Non-line-of-sight Surface Reconstruction
Using the Directional Light-cone Transform

Sean I. Young
Stanford University
sean0@stanford.edu

David B. Lindell
Stanford University
lindell@stanford.edu

Bernd Girod
Stanford University
bgirod@stanford.edu

David Taubman
UNSW Sydney
d.taubman@unsw.edu.au

Gordon Wetzstein
Stanford University
gordon.wetzstein@stanford.edu

Abstract

We propose a joint albedo–normal approach to non-line-of-sight (NLOS) surface reconstruction using the directional light-cone transform (D-LCT). While current NLOS imaging methods reconstruct either the albedo or surface normals of the hidden scene, the two quantities provide complementary information of the scene, so an efficient method to estimate both simultaneously is desirable. We formulate the recovery of the two quantities as a vector deconvolution problem, and solve it using the Cholesky–Wiener decomposition. We show that surfaces fitted non-parametrically using our recovered normals are more accurate than those produced with NLOS surface reconstruction methods recently proposed, and are 1,000\(\times\) faster to compute than using inverse rendering.

1. Introduction

Non-line-of-sight (NLOS) imaging deals with the capture and rendering of a scene that is not in the direct line of sight from the sensor. In recent years, NLOS imaging has emerged as an important vision problem, with applications in remote sensing, defense, robotic vision and autonomous driving. A common imaging setup is to “look around the corner” using the confocal configuration depicted in Figure 1. Typically, a light source, such as a laser beam, indirectly illuminates the scene by reflecting onto a surface that can be seen from both the sensor and the scene. The sensor then captures the scene reflections from the same surface location, and records them as a time-resolved sequence of two-dimensional images (or transients), from which one can computationally reconstruct the scene. Apart from transient-based imaging, other NLOS imaging modalities include those based on speckle [1–3], or incoherent intensity measurements [4, 5], as well as passive sensing [6–9] and acoustic imaging [10] techniques.

Here, we will consider exclusively NLOS imaging based on transients [11–19], from which the hidden NLOS scene is typically rendered as a spatial, three-dimensional volume of albedo (volumetric albedo), or as a set of object surfaces. In the volumetric albedo paradigm, the objective is to estimate albedo values for scene voxels [11–16], while in the surface reconstruction paradigm, one seeks more directly to recover object surfaces in the three-dimensional scene by estimating their surface normals [17–19]. While surface-based methods have the potential to reconstruct object geometry with finer detail than the albedo ones, current approaches to estimating surface normals are sensitive to noise, limited to scenes with simpler object geometry [18], sensitive to initialization [19] or entail a high computational complexity [13], all of which motivate our present work. Embedded in transients is a mix of surface normal and albedo information, so that explicitly accounting for the presence of both in transients can present new opportunities for recovering both quantities robustly.
In this work, we propose a joint albedo–normal approach to NLOS scene reconstruction based on the directional light-cone transform (D-LCT), which is a vectorial generalization of the (scalar) light-cone transform (LCT) recently proposed by O’Toole et al. [12]. We formulate the recovery of normals and albedos as a vector deconvolution problem, and having found the two quantities, we fit a surface onto the recovered normals. By solving this vectorial deconvolution problem in the joint albedo–normal space, we recover the two quantities robustly and efficiently, allowing us to obtain a better surface reconstruction. In particular, our approach is many orders of magnitude faster than recent work [13], which can similarly estimate albedos and surface normals in a joint manner. Our proposed approach also works with existing confocal NLOS imaging hardware. We illustrate our overall method pipeline in Figure 2. To summarize, our main contributions are:

- **Directional LCT**: We express non-line-of-sight surface normal recovery as a vector deconvolution problem on time-resolved measurements, via the Directional Light-cone Transform (D-LCT).

- **Cholesky–Wiener Solver**: We solve the above vectorial deconvolution problem efficiently in the Fourier domain to recover the surface normals.

- **Surface Fitting**: We reconstruct highly-accurate object surface descriptions by fitting surface parameters on the recovered normals.

2. **Related Work**

Transient imaging was first conceptualized by Kirmani et al. [4], who believed that we can look around the corner by probing a wall with an ultrafast laser and detector. Later, the idea was demonstrated in practice by Velten et al. [11] using a femtosecond laser and a streak camera. Owing to the cost of such imaging hardware, researchers have also used other sensing technologies such as time-of-flight cameras [20, 21] or even regular consumer cameras [9, 22]. However, single-photon avalanche diodes (SPADs) [23–27] have been shown to be particularly versatile for sensing, allowing us to image under ambient lighting, at fast rates [28], or at long distances [29]. Altmann et al. [30] provide a comprehensive review of SPADs as well as their applications.

Having captured the transient data, the NLOS scene can be reconstructed as a volume of albedo [11–16], or surfaces of objects [17–19]. Generally speaking, estimation of albedo can be posed as an inverse-filtering problem whereas that of surfaces is often posed as an inverse-rendering problem. We now provide a brief review of the two inverse approaches.

2.1. **Inverse Filtering Approaches**

Velten et al. [11] are the first to pose the recovery of the albedo volume as an inverse problem. Relating the transient measurements to some NLOS scene using higher-order light transport, they formulated scene reconstruction as a (linear) least-squares problem and solved the resulting dense system of equations with filtered back-projection (FBP). While FBP produces promising results, it only approximately solves the original least-squares problem, and the reconstructed scenes can lack fine details. Since the computational complexity of FBP is still high at $O(V^3)$ in the number $V^3$ of voxels, later authors sought to refine FBP by improving the quality of the iterative solvers [16, 31] or their speed on GPUs [26].

In the confocal case, O’Toole et al. [12] note that higher-order light transport can be expressed as a convolution with a change of variables. Their overall transformation, referred to as the light-cone transform (LCT), expresses the problem of Velten et al. [11] as a three-dimensional signal deblurring problem. In contrast to the FBP, the LCT solves the inverse problem exactly, and has a low computational complexity of $O(V^3 \log V)$ in the number $V^3$ of voxels, thanks to the use of the Fourier transform. Recently, Ahn et al. [32] proposed an approximate convolutional imaging transform similar to the LCT for the non-confocal setting.

In contrast with deconvolution, which we can ultimately relate to the diffusion equation, Lindell et al. [14] suggested...
to solve the transient imaging problem by modeling higher-order light transport as wave propagation in a 3-dimensional space, and solve the resulting inverse problem efficiently in the Fourier domain using “f–k” migration. This method also has a $O(V^3 \log V)$ computational complexity due to the use of the Fourier transform. Methods based on diffractive wave propagation, e.g. phasor fields [33], aim further to overcome the limitations in the imaging model due to assumptions such as single scattering, and the lack of occlusions in the hidden scene. If the confocal setup is used, the phasor field method can be implemented in terms of the LCT for a $O(V^3 \log V)$ computational complexity.

Whereas both the LCT and the f–k migration approaches are extremely efficient, they do not innately have the ability to estimate the surface normals of scene objects. To find the surface normals along with the albedo volume, Heide et al. [13] pose the transient imaging problem as an optimization in the albedo and the surface normal variables. Although the reported results are promising, one major limitation of such a method is the $O(V^5)$ complexity in both computation and memory, as well as the nonconvexity of the overall problem formulation. Our D-LCT facilitates solving a similar albedo-normal estimation problem with the same low complexity of the LCT and f–k migration. Given the shared Fourier roots across the LCT, f–k and phasor fields, it may be possible to apply our directional approach to f–k migration and phasor fields as well, although we do not attempt this in our work.

2.2. Inverse Rendering Approaches

In contrast with the inverse filtering approaches, inverse rendering (analysis-by-synthesis) methods, e.g. [19], search for values of the NLOS surface parameters (e.g. BRDFs and surface normals) that would produce the observed transients if the NLOS surface were to be rendered. Since a full search in the surface parameter space would be intractable, inverse rendering is typically performed via differentiable rendering (i.e., an energy minimization in surface parameters). Surface parameters recoverable in this way include surface locations and normals [34], and illumination effects such as scattering [35–38] and interreflections [39, 40].

2.3. Surface Fitting Methods

Object surface can be fit on the obtained surface normals either parametrically or non-parametrically. Non-parametric fitting is commonly used for stereo-based 3D reconstruction [41–43] whereas parametric approaches are more suited for mesh refinement [44–47]. Surface fitting may be seen as an inverse problem where the goal is to interpolate a smooth manifold without violating the given normal conditions. We can naturally formulate surface fitting as diffusion processes or energy minimization methods, both of which seek to find the right trade-off between regularity (smoothness) and data fidelity (surface orthogonality to the given normals). We use the energy-minimization approach of [48] to fit a surface on our recovered normals.

3. Mathematical Framework

After briefly reviewing the volumetric albedo model and discussing its limitations, we develop our directional albedo model, and propose efficient ways for solving the associated inverse problem of estimating the surface normals.

3.1. The Volumetric Albedo Model

In transient imaging approaches, a time-resolved detector is used to measure the incident flux of photons as a function of emitted light impulses. Each of these time measurements records the impulse response of the NLOS scene at positions on a visible surface to produce a volume of transients.

Let us denote the three-dimensional scene coordinates by $(x, y, z)$, and assume the visible surface is positioned along $z = 0$. We denote by $(x', y', z = 0)$ positions on this visible surface; see Figure 1. A common transient imaging model is the confocal volumetric albedo model

$$\tau(x', y', t) = \iiint_{\Omega} dx dy dz \frac{\rho(x, y, z)}{r^4} \delta \left(2(2-r_i^2-x^2+y^2)^2+z^2-tc\right),$$

in which $\rho$ denotes a three-dimensional albedo volume with finite support $\Omega$, and $\delta(\cdot)$ relates the round-trip time of flight of light with twice the distance $r$ between the scene $(x, y, z)$ and the sensing $(x', y', z = 0)$ locations. Here, $c \approx 3 \times 10^8$ denotes the speed of light while $1/r^4 = (2/ct)^4$ models the radiometric fall-off due to distance. The scaling $1/r^4$ can be removed from (1) if we prescale $\tau$ by $(2/ct)^4$ in advance. In the case of retro-reflective surfaces, a fall-off factor of $1/r^2$ is more commonly assumed.

To discretize model (1), we sample $\Omega$ using $N, N$ and $M$ points on the $x$-, $y$- and $z$-axes, respectively. Assuming that the transient $\tau$ has been pre-scaled by $(2/ct)^4$, we can write the discretized model compactly using matrix notation as

$$\mathbf{\tau} = \mathbf{K}\mathbf{\rho},$$

in which $\mathbf{\tau}, \mathbf{\rho} \in \mathbb{R}^{NM \times 4M}$ and $\mathbf{K}$ is a binary matrix with values obtained by sampling $\delta(\cdot)$. Since $\mathbf{K}$ is a low-pass operator of high condition number, the task of finding $\mathbf{\rho}$ from given $\mathbf{\tau}$ is an ill-posed problem [49]. Rather than compute the solution directly as $\mathbf{\rho}^{\text{opt}} = \mathbf{K}^{-1}\mathbf{\tau}$, we should find it as the solution of the regularized least-squares problem

$$\text{minimize } f(\mathbf{\rho}) = ||\mathbf{K}\mathbf{\rho} - \mathbf{\tau}||_2^2 + \lambda||\mathbf{\rho}||_2^2,$$

in which $\lambda$ represents the trade-off between data fidelity and regularity (smoothness) of the solution. O’Toole et al. [12] note that problem (3) can be solved in an efficient manner with their so-called light-cone transform (LCT). If we denote their resampling operator by $\mathbf{T}$, we can
In the case of isotropic point emitters, (1) is an adequate model for higher-order light transport. However, for typical diffuse or Lambertian object surfaces, such a model ignores the radiometric fall-off due to Lambert’s cosine law, i.e., the fall-off due to the angle between the incident light rays and the surface normals; see [50]. Incorporating cosine terms in (1) not only yields a more accurate forward model, but more importantly, it enables recovery of surface normals from the transients via the inverse model.

Denoting the two spatial coordinates by \( s = (x, y, z) \) on the object, directional albedo \( \psi(s) \) has direction and magnitude of the normal \( \mathbf{n}(s) \) and the albedo \( \rho(s) \), respectively. Contribution of albedo \( \rho(s) \) to the surface at \( s' = (x', y', z = 0) \) decreases, in the first order, as the cosine of the angle \( \theta \) between \( \mathbf{n}(s) \) and \( s' - s \).

Express problem (3) equivalently in terms of the resampled albedo \( \tilde{\rho} = T^*\rho \) and transients \( \tilde{\tau} = T^*\tau \) as

\[
\text{minimize } f(\tilde{\rho}) = \|H\tilde{\rho} - \tilde{\tau}\|_2^2 + \lambda\|\tilde{\rho}\|_2^2,
\]

in which \( H = T^*K^*T \) is a three-dimensional filter (that is, a linear space-invariant operator), whose impulse response is shown in Figure 3 (a). Section 1 of the supplement provides the details of the resampling operator \( T^* \).

Since \( H \) is a three-dimensional filter, we can compute the solution \( \tilde{\rho}^\text{opt} = (H^*H + \lambda I)^{-1}H^*\tilde{\tau} \) of (4) efficiently in the Fourier domain using Wiener deconvolution. The regularity parameter \( \lambda \) can be interpreted as the noise-to-signal ratio in this filtering context. The solution of our original problem (3) is obtained by resampling the deconvolved solution using the adjoint resampling operator, that is, \( \rho^\text{opt} = T\tilde{\rho}^\text{opt} \).

### 3.2. The Directional Albedo Model

In the case of isotropic point emitters, (1) is an adequate model for higher-order light transport. However, for typical diffuse or Lambertian object surfaces, such a model ignores the radiometric fall-off due to Lambert’s cosine law, i.e., the fall-off due to the angle between the incident light rays and the surface normals; see [50]. Incorporating cosine terms in (1) not only yields a more accurate forward model, but more importantly, it enables recovery of surface normals from the transients via the inverse model.

Denoting the two spatial coordinates by \( s = (x, y, z) \) on the object, directional albedo \( \psi(s) \) has direction and magnitude of the normal \( \mathbf{n}(s) \) and the albedo \( \rho(s) \), respectively. Contribution of albedo \( \rho(s) \) to the surface at \( s' = (x', y', z = 0) \) decreases, in the first order, as the cosine of the angle \( \theta \) between \( \mathbf{n}(s) \) and \( s' - s \).

Denoting the two spatial coordinates by \( s = (x, y, z) \) on the object, directional albedo \( \psi(s) \) has direction and magnitude of the normal \( \mathbf{n}(s) \) and the albedo \( \rho(s) \), respectively. Contribution of albedo \( \rho(s) \) to the surface at \( s' = (x', y', z = 0) \) decreases, in the first order, as the cosine of the angle \( \theta \) between \( \mathbf{n}(s) \) and \( s' - s \).

Express problem (3) equivalently in terms of the resampled albedo \( \tilde{\rho} = T^*\rho \) and transients \( \tilde{\tau} = T^*\tau \) as

\[
\text{minimize } f(\tilde{\rho}) = \|H\tilde{\rho} - \tilde{\tau}\|_2^2 + \lambda\|\tilde{\rho}\|_2^2,
\]

in which \( H = T^*K^*T \) is a three-dimensional filter (that is, a linear space-invariant operator), whose impulse response is shown in Figure 3 (a). Section 1 of the supplement provides the details of the resampling operator \( T^* \).

Since \( H \) is a three-dimensional filter, we can compute the solution \( \tilde{\rho}^\text{opt} = (H^*H + \lambda I)^{-1}H^*\tilde{\tau} \) of (4) efficiently in the Fourier domain using Wiener deconvolution. The regularity parameter \( \lambda \) can be interpreted as the noise-to-signal ratio in this filtering context. The solution of our original problem (3) is obtained by resampling the deconvolved solution using the adjoint resampling operator, that is, \( \rho^\text{opt} = T\tilde{\rho}^\text{opt} \).

### 3.2. The Directional Albedo Model

In the case of isotropic point emitters, (1) is an adequate model for higher-order light transport. However, for typical diffuse or Lambertian object surfaces, such a model ignores the radiometric fall-off due to Lambert’s cosine law, i.e., the fall-off due to the angle between the incident light rays and the surface normals; see [50]. Incorporating cosine terms in (1) not only yields a more accurate forward model, but more importantly, it enables recovery of surface normals from the transients via the inverse model.

Denoting the two spatial coordinates by \( s = (x, y, z) \) on the object, directional albedo \( \psi(s) \) has direction and magnitude of the normal \( \mathbf{n}(s) \) and the albedo \( \rho(s) \), respectively. Contribution of albedo \( \rho(s) \) to the surface at \( s' = (x', y', z = 0) \) decreases, in the first order, as the cosine of the angle \( \theta \) between \( \mathbf{n}(s) \) and \( s' - s \).
Figure 5. **Transient imaging using the D-LCT**: D-LCT (c) captures fine details of object surfaces (a) not captured by the LCT (b). Volumes (b)–(c) are rendered using maximum intensity projection. D-LCT surfaces (d) are fit directly onto the D-LCT normals (c). For (d), we used known background masks to first remove the background points. All hidden objects have diffuse surfaces.

To discretize our directional albedo model, we sample Ω similarly to (2). This produces the system of linear equations in our directional albedos \( \hat{\mathbf{v}} = (\hat{v}_x^*, \hat{v}_y^*, \hat{v}_z^*)^\top \):

\[
\tau = \mathbf{K} \hat{\mathbf{v}} \tag{9}
\]

in which \( \mathbf{K} \) is the matrix from (2), and we obtain the entries of \( \mathbf{S} = (S_x, S_y, S_z) \) by sampling \((s' - s)\) on \( \Omega \). Recovering the directional albedo \( \hat{\mathbf{v}} \) given the transients \( \tau \) is ill-posed in that it requires us to find the values of \( 3N^2M \) variables with only \( N^2M \) equations. Such a rank-deficient problem can be solved by formulating (9) as the least-squares problem

\[
\min_{\hat{\mathbf{v}}} f(\hat{\mathbf{v}}) = \| \mathbf{K} \hat{\mathbf{v}} - \tau \|_2^2 + \lambda \| \hat{\mathbf{v}} \|_2^2 \tag{10}
\]

similarly to the regularized approach in (3). While we could alternatively optimize the TV-L1 version of problem (10) to obtain a better solution, we focus on our L2 variant for now to solve (10) efficiently as a vector deconvolution problem.

### 3.3. Directional Light-cone Transform

Whereas our least-squares optimization problem (10) has the simple, closed-form solution

\[
\hat{\mathbf{v}}^{\text{opt}} = (\mathbf{S}^\top \mathbf{K} \mathbf{S} + \lambda \mathbf{I})^{-1} \mathbf{S}^\top \mathbf{K} \tau, \tag{11}
\]

this solution is too expensive to compute naively for typical problems with \( V^3 = N^2M \approx 10^8 \) voxels. Computing \( \hat{\mathbf{v}}^{\text{opt}} \) directly with numerical methods such as Cholesky and LDL decompositions would incur a \( O(V^3) \) cost whereas iterative ones (e.g. conjugate gradients), a \( O(V^6) \) cost. These general methods are therefore unsuited to practical problem sizes.

To solve problem (10) efficiently, we generalize the LCT technique used in (4) to the vectorial problem. We can write problem (10) equivalently as

\[
\min_{\hat{\mathbf{v}}} f(\hat{\mathbf{v}}) = \| \mathbf{H}(\mathbf{S}_x, \mathbf{S}_y, \mathbf{I}) \hat{\mathbf{v}} - \tilde{\tau} \|_2^2 + \lambda \| \hat{\mathbf{v}} \|_2^2, \tag{12}
\]

in which

\[
\begin{pmatrix}
\tilde{v}_x \\
\tilde{v}_y \\
\tilde{v}_z
\end{pmatrix} =

\begin{pmatrix}
\mathbf{T}_x & \mathbf{T}_y & \mathbf{T}_z \\
\mathbf{T}_x^* & \mathbf{T}_y^* & \mathbf{T}_z^*
\end{pmatrix}

\begin{pmatrix}
\hat{v}_x \\
\hat{v}_y \\
\hat{v}_z
\end{pmatrix}
\]

(13)

is the resampled variable, and \( \tilde{\tau} = \mathbf{T}^* \tau \) as before. From the solution \( \hat{\mathbf{v}}^{\text{opt}} \) of (12), we recover the solution of the original problem (10) as \( \hat{\mathbf{v}}^{\text{opt}} = \mathbf{T}_d \hat{\mathbf{v}}^{\text{opt}} \). Section 2 of the supplement derives the resampler \( \mathbf{T}_d \) and its relationship to \( \mathbf{T}^* \) in (4).

Note in (12) that \( \mathbf{S}_x, \mathbf{S}_y, \mathbf{I} \) are shift-invariant, so they can be composed with the light-cone filter \( \mathbf{H} \) to produce the directional light-cone filters \( \mathbf{H} \mathbf{S}_x, \mathbf{H} \mathbf{S}_y \) and \( \mathbf{H} \mathbf{I} \). Their filter kernels are shown in Figure 3 (b). Since all the operators in problem (12) are filters, one can interpret (12) as a vectorial deconvolution problem, with a fixed noise-to-signal ratio \( \lambda \) across the vector frequencies. One can solve (12) efficiently using our vector extension of Wiener deconvolution.

### 3.4. Cholesky–Wiener Deconvolution

Denoting the matrices in (12) as \( \mathbf{H}_x = \mathbf{H} \mathbf{S}_x, \mathbf{H}_y = \mathbf{H} \mathbf{S}_y \) and \( \mathbf{H}_z = \mathbf{H} \mathbf{I} \) for simplicity, we write the normal equations...
associated with the least-squares problem (12) as
\[\begin{bmatrix}
\mathbf{H}_x^2 + \lambda I & \mathbf{H}_x^h \\
\mathbf{H}_y^h & \mathbf{H}_y^2 + \lambda I
\end{bmatrix}
\begin{bmatrix}
\mathbf{v}_x \\
\mathbf{v}_y
\end{bmatrix}
= \begin{bmatrix}
\mathbf{H}_x^2 \mathbf{T} \\
\mathbf{H}_y^2 \mathbf{T}
\end{bmatrix},
\]
(14)
that is, a $3 \times 3$ block system of equations, where each block element is a filter or a filter signal. The structure of (14) thus suggests that we solve the $3 \times 3$ system using the Cholesky decomposition, performing the two associated forward- and back-substitutions using filtering operations. The right-hand side vector $\mathbf{H}^T \mathbf{T}$ can be computed in the Fourier domain, as each $\mathbf{H}_x$, $\mathbf{H}_y$ and $\mathbf{H}_z$ is a filter.

Using the LDL variant of the Cholesky factorization, we factor the matrix $\mathbf{A} = \mathbf{H}^T \mathbf{H} + \lambda I$ as $\mathbf{A} = \mathbf{LDL}^T$, where
\[
\mathbf{L} = \begin{bmatrix}
\mathbf{I} & & \\
\mathbf{L}_{yx} & \mathbf{I} & \\
\mathbf{L}_{zx} & \mathbf{L}_{zy} & \mathbf{I}
\end{bmatrix},
\]
\[
\mathbf{D} = \begin{bmatrix}
\mathbf{D}_{xx} & & \\
& \mathbf{D}_{yy} & \\
& & \mathbf{D}_{zz}
\end{bmatrix},
\]
(15)
and the elements of $\mathbf{L}$ and $\mathbf{D}$ are given by
\[
\mathbf{D}_{ij} = \mathbf{H}_i^T \mathbf{H}_j + \lambda - \sum_{k=1}^{j-1} \mathbf{L}_{ik} \mathbf{D}_{kk} \mathbf{L}_{jk}^*,
\]
\[
\mathbf{L}_{ij} = \mathbf{D}_{ij}^{-1}(\mathbf{H}_i^T \mathbf{H}_j - \sum_{k=1}^{j-1} \mathbf{L}_{ik} \mathbf{D}_{kk} \mathbf{L}_{jk}^*),
\]
(16)
using the convention $1 = x$, $2 = y$, $3 = z$ in both sums. We can readily verify the dynamic programming procedure (16) by applying the elimination steps of the Cholesky algorithm [51] to the block elements of matrix $\mathbf{A}$.

Finally, the triangularized system $\mathbf{LDL}^T \mathbf{v} = \mathbf{H}^T \mathbf{T}$ can be solved using forward- and back-substitutions
\[
\mathbf{v} = \mathbf{L}^{-1} \mathbf{H}^T \mathbf{T}, \quad \mathbf{v} = \mathbf{L}^{-1} \mathbf{D}^{-1} \mathbf{v},
\]
(17)
both of which can be performed as a series of filtering steps in the Fourier domain. For example, the block-elements of $\mathbf{v}$ can be obtained using $\mathbf{v}_1 = \mathbf{H}_x^T \mathbf{T}$, $\mathbf{v}_2 = \mathbf{H}_y^T \mathbf{T} - \mathbf{L}_{yx} \mathbf{v}_1$ and $\mathbf{v}_3 = \mathbf{H}_z^T \mathbf{T} - \mathbf{L}_{zx} \mathbf{v}_1 - \mathbf{L}_{zy} \mathbf{v}_2$. Observe here $\mathbf{L}_{yx}, \mathbf{L}_{zx}$ and $\mathbf{L}_{zy}$ may be computed in the Fourier domain, as each matrix $\mathbf{L}_{yx}, \mathbf{L}_{zx}$ and $\mathbf{L}_{zy}$ represents a 3D filter. We compute the elements of $\mathbf{v}$ can be computed in a similar manner.

For large problems where storing the Fourier coefficients of the block elements of $\mathbf{L}$ and $\mathbf{D}$ is not feasible, we can use an iterative solver like conjugate gradients, and compute the forward mapping $\mathbf{x} \mapsto (\mathbf{H}^T \mathbf{H} + \lambda I) \mathbf{x}$ using, again, filtering operations in the Fourier domain.

### 3.5. Surface Reconstruction

Having obtained the field $\mathbf{v}$ of directional albedo, we use the method of [52] to fit a nonparametric surface. Fitting the surface amounts to recovering an indicator function $\chi$ of the scene object so that the gradient of $\chi$ equals $\mathbf{v}$. Expressed as an optimization problem, we have
\[
\text{minimize} \ f(\chi) = \| \mathbf{G}^T \mathbf{G} \chi - \mathbf{G}^T \mathbf{v} \|^2 + \lambda \| \chi \|^2,
\]
(18)
in which $\mathbf{G}$ denotes the discretization of the 3-dimensional gradient operator. In practice, however, the zero-level-set of the solution $\chi^{\text{opt}}$ of (18) may deviate from the surface of the true scene object due to the noise in $\mathbf{v}$ and the discretization
To mitigate such issues, we follow again the approach of [52] to extract an isosurface of $\delta_{\text{opt}}$ instead.

### 4. Experimental Results

Since our directional LCT approach recovers both albedo and surface normals, it can be used in the traditional context of two-dimensional NLOS imaging as well as to reconstruct surfaces in three-dimensions. We demonstrate our approach for both use-cases, also comparing it to methods specialized to each one. We use ZNLOS [53] and Stanford [14] datasets for experimental validation. The ZNLOS dataset consists of multiple-bounce transients of synthetic objects 0.50m away from a 1m $\times$ 1m visible surface. The dataset has a temporal resolution of 512 pixels with bins of width 10ps, and spatial resolutions of 256 $\times$ 256 pixels. The Stanford set consists of transients measured on a 2m $\times$ 2m surface of natural hidden objects 1m away, with ambient light and noise. This dataset has a spatial resolution of 512 $\times$ 512 or 64 $\times$ 64 pixels, and a temporal resolution of 512 with bins of width 32ps.

#### 4.1. NLOS Imaging Experiments

**Directional Transient Imaging.** Figure 5 shows the normal images obtained using the D-LCT. These images contain the fine variations in object surfaces such as the smooth surface of the spheres and the fur of the bunny. These details would be difficult to recover post-hoc from albedo-only images using detail-enhancement techniques, for example. The results for the LCT resemble the $z$-component of the D-LCT ones with subtle differences that can be expected from the simplifying assumption (6). We render directional albedo volumes using maximum intensity projection: for each point ($x'$, $y'$, $z = 0$) on the image plane, we find along the $z$-axis the directional-albedo with maximum $z$-component values. We reconstruct the surfaces by first masking out the background pixels with ground-truth masks, and performing Poisson reconstruction on the foreground points and the directional albedo. We use $\lambda = 2^{3\frac{1}{3}}$ for all scenes.

**Accuracy of Depth and Surface Normals.** Figure 6 shows the error maps for the recovered depth and surface normals of the “Bunny”. The recovered LCT and D-LCT depth maps have root-mean-squared errors (RMSE) 5.97cm and 4.96cm.
and the mean absolute errors (MAE) of 1.87cm and 1.59cm respectively, across foreground pixels. The surface normals estimated using the LCT and the D-LCT have the end-point RMSE of 0.91cm and 0.52cm, MAE of 0.61cm and 0.38cm respectively. Note, the LCT does not, by itself, produce any surface normals, so we obtain the normals using the method [54] with the LCT depth as the input (we use 6 neighboring points to produce the optimum results). In Figure 7, we plot the influence of regularization parameter $\lambda$ on the depth and surface normal errors, illustrating that the D-LCT performs stable over a wide range of $\lambda$. This can be useful in practical imaging scenarios in which the signal-to-noise ratio $\lambda$ of the captured transient data is not known exactly.

**Surface Reconstruction with Captured Data.** To show the robustness of the D-LCT against different types of noise that are present in real capture environments, we perform surface reconstruction with the Stanford dataset. Figure 8 shows the directional albedo and surfaces of recovered SU, Discobolus and the Dragon objects. We reconstruct the surfaces by first thresholding the norm of directional albedo vectors to mask out the background points, then performing Poisson surface reconstructions on the remaining foreground points. We use $\lambda = 2^0, 2^2$ and $2^3$ for the three scenes. In the SU scene, the normal volumes reveal the orientation of the letters S and U (S points to the upper-left, U points to the upper-right). The left part of U is partially occluded, so the different methods produce different maximal intensity projections along the z-axis. For these reconstruction tasks with noisy transients, the method of Tsai et al. [19] and Fermat flow [18] reconstruct only the rough shapes of the objects. We initialize [19] using the LCT, but other initializations are also possible. We used the --density flag in the Poisson reconstruction software [48] to avoid the fusion of nearby surface segments.

**Computational Efficiency.** While the D-LCT has the same computational complexity as the LCT, we perform $9 \times$ more computations per voxel due to the outer Cholesky factoring required. The D-LCT is $1000 \times$ faster compared with similar methods that are capable of recovering the surface normals of objects with a complex geometry. While Fermat flow [18] is $4 \times$ faster than our approach, it is applicable mostly to the reconstruction of surfaces of objects with simpler geometry such as a bowl or a sphere (see Section 3 of the supplement for the reconstructions). Table 1 provides the running times of different methods on an 8-core, 2.70GHz CPU.

**Limitations.** Our forward model (8) assumes the scene has mostly non-specular surfaces. Fortunately, our least-squares inverse method provides some degree of robustness against specularities by treating them as outliers (see e.g. the Dragon reconstruction, Figure 8). Similarly, we treat occlusions in the scene as outliers to our least-squares formulation. Using an $L_1$-based data fidelity term instead of our $L_2$-based one (10) could further improve the robustness of our method, at the cost of increased computation times. The $L_1$-based data-term also enforces sparsity, which may remove the need for masking out background pixels.

Our forward model also linearizes the cosine fall-off due to the interaction between surface normals and the two light rays (incident and reflected). Our linearized fall-off model is an under-estimator of the true fall-off, and surface locations that make larger angles on average with the visible wall are estimated to be at positions closer to the visible wall, where the fall-off is indeed less. This causes rounder surfaces to be estimated slightly flatter than they should be (see the arms of the Discobolus, Figure 8), but not as flat as the estimates of the LCT, which assumes zero cosine fall-off. This issue can be overcome by iteratively reweighting the first term of (10) using the ratio of the true fall-off to the linear one, based on the normals last estimated; see Section 3 of the supplement.

**Future Work.** To improve reconstruction times, we plan to implement the D-LCT procedure on a GPU. Similarly to the original LCT, the D-LCT is highly parallelizable and can be significantly accelerated using a GPU implementation. Like the GPU implementation of the LCT, we expect GPU-based D-LCT to require milliseconds of processing time for lower spatial resolutions, e.g. $32 \times 32$ or $64 \times 64$ pixels. We plan also to consider $L_1$ or TV regularizers to better preserve the discontinuities in the reconstructed surfaces.

**6. Conclusion**

NLOS imaging approaches have typically been classified as recovering either the albedo or the surface normals of the hidden objects. In this work, we showed that it is possible to recover both quantities jointly. In closing, reconstruction of surfaces of hidden 3D objects can be regarded as the next frontier for NLOS imaging because it allows us to represent better the 3D environment we ultimately live in. We believe the D-LCT is a big step towards pushing beyond volumetric albedo approaches, providing a practical way to estimate the hidden surface normals needed for surface reconstruction.

**Acknowledgements.** We thank M. J. Galindo for help with dataset [53] and I. Gkioulekas for the code of [18, 19]. D.L. was supported by a Stanford Graduate Fellowship. G.W. was supported by an NSF CAREER Award (IIS 1553333), a Sloan Fellowship, by the KAUST Office of Sponsored Research through the Visual Computing Center CCF grant, the DARPA REVEAL program, and a PECASE by the ARL.
References


