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

Volume 43 (2024), Number 4

Residual path integrals for re-rendering: Supplemental document

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

[1]University of California, San Diego, USA
[2]Adobe Research, UK

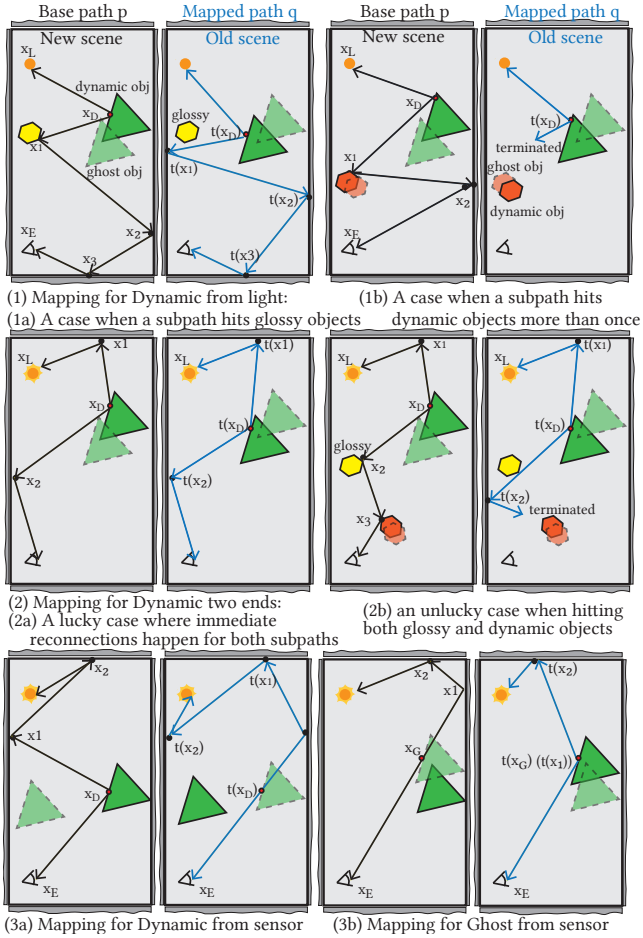

Figure 1: **Path mappings for various dynamic sampling techniques.** On the left of each pair, we sample the base path (black) using our methods shown in Sec.4 of the main paper. Mapped paths are blue on the right. Two frames can be flipped for the two-way path mapping. We include this figure again for easy reference.

In this supplemental material, we first present the path mappings for complete dynamic paths in detail. Then we present supplemental figures for the main paper.

1. Path mappings for complete dynamic paths

Building upon the operators we define in Sec.4 of the main paper, here we summarize the path mapping strategies for the complete dynamic paths (see Fig. 1). Without loss of generality, we map base paths $\boldsymbol{p}$ sampled from the new scene (Fig. 1 black paths) to paths in the old scene (Fig. 1 blue paths).

Standard path tracing can simply reuse the random sequence to significantly increase correlation between paths in two frames([MKD*16]), since the pair inherently starts from the same pixel. It is much more challenging in our dynamic scenario. Our dynamic path sampling instead starts from the middle. Therefore, a pair of paths has a large chance of landing on different pixels, and the correlation will not contribute to variance reduction in 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 the term *path mapping* in the remainder of this section. The main idea is to always do path re-connection as soon as possible (supported by operator *Keep Vertex the Same*). This is both for 1) efficiency, in terms of path reuse, and 2) increasing correlation between the two paths. Before the re-connection condition is met, the paths are diverged, meaning that they are not sharing the same path vertex. We usually opt for *Random Seed Replay* until the paths reconnect. One example is given in Fig. 1(1a), where the path fails the reconnection condition given x_1 is too glossy and only manages to reconnect at x_3.

We show PM for *Dynamic From Light* in Fig. 1(1). Concretely, the first point sampled on a dynamic object ($\boldsymbol{x}_D$) is mapped to the same position ($t(\boldsymbol{x}_D)$) by *Transform with Object Movement* operator. We generally stick to using random seed replay until path reconnection. A special case is illustrated in Fig. 1(1b) which only exists for dynamic scenes. When the base path interacts with dynamic objects more than once, we terminate the mapped path proactively using the *Rejection* operator. Theoretically we can apply the *Transform with Object Movement* operator again for any dynamic vertex, but it could easily lead to a large Jacobian in practice, especially given large movement. In Fig. 1(2), we show PM for *Dynamic Two Ends*. For either one of the subpaths, the base path is mapped in the same fashion.

One detail is special for dynamic scenes: the paths can diverge again whenever hitting dynamic objects after a path re-connection. We maintain an extra acceleration structure containing only the

dynamic objects to re-check the visibility for each edge after the re-connection. Since we assume only a small portion of the scene is moving, this intersection re-test adds relatively small 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 largely contribute to different pixels, squandering the correlation. We instead do the mapping between the corresponding dynamic and ghost objects as shown in Fig. 1(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 keep tracing in the direction of $x_E \rightarrow t(x_D)$ 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. We then simply construct the mapped path using *Random Seed Replay* for the remainder of the vertices. We select not to do path re-connection here due to the potentially large Jacobian caused by the starting points located on different objects in common scenarios. By symmetry, we design a similar mapping for *Ghost From Sensor* (Fig. 1(3b)).

There are two sampling techniques for ghost objects left: *Ghost From Light* and *Ghost Two Ends*. We designed and implemented their path mapping strategies in a similar fashion, but ended up finding them not helping as for dynamic objects. For ghost objects, we observe a common scenario that one of the corresponding paths starting from the ghost will largely be blocked by its nearby dynamic counterpart, which reduces to finite differences. Also since a ghost object can be an (partly) open space, the through-the-ghost rays mapped between the two frames will largely hit different objects, resulting in low correlation. Hence we simply adopt independent tracing in both frames.

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 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.

2. Supplemental figures for tables and convergence graph

We present the convergence graph in Fig. 2 and supplemental figures for Table.3(b) in Fig. 3.

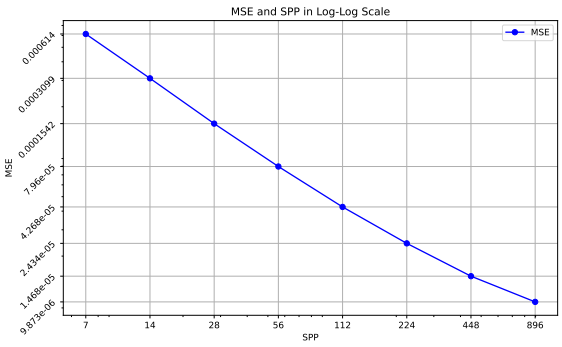

Figure 2: **Convergence plot** of our approach for Ninja Sponza scene as an example. We show MSE with respect to increasing sample per pixel in Log-Log scale. The expected slope verifies the unbiasedness of our approach.

References

[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). 1

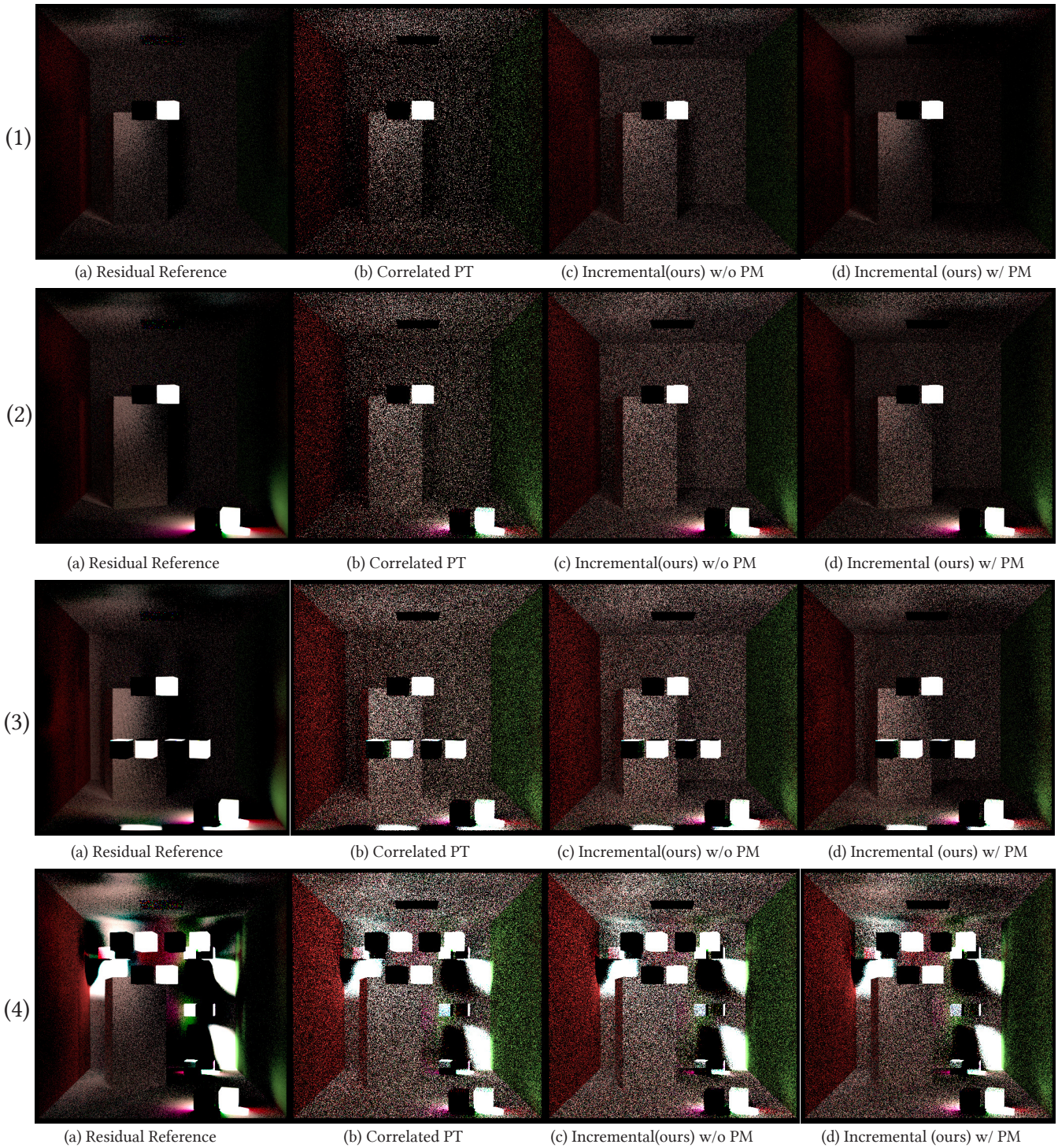

Figure 3: Equal-time comparisons of residual images for Table3(a) of the main paper; (1) 1 dynamic objects; (2) 2 dynamic objects; (3) 4 dynamic objects; (4) 8 dynamic objects dispersedly scattered across the scene. Note again that this does *not* adhere to the single-control-variable protocols.