TurboSL: Dense, Accurate and Fast 3D by Neural Inverse Structured Light

Parsa Mirdehghan[1,2] Maxx Wu[1,2] Wenzheng Chen[1,2] David B. Lindell[1,2] Kiriakos N. Kutulakos[1,2]
[1]University of Toronto [2]Vector Institute
{parsa,wumaxx,wenzheng,lindell,kyros}@cs.toronto.edu

Abstract

We show how to turn a noisy and fragile active triangulation technique—three-pattern structured light with a grayscale camera—into a fast and powerful tool for 3D capture: able to output sub-pixel accurate disparities at megapixel resolution, along with reflectance, normals, and a no-reference estimate of its own pixelwise 3D error. To achieve this, we formulate structured-light decoding as a neural inverse rendering problem. We show that despite having just three or four input images—all from the same viewpoint—this problem can be tractably solved by *TurboSL*, an algorithm that combines (1) a precise image formation model, (2) a signed distance field scene representation, and (3) projection pattern sequences optimized for accuracy instead of precision. We use TurboSL to reconstruct a variety of complex scenes from images captured at up to 60 fps with a camera and a common projector. Our experiments highlight TurboSL's potential for dense and highly-accurate 3D acquisition from data captured in fractions of a second.

1. Introduction

Structured-light (SL) triangulation is one of the oldest and most widely used techniques for precise and reliable 3D shape acquisition [12]. Although many other depth-sensing modalities have advanced over the past decade [2, 23, 39], SL remains the most accessible way to capture mm- and sub-mm accurate geometry for a broad range of materials and appearances: all that is needed is to arrange one camera and a projector in a stereo configuration, adjust their baseline, and capture images while a sequence of patterns is projected onto a scene [9, 32, 40]. With multi-megapixel cameras readily available and inexpensive projectors boasting 4K resolution [10], SL acquisition of sub-mm-accurate, multi-megapixel depth maps is now fairly straightforward. Unfortunately, accurate SL is *slow*: many patterns must be projected onto a scene while it remains stationary [11], making 3D acquisition laborious and restrictive.

At the same time, the last few years have seen tremendous progress on learning 3D scene representations directly from 2D images, entirely passively [48]. These approaches use just a camera for acquisition and typically need a relatively large set of images even for desktop scenes. Their reliance on ambient illumination, however, is not sufficient for accurate surface reconstruction, especially for scenes that lack surface texture or have complex reflectance [15, 46, 50].

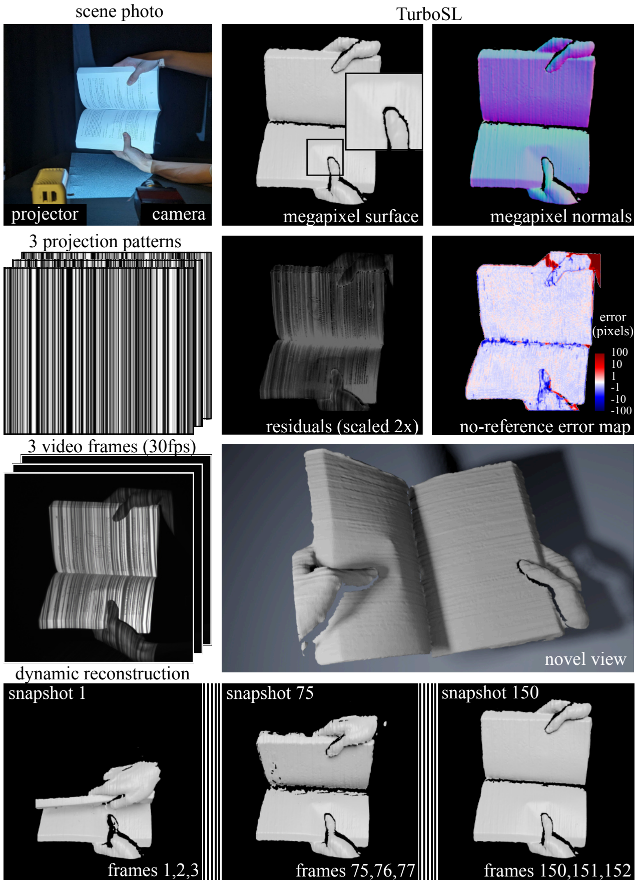

Figure 1: TurboSL for accurate and fast 3D sensing. From three consecutive grayscale SL video frames of a dynamic scene, TurboSL infers a detailed 3D surface; per-pixel surface normals and reflectances; and a disparity map with sub-pixel accuracy. It also estimates residual contributions for indirect and ambient light, and provides a no-reference proxy for the disparity error at each pixel.

In this work, we show how to leverage recent advances on neural 3D scene modeling to make SL image acquisition much faster. Specifically, by recasting grayscale multi-pattern SL as an inverse rendering problem—and by using a neural signed distance field method to solve it—we show that just three or four projection patterns are enough to obtain highly detailed geometric and photometric information about an unknown scene: (1) sub-pixel-accurate stereo correspondences, (2) surface normals and reflectances, (3) residual contributions from ambient and indirect illumination [35], and (4) a proxy for the depth uncertainty at each pixel. Our approach requires no pre-trained models and relies only on the captured SL images for supervision.

In contrast to conventional settings for neural scene modeling, few-pattern SL is an extreme case that involves just one camera viewpoint and a very small number of input images—as few as three. In this extreme case, one does not have the benefit of the implicit geometric constraints provided by multi-viewpoint image datasets, which have already been employed for 3D shape acquisition in conjunction with time of flight [1] and single-pattern SL data [44].

From the perspective of SL imaging, our approach can be thought of as addressing the classical problem of *SL decoding* [31]: given the n intensities observed at a pixel for n known projection patterns, the goal is to "decode" this information into a stereo correspondence on the projector's plane. Existing methods perform this decoding independently at each pixel—an approach that yields dense and highly-accurate correspondences for sufficiently large n [13] but introduces significant noise and outliers as n approaches the theoretical minimum of three patterns.

To tackle the data-starved regime of few-pattern SL, we take a global approach instead. Intuitively, even though the intensities observed at a single pixel may constrain the pixel's correspondence rather poorly, the SL images as a whole constrain the scene far more strongly—its 3D shape, its appearance, and even the optical properties of the projector illuminating it. We exploit this observation by formulating SL decoding as the joint inference of per-pixel correspondences; normals; reflectances; ambient and indirect contributions; and of a blur kernel for the projector's imperfect optics. We refer to this problem as *inverse structured light*.

Overall, our work makes the following contributions. First, we highlight an unexplored instance of inverse rendering that can break the longstanding SL tradeoff between speed, accuracy and density of 3D measurements. Second, we show that neural representations based on signed distance fields (SDFs) [25, 51, 52] are uniquely suited for this task. Specifically, alternative volumetric formulations [30] can easily overfit the limited image data in few-pattern SL, and 2D neural representations [33, 45] are sensitive to the frequency content of projection patterns and thus unsuitable as a general backbone for SL. Third, we show that precise modeling of SL image formation—from foreshortening of incident irradiance to imperfect projector optics—is key for pushing SL performance to the sub-pixel accuracy regime. Fourth, we observe that inverse SL produces particularly accurate reconstructions when paired with SL patterns that optimize the distinctiveness of adjacent pixels [31]. These patterns have received little attention in traditional SL because they produce 3D point clouds with many outliers among their very accurate inliers, making downstream geometry processing a major challenge. The inductive bias of our SDF representation, on the other hand, acts as an effective mechanism for outlier rejection that takes full advantage of the patterns' inherent accuracy. In TurboSL, we combine these patterns with a neural method for inverse SL. Fifth, we present the first experimental demonstration of acquiring megapixel-resolution disparity, normals, reflectance, and indirect/ambient contributions from three-pattern structured light, achieving sub-pixel accuracy with a standard projector and grayscale camera running at up to 60 fps.

2. Multi-Pattern Structured Light

Consider a projector and camera in a stereo configuration in front of an unknown scene (Figure 2, left). We seek to compute correspondences between their planes by capturing $n > 1$ images under n grayscale projection patterns. Unlike single-pattern SL whose spatial-smoothness assumptions limit 3D accuracy and spatial resolution [26], multi-pattern SL can theoretically compute the correspondence of each camera pixel individually [21].

Correspondence-finding in multi-pattern SL involves an encoding-decoding problem [16]: the n-dimensional vector of intensities emitted by a projector pixel encode its position on the projector's plane, and the n intensities measured at a camera pixel are "decoded" to localize the source of the incident light. Our focus is on the most extreme form of multi-pattern SL: establishing sub-pixel-accurate correspondences with the fewest possible patterns.

2.1. General SL Image Formation

Suppose the scene is within the depth of field of both the projector and the camera so that any patterns projected onto it are in focus. In this case, an infinitesimal patch $\mathbf{c}$ on the camera's plane will receive contributions from three types of light paths: (1) direct reflections of a projected pattern P_k, (2) indirect light originating from the projector, and (3) contributions from other light sources in the environment. The intensity measured at a discrete pixel $[i, j]$ is an integral over the pixel's footprint of these three contributions:

$$\begin{aligned} \mathrm{I}_k[i,j] \;=\; \int \Big\{ & \underbrace{\mathsf{r}(\mathbf{c})\,\big[\,\vec{\mathbf{n}}(\mathbf{c})\cdot\vec{\mathbf{i}}(\mathbf{c})\,\big]\,\big(\mathrm{B}*\mathrm{P}_k\big)(\mathbf{p})}_{\text{direct}} + \underbrace{\mathsf{a}(\mathbf{c})}_{\text{ambient}} \\ & + \underbrace{\int \mathrm{g}(\mathbf{p}' \to \mathbf{c})\,\big(\mathrm{B}*\mathrm{P}_k\big)(\mathbf{p}')\,d\mathbf{p}' \Big\}}_{\text{indirect}}\, d\mathbf{c} + \text{noise} \qquad (1) \end{aligned}$$

where $\mathbf{p}$ is the infinitesimal patch on the projector plane corresponding to $\mathbf{c}$; the convolution $\mathrm{B}*\mathrm{P}_k$ models optical degradations of the projection pattern, if any, due to the projector's imperfect optics; $\mathsf{r}(\mathbf{c})$ and $\vec{\mathbf{n}}(\mathbf{c})$ are the reflectance and unit normal of the surface point projecting to $\mathbf{c}$; the cosine factor $\vec{\mathbf{n}}(\mathbf{c})\cdot\vec{\mathbf{i}}(\mathbf{c})$ accounts for surface foreshortening relative to the incident light direction $\vec{\mathbf{i}}(\mathbf{c})$; and $\mathrm{g}(\mathbf{p}' \to \mathbf{c})$ represents indirect light transport from a general patch $\mathbf{p}'$ on the projector's plane to patch $\mathbf{c}$ on the camera.

The geometric relation between corresponding patches on the camera and projector can be conveniently expressed as a 2D displacement by assuming, without loss of generality, a rectified stereo configuration [42]:

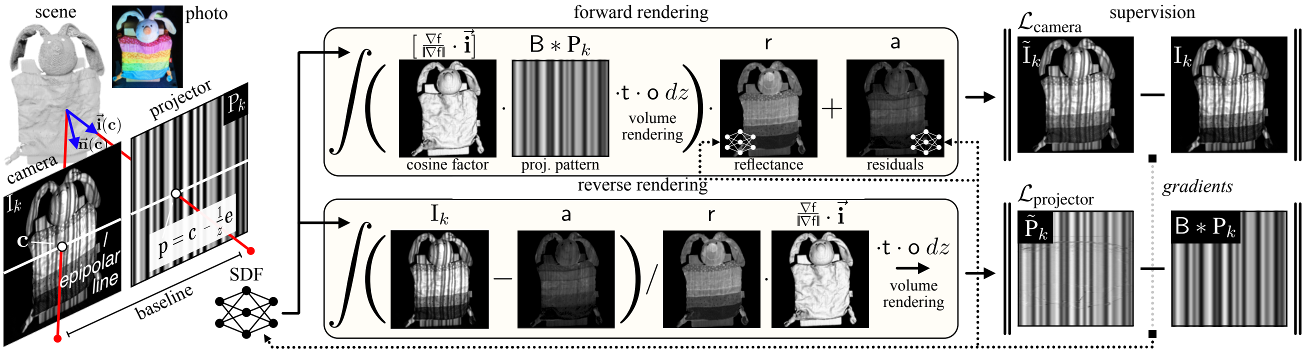

Figure 2: Imaging geometry and the TurboSL decoding algorithm.

$$\mathbf{p} \;=\; \mathbf{c} - \frac{1}{\mathsf{z}(\mathbf{c})}\begin{bmatrix}0\\ e\end{bmatrix} \quad , \qquad (2)$$

where $\mathsf{z}(\mathbf{c})$ is the depth of the point projecting to $\mathbf{c}$ and the scalar e depends on the SL system's geometry (*i.e.*, baseline, lens focal lengths and physical dimensions of camera and projector pixels). In the following we refer to the displacement in Eq. (2) as the pixel's *disparity*.

2.2. Key Challenges in Pixelwise SL Decoding

Due to the limited observations at each pixel, prior work on multi-pattern SL considers a simplified form of Eq. (1) that ignores indirect light and imperfect optics, and treats camera and projector pixels as infinitesimally small. This eliminates all integrals from the equation, leaving just three unknowns per pixel (Eq. (T1) in Table 1): the depth $\mathsf{z}(\mathbf{c})$, a reflectance $\mathsf{r}(\mathbf{c})$ that absorbs surface foreshortening, and a residual term $\mathsf{a}(\mathbf{c})$ that also absorbs contributions from ambient and indirect light. Unfortunately, while pixelwise inversion of this simplified model is possible when n is large enough, significant challenges remain for small n.

Near-minimal decoding. In theory, three patterns suffice to invert Eq. (T1) as long as they can be expressed as a non-intersecting curve in $\mathbb{R}^3$ [16]. In practice, however, few-pattern SL performs poorly even for highly-optimized pattern sequences [31]. This is because SL patterns are designed to tackle several competing objectives—providing a large range of unambiguous disparities [21]; ensuring the codes of nearby projector pixels are distinguishable with high probability in presence of noise [17]; and making indirect light approximately constant across patterns so it can be absorbed by the ambient term [18]. These objectives become mutually contradictory when n is small, leading to unavoidable pixelwise decoding errors regardless of pattern choice [41]. Our work sidesteps this problem with a neural approach that inverts the full Eq. (1)—except for the indirect light integral—over the whole image.

Sub-pixel-accurate decoding. The demands on the SL decoder are particularly stringent when the stereo baseline is small. Small baselines lead to compact systems and reduce shadows and occlusions, but achieving millimeter-scale accuracy with small baselines and large fields of view can easily push the required disparity error to *well below one pixel*.[1][2] This is well beyond the state of the art in few-pattern SL [5], and outside the reach of SL techniques that compute pixel-resolution disparities only [31]. In contrast, our SDF-based approach optimizes correspondences over a continuous domain and takes both geometric and photometric image cues into account for higher accuracy.

Pattern-agnostic decoding. In almost all prior work on SL [16, 17, 32], decoding algorithms have been designed in conjunction with the projection patterns themselves. Given the wide range of SL encoding schemes available today, an important question is how to distinguish the performance of the encoding scheme from the performance of the decoder itself. To that end, the ZNCC decoder [31] treats SL decoding as a general optimization problem and was shown to be optimal under an additive noise model, *i.e.*, it returns the maximum-likelihood pixel correspondence for any given sequence of projection patterns. Despite its theoretical guarantees and state-of-the-art performance [53], few-pattern SL remains a challenge for this decoder as well (Figure 3, top). By formulating decoding as an inverse rendering problem, our approach is equally agnostic to the SL patterns used, but far more accurate (Figure 3, bottom).

Normal estimation & geometry processing. Pixelwise decoders conflate reflectance and orientation, outputing unoriented point clouds [17]. This leaves normal estimation and surface extraction to downstream pipelines [3], a challenging task for few-pattern SL, whose noisy disparities and outliers can be quite significant. In contrast, our approach computes disparities and normals jointly, taking advantage of the strong relation between orientation, appearance, and sub-pixel disparity (Eq. (1)) present in the raw SL images. This is in the spirit of end-to-end processing of raw sensor measurements for rendering and reconstruction [1, 27, 47].

[1] In our experimental setup, for example, a 1 mm depth error at 70 cm corresponds to a disparity error of 0.5 projector pixels.

[2] While SL performance on real scenes is usually measured in terms of metric depth error [16], this error can be reduced by merely increasing a system's baseline without any decoder improvements. A more appropriate performance measure for SL decoders is the *disparity error* [42], also adopted in this work, which is invariant to the stereo baseline.

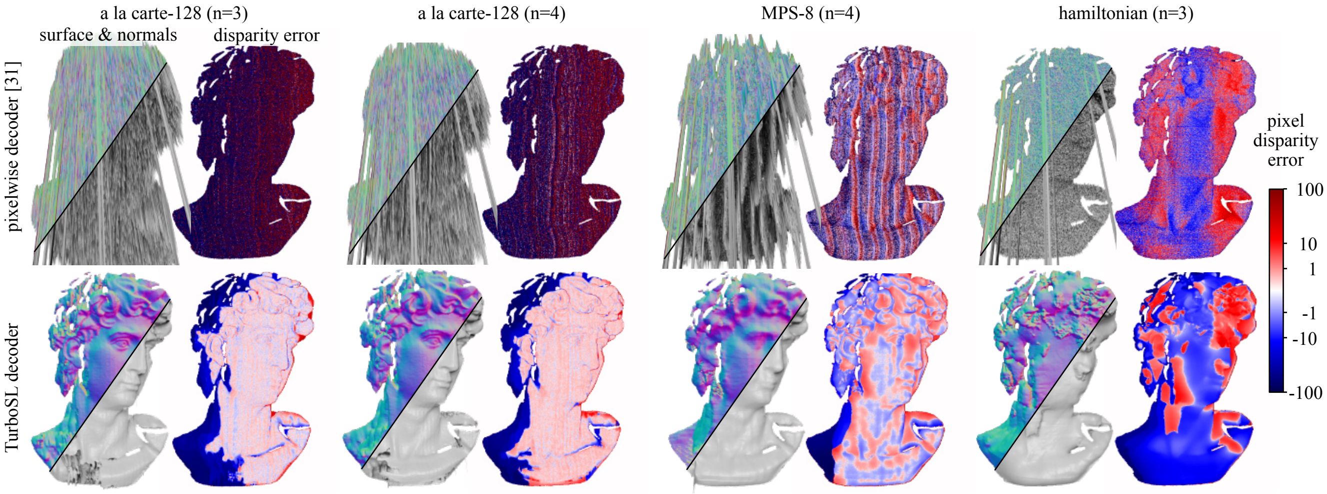

Figure 3: Pixelwise decoding vs TurboSL decoding for few-pattern SL. Top: Reconstructions from ZNCC [31], a state-of-the-art pixelwise decoder, for SL images of the real scene in Figure 4. We show results for three pattern families: a la carte [31] with maximum frequency 128, optimized for zero disparity error; MPS [17] with maximum frequency 8, designed for resistance to indirect light; and Hamiltonian [16], optimized for low mean disparity error. All 3D meshes are poor in this setting (see supplement Section A.4 for mesh extraction details and Section B.2 for results with $n = 6$, which are much better). **Bottom:** The TurboSL decoder's results on the same input. Note the far better accuracy of a la carte SL patterns and the relatively small difference between $n = 3$ and $n = 4$ for that family.

Quantifying spatial uncertainty. SL decoding errors can vary a lot across scenes and even from pixel to pixel within a scene. This occurs for two reasons. First, the signal-to-noise ratio (SNR) of individual pixels is not constant because of differences in reflectance, orientation, projector distance, and incident ambient/indirect light. Second, all SL encoding schemes introduce error non-uniformities to some extent, because the n-dimensional code vector assigned to some projector pixels may be less distinctive than others [21]. These non-uniformities are especially pronounced in few-pattern SL where very few degrees of freedom are available for encoding position. As a result, variations in depth uncertainty can be significant even among camera pixels with identical SNR. The question of how depth uncertainty can be estimated from the SL images themselves is poorly understood, yet this information can be potentially very useful for geometry processing downstream [43]. We partially address this question in Section 4.2 by computing two distinct reconstructions as proxies for uncertainty.

3. Neural Inverse Structured Light

Equation (1) is nothing other than a restricted version of the general rendering equation [22], where a projector is the scene's main light source and the projection patterns are designed to create images highly informative of the scene's 3D shape. From this perspective, pixelwise SL decoding can be viewed as an instance of inverse rendering [28], but solved independently at every camera pixel.

Neural multi-view reconstruction methods solve an inverse rendering problem through a global optimization of scene appearance and geometry, which are each parameterized using neural networks [4, 51, 54]. The optimization generally follows an analysis-by-synthesis approach, wherein (1) an image formation model is used to render camera images based on the current estimates of scene appearance and geometry, (2) the rendered images are compared to the multi-view captured dataset, (3) gradients are computed to update scene geometry and appearance, and (4) this process is repeated iteratively to improve reconstruction fidelity.

Application to inverse SL. Applying neural reconstruction involves choosing how to simplify Eq. (1) and parameterize its various terms. Table 1 summarizes three general choices.

The first is to use a decoding approach based on 2D multi-layer perceptrons (MLPs) to invert the same image formation model employed by pixelwise-decoding methods (Eq. (T2) in Table 1). In this case, the MLPs learn to map continuous-valued 2D input coordinates to the corresponding values of depth, reflectance, and residual contributions, with a global representation for geometry and appearance.

A second option is to rely on a 3D neural representation that employs MLPs [30] or feature-based networks [33] to map 3D input coordinates to values of an opaque density field, which models the differential probability of camera ray termination at each point in space. This allows use of a more general—albeit approximate—image formation model based on the volume rendering equation (Eq. (T3) in Table 1). Implementing this equation to render the intensity of a camera pixel $\mathbf{c}$ involves marching along its 3D ray; retrieving the intensity of the corresponding projector pixel $\mathrm{P}_k(\mathbf{c}-\frac{1}{z}\mathbf{e})$ for each 3D point along the ray; and alpha-compositing these intensities based on the accumulated density [29]. The result of this integration is scaled by the reflectance $\mathsf{r}(\mathbf{c})$ and combined with the residual term $\mathsf{a}(\mathbf{c})$, both of which are represented by 2D MLPs.

Decoding approach	Scene representation	Density $\mathsf{o}(\mathbf{c},z)$	Transmittance $\mathsf{t}(z)$	Image formation model	
Pixel-wise decoding	depth $\mathsf{z}(\mathbf{c})$, reflectance $\mathsf{r}(\mathbf{c})$, residual $\mathsf{a}(\mathbf{c})$	N/A	N/A	$\mathsf{r}(\mathbf{c})\ \mathrm{P}_k(\mathbf{c}-\frac{1}{\mathsf{z}(\mathbf{c})}\mathbf{e})\ +\ \mathsf{a}(\mathbf{c})$	(T1)
Learn 2D MLPs	depth $\mathsf{z}(\mathbf{c})$, reflectance $\mathsf{r}(\mathbf{c})$, residual $\mathsf{a}(\mathbf{c})$	N/A	N/A	$\mathsf{r}(\mathbf{c})\ \mathrm{P}_k(\mathbf{c}-\frac{1}{\mathsf{z}(\mathbf{c})}\mathbf{e})\ +\ \mathsf{a}(\mathbf{c})$	(T2)
Learn 3D opaque density field	density $\mathsf{o}(\mathbf{c},z)$, reflectance $\mathsf{r}(\mathbf{c})$, residual $\mathsf{a}(\mathbf{c})$	N/A	$\exp\left(-\int_{z_{\min}}^{z}\mathsf{o}(\mathbf{c},u)du\right)$	$\mathsf{r}(\mathbf{c})\int_{z_{\min}}^{z_{\max}}\mathrm{P}_k(\mathbf{c}-\frac{1}{z}\mathbf{e})\,\mathsf{t}(z)\,\mathsf{o}(\mathbf{c},z)dz\ +\ \mathsf{a}(\mathbf{c})$	(T3)
Learn 3D SDF (forward rendering)	0-level set of SDF $\mathsf{f}(\mathbf{c},z)$, reflectance $\mathsf{r}(\mathbf{c})$, residual $\mathsf{a}(\mathbf{c})$, normal $\vec{\mathbf{n}}(\mathbf{c})$	$\max\left(\frac{-\frac{d}{dz}\sigma(\mathsf{f}(\mathbf{c},z))}{\sigma(\mathsf{f}(\mathbf{c},z))},0\right)$	$\exp\left(-\int_{z_{\min}}^{z}\mathsf{o}(\mathbf{c},u)du\right)$	$\mathsf{r}(\mathbf{c})\int_{z_{\min}}^{z_{\max}}\left[\hat{\nabla}\mathsf{f}(\mathbf{c},z)\cdot\vec{\mathbf{i}}(\mathbf{c},z)\right](\mathrm{B}*\mathrm{P}_k)(\mathbf{c}-\frac{1}{z}\mathbf{e})\,\mathsf{t}(z)\,\mathsf{o}(\mathbf{c},z)dz\ +\ \mathsf{a}(\mathbf{c})$	(T4)
Learn 3D SDF (reverse rendering)	0-level set of SDF $\mathsf{f}(\mathbf{c},z)$, reflectance $\mathsf{r}(\mathbf{c})$, residual $\mathsf{a}(\mathbf{c})$, normal $\vec{\mathbf{n}}(\mathbf{c})$	$\max\left(\frac{-\frac{d}{dz}\sigma(\mathsf{f}(\mathbf{c},z))}{\sigma(\mathsf{f}(\mathbf{c},z))},0\right)$	$\exp\left(-\int_{z_{\min}}^{z}\mathsf{o}(\mathbf{c},u)du\right)$	$\int_{z_{\min}}^{z_{\max}}\frac{\mathrm{I}_k(\mathbf{p}+\frac{1}{z}\mathbf{e})-\mathsf{a}(\mathbf{p}+\frac{1}{z}\mathbf{e})}{\mathsf{r}(\mathbf{p}+\frac{1}{z}\mathbf{e})[\hat{\nabla}\mathsf{f}(\mathbf{c},z)\cdot\vec{\mathbf{i}}(\mathbf{c},z)]}\,\mathsf{t}(z)\,\mathsf{o}(\mathbf{p}+\frac{1}{z}\mathbf{e},z)dz$	(T5)

Table 1: Basic approaches, scene representations and image formation models for inverse SL. We define $\mathbf{e}=[0\ \ e]^T$, $\hat{\nabla}\mathsf{f}()=\nabla\mathsf{f}()/\|\nabla\mathsf{f}()\|$.

Neither of these two choices proved effective in our experiments. In particular, contrary to their success in multi-view 3D reconstruction [6, 7], we find that 2D MLPs fail to recover accurate geometry when applied to inverse SL, especially when high-frequency patterns are used for projection (Figure 4, middle). Intuitively, using a depth map representation for inverse SL is challenging because it explicitly associates each camera pixel with a single depth and thus a single projector pixel. This fails to handle the ambiguities inherent in SL with high-frequency patterns, where image noise can make individual projector pixels less distinguishable. Volumetric approaches, on the other hand, are *too flexible*: they can reproduce the (very few) SL images given as input very accurately, but extracting a surface from the volume involves accumulating density along individual camera rays and finding the expected ray termination distance [8], which yields inaccurate geometry (Figure 4, right).

The third option, which we adopt, draws on the advantages of both 2D and volumetric representations: modeling surfaces explicitly while flexibly handling ambiguity.

4. The TurboSL Decoder

The TurboSL decoding algorithm uses a hybrid representation based on volume rendering and a signed distance field representation [36, 51, 55]. We represent geometry using an efficient feature-based network [25, 33] to map input 3D coordinates to the values of an SDF. The SDF values are then converted to density following Wang *et al*. [51]. We render camera images by volume rendering according to Eq. (T4). This representation provides an explicit model for surfaces, defined as the zero level set of the SDF.

4.1. Bidirectional Rendering

The data-starved conditions in which TurboSL operates require exploiting all available constraints in the optimization for better 3D fidelity. To that end, we introduce *forward* and *reverse* rendering procedures that enforce consistency of the scene representation and the projector blur kernel with the SL input images, *as well as the patterns that produced them*.

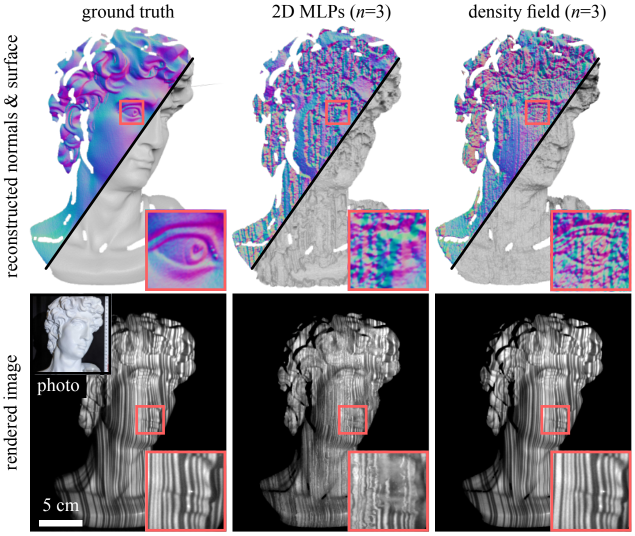

Figure 4: MLPs and density fields for inverse SL. Reconstructing a statue from $n = 3$ SL images using a la carte projection patterns of maximum frequency 128 [31]. The MLP-based reconstruction has many artifacts and does not reproduce the input images. While the volumetric approach reproduces the images faithfully, the recovered surface is inaccurate.

Forward rendering. In the forward-rendering direction we reconstruct a camera image according to Eq. (T4). We march along rays from the camera center of projection, through the image plane, and into the scene (Figure 5, bottom left). We query the neural network with 3D points sampled along the ray to recover the corresponding SDF values and density. Then, we project these points into the projector plane, retrieve the projector pattern's intensity, and alpha-composite the resulting intensities along the ray. We incorporate the cosine factor $\frac{\nabla\mathsf{f}(\mathbf{c},z)}{\|\nabla\mathsf{f}(\mathbf{c},z)\|}\cdot\vec{\mathbf{i}}(\mathbf{c},z)$ into the rendering along the ray by calculating the SDF's gradient to query surface normals. The result of the integral is scaled by a learned reflectance term $\mathsf{r}(\mathbf{c})$ parameterized by a 2D MLP. An additional 2D MLP is used to parameterize the residual component $\mathsf{a}(\mathbf{c})$, which absorbs ambient and global illumination and any other unmodeled contributions. The 2D and 3D neural networks are optimized to minimize differences between the rendered and captured camera images, and af-

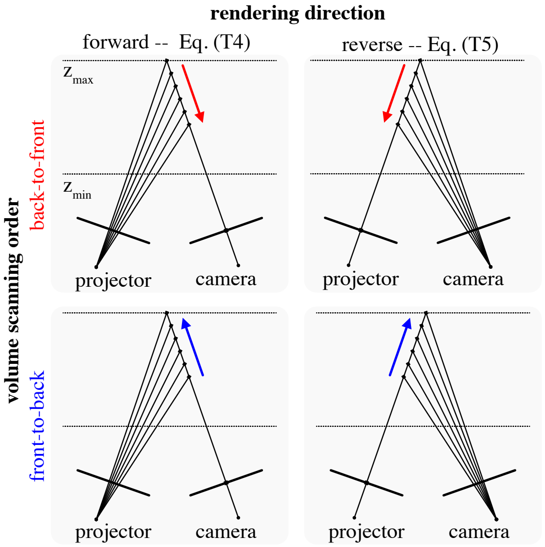

Figure 5: The four ray-marching configurations of TurboSL.

ter optimization, we extract a mesh from the SDF (see supplement Section A.4 for details).

To account for projector non-idealities, we convolve the (known) projection patterns with a single kernel B before rendering the camera image. This kernel is optimized with the scene representation using just the SL input images.

Reverse rendering. In the reverse-rendering direction we reconstruct the projector pattern according to Eq. (T5) using the same 2D and 3D neural networks as before. We march along rays through each projector pixel and project the 3D points along each ray into the camera image plane to retrieve the corresponding camera measurement, $\mathrm{I}_k(\mathbf{p}+\frac{1}{z}\mathbf{e})$ (Figure 5, bottom right). We incorporate the reflectance, cosine factor, and residual components, and we optimize the neural networks to minimize the difference between the rendered and projected patterns.

4.2. Bidirectional Scans as an Uncertainty Proxy

To capture reconstruction uncertainty, we optimize two separate scene models—one using the procedure of Section 4.1 and one by scanning the volume in a back-to-front order (Figure 5, top). The intuition behind this approach is as follows. In SL, images are captured from a single viewpoint and so each camera ray encounters one surface at most (we assume no semi-transparent surfaces). Effectively, the volume rendering procedure finds projector-pattern correspondences by scanning along the epipolar line from one direction. As we march along a camera ray in front-to-back volume scanning, we trace out points in one direction along an epipolar line on the projector plane (Figure 5, bottom left). Back-to-front scanning proceeds in the same fashion, but starts from the other side of the epipolar line and scans in the opposite direction (Figure 5, top left). This bidirectional scanning is unique to inverse SL and cannot be applied to conventional multi-view reconstruction where camera rays can pass through multiple reconstructed surfaces.

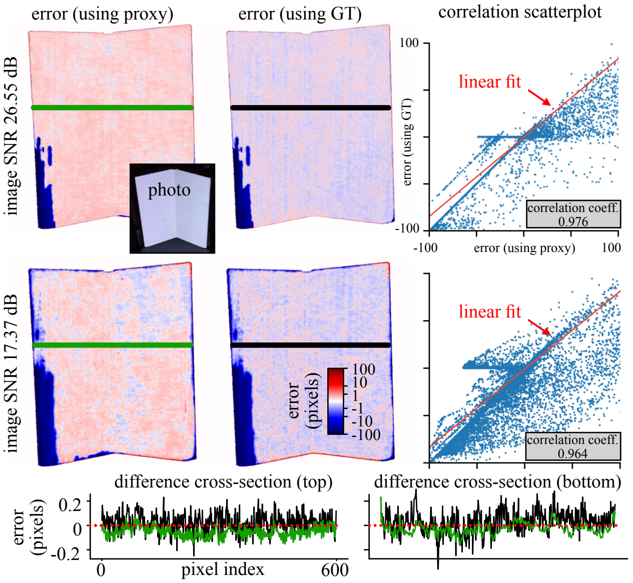

Figure 6: No-reference proxy for disparity error. Comparison of the signed difference between front-to-back and back-to-front disparity maps (left) and between front-to-back and ground-truth maps (middle) for a wedge-like surface reconstructed from two sets of SL images—one captured (top) and one with extra 1% Gaussian noise added to the captured set (bottom). The difference cross-sections show agreement to within a fraction of a pixel, and the scatter-plots demonstrate strong correlation across all disparity errors. The SL patterns used were the same as in Figure 4.

Our empirical observation is that pixelwise subtraction of reconstructed front-to-back and back-to-front disparity maps provides a proxy for the *actual* disparity error, *i.e.*, it is highly correlated with the pixelwise disparity error against an accurate reference (Figure 6). This type of uncertainty modeling could be useful for downstream tasks, and is reminiscent of reference-free 3D reconstruction metrics. For example, in structural biology and other fields, the Fourier shell correlation metric similarly assesses correlation between two independently reconstructed volumes [49]. We use this proxy to remove low-confidence pixels in the experiments of Section 5.

4.3. Optimization

The TurboSL decoder trains by minimizing the following losses with respect to the parameters of the 2D and 3D neural networks. The camera loss penalizes the difference between the rendered and measured camera images, and the projector loss operates similarly for the projector patterns:

$$\mathcal{L}_{\text{cam}} = \sum_{k,\mathbf{c}} \|\mathrm{I}_k(\mathbf{c}) - \tilde{\mathrm{I}}_k(\mathbf{c})\|,\ \ \mathcal{L}_{\text{proj}} = \sum_{k,\mathbf{p}} \|(\mathrm{B} * \mathrm{P}_k)(\mathbf{p}) - \tilde{\mathrm{P}}_k(\mathbf{p})\|. \quad (3)$$

We also incorporate a loss $\mathcal{L}_{\text{mask}}$ to remove density along camera rays corresponding to shadowed parts of the scene and to enforce that rays passing through points illuminated by the projector are completely attenuated by density (to prevent transparent surfaces). We calculate a binary mask

$\mathcal{M}(\mathbf{c})$ for illuminated camera pixels by applying a threshold on the difference between the brightest and darkest pixel values across patterns (shadowed pixels show little variability across patterns). The loss is computed as

$$\mathcal{L}_{\text{mask}} = \underset{z,\mathbf{c}}{\mathsf{BCE}}\left(\mathcal{M}(\mathbf{c}), \int_{z_{\min}}^{z_{\max}} \mathsf{t}(z)\,\mathsf{o}(\mathbf{c},z)\,dz\right), \tag{4}$$

where BCE is the binary cross entropy loss, and the integral argument computes the total attenuation along the ray. Last, we incorporate an eikonal penalty to enforce that the gradient of the SDF be equal to one, and we add an L1 penalty on densities sampled along the rays to avoid regions of spurious non-zero density:

$$\mathcal{L}_{\text{eik}} = \sum_{z,\mathbf{c}} (\|\nabla \mathsf{f}(\mathbf{c},z)\|_2 - 1)^2,\ \mathcal{L}_{\text{sp}} = \sum_{z,\mathbf{c}} \exp(-|\mathsf{f}(\mathbf{c},z)|)\,. \tag{5}$$

In practice, we apply both L1 and L2 penalties for the projector and camera losses, and we compute a weighted combination of all losses (see supplement Section A for details).

Implementation details. Our implementation of TurboSL builds on top of the NerfAcc framework [24], augmented to include the hybrid SDF and volumetric surface parameterization [14]. Each model is trained by sampling rays from both the camera and projector and performing forward and backward rendering. Training for $n = 3$ input images takes 12 minutes and uses 10GB of VRAM on an RTX6000 GPU, *i.e*., on par with state-of-the-art neural rendering methods [33]. To avoid numerical instability because of the division in Eq. (T5), we use only camera loss for the first 1000 iterations and add the projector loss until convergence. To capture uncertainty, we train two models for each scene using front-to-back and back-to-front volume scanning, and render all surfaces using the zero level set of the front-to-back-trained model. To model projector blur, we optimize two separable one-dimensional filters of size 11 pixels and use their outer product as the blur kernel (see supplement Section A.3). While the camera mask can be estimated from input images, the projector mask is unknown. Randomly sampling the projector rays with a constant "background" intensity leads to the creation of spurious densities. To avoid this, we assume that scene points at infinite depth along the ray through a projector pixel have reflectance equal to the projection pattern's intensity at that pixel. This allows the projector loss to carve out the space.

5. Experimental Results

We show several results from two SL system configurations, one geared toward 60 fps image acquisition (IDS UI-3240CP camera) and one for 30 fps acquisition (Prosilica GT1920c camera). We use the same off-the-shelf projector (LG PH550 LED mini projector) and the same baseline for both configurations. We use TurboSL's a la carte pattern family for all results discussed below. For more results, see supplement Section B and the supplemental video.

Dynamic scene reconstruction. To demonstrate the rapid image acquisition capabilities of TurboSL, we use live video data to reconstruct a dynamic scene exhibiting non-rigid motion. Three frequency-128 a la carte patterns are repeatedly projected onto a hand-held book, while capturing synchronized 30 fps video with our camera. Figure 1 (row 4) shows three snapshots of the reconstructions.

Ablation study of TurboSL components. We conduct an ablation study of several components of the TurboSL decoder in the top row of Figure 7. We capture SL images of a statue with three a la carte patterns of maximum frequency 32, and reconstruct the scene using only the camera loss. Without the other components, the approach fails to accurately model the captured images, and spatial structures from the projector patterns leak into the reconstructed surface. Adding the projector loss and cosine factor improves the reconstructed surface somewhat and eliminates some of the minor banding artifacts from the geometry. Finally, incorporating the learnable projector blur kernel yields a low image reconstruction error across most of the statue, and the recovered geometry closely matches the ground truth.

Impact of more patterns. Although TurboSL achieves sub-pixel-level accuracy with as few as three SL patterns (Figure 3), increasing the number of patterns improves the accuracy of the recovered geometry. This can be observed in the second row of Figure 7, where we capture a plush toy using 4, 5, and 6 a la carte patterns with maximum frequency 32. While using four patterns already reconstructs geometry at the sub-pixel level, additional patterns improve the smoothness of the surface normals and reveal additional wrinkles and fine details in the fabric of the toy. Inspecting the disparity error maps also shows that the reconstruction improves in many areas, albeit with some outliers.

Bidirectional rendering. The third row of Figure 7 shows the outputs of TurboSL's bidirectional rendering scheme, including surfaces and normals recovered using front-to-back and back-to-front rendering. The scene was reconstructed using four patterns with a maximum frequency of 64. While both rendered surfaces capture similar details, the reconstructions disagree in areas such as the legs and ears. As can be seen, the no-reference error map computed by subtracting the front-to-back and back-to-front disparity maps helps identify regions of high ground-truth error. Although the no-reference error map overestimates the ground-truth error in some regions (*e.g*. the right ear), it contains very few false negatives, *i.e*., points where the ground-truth error is underestimated. By using a threshold on this conservative no-reference error map (described in supplement Section C.2), we obtain an inlier mask that identifies the high-confidence 3D points and normals.

Indirect light. Scenes exhibiting strong indirect lighting effects are a challenge for SL systems [19, 20, 37, 38]. Row 4 of Figure 7 shows TurboSL's results for a bowl filled with a pumpkin and two clementines, using three SL patterns

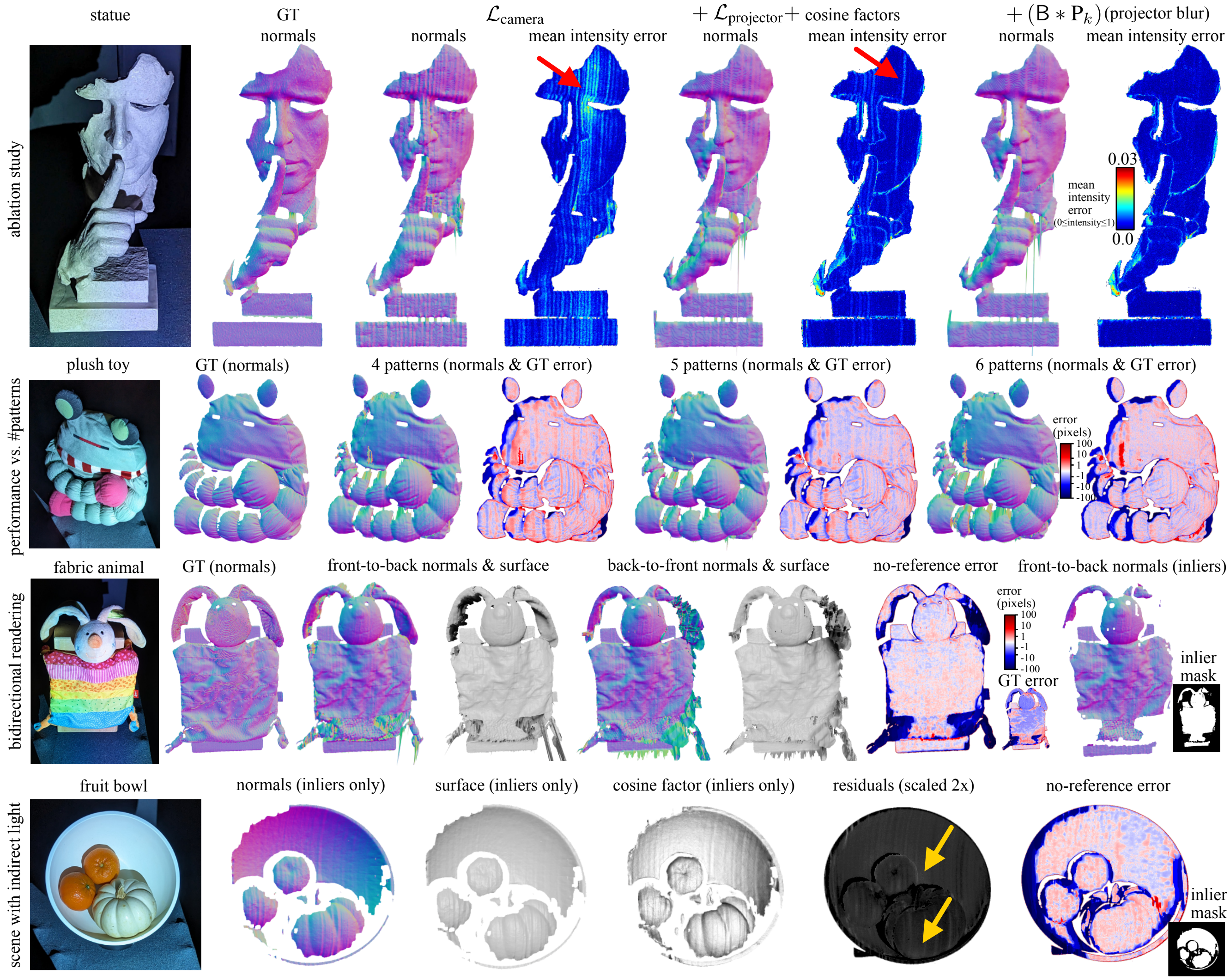
statue
GT normals
$\mathcal{L}_{\text{camera}}$
normals
mean intensity error
$+\mathcal{L}_{\text{projector}}$ + cosine factors
normals
mean intensity error
$+(\mathsf{B} * \mathsf{P}_k)$ (projector blur)
normals
mean intensity error
ablation study
0.03
mean intensity error (0≤intensity≤1)
0.0
plush toy
GT (normals)
4 patterns (normals & GT error)
5 patterns (normals & GT error)
6 patterns (normals & GT error)
performance vs. #patterns
error (pixels)
100
10
1
-1
-10
-100
fabric animal
GT (normals)
front-to-back normals & surface
back-to-front normals & surface
no-reference error
front-to-back normals (inliers)
bidirectional rendering
error (pixels)
100
10
1
-1
-10
-100
GT error
inlier mask
fruit bowl
normals (inliers only)
surface (inliers only)
cosine factor (inliers only)
residuals (scaled 2x)
no-reference error
scene with indirect light
inlier mask

Figure 7: TurboSL results. **Row 1:** We assess the impact on image error of including the camera loss, projector loss, cosine factor, and projector blur terms for three patterns of maximum frequency 32. Including each term in sequence improves image and 3D reconstruction fidelity (red arrows). **Row 2:** Increasing the number of patterns up to 6 further improves reconstruction accuracy. **Row 3:** Reconstructions using bidirectional rendering have strong agreement in some areas, less so in others. Regions where the reconstructions differ correlate with large ground-truth error, providing a conservative estimate of reconstruction uncertainty. **Row 4:** TurboSL captures fine geometric details in this challenging scene with indirect lighting; areas affected by indirect light can be observed in the residuals image (yellow arrows).

of maximum frequency 64. The impact of indirect light is particularly strong near the objects' boundaries deep inside the bowl, where diffuse interreflections contribute the most [34, 35]. The residual term does capture their contributions (*e.g.* lower edge of pumpkin, boundary of uppermost clementine), along with unmodeled stripe-like contributions from the projector. Despite these effects, the no-reference metric allows automatic exclusion of geometry near the boundary where depth errors are most severe, yielding high quality geometry despite them. A similar behavior can be observed in the results of Figure 1, near the book's crease. We believe that further exploration of this metric—and bidirectional rendering more generally—is needed to fully understand its practical implications.

6. Concluding Remarks

Approaching neural inverse rendering from the perspective of SL yields unique insights. While much recent work in inverse rendering focuses on what can be achieved using images captured from tens to hundreds of viewpoints, SL imaging offers a counterpoint: careful camera–projector modeling and modern neural rendering techniques can enable robust, sub-mm 3D reconstruction from a single viewpoint with commodity hardware.

Acknowledgements. KNK and DBL gratefully acknowledge the support of NSERC under the RGPIN and RTI programs. MW thanks the support of Vector Institute.

References

[1] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. In *NeurIPS*, 2021.

[2] Seung-Hwan Baek, Noah Walsh, Ilya Chugunov, Zheng Shi, and Felix Heide. Centimeter-wave free-space neural time-of-flight imaging. *ACM Trans. Graph.*, 42(1):1–18, 2023.

[3] Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Joshua A Levine, Andrei Sharf, and Claudio T Silva. State of the art in surface reconstruction from point clouds. In *Eurographics*, 2014.

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *CVPR*, 2021.

[5] Wenzheng Chen, Parsa Mirdehghan, Sanja Fidler, and Kiriakos N Kutulakos. Auto-tuning structured light by optical stochastic gradient descent. In *CVPR*, 2020.

[6] Ilya Chugunov, Yuxuan Zhang, Zhihao Xia, Xuaner Zhang, Jiawen Chen, and Felix Heide. The implicit values of a good hand shake: Handheld multi-frame neural depth refinement. In *CVPR*, 2022.

[7] Ilya Chugunov, Yuxuan Zhang, and Felix Heide. Shakes on a plane: Unsupervised depth estimation from unstabilized photography. In *CVPR*, 2023.

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, 2022.

[9] Matea Donlic, Tomislav Petkovic, and Tomislav Pribanic. On tablet 3D structured light reconstruction and registration. In *ICCV Workshops*, 2017.

[10] Marc-Antoine Drouin, Francois Blais, and Guy Godin. High resolution projector for 3D imaging. In *3DV*, 2014.

[11] Eyþór Rúnar Eiríksson, Jakob Wilm, David Bue Pedersen, and Henrik Aanæs. Precision and accuracy parameters in structured light 3-D scanning. *ISPRS Archives*, 5:7–15, 2016.

[12] Jason Geng. Structured-light 3D surface imaging: A tutorial. *Adv. Opt. Photonics*, 3(2):128–160, 2011.

[13] Jens Gühring. Dense 3D surface acquisition by structured light using off-the-shelf components. In *Videometrics*, 2000.

[14] Yuan-Chen Guo. Instant neural surface reconstruction, 2022. https://github.com/bennyguo/instant-nsr-pl.

[15] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. NeRFReN: Neural radiance fields with reflections. In *CVPR*, 2022.

[16] Mohit Gupta and Nikhil Nakhate. A geometric perspective on structured light coding. In *ECCV*, 2018.

[17] Mohit Gupta and Shree K Nayar. Micro phase shifting. In *CVPR*, 2012.

[18] Mohit Gupta, Amit Agrawal, Ashok Veeraraghavan, and Srinivasa G Narasimhan. Structured light 3D scanning in the presence of global illumination. In *CVPR*, 2011.

[19] Mohit Gupta, Yuandong Tian, Srinivasa G Narasimhan, and Li Zhang. A combined theory of defocused illumination and global light transport. *Int. J. Comput. Vis.*, 98:146–167, 2012.

[20] Mohit Gupta, Amit Agrawal, Ashok Veeraraghavan, and Srinivasa G Narasimhan. A practical approach to 3D scanning in the presence of interreflections, subsurface scattering and defocus. *Int. J. Comput. Vis.*, 102:33–55, 2013.

[21] Eli Horn and Nahum Kiryati. Toward optimal structured light patterns. *Image Vis. Comput.*, 17(2):87–97, 1999.

[22] James T Kajiya. The rendering equation. In *SIGGRAPH*, 1986.

[23] Alankar Kotwal, Anat Levin, and Ioannis Gkioulekas. Passive micron-scale time-of-flight with sunlight interferometry. In *CVPR*, 2023.

[24] Ruilong Li, Matthew Tancik, and Angjoo Kanazawa. NerfAcc: A general NeRF acceleration toolbox. *arXiv preprint arXiv:2210.04847*, 2022.

[25] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *CVPR*, 2023.

[26] Haibo Lin, Lei Nie, and Zhan Song. A single-shot structured light means by encoding both color and geometrical features. *Pattern Recognit.*, 54:178–189, 2016.

[27] Anagh Malik, Parsa Mirdehghan, Sotiris Nousias, Kiriakos N Kutulakos, and David B Lindell. Transient neural radiance fields for lidar view synthesis and 3D reconstruction. In *NeurIPS*, 2023.

[28] Stephen Robert Marschner. *Inverse rendering for computer graphics*. Phd thesis, Cornell University, 1998.

[29] Nelson Max. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 1(2):99–108, 1995.

[30] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2021.

[31] Parsa Mirdehghan, Wenzheng Chen, and Kiriakos N Kutulakos. Optimal structured light a la carte. In *CVPR*, 2018.

[32] Daniel Moreno, Fatih Calakli, and Gabriel Taubin. Unsynchronized structured light. *ACM Trans. Graph.*, 34(6):1–11, 2015.

[33] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):1–15, 2022.

[34] Shree K Nayar, Katsushi Ikeuchi, and Takeo Kanade. Shape from interreflections. *Int. J. Comput. Vis.*, 6:173–195, 1991.

[35] Shree K Nayar, G Krishnan, Michael D Grossberg, and Ramesh Raskar. Fast separation of direct and global components of a scene using high frequency illumination. In *SIGGRAPH*, 2006.

[36] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *CVPR*, 2021.

[37] Matthew O'Toole, Supreeth Achar, Srinivasa G Narasimhan, and Kiriakos N Kutulakos. Homogeneous codes for energy-efficient illumination and imaging. *ACM Trans. Graph.*, 34 (4):1–13, 2015.

[38] Matthew O'Toole, John Mather, and Kiriakos N Kutulakos. 3D shape and indirect appearance by structured light transport. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(7):1298–1312, 2016.

[39] Joshua Rapp, Julian Tachella, Yoann Altmann, Stephen McLaughlin, and Vivek K Goyal. Advances in single-photon lidar for autonomous vehicles: Working principles, challenges, and recent advances. *IEEE Signal Process. Mag.*, 37(4):62–71, 2020.

[40] Claudio Rocchini, Paulo Cignoni, Claudio Montani, Paolo Pingi, and Roberto Scopigno. A low cost 3D scanner based on structured light. *Comput. Graph. Forum*, 20(3):299–308, 2001.

[41] Joaquim Salvi, Jordi Pages, and Joan Batlle. Pattern codification strategies in structured light systems. *Pattern Recognit.*, 37(4):827–849, 2004.

[42] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.*, 47:7–42, 2002.

[43] Silvia Sellán and Alec Jacobson. Stochastic Poisson surface reconstruction. *ACM Trans. Graph.*, 41(6):1–12, 2022.

[44] Aarrushi Shandilya, Benjamin Attal, Christian Richardt, James Tompkin, and Matthew O'Toole. Neural fields for structured lighting. In *ICCV*, 2023.

[45] Vincent Sitzmann, Julien Martel, Alexander Bergman, David B Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020.

[46] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021.

[47] Shuochen Su, Felix Heide, Gordon Wetzstein, and Wolfgang Heidrich. Deep end-to-end time-of-flight imaging. In *CVPR*, 2018.

[48] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. *Comput. Graph. Forum*, 41(2):703–735, 2022.

[49] Marin Van Heel and Michael Schatz. Fourier shell correlation threshold criteria. *J. Struct. Biol.*, 151(3):250–262, 2005.

[50] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022.

[51] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021.

[52] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *CVPR*, 2023.

[53] Xianmin Xu, Yuxin Lin, Haoyang Zhou, Chong Zeng, Yaxin Yu, Kun Zhou, and Hongzhi Wu. A unified spatial-angular structured light for single-view acquisition of shape and reflectance. In *CVPR*, 2023.

[54] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020.

[55] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021.