Progressive Transient Photon Beams

Julio Marco\textsuperscript{1} Ibón Guíllén\textsuperscript{1} Wojciech Jarosz\textsuperscript{2} Diego Gutierrez\textsuperscript{1} Adrian Jarabo\textsuperscript{1}

\textsuperscript{1} Universidad de Zaragoza, I3A \textsuperscript{2} Dartmouth College

Abstract

In this work we introduce a novel algorithm for transient rendering in participating media. Our method is consistent, robust, and is able to generate animations of time-resolved light transport featuring complex caustic light paths in media. We base our method on the observation that the spatial continuity provides an increased coverage of the temporal domain, and generalize photon beams to transient state. We extend steady-state photon beam radiance estimates to include the temporal domain. Then, we develop a progressive variant of our approach which provably converges to the correct solution using finite memory by averaging independent realizations of the estimates with progressively reduced kernel bandwidths. We derive the optimal convergence rates accounting for space and time kernels, and demonstrate our method against previous consistent transient rendering methods for participating media.

CCS Concepts

\textbullet Computer Graphics\rightarrow Three-dimensional graphics and realism; Raytracing; Transient rendering;

1. Introduction

The emergence of transient imaging has led to a vast number of applications in graphics and vision [JMMG17], where the ability of sensing the world at extreme high temporal resolution allows new applications such as imaging light in motion [VWI\textsuperscript{*}13], appearance capture [NZV\textsuperscript{*}11], geometry reconstruction [BH04, MHM\textsuperscript{*}17], or vision through media [Bus05, WJS\textsuperscript{*}18] and around the corner [VWG\textsuperscript{*}12, AGJ17]. Sensing through media is one of the key applications: The ability of demultiplexing light interactions in the temporal domain is a very promising approach for important practical domains such as non-invasive medical imaging, underwater vision, or autonomous driving through fog. Accurately simulating light transport could help enormously in these applications, potentially serving as a benchmark, a forward model in optimization, or as a training set for machine learning.

Transient rendering in media is, however, still challenging: The increased dimensionality (time) increases variance dramatically in Monte Carlo algorithms, potentially leading to impractical rendering times. This variance is especially harmful in media, where the signal tends to be smooth due to the low-pass filtering behavior of scattering, in both the spatial and temporal domains. One of the major drawbacks of transient rendering is that it requires much
higher sampling rates to fill up the extended temporal domain, special-

ially when using 0D (photon) point samples, which are sparsely
distributed across both time and space. We make the observation
that 1D photon trajectories populate both space and time much
more densely; hence, a technique based on photon beams [JNSJ11]
should significantly reduce the rendering time when computing a
noise-free time-resolved render, and, given its density estimation
nature, it could naturally combine with the temporal domain den-
sity estimation proposed by Jarabo et al. [JMM*14].

We present a new method for transient-state rendering of par-
ticipating media, that leverages the good properties of density es-
timation for reconstructing smooth signals. Our work improves
Jarabo et al. [JMM*14] by extending \textit{progressive photon beams}
(PPB) [JNT*11] to the transient domain, and combining it with the
temporal density estimation for improved reconstruction in both the
spatial and temporal domains. Our technique is biased but consist-
ent, converging to the ground truth using finite memory by taking
advantage on the progressive [HOJ08, KZ11] nature of density es-
timation. We analyze the asymptotic convergence of our proposed
space-time density estimation, computing the optimal kernel reduc-
ion ratios for both domains. Finally, we demonstrate our method
on a variety of scenes with complex volumetric light transport, fea-
turing high-frequency occlusions, caustics, or glossy reflections,
and show its improved performance over naïvely extending PPB
to the transient domain.

This paper is an extension of our previous work on rendering
transient volumetric light transport [MJGJ17], where we proposed
a naive extension of photon beams to transient state. Here we in-
crease the applicability of the method, by proposing a progressive
version of the space-time density estimation, and rigorously ana-
lyze its convergence.

2. Related Work

Rendering participating media is a long-standing problem in com-
puter graphics, with a vast literature on the topic. Here we focus
on works related directly with the scope of the paper. For a wider
overview on the field, we refer to the recent survey by Novák et
al. [NGHJ18].

\textbf{Photon-based Light Transport.} Photon mapping [Jen01] is one
of the most versatile and robust methods for rendering complex
\textit{global} illumination, with several extensions for making it com-
patible with motion blur [CJ02], adapting the distribution of pho-
ton density for the same photon tracing step. Their progressive
and hybrid counterparts [JNT*11, KGH*14] leveraged the benef-
fits of photon beams while providing consistent solutions using fi-
nite memory. Recently, Bitterli and Jarosz [BJ17] generalized 0D
photon points and 1D photon beams to even higher dimensions,
proposing the use of photon planes (2D), volumes (3D) and, in
theory, higher-dimensional geometries, leading to unbiased density
estimation. All these works are, however, restricted to steady-state
renders; we instead focus on simulating light transport in transient
state.

\textbf{Transient rendering.} Though the transport equations [Cha60,
Gla95] are time- resolved, most rendering algorithms focus on
steady-state light transport. Still, several works have been pro-
duced to deal with light transport in a time-resolved manner.
In particular, most previous work on transient rendering has fo-
cused on simulating surfaces transport: Kleim et al. [KPM*16] ex-
tended Smiths’\textit{’} transient radiosity [SSD08] for second bounce dif-
fuse illumination, while other work has used more general meth-
ods based on transient extensions of Monte Carlo (bidirectional)
path tracing [Jar12, JMM*14, PBSC14, JA18] and photon mapping
[MNJ13, OHX*14]. Several works have also dealt with time-
resolved transport on the field of neutron transport [CPH13, BG70,
WJ11, DM79]. Closer to our work, Ament and colleagues [ABW14]
rendered transient light transport in refractive media using volu-
metric photon mapping, but they do not provide an efficient ap-
proach that guarantees consistency. Jarabo et al. [JMM*14] pro-
duced a transient extension of the path integral, and introduced an
efficient technique for reconstructing the temporal signal based on
density estimation. They also proposed a set of techniques for sam-
ping media interactions uniformly in time. Their method is how-
ever limited to bidirectional path tracing and photon mapping, of-
ten failing to densely populate media in the temporal domain. Fi-
ally, Bitterli [Bit16b] and Marco et al. [Mar13, MJGJ17] proposed
a transient extension of the photon beams algorithm, but these ap-
proaches are not progressive, therefore not converging to the cor-
rect solution in the limit. Our work extends the latter, proposing a
progressive, consistent, and robust method for rendering transient
light transport. We leverage beams continuity and spatio-temporal
density estimation to mitigate variance in the temporal domain, and
derive the parameters for optimal convergence of the method.

3. Transient Radiative Transfer

The \textit{radiative transfer equation} (RTE) [Cha60] models the behav-
ior of light traveling through a medium. While the original for-
mulation is time-resolved, its integral form used in traditional render-
ing ignores this temporal dependence, and computes the radiance $L$
reaching any point $x$ from direction $\hat{\omega}$ as

$$L(x, \hat{\omega}) = T_r(x, x_0) L_0(x_0, \hat{\omega}) + \int_0^1 \mu_s(x_q) T_r(x, x_q) L_0(x_q, \hat{\omega}) dq,$$

where $x_q = x - d \cdot \hat{\omega}$ is a point at distance $d$, $\mu_s$ is the scattering
coefficient, and $T_r(x, x_q) = \exp(-\int_0^d \mu_s(x_q') dq')$ is the \textit{transmit-
tance} describing the fraction of photons that make it between $x$ and
$x_0$ without undergoing extinction at any point $x_q$, determined by the
\textit{extinction coefficient} $\mu_s(x_q')$. The outgoing radiance $L_0$ in
direction $\hat{\omega}$ from a medium point $x_q$ at distance $q$ is defined by the
\begin{equation}
\end{equation}
scattering integral:

\[ L_s(x_q, \omega) = L_o(x_q, \omega) + \int_S f_s(x_q, \omega_0, \omega) L(x_q, \omega_0) \, d\omega_0, \]  

(2)

where \( S \) is the spherical domain, and \( f_s \) is the phase function. \( L_o \) is defined analogously via the rendering equation [Kaj86], but integrated over the hemispherical domain, and using the cosine-weighted BSDF in place of the phase function.

Transient RTE  Equations 1 and 2 assume that the speed of light is infinite. However, if we want to solve the RTE at time scales comparable to the speed of light we need to incorporate the different delays affecting light. In the following we review the main practical considerations for accounting time into the integral form of the RTE for its application in transient rendering. Light takes a certain amount of time to propagate through space, and therefore light transport from a point \( x_0 \) towards a point \( x_1 \) does not occur immediately. In the absence of scattering effects, transport between two points \( x_0 \) and \( x_1 \) occurs as

\[ L(x_1, \omega_0, t) = L(x_0, -\omega_0, t - \Delta t), \]  

(3)

where \( \Delta t \) is the time it takes the light to go from \( x_0 \) to \( x_1 \). In turn, \( \Delta t \) is defined by

\[ \Delta t(x_0 \leftrightarrow x_1) = \int_{x_0}^{x_1} \frac{\eta(x)}{c} \, dx, \]  

(4)

where \( \eta(x) \) is the index of refraction at a medium point \( x \) and \( c \) is the speed of light in vacuum. Note that in this case light does not travel in a straight line, but by following the Eikonal equation [ABW14,GMA05]. In a medium with a constant index of refraction \( \eta(x) = \eta_0 \), then \( \Delta t(x_0 \leftrightarrow x_1) \) can be expressed as

\[ \Delta t(x_0 \leftrightarrow x_1) = \frac{\eta_0}{c} |x_1 - x_0|. \]  

(5)

The second form of delay occurs in the scattering events, and might occur from different sources, including electromagnetic phase shift, fluorescence and phosphorescence, or multiple scattering within the surface (or particle) microgeometry. To account for these sources of scattering delays, we introduce a temporal variable in the phase function as \( f_t(x, \omega_0, \omega, t) \), where \( t \) is the instant of light interacting with the particle before it is scattered. With those delays in place, we reformulate the RTE (Equations 1 and 2) introducing the temporal dependence as [Gla95]

\[ L(x, \omega_0, t) = T_r(x, x_0) L_s(x_0, \omega_0, t - \Delta \rho), \]  

(6)

\[ L(x_0, \omega_0, t) = \int_0^\rho \mu_s(x_0) T_l(x, x_0) L_s(x_0, \omega_0, t - \Delta \rho) \, dq, \]  

(7)

\[ L(x_0, \omega_0, t) = \int_0^\rho \mu_s(x_0) T_l(x, x_0) L_s(x_0, \omega_0, t - \Delta \rho) \, dq, \]  

(8)

with \( \Delta \rho = \Delta t(x_0 \leftrightarrow x_0) \) and \( \Delta \rho_t = \Delta t(x_0 \leftrightarrow x_0) \) (Equation (4)). \( L_s \) changes analogously. Note that we assume that the matter does not change at time-scales comparable to the speed of light, and therefore avoid any temporal dependence on \( \mu_s \) and \( \mu_t \). Introducing temporal variation at such speeds would produce visible relativistic effects [WK09, JMV15].
seen by a camera ray is computed by performing a density estimation on every ray-beam intersection. For 1D and 2D kernels, this radiance is computed as

$$L^\text{1D}_b(x, \bar{\omega}_b) = K_{1D}(R_b) \Phi_b f_b(\theta_b) \mu_b e^{-\mu_b s_b} e^{-\mu_b s_b},$$

$$L^\text{2D}_b(x, \bar{\omega}_b) = K_{2D}(R_b) \Phi_b f_b(\theta_b) \mu_b e^{-\mu_b (s' - s')} e^{-\mu_b (\cos \theta_b - \cos \theta_b)},$$

where the beam is defined by $x_b + s_b \bar{\omega}_b$ and the ray is defined by $x + s \bar{\omega}_b$ (see setups in Figure 3a).

4.1. Our algorithm

To generalize photon beams to the transient domain, we need to account for the duration of light paths. This requires considering propagation and scattering delays along the camera and light subpaths, but also the effect of time in the density estimation connecting these two subpaths.

Creating the photon map We compute the photon propagation as a standard random walk through the scene, which can be modeled using the subpath formulation defined by Jarabo et al. [JMM+14]. Let us define a light subpath $x_i = x_0, x_1, ..., x_k$ with $k$ vertices, where $x_0$ is the light source. This path defines $k - 1$ photon beams, in which a beam is defined by its origin at $x_0 = x_i$ and direction $\bar{\omega}_b = \frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}$. Using Jarabo’s definition of the path integral (and therefore of the contribution of the subpaths), we compute the flux of each photon as:

$$\Phi_b = \frac{f(S_j, t_j)}{M p(S_j, t_j)} = L_e(x_0 \rightarrow x_1, \tau_0) T(S_j, t_j),$$

with $S_j$ the subpath of $S_i$ up the vertex $j$, $f$ the subpath contribution function, $\tau_j = \tau_0, ..., \tau_j$ the sequence of time delays up to vertex $j$, $M$ the number of photon random walks sampled, $L_e(x_0 \rightarrow x_1, \tau_0)$ the emission function, $p(x_1, \tau_j)$ the probability density of sampling vertex $x_i$ with time delay $\tau_j$. The throughput, $T(S_j, t_j)$, of subpath $(S_j, \tau_j)$ is defined as:

$$T(S_j, t_j) = \prod_{i=0}^{j-1} f_r(x_i, \tau_i) \prod_{i=0}^{j-1} G(x_i, x_{i+1}) V(x_i, x_{i+1}),$$

with $f_r(x_i, \tau_i)$ the scattering event at vertex $x_i$ with delay $\tau_i$, and $G(x_i, x_{i+1})$ and $V(x_i, x_{i+1})$ the geometry and visibility terms between vertices $x_i$ and $x_{i+1}$, respectively. Finally, for transient state we need to know the instant $t_b$ at which the photon beam is created (through emission or scattering), defined as:

$$t_b = \sum_{j=0}^{j-1} \tau_j + \sum_{j=0}^{j-1} \Delta t(S_i, x_{i+1}).$$

Rendering For rendering, we adapt Equation (8) to account for the temporal domain, as

$$L(x, \bar{\omega}_b, t) = \sum_{b \in R_b} L_b(x, \bar{\omega}_b, t),$$

with $L_b(x, \bar{\omega}_b, t)$ the radiance estimation for beam $b$ to ray $t$ at instant $t$. In essence, $L_b(x, \bar{\omega}_b, t)$ will return zero radiance if $t$ is out of the temporal footprint of the density estimation kernel. Depending on the dimensionality of the density estimation, Jarosz and colleagues [JNSJ11] proposed three different estimators based on 3D, 2D, and 1D kernels. Since the 3D kernel results impractical due to costly 3D convolutions, we focus on 1D and 2D kernels (Equations (9) and (10)), and extend them to transient state, assuming homogeneous media.

Kernel 2D We generalize Jarosz’s et al.’s 2D estimate $L^\text{2D}_b$ (Equation (10)) by introducing a temporal function $W(t)$ as

$$L^\text{2D}_b(x, \bar{\omega}_b, t) = K_{2D}(R_b) \Phi_b f_b(\theta_b, t) \mu_b e^{-\mu_b (s' - s')} e^{-\mu_b (\cos \theta_b - \cos \theta_b)},$$

where $[s', s']$ are the limits of the ray-beam intersection (Figure 3a), $\theta_b$ is the angle between $\bar{\omega}_b$ and $\bar{\omega}_t$, and $K_{2D}(R_b)$ is a canonical 2D kernel with radius $R_b$. The temporal function $W(t)$ models the temporal footprint of the 2D kernel as

$$W(t) = \begin{cases} \frac{1}{t} & \text{if } t \in (-t^- \cdot t^+), \\ 0 & \text{otherwise}, \end{cases}$$

with $t^- = t_b + t_b + \frac{\mu_b}{\mu_b}(s^+ + s^-)$ and $t^+ = t_b + t_b + \frac{\mu_b}{\mu_b}(s^+ + s^-)$, and $t_b$ and $t_b$ are the initial times of the camera ray and beam, respectively. Note that due to transmittance, the photon energy varies as it travels across the blur region. Evenly distributing the integrated radiance $L_b$ across this interval introduces temporal bias, in addition to the inherent spatial bias introduced by density estimation. However we observed this even distribution provides a good trade-off between bias, variance, and computational overhead.

Kernel 1D In the 1D kernel defined for density estimation by Jarosz et al., the spatial blur is performed over a line. Therefore, the energy of the beam is just spread on the ray on a single point at $r(s_r)$, from a single point of the beam $b(s_b)$ (see Figure 3a). In consequence, $s_b \rightarrow s_r$ and $s_b \rightarrow s_r$, which implies that $t^- \rightarrow t_b$, and the temporal function reduces to $W_{1D}(t - t_b) = \delta(t)$, with $\delta(t)$ the Dirac delta function. With that in place, we transform Jarosz et al. 1D estimate to

$$L^\text{1D}_b(x, \bar{\omega}_b, t) = K_{1D}(R_b) \Phi_b f_b(\theta_b, t) \mu_b e^{-\mu_b s_b} e^{-\mu_b s_b},$$

with $K_{1D}(R_b)$ a 1D kernel with radius $R_b$.

Implementation Since photon beams correspond to full photon trajectories, they allow us to estimate radiance at any position $x_b + s_b \bar{\omega}_b$ of the beam, and therefore at any arbitrary time $t(x_b + s_b \bar{\omega}_b)$. As mentioned, one-dimensional radiance estimate corresponds to a single time across the beam. In a traditional rendering process where camera rays are traced through view-plane pixels against the beams map, the temporal definition within a pixel will be proportional to the amount of samples per pixel taken. Additionally, 2D blur requires distributing every radiance estimate along a time interval, which reduces variance in the time dimension of a pixel at the expense of introducing additional temporal bias.
Finally, note that the temporal footprint of the density estimation might be arbitrarily small, so the probability of finding a beam $b$ at an specific time might be very low. We alleviate this issue using path reuse via density estimation [JMM+14]. In particular, for the non-progressive results we use histogram temporal density estimation. In this technique, the samples in the temporal domain are reused across all frames by evaluating their contribution functions, which correspond to the temporal window covered by each frame. In Section 5 we introduce temporal kernel-based density estimation, and combine it with the spatial density estimation of the beam.

5. Progressive Transient Photon Beams

By means of Equations (15) and (17) we have introduced temporal dependence on the spatial density estimations that use 2D and 1D kernels, respectively. These density estimations reduce variance at the expense of introducing bias in the results, which means both Equations (8) and (14) will not converge to the correct solution, even with an infinite number of photons $M$. To avoid this, progressive density estimation aims to provide a biased, yet consistent technique, that in the limit converges to the expected value (in other words, the bias vanishes in the limit). The key idea is to average several render passes with a finite number of photon random walks $M$, progressively reducing the bias in each iteration while allowing variance to slightly increase.

In order to fully leverage a progressive approach, we propose to combine our time-resolved spatial density estimations (Section 4) with additional temporal density estimations. While our time-resolved 2D spatial kernel implicitly performs a temporal blur over the interval $[t_i, t_{i+1}]$, it is coupled with the spatial blur. This does not allow to choose its own initial kernel size for the temporal density estimation, which is a desirable degree of freedom since the temporal resolution may not be proportional to the spatial one. In contrast, our time-resolved 1D spatial kernel does not perform a temporal blur, since the footprint is a single instant in time. As we show in the remainder of this section, this allows us to perform additional progressive density estimations with an independent initial kernel size, while keeping the same two-dimensionality (1D spatial and 1D temporal). In the following, we introduce our spatio-temporal beam density estimation based on our time-resolved 1D kernel, and then present our progressive approach.

Spatio-Temporal Beam Estimation Jarabo et al. [JMM+14] showed that progressive density estimations in the temporal domain can in fact improve the convergence rate for transient rendering, in particular when compared with the histogram method used in Section 4 for rendering the temporal domain. To combine such approach with the (progressive) spatial density estimation in photon beams [JNT+11], we reformulate the 1D kernel in Equation (17), by convolving it with a 1D temporal kernel $K_T(t)$ so that

$$L_{b\text{T}}^1(x, \theta_b, t) = K_{1D}(R_b) \Phi_b f_s(\theta_b, t) \mu_b \frac{e^{-\mu_b h_b}}{\sin \theta_b} K_T(t - t_b).$$

(18)

Progressive Transient Photon Beams We generalize the computation of $L(x, \theta_b, t)$ (Equation (14)) using an iterative estimator, defined as

$$L_b \leftarrow 0$$

$$R_b \leftarrow R_b$$

$$T \leftarrow T_0$$

\textbf{for} $i \in [0..N)$ \textbf{do}

$$r \leftarrow \text{traceRay}(\cdot)$$

$$B \leftarrow \text{beamsMap}(\cdot)$$ \textbf{(Eqs. (6), (7), (11)-(13))}

$$R_b \leftarrow R_b \sqrt{\frac{t_i + 2/3}{t_{i+1} + 2/3}} \textbf{ (Eq. (20), left)}$$

$$T \leftarrow T \sqrt{\frac{t_i + 2}{t_{i+1} + 2}} \textbf{ (Eq. (20), right)}$$

$$L_b \leftarrow 0$$

\textbf{for} $b \in B$ \textbf{do}

$$L_b \leftarrow L_b + \text{radiance}(r, b, R_b, T) \textbf{ (Eq. (18))}$$

\textbf{end for}

$$L_n \leftarrow L_n + L_b$$

\textbf{end for}

Algorithm 1 Pseudo-code of our progressive spatio-temporal density estimation.

$$L_0 \leftarrow 0$$

$$R_0 \leftarrow R_0$$

$$T \leftarrow T_0$$

\textbf{for} $i \in [0..N)$ \textbf{do}

$$r \leftarrow \text{traceRay}(\cdot)$$

$$B \leftarrow \text{beamsMap}(\cdot)$$ \textbf{(Eqs. (6), (7), (11)-(13))}

$$R_b \leftarrow R_b \sqrt{\frac{t_i + 2/3}{t_{i+1} + 2/3}} \textbf{ (Eq. (20), left)}$$

$$T \leftarrow T \sqrt{\frac{t_i + 2}{t_{i+1} + 2}} \textbf{ (Eq. (20), right)}$$

$$L_b \leftarrow 0$$

\textbf{for} $b \in B$ \textbf{do}

$$L_b \leftarrow L_b + \text{radiance}(r, b, R_b, T) \textbf{ (Eq. (18))}$$

\textbf{end for}

$$L_n \leftarrow L_n + L_b$$

\textbf{end for}

$$L_b \leftarrow L_b + \text{radiance}(r, b, R_b, T) \textbf{ (Eq. (18))}$$

with $L_n$ the estimate of $L$ after $n$ iterations, and $B_i$ the set of photon beams in iteration $i$. Note that the previous equation assumes that the camera ray $r$ is the same for all iterations. That is not necessarily true (and in fact it is not) but for simplicity we express this way.

The error of the estimate $L_n$ is defined by its bias and variance, which as shown in Appendix B is dependent on the bandwidth of the kernels, while bias is reduced at the same rate. Then, on each iteration we reduce the bias by allowing the variance to increase at a controlled rate of $(i + 1)/(i + 1 + \alpha)$, with $\alpha \in [0, 1]$ being a parameter that controls how much the variance is allowed to increase at each iteration. To achieve that reduction, on each iteration $i + 1$ we reduce the foot-print of kernels $K_D$ and $K_T$ ($R_{b|i}$ and $T_{i+1}$) by

$$\frac{R_{b|i+1}}{R_{b|i}} = \left(\frac{i + 1 + \alpha}{i + 1}\right)^{\beta_b} , \quad \frac{T_{i+1}}{T_i} = \left(\frac{i + 1 + \alpha}{i + 1}\right)^{\beta_T},$$

(20)

where $\beta_b$ and $\beta_T$ control the individual reduction ratio of each kernel, with $\beta_T = 1 - \beta_b$. A pseudo code of the main steps of our progressive approach can be found in Algorithm 1. In the following, we analyze the convergence rate of the method, and compute the optimal values for the parameters $\alpha$, $\beta_T$ and $\beta_b$.

Convergence analysis We analyze the convergence of the algorithm as a function of the asymptotic mean squared error (AMSE) defined as

$$\text{AMSE}(L_n) = \text{Var}(L_n) + E[e_n]^2,$$

(21)

where $\text{Var}(L_n)$ is the variance of the estimate and $E[e_n]$ is the bias at iteration $n$ (see Appendix A). As shown in Appendix C, the variance converges with rate

$$\text{Var}(L_n) \approx O(n^{-1}) + O(n^{-\alpha}) = O(n^{-\alpha}),$$

(22)
while the bias converges with rate
\[ E(n_a) = O(n^{-\alpha}) - 2\beta T + O(n^{-\alpha_2}2\beta T - 2). \]

Plugging Equation (22) and (23) into Equation (21), we can model the AMSE as
\[ \text{AMSE}(\hat{L}_n) = O(n^{-\alpha}) + (n^{-\alpha} - 2\beta T + O(n^{-\alpha_2}2\beta T - 2))^2. \]

Finally, by minimizing Equation (24) (see Appendix D) we obtain the values for optimal asymptotic convergence $\beta T = 1/2$ and $\alpha = 2/3$, which by substitution gives us the final asymptotic convergence rate of our progressive transient photon beams
\[ \text{AMSE}(\hat{L}_n) = O(n^{-\frac{2}{3}}). \]

6. Results

In the following we illustrate the results of our proposed method in five scenes: Cornell spheres, Mirrors, Pumpkin, Soccer [SZLG10], and Juice. See Figures 1 (right), 4, and 8 (left) for steady-state renders of the scenes. Results of Figures 5 and 6 were taken on a desktop PC with Intel i7 and 4GB RAM using a transient 2D kernel (Equation 15). Figures 1, 7, and 8 were rendered on an Intel Xeon E5 with 256GB RAM, using our progressive spatio-temporal kernel density estimations (Section 5) derived from the transient spatial 1D kernel (Equation 17). In each iteration, we use a fixed radius for our spatio-temporal density estimators (instead of using a nearest neighbor approach). Please refer to the supplemental content for the full sequences of all the scenes.

Figure 5 shows a Cornell box filled with a scattering medium, and demonstrates the effect of camera unwarping [VWJ+13] when rendering. Camera unwarping is an intuitive way of visualizing how light propagates locally on the scene without accounting for the time light takes to reach the camera. The scene consists of a diffuse Cornell box with a point light on the top, a glass refractive sphere (top, IOR = 1.5) and a mirror sphere (bottom). While Figure 5b shows the real propagation of light—including camera time—, Figure 5a depicts more intuitively how light comes out from the point light, travels through the refractive sphere, and the generated caustic bounces on the mirror sphere. Note how in the top sequence we can clearly see how light is slowed down through the glass sphere due to the higher index of refraction. We can also observe multiple scattered light (particularly noticeable in frames t=4ns and t=6ns) as a secondary wavefront.

Figure 6 compares visualizations of light propagation within the Mirrors scene using Heaviside and Dirac delta light emission. The scene is composed by two colored mirrors and a glass sphere with IOR = 1.5, and was rendered using the previously mentioned camera unwarping. We can observe how delta emission generates wavefronts that go through the ball and bounce in the mirrors, creating wavefront holes where constant emission creates medium shadows. In the last frame of the top row Delta emission clearly depicts the slowed down caustic through the glass ball respect to the main wavefront.

Our progressive method combines time-resolved 1D spatial kernels of photon beams and temporal density estimations, reducing bias while providing consistent solutions in the limit with an optimal convergence rate of $O(n^{-\frac{2}{3}})$. In Figure 7 we analyze its convergence with respect to progressive transient path tracing with temporal KDE [JMM+14] (PTPT). In the middle graph we show the temporal profile on a single pixel for both our algorithm and PTPT after 4096 equal-time iterations, where both algorithms converge to the reference solution taken with transient path tracing (no temporal KDE) with 64 million samples. While PTPT presents faster convergence (see Figure 7, right graph), our algorithm presents a better behavior over time where variance increases due to the lack of samples (center graph). Additionally, it requires much fewer iterations than PTPT to achieve a similar MSE (see log-log right graph).

In Figure 1 we show a more complex scenario, with different caustics rendered, with our progressive algorithm. It contains a smooth dielectric figurine with different transmission albedos placed within a participating medium with an isotropic phase function. Our method is capable of handling complex caustics transmitted from light sources through the player, and then through the ball. Our algorithm progressively reduces bias and variance to provide a consistent solution.

Finally in Figure 8 we illustrate a setup combining different media properties, and specular refractive and reflective materials. The liquid has a very forward phase function, making the light first travel through the direction of the stream ($t = 4.6$ ns), and then...
Dirac delta emission lets us see how a pulse of light travels and scatters across the scene, depicting the light wavefronts bouncing on the mirrors and going through the glass ball. Continuous emission shows how light is emitted until it reaches every point in the scene, as if we were taking a picture with a camera at very slow-motion.

**Figure 7:** The PUMPKIN scene shows a jack o’lantern embedding a point light that creates hard shadows through the holes. The left frames show a sequence of the time-resolved renders after 4096 iterations of our algorithm (10k beams / iteration), and temporal KDE on a progressive transient path tracer (PTPT, 16spp / iteration) [JMM*14]. The middle plot compares the whole temporal footprint at the pink marker. Reference solution (dark grey) was obtained with a transient path tracer (no KDE) using 64M samples per pixel. Right plot shows MSE convergence with respect to the number of progressive iterations (in log-log scale), at 1 minute/iteration on each algorithm. As expected, the convergence of our method \(O(n^{-\frac{2}{3}})\) is slower than PTPT \(O(n^{-\frac{4}{5}})\); however, as shown in the equal-time comparison, our algorithm presents better temporal behavior with much less variance on later timings.

**Figure 8:** We illustrate the potential of our method in the JUICE scene [Bit16a], which presents a scene very difficult to render for path tracing methods, but well-handled by photon-based methods. The scene is filled by a thin participating medium, while the glass contains ruby grapefruit juice as measured by Narasimhan et al. [NGD*06]. The highly forward phase function of the juice, as well as the delta interactions on the glass, ice cubes, and the mirror floor surface, generate complex caustic patterns which our method is able to simulate in transient state. Bottom row has increased exposure respect to top row to show the radiance at later timings.

7. Conclusions

In this paper we have presented a robust progressive method for efficiently rendering transient light transport with consistent results. We derived our method based on progressive photon beams [JNT*11], extending its density estimators to account for light time-of-flight, and deriving a new progressive scheme. We then compute the convergence of the method, and derive the parameters for optimal asymptotic convergence. Our results demonstrate that combining continuous photon trajectories in transient state and our optimal spatio-temporal convergence rates allow to robustly compute a noise-free solutions to the time-resolved RTE for complex light paths. We believe that our work might be very useful for developing new techniques for transient imaging and reconstruction.
in media, as well as to obtain new insights on time-resolved light transport.

As future work it would be interesting to analyze more thoroughly the optimal performance and kernels for variance reduction and bias impact in transient state, under varying media characteristics. In addition, extending our method to leverage recent advances in media transport, such as transient-state adaptations of higher-dimensional photon estimators [BJ17] as well as hybrid techniques [KGH*14], could improve performance of time-resolved rendering for a general set of geometries and media characteristics.

Acknowledgments

We want to thank the reviewers for their insightful comments. This project has been funded by DARPA (project REVEAL), the European Research Council (ERC) under the EU’s Horizon 2020 research and innovation programme (project CHAMELEON, grant No 682080), the Spanish Ministry of Economy and Competitiveness (project TIN2016-78753-P), the BBVA Foundation, and the Gobierno de Aragón.

Appendix A: Error in Transient Progressive Photon Beams

Here we analyze the consistency of the transient progressive photon beams algorithm described in Section 5. For our analysis on the error of the estimate, we use the asymptotic mean squared error (AMSE) defined as

\[ \text{AMSE}(\hat{L}_n) = \text{Var}[\hat{L}_n] + E[\varepsilon_n]^2, \]  

where \( \text{Var}[\hat{L}_n] \) is the variance of the estimate and \( E[\varepsilon_n] \) is the bias at iteration \( n \). We model \( \text{Var}[\hat{L}_n] \) as [KZ11]

\[ \text{Var}[\hat{L}_n] = \frac{1}{n} \text{Var}[\Psi L] + \frac{1}{n^2} \sum_{j=1}^{n} \text{Var}[\Psi \varepsilon_j], \]

where \( \Psi \) is the contribution of the eye ray, and \( \varepsilon_j \) is the bias for iteration \( j \). The first term is the standard variance of the Monte Carlo estimate, which is unaffected by the kernel. The second term, on the other hand, is the variance of the error, and is dependent on density estimation. On the other hand, the estimated value of the error (bias) \( E[\hat{L}_n] \) is defined as

\[ E[\hat{L}_n] = L + E[\Psi] E[\varepsilon_n], \]

where \( E[\varepsilon_n] \) is the bias of the estimator after \( n \) steps:

\[ E[\varepsilon_n] = \frac{1}{n} \sum_{j=1}^{n} E[\varepsilon_j], \]

with \( E[\varepsilon_j] \) the expected error at iteration \( j \). In the following, we first derive the variance and expected value of the error for a single iteration. Then, we analyze the asymptotic behavior of these terms, and compute the values for optimal convergence for \( \beta_T, \beta_R \) and \( \alpha \).

Appendix B: Variance and Expected Value of the Error of the Time-Resolved Beam Radiance Estimate

We first analyze the variance and expected value of the error (bias) introduced by the time-resolved estimate at each iteration. Let us first define the error in each iteration as:

\[ \varepsilon = \hat{L}_n(x, \omega_r, t) - L(x, \omega_r, t) \]

\[ = \sum_{i=1}^{M} K_{1D}(R_b) K_T(t - t_i) \Phi_i - L(x, \omega_r, t). \]  

(30)

\[ \text{Variance} \]  

We first define the variance of the error \( \text{Var}[\varepsilon] \) as (in the following, we omit dependences for clarity):

\[ \text{Var}[\varepsilon] = \sum_{i=1}^{M} K_{1D} K_T \Phi - L \]  

\[ = \left( \text{Var}[K_{1D}] + E[K_{1D}]^2 \right) \left( \text{Var}[K_T] + E[K_T]^2 \right) \]

\[ \left( \text{Var}[\Phi] + E[\Phi]^2 \right) - E[K_{1D}]^2 E[K_T]^2 E[\Phi]^2, \]

In order to compute the variance of the error \( \text{Var}[\varepsilon] \) we need to make a set of assumptions: First, we assume that the beams’ probability density is constant within the kernel \( K_{1D} \) in the spatial domain [JNT*11], and within \( K_T \) in the temporal domain [JMM*14]. We denote these probabilities as \( p_{R_b} \) and \( p_T \) respectively. We also assume that the distance between view ray and photon beam, time \( t_b \) and beams’ energy \( \Phi_i \) are independent samples of the random variables \( D, T \) and \( \Phi \), respectively, which are mutually independent. Finally, we assume that \( D \) and \( T \) have probability densities \( p_{R_b} \) and \( p_T \).

With these assumptions, and taking into account that \( E[K_{1D}] = p_{R_b} \) and \( E[K_T] = p_T \), we can model the the variance introduced by the temporal kernel \( \text{Var}[K_T] \) as [JMM*14]

\[ \text{Var}[K_T] = \int_{\mathbb{R}} k_T(\psi)^2 d\psi - p_T^2, \]

where we express \( K_T \) as a canonical kernel \( k_T \) with unit integral such that \( K_T(\xi) = k_T(\xi / T)^{-1} \). Analogously, \( \text{Var}[K_{1D}] \) is [JNT*11]:

\[ \text{Var}[K_{1D}] = \int_{\mathbb{R}} k_{1D}(\psi)^2 d\psi - p_{R_b}^2. \]

(33)

This allows us to express the variance of the error \( \text{Var}[\varepsilon] \) as:

\[ \text{Var}[\varepsilon] \approx \left( \text{Var}[\Phi] + E[\Phi]^2 \right) \left( \frac{p_{R_b}}{p_T} C_{1D} \right) \left( \frac{p_T}{p_T} C_T \right), \]

(34)

where \( C_{1D} \) and \( C_T \) are kernel-dependent constants. The last term can be neglect by assuming that the kernels cover small areas in their respective domains, which effectively means that \( C_{1D} \gg p_{R_b} \) and \( C_T \gg p_T \). Equation (34) shows that for transient density estimation, the variance \( \text{Var}[\varepsilon] \) is inversely proportional to \( R_b T \).

\[ \text{Bias} \]  

Bias at each iteration \( j \) is defined as the expected value of the error \( E[\varepsilon_j] \) as

\[ E[\varepsilon_j] = \sum_{i=1}^{M} K_{1D} K_T \Phi - L \]

\[ = E[K_{1D}] E[K_T] E[\Phi] - L. \]

© 2020 The Author(s)

Computer Graphics Forum © 2020 The Eurographics Association and John Wiley & Sons Ltd.
Using a second-order expansion of \( p_T \) and \( p_{\Theta_b} \), instead of the zero-th order used when modeling variance, we can express the expected value of \( K_T \) as [JMM+14]

\[
E[K_T] \approx p_T + T^2 \int_{\mathbb{R}} k_T(\psi)O(\|\psi\|^2)\,d\psi = p_T + T^2 c_{T,2}^D,
\]

(35)

while the expected value of \( K_{1D} \) is [INT+11]

\[
E[K_{1D}] \approx p_{R_c} + R_b \int_{\mathbb{R}^2} k_{1D}(\psi)O(\|\psi\|^2)\,d\psi = p_{R_c} + R_b c_{1D}^D,
\]

(36)

where \( c_{T,2}^D \) and \( c_{1D}^D \) are constants dependent on the higher-order derivatives of the spatio-temporal light distribution. Using (35) and (36), and \( L = p_{R_c} p_T E[\Phi] \) we finally compute \( E[\epsilon_j] \) for iteration \( j \) as

\[
E[\epsilon_j] \approx (p_{R_c} + R_b)^2 c_{1D}^D (p_T + T^2 c_{T,2}^D) E[\Phi] - p_{R_c} p_T E[\Phi] = E[\Phi] \left( p_{R_c} T^2 c_{T,2}^D + p_T R_b c_{1D}^D + T^2 c_{T,2}^D R_b c_{1D}^D \right).
\]

(37)

### Appendix C: Convergence Analysis of Progressive Transient Photon Beams

Based on the expressions for \( \text{Var}[\epsilon] \) and \( E[\epsilon_j] \) defined above (Equations (34) and (37)), we can know derive the asymptotic behaviour of Equation (21). For that, we will compute the variance \( \text{Var}[\hat{L}_n] \) and bias \( E[\epsilon_n] \) after \( n \) iterations.

**Variance** Assuming that the random variables \( \Psi \) and \( \epsilon_j \) are independent, we model the variance of the estimator \( \text{Var}[\hat{L}_n] \) in Equation (27) as [KZ11]:

\[
\text{Var}[\hat{L}_n] = \frac{1}{n} \text{Var}[\Psi] L + \frac{1}{n^2} \sum_{j=1}^{n} \text{Var}[\Psi_j] + E[\Psi]^2 \left( \frac{1}{n} \sum_{j=1}^{n} \text{Var}[\epsilon_j] + \text{Var}[\epsilon] \frac{1}{n^2} \sum_{j=1}^{n} E[\epsilon_j]^2 \right).
\]

Following [KD13], we can approximate \( \text{Var}[\epsilon_n] \) as a function of the variance at the first iteration \( \text{Var}[\epsilon_1] \) as:

\[
\text{Var}[\epsilon_n] \approx \frac{\text{Var}[\epsilon_1]}{(2 - \alpha)n^\alpha} = O(n^{-\alpha}).
\]

(39)

Finally, by applying \( \text{Var}[\epsilon_n] \) and asymptotic simplifications, we can formulate \( \text{Var}[\hat{L}_n] \) as:

\[
\text{Var}[\hat{L}_n] \approx \frac{1}{n} \text{Var}[\Psi] L + E[\Psi]^2 \text{Var}[\epsilon_1] \approx \frac{1}{n} \text{Var}[\Psi] L + \text{Var}[\epsilon_1] \frac{1}{(2 - \alpha)n^\alpha} = O(n^{-\alpha^{-1}}) + O(n^{-\alpha}) = O(n^{-\alpha}).
\]

(40)

**Bias** The expected value of the error \( E[\epsilon_n] \) is modeled in Equation (28) as a function of the averaged bias introduced at each iteration \( E[\epsilon_j] \) (37). Computing the kernels’ bandwidth \( T_j \) and \( R_{0,j} \) at iteration \( j \) by expanding Equation (20) as a function of their initial value by we get

\[
T_j = T_1(j \alpha \Theta(\alpha, j))^{-\beta_T},
\]

(41)

\[
R_{0,j} = R_1(j \alpha \Theta(\alpha, j))^{-\beta_{\Theta_b}},
\]

(42)

where \( \Theta(x, y) \) is the Beta function. Using (41) and (42) in Equation (37) we can express \( E[\epsilon_j] \) as a function of the initial kernel bandwidths

\[
E[\epsilon_j] = E[\Phi] p_{R_c} c_{T,2}^D T_j^{-2} \Theta(j^{-1} - \alpha)^{-2\beta_T} + E[\Phi] p_T c_{1D}^D R_b \Theta(j^{-1} - \alpha)^{-2\beta_{\Theta_b}} + E[\Phi] c_{T,2}^D c_{1D}^D T_j^{-2} R_b \Theta(j^{-1} - 2\beta_T + \beta_{\Theta_b}).
\]

(43)

Finally, we use \( \sum_{j=1}^{n} \Theta(j^2) = n \Theta(n^2) \) to plug Equation (43) into Equation (29) to get the asymptotic behavior of \( E[\epsilon_n] \) in transient progressive photon beams:

\[
E[\epsilon_n] = O(n^{-1} - 2\beta_T + O(n^{-1}) - 2\beta_{\Theta_b} + O(n^{-1})(2\beta_T + \beta_{\Theta_b})\).
\]

which, by using the equality \( \beta_{\Theta_b} = 1 - \beta_T \), becomes:

\[
E[\epsilon_n] = O(n^{-1} - 2\beta_T + O(n^{-1}) - 2\beta_T + O(n^{-1})(2\beta_T + \beta_{\Theta_b})).
\]

(44)

### Appendix D: Minimizing Asymptotic Mean Squared Error

Using the asymptotic expression for variance and bias in Equations (40) and (44), we can express the AMSE (21) as

\[
\text{AMSE}(\hat{L}_n) = O(n^{-\alpha}) + \left( O(n^{-1}) - 2\beta_T + O(n^{-1})(2\beta_T - 2) \right) O(n^{-1} - 2\beta_T + O(n^{-1})(2\beta_T + \beta_{\Theta_b})).
\]

(45)

which is a function of the parameters \( \alpha \) and \( \beta_T \). Given that the variance is independent of \( \beta_T \), we first obtain the optimal value for this parameter that yields the highest convergence rate of the bias \( E[\epsilon_n] \). We differentiate Equation (44), apply asymptotic simplifications and equating to zero, we obtain the optimal value \( \beta_T = 1/2 \). By plugging this value in Equation (45), we obtain:

\[
\text{AMSE}(\hat{L}_n) = O(n^{-\alpha}) + O(n^{-2(1-\alpha)}).
\]

(46)

Finally, by finding the minimum again with respect to \( \alpha \) we get the optimal parameter \( \alpha = 2/3 \), which results in the optimal convergence rate of the AMSE for our transient progressive photon beams as

\[
\text{AMSE}(\hat{L}_n) = O(n^{-\frac{2}{3}}) + O(n^{-2(1-\frac{2}{3})}) = O(n^{-\frac{2}{3}}).
\]

(47)

### References


The Author(s)

© 2020 The Eurographics Association and John Wiley & Sons Ltd.

Computer Graphics Forum © 2020 The Author(s).


Marco et al. / Progressive Transient Photon Beams


[Wil71] Williams M. M. R.: Mathematical methods in particle transport theory. 2
