Supplementary Material: Looking Around Flatland: End-to-End 2D Real-Time NLOS Imaging

María Peña, Diego Gutierrez, Julio Marco

***Abstract*—In this supplementary document, we list the most important symbols used throughout the paper and provide implementation details and additional experiments.**

S.I. Symbols

In the following, we present two tables with the most relevant symbols used throughout the paper to represent terms related to transient light transport (Table II) and NLOS imaging (Table III).

TABLE II
SYMBOLS RELATED TO TRANSIENT LIGHT TRANSPORT.

Symbol	Meaning
I_j	Spatio-temporal measurement
Ψ_{2D}	Space of light transport paths in 2D
$\overline{\mathbf{x}} \in \Psi_{\text{2D}}$	Spatial coordinates of the vertices of a light path between a light source and a camera sensor
ΔT	Space of temporal delays of all paths
$\overline{\mathbf{\Delta t}} \in \Delta T$	Sequence of time delays resulting from the optical distance and scattering events of a path $\overline{\mathbf{x}}$
Δts_i	Scattering delay at path vertex $\mathbf{x}_i$
$\rho_j(\overline{\mathbf{x}}, \overline{\mathbf{\Delta t}})$	Measurement contribution function
$\mathrm{d}\mu_{\text{2D}}(\overline{\mathbf{x}})$	Spatial integration at each path vertex
$\mathrm{d}\mu(\overline{\mathbf{\Delta t}})$	Temporal integration at each path vertex
ω	3D direction or solid angle
θ	2D direction or planar angle
$\vec{n}$	Normal direction to a point in a surface or curve
$\mathrm{d}a$	Differential area at a point in 3D
$\mathrm{d}l$	Differential arc-length at a point in 2D
$L_{3D}(\mathbf{x}, \omega)$	Radiance in 3D arriving at or leaving point $\mathbf{x}$ in direction ω
$L_{2D}(\mathbf{x}, \theta)$	Radiance in 2D arriving at or leaving point $\mathbf{x}$ in direction θ
$L_e(\mathbf{x}_0, \mathbf{x}_1, \mathbf{\Delta} t_0)$	Emitted radiance from $\mathbf{x}_0$ toward $\mathbf{x}_1$ at time $\mathbf{\Delta} t_0$
$W_e^{(j)}(\mathbf{x}_{m-1}, \mathbf{x}_m, \mathbf{\Delta} t_m)$	Temporal sensor importance at pixel j
$f(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{\Delta} t_i)$	Scattering coefficient at $\mathbf{x}_i$ after a time delay $\mathbf{\Delta} t_i$
$f(\overline{\mathbf{x}})$	Total scattering attenuation at path $\overline{\mathbf{x}}$ vertices
$D(\theta), s$	Microfacet distribution and roughness
$G_{\text{2D}}(\mathbf{x}_i, \mathbf{x}_{i+1})$	Geometric term in 2D, attenuation between any pair of path vertices
$G_{\text{2D}}(\overline{\mathbf{x}})$	Total geometric attenuation of path $\overline{\mathbf{x}}$
$H(\mathbf{x}_l, \mathbf{x}_s, t)$	Impulse response function

S.II. Implementation Details

We implement our end-to-end pipeline using the WebGL API [1], relying on vertex and fragment shaders and storing intermediate and final results in textures for efficient time-resolved light transport simulation and NLOS imaging in 2D scenes.

Simulation of 2D Transient Light Transport: We implement transient light transport in 2D by extending the WebGL-based rendering engine Tantalum [2]. Tantalum simulates conventional steady-state light transport in 2D scenes using light tracing, assuming infinite speed of light and displaying conventional fluence images of 2D scenes without any temporal dependency. Based on our 2D transient path integral formulation, we extend Tantalum to support time-gated emission and capture of irradiance values in 2D scenes. For this purpose, we incorporate temporal dependency of light propagation during the light tracing process by accounting for the optical distances and refractive indices of the media traversed by the light paths to compute the time-of-flight of light. Specifically, we leverage the fourth channel of the radiance textures to store the time-of-flight light samples.

Replicating the NLOS capture process: Real NLOS imaging capture setups implement SPAD sensors focused at a finite region on the relay wall to gather indirect photons from the hidden scene. We add support for this process by implementing a new kernel for computing $\mathbf{H}$ from the generated light samples. This kernel gathers sampled paths in the vicinity of a captured point $\mathbf{x}_s$ and connects deterministic rays to the sensor origin $\mathbf{x}_{s0}$. While SPAD-based NLOS imaging setups typically return a list of individual photons and their corresponding time of flight, it is common practice to accumulate those in a temporal histogram for each captured point $\mathbf{x}_s$, i.e. $\mathbf{H}$, where each temporal bin accumulates photons with a time of flight within a specific time interval based on a user-defined sensor temporal resolution Δt_e. To mimic this process, we accumulate the resulting light samples of our simulation in their corresponding temporal bins—represented by a texture—based on their time of flight and Δt_e. Similarly to Tantalum, we configure NLOS scenes using shaders that contain a list of geometric primitives and their materials. To facilitate the process of scene configuration, we implement a visual interface to create new scenes and modify existing ones, which automates the generation of scene shaders.

NLOS imaging: Based on the scene geometry and parameters defined by the user, we implement real-time NLOS imaging by feeding our implementation of different NLOS imaging models with textures that accumulate small batches

TABLE III
SYMBOLS RELATED TO NLOS IMAGING.

Symbol	Meaning
$\mathcal{L}$	Set of points illuminated in the relay wall
$\mathcal{S}$	Set of points captured in the relay wall
$\mathcal{V}$	Hidden scene volume
$\mathbf{x}_l \in \mathcal{L}$	A single illuminated point in the relay wall
$\tilde{\mathbf{x}}_l$	Virtual light source
$\mathbf{x}_s \in \mathcal{S}$	A single captured point in the relay wall
$\mathbf{x}_v \in \mathcal{V}$	A single point in the hidden scene
$\mathbf{x}_{l0}, \mathbf{x}_{s0}$	Location of the laser and sensor devices, respectively
$\overline{\mathbf{x}}$	Three-bounce light path between a light source and a camera sensor
Δx	Lateral resolution of captured points
t	Time
Δt_e	Sensor temporal resolution
T_{max}	Sensor temporal range
N_l, N_s	Number of illuminated and captured points, respectively
N_t	Number of temporal bins
$\mathcal{F}_t$	Fourier transform
Ω	Imaging frequency
$K(t)$	Pulsed function in the temporal domain
$\Omega_c, \lambda_c, \sigma$	Carrier frequency, central wavelength and width of a pulsed virtual illumination function with a Gaussian envelope
$H(\mathbf{x}_l, \mathbf{x}_s, t)$	Impulse response function
$\mathbf{H}$	Discretized approximation of $H(\mathbf{x}_l, \mathbf{x}_s, t)$
$H_l(\tilde{\mathbf{x}}_l, \mathbf{x}_s, t), \mathbf{H_l}$	Impulse response under virtual illumination emitted from $\tilde{\mathbf{x}}_l$
$H_K(\mathbf{x}_l, \mathbf{x}_s, t), \mathbf{H_K}$	Impulse response under an arbitrary virtual illumination function
$\widehat{H}(\mathbf{x}_l, \mathbf{x}_s, \Omega)$	Frequency-domain impulse response function
$f(\mathbf{x}_v)$	Albedo of diffuse surfaces in the hidden scene
$f(\mathbf{x}_v, t)$	Time-resolved image of the hidden scene
$\widehat{f}(\mathbf{x}_v, \Omega)$	Set of phasors that represents an image of the hidden scene at imaging frequencies Ω
$\mathbf{f}$	Discretized approximation of $f(\mathbf{x}_v, t)$
$f_{tc}(\mathbf{x}_v, t), \mathbf{f}_{tc}$	Image obtained by the time-gated camera
$f_{cc}(\mathbf{x}_v, t), \mathbf{f}_{cc}$	Image obtained by the confocal camera
$f_{sc}(\mathbf{x}_v), \mathbf{f}_{sc}(\mathbf{x}_v)$	Image obtained by the steady-state camera
j_t	Discrete time index of $\mathbf{f}$
$G(\overline{\mathbf{x}})$	Geometric attenuation of path $\overline{\mathbf{x}}$
Δt_v	Time of flight of path $\overline{\mathbf{x}}$
d_{sv}, t_{sv}	Distance from $\mathbf{x}_v$ to $\mathbf{x}_s$ and time to travel it in free space, respectively
d_{lv}, t_{lv}	Distance from $\mathbf{x}_l$ to $\mathbf{x}_v$ and time to travel it in free space, respectively
$\mathcal{P}(\mathbf{x}_l, t)$	Virtual time-resolved illumination function
$\mathcal{P}(\mathbf{x}_s, t)$	Virtual response of the scene
$\widehat{\mathcal{P}}(\mathbf{x}_l, \Omega), \widehat{\mathcal{P}}(\mathbf{x}_s, \Omega)$	Virtual illumination and response phasors at frequency Ω, respectively
Φ	Wave-based imaging operator

of a few tens to hundreds of transient light transport samples. We compute each pixel of $\mathcal{V}$ in parallel according to Eq. (11) or Eq. (15), depending on the chosen NLOS imaging camera model. For the steady-state camera model, we evaluate Eq. (11) for all possible j_t and accumulate all temporal bins. For the filtering step, we implement three different convolution shaders. Laplacian filtering is implemented as a 2D convolution of a texture with a 3×3 Laplacian kernel. Laplacian of Gaussian (LoG) kernels are usually larger, so we perform the 2D convolution between two textures cntaining $\mathbf{H}$ and the LoG kernel, respectively. For phasor-based virtual illumination we implement a 1D convolution between the texture storing $\mathbf{H}$, and a texture that stores the virtual illumination function, which is one dimensional and complex valued. To provide an intuitive visualization of the absolute monochromatic intensity of the resulting reconstructions, we apply a high contrast color map to the absolute intensity values of the imaging output, and store it in a texture which is finally shown to the user. The resulting reconstruction can be saved as floating-point values without loss of precision for additional analysis.

The final algorithm coupling 2D light transport simulation and NLOS imaging is summarized in Algorithm 1, where **gpuParallelFor** stands for a shader call, and captureSample$(\mathbf{x}_i, \theta_i)$ accumulates light samples in our impulse response $\mathbf{H}$ when $\mathbf{x}_i$ is in the chosen vicinity of one of the scanned points $\mathbf{x}_s$.

S.III. SENSOR NOISE

NLOS imaging with real-world captured data is usually photon-starved and suffers from Poisson noise and time jitter introduced by SPAD sensors [3], [4]. Simulation often overlooks these factors, making it difficult to match simulated results to capture conditions. We mimic capture conditions of real setups by post processing our simulated data. Following common practice, we simulate time jitter by convolving the temporal domain of our simulated impulse response function with a Gaussian pulse, adjusting its Full Width at Half Maximum (FWHM). We then apply Poisson noise to the convolved signal assuming the entire sensor domain gathers a total M_p photons. In Fig. 13 we compare the signal (top row) and resulting images (bottom row) for a reference simulation of the BUNNY scene from Fig. 6 with no jitter or capture noise (first column), including only time jitter with FWHM $= 160$ ps (second column), adding low capture noise ($M_p = 500000$, third column), and adding high capture noise ($M_p = 50000$, fourth column). Including time jitter (second column) produces ambiguities when estimating the position of the surfaces along the axis coaxial to the relay wall (horizontal dimension in the image), losing sharpness in the final image. The effect of introducing Poisson noise (third and fourth columns) is more noticeable: the signal is noticeably degraded, producing noisier results, and showing difficulty in recovering the shape of the hidden object for low photon counts (e.g., $M_P = 50000$, fourth column). This replicates results obtained in real captures, which usually need long capture times to gather a sufficient number of photons. Incorporating these factors into the simulation may help researchers establish a lower threshold of photon counts to estimate optimal capture times.

Algorithm 1 Simulation and imaging algorithm in our system

Require: N: number of paths
Require: M: batch size
Require: m: path length
Require: $filter$: filtering strategy
Require: $\mathbf{K}$: filtering kernel
Require: N_v: number of pixels in virtual image
Require: j_t
  for $i \leftarrow 1$ **to** N/M **do**
    // Light tracing
    gpuParallelFor(1 **to** M)**:** $\mathbf{x}_i, \theta_i \leftarrow \text{sample}L_e()$
    for $j \leftarrow 1$ to m **do**
      gpuParallelFor(1 **to** M)**:**
        $\mathbf{x}'_i \leftarrow \text{raytrace}(\mathbf{x}_i, \theta_i)$
        $\theta'_i \leftarrow \text{sampleBrdf}(\mathbf{x}'_i, \theta_i)$
      gpuParallelFor(1 **to** M)**:** drawLine($\mathbf{x}_i, \mathbf{x}'_i$)
      gpuParallelFor(1 **to** M)**:** captureSample($\mathbf{x}_i, \theta_i$)
      $\mathbf{x}_i, \theta_i \leftarrow \mathbf{x}'_i, \theta'_i$
    end for
    // NLOS imaging
    if $filter$ is phasor-based virtual illumination **then**
      $\mathbf{H}' \leftarrow \text{convolution1d}(\mathbf{H}, \mathbf{K})$
    else
      $\mathbf{H}' \leftarrow \mathbf{H}$
    end if
    gpuParallelFor($\mathbf{x}_v \in \mathcal{V}$)**:**
      $\mathbf{f}[\mathbf{x}_v] \leftarrow 0$
      for $\mathbf{x}_s \in \mathcal{S}$ **do**
        $d = \text{distance}(\mathbf{x}_{l0} \rightarrow \mathbf{x}_l \rightarrow \mathbf{x}_v \rightarrow \mathbf{x}_s \rightarrow \mathbf{x}_{s0})$
        $t = d/c + j_t$
        $\mathbf{f}[\mathbf{x}_v] \leftarrow \mathbf{f}[\mathbf{x}_v] + \mathbf{H}'[\mathbf{x}_l, \mathbf{x}_s, t]$
      end for
    if $filter$ is Laplacian or LoG **then**
      $\mathbf{f} \leftarrow \text{convolution2d}(\mathbf{f}, \mathbf{K})$
    end if
  end for

S.IV. Exhaustive Capture Setups

The development of SPAD arrays facilitates simultaneous capture of multiple sensor positions on the relay wall, dramatically improving capture SNR at low exposure times [4]–[6], or enabling exhaustive capture setups where multiple sensor pixels $\mathbf{x}_s$ capture the response of multiple illuminated points $\mathbf{x}_l$ for more complex imaging modalities [7]. However, exhaustive capture and simulation in 3D scenes leads to a dramatic increase of the memory requirements, as the size of the impulse response function grows with the fourth power of the capture spatial resolution ($N_l \cdot N_s$, with both growing quadratically). Existing works proved the applicability of exhaustive captures with lateral resolutions of up to 32 illuminated and captured points [7]. Our 2D pipeline becomes fundamental to analyze the performance of exhaustive captures with higher resolution, as dimensionality reduction provides a tractable memory footprint. We include proof-of-concept results obtained with a 2D exhaustive capture, by capturing and illuminating the same 128 points for both sensor and laser, shown in Fig. 14, where we use the BUNNY scene described in Fig. 6. In contrast to single-point illumination at the center of the relay segment $\mathbf{x}_l$ (first image), exhaustive captures (second image) reveal the shape of the ear and head of the bunny.

References

[1] T. K. G. Inc. Webgl. [Online]. Available: https://www.khronos.org/api/webgl

[2] B. Bitterli. (2015) The secret life of photons. [Online]. Available: https://benedikt-bitterli.me/tantalum/

[3] Q. Hernandez, D. Gutierrez, and A. Jarabo, "A computational model of a single-photon avalanche diode sensor for transient imaging," 2017.

[4] M. Renna, J. H. Nam, M. Buttafava, F. Villa, A. Velten, and A. Tosi, "Fast-gated 16×1 spad array for non-line-of-sight imaging applications," *Instruments*, vol. 4, no. 2, p. 14, 2020.

[5] S. Riccardo, E. Conca, V. Sesta, A. Velten, and A. Tosi, "Fast-gated 16× 16 spad array with 16 on-chip 6 ps time-to-digital converters for non-line-of-sight imaging," *IEEE Sensors Journal*, vol. 22, no. 17, pp. 16 874–16 885, 2022.

[6] C. Jin, M. Tang, L. Jia, X. Tian, J. Yang, K. Qiao, and S. Zhang, "Scannerless non-line-of-sight three dimensional imaging with a 32x32 spad array," 2020.

[7] J. Marco, A. Jarabo, J. H. Nam, X. Liu, M. A. Cosculluela, A. Velten, and D. Gutierrez, "Virtual light transport matrices for non-line-of-sight imaging," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 2440–2449.