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

Sean I. Young
Stanford University
Bernd Girod
Stanford University

David B. Lindell
Stanford University
David Tauman
UNSW Sydney
Gordon Wetzstein
Stanford University

S1. LCT Resampling

Here, we provide a derivation for the LCT resampler \( T^r \)
introduced in Section 3.1 of the main paper. Recapitulating
that the forward confocal model \([1]\) is given by
\[
\tau(x', y', t) = \iint \mathcal{D} \Omega \frac{\rho(x, y, z)}{\rho^4} \cdot \delta \left( 2\sqrt{(x'-x)^2 + (y'-y)^2 + z^2 - tc} \right), \tag{S1}
\]
the objective of the LCT is to convert (S1) to a convolution
operator different from that used by the LCT. The D-LCT
aims to express the directional-albedo model
in the form of linear integrals (S3).

Applying the LCT (S3) to the three integrals, we obtain
\[
\mathcal{T}^r_x [\tau](x', y', u') = \iint \mathcal{D} \Omega \mathcal{T}^r_x [v_x](x, y, u) h_x(\cdot) \tag{S5}
\]
\[
+ \iint \mathcal{D} \Omega \mathcal{T}^r_y [v_y](x, y, u) h_y(\cdot) \tag{S6}
\]
\[
+ \iint \mathcal{D} \Omega \mathcal{T}^r_z [v_z](x, y, u) h_z(\cdot)
\]

in which \( \delta(\cdot) \) denotes the last term in the integral of (S1).

Applying the LCT (S3) to the three integrals, we obtain
\[
\mathcal{T}^r_x [\tau](x', y', u') = \iint \mathcal{D} \Omega \mathcal{T}^r_x [v_x](x, y, u) h_x(\cdot) \tag{S5}
\]
\[
+ \iint \mathcal{D} \Omega \mathcal{T}^r_y [v_y](x, y, u) h_y(\cdot) \tag{S6}
\]
\[
+ \iint \mathcal{D} \Omega \mathcal{T}^r_z [v_z](x, y, u) h_z(\cdot)
\]

in which the resampling operators
\[
\mathcal{T}^r_x [v_x](x, y, u) = (1/2\sqrt{u})v_x(x, y, \sqrt{u}/2)
\]
\[
\mathcal{T}^r_y [v_y](x, y, u) = (1/2\sqrt{u})v_x(x, y, \sqrt{u}/2)
\]
\[
\mathcal{T}^r_z [v_z](x, y, u) = (1/4)v_z(x, y, \sqrt{u}/2)
\]

and note the \( \mathcal{T}^r_x \) and \( \mathcal{T}^r_y \) are identical to the LCT resampling
operator \( \mathcal{T}^r \), while \( \mathcal{T}^r_z \) differs due to the presence of \( z \) in the
last integral of (S5). We have also denoted by
\[
h_x(\cdot) = h(x'-x, y'-y, u'-u)(x-x')
\]
\[
h_y(\cdot) = h(x'-x, y'-y, u'-u)(y-y') \tag{S8}
\]
\[
h_z(\cdot) = h(x'-x, y'-y, u'-u)
\]

the three, shift-invariant D-LCT operators (filters) in (S6).

S2. D-LCT Resampling

In the case of the D-LCT, the definition of the resampling
operator is different from that used by the LCT. The D-LCT
aims to express the directional-albedo model
\[
\tau(x', y', t) = \iint \mathcal{D} \Omega \frac{\rho(s', s - s)}{\rho^5} \cdot \delta \left( 2\sqrt{(x'-x)^2 + (y'-y)^2 + z^2 - tc} \right), \tag{S4}
\]
as a sum of convolutions. Absorbing \( 1/r^5 = (2/tc)^5 \) in the

definition of \( \tau \) and unwrapping \( \langle v(s), s' - s \rangle \) produces
\[
\tau(x', y', t) = \iint \mathcal{D} \Omega v_x(x, y, z) \delta(x - x') \tag{S5}
\]
\[
+ \iint \mathcal{D} \Omega v_y(x, y, z) \delta(y - y') \tag{S5}
\]
\[
+ \iint \mathcal{D} \Omega v_z(x, y, z) \delta(z)
\]

S3. Model Linearization

Here, we show the directional-albedo model corresponds
to the linearization of a physically-based, higher-order light
transport model. We can express the confocal version of the
physical model proposed by Tsai et al. [2] as
\[
\tau(x', y', t) = \int \int d\Omega \frac{\rho(s)}{r^4} \left( \frac{n(s), s' - s}{|s' - s|} \right)^2 \frac{f(n(s), s' - s)}{f(n(s), s' - s)} \cdot \delta \left( 2\sqrt{(x' - x)^2 + (y' - y)^2 + z^2} - tc \right),
\]  
(S9)
in which the squaring of the inner product takes into account the cosine fall-offs due to both the incident and the reflected rays at \( s = (x, y, z) \).

Due to the squaring of the dot product in (S9), recovering \( n(s) \) requires solving a nonlinear least-squares problem. We can apply the Gauss-Newton method to solve this nonlinear least-squares problem iteratively, by successively linearizing the term \( f(n(s), s' - s) \) as
\[
\hat{f}(n(s), s' - s) = a(s) \left( \frac{n(s), s' - s}{|s' - s|} \right) + b(s),
\]  
(S10)
in which \( a(s) \) and \( b(s) \) are respectively the slope and offset parameters of each linearized model. Observe that both \( n(s) \) and \( (s' - s)/|s' - s| \) are unit vectors, so that linearizations \( \hat{f}(n(s), s' - s) \) depend only on the angle between \( n(s) \) and \( s' - s \) about which the linear approximations are formed.

Our object is to derive a linearized model in terms of the directional albedos \( \nu(s) = \rho(s)n(s) \). Substituting the linear term \( \hat{f}(n(s), s' - s) \) in (S9) and using the bilinearity of dot products, we can write the linearized model associated with the \( k + 1 \)th iteration of Gauss-Newton as \( \tau(x', y', t) \)
\[
\tau(x', y', t) = \int \int d\Omega \left( a_k(s) \left( \frac{n_k(s), s' - s}{|s' - s|} \right) - b_k(s) \rho_k(s) \right) \cdot \delta \left( 2\sqrt{(x' - x)^2 + (y' - y)^2 + z^2} - tc \right) d\Omega,
\]  
(S11)
in which
\[
a_k(s) = 2 \left( \frac{n_k(s), s' - s}{|s' - s|} \right),
\]  
(S12)\[
b_k(s) = - \left( \frac{n_k(s), s' - s}{|s' - s|} \right)^2,
\]
and \( n_k(s) \) and \( \rho_k(s) \) are respectively the normal and albedo at \( s \) estimated during the \( k \)th iteration. Figure S1 (left) plots the linearization of \( f(n(s), s' - s) \) at different values of the incident angle \( w \) between \( n_k(s) \) and \( s' - s \).

In the first iteration (\( k = 0 \)) of Gauss-Newton, we do not have the previous estimate \( \rho_{-1} \) of the albedo. Therefore, we require a linearization \( \hat{f}(n(s), s' - s) \) that does not involve the offset \( b(s) \). Constraining \( b(s) = 0 \) for all \( s \), and using the fact that (S9) is invariant to a constant scaling of the normal \( n(s) \), we can choose
\[
a(s) = 1, \quad b(s) = 0
\]  
(S13)
for all \( s \), from which we obtain (5) and consequently (8). In Figure S1 (right), we plot the offset-constrained linearization \( \hat{f}_0(w) \) of \( f(w) \). Flatness can be more prominent for scenes that are further away from the relay wall since the light rays are almost collinear, providing less directional information in the transients, as seen with Discobolus in Figure S2.

### S4. Thresholding and Masking

Having thus obtained the 3D volume of directional albedo \( \nu = (\nu_x, \nu_y, \nu_z) \in \mathbb{R}^{V^3 \times 3} \), we generate foreground mask as \( \mathbf{m} = \nu_x > \alpha \max(\nu_z) \in \mathbb{R}^{V^3} \). Using the threshold \( \alpha = 0.2 \) gives us good masks in practice. The masked directional-
The albedo is \( \mathbf{mu} = (\mathbf{mu}_x, \mathbf{mu}_y, \mathbf{mu}_z) \). This is similar to the procedure of Tsai et al. [2]. For simulations, the ground truth masks \( \mathbf{m} \in \mathbb{R}^{HW} \) are two-dimensional binary images, so we first replicate \( \mathbf{m} \) along the depth dimension.

### S5. Additional Results

We provide a more extensive set of experimental results for comparison with various baselines. We show the surface reconstructions from different viewpoints in Figure S2. We show in Figure S3, the albedo and surface reconstructions produced by a number of methods: the light-cone transform [1], \( f-k \) migration [3], phasor fields [4], Tsai et al. [2], and Fermat flow [5].

![Images of reconstructions](image)

**Figure S3. Albedo and surface reconstructions using captured data:** The extracted normals (c) are used to fit surfaces (d). SU has a spatial resolution of \( 64 \times 64 \) pixels (1 min exposure), and the remaining scenes, \( 512 \times 512 \) (180 min exposure). We use \( \lambda = 2^0, 2^2 \) and \( 2^3 \) for SU, Discobolus and Dragon, respectively. Parameters of the other methods were optimized using grid search.

### References


