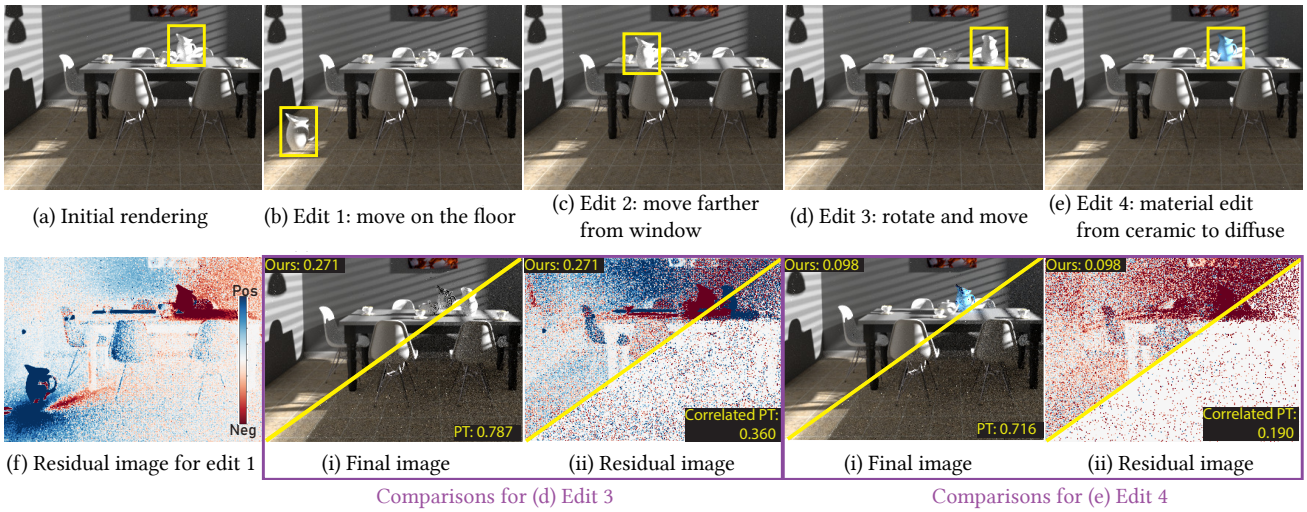
Eurographics Symposium on Rendering 2024
E Garces and E. Haines
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 4

Residual path integrals for re-rendering

Bing Xu[1] Tzu-Mao Li[1] Iliyan Georgiev[2] Trevor Hedstrom[1] Ravi Ramamoorthi[1]



(a) Initial rendering (b) Edit 1: move on the floor (c) Edit 2: move farther from window (d) Edit 3: rotate and move (e) Edit 4: material edit from ceramic to diffuse

(f) Residual image for edit 1 (i) Final image (ii) Residual image (i) Final image (ii) Residual image

Comparisons for (d) Edit 3 Comparisons for (e) Edit 4

Figure 1: We consider the problem of efficiently rendering versions of a given an initial rendering (a), with object movement or material editing (b-e). We devise sampling techniques to focus on the light-transport paths affected by the scene changes and a method for incremental re-rendering faster than path tracing from scratch (PT) or computing correlated-path differences (Correlated PT). Our approach renders a *residual image* (f) between the original and the edited scene. We show equal-time comparisons with baselines for edits 3 and 4, including final images for the edited scenes and residual images. Numbers shown are MSE$\times 10^3$. More equal-time comparisons are shown in Figs. 6 to 8.

Abstract

Conventional rendering techniques are primarily designed and optimized for single-frame rendering. In practical applications, such as scene editing and animation rendering, users frequently encounter scenes where only a small portion is modified between consecutive frames. In this paper, we develop a novel approach to incremental re-rendering of scenes with dynamic objects, where only a small part of a scene moves from one frame to the next. We formulate the difference (or residual) in the image between two frames as a (correlated) light-transport integral which we call the residual path integral. Efficient numerical solution of this integral then involves (1) devising importance sampling strategies to focus on paths with non-zero residual-transport contributions and (2) choosing appropriate mappings between the native path spaces of the two frames. We introduce a set of path importance sampling strategies that trace from the moving object(s) which are the sources of residual energy. We explore path mapping strategies that generalize those from gradient-domain path tracing to our importance sampling techniques specially for dynamic scenes. Additionally, our formulation can be applied to material editing as a simpler special case. We demonstrate speed-ups over previous correlated sampling of path differences and over rendering the new frame independently. Our formulation brings new insights into the re-rendering problem and paves the way for devising new types of sampling techniques and path mappings with different trade-offs.

1. Introduction

It is common practice to render the frames of an animation sequence independently, which can be expensive. There is considerable coherence from one frame to the next, for example when only the main character is moving around a complex open world. Another scenario is scene authoring or gaming applications where one may edit or move the scene setup only slightly at a time (Fig. 1). However, exploiting this coherence is not as simple as focusing attention only on the pixels corresponding to dynamic objects, since moving objects may affect the shading in other parts of the scene through shadows and global illumination.

In this paper, we develop a method for efficient incremental

(re-)rendering of such scenes with moving objects (Fig. 1), which can also be applied to material editing as a simpler special case. We assume the previous scene state has been already rendered accurately, and our goal is to render the difference, or the residual, between the two states. To consider only the light paths that contribute to the difference, we formulate a *residual path integral*. This integral characterizes the difference in light transport between the old and the new scene. It opens up a plethora of importance sampling techniques that we can use to efficiently re-render the scene.

Concretely, we make the following contributions:

1. We extend the classical path integral [Vea98] to a *residual path integral* to account for differences before and after scene changes using control variates.
2. We devise importance sampling techniques for spawning paths with non-zero residual contribution in the path space, where at least one vertex or edge is affected by the scene changes.
3. We construct "path mappings"[†], akin to shift mapping in gradient-domain rendering [LKL*13], to form bijections between the path spaces of the old and new frames. These mappings are tailored to our importance sampling techniques.

2. Related work

Temporal reprojection. A number of previous methods reproject light paths or color in their temporal trajectories to facilitate image reconstruction [BFMZ94, WDP99, BDT99, TPWG02, NSL*07]. These ideas are also often incorporated in Monte Carlo denoising [MA06, ZRJ*15, SKW*17, CKS*17]. We focus on unbiased rendering of the new scene state while exploiting the coherence between frames using importance sampling.

Spatio-directional caching, path guiding, and resampling. Some methods do not reproject the light paths, but instead reuse and adapt the statistics of spatio-directional caches over time. These caches can then be used for importance sampling [LW95, VKS*14, RHJD18] or as a direct approximation of the radiance [MRKN20, MRNK21]. Furthermore, ReSTIR [BWP*20] and its variants [OLK*21, LKB*22, KLR*23] reuse samples from previous frames and spatial neighbors through resampled importance resampling [TCE05].

We focus on directing the light-path generation using geometry information, so that most sampling efforts can be spent on the differences between frames. Reprojection, denoising, spatio-directional caches, and resampling can then be potentially applied on top of our method to further improve the results.

Gradient-domain rendering. Our residual path integral is highly related to gradient domain rendering [LKL*13, KMA*15, HGP*19a] and particularly its temporal variant [MKD*16]. Both our method and gradient-domain rendering address the problem of computing the difference between the colors of two pixels. The key difference lies in how we characterize dynamic objects in the path space. Temporal gradient-domain rendering correlates the samples for rendering the two frames by reusing the same random-number sequence. Instead, we explicitly model the difference of the two path spaces and design importance sampling strategies and path mappings to direct sampling towards the part of path space that is affected by the scene change. We compare to the baseline of correlated path tracing and show reduction in variance due to our importance sampling strategies. In principle, one can apply a spatio-temporal Poisson reconstruction to obtain the final image/video for our method just like temporal gradient domain rendering; we leave it as future work.

Incremental radiosity. Earlier work in radiosity explores incremental updates of form factors between elements [Che90, FYT95, DS97, GD01, MPT03]. We explore related ideas in Monte Carlo path tracing.

Dynamic photon mapping and virtual point lights. Some methods reuse photon maps [Jen96] or virtual point lights [Kel97] in a temporally coherent manner [DBMS02, LSK*07, WG12]. We focus on the path tracing regime, and the combination with these methods is an interesting future direction.

Scene editing and re-rendering. Our problem setting is related to previous work on efficient scene re-rendering [GG15]. In particular, control variates and correlated sampling have been applied to re-rendering previously [RJN16]. The former relied on simple heuristics to reconstruct the final image and generate biased results. We use the same high-level control variate framework as Rousselle et al. [RJN16]. They analyzed a multi-level Monte Carlo method, focusing on finding the optimal parameters to combine two MC estimators (primal and residual rendering) and use path tracing with the same random sequence to compute the residual image. This is equivalent to the correlated path tracing we compared as a baseline. Their work supports material editing but not object movement. We instead focus on formulating and rendering of the difference image in an unbiased fashion, enabling object movement. The composition with the standard MC estimator and optimizing the parameters of the external multi-level Monte Carlo control variates are orthogonal to our work. We mainly target at the unsolved challenging scenario involving moving objects, while gaining significant advantage for material editing. Our contribution is the characterization of the light paths that interact with a dynamic object, the importance sampling strategies for sampling such light paths, and the path mappings for finding correspondence between the two path spaces. In the results, we compare to correlated path tracing and show variance reduction.

Portal sampling. Our importance sampling strategies share similarities to the *portal sampling* techniques [BNJ15, ALLD17, OHD20] used for sampling sparse path space (e.g., lights that go through a pinhole of a scene). We apply portal sampling to the problem of incremental rendering using the residual path integral formulation. The existence of two path spaces introduces new challenges and opportunities in designing new sampling strategies.

3. Residual path integral

In this section we formulate our residual path integral. We first revisit the classical path integral (Section 3.1) which we then extend to

[†] The term "shift mapping" is extensively used in both gradient-domain rendering and ReSTIR literature. We refrain from using the same term, as the mapped path in our method is not generated by shifting the target pixel.

Table 1: A list of commonly used symbols in this document.

Symbol	Description
x	path vertex in the path integral formulation
x_D, x_G, x_S	dynamic vertex, ghost vertex, static vertex in the path representation
x_E, x_L	sensor and emitter vertices
$\bar{x} = x_0 \dots x_{k-1}$	full, path with k vertices
$\bar{y} = y_0 \dots y_{s-1}$	emitter sub-path, with the last vertex y_{s-1} on an emitter. Note the direction of the path here is opposite to BDPT convention.
$\bar{z} = z_0 \dots z_{t-1}$	sensor sub-path, with the last vertex z_{t-1} on a sensor.
$\overrightarrow{p_i}, \overleftarrow{p_i}$	forward and reverse area PDFs for sub-path vertex i
E, L	sensor and emitter vertices in path regular expression grammar
$\mathbb{S}, \mathbb{D}$	static and a dynamic vertices in regular expression
$\bar{\mathbb{S}}$	static path edge, not passing through ghost objects
$\bar{\mathbb{D}}, \bar{\mathbb{D}}_n$	dynamic path edge, passing through ghost objects n times; subscript n may be dropped for convenience
$\boldsymbol{p}, \boldsymbol{q} = T(\boldsymbol{p})$	base and transformed paths in the other frame.
f_d	difference radiance contribution function (Eq. (5))

a residual formulation via control variates (Section 3.2). This extension is designed to handle differences introduced by scene changes (Section 3.3) and helps characterize the paths that are affected by those changes (Section 3.4).

In Section 4 we will devise sampling techniques that spawn paths with likely non-zero contributions in the residual path integral. In Section 5, we describe "path mappings" that form bijections between the old and new path spaces for efficient control variates. Table 1 lists some commonly used notations throughout the paper.

3.1. Primal path integral

Following Veach's [Vea98] path integral formulation, the intensity of each pixel in image I can be written as[‡]

$$I = \int_\Omega f(\boldsymbol{p}) d\mu(\boldsymbol{p}), \tag{1}$$

where $d\mu(\boldsymbol{p})$ is a measure over the space Ω of light-transport paths $\boldsymbol{p} = x_0 x_1 \dots x_{k-2} x_{k-1}$ with k vertices. The contribution of a path is

$$f(\boldsymbol{p}) = L(x_{k-2} \to x_{k-1}) \cdot \prod_{i=1}^{k-2} \rho(x_{i-1} \to x_i \to x_{i+1}) G(x_i \leftrightarrow x_{i+1}) V(x_i \leftrightarrow x_{i+1}), \tag{2}$$

where L is the light emission, ρ is the bidirectional scattering distribution function (BSDF), G is the geometric term, and V is the visibility function [Vea98].

3.2. The residual path integral

We assume we are given the image at the previous frame, I_1 (without auxiliary information or any cache), and we seek to compute the current frame I_2. We are also given the geometric and material configurations of both frames, which differ only slightly, because only some object(s) have moved or changed material.

To enable incremental rendering, we seek to render the difference, or residual, $I_2 - I_1$ between the two frames, effectively treating the first image as a control variate:

$$I = I_1 + (I_2 - I_1). \tag{3}$$

We now need to compute the residual

$$I_2 - I_1 = \int_{\Omega_2} f_2(\boldsymbol{p}) d\mu(\boldsymbol{p}) - \int_{\Omega_1} f_1(\boldsymbol{q}) d\mu(\boldsymbol{q}), \tag{4}$$

where we have simply subtracted out path integrals for frames I_2 and I_1. The path space may be different between the two frames.

To evaluate the result as one integral, we seek to find a mapping between paths $\boldsymbol{p}$ in frame I_2 and paths $\boldsymbol{q}$ in frame I_1, writing $\boldsymbol{q} = T(\boldsymbol{p})$. We will discuss the form of the mapping T we use in Section 5, but the theory below applies to any mapping. Now, T can be seen as a change of variable that brings the integral I_1 into the same domain and parameterization as I_2:

$$\begin{aligned} I_2 - I_1 &= \int_\Omega \left(f_2(\boldsymbol{p}) - f_1(T(\boldsymbol{p})) \left| T'(\boldsymbol{p}) \right| \right) d\mu(\boldsymbol{p}) \\ &= \int_\Omega f_\mathrm{d}(\boldsymbol{p}) \, d\mu(\boldsymbol{p}), \end{aligned} \tag{5}$$

where $\left| T' \right|$ accounts for the change of measure. We have dropped the subscript on Ω_2 for clarity. However, we retain the subscripts on f_2 and f_1 since the path contribution must be evaluated with respect to each frame's geometry which can differ even for the same path, on account of visibility changes. We can also view our problem as simulating a *difference radiance* quantity and Eq. (5) as integrating a *difference path contribution* $f_\mathrm{d}(\boldsymbol{p})$.

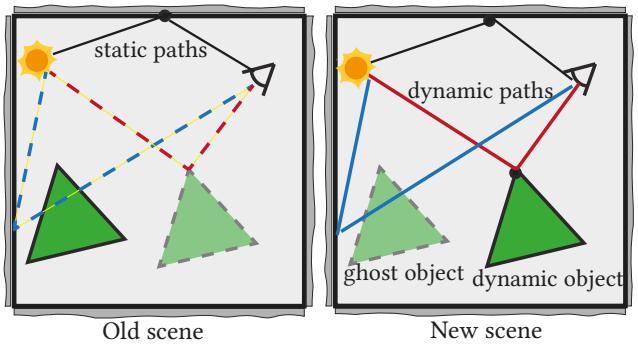


Figure 2: **Examples of dynamic paths.** We show three-vertex paths involving a moving triangle. In each scene, the location of the triangle in the other scene is in light green. In the new scene, we can see two types of *"dynamic" paths*. One hits the dynamic triangle (path in red) while the other one passes through the ghost triangle (in blue; as per the position of the dynamic counterpart in the old scene). The red and blue paths did not exist in the previous scene (thus shown as dashed lines), so they would contribute to the difference integral. The black paths represent *static paths* that do not contribute to the difference path integral.

Ghost objects. Equation (5) reveals a key difference between the classical and residual path integrals: the contribution of a single path $\boldsymbol{p}$ in frame I_2 needs to be evaluated at two different scenes: f_1 and f_2. This means that we need to account for paths that interact with the dynamic object in the new scene f_2 (red path in Fig. 2), and also

[‡] We omit the pixel filter for simplicity.

paths that interact with where the dynamic object was at in the old scene f_1 (blue path in Fig. 2). We use "ghost object(s)" to represent where the dynamic objects used to be, which can be an empty space in the current frame or (partly) overlapped by the dynamic object.

3.3. Characterizing dynamic paths

We refer to paths that interact with both the dynamic and ghost objects as *dynamic paths*.[§] A path $\boldsymbol{p}$ is dynamic if it meets either of the following criteria. It contains

1. at least one *dynamic vertex*, i.e., a vertex located on a dynamic object (red path in Fig. 2); or
2. at least one *dynamic edge*, i.e., an edge passing through ghost surfaces (blue path in Fig. 2).

Otherwise, the path is *static*, composed of *static vertices* and *static edges* (black path in Fig. 2).

We extend Heckbert's regular expression grammar for paths [Hec90] by explicitly including an edge notation (Table 1). We use $\mathbb{S}$ and $\mathbb{D}$ to represent static and dynamic vertices, respectively (which originally denote specular and diffuse), and $\bar{\mathbb{S}}$ and $\bar{\mathbb{D}}$ for static and dynamic *edges*, respectively.

Starting from the eye E and ending at the light L, dynamic paths corresponding to the two conditions above are denoted as

1. $E[(\bar{\mathbb{S}}|\bar{\mathbb{D}})(\mathbb{S}|\mathbb{D})]^*(\bar{\mathbb{S}}|\bar{\mathbb{D}})\mathbb{D}(\bar{\mathbb{S}}|\bar{\mathbb{D}})[(\mathbb{S}|\mathbb{D})(\bar{\mathbb{S}}|\bar{\mathbb{D}})]^*L$,
2. $E[(\bar{\mathbb{S}}|\bar{\mathbb{D}})(\mathbb{S}|\mathbb{D})]^*\bar{\mathbb{D}}[(\mathbb{S}|\mathbb{D})(\bar{\mathbb{S}}|\bar{\mathbb{D}})]^*L$.

In condition (1), the $\mathbb{D}$ in the center is a dynamic vertex; the rest of the path can include static/dynamic vertices and edges. In condition (2), the $\bar{\mathbb{D}}$ in the middle is a dynamic edge; the rest of the path can have arbitrary vertices/edges.

3.4. Dynamic path space

Assuming small changes between the two frames, most paths in the residual integral in Eq. (5) are static and have zero contribution. Simulating those paths would be a waste of computation. We therefore isolate all dynamic paths into a *dynamic path space* $\mathcal{D}$, while $\mathcal{S}$ is the zero-contribution *static path space*, with $\Omega = \mathcal{S} \cup \mathcal{D}$:

$$I_2 - I_1 = \int_{\mathcal{S}} 0 \, d\mu(\boldsymbol{p}) + \int_{\mathcal{D}} f_\mathrm{d}(\boldsymbol{p}) \, d\mu(\boldsymbol{p}) \,. \tag{6}$$

Our goal is then to devise sampling techniques that can generate paths within $\mathcal{D}$, i.e., paths impacted by the dynamic or ghost objects. However, $\mathcal{D}$ forms a complex sub-space, and sampling paths in it is challenging.

4. Sampling the dynamic path space

Equation (6) enables us to separate out the dynamic-path subspace affected by the scene movement. In this section, we present a set of techniques to sample paths in that space.

Recall that a path could contribute to the residual integral if it has a dynamic vertex or a dynamic edge (namely, an edge passing through ghost surfaces). Contrary to traditional path construction starting from an emitter or the sensor, our key idea to guarantee sampling dynamic paths is to start from dynamic (Section 4.1) and ghost (Section 4.2) objects towards the emitters and the sensor. We also derive their corresponding density functions which will allow us to combine diverse and non-exclusive techniques via multiple importance sampling (MIS).

Notations. We generally follow the notations used by Veach [Vea98] and Georgiev et al. [GKDS12] (see Table 1). After sampling a dynamic/ghost starting point, two subpaths are built, one ending on an emitter and the other on the sensor. We denote an emitter sub-path with s vertices by $\bar{y} = y_0 y_1 \dots y_{s-1}$ and a sensor sub-path with t vertices by $\bar{z} = z_0 z_1 \dots z_{t-1}$. For convenience, the vertices are indexed in the order of their generation. We give y_0 and z_0 different aliases depending on their locations:

- When they are sampled on a dynamic object, $y_0 = z_0 = x_D$;
- When they are sampled on a ghost object, $y_0 = z_0 = x_G$.

y_{s-1} and z_{t-1} are on an emitter and the sensor, respectively. When they are the only vertex of their corresponding sub-paths, we denote them as x_L and x_E, respectively. The derivations below consider emitter sub-paths $\bar{y}$, but they apply symmetrically to sensor sub-paths too.

Following Georgiev et al. [GKDS12], we use $\overrightarrow{p_i}$ for *forward* vertex probabilities (w.r.t. the random-walk direction). p_0 is the probability density function (PDF) of the initial path vertex (on a dynamic or ghost object). The PDF of a sub-path $\bar{y}$ with s vertices is

$$p_s(\bar{y}) = p_0 \prod_{i=1}^{s-1} \overrightarrow{p_i}(\bar{y}). \tag{7}$$

The reverse PDF $\overleftarrow{p_i}(\bar{y})$ is analogous to the forward one. It represents the probability for sampling y_i in direction opposite to that of the random walk. We will need the reverse PDFs to compute the MIS weight for each dynamic path. We always start the dynamic path by sampling the initial vertex on a dynamic/ghost surface. We sample emitters as in regular next-event estimation and the sensor as in light tracing, again w.r.t. area.

4.1. Sampling dynamic vertices

We introduce three path sampling techniques that start from a point on the dynamic surface.

(1) Dynamic from emitter. (Fig. 3A1, Fig. 4a1) The first vertex x_D is sampled on a dynamic surface. We sample the second path vertex x_L on an emitter. Sampling fails if the connection is blocked. At x_D we initiate a random walk towards the sensor using BSDF sampling. At each sampled vertex, we directly connect to the sensor as in light tracing. The paths generated from this techniques are in the form of $E(\bar{\mathbb{S}}|\bar{\mathbb{D}})[(\mathbb{S}|\mathbb{D})(\bar{\mathbb{S}}|\bar{\mathbb{D}})]^*\mathbb{D}(\bar{\mathbb{D}}|\bar{\mathbb{S}})L$. The path density is

$$p(\bar{x}) = p(x_L)p(x_D) \prod_{i=1}^{t-1} \overrightarrow{p_i}(\bar{z}). \tag{8}$$

[§] Dynamic paths refer to all paths affected by either object movement or material editing.

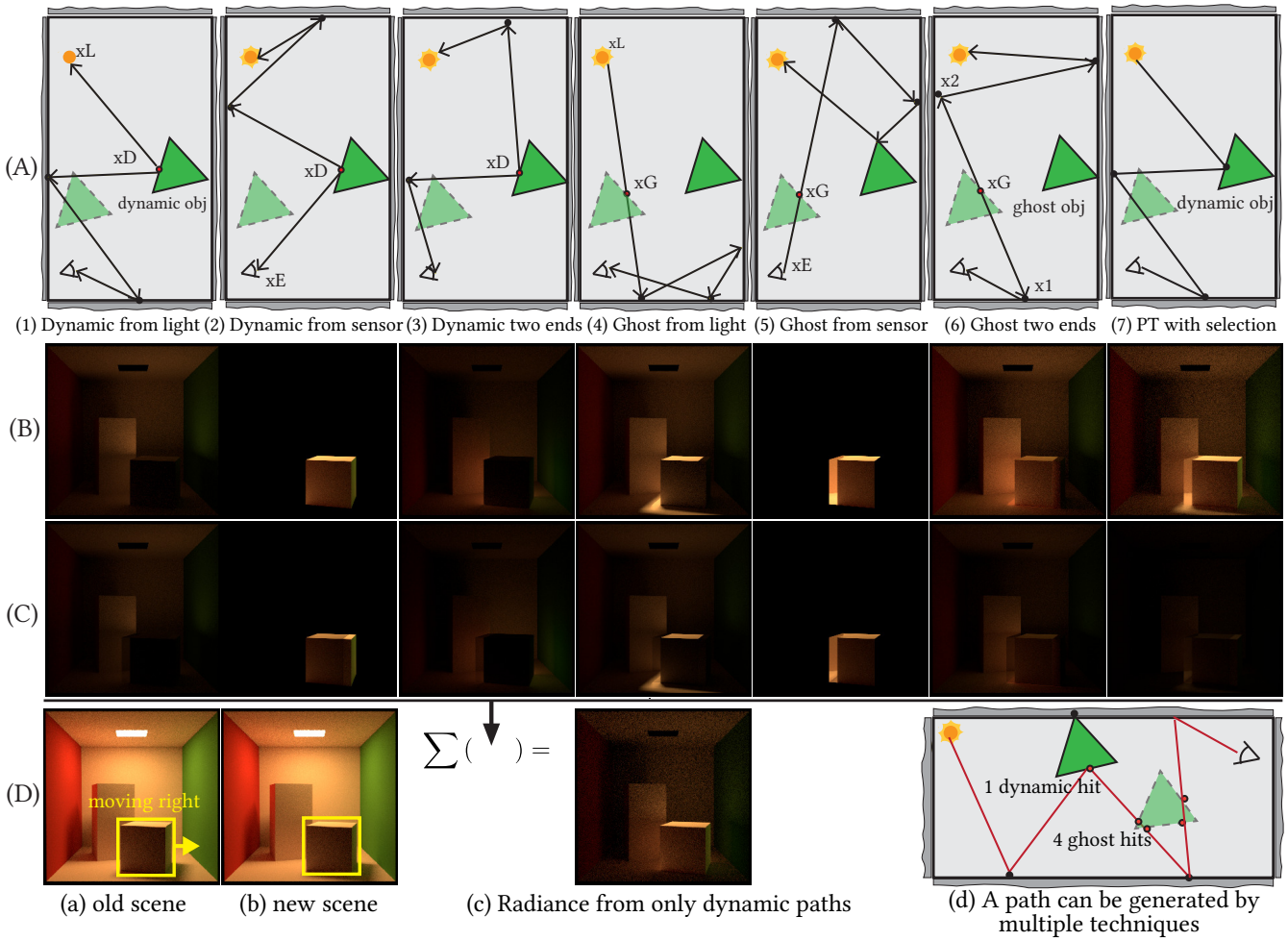


Figure 3: **Dynamic-path sampling techniques.** Row A illustrates six techniques plus the regular path tracing with non-affected paths rejected (Section 4). We move the small box slightly to the right in the CORNELL BOX scene (a, b). We render the difference radiance contribution from each sampling technique separately in Row B. Row C is Row B weighted by the corresponding multiple importance sampling weight for each column. A sum of Row C gives the total contribution by dynamic paths in (c). (d) An example for multiple importance sampling computation in Section 4.3, where a dynamic path hits dynamic objects once and ghost objects 4 times.

(2) Dynamic from sensor. (Fig. 3A2, Fig. 4a2) This is symmetric to technique (1), swapping emitter and sensor. Paths generated from this technique have the form $E(\bar{\mathbb{D}}|\bar{\mathbb{S}})D[(\bar{\mathbb{S}}|\bar{\mathbb{D}})(\mathbb{S}|\mathbb{D})]^*(\bar{\mathbb{S}}|\bar{\mathbb{D}})L$. The path density is

$$p(\bar{x}) = p(x_E)p(x_D)\prod_{i=1}^{s-1}\overrightarrow{p_i}(\bar{y}). \tag{9}$$

Techniques (1) and (2) become identical for the case of direct lighting (i.e., 3-vertex paths). We handle direct lighting in technique (1), connecting the dynamic vertex to both emitter and sensor.

(3) Dynamic two ends. (Fig. 3A3, Fig. 4a3) The vertex x_D is again sampled on a dynamic surface. We then sample a direction towards an emitter using cosine hemisphere sampling. Given that direction, another direction, towards a sensor, is sampled using BSDF sampling. Finally, we initiate random walks along these two directions. Connections to emitter/sensor are performed at every sampled vertex. This process yields emitter and sensor subpaths, akin to bidirectional path tracing. The path expression is $E[(\bar{\mathbb{S}}|\bar{\mathbb{D}})(\mathbb{S}|\mathbb{D})]^+(\bar{\mathbb{S}}|\bar{\mathbb{D}})\mathbb{D}(\bar{\mathbb{S}}|\bar{\mathbb{D}})[(\mathbb{S}|\mathbb{D})(\bar{\mathbb{S}}|\bar{\mathbb{D}})]^+L$. The path density is

$$p(\bar{x}) = \prod_{i=1}^{t-1}\overrightarrow{p_i}(\bar{z})p(x_D)\prod_{i=1}^{s-1}\overrightarrow{p_i}(\bar{y}). \tag{10}$$

A path generated with the above techniques can hit a dynamic or ghost object again, meaning that the path can also be sampled from other techniques or multiple starting points by the same technique (see Fig. 3 bottom right). We discuss the details of handling the MIS weights correctly in Section 4.3.

4.2. Sampling dynamic edges

Since ghost objects do not physically exist in the new frame, paths pass through them. We propose a mechanism to sample dynamic edges by sampling points on ghost surfaces.

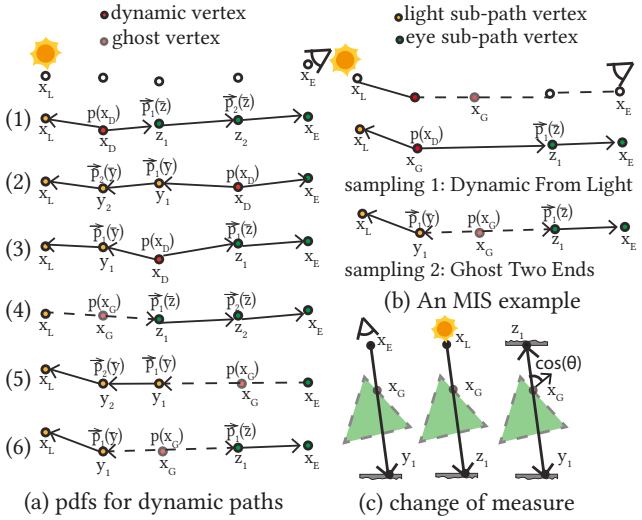


Figure 4: (a) Illustration of pdf construction for our set of dynamic path sampling strategies. Indices match Fig. 3. (b) An MIS example in terms of pdfs showing the same path can be sampled by two techniques. (c) Geometries used for change of measure when computing pdfs for dynamic paths starting from a ghost vertex. We represent dynamic edges passing through ghost objects by dashed segments.

(4) Ghost from emitter. (Fig. 3A4, Fig. 4a4) Analogously to technique (1) (Section 4.1), we begin by sampling a point x_G on a ghost surface using area sampling, followed by a point x_L on an emitter. The point x_G does *not* act as a path vertex but determines the direction $\overrightarrow{x_L x_G}$ of a ray initiating a random walk from the emitter vertex x_L. The ray will go through the ghost surface as desired. We connect each random-walk vertex to the sensor as before. Paths generated with this technique have the form $E(\bar{\mathbb{S}}|\bar{\mathbb{D}})[(\mathbb{S}|\mathbb{D})(\bar{\mathbb{S}}|\bar{\mathbb{D}})]^*\bar{\mathbb{D}}L$.

The path PDF of this light-tracing technique has the familiar form

$$p(\bar{x}) = p(x_L)\prod_{i=1}^{t-1}\overrightarrow{p_i}(\bar{z}), \tag{11}$$

with the peculiarity that the direction towards the second path vertex x_1 is sampled via an auxiliary surface point x_G. To obtain the area-PDF of x_1, we first convert the area-PDF $p(x_G)$ to solid angle measure at x_L and then convert that to area measure at x_1:

$$\overrightarrow{p_1}(\bar{z}) = p(x_G)\frac{\| x_G - x_L \|^2}{\left|n_{x_G}\cdot\overrightarrow{x_G x_1}\right|}\cdot\frac{\left|n_{x_1}\cdot\overleftarrow{x_G x_1}\right|}{\| x_1 - x_L \|^2}, \tag{12}$$

where n_x denotes surface normal at point x.

(5) Ghost from sensor. (Fig. 3A5, Fig. 4a5) This technique is symmetric to (4) by switching the light source and sensor. Generated paths have the form $E\bar{\mathbb{D}}[(\mathbb{S}|\mathbb{D})(\bar{\mathbb{S}}|\bar{\mathbb{D}})]^*(\bar{\mathbb{S}}|\bar{\mathbb{D}})L$. The path PDF reads

$$p(\bar{x}) = p(x_E)\prod_{i=1}^{s-1}\overrightarrow{p_i}(\bar{y}), \tag{13}$$

where $\overrightarrow{p_1}(\bar{z})$ is the same as in Eq. (12) but with x_L replaced with x_E.

(6) Ghost two ends. (Fig. 3A6, Fig. 4a6) After sampling the initial ghost vertex x_G, we randomly sample a direction uniformly toward the light, with PDF p_{dir}. This gives us a dynamic edge which passes through x_G. By casting two rays with the same origin x_G but opposite directions, we obtain two vertices of this edge, x_1 and x_2. We then incrementally construct the two sub-paths accordingly using unidirectional path tracing. Again, each intersection is BSDF-sampled along with next event estimation. The paths generated from this technique have the form $E[(\bar{\mathbb{S}}|\bar{\mathbb{D}})(\mathbb{S}|\mathbb{D})]^+\bar{\mathbb{D}}[(\mathbb{S}|\mathbb{D})(\bar{\mathbb{S}}|\bar{\mathbb{D}})]^+L$. The geometric term transforming the density of the sampled ghost vertex to the product of densities of x_1 and x_2 (Fig. 4b), and the path PDF, are

$$g_{G,x1,x2} = \frac{\left|\cos(N_{x_1},\overleftarrow{x_1x_2})\right|\cdot\left|\cos(N_{x_2},\overleftarrow{x_1x_2})\right|}{\left|\cos(N_{x_G},\overleftarrow{x_1x_2})\right|}\cdot\frac{p_{\text{dir}}}{\| x_1 - x_2 \|^2}, \tag{14}$$

$$p(\bar{x}) = \prod_{i=1}^{t-1}\overrightarrow{p_i}(\bar{z})\prod_{i=1}^{s-1}\overrightarrow{p_i}(\bar{y}),\ \text{where}\ \overrightarrow{p_1}(\bar{z})\overrightarrow{p_1}(\bar{y}) = p(x_G)g_{G,x1,x2}. \tag{15}$$

The effectiveness of each technique is scene-dependent. For instance, when direct lighting reflected by dynamic objects is significant, directly connecting to the emitter is beneficial. If dynamic objects are not easily reachable by the emitter or sensor, techniques (3) and (6) are more effective.

4.3. Technique combination

In most cases, there may be more than one sampling technique capable of generating one specific dynamic path. Figure 3d shows an example where a path hits the dynamic object once and passes through a ghost surface four times. The path has the form $E\bar{\mathbb{S}}\bar{\mathbb{S}}\bar{\mathbb{D}}_2\mathbb{S}\bar{\mathbb{D}}_2\mathbb{D}\bar{\mathbb{S}}\bar{\mathbb{S}}\bar{\mathbb{S}}L$ (the subscript means 2 intersections with ghost objects for the edge). Regardless which vertex is sampled first, there are 5 techniques capable of sampling this path. More specifically, sampling may start from the ghost point next to the sensor using technique (5) (Section 4.2), from any of the other 3 ghost points using technique (5), or from the dynamic vertex next to the light using technique (1) (Section 4.1). Another example is shown in Fig. 4c. In effect, if a path containing N_k vertices intersects the dynamic objects N_D times, and its edges pass through ghost objects N_G times, we need to correctly weight $N_D + N_G$ different sampling techniques to produce an unbiased result. We apply multiple importance sampling (MIS) to combine them, based on their derived path densities.

Finally, we employ regular path tracing with static paths rejected as another sampling technique to complement the entire set (Fig. 3A7). Figure 3B,C shows the rendering of various techniques before and after the MIS weighting:

$$w(\bar{x}) = \frac{p(\bar{x})}{\sum_{l=1}^{N_D+N_G+1}p(\bar{x}_l)}. \tag{16}$$

To facilitate the weight computation, we record some information of the ghost intersections for every path segment (normal, position, shape ID). Given the assumption that the scale of moving objects is small compared to the whole scene, this overhead is negligible.

Similarly to prior implementations of bidirectional methods [Geo12, Jak10], we record forward and backward PDFs at every vertex for emitter/sensor sub-paths. Instead of one path traversal for each of them, we preprocess the path and build two tables: one for cumulative products of forward PDFs and another for backward PDFs. To compute the PDF for any path segment, we just query

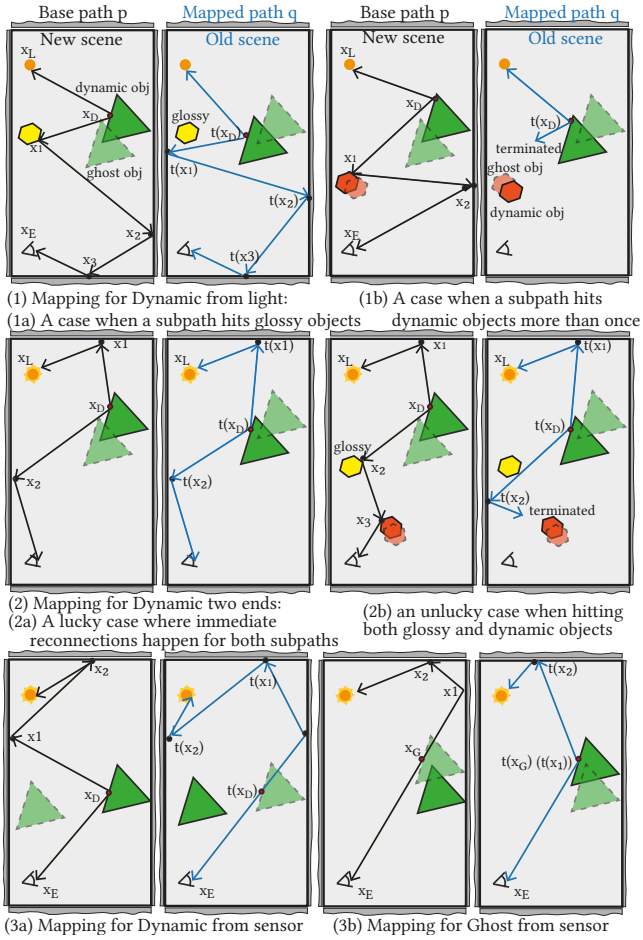


Figure 5: **Path mappings for various dynamic sampling techniques.** On the left of each pair, we sample the base path (black) using our methods shown in Fig. 3. Mapped paths are blue on the right. Two frames can be flipped for the two-way path mapping.

the tables at two indices and perform a division, leading to time complexity reduction from $O((N_D+N_G)*N_k)$ to $O(N_D+N_G)$.

5. Path mapping

Recall from Section 3.2 that we seek to find a correspondence between paths in two frames (Eqs. (5) and (6)). In Section 4, we presented our sampling techniques to sample paths in the dynamic path space, which are affected by the movement or material edits. Once we have sampled such a path $\boldsymbol{p}$ in one frame, we need to map it to a path $T(\boldsymbol{p})$ in the other frame, to compute the residual contribution (Eq. (5)). We extend the shift-mapping operators used in prior work (*Random-seed replay*, *Keep vertex the same*) [KMA*15, HGP*19b, LKB*22] to dynamic scenes. Additionally, we present new operators tailored to our dynamic path sampling techniques.

We build the path mapping T by applying a transformation t to each vertex:

$$T(\boldsymbol{p}) = T(x_0x_1\dots x_{k-2}x_{k-1}) = t(x_0)t(x_1)\dots t(x_{k-2})t(x_{k-1}). \quad (17)$$

The vertices x_0 and x_{k-1} lie respectively on the sensor and emitter.

Figure 5 illustrates how path mapping is applied to our techniques. Below we present the five basic vertex transformation operators t we use and provide their Jacobians $|t'|$. We choose the mapping operators according to the status of the vertex, such as the path sampling technique used and the mapping history of its predecessors. Below we consider the individual cases. Please refer to Appendix A for mappings of the complete paths.

Transform with object movement. We support rigid transformations t_{obj} of objects. If x_i is on a dynamic/ghost surface in frame I_2, we apply T_{obj} to bring it to its (canonical) position in frame I_1:

$$t(x_i) = T_{obj}(x_i) \quad \text{if} \quad x_i \in \{x_D, x_G\}. \quad (18)$$

For rigid motions of articulated bodies, $|t'| = 1$. For example, in Fig. 5(1a), x_D is mapped to $t(x_D)$ along with the moving triangle. Whenever a base path (Fig. 5) starts from a dynamic object or hits a dynamic object, the two paths diverge in two frames – they do not share the same path vertices in the rest of the paths. When applying this operator multiple times after the paths already diverge, significant deviations can occur, especially with large movements. Hence, we *reject* the mapped path when the base path hits dynamic objects more than once, as shown in Fig. 5 (1b). The detail of the rejection will be discussed below.

Keep vertex the same. If the vertex x_i lies on a static object, we set t to the identity function:

$$t(x_i) = x_i \quad \text{if} \quad x_i \in \{\mathbf{x}_S\}, \quad (19)$$

$$|t'| = 1 \quad \text{if} \quad t(x_{i-1}) = x_{i-1}. \quad (20)$$

This enables path re-connection, where paths that deviated at x_{i-1} can re-connect at x_i, creating more correlated paths $\boldsymbol{p}$ and $T(\boldsymbol{p})$, reducing variance in the residual contribution. In Fig. 5(2a), $t(x_1)$ is formed by reconnecting x_D to x_1, effectively reusing x_1 as $t(x_1)$.

If the previous vertex x_{i-1} was mapped to a different location in the previous frame (and hence directions $x_{i-1} \to x_i$ are not the same). The Jacobian $|t'|$ is no longer 1 as we need to account for the change of projected solid angle:

$$|t'| = \frac{\cos(t(x_{i-1}) \leftarrow x_i)}{\cos(x_{i-1} \leftarrow x_i)} \cdot \frac{\|x_{i-1} - x_i\|^2}{\|t(x_{i-1}) - x_i\|^2}. \quad (21)$$

We avoid multiple (re)-connections for one path to prevent a potentially large Jacobian, which is a multiplication of all the corresponding Jacobians. We add an additional re-connection criterion (1) to the two conditions (2,3) introduced by Lin et al. [LKB*22]:

1. x_i is not a dynamic vertex;
2. $\min(\alpha(x_{i-1}), \alpha(t(x_{i-1})), \alpha(x_i)) \geq \alpha_{min}$ (roughness threshold);
3. $\min(\|t(x_{i-1}) - x_i\|, \|x_{i-1} - x_i\|) \geq d_{min}$ (distance threshold).

Random-seed replay. This operator refers to using the same random numbers to generate ray directions ($\overrightarrow{x_{i-1}x_i}$ and $\overrightarrow{t(x_{i-1})t(x_i)}$). We use this approach to do independent tracing for diverged paths until they are merged by path re-connection.

Mapping between the dynamic and ghost pair. As shown in Fig. 5(3a,3b), we map the starting point x_D to the same location on its ghost counterpart in the other frame. This operator is exclusive to

techniques *Dynamic from sensor* and *Ghost from sensor*. The effect is the same as tracing the camera rays towards the same direction in both frames. Unlike the same primary ray as in correlated path, our paths start from the middle.

Path rejection. In cases of path mapping failure or proactive termination, we reject the mapped path and continue tracing the base path. Similarly to symmetric gradient computation used in gradient-domain rendering [MRK*14], we sample from both frames and employ two-way path mapping to ensure the coverage of the entire integral domain. This is crucial to maintain unbiasedness. When the rejection operator is used, paths revert to independent tracing for both frames, reducing Eq. (5) to finite differences. Cases where we terminate the mapped path are:

1. The mapped path may be occluded during reconnection or even afterward by a dynamic object.
2. The Jacobian exceeds a threshold, potentially causing fireflies.
3. The base path hits dynamic objects more than once.

Conditions 1 and 3 occur when paths repeatedly hit dynamic objects, which have limited impact if only a small portion of the scene is moving. Rejection happens more frequently when dynamic objects occupy a larger (Table 3) or a significant part of the path space, such as being close to the camera. The Jacobian threshold is set to 10. Condition 2 is rarely met, as most cases are already covered by other conditions. This threshold serves as a final safety net while maintaining unbiasedness.

6. Results and discussion

In this section, we show our results. Following a discussion of implementation details, we demonstrate the benefits brought by our incremental rendering approach, and discuss timings and overhead.

6.1. Implementation

We implemented our approach in a stand-alone C++ renderer on top of Intel Embree. Our experiments are run on an AMD Ryzen 16-Core Processor. The ghost objects are implemented by modifying specific Embree ray intersection behaviors. We maintain two extra acceleration structures: one containing all the dynamic objects and the other containing all the ghost objects. The former is used for the visibility re-check for dynamic objects during path mapping mentioned in Section 5; the other is for updating the path PDFs when computing the multiple importance sampling weights (Section 4.3). Based on the assumption that only a small part of the scene is moving, this adds a very small overhead.

The dynamic path sampling techniques we introduced in Section 4 are the most fundamental building block for our approach. We separately display the radiance contribution from each dynamic path sampling technique in Fig. 3. We have verified the unbiasedness with images ultimately converging to the reference; the plot with increasing number of samples per pixel is shown in the supplemental material with expected slope.

6.2. Incremental re-rendering

Baseline methods. We compare to two baselines. 1) *Path tracing (PT)*: Rendering the new scene from scratch. To ensure a fair comparison, we only consider scenes where the moving objects are directly visible to the camera. In such cases, path tracing represents the most competitive and efficient light transport method. Our approach consistently outperforms path tracing, especially in scenarios when the probability of hitting the dynamic objects is low using paths traced from camera; 2) *Correlated PT*: Rendering the residual image using correlated path tracing and adding it back to the ready-to-use old scene rendering (Eq. (5)). This approach is equivalent to the residual image computation from the state-of-the-art temporal gradient domain path tracing [MKD*16] and control-variates-based re-rendering method [RJN16]. The path pairs are generated using path tracing with the same random sequence, ensuring strong correlation between paths traced from the same pixel. Components such as spatial gradient-domain path tracing, camera reprojection and Poisson blending from temporal gradient domain path tracing, and the optimization of multi-level Monte Carlo or control variates coefficients [RJN16] are orthogonal and can complement our method.

In Figs. 1 and 6 to 8, we show an equal-time comparison of renderings for the new scene by the aforementioned two baselines and our method: 3) *Incremental*: Rendering the residual image using our dynamic path sampling techniques and corresponding path mappings. The residual image is then added back to the old frame to produce the new frame. Each scene showcases both the comparison for the final rendering of the new scene as well as the comparison of the residual image rendering between Correlated PT and our approach. Notably, the residual images typically exhibit lower and sparser energy levels compared to the primal renderings. This observation underscores our initial motivation to selectively sample the residual path integral, focusing on regions where the difference in radiance contribution is non-zero (Eq. (5) and Eq. (6)). Additionally, the residual images contain both negative and positive values, occasionally resulting in visual disparities in brightness or color when comparing unbiased images with varying noise levels. It is anticipated but can be counter-intuitive given familiarity with traditional rendering. Dark pixels (e.g., third row in Fig. 8) in some final renderings from the baseline or our method result from negative values in the residual image. These disappear with sufficient samples. Future work could apply filtering/denoising techniques tailored to residual images.

We can also view our problem as simulating a *difference radiance* quantity (Section 3.2), where the visual impact of the moving object on its surrounding environment is predominantly influenced by its proximity to the light source. When a dynamic object is situated closer to the light source (LIVINGROOM in Fig. 6), it is more likely to reflect more light, resulting in affected paths that generally carry greater energy. To cover this variability, we showcase a range of scenarios across multiple scenes (Figs. 6 to 8). Our approach generally converges faster irrespective of the distance between dynamic objects and the light source. Overall, we achieve speedups ranging from 2x to 5x (and 4x to 16x for material editing) compared to Correlated PT, and frequently observe orders of magnitude speedups compared to traditional path tracing. In some cases, with significant fireflies, mean square error may not be the best metric, so we encour-

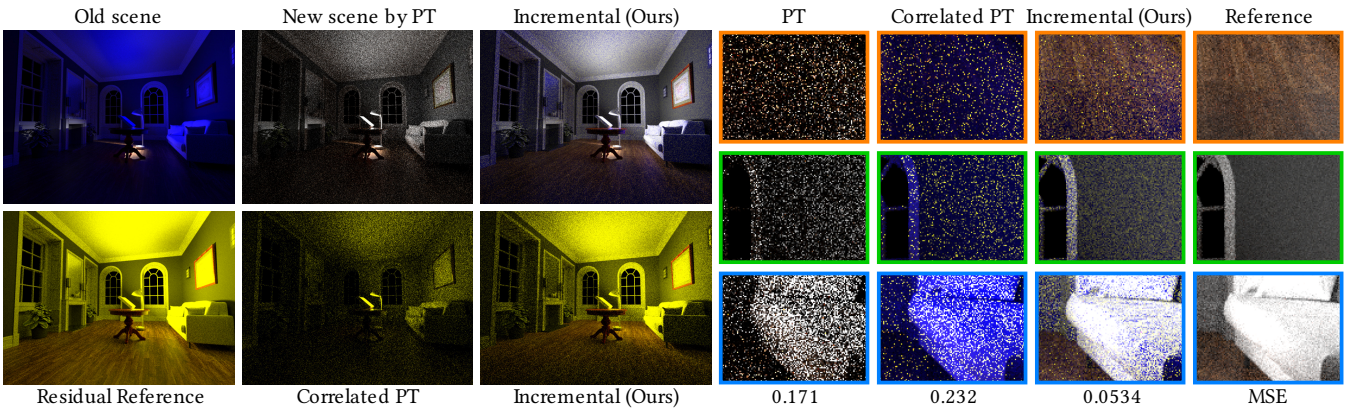


Figure 6: **Equal-time comparison**. We edit the material of the book (marked in yellow box) below a lamp shedding white light. The albedo is edited from blue to white, thereby the book is reflecting differently shaded light into the scene and largely changing the global illumination. Path tracing: 64spp. Correlated PT: 32spp. Incremental(Ours): 10spp*3; Reference for each frame: 6400spp. Please zoom in for better inspection of noise.

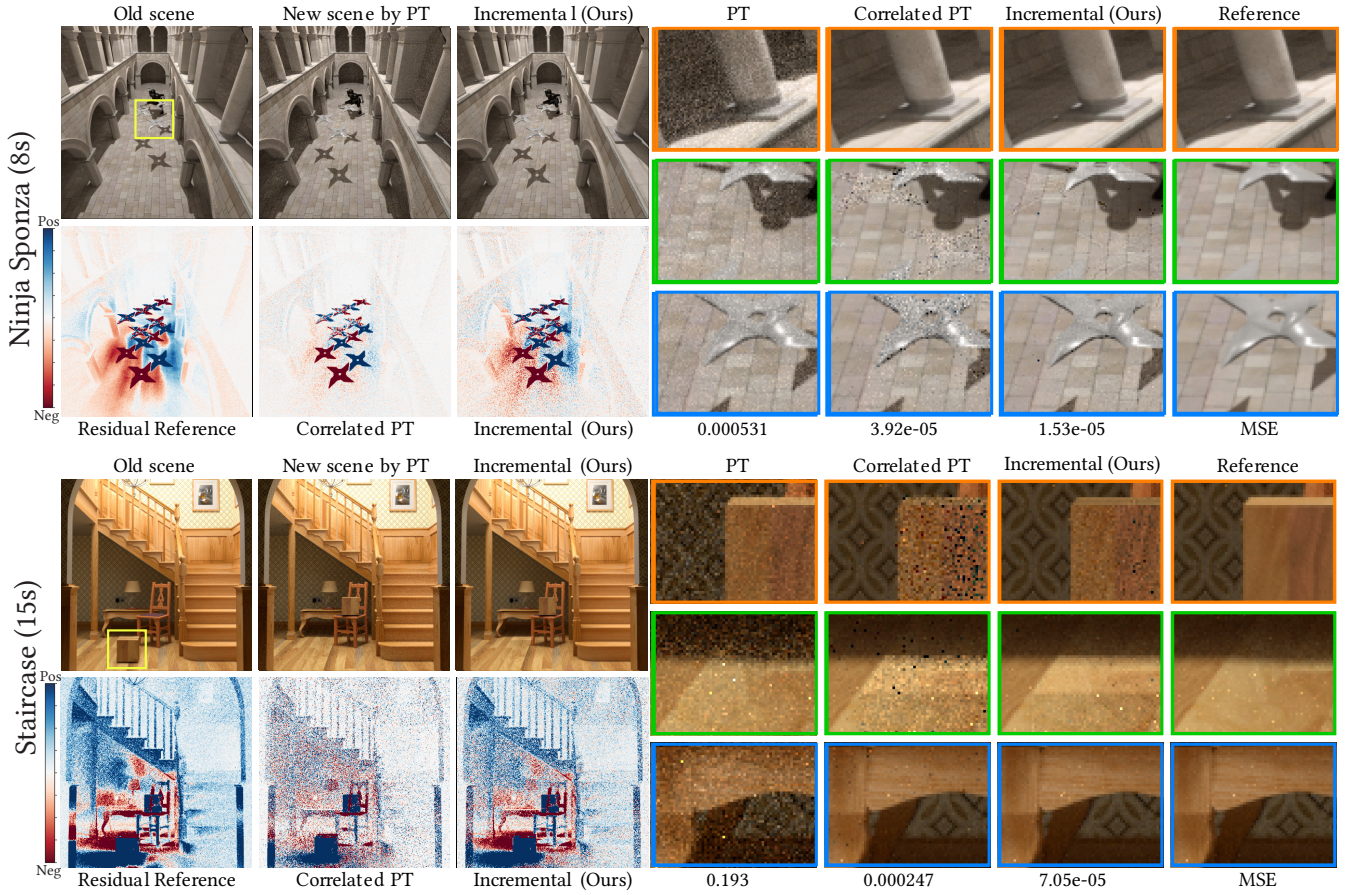


Figure 7: **Equal-time comparison**. Path tracing: 64 samples per pixel (spp). Correlated PT: 32spp. Incremental(Ours): 4spp*7; Reference for each frame: 2048spp. Moving objects are marked in the Old scene on the top left: three stars are flying in Ninja Sponza and a treasure chest is put on the chair in Staircase. We show extra comparison for the residual images with Correlated PT on the bottom left. Numbers are calculated on the full-res images. Please zoom in for better inspection of noise for all images.

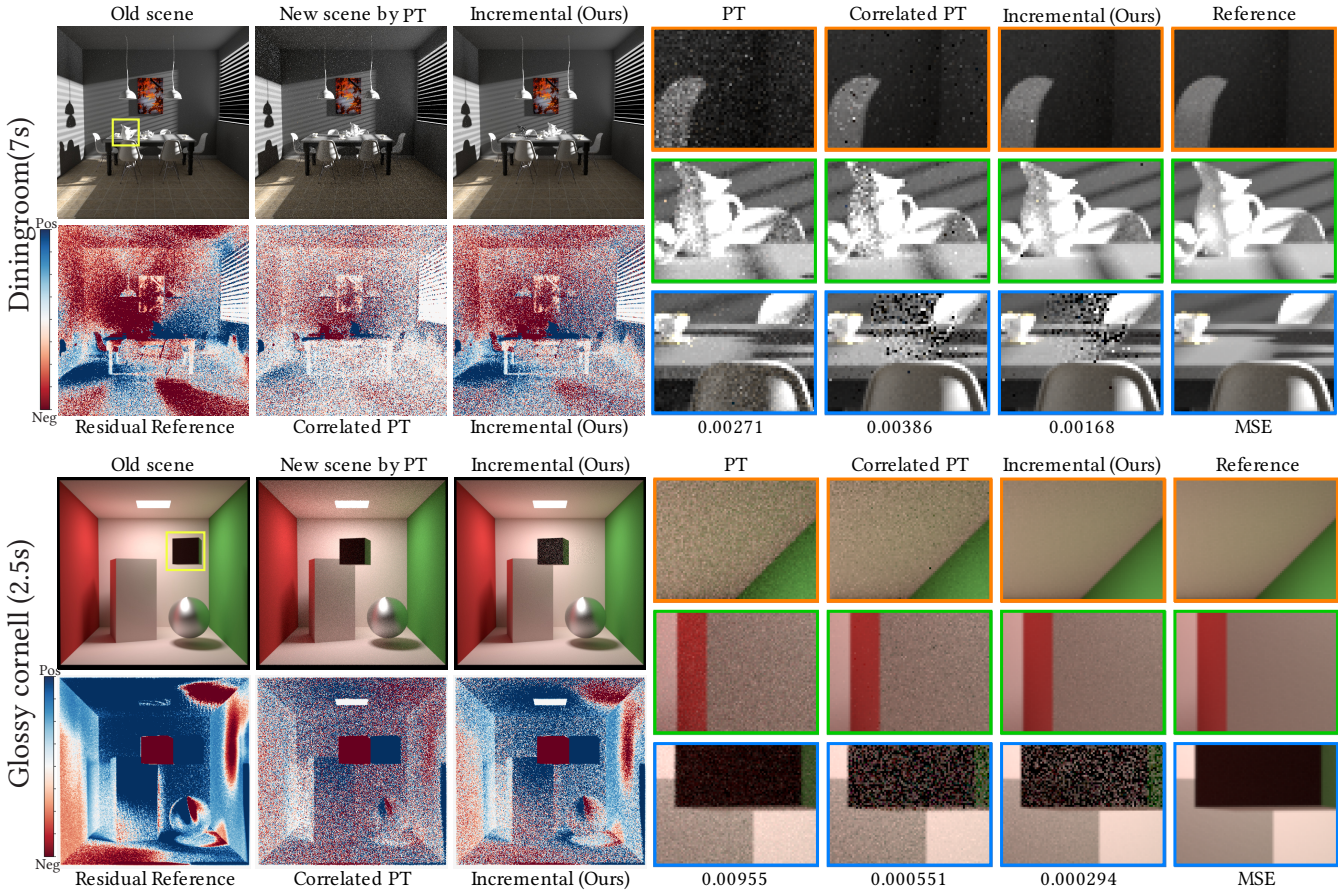


Figure 8: **Equal-time comparison**. Moving objects are marked in the Old scene on the top left: the pitcher is moving closer to the window in Dining room and the box is moving left in Glossy Cornell. For top row, Path tracing: 88spp. Correlated PT: 44spp. Incremental(Ours): 4spp*7; Reference for each frame: 4096spp. Bottom row:64spp/32spp/4spp*7/2048spp. Please zoom in for better inspection of noise.

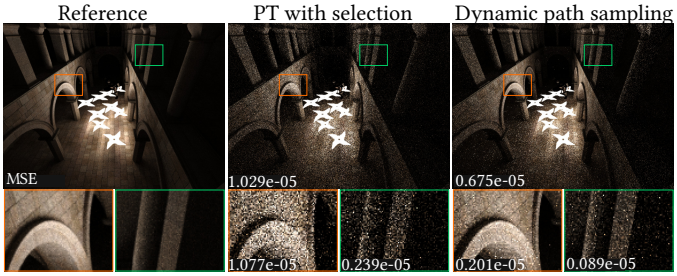


Figure 9: **Equal-time ablation for the rendering of dynamic paths**. 1) PT with selection: path tracing where only the paths affected by the dynamic objects are selected and contributing to the shown image and 2) Our dynamic path sampling technique.

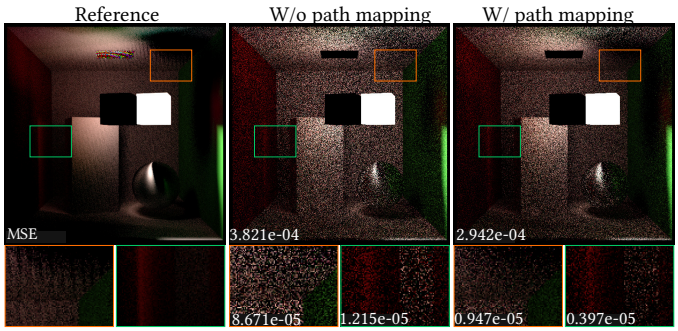


Figure 10: **Equal-time ablation of the effect of path mapping**. In the middle, we compute the residual image by independently rendering the two frames using our dynamic path sampling techniques. On the right,we further apply path mapping to correlate the two frames. See more ablation in Table 3.

age readers to zoom in to evaluate the visual quality of the images. As shown in Fig.11 where the editing clearly affects the GI of the entire scene, we consistently gain 4× runtime benefit for visually more converged images against path tracing.

Most of the scenes we show are easy to render for path tracing (e.g. the Glossy Cornell scene in Fig. 8). Occasionally, the camera tracing outperforms our dynamic path sampling when the dynamic object is close to the camera. This can be seen in the third row of the Glossy Cornell scene, and also applies to correlated path tracing. Nevertheless, our approach consistently performs better than the baselines both numerically and visually. The advantage brought by the dynamic path sampling techniques is especially obvious when

Table 2: Equal-time results for three more material editing operations for GLOSSY CORNELL. All MSE values are scaled by 10^5.

Metric	Editing operation	PT from scratch	Correlated PT	Incremental (ours)
MSE	Lambertian color:white to blue	990	17.9	2.10
MSE	GGX roughness: 0.1 to 0.5	990	22.2	2.91
MSE	Lambertian to metalic	990	88.9	5.48
SSIM	Lambertian color:white to blue	0.27	0.56	0.84
SSIM	GGX roughness: 0.1 to 0.5	0.28	0.59	0.79
SSIM	Lambertian to metalic	0.27	0.40	0.73

the light source is difficult to reach. The dynamic path finding would also not be easy for other light transport methods like bidirectional path tracing or light tracing when the dynamic object is not close to the light source as well (e.g., the DINING ROOM scene in Fig. 8).

Scene editing. In a scene editing scenario, designers can quickly initiate multiple edits from one ready-to-use rendering using our incremental re-rendering, as depicted in Fig. 1.

Our method aims to fill the gap left in the previous literature on re-rendering for dynamic objects. However, this more general formulation of the residual path space also encapsulates special cases, such as dynamic camera/lights and material editing. When the endpoint such as camera/lights move, the best sampling strategy degenerates into camera/light tracing. Material editing is a simpler problem without path space domain transformation. It requires only half of our solution set. With no movement, dynamic objects and their ghost counterparts coincide perfectly, eliminating the need for ghost object representation and associated computation. We can easily handle this application by partially disabling sampling and multiple importance sampling implementation, showcasing significant advantages across various material editing operations in Fig. 1e, Fig. 6 and Table 2.

6.3. Timings and overhead

In our implementation, we observe that under equal number of random walks (also with very similar CPU instruction counts), we perform essentially the same number of intersection tests as standard path tracing. However, our sampling methods may experience a higher rate of cache misses. This may be because the path tracer starts all the paths from the camera, while ours start from the middle of the scene. We leave the potential batching and coherency optimization as future work.

For the reason above, our sampling techniques do introduce some overhead. Thus for all the equal time comparisons, path tracing usually performs more random walks than ours. Nevertheless, this overhead is small, on the order of 10-20%. For each sample per pixel[¶], we apply 7 sampling techniques, where some paths directly connecting to the light source or the sensor will be rejected if the first intersection fails. Hence, for n samples per pixel, we typically generate less than $7n$ paths.

[¶] The spp shown for our method is the total number of samples splatted into the image plane divided by the number of pixels.

Table 3: **Hard case setups**. We show the behavior of our method with respect to the number of dynamic objects and the magnitude of displacements in the editing, using the CORNELL BOX scene. This includes ablations with and without path mappings (PM). Note that these experiments do ***not*** adhere to single-control-variate protocols. Other variables such as object locations, sizes and proximity to the light, significantly influence the results. We intentionally place the objects far away from each other in the space to affect more paths for (a). All MSE values are scaled by 10^5. Please refer to Fig. 14 in the Appendix for visuals.

(a) Increasing the number of dynamic objects.

Metric	Dynamic count	PT from scratch	Correlated PT	Incremental w/o PM (ours)	Incremental w/ PM (ours)
MSE	1	920	14.0 (5.1x)	3.38 (1.22x)	2.76 (1x)
MSE	2	960	18.2 (2.8x)	7.84 (1.20x)	6.53 (1x)
MSE	4	720	22.9 (1.4x)	20.0 (1.23x)	16.2 (1x)
MSE	8	1000	96.2 (1.1x)	109 (1.24x)	87.4 (1x)
SSIM	1	0.27	0.68 (0.7x)	0.89	0.94 (1x)
SSIM	2	0.28	0.61 (0.7x)	0.79	0.87 (1x)
SSIM	4	0.32	0.56 (0.7x)	0.69	0.80 (1x)
SSIM	8	0.31	0.40 (0.7x)	0.46	0.57 (1x)

(b) Increasing the movement displacement of the dynamic object.

Metric	Distant	PT from scratch	Correlated PT	Incremental w/o PM (ours)	Incremental w/ PM (ours)
MSE	40	990	10.0 (4.4x)	2.76 (1.22x)	2.26 (1x)
MSE	100	990	11.5 (4.2x)	3.31 (1.20x)	2.75 (1x)
MSE	160	990	11.7 (5.2x)	2.69 (1.18x)	2.27 (1x)
MSE	220	920	8.72 (5.0x)	1.95 (1.12x)	1.74 (1x)
SSIM	40	0.28	0.71 (0.7x)	0.90	0.95 (1x)
SSIM	100	0.28	0.69 (0.7x)	0.89	0.94 (1x)
SSIM	160	0.28	0.69 (0.7x)	0.89	0.93 (1x)
SSIM	220	0.27	0.74 (0.7x)	0.91	0.93 (1x)

6.4. Ablations and analysis

Figure 9 shows an equal time comparison for the rendering of dynamic paths versus simply doing path tracing with rejection of paths not passing through dynamic and ghost objects for the Ninja Sponza scene. Our dynamic path sampling technique substantially reduces noise, motivating the use of the novel sampling strategies proposed in this paper that start from dynamic and ghost objects. In Fig. 10, we justify the use of our path mapping approach, showing an equal time comparison of computing the difference images with and without path mapping. A naïve way to compute the residual image is to directly subtract I_1 from I_2 (Eq. (4)). I_2 and I_1 have their own domains and can be rendered independently. For example, path tracing can use different random seeds for two frames and the residual images are much more noisy than using Corrected PT. Without path mapping, we can still render the two frames separately using our dynamic path sampling techniques extracting paths affected by the motion. We can see that rendering the two frames separately does not preserve the correlation introduced by the path mapping approaches. However, the benefit brought by path mapping will drop with more dramatic scene changes.

Scale of moving objects. The residual path space will reduce to a finite difference of the whole path space if all paths are affected (e.g., when most of the objects are moving). This breaks our motivational

Figure 11: **Runtime benefit** in the LIVING ROOM example, where the editing clearly impacts the whole scene through global illumination. We consistently achieve a 4x runtime benefit, producing visually more converged images compared to the baselines.

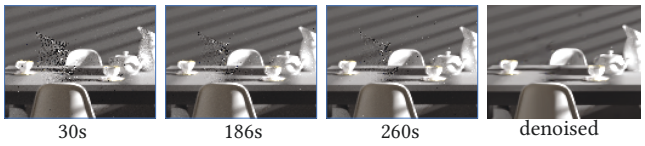


Figure 12: **The ghosting** in the DININGROOM of Fig.7 reduces during the convergence of the residual rendering. The artifacts are inherent to the image-space control variates methods. Note that this is unbiased. On the right, we apply the off-the-shelf OptiX denoiser to demonstrate that filtering helps mitigate these artifacts. Developing denoisers specifically tailored for our incremental rendering could be an interesting direction for future work.

assumption and our dynamic path sampling techniques have no advantage in this case. Scalability analysis is presented in Table 3 (a). We uniformly scatter eight moving objects across the scene to affect the whole path space. As dynamic objects occupy more of the primal path space, the benefits of our sampling technique diminish.

Magnitude of movements. Our approach demonstrates robustness with multiple dynamic objects and rigid transformations. We do not impose constraints on the amount of movement. In the worst case scenario, a large motion will break the possibility of any path mapping and essentially fall back to finite differences. We can see from Table 3 (b): as the displacement increases, the benefit of our sampling techniques remains consistent, while the benefit of path mapping diminishes (220 completely spans one side of the scene).

6.5. Limitations

Our dynamic path sampling typically starts from the dynamic or ghost objects in the middle of the scene. Path tracing can inherently perform better when the scene favors camera tracing and image domain sampling. We currently employ path tracing as one extra sampling technique to incorporate the advantage brought by camera tracing. Ideally, the eye sub-paths can be extended to combine with the camera tracing using bidirectional methods. Symmetrically, the light sub-paths can be combined with light tracing. However, this will bring more challenges in multiple importance sampling.

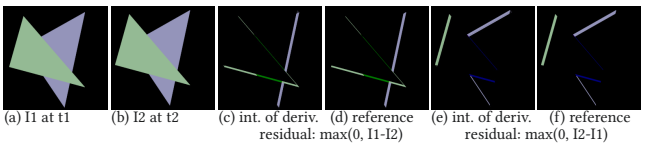


Figure 13: **Residual rendered as integral of derivatives w.r.t time**. We render the derivative using edge sampling [LADL18] and do Monte Carlo integration by uniformly sampling the time axis.

Additionally, our algorithm is not targeted at a dynamic camera or light which will affect every pixel and reduce the incremental path space into the primal path space.

In some of the new frames generated by our incremental rendering method (and by correlated path tracing), the variance is concentrated at the positions of the dynamic and ghost objects. Especially when the higher variance is concentrated at the ghost objects (the previous position of the dynamic objects), the inconsistent noise level for neighboring pixels can sometimes be visually disturbing (See the 3rd row in Fig. 8 Dining room). These areas will gradually converge with more samples, and methods like adaptive sampling and denoising can also be used to alleviate the noise (see Fig.12).

6.6. Relationship to differentiable rendering

The *Dynamic Two Ends* sampling technique shares a similarity in starting from the middle of the path with path-space differential rendering (PSDR) [ZMY*20]. However, PSDR starts paths from edges (one less dimension than surfaces), so the paths have statistically zero chance of intersecting the edges multiple times.

The residual image rendered by our method can be seen as an integral of derivatives w.r.t. time, which can be calculated by differentiable rendering methods [LADL18, LHJ19, ZWRY21, BLD20, ZMY*20, ZRJ23]. In Fig. 13, we show one 2D example with moving triangles where the residual is calculated as integral of derivatives using edge sampling [LADL18]. The integral can be difficult to solve efficiently due to potential high variance along the time axis. It is possible to derive better importance sampling techniques on the extra time dimension. In contrast, we directly solve the integral by treating it as a regular point sampling problem.

7. Conclusion and future work

We have introduced a theoretical framework for, and initial practical applications of, the residual path integral and scene re-rendering with moving objects and material authoring. Most previous rendering techniques, for instance, the acceleration structures, intersection behaviors, sampling strategies are specially tailored and highly optimized for primal path space. The rendering algorithms for the residual path integral are far from being exhaustively explored and optimized. We have taken a first step towards shedding light on this research direction.

Our work opens up several future directions. We sample the residual path integral where the difference radiance contribution is nonzero. There can be better importance sampling techniques to sample proportionally to the actual difference integrand. Better path mappings can also be designed to increase the correlation, to avoid compromising the advantage brought by the novel sampling strategies. More ambitiously, joint optimization of path mapping and importance sampling of the difference integrand can be achieved. Another direction would be to support deformable movement. Finally, our current approach only focuses on surface scattering. Extensions to participating media is an interesting future work.

Acknowledgements. This work was funded in part by NSF grants 2105806, 2212085, 2110409, gifts from Adobe, Google, the Ronald L. Graham Chair and the UC San Diego Center for Visual Computing. We especially thank Shilin Zhu for countless discussions in the early stages of this project. We thank Rui Tang, Lifan Wu, Ling-Qi Yan, Fujun Luan, Alexandr Kuznetsov, Yash Belhe for earlier discussions; Xuanda Yang, Yang Zhou for later performance discussions; Varun Munagala for early visualization experiments, and the anonymous reviewers for their constructive feedback. Scenes courtesy of Benedikt Bitterli.

References

[ALLD17] ANDERSON L., LI T.-M., LEHTINEN J., DURAND F.: Aether: An embedded domain specific sampling language for Monte Carlo rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH) 36*, 4 (2017), 99:1–99:16.

[BDT99] BALA K., DORSEY J., TELLER S.: Radiance interpolants for accelerated bounded-error ray tracing. *ACM TOG 18*, 3 (1999), 213–256.

[BFMZ94] BISHOP G., FUCHS H., MCMILLAN L., ZAGIER E. J. S.: Frameless rendering: Double buffering considered harmful. In *SIGGRAPH* (1994), p. 175–176.

[BLD20] BANGARU S. P., LI T.-M., DURAND F.: Unbiased warped-area sampling for differentiable rendering. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–18.

[BNJ15] BITTERLI B., NOVÁK J., JAROSZ W.: Portal-masked environment map sampling. In *Computer Graphics Forum (Proc. EGSR)* (2015), vol. 34, Wiley Online Library, pp. 13–19.

[BWP*20] BITTERLI B., WYMAN C., PHARR M., SHIRLEY P., LEFOHN A., JAROSZ W.: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph. (Proc. SIGGRAPH) 39*, 4 (2020).

[Che90] CHEN S. E.: Incremental radiosity: an extension of progressive radiosity to an interactive image synthesis system. *Comput. Graph. (Proc. SIGGRAPH) 24*, 4 (1990), 135–144.

[CKS*17] CHAITANYA C. R. A., KAPLANYAN A. S., SCHIED C., SALVI M., LEFOHN A., NOWROUZEZAHRAI D., AILA T.: Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph. (Proc. SIGGRAPH) 36*, 4 (2017).

[DBMS02] DMITRIEV K., BRABEC S., MYSZKOWSKI K., SEIDEL H.-P.: Interactive global illumination using selective photon tracing. In *Eurographics Workshop on Rendering* (2002), p. 25–36.

[DS97] DRETTAKIS G., SILLION F. X.: Interactive update of global illumination using a line-space hierarchy. In *SIGGRAPH* (1997), pp. 57–64.

[FYT95] FORSYTH D. W., YANG C., TEO K.: Efficient radiosity in dynamic environments. *Photorealistic Rendering Techniques (Proc. EGWR)* (1995).

[GD01] GRANIER X., DRETTAKIS G.: Incremental updates for rapid glossy global illumination. *Comput. Graph. Forum (Proc. Eurographics) 20*, 3 (2001), 268–277.

[Geo12] GEORGIEV I.: *Implementing Vertex Connection and Merging*. Tech. rep., Saarland University, 2012.

[GG15] GÜNTHER T., GROSCH T.: Consistent scene editing by progressive difference images. *Comput. Graph. Forum 34*, 4 (2015), 41–51.

[GKDS12] GEORGIEV I., KŘIVÁNEK J., DAVIDOVIČ T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6 (2012), 192:1–192:10.

[Hec90] HECKBERT P. S.: Adaptive radiosity textures for bidirectional ray tracing. vol. 24, p. 145–154.

[HGP*19a] HUA B.-S., GRUSON A., PETITJEAN V., ZWICKER M., NOWROUZEZAHRAI D., EISEMANN E., HACHISUKA T.: A survey on gradient-domain rendering. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 455–472.

[HGP*19b] HUA B.-S., GRUSON A., PETITJEAN V., ZWICKER M., NOWROUZEZAHRAI D., EISEMANN E., HACHISUKA T.: A survey on gradient-domain rendering. *Comput. Graph. Forum (Proc. Eurographics STAR) 38*, 2 (2019), 455–472.

[Jak10] JAKOB W.: Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[Jen96] JENSEN H. W.: Global illumination using photon maps. In *Rendering Techniques (Proc. EGWR)* (1996), pp. 21–30.

[Kel97] KELLER A.: Instant radiosity. In *SIGGRAPH* (1997), pp. 49–56.

[KLR*23] KETTUNEN M., LIN D., RAMAMOORTHI R., BASHFORD-ROGERS T., WYMAN C.: Conditional resampled importance sampling and restir. In *SIGGRAPH Asia 2023 Conference Papers* (2023), pp. 1–11.

[KMA*15] KETTUNEN M., MANZI M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain path tracing. *ACM Trans. Graph. (Proc. SIGGRAPH) 34*, 4 (2015).

[LADL18] LI T.-M., AITTALA M., DURAND F., LEHTINEN J.: Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–11.

[LHJ19] LOUBET G., HOLZSCHUCH N., JAKOB W.: Reparameterizing discontinuous integrands for differentiable rendering. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–14.

[LKB*22] LIN D., KETTUNEN M., BITTERLI B., PANTALEONI J., YUKSEL C., WYMAN C.: Generalized resampled importance sampling: Foundations of ReSTIR. *ACM Trans. Graph. (Proc. SIGGRAPH) 41*, 4 (2022).

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-domain Metropolis light transport. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4 (2013).

[LSK*07] LAINE S., SARANSAARI H., KONTKANEN J., LEHTINEN J., AILA T.: Incremental instant radiosity for real-time indirect illumination. *Rendering Techniques (Proc. EGSR)* (2007).

[LW95] LAFORTUNE E. P., WILLEMS Y. D.: A 5D tree to reduce the variance of Monte Carlo ray tracing. pp. 11–20.

[MA06] MEYER M., ANDERSON J.: Statistical acceleration for animated global illumination. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (2006), 1075–1080.

[MKD*16] MANZI M., KETTUNEN M., DURAND F., ZWICKER M., LEHTINEN J.: Temporal gradient-domain path tracing. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 35*, 6 (2016).

[MPT03] MARTÍN I., PUEYO X., TOST D.: Frame-to-frame coherent animation with two-pass radiosity. *IEEE Trans. Vis. Comput. Graph. 9*, 1 (2003), 70–84.

[MRK*14] MANZI M., ROUSSELLE F., KETTUNEN M., LEHTINEN J., ZWICKER M.: Improved sampling for gradient-domain Metropolis light transport. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 33*, 6 (2014).

[MRKN20] MÜLLER T., ROUSSELLE F., KELLER A., NOVÁK J.: Neural control variates. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 39*, 6 (2020).

[MRNK21] MÜLLER T., ROUSSELLE F., NOVÁK J., KELLER A.: Real-time neural radiance caching for path tracing. *ACM Trans. Graph. (Proc. SIGGRAPH) 40*, 4 (2021).

[NSL*07] NEHAB D., SANDER P. V., LAWRENCE J., TATARCHUK N., ISIDORO J. R.: Accelerating real-time shading with reverse reprojection caching. In *Graphics Hardware* (2007), p. 25–35.

[OHD20] OTSU H., HANIKA J., DACHSBACHER C.: Portal-based path perturbation for metropolis light transport. In *Proceedings of Vision, Modeling and Visualization* (2020).

[OLK*21] OUYANG Y., LIU S., KETTUNEN M., PHARR M., PANTALEONI J.: ReSTIR GI: Path resampling for real-time path tracing. *Comput. Graph. Forum (Proc. HPG) 40*, 8 (2021), 17–29.

[RHJD18] REIBOLD F., HANIKA J., JUNG A., DACHSBACHER C.: Selective guided sampling with complete light transport paths. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 37*, 6 (2018), 223:1–223:14.

[RJN16] ROUSSELLE F., JAROSZ W., NOVÁK J.: Image-space control variates for rendering. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 35*, 6 (2016).

[SKW*17] SCHIED C., KAPLANYAN A., WYMAN C., PATNEY A., CHAITANYA C. R. A., BURGESS J., LIU S., DACHSBACHER C., LEFOHN A., SALVI M.: Spatiotemporal variance-guided filtering: Real-time reconstruction for path-traced global illumination. In *High Performance Graphics* (2017).

[TCE05] TALBOT J. F., CLINE D., EGBERT P.: Importance resampling for global illumination. *Rendering Techniques (Proc. EGSR)* (2005), 139–146.

[TPWG02] TOLE P., PELLACINI F., WALTER B., GREENBERG D. P.: Interactive global illumination in dynamic scenes. *ACM Trans. Graph. (Proc. SIGGRAPH) 21*, 3 (2002), 537–546.

[Vea98] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1998.

[VKS*14] VORBA J., KARLÍK O., SIK M., RITSCHEL T., KRIVÁNEK J.: On-line learning of parametric mixture models for light transport simulation. *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4 (2014), 101:1–101:11.

[WDP99] WALTER B., DRETTAKIS G., PARKER S.: Interactive rendering using the render cache. In *Eurographics Workshop on Rendering Techniques* (1999), pp. 19–30.

[WG12] WEISS M., GROSCH T.: Stochastic progressive photon mapping for dynamic scenes. *Comput. Graph. Forum (Proc. Eurographics) 31*, 2 (2012), 719–726.

[ZMY*20] ZHANG C., MILLER B., YAN K., GKIOULEKAS I., ZHAO S.: Path-space differentiable rendering. *ACM Trans. Graph. 39*, 4 (2020), 143:1–143:19.

[ZRJ*15] ZIMMER H., ROUSSELLE F., JAKOB W., WANG O., ADLER D., JAROSZ W., SORKINE-HORNUNG O., SORKINE-HORNUNG A.: Path-space motion estimation and decomposition for robust animation filtering. *Comput. Graph. Forum (Proc. EGSR) 34*, 4 (2015), 131–142.

[ZRJ23] ZHANG Z., ROUSSEL N., JAKOB W.: Projective sampling for differentiable rendering of geometry. *Transactions on Graphics (Proceedings of SIGGRAPH Asia) 42*, 6 (Dec. 2023). doi:10.1145/3618385.

[ZWRY21] ZHOU Y., WU L., RAMAMOORTHI R., YAN L.-Q.: Vectorization for fast, analytic, and differentiable visibility. *ACM Transactions on Graphics (TOG) 40*, 3 (2021), 1–21.

Appendix A: Path mappings for complete dynamic paths

Building upon the operators we define in Sec.4, here we summarize the path mapping strategies for the complete dynamic paths. Without loss of generality, we map base paths $\boldsymbol{p}$ sampled from the new scene (Fig. 5 black paths) to paths in the old scene (Fig. 5 blue paths).

Standard path tracing can simply reuse the random sequence to significantly increase the correlation between paths in two frames [MKD*16], as the pair inherently starts from the same pixel. This is more challenging in the dynamic scenarios. Our dynamic path sampling instead starts from the middle, causing pairs of paths to often land on different pixels. Consequently, the correlation does not contribute to variance reduction in the image domain.

Below we briefly present the path mapping we designed for each dynamic path sampling technique. To reduce clutter, we use acronym *PM* to refer to *path mapping* throughout this section. The main idea is to do path reconnection as soon as possible (by *Keep vertex the same*). This is both for 1) efficiency, in terms of path reuse, and 2) increasing correlation between the two paths. Until the re-connection condition is met, the paths are diverged, meaning that they do not share the same path vertex. We usually employ *Random seed replay* until the paths reconnect. An example is shown in Fig. 5(1a), where the path fails to meet the reconnection condition at x_1 due to excessive glossiness and only manages to reconnect at x_3.

We demonstrate PM for *Dynamic from light* in Fig. 5(1). Specifically, the first point sampled on a dynamic object (x_D) is mapped to the same position $t(x_D)$ using *Transform with object movement*. We generally continue using *Random seed replay* until path reconnection. A special case is shown in Fig. 5(1b), which only arises in dynamic scenes. When the base path interacts with dynamic objects more than once, we proactively terminate the mapped path using the *Rejection* operator. While it is theoretically possible to apply the *Transform with object movement* operator again for any dynamic vertex, this can easily result in a large Jacobian in practice, especially given large movement. In Fig. 5(2), we show PM for *Dynamic two ends*. The base path for each subpath is mapped similarly.

One special detail for dynamic scenes is that paths can diverge again whenever they encounter dynamic objects after a path reconnection. We maintain an additional acceleration structure that contains only the dynamic objects to re-check visibility for each edge after reconnection. Since we assume only a small portion of the scene is moving, this intersection re-test adds relatively little overhead.

For sampling techniques *Dynamic from sensor*, simply applying the same strategies as in above cases does not work. If we similarly use the operator *Transform with object movement*, the corresponding paths start from the middle of the scene and then directly reconnect to the sensor. Without a strict constraint on the scale of the movement, the paths will contribute largely to different pixels, thus wasting the correlation. We instead map between the corresponding dynamic and ghost objects as shown in Fig. 5(3). In (3a) we map x_D sampled on a dynamic object to the same position on its ghost counterpart $t(x_D)$. Since $t(x_D)$ is a ghost vertex, we continue tracing in the direction of $\overrightarrow{x_E t(x_D)}$ until it hits a solid object x_1. This has the same effect of having primary rays from the sensor shoot into the same direction. Subsequently, we construct the mapped path using *Random seed replay* for the remaining vertices. We choose not to do path re-connection here due to the potentially large Jacobian caused by starting points located on different objects. Symmetrically, we design a similar mapping for *Ghost from sensor* (Fig. 5(3b)).

Two sampling techniques remain for ghost objects: *Ghost from light* and *Ghost two ends*. We designed and implemented their path mapping strategies similarly, but found that they did not assist as they did for dynamic objects. For ghost objects, we commonly observe a scenario where one of the corresponding paths starting from the ghost is largely blocked by its nearby dynamic counterpart, which reduces to finite differences. Additionally, since a ghost object can be (partly) open space, rays passing through the ghost mapped between frames largely intersect different objects, leading to low correlation. Hence we simply adopt independent tracing. Due to the difficulty of PM inherently imposed by the dynamic path sampling techniques, part of the advantage over path tracing gained from the better sampling strategies is equalized by a simple correlated

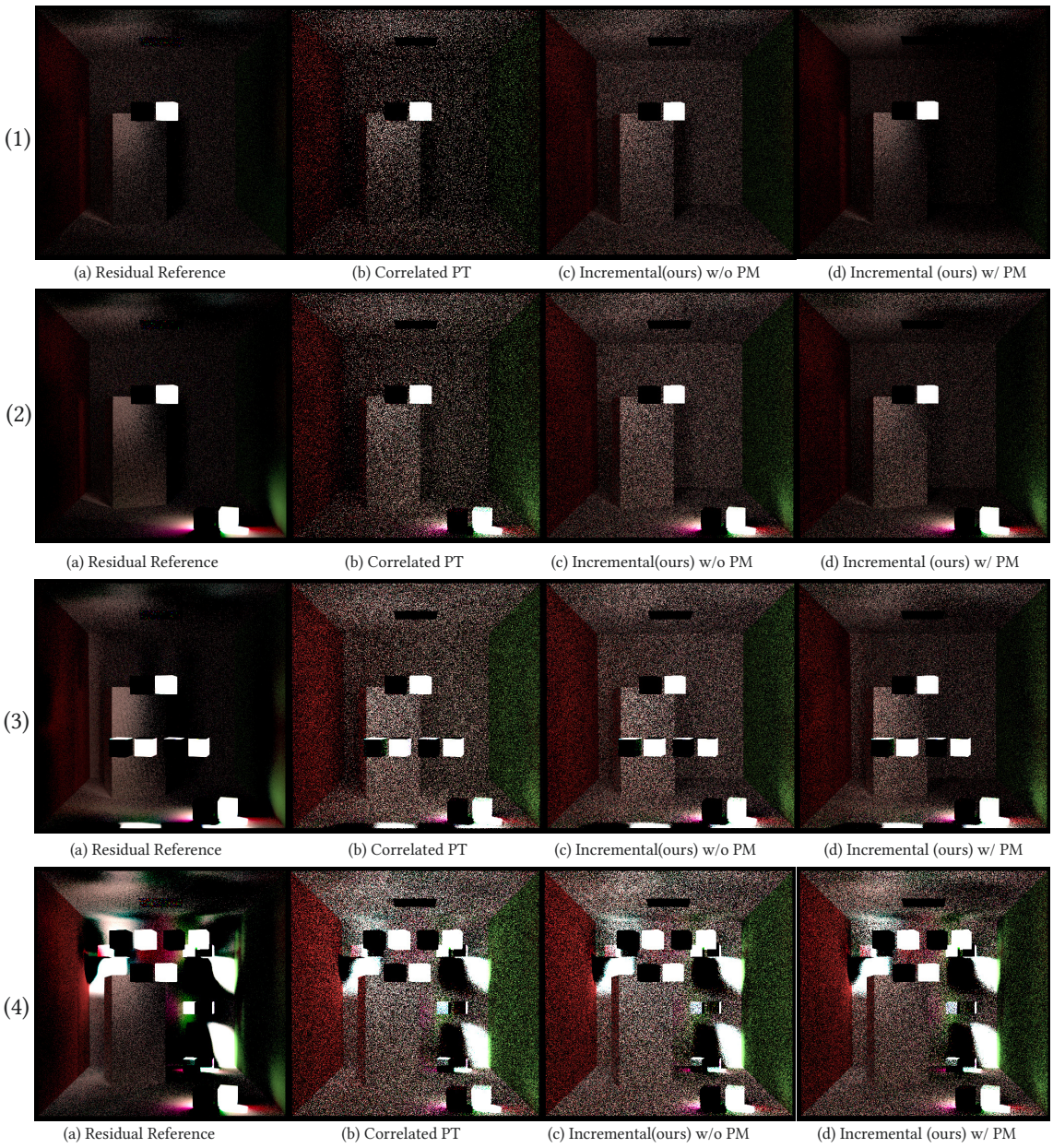


Figure 14: Equal-time comparisons of residual images for Table3(a) of the main paper; (1) 1 dynamic objects; (2) 2 dynamic objects; (3) 4 dynamic objects; (4) 8 dynamic objects dispersedly scattered across the scene. Note again that this does *not* adhere to the single-control-variable protocols.

path tracing method for some cases. Nevertheless, we can see in our results that sampling the dynamic and ghost objects leads to significant performance improvements in most cases.