Eurographics Symposium on Rendering 2026
I. Gkioulekas and A. Jarabo
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 4

Light-Cone Shell Queries for Scalable Time-Gated Rendering

Jack Cui Juhyeon Kim Adithya Pediredla Wojciech Jarosz

Dartmouth College

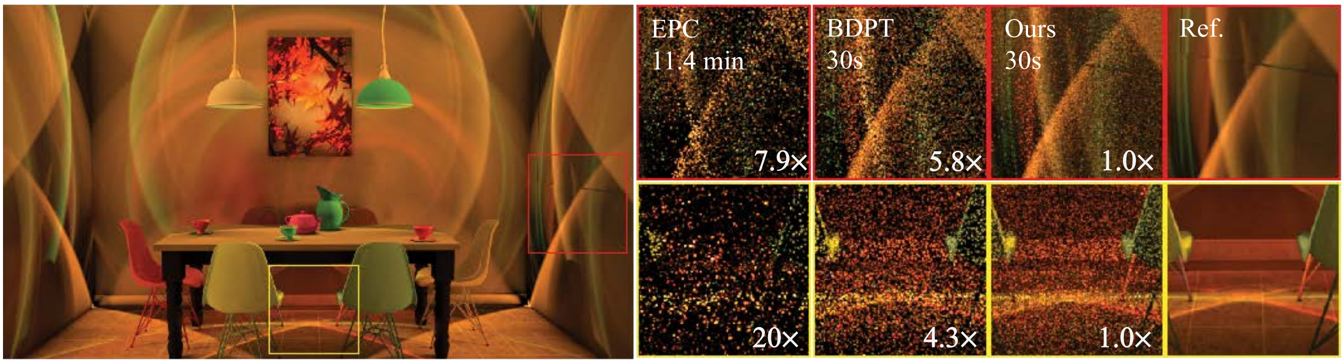

Figure 1: *Equal-time comparison of various time-gated rendering methods on the* Dining Room *scene (270k triangles, collimated beam emitter). BDPT and our method are shown at* 30 s*; Ellipsoidal path connections (EPC) [PVG19] require explicit ellipsoid–scene intersections and at this complexity produce a single sample in* 11.4 min *(shown at 1 spp). Our method replaces geometric intersections with point-based range queries over the light-cone shell, achieving time-gating cost independent of scene geometry and significantly reduced variance over both baselines (relMSE shown in insets).*

Abstract

Monte Carlo time-gated rendering requires sampling light paths that not only connect a sensor to an emitter, but which also have a total travel time that falls within a narrow interval, a constraint that is difficult to importance sample. We show that this problem has an underlying geometric structure: in the joint space of position and accumulated travel time, the points yielding a time-valid connection to a given query point form a light-cone shell. Prior methods sample this shell indirectly. Steady-state algorithms sample the full space and reject points outside it, giving high variance under tight gates. Ellipsoidal path connections target a single cone surface by intersecting an ellipsoid with scene geometry, coupling cost to scene complexity. Our key observation is that shell membership is cheap to test, needing only accumulated travel time and a Euclidean distance. We therefore store the vertices of traced light subpaths in a 4D spatiotemporal hierarchy and recast time-gated connection as a range query, using pruning and importance sampling over the shell to select time-valid vertices without intersecting scene geometry. This decouples the cost of time gating from scene complexity. Within a bidirectional path tracing framework, our method significantly reduces variance over existing approaches on scenes with up to 2.4M triangles.

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing;*** *Modeling and simulation;*

1. Introduction

Time-of-flight (ToF) sensors measure photon travel time in addition to intensity. A particularly useful variant is the *time-gated* sensor, which records only photons whose total travel time falls within a narrow interval. Since a vanishingly small fraction of all transport paths fall within narrow time gates, the difficulty in simulating such sensors lies in *importance sampling* this constraint.

Time-gated sensors reject photons outside a narrow temporal window, enabling applications such as seeing through fog [GJBH19], recovering objects hidden around corners via non-line-of-sight (NLOS) imaging [PDV19], and performing depth-selective proximity detection. Developing and validating these applications relies on physically accurate simulation of time-gated sensors, which in turn requires rendering complex scenes under tight travel-time constraints at practical computation times.

Steady-state methods such as bidirectional path tracing (BDPT) sample paths without awareness of travel time, evaluating the constraint only after a path is formed. For narrow gates, the fraction of valid paths is small and the vast majority of effort is wasted. Photon-tracing methods [LJJ22; MGJ*19] deposit light vertices into spatial data structures and can exploit travel-time information during density estimation, but they primarily target transport in participating media. Ellipsoidal path connections (EPC) [PVG19] exactly sample paths of a desired travel time by introducing an auxiliary vertex at the intersection of an ellipsoid with scene geometry. This targets time-valid paths precisely, but the required ellipsoid–scene intersection couples the cost of time-gating directly to scene complexity.

We show that these approaches—and time-gated rendering in general—can be understood through a single geometric structure. In the joint space of position and accumulated travel time, the set of vertices producing a time-valid connection to a given sensor vertex forms a light-cone shell (LCS)—a region bounded by two cones, each corresponding to a gate boundary (Sec. 4). Steady-state methods sample the full space and discard vertices outside the shell. Ellipsoidal connections target a single cone surface precisely, but require explicit geometric intersections to do so.

Our key observation is that testing whether a light vertex lies within this shell requires only its accumulated travel time and its distance to the shade point — quantities readily available from stored light subpaths. This recasts the problem of finding time-valid connections as an efficient search over stored vertices.

We then propose an acceleration structure that queries the shell efficiently and independently of scene complexity. We organize stored light vertices into a spatiotemporal hierarchy over position and accumulated travel time. During traversal, nodes whose spatiotemporal bounding boxes fall entirely outside the shell are pruned, and contribution bounds are tightened to steer samples toward time-valid vertices. Each query costs $O(\log N)$ in the number of stored light vertices, decoupling time-gating from scene complexity (Sec. 5).

We integrate light-cone shell queries into a bidirectional path tracing framework (Sec. 6) and evaluate on scenes spanning 226k–2.4M triangles with narrow time gates, achieving up to 9× relMSE reduction over BDPT at equal time. Unlike EPC, our method's time-gating cost is independent of scene complexity, maintaining consistent performance as triangle count grows. We further demonstrate our method on time-gated imaging applications including seeing through fog and proximity detection (Sec. 7).

2. Related Work

Time-gated imaging and applications. Time-gated sensors selectively capture photons within a narrow travel-time window, enabling applications that are impossible with steady-state imaging [JMMG17]. In non-line-of-sight (NLOS) imaging, time-gated measurements recover objects hidden around corners by exploiting multi-bounce time-of-flight information [OLW18; VWG*12]. Time gating can also suppress multiply-scattered photons in participating media, allowing sensors to see through fog or tissue [GJBH19]. These applications motivate physically accurate simulation of time-gated sensors in complex scenes—the problem we address.

Time-of-flight rendering. Rendering time-resolved light transport requires evaluating the path integral subject to a path travel-time constraint. Smith et al. [SSD08] and Jarabo et al. [JMM*14] extended the rendering equation and path integral formulation to the transient domain, enabling Monte Carlo simulation of time-resolved transport. Jarabo et al. [JMM*14] also proposed density estimation and importance sampling strategies that exploit the extra degree of freedom in participating media. Subsequent work developed photon-tracing approaches for transient transport [Bit16b; HDJJ24; LJJ22; Mar13; MGJ*19; MJGJ17], which primarily target scenes dominated by participating media. Jarabo and Arellano [JA18] extended the transient path integral to vector (polarized) transport, Royo et al. [RGMJ22] derived a reduced path space for importance sampling NLOS scenes, and differentiable [YKC*21] and Doppler [KJGP23] extensions have also been explored. For surface-dominated scenes, naively applying steady-state methods treats the travel-time constraint as a rejection filter. Pediredla et al. [PVG19] addressed this with ellipsoidal path connections (EPC), which construct paths of a desired travel time by intersecting an ellipsoid with scene geometry. This directly importance samples the travel-time constraint but couples the per-path cost to scene complexity through the required ellipsoid–BVH traversal. Our method replaces the geometric intersection with a hierarchical range query over stored light vertices, decoupling time-gating cost from triangle count.

Photon mapping and many-lights. Photon mapping [Jen01] and many-light methods [DKH*14] both approximate global illumination by tracing many light subpaths and storing the resulting vertices as either photons or virtual point lights (VPLs) [Kel97]. By organizing the subpath vertices into hierarchical data structures, these methods are likewise able to form valid connections with a cost independent of the geometric complexity of the scene. Photon mapping does this via a density estimation search in a kd-tree, while Lightcuts [WFA*05] selects representative clusters via contribution upper bounds, reducing the number of evaluated connections. Stochastic Lightcuts (SLC) [Yuk19] replaces the deterministic set of VPLs with a stochastic traversal, descending the tree by choosing children proportional to their bounds. While extensions to photon beams and higher-dimensional primitives [BJ17; DJBJ19; JNSJ11; JZJ08] reduce variance, these only apply to participating media, and the increasingly elaborate range queries become expensive enough to diminish gains. Prior work has shown that time-gated transport paths follow sliced balls [LJJ22], but the resulting geometric queries carry a similar trade-off. We use SLC-style hierarchical traversal to efficiently query the light-cone shell (Sec. 4) via simple space-time *point* queries, incorporating temporal pruning and tightened bounds to steer samples toward time-valid connections.

Bidirectional path tracing. Bidirectional path tracing [LW93; Vea97] (BDPT) traces subpaths independently from emitters and sensors and connects them to form complete transport paths. Each connection defines a strategy indexed by subpath lengths, and multiple importance sampling (MIS) [Vea97] combines strategies to reduce variance. In standard BDPT, there is little control over which light subpaths a sensor vertex connects to. Probabilistic connections [PRDD15] address this by importance sampling connections from a shared pool of light vertices, and subsequent work introduced hierarchical selection [TH19], two-stage resampling [NID20a], resampling-aware MIS weights [NID20b], and subspace-based selection [SLW22]. These methods allow computing MIS weights that account for the resampling process, improving variance reduction in the steady-state setting. We focus on developing an efficient range search to quickly select time-valid connections by leveraging the space-time geometry of the light-cone shell. We currently use standard MIS weights to suppress singularities from the subpath sampling densities. Additionally incorporating the tree-traversal probability mass function (PMF) into the MIS weights, as these methods do for steady-state rendering, is a natural extension that we leave to future work.

3. Preliminaries

3.1. Time-of-Flight Rendering

The path integral formulation [Vea97] expresses the steady-state pixel measurement as

$$I = \int_{\Omega} f(\bar{x})\,\mathrm{d}\mu(\bar{x}), \tag{1}$$

where Ω is the space of all transport paths $\bar{x} = x_0 x_1 \cdots x_k$ with x_0 on a light source and x_k on the sensor, $\mathrm{d}\mu(\bar{x})$ is the area-product measure, and $f(\bar{x})$ is the measurement contribution of the path.

A time-of-flight sensor does not integrate over all paths indiscriminately; it is sensitive only to photons whose travel time falls within a specific interval. Following Pediredla et al. [PVG19], this can be modeled by incorporating a *travel-time* importance function W into the path integral:

$$I_\tau = \int_{\Omega} W(\tau(\bar{x}))\, f(\bar{x})\,\mathrm{d}\mu(\bar{x}), \tag{2}$$

where the optical path length is $\tau(\bar{x}) = \sum_{i=0}^{k-1} \eta_i \,\|x_{i+1} - x_i\|$, with η_i being the refractive index of the i-th segment. We set the speed of light $c = 1$ throughout, so optical path length and travel time are interchangeable and we denote both by τ. Fig. 2 shows how a 2D scene in space can be extended to 3D in space-time.

A common choice for W is a box function over a gate interval $[\tau_{\min}, \tau_{\max}]$:

$$W(\tau(\bar{x})) = \begin{cases} 1 & \text{if } \tau_{\min} \le \tau(\bar{x}) \le \tau_{\max}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Under this gating function, only paths whose travel time falls within the interval contribute to the measurement. The central challenge of time-gated rendering is that the constraint $\tau_{\min} \le \tau(\bar{x}) \le \tau_{\max}$ couples all vertices of a path through their cumulative distances, making it difficult to importance sample efficiently.

3.2. Many-Light Methods

Many-light methods compute Eq. (1) by tracing light subpaths from emitters and storing the resulting vertices as *virtual point lights* (VPLs) [Kel97]. Given a pool of N stored light vertices, the illumination at a shade point q is expressed as a sum

$$L(q) = \sum_{j=1}^{N} \phi_j\, f_j(q), \tag{4}$$

where ϕ_j is the intensity of light vertex j (subpath throughput divided by its sampling density) and $f_j(q)$ is the connection factor incorporating the BSDFs at both endpoints, the geometry term, and visibility. Evaluating this sum exactly is prohibitively expensive when N is large.

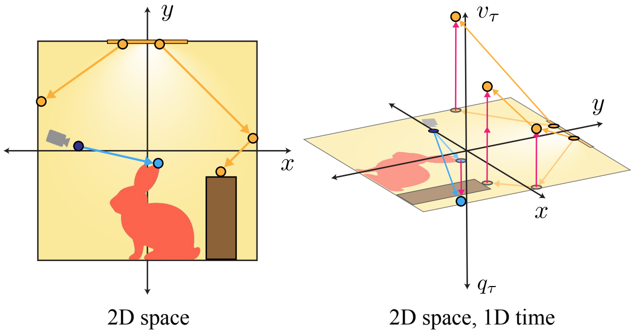

Figure 2: *Lifting a scene from 2D space to 3D space-time.* ***Left:*** *A light path (yellow) starts from the emitter and scatters through the scene; a camera subpath (blue) extends from the sensor through to the shade point.* ***Right:*** *Each vertex is lifted along the vertical axis (pink arrows) by its accumulated travel time. Note how emitter and camera vertices lie on the plane at $v_\tau = q_\tau = 0$. Sensor vertices are plotted downward for visual separation, but q_τ remains a positive accumulated quantity like v_τ.*

Lightcuts [WFA*05] approximates this sum by organizing the VPLs into a balanced binary tree and selecting a small subset of representative clusters reducing the number of connections to be evaluated. Stochastic Lightcuts (SLC) [Yuk19] replaces the deterministic cut with a stochastic traversal. Starting from the root, the traversal descends the tree by choosing between the two children $\mathcal{N}_L$ and $\mathcal{N}_R$ of the current node $\mathcal{N}$ with probability proportional to their contribution upper bounds:

$$P(\mathcal{N}_L) = \frac{B(\mathcal{N}_L, q)}{B(\mathcal{N}_L, q) + B(\mathcal{N}_R, q)}. \tag{5}$$

The traversal terminates at a leaf, yielding a single light vertex j with selection probability $P(j \mid q) = \prod_d P(\text{child taken at level } d)$. Multiple independent traversals can be performed per shade point to reduce variance. The estimate is unbiased as long as every vertex with nonzero contribution has a nonzero selection probability [Yuk19]; the tightness of the bound determines variance.

The SLC framework assumes steady-state rendering: the bound $B(\mathcal{N}, q)$ depends on spatial, material, and power factors but has no notion of travel time. Incorporating the time-gate constraint into this hierarchical selection process requires understanding the geometric structure of valid connections in the joint space of position and travel time, which we develop in the following sections.

4. Light Cones in Time-Gated Rendering

Time-gated rendering requires paths whose total optical path length falls within a narrow interval $[\tau_{\min}, \tau_{\max}]$. In this section, we show that the set of valid connections for a given sensor vertex has a natural geometric structure—a light cone in the space-time domain—and that this structure unifies existing approaches while motivating a different strategy based on range queries.

4.1. The Light Cone

Consider a shade point q (the endpoint of a sensor subpath) with accumulated travel time q_τ, and a light vertex v with accumulated travel

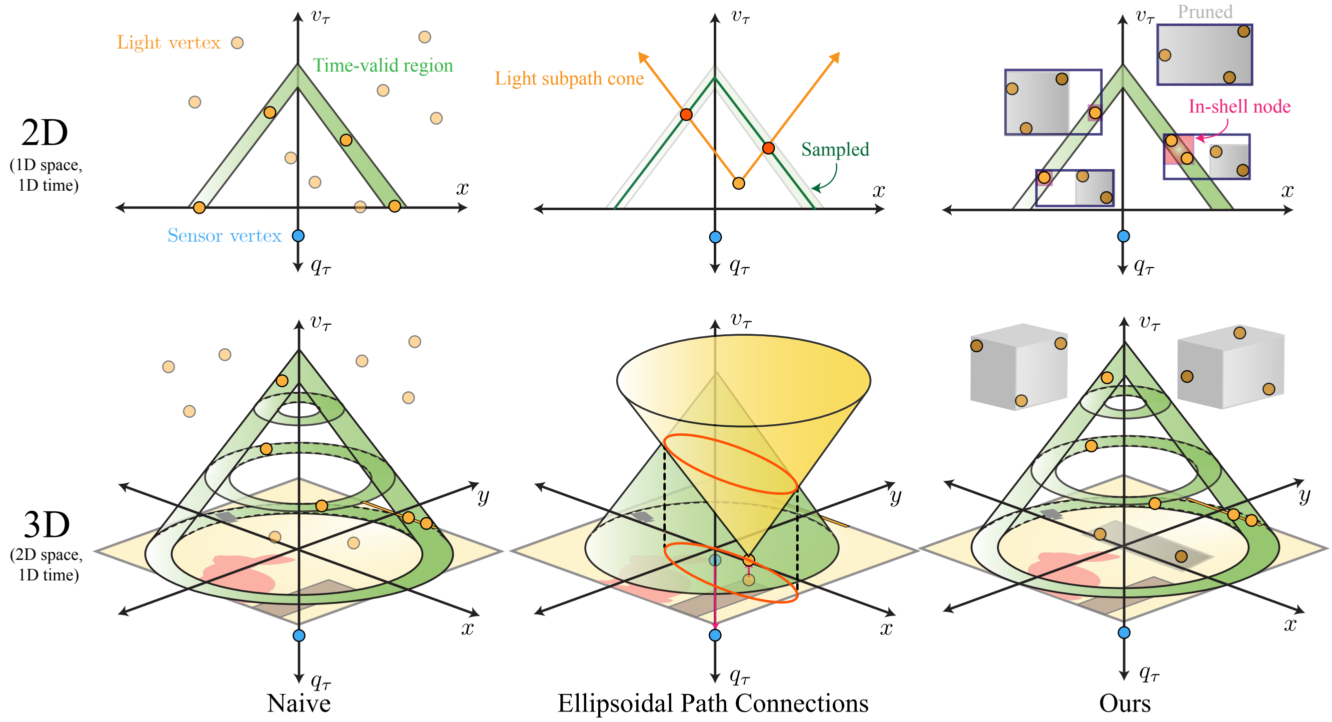

Figure 3: *Comparison of time-gated connection strategies in 2D (1D space + 1D time, top) and 3D (2D space + 1D time, bottom). Spatial axes show displacement from the sensor vertex (origin); light and sensor vertices are endpoints of independently traced subpaths from emitters and the sensor, respectively.* ***Left:*** *Steady-state methods sample the full space and rejection sample the LCS, discarding vertices outside the shell.* ***Center:*** *EPC samples a time and then extends a light subpath by one segment to an intersection with the light cone. This traces a light subpath cone centered at the light vertex, representing straight-line propagation from it. The intersection of the two light cones yields the set of time-valid connection points—an ellipsoidal region (in red). EPC then intersects this ellipsoid with scene geometry to find a vertex that lies on a surface, at a cost that scales with scene complexity.* ***Right:*** *Our method organizes light vertices into a space-time hierarchy and prunes entire nodes against the light-cone shell, steering samples toward the LCS without intersecting scene geometry.*

time v_τ. A direct connection between them produces a complete path whose total optical path length is

$$\tau = v_\tau + \|v - q\| + q_\tau. \tag{6}$$

Note that this is a sum of Euclidean spatial distance and scalar accumulated travel times, not a single distance in 4D space-time. This path satisfies the time-gate whenever $\tau_{\min} \le \tau \le \tau_{\max}$, which constrains the connection distance to the interval

$$\tau_{\min} - v_\tau - q_\tau \;\le\; \|v - q\| \;\le\; \tau_{\max} - v_\tau - q_\tau. \tag{7}$$

The inner and outer radii of the valid region are therefore

$$r_{\min} = \max\big(0,\; \tau_{\min} - v_\tau - q_\tau\big), \tag{8}$$
$$r_{\max} = \tau_{\max} - v_\tau - q_\tau, \tag{9}$$

with no valid connections when $r_{\max} < 0$, i.e. when $v_\tau + q_\tau > \tau_{\max}$.

First consider a single target travel time τ^*. The condition $\tau = \tau^*$ constrains the connection distance to $\|v - q\| = \tau^* - v_\tau - q_\tau$. Thus, a light vertex at position v must have accumulated travel time

$$v_\tau = \tau^* - q_\tau - \|v - q\|. \tag{10}$$

As v_τ increases, the valid connection distance shrinks monotonically until it reaches zero at $v_\tau = \tau^* - q_\tau$, beyond which no valid connection exists. The locus of valid connections traces out a cone in the joint space. We refer to this as a *light cone* for time-gated rendering, in analogy with the light cone of special relativity. Concretely, ours is the *past* light cone of the photon's arrival at the shade point: a future light cone opens with increasing time, but the past cone converges on its apex, so ours narrows as the accumulated travel time v_τ grows and less budget remains for the connection.

As in special relativity, the cone can be centered at either endpoint; we adopt the shade-point perspective throughout, querying which light vertices fall within the cone. In the accompanying diagrams (Fig. 2, Fig. 3), we plot sensor vertices downward along the vertical axis to visually distinguish the two subpaths; q_τ nonetheless represents a positive accumulated travel time.

A single target travel time defines a cone surface, but in practice the time-gate spans an interval $[\tau_{\min}, \tau_{\max}]$. Each boundary defines its own cone, and the valid region is the volume between them: a *shell* bounded by the $\tau_{\max}$ cone on the outside and the $\tau_{\min}$ cone on the inside. We call this the *light-cone shell* (LCS). A light vertex v produces a time-valid connection to the shade point q if and only if $\tau_{\min} \le \tau \le \tau_{\max}$, i.e. it lies within this shell (top-right of Fig. 3).

Fig. 3 illustrates the light-cone shell with increasing spatial dimension. In the full 4D (3D space, 1D time) setting, cross-sections at fixed v_τ are spherical shells centered at q; all share the same thickness $\tau_{\max} - \tau_{\min}$, but their radii shrink with increasing v_τ as more of the travel-time budget is consumed.

4.2. Existing Approaches

Steady-state methods sample paths over the entire space and rejection sample the LCS, discarding those outside the shell (Fig. 3, left). For narrow gates this wastes the vast majority of samples. Ellipsoidal path connections (Fig. 3, middle) take a fundamentally different approach, which we now reinterpret through the light-cone framework.

4.2.1. Ellipsoidal Connections as Cone Intersections

The light-cone shell describes where *existing* light vertices must lie to form time-valid connections. But a connection need not terminate at an existing vertex—one can extend a subpath by tracing an additional segment to a new vertex e on scene geometry, and then connect from there. Ellipsoidal path connections take exactly this approach. We now show that the ellipsoid arises naturally as the intersection of two light cones.

Extending a subpath in space-time. Consider a light vertex (v, v_τ) in the space-time diagram. If we extend the light subpath by one segment to a new point e, the segment has length $\|e - v\|$, so e sits at accumulated travel time

$$e_\tau = v_\tau + \|e - v\|\,. \tag{11}$$

As e varies over all spatial positions, this traces a cone opening upward from (v, v_τ) (top-middle of Fig. 3). We call this the *light subpath cone*: the locus of all points reachable by extending the light subpath by one segment.

Intersection with the light cone. Given a light vertex (v, v_τ) and a shade point (q, q_τ), EPC first samples a target travel time $\tau^* \in [\tau_{\min}, \tau_{\max}]$. This target τ^* defines a light cone from q. A vertex e lies on this cone when

$$e_\tau = \tau^* - q_\tau - \|e - q\|\,. \tag{12}$$

For the connection through e to achieve the target travel time, e must lie on *both* the light subpath cone (Eq. (11)) and the light cone from q (Eq. (12)). Setting these equal:

$$v_\tau + \|e - v\| = \tau^* - q_\tau - \|e - q\|\,, \tag{13}$$

which rearranges to

$$\|e - v\| + \|e - q\| = \tau^* - v_\tau - q_\tau \equiv \tau_e\,. \tag{14}$$

This is the equation of a prolate ellipsoid $\mathcal{E}(v, q, \tau_e)$ with foci at v and q. The ellipsoid is the intersection of the light subpath cone with the light cone from q (Fig. 3).

Since e must also lie on scene geometry $\mathcal{M}$, EPC intersects this ellipsoid with the scene, which is represented as a BVH, to sample the auxiliary vertex. Each sampled τ^* produces a different ellipsoid, and the ellipsoid must be intersected against the scene BVH to find candidate primitives. While BVH traversal is logarithmic on average, the query is output-sensitive: the ellipsoid may overlap many primitives in regions of high local triangle density, and each candidate requires an analytical ellipse–polygon intersection [PVG19, Sec. 4]. Pediredla et al. [PVG19, Sec. 7] note that this limits scalability to large meshes.

4.3. Light-Cone Shell Queries

Our key observation is that the LCS can be queried *directly* from stored light subpaths, without any geometric construction. The membership test of Eq. (7) requires only quantities already carried by stored light subpaths—the accumulated travel times and the Euclidean connection distance—making it $O(1)$ and independent of scene geometry. The problem thus reduces to efficient search: among a large pool of stored light vertices, rapidly identify and importance sample those within the shell. We develop the machinery for this in the following section.

5. Accelerating Light-Cone Shell Queries

Sec. 4.3 reformulated time-gated rendering as a search problem: among all of the stored light vertices, identify those that fall within the light-cone shell for a given sensor vertex. We now make this precise and develop the machinery to efficiently do so. We first present the estimator and acceleration structure for a single shade point; Sec. 6 embeds this into a BDPT framework where the stored vertices are subpath endpoints at varying depths, combined via MIS.

5.1. Per-Shade-Point Estimation

Suppose we trace M light subpaths from the emitters, producing a pool of N light vertices. Each vertex v_j occupies a point $(v_j, v_{j,\tau}) \in \mathbb{R}^4$ and carries a weight $\phi_j = f(\bar{v}_j)/p(\bar{v}_j)$, the subpath throughput divided by its sampling density.

Consider a shade point q with accumulated optical path length q_τ. Connecting it to light vertex j produces a complete path with total optical path length

$$\tau_j = v_{j,\tau} + \left\|v_j - q\right\| + q_\tau. \tag{15}$$

The time-gated contribution at q from the entire light vertex pool is the sum

$$L_\tau(q) = \sum_{j=1}^{N} W(\tau_j)\,\phi_j f_j(q), \tag{16}$$

where $f_j(q)$ is the connection factor (BSDFs at both endpoints, geometry term, and visibility) and $W(\tau_j)$ zeroes out connections whose total travel time falls outside the gate $[\tau_{\min}, \tau_{\max}]$ or equivalently, those whose light vertex lies outside the LCS.

Evaluating this sum exactly requires up to N connections to light vertices which is prohibitively expensive. We instead estimate it by sampling K light vertices from a PMF $P(j \mid q)$ over the pool, yielding the following estimator

$$\hat{L}_\tau(q) = \frac{1}{K} \sum_{k=1}^{K} \frac{W(\tau_{j_k})\,\phi_{j_k} f_{j_k}(q)}{P(j_k \mid q)}, \tag{17}$$

where j_k indexes the k-th selected vertex. This estimator is unbiased for any PMF with $P(j \mid q) > 0$ whenever the j-th light vertex has nonzero contribution. Its variance is minimized when $P(j \mid q)$ is proportional to the magnitude of the j-th summand—i.e., when the sampling probability reflects both the vertex's intensity and whether it lies within the LCS.

The challenge is constructing a good PMF efficiently. A naive approach would evaluate the time-gate and contribution for every vertex, normalize, and sample—but this is $O(N)$ per shade point, no better than evaluating the sum directly. Hierarchical importance sampling reduces this to $O(\log N)$ by organizing the vertices into a tree and making incremental selection decisions at each node: at every level, we choose between the two children in proportion to upper bounds on their contribution to the sum. This results in a PMF over the sampled light vertices.

The key insight is that the light-cone shell provides a powerful additional signal during this traversal. At each node, we can test whether the node's spatiotemporal extent overlaps the LCS and, if so, estimate what fraction of its descendants are time-valid. Subtrees that lie entirely outside the shell are pruned (assigned zero probability), while surviving nodes receive tightened bounds that reflect their overlap with the shell. This steers samples toward time-valid, high-contribution vertices simultaneously.

5.2. 4D Hierarchy Construction

We organize the N light vertices into a balanced binary tree following the Stochastic Lightcuts (SLC) framework [Yuk19] where individual vertices reside in leaves. Each inner node $\mathcal{N}$ stores aggregate quantities summarizing its descendants: a bounding box over their positions, total power $\Phi_{\mathcal{N}}$, and material bounds used to upper-bound the connection factor. These quantities define an upper bound $B(\mathcal{N}, q)$ on the contribution of the entire subtree to a given shade point (see Sec. 3 for details).

For time-gated rendering, position alone is insufficient. We therefore augment each node with a travel-time interval $[\mathcal{N}_\tau^-, \mathcal{N}_\tau^+]$ bounding the accumulated travel times of all descendant vertices, extending the hierarchy to four dimensions. Geometrically, a node's 4D extent—the Cartesian product of its spatial bounding box and travel-time interval—is a rectangular region in the light-cone diagram, and the pruning and bounding strategies that follow operate by testing this region against the LCS. We split along the dimension of greatest extent among all four.

5.3. Pruning via the Light-Cone Shell

The hierarchy enables a cheap test that discards entire subtrees whose descendants cannot produce time-valid connections. Consider a shade point q with accumulated optical path length q_τ, and a tree node $\mathcal{N}$ with spatial bounding box and travel-time interval $[\mathcal{N}_\tau^-, \mathcal{N}_\tau^+]$. From Eq. (15), the total optical path length of any connection to a descendant of $\mathcal{N}$ lies in the range

$$\tau \in \left[\mathcal{N}_\tau^- + d_{\min}(\mathcal{N}, q) + q_\tau,\ \ \mathcal{N}_\tau^+ + d_{\max}(\mathcal{N}, q) + q_\tau\right], \tag{18}$$

where $d_{\min}(\mathcal{N}, q)$ and $d_{\max}(\mathcal{N}, q)$ are the minimum and maximum Euclidean distances from q to the axis-aligned spatial bounding box of $\mathcal{N}$. Node $\mathcal{N}$ can be pruned whenever this range does not overlap the time-gate $[\tau_{\min}, \tau_{\max}]$:

$$\mathcal{N}_\tau^+ + d_{\max} + q_\tau < \tau_{\min} \quad \text{or} \quad \mathcal{N}_\tau^- + d_{\min} + q_\tau > \tau_{\max}. \tag{19}$$

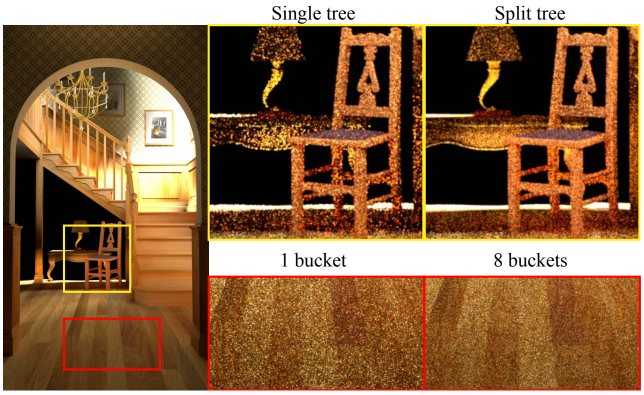

Figure 4: *Scene illuminated with an area emitter. Equal-time (30s) comparisons of:* ***Top:*** *single tree vs. split trees (Sec. 5.4).* ***Bottom:*** *1 vs. 8 temporal power buckets (Sec. 5.5.2).*

This has a direct geometric interpretation: a node is pruned when its spatiotemporal bounding box lies outside the light-cone shell. We see this in the top-right of Fig. 3 through the grayed out boxes.

The test is conservative—it treats the travel time and distance ranges as independent, so some nodes whose descendants are all time-invalid may survive, but no valid nodes are ever discarded, preserving unbiasedness.

5.4. Separating Vertex Populations

The pruning test of the preceding section applies uniformly to any node in the hierarchy. However, different vertex populations have different characteristics in the joint space, and separating them enables tighter contribution bounds during traversal.

Consider emitter vertices: all share $v_\tau = 0$ (Fig. 2), lying on a hyperplane in space-time. For these vertices, the time-gate constraint reduces to a simple distance condition, and the overlap between the light-cone shell and a node's spatial extent can be computed directly. Vertices that have scattered at least once, by contrast, vary in both position and accumulated travel time, requiring a different strategy to resolve their overlap with the shell.

Mixing the two populations in a single tree forfeits two opportunities: shared ancestor nodes acquire inflated travel-time intervals from the scattered vertices, loosening the pruning test of Eq. (19), while the simple distance-based bound available for emitter vertices cannot be applied to nodes containing both types. We therefore maintain two separate trees:

- A **3D spatial tree** over vertices with $v_\tau = 0$, using only spatial bounding boxes.
- A **4D spatiotemporal tree** over vertices distributed in both position and accumulated travel time.

We query both trees independently at each shade point, and their contributions are combined as described in Sec. 6. For scenes with point emitters, we evaluate them through next-event estimation. We compare renderings using the split-tree architecture against a single mixed tree in Fig. 4 (top). The bounding strategies enabled by this separation are developed next.

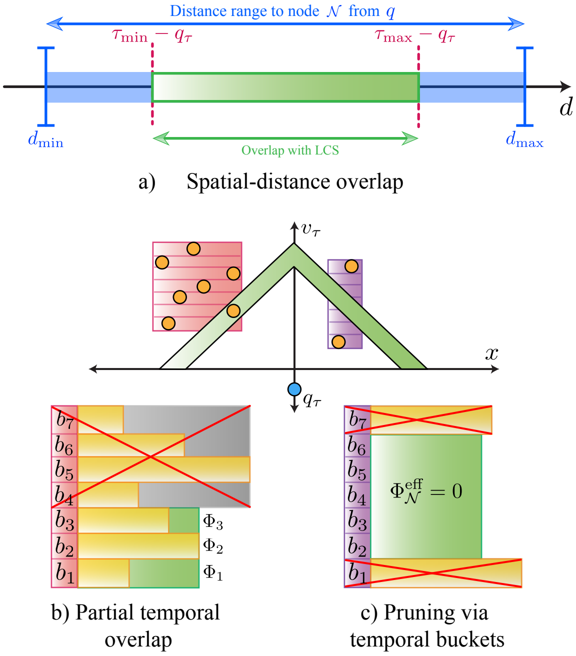

Figure 5: *Tighter bounds via LCS overlap.* ***a)*** *For vertices with $v_\tau = 0$, effective power scales by the overlap of the time-valid interval (green) with the node's distance range (blue).* ***b–c)*** *For vertices varying in 4D, per-node temporal power buckets resolve which portions of the travel-time range intersect the shell. In* ***b)****, only buckets b_1–b_3 contribute. In* ***c)****, binary pruning keeps the node but all power resides in buckets outside the shell, yielding $\Phi^{\text{eff}}_{\mathcal{N}} = 0$.*

5.5. Tighter Bounds via Light-Cone Shell Overlap

Pruning provides a binary decision: a node either survives or is discarded entirely. But among the descendants of a surviving node, not all necessarily produce time-valid connections. The standard SLC contribution upper bound $B(\mathcal{N}, q)$ incorporates material, geometric, and power factors, each maximized over the node's extent [Yuk19]. Of these, the material and geometric factors already vary per shade point. We apply the same principle to the power factor: rather than using the total node power $\Phi_{\mathcal{N}}$ regardless of the time gate, we compute an *effective power* $\Phi^{\text{eff}}_{\mathcal{N}}$ that reflects the fractional overlap between the node's extent and the light-cone shell for the current shade point. This steers traversal toward regions of the tree with greater LCS overlap.

The core idea is the same for both trees: estimate what fraction of a node's descendants could plausibly produce time-valid connections, and scale the power accordingly. The split-tree architecture (Sec. 5.4) enables tailored strategies for each population.

5.5.1. Adaptive 3D Spatial Bounds

For vertices in the 3D spatial tree, all descendants share $v_\tau = 0$. The total travel time of a connection simplifies to

$$\tau = \|v - q\| + q_\tau, \tag{20}$$

so the connection distance is the sole unknown. The LCS cross-section at $v_\tau = 0$ is a single spherical shell: the valid connection distances are exactly $[\tau_{\min} - q_\tau,\ \tau_{\max} - q_\tau]$.

The bounding box of $\mathcal{N}$ provides a range $[d_{\min}, d_{\max}]$ of possible distances to q. The overlap between this range and the time-valid distance interval is

$$d \in \big[\max(d_{\min},\ \tau_{\min} - q_\tau),\ \min(d_{\max},\ \tau_{\max} - q_\tau)\big]. \tag{21}$$

When this subinterval is strictly smaller than $[d_{\min}, d_{\max}]$, only a fraction of the node's spatial extent can produce time-valid connections. Assuming a uniform spatial distribution of vertices within the bounding box, the estimated effective power becomes

$$\Phi^{\text{eff}}_{\mathcal{N}} = \Phi_{\mathcal{N}} \cdot \frac{\min(d_{\max}, \tau_{\max} - q_\tau) - \max(d_{\min}, \tau_{\min} - q_\tau)}{d_{\max} - d_{\min}}. \tag{22}$$

This scales the node's power by the fractional overlap between the shell and the node's extent (Fig. 5a), assigning more weight to nodes whose spatial extent has greater overlap with the time-valid region.

5.5.2. Adaptive 4D Spatiotemporal Bounds

For vertices in the 4D spatiotemporal tree, both position and accumulated travel time v_τ vary across descendants of a node. Here, the shell cross-section itself varies with v_τ, so we must first resolve which portions of the node's travel-time range intersect the shell.

We do this by partitioning each node's travel-time interval $[\mathcal{N}^-_\tau, \mathcal{N}^+_\tau]$ into B uniform *temporal power buckets*. Each bucket b covers a subinterval $[\mathcal{N}^-_\tau[b], \mathcal{N}^+_\tau[b]]$ of accumulated travel time—geometrically, a horizontal slab in the light-cone diagram (Fig. 5b). During tree construction, each vertex is assigned to the bucket containing its v_τ, and the total power Φ_b of each bucket is accumulated and stored at the node, requiring $O(B)$ additional storage.

At query time, given a shade point q, we identify which buckets can contain time-valid vertices. From Eq. (7), a bucket b can contain time-valid vertices only if its travel-time range is compatible with the node's distance range:

$$\mathcal{N}^-_\tau[b] + d_{\min} + q_\tau \le \tau_{\max} \quad \text{and} \quad \mathcal{N}^+_\tau[b] + d_{\max} + q_\tau \ge \tau_{\min}. \tag{23}$$

The effective power for the node is then the sum over only the valid buckets:

$$\Phi^{\text{eff}}_{\mathcal{N}} = \sum_{b \,\in\, \text{valid}} \Phi_b. \tag{24}$$

This is inexpensive: $d_{\min}$ and $d_{\max}$ are computed once for the node's spatial bounding box, and identifying the range of valid buckets requires only a constant-time index computation. Within each valid bucket, one could further refine by computing the fractional distance overlap as in Sec. 5.5.1, scaling each Φ_b by the overlap between the bucket's valid distance range and $[d_{\min}, d_{\max}]$; we leave analyzing this refinement to future work.

The number of buckets B controls a precision–storage trade-off. With $B = 1$ (no buckets), the bound reduces to the total node power whenever the node survives pruning. As B increases, the per-bucket travel-time range narrows and the summed power more closely reflects the true time-valid power in the node. Fig. 5c shows a case where binary pruning keeps a node but all power resides in buckets outside the shell, yielding $\Phi^{\text{eff}}_{\mathcal{N}} = 0$—a refinement impossible without the histogram. In practice, $B = 8$ provides a good trade-off (Fig. 4 bottom).

6. BDPT with Light-Cone Shell Queries

We now show how light-cone shell queries can be incorporated into bidirectional path tracing (BDPT). Each connection pairs a shade point $q^{(t)}$ with a light vertex $v^{(s)}$, defining an (s,t) strategy. Algorithm 1 summarizes the complete algorithm.

6.1. Per-Pixel Estimate

The illumination at a shade point $q^{(t)}$ from the stored light vertex pool is estimated via Eq. (17). To obtain a pixel measurement, we multiply by the sensor subpath throughput weight ω_t and an MIS weight $w_{s,t}$. Here ω_t is the sensor side analog of ϕ_j: the accumulated sensor throughput divided by its sampling density up to depth t. The per-sample pixel contribution is

$$\hat{I}_{s,t} = w_{s,t} \cdot \omega_t \cdot \frac{W(\tau_j)\, \phi_j\, f_j(q^{(t)})}{P(j \mid q^{(t)})}, \tag{25}$$

where s is the subpath depth of the selected light vertex j. For each shade point $q^{(t)}$ along the sensor subpath, we perform K independent tree queries. The complete pixel estimate is:

$$\hat{I}_\tau = \sum_{t=2}^{|\bar{q}|} \frac{1}{K} \sum_{k=1}^{K} \hat{I}_{s_k,t}, \tag{26}$$

where s_k varies per sample since the vertex pool contains light vertices at different subpath depths.

We omit the boundary strategies $s = 0$ (unidirectional path tracing), $t = 0$ (light tracing), and $t = 1$ (direct sensor connection). These strategies trace complete subpaths without a controllable connection point, preventing importance sampling of the travel-time constraint. In scenes requiring caustic transport, they can be recovered by tracing a fresh emitter subpath outside the tree framework or combining with steady-state methods as discussed in Sec. 8.

6.2. MIS Weighting

Each connection corresponds to a BDPT (s,t) strategy. We combine strategies using the power heuristic [Vea97]:

$$w_{s,t}(\bar{x}) = \frac{p_{s,t}(\bar{x})^\beta}{\sum_{s',t'} p_{s',t'}(\bar{x})^\beta}, \tag{27}$$

where $\beta = 2$ and $p_{s,t}(\bar{x})$ is the area-product density of generating the full path $\bar{x}$ via strategy (s,t), evaluated as though the light vertices were sampled by ordinary path tracing rather than by tree traversal.

This weighting deliberately omits the tree-traversal PMF $P(j \mid q^{(t)})$. The full sampling density of a connection includes this PMF as a factor, and the estimator correctly divides by it, so the estimate is unbiased regardless of the MIS weight. The resulting weights are suboptimal in the variance-minimization sense but remain valid, and they retain the geometry-term singularity suppression of standard BDPT MIS.

7. Results

We evaluate our method against two baselines: standard BDPT and EPC. We build upon the Mitsuba 0.5.0 [Jak13] code from Pediredla et al. [PVG19]. All experiments were conducted on a 16-core AMD

Algorithm 1 Time-gated BDPT with light-cone shell queries.

```
// Phase 1: Light tracing
Trace M light subpaths from emitters
Store each vertex with position, v_τ, weight φ_j
Separate based on whether they are a delta in time or space

// Phase 2: Hierarchy construction
Build 3D spatial tree over delta-in-time vertices
Build 4D spatiotemporal tree over vertices varying in space and time

// Phase 3: Sensor tracing and connection
for each pixel do
    Trace sensor subpath; record q^(t), q_τ, ω_t at each vertex
    for each shade point q^(t) with non-delta scattering do
        for k = 1, ..., K do
            Query appropriate tree
            if leaf v^(s) reached (not a dead node) then
                Evaluate f_j(q^(t)), test visibility
                Accumulate Î_{s,t}
            end if
        end for
    end for
end for
```

Table 1: *Image quality as a function of the number of light vertices used in our method at equal render time (30s). relMSE values are in units of* 10^{-3}. *Best values per scene are* ***bolded***.

	Veach-Ajar		Staircase		Bedroom	
Vertices	relMSE ↓	ꟻLIP ↓	relMSE ↓	ꟻLIP ↓	relMSE ↓	ꟻLIP ↓
25k	22.60	0.2087	9.43	0.1750	46.90	0.2192
50k	21.93	0.1942	8.67	0.1625	**46.15**	0.2113
100k	**21.77**	0.1854	8.69	0.1543	47.32	**0.2082**
200k	22.35	**0.1788**	**8.60**	0.1536	50.80	0.2093
400k	23.69	0.1795	9.00	**0.1522**	52.69	0.2112
800k	25.59	0.1813	9.21	0.1541	57.93	0.2169
1.6M	27.29	0.1851	9.97	0.1584	69.05	0.2302

Ryzen 9 5900XT. The scenes in Fig. 6 are from Bitterli [Bit16a] but modified for time-gated rendering. Scene BVH build time is shared across all methods and excluded from timing; tree build time for our method is included. All reference images were rendered with BDPT.

7.1. Equal-Time Comparisons

Fig. 6 compares convergence on three scenes spanning 226k–1.5M triangles with narrow time gates (1–2% of scene size). Our method achieves 5.5×, 4.5×, and 9.3× relMSE improvement over BDPT at 30 s on Veach-Ajar, Bedroom, and Staircase, respectively. EPC produces higher variance at substantially longer render times (758 s–6600 s) due to the high per-path cost of the ellipsoidal intersection. Notably, EPC's render time on Bedroom (6600 s) is 7–9× longer than on the other scenes, reflecting the poor scaling of ellipsoidal intersection with triangle count (1.5M).

The improvement is largest on Staircase (9.3×), which uses

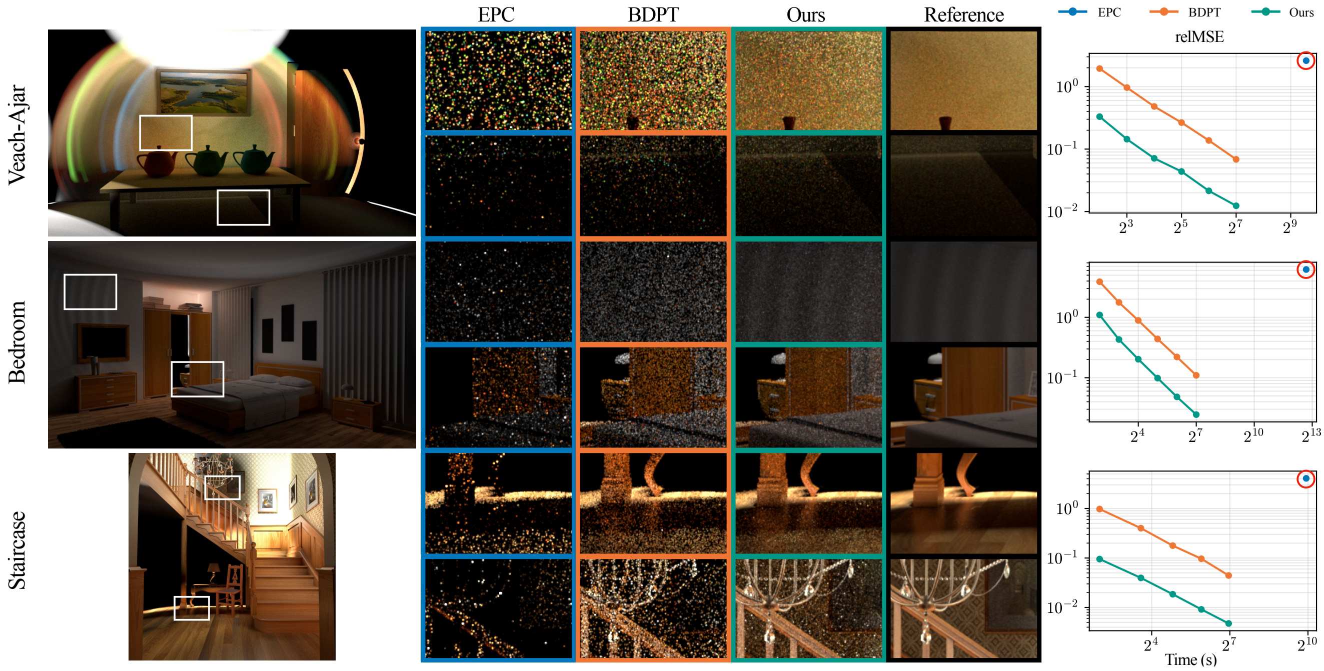

Figure 6: *Equal-time comparison of time-gated rendering on three scenes of varying complexity (VEACH-AJAR: 383k triangles; BEDROOM:* 1.5*M triangles; STAIRCASE:* 226*k triangles). VEACH-AJAR and BEDROOM use a collimated beam emitter with a* 2% *gate width; STAIRCASE uses an area emitter with a* 1% *gate width. Crops compare EPC, BDPT, and our method at* 30*s (BDPT and ours) alongside EPC at* 1 *sample per pixel (*758*s,* 6600*s, and* 977*s, respectively). Convergence plots show relMSE vs. time (s); the circled point marks EPC. Our method achieves* 5.5×, 4.5×, *and* 9.3× *relMSE reduction over BDPT at equal time on VEACH-AJAR, BEDROOM, and STAIRCASE, respectively.*

an area emitter whose vertices are distributed in both space and travel time. This engages both the 3D spatial and 4D spatiotemporal trees (Sec. 5.4), allowing the spatial tree's distance-based pruning to eliminate emitter vertices outside the light-cone shell. VEACH-AJAR and BEDROOM use a collimated beam emitter, for which first-bounce vertices are degenerate in position and travel time; these scenes rely on the 4D tree over vertices varying in space-time.

7.2. Parameter Sensitivity

Light vertex count. Table 1 reports image quality as a function of the number of stored light vertices at equal render time. At low vertex counts, the few time-valid vertices are reused across many shade points, producing the spatially correlated error characteristic of many-light methods, which we show in the supplemental. This correlation is reflected more clearly in ꟻLIP, which penalizes structured error, than in relMSE, which evaluates each pixel independently. As the vertex count increases both metrics reach their optimum in the range of 50k–400k vertices depending on the scene. Beyond this range, additional vertices yield diminishing returns in coverage while increasing tree construction time and per-query traversal cost, leaving fewer samples per pixel within the equal-time budget. The main results in Fig. 6 use 200k vertices.

Fig. 7 isolates the effect of gate width, triangle count, and material roughness on a Cornell box scene with a tessellated bunny. We normalize EPC and our method to BDPT.

Gate width. As the gate narrows, our method's advantage grows compared to BDPT, reaching over an order of magnitude at the narrowest gate (0.5 scene units, 0.03%). At the tightest gates, EPC outperforms our method: ellipsoidal connections can target paths of an exact desired travel time, whereas our method requires pre-traced light vertices to fall within the shell, which becomes vanishingly unlikely as the gate approaches a delta. However, this advantage is modest on the low-complexity scene used here (66 triangles); on scenes with higher triangle counts, EPC's per-path intersection cost would offset this gain. At wide gates, both EPC and our method become worse than BDPT: the gate admits most paths, so BDPT's rejection rate is low and the overhead of the tree and ellipsoidal intersection outweighs the benefit of targeted sampling.

Triangle count. EPC's relMSE increases sharply with triangle count, reflecting the scaling of ellipsoidal intersection. Our method stays flat compared to BDPT confirming that our approach decouples the cost of time-gating from scene complexity.

Specularity. Both methods show higher error at low roughness ($\alpha = 0.05$). For EPC, the auxiliary vertex introduces two edges that are unlikely to align with narrow specular lobes, a limitation noted by Pediredla et al. [PVG19]. For our method, the contribution bounds assume the highest material response, which becomes looser for near-specular BSDFs; tighter bounds for glossy materials have been explored in steady-state many-light methods and could be applied to our framework in future work.

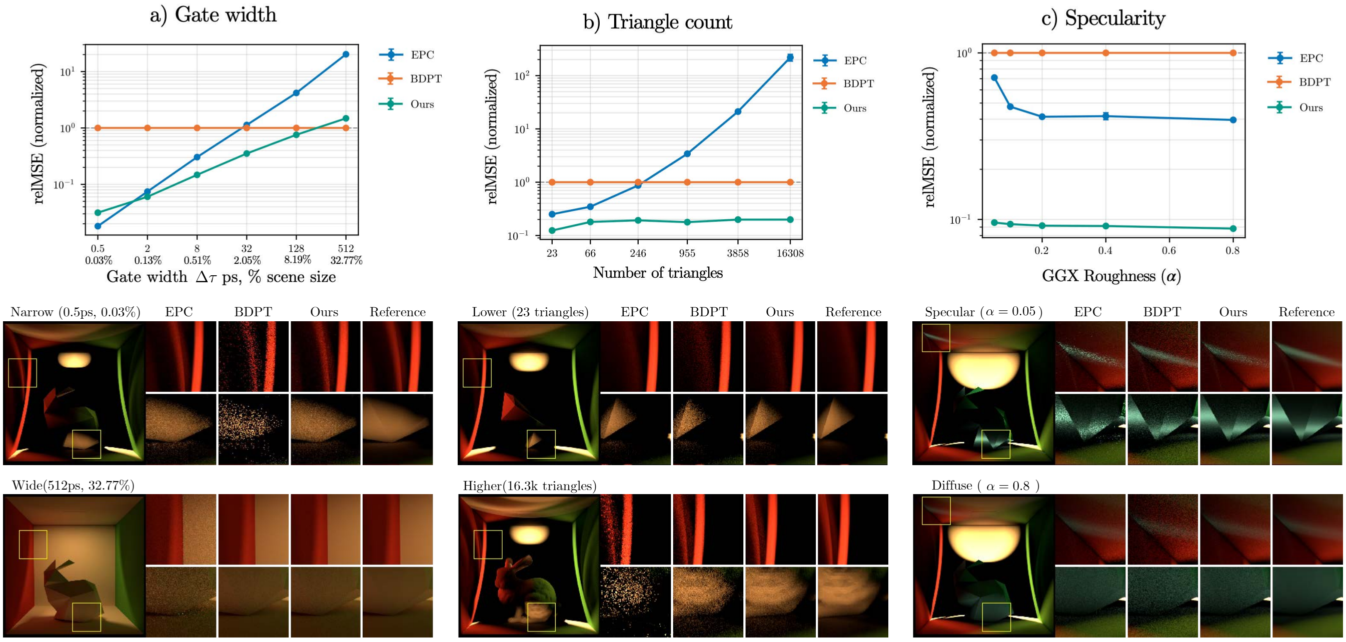

Figure 7: *Parameter sensitivity on a Cornell box scene with a tessellated bunny (66 triangles unless varied) at 30 s equal time with relMSE normalized to BDPT.* ***a)*** *Gate width sweep from* 0.5–512 *scene units* (0.03%–32.77% *of scene size).* ***b)*** *Triangle count sweep from* 23–16.3*k triangles.* ***c)*** *GGX roughness sweep from* $\alpha = 0.05$ *(near-specular) to* $\alpha = 0.8$ *(diffuse). Insets show crops at the extremes of each sweep.*

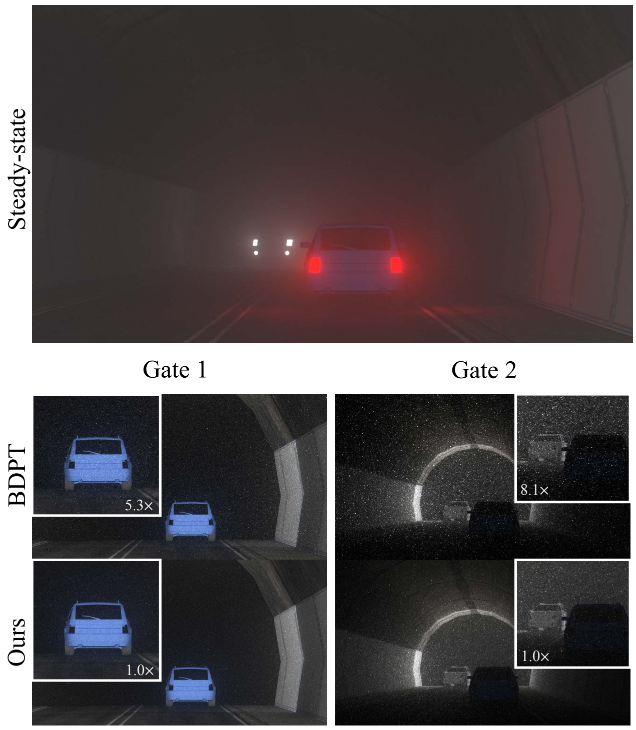

Figure 8: *Seeing through fog.* ***Top:*** *Steady-state render of a foggy tunnel scene; the far car is barely visible through the scattering medium.* ***Bottom:*** *Time-gated renders at two gate intervals targeting the near and far car, comparing BDPT and our method at equal time (*30 s*). Our method achieves* 5.3× *and* 8.1× *relMSE reduction on the two cropped regions.*

7.3. Applications

Time-gated imaging has many applications in computational photography and autonomous sensing. We demonstrate two use cases enabled by our method's efficiency on complex scenes.

Seeing through fog. Fig. 8 shows a foggy tunnel scene where a distant car is barely visible in the steady-state render. By time-gating at two different intervals targeting the near and far car, the multiply-scattered fog photons are rejected and the car becomes visible [GJBH19]. Our method achieves 5.3× and 8.1× relMSE reduction over BDPT at equal time (30 s) for the two gates, respectively.

Proximity detection. Time-gated sensors mounted on vehicles can detect objects within a specific distance range by rejecting photons outside a narrow travel-time window, functioning as proximity sensors. Simulating such sensors in dynamic scenes requires rendering a sequence of time-gated frames, each for a slightly different scene configuration. Fig. 9 shows a city scene with 2.4M triangles where we simulate measurements from a camera equipped on an automobile. We render two frames at different positions along the vehicle's trajectory, each time-gated to a narrow depth range. Our method produces visibly cleaner results than BDPT at equal time (10 s), and because its time-gating cost is independent of scene complexity, it is practical to simulate these sensors in realistic urban environments.

8. Limitations and Future Work

MIS weighting. Our MIS weights omit the tree-traversal PMF (Sec. 6.2). Incorporating selection probabilities into the weights, as steady-state methods do [NID20b; PRDD15; SLW22], results

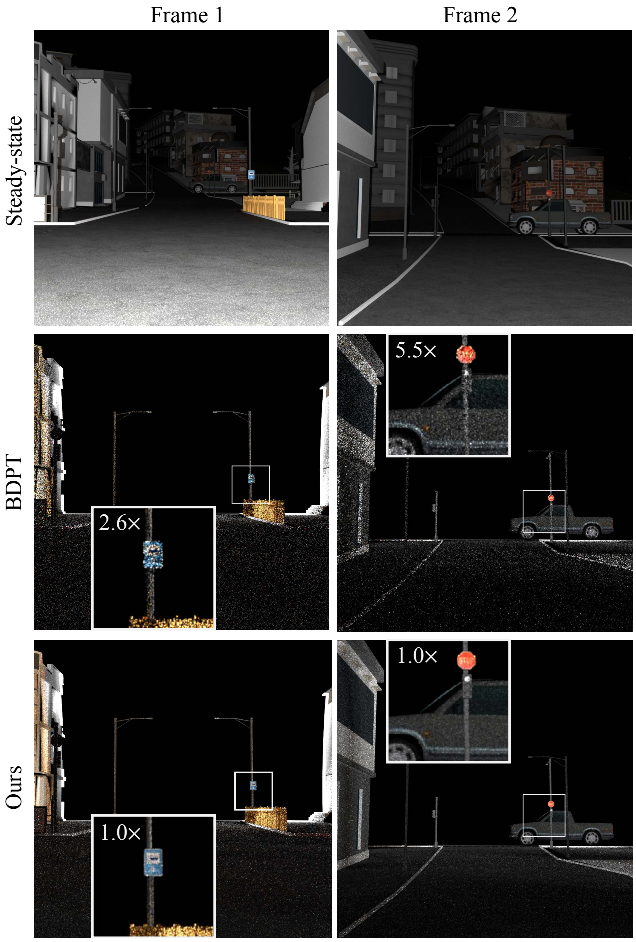

Figure 9: *Proximity detection camera. Using a city scene with* 2.4*M triangles, we simulate measurements from a camera equipped on a travelling automobile. From top to bottom we show steady-state renderings of 2 frames and then time-gated rendering with BDPT and our method. All images are rendered for* 10 s *with our method producing* 2.6× *and* 5.5× *lower relMSE for the two crops.*

in resampling-aware weights but increases the per-sample cost; exploring this trade-off is an interesting direction.

Geometry-term singularities. Many-light methods suffer from geometry-term singularities at short connection distances. We mitigate this through BDPT-style MIS weighting (Eq. (27)), but bias compensation via final gather [KK06] offers an alternative worth exploring in the time-gated setting.

Dead nodes. The conservative pruning test (Sec. 5.3) means some traversals reach leaves whose vertices are time-invalid, contributing nothing. The dead-node rate increases as the light-cone shell becomes thin relative to node extents. Temporal power buckets (Sec. 5.5.2) steer traversal away from these regions but cannot eliminate dead nodes entirely. Tighter bounding volumes such as oriented bounding boxes, or adaptive re-partitioning in regions where the shell is thin, could further reduce dead rates at the cost of more expensive traversal.

Very narrow gates. In realistic sensing applications, time gates are sufficiently wide that range searching over pre-traced vertices is more cost-effective than directly generating paths. As the gate approaches a delta function, however, the effective pool of time-valid vertices shrinks and our method's advantage diminishes. A progressive approach akin to progressive photon mapping [HOJ08] could address this regime by tracing light subpaths across iterations while narrowing the gate width, converging to a delta gate in the limit without explicit geometric intersection.

Near-specular materials and caustics. Our contribution bounds assume worst-case material response over each node's extent, which becomes increasingly loose for near-specular BSDFs. We also omit the $s = 0$, $t = 0$, and $t = 1$ BDPT strategies, which trace complete subpaths without a controllable connection point and therefore cannot importance sample the travel-time constraint via our framework. These strategies are important for caustic transport. Combining our method with VCM/UPS [GKDS12; HPJ12] could address both limitations: merging handles caustic paths that our connection-based approach misses, while our light-cone shell queries would accelerate the connection strategies that VCM/UPS shares with BDPT.

Other extensions. Our framework targets time-gated rendering, but other imaging modalities spread contributions across the temporal domain. For full transient rendering, the per-node power buckets could provide per-bin contribution weights during a single traversal, though whether this amortization offsets the loss of aggressive pruning is an open question. Continuous-wave ToF poses a harder case: the periodic modulation results in all paths contributing with varying weight. Separately, our proximity-detection results (Fig. 9) rebuild the tree each frame; for animated sequences, temporal coherence could amortize construction across frames.

Acknowledgments

We thank the anonymous reviewers for their helpful suggestions and the members of the Dartmouth Visual Computing Lab for valuable discussions. We are grateful to the creators of the 3D assets used in our scenes: the city scene by Alex Samusenko (BlenderKit), tunnel road by Low poly 3d Models (Sketchfab), and car meshes by Comrade1280 (Sketchfab) and MaX3Dd (Sketchfab). This work was supported by NSF award 2403122.

References

[Bit16a] Bitterli, B. *Rendering Resources*. 2016. url: https://benedikt-bitterli.me/resources/ 8.

[Bit16b] Bitterli, B. *Virtual Femto Photography*. 2016. url: https://benedikt-bitterli.me/femto.html 2.

[BJ17] Bitterli, B. and Jarosz, W. "Beyond points and beams: Higher-dimensional photon samples for volumetric light transport". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 36.4 (July 2017). issn: 0730-0301. doi: 10/gfznbr 2.

[DJBJ19] Deng, X., Jiao, S., Bitterli, B., and Jarosz, W. "Photon surfaces for robust, unbiased volumetric density estimation". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38.4 (July 2019). doi: 10/gf6rx9 2.

[DKH*14] Dachsbacher, C., Křivánek, J., Hašan, M., Arbree, A., Walter, B., and Novák, J. "Scalable realistic rendering with many-light methods". *Computer Graphics Forum* 33.1 (Feb. 1, 2014). issn: 0167-7055. doi: 10/f5twgd 2.

[GJBH19] Gruber, T., Julca-Aguilar, F., Bijelic, M., and Heide, F. "Gated2depth: real-time dense lidar from gated images". *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019). doi: 10/ggr9s8 1, 2, 10.

[GKDS12] Georgiev, I., Křivánek, J., Davidovič, T., and Slusallek, P. "Light transport simulation with vertex connection and merging". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 31.6 (Nov. 2012). issn: 0730-0301. doi: 10/gbb6q7 11.

[HDJJ24] He, Q., Du, D., Jiang, H., and Jin, X. "Darts: diffusion approximated residual time sampling for time-of-flight rendering in homogeneous scattering media". *ACM Transactions on Graphics* 43.6 (Nov. 2024). issn: 0730-0301. doi: 10/q9nj 2.

[HOJ08] Hachisuka, T., Ogaki, S., and Jensen, H. W. "Progressive photon mapping". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 27.5 (Dec. 1, 2008). issn: 07300301. doi: 10/cn8h39 11.

[HPJ12] Hachisuka, T., Pantaleoni, J., and Jensen, H. W. "A path space extension for robust light transport simulation". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 31.6 (Jan. 11, 2012). issn: 0730-0301. doi: 10/gbb6n3 11.

[JA18] Jarabo, A. and Arellano, V. "Bidirectional rendering of vector light transport". *Computer Graphics Forum* 37.6 (Sept. 1, 2018). issn: 0167-7055. doi: 10/gdvk6m 2.

[Jak13] Jakob, W. *Mitsuba Renderer*. 2013. url: http://www.mitsuba-renderer.org 8.

[Jen01] Jensen, H. W. *Realistic Image Synthesis Using Photon Mapping*. Natick, MA, USA: AK Peters, Ltd., 2001. isbn: 1-56881-147-0 2.

[JMM*14] Jarabo, A., Marco, J., Munoz, A., Buisan, R., Jarosz, W., and Gutierrez, D. "A framework for transient rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 33.6 (Nov. 2014). issn: 15577368. doi: 10/gfznb8 2.

[JMMG17] Jarabo, A., Masia, B., Marco, J., and Gutierrez, D. "Recent advances in transient imaging: a computer graphics and vision perspective". *Visual Informatics* 1.1 (2017). doi: 10/gfz5mh 2.

[JNSJ11] Jarosz, W., Nowrouzezahrai, D., Sadeghi, I., and Jensen, H. W. "A comprehensive theory of volumetric radiance estimation using photon points and beams". *ACM Transactions on Graphics* 30.1 (Jan. 1, 2011). issn: 0730-0301. doi: 10/fcdh2f 2.

[JZJ08] Jarosz, W., Zwicker, M., and Jensen, H. W. *The Beam Radiance Estimate for Volumetric Photon Mapping*. CS2008-0914. University of California, San Diego, 2008 2.

[Kel97] Keller, A. "Instant radiosity". *Annual Conference Series (Proceedings of SIGGRAPH)*. ACM Press, Aug. 1997. isbn: 978-0-89791-896-1. doi: 10/fqch2z 2, 3.

[KJGP23] Kim, J., Jarosz, W., Gkioulekas, I., and Pediredla, A. "Doppler time-of-flight rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 42.6 (Dec. 5, 2023). issn: 0730-0301. doi: 10/gs82tm 2.

[KK06] Kollig, T. and Keller, A. "Illumination in the presence of weak singularities". *Monte Carlo and Quasi-Monte Carlo Methods*. Ed. by Niederreiter, H. and Talay, D. Springer-Verlag, 2006. isbn: 978-3-540-25541-3. doi: 10/d42nk6 11.

[LJJ22] Liu, Y., Jiao, S., and Jarosz, W. "Temporally sliced photon primitives for time-of-flight rendering". *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 41.4 (2022). issn: 1467-8659. doi: 10/jgzq 1, 2.

[LW93] Lafortune, E. P. and Willems, Y. D. "Bi-directional path tracing". *Proceedings of the International Conference on Computational Graphics and Visualization Techniques (Compugraphics)*. Vol. 93. Alvor, Portugal, Dec. 1993 2.

[Mar13] Marco, J. "Transient Light Transport in Participating Media". PhD thesis. Universidad de Zaragoza, 2013 2.

[MGJ*19] Marco, J., Guillén, I., Jarosz, W., Gutierrez, D., and Jarabo, A. "Progressive transient photon beams". *Computer Graphics Forum* 38.1 (Mar. 2019). issn: 1467-8659. doi: 10/gfvr9w 1, 2.

[MJGJ17] Marco, J., Jarosz, W., Gutierrez, D., and Jarabo, A. "Transient photon beams". *Congreso Espanol de Informatica Grafica*. Eurographics Association, June 2017. isbn: 978-3-03868-046-8. doi: 10/gfzncc 2.

[NID20a] Nabata, K., Iwasaki, K., and Dobashi, Y. "Two-stage Resampling for Bidirectional Path Tracing with Multiple Light Sub-paths". *Computer Graphics Forum* 39.7 (2020). issn: 1467-8659. doi: 10/hbw4cd 2.

[NID20b] Nabata, K., Iwasaki, K., and Dobashi, Y. "Resampling-aware weighting functions for bidirectional path tracing using multiple light sub-paths". *ACM Transactions on Graphics* 39.2 (Mar. 2020). issn: 0730-0301. doi: 10/q9nk 2, 10.

[OLW18] O'Toole, M., Lindell, D. B., and Wetzstein, G. "Confocal non-line-of-sight imaging based on the light-cone transform". *Nature* 555.7696 (Mar. 2018). issn: 1476-4687. doi: 10/gc3gs5 2.

[PDV19] Pediredla, A., Dave, A., and Veeraraghavan, A. "SNLOS: Non-line-of-sight scanning through temporal focusing". *IEEE International Conference on Computational Photography (ICCP)*. IEEE Computer Society, May 2019. doi: 10/gspq34 1.

[PRDD15] Popov, S., Ramamoorthi, R., Durand, F., and Drettakis, G. "Probabilistic connections for bidirectional path tracing". *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 34.4 (2015). doi: 10/gfzwbh 2, 10.

[PVG19] Pediredla, A., Veeraraghavan, A., and Gkioulekas, I. "Ellipsoidal path connections for time-gated rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38.4 (July 2019). issn: 0730-0301. doi: 10/gf5jbm 1–3, 5, 8, 9.

[RGMJ22] Royo, D., García, J., Muñoz, A., and Jarabo, A. "Non-line-of-sight transient rendering". *Computers & Graphics* 107 (Oct. 1, 2022). issn: 0097-8493. doi: 10/gspq3r 2.

[SLW22] Su, F., Li, S., and Wang, G. "SPCBPT: subspace-based probabilistic connections for bidirectional path tracing". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 41.4 (July 22, 2022). issn: 0730-0301. doi: 10/gqjn7d 2, 10.

[SSD08] Smith, A., Skorupski, J., and Davis, J. *Transient Rendering*. UCSC-SOE-08-26. School of Engineering, University of California, Santa Cruz, 2008 2.

[TH19] Tokuyoshi, Y. and Harada, T. "Hierarchical Russian roulette for vertex connections". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38.4 (July 2019). issn: 0730-0301. doi: 10/gf5jbg 2.

[Vea97] Veach, E. "Robust Monte Carlo Methods for Light Transport Simulation". PhD thesis. Stanford University, Dec. 1997 2, 3, 8.

[VWG*12] Velten, A., Willwacher, T., Gupta, O., Veeraraghavan, A., Bawendi, M. G., and Raskar, R. "Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging". *Nature Communications* 3 (2012) 2.

[WFA*05] Walter, B., Fernandez, S., Arbree, A., Bala, K., Donikian, M., and Greenberg, D. P. "Lightcuts: a scalable approach to illumination". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 24.3 (Aug. 2005). issn: 0730-0301. doi: 10/dhp5d3 2, 3.

[YKC*21] Yi, S., Kim, D., Choi, K., Jarabo, A., Gutierrez, D., and Kim, M. H. "Differentiable transient rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 40.6 (Dec. 10, 2021). doi: 10/h2db 2.

[Yuk19] Yuksel, C. "Stochastic lightcuts". *Proceedings of High Performance Graphics – Short Papers*. Ed. by Steinberger, M. and Foley, T. Eurographics Association, 2019. isbn: 978-3-03868-092-5. doi: 10/ggjcgw 2, 3, 6, 7.