Ellipsoidal Path Connections for Time-Gated Rendering

ADITHYA PEDIREDLA, Rice University, USA and Carnegie Mellon University, USA
ASHOK VEERARAGHAVAN, Rice University, USA
IOANNIS GKIOULEKAS, Carnegie Mellon University, USA

During the last decade, we have been witnessing the continued development of new time-of-flight imaging devices, and their increased use in numerous and varied applications. However, physics-based rendering techniques that can accurately simulate these devices are still lacking: while existing algorithms are adequate for certain tasks, such as simulating transient cameras, they are very inefficient for simulating time-gated cameras because of the large number of wasted path samples. We take steps towards addressing these deficiencies, by introducing a procedure for efficiently sampling paths with a predetermined length, and incorporating it within rendering frameworks tailored towards simulating time-gated imaging. We use our open-source implementation of the above to empirically demonstrate improved rendering performance in a variety of applications, including simulating proximity sensors, imaging through occlusions, depth-selective cameras, transient imaging in dynamic scenes, and non-line-of-sight imaging.

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

Additional Key Words and Phrases: time-gated cameras, transient imaging.

1 INTRODUCTION

Time-of-flight (ToF) imaging is revolutionizing a large array of applications including robotics, autonomous navigation, atmospheric sciences, medicine, and even entertainment and human-computer interaction. Unlike conventional intensity sensors, ToF sensors exploit the fact that the speed of light is finite, and record information about the time it takes for photons to travel from a light source to the sensor. This information can then be used to recover geometric and material properties of the scene the photon traveled through, including even parts of the scene that are not directly visible to the sensor because of opaque or translucent occluders.

Within computer graphics, imaging and rendering have consistently advanced in lockstep: As new imaging sensors are being developed, rendering algorithms that can accurately and efficiently simulate measurements of such sensors quickly follow. In turn, the availability of rendering algorithms greatly enhances the ability of the sensor designers to optimize their designs, resulting in novel

Authors' addresses: Adithya Pediredla, Rice University, USA, adithya.k.pediredla@rice.edu, Carnegie Mellon University, USA, apedired@andrew.cmu.edu; Ashok Veeraraghavan, Rice University, USA, vashok@rice.edu; Ioannis Gkioulekas, Carnegie Mellon University, USA, igkioule@andrew.cmu.edu.

sensors of superior performance. In particular, as state-of-the-art machine learning tools are increasingly used for both designing the sensors [Marco et al. 2017a] and processing their outputs [Gruber et al. 2019]. Efficient rendering tools are necessary to generate the large, diverse, and realistic datasets needed for training.

Existing physically-accurate rendering algorithms are primarily tailored to conventional, steady-state rendering. As a consequence, they can be suboptimal for simulating ToF sensors, with their performance varying considerably between different ToF rendering tasks. To characterize the performance characteristics of existing algorithms, we will broadly classify ToF rendering tasks into two categories. The first category includes tasks such as simulating *continuous-wave time-of-flight cameras*, which accumulate all photons with a weight that depends on their time of travel; as well as *transient cameras*, which aggregate all photons but separate them into a sequence of images, each recording contributions only from photons with a specific time of travel. Existing steady-state rendering algorithms remain efficient for simulation tasks in this category, as the majority of paths they generate will have a non-zero contribution, regardless of their pathlength. Tasks in this category have generally been the main focus of previous research on ToF rendering; for instance, Jarabo et al. [Jarabo et al. 2014] improve rendering performance by introducing path-sampling schemes and reconstruction techniques tailored to the transient imaging setting.

The second type of ToF rendering tasks involves simulating images that accumulate contributions only from a small subset of photons, whose time of travel is within some narrow interval. These *time-gated* rendering tasks arise in a large number of practical situations; examples include time-gated sensors used as proximity detectors, gated laser ranging cameras, as well as situations where transient imaging is performed in dynamic scenes such as outdoors environments or tissue with blood flow. Unfortunately, existing rendering algorithms cannot be used for efficient time-gated rendering: the vast majority of the paths generated by these algorithms end up being rejected, for having length outside the narrow range accumulated by the simulated sensor.

Given this challenge and the importance of the problem, we focus on developing efficient Monte Carlo algorithms for time-gated rendering tasks. At the core of the challenge is the fact that current physics-based rendering algorithms cannot sample paths that satisfy specific constraints on their length. We take first steps towards addressing this fundamental limitation, by making the following technical contributions: We develop a path-sampling technique that generates paths with a predetermined target length. This technique is a variant of standard bidirectional path tracing [Lafortune and Willems 1996; Veach and Guibas 1995a], which we term *bidirectional path tracing with ellipsoidal connections*. This allows us to simulate time-gated sensors without having to reject an excessive number of sample paths. We develop the mathematical machinery

required to ensure that this new path-sampling technique remains consistent and unbiased. Additionally, we propose algorithms for accelerating path sampling using our proposed technique, including efficient geometric query procedures analogous to those used for accelerating ray-mesh intersections. Finally, we implement all of our contributions within an open-source renderer [Jakob 2010], and use this renderer to demonstrate improved rendering performance in a variety of time-gated rendering tasks. We hope that our publicly-available implementation [Pediredla et al. 2019b] will facilitate the design of new ToF cameras and related inverse algorithms.

2 RELATED WORK

Time-of-flight imaging. We can broadly distinguish between two types of time-of-flight imaging sensors. First, transient imaging sensors record a sequence of images, with each image comprising contributions from photons that have traveled for a specific time between emission and measurement [Faccio and Velten 2018; Jarabo et al. 2017]. Transient cameras have been implemented using a number of different technologies, with temporal resolutions ranging from a few nanoseconds to a few femtoseconds. These include streak cameras [Velten et al. 2013], intensified CCD (ICCD) sensors [Cester et al. 2019; Pediredla et al. 2017b], single-photon avalanche diodes (SPADs) [Gariepy et al. 2015], continuous-wave time-of-flight sensors [Heide et al. 2013; Peters et al. 2015], Kerr gates [Ham 2019; Schmidt et al. 2003; Takahashi et al. 1994; Zhan et al. 2016], and interferometry [Gkioulekas et al. 2015]. Transient imaging has recently attracted increased interest within computer graphics, for applications such as non-line-of-sight imaging [Buttafava et al. 2015; Chan et al. 2017; Heide et al. 2019, 2014; Kirmani et al. 2009; O'Toole et al. 2018; Pediredla et al. 2017a, 2019a; Tsai et al. 2017; Velten et al. 2012; Xin et al. 2019], separating light transport components [O'Toole et al. 2014; Wu et al. 2014a,b], and inverting and seeing through scattering [Gkioulekas et al. 2016; Satat et al. 2016].

Second, there are time-of-flight sensors that produce single images through the (weighted) integration of contributions from photons with travel time within the same range. One example is time-gated sensors, which couple ultrafast pulsed lasers with ultrafast shutters, activated only for a specific time interval after each laser pulse is emitted. We discuss this case below. Another example is sensors that record only the time of arrival of the first-arriving photons, such as lidar and laser scanning systems. Continuous-wave time-of-flight (CW-ToF) sensors combine an amplitude-modulated light source with an exposure-modulated sensor, to measure a weighted integral of a transient sequence [Gupta et al. 2015; Lin et al. 2014]. Both lidar and CW-ToF systems are primarily used for 3D sensing and velocity sensing [Shrestha et al. 2016] in a wide range of applications, including robotics and autonomous vehicles.

Time-gated imaging. Our focus is on time-of-flight imaging tasks which require capturing individual images that receive non-zero contributions from only photons with the time-of-travel within a very narrow range. The time-gated cameras mentioned above naturally fall under this category. Such time-gated sensors find applications in a variety of areas. For example, SPAD sensors time-gated at a few tens of picoseconds are used in biology and medical settings

for fluorescence lifetime imaging [Lakowicz et al. 1992], and optical sectioning of layers within scattering media [Das et al. 1993]. Time-gated sensors are also used to enhance visibility, for instance by allowing imaging through fog and smoke [Satat et al. 2018]. The recent decade saw a proliferation of advanced driver assistance systems (ADAS) to increase automobile safety, with fixed time-gated cameras of gate widths as narrow as 200 ps used to avoid collision and improve night-time navigation [David et al. 2006; Grauer and Sonn 2015]). Coupled with machine learning algorithms, time-gated sensors are also used as alternatives to lidar for dense depth sensing [Gruber et al. 2019]. Gated laser ranging and short-wave-infrared sensors are used in military applications, to survey objects at specific distances [las 2019; Baker et al. 2004]. Finally, even though most non-line-of-sight imaging algorithms require entire transients, some techniques use time-gated measurements [Laurenzis and Velten 2014; Pediredla et al. 2019a; Thrampoulidis et al. 2018]

Time-gated imaging scenarios can arise even when using ToF technologies other than time-gated sensors. For example, even though typical CW-ToF sensors accumulate non-zero contributions from all photons regardless of their time-of-travel, it is possible to use special modulation patterns to create a camera that almost exclusively measures photons within a specific time-of-travel range [Tadano et al. 2015]. Such cameras have been used for applications such as photography through partial occlusions, and optical Z-keying. Another example is when using transient cameras such as ICCDs or Kerr gates to image dynamic scenes. These cameras capture entire transients by progressively shifting a narrow time gate. In dynamic scenes, such cameras effectively capture a single time-gated slice within the time duration in which the scene can be assumed static. During the time it takes to change the gate (a few milliseconds), the scene motion effectively means that the next frame in the sequence will be a time-gated slice of a (slightly) different scene.

Time-of-flight rendering. Analogously to the imaging case, we use the term *time-of-flight (ToF) rendering* to refer to the simulation of both transient image sequences, and time-gated images. The theoretical framework for ToF rendering is provided by the time-resolved versions of the radiative transfer equation [Chandrasekhar 1960], rendering equation [Smith et al. 2008], and path-integral formulation of light transport [Jarabo et al. 2014].

Most prior research on ToF rendering focuses on simulating entire transients. Several works have adapted steady-state Monte Carlo rendering algorithms to the transient case, both for forward simulations [Jarabo 2012; Jarabo et al. 2014] and for inverse rendering problems [Gkioulekas et al. 2016; Naik et al. 2011; Tsai et al. 2019]. Most of these works directly repurpose algorithms such as bidirectional path tracing (BDPT) to generate samples. Notably, Jarabo et al. [2014] additionally introduced a modified BDPT procedure for generating paths in scattering media with a more uniform path-length distribution. An alternative approach to transient rendering, which focuses on increased efficiency at the cost of introducing bias, has been to extend to the transient setting reconstruction algorithms based on density estimation, such as photon mapping [Ament et al. 2014; Jarabo et al. 2014; Marco et al. 2018, 2017b].

Unlike these previous works, our focus is on developing rendering algorithms for time-gated imaging scenarios. In this case, directly

repurposing steady-state algorithms such as BDPT is very inefficient, because most sampled paths end up being rejected. For biased time-gated rendering, the transient photon mapping techniques mentioned previously can be extended straightforwardly to this case, by using kernels that weigh photons based on how much their time-of-flight deviates from the target time gate. In the context of unbiased time-gated rendering, some of the path sampling techniques introduced by Jarabo et al. [2014] can help improve performance when rendering volumetric media, by allowing to specify lower and upper bounds on the length of sampled paths. Inspired from this work, we aim to develop new path sampling techniques that operate on surfaces, rather than volumetric media, and that provide a more expansive range of options for controlling pathlength—for example, allowing for specifying exact pathlength values. This flexibility can help expand the range of applications for which unbiased, physically-accurate time-gated rendering is practical.

3 BACKGROUND ON TIME-OF-FLIGHT RENDERING

The foundation of modern Monte Carlo rendering algorithms is the *path integral formulation* of light transport [Veach 1997]. According to this formulation, the image I recorded by any *steady-state* intensity sensor can be written as a path integral of the form,

$$I = \int_{\mathcal{P}} f(\bar{x}) d\mu(\bar{x}). \tag{1}$$

In this integral, $\bar{x} = x_0 \rightarrow \ldots \rightarrow x_{B+1}$ is an ordered sequence of $B + 2$ three-dimensional points, for any $B \geq 0$, and represents a path photons can follow from a light source (path vertex x_0) to a sensor (path vertex x_{B+1}). The function $f(\bar{x})$ is the *radiance throughput* of the path, which depends on the geometry and material of the scene, the spatio-angular emission of light sources, and the spatio-angular sensitivity (or importance) of the sensor. Finally, $\mathcal{P}$ is the space of all paths $\bar{x}$, and μ is its Lebesgue measure.

Following Jarabo et al. [2014], the path integral formulation can be extended to time-of-flight sensors, by incorporating a *pathlength importance*[1] term W_τ that depends on the length $\|\bar{x}\|$ of a path $\bar{x}$,

$$I = \int_{\mathcal{P}} W_\tau(\|\bar{x}\|) f(\bar{x}) d\mu(\bar{x}). \tag{2}$$

The pathlength importance function W_τ is independent of the scene; instead, it is uniquely determined by the technology, hardware design, and parameters of the time-of-flight sensor. We will broadly distinguish between three time-of-flight imaging modalities, which can be modeled using Equation (2) with different functions W_τ.

Time-gated cameras. A time-gated camera captures only photons with time-of-flight within some narrow range, and rejects all other photons. If the time range is $[\tau - \frac{\Delta\tau}{2}, \tau + \frac{\Delta\tau}{2}]$, then such a camera can be modeled using a pathlength importance function,

$$W_\tau(\|\bar{x}\|) = \text{rect}\left(\frac{\|\bar{x}\| - \tau}{\Delta\tau}\right). \tag{3}$$

In this context, the pathlength importance function is also known as the *gate* of the camera, a term that we will also use throughout the paper. In practice, such cameras have imperfect gating mechanisms,

[1]We will be focusing on scenes that do not include refractive media. Therefore, we can use the time-of-flight t of a photon and the length τ of its path interchangeably. The two are related as $\tau = ct$, where $c = 1$ is the speed of light in vacuum.

and therefore may be better modeled with lower-frequency but narrow pathlength importance functions; for example, a Gaussian of mean τ and standard deviation $\Delta\tau$,

$$W_\tau(\|\bar{x}\|) = \text{Gaussian}(\|\bar{x}\|; \tau, \Delta\tau). \tag{4}$$

Transient cameras. A transient camera records a *sequence* of images $I(\tau)$, $\tau \in \{\tau_{\min}, \ldots, \tau_{\max}\}$, with the entire sequence often termed a *transient image*. Each frame $I(\tau)$ in the sequence records contributions only from photons accepted by a narrow time gate centered around τ. For an *ideal* transient camera, this gate only permits photons with time-of-flight exactly τ, corresponding to a *per-frame* pathlength importance function that is a Diract delta. In practice, the per-frame time gates are imperfect, and are better modeled as in Equations (3) and (4). We note that a transient camera captures measurements equivalent to those captured by multiple time-gamed cameras, each with a gate centered at a different pathlength τ.

Continuous-wave time-of-flight (CW-ToF) cameras. A CW-ToF camera uses illumination with temporally-modulated amplitude, coupled with a sensor with temporally-modulated sensitivity. This corresponds to a pathlength importance function [Lin et al. 2014],

$$W_\tau(\|\bar{x}\|) = C(\|\bar{x}\|), \tag{5}$$

where C is the cross-correlation between the illumination and sensor modulation functions. Typically, C is a sinusoid, with frequency in the range of tens to hundreds of MHz. However, cross-correlations of trapezoidal [Gupta et al. 2018] and narrow rectangular [Tadano et al. 2015] shapes have also been demonstrated. In the latter case, the camera effectively reduces to a time-gated camera for some narrow pathlength importance function W_τ.

3.1 Rendering time-of-flight cameras

Simulating measurements of a time-of-flight sensor requires evaluating the integral of Equation (2). As in the steady-state case ($W_\tau(\|\bar{x}\|) = 1$), analytic integration is impossible except for trivial scenes. Instead, Monte Carlo rendering algorithms [Dutre et al. 2006; Veach 1997] attempt to approximate the integral using Monte Carlo integration, forming an estimate of the form,

$$\tilde{I} = \sum_{n=1}^{N} \frac{W_\tau(\|\bar{x}_n\|) f(\bar{x}_n)}{p(\bar{x}_n)}, \tag{6}$$

where each $\bar{x}_n$ is a randomly sampled path, and $p(\bar{x}_n)$ is the probability of sampling this path. When the path-sampling distribution $p(\bar{x})$ has non-zero measure for all paths $\bar{x}$, estimators of this form are both consistent ($\lim_{N \rightarrow \infty} \tilde{I} = I$) and unbiased ($\mathbb{E}_{\{\bar{x}_n\}}[\tilde{I}] = I$).

The efficiency of these Monte Carlo estimators critically depends on the availability of path sampling algorithms with probability distributions $p(\bar{x})$ that approximate the target integrand. For rendering in the steady-state case, there exist powerful algorithms that approximate the radiance throughput term $f(\bar{x})$, including path tracing, bidirectional path tracing (BDPT), Metropolis light transport, and numerous variants; see Pharr et al. [2016] and Dutre et al. [2006] for a review. The same algorithms can also be used to generate paths for time-of-flight rendering. We will be focusing on using BDPT in this way, an approach we term baseline BDPT.

Baseline BDPT, however, is suboptimal for time-of-flight rendering, as it does not approximate the correct integrand: Instead of sampling paths with a probability proportional to the product $W_\tau(\|\bar{x}\|)f(\bar{x})$ of path integral (2), it samples proportionally to the radiance throughput term $f(\bar{x})$ of path integral (1). The effect of this mismatch on performance is critically dependent on the pathlength importance function W_τ. For example, regular CW-ToF cameras with sinusoidal pathlength importance functions will still accept most sampled paths with high importance; therefore baseline BDPT remains a relatively efficient rendering algorithm for these cameras.

Baseline BDPT can also be used to efficiently render transient sequences $I(\tau)$: In this case, even though each individual image in the transient sequence corresponds to a very narrow pathlength importance function, collectively the entire sequence has an effective pathlength importance function that covers all possible pathlengths. Therefore, baseline BDPT can operate as follows: for each sampled path $\bar{x}$, assign it to all images in the transient sequence with a gate that is non-zero at $\|\bar{x}\|$. This procedure, termed *temporal path reuse* in Jarabo et al. [2014], guarantees that all sampled paths contribute to one image in the transient sequence $I(\tau)$.

3.2 Our focus: time-gated rendering

Unfortunately, baseline BDPT becomes very inefficient for rendering tasks requiring the simulation of isolated images (rather than sequences) corresponding to pathlength importance functions W_τ with narrow support. As baseline BDPT does not provide control over the length of sampled paths $\bar{x}$, its use in such cases results in most sampled paths having zero net contribution, making baseline BDPT a very wasteful path sampling procedure. Tasks like this arise in a number of different time-of-flight imaging contexts, and we discuss three of them as examples. An obvious first case is when simulating measurements from time-gated sensors, as described above. A second case is simulating transient cameras operating in dynamic environments, where the measured scene changes as the camera captures a transient sequence. Then, each frame of the transient sequence is a time-gated measurement of an effectively different scene, and therefore temporal path reuse cannot be used. Finally, a third case is when rendering CW-ToF cameras where the cross-correlation of illumination and sensor modulation has very narrow support, as in the depth-selective camera of Tadano et al. [2015]. We use the term *time-gated rendering* to describe these and all other rendering tasks dealing with very narrow pathlength importance functions; these tasks will be the focus of the rest of the paper.

BDPT with ellipsoidal connections. We introduce a path sampling algorithm that helps ameliorate the inefficiency of baseline BDPT for time-gated rendering tasks. Rather than directly approximate the product $W_\tau(\|\bar{x}\|)f(\bar{x})$ of the path integral of Equation (2), our algorithm operates in three steps:

(1) A target pathlength τ is selected, for instance through importance sampling or stratified sampling of the narrow pathlength importance function $W_\tau(\|\bar{x}\|)$.
(2) Standard BDPT is used to trace source and camera subpaths, up until a path length that does not exceed τ.
(3) Complete paths are formed by connecting every pair of vertices in the source and camera subpaths through an additional
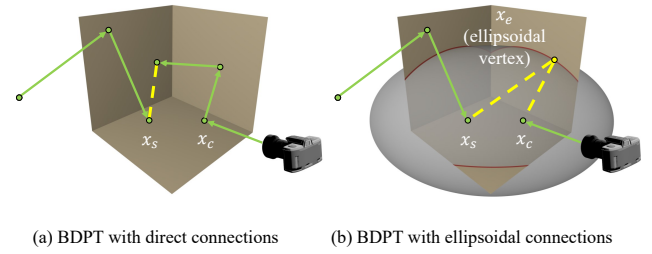
(a) BDPT with direct connections   (b) BDPT with ellipsoidal connections

Fig. 1. **BDPT with direct and ellipsoidal connections.** (a) Given sampled source and camera subpaths, standard BDPT forms complete paths by directly connecting every vertex in one subpath to every vertex in the other. (b) By contrast, BDPT with ellipsoidal connections forms complete paths by connecting each such pair of vertices through an additional vertex, which lies on the intersection of an ellipsoid with the scene (shown in red).

vertex, rather than directly as in standard BDPT, with the new vertex selected so that the total pathlength equals τ.

We call the new type of connections in the third step of our algorithm *ellipsoidal connections*. We describe them in detail in the following section, where we also explain the reasoning for this name.

By producing paths of a length τ sampled from the pathlength importance function W_τ, BDPT with ellipsoidal connections guarantees that all sampled paths will have non-zero contribution to the integral of Equation (2). This guarantee, however, comes at a price, as each of these paths is considerably more expensive to generate than with baseline BDPT, especially for scenes with complicated geometry. This trade-off between high pathlength importance and high sampling efficiency implies that there is a scene-dependent upper-bound on the width of the gate W_τ, beyond which it is better to use baseline BDPT than BDPT with ellipsoidal connections. In Section 5, we describe techniques for improving the efficiency of BDPT with ellipsoidal connections, and in Section 6, we qualify the resulting trade-off through extensive experiments.

We note that we are not the first to propose using a new vertex as a means to control the pathlength of the final, complete path. Jarabo et al. [2014] proposed a path sampling procedure that, likewise, uses a new vertex when connecting BDPT subpaths, in order to improve performance for time-of-flight rendering *in volumetric media*. Our path sampling procedure is different in two ways: First, it targets rendering of *surfaces*, rather than volumetric media. Second, it allows for *exact* control of the length of sampled paths, instead of ensuring uniform distribution within some pathlength range. Conceptually, our proposed ellipsoidal connections can be extended for sampling in volumetric media, in which case it would be interesting to compare them to the corresponding path sampling technique of Jarabo et al. [2014]. We leave this for future work.

4 COMPUTING ELLIPSOIDAL CONNECTIONS

Throughout this section, we assume that we are given source and camera subpaths, $\bar{x}_S$ and $\bar{x}_C$ respectively, created using bidirectional path tracing. Additionally, we are given a target pathlength τ satisfying $\|\bar{x}_S\| + \|\bar{x}_C\| < \tau$. Our objective is to form complete paths of length τ, by using intermediate vertices x_e to connect each vertex $x_s \in \bar{x}_S$, $s \in \{0, \ldots, S\}$ in the source subpath, to every vertex
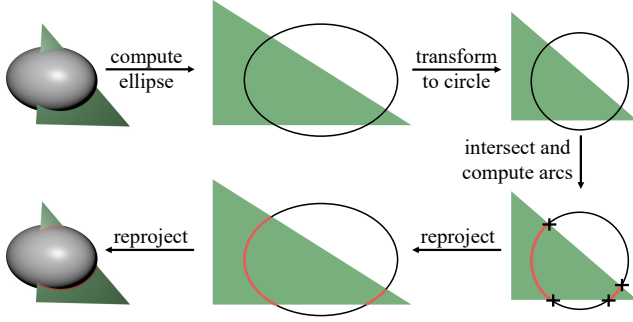
Fig. 2. **Ellipsoid-polygon intersection.** We first compute the ellipse formed by intersecting the ellipsoid and the plane containing the polygon. We then transform both shapes so that the ellipse becomes a unit circle. We identify the circular arcs contained inside the polygon. Finally, we reproject these arcs to the original three-dimensional coordinate system.

$x_c \in \bar{x}_C$, $c \in \{0, \ldots, C\}$ in the camera subpath, where $S + 1$ and $C + 1$ are the number of vertices in the two subpaths and $S, C \geq 0$.

We denote by $\bar{x}_{0:s}$ the part of the source subpath from its start until the vertex x_s, and likewise for $\bar{x}_{0:c}$ on the camera subpath. Then, the intermediate vertex x_e must be selected so that,

$$\|\bar{x}_{0:s}\| + \|x_e - x_s\| + \|x_e - x_c\| + \|\bar{x}_{0:c}\| = \tau \qquad (7)$$

$$\implies \|x_e - x_s\| + \|x_e - x_c\| = \underbrace{\tau - \|\bar{x}_{0:s}\| - \|\bar{x}_{0:c}\|}_{\tau_e}. \qquad (8)$$

Equation (8) defines a three-dimensional prolate ellipsoid $\mathcal{E}(x_s, x_c, \tau_e)$ with foci x_s and x_c, and pathlength τ_e. This is our key observation: For the complete path to have pathlength τ, we need to connect x_s and x_c through an intermediate vertex x_e that satisfies the ellipsoid constraint (8) (see also Figure 1). We refer to vertex x_e as an *ellipsoidal vertex*, and to path connections using such vertices as *ellipsoidal connections*. This ellipsoidal constraint is well-known in NLOS imaging applications, which reconstruct occluded geometry from pathlength measurements of partially observed paths; for example, it underlies the ellipsoidal backprojection [Arellano et al. 2017] and space carving [Tsai et al. 2017] algorithms. To the best of our knowledge, we are the first to use this constraint for rendering.

Ellipsoid-scene intersections. In addition to satisfying Equation (8), the vertex x_e must lie on the union of scene surfaces $\mathcal{M}$. Therefore, to create the complete path, we first need to compute the intersection $\mathcal{E}(x_s, x_c, \tau_e) \cap \mathcal{M}$ between the ellipsoid $\mathcal{E}(x_s, x_c, \tau_e)$ and $\mathcal{M}$. Here we address how to compute this intersection for the case when $\mathcal{M}$ is the union of oriented polygonal primitives. This subsumes the case of a scene comprising polygonal meshes, arguably the most common surface representation in computer graphics applications.

In particular, we develop an algorithm for *analytically* intersecting the ellipsoid $\mathcal{E}(x_s, x_c, \tau_e)$ with an arbitrary polygon $\mathcal{T}$. We can then use this intersection operator to determine $\mathcal{E}(x_s, x_c, \tau_e) \cap \mathcal{M}$, by applying it iteratively to all polygons in $\mathcal{M}$ (though see Section 5 for strategies to accelerate this procedure). We outline here the steps involved in this operation, and derive it in detail in the supplement.

We start by computing the two-dimensional ellipse arising as the intersection of the ellipsoid $\mathcal{E}(x_s, x_c, \tau_e)$ with the plane containing the polygon $\mathcal{T}$. Then, we use the procedure shown in Figure 2 to determine the portions of this ellipse that are inside $\mathcal{T}$: First, we compute the transformation that converts the two-dimensional ellipse to an axis-aligned unit circle, and apply this transformation to $\mathcal{T}$. Second, we compute the intersections of the unit circle with the edges of the transformed oriented polygon, and determine whether each intersection is moving towards the interior or exterior of the circle. Third, we identify all circular arcs lying between an outgoing and an ingoing intersection. Finally, we convert these circular arcs back to elliptic curves in the original coordinate system by applying the inverse of the transformation we used on $\mathcal{T}$.

Ellipsoidal vertex sampling. After computing $\mathcal{E}(x_s, x_c, \tau_e) \cap \mathcal{M}$, we must sample from it a vertex x_e to form a complete path, compute the sampling probability of x_e, and use these quantities to update the Monte Carlo estimate of Equation (6). Effectively, by sampling x_e, we form a Monte Carlo approximation of the integral,

$$\sum_{i=1}^{T} \int_{\mathcal{E}(x_s, x_c, \tau_e) \cap \mathcal{T}_i} f(\bar{x}_{0:s} \to x_e(\tau, \theta) \to \bar{x}_{c:0}) J(\mu; \tau, \theta) \, d\theta \, d\tau, \quad (9)$$

where $\{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$ are the polygons intersected by the ellipsoid $\mathcal{E}(x_s, x_c, \tau_e)$, and $x_e(\tau, \theta)$ is a parameterization of the corresponding elliptic arc on each polygon. As we reparameterize the polygon surface in terms of τ and θ, in addition to the throughput term f for area sampling [Veach and Guibas 1995b], we need to compute the Jacobian $J(\mu; \tau, \theta)$ corresponding to this reparameterization.

In our implementation, we sample a point x_e by first uniformly selecting one of the polygons $\mathcal{T}_i$, then parameterizing its elliptic arc in terms of center angle θ, and finally sampling a point on it uniformly with respect to θ. These sampling and parameterization choices are motivated by the fact that they facilitate the computation of the path sampling probability and the Jacobian $J(\mu; \tau, \theta)$. In the supplement, we derive *analytic* and easy to compute expressions for these quantities. We have verified these expressions numerically through comparisons with automatic differentiation [Anderson et al. 2017], to ensure unbiasedness of our path sampling algorithm.

Spherical connections for direct illumination. In order for our path sampling technique to be consistent, it must sample from the same space of paths as BDPT with direct connections. Unfortunately, this is not possible using only ellipsoidal connections: The addition of the extra vertex x_e means that the minimum number of vertices of a resulting full path is 3 ($B = 1$), with the other two vertices being on the source and camera. This excludes two-bounce paths ($B = 0$), corresponding to source-camera connections for direct illumination.

To handle this type of path, we extend our path sampling technique by also performing *spherical connections* when creating paths from the first points of the source and camera subpaths: Let x_0 be the first point of the camera subpath $\bar{x}_C$, located on the sensor. We intersect the source with a sphere $\mathcal{E}(x_0, \tau/2)$, centered at x_0 and of radius $\tau/2$ equal to half the target pathlength. Then, we form a full two-bounce path by connecting x_0 with the resulting intersection point. And likewise for the first point of the source subpath. Most aspects of ellipsoidal connections, such as Jacobian computation

(a) the bounding box is in the negative half-space of a focus point   (b) the bounding box is contained inside the ellipsoid   (c) the bounding box does not intersect the ellipsoid's AABB

(d) the primitive is in the negative half-space of a focus point   (e) a focus point is in the negative half-space of the primitive   (f) the primitive is contained inside the ellipsoid
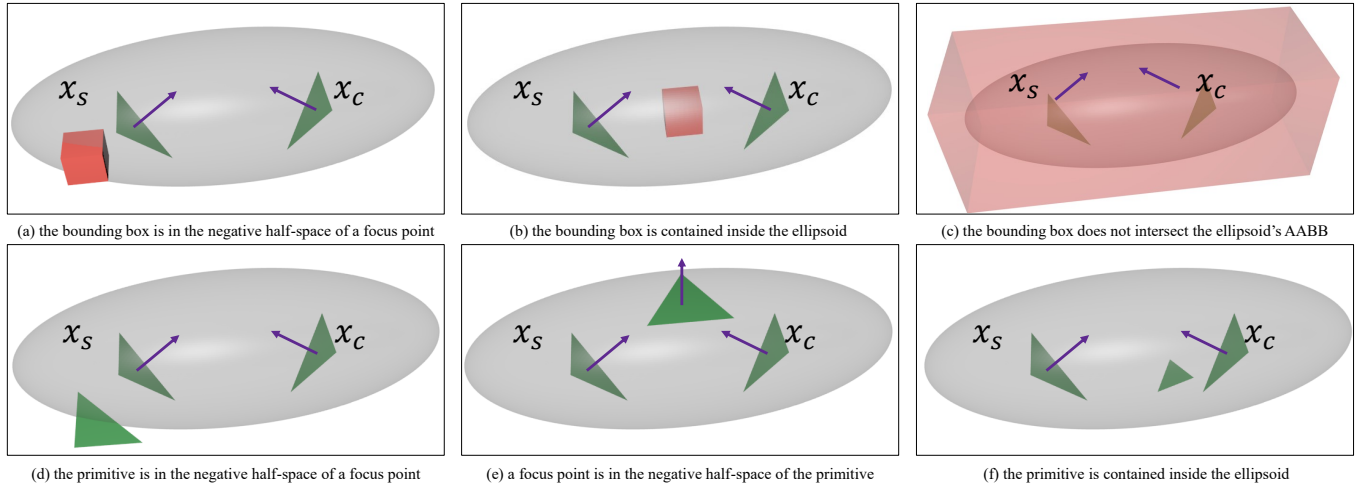
Fig. 3. **Accelerating ellipsoidal intersection queries.** We use a bounding-volume-hierarchy (BVH) structure to store the scene triangles. We parse the BVH tree in a depth-first manner to identify all the triangles intersecting the ellipsoid. At every level of the tree, we use pruning techniques to either reject an entire subtree (top row) or individual triangles (bottom row).

and the acceleration techniques of the next section, can be extended straightforwardly to spherical connections. By combining ellipsoidal and spherical connections, our path sampling technique samples from the same path space as BDPT, and therefore is consistent.

Multiple importance sampling. A critical component for the performance of standard BDPT algorithms is the use of multiple important sampling (MIS) [Veach and Guibas 1995b]: The contribution of each complete path is weighed based on the probability of sampling this path using different combinations of source and camera subpath vertices. Even though the exact MIS formulation from standard BDPT can be adapted for the case ellipsoidal connections, doing so would require performing multiple additional ellipsoid-scene intersections per path, which is computationally expensive. Instead, we use the following approximation: For each complete path, we form an MIS weight by considering the probabilities of sampling the path *with direct connections*, for every possible pair of adjacent edges in the path (as ellipsoidal connections add two edges). We have verified numerically that the resulting estimate remains unbiased, and shows a strong reduction in variance (e.g., by suppressing "fireflies").

5 ACCELERATING ELLIPSOIDAL CONNECTIONS

We now discuss techniques for accelerating BDPT rendering with ellipsoidal connections.

Space partitioning. Space-partitioning data structures such as k-d trees, octrees, and bounding volume hierarchies (BVH) have been central for accelerating the ray-mesh intersection operations required by all path tracing algorithms. We use these data structures to also accelerate the computation of the ellipsoid-scene intersection $\mathcal{E}(x_s, x_c, \tau_e) \cap \mathcal{M}$ described in Section 4. We describe our use of BVH, though our arguments extend to other similar data structures.

We first form an axis-aligned bounding box (AABB) for the ellipsoid $\mathcal{E}(x_s, x_c, \tau_e)$, and a BVH for the scene polygon soup $\mathcal{M}$ (usually one already exists for BDPT). Then, we traverse the BVH tree in a depth-first manner, to form a list of all primitives that the ellipsoid

may intersect. At every tree node, we can prune an entire child sub-tree if its AABB satisfies any of the following conditions:

- It is contained in the negative half-space of a primitive containing at least one of the foci of $\mathcal{E}(x_s, x_c, \tau_e)$ (Figure 3(a)).
- It is contained within the $\mathcal{E}(x_s, x_c, \tau_e)$ (Figure 3(b)).
- It does not intersect the ellipsoid's AABB (Figure 3(c)).

Early rejection of primitives. The BVH traversal produces a list of polygon primitives that we need to examine for intersections with the ellipsoid $\mathcal{E}(x_s, x_c, \tau_e)$. We devise a set of criteria for rejecting some of these primitives without having to perform the ellipsoid-polygon intersection operation of Section 4. Namely, we reject a candidate primitive if it satisfies any of the following conditions:

- Its negative half-space includes at least one of the foci of $\mathcal{E}(x_s, x_c, \tau_e)$ (Figure 3(d)).
- It is contained in the negative half-space of a primitive containing at least one of the foci of $\mathcal{E}(x_s, x_c, \tau_e)$ (Figure 3(e)).
- It is contained within $\mathcal{E}(x_s, x_c, \tau_e)$ (Figure 3(f)).

Multiple ellipsoidal connections. In standard BDPT, there is only one possible direct connection between two vertices x_s and x_c of the source and camera subpaths. By contrast, in our setting, many possible ellipsoidal connections can be used to form complete paths. In particular, after computing the ellipsoid-scene intersection $\mathcal{E}(x_s, x_c, \tau_e) \cap \mathcal{M}$, we can amortize the cost of path generation by sampling from it multiple ellipsoidal vertices x_e and using them to form multiple complete paths. These paths are correlated, as they have the same source and camera sub-paths, and therefore do not reduce the variance of the image estimate $\tilde{I}$ by the same amount as paths that are sampled completely independently. However, this is outweighed by the fact that these paths are a lot cheaper to generate than sampling a new path, since they reuse the already computed $\mathcal{E}(x_s, x_c, \tau_e) \cap \mathcal{M}$. These correlated paths can critically improve performance when rendering scenes that include meshes with very large numbers of polygons. We note that, despite being correlated,
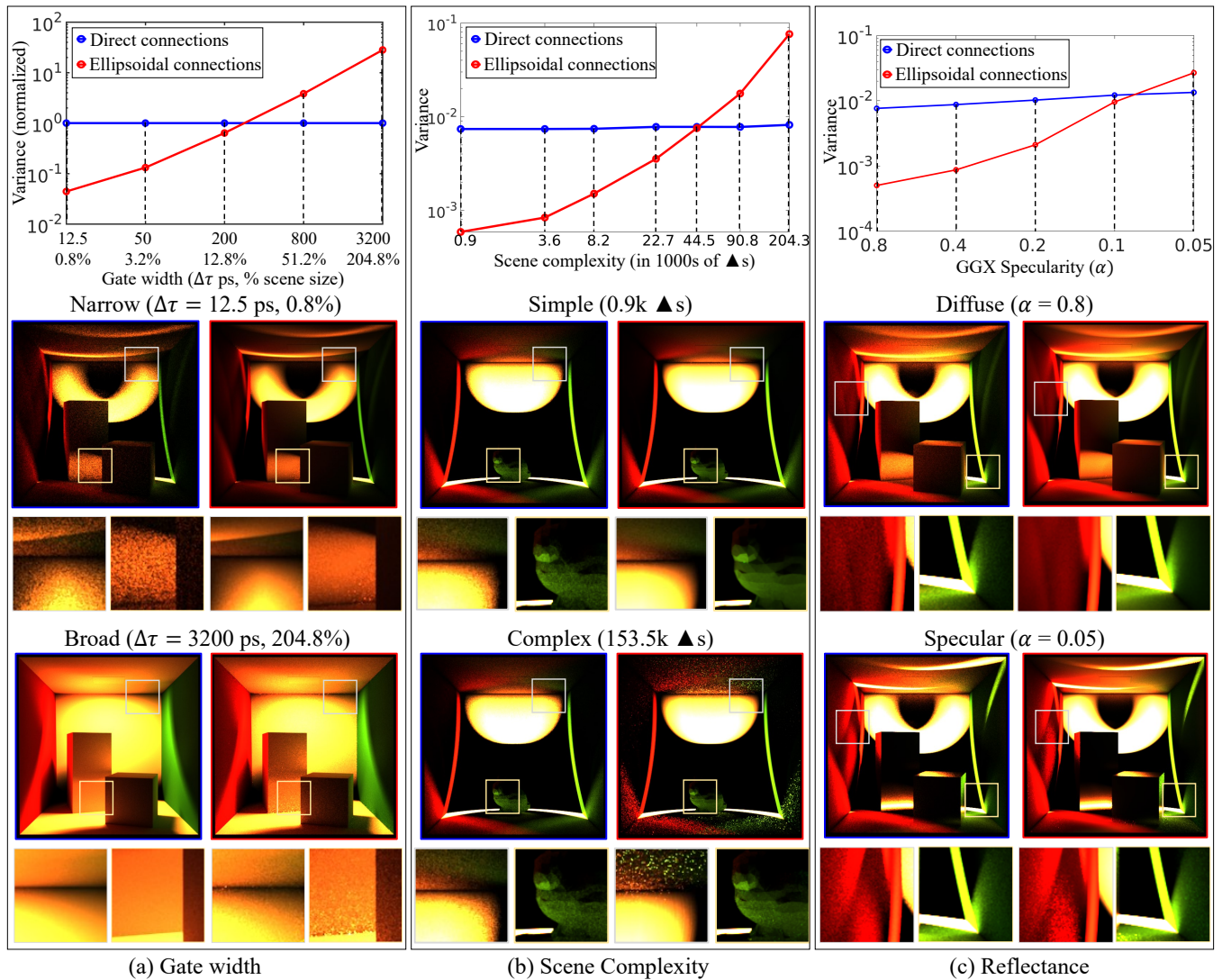
Fig. 4. **Effect of gate width, scene complexity, and reflectance.** (a) We compare the variance (computed over 50 renderings) of BDPT with direct and ellipsoidal connections, as a function of: (a) gate width $\Delta\tau$; (b) number of triangles in the scene (increased by tessellating the Stanford bunny); and (c) specularity (increased by changing the roughness parameter of GGX BRDF). In all cases, we additionally visualize renderings produced by the two algorithms for the corresponding most extreme settings. All images are rendered for 60 s. In (b) and (c), a gate width of $\Delta\tau = 12.5$ ps (0.8% of scene size) is used.

these additional path samples do not introduce bias: They effectively correspond to using multiple samples x_e for the Monte Carlo approximation of the integral of Equation (9).

Computational complexity. Using the above acceleration techniques, the average complexity of ellipsoidal connections is logarithmic in the number of scene triangles, but with a worse constant than for ray intersections due to the need for multiple BVH traversals. We note, though, that after the first ellipsoidal connection, subsequent amortized connections have constant complexity.

6 EXPERIMENTS

We evaluate the performance of BPDT with ellipsoidal connections, both quantitatively and qualitatively, for various time-gated rendering tasks. In all experiments, we additionally ran both standard and our proposed versions of BDPT to convergence and compared the resulting images, verifying that both algorithms converge to the same result. We did this to numerically confirm the unbiasedness and consistency of our proposed rendering algorithm. We use scene dimensions comparable to real-world scenes, and gate widths based on real cameras used in analogous situations. To enable meaningful comparisons, we also report gate widths relative to scene size (percentage of half the maximum time-of-flight of directly-reflected
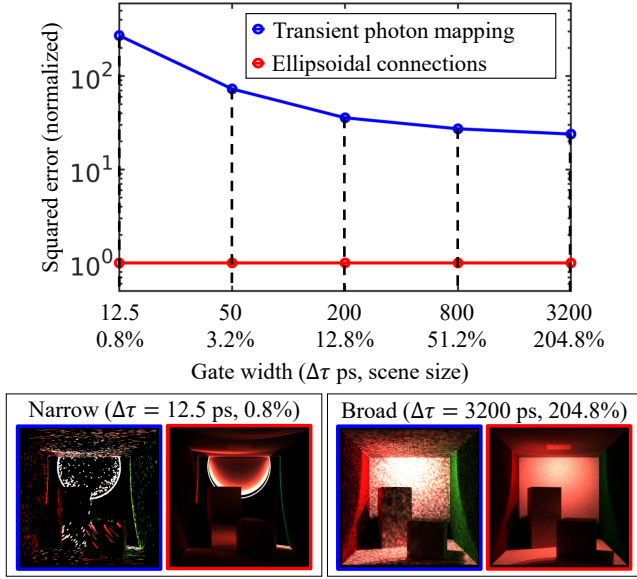
Fig. 5. **Comparison with transient photon mapping [Jarabo et al. 2014].** We compare squared error (computed over 50 renderings and compared to groundtruth rendered with standard BDPT for 40,000 samples-per-pixel) and visualize renderings produced, for the same scene and different gate widths, using BDPT with ellipsoidal connections and transient photon mapping. All images are rendered for 60 s.
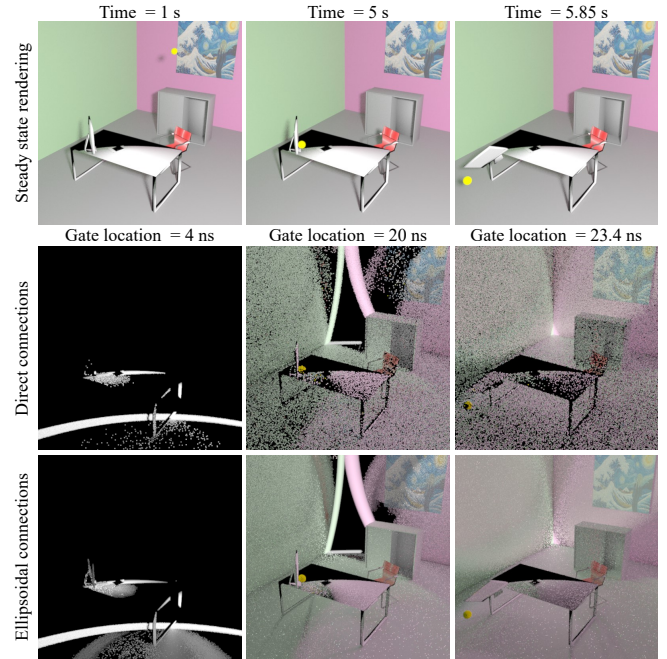

Fig. 6. **Transient imaging in dynamic scenes.** To render transients of dynamic scenes as captured by gated transient cameras, we have to render a sequence of time-gated frames, each for a slightly different scene due to motion. From top to bottom, we show steady-state renderings, transient renderings with direct connections, and transient renderings with ellipsoidal connections. Different columns are different frames of the video. All images are rendered for 15 s, with a gate width $\Delta\tau = 200$ ps (1.74% of scene size), and per-gate exposure 50 ms. See supplement for entire video.

photons). All of the comparisons shown are same-time comparisons, with rendering times reported separately for each experiment. Except where mentioned otherwise, we use our own implementation of BDPT for time-of-flight rendering, with either direct or ellipsoidal connections. Our implementation is publicly available [Pediredla et al. 2019b], and is based on the Mitsuba renderer [Jakob 2010]. All experiments are run on an Amazon EC2 c4.8xlarge instance with 36 cores. All rendered images are shown in the linear RGB color space.

Quantitative comparison. We perform a series of experiments to quantify the relative performance of BDPT with ellipsoidal and direct connections. For these experiments, we use variants of the Cornell box scene, scaled to approximately one-meter length on all sides. We image this scene with a time-gated sensor, with pathlength importance function W_τ as in Equation (3). We compare the performance of the two algorithms as we vary three parameters: temporal gate width, geometric scene complexity, and BRDF specularity.

In Figure 4(a), we compare the performance of the two BDPT variants for different gate widths $\Delta\tau$. For each gate width, we perform 50 renderings and compute variance statistics, shown at the top of Figure 4(a). We observe that, for very small gate widths, ellipsoidal connections result in a strong reduction in variance. Conversely, as the gate width increases, the increased cost of ellipsoidal connections starts to outweigh the utility of generating paths that are guaranteed to be within the time gate. Note that, as shown in the insets of Figure 4(a), at $\Delta\tau = 3200$ ps, the rendering is almost an instance of steady-state rendering, where all paths are accepted regardless of pathlength. The insets of Figure 4(a) visualize these differences, where we observe that for the narrow gate, the rendering produced using ellipsoidal connections is visibly less noisy.

In Figure 4(b), we characterize the effect of geometric scene complexity on the two BDPT variants. For this comparison, we replace the square boxes with a Stanford bunny, which we tessellate to progressively increase the total number of triangles in the scene. The top of Figure 4(b) plots variance as a function of triangle number, with variance computed as before. We observe that the performance of direct connections remains relatively unaffected by the increase in triangles, whereas ellipsoidal connections become considerably slower. The tipping point between the two algorithms is at around 44,000 triangles. The insets compare renderings from the two algorithms for the cases of the simplest and most complex scene.

Finally, in Figure 4(c), we quantify how changes in reflectance affect the two BDPT variants. For this, we assign to each box in the scene a rough-specular BRDF (using the GGX model [Walter et al. 2007]), and we progressively decrease the roughness parameter, making the boxes more specular. For each specularity setting, we compute variance statistics as before, and plot them in the top of Figure 4(c). We observe that the performance of both direct and ellipsoidal connections worsens as the objects in the scene become more specular. Ellipsoidal connections deteriorate faster, eventually becoming worse than direct connections for the most specular BRDF.
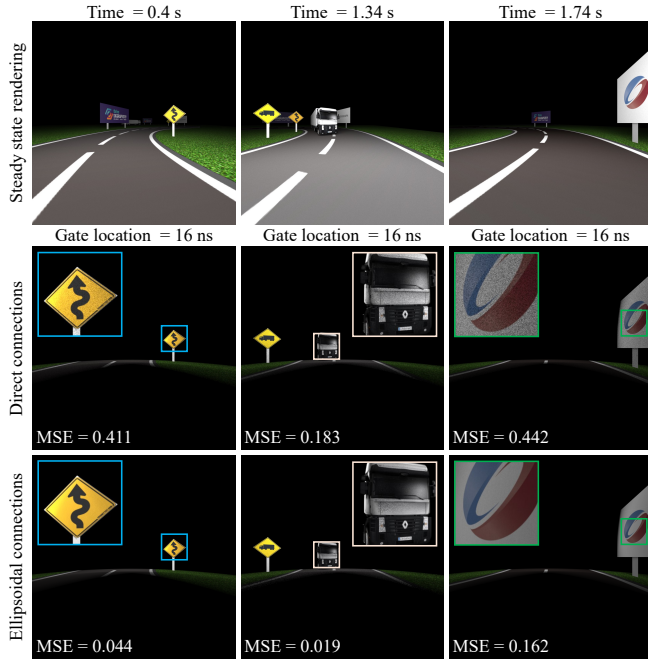
Time = 0.4 s   Time = 1.34 s   Time = 1.74 s

Gate location = 16 ns   Gate location = 16 ns   Gate location = 16 ns

MSE = 0.411   MSE = 0.183   MSE = 0.442

MSE = 0.044   MSE = 0.019   MSE = 0.162

Fig. 7. Proximity detection camera. We simulate measurements from such a camera, equipped on a travelling automobile. From top to bottom, we show steady state renderings, time-gated renderings with direct connections, and time-gated renderings with ellipsoidal connections. Different columns are different frames of the video. All images are rendered for 10 s, with gate width $\Delta\tau = 200$ ps (1.14% of scene size). The MSE improvement for the shown frames is 9.25×, 9.39×, 2.73×. See supplement for entire video.

The worse performance, in that case, can be attributed to the fact that ellipsoidal connections create two path edges ($x_e \rightarrow x_s$ and $x_e \rightarrow x_c$) that are not importance-sampled from the BRDF, whereas direct connections add only one such edge ($x_c \rightarrow x_s$).

Comparison with transient photon mapping. We additionally compare our algorithm with the transient photon mapping technique of Jarabo et al. [2014]. This is an extension to the transient setting of steady-state photon mapping techniques [Hachisuka et al. 2008; Jensen 2001], which operate in two passes: In the first pass, a *photon map* is created using paths traced from the light source. In the second pass, the contributions of rays traced from the detector are computed through kernel-density estimation based on the photon map from the first pass. In the time-gated setting, the kernel used for density estimation additionally weights photons based on how much their time-of-flight deviates from the target time gate.

In Figure 5, we compare BDPT with ellipsoidal connections and the implementation of transient photon mapping provided by Jarabo et al. [2014]. We use the settings of Figure 4(a), with three differences: First, as the photon mapping implementation only supports point sources, we replace the area source on the roof of the Cornell box with a point source slightly below the roof. Second, as the photon mapping implementation is single-threaded, we run it for 36×60 s, to achieve a same-time comparison with our multi-threaded code for ellipsoidal connections (60 s on a 36-core machine). Third, as photon mapping is biased, we use squared error (bias^2 + variance) for this

(a) Scene setting   (b) Steady state rendering

MSE = 18.81   MSE = 5.58

(c) Time gated rendering   (d) Time gated rendering
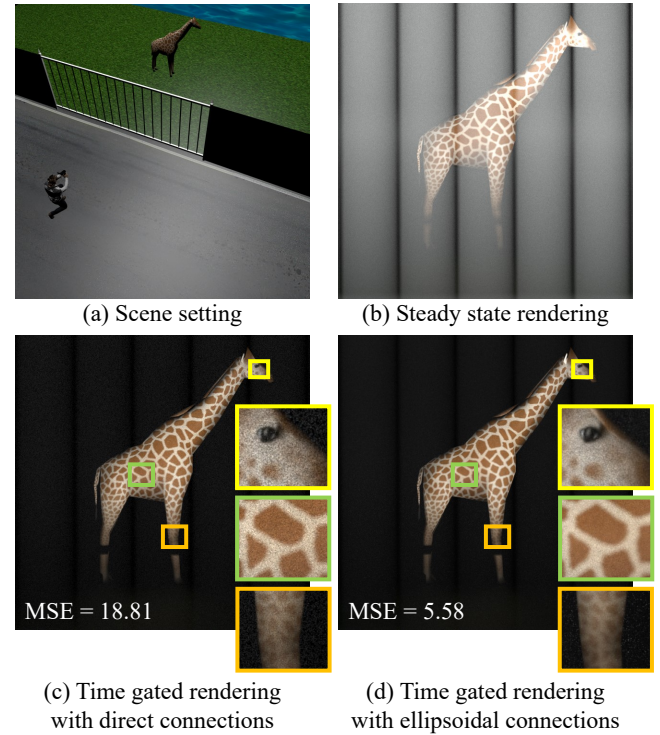with direct connections   with ellipsoidal connections

Fig. 8. Imaging through partial occluders. We simulate lens-based cameras used to image through a fence. All images are rendered for 6 min, with gate width $\Delta\tau = 1$ ns (2.35% of scene size). The MSE improvement is 3.37×.

comparison. We observe that BDPT with ellipsoidal connections results in lower error and less noisy images for all gate widths.

Transient cameras in dynamic scenes. As discussed in Section 2, transient cameras such as ICCDs or Kerr gates operate by capturing sequences of time-gated images, one at a time. In dynamic scenes, each such time-gated frame effectively images a different scene due to scene motion. Consequently, simulating this situation requires performing a sequence of time-gated rendering operations. We demonstrate this in Figure 6, where we compare the relative performance of BDPT with direct and ellipsoidal connections for this time-gated rendering task. We show a few frames of transient sequences from a dynamic scene, rendered for a transient camera with a per-frame gate width of 200 ps, typical of ICCDs [Cester et al. 2019]. We observe that, at all frames, ellipsoidal connections produce renderings that are considerably less noisy.

We emphasize that, when transient rendering a static scene, ellipsoidal connections would not offer a performance advantage over direct connections. In that case, temporal path reuse means that all paths sampled by BDPT with direct connections have high importance, and therefore it is not necessary to use ellipsoidal connections.

Proximity sensors. As discussed in Section 2, fixed time-gated sensors are increasingly used as proximity sensors in automobiles. Simulating these sensors is a time-gated rendering task, for which ellipsoidal connections can be beneficial. We show this in Figure 7, where we simulate a vehicle traveling through a road scene, equipped with

(a) BDPT with direct connections
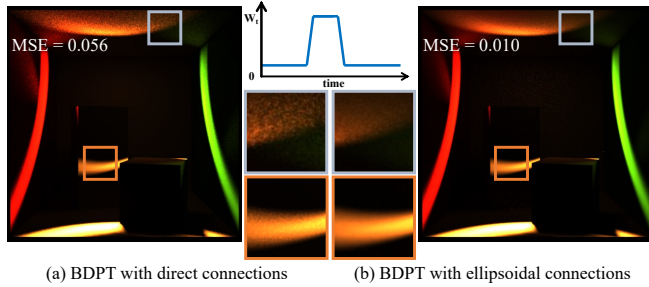(b) BDPT with ellipsoidal connections

Fig. 9. **CW-ToF depth-selective camera.** We simulate a CW-ToF camera using modulation codes resulting in depth-selectivity [Tadano et al. 2015]. The inset at the top shows the corresponding pathlength importance function, with a high-importance area of width $\Delta\tau = 20$ ps (1.28% of scene size). All images are rendered for 60 s. The MSE improvement is 5.49×.

(a) Scene
(b) Temporal Focusing with Direct Connections
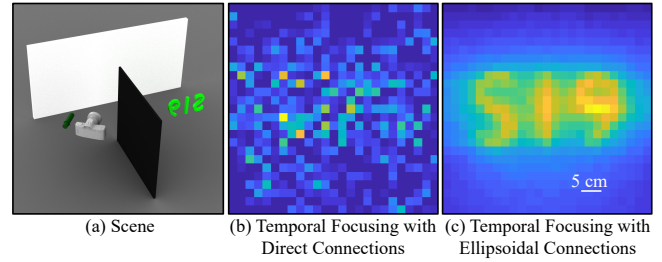(c) Temporal Focusing with Ellipsoidal Connections

Fig. 10. **Non-line-of-sight imaging with temporal focusing.** We simulate the system of Pediredla et al. [2019a]. The full scene is shown in (a), with a time-gated camera on one side of a barrier, and three objects (alphanumerics S, 1, 9) on the other side. The temporal focusing camera scans one hidden voxel at a time. We render these per-voxel measurements using direct connections (b) and ellipsoidal connections (c). Each voxel measurement is rendered for 17 s, with gate width $\Delta\tau = 4$ ps (0.4% of scene size).

a time-gated proximity camera. The figure shows a few frames of the rendered sequence, where we observe that ellipsoidal connections outperform direct connections.

Imaging through partial occluders. As discussed in Section 2, narrow time-gated cameras can be used to enhance images captured through partial occluders such as semi-blocking interfaces (e.g., fences) or participating media (e.g., fog). We simulate such a scenario in Figure 8. We create a zoo scene with a giraffe inside a fence and a photographer outside the fence (Figure 8(a)). Using a lens-based steady-state camera results in a blurry image (8(b)). By contrast, using a time-gated camera produces cleaner images (Figure 8(c,d)). We also observe that, even though the time gate used is not as narrow as before ($\Delta\tau = 1$ ns, or 2.35% of scene size), rendering with ellipsoidal connections is still beneficial for this larger scene.

Depth-selective CW-ToF cameras. Depth selectivity analogous to the previous case can also be achieved using a CW-ToF camera, as shown by Tadano et al. [2015]. We simulate this camera in Figure 9, where we compare renderings produced with direct connections and ellipsoidal connections. The function W_τ in this case has low (but non-zero) values for all pathlengths, except for a narrow trapezoidal area of width $\Delta\tau = 20$ ps (1.28% of scene size). Therefore, unlike the case of a time-gated sensor, here the contributions of all paths are non-zero, albeit in most cases very small. Comparing to ellipsoidal connections, we observe that the ability to importance sample the pathlength importance function W_τ results in significant improvement in the quality of the rendered image.

We emphasize again that ellipsoidal connections would not offer an advantage for simulating a conventional CW-ToF camera with a sinusoidal pathlength importance function. The performance benefits in Figure 9 are due to the special modulation codes used by Tadano et al. [2015], which result in depth-selectivity.

Non-line-of-sight imaging with temporal focusing. Finally, we show simulations for an NLOS imaging application. We follow Pediredla et al. [2019a], who use a temporal focusing technique to scan voxel-by-voxel a "looking around the corner" scene, as in Figure 10. Such a temporal focusing camera requires narrow time-gating for high-fidelity reconstruction. Following Pediredla et al. [2019a], we simulate such a camera with gate width $\Delta\tau = 4$ ps. In Figure 10(b,c), we

show scanning results simulated by rendering temporal focusing measurements with direct and ellipsoidal connections. For the same rendering time of around 17 s per voxel measurement, the scene is rendered with high fidelity using ellipsoidal connections, whereas it is indistinguishable from noise using direct connections.

7 LIMITATIONS AND FUTURE DIRECTIONS

We proposed combining BDPT with ellipsoidal connections as a way to sample paths that satisfy a predetermined pathlength constraint. We have used this new path sampling technique to take first steps towards addressing critical shortcomings of existing algorithms for *time-gated rendering tasks*, namely the rejection of large numbers of sampled paths. In the rest of this section, we discuss limitations of path sampling using BDPT with ellipsoidal connections, future directions towards addressing these limitations, as well as some interesting research directions that arise from the availability of techniques for pathlength-constrained path sampling.

Scalability to large meshes. In Section 5, we showed that data structures used to accelerate ray-mesh intersections can also be used to accelerate ellipsoid-scene intersections. While this provides modest speed-ups, the number of primitives that remain as candidates for intersection after tree traversal is typically much larger than in the case of ray-mesh intersection operations. Consequently, BDPT with ellipsoidal connections does not scale well to scenes with very large meshes, compared to standard BDPT. We quantified this scaling behavior in the experiments of Section 6. As we discuss in Section 5, and show in experiments, the scalability of our technique can be improved by amortizing each ellipsoid-scene intersection to perform multiple ellipsoidal connections (Section 5). Still, investigating techniques to further reduce the number of primitives that need to be intersected is an important future research direction.

Importance sampling of ellipsoidal vertices. Currently, after performing an ellipsoid-scene intersection, we select an ellipsoidal vertex uniformly at random. This is suboptimal, as ideally ellipsoidal vertices should be importance-sampled based on their throughput. Unfortunately, we do not know of an efficient algorithm to perform this sampling operation, which requires evaluating the throughput along multiple one-dimensional elliptical arcs. We experimented

with an approximate importance sampling scheme, where candidate ellipsoidal vertices are sampled with a probability proportional to the throughput evaluated at the barycenter of their corresponding triangles. However, this resulted in negligible improvement, indicating that we should explore better approximations.

Combining direct and ellipsoidal connections. Direct and ellipsoidal connections can both produce paths with zero contributions, but their failure cases occur under different conditions. A direct connection always creates a complete path, but that path may have pathlength importance W_τ zero. By contrast, an ellipsoidal connection may fail to produce a complete path altogether when the ellipsoid-scene intersection $\mathcal{E}(x_s, x_c, \tau_e) \cap \mathcal{M}$ is empty. Additionally, both types of connections can produce zero-contributing paths because of visibility terms, with this issue being more severe for ellipsoidal connections (two visibility terms, $x_e \to x_c$ and $x_e \to x_s$) than for direct ones (one visibility term, $x_s \to x_c$).

As we demonstrated empirically in Section 6, ellipsoidal connections perform better as the support of W_τ decreases, and conversely for direct connections. This motivates the question: Rather than committing to one type of connections, is there a way to optimally combine the two? It will be interesting to investigate randomized procedures that alternate between the two types of connections, in a way that adapts to the characteristics of the simulated sensor and scene, and potentially that takes advantage of information from previously sampled paths [Hachisuka et al. 2014].

Adaptive sampling and Metropolis light transport. Adaptive sampling techniques have been successful for steady-state rendering, with the key idea being that we can allocate more samples to spatial pixels that are difficult to render (e.g., edges, textures, caustics) [Zwicker et al. 2015]. BDPT with ellipsoidal constraints provides a way to adapt these techniques to time-of-flight rendering tasks, by making it possible to allocate more samples to temporal frames $I(\tau)$ that require them (e.g., high-frequency parts of a transient [Wu et al. 2014b]). This was not possible previously, as there was no way to sample paths for frames corresponding to specific pathlengths. Experimentally, we found that the increased cost of ellipsoidal connections generally outweighs the benefit of adaptive sampling, but any future discoveries improving the efficiency of ellipsoidal connections would make this approach worth revisiting.

Ellipsoidal connections additionally show promise within the framework of path-space Metropolis light transport [Veach and Guibas 1997]: Given a path, we can perturb it by removing one of its vertices and replacing it with a new ellipsoidal vertex, selected so that the new path has the same length as the original one. Using ellipsoidal connections as pathlength-preserving perturbation operations open up the possibility of using Metropolis light transport algorithms to efficiently explore path manifolds defined by pathlength constraints [Jakob and Marschner 2012].

Inverse rendering. We have focused on forward rendering tasks, that is, simulating sensor measurements of a known scene. However, physics-based rendering is also an indispensable tool for *inverse rendering* tasks [Li et al. 2018; Marschner and Greenberg 1998]: Rendering operations are used within optimization procedures that attempt to match a set of sensor measurements, in order to infer

properties of the scene these measurements were captured from. Different types of time-of-flight imaging have recently been used, together with inverse rendering optimization, for inverse scattering [Gkioulekas et al. 2016] and non-line-of-sight imaging [O'Toole et al. 2018; Tsai et al. 2019; Velten et al. 2012] applications. We expect that our algorithm will be useful for such inverse rendering tasks.

REFERENCES

2019. FESCA-100 Femtosecond streak camera. https://www.hamamatsu.com/us/en/product/photometry-systems/streak-camera/fesca-100-femtosecond-streak-camera/index.html. Accessed: 01-12-2019.

2019. Laser Range-Gated Imaging for Imaging at Long Ranges and Through Obscurants. http://www.sensorsinc.com/applications/military/laser-range-gating. Accessed: 01-12-2019.

Marco Ament, Christoph Bergmann, and Daniel Weiskopf. 2014. Refractive radiative transfer equation. *ACM Transactions on Graphics (TOG)* 33, 2 (2014), 17.

Luke Anderson, Tzu-Mao Li, Jaakko Lehtinen, and Frédo Durand. 2017. Aether: An embedded domain specific sampling language for Monte Carlo rendering. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 99.

Victor Arellano, Diego Gutierrez, and Adrian Jarabo. 2017. Fast back-projection for non-line of sight reconstruction. *Optics Express* 25, 10 (2017), 11574–11583.

Ian M Baker, Stuart S Duncan, and Jeremy W Copley. 2004. A low-noise laser-gated imaging system for long-range target identification. In *Infrared Technology and Applications XXX*, Vol. 5406. International Society for Optics and Photonics, 133–145.

Mauro Buttafava, Jessica Zeman, Alberto Tosi, Kevin Eliceiri, and Andreas Velten. 2015. Non-line-of-sight imaging using a time-gated single photon avalanche diode. *Optics express* 23, 16 (2015), 20997–21011.

Lucrezia Cester, Ashley Lyons, Maria Braidotti, and Daniele Faccio. 2019. Time-of-Flight Imaging at 10 ps Resolution with an ICCD Camera. *Sensors* 19, 1 (2019), 180.

Susan Chan, Ryan E Warburton, Genevieve Gariepy, Jonathan Leach, and Daniele Faccio. 2017. Non-line-of-sight tracking of people at long range. *Optics express* 25, 9 (2017), 10109–10117.

Subrahmanyan Chandrasekhar. 1960. *Radiative transfer.* Dover Publication.

BB Das, KM Yoo, and RR Alfano. 1993. Ultrafast time-gated imaging in thick tissues: a step toward optical mammography. *Optics letters* 18, 13 (1993), 1092–1094.

Ofer David, Norman S Kopeika, and Boaz Weizer. 2006. Range gated active night vision system for automobiles. *Applied optics* 45, 28 (2006), 7248–7254.

Philip Dutre, Philippe Bekaert, and Kavita Bala. 2006. *Advanced global illumination.* AK Peters/CRC Press.

Daniele Faccio and Andreas Velten. 2018. A trillion frames per second: the techniques and applications of light-in-flight photography. *Reports on Progress in Physics* 81, 10 (2018), 105901.

Genevieve Gariepy, Nikola Krstajić, Robert Henderson, Chunyong Li, Robert R Thomson, Gerald S Buller, Barmak Heshmat, Ramesh Raskar, Jonathan Leach, and Daniele Faccio. 2015. Single-photon sensitive light-in-fight imaging. *Nature communications* 6 (2015), 6021.

Ioannis Gkioulekas, Anat Levin, Frédo Durand, and Todd Zickler. 2015. Micron-scale light transport decomposition using interferometry. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 37.

Ioannis Gkioulekas, Anat Levin, and Todd Zickler. 2016. An Evaluation of Computational Imaging Techniques for Heterogeneous Inverse Scattering. *European Conference on Computer Vision* (2016).

Yoav Grauer and Ezri Sonn. 2015. Active gated imaging for automotive safety applications. In *Video Surveillance and Transportation Imaging Applications 2015*, Vol. 9407. International Society for Optics and Photonics, 94070F.

Tobias Gruber, Frank Julca-Aguilar, Mario Bijelic, Werner Ritter, Klaus Dietmayer, and Felix Heide. 2019. Gated2Depth: Real-time Dense Lidar from Gated Images. *arXiv preprint arXiv:1902.04997* (2019).

Mohit Gupta, Shree K Nayar, Matthias B Hullin, and Jaime Martin. 2015. Phasor imaging: A generalization of correlation-based time-of-flight imaging. *ACM Transactions on Graphics (ToG)* 34, 5 (2015), 156.

Mohit Gupta, Andreas Velten, Shree K Nayar, and Eric Breitbach. 2018. What are optimal coding functions for time-of-flight imaging? *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 13.

Toshiya Hachisuka, Anton S Kaplanyan, and Carsten Dachsbacher. 2014. Multiplexed metropolis light transport. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 100.

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. 2008. Progressive photon mapping. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 130.

Felix Heide, Matthias B Hullin, James Gregson, and Wolfgang Heidrich. 2013. Low-budget transient imaging using photonic mixer devices. *ACM Transactions on Graphics (ToG)* 32, 4 (2013), 45.

Felix Heide, Matthew O'Toole, Kai Zhang, David Lindell, Steven Diamond, and Gordon Wetzstein. 2019. Non-line-of-sight Imaging with Partial Occluders and Surface Normals. *ACM Transactions on Graphics (ToG)* (2019).

Felix Heide, Lei Xiao, Wolfgang Heidrich, and Matthias B Hullin. 2014. Diffuse mirrors: 3D reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3222–3229.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: A markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics* (2012).

Adrian Jarabo. 2012. Femto-photography: Visualizing light in motion. *Master's thesis, Universidad de Zaragoza* (2012).

Adrian Jarabo, Julio Marco, Adolfo Muñoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. 2014. A framework for transient rendering. *ACM Transactions on Graphics (ToG)* 33, 6 (2014), 177.

Adrian Jarabo, Belen Masia, Julio Marco, and Diego Gutierrez. 2017. Recent advances in transient imaging: A computer graphics and vision perspective. *Visual Informatics* 1, 1 (2017), 65–79.

Henrik Wann Jensen. 2001. *Realistic image synthesis using photon mapping*. AK Peters/CRC Press.

Ahmed Kirmani, Tyler Hutchison, James Davis, and Ramesh Raskar. 2009. Looking around the corner using transient imaging. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 159–166.

Eric P Lafortune and Yves D Willems. 1996. Rendering participating media with bidirectional path tracing. In *Rendering Techniques' 96*. Springer, 91–100.

Joseph R Lakowicz, Henryk Szmacinski, Kazimierz Nowaczyk, Klaus W Berndt, and Michael Johnson. 1992. Fluorescence lifetime imaging. *Analytical biochemistry* 202, 2 (1992), 316–330.

Martin Laurenzis and Andreas Velten. 2014. Nonline-of-sight laser gated viewing of scattered photons. *Optical Engineering* 53, 2 (2014), 023102.

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable monte carlo ray tracing through edge sampling. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 222.

Jingyu Lin, Yebin Liu, Matthias B Hullin, and Qionghai Dai. 2014. Fourier analysis on transient imaging with a multifrequency time-of-flight camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3230–3237.

Julio Marco, Ibón Guillén, Wojciech Jarosz, Diego Gutierrez, and Adrian Jarabo. 2018. Progressive Transient Photon Beams. *arXiv preprint arXiv:1805.09562* (2018).

Julio Marco, Quercus Hernandez, Adolfo Muñoz, Yue Dong, Adrian Jarabo, Min H Kim, Xin Tong, and Diego Gutierrez. 2017a. DeepToF: off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 219.

Julio Marco, Wojciech Jarosz, Diego Gutierrez, and Adrian Jarabo. 2017b. Transient photon beams. In *ACM SIGGRAPH 2017 Posters*. ACM, 52.

Stephen Robert Marschner and Donald P Greenberg. 1998. *Inverse rendering for computer graphics*. Cornell University.

Nikhil Naik, Shuang Zhao, Andreas Velten, Ramesh Raskar, and Kavita Bala. 2011. Single View Reflectance Capture Using Multiplexed Scattering and Time-of-flight Imaging. *ACM Trans. Graph.* 30, 6 (2011), 171:1–171:10.

Matthew O'Toole, Felix Heide, Lei Xiao, Matthias B Hullin, Wolfgang Heidrich, and Kiriakos N Kutulakos. 2014. Temporal frequency probing for 5D transient analysis of global light transport. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 87.

Matthew O'Toole, David B Lindell, and Gordon Wetzstein. 2018. Confocal non-line-of-sight imaging based on the light-cone transform. *Nature* 555, 7696 (2018), 338.

Adithya Pediredla, Mauro Buttafava, Alberto Tosi, Oliver Cossairt, and Ashok Veeraraghavan. 2017a. Reconstructing rooms using photon echoes: A plane based model and reconstruction algorithm for looking around the corner. In *2017 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 1–12.

Adithya Pediredla, Akshat Dave, and Ashok Veeraraghavan. 2019a. SNLOS: Non-line-of-sight Scanning through Temporal Focusing. (2019).

Adithya Pediredla, Nathan Matsuda, Oliver Cossairt, and Ashok Veeraraghavan. 2017b. Linear systems approach to identifying performance bounds in indirect imaging. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 6235–6239.

Adithya Pediredla, Ashok Veeraraghavan, and Gkioulekas Ioannis. 2019b. Mitsuba Time-of-Flight renderer. https://github.com/cmu-ci-lab/MitsubaToFRenderer.

Christoph Peters, Jonathan Klein, Matthias B Hullin, and Reinhard Klein. 2015. Solving trigonometric moment problems for fast transient imaging. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 220.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically based rendering: From theory to implementation*. Morgan Kaufmann.

Guy Satat, Barmak Heshmat, Dan Raviv, and Ramesh Raskar. 2016. All photons imaging through volumetric scattering. *Scientific reports* 6 (2016), 33946.

Guy Satat, Matthew Tancik, and Ramesh Raskar. 2018. Towards photography through realistic fog. In *Computational Photography (ICCP), 2018 IEEE International Conference on*. IEEE, 1–10.

B Schmidt, S Laimgruber, W Zinth, and P Gilch. 2003. A broadband Kerr shutter for femtosecond fluorescence spectroscopy. *Applied Physics B* 76, 8 (2003), 809–814.

Shikhar Shrestha, Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. 2016. Computational imaging with multi-camera time-of-flight systems. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 33.

Adam Smith, James Skorupski, and James Davis. 2008. Transient rendering. (2008).

Ryuichi Tadano, Adithya Kumar Pediredla, and Ashok Veeraraghavan. 2015. Depth selective camera: A direct, on-chip, programmable technique for depth selectivity in photography. In *Proceedings of the IEEE International Conference on Computer Vision*. 3595–3603.

Akira Takahashi, Mitsunori Nishizawa, Yoshinori Inagaki, Musubu Koishi, and Katsuyuki Kinoshita. 1994. New femtosecond streak camera with temporal resolution of 180 fs. In *Generation, amplification, and measurement of ultrashort laser pulses*, Vol. 2116. International Society for Optics and Photonics, 275–285.

Christos Thrampoulidis, Gal Shulkind, Feihu Xu, William T Freeman, Jeffrey Shapiro, Antonio Torralba, Franco Wong, and Gregory Wornell. 2018. Exploiting occlusion in non-line-of-sight active imaging. *IEEE Transactions on Computational Imaging* (2018).

Chia-Yin Tsai, Kiriakos N Kutulakos, Srinivasa G Narasimhan, and Aswin C Sankaranarayanan. 2017. The geometry of first-returning photons for non-line-of-sight imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7216–7224.

Chia-Yin Tsai, Aswin C. Sankaranarayanan, and Ioannis Gkioulekas. 2019. Beyond Volumetric Albedo—A Surface Optimization Framework for Non-Line-of-Sight Imaging. In *CVPR*.

Eric Veach. 1997. *Robust monte carlo methods for light transport simulation*. Number 1610. Stanford University PhD thesis.

Eric Veach and Leonidas Guibas. 1995a. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 145–167.

Eric Veach and Leonidas J Guibas. 1995b. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 419–428.

Eric Veach and Leonidas J Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 65–76.

Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G Bawendi, and Ramesh Raskar. 2012. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications* 3 (2012), 745.

Andreas Velten, Di Wu, Adrian Jarabo, Belen Masia, Christopher Barsi, Chinmaya Joshi, Everett Lawson, Moungi Bawendi, Diego Gutierrez, and Ramesh Raskar. 2013. Femto-photography: capturing and visualizing the propagation of light. *ACM Transactions on Graphics (ToG)* 32, 4 (2013), 44.

Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*. Eurographics Association, 195–206.

Di Wu, Andreas Velten, Matthew O'Toole, Belen Masia, Amit Agrawal, Qionghai Dai, and Ramesh Raskar. 2014a. Decomposing global light transport using time of flight imaging. *International journal of computer vision* 107, 2 (2014), 123–138.

Di Wu, Gordon Wetzstein, Christopher Barsi, Thomas Willwacher, Qionghai Dai, and Ramesh Raskar. 2014b. Ultra-fast lensless computational imaging through 5D frequency analysis of time-resolved light transport. *International journal of computer vision* 110, 2 (2014), 128–140.

Shumian Xin, Sotiris Nousias, Kyriakos N. Kutulakos, Aswin C. Sankaranarayanan, Srinivasa G. Narasimhan, and Ioannis Gkioulekas. 2019. A Theory of Fermat Paths for Non-Line-of-Sight Shape Reconstruction. In *CVPR*.

Pingping Zhan, Junyi Tong, Wenjiang Tan, and Shichao Xu. 2016. Comparison between traditional and heterodyned optical Kerr gated imaging for CS 2 and Te glass Kerr media. *Optical Engineering* 55, 11 (2016), 110502.

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and S-E Yoon. 2015. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. In *Computer graphics forum*, Vol. 34. Wiley Online Library, 667–681.