

# Recovering Parametric Scenes from Very Few Time-of-Flight Pixels

Carter Sifferman<sup>†</sup>, Yiquan Li<sup>†</sup>, Yiming Li, Fangzhou Mu, Michael Gleicher, Mohit Gupta, Yin Li  
University of Wisconsin-Madison

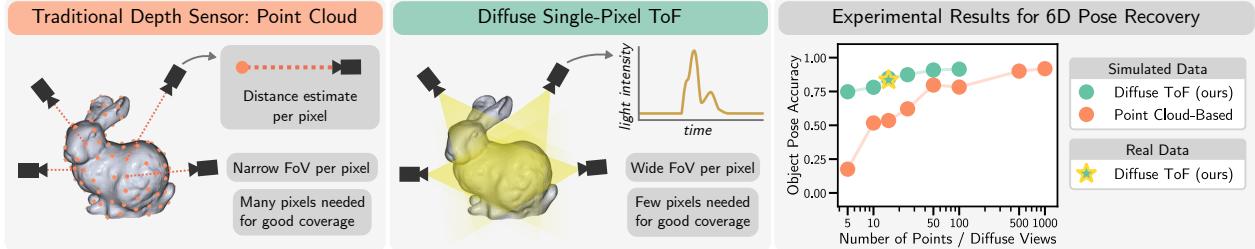


Figure 1. We introduce a method for recovering the geometry of parametric 3D scenes, such as the 6D pose of a known object, from a distributed set of **very few** (*e.g.*, 15), **diffuse** (*i.e.*, wide field-of-view) single-pixel ToF Sensors. Methods based on **traditional** depth sensors suffer poor performance under a low-pixel-count regime due to their sparse coverage. Our method outperforms a point cloud-based baseline by utilizing the entirety of data recovered by a diffuse ToF sensor. See Sec. 4.4 for details of the experiment.

## Abstract

We aim to recover the geometry of 3D parametric scenes using **very few** depth measurements from low-cost, commercially available time-of-flight sensors. These sensors offer very low spatial resolution (*i.e.*, a single pixel), but image a wide field-of-view per pixel and capture detailed time-of-flight data in the form of time-resolved photon counts. This time-of-flight data encodes rich scene information and thus enables recovery of simple scenes from sparse measurements. We investigate the feasibility of using a distributed set of few measurements (*e.g.*, as few as 15 pixels) to recover the geometry of simple parametric scenes with a strong prior, such as estimating the 6D pose of a known object. To achieve this, we design a method that utilizes both feed-forward prediction to infer scene parameters, and differentiable rendering within an analysis-by-synthesis framework to refine the scene parameter estimate. We develop hardware prototypes and demonstrate that our method effectively recovers object pose given an untextured 3D model in both simulations and controlled real-world captures, and show promising initial results for other parametric scenes. We additionally conduct experiments to explore the limits and capabilities of our imaging solution. Our project webpage is available at [cpsiff.github.io/recovering\\_parametric\\_scenes](https://cpsiff.github.io/recovering_parametric_scenes)

## 1. Introduction

Time-of-flight (ToF) cameras such as LiDARs are a key technology for modern 3D vision, enabling tasks like pose

estimation, shape reconstruction, and object recognition, with applications spanning fields such as robotics, augmented reality, and autonomous driving. Most current methods for inference from ToF imagery depend on dense 3D data, usually represented as a point cloud captured by high-resolution camera(s). It is generally accepted that a dense collection of depth measurements (*e.g.*, ToF pixels as points) is vital for precise 3D vision. While certain applications do require high-resolution geometry, can some vision tasks be accomplished with only sparse 3D measurements?

This question is particularly relevant given the recent emergence of low-cost (< \$3 USD per unit), miniature (< 5 mm across) ToF sensors [1, 30]. These sensors, already deployed in mobile [41] and wearable applications [9], are often implemented with a single photon avalanche diode (SPAD) array [31, 33], featuring very limited pixel counts (even a single pixel) yet a wide field-of-view (FoV) per pixel (*e.g.*, 30°). They capture raw ToF data with a *transient histogram*—a 1D waveform that encodes the intensity of light returning from the scene at pico-to-nanosecond timescales, integrated over a pixel's entire FoV. Traditionally, these histograms are processed into a point cloud by detecting and converting their peaks into depth estimates. However, this processing reduces the high-dimensional histogram to a single number, eliminating potentially useful information. Our key hypothesis, inspired by recent studies [3, 15, 24, 32], is that while these sensors cannot recover high-resolution point clouds, even a few transient histograms encode rich scene information sufficient for various downstream 3D vision tasks. An example is recovering scenes with low geometric complexity or scenarios where a strong geometric prior (*e.g.*, a low-dimensional parametric shape model) is available. Our question is, under these con-

<sup>†</sup>Co-first author

ditions, what is the minimal number of depth measurements required to recover 3D scenes?

As a first step towards addressing this question, we investigate the recovery of simple parametric 3D scenes using *very few* ToF measurements, each with a wide FoV (see Fig. 1), *e.g.*, as few as 15 pixels captured by spatially distributed, low-cost single-pixel SPAD sensors. We assume a 3D Lambertian scene defined by a parametric shape model and aim to recover the parameters of that model using a limited number of transient histograms captured from known fixed poses. We place special emphasis on the task of 6D pose estimation, which is a specific case of parametric scene recovery. In this case, the parameters that we aim to recover are the position and orientation of a known object mesh. We focus on 6D pose estimation because it is a well-defined problem with practical applications in robotics and augmented reality. This makes it a good testbed to tackle the fundamental challenge: utilizing very low resolution sensor data. With very few pixels, each of which integrates over a wide FoV, recovering 6D pose is challenging.

To solve this problem, we present an *analysis-by-synthesis* based approach. Our method integrates (1) a learning-based feedforward model which predicts an initial estimate of scene parameters; (2) a differentiable renderer that synthesizes sensor measurements given scene parameters in the parametric model; and (3) an optimization-based refiner that iteratively renders sensor measurements to optimize scene parameters using the differentiable renderer. To address the scarcity of real-world imaging data, we re-use our renderer to generate a large-scale synthetic dataset for training our feedforward model, and explore its ability to transfer to real-world captures.

We develop hardware prototypes for real-world capture and evaluate our approach using both simulated and real-world data. In real-world tests, our approach successfully estimates poses of even non-Lambertian objects using only 15 ToF pixels and an untextured object mesh. Moreover, leveraging our approach and hardware, we also briefly investigate two other forms of parametric scene recovery: parametric shape recovery (*i.e.*, the position and scale of a known spherical object), and human hand pose recovery (*i.e.*, pose and articulation), for which we demonstrate encouraging preliminary results in real-world settings.

**Scope and Limitations.** While our problem formulation is general, we focus on 6D pose estimation, with a limited exploration of two other forms of parametric scene recovery. Our main objective is to establish feasibility rather than present an immediately deployable solution. To simplify real-world experiments, we make key assumptions, such as Lambertian surfaces, co-located sensor and light source, and known sensor poses. While robustness to varying scene reflectance and imperfect sensor poses are assessed in our experiments, our approach and prototype are not yet prac-

tical for widespread use. Moreover, we rely on currently available consumer hardware, which has a restricted sensing range, limiting our experiments to tabletop scenes.

## 2. Related Work

**Time-of-flight (ToF) Imaging with SPADs.** A ToF camera emits light pulses and measures the return time of incident photons to estimate distance. SPAD sensors have increasingly been adopted for ToF imaging, typically combined with a co-located light source (*e.g.*, laser) [31]. This setup has been successfully applied to fluorescence lifetime imaging [38], novel view synthesis [29], and non-line-of-sight (NLOS) imaging [11, 14, 17, 18, 48]. Many of these systems rely on large, costly SPAD arrays (>\$10K USD) with high spatial and temporal resolution. Recent works have explored low-cost SPAD sensors (<\$3 USD) with limited pixel counts and lower temporal resolution for applications such as NLOS imaging [6, 50], shape reconstruction [32], human pose estimation [37], and SLAM [25]. Our work also explores low-cost SPAD sensors for ToF imaging; however, our primary focus is on the feasibility of using a minimal number of SPAD sensors for parametric scene recovery.

A key component in SPAD imaging is modeling the image formation process. Sifferman *et al.* [39] introduce a simple model for miniature ToF sensors. Recent works model the SPAD image formation process with differentiable functions for laboratory-grade [27, 28] or commodity [32] sensors, enabling gradient-based optimization and facilitating 3D tasks such as pose estimation and shape reconstruction. Our work modifies the sensor model in [32] to accommodate differentiable mesh rendering.

**3D Vision with Low-Cost SPADs.** Prior works have explored feedforward neural networks for inference from transient histograms. Pixels2Pose [37] proposes a learning-based method to estimate whole-body human pose from a single  $4 \times 4$  pixel transient histogram. DELTAR [24] and Jungerman *et al.* [15] both predict high resolution depth images from ToF transient(s) plus an RGB image. The neural networks in these prior works are generally trained on real-world data only, or on simulated data generated from a simple sensor model. In this work, we train on simulated data generated via a high-fidelity sensor model.

Recent works have considered an analysis-by-synthesis (AbS) paradigm, which uses differentiable rendering to align a set of underlying scene parameters with observed transient measurements. Sifferman *et al.* [39] design an AbS pipeline to recover 3DoF plane pose and albedo from a set of  $3 \times 3$  transient measurements. Mu *et al.* [32] present a method to recover arbitrary 3D scenes from a distributed set of >100 single-pixel transient measurements. Behari *et al.* [3] leverage AbS to reconstruct arbitrary 3D scenes from a miniature ToF sensor plus an RGB camera. Liu *et al.* [25] build a neural radiance field for dense SLAM by

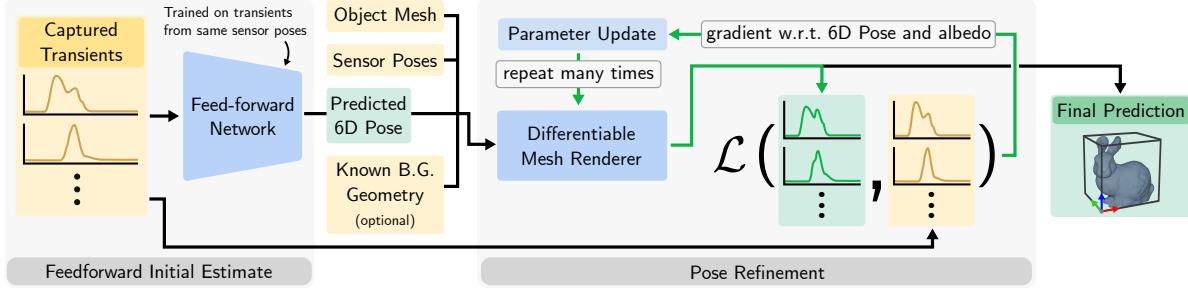


Figure 2. **Overview of our method** as applied to 6D pose estimation. Our method consists of two components: 1) a pose prediction module, where a feedforward network estimates initial object pose from a sparse set of input transient histograms; and 2) a pose refiner, where a differentiable renderer is integrated into an analysis-by-synthesis framework to iteratively optimize the pose estimates. Yellow boxes indicate inputs. Green boxes indicate (intermediate) outputs. The optimization loop is illustrated with green arrows.

integrating data from a miniature ToF sensor with an RGB camera. Luo *et al.* [27] reconstruct 3D scenes from transient measurements from very few viewpoints, however they use a high-resolution scanning LiDAR with higher fidelity and data rate than the miniature ToF sensors we consider.

Our work shares a similar goal to prior studies that use multiple sensors for 3D scene recovery. However, our focus is on leveraging strong geometric priors to push the limits of scene recovery using only a few low-fidelity sensors.

**6D Pose Estimation.** 6D pose estimation aims to determine the 6 degree-of-freedom pose of a known rigid object relative to the camera, given its 3D mesh model. Recent approaches use supervised learning to directly regress object pose from RGB and/or depth images [5, 20, 34, 46, 47]. The predicted pose can be further refined via an AbS pipeline [19, 20, 43, 46]. We take inspiration from the success of these works in designing our approach, integrating a feedforward network for initial pose prediction and an AbS pipeline for pose refinement.

### 3. Scene Recovery from a Few ToF Pixels

**Problem Formulation.** We aim to recover 3D geometry of a *Lambertian* scene specified by a set of parameters  $\mathbf{P}$  from a distributed set of  $n$  ToF sensors with known poses. We make two key assumptions regarding the *sensing setup* and *scene representation*. *First*, we assume that each ToF sensor operates via a co-located diffuse light source with a finite field-of-view, and reports a transient histogram  $\mathbf{h}$  which captures the intensity of light returning from the scene after a controlled pulse of illumination. This assumption covers a range of ToF sensors, including flash LiDAR and the low-cost SPAD sensor considered in this paper. *Second*, we utilize a mesh-based 3D scene representation, where the scene is modeled as a polygonal mesh composed of interconnected triangles that define its shape and surface. Mesh-based representations are widely used in graphics and many shape models are built on meshes [23, 26, 36, 49]. In this case,  $\mathbf{P}$  can be the 6 DoF pose of a 3D object mesh or parameters for a mesh-based shape model. Our goal is to esti-

mate  $\mathbf{P}$  from the set of measured histograms  $\{\mathbf{h}_i\}_{i=1}^n$ .

**Method Overview.** Our method consists of two steps: 1) given a set of input transient histograms  $\{\mathbf{h}_i\}_{i=1}^n$ , a feedforward network outputs a prediction  $\mathbf{P}_{\text{FF}}$  of the scene parameters, and 2) an analysis-by-synthesis based refiner takes  $\mathbf{P}_{\text{FF}}$  as an initial estimate, alongside camera pose and any other scene prior (*e.g.*, a parametric model), and iteratively optimizes  $\mathbf{P}_{\text{FF}}$  to minimize the difference between the measured histograms  $\{\mathbf{h}_i\}_{i=1}^n$  and histograms synthesized by our differentiable renderer. An illustration of our method as applied to the task of 6D pose estimation is shown in Fig. 2. In what follows we introduce the SPAD image formation process and present each component of our method.

#### 3.1. Background: Transient Formation Model

We utilize physics-based sensor modeling to accurately render the transient histograms  $\{\mathbf{h}_i\}_{i=1}^n$ . For each captured histogram, the laser source emits  $N_{\text{emit}}$  photons. Assuming that the source is co-located with the sensor at the origin  $\mathbf{o}$ , the rays of the emitted photons can be parametrized by a direction  $\omega$ . As in [15, 32, 39], we ignore high-order light paths and consider one-bounce paths only. Therefore, each photon travels from  $\mathbf{o}$  to a point on the scene  $\mathbf{x}$  and then back to  $\mathbf{o}$ , where  $\mathbf{x} = \mathbf{x}(\omega, \mathcal{M})$  is the first intersection between the ray in the direction of  $\omega$  and the scene  $\mathcal{M}$ .

Following prior work [15, 32], the expected number of photons  $N[i]$  received by the sensor within the  $i$ -th bin, in its angular integral form, is

$$N[i] = N_{\text{emit}} \int_{\Omega} I(\omega) \frac{\rho(\mathbf{x})}{\pi} \frac{\langle -\omega, \hat{n}(\mathbf{x}) \rangle}{\|\mathbf{x}\|^2} W\left(\frac{2\|\mathbf{x}\|}{c}, t_i\right) d\omega, \quad (1)$$

where  $\Omega$  is the space of solid angles within the FoV of the sensor.  $I(\omega)$  encodes the intensity of the laser along the direction  $\omega$ .  $\rho(\mathbf{x})$  is the albedo, and  $\hat{n}(\mathbf{x})$  is the normal of the surface at  $\mathbf{x}$ .  $t_i = i\Delta t$  corresponds to the time of the leading edge of the  $i^{\text{th}}$  bin, with  $\Delta t$  as the bin width. Lastly,

$$W(t, t_i) = \begin{cases} 1 & \text{if } t \in [t_i, t_i + \Delta t), \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

bins the photons. Due to hardware effects, the binning of photons is not perfect in reality.  $\mathbf{h}[i]$  in fact might also detect photon arrivals near but outside that bin, and is affected by the shape of the outgoing laser pulse. Therefore, we convolve the expected photon numbers with an empirically derived discrete jitter kernel  $\mathbf{s}$  to account for this effect

$$\mathbf{h}[i] = \sum_j N[i + \Delta i - j] \mathbf{s}[j], \quad (3)$$

where  $\Delta i$  is sensor-specific, constant temporal offset to the histogram, which needs to be calibrated.

**Pile-up Correction.** Previous works [13, 32] model the pile-up effect, which can significantly alter the measured histogram at high levels of ambient or active flux. To mitigate this effect, many existing sensors pre-process the histogram with algorithms like Coates' correction [8]. With this correction and under reasonable lighting conditions, the pile-up effect is often negligible. We thus do not include pile-up in our imaging model.

### 3.2. Differentiable Rendering

We numerically integrate Eq. (1) using the weighted sum

$$N[i] \approx \frac{N_{\text{emit}}}{\pi} \sum_{\omega \in \mathcal{W}} Q(\omega) I(\omega) \rho(\mathbf{x}) \frac{\langle -\omega, \hat{n}(\mathbf{x}) \rangle}{\|\mathbf{x}\|^2} W\left(\frac{2\|\mathbf{x}\|}{c}, t_i\right), \quad (4)$$

where  $\mathcal{W}$  is a specified set of rays, and  $Q(\omega)$  is the associated quadrature. To make the rendering differentiable, we calculate  $\partial \mathbf{x} / \partial \mathcal{M}$  and  $\partial \hat{n} / \partial \mathcal{M}$  using off-the-shelf differentiable rendering libraries. Specifically, we set  $\mathcal{W}$  to a  $h \times w$  grid of rays, fully covering the FoV and resembling the rendering of classical pixels, and the rasterization computes the rays'  $\mathbf{x}$ ,  $\hat{n}$ , and the gradients. Suppose that the center of the FoV is the z-axis and the imaging plane is  $z = 1$ , each pixel has area  $A_{\text{pixel}} = (2 \tan(FoV/2))^2 / (h \cdot w)$ . Assuming  $\omega = (\omega_x, \omega_y, \omega_z)$ , the quadrature  $Q(\omega)$  transforms the square pixel areas to solid angle differentials by

$$Q(\omega) = \frac{A_{\text{pixel}} \langle \omega, \hat{z} \rangle}{(1/\omega_z)^2} = \frac{4 \tan^2(FoV/2) \omega_z^3}{h \cdot w}. \quad (5)$$

The binning function  $W$  in Eq. (2) is discontinuous. We thus approximate it with the sigmoid function  $\sigma(x)$  by

$$W(t, t_i) = \sigma(k(t - t_i)) - \sigma(k(t - t_i - \Delta t)), \quad (6)$$

where  $k$  is a hand-picked constant to balance smoothness and realism. Further, the intensity map  $I(\omega)$  is discontinuous under an idealized diffuse laser source, since it provides uniform illumination within its FoV and zero illumination elsewhere. However, real-world lasers exhibit a non-uniform distribution, where intensity is highest at the center and gradually decreases toward the edges of the FoV. This allows us to approximate  $I(\omega)$  using a differentiable, spatially-varying function. We fit this function using real-world sensor properties in our experiments (see Sec. 4.2).

### 3.3. Feedforward Estimation of Scene Parameters

We learn a neural network  $f_\theta$  to predict initial scene parameters  $\mathbf{P}_{\text{FF}}$ , which will be further refined. This is given by

$$f_\theta(\{\mathbf{h}_i\}_{i=1}^n) \rightarrow \mathbf{P}_{\text{FF}}, \quad (7)$$

where  $\theta$  is the network weights learned from data,  $\{\mathbf{h}_i\}_{i=1}^n$  is the input of  $n$  transient histograms from ToF sensors. Namely, this feedforward network directly regresses the scene parameters based on sensor data.

**Network Architecture.** Specifically,  $f_\theta$  is a standard Transformer model [45]. The input transient histograms are first normalized, and then embedded using an MLP. These embeddings are added to positional embeddings, and further processed by a stack of Transformer blocks (4 in our implementation). The output embeddings are concatenated and fed into another MLP to predict scene parameters  $\mathbf{P}$ . This network is trained with full supervision, and the loss function varies depending on the scene parameterization used, as described in the supplement.

**Sim-to-Real Transfer.** A key challenge for training is the lack of real-world sensor data. We explore training on a large-scale synthetic dataset and transfer the learned model directly to real-world captures. This is made possible thanks to our efficient renderer in Sec. 3.2 and availability of 3D models [7]. We demonstrate strong results using this sim-to-real transfer in our experiments.

**Discussion.** Our network assumes fixed sensor poses and requires re-training for every sensor configuration. This design is highly tailored for our current hardware prototypes, yet can be easily extended to accommodate varying sensor poses, e.g., encoding sensor pose as part of the input [22].

### 3.4. Parameter Refinement

Given an estimate  $\mathbf{P}_{\text{FF}}$  of the scene parameters from the feedforward network and the differentiable renderer  $\mathcal{R}$ , we propose an analysis-by-synthesis approach to further refine  $\mathbf{P}_{\text{FF}}$ . This is done by directly optimizing  $\mathbf{P}$  to minimize the difference between the measured histograms  $\{\mathbf{h}_i\}_{i=1}^n$  and rendered histograms  $\mathcal{R}(\mathbf{P})$ , given by

$$\arg \min_{\mathbf{P}} \sum_{i=1}^n \|\mathcal{R}(\mathbf{P})_i - \mathbf{h}_i\|. \quad (8)$$

Since the renderer  $\mathcal{R}$  is fully differentiable, gradient descent can be used to solve this optimization locally. Specifically, this optimization starts from the initial estimate  $\mathbf{P}_{\text{FF}}$  and applies gradient steps until convergence. Since the rendering process  $\mathcal{R}$  is highly nonlinear, the quality of the solution depends on the accuracy of initial estimate  $\mathbf{P}_{\text{FF}}$ .

## 4. Experiments on 6D Pose Estimation

To adopt our method for 6D pose recovery, we set the scene parameters  $\mathbf{P}$  to a rotation  $\mathbf{R}$  and translation  $\mathbf{T}$  which trans-

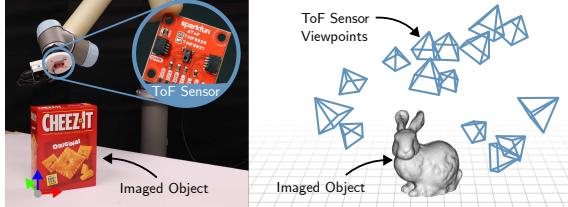


Figure 3. **Left:** Illustration of our capture setup, where a ToF sensor is mounted on a robot arm and moved between a set of positions. **Right:** 15 sensor poses used for our experiments.

form a known object mesh to its position in a global coordinate frame. We evaluate our method for 6D pose estimation in simulation and on real-world captures.

#### 4.1. Hardware and Real-world Capture

**Hardware Prototype.** Our imaging system assumes known relative position of the ToF sensors. In practical deployment, this may be achieved by placing multiple sensors in a stationary position in the environment, or by attaching them each to a rigid object, like a mobile device. To allow flexibility for our experiments, we instead place a single sensor on an industrial robot arm, and move the sensor to multiple positions by controlling the robot while the scene remains static. We rely on the robot’s kinematics, which are quoted as repeatable to  $\pm 0.1\text{mm}$  [44], to record sensor pose.

We use the AMS TMF8820 SPAD-based ToF sensor. Like other sensors of its class, the sensor is very small ( $12.8\text{mm}^3$ ,  $< 1\text{g}$ ) and low-power ( $< 100\text{mW}$ ) [1]. The illumination source is an integrated low-power 940nm VCSEL laser. We operate the sensor in “low range, high accuracy mode,” and 4 million iterations per measurement, giving it a maximum range of 1.5m and a bin size equivalent to  $\sim 1.4\text{cm}$ . We interface with the sensor via an attached microcontroller, which forwards transient histogram measurements from the sensor to a connected computer.

**Capture Setup.** We sample random sensor positions between 30cm and 80cm from the workspace center (within the range of TMF8820), and random orientations which face the camera to the workspace center ( $\pm 15^\circ$ ). For a fair comparison, the same 15 randomly sampled sensor poses are used for all real-world experiments. This number (15) allows for practical data capture, and was chosen to strike a balance between estimation accuracy and information sparsity based on our simulation results (see Sec. 4.4). An Intel RealSense D405 depth-from-stereo camera is affixed next to the ToF sensor for ground truth capture. We utilize the merged point cloud from the depth camera views alongside ICP [4] and manual registration to generate ground truth object poses. We measure the geometry of the known background (the tabletop) by touching the robot to the surface at multiple points. This capture setup is shown in Fig. 3.

**Data Capture.** We capture each object at 25 poses (10 for the highly symmetric basketball, softball, and tennis ball)

by manually positioning the object such that the object center is within 15cm of the workspace center. Effort is made to distribute the object poses uniformly within the workspace and to vary the object orientation uniformly. Because the objects are placed on a tabletop, we are restricted to orientations that provide stable support on the surface.

#### 4.2. Implementation Details

**TMF8820 Modeling.** The TMF8820 reports transient histograms for nine separate fields-of-view, called “zones”, each of which correspond to different sets of pixels on the SPAD array. Significant bloom artifacts are present between zones, and the exact zone dimensions are poorly defined [3, 39]. To address this challenge, and in line with our focus on very-low-pixel-count regimes, we aggregate the zones into one by summing across the zone dimension, yielding a single 128-bin histogram per sensor measurement. This approach is consistent with prior work [32, 40].

Further, in the TMF8820 datasheet, the laser illumination is reported as non-uniform across the FoV [1]  $I(\omega)$ . We therefore approximate the intensity map using

$$I(\omega) = K_1 \exp(-K_2(\omega_x^2 + \omega_y^2) - K_3(\omega_x^4 + \omega_y^4)). \quad (9)$$

We calibrate the constants  $K$  to match the intensity map shown in the TMF8820 datasheet. A visualization of  $I(\omega)$  is included in the supplement (Fig. D). We find that pile-up is not very apparent even at high ambient light levels.

**Jitter Kernel.** The TMF8820 reports the shape of the outgoing laser pulse in a “reference histogram” alongside each measurement, which is measured from a SPAD sensor inside the laser cavity. Because this reference histogram is itself captured by a SPAD sensor, it encapsulates the shape of the outgoing laser pulse and the temporal response function of the SPAD itself. We make use of this histogram as the jitter kernel  $s$  in our imaging model.

**Sensor Calibration.** The reference histogram reported by the TMF8820 is not reported at the same temporal resolution as the transient histogram [32, 39]. We resample the reference histogram by a factor of  $s_{\text{scale}}$  before use in our imaging model. Additionally, the temporal resolution  $\Delta t$  and a constant temporal offset to the histogram  $\Delta i$  are not known. Following prior work, we recover the parameters  $s_{\text{scale}}$ ,  $\Delta t$ , and  $\Delta i$  by calibrating on some set of reference captures of a planar surface with known geometry. To do so, we keep scene geometry fixed, and optimize the sensor parameters to minimize the loss between captured and rendered histograms, akin to Eq. (8).

**Feedforward Network.** We train the network described in Sec. 3.3 to predict 6D pose. For non-symmetric objects, we use a combination of rotation loss, translation loss, and a point matching loss. For symmetric objects, we use the ADD-S loss [47]. See the supplement (Sec. A) for details

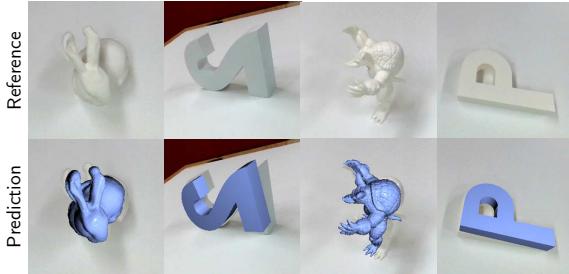


Figure 4. **Visualization** of 6D pose estimation results with 3D-printed objects using our method (feedforward + refiner). For each object, the pose prediction with the median pose error (ADD) over the 25-capture dataset is shown.

of the loss functions and training parameters. A single-instance forward pass takes  $\sim 4.6$ ms on Nvidia RTX 4080.

**Pose Refinement.** We use adaptive gradient descent following Adam [16] to solve Eq. (8). We set the step size (*i.e.*, learning rate) to 0.01 for  $\mathbf{R}$  and 0.001 for  $\mathbf{T}$ . Additionally, we optimize the albedo of both the object ( $\rho_{\text{obj}}$ ) and the planar surface ( $\rho_{\text{plane}}$ ). These albedos are not predicted by the feedforward network; instead they are initialized to 1 and further optimized. We find empirically that optimizing albedos improves performance. We set the number of optimization steps to 200. We represent  $\mathbf{R}$  using the 6D rotation representation proposed in [51]. Differentiable rendering is implemented via Nvdiffrast [21] and PyTorch [2].

### 4.3. Experiment Protocol

**Datasets.** We evaluate our method for 6D pose estimation on two sets of objects: 1) 3D printed test objects and 2) seven readily available objects from the YCB dataset [7] — a standard benchmark for 6D pose recognition. See the supplement (Fig. I) for images of the objects. For YCB objects, the high-resolution “Google 16k” meshes provided by the YCB dataset are used for data generation and as input to the refiner. A child’s basketball is used in place of the child’s soccer ball in the YCB object set, along with a manually constructed spherical mesh.

**Synthetic Training Data.** We generate synthetic data with our renderer to train the feedforward model. For each object, we synthesize 200K samples by limiting the object center within 15cm of the workspace center. Object orientations are randomly sampled. To ensure physical plausibility, the object height is adjusted so that at least one vertex of the mesh lies on the planar surface, preventing the object from appearing to float in space, though this configuration may not correspond to a stable resting pose. This setup imposes a conservative prior on object placement. The planar surface is included in the scene when rendering synthetic data. We also perform domain randomization [42]; we add Gaussian noise with a standard deviation of 1.5cm to the sensor positions independently for each sample, and vary the albedo of the object and the planar surface.



Figure 5. **Visualization** of 6D pose estimation results with objects from the YCB dataset using our method (feedforward + refiner). For each object, the pose prediction with the median pose error (ADD-S) over the 25-capture dataset is shown.

**Evaluation Metrics.** We follow standard metrics [47] for evaluation, including ADD for non-symmetric objects, and ADD-S for symmetric objects. ADD captures the average distance between corresponding points on the predicted and ground truth object. ADD-S captures the average distance between a point on the predicted object and *the nearest* point on the ground truth object. We report the AUC-ADD(-S) in order to capture the distribution of scores over the entire dataset, with the maximum threshold in calculating AUC set to 10cm. Note that ADD(-S) score is highly dependent on the scale and geometry of a specific object, thus should not be compared between objects.

**Point Cloud Baselines.** We implement a point-cloud based baseline which represents an upper bound on point-cloud based system performance. To avoid pitfalls of any one particular depth camera, we simulate *idealized* sensor measurements; for each sensor pixel we project rays to get points of intersection with scene geometry, which are combined to form a point cloud. Points which lie on the planar surface are removed. The result is a noise-free point cloud of points on the object mesh. We then use ICP [4] to align the object mesh to this point cloud. To maximize the chance of success, the ground truth pose is used as initialization for ICP. Therefore, this baseline represents the best possible performance of a vanilla ICP-based registration system. We consider two variants: single-pixel, in which the same number of imaging pixels are used as in our system (one per view), and  $16^2$ , in which a  $16 \times 16$  point cloud sampled from a pixel grid spanning an FoV matching the diffuse sensor is generated *per view* (yielding  $256 \times$  imaging pixels of our system).

**RGB(D) Baselines.** For reference, we compare to two recent deep models: FoundPose [52], which uses single-view RGB input, and FoundationPose [46], which uses single-view RGB-D input. For both methods, the RGB image from viewpoint 15 is used as it provides a wide view. For FoundationPose, a high resolution depth camera view is simulated from the ground truth object pose.

Data	Total Pixels	Method	Crackers	Mustard	Chips	AUC-ADD-S ( $\uparrow$ )				Softball	Mean
						SPAM	Basketball	Tennis Ball			
Sim.	15	1 Px Point Cloud <sup>†</sup>	78.36	82.12	77.83	85.07	82.92	88.09	86.36	82.96	
Real	15	Ours: Feedforward	88.02	90.04	88.38	<b>90.04</b>	95.15	<b>96.26</b>	<b>94.95</b>	91.83	
Real	15	Ours: FF + Refiner	<b>90.04</b>	<b>90.07</b>	<b>88.50</b>	90.00	<b>95.76</b>	96.06	<b>94.95</b>	<b>92.20</b>	
Sim.	3840	16 <sup>2</sup> Point Cloud <sup>†</sup>	95.17	97.23	97.23	97.19	97.67	97.57	97.37	97.06	
Real	407K	Single-View RGB [52]	60.71	87.93	40.12	58.95	65.46	77.68	72.42	66.18	
Real	407K	Single-View RGB-D* [46]	90.49	92.10	92.54	93.80	94.24	86.67	94.24	92.01	

Table 1. **Results of 6D pose estimation with (symmetric) objects from the YCB object set.** gray: methods using additional pixels; <sup>†</sup>: methods using oracle ground-truth pose; \*: methods using simulated high-resolution point cloud data. See details in Sec. 4.3.

Data	Total Pixels	Method	Two	AUC-ADD ( $\uparrow$ )				Mean
				P	L	Bunny	Armadillo	
Sim.	15	1 Px Point Cloud <sup>†</sup>	73.87	59.65	55.11	56.14	53.89	59.73
Real	15	Ours: Feedforward	74.67	77.31	69.11	67.78	65.93	70.96
Real	15	Ours: FF + Refiner	<b>83.47</b>	<b>84.94</b>	<b>77.75</b>	<b>77.68</b>	<b>79.71</b>	<b>80.71</b>
Sim.	3840	16 <sup>2</sup> Point Cloud <sup>†</sup>	97.55	96.18	95.17	96.18	96.90	96.39
Real	407K	Single-View RGB [52]	56.99	65.40	51.51	64.63	86.28	64.96
Real	407K	Single-View RGB-D* [46]	86.57	85.51	82.72	88.30	87.58	86.14

Table 2. **Results of 6D pose estimation with (non-symmetric) 3D printed objects.** gray: methods using additional pixels; <sup>†</sup>: methods using oracle ground-truth pose; \*: methods using simulated high-resolution point cloud data. See details in Sec. 4.3.

#### 4.4. 6D Pose Estimation Results

**Main Results.** Our main results are presented in Tab. 1 and Tab. 2. Visualization of sample results with median pose errors are shown in Fig. 4 and Fig. 5. In all cases, our method outperforms the best-case performance of using a single-pixel point cloud. Additionally, our refiner improves performance in most cases. We also notice that our refiner yields a much larger performance improvement over the feedforward network on 3D printed objects. We hypothesize that this is due to the ambiguity of symmetric objects, and the symmetric loss can easily converge to a local minimum. Despite reasonable metrics, the SPAM is a failure case for our method; because the object is very small and near-symmetric along many axes, a high ADD-S score can be achieved by matching object translation. However, qualitative results (Fig. 5) show that orientation is not predicted reliably. Our method approaches the performance of the RGB-D baseline, while exceeding the performance of the RGB baseline which struggles due to the lack of metric depth information and/or lack of object texture.

**Varying Viewpoint Count.** In simulation, we evaluate the performance of our method with varying number of views. We compare to the point cloud-based baseline described in Sec. 4.3. We evaluate on synthetic data of the 25 real “2” poses. Camera poses are sampled via the same process as real poses ( Sec. 4.1). Results are shown in the rightmost panel of Fig. 1. See the supplement for full results (Tab. D). Our approach exceeds the performance of the point cloud-based baseline for 5-100 total pixels. The point cloud-based method suffers with very few input pixels because, despite some variation in sensor orientation, the poses do not achieve good coverage of the scene.

**Additional Results.** Due to space limit, the following are included in the supplement: (1) results with varying scene reflectance and ambient light; (2) ablations on the sensor model and feedforward network training; and (3) experiments on sensor interference.

### 5. Exploration Beyond 6D Pose Estimation

#### 5.1. Size and Position of Spherical Objects

We experiment with recovering the size and location of a sphere resting on a planar surface. We use an identical method to that used for 6D pose estimation (Sec. 4), except the predicted parameters  $\mathbf{P}$  consist of the center point and diameter of a sphere, rather than the rotation and translation of a known mesh. Both parameters are predicted by the feedforward network and optimized during refinement. We evaluate on our pre-existing captures of the basketball, softball, and tennis ball objects. The results of this experiment are shown in Tab. 3. We find that our method can recover the position and diameter of the sphere with an average error of  $< 1\text{cm}$  in all cases, with often  $< 0.5\text{cm}$  of error, despite the temporal resolution of an individual SPAD being  $\sim 1.4\text{cm}$ .

#### 5.2. Human Hand Pose

We recover human hand pose (absolute pose and articulation) from a ring of eight sensors encircling the wrist, as shown in Fig. 6. We modify the feedforward prediction to include pose, shape, global translation, and rotation of the human hand. The hand pose and shape is represented using the parameters of the MANO hand model [36].

**Data Capture.** As an initial feasibility study, we gather 250 measurements of a single individual’s hand from the

Object	Mean Error in Diameter (cm) (↓)	Mean Error in Position (cm) (↓)
Basketball	0.35 (1.9%)	0.84
Softball	0.28 (2.9%)	0.24
Tennis Ball	0.33 (4.8%)	0.39

Table 3. Results of recovering position / size of spherical objects.

Method	PA-MPJPE (mm) (↓)
ToF-based Prior Work <sup>†</sup> [10]	11.96
Trained on Sim. Data Only (ours)	19.56
Trained on Real Data Only (ours)	9.98
P.T. on Sim., F.T. on Real (ours)	8.18
RGB-Based Method <sup>†</sup> [35]	6.0

Table 4. Results of hand pose and shape estimation. <sup>†</sup>Related works are provided for context only; metrics are over a different dataset and should not be directly compared.

ring of sensors. To sample hand poses, we prompt the user to match their hand pose to a random hand pose selected from the DART dataset [12]. RGB-D cameras are mounted above and below the hand to capture ground truth, which is provided by the RGB-based method HaMeR [35], and aligned to a fused point cloud from the two depth maps via ICP [4]. We reserve 50 captures for testing.

**Results.** We report Procrutes aligned mean per joint position error (PA-MPJPE), a standard metric for hand tracking [10, 35]. PA-MPJPE captures the average distance between corresponding joints in the predicted and ground truth hand pose. The results of our experiment are shown in Tab. 4. We find that, in this setting, training on simulated data alone yields unsatisfactory results. A closer inspection reveals that the simulated histograms become inaccurate at distances below 15cm. We attribute this to unmodeled sensor effects, such as unmodeled effects such as gating and/or pile-up from the high intensity of returning light. While we attempted to mitigate this issue by modeling these effects and learning a custom gating function, these efforts did not lead to improved performance. For the same reason, our refiner module is also not effective for hand pose estimation.

We observe significantly improved results when training on real data, with the best results achieved through a two-stage process: pre-training on simulated data followed by fine-tuning on real data. Despite the limited realism of the simulated data, pre-training still provides benefits, likely because the network is able to learn transferable high-level features. To contextualize our results, we include comparisons to related works in Tab. 4. However, results from prior works are based on different datasets and experimental conditions, and thus are not directly comparable.

## 6. Discussion

Our work demonstrate that a few (*e.g.*, 15) diffuse ToF pixels are sufficient to recover simple scene geometry. More-

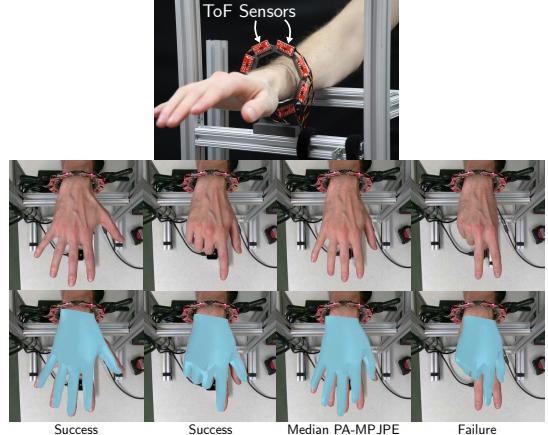


Figure 6. **Top:** Setting for hand pose capture, in which eight ToF sensors encircle the wrist. **Bottom:** Visualization of results of applying our method to hand pose estimation, corresponding to method "Pretrained on Sim., F.T. on Real" in Tab. 4.

over, we showcased the potential of our approach for more complex geometry, including recovering the position and scale of a spherical object, and human hand pose estimation. Our work offers an initial step toward enabling a range of practical applications and open up several promising directions for future research.

**Practical Implications.** While still in its early stage, our approach has great potential for 3D vision applications that benefit from low-cost, low-power, and distributed sensing. One promising application domain is *wearable computing*. Inspired by our experiments on hand tracking, we envision that an array of miniature ToF sensors could be deployed in head-mounted or wrist-worn devices to track a user’s body motion (*e.g.*, arm and hand pose), enabling gesture-based user interfaces. Another key application domain is *robotics*. Imagine a robotic arm or drone equipped with a distributed array of lightweight, energy-efficient ToF sensors. These sensors could function as a network of spatially distributed cameras, reconstructing the environment from an inside-out perspective, enhancing tasks like grasping, navigation, and human-robot interaction.

**Future Directions.** Our work demonstrates the estimation of 6D pose of rigid objects in a tabletop setting. Future work should aim to improve robustness to environmental factors such as ambient light and varied surface reflectance. This could be achieved by explicitly modeling these factors or developing methods that are inherently invariant to them. Additionally, future work should explore recovery of more complex scene geometries at larger scales, *e.g.*, multiple deformable, articulated objects in room- or playground-sized environments. A promising future direction is learning from large-scale synthetic data. Encouragingly, our results have demonstrated that effective sim-to-real transfer is possible with ToF histogram data. We are hopeful that large-scale synthetic data can be applied to a range of inference tasks.

**Acknowledgements:** This work was supported in part by the National Science Foundation under Grant Numbers 2152163 (NRT), 2333491 (CPS Frontier), 2442739 (CA-REER), 1943149 (CAREER), 2107060 (CNS Core), by ARL under contract number W911NF-2020221, and by ONR grant N000142412155.

## References

- [1] AMS OSRAM AG. *TMF882X Datasheet*. AMS OSRAM AG. [1](#), [5](#)
- [2] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsich, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanian, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, page 929–947, New York, NY, USA, 2024. Association for Computing Machinery. [6](#)
- [3] Nikhil Behari, Aaron Young, Siddharth Somasundaram, Tzofi Klinghoffer, Akshat Dave, and Ramesh Raskar. Blurred LiDAR for Sharper 3D: Robust Handheld 3D Scanning with Diffuse LiDAR and RGB, 2024. arXiv:2411.19474 [eess]. [1](#), [2](#), [5](#)
- [4] P.J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. [5](#), [6](#), [8](#)
- [5] Dingding Cai, Janne Heikkia, and Esa Rahtu. OVE6D: Object Viewpoint Encoding for Depth-based 6D Object Pose Estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6793–6803, New Orleans, LA, USA, 2022. IEEE. [3](#)
- [6] Clara Callenberg, Zheng Shi, Felix Heide, and Matthias B. Hullin. Low-cost SPAD sensing for non-line-of-sight tracking, material classification and depth imaging. *ACM Transactions on Graphics*, 40(4):1–12, 2021. [2](#)
- [7] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB object and Model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 510–517, 2015. [4](#), [6](#)
- [8] PB Coates. The correction for photopile-up’ in the measurement of radiative lifetimes. *Journal of Physics E: Scientific Instruments*, 1968. [4](#)
- [9] Dan O’shea. UK start-up Singular Photonics touts SPAD sensors, Meta collab, 2025. Accessed: 2025-03-07. [1](#)
- [10] Nathan Devrio and Chris Harrison. DiscoBand: Multiview Depth-Sensing Smartwatch Strap for Hand, Body and Environment Tracking. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13, New York, NY, USA, 2022. Association for Computing Machinery. [8](#)
- [11] Daniele Faccio, Andreas Velten, and Gordon Wetzstein. Non-line-of-sight imaging. *Nature Reviews Physics*, 2(6):318–327, 2020. [2](#)
- [12] Daiheng Gao, Yuliang Xiu, Kailin Li, Lixin Yang, Feng Wang, Peng Zhang, Bang Zhang, Cewu Lu, and Ping Tan. DART: Articulated Hand Model with Diverse Accessories and Rich Textures, 2022. arXiv:2210.07650 [cs]. [8](#)
- [13] Anant Gupta, Atul Ingle, Andreas Velten, and Mohit Gupta. Photon-Flooded Single-Photon 3D Cameras. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6763–6772, Long Beach, CA, USA, 2019. IEEE. [4](#), [5](#)
- [14] Felix Heide, Matthew O’Toole, Kai Zang, David B Lindell, Steven Diamond, and Gordon Wetzstein. Non-line-of-sight imaging with partial occluders and surface normals. *ACM Transactions on Graphics (ToG)*, 38(3):1–10, 2019. [2](#)
- [15] Sacha Jungerman, Atul Ingle, Yin Li, and Mohit Gupta. 3D Scene Inference from Transient Histograms. In *Computer Vision – ECCV 2022*, pages 401–417. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science. [1](#), [2](#), [3](#)
- [16] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, 2017. arXiv:1412.6980 [cs]. [6](#)
- [17] Ahmed Kirmani, Tyler Hutchison, James Davis, and Ramesh Raskar. Looking around the corner using transient imaging. In *2009 IEEE 12th International Conference on Computer Vision*, pages 159–166. IEEE, 2009. [2](#)
- [18] Tzofi Klinghoffer, Xiaoyu Xiang, Siddharth Somasundaram, Yuchen Fan, Christian Richardt, Ramesh

- Raskar, and Rakesh Ranjan. Platonerf: 3d reconstruction in plato’s cave via single-view two-bounce lidar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14565–14574, 2024. [2](#)
- [19] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 954–962, Santiago, Chile, 2015. IEEE. [3](#)
- [20] Yann Labb  , Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare, 2022. arXiv:2212.06870 [cs]. [3](#)
- [21] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. [6](#)
- [22] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *arXiv preprint arXiv:2507.10496*, 2025. [4](#)
- [23] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. [3](#)
- [24] Yijin Li, Xinyang Liu, Wenqi Dong, Han Zhou, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. DELTAR: Depth Estimation from a Light-Weight ToF Sensor and RGB Image. In *Computer Vision – ECCV 2022*, pages 619–636. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science. [1, 2](#)
- [25] Xinyang Liu, Yijin Li, Yanbin Teng, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2023. [2](#)
- [26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. [3](#)
- [27] Weihan Luo, Anagh Malik, and David B. Lindell. Transientangelo: Few-Viewpoint Surface Reconstruction Using Single-Photon Lidar, 2024. arXiv:2408.12191. [2, 3](#)
- [28] Anagh Malik, Parsa Mirdehghan, Sotiris Nousias, Kyros Kutulakos, and David Lindell. Transient Neural Radiance Fields for Lidar View Synthesis and 3D Reconstruction. *Advances in Neural Information Processing Systems*, 36:71569–71581, 2023. [2](#)
- [29] Anagh Malik, Noah Juravsky, Ryan Po, Gordon Wetzstein, Kirriakos N Kutulakos, and David B Lindell. Flying with photons: Rendering novel views of propagating light. In *European Conference on Computer Vision*, pages 333–351. Springer, 2024. [2](#)
- [30] ST Microelectronics. *VL6180X Proximity and Ambient Light Sensing Module Datasheet*. ST Microelectronics. [1](#)
- [31] Kazuhiro Morimoto, Andrei Ardelean, Ming-Lo Wu, Arin Can Ulku, Ivan Michel Antolovic, Claudio Bruschini, and Edoardo Charbon. Megapixel time-gated SPAD image sensor for 2D and 3D imaging applications. *Optica*, 7(4):346, 2020. [1, 2](#)
- [32] Fangzhou Mu, Carter Sifferman, Sacha Jungerman, Yiquan Li, Mark Han, Michael Gleicher, Mohit Gupta, and Yin Li. Towards 3D Vision with Low-Cost Single-Photon Cameras. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5302–5311, Seattle, WA, USA, 2024. IEEE. [1, 2, 3, 4, 5](#)
- [33] C. Niclass, A. Rochas, P.-A. Besse, and E. Charbon. Design and characterization of a CMOS 3-D image sensor based on single photon avalanche diodes. *IEEE Journal of Solid-State Circuits*, 40(9):1847–1854, 2005. Conference Name: IEEE Journal of Solid-State Circuits. [1](#)
- [34] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7667–7676, 2019. arXiv:1908.07433 [cs]. [3](#)
- [35] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing Hands in 3D with Transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9826–9836, Seattle, WA, USA, 2024. IEEE. [8](#)
- [36] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. [3, 7, 1](#)
- [37] Alice Ruget, Max Tyler, Germ  n Mora-Mart  n, Sterling Scholes, Feng Zhu, Istvan Gyongy, Brent Hearn, Steve McLaughlin, Abderrahim Halimi, and Jonathan Leach. Pixels2Pose: Super-Resolution Time-of-Flight Imaging for 3D Pose Estimation. In *Imaging and Applied Optics Congress 2022 (3D, AOA, COSI, ISA, pcaOP)*, page ITh5D.5, Vancouver, British Columbia, 2022. Optica Publishing Group. [2](#)

- [38] David Eric Schwartz, Edoardo Charbon, and Kenneth L Shepard. A single-photon avalanche diode array for fluorescence lifetime imaging microscopy. *IEEE journal of solid-state circuits*, 43(11):2546–2557, 2008. 2
- [39] Carter Sifferman, Yeping Wang, Mohit Gupta, and Michael Gleicher. Unlocking the Performance of Proximity Sensors by Utilizing Transient Histograms. *IEEE Robotics and Automation Letters*, 8(10):6843–6850, 2023. 2, 3, 5
- [40] Carter Sifferman, William Sun, Mohit Gupta, and Michael Gleicher. Using a Distance Sensor to Detect Deviations in a Planar Surface. *IEEE Robotics and Automation Letters*, 9(10):8515–8522, 2024. Conference Name: IEEE Robotics and Automation Letters. 5
- [41] TechInsights. iPhone 15 Pro Max Rear LiDAR Camera Process Flow Analysis, 2023. Accessed: 2025-03-07. 1
- [42] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 6
- [43] Jonathan Tremblay, Bowen Wen, Valts Blukis, Balakumar Sundaralingam, Stephen Tyree, and Stan Birchfield. Diff-DOPE: Differentiable Deep Object Pose Estimation, 2023. arXiv:2310.00463 [cs]. 3
- [44] Universal Robots. Ur5 technical specifications. Online, 2016. Accessed: 2025-02-28. 5
- [45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4
- [46] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17868–17879, Seattle, WA, USA, 2024. IEEE. 3, 6, 7
- [47] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes, 2018. arXiv:1711.00199 [cs]. 3, 5, 6, 1
- [48] Shumian Xin, Sotiris Nousias, Kiriakos N. Kutulakos, Aswin C. Sankaranarayanan, Srinivasa G. Narasimhan, and Ioannis Gkioulekas. A Theory of Fermat Paths for Non-Line-Of-Sight Shape Reconstruction. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 6800–6809, 2019. 2
- [49] Wenqiang Xu, Zhenjun Yu, Han Xue, Ruolin Ye, Siqiong Yao, and Cewu Lu. Visual-Tactile Sensing for In-Hand Object Reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8803–8812, Vancouver, BC, Canada, 2023. IEEE. 3
- [50] Aaron Young, Nevindu M. Batagoda, Harry Zhang, Akshat Dave, Adithya Pediredla, Dan Negrut, and Ramesh Raskar. Enhancing Autonomous Navigation by Imaging Hidden Objects using Single-Photon LiDAR, 2024. arXiv:2410.03555 [cs]. 2
- [51] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5738–5746, 2019. ISSN: 2575-7075. 6, 1
- [52] Evin Pınar Örnek, Yann Labb  , Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. FoundPose: Unseen Object Pose Estimation with Foundation Features. In *Computer Vision – ECCV 2024*, pages 163–182. Springer Nature Switzerland, Cham, 2025. Series Title: Lecture Notes in Computer Science. 6, 7

# Recovering Parametric Scenes from Very Few Time-of-Flight Pixels

## Supplementary Material

In this supplementary material, we provide (1) a description of loss functions for training our feedforward model (Sec. A); (2) additional results on 6D pose estimation (Sec. B); (3) an analysis of runtime and complexity (Sec. C); (4) experiments and discussion on sensor interference (Sec. D); and (5) additional visualization of our results on 6D pose estimation (Sec. E).

For sections, figures and equations, we use numbers (*e.g.*, Sec. 1) to refer to the main paper and capital letters (*e.g.*, Sec. A) to refer to this supplement.

### A. Training Loss of Feedforward Models

#### A.1. 6D Pose Estimation

As described in Sec. 4.2, we utilize one of two losses to train the feedforward model depending on if the object is symmetrical. For non-symmetrical objects, we utilize a combination rotation, translation, and point matching loss. Given a ground truth object rotation  $\mathbf{R}_{gt}$  (represented by the 6D representation proposed by [51]) and translation  $\mathbf{t}_{gt}$ . Given a set of 3D points  $\mathbf{x}_i$  on the object, the loss of the predicted rotation  $\mathbf{R}$  and translation  $\mathbf{t}$  is given by:

$$\mathcal{L} = \lambda_r \mathcal{L}_{rot} + \lambda_t \mathcal{L}_{trans} + \lambda_p \mathcal{L}_{pm}$$

where the loss terms are given by

$$\begin{aligned}\mathcal{L}_{rot} &= \|\mathbf{R} - \mathbf{R}_{gt}\|_1, \\ \mathcal{L}_{trans} &= \|\mathbf{t} - \mathbf{t}_{gt}\|_1, \\ \mathcal{L}_{pm} &= \frac{1}{N} \sum_{i=1}^N \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - (\mathbf{R}_{gt}\mathbf{x}_i + \mathbf{t}_{gt})\|_2\end{aligned}$$

We set  $\lambda_r = 1.0$ ,  $\lambda_t = 0.5$ ,  $\lambda_p = 0.1$  for our experiments.

For symmetric objects, we use ADD-S loss introduced in [47], where  $\mathcal{X}$  represents the set of object points:

$$\mathcal{L}_{ADD-S} = \frac{1}{N} \sum_{i=1}^N \min_{\mathbf{x}_j \in \mathcal{X}} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - (\mathbf{R}_{gt}\mathbf{x}_j + \mathbf{t}_{gt})\|_2$$

#### A.2. Spherical Object Recovery

For spherical object recovery (Sec. 5.1), the scene is parameterized by the center point  $\mathbf{c} \in \mathbb{R}^3$  and diameter  $d$ . Our loss function is a simple combination of error in the two components:

$$\mathcal{L} = \|\mathbf{c} - \mathbf{c}_{gt}\| + \lambda |d - d_{gt}|$$

We set  $\lambda = 1$  for our experiments.

### A.3. Human Hand Pose Estimation

For hand pose estimation (Sec. 5.2), we predict the MANO model [36] shape parameters  $\beta$ , pose parameters  $\theta$ , global 3D rotation  $\mathbf{R}$  (represented by the 6D representation proposed by [51]), and global 3D translation  $\mathbf{t}$ . The loss for a given prediction is given by:

$$\begin{aligned}\mathcal{L} = \lambda_s \mathcal{L}_{shape} + \lambda_p \mathcal{L}_{pose} + \lambda_r \mathcal{L}_{rot} \\ + \lambda_t \mathcal{L}_{trans} + \lambda_j \mathcal{L}_{joint} + \lambda_v \mathcal{L}_{vertex}\end{aligned}$$

where the loss terms are given by

$$\begin{aligned}\mathcal{L}_{shape} &= \|\beta - \beta_{gt}\|_1, \\ \mathcal{L}_{pose} &= \|\theta - \theta_{gt}\|_1, \\ \mathcal{L}_{rot} &= \|\mathbf{R} - \mathbf{R}_{gt}\|_1, \\ \mathcal{L}_{trans} &= \|\mathbf{t} - \mathbf{t}_{gt}\|_1, \\ \mathcal{L}_j &= \|(\mathbf{R}\mathcal{M}_j(\beta, \theta) + \mathbf{t}) - (\mathbf{R}_{gt}\mathcal{M}_j(\beta_{gt}, \theta_{gt}) + \mathbf{t}_{gt})\|_2, \\ \mathcal{L}_v &= \|(\mathbf{R}\mathcal{M}_v(\beta, \theta) + \mathbf{t}) - (\mathbf{R}_{gt}\mathcal{M}_v(\beta_{gt}, \theta_{gt}) + \mathbf{t}_{gt})\|_2\end{aligned}$$

Where  $\mathcal{M}_j$  is the MANO model that outputs joint keypoint positions, and  $\mathcal{M}_v$  is the MANO model that outputs mesh vertex positions. We set  $\lambda_s = 0.1$ ,  $\lambda_p = 0.1$ ,  $\lambda_r = 1.0$ ,  $\lambda_t = 1.0$ ,  $\lambda_j = 0.1$ ,  $\lambda_v = 0.1$  for our experiments.

### B. Additional 6D Pose Estimation Experiments

#### B.1. Data Visualization

We visualize the transient histograms captured by multiple, distributed ToF sensors across two different 3D scenes in Fig. A. The measurement has a complex relationship with scene geometry. We aim to solve the inverse problem (multi-view transient histogram  $\rightarrow$  geometry) for simple parametric scenes.

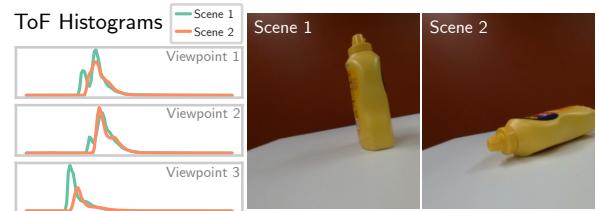


Figure A. Transient histograms from multiple viewpoints alongside corresponding 3D scenes.

#### B.2. Fine-Tuning on Real Data

We investigate the effects of fine-tuning our feedforward model on real data. To do so, we capture 80 additional measurements of the matte white “2” object used in prior

Training Data	AUC-ADD ( $\uparrow$ )	
	Feedforward	FF + Refiner
Fully Sim.	74.67	83.47
Finetune on Real	<b>86.16</b>	<b>90.36</b>

Table A. Results of fine-tuning the 6D pose estimation method on real data, over 25 measurements of the “2” object.



Figure B. “2” objects with different reflectance properties used in the varying scene reflectance experiment (Sec. B.3). From left to right: matte white, glossy white, and spotted black and white.

experiments, and fine-tune the model trained on simulated data on these measurements. We leave the refiner unmodified.

The results of the fine-tuning experiment are presented in Tab. A. We see a significant improvement in the performance of the feedforward network. We also see a significant improvement in the result after refinement due to the improved starting estimate from the feedforward network. These results are encouraging as they indicate that a minimal amount of real-world data could improve the performance of our method.

### B.3. Varying Scene Reflectance

The transient is a product of scene geometry *and* reflectance, so scenes of varying reflectance could affect the performance of our method. We conduct a systematic test in which we modify the reflectance properties of the 3D printed digit “2” and the tabletop surface. We test “2” objects with three surface finishes, as shown in Fig. B. We test two table materials: matte white and matte black.

The results of varying surface properties are presented in Tab. B. A modest decline in performance is observed with the glossy white object and the matte black tabletop, while a significant drop in performance occurs with the spotted black-and-white object. We attribute this drop to the fact that the spotted object has strong low-frequency variations in albedo across the surface. This sort of albedo variation is not included in our domain randomization when generating simulated data, nor is it able to be modeled by our refiner.

### B.4. Varying Ambient Light

We evaluate the performance of our method under varying levels of ambient lighting in Tab. C, on a new set of 10 captures of the “2” object at each light level. We see consistent

Obj. Material	Table Material	AUC-ADD ( $\uparrow$ )	
		FF	FF + Refiner
Matte White	Matte White	74.67	83.47
Matte White	Matte Black	66.59	79.55
Glossy White	Matte White	69.86	77.74
Spotted B/W	Matte White	50.46	61.49

Table B. 6D Pose Estimation of the “2” object with varying object and tabletop surface reflectance.

Ambient Light Level	AUC-ADD( $\uparrow$ )	AUC-ADD-S( $\uparrow$ )
< 0.1 lux	65.69	90.47
300 lux	72.45	93.10
3000 lux (heavy IR)	25.45	26.53

Table C. 6D Pose Estimation of the “2” object under varying levels of ambient illumination.

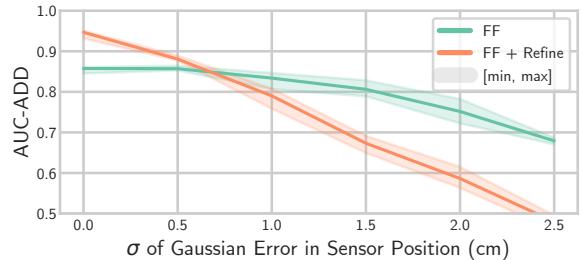


Figure C. Effect of adding Gaussian error to sensor poses on the “2” pose estimation task before feeding into our method.

performance in darkness (<0.1 lux) and the same indoor lights as used in other captures (300 lux), but a heavy falloff in performance under a very bright halogen spotlight (3000 lux), which emits high amounts of infrared light, leading to a high DC offset in the transient histogram. This performance drop is expected as we assume negligible ambient light in both synthetic data generation and the refiner. Future work could aim to alleviate this problem by including ambient light level in domain randomization to make the feedforward network more robust, pre-processing histograms to mitigate the effect of ambient light, and/or optimizing for ambient light level in the refiner.

### B.5. Sensitivity to Inaccuracy in Sensor Pose

When generating synthetic data to train our method, we add random Gaussian noise to the simulated sensor position to increase robustness to real-world inaccuracies in sensor position. We perform a simulated experiment to test this robustness, the results of which are shown in Fig. C. Both the feedforward model and refiner are robust to modest variations in sensor pose (< 1cm), which are likely achievable in realistic settings. We find that the feedforward method is more robust to variations than the refiner, and when there is high variation in sensor pose, foregoing the refiner leads to higher accuracy in the recovered object pose.

Number of Views	AUC-ADD (AUC-ADD-S) ( $\uparrow$ )	
	Feedforward	FF+Refiner
5 Pixels (Views)	74.79 (90.34)	74.80 (90.34)
10 Pixels (Views)	78.29 (90.57)	78.07 (90.63)
15 Pixels (Views)	84.65 (91.27)	84.79 (91.66)
25 Pixels (Views)	87.48 (91.51)	87.40 (91.42)
50 Pixels (Views)	90.54 (94.33)	90.87 (94.59)
100 Pixels (Views)	91.47 (94.58)	91.49 (94.37)

Table D. 6D Pose Estimation with Different Numbers of Views.

Ablation	AUC-ADD ( $\uparrow$ )	
	Feedforward	FF + Refiner
Full Model	73.67	83.47
Idealized Jitter Kernel	22.64	24.50
Incorrect Bin Size	49.98	33.23
Incorrect FoV	29.70	44.22

Table E. Results of 6D Pose Estimation under varying sensor model ablations, over a dataset of 25 captures of the “2” object.

## B.6. Sensor Model Ablation Study

We perform an ablation study over key components of our sensor model as described in Sec. 3.1. We consider the following variants:

1. **Full:** The full sensor model as described in Sec. 3.1 and used for all previous experiments.
2. **Idealized Jitter Kernel:** The jitter kernel  $s$  is replaced by a Dirac delta function at the location of the peak of  $s$ .
3. **Inaccurate Bin Size:** The temporal bin size  $\Delta t$  of the transient histogram is  $\sim 10\%$  smaller than as calibrated (from 1.38cm to 1.2cm).
4. **Inaccurate FoV Size:** The angular size of the FoV is incorrect by  $\sim 20\%$ , increasing from  $32^\circ$  to  $38^\circ$ . Additionally, the intensity map  $I(\omega)$  is replaced with a constant function.

For each variant, we train a feedforward model on synthetic data generated with the ablated sensor model, and use the same ablated sensor model in our refiner. Results over the 25-pose “2” digit dataset are shown in Tab. E. The results demonstrate that each of these aspects of sensor modeling are important to achieve good performance.

## C. Runtime and Complexity Analysis

While our method foregoes some computation performed by traditional methods (*e.g.* peak finding and ICP), it is replaced by relatively costly neural network inference and iterative pose refinement. Therefore we do not foresee efficiency improvements compared to point cloud-based methods. One feed-forward pass of our network takes  $\sim 4.8$  ms. The (unoptimized) refiner takes  $\sim 2$  seconds. With attention paid to efficiency, refiner speed could likely be increased. The costs of both the forward pass and optimization scale linearly with the number of viewpoints.

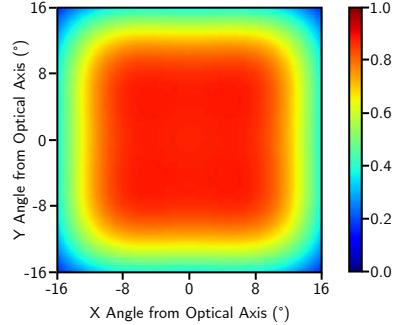
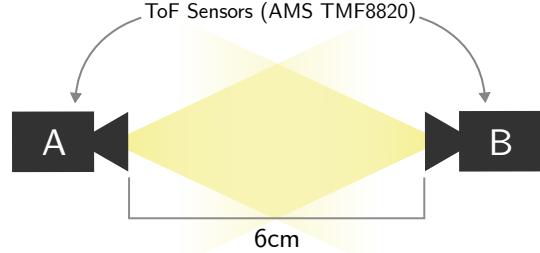
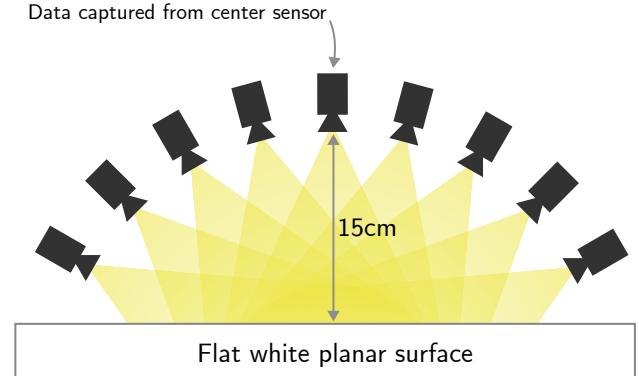


Figure D. Visualization of the laser intensity function  $I(\omega)$  that we use for the TMF8820 sensor, as given by Eq. (9). We set  $K_1 = 0.88$ ,  $K_2 = -3.16$ ,  $K_3 = 250.51$ .



(a) Sensor configuration for interference experiment 1.



(b) Sensor configuration for interference experiment 2.

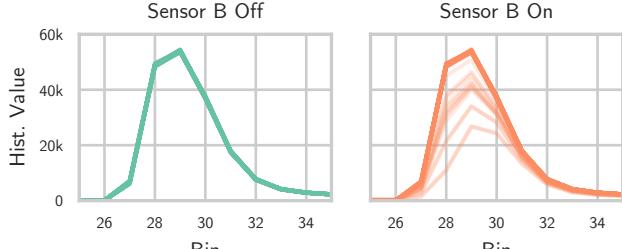
Figure E. Sensor configurations used for interference experiments.

## D. Test of Between-Sensor Interference

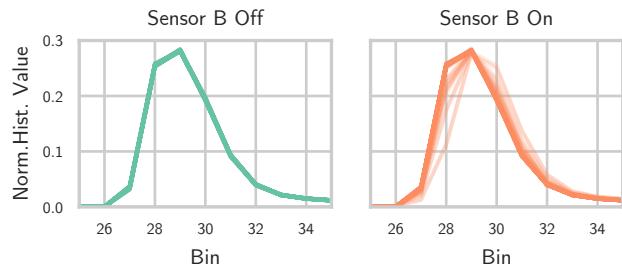
In our prototype system, a single sensor is moved to multiple positions while the scene remains static. However many practical applications for our method may involve multiple sensors imaging the scene at the same time, which could lead to interference between sensors. We perform controlled experiments to investigate the effect of interference.

### D.1. Two Sensors Facing Each Other

We position two AMS TMF8820 sensors facing directly at each other at a distance of 6cm, as illustrated in Fig. Ea. We compare measurements captured by sensor A between two conditions: sensor B on and sensor B off. The raw and nor-



(a) Raw histograms



(b) Histograms normalized to have a sum of 1.

Figure F. Comparison of the histograms captured in interference experiment 1. Each plot shows 128 sensor measurements overlaid. About 90% of samples in the right column exhibit no interference artifacts, comprising the dark orange lines.

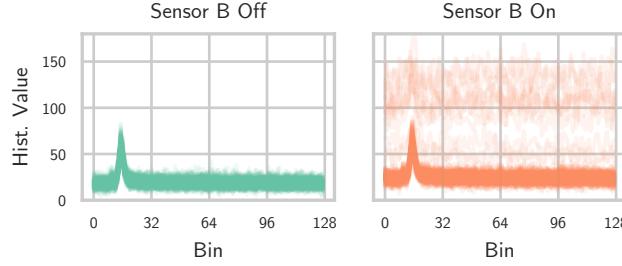
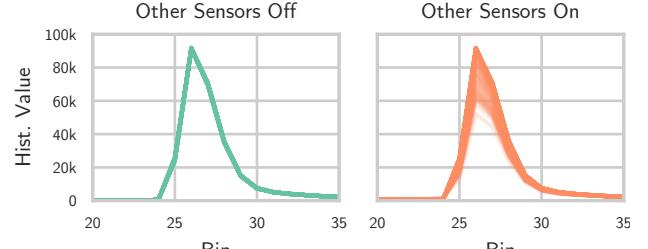


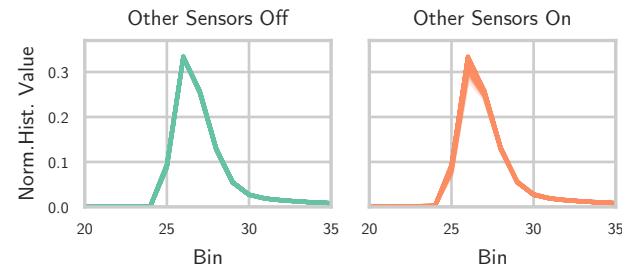
Figure G. Comparison on the histograms captured in interference 1, with the light source of sensor A covered. Each plot shows 128 sensor measurements overlaid. About 90% of the samples in the right column exhibit no interference artifacts, comprising the dark orange line.

Normalized histograms for both conditions are shown in Fig. F. We find that the operation of sensor B causes an effect in the histogram captured by sensor A  $\sim 10\%$  of the time. Even after normalization, the effect is still present. This effect appears similar to the effect caused by ambient light [13], and is consistent with what we would expect to see if sensor B's light source is not correlated with the light source of sensor A; *i.e.*, because the laser pulse trains of the two sensors are not synchronized, sensor B's operation leads to photons arriving uniformly at any time relative to sensor A's pulse train, just as ambient light arrives uniformly.

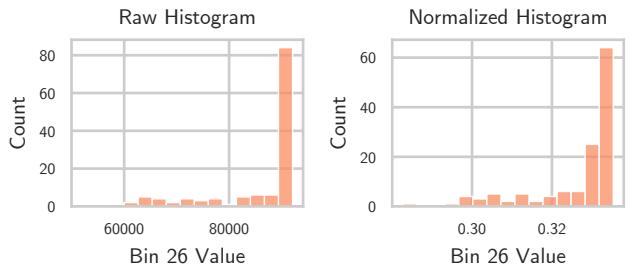
To further validate this hypothesis, we perform another test using the same sensor configuration in which the laser



(a) Raw histograms



(b) Histograms normalized to have a sum of 1



(c) Histogram of values of bin 26 (the peak) for the "Other Sensors On" condition. The bin values are grouped together in about 75% of measurements.

Figure H. Comparison of the histograms captured in interference experiment 2.

light source of sensor A is covered, so that only ambient light and the effect of sensor B are captured by sensor A. The results of this experiment are shown in Fig. G. In this case we can clearly see interference manifest as a DC offset in the captured histogram, again matching the signature of ambient light.

## D.2. Nine Sensors Imaging a Plane

We perform a second experiment in which nine sensors are all operating simultaneously and imaging the same portion of a planar surface. The experimental setup is illustrated in Fig. Eb. We position the sensors such that the centers of their optical axes each intersect with a planar surface at the same point, and record data only from the center sensor. Again, we compare between two conditions: the other 8 sensors on, and the other 8 sensors off. The results of this experiment are shown in Fig. H. We see the same effect as

in the previous experiment, but with a slightly higher occurrence rate of  $\sim 25\%$ .

### D.3. Discussion: Between-Sensor Interference

We have demonstrated that, at least for the AMS TMF8820 sensor, the effect of interference between sensors happens only occasionally even in the worst case. In practical scenarios, the rate of interference is likely to be quite low (*i.e.*  $< 10\%$ ). Further, the effect of interference on the histogram appears to be similar to the effect of ambient light. Adjusting captured histograms to account for ambient light is a well-studied problem [13], and it is likely that methods which are robust to changes in ambient light will be robust to between-sensor interference. While future applications should take interference into account, we believe it is unlikely to be a major obstacle for future deployments of distributed miniature ToF sensors.

## E. Visualization of 6D Pose Results

We provide visualization of our results on 6D pose estimation in Figures J to R.



Figure I. Objects used for 6D pose estimation experiments.

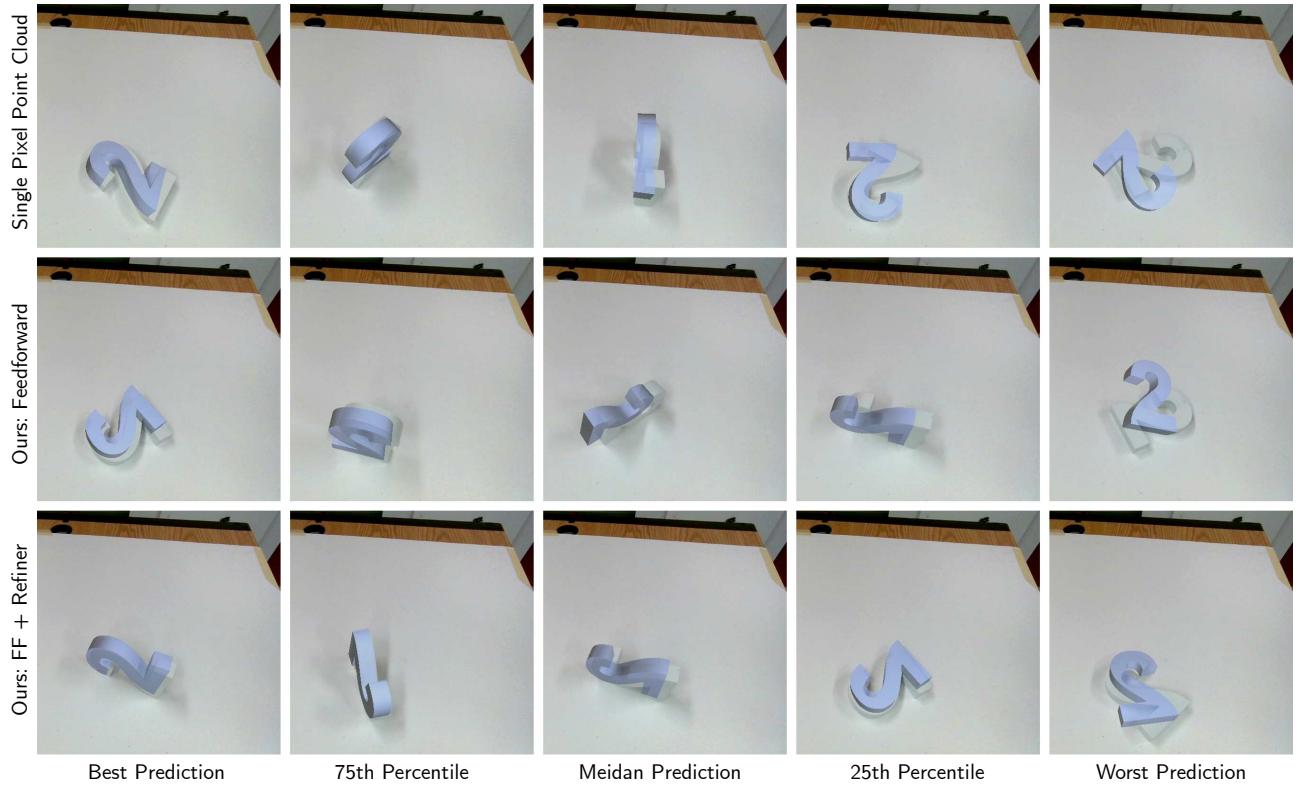


Figure J. Visualization of results on the 3D printed “two” object.

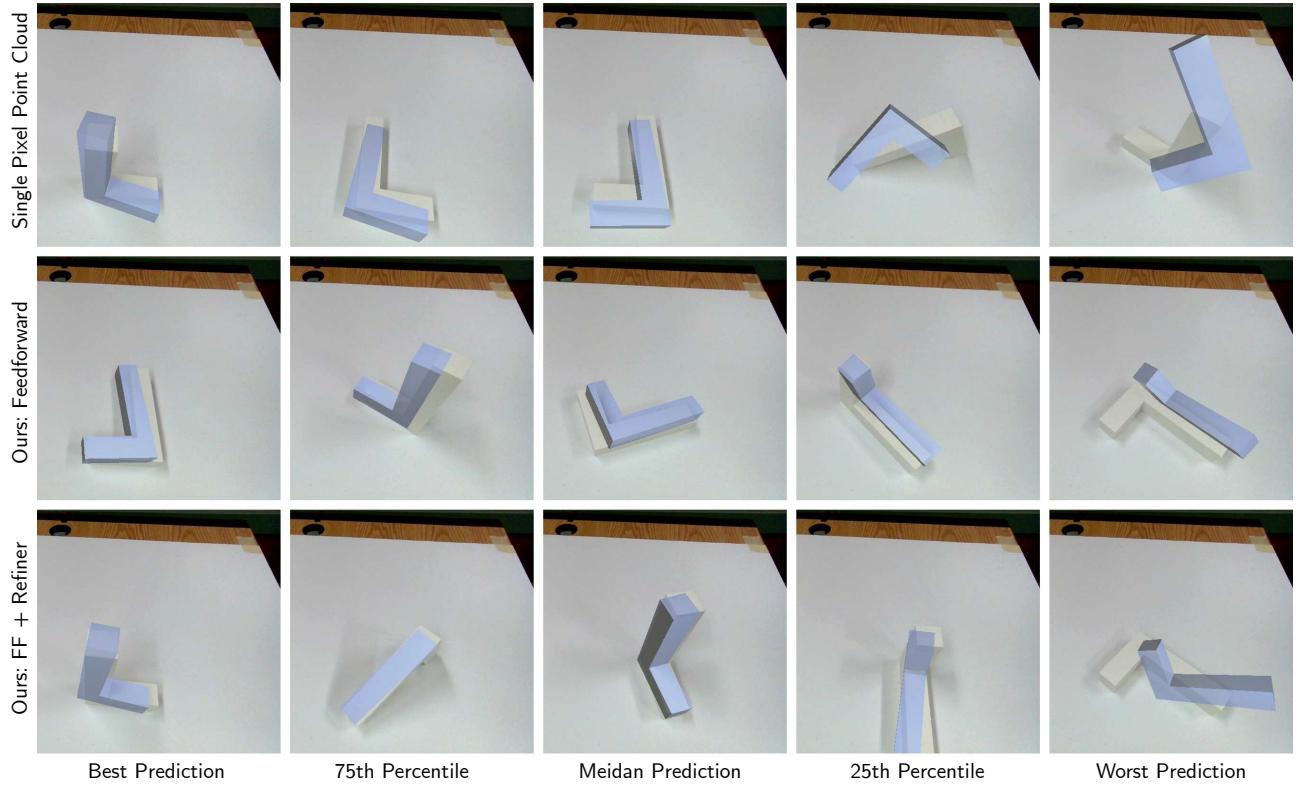


Figure K. Visualization of results on the 3D printed “L” object.

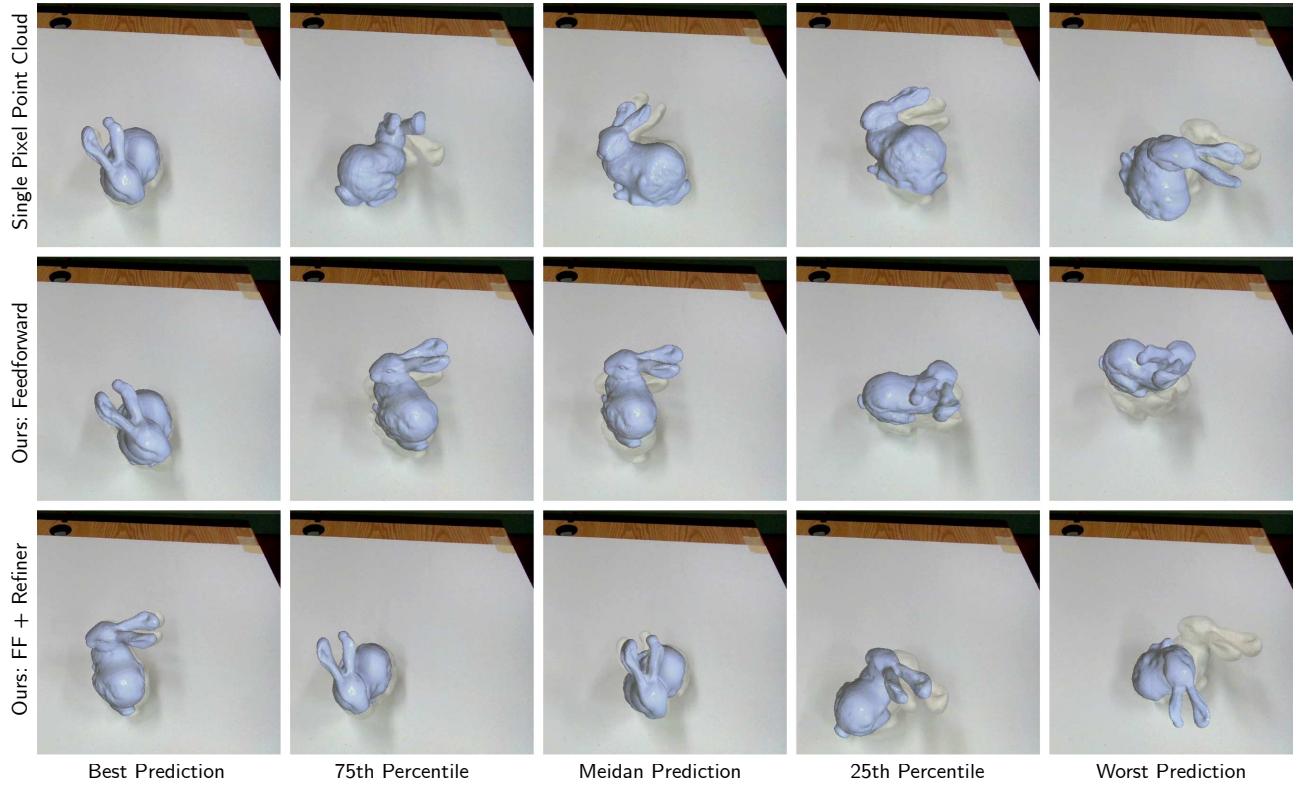


Figure L. Visualization of results on the 3D printed “bunny” object.

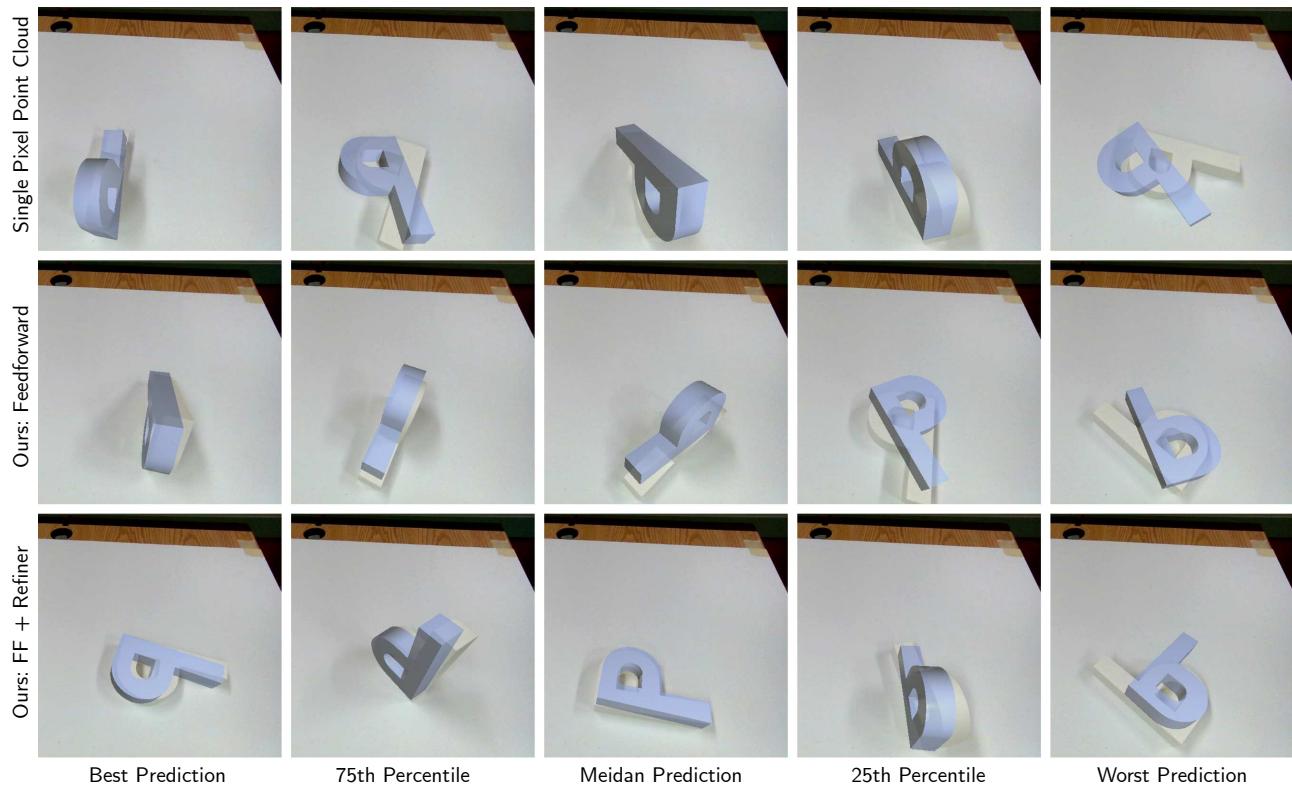


Figure M. Visualization of results on the 3D printed “P” object.

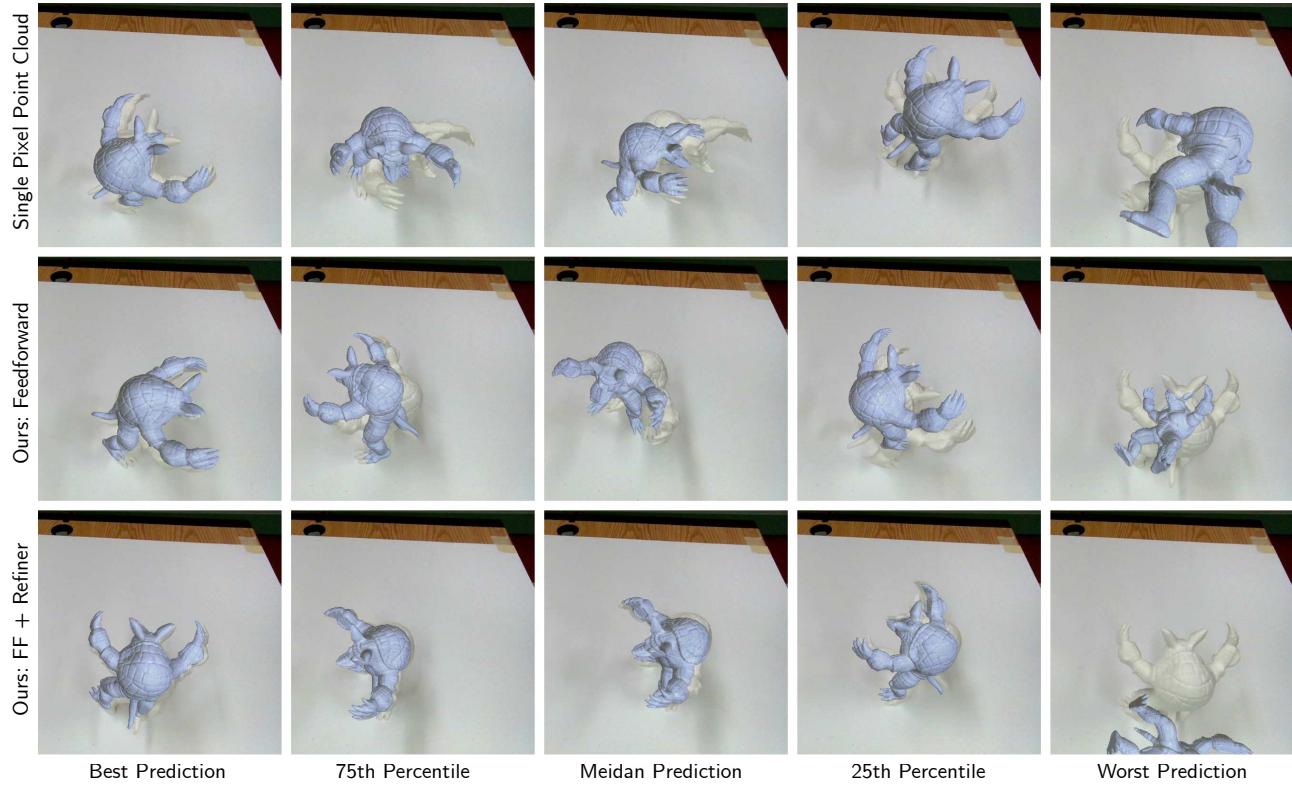


Figure N. Visualization of results on the 3D printed “armadillo” object.

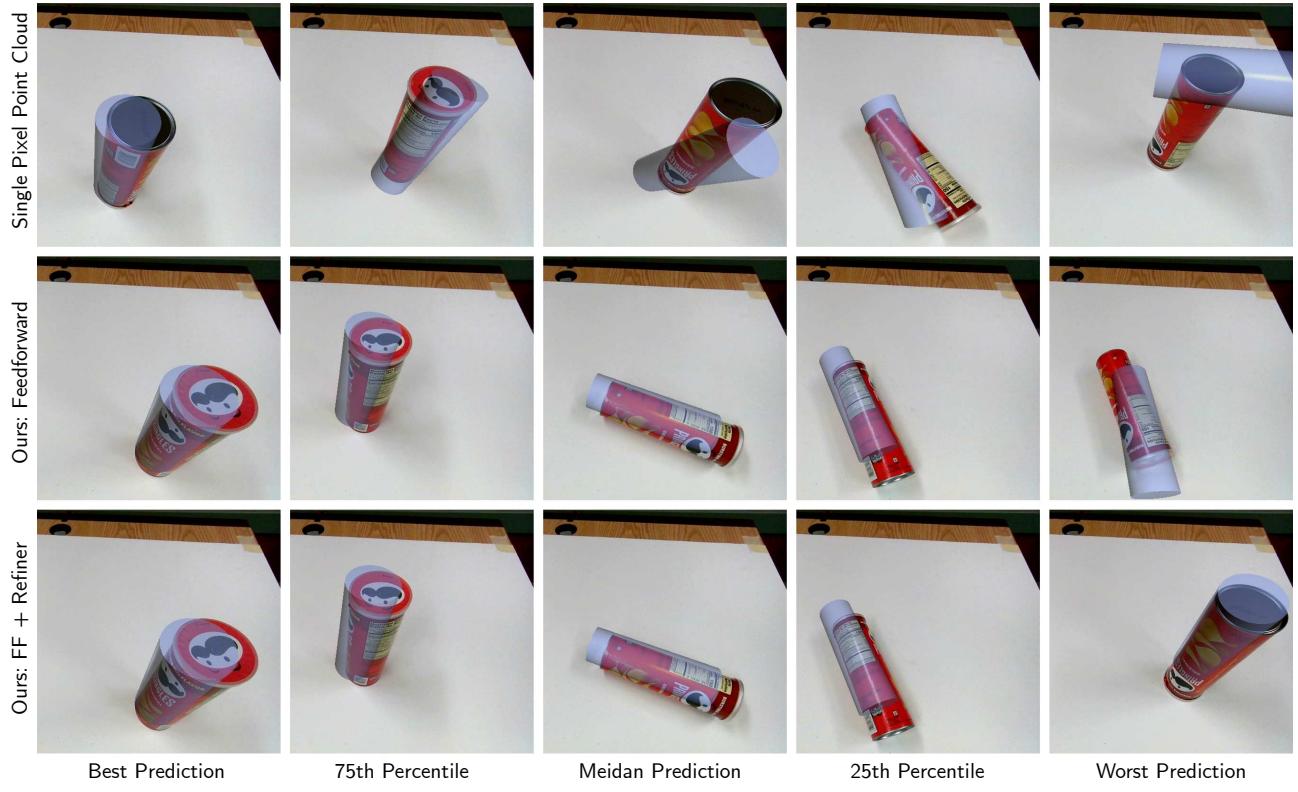


Figure O. Visualization of results on the “chips” object from the YCB dataset.

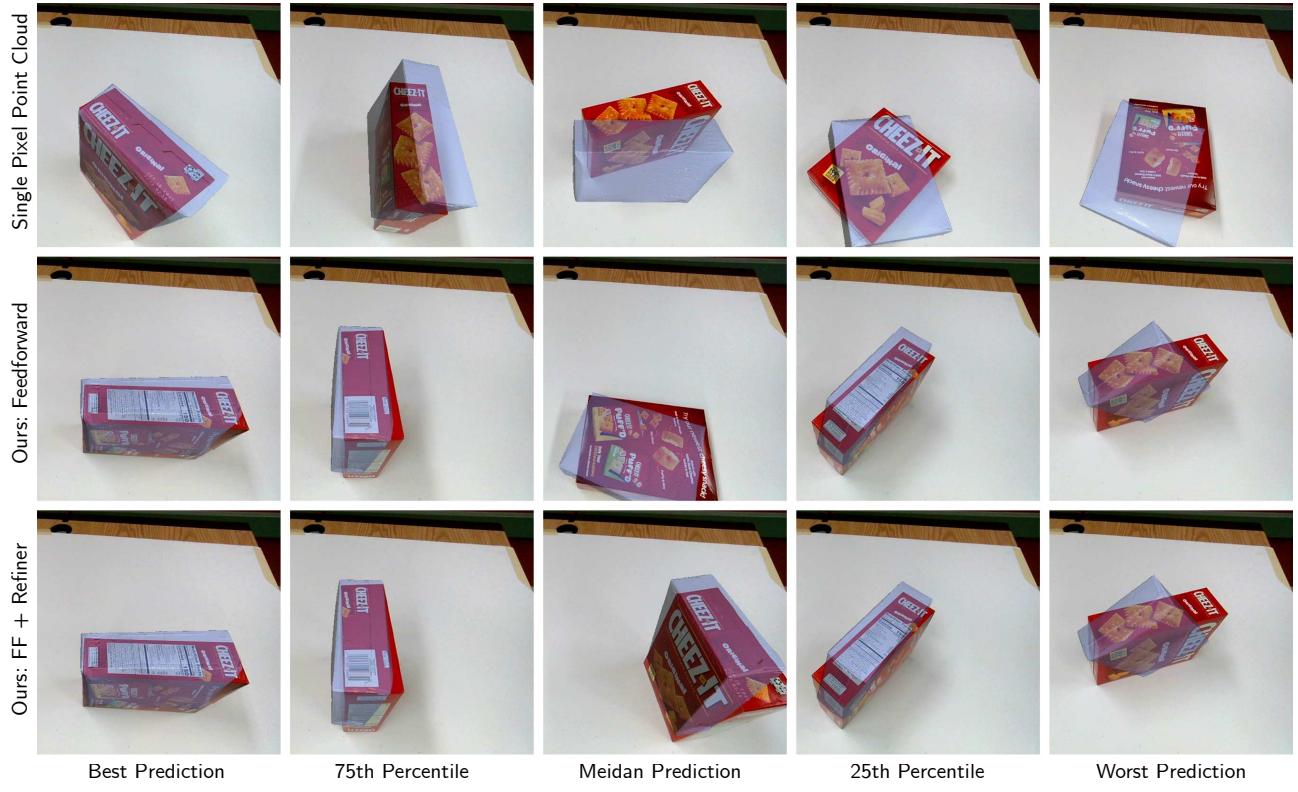


Figure P. Visualization of results on the “crackers” object from the YCB dataset.

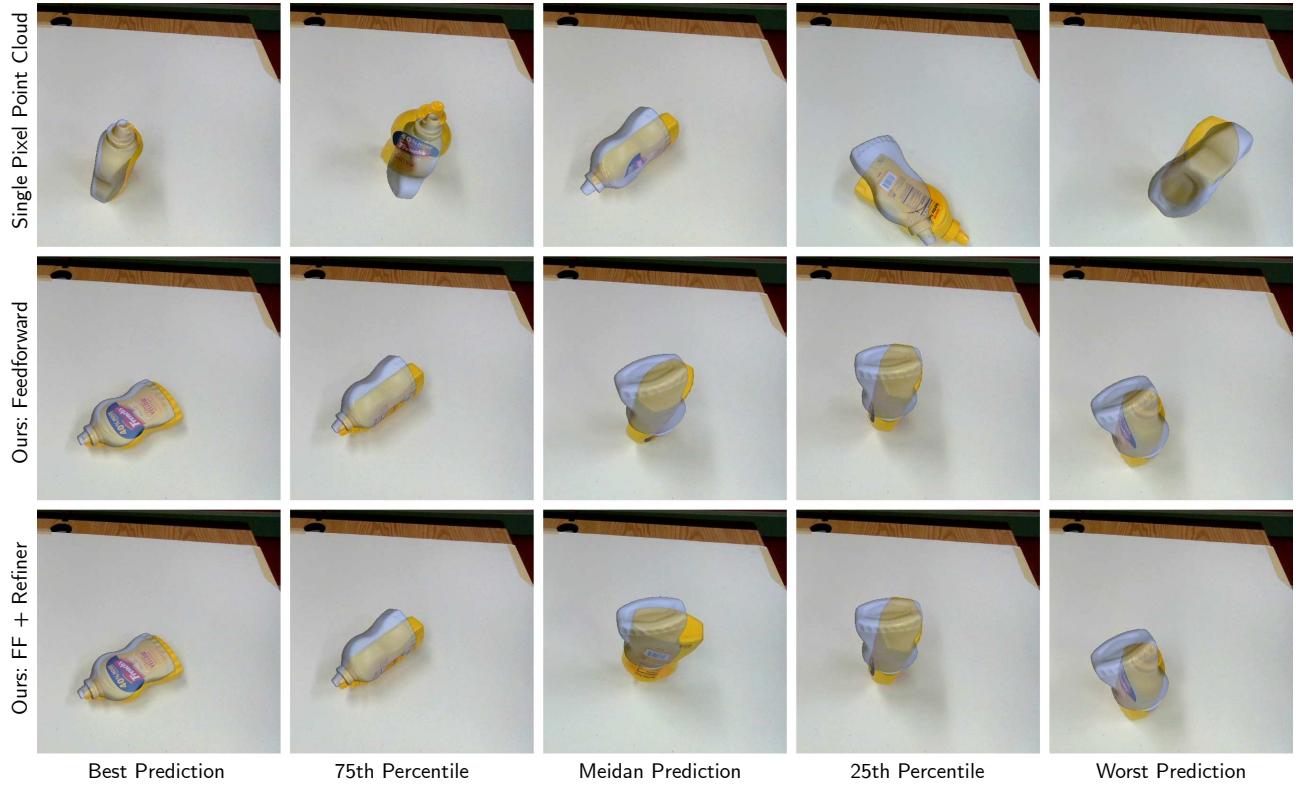


Figure Q. Visualization of results on the “mustard” object from the YCB dataset.

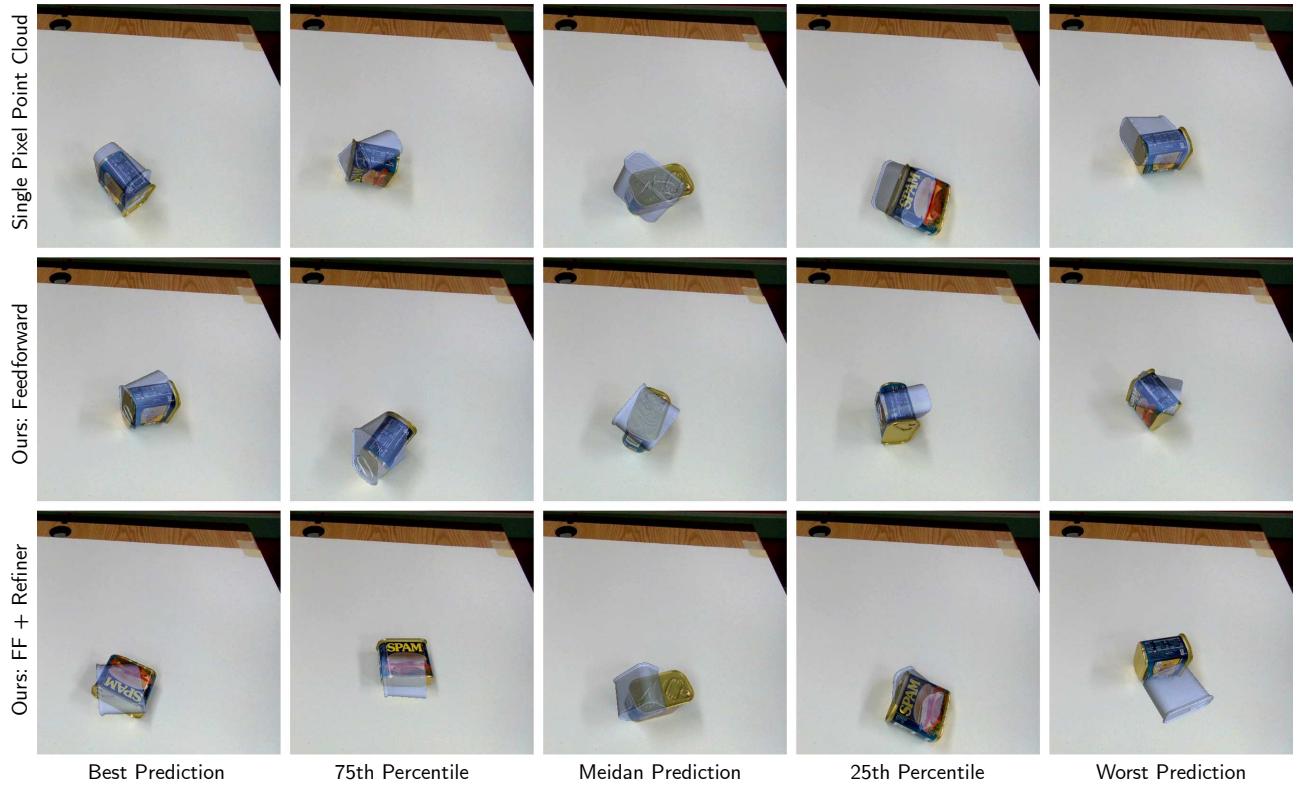


Figure R. Visualization of results on “SPAM” object from the YCB dataset. The SPAM is a failure case for our method due to its specular surface, small size, and many near-symmetries which make optimization difficult.