Differentiable Transient Rendering

SHINYOUNG YI, DONGGUN KIM, and KISEOK CHOI, KAIST
ADRIAN JARABO and DIEGO GUTIERREZ, Universidad de Zaragoza, I3A
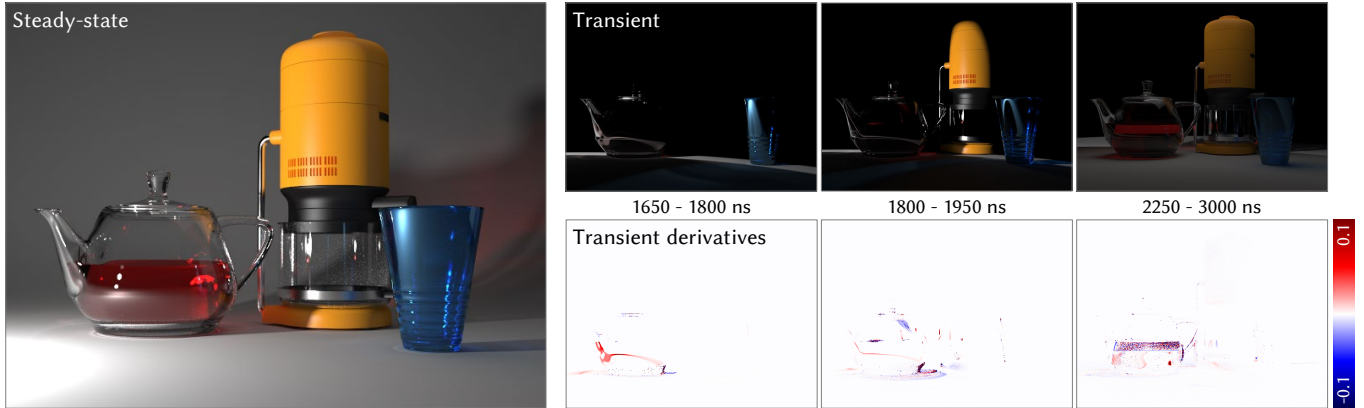MIN H. KIM, KAIST

Fig. 1. Our general-purpose differentiable transient rendering framework allows to compute derivates of complex, multi-bounce transient sequences with respect to scene parameters, even in the presence of discontinuous light and sensor functions. The figure shows steady-state and transient renders of a table-top scene with light coming from the left, then being bounced back by two off-camera diffusers. The bottom row shows the transient light transport scene derivatives with respect to the index of refraction of the red tea. Please refer to Figure 10 for additional results optimizing the index of refraction of the tea in the teapot, which in turn changes the speed of light.

Recent differentiable rendering techniques have become key tools to tackle many inverse problems in graphics and vision. Existing models, however, assume steady-state light transport, i.e., infinite speed of light. While this is a safe assumption for many applications, recent advances in ultrafast imaging leverage the wealth of information that can be extracted from the exact time of flight of light. In this context, physically-based transient rendering allows to efficiently simulate and analyze light transport considering that the speed of light is indeed finite. In this paper, we introduce a novel differentiable transient rendering framework, to help bring the potential of differentiable approaches into the transient regime. To differentiate the transient path integral we need to take into account that scattering events at path vertices are no longer independent; instead, tracking the time of flight of light requires treating such scattering events at path vertices jointly as a multidimensional, evolving manifold. We thus turn to the generalized transport theorem, and introduce a novel *correlated importance* term, which links the time-integrated contribution of a path to its light throughput, and allows us to handle discontinuities in the light and sensor functions. Last, we present results in several challenging scenarios where the time of flight of light plays an important role such as optimizing indices of refraction, non-line-of-sight tracking with nonplanar relay walls, and non-line-of-sight tracking around two corners.

Authors' addresses: Shinyoung Yi; Donggun Kim; Kiseok Choi, KAIST, School of Computing, Daejeon, South Korea, 34141; Adrian Jarabo; Diego Gutierrez, Universidad de Zaragoza, I3A, , Zaragoza, 50018; Min H. Kim, KAIST, School of Computing, Daejeon, South Korea, 34141, corresponding_author:minhkim@kaist.ac.kr.

1 INTRODUCTION

Physically-based differentiable rendering deals with the computation of the derivatives of radiometric measurements, according to changes in scene parameters (see recent references [Kato et al. 2020; Zhao et al. 2020] for a wide overview of the field). It has recently become a key tool not only for inverse rendering or scene reconstruction problems that require gradient-based optimization, but also to enable the integration of physics-based simulations in machine learning pipelines, computing the loss function in rendering space. While many different approaches have been presented in recent years (e.g., [Li et al. 2018; Zhang et al. 2020, 2019]), all of them assume *steady-state* configurations, where time delays due to the propagation of light are ignored.

Transient imaging, on the other hand, has already enabled many non-trivial applications in computer vision and scene recognition [Jarabo et al. 2017], such as visualizing light in motion [Heide et al. 2013; Velten et al. 2013], looking around corners [Liu et al. 2019a; O'Toole et al. 2018], looking through turbid media [Heide et al.

2014; Wu et al. 2018], or the decomposition of global illumination components [Marco et al. 2021; Wu et al. 2014], to name a few. Transient rendering techniques, which lift the assumption that light speed is infinite and take explicitly into account light's time of flight, have thus become an increasingly relevant tool to accurately simulate scenarios and hypotheses without requiring cumbersome data capture sessions, as well as to help provide rigorous mathematical analyses [Jarabo et al. 2014; Pediredla et al. 2019].

Computing the time-resolved radiance derivatives as a function of the scene is currently limited to expensive and fragile finite differences [Gkioulekas et al. 2016; Iseringhausen and Hullin 2020], which hinders the applicability of current transient rendering in inverse problems or machine learning approaches. To overcome this, we introduce *path-space differentiable transient rendering*. Our goal is to differentiate the transient path integral [Jarabo et al. 2014] to help unlock the potential of differential approaches in transient scenarios. We make the observation that, when tracking the time of flight of light, scattering events at path vertices are no longer independent, as assumed in recent work [Zhang et al. 2020]. As a consequence, we are no longer dealing with two-dimensional evolving manifolds, which in turn means that we cannot apply the Reynolds transport relation recursively between path vertices [Zhang et al. 2020].

We thus rely on the *generalized transport theorem* [Seguin and Fried 2014], applied on the *higher-dimensional manifold* that results from all path vertices contributing to the temporal domain, and introduce a novel *correlated importance* term, which links the time-integrated contribution of a path to its light throughput and allows us to handle discontinuous light and sensor functions. From our theoretical formulation, we then demonstrate that, despite the higher dimensionality of transient light transport, our practical formulation of the boundary integral converges to Zhang's original formulation of differentiable steady-state path tracing [2020]. In that sense, our work can also be seen as a generalization of the existing path-space formulation, lifting the steady-state assumption of independent scattering events while being able to handle discontinuities in the light and sensor functions. In addition, we show how to incorporate our work in a Monte Carlo framework, defining estimators for both the interior and the boundary integrals.

We validate our results against the baseline transient path integral technique and using finite differences. In addition, we demonstrate how our formulation enables several novel applications in challenging scenarios, where the time of flight of light plays an important role and traditional non-line-of-sight (NLOS) imaging has been inapplicable. In particular, we show optimization results for indices of refraction; NLOS tracking with a nonplanar, wavy relay wall; and tracking the motion of a hidden object around two corners. These last two examples are the first demonstrations of NLOS imaging under such challenging scenarios, and demonstrate the potential of differential transient rendering for solving difficult inverse problems. We hope that our code and datasets[1] will help future developments on the field.

---

[1]http://vclab.kaist.ac.kr/siggraphasia2021/

2 RELATED WORK

For a wide overview of the applications of transient imaging, we refer the reader to existing surveys on the topic [Jarabo et al. 2017; Satat et al. 2016]. In the following, we discuss related works that focus on transient and differentiable rendering.

Transient rendering. The term "transient rendering" was coined by Smith et al. [2008], who proposed to extend the rendering equation by including the temporal delay of light propagation. Based on this framework, several works proposed transient variations of Monte Carlo rendering [Adam et al. 2016; Ament et al. 2014; Jarabo 2012; Krivánek et al. 2013; O'Toole et al. 2014; Pitts et al. 2014]. Jarabo et al. [2014] presented a practical framework by introducing the *transient path integral*, including an efficient reconstruction technique, as well as a novel sampling strategy by adding indirect shadow vertices, to control path lengths in participating media. This framework was later extended to include vector-based effects such as polarization or fluorescence [Jarabo and Arellano 2018]. Pediredla et al. [2019] introduced ellipsoidal connections for time-gated rendering, also using additional vertices to control path length, but for surfaces instead of media. Other authors have recently focused on increasing efficiency when rendering transient light transport, usually based on simplified models [Chen et al. 2020; Iseringhausen and Hullin 2020; Pan et al. 2019; Tsai et al. 2019], or by means of filtering [Marco et al. 2019]. All these works render time-resolved zeroth-order radiance; in contrast, we develop a differentiable transient path integral, and focus on the derivatives of transient light transport.

Analysis-by-synthesis in transient imaging. Several works have leveraged transient rendering for inverse problems, using it as the generative model for gradient descent-based optimization. Iseringhausen and Hullin [2020] used an efficient three-bounce renderer for non-line-of-sight (NLOS) reconstruction based on finite differences, while Tsai et al. [2019] explicitly derived the light transport gradients of hidden surface properties. None of these works account for visibility changes, and are limited to the particular problem of three-bounce NLOS reconstruction. Gkioulekas et al. [2016] analyzed the use of transient rendering in the context of stochastic gradient descent for recovering heterogeneous media. In contrast, our framework is general and does not rely on explicitly computed gradients, while taking into account complex derivatives including singularities produced by, e.g., visibility changes.

Differentiable rendering. While several special-purpose differentiable rendering systems have been proposed in the past (e.g., [Chen et al. 2019; Gkioulekas et al. 2016; Kato et al. 2018; Liu et al. 2019b]) in this work we focus on general-purpose differentiable rendering. It was first proposed in OpenDR [Loper and Black 2014], targeting scalability and efficiency; however, it was based on a simple physical model, which reduced its applicability to direct illumination only. Laine et al. [2020] included additional features such as antialiasing or texture filtering, but still limited only to direct illumination.

Li et al. [2018] proposed a general-purpose, differentiable, and physically-based rendering model, computing the differential form of the rendering equation with Monte Carlo ray tracing, while taking into account global illumination and geometric discontinuities. The

Table 1. Main symbols used in the paper.

Symbol	Description
$\bar{\mathbf{x}} = \mathbf{x}_0...\mathbf{x}_k$	Light path of $k+1$ vertices
$\mathbf{t} = t_0...t_k$	Time delays on $k+1$ vertices
Ω	Space of all light paths
Ω_k	Space of light paths of $k+1$ vertices
$\mathcal{T}$	Space of temporal delays
c	Speed of light in vacuum
η_i	Refractive index of the medium between $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$
$\text{tof}(\bar{\mathbf{x}})$	Total delay of path $\bar{\mathbf{x}}$
$L_e(\mathbf{x}_0 \to \mathbf{x}_1, t)$	Light source emission function with direction $\mathbf{x}_0 \to \mathbf{x}_1$, time delay t
$W_e(\mathbf{x}_{k-1} \to \mathbf{x}_k, t)$	Sensor sensitivity function with direction $\mathbf{x}_{k-1} \to \mathbf{x}_k$, time delay t
$S_e(\bar{\mathbf{x}})$	Correlated importance as a function of light path $\bar{\mathbf{x}}$
$\rho(\mathbf{x}_i, t_i)$	Scattering function at vertex $\mathbf{x}_i$
$G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1})$	Geometric function
$V(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1})$	Visibility function
$\mathfrak{T}(\bar{\mathbf{x}}, \mathbf{t})$	Light throughput, the product of ρ, G, and V
$\boldsymbol{\theta} = \theta_1...\theta_d$	Set of d parameters describing a scene
$\mathcal{M}(\boldsymbol{\theta})$	Manifold that evolves with $\boldsymbol{\theta}$
$\partial\mathcal{M}(\boldsymbol{\theta})$	Boundary of $\mathcal{M}(\boldsymbol{\theta})$
$v(\mathbf{x})$	Local velocity at $\mathbf{x}$
$\mathcal{V}_{\mathcal{M}}(\mathbf{x})$	Scalar normal velocity of $\mathcal{M}(\boldsymbol{\theta})$ at $\mathbf{x}$
$\mathcal{V}_{\partial\overline{\mathcal{M}}}(\mathbf{x})$	Scalar normal velocity of $\overline{\partial\mathcal{M}(\boldsymbol{\theta})}$ at $\mathbf{x}$
$\kappa(\mathbf{x})$	Total curvature at $\mathbf{x}$
$I(\boldsymbol{\theta})$	Rendered image as a function of $\boldsymbol{\theta}$
$f(\bar{\mathbf{x}})$	Contribution to I of path $\bar{\mathbf{x}}$
$f_t(\bar{\mathbf{x}}, \mathbf{t})$	Time-resolved contribution to I of path $\bar{\mathbf{x}}$ with time delay $\mathbf{t}$
$f_{\mathcal{T}}(\bar{\mathbf{x}})$	Time-integrated contribution to I of path $\bar{\mathbf{x}}$
$\dot{f}_{\mathcal{T}}(\bar{\mathbf{x}}, \boldsymbol{\theta})$	Derivative of $f_{\mathcal{T}}$ for path $\bar{\mathbf{x}}$ w.r.t. $\boldsymbol{\theta}$
$\overset{\square}{f}_{\mathcal{T}}(\bar{\mathbf{x}}, \boldsymbol{\theta})$	Normal derivative of $f_{\mathcal{T}}$ for $\bar{\mathbf{x}}$ w.r.t. $\boldsymbol{\theta}$
$\Delta f_{\mathcal{T}}(\bar{\mathbf{x}})$	Boundary contribution of function $f_{\mathcal{T}}$ for $\bar{\mathbf{x}}$ on $\partial\overline{\Omega}$
$\Delta\overline{\Omega}[f_{\mathcal{T}}](\boldsymbol{\theta})$	Set of discontinuities of function $f_{\mathcal{T}}$ on Ω w.r.t. $\boldsymbol{\theta}$
$\partial\overline{\Omega}[f_{\mathcal{T}}](\boldsymbol{\theta})$	Extended boundary of Ω w.r.t. $f_{\mathcal{T}}$ and $\boldsymbol{\theta}$
$\mathbf{n}(\mathbf{x}_i)$	Unit normal at $\mathbf{x}_i$
$\delta(t_i)$	Dirac delta at t_i
$\|\mathbf{x}_i - \mathbf{x}_{i-1}\|^{\square}$	Normal derivative of $\|\mathbf{x}_i - \mathbf{x}_{i-1}\|$ w.r.t. $\boldsymbol{\theta}$

authors proposed searching for the discontinuities on the integral using edge sampling. The discontinuity handling was later improved by the works of Loubet et al. [2019], who reparameterized the integral on top of the differentiable renderer Mitsuba 2 [Nimier-David et al. 2019] to ease computations at the price of introducing bias, and Bangaru et al. [2020], who combined edge and area sampling for increasing robustness and efficiency. Zhang et al. [2019] extended the differential rendering equation to the volumetric rendering equation, modeling light transport in participating media. These approaches work on the local domain of the rendering equation, which makes it difficult to extend them to bidirectional rendering algorithms. Zhang et al. [2020] introduced a differential path-space light transport integral suitable for bidirectional methods. Our work builds on top of such integral, and generalizes it to time-resolved light transport. Finally, orthogonal to these works, Nimier-David et al. [2020] proposed an adjoint differential light transport formulation that propagates the derivatives from the camera to the light sources, dramatically improving speed and memory requirements.

3 BACKGROUND

We introduce here the main aspects of both the transient path integral and path-space differentiable rendering. Table 1 summarizes the main symbols used throughout the paper.

3.1 Transient Path Integral

The path integral [Veach 1997] models the intensity I recorded by a virtual sensor as the integral of all light paths Ω contributing to a pixel, assuming that both the emission and sensor response are orders of magnitude larger than the propagation time of light. Incorporating the temporal domain we obtain the *transient path integral* [Jarabo et al. 2014] as

$$I = \int_{\Omega} \int_{\mathcal{T}} f_t(\bar{\mathbf{x}}, \mathbf{t}) \mathrm{d}\mu(\mathbf{t}) \mathrm{d}\mu(\bar{\mathbf{x}}), \tag{1}$$

where $\mathcal{T}$ represents the space of temporal delays, $\mathbf{t} = t_0...t_k$ is the sequence of time delays on each vertex, $\mathrm{d}\mu(\mathbf{t})$ denotes temporal integration at each vertex, $\mathrm{d}\mu(\bar{\mathbf{x}})$ is the differential measure, and $\bar{\mathbf{x}} = \mathbf{x}_0...\mathbf{x}_k$ is a path of $k+1$ vertices. Vertices $\mathbf{x}_0$ and $\mathbf{x}_k$ lie on the light source and the sensor, respectively. For convenience, we define the path space as $\Omega = \cup_{k=1}^{\infty} \Omega_k$, with Ω_k being the space of all paths with k vertices. The path contribution $f_t(\bar{\mathbf{x}}, \mathbf{t})$ is given by

$$f_t(\bar{\mathbf{x}}, \mathbf{t}) = L_e(\mathbf{x}_0 \to \mathbf{x}_1, t_0)\, \mathfrak{T}(\bar{\mathbf{x}}, \mathbf{t})\, W_e(\mathbf{x}_{k-1} \to \mathbf{x}_k, t_k + \text{tof}(\bar{\mathbf{x}})), \tag{2}$$

where $\text{tof}(\bar{\mathbf{x}}) = c^{-1}\sum_{i=0}^{k-1}\|\mathbf{x}_i - \mathbf{x}_{i-1}\|\eta_i + \sum_{i=0}^{k-1} t_i$ is the total delay of path $\bar{\mathbf{x}}$ (i.e., its time of flight), with c the speed of light and η_i the index of refraction of the medium between vertices $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. The temporal delay on the emission $L_e(\mathbf{x}_0 \to \mathbf{x}_1, t_0)$ is continuous in real-world applications, ranging from a few femto-seconds in ultrashort lasers to continuous light modulation in time-of-flight cameras. The sensor sensitivity $W_e(\mathbf{x}_{k-1} \to \mathbf{x}_k, t_k + \text{tof}(\bar{\mathbf{x}}))$ models the temporal response of each specific timestamp, in addition to the spatio-angular response of the pixel. Last, the light throughput $\mathfrak{T}(\bar{\mathbf{x}}, \mathbf{t})$ is obtained from the product of the scattering function at inner vertices ρ, the geometry function G, and the visibility term V for each path segment as

$$\mathfrak{T}(\bar{\mathbf{x}}, \mathbf{t}) = \left[\prod_{i=1}^{k-1} \rho(\mathbf{x}_i, t_i)\right]\left[\prod_{i=0}^{k-1} G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1})\, V(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1})\right], \tag{3}$$

where the time dependency is explicitly incorporated in the scattering function as $\rho(\mathbf{x}_i, t_i)$.

3.2 Path-space Differentiable Rendering

Differentiable rendering deals with the computation of the derivatives of the image $I(\boldsymbol{\theta})$ with respect to a set of parameters $\boldsymbol{\theta}$ describing the scene. Unfortunately, the path space $\Omega(\boldsymbol{\theta})$ presents a large number of discontinuities due to occlusions and surface discontinuities, resulting into potential interactions between the different parameters; a naïve derivation of $\frac{\mathrm{d}I(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}$ cannot handle such *evolving* discontinuities, and therefore require sophisticated methods to handle them [Bangaru et al. 2020; Li et al. 2018; Loubet et al. 2019].

Zhang et al. [2020] observed that, given a fixed set of parameters $\boldsymbol{\theta}$, the vertices $\mathbf{x}_{0...k}$ of a path $\bar{\mathbf{x}} \in \Omega(\boldsymbol{\theta})$ lie on a surface manifold $\mathcal{M}(\boldsymbol{\theta}) \subset \mathbb{R}^2$ that *evolves* with $\boldsymbol{\theta}$. In particular, it follows a vector field defined by the local velocities $v(\mathbf{x}_{0...k}, \boldsymbol{\theta}) = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\mathbf{x}_{0...k}(\boldsymbol{\theta})$, with

$\mathbf{x}_{0\dots k}(\boldsymbol{\theta}) \in \mathcal{M}(\boldsymbol{\theta})$ a path vertex for a parameter set $\boldsymbol{\theta}$ (in the following, we omit the dependence on $\boldsymbol{\theta}$ on paths and vertices for clarity).

The manifold $\mathcal{M}(\boldsymbol{\theta})$ can be assumed C^0-continuous except in the set $\partial\overline{\mathcal{M}(\boldsymbol{\theta})}[f] \subset \mathcal{M}(\boldsymbol{\theta})$ that includes the boundaries of $\mathcal{M}(\boldsymbol{\theta})$ and the discontinuity curves of $f(\bar{\mathbf{x}})$, i.e., the jump discontinuity points in $f(\bar{\mathbf{x}})$, $\partial\mathcal{M}$, and $\Delta\mathcal{M}[f]$. For clarity, in the following we omit the dependence on f of the extended boundary. We denote $\partial\overline{\Omega(\boldsymbol{\theta})} \subset \Omega(\boldsymbol{\theta})$ the subspace of paths with at least one vertex $\mathbf{x}_{0..k} \in \partial\overline{\mathcal{M}(\boldsymbol{\theta})}$, i.e., lying on the boundary manifold. Tracking this particular subset of the manifold allows to explicitly account for the derivatives due to discontinuities in the path space.

By following the Reynolds transport theorem [Cermelli et al. 2005] from fluid mechanics, the derivatives of the path integral can be computed by defining the integral on the evolving manifold $\mathcal{M}(\boldsymbol{\theta})$ as the sum of the so-called *interior* and *boundary* integrals, as

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}I(\boldsymbol{\theta}) = \int_\Omega \left[\overset{\square}{f}(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}})\sum_{i=0}^k \kappa(\mathbf{x}_i)\mathcal{V}_\mathcal{M}(\mathbf{x}_i)\right]\mathrm{d}\mu(\bar{\mathbf{x}}) \qquad (4)$$

$$+ \int_{\partial\overline{\Omega}} \Delta f_i(\bar{\mathbf{x}})\mathcal{V}_{\partial\overline{\mathcal{M}}_i}(\mathbf{x}_i)\mathrm{d}\mu_{\partial\overline{\Omega}}(\bar{\mathbf{x}}),$$

where $\overset{\square}{f}(\bar{\mathbf{x}})$ is the normal derivative of $f(\bar{\mathbf{x}})$ as a function of $\boldsymbol{\theta}$, $\Delta f_i(\bar{\mathbf{x}})$ is the boundary contribution function for a vertex $\mathbf{x}_i \in \partial\overline{\mathcal{M}}$ on the boundary, $\kappa(\mathbf{x}_i)$ is the total curvature at $\mathbf{x}_i$, and $\mathcal{V}_\mathcal{M}(\mathbf{x}_i)$ (and $\mathcal{V}_{\partial\overline{\mathcal{M}}}(\mathbf{x}_i)$) are the scalar normal velocities defined as $\mathcal{V}_\mathcal{M}(\mathbf{x}_i) = \mathbf{n}(\mathbf{x}_i)\cdot v(\mathbf{x}_i)$ with $\mathbf{n}(\mathbf{x}_i)$ the unit normal at $\mathbf{x}_i$. The normal derivative of $f(\bar{\mathbf{x}})$ is defined as $\overset{\square}{f}(\bar{\mathbf{x}}) = \dot{f}(\bar{\mathbf{x}}) - (v(\bar{\mathbf{x}}) - \mathcal{V}_\mathcal{M}\mathbf{n}))\,\mathrm{grad}_\mathcal{M}(f(\bar{\mathbf{x}}))$, with $\mathrm{grad}_\mathcal{M}(f(\bar{\mathbf{x}}))$ the manifold gradient of $f(\bar{\mathbf{x}})$ (see Figure 2.a). Note that for zero tangential velocity of the evolving manifold with respect to the local parameterization we get $\overset{\square}{f}(\bar{\mathbf{x}}) = \dot{f}(\bar{\mathbf{x}})$. Figure 2.b depicts the derivatives of the path integral as interior and boundary integrals.

4 DIFFERENTIABLE TRANSIENT RENDERING

In this section we obtain the differentiable version of the transient path integral in Equation (1). Following the path-space model proposed by Zhang et al. [2020], we compute two different terms for the path integral, the interior and boundary terms.

In the original *steady-state* formulation of Zhang and colleagues, all scattering events in path $\bar{\mathbf{x}}$ are considered *independent*, which allows the authors to apply the two-dimensional Reynolds transport theorem recursively. However, in *transient state* this is no longer the case; the total time of flight of the path $\mathrm{tof}(\bar{\mathbf{x}})$ needs to be taken into account, and as a consequence scattering events are no longer independent. To deal with this and derive our differential transient path integral, we rely on the *generalized transport theorem* [Seguin and Fried 2014] applied on a higher-dimensional manifold.

Differential transient path integral. We first define an intermediate path contribution term $f_\mathcal{T}$ that only depends on spatial variables as

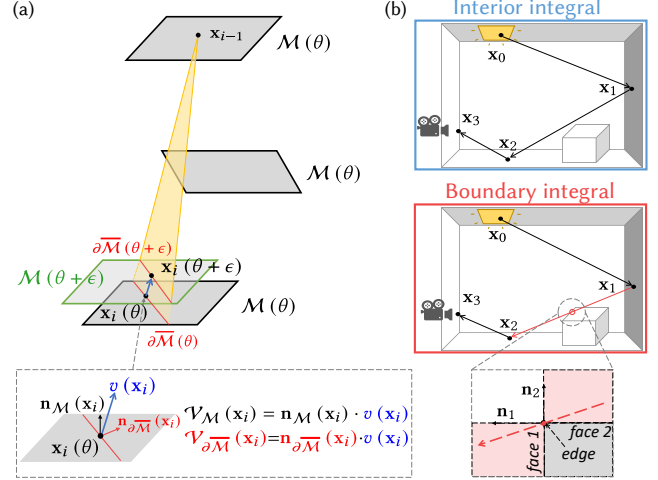$$f_\mathcal{T}(\bar{\mathbf{x}}) = \int_\mathcal{T} f_\mathrm{t}(\bar{\mathbf{x}}, \mathbf{t})\mathrm{d}\mu(\mathbf{t}), \qquad (5)$$

Fig. 2. (a) A surface manifold $\mathcal{M}(\theta)$ and a scalar field with a jump discontinuity evolve with respect to a scene parameter θ. The red line represents the discontinuity of the scalar field (e.g., the visibility function $V(\mathbf{x}_i, \mathbf{x}_{i-1})$ for a fixed $\mathbf{x}_{i-1}$). Such discontinuity, belonging to $\partial\overline{\mathcal{M}}$, also evolves with θ. The blue arrow indicates the local velocity of a point $\mathbf{x}_i \in \partial\overline{\mathcal{M}}$, denoted by $v(\mathbf{x}_i)$. (b) Illustration of the sampling process of the interior integral (top) and the boundary integral (bottom) in path-space differentiable rendering [Zhang et al. 2020]. In the latter, light paths should contain a boundary segment (in red), which grazes a sharp edge in the scene geometry.

which leads to the following expression for the transient path integral defined in Equation (1):

$$I = \int_\Omega f_\mathcal{T}(\bar{\mathbf{x}})\mathrm{d}\mu(\bar{\mathbf{x}}). \qquad (6)$$

Similar to previous works [Chen et al. 2020; Jarabo et al. 2014; Tsai et al. 2019], we assume that scattering delays in materials (due to e.g., multiple scattering or electromagnetic phase shift) are negligible compared with the temporal resolution of sensors and the propagation delays. This allows us to approximate $\rho(\mathbf{x}_i, t_i) \approx \rho(\mathbf{x}_i)\delta(t_i)$, therefore removing the temporal dependence on the light throughput $\mathfrak{T}(\bar{\mathbf{x}})$ defined in Equation (3). Moreover, the integral domain of Equation (5) no longer depends on scattering delays, and thus it reduces to a one-dimensional time domain. Therefore, when the temporal response of the source L_e and sensor W_e are independent, the path contribution $f_\mathcal{T}$ becomes

$$f_\mathcal{T}(\bar{\mathbf{x}}) = \mathfrak{T}(\bar{\mathbf{x}})\int_{-\infty}^\infty L_e(\mathbf{x}_0 \to \mathbf{x}_1, t)W_e(\mathbf{x}_{k-1} \to \mathbf{x}_k, t + \mathrm{tof}(\bar{\mathbf{x}}))\mathrm{d}t$$
$$= \mathfrak{T}(\bar{\mathbf{x}})\,S_e(\bar{\mathbf{x}}), \qquad (7)$$

where we introduce a novel *correlated importance* function $S_e(\bar{\mathbf{x}})$. Note that $S_e(\bar{\mathbf{x}})$ has an implicit dependence on time through $\mathrm{tof}(\bar{\mathbf{x}})$; this dependence will be relevant when we obtain its derivatives later in this section, since it allows us to handle discontinuous light and sensor functions (see Figure 3).

To compute the derivative of image I in Equation (6) we need to differentiate a high-dimensional integral, since our evolving manifolds are $2(k + 1)$-dimensional in path space Ω; more generally, Ω is an m-dimensional subset of $\mathbb{R}^n$, with m and n integers such that

$n > m$. However, due to the influence of $\mathrm{tof}(\bar{\mathbf{x}})$ in path contribution $f_{\mathcal{T}}(\bar{\mathbf{x}})$, we cannot separate it into two-dimensional integrals on $\mathcal{M}$ as in previous work [Zhang et al. 2020], and as a consequence we cannot recursively apply Reynolds transport theorem for two-dimensional manifolds.

Instead, we turn to the *generalized transport theorem* [Seguin and Fried 2014] to compute the derivative of I. As opposed to Reynolds transport theorem, the generalized transport theorem considers a higher-dimensional manifold Ω defined as the product of multiple two-dimensional manifolds. Intuitively, each of these low-dimensional manifolds define a light interaction at each path vertex. Applying the generalized transport theorem allows to directly compute the derivative of the full path, leading to our *differential transient path integral* (see the supplemental document for the complete rigorous derivation):

$$
\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} \int_{\Omega} f_{\mathcal{T}}(\bar{\mathbf{x}}) \mathrm{d}\mu(\bar{\mathbf{x}}) = \int_{\Omega} \left[\overset{\square}{f}_{\mathcal{T}}(\bar{\mathbf{x}}) - f_{\mathcal{T}}(\bar{\mathbf{x}}) \sum_{i=0}^{k} \kappa(\mathbf{x}_i) \mathcal{V}_{\mathcal{M}}(\mathbf{x}_i) \right] \mathrm{d}\mu(\bar{\mathbf{x}})
$$
$$
+ \int_{\partial\overline{\Omega}} \Delta f_{\mathcal{T}}(\bar{\mathbf{x}}) \mathcal{V}_{\partial\overline{\mathcal{M}}}(\bar{\mathbf{x}}) \, \mathrm{d}\mu_{\partial\overline{\Omega}}(\bar{\mathbf{x}}), \tag{8}
$$

where the first term represents the interior integral, and the second describes the boundary integral. Note that the boundary integral is now defined over the extended boundary $\partial\overline{\Omega}[f_{\mathcal{T}}]$; for clarity on the notation, we omit this dependence. Equation (8) has a similar structure to Zhang et al.'s steady-state differential path integral (4), but the term $f_{\mathcal{T}}(\bar{\mathbf{x}})$ now takes explicitly into account the temporal delays due to light's time of flight. Our transient formulation can thus be seen as a generalization of Zhang's formulation, but not limited to 2D manifolds. In the following, we derive individually the interior and boundary integrals.

Interior term. From Equations (7) and (8), the normal derivative of path contribution $f_{\mathcal{T}}(\bar{\mathbf{x}})$ with respect to $\boldsymbol{\theta}$ is

$$
\overset{\square}{f}_{\mathcal{T}}(\bar{\mathbf{x}}) = \overset{\square}{\mathfrak{T}}(\bar{\mathbf{x}}) S_e(\bar{\mathbf{x}}) + \mathfrak{T}(\bar{\mathbf{x}}) \overset{\square}{S}_e(\bar{\mathbf{x}}). \tag{9}
$$

The normal derivative of the light throughput $\overset{\square}{\mathfrak{T}}(\bar{\mathbf{x}})$ is obtained by the product rule in a similar fashion as Zhang et al.:

$$
\overset{\square}{\mathfrak{T}}(\bar{\mathbf{x}}) = \sum_{j=1}^{k-1} \left[\overset{\square}{\rho}_j \prod_{\substack{i=1 \\ i \neq j}}^{k-1} \rho_i + \prod_{i=0}^{k-1} G_i V_i \right] + \sum_{j=0}^{k-1} \left[\prod_{i=1}^{k-1} \rho_i + \left(\overset{\square}{G}_j V_j + G_j \overset{\square}{V}_j \right) \prod_{\substack{i=0 \\ i \neq j}}^{k-1} G_i V_i \right], \tag{10}
$$

where $\rho_i = \rho(\mathbf{x}_{i-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{x}_{i+1})$, $G_i = G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1})$, and $V_i = V(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1})$, for any index i.

We then compute the normal derivative of our novel term $S_e(\bar{\mathbf{x}})$ as

$$
\overset{\square}{S}_e(\bar{\mathbf{x}}) = \frac{\partial S_e(\bar{\mathbf{x}})}{\partial t} \overset{\square}{\mathrm{tof}}(\bar{\mathbf{x}}) + \sum_{i=0,1,k-1,k} \frac{\partial S_e(\bar{\mathbf{x}})}{\partial \mathbf{x}_i} \overset{\square}{\mathbf{x}}_i. \tag{11}
$$

Considering S_e a function of t, $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_{k-1}$, and $\mathbf{x}_k$, the partial derivatives are evaluated as follows:

$$
\frac{\partial S_e(\bar{\mathbf{x}})}{\partial t} = \int_{-\infty}^{\infty} \left[\frac{\partial L_e}{\partial t}(\mathbf{x}_0, \mathbf{x}_1, t') W_e(\mathbf{x}_{k-1}, \mathbf{x}_k, t' + t) \right.
$$
$$
\left. + L_e(\mathbf{x}_0, \mathbf{x}_1, t') \frac{\partial W_e}{\partial t}(\mathbf{x}_{k-1}, \mathbf{x}_k, t' + t) \right] \mathrm{d}t' \bigg|_{t=\mathrm{tof}(\bar{\mathbf{x}})},
$$
$$
\frac{\partial S_e(\bar{\mathbf{x}})}{\partial \mathbf{x}_i} = \int_{-\infty}^{\infty} \frac{\partial L_e}{\partial \mathbf{x}_i}(\mathbf{x}_0, \mathbf{x}_1, t') W_e(\mathbf{x}_{k-1}, \mathbf{x}_k, t' + t) \mathrm{d}t' \bigg|_{t=\mathrm{tof}(\bar{\mathbf{x}})} \quad \text{for } i=0,1,
$$
$$
\frac{\partial S_e(\bar{\mathbf{x}})}{\partial \mathbf{x}_i} = \int_{-\infty}^{\infty} L_e(\mathbf{x}_0, \mathbf{x}_1, t') \frac{\partial W_e}{\partial \mathbf{x}_i}(\mathbf{x}_{k-1}, \mathbf{x}_k, t' + t) \mathrm{d}t' \bigg|_{t=\mathrm{tof}(\bar{\mathbf{x}})} \quad \text{for } i=k-1,k.
$$
$$
\tag{12}
$$

The normal derivative of the path time of flight is

$$
\overset{\square}{\mathrm{tof}}(\bar{\mathbf{x}}) = \sum_{i=1}^{k} \frac{\eta_i}{c} \|\mathbf{x}_i - \mathbf{x}_{i-1}\|^{\square} + \frac{\overset{\square}{\eta}_i}{c} \|\mathbf{x}_i - \mathbf{x}_{i-1}\| \tag{13}
$$
$$
= \sum_{i=1}^{k} \frac{\eta_i}{c} \frac{(\mathbf{x}_i - \mathbf{x}_{i-1}) \cdot \overset{\square}{\mathbf{x}}_i + (\mathbf{x}_{i-1} - \mathbf{x}_i) \cdot \overset{\square}{\mathbf{x}}_{i-1}}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|} + \frac{\overset{\square}{\eta}_i}{c} \|\mathbf{x}_i - \mathbf{x}_{i-1}\|.
$$

In both Equations (11) and (13) the derivatives of the vertices $\overset{\square}{\mathbf{x}}_i$ lie on the velocity field of $\mathcal{M}$ as $\overset{\square}{\mathbf{x}}_i = \mathcal{V}_{\mathcal{M}}(\mathbf{x}_i)$.

To obtain the final expression for $\overset{\square}{f}_{\mathcal{T}}(\bar{\mathbf{x}})$, we plug Equation (11) into Equation (9), which yields:

$$
\overset{\square}{f}_{\mathcal{T}}(\bar{\mathbf{x}}) = \overset{\square}{f}_s(\bar{\mathbf{x}}) + \mathfrak{T}(\bar{\mathbf{x}}) \frac{\partial S_e(\bar{\mathbf{x}})}{\partial t} \overset{\square}{\mathrm{tof}}(\bar{\mathbf{x}}), \tag{14}
$$

where $\overset{\square}{f}_s(\bar{\mathbf{x}})$ is defined as

$$
\overset{\square}{f}_s(\bar{\mathbf{x}}) = \overset{\square}{\mathfrak{T}}(\bar{\mathbf{x}}) S_e(\bar{\mathbf{x}}) + \mathfrak{T}(\bar{\mathbf{x}}) \sum_{i=0,1,k-1,k} \frac{\partial S_e(\bar{\mathbf{x}})}{\partial \mathbf{x}_i} \overset{\square}{\mathbf{x}}_i. \tag{15}
$$

The second term in Equation (14) models the temporal derivatives for transient rendering (which do not exist in steady-state differentiable rendering), while $\overset{\square}{f}_s(\bar{\mathbf{x}})$ models the spatial derivatives of the path contribution, similar to the steady-state differential path integral in Equation (4).

Boundary term. The main problem in the second term of Equation (8) (the boundary term) is to determine the domain of integration $\partial\overline{\Omega}[f_{\mathcal{T}}]$, in which we will compute the path contribution $\Delta f_{\mathcal{T}}(\bar{\mathbf{x}})$, as well as how this domain evolves with $\boldsymbol{\theta}$ according to the velocity field $\mathcal{V}_{\partial\overline{\mathcal{M}}}(\bar{\mathbf{x}})$. Recall that, similarly to its steady-state counterpart in Equation (4), $\partial\overline{\Omega}[f_{\mathcal{T}}]$ represents the extended boundary of $f_{\mathcal{T}}$, where we include both the manifold boundaries $\partial\Omega$ and the discontinuity set $\Delta\Omega[f_{\mathcal{T}}]$ of $f_{\mathcal{T}}$ as $\partial\overline{\Omega}[f_{\mathcal{T}}] = \partial\Omega \cup \Delta\Omega[f_{\mathcal{T}}]$. For our derivation, we consider the set of discontinuities $\Delta\Omega_k[f_{\mathcal{T}}]$ for paths of length k; it follows that $\Delta\Omega[f_{\mathcal{T}}] = \bigcup_{k=1}^{\infty} \Delta\Omega_k[f_{\mathcal{T}}]$. We summarize here the main results of this derivation, and refer the reader to the supplemental material for the full details.

By the generalized transport theorem, the extended boundary of $f_{\mathcal{T}}$ is the union of the boundary of manifold Ω_k, and the discontinuities in the different terms of Equation (7), namely the visibility

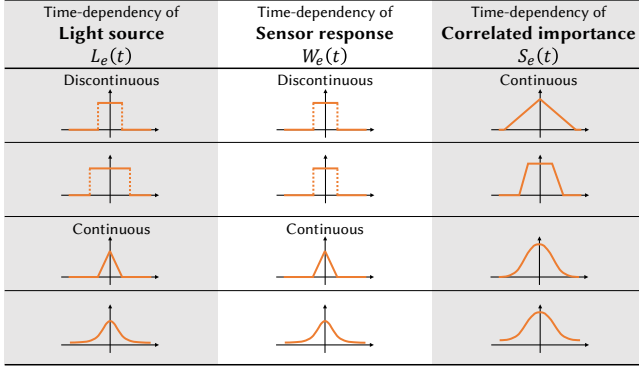| Time-dependency of **Light source** $L_e(t)$	Time-dependency of **Sensor response** $W_e(t)$	Time-dependency of **Correlated importance** $S_e(t)$
Discontinuous	Discontinuous	Continuous
Continuous	Continuous	

Fig. 3. Example shapes of time-dependent light source L_e, sensor response W_e, and the resulting correlated importance S_e functions. These shapes include combinations of short square wave laser pulses and time-gated sensors, as well as Gaussian laser pulses and time-gated sensors, used in real applications.

V, geometry G, scattering ρ, and correlated importance S_e terms, as

$$\partial\overline{\Omega}_k[f_{\mathcal{T}}] = \partial\Omega_k \cup \Delta\Omega_k[f_{\mathcal{T}}]$$
$$= \partial\Omega_k \cup \Delta\Omega_k[S_e] \cup \Delta\Omega_k[G_1\cdots G_k]$$
$$\cup \Delta\Omega_k[V_1\cdots V_k] \cup \Delta\Omega_k[\rho_1\cdots\rho_{k-1}]. \quad (16)$$

By assuming realistic non-singular scattering functions and light sources following previous work [Bangaru et al. 2020; Li et al. 2018; Loubet et al. 2019; Zhang et al. 2020, 2019], and applying the product space rule, we can express Equation (16) as

$$\partial\overline{\Omega}_k[f_{\mathcal{T}}] = \partial\Omega_k \cup \Delta\Omega_k[S_e]$$
$$\cup (\Delta\Omega_k[G_1V_1\cdots G_kV_k] - \partial\Omega_k), \quad (17)$$

with

$$\Delta\Omega_k[G_1V_1\cdots G_kV_k] - \partial\Omega_k = \bigcup_{i=1}^{k}\mathcal{M}_0\times\cdots\times\mathcal{M}_{i-2} \quad (18)$$
$$\times\left(\Delta\mathcal{M}^2[GV] - \partial\mathcal{M}^2\right)\times\mathcal{M}_{i+1}\times\cdots\times\mathcal{M}_k,$$

where $\Delta\mathcal{M}^2[GV] - \partial\mathcal{M}^2$ represents the path segments that intersect a silhouette edge of $\mathcal{M}$ (see Figure 2 and Lemma 2.4 in the supplemental). Note that under these conditions $\Delta\Omega_k[G_1V_1\cdots G_kV_k]$ represents a similar boundary as in steady state [Zhang et al. 2020], but derived from the generalized transport theorem without restrictions in the dimensionality of the manifolds.

The time-dependent effects on the boundary space $\partial\overline{\Omega}_k[f_{\mathcal{T}}]$ are implicitly encoded in $\Delta\Omega_k[S_e]$. For theoretical Dirac-delta emissions and sensor responses this becomes relevant, while for most practical applications (from ultrashort lasers to amplitude-modulated time-of-flight sensors) this can be handled without additional sampling. Figure 3 illustrates these practical scenarios.

4.1 Material Form of Differential Transient Path Integral

While our differential transient path integral in Equation (8) is a general formulation, its material form is more efficient for algorithmic implementation. Such material form allows integrating over a linear space parametrizing the spatial curved manifold, which in turn simplifies significantly how boundaries are handled [Zhang et al. 2020].

Thus, instead of integrating over the curved manifold $\mathcal{M}(\theta)$, which lies on $\mathbb{R}^3$, we can integrate over a planar domain $\mathcal{B}(\theta)\subset\mathbb{R}^2$, called a reference configuration, when there exists a global parameterization $\hat{\mathbf{x}}(\cdot,\theta):\mathcal{B}(\theta)\to\mathcal{M}(\theta)$. Following Zhang et al. [2020] we use a barycentric parametrization mapping $\bar{\mathbf{x}} = \hat{\mathbf{x}}(\bar{\mathbf{p}},\theta)$ from $\bar{\mathbf{p}}$ to $\bar{\mathbf{x}}$; this results in $\left(\hat{f}_{\mathcal{T}}\right)^{\square}(\bar{\mathbf{p}}) = \left(\hat{f}_{\mathcal{T}}\right)^{\cdot}(\bar{\mathbf{p}})$, as well as zero local velocity on boundaries and sharp edges. This global parameterization converts our differential transient path integral in Equation (8) into a simpler form:

$$\frac{dI}{d\theta} = \int_{\hat{\Omega}}\left(\hat{f}_{\mathcal{T}}\right)^{\cdot}(\bar{\mathbf{p}})\,d\mu(\bar{\mathbf{p}}) + \int_{\partial\hat{\Omega}}\Delta\hat{f}_{\mathcal{T}}(\bar{\mathbf{p}})\mathcal{V}_{\Delta\mathcal{B}}(\bar{\mathbf{p}})\,d\mu_{\partial\hat{\Omega}}(\bar{\mathbf{p}}), \quad (19)$$

where $\hat{\Omega} = \bigcup_{k=1}^{\infty}\mathcal{B}^{k+1}$ is the space of all material paths $\bar{\mathbf{p}} = \mathbf{p}_0\ldots\mathbf{p}_k$. The terms in Equation (19) can be easily defined by considering the equivalent terms in Equation (8) as functions of material paths $\bar{\mathbf{p}}$ instead of $\bar{\mathbf{x}}$, and taking the Jacobian of $\hat{\mathbf{x}}$ into account, similar to the case of steady-state differentiable rendering [Zhang et al. 2020]. In particular, the path contribution in material form is defined as

$$\hat{f}_{\mathcal{T}}(\bar{\mathbf{p}}) = \hat{\mathfrak{T}}(\bar{\mathbf{p}})\hat{S}_e(\mathbf{p}), \quad (20)$$

where $\hat{\mathfrak{T}}(\bar{\mathbf{p}}) = \mathfrak{T}(\hat{\mathbf{x}}(\bar{\mathbf{p}},\theta))\prod_{i=1}^{k-1}J(\mathbf{p}_i,\theta)$, with $J(\mathbf{p},\theta) = \left|\frac{\partial\hat{\mathbf{x}}(\mathbf{p},\theta)}{\partial\mathbf{p}}\right|$ denoting the Jacobian determinant of the change of variable $\hat{\mathbf{x}}$. The derivatives of the Jacobian determinant $\frac{\partial J(\mathbf{p}_i,\theta)}{\partial\theta}$ are computed in the same fashion as in the steady-state case [Zhang et al. 2020].

We compute the material form of the correlated importance as

$$\hat{S}_e(\bar{\mathbf{p}}) = S_e(\hat{\mathbf{x}}(\bar{\mathbf{p}},\theta))J(\mathbf{p}_0,\theta)J(\mathbf{p}_k,\theta), \quad (21)$$

which only depends on the Jacobian of the first and last vertices of the path $\mathbf{p}_0$ and $\mathbf{p}_k$. Finally, the derivative of the correlated importance $\hat{S}_e(\bar{\mathbf{p}})$ is

$$(\hat{S}_e)^{\cdot}(\bar{\mathbf{p}}) = \frac{\partial S_e(\hat{\mathbf{x}}(\bar{\mathbf{p}},\theta))}{\partial\theta}J(\mathbf{p}_0,\theta)J(\mathbf{p}_k,\theta) \quad (22)$$
$$+ S_e(\hat{\mathbf{x}}(\bar{\mathbf{p}},\theta))\left[\frac{\partial J(\mathbf{p}_0,\theta)}{\partial\theta}J(\mathbf{p}_k,\theta) + J(\mathbf{p}_0,\theta)\frac{\partial J(\mathbf{p}_k,\theta)}{\partial\theta}\right],$$

with $\frac{\partial S_e(\hat{\mathbf{x}}(\bar{\mathbf{p}},\theta))}{\partial\theta}$ computed in an analogous way as Equation (11):

$$\frac{\partial S_e(\hat{\mathbf{x}}(\bar{\mathbf{p}},\theta))}{\partial\theta} = \frac{\partial S_e(\hat{\mathbf{x}}(\mathbf{p}_i,\theta))}{\partial t}\frac{\partial\mathrm{tof}(\hat{\mathbf{x}}(\bar{\mathbf{p}},\theta))}{\partial\theta}$$
$$+ \sum_{i=0,1,k-1,k}\frac{\partial S_e(\bar{\mathbf{x}})}{\partial\mathbf{x}_i}\frac{\partial\hat{\mathbf{x}}(\mathbf{p}_i,\theta)}{\partial\theta}. \quad (23)$$

5 DIFFERENTIABLE MONTE CARLO TRANSIENT RENDERING

We approximate the material form of our differential transient path integral (19) as the sum of two Monte Carlo estimators for the interior and boundary integrals, as

$$\frac{dI}{d\theta} \approx \frac{1}{N_{\mathrm{i}}}\sum_{j=1}^{N_{\mathrm{i}}}\frac{\left(\hat{f}_{\mathcal{T}}\right)^{\cdot}(\bar{\mathbf{p}}_j)}{p_{\mathrm{i}}(\bar{\mathbf{p}}_j)} + \frac{1}{N_{\mathrm{b}}}\sum_{l=1}^{N_{\mathrm{b}}}\frac{\Delta\hat{f}_{\mathcal{T}}(\bar{\mathbf{p}}_l)\mathcal{V}_{\partial\overline{\mathcal{B}}}(\bar{\mathbf{p}}_l)}{p_{\mathrm{b}}(\bar{\mathbf{p}}_l)}, \quad (24)$$

where N_{i} and N_{b} are the number of samples for the interior and boundary integrals, respectively, and $p_{\mathrm{i}}(\bar{\mathbf{p}})$ and $p_{\mathrm{b}}(\bar{\mathbf{p}})$ are the probabilities of sampling the material path $\bar{\mathbf{p}}$ for each estimator.
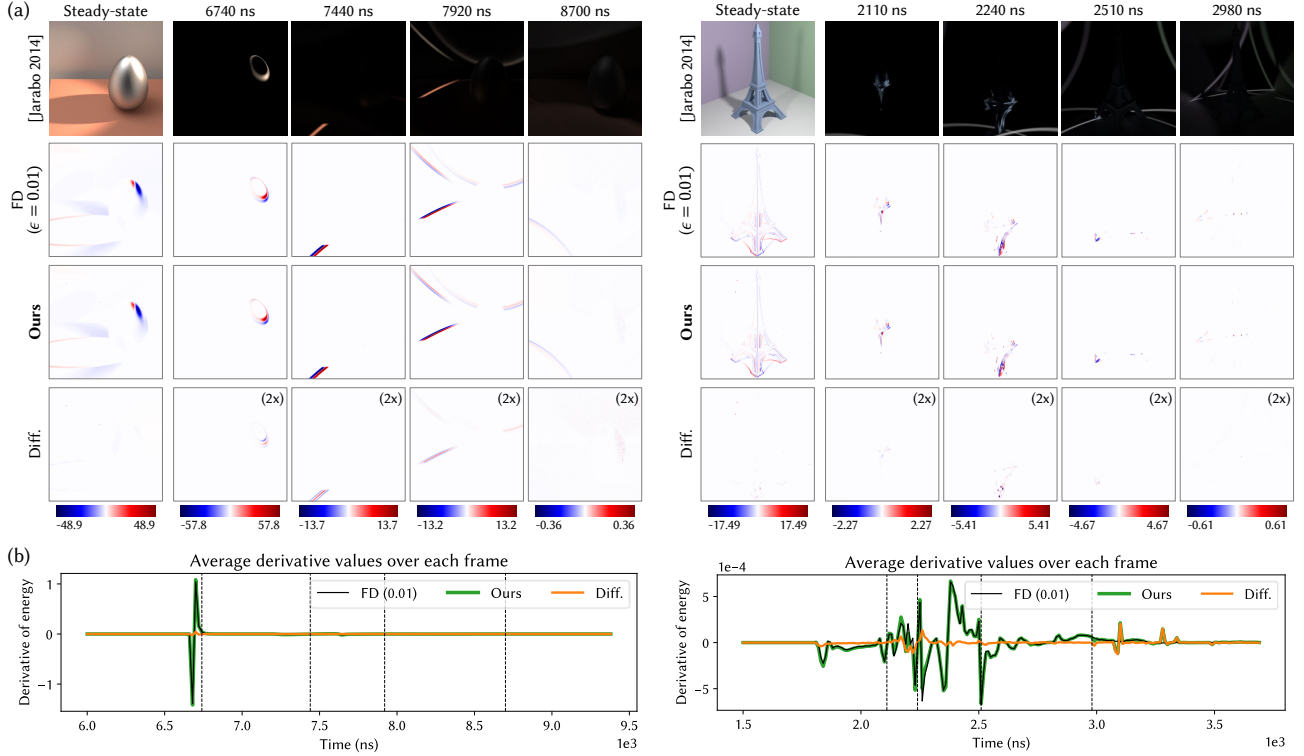
Fig. 4. Validation of our derivatives against finite differences with $\epsilon = 0.01$. Left: *Egg* scene with a moving light source. Right: *Tower* scene with a rotating object. (a) Transient sequences and derivatives, including the absolute difference amplified by a factor of 2. (b) Gradient plots. Dotted vertical lines indicate the timestamp of the frames shown above. In both scenes our framework matches the finite differences (FD) method closely. The *Egg* scene consists of 1,000 transient frames with 20.0 ns exposure time. Our result is rendered with 410+919 spp (interior + boundary integrals, respectively), which takes 28.9s per frame. The FD result is rendered with 4,096 spp at 267.8s per frame. The *Tower* scene consists of 500 transient frames taken with 10.0 ns exposure time. Our result is rendered with 819+1,720 spp at 67.7s per frame. The FD result is rendered with 4,096 spp at 144.0s per frame.

Similar to previous work on transient rendering [Jarabo et al. 2014], we reconstruct the whole temporal domain at the same time by reusing samples between different time frames. We track the time of flight of the path $\hat{\mathbf{x}}(\bar{\mathbf{p}}_j, \boldsymbol{\theta})$, and bin the path's contribution along the temporal domain. The temporal footprint of the path $\hat{\mathbf{x}}(\bar{\mathbf{p}}_j, \boldsymbol{\theta})$ is selected based on a time window determined by the non-zero regions of $S_e\left(\hat{\mathbf{x}}(\bar{\mathbf{p}}_j, \boldsymbol{\theta})\right)$, which acts as a temporal smoothing kernel potentially removing temporal discontinuities between frames (see Figure 3).

Interior integral estimator. We estimate the interior integral by tracing the path $\hat{\mathbf{x}}(\bar{\mathbf{p}}_j, \boldsymbol{\theta})$ and computing $\left(\hat{f}_{\mathcal{T}}\right)^{\cdot}(\bar{\mathbf{p}}_j)$. Different from Zhang et al. [2020], we track the temporal delay of the path encoded in $\text{tof}(\hat{\mathbf{x}}(\bar{\mathbf{p}}_j, \boldsymbol{\theta}))$. Note that during this process we compute the throughput $\hat{\mathfrak{T}}(\bar{\mathbf{p}}_j)$ and its derivative $\left(\hat{\mathfrak{T}}\right)^{\cdot}(\bar{\mathbf{p}}_j)$, which are later convolved by $\left(\hat{S}_e\right)^{\cdot}(\bar{\mathbf{p}}_j)$ and $\hat{S}_e(\bar{\mathbf{p}}_j)$ when binning the contribution on the temporal domain following Equation (9).

Boundary integral estimator. The main difference with respect to the steady-state boundary integral is the existence of the discontinuity term $\Delta\Omega_k[S_e]$. For most practical situations, both light sources and sensors have a finite temporal response making $S_e\left(\hat{\mathbf{x}}(\bar{\mathbf{p}}_j, \boldsymbol{\theta})\right)$ continuous (Figure 3), so $\Delta\Omega_k[S_e]$ vanishes; the transient boundary integral estimation is then similar to its steady-state counterpart, except that we again need to keep track of the path time of flight $\text{tof}(\hat{\mathbf{x}}(\bar{\mathbf{p}}_l, \boldsymbol{\theta}))$ for binning.

6 RESULTS AND APPLICATIONS

We have implemented our differentiable transient rendering on top of the code provided by Zhang et al. [2020], by extending the bidirectional path tracer to keep track of the propagation delay of the paths. We compute the interior integral using automatic differentiation on top of next-event estimation and radiance-based multiple importance sampling. For the boundary integral, we use the multidirectional sampling of boundaries, next-event estimation, and the grid-based importance sampling proposed by Zhang et al. For temporal reconstruction we use histogram binning, though more advanced density estimation techniques [Jarabo et al. 2014] could be used without changes in our derivatives estimation. We refer to the supplemental material for pseudocode describing the computation of both the interior and boundary integrals.

All our results have been obtained using a conventional desktop computer equipped with an Intel Core I9-10920X CPU of 3.5 GHz with 128 GB RAM, and an NVIDIA Titan RTX graphics card. Without loss of generality, the examples shown are based on a triangular correlated importance function S_e (using discontinuous L_e and W_e
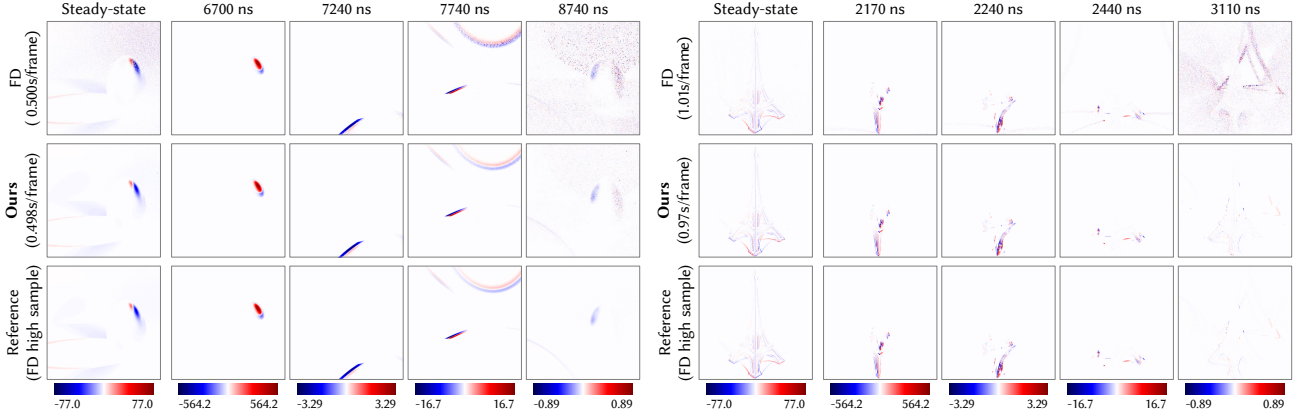
Fig. 5. Equal-time comparisons for FD and our derivatives on the *Egg* (left) and *Tower* (right) scenes. As the first and second rows show, our derivatives are visibly less noisy. The third row shows a high-sample reference (4,096 spp) rendered with FD, similar to the second row of Figure 4.
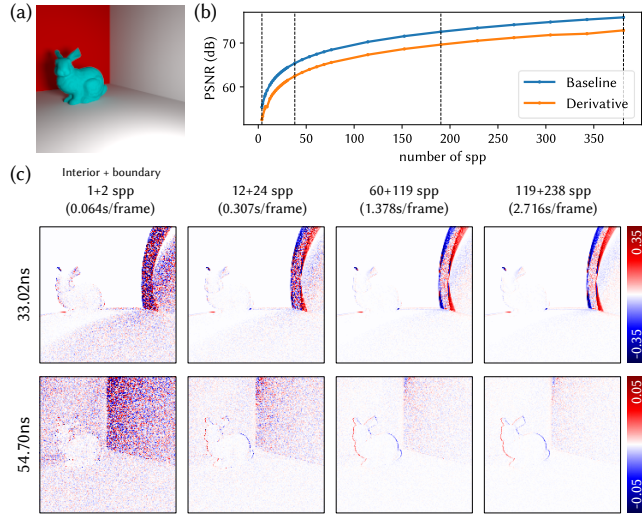

Fig. 6. Evolution of variance with the number of samples per pixel. (a) Steady-state scene; (b) PSNR of our baseline [Jarabo et al. 2014] and derivative transient images; (c) selected derivatives of transient frames, indicated by the dashed lines in (b) for each spp (interior plus boundary integrals). The scene parameter being optimized is the x-translation of the camera.

functions). Please refer also to our supplementary video for transient rendering results.

We first compare our derivatives against the finite differences (FD) method on two different scenes. To obtain the FD, we use our transient renderer by computing radiance with the same (radiance-based) sampling routines we use for the interior integral. Figure 4, left, shows the *Egg* scene, with a vertical translation of the light source as the varying parameter. Figure 4, right, shows the *Tower* scene, where the tower rotates along its vertical axis. In both scenes our method matches the results from finite differences closely. In addition, Figure 5 provides equal-time comparisons, showing how our method provides less noisy results than FD.

We also evaluate in Figure 6 the evolution of variance with the number of samples per pixel. We observe that it follows a trend similar to baseline transient rendering [Jarabo et al. 2014].

We have also validated our scene derivatives when varying several parameters at the same time. Figure 7 shows the transient sequence for the *Teapot* and its scene derivatives with respect to three parameters. θ_1 refers to a translation of the teapot, θ_2 to a translation of the light source, and θ_3 to the combined motion of the object and the light source. As the absolute difference row shows (amplified by a factor of 1000) $\frac{\partial I}{\partial \theta_1} + \frac{\partial I}{\partial \theta_2} = \frac{\partial I}{\partial \theta_3}$ when θ_3 is the combined motion of θ_1 and θ_2, which is a general key property of differentiable rendering.

6.1 Applications

We begin by presenting the first demonstrations of two challenging NLOS scenarios where the time of flight of light plays a key role. We use ADAM optimizer with a scene-dependent learning rate (see Table 2). First, our *Wavy* scene in Figure 8 shows a variation of the classic confocal NLOS imaging setup; we lift the requirement of using a planar surface as relay wall, and use a wavy surface instead. In the hidden scene, the cyan bunny moves to a different target position. We jointly optimize the three coordinates x, y, and z defining its position. Our sampling resolution is $60 \times 60 \times 1200$. We include a comparison of our derivative computation and FD equipped with the same optimizer and learning rate. Our results are rendered with 17 and 34 spp for interior and boundary integrals, respectively, while FD results are rendered with 4×17 spp. FD takes 61 min. per iteration, compared to 36 min. in our approach. Moreover, it can be seen how FD fails to optimize the parameters successfully. The oscillating nature of converge in our method is similar to a general stochastic gradient descent.

Next, we show a particularly difficult NLOS scenario: being able to track an object around *two* corners. Our *Corners* scene can be seen in Figure 9, where the hidden object moves along one axis. Despite the extremely challenging conditions of the scene, we are able to provide reasonable optimized results.

Figure 10 shows two different cases of parameter optimization. The *Coffee* scene (also shown in Figure 1) includes two transparent, colored objects: a glass teapot with reddish tea and an empty blue cup. Using our scene derivatives, we optimize the index of refraction of the tea in the teapot from an arbitrary initial value to the final
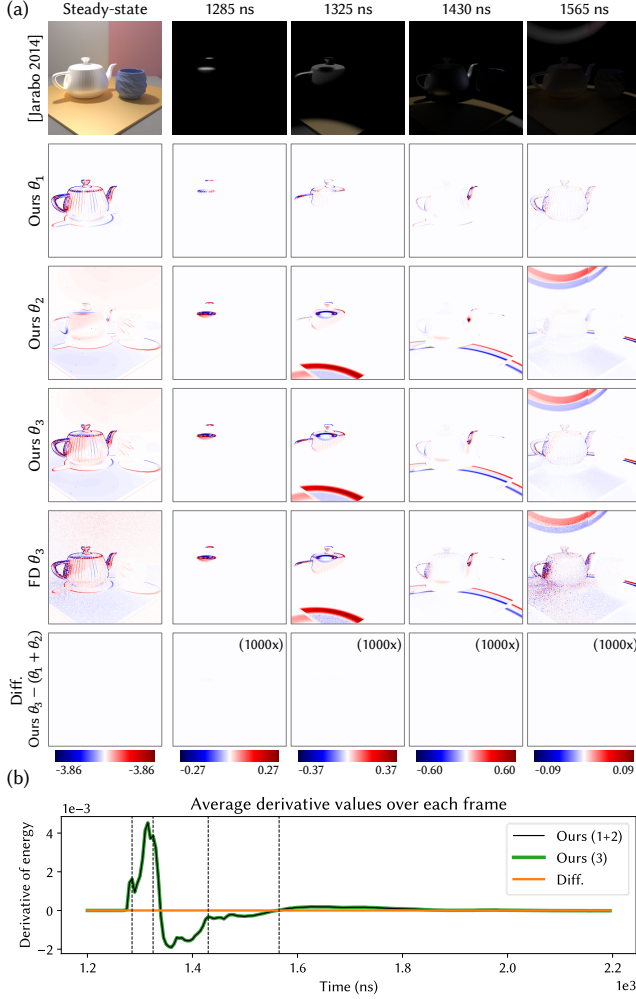
(a) Steady-state   1285 ns   1325 ns   1430 ns   1565 ns

[Jarabo 2014]

Ours θ_1

Ours θ_2

Ours θ_3

FD θ_3

Diff.
Ours $\theta_3 - (\theta_1 + \theta_2)$

(1000x)   (1000x)   (1000x)   (1000x)

-3.86 3.86   -0.27 0.27   -0.37 0.37   -0.60 0.60   -0.09 0.09

(b) Average derivative values over each frame

Fig. 7. Validation of our derivatives when varying several parameters at the same time. a) Transient sequence and derivatives. θ_1 corresponds to the translation of the teapot, θ_2 to the translation of the light source, and θ_3 to the combined translation of both teapot and light source. The absolute difference (x1000) shows how the sum of scene derivatives w.r.t. θ_1 and θ_2 is identical to the scene derivatives w.r.t. θ_3. (b) Radiance and absolute difference plots. Dotted vertical lines indicate the exact frames shown above. The scene consists of 600 transient frames with an exposure of 5.0 ns. Our results are rendered with 683+1,382 spp (interior + boundary integrals, resp.) taking 94.7s per frame. The FD result is rendered with 4×683 spp taking 173.4s per frame.

target value. Last, the *Stars* scene is composed of two translucent, colored stars. The red star rotates around one axis, while the index of refraction of the blue star changes. In both scenes, the streak images show how changing the index of refraction affects the time of flight of light through the transparent objects.

Figure 10 also includes and equal-sample comparison with Zhang et al.'s steady-state renderer [2020]. It can be seen how the index of refraction converges faster with our method in the *Coffee* scene, while for *Stars* the steady-state approach fails to converge. Table 2 summarizes the main information for all scenes shown. Please refer to the supplemental video for the full transient sequences.
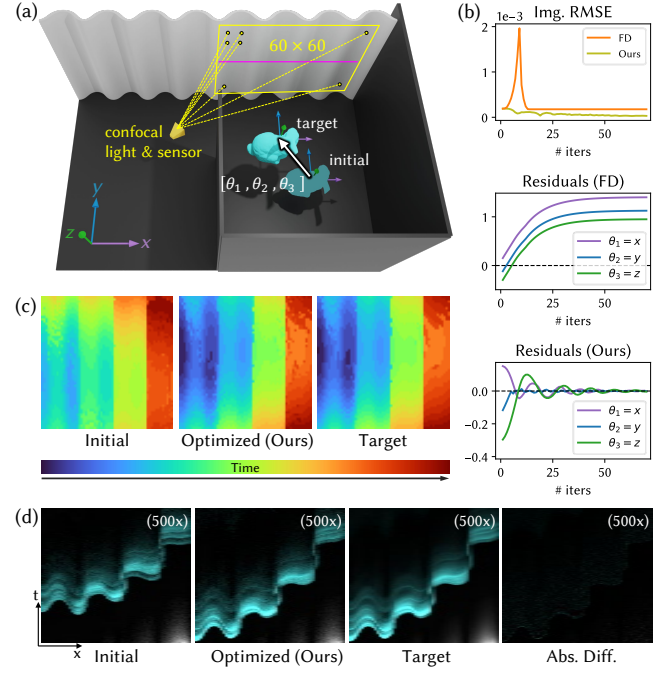

Fig. 8. Three-parameter optimization in an NLOS tracking scenario with a *wavy* relay wall. (a) The hidden bunny moves to the target position, which implies changes in three parameters: $\theta_1 = x$, $\theta_2 = y$, and $\theta_3 = z$. The yellows dots illustrate sampled positions. (b) RMSE of rendered transient images (top), as well as residuals for each parameter obtained by FD (middle) and our method (bottom). (c) Color-coded transient results: initial, optimized parameters after 69 iterations using our method, and target. (d) Streak images in $x - t$ for the coordinates indicated by the magenta line in (a).
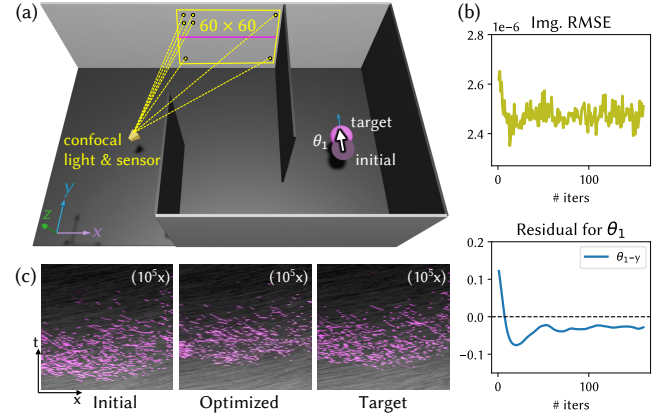

Fig. 9. NLOS tracking around two corners. (a) Scene setup, where the hidden object moves along one axis. (b) RMSE of the image and residual of the optimized parameter. (c) Streak images in $x - t$ for the coordinates indicated by the magenta line in (a) for the initial situation, optimized parameter after 159 iterations, and target. Even in this challenging scenario, our method converges with just a small residual bias.
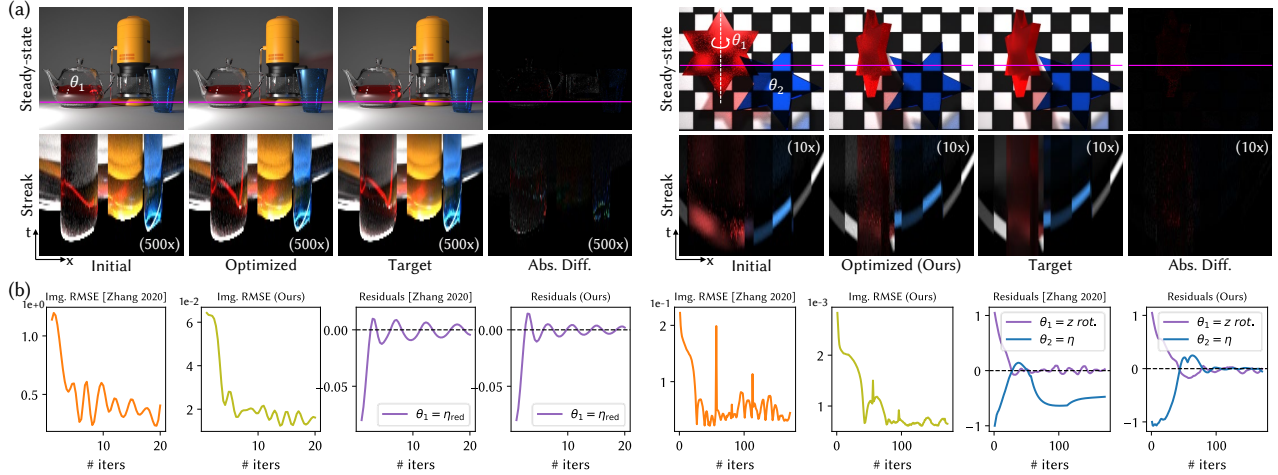
Fig. 10. Two examples of parameter optimization involving indices of refraction. Left: The same *Coffee* scene as in Figure 1, depicting a glass teapot with red tea. We optimize the index of refraction of the tea θ_1 from an arbitrary value (1.33) to the target value (1.41). Right: *Stars* scene showing two translucent glass stars. θ_1 represents the single-axis rotation of the red star while θ_2 represents the index of refraction of the blue star, changing from 1.5 to 2.5. For both scenes: (a) The top row shows steady-state images for the initial, optimized, and target parameters, as well as the difference between the last two. The bottom row shows the respective $x - t$ streak images, corresponding to the magenta line. It can be observed how changing the index of refraction affects not only the BSDF of the object, but also the time of flight of light. (b) RMSE of the image and evolution of the residuals for each parameter, as well as equal-sample comparison of our results with Zhang et al.'s steady-state renderer [2020]. Although the error scales are not directly comparable, the optimization of the index of refraction in the steady-state approach converges more slowly in the *Coffee* scene, while failing to converge in *Stars*.

Table 2. Rendering and optimization data for our application scenarios. Since there is no geometry change in the *Coffee* scene, we do not need to perform boundary integral for this scene. Note that *Wavy* and *Corners* scenes requires high spps due to indirect-dominant NLOS scenario.

Scene	Coffee	Stars	Wavy	Corners
# spp (interior)	41	1	17	85
# spp (boundary)	0	4	34	85
# frames	250	800	1200	1200
# param.	2	2	3	1
# iter.	19	129	69	159
Time per iter.	28.5min.	5.7min.	36min.	32.8min.
Learning rate	0.07	0.07	0.07	40.0

7 DISCUSSION AND CONCLUSIONS

We have presented a framework for differential transient rendering, based on the key observation that in transient state scattering events at path vertices are no longer independent. Instead, tracking the time of flight of light requires treating such scattering events jointly as a multidimensional, evolving manifold. For this, we have relied on the generalized transport theorem, and introduced a correlated importance function which allows us to handle discontinuous importance functions for both the light and the sensor. We have shown how to incorporate our framework in a Monte Carlo renderer, and demonstrated its application in several challenging scenarios. The cost of computing our derivates depends on the scene, due to next-event estimation and importance sampling over edges in the boundary integral (similar to Zhang's method). In the worst case (*Tower*) the overhead is 72% with respect to standard transient rendering, whereas for our NLOS results (*Wavy* and *Corners*) it is just 10%. Since a finite differences approach scales with the number of optimized parameters, our method is faster for all scenes shown.

Relation to the steady-state differential path integral. Our work can be seen as a generalization of the steady-state differential path integral [Zhang et al. 2020] to higher-dimensional domains, and handling discontinuities beyond the geometric visibility term. As expected, when our time-dependent terms disappear, the temporal boundary in path space vanishes and our differential transient path integral converges to its steady-state counterpart.

Limitations and future work. Several exciting avenues of future work lie ahead. From a theoretical point of view, it would be interesting to include Dirac delta functions in the temporal domain for both the light and the sensor. From a more practical perspective, assuming non-negligible scattering delays would allow handling fluorescent materials [Gutierrez et al. 2008], thus extending the applicability of our method to time-resolved fluorescence lifetime imaging (FLI) [Satat et al. 2015]. This would require re-introducing the temporal dimension on the paths throughput, as well as the scattering delays inside the path time of flight. Extending our differentiable transient framework to participating media [Zhang et al. 2021] should be possible, adding *indirect shadow vertices* to control the path duration during sampling [Jarabo et al. 2014]. Exploiting this additional degree of freedom is a promising avenue for improving sampling of the derivatives. Currently our work shares Zhang et al.'s [2020] limitation of lack of reciprocity. However, by using the generalized manifold transport theorem over the k-dimensional evolving manifold of each path, a fully reciprocal boundary path space $\overline{\partial \Omega}$ and normal velocity $\mathcal{V}_{\overline{\partial \Omega}}$ could be derived. Last, our framework can be used to optimize any parameters describing the scene, including geometry, light sources, or materials. However, similar to Zhang's work [2020], our current implementation is memory-bounded, which in our case is aggravated by having to compute a

full transient sequence. Computing the derivatives in more complex scenarios with potentially thousands of parameters would benefit from faster differentiation strategies, which could be obtained by combining our work with recent advances on efficient backpropagation of derivatives [Nimier-David et al. 2020; Vicini et al. 2021].

ACKNOWLEDGMENTS

REFERENCES

A Adam, C Dann, O Yair, S Mazor, and S Nowozin. 2016. Bayesian Time-of-Flight for Realtime Shape, Illumination and Albedo. *IEEE Trans. Pattern Analysis and Machine Intelligence* (2016).

Marco Ament, Christoph Bergmann, and Daniel Weiskopf. 2014. Refractive radiative transfer equation. *ACM Trans. Graph.* 33, 2 (2014).

Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased warped-area sampling for differentiable rendering. *ACM Trans. Graph.* 39, 6 (2020), 1–18.

Paolo Cermelli, Eliot Fried, and Morton E Gurtin. 2005. Transport relations for surface integrals arising in the formulation of balance laws for evolving fluid interfaces. *Journal of Fluid Mechanics* 544 (2005), 339–351.

Wenzheng Chen, Jun Gao, Huan Ling, Edward J Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. 2019. "Learning to predict 3D objects with an interpolation-based differentiable renderer". In *Proc. NeurIPS 2019*.

Wenzheng Chen, Fangyin Wei, Kiriakos N. Kutulakos, Szymon Rusinkiewicz, and Felix Heide. 2020. Learned Feature Embeddings for Non-Line-of-Sight Imaging and Recognition. *ACM Trans. Graph.* 39, 6 (2020).

Ioannis Gkioulekas, Anat Levin, and Todd Zickler. 2016. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *European Conference on Computer Vision (ECCV)*. Springer, 685–701.

Diego Gutierrez, Francisco J Seron, Adolfo Munoz, and Oscar Anson. 2008. Visualizing underwater ocean optics. *Computer Graphics Forum* 27, 2 (2008), 547–556.

Felix Heide, Matthias Hullin, James Gregson, and Wolfgang Heidrich. 2013. Low-budget Transient Imaging Using Photonic Mixer Devices. *ACM Trans. Graph.* 32, 4 (2013).

Felix Heide, Lei Xiao, Andreas Kolb, Matthias B Hullin, and Wolfgang Heidrich. 2014. Imaging in scattering media using correlation image sensors and sparse convolutional coding. *Opt. Express* 22, 21 (2014).

Julian Iseringhausen and Matthias B Hullin. 2020. Non-line-of-sight reconstruction using efficient transient rendering. *ACM Trans. Graph.* 39, 1 (2020), 1–14.

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

Adrian Jarabo and Victor Arellano. 2018. Bidirectional rendering of vector light transport. *Computer Graphics Forum* 37, 6 (2018), 96–105.

Adrian Jarabo, Julio Marco, Adolfo Munoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. 2014. A framework for transient rendering. *ACM Trans. Graph.* 33, 6 (2014), 1–10.

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).

Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. 2020. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057* (2020).

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3907–3916.

Jaroslav Krivánek, Iliyan Georgiev, Anton Kaplanyan, and Juan Canada. 2013. Recent Advances in Light Transport Simulation: Theory and Practice. In *ACM SIGGRAPH 2013 Courses*.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. 2020. Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.* 39, 6 (2020), 1–14.

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6 (2018), 1–11.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. 2019b. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *IEEE International Conference on Computer Vision (ICCV)*. 7708–7717.

Xiaochun Liu, Ibón Guillén, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Adrian Jarabo, Diego Gutierrez, and Andreas Velten. 2019a. Non-Line-of-Sight Imaging using Phasor Fields Virtual Wave Optics. *Nature* (2019).

Matthew M Loper and Michael J Black. 2014. OpenDR: An approximate differentiable renderer. In *European Conference on Computer Vision (ECCV)*. Springer, 154–169.

Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.* 38, 6 (2019), 1–14.

Julio Marco, Ibón Guillén, Wojciech Jarosz, Diego Gutierrez, and Adrian Jarabo. 2019. Progressive Transient Photon Beams. *Computer Graphics Forum* (2019).

Julio Marco, Adrian Jarabo, Ji Hyun Nam, Xiaochun Liu, Miguel Ángel Cosculluela, Andreas Velten, and Diego Gutierrez. 2021. Virtual light transport matrices for non-line-of-sight imaging. In *IEEE International Conference on Computer Vision (ICCV)*.

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020), 146–1.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans. Graph.* 38, 6 (2019), 1–17.

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 Trans. Graph.* 33, 4 (2014).

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.

Xian Pan, Victor Arellano, and Adrian Jarabo. 2019. Transient instant radiosity for efficient time-resolved global illumination. *Computers & Graphics* 83 (2019), 107–113.

Adithya Pediredla, Ashok Veeraraghavan, and Ioannis Gkioulekas. 2019. Ellipsoidal path connections for time-gated rendering. *ACM Trans. Graph.* 38, 4 (2019), 1–12.

Phil Pitts, Arrigo Benedetti, Malcolm Slaney, and Phil Chou. 2014. *Time of Flight Tracer.* Technical Report. Microsoft.

Guy Satat, Barmak Heshmat, Christopher Barsi, Dan Raviv, Ou Chen, Moungi G Bawendi, and Ramesh Raskar. 2015. Locating and classifying fluorescent tags behind turbid layers using time-resolved inversion. *Nature communications* 6, 1 (2015), 1–8.

Guy Satat, Barmak Heshmat, Nikhil Naik, Albert Redo-Sanchez, and Ramesh Raskar. 2016. Advances in ultrafast optics and imaging applications. In *Ultrafast Bandgap Photonics*, Vol. 9835. International Society for Optics and Photonics, 98350Q.

Brian Seguin and Eliot Fried. 2014. Roughening it—evolving irregular domains and transport theorems. *Mathematical Models and Methods in Applied Sciences* 24, 09 (2014), 1729–1779.

Adam Smith, James Skorupski, and James Davis. 2008. *Transient Rendering.* Technical Report UCSC-SOE-08-26. School of Engineering, University of California, Santa Cruz.

Chia-Yin Tsai, Aswin C Sankaranarayanan, and Ioannis Gkioulekas. 2019. Beyond Volumetric Albedo–A Surface Optimization Framework for Non-Line-Of-Sight Imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1545–1555.

Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation.* Vol. 1610. Stanford University PhD thesis.

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 Trans. Graph.* 32, 4 (2013), 1–8.

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *ACM Trans. Graph.* 40, 4 (Aug. 2021), 108:1–108:14. https://doi.org/10.1145/3450626.3459804

Di Wu, Andreas Velten, Matthew O'Toole, Belen Masia, Amit Agrawal, Qionghai Dai, and Ramesh Raskar. 2014. Decomposing Global Light Transport using Time of Flight Imaging. *IEEE International Conference on Computer Vision (ICCV)* 107, 2 (2014).

Rihui Wu, Adrian Jarabo, Jinli Suo, Feng Dai, Yongdong Zhang, Qionghai Dai, and Diego Gutierrez. 2018. Adaptive Polarization-Difference Transient Imaging for Depth Estimation in Scattering Media. *Optics Letters* 43, 6 (2018).

Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM Trans. Graph.* 39, 6 (2020), 143.

Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. 2019. A differential theory of radiative transfer. *ACM Trans. Graph.* 38, 6 (2019), 1–16.

Cheng Zhang, Zihan Yu, and Shuang Zhao. 2021. Path-Space Differentiable Rendering of Participating Media. *ACM Trans. Graph.* 40, 4 (2021), 76:1–76:15.

Shuang Zhao, Wenzel Jakob, and Tzu-Mao Li. 2020. Physics-Based Differentiable Rendering: From Theory to Implementation. In *ACM SIGGRAPH 2020 Courses*.