3D Vision with Miniature Time-of-Flight Sensors

By
Carter Sifferman

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2026

Date of final oral examination: 02/06/2026

The dissertation is approved by the following members of the Final Oral Committee:

Michael Gleicher, Professor, Computer Sciences
Mohit Gupta, Associate Professor, Computer Sciences
Yin Li, Associate Professor, Biostatistics & Medical Informatics
Yong Jae Lee, Professor, Computer Sciences

ACKNOWLEDGMENTS

Thank you to my advisors, Michael Gleicher and Mohit Gupta, for taking a chance on me and providing me with the mentorship, support, and expertise that I needed to succeed. They have taught me how to be a good researcher and communicator, encouraged me to push myself, and helped me to discover my own capabilities. I will carry these things with me for a lifetime.

Thank you to the rest of my dissertation committee, Yong Jae Lee and Yin Li, for their feedback and support, which have undoubtedly improved the quality of my research.

I'm fortunate to have worked with many talented individuals throughout my PhD. Thank you to my student collaborators Yiquan Li, Yiming Li, Fangzhou Mu, Sacha Jungerman, William Sun, and Yeping Wang. Much of my dissertation would not have been possible without their collaboration and hard work.

Thank you to my labmates and friends, who kept me grounded and made Madison a great place to call home. Thank you to my parents for supporting me and encouraging me to get out of my comfort zone. Thank you most of all to my wife, Lyndsey, who has given me her unwavering encouragement and support in all that I do.

The work in this dissertation has been funded in part by National Science Foundation awards 1830242, 1943139, 2107060, 2003129, 2333491, 2442739, and 2152163, ONR grant N000142412155, ARL under contract number W911NF-2020221, Los Alamos National Laboratory and the Department of Energy, Wisconsin Alumni Research Foundation, a University of Wisconsin Vilas Award, and a SONY Faculty Innovation Award

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Optical time-of-flight sensors operate by illuminating the scene with a pulse of light and measuring the time it takes for that pulse to return back to the sensor. These sensors convert that time-of-flight to a per-pixel distance to the scene, forming a depth image. Recently, miniature *time-resolved* time-of-flight sensors have become available. In addition to reporting the distance to the scene, these sensors capture a 1D waveform per-pixel which encodes the intensity of returning light at picosecond-to-nanosecond resolution. This waveform therefore encodes sub-pixel information about the scene geometry and photometric properties. However, existing downstream algorithms for these sensors do not utilize the entirety of this 1D waveform, instead summarizing it to a single distance estimate per-pixel. Because of this, these miniature sensors have so far been limited to crude tasks such as proximity sensing and presence detection, rather than 3D computer vision tasks that require more complex processing, such as object tracking or pose estimation.

In this dissertation, we develop new algorithms and frameworks for making use of the sub-pixel information captured by miniature time-of-flight sensors. We create realistic models of existing off-the-shelf sensor hardware, and apply those models to demonstrate our algorithms in the real world. We primarily focus on robotics applications—robot arms and mobile robots—with additional work on general scene recovery. Our algorithms demonstrate a path towards enabling a new level of computer vision on minimal sensing hardware.

1 INTRODUCTION

3D imaging systems offer two primary advantages over monocular RGB imagery: 1) 3D sensors provide metric depth information, and 2) the 3D data reported by these sensors feeds directly into downstream geometric algorithms for *e.g.* mapping or reconstruction. As a result, 3D imaging systems are often a practical and reliable solution for both real-time systems and offline reconstruction and mapping. 3D imaging systems come in many forms; for example, depth-from-stereo pairs, long-range direct time-of-flight (ToF) LiDAR, and high-precision structured light arrays. Each of these systems offers a different set of tradeoffs in terms of requirements (power, compute), form factor (size, weight), accuracy, operating range, and robustness to environmental conditions.

Thanks in part to this broad spectrum of capabilities and configurations, 3D sensors are used in a wide range of applications, including smartphone camera systems [4], autonomous vacuum cleaners [96], industrial robots [10], and autonomous vehicles [70]. In offline settings, LiDAR maps aided in restoration of Notre Dame Cathedral after a fire [45], LiDAR has been used to discover lost cities in the jungle [98], and is used to map and inspect critical infrastructure [19].

Recently, miniature direct ToF sensors have become commercially available. These sensors lie at the extreme end of the spectrum in terms of size, weight, and power requirements. Consequently, they offer very little spatial resolution, presently ranging from a single pixel to an 8×8 pixel array. Existing applications for these sensors are for simple tasks which require minimal processing of the distance estimates reported by the sensor. For example, cliff detection for mobile robots [91], collision avoidance for micro-drones [43], and aiding auto-focus for smartphones [32]. The vast majority of existing applications treat the sensor measurements as single distance estimates per-pixel. In reality, these sensors gather richer underlying information, in the form of a 1D waveform called the *transient histogram*, which encodes the intensity of returning light over very short time scales. In such a resource constrained setting, taking advantage of this

information is crucial to enabling 3D computer vision (*e.g.* object tracking, pose recognition, 3D reconstruction).

I claim that, by leveraging the sub-pixel transient histogram information captured by miniature time-of-flight sensors, these sensors can be used for 3D computer vision, unlocking new applications for real-time systems. In this dissertation, we develop new algorithms for making use of the sub-pixel transient histogram information captured by miniature ToF sensors. We demonstrate that these algorithms can unlock new possibilities in real-time systems often with minimal compute overhead, and can enable higher fidelity 3D vision with minimal sensing hardware. We explore data-driven approaches for detecting deviations in transient histogram measurements, and apply these to detect deviations in planar surfaces (Chapter 3) and to detect objects near a robot arm (Chapter 4). Next, we investigate both data-driven and first-principles approaches for recovery of planar surfaces from a single multi-pixel miniature ToF sensor measurement (Chapter 5). Finally, we demonstrate a general framework for recovery of parametric scenes from a spatially distributed set of miniature ToF sensors (Chapter 6).

2 DIRECT TIME-OF-FLIGHT IMAGING WITH SINGLE-PHOTON CAMERAS

Direct time-of-flight (ToF) sensors operate by illuminating the scene with a very short (on the order of picoseconds) burst of light, and measuring the time it takes for that light to bounce off the scene and return back to the sensor. In this dissertation, we focus on utilizing the raw time-of-flight information captured by *time-resolved* direct time-of-flight sensors. These sensors capture not only the *time* it takes for the light to return to the image sensor, but the *intensity* of photon flux at a very high temporal resolution (pico-to-nanoseconds). This 1D signal, which encodes the intensity of photon flux per-pixel is called the scene *transience* and the quantized version captured by the sensor is called the *transient histogram* [42, 38]. Single photon avalanche diodes (SPADs) [79, 133] are the most mature and widely available technology enabling this type of sensing. While SPAD sensors are used for real-world tests in this work, the same principles developed here could be applied to any time-resolved direct time-of-flight sensor, should new sensor technologies become available.

Time-resolved direct ToF sensors exist in a range of form factors, including 360 automotive LiDAR [83], smartphone camera modules [4], and laboratory grade benchtop setups [68]. In this dissertation, we focus on miniature time-resolved direct ToF sensors, which are often marketed as "direct time-of-flight (dToF)" or "proximity" sensors. These sensors are available for less than $5 USD and are widely used in robotics applications [118, 80]. In addition to their low cost, they are very small ($<$20 mm^3) and low-power ($<$10 milliwatts per frame) [108, 2]. However, these sensors provide very low spatial resolution, ranging from a single pixel [109] to an 8×8 array [108]. These sensors operate as a flash LiDAR system, illuminating the entire field-of-view at once with a diffuse laser.

The transient histogram signal captured by these sensors is a product of the geometric and photometric properties of the scene, as well as the properties of the

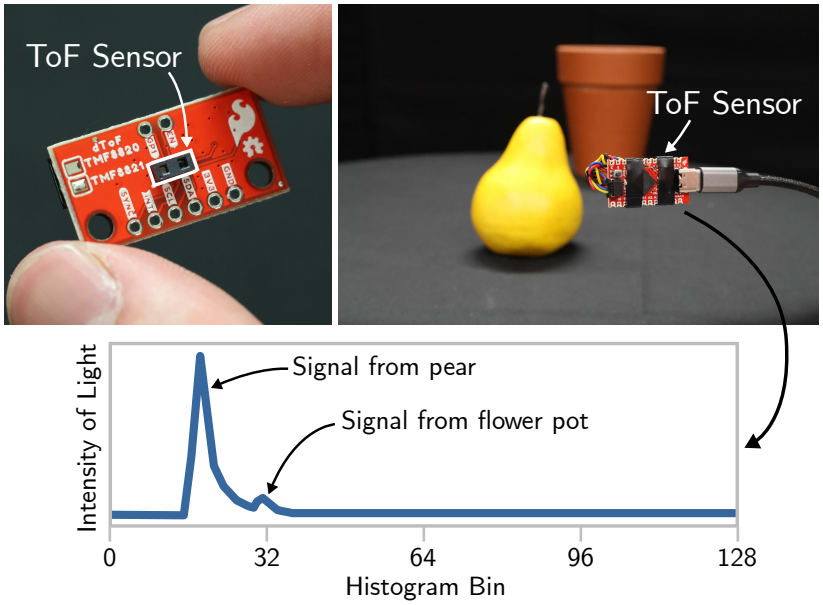


Figure 2.1: **Miniature Time-of-Flight Sensor and Accompanying Histogram Measurement**: Due to the sensor's wide field-of-view, it captures returning light from both the pear and flower pot. Despite the flower pot being larger in size, its corresponding signal in the histogram is smaller in magnitude due to its increased distance from the sensor.

sensor. An example transient histogram for a simple scene is shown in Fig. 2.1. More details of the image formation process, including nonlinearities caused by SPAD technology and intricacies of specific sensors, are covered in subsequent chapters when necessary.

The key challenge addressed in this dissertation is how to make use of the transient histogram signal for 3D computer vision tasks, *e.g.*, pose estimation, 3D reconstruction, or obstacle detection. Traditional algorithms based on features corresponding to distinct visual patterns (*e.g.*, textures, edges, or corners) [63, 99, 23] are not effective due to the lack of spatial resolution. Supervised learning-based methods can be difficult to train because existing large-scale datasets do not include transient histogram measurements. Additionally, the image formation process contains nonlinearities, making it difficult to invert, and the histogram is a product of both the photometric and geometric properties of the scene, which are

difficult to disentangle. Throughout this dissertation, we take multiple approaches to address these challenges for multiple downstream tasks.

3 DETECTING DEVIATIONS IN A PLANAR SURFACE

In this chapter, we investigate methods for determining if a planar surface contains geometric deviations (*e.g.* protrusions, objects, divots, or cliffs) using only an instantaneous measurement from a miniature optical time-of-flight sensor. The key to our method is to utilize the entirety of information encoded in raw time-of-flight data captured by off-the-shelf distance sensors. We provide an analysis of the problem in which we identify the key ambiguity between geometry and surface photometrics. To overcome this ambiguity, we fit a Gaussian mixture model to a small dataset of planar surface measurements. This model implicitly captures the expected geometry and distribution of photometrics of the planar surface and is used to identify measurements that are likely to contain deviations. We characterize our method on a variety of surfaces and deviation types, and provide an example application in which our method enables mobile robot obstacle avoidance and cliff detection over a wide field-of-view.

Project website: cpsiff.github.io/using_a_distance_sensor/

3.1 Introduction

Optical time-of-flight distance sensors are widely used in robotics to sense the distance to nearby objects for tasks such as obstacle avoidance [28] or localization [43, 44]. These sensors are low-cost, low-power, and are available in low-resolution (*e.g.*, 4x4 pixel) arrays requiring minimal data bandwidth. The distance estimates reported by these sensors are summaries over a wide (*e.g.*,

This work was completed under the supervision of Michael Gleicher and Mohit Gupta. William Sun contributed by gathering testing data and evaluating methods. Carter led the project, designed the experiments, designed the algorithm, conducted experiments, created figures, and wrote up the results.

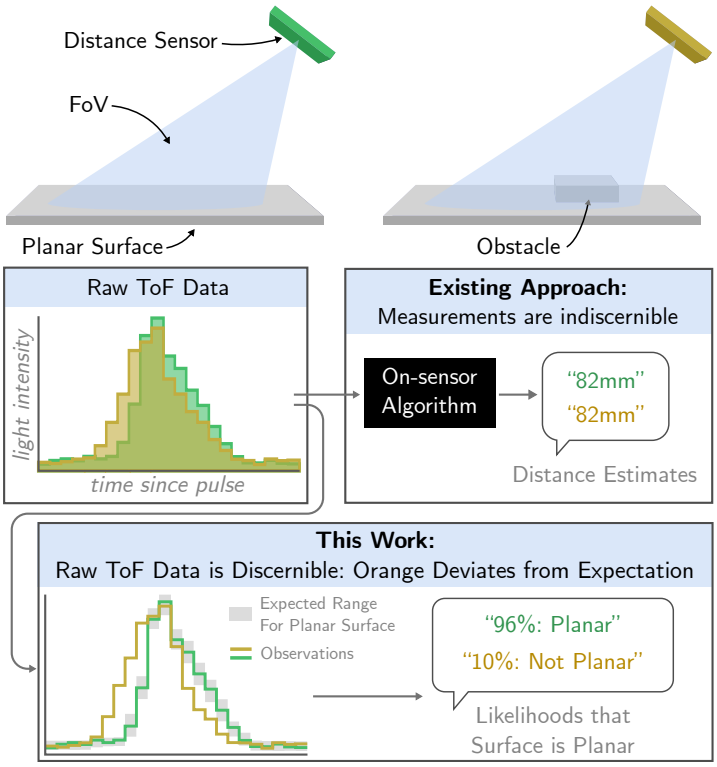


Figure 3.1: **Method Overview**: Our method uses raw time-of-flight data to detect deviations from planar surfaces (*e.g.*, objects, divots, cliffs, or walls). It is able to do so more accurately than methods that utilize on-sensor distance estimates.

10°-30°) field-of-view per pixel, which is advantageous for some applications (*e.g.*, conservative obstacle avoidance), but these distance estimates are ineffective at detecting small geometric deviations. In this work, we show that this limitation can be overcome by utilizing readily available raw time-of-flight information captured by these sensors. With this data, we are able to detect geometric deviations in a planar surface with more accuracy than is possible with distance estimates alone.

Our method utilizes raw time-of-flight data captured by consumer-grade *time resolved* active time-of-flight sensors. These sensors operate by illuminating a wide (30° +) patch of the scene with a pulse of light, and capturing the intensity

of light over time as it bounces back from the scene in a 1D temporal waveform called a *transient histogram* [16, 42]. These sensors are available for less than $5 USD and are widely used in robotics applications [118, 80]. In addition to their low cost, they are very small ($<$20 mm^3) and low-power ($<$10 milliwatts per measurement) [108, 2]. Typical applications do not utilize the transient histogram captured by these sensors, instead relying on a proprietary algorithm onboard the sensor to extract a single distance estimate per pixel. While this estimate is convenient for many tasks, it is not ideal for many others, as it obscures relevant information about the scene which is encoded in the shape and magnitude of the transient histogram.

In this chapter, we aim to detect geometric deviations on planar surfaces (*e.g.*, objects, divots, cliffs, or walls). Our method assumes that the relative orientation and distance of the sensor to the planar surface remains fixed (*i.e.*, only translation parallel to the surface is permitted), and our method requires a small dataset of measurements from the sensor to fit a surface model. We do not aim to detect the exact nature of the deviation; we only classify a measurement as "planar" or "not planar" (the latter class including planes viewed from a different orientation and/or distance). This capability is useful for robotics applications like mobile robot and drone navigation. It could also be useful for *e.g.*, safely landing a drone on a flat and level surface or safely placing a cup of liquid on a clear and level portion of tabletop with a robot manipulator. Our method enables deviation detection in a very small size, weight, and compute footprint, making it particularly useful for resource-constrained scenarios like micro-drones, and for distributed sensing with sensors mounted at many points on a larger robot.

The key contributions of the work in this chapter are: 1) an analysis of the problem of detecting deviations from a planar surface with a distance sensor, in which we identify the key ambiguity that makes the problem challenging, 2) a computationally lightweight method for detection of said deviations using raw sensor time-of-flight information, 3) characterization of the sensitivity and accuracy of said method compared to methods which use only sensor distance

estimates, and 4) an example application in which our method enables mobile robot obstacle avoidance.

3.2 Related Work

Obstacle Detection on Planar Surfaces

Detecting obstacles on top of planar surfaces is a widely studied problem in computer vision and robotics [5, 132]. Approaches vary in their problem setup (per-frame detection or utilizing constrained robot motion) and imaging modality (RGB, depth from stereo, or structured light).

Utilizing Robot Motion

One class of methods utilize robot motion parallel to a ground plane and the parallax effect to detect deviations from a flat plane in RGB images. This includes methods which compute homographies between subsequent images [134, 21], and those based on optical flow [111, 18]. Kumar *et al.* [51] use an RGB-D camera and robot motion to detect very small obstacles (0.5-2cm) on a planar surface. While such methods work well for moving, wheeled robots, they are not useful for detecting deviations when camera motion is unconstrained or nonexistent, and they assume that the scene is static from frame to frame. Our method operates on a per-frame basis, meaning it does not rely on robot motion nor assume a static scene from frame to frame.

Stereo-based Methods

Some methods for obstacle avoidance [11, 90, 41] use depth from stereo to determine the distance to the ground plane, and detect an obstacle when it deviates greatly from that ground plane. Compared to our work, these methods use larger sensors, as they require a stereo baseline, and the algorithms have higher compute requirements.

Learning-Based Methods

Many methods for per-frame object detection utilize a large labeled dataset to train a detector via supervised learning [41, 59, 95, 37]. Other works aim to generally detect any anomaly in an image, relying only on a dataset of anomaly-free images [26, 92]. While many of these methods are effective, the neural networks employed are generally compute intensive, making them unfit for resource constrained applications.

Time-Resolved Sensors in Robotics

Time-resolved time-of-flight sensors which report transient histograms are widely used in robotics to sense the distance to an object. One line of work places these sensors in a distributed manner on robot arms and uses their measurements to ensure safe movement [28, 117]. There exist methods for calibrating the position of these sensors on a robot arm [105]. The sensors' small size makes them well suited for use on small drones for obstacle avoidance [118] and SLAM [43], and for use on very small mobile robots [80].

Low-cost Distance Sensor Transient Histograms

Transient histograms from low-cost distance sensors have been utilized for human pose estimation [101], object tracking [16], material classification [6, 16], depth estimation [42, 81, 58], to assist an RGB camera in SLAM [61], and for multi-view 3D reconstruction [75]. While many of these works are similar in spirit to this work, *i.e.*, they aim to take advantage of transient histograms to unlock new capabilities, they do not attempt to solve the same problem of detecting if a surface is planar from a single sensor reading.

3.3 Problem Analysis

Background: Transient Histograms

A time-resolved distance sensor operates by illuminating a wide patch of the scene with a very short pulse of light and capturing the intensity of that light over pico-to-nanosecond timescales as it returns to the sensor after bouncing off of the scene in a *transient histogram* [42, 38]. Single photon avalanche diodes (SPADs) [79, 133] are the most mature and widely available technology enabling this type of sensing.

In this chapter, we utilize the AMS TMF8820 sensor, a SPAD-based distance sensor with 9 pixels in a 3×3 configuration, making it akin to a very low-resolution depth camera. Unlike a typical depth camera, each pixel captures a 128-bin transient histogram, which can be read out in addition to a per-pixel distance estimate generated by a proprietary algorithm onboard the sensor. We choose this sensor as it reports histograms at a relatively high temporal resolution, with each bin corresponding to $\sim$ 1.2 cm of depth range. In principle, our method can be applied to any time-resolved optical sensor with a diffuse light source, which currently includes dozens of SPAD-based consumer distance sensor models, in addition to research-grade benchtop setups.

Sensor Sensitivity

Miniature time-resolved sensors can be very sensitive at close distances and have a high signal-to-noise ratio. To demonstrate this with the TMF8820, we perform an experiment in which we place a $20 \times 20 \times 1$ mm piece of heavyweight paper on a background of the same material, as shown in Figure 3.2. Without moving the sensor or background, we capture 16 measurements of the flat surface and 16 measurements of the surface with the small square of paper in the same position 20cm from the sensor. The sensor is set to its default integration period of 230ms, during which it integrates over 4 million laser pulses, giving it an

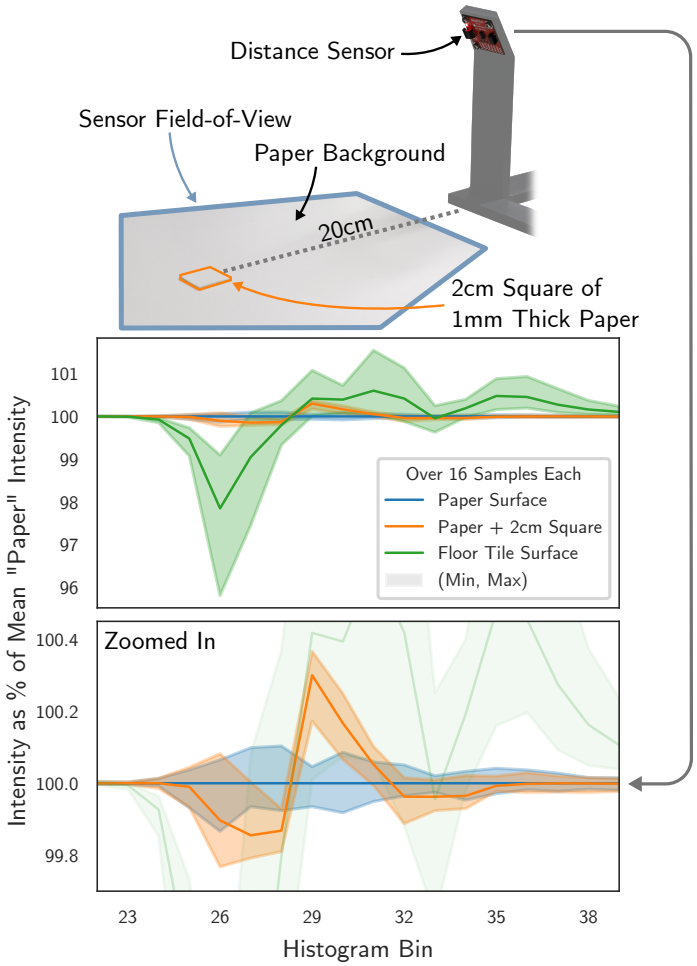


Figure 3.2: **Sensor Signal-to-Noise Ratio Test**: A low-cost distance sensor can distinguish between a flat surface and one with a small piece of heavyweight paper under controlled conditions. The effect of a change in surface photometrics caused by a change to a tile surface is much larger than the effect of a change in geometry caused by the presence of a small piece of paper on a background of the same material.

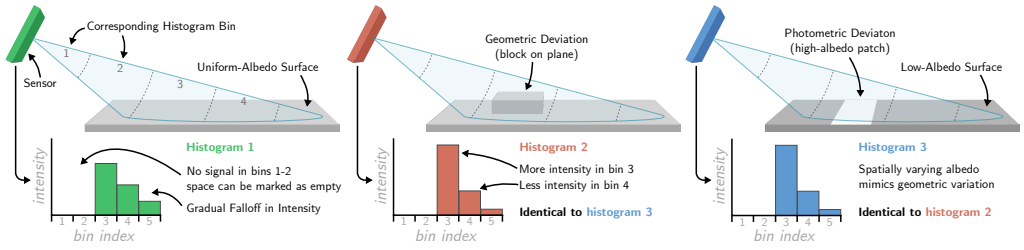


Figure 3.3: **Geometry-Albedo Ambiguity Demonstration**: Time-resolved distance sensors exhibit a fundamental ambiguity between geometry and albedo. While a geometric deviation from a flat plane does affect the histogram, a photometric deviation, in the form of a patch with a higher albedo, can affect the histogram in an identical way. This makes the detection of deviation an ill-posed problem aside from geometric variations that violate the space carving assumption. This space carving assumption is weaker at a steep angle of incidence with the plane.

effective frame rate of ∼4.3FPS. As shown in Figure 3.2A, transient histograms between the two scenarios are easily separable when the background surface is highly controlled. This might lead one to expect that the distance sensor would be capable of detecting even sub-mm deviations in a planar surface. However, sensor noise is typically not the limiting factor as discussed in the following subsection.

Ambiguity Between Geometry and Albedo

The captured transient histogram is a product of scene geometry, photometric effects (spatially varying scene albedo and reflectance), and sensor intrinsics (laser power, FoV, etc.). As described in previous work [42], this leads to an ambiguity between scene geometry and photometrics. The intensity of light captured in a given histogram bin is affected by both how *much* surface is within the range corresponding to that bin, and the effective *albedo* of that surface in the wavelength of the light source. From a single measurement, it is impossible to differentiate between a large low-albedo scene patch, and a small high-albedo one.

This ambiguity poses a problem for our goal of detecting deviations in a

planar surface. The only constraint that can be placed on the scene from a single measurement is that bins in which no light returns do not contain any geometry, commonly referred to as *space carving* [52, 68, 75]. This constraint is especially ineffective when the sensor is at a steep angle-of-incidence to the planar surface, *e.g.*, the configuration in Figure 3.3. Without making assumptions about the plane photometrics or introducing implicit bias into the inference process, it is impossible to detect planar deviations beyond those that violate the space carving constraint. This ambiguity exists regardless of the temporal resolution of the sensor, but is reduced by an increase in spatial resolution (*i.e.*, more pixels, each with a smaller FoV).

We demonstrate that, despite this fundamental ambiguity, it is practical to detect deviations on real-world surfaces by implicitly modeling the expected distribution of surface reflectance, and identifying measurements that are unlikely under the model. We are inspired by works on monocular depth estimation (MDE) [27, 31], which have been successful at predicting ordinal depth despite MDE exhibiting a very similar geometric-photometric ambiguity to our problem. This is possible because real-world data tends to be well-behaved, and adversarial examples that abuse the ambiguity are exceedingly rare. While approaches for MDE take advantage of implicit bias by training a neural network on a large dataset, the relative low-dimensionality of the measurements captured by our sensor means that simpler classical vision techniques can be effective.

The existence of this ambiguity informs the design of our method. We know that the image formation process cannot be directly inverted to recover geometry, so a model of the expected reflectance properties of the background surface is necessary to make predictions about scene geometry.

3.4 Method

Given a single measurement from a time-resolved distance sensor, we aim to predict whether the geometry of the imaged scene area matches that of a

plane viewed from a fixed distance and angle-of-incidence. This is a binary classification problem: given a measurement, classify whether it is an image of a deviation-free plane or not.

At inference time, our method takes as input a set of query histograms $\mathbf{H} = \{\mathbf{h}_i = [h_1, h_2, ..., h_b]\}_{i=0}^{p}$ captured simultaneously by a sensor with p pixels and b bins per histogram. We assume that the histograms have had some form of pile-up correction, like Coates' correction [20] applied, but have not been processed to remove ambient light, which manifests as a constant DC bias in the measured signal [36]. This is consistent with the transient histograms reported by currently available low-cost distance sensors, including the TMF8820. The output of our method is a likelihood $\ell \in [0, 1]$ of $\mathbf{H}$ being an image of a deviation-free planar surface viewed from a fixed position, *i.e.*, $p < 0.5$ could mean that there is an object atop the plane, the plane contains a divot downwards, the plane is not being viewed from the fixed distance and angle-of-incidence, or the imaged scene is not a plane at all.

Our method utilizes a dataset D of measurements $\mathbf{H}$, each of which images a flat planar surface from a fixed distance and angle-of-incidence. This dataset can consist of measurements of a single surface material (to be used *e.g.*, when the surface material is known) or a larger corpus of measurements of multiple surfaces. We investigate performance under both dataset types in Section 3.5. Fitting a model of the deviation-free surface to this dataset consists of two steps: 1) pre-processing D to remove the effect of ambient light and normalize based on albedo, and 2) modeling the distribution of D with a multi-dimensional Gaussian mixture model. This model was chosen based on empirical performance, because it allows lightweight inference, and because it can be trained on only samples of deviation-free planar surfaces.

Pre-processing

We approximate the ambient light level a_i for each histogram $\mathbf{h}_i$ by calculating kernel density on the values of $\mathbf{h}_i$ with a Gaussian kernel with bandwidth σ, and

choosing the value with the highest density:

$$a_i = \underset{x}{\text{argmax}} \sum_{h \in \mathbf{h_i}} \mathcal{N}(x; h, \sigma) \tag{3.1}$$

This serves as a way of estimating the *modal* value in the histogram with more robustness to noise than the mode. For most scenes, the modal bin value is equal to the influence from ambient light, as bins which do not capture light returning from the scene will capture only ambient light, and even a few such bins are typically enough to make it the modal value. In practice, the ideal σ varies based on sensor noise, and can be found by searching for the σ which minimizes the variation in recovered a over a set of measurements taken under the same ambient light.

Surfaces of different, but uniform albedos will result in histograms of similar shape, but different magnitudes, equivalent to scaling the bin values uniformly. To compensate for this effect, we normalize each histogram to have a unit L^1 norm after ambient light has been removed. The pre-processed measurement $\tilde{\mathbf{H}}$ is given by:

$$\tilde{\mathbf{H}} = \left\{ \frac{\mathbf{h}_i - a_i}{\|\mathbf{h}_i\|_1} \right\}_{i=1}^{p} \tag{3.2}$$

These pre-processing steps serve to align histograms which have similar shapes, but different magnitudes due to surface albedo, or different DC biases due to ambient light.

Gaussian Mixture Model

We model the distribution of measured histogram values using a multi-dimensional Gaussian mixture model [110]. We flatten each measurement $\tilde{\mathbf{H}}$ across the pixel dimension to get a one dimensional vector $\hat{\mathbf{H}} \in \mathbb{R}^k$ where $k = p * b$ containing each $\tilde{\mathbf{h}}_i \in \tilde{\mathbf{H}}$ concatenated one after another. Our method does not account for the ordering of the histogram bins or the spatial position of their fields-of-view, so this flattening serves to simplify notation. A Gaussian mixture model is fit to

the pre-processed measurements $\hat{\mathbf{H}}$ in the dataset D.

We fit a Gaussian mixture with c components, a per-component per-bin mean vector $\boldsymbol{\mu} \in \mathbb{R}^{c \times k}$, a per-component variance vector $\boldsymbol{\sigma}^2 \in \mathbb{R}^c$, and a per-component weight parameter $\boldsymbol{\alpha} \in \mathbb{R}^c$. This means that a separate Gaussian mixture is fit to each bin, with each Gaussian mixture constrained to use the same variances, weights, and number of components. The likelihood ℓ of a given histogram being an image of a flat plane is calculated by taking the joint probability over all bins, with each bin's likelihood calculated by summing all weighted components of the mixture:

$$\mathcal{L}(\hat{\mathbf{H}}) = \prod_{i=1}^{k} \sum_{j=1}^{c} \mathcal{N}(\hat{\mathbf{H}}_i; \boldsymbol{\mu}_{i,j}, \boldsymbol{\sigma}_j^2) * \boldsymbol{\alpha}_j \tag{3.3}$$

The Gaussian parameters are estimated over the dataset D of flat planar images using the expectation-maximization algorithm [24] to optimize the parameters $\boldsymbol{\mu}, \boldsymbol{\sigma}^2$, and $\boldsymbol{\alpha}$. To choose the number of Gaussian components c to use, we fit models with a range of c values and choose that which minimizes the Akaike information criterion (AIC), given by:

$$\text{AIC} = 2\rho - 2\ln\left(\sum_{\hat{\mathbf{H}} \in D} \mathcal{L}(\hat{\mathbf{H}})\right) \tag{3.4}$$

where the parameter count $\rho = kc + 2c$. The chosen c varies depending on the contents of D.

To predict the likelihood that a query histogram $\mathbf{H}_{\text{query}}$ is an image of a flat planar surface from the same fixed pose as the measurements in D, we perform the same pre-processing steps given in Equation 3.2 and calculate $\mathcal{L}(\hat{\mathbf{H}}_{\text{query}})$ as in Equation 3.3. To generate a binary prediction, we apply a threshold to this likelihood, and we label likelihoods above the threshold as "no deviation" (the negative class) and those below the threshold as "deviation" (the positive class).

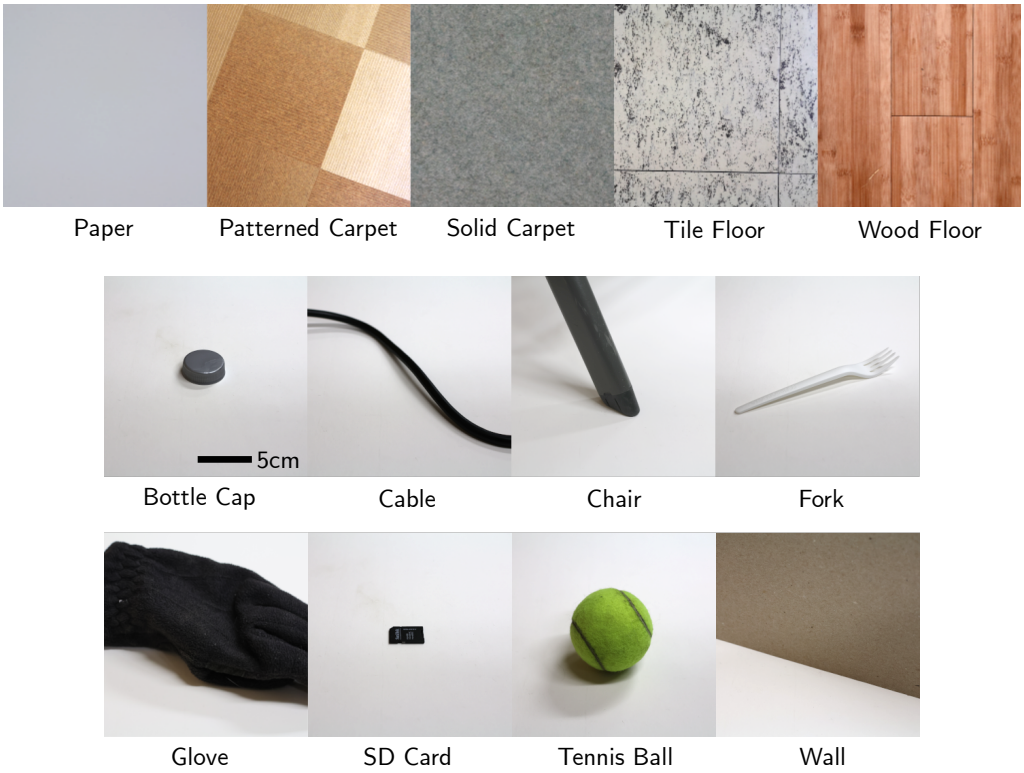


Figure 3.4: **Surfaces and Obstacles used in Object Detection Experiments.**

Baseline Methods

We compare our method to two baselines, both of which extract from the histograms a single value per-pixel before performing classification on those values. The *proprietary distances* method utilizes the per-pixel distance estimates generated onboard the sensor via a proprietary algorithm. The *histogram peaks* method finds the location of the histogram peak by fitting a cubic curve to the histogram and sampling that curve at a high frequency, following the method of [107]. For either method, we plug the distance estimates into the same Gaussian mixture model as described above, but effectively treating each distance estimate as a histogram with one bin, leading to p-dimensional feature vector. Empirically, we found this to be the best performing approach for utilizing distance estimates.

		Per-Object AUROC ↑							
Method	Overall AUROC ↑	Bottle Cap	Cable	Chair	Fork	Glove	SD Card	Tennis Ball	Wall
Ours (Histograms)	**0.84**	**0.83**	**0.87**	**0.80**	**0.84**	**0.87**	**0.76**	**0.86**	**0.89**
Histogram Peaks	0.63	0.61	0.65	0.61	0.63	0.66	0.57	0.71	0.62
Proprietary Distances	0.60	0.58	0.55	0.59	0.54	0.63	0.70	0.52	0.68

Table 3.1: **Forward-Facing Obstacle Detection Results**: Our method outperforms baselines overall and across each object.

3.5 Experimental Results

We perform real-world experiments with the AMS TMF8820 distance sensor to assess the performance of our method for detecting planar surfaces across many conditions and compare our method to baselines that utilize only a summary statistic for each histogram.

Implementation Details

Measurements are taken with an AMS TMF8820 sensor connected to an Arduino microcontroller via I^2C, which forwards the measurements to a connected computer. The sensor is set to 4 million iterations and a measurement period of 230ms, leading to an effective frame rate of $\sim$4.3FPS. KDE kernel bandwidth σ in Equation 3.1 is set to 5. Expectation-maximization for Gaussian mixture model fitting is done via the scikit-learn [87] GaussianMixture class. On a mid-range laptop, model fitting takes about 3 seconds on a dataset of 75 measurements, and inference runs at $\sim$108 FPS. We run the sensor in "low range, high accuracy" mode, which gives it a maximum range of 120cm. We limit our testing to within 80cm of the sensor and trim all sensor measurements $\mathbf{h}$ to bins in range $(13, 73)$ after the pre-processing step, corresponding to distances 0-80cm from the sensor. This range makes it more practical to capture measurements of a surface that is planar over the entire sensor FoV.

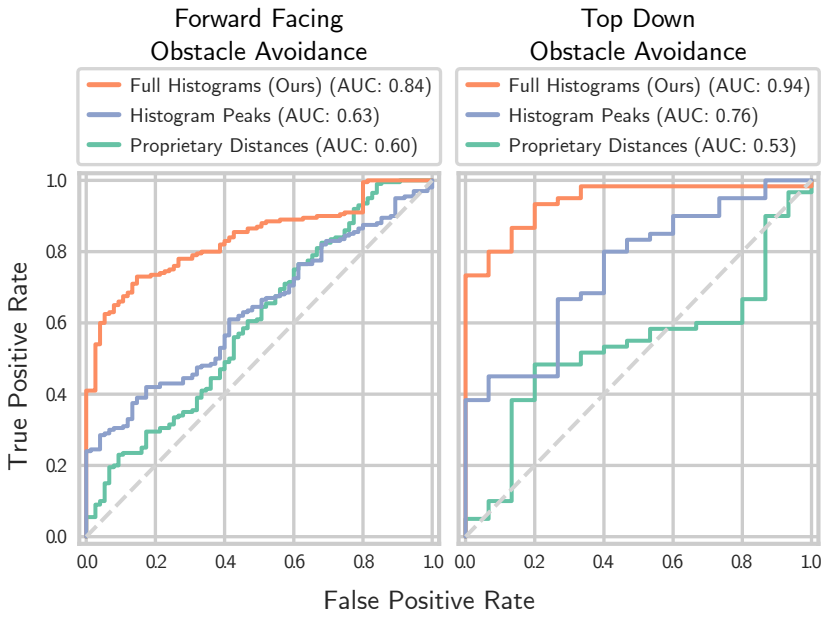


Figure 3.5: **Obstacle Detection ROC Curves**: Our method outperforms baseline methods on AUROC on our forward-facing and top-down obstacle detection datasets.

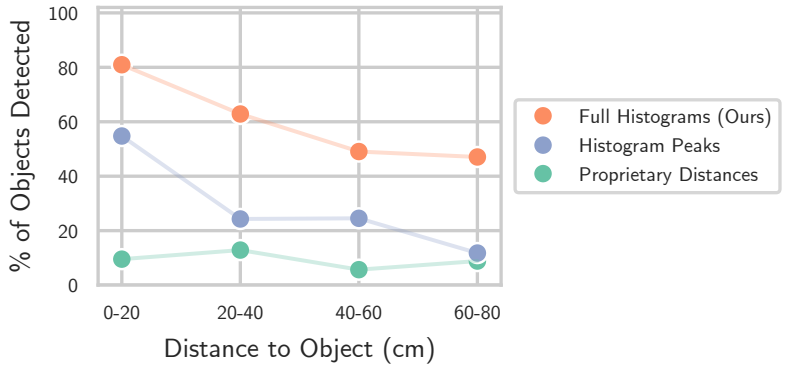


Figure 3.6: **Obstacle Detection Performance as a Factor of Distance to Obstacle**: Our method achieves a higher detection rate than baselines across the entire distance range. Detection threshold is limited to a $< 5\%$ false positive rate on the test set for fair comparison.

Forward-Facing Obstacle Detection

We capture a large dataset to emulate an obstacle detection scenario for a small mobile robot. We build a mount that holds the sensor 10cm from the ground and at a 60° angle-of-incidence to the surface, as in Figure 3.2, so that the top of its field-of-view is parallel to the ground. We capture measurements of five planar floor surfaces; for each surface, we capture 30 measurements of the surface with no deviations and 10 measurements each of 8 different objects placed at various distances between 10cm and 80cm from the sensor. The surfaces and objects are shown in Figure 3.4. Every capture is under artificial lighting, aside from the wood floor surface, which is under direct sunlight filtered through a window. In total, we capture 550 measurements. The sensor and objects are moved between every capture to ensure diverse coverage of the spatially varying surface. For every measurement, we also capture an RGB and depth image from an Intel RealSense D405 depth-from-stereo camera placed next to the sensor. These depth images are used to manually verify that the objects lie within the sensor's FoV during capture and to label the distance to each object.

Per-Object Performance

We use half of the captures of empty surfaces to fit a single surface model ($N = 15$ per surface, 75 total) as described in Section 3.4. This emulates a scenario in which the material of the planar surface is constrained, but not exactly known. The model selected by Equation 3.4 has $c = 10$ components. For every query measurement in the test set, Equation 3.3 is used to calculate the probability of the sample *not* being a measurement of a deviation-free planar surface. These per-sample probabilities are used to calculate the area under the receiver operating characteristic (AUROC) for the binary classification problem, and those results shown in Table 3.1. The ROC curves are shown in Figure 3.5. Our method leads to significantly higher AUROC than the baselines, meaning it is better at discriminating between measurements of a flat planar surface and measurements of a surface with an obstacle. When broken down by object, we find that our

method is marginally better at detecting large objects (*e.g.*, wall, glove) than small ones (*e.g.*, SD card, bottle cap).

Performance vs. Distance to Object

Using the same per-sample probabilities, we utilize the distance labels generated from the depth-from-stereo camera to evaluate obstacle detection performance as a factor of distance to the object. To ensure a fair comparison between methods, we restrict the detection threshold to generate a $< 5\%$ false positive rate on the test dataset. Obstacle detection accuracy as a factor of distance is shown in Figure 3.6. We find that, at all distances, our method outperforms baselines. We also observe that the distance to the object has a larger effect on detection rate than the object size. This may be because the amount of light returning from an object at distance d falls off at a rate of $1/d^2$, meaning SNR goes down quickly as d increases.

Performance as a Factor of Training Surfaces

To evaluate the performance of our method on out-of-distribution surfaces, we perform an experiment in which we vary the surfaces used to fit the surface model and split the AUROC by test surface. This means the model is trained on four surfaces, and must generalize to a fifth. We test three conditions: "All" training surfaces, which is the same surface model fit to 75 total measurements of all surfaces, as used for Table 3.1 and Figures 3.5 and 3.6; "Test only" training surfaces, in which only 15 measurements from the test surface are used to fit the surface model; and "All but test", in which 60 total measurements of all surfaces *except for* the test surface are used to fit the surface model. The results of this experiment are shown in Table 3.2. When the surface model is fit to all five surfaces, performance is uniform across each surface. When fit to and tested on one surface only, our method's performance increases significantly for most surfaces, as per-bin measurements have lower variance and the surface model is able to fit more tightly to the expected distribution. Performance is

	AUROC by Training Surfaces ↑ (# of samples)		
Test Surface	All (75)	Test Only (15)	All but Test (60)
Paper	0.85	**0.99**	0.60
Patterned Carpet	**0.83**	0.65	0.70
Solid Carpet	0.84	**0.92**	0.79
Tile Floor	0.84	**0.97**	0.48
Wood Floor	0.83	**0.91**	0.62
(Average)	0.84	**0.89**	0.71

Table 3.2: **Performance as a Factor of Surfaces Used to Fit the Surface Model**: Performance is highest when the test surface is the only surface in the training set (Test Only), is lower when the test surface is one of five in the training set (All), and is much lower when the test surface is not in the training set (All but Test).

reduced, however, on the highly spatially varying "patterned carpet", as fifteen measurements of the surface alone is not enough to capture a comprehensive range of surface textures. Lastly, when the surface model is fit to all surfaces *but* the test surface, we see a noticeable drop in the performance of our method, with the highest performance decrease present in the highly specular tile floor, and the lowest in the solid carpet. Performance on the tile floor likely suffers because it is specular, while all other surfaces are nearly fully diffuse. Meanwhile, the solid carpet is photometrically more similar to the other four surfaces, leading to better generalization.

Top-Down Obstacle Detection

We also apply our method to a top-down obstacle detection scenario, as might be utilized by *e.g.*, a drone finding a safe spot to land, or a manipulator finding a safe spot to place an object. In this experiment, the sensor is placed 28cm above the ground facing straight downwards. We use the "solid carpet" material from Section 3.5 and the bottle cap, cable, fork, glove, SD card, and tennis ball objects. Similar to the forward-facing obstacle detection experiment, 30 total measurements are taken of the empty surface, half of which are reserved

for fitting the surface model. Ten pictures of each object are taken, with the object moved within the FoV and the sensor moved relative to the surface for each capture. The model selected by Equation 3.4 has $c = 4$ components. The resulting ROC curves of each method are shown in Figure 3.5. Our results echo those of the forward-facing obstacle detection experiment: utilizing the entirety of the histograms makes it much easier to discern between planar surfaces and non-planar surfaces. Performance in this setting at this distance is similar to that achieved in our forward-facing obstacle detection experiment, when only the solid carpet surface is used to fit the surface model.

Cliff Detection

In principle, our method can detect any deviation in a planar surface, *e.g.*, a protrusion upwards, cliff, or ledge. To evaluate cliff detection performance, we gather a dataset in which the sensor is placed atop a wooden table, using the same forward-facing mount. We use 15 measurements of the empty tabletop to fit a model for the surface, and the sensor is placed at varying distances from the edge of the table in the range (5cm, 75cm) in 5cm increments. At each distance, four measurements are taken from various positions. An additional 15 measurements of the empty planar surface are reserved for testing to test false positive rate. Our method is able to detect cliffs 100% of the time up to 35cm away with no false positives. Cliffs 45cm away or further are not detected. Under the same conditions, the histogram peak-based method detects cliffs up to 25cm, while the proprietary distance-based method is only effective up to 10cm.

Ablation Study

We perform an ablation study to assess the importance of each aspect of our method on the forward-facing obstacle detection dataset. The results of this study are shown in Table 3.3. We find that each aspect of our method has some effect on performance, with exclusion of both pre-processing steps leading to a drop of

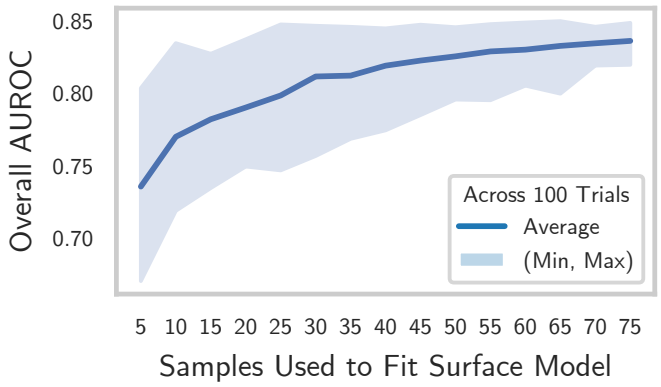


Figure 3.7: **Effect of varying the number of samples used to fit the surface model on AUROC on our forward-facing obstacle detection dataset**: The shaded region represents the minimum and maximum over 100 samples, while the solid line represents the average.

0.8 AUROC, and limiting the Gaussian mixture model to one component leading to a large drop of 0.21 AUROC. We believe that this is because bin values tend to be multi-modal, especially when the surface model is fit to a multiple-surface dataset, as different surfaces' varying reflectance properties mean that the shape of the histogram is consistent within a surface (aside from patterned surfaces), but varies between surfaces. By limiting the model to a single Gaussian component, the varying histogram shapes are not effectively modeled.

We perform an additional study in which we vary the number of samples used to fit the surface model for forward-facing obstacle detection. For a given number of samples, we pull an even number of samples from each of the five surfaces. For each number of samples, we repeat the experiment 100 times, each with a randomly sampled dataset for surface model fitting and the same test set. The results of this experiment are shown in Figure 3.7. We find that performance begins to level out as we approach 15 samples per surface (75 total). Reasonable performance is still possible with fewer samples per surface, *e.g.*, 5 samples per surface (25 total) yields an average AUROC of 0.80.

Method	AUROC ↑
Base	0.84
No Ambient Light Correction ($a_i = 0$)	0.82
No Normalization (modify Eqn. 3.2)	0.78
No Norm. or Ambient Light Correction (skip Eqn. 3.2)	0.76
Limit to one Gaussian Component ($c = 1$)	0.63
No Norm, No ALC, & Limit to One Component	0.53

Table 3.3: **Ablation Study Results on Obstacle Detection Dataset**

3.6 Example Application

We build an example application for our method in which a mobile robot is equipped with three distance sensors in a forward-facing configuration, providing a wide field-of-view. The sensor is in the same position relative to the ground as in our forward-facing obstacle detection and cliff detection experiments. We assume that the robot begins in an obstacle-free area and can safely drive forward for 10 seconds to characterize the surface. After capturing 30 measurements per sensor to characterize the ground surface and fit a surface model, the robot is able to avoid obstacles and cliffs using measurements from the sensors alone. The maximum range at which obstacles are detected is configurable by trimming the histogram bins per sensor during inference. A visualization of the application is shown in Figure 3.8. See youtube.com/watch?v=vtP5Ktp-oIo for a video demonstration.

3.7 Limitations and Conclusion

While we have shown that real-world deviations can be detected with reasonable accuracy, our method is still subject to the photometric-geometric ambiguity described in Section 3.3. Additionally, because of the presence of photometric effects, our method is most effective when the surface is well-known, and it is much less effective when an observed surface has not been previously seen.

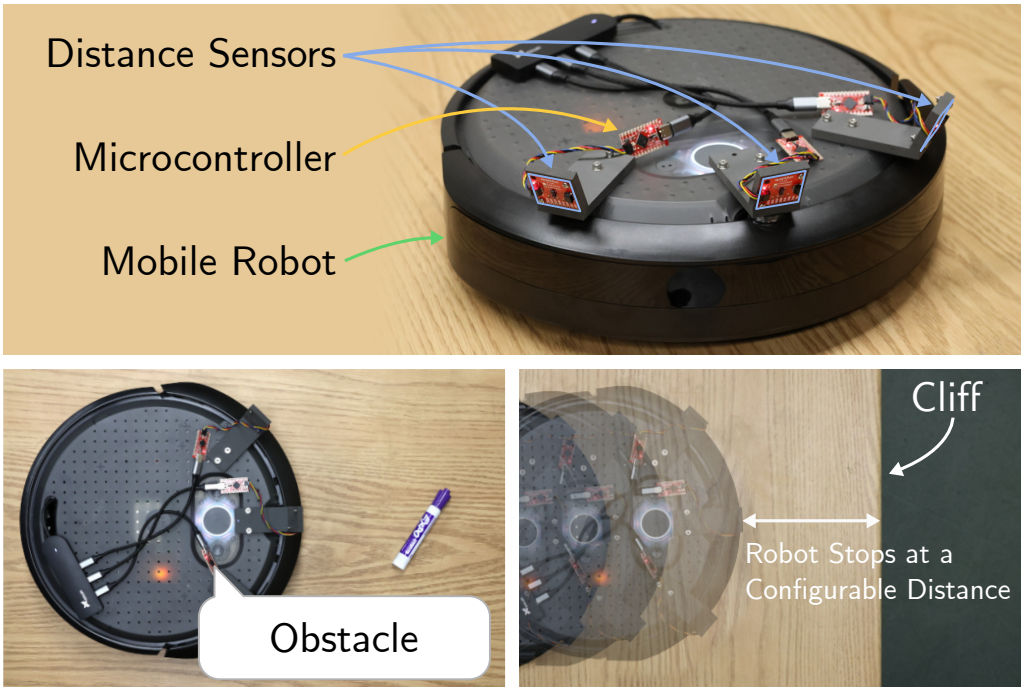


Figure 3.8: **Example Application of our Method being Applied to Mobile Robot Obstacle Avoidance**: The robot is equipped with three distance sensors. After characterizing the surface, it is able to avoid obstacles and cliffs in its path with a configurable buffer distance.

Future work should investigate ways to overcome this limitation by, *e.g.*, creating a general-purpose surface model by fitting to a large dataset of surfaces or utilizing an explicit photometric model to set bounds on the expected photometric deviations of the surface. Lastly, we assume that the relative orientation and distance to the planar surface is fixed. Future work should investigate the possibility of detecting surface geometry regardless of relative sensor pose and investigate robustness to subtle changes in sensor pose due to robot motion.

This work provides a way to extend the capabilities of distance sensors with no additional hardware and minimal compute overhead. We look forward to future robotics applications that make use of our method to improve robot sensing, particularly in resource constrained scenarios.

4 DETECTING OBJECTS NEAR A ROBOT MANIPULATOR

In this chapter, we provide a method for detecting and localizing objects near a robot arm using arm-mounted miniature time-of-flight sensors. A key challenge when using arm-mounted sensors is differentiating between the robot itself and external objects in sensor measurements. To address this challenge, we propose a computationally lightweight method which utilizes the raw time-of-flight information captured by many off-the-shelf, low-resolution time-of-flight sensors. We build an empirical model of expected sensor measurements in the presence of the robot alone, and use this model at runtime to detect objects in proximity to the robot. In addition to avoiding robot self-detections in common sensor configurations, the proposed method enables extra flexibility in sensor placement, unlocking configurations which achieve more efficient coverage of a radius around the robot arm. Our method can detect small objects near the arm and localize the position of objects along the length of a robot link to reasonable precision. We evaluate the performance of the method with respect to object type, location, and ambient light level, and identify limiting factors on performance inherent in the measurement principle. The proposed method has potential applications in collision avoidance and in facilitating safe human-robot interaction.

Project website: cpsiff.github.io/efficient_detection/

4.1 Introduction

Detection of objects near a robot arm is useful for tasks such as collision avoidance [113, 55] or to enable proximity-based human-robot interactions [28].

This work was completed under the supervision of Michael Gleicher and Mohit Gupta.

Externally mounted cameras are one way of detecting such objects, but they suffer from occlusion and require the robot to remain in view of the cameras, limiting their practicality when used with mobile manipulators. Therefore, we seek a solution which uses sensors mounted on the robot. Miniature time-of-flight (ToF) sensors [16, 107, 2, 108] are particularly attractive for for mounting on-robot because of their small size and low power consumption. In order to cover the space around the robot with a small number of sensors, we need the ability to choose efficient sensor configurations, such as placing the sensor peering down the length of an arm segment, as shown in Fig. 4.1F. However, miniature ToF sensors have very low pixel counts (*e.g.* 3x3) with a wide field-of-view (FoV) per-pixel (5° - 40°), meaning that in such configurations, the robot is constantly detected, and other objects can only be detected if they are closer than the robot is to the ToF sensor. As illustrated in Fig. 4.2, simple filtering of pixels which view the robot is ineffective in this case; therefore, to enable such sensor configurations, any approach to object detection must be able to differentiate between *self-detections* (the robot itself) and external object detections *within each pixel*.

In this work, we provide a method for detecting and localizing objects near a robot arm using miniature ToF sensors. Our method allows for flexibility in sensor placement and avoids self-detections of the robot. We address the key challenge of differentiating robot self-detections from other objects by implicitly modeling the expected appearance, reflectance, and geometry of the robot through sampling of raw ToF measurements. We utilize the raw ToF data captured by commonly used off-the-shelf sensors; at runtime our method finds differences between the measured ToF data and the expected appearance of the robot. Therefore, our method can detect objects even in sensor pixels for which the robot is prominent in view. This enables configurations such as that shown in Fig. 4.1F, which provide coverage of a small radius around the robot surface using few sensors. Our method also prevents self-detection in more typical outward facing sensor configurations as shown in Fig. 4.1C.

Our contributions are: 1) a method for detection and localization of objects

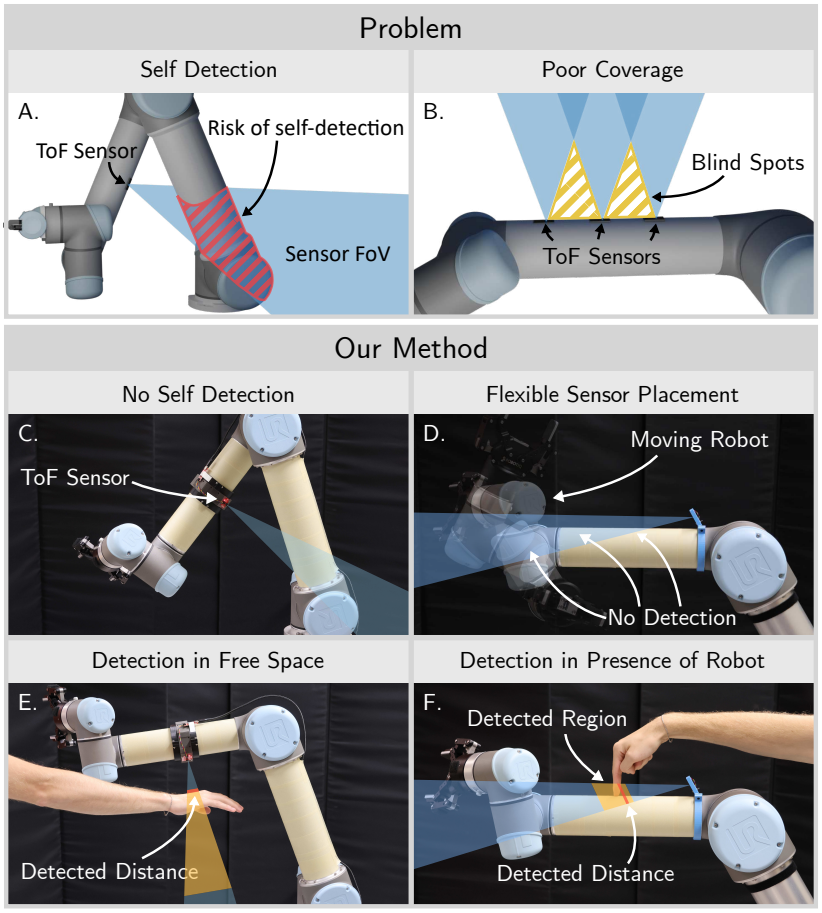


Figure 4.1: **Overview of Near-Robot Detection Problem**: Time-of-flight sensors attached to robot arms are prone to self-detection (A), and typical configurations provide inefficient coverage of the radius near the robot surface (B). Our method enables self-detection free proximity sensing, which enables new sensor configurations that provide more efficient coverage of a radius around the robot surface (C-F).

near a robot arm with a known joint state using a miniature ToF sensor while ignoring robot self-detections; 2) experiments demonstrating that our method is effective at detecting and estimating distance to objects with the configuration shown in Fig. 4.1F, and experiments investigating the limits and inherent constraints on the performance of our method; and 3) a live demonstration.

Scope and Limitations. While our method can scale to multiple sensors, in this work we build a prototype which includes one sensor at a time. Our demonstration shows live output of our method, but is not integrated with robot control for *e.g.* collision avoidance, and the sensor frame rate is limited to 3.5 Hz by the data interface of currently available sensors. Our method enables sensor configurations which efficiently cover a small radius around the robot surface, and is computationally efficient at runtime, but requires one-time overnight reference data capture per sensor position. Additionally, our method foregoes the need for any geometric calibration of sensor position, which is required by most alternative methods.

4.2 Related Work

Whole-Robot Proximity Detection

Research on robotic "sensitive skin" comprised of touch or proximity sensors dates back to the 1980s [119, 64]. While tactile sensors [130, 126, 60] are useful for collision detection and human-robot interaction, the ability to sense objects before touch occurs (proximity detection) enables a different set of applications including collision avoidance and safety in human-robot interaction. Systems have been proposed for whole-robot proximity detection, including those based on optical ToF [117, 34, 136, 46], or other sensing principles [30, 77]. These works do not address the problem of self-detection directly. Therefore, they are limited to outward facing sensor configurations. Our work demonstrates using flexibility in sensor placement to enable novel configurations that cover regions

efficiently.

Avoiding Self-Detection

Self-detection, when the robot itself is detected as an external object, is a challenging problem for robot manipulator perception systems due to the manipulator's dynamic shape during operation. There exists work on avoiding the self-detection problem when using an external depth camera which provides a 3D point cloud, by filtering out points belonging to the robot. Some approaches rely on extrinsic calibration between the camera and the robot, and use a 3D model of the robot to simulate its expected signal in the point cloud [94, 112]. Other works do not rely on extrinsic calibration, recognizing and removing the robot from the point cloud directly [124], or using temporal cues along with proprioception [66, 71]. These approaches are a reasonable solution for high resolution point clouds; when the robot points are removed, there is still sufficient information remaining to avoid collisions. However, with a low resolution sensor, filtering point clouds is not effective because the robot may be visible in all or nearly-all pixels. Therefore objects can only be detected when they are closer than the detected distance to the robot in a given pixel, severely limiting the effective detection area for some sensor configurations. A visualization of this limitation is shown in Fig. 4.2.

There is little prior work on avoiding self-detection with arm-mounted proximity sensors. The works which address the problem directly follow an approach similar to that used for point clouds. Avanzini *et al*. [13] place distance sensors on the links of a robot arm. To avoid self-detections, they use a 3D model of the robot to simulate the expected distance reading if only the robot were present. If the distance reading is less than the simulated reading, an object is detected. Himmelsbach *et al*. [40] use a similar approach. Such an approach means the sensor will not see objects within its FoV which are further than the simulated distance estimate, again being prone to the problem shown in Fig. 4.2.

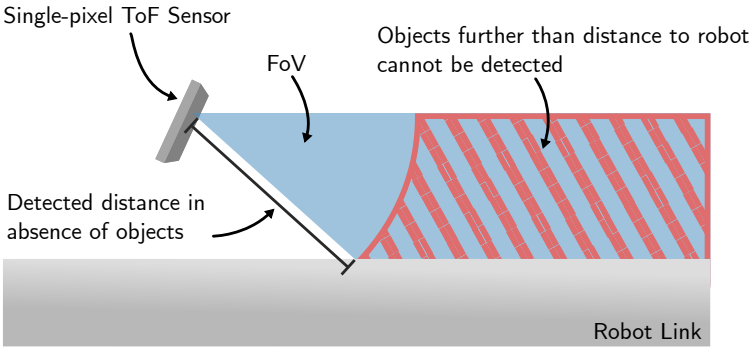


Figure 4.2: **Limitation of Distance Estimates for Object Detection**: When sensor measurements are treated as single distances per-pixel, objects further than the detected distance to the robot cannot be detected. This leads to significant blind spots, precluding sensor configurations such as the one shown. Our work enables such sensor configurations.

Miniature Time-of-Flight Sensors

Miniature ToF sensors are widely used in robotics due to their small size and low power requirements. Applications have been developed for the sensors mounted on miniature drones [118, 76, 137, 80, 43]. Other works place a sparse set of sensors around a robot [28, 46] or at a robot wrist [1], and build applications for collision avoidance. There exist methods which use the sensors to detect the 3DoF pose of a planar surface [107], and methods for calibrating the extrinsic position of a sensor attached to a robot arm [105]. This work builds on previous work which provides a method for detecting geometric deviations on a planar surface [106]. In contrast to this previous work, the method presented in this chapter works for articulated robots and non-planar surfaces, and is able to determine the distance to unknown objects.

There is a body of research which aims to make use of the raw ToF data from low-cost sensors akin to the one used in this work. Callenberg *et al*. [16] demonstrate in-contact material classification and, utilizing additional hardware, high-resolution imaging and non-line-of-sight tracking. Other works look to recover more detailed 3D information from the low-resolution measurements of

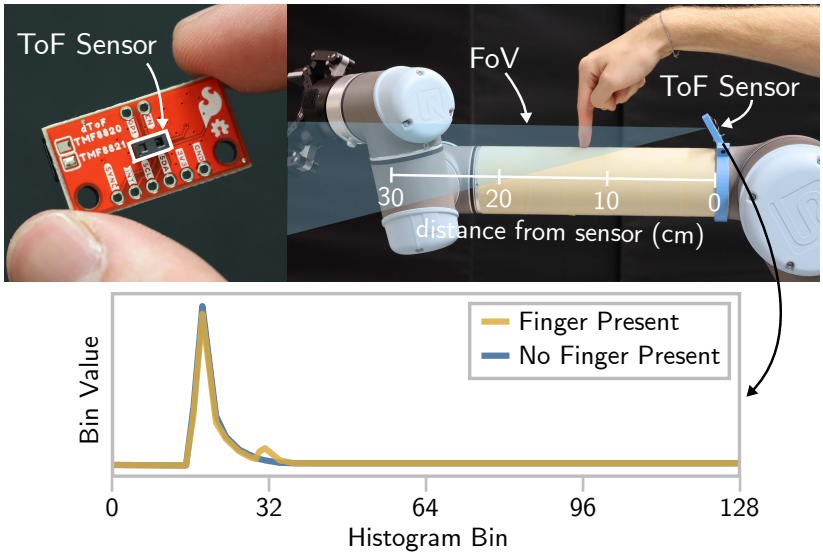


Figure 4.3: **Demonstration of Near-Robot Objects Captured in Histogram**: AMS TMF8820 sensor used in this work, and an example histogram captured by the sensor mounted on a robot arm. The large, leftmost peak is due to the surface of the robot link. The presence of the finger causes the appearance of an additional peak.

these sensors. There exist approaches for recovering 3D human pose [101], high resolution depth images [58], and general 3D reconstruction (from a distributed set of sensors) [75]. Miniature ToF sensors have also been used to refine monocular depth estimates [42, 81] and augment RGB SLAM [61]. Aforementioned work in robotics also takes advantage of raw ToF data [107, 106].

4.3 Problem Overview

Background: Direct Time-of-Flight

The miniature direct ToF sensors that we utilize operate by illuminating a patch of the scene with a pulse of (typically infrared) light and directly measuring the time of travel of the returning light at high (nano- to picosecond) time resolution. The

returning light waveform is called the scene *transience*, and the quantized version of that waveform recorded by the sensor is the *transient histogram* [42, 38]. The transient histogram is a function of scene geometry and reflectance properties (in addition to sensor and light-source characteristics) integrated over the FoV of the sensor which, for miniature sensors, is typically between 10 and 40 degrees per pixel. Single photon avalanche diodes (SPADs) [79, 133] are the most mature and widely available technology enabling direct ToF, and the basis of currently available miniature direct ToF sensors. These sensors are very small ($<$20 mm^3), lightweight ($<$1 gram), and power efficient ($<$10 milliwatts per measurement) [2, 108, 109].

Typically, miniature ToF sensors use an onboard algorithm to calculate a distance estimate from the transient histogram, which is reported. The goal of this work is to recover the distance to the closest point on unknown geometry while ignoring the robot itself. To accomplish this, our method utilizes the transient histogram directly, rather than on-sensor distance estimates. An example of a transient histogram is shown in Fig. 4.3. Using the transient histogram directly allows us to pick out subtle variations in the measured ToF signal in an individual pixel which are not contained in on-sensor distance estimates alone. In Sec. 4.5, we demonstrate that on-sensor distance estimates are not sufficient.

Problem Analysis

In order to estimate the distance to unknown objects in a transient histogram while ignoring the robot itself, we first must identify and remove the signal caused by the robot. In the case of an articulated robot, the geometry of the robot varies with respect to the robot joint state. Our goal is then to create a mapping from robot joint state to a probabilistic model of the expected transient histogram, as it would appear if only the robot were present. This mapping could be achieved in multiple ways. Previous work [75, 107] demonstrated an effective forward model for miniature direct ToF sensors, which allows simulation of a histogram measurement given scene geometry and reflectance

(*i.e.* surface albedo and specularity). However, in order to accurately simulate sensor measurements of the robot, the spatially varying reflectance properties of the robot would need to be known. Gathering such a measurement requires highly specialized equipment, making a simulation-based approach impractical. Further, previous work has established that detecting objects based on known geometry but unknown reflectance is fundamentally ambiguous under many settings [106, 42].

Rather than modeling the ToF signal of the robot explicitly, we utilize a data-driven approach in which measurements from the sensor are sampled at many robot states with only the robot present. Those measurements are used to create a probabilistic model of the expected transient histogram for any single joint state within the sampled range. Our method is applicable to any direct ToF sensor which reports a transient histogram. We evaluate our method using the AMS TMF8820, shown in Fig. 4.3.

4.4 Method

Given a b-bin transient histogram $\mathbf{h}_{\text{obs}} \in \mathbb{N}^b$ captured by a miniature ToF sensor attached to an n degree-of-freedom robot with joint state $\mathbf{q} \in \mathbb{R}^n$, we aim to recover the distance d to the point nearest the sensor on any object in the sensor FoV excluding the robot itself (and any attached accessories, *e.g.* a gripper). In practice, depending on the mounting position of the sensor, the sensor may not be able to see every link of the robot. In this case only degrees-of-freedom which affect sensor readings are included in $\mathbf{q}$.

Histogram Pre-Processing

Transient histograms are affected by ambient light, which manifests as a DC offset in the captured signal [36]. To avoid falsely detecting changes in ambient light as objects, we pre-process histograms by subtracting the DC offset and normalizing the area under the signal, following the approach of previous work [106]. For the histogram $\mathbf{h}$ the DC offset h_{offset} induced by ambient light is approximated by

finding the maximum kernel density on the values of $\mathbf{h}$, which acts as a robust way of estimating the modal value of $\mathbf{h}$. The kernel bandwidth σ is a tune-able parameter, which can vary by sensor model:

$$h_{\text{offset}} = \underset{x}{\text{argmax}} \sum_{h_i \in \mathbf{h}} \mathcal{N}(x; h_i, \sigma) \tag{4.1}$$

The area under the signal is normalized after h_{offset} is subtracted. The pre-processed histogram $\tilde{\mathbf{h}}$ is given by:

$$\tilde{\mathbf{h}} = \frac{\mathbf{h} - h_{\text{offset}}}{\|\mathbf{h} - h_{\text{offset}}\|_1} \tag{4.2}$$

Modeling Known Objects

To detect the distance to unknown objects in the FoV imaged by $\mathbf{h}_{\text{obs}}$, we rely on a probabilistic model of the per-bin mean $\mu_{\mathbf{q}} \in \mathbb{R}_+^b$ and per-bin variance $\sigma_{\mathbf{q}} \in \mathbb{R}_+^b$ of the expected histogram if only known objects were in the sensor FoV, given the current joint state $\mathbf{q}$. For reasons explained in Sec. 4.3, we approximate $\mu_{\mathbf{q}}$ and $\sigma_{\mathbf{q}}$ by interpolating between real samples over a range of possible $\mathbf{q}$ rather than an analytical approach.

Our method requires a set of per-bin histogram means $\mathbf{M}$, variances $\mathbf{V}$ and corresponding joint angles $\mathbf{J}$ which sample the robot configuration space. In practice, this dataset can be captured by, *e.g.* grid search or random sampling in configuration space. For each joint position sampled, multiple histograms are captured to generate a good approximation of $\mathbf{V}$ to capture sensor noise. Details of how we perform this sampling for real-world experiments are given in Sec. 4.5. To approximate $\mu_{\mathbf{q}}$ and $\sigma_{\mathbf{q}}$ from samples in $\mathbf{M}$ and $\mathbf{V}$, we perform barycentric interpolation, which requires finding a convex hull of $n+1$ points around $\mathbf{q}$ in $\mathbf{J}$. $\mu_{\mathbf{q}}$ and $\sigma_{\mathbf{q}}$ are then interpolated between the corresponding values in $\mathbf{M}$ and $\mathbf{V}$. The importance of interpolation and the density of samples needed to achieve good approximation of $\mu_{\mathbf{q}}$ and $\sigma_{\mathbf{q}}$ is investigated in Sec. 4.5.

Detecting Distance to Unknown Objects

Given $\mu_{\mathbf{q}}$ and $\sigma_{\mathbf{q}}$, we calculate the normalized probability vector $\mathbf{p} \in \mathbb{R}_{+}^{b}$ for $\mathbf{h}_{\text{obs}}$, which encodes the per-bin likelihood that a given bin is in the distribution expected when the robot alone is present, normalized so that for a given bin index i, when $h_{\text{obs},i} = \mu_{\mathbf{q},i}$ the likelihood is 1:

$$p_i = e^{\frac{-(h_{\text{obs},i} - \mu_{\mathbf{q},i})^2}{2(\sigma_{\mathbf{q},i})^2}} \tag{4.3}$$

To detect objects and estimate their distance, we transform $\mathbf{p}$ to a binary vector $\mathbf{g}$ which encodes bins that are likely to contain an unknown object. The threshold for detection t is a hyper-parameter which can be tuned to adjust sensitivity:

$$g_i = \begin{cases} 1 & \text{if } p_i < t \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

We then search for segments of the value 1 in $\mathbf{g}$ which span c or more contiguous bins, each of which corresponds to one detected object. For each segment, we extract the values of $\mathbf{h}$ over the corresponding range. We find the peak in these extracted values of $\mathbf{h}$. The position (bin index) of this peak corresponds to the distance to the detected object. We convert from bin index i_{peak} to distance using the conversion from bin index to distance for the TMF8820 sensor established by previous work [107]: distance (m) $= 0.01387 i_{\text{peak}} - 0.1825$. We empirically observe that this calibration is stable between multiple instances of the same sensor model.

TMF8820 Calibration

We observe that a varying bias is applied to TMF8820 histogram measurements between sensor power cycles. This bias can lead to false positives if the sensor is cycled off after the reference measurements ($\mathbf{M}$, $\mathbf{V}$ and $\mathbf{J}$) are captured. While we do not know the exact cause of this effect, or if it applies to other ToF sensors,

we are able to mitigate it by performing a one-off calibration step every time the sensor is powered on after reference capture. We move the robot to the first reference joint position $\mathbf{J}_1$ and capture a set of 50 measurements, which we average per-bin and store as $\mathbf{h}_{\text{ref}}$. We then calculate $\mathbf{h}_{\text{calib}} = \mathbf{M}_1 - \mathbf{h}_{\text{ref}}$. $\mathbf{h}_{\text{calib}}$ is stored and at query time is added to $\mathbf{h}$ prior to Eq. (4.1).

4.5 Experimental Results

Implementation Details

We perform a series of experiments in which a TMF8820 sensor is attached to link two of a Universal Robots UR5 robot arm. The sensor is positioned facing the end effector, as shown in Fig. 4.3. In this position, sensor readings are invariant to movement in the three most proximal joints. Thus, we only sample the 3DoF of the three most distal joints (*i.e.* those comprising the wrist) to capture reference histograms. Each experiment aside from Sec. 4.5 relies on the same reference dataset, which is captured over a 3D grid in joint space, in which $\mathbf{q}_4 \in [-\pi, -\pi/12]$, $\mathbf{q}_5 \in [-5\pi/6, 5\pi/12]$, $\mathbf{q}_6 \in [-\pi/2, 5\pi/12]$. Joint positions are sampled in $\pi/12$ radian increments, for a total of 2,304 joint positions. It takes ~ 10 hours to programmatically capture 50 measurements per joint position. The brushed aluminum surface of the UR5 robot is highly specular. While our data-driven approach models the effect of the specular surface when the robot alone is present, when other objects are present outside of the sensor field-of-view the surface acts like a mirror. This leads to false detections when objects outside of the sensor FoV are detected via three-bounce paths. We cover the metallic surfaces of the robot in masking tape to minimize this effect, and further investigate in Sec. 4.5

We capture data from an AMS TMF8820 sensor connected to an Arduino microcontroller, using the microcontroller code provided by prior works [107, 75, 106] to extract both distance and histogram measurements. The TMF8820 reports 9 histograms over 9 non-overlapping zones, for a total FoV of 30° diagonally. For

simplicity and to limit the FoV to avoid unwanted detections, we utilize only one zone of the sensor, yielding a 10° diagonal FoV. Sensor frame rate is limited by the speed at which the I^2C interface, which is not designed for histogram data capture, can transmit histograms, so we modify the microcontroller code to only report bins 1-80 to increase frame rate. This means the maximum range of the sensor as configured is ~ 90cm. The sensor reports measurements at 3.5 FPS. The execution of our algorithm takes 0.35 ms (2803 FPS) on a mid-range laptop CPU (Intel i5 1340P), and the runtime scales linearly with the number of sensors used. In our testing, interference between multiple TMF8820 sensors is minimal, making them well-suited to future systems with many sensors.

Unless otherwise stated, we set the probability threshold $t = 0.001$, and minimum segment size $c = 4$. These values were manually tuned to create a reasonably low false positive rate for the main experiments. We investigate the effect of changing these parameters in Sec. 4.5. A peak in bin $\sim$14 from the TMF8820 corresponds to an object at distance zero; therefore we trim the histogram $\tilde{\mathbf{h}}$ to bin range $(15, 80)$ before applying Eq. (4.3). We set σ in Eq. (4.1) to 5, following previous work [106]. In each experiment, we only consider the *closest* detection. All captures are in a windowless room with fluorescent lights (~ 500 lux).

Self-Detection Rate

We perform an experiment to understand the effect of joint-space sampling density on the rate at which the robot is falsely detected. The results of this experiment are highly dependent on robot geometry and sensor position. The experiment serves to provide a rough approximation of performance in general, and provides context for other experiments, which use the same robot and the same sensor position.

We capture a dataset of ToF measurements from 1000 uniformly sampled random joint positions within the joint range of the reference dataset, with only the robot present. The self-detection rate (false positive rate) is the rate at which

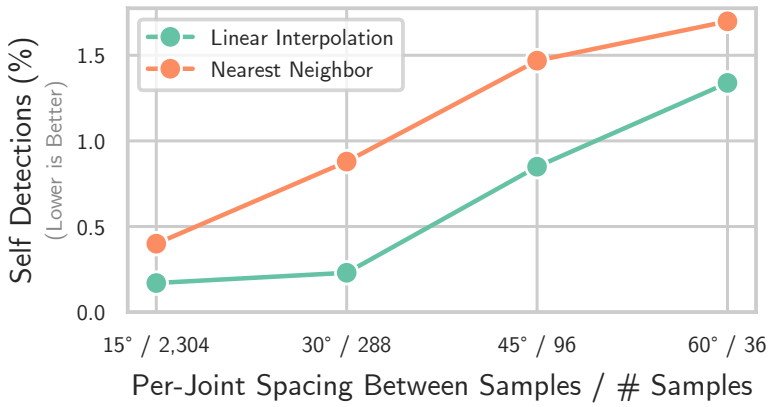


Figure 4.4: **Effect of Joint-Space Sampling Density on Self-Detection (False Positive) Rate.** Linear interpolation of background histograms between nearby joint states outperforms nearest neighbor interpolation (lower is better). 3 DoF of robot wrist joints are sampled.

detections occur in this dataset. To investigate the effect that sampling density has on self-detection rate, we sub-sample the reference dataset by a factor of 2, 3, and 4 per dimension, creating a coarser grid of samples on which the model is built, and plot the effect on self-detection rate in Fig. 4.4. Self-detection rate increases as the density of joint-space samples decreases, and linear interpolation leads to a lower self-detection rate than nearest neighbor interpolation. Self-detection rate levels off at high sampling densities; we hypothesize that this is because in rare cases measurement noise leads to self-detections, and measurement noise is constant regardless of sample density. A coarser sampling of every 30 achieves similar performance to 15 while requiring an order of magnitude fewer samples. Coupled with future sensors with a higher frame rate, this means that reference data capture could be made orders of magnitudes faster for little performance penalty.

True Positive Rate

To evaluate the true positive rate of our method (*i.e.* the rate at which an object is detected when one is present), we capture a dataset of ToF measurements

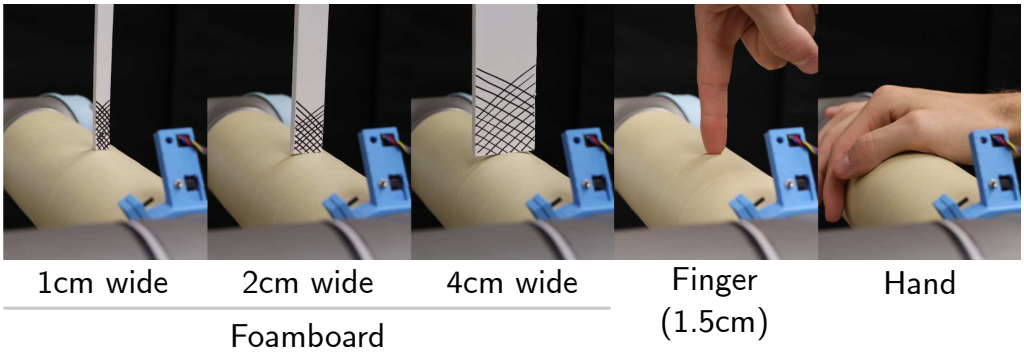


Figure 4.5: **Objects Used in Experiments, Shown as Placed on the Robot for Data Capture**: Pieces of foamboard have a hatching pattern applied to provide visual features for ground-truth depth-from-stereo camera.

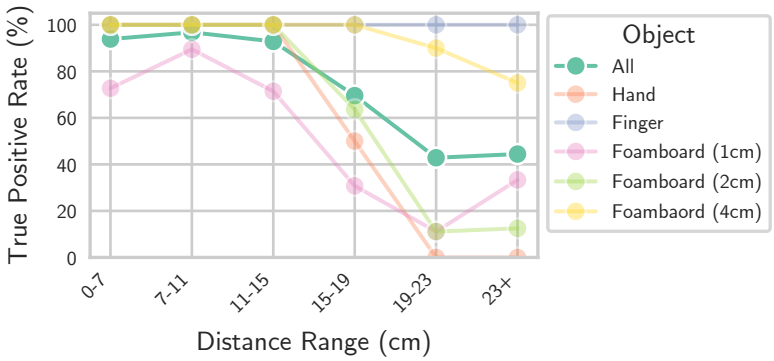


Figure 4.6: **True Positive Rate as a Factor of the Distance from the Sensor to the Object, by Object Type**: Note that due to random object placement, each data point may not represent the same number of samples.

in which objects are touching or nearly touching the robot arm. Between each measurement, the robot is moved to a random uniformly sampled joint state within the bounds of the reference dataset, and moved the object to a random distance from the sensor along the robot arm (1-28cm from the sensor); random distances are used to make data capture faster, ultimately allowing a larger dataset. We utilize five objects: a human pointer finger, a human hand, and long pieces of white foamboard cut to 1cm, 2cm, and 4cm in width. The objects are shown in Fig. 4.5. The finger and hand were captured touching the robot arm, while each

piece of foamboard was captured at 0cm, 1cm, and 2cm proximity from the arm itself. In total, the dataset contains 275 captures, each with varying conditions (object, distance to sensor, and proximity to arm).

We achieve a true positive rate of 78.9%; this is broken down by object and distance to the sensor in Fig. 4.6. The 4cm wide foamboard and pointer finger are the easiest to detect at all distances, while the narrower foamboard and hand are the most difficult. The hand, 1cm, and 2cm-wide foamboard are rarely recognized beyond 19cm from the sensor. We hypothesize that the difference between objects is due to a difference in their cross-sectional area and geometric deviation from the robot. While the hand is a large object, its cross sectional area is small from the point-of-view of the sensor when the hand is resting flat on the robot, and the hand does not extend far above the robot surface, leading to a relatively small change in the histogram. Object albedo might also play a factor; an object which is much brighter or darker than the robot will cause a larger change in the measured histogram than one with the same albedo. Lastly, non-line-of-sight effects (see Sec. 4.5) caused by the rest presence of the wrist and rest of the arm above the finger may make it easier to detect than the narrow foamboard. Future work should aim to isolate and negate the cause of the performance gap between objects.

Distance Estimation

We use the same dataset as in the previous subsection (Sec. 4.5) to evaluate the accuracy of our distance estimate. Ground truth distance labels are captured via an Intel Realsense D405 depth-from-stereo camera positioned next to the ToF sensor. The closest point on each object is labeled, and the depth estimate extracted from the D405 depth image. Objects closer than the minimum depth range of the camera are labeled by overlaying an image of the robot arm with ruler marks onto the captured RGB image of the object.

Fig. 4.7A compares the distance estimate from our method to the actual distance. We achieve an average absolute error of 2.08cm. Our method under-

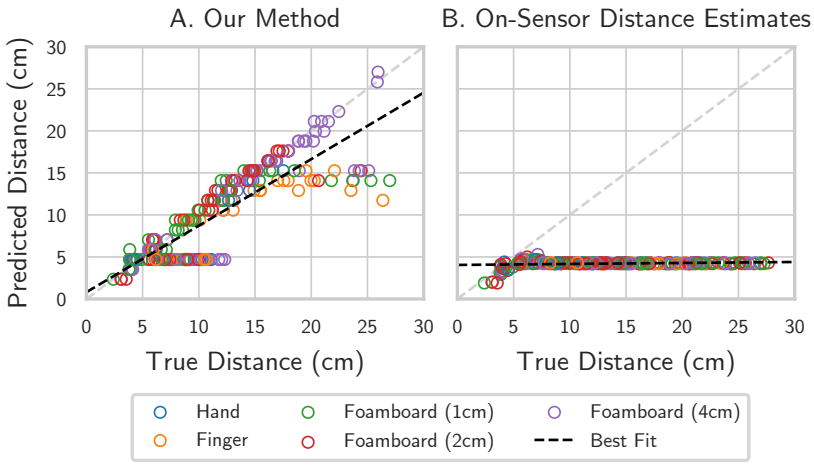


Figure 4.7: **Distance Detection Results**: A. Actual distance vs. distance predicted by our method. We achieve an average distance error of 2.08cm. B. Actual distance vs. distance predicted by a baseline method which utilizes on-sensor distance estimates. The baseline method never estimates a distance further than 5cm due to the limitation illustrated in Fig. 4.2.

estimates the distance to objects in some cases. One case is when a nearby object fills a large portion of the FoV, completely changing the shape of the histogram. When this happens, it is difficult to align the observed histogram to the reference to localize the deviation. Our method also sometimes under-estimates the distance to far away objects. We hypothesize that this could be due to a low signal-to-noise ratio, and/or the presence of the dynamic robot wrist links at those distances.

On-Sensor Distance Estimation

We compare the distance estimation results achieved by on-sensor distance estimates to our method. The TMF8820 reports up to two distance estimates per-zone, corresponding to up to two objects in the FoV. For each sample we choose the distance estimate of the two which achieves the lowest absolute error from the ground truth to demonstrate the best case for on-sensor distance estimates. In Fig. 4.7B, we show the distance estimated by this method compared to the true

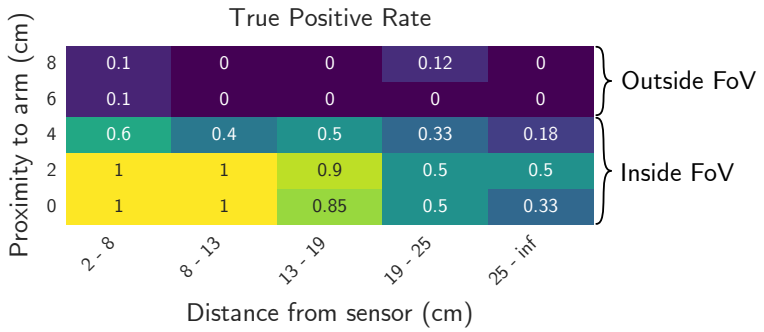


Figure 4.8: **Detection Rate as a Factor of Object Distance from the Sensor and Proximity to the Arm**: We see that objects beyond 4cm proximity to the arm are rarely detected, and that the proximity of detection is consistent regardless of distance from the sensor. This well-defined field-of-view is desirable for downstream applications.

distance. The distance estimates never exceed 5cm, roughly the distance to the nearest point on the robot. This experiment demonstrates the limitation illustrated in Fig. 4.2 and makes it clear that the results achieved by our method cannot be achieved using on-sensor distance estimates alone.

Sensor Field-of-View

We characterize the FoV of the sensor by placing the 4cm and 2cm foamboard at varying distance from the sensor and proximity to the arm and plot the TPR per object position in Fig. 4.8. The sensor is 4cm from the robot surface, with the top edge of the FoV aligned with the surface. Accordingly, we see that detection is less likely at 4cm from the surface and much less likely at 6cm+. The maximum detection proximity is consistent across all distances from the sensor, making the sensor configuration effective for detecting objects that come within 4cm of the robot surface.

Condition	TPR (↑)	FPR (↓)
Base	0.789	0.002
No Preprocessing	0.785	0.002
No Calibration	0.996	0.917
No Bin Trimming	0.938	0.706

Table 4.1: **Ablation study**: Calibration and bin trimming are necessary to avoid high false positives. Preprocessing has no effect as the test dataset does not contain changes in ambient light.

Non-Line-of-Sight Objects

Direct ToF sensors are subject to non-line-of-sight (NLOS) effects, which occur when photons bounce multiple times before returning to the sensor, as illustrated by the blue path in Fig. 4.9. As previously noted, this effect is very noticeable with the brushed aluminum robot surface. We cover the arm with masking tape to lower this effect for our experiments, but it is still present. In this subsection, we investigate the prevalence of NLOS effects in our setting.

We investigate NLOS effects over the previously captured dataset (Sec. 4.5). We treat objects outside of the direct FoI of the sensor (*i.e.*, 6cm and 8cm proximity to the arm; the top two rows in Fig. 4.8) as NLOS objects. We compare false positive rate between this set of objects and the case where only the robot is present. As shown in Fig. 4.10, false positives are much more likely when an NLOS object is present. The NLOS effect limits the performance of our system when the detection threshold is limited to not detect NLOS objects; increasing the sensitivity to detect more distant objects comes at the expense of increased detection of NLOS objects. On the other hand, for some applications, detection of NLOS objects may be welcome. In such cases the sensitivity can be increased to detect more distant line-of-sight objects with little increase in false positives in the absence of any objects.

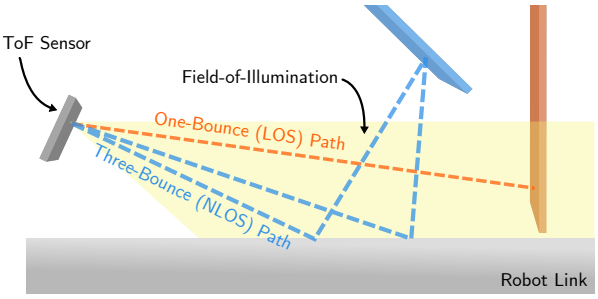


Figure 4.9: **LOS-NLOS Ambiguity**: There is an ambiguity between distant line-of-sight (LOS) objects and nearer non-line-of-sight (NLOS) objects outside of the sensor field-of-illumination. This limits the performance of our method on distant objects. If the detection threshold is lowered to determine distance LOS objects, nearer NLOS objects will also be detected.

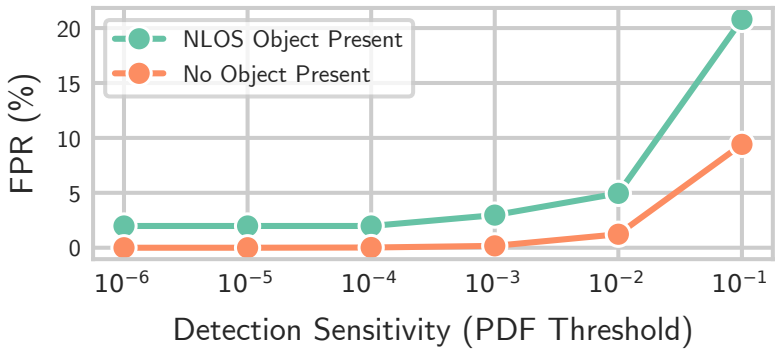


Figure 4.10: **False Positive Rate Compared Between NLOS Object Present and no NLOS Object Present**: When there is a large object outside of the sensor field-of-view, false positives are more likely due to three-bounce paths from the object.

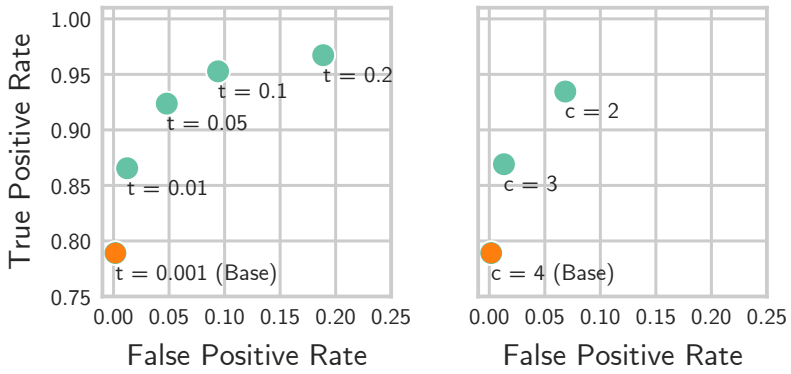


Figure 4.11: **Parameter Tuning Demonstration**: Changing method parameters (t or c described in Sec. 4.4) leads to different operating points on the ROC curve.

Parameter Tuning and Ablation Study

Tuning the parameters t and c (see Sec. 4.4) allows for a tradeoff between false positive rate and true positive rate. We use the dataset from Sec. 4.5 to test false positive rate and that from Sec. 4.5 (containing objects 0cm, 1cm, and 2cm from the robot surface) to test true positive rate. The operating points achieved by varying t and c are shown in Fig. 4.11. Varying these parameters allows for tuning of the FPR/TPR trade-off for different scenarios. However, as shown in Fig. 4.10, increasing the sensitivity of the method also increases the detection rate of non-line-of-sight objects.

We ablate components of our method and show the effect on true positive rate and false positive rate in Tab. 4.1. We find that calibration (Sec. 4.4), and bin trimming are necessary to prevent a high false positive rate. Normalization (Eq. (4.1)) has no significant effect on the results because the test dataset does not exhibit strong changes in ambient light.

Performance Under Varying Ambient Light

Ambient light affects the histogram, potentially leading to a false positive if ambient light level changes after reference capture. To evaluate our method under

		FPR (↓)	
Lighting	Lux	Base Method	No Pre-Processing
Dark	< 0.1	0.0099	0.0198
Dim LEDs*	100	0.0198	0.0198
Flourescent Lights	500	0.0099	0.1089
Halogen Lights	1000	0.1782	1.0000

Table 4.2: **False Positive Rate Under Varying Ambient Light**: Poor performance is observed under bright halogen lights, and performance is worse with no pre-processing (Eq. (4.2)). * Dim LED lights match the lighting used during reference capture.

such changes, we capture new reference captures at the same joint locations as previous experiments. For each of four ambient light levels, 100 measurements of the robot are captured at random joint positions with only the robot present. We calculate the false positive rate of our method at each of these four light levels. The results are shown in Tab. 4.2. Moderate changes in ambient light from the reference capture (which was taken under the "Dim LED" lighting) do not lead to an increase in false positive rate. The bright, IR-heavy halogen light source leads to a sharp increase in false positive rate. This means our method is impractical when ambient light changes significantly (*e.g.* moving from indoors to sunlight). Additionally, we observe that the pre-processing steps described in Sec. 4.4 are somewhat effective at improving performance under high ambient light. It is possible that more sophisticated methods for reversing the effect of ambient light [36] could improve the performance of our method in these scenarios, but they require an accurate model of sensor hardware which is not available with current commercially available sensors.

4.6 Demonstration

We demonstrate our method on a UR5 robot arm. As objects approach the arm, they are detected and their position shown overlaid on an image of the robot as

a yellow region corresponding to the range of the detected segments, and a red line for the detected distance, shown in Fig. 4.1E and Fig. 4.1F. In addition to the configuration used for experiments (Sec. 4.5), we demonstrate an additional configuration in which the sensor is positioned orthogonal to the robot surface. The only change made to accommodate this configuration is that joints 2 and 3 of the robot are sampled for reference measurements rather than joints 4, 5, and 6. See youtu.be/YXwGyWGNJhg?t=66 for a video version of the demonstration. This demonstration shows that our method is effective under multiple sensor configurations with minimal adjustments.

4.7 Conclusion and Future Work

The work in this chapter demonstrates that it is possible to use raw ToF measurements to extract information about objects near a robot arm that would be impossible to obtain from distance estimates alone. We see poor performance on small objects when they are placed beyond $\sim$ 15cm from the sensor, limited partially by non-line-of-sight effects. Future work should investigate fusing sensor measurements to resolve the ambiguity between line-of-sight and non-line-of-sight objects. Our method could also be made more practical. We sample joint space uniformly to capture reference data, which is inefficient; future work should explore adaptive sampling or vary sampling density based on robot geometry. Creating a robot system which utilizes our method will require improved sensing hardware design (*i.e.* PCBs and integrated wiring) to decrease footprint and increase frame rate, in addition to integration with kinematics and control algorithms, akin to [28, 46, 93], to enable human-robot interaction and collision avoidance. A full system could additionally leverage temporal filtering to improve detection stability and performance. Further work should also further investigate performance as a factor of the sensor placement and the geometry and reflectance properties of detected objects. We believe the method presented in this chapter is a step towards whole-body proximity sensing with minimal hardware sensing cost.

5 RECOVERING PLANAR GEOMETRY

In this chapter, we provide methods which recover planar scene geometry by utilizing the transient histograms captured by a class of close-range time-of-flight (ToF) distance sensor. Typically, a sensor processes the transient histogram using a proprietary algorithm to produce distance estimates, which are commonly used in robotics applications. Our methods utilize the transient histogram directly to enable recovery of planar geometry more accurately than is possible using only proprietary distance estimates, and consistent recovery of the albedo of the planar surface, which is not possible with proprietary distance estimates alone. This is accomplished via a differentiable rendering pipeline, which simulates the transient imaging process, allowing direct optimization of scene geometry to match observations. To validate our methods, we capture 3,800 measurements of eight planar surfaces from a wide range of viewpoints, and show that our method outperforms the proprietary-distance-estimate baseline by an order of magnitude in most scenarios. We demonstrate a simple robotics application which uses our method to sense the distance and angle-of-incidence of a planar surface from a sensor mounted on the end effector of a robot arm.

Project website: cpsiff.github.io/unlocking_proximity_sensors/

5.1 Introduction

Optical time-of-flight proximity sensors which measure scene *transients* have recently become widely available. These sensors operate by illuminating the scene with a pulse of light, and measuring the *shape* of that pulse over time as it returns back from the scene in a *transient histogram*. These *transient sensors*

This work was completed under the supervision of Michael Gleicher and Mohit Gupta. Yeping Wang programmed the robot arm for the live demo, and helped set up and capture the live demo. The rest of the work was completed by Carter.

have seen use in robotics due to their ability to reliably report a distance estimate over a wide range (1cm - 5m) while being small (< 20 mm^3), lightweight, and low-power (on the order of milliwatts per measurement) [2, 109]. Because of their form factor, transient sensors can be placed in locations where higher resolution 3D sensors cannot, such as on the gripper or links of a robot manipulator, or on very small robots. While these sensors have many desirable properties, existing robotics applications do not utilize the transient histograms, instead relying on low-resolution (at most 4×4 pixel) proximity measurements generated onboard the sensor. Due to the coarseness of their measurements, these sensors are presently only used in robotics for coarse sensing, *e.g.,* detecting the presence of obstacles or distance to a target.

In this work, we utilize transient histograms directly to recover accurate planar scene geometry, and consistent planar albedo from a single 3×3 transient sensor measurement. Planar geometry is an initial use case for our methods, and is a special case of 3D sensing that has many applications in robotics. A robot interacting directly with any planar surface will benefit from sensing the geometry of that surface accurately and at a close range. For example: a robot arm placing an object on a tabletop, sweeping a floor, or writing on a flat surface; a mobile robot localizing the floor and walls of a room; or a drone finding a safe spot to land. Our method enables accurate recovery of this planar geometry that otherwise would have required multiple proximity sensors or a depth camera, while maintaining the same very small form factor and operating at ranges as low as 1cm.

This work is the first to demonstrate that utilizing transient histograms can improve the performance of proximity sensors over utilizing proprietary on-sensor distance estimates. To achieve this, our contributions are 1) an effective *forward imaging model* for commodity proximity sensors, 2) a *differentiable rendering pipeline* which implements the forward imaging model and utilizes it to recover planar geometry and albedo directly from transient histograms, 3) an *empirically calibrated approach* which approximates the performance of the

differentiable rendering pipeline and acts as a baseline, and 4) *empirical evidence* that our approaches outperform alternative methods which do not utilize transient histograms.

We present two methods for recovery of planar geometry, one of which can also be used to consistently recover the albedo of the planar surface. To evaluate our methods, we gather thousands of measurements of eight planar surfaces with a commodity transient sensor from a range of angles-of-incidence and distances. We find that our methods which utilize the transient histogram are more accurate and robust than those which rely on proprietary distance estimates. We also find that our method recovers consistent planar albedo, which is not possible to recover from proprietary distance estimates, as they do not encode intensity information. We build a demonstration application which takes advantage of the small size of a transient sensor by mounting the sensor to the gripper of a robot arm. Measurements from the sensor are used to measure the distance to the surface below the gripper and to ensure that the surface is level before placing an object.

5.2 Background: Transient Histograms

A *transient* is a one dimensional temporal waveform which measures the light reflected from a scene over time in response to a pulsed light source. Recently, sensors which are able to capture a transient quantized over short (picosecond) time scales have become available for distance/range measurement using the time-of-flight principle. We refer to these sensors as *transient sensors*. These sensors come in a range of form factors: from high resolution lab-grade arrays to mobile device LiDAR modules, to very small proximity sensors. Most notable of the currently available transient sensors is the single photon avalanche diode (SPAD) [22, 88], which is inexpensive and commonly used in robotics (see Sec. 5.3).

A SPAD-based sensor approximates the transient histogram through a repeated process. A controlled pulse of light (typically infrared) flood-illuminates the scene

in front of the sensor. Each sensor pixel records the elapsed time between this pulse being sent and a single photon arriving at the pixel. This arrival time is quantized to a discrete bin and accumulated in a *transient histogram*. Over many photon arrivals, this histogram approximates the true transient. In practice, a commodity sensor may record millions of photon arrivals to form a transient histogram. In sensors with an array of pixels, a transient histogram is generated for each pixel.

Currently available commodity transient sensors have many desirable properties. Many are capable of gathering transient histograms at 30 frames per second. Maximum range varies by model, but may be as high as 5m, with a typical minimum range of 1cm. There exist techniques for mitigating the effects of high ambient light on these sensors, enabling their operation in diverse environments [86, 35].

In this work, we evaluate our method using the SPAD-based TMF8820 sensor manufactured by AMS. We choose this sensor because it 1) allows access to transient histograms through an official driver, 2) captures a 3×3 grid of transient histograms at a time, each from a different region of its field-of-view, and 3) provides access to a "reference histogram" which encodes the intensity of the laser pulse over time. In the sensor's default configuration, transient histograms are summarized onboard the sensor via a proprietary algorithm, and a distance and confidence estimate are reported for each field-of-view region. We reconfigure the sensor to report a transient histogram *and* proprietary distance estimate for each FoV region. While we utilize the TMF8820 in this work, the methods we propose can be applied to any sensor which reports a transient histogram.

5.3 Related Work

Transient Sensors in Robotics

Transient sensors are widely used in robotics applications as they provide highly reliable distance measurements, while being lightweight, low-cost and low-power.

Tsuji and Kohama [117] demonstrate a "sensitive skin" for a robot arm consisting of many single-pixel transient sensors. Similarly, Adamides et al. [1] propose an array of transient sensors mounted around a robot wrist to achieve safe human-robot collaboration. Escobedo et al. place transient sensor on robot joints and use them to actively avoid collisions [28]. Transient sensors have been used to detect obstacles when mounted on a drone [118]. Our previous work characterized two transient sensors and demonstrated a method for extrinsically calibrating their position relative to a robot arm to which they are attached [105]. Outside of robotics, commodity transient sensors have seen use in wearable computing [103] and inspection applications [69]. In these prior works, only the sensor's proprietary distance estimates are utilized. To our knowledge, our work is the first to utilize the transient histogram in a robotics setting.

Inference from Transient Histograms

Our differentiable rendering pipeline and forward imaging model are heavily inspired by prior work in imaging. Photon arrival times, like those encoded by a transient histogram, are heavily utilized in non-line-of-sight (NLOS) imaging, pioneered by Velten et al. [123]. In NLOS imaging, scene geometry is recovered from around the corner by reflecting a powerful pulsed laser off a diffuse surface. Recent NLOS works utilize the same single photon avalanche diode (SPAD) technology as the sensor that we use in this work [14, 104]. However, the imaging setup used in NLOS imaging requires a high-powered laser and relatively large, expensive laboratory grade SPAD sensors, which have thousands of histogram bins and very precise timing. In contrast, the sensor that we use in this work is readily available, small, lightweight, and eye safe, but reports only 128 histogram bins, and has less precise timing and optical characteristics.

A number of recent papers have utilized transient histograms from commodity SPADs to perform scene inference. Each of these works uses sensors which are very similar to the one used in this chapter in terms of technology, form factor, and cost. Callenberg et al. [16] propose the use of transient histograms from

a SPAD to classify materials based on subsurface scattering (with the sensor placed in direct contact), generate higher resolution depth imagery, and perform non-line-of-sight imaging (with additional hardware). Becker and Koerner [6] also classify materials, but in a non-contact setting. Ruget et al. [101] perform super resolution and use supervised machine learning to estimate human poses from transient histograms. Other works also perform super resolution to resolve higher resolution depth images from relatively few transient histograms [9, 100].

The differentiable rendering approach used in this work is inspired by Jungerman et al. [42], who use differentiable rendering to recover partial plane parameters from a single transient histogram. Because the sensor used by Jungerman et al. reports only a single transient histogram, only two of the three planar degrees of freedom could be recovered from a single sensor measurement. In contrast, the multiple transient histograms reported by the sensor used in this work enable recovery of all plane parameters from a single measurement, and our work is the first to do so.

5.4 Forward Imaging Model

An accurate forward imaging model is crucial to enabling our differentiable rendering method. In this section, we give an overview of our forward imaging model, which is designed for the TMF8820 sensor, but can in principle be adapted to other sensors. Our model assumes planar scene geometry, with uniform albedo and reflectance model parameters per-plane. For a more general forward imaging model that is sensor agnostic, refer to previous work [42]. We consider a set of transient histograms which are simultaneously captured by a transient sensor over different fields-of-view. We refer to this set of histograms as an *image* Φ. Each image consists of n histograms $\varphi \in \Phi$. Each histogram consists of m bins, $\varphi_i : 1 \leq i \leq m$.

Surface Reflection Model

We utilize the Phong reflection model [89], in which a surface's reflection properties are parameterized by its albedo α, specular exponent k_e, and specular weight k_s. We assume that the light source and sensor are co-located, the pulsed laser source is the only light in the scene, and the strength of illumination is uniform over the field of view. The intensity I of incident light returned by a ray $\mathbf{r} \in \mathbb{R}^3$ intersecting with plane $\mathbf{ax} + d = 0$ is given by:

$$I = \alpha * (1 - k_s)(\mathbf{r} \cdot \mathbf{a}) + k_s((2(\mathbf{r} \cdot \mathbf{a})\mathbf{a} + \mathbf{r}) \cdot \mathbf{r})^{k_e} \tag{5.1}$$

SPAD Saturation

The Phong reflection model alone does not take into account light falloff or unique properties of the SPAD sensor. Previous work [36] has established that, due to the nature of SPADs, the number of detected photons p follows a soft saturation curve in relation to the number of incident photons ϕ, given by $p = 1 - e^{-\phi}$. Due to the inverse-square law, a ray which travels distance r from the sensor before bouncing off the scene returns with an intensity of $1/r^2$. We incorporate the plane's albedo α, as well as the output I from the lighting model given in Eq. (5.1). The asymptotic highest possible photon detection count σ is a property of the sensor. The sensor gain parameter g scales the intensity for an individual ray–this is included because in practice we do not simulate as many rays as photons are actually measured by the sensor. The number of detected photons p is then given by:

$$p = \sigma(1 - e^{-gI/(\sigma r^2)}) \tag{5.2}$$

Histogram Formation

Consider a histogram φ which images a plane given by $\mathbf{ax} + d = 0$, with a uniform albedo and reflectance parameters. Let the sensor reside at the origin, and let R be a set of rays uniformly sampled from the field-of-view which φ

images. If φ has n bins, a bin temporal "size" of t, and a bin offset ω (meaning a flight time of t is recorded as $t + \omega$), the value of an individual histogram bin is given by:

$$i_s = \omega + t(i-1) \qquad i_e = \omega + ti$$

$$\varphi_i^{raw} = \sum_{\mathbf{r} \in R} \begin{cases} p(\mathbf{r}) & \text{if } i_s \leq \dfrac{2||\texttt{isect}(\mathbf{r}, \mathbf{a}, d)||_2}{c} < i_e \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Where $p(\mathbf{r})$ is the intensity of light returned by ray $\mathbf{r}$, as given in Eq. (5.2), c is the speed of light, and $\texttt{isect}(\cdot) \in \mathbb{R}^3$ is the intersection point of $\mathbf{r}$ with $\mathbf{ax} + d = 0$. Because the sensor that we model (TMF8820) filters out ambient light on-sensor, we assume no ambient light in our imaging model.

Laser Impulse

The sensor which we model records the intensity of its laser impulse over time by piping the laser pulse directly to a SPAD [2], and reports the result as a "reference histogram". The captured transient histogram is effectively temporally blurred by a kernel matching the reference histogram. To replicate this effect, we cross-correlate the reference histogram with the generated histogram as a step in our forward process, as shown in Fig. 5.1. In the case of the TMF8820 sensor that we utilize, the temporal scale (bin size) is not the same in the reference histogram δ as in the transient histogram φ, so we temporally scale δ by a factor s_δ before applying the cross-correlation. The histogram after correlation is given by

$$\varphi^{corr} = \varphi^{raw} \star \texttt{rescale}(\delta, s_\delta) \tag{5.4}$$

Where $\star$ denotes the cross-correlation operation, and the `rescale` function scales the function δ temporally by s_δ.

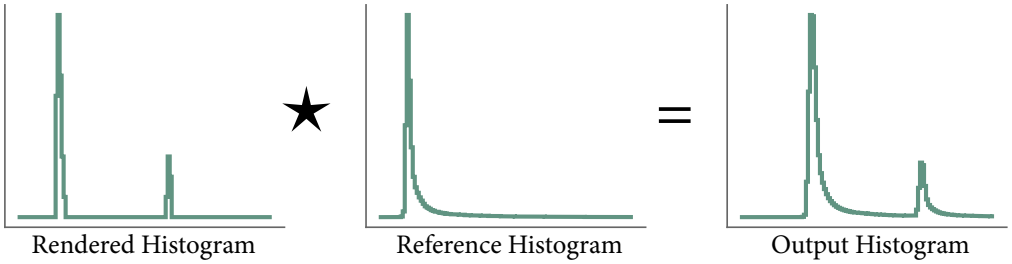


Figure 5.1: **Illustration of Convolution with the Laser Impulse**: The raw rendered histogram is cross-correlated with the reference histogram (which encodes the laser pulse intensity over time) to generate the output histogram of our forward imaging model.

Inter-histogram Interference

The sensor that we model suffers from *inter-histogram interference*, meaning light detected in one histogram is also detected in other histograms, scaled by a factor. We assume that one histogram interferes with all other histograms with an equal magnitude ψ, meaning that a bin value of x in one histogram will manifest as ψx in all other histograms. Formally, for a histogram $\varphi \in \Phi$, where φ_i is the i^{th} bin of φ,

$$\varphi_i = \varphi_i^{corr} + \psi \sum_{\tilde{\varphi}^{corr} \in \Phi^{corr}} \tilde{\varphi}_i^{corr} \tag{5.5}$$

5.5 Differentiable Rendering Pipeline

Our differentiable rendering pipeline recovers plane parameters by minimizing the loss between an observed histogram and the output of a render function. The render function renders a histogram image Φ^r as a function of four sets of variables: the scene geometry G, reflectance model parameters F, the sensor's forward imaging model parameters C, and the sensor's reported laser impulse δ:

$$\Phi^r = R(G, F, C, \delta) \tag{5.6}$$

The render function assumes that the sensor is placed at the origin and the

optical axis is aligned with the positive z axis. The planar geometry G is given by the angle of incidence θ of the optical axis to the plane, the intersection point of the plane with the z axis Z_0, and the azimuth angle ϕ, which denotes rotation about the optical axis.

The lighting parameters F are comprised of the Phong reflection model parameters (k_s, k_e, α). The camera parameters C are comprised of those described in Sec. 5.4 $(n, t, g, \psi, s_\delta, \sigma)$, along with 36 scalar parameters which define the height, width, and center of each of the sensor's 9 FoV regions. These FoV parameters are derived from the TMF8820 specification sheet [2], and are not differentiable in our implementation. Also included in C is an integer which denotes the number of random ray samples used to render the transient histogram. We keep this fixed at 2304 per FoV region. The impulse response function δ is reported by the sensor along with every image.

To compare the rendered histogram Φ^r to the observed histogram Φ^o, the following loss function $\mathcal{L}$ is used:

$$\mathcal{L}(\Phi^r, \Phi^o) = \sum_{(\varphi^r, \varphi^o) \in \Phi^r, \Phi^o} \left|\left| \frac{\varphi^r}{\texttt{max}(\varphi^o)} - \frac{\varphi^o}{\texttt{max}(\varphi^o)} \right|\right|_2 \tag{5.7}$$

Dividing by the magnitude of the observed histogram ensures that high magnitude histograms do not dominate the loss. Unlike previous work [42], we do not use a Fourier transform-based loss function. In our tests, the L2-norm function above performed slightly better. We believe this is because we utilize a good initial estimate from the histogram peak based approach described in Sec. 5.6. A Fourier-based loss excels when the rendered and observed histograms are very different, but may not provide as strong of a signal when they are similar. We adapt Mu et al.'s Python implementation [74] of the algorithm given by Urea et al. [121] to uniformly sample rays from the rectangular FoV of the TMF8820.

The process of assigning a value to a histogram bin is inherently non-differentiable, as there is an instantaneous change in the output histogram as the input crosses a bin boundary. Following previous work [42], we make the render

function differentiable via a soft binning process, in which a Gaussian kernel is generated centered at each input datapoint, and these Gaussians sampled at the bin centers and summed across the datapoint dimension to generate an approximation of the histogram. In our implementation, each Gaussian is also weighted according to its intensity, which is given by Eq. (5.2). The same soft binning process is used to temporally scale the reference histogram δ.

To tune the parameters of our forward imaging model, we utilize a large dataset D of (Φ^o, δ) pairs, each with an associated ground truth geometry G. We minimize the reconstruction loss, as given in Eq. (5.7) over the entire dataset to find the ideal camera parameters C^*:

$$C^* = \arg\min_{C} \sum_{(\Phi^o, G, \delta) \in D} \mathcal{L}(R(G, F, C, \delta), \Phi^o) \tag{5.8}$$

Where the reflectance model parameters F are free variables. This optimization only needs to be performed once, as C^* remains fixed for a given sensor.

To recover the geometric parameters G^* of a planar surface from a single image Φ^o with reference histogram δ, we use the optimized forward imaging parameters C^*, while allowing the scene geometry G and reflectance model parameters F to change:

$$G^*, F^* = \arg\min_{G,F} \quad \mathcal{L}(R(G, F, C^*, \delta), \Phi^o) \tag{5.9}$$

Performing this optimization also recovers the reflectance model parameters F^* of the surface. We evaluate the consistency of the surface albedo recovered by this approach in section Sec. 5.7. Recovery of other reflectance parameters is left for future work as it is outside of the scope of this chapter, and evaluation of these parameters is difficult.

Optimization is performed via stochastic gradient descent using the Adam optimizer [47]. The render function R is implemented in PyTorch with gradients generated through automatic differentiation. To initialize G in Eq. (5.9), we use

the output of the histogram peak based approach described in Sec. 5.6. We observe that regardless of what reasonable starting estimate is used, the optimization tends to converge to the same solution for planar geometry.

5.6 Histogram Peak Based Approach

We provide an empirically calibrated approach which is able to approximate the performance of differentiable rendering on the plane recovery task. This method operates by estimating the distance to the plane in each field of view, projecting outwards by the distance, and fitting a plane to the projected points. To tune the method, we optimize a linear mapping from histogram bin to distance (given by parameters m and b below). We also optimize the angle at which points are projected outwards; a different angle is used depending on whether the histogram corresponds to a field of view region on the edge or corner of the overall 3×3 region field-of-view (s_e or s_c respectively). The algorithm for this approach is shown in Algorithm 1.

To find the location of the peak in a histogram φ, we fit a piecewise cubic curve to the 128-bin histogram, and sample that curve at $10\times$ density around the highest individual bin. The temporal position of the highest point on the interpolated curve is the location of the peak. This process captures variations smaller than the ~ 1.2cm equivalent bins of the histogram by using the relative intensity between adjacent bins. We find empirically that this approach outperforms picking the highest bin without interpolation.

To determine the optimal parameters m, b, s_e, and s_c to the RecoverPlane function, we find the parameters which minimize the error in the reconstructed plane over some calibration dataset D which contains images Φ along with ground truth planar geometry $\mathbf{a}, d$:

$$m^*, b^*, s_e^*, s_c^* = \underset{m,b,s_e,s_c}{\arg\min} \sum_{\Phi,\mathbf{a},d \in D} \epsilon_p(f(\Phi, m, b, s_e, s_c), \mathbf{a}, d) \tag{5.10}$$

Algorithm 1 Empirically calibrated algorithm for recovering planar geometry from a set of transient histograms using histogram peaks

function RECOVERPLANE(Φ, m, b, s_e, s_c)
  $pts \leftarrow [\,]$
  for φ in Φ **do**
    $i \leftarrow$ the temporal coordinate of the peak of φ
    $dist \leftarrow i * m + b$
    $\mathbf{u} \leftarrow$ unit vector pointing to center of FoV of φ
    if φ images an edge FoV region **then**
      Scale angle of $\mathbf{u}$ from optical axis by s_e
    else if φ images a corner FoV region **then**
      Scale angle of $\mathbf{u}$ from optical axis by s_c
    end if
    $pt \leftarrow \mathbf{u} * dist$
    Append pt to pts
  end for
  Fit a plane to pts via SVD [12]
  return the parameters $\mathbf{a}, d$ of the fit plane
end function

where the ϵ_p is the point error between two planes, as defined in Eq. (5.11), and f is the RecoverPlane function given in Algorithm 1. We perform this optimization using the Nelder-Mead method [78] with finite difference estimation of derivatives, via the SciPy Python library. As this optimization is performed only once, speed is not crucial.

5.7 Experimental Results

Sensor Configuration

We run the TMF8820 in "short range, high accuracy" mode, in which it reports 128 bins with an individual bin size equivalent to ~ 1.2cm of distance. We run the sensor in the default configuration of 4 million iterations (light pulses) per measurement, and use the default field-of-view configuration, which gives an

FoV of 33×34, divided into 3×3 regions, with a transient histogram reported for each region.

Metrics

We use three metrics to measure the accuracy of plane recovery. Assume that we are comparing two planes given by $\mathbf{a}_1\mathbf{x} + d_1 = 0$ and $\mathbf{a}_2\mathbf{x} + d_2 = 0$, where $d_1 > 0$, $d_2 > 0$, then the *angular error* $\epsilon_a = \arccos(\mathbf{a}_1 \cdot \mathbf{a}_2)$. *Linear error* is given by $\epsilon_l = |d_1 - d_2|$. These metrics are intuitive, but the trade-off between the two is not clear. To capture error with a single metric, we define point error ϵ_p. Given a random ray originating at the sensor and within the sensor's FoV, point error captures the expected difference between the intersection of that ray with the predicted plane and with the ground truth plane. Formally:

$$\epsilon_p = \frac{\sum_{\mathbf{r} \in R} ||\texttt{isect}(\mathbf{a}_1, d_1, \mathbf{r}) - \texttt{isect}(\mathbf{a}_2, d_2, \mathbf{r})||_2}{|R|} \tag{5.11}$$

where $\texttt{isect}(\mathbf{a}, d, \mathbf{r})$ returns the 3D point of intersection between plane $\mathbf{a}\mathbf{x}+d = 0$ and ray $\mathbf{r}$, and R is a randomly sampled set of rays originating at the sensor and within the sensor's FoV. In practice, we set R to be an 8×8 grid of rays which uniformly cover the sensor's FoV for repeatability.

Planar Recovery

We evaluate five different approaches for planar recovery:

1. Differentiable rendering, the optimization problem in Eq. (5.9) is solved.
2. Peak finding - calibrated, the histogram peak based approach given by Algorithm 1 is performed with optimized parameters given by Eq. (5.10).
3. Proprietary distances - calibrated, the same as 2), but utilizing distance estimates generated onboard the sensor rather than histogram peak locations.

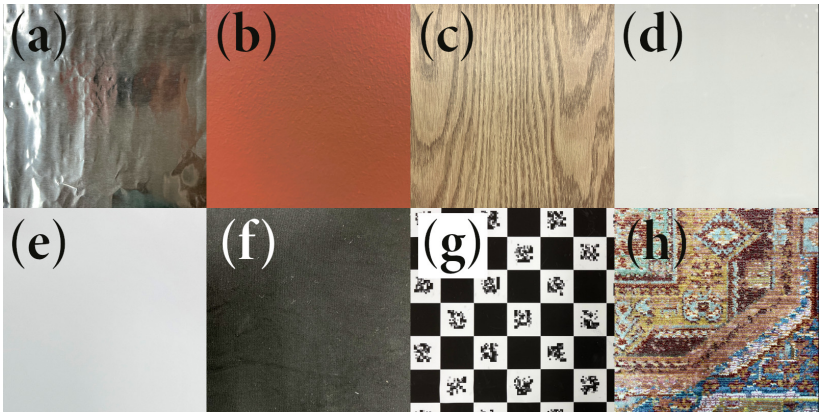


Figure 5.2: **Materials Used for Evaluation of Planar Recovery**: (a) aluminum foil; (b) red painted drywall; (c) wooden table; (d) whiteboard; (e) white paper; (f) black fabric; (g) checkerboard; (h) patterned rug.

4. Peak finding - naive, the histogram peak based approach is used, but without optimized parameters.

5. Proprietary distances - naive, the same as 4), but utilizing distance estimates generated onboard the sensor rather than histogram peaks.

To generate ground truth data, we mount a TMF8820 sensor to a custom 3D printed end effector for a Universal Robots UR5 robot arm. We manually calibrate the position of the sensor relative to the end effector, and use the robot's forward kinematics (which are quoted as precise to $\pm$ 0.1mm) to gather ground truth sensor poses. To determine the position of the plane relative to the robot, the end effector is touched to the plane at a number of points, and a ground truth plane is fit to these points. The robot is used to automatically move the sensor, allowing us to generate a large dataset of planar images (3,800 images total) from a variety of distances (Z_0), angles-of-incidence (θ), and azimuth angles (ϕ). All measurements were captured in an artificially lit room.

To ensure the validity of our results when comparing differentiable rendering to other approaches, we use the worst-performing naive proprietary distances approach as a starting estimate, and perform 100 iterations of gradient descent.

	Angular Error ()			Linear Error (mm)			Point Error (mm)		
Method	Mean	Median	95%	Mean	Median	95%	Mean	Median	95%
Differentiable Rendering*	**3.40**	**1.97**	**12.90**	**2.46**	**1.90**	**6.51**	**3.79**	**3.17**	8.46
Peak Finding - Calibrated†	3.57	2.22	13.44	2.67	2.11	7.13	3.94	3.52	**7.92**
Peak Finding - Naive	5.68	3.87	18.42	6.15	5.28	13.56	7.70	6.96	14.28
Proprietary Distances - Calibrated†	7.34	4.71	25.97	49.20	60.31	68.41	52.44	62.96	69.60
Proprietary Distances - Naive	8.87	4.71	30.06	60.31	71.96	78.14	65.45	76.26	83.15

Table 5.1: **Comparison Between Our Method and Methods Using Proprietary Distance Estimates**: Methods which utilize the histogram outperform those which use proprietary distance estimates in all metrics. Images in range 1-30cm to plane, 0-30AoI on surfaces (c) - (h). 400 measurements per surface. Measurements of surface (b) from the same range were used optimize forward model of differentiable method (*) and calibrate "calibrated" methods (†). 95% refers to the 95th percentile of error. See Sec. 5.7 for a description of methods.

	Point Error (mm)		
Method	Mean	Median	95%
Differentiable Rendering*	**6.26**	**3.52**	**22.31**
Peak Finding - Calibrated†	6.80	3.78	23.58
Peak Finding - Naive	15.84	11.45	44.56
Proprietary Distances - Naive	42.23	22.15	130.46
Proprietary Distances - Calibrated†	74.45	75.45	143.51

Table 5.2: **Comparison Between Our Method and Methods Using Proprietary Distance Estimates Over a Wide Range**: Methods which utilize the histogram outperform those which use proprietary distance estimates in larger range of plane parameters. Measurements cover range 1-70cm, 0-45AoI of surface (b). Measurements of surface (e) from range 0-30cm, 0-30 AoI were used optimize forward model of renderer (*) and to calibrate "calibrated" methods (†).

One iteration takes about 0.05 seconds on a mid-range laptop (i7-10705H, NVIDIA GTX 1650Ti). In real-world operation, a better starting estimate could be used and fewer iterations performed. The peak-based approaches operate at 95Hz on the same hardware, exceeding the 30Hz at which the sensor reports data.

A comparison between the five approaches is given in Tab. 5.1. Methods which utilize transient histograms consistently outperform those which rely on

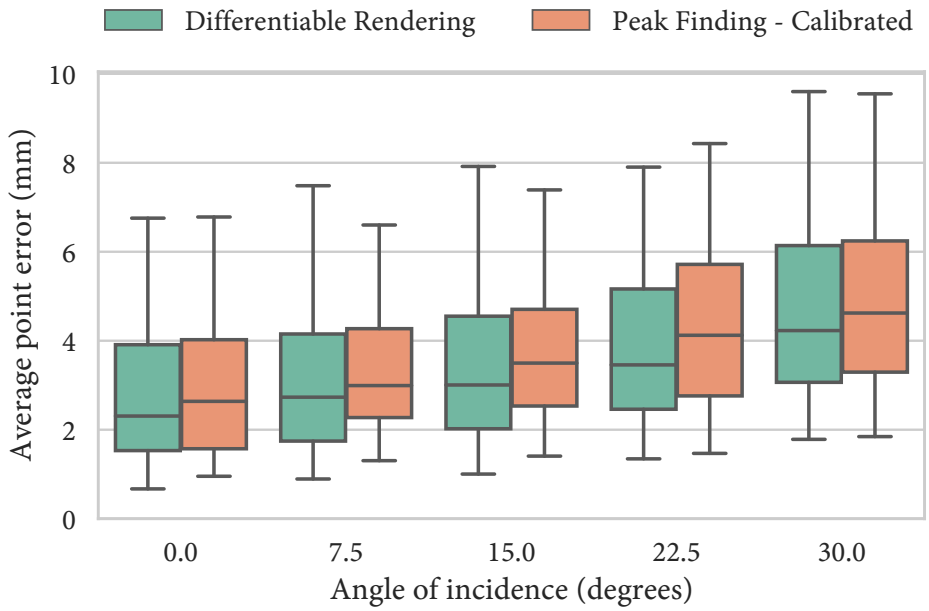


Figure 5.3: **Higher Angle-of-Incidence Leads to Higher Error in Reconstruction**: Measurements of materials (c)-(h) cover distance range 0-30cm. Whiskers extend to 5th and 95th percentile.

proprietary distance estimates. We see that the differentiable rendering approach, which utilizes the entirety of the information in all nine histograms, outperforms peak finding approaches, in which each histogram is reduced to a single value. Our peak finding approach comes close to the performance of differentiable rendering across the board, even outperforming it in some cases, offering speed at the expense of generality. We believe the large gap between the "peak finding" and "proprietary distances" approaches can partially be explained by a difference in interpolation method; the interpolation method used onboard the sensor may be less accurate than the one used in our peak finding method. However, in our testing we found that even when using *no interpolation at all*, our peak finding approach outperformed the proprietary distances approaches, necessitating an additional explanation.

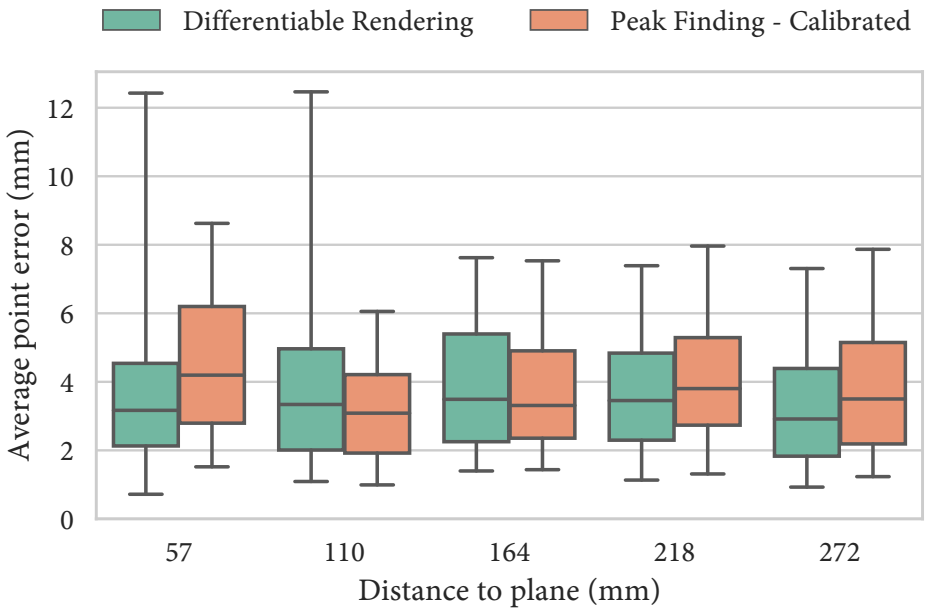


Figure 5.4: **Distance to the Planar Surface has Little Effect on Reconstruction Error**: Measurements of materials (c)-(h) cover AoI range 0-30. Whiskers extend to 5th and 95th percentile. Ticks on x axis denote center of 54mm bins.

We suspect that the proprietary algorithm onboard the sensor is not overly naive, but instead is designed to be more general purpose than our approach. Plane fitting is a special case; an algorithm which performs well for a variety of potential use cases may not be optimal for plane fitting. The peak finding method that we use was chosen *because* it is effective at recovering planar surfaces. By accessing the transient histograms directly, we were afforded the ability to make this choice. Results of planar recovery over a wider range of sensor poses are shown in Tab. 5.2. The effect of angle-of-incidence (AoI) and distance on reconstruction error is shown in Fig. 5.3 and Fig. 5.4.

We evaluate our method on a range of surfaces and report the results in Tab. 5.3. The parameters of the "calibrated" methods and imaging model parameters of the differentiable rendering methods were trained on measurements of the red

	Mean Point Error (mm)		
Material	Diff. Render	Peak Find	Propr. Dist.
(b) Red drywall*	**2.05**	3.09	4.68
(e) White paper	**2.51**	3.45	63.0
(f) Black fabric	**2.51**	3.35	72.5
(h) Patterned rug	**2.69**	3.62	62.7
(c) Wood	4.19	**4.03**	60.7
(d) Whiteboard	**5.12**	5.82	54.9
(g) Checkerboard	6.94	**4.12**	61.1
(a) Aluminum foil	**12.7**	15.0	25.3

Table 5.3: **Performance of Planar Recovery by Surface**: Our methods are generally robust to surface properties, aside from highly specular aluminum foil. Images in range 1-30cm, 0-30 AoI. *Measurements of red drywall are used to optimize forward model of differentiable method and to calibrate peak finding and proprietary distance approaches.

painted drywall. We see that our methods are generally robust to this change from training to testing surface, particularly when that surface has a uniform texture and albedo. Our methods are slightly less robust to textured surfaces such as wood and a patterned rug. We see diminished performance with the slightly glossy whiteboard, and the checkerboard surface, which has spatially varying albedo. Performance is significantly diminished on the specular aluminum foil.

We observe that the proprietary distance based approach tends to overfit when calibrated to a dataset. There is evidence of this overfitting in the longer range test in Tab. 5.2; the calibrated histogram approach improves over the naive approach, while the calibrated proprietary distance approach performs *worse* than the naive. This is because the parameters of the "calibrated" approaches were calibrated to recover planar geometry on images of a different surface over a different range of distances and angles of incidence. While the histogram based approaches, including differentiable rendering, are robust to this change in surface, the approaches which utilize proprietary distances are not.

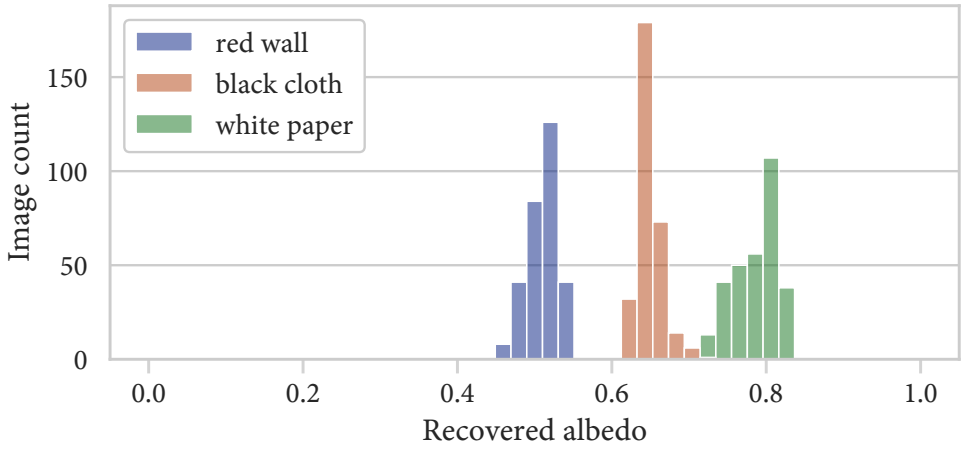


Figure 5.5: **Albedo Recovery Results**: Our method recovers consistent surface albedo under various distances and angles-of-incidence. Recovered albedo is in the wavelength of the sensor light source (940nm IR), and may vary significantly from the albedo as it appears to the human eye under visible light. Each surface is observed from 300 poses in range 7-40cm, 0-30 AoI.

Albedo Recovery

We evaluate the performance of our differentiable renderer for recovering surface albedo, as given in Eq. (5.9) by recovering the albedo from images of three planar surfaces which have a uniform texture and albedo. We only evaluate the *consistency* of the recovered albedo, not the *accuracy*. Evaluating the accuracy would require an accurate characterization of the wavelength of the sensor light source, and surfaces with a known albedo in that wavelength, which is outside the scope of this work. We find that this method recovers a consistent albedo per-surface relatively invariant to distance in the range 7-40cm and angle-of-incidence in the range 0-30, allowing discrimination between surfaces, as shown in Fig. 5.5.

5.8 Example Application

We build an application to showcase our methods, in which a robot arm is holding a cup of liquid. The robot's goal is to safely place the cup on a tabletop below, which is at an unknown distance and may have regions which are not level. In our application, we attach a TMF8820 transient sensor directly to the gripper of the robot arm. Due to its small size, the sensor can be placed centimeters away from the jaws of the gripper, where it senses the surface below directly, making it invulnerable to occlusions.

Using our approach for recovering planar geometry, the robot is able to sense the distance to and slope of the surface below the cup being held in the end effector, as shown in Fig. 5.6. The robot uses this information to know when it is close enough to the surface to place the cup down, and to ensure that the surface is level enough to safely release the cup. See youtu.be/vJdfpmd6OE0?t=258 for a video demonstration.

5.9 Limitations

While the differentiable rendering method given in this work can in principle recover any unknown parameters to the render function, we only evaluate recovery of scene geometry and albedo. A next step is to investigate recovery of the reflectance model parameters of a surface. While our method in principle enables such recovery, evaluation is difficult. Another next step is recovering other types of geometry. Our approach can in principle easily be adapted to other parameterized surfaces, *e.g.*, a sphere or cube. Extending to arbitrary geometry would require a more general differentiable representation, *e.g.* a neural representation akin to NeRF [72]. As both of these tasks introduce extra degrees of freedom into the optimization process, they may require a more accurate and/or sophisticated model of the transient histogram imaging process to sufficiently constrain optimization.

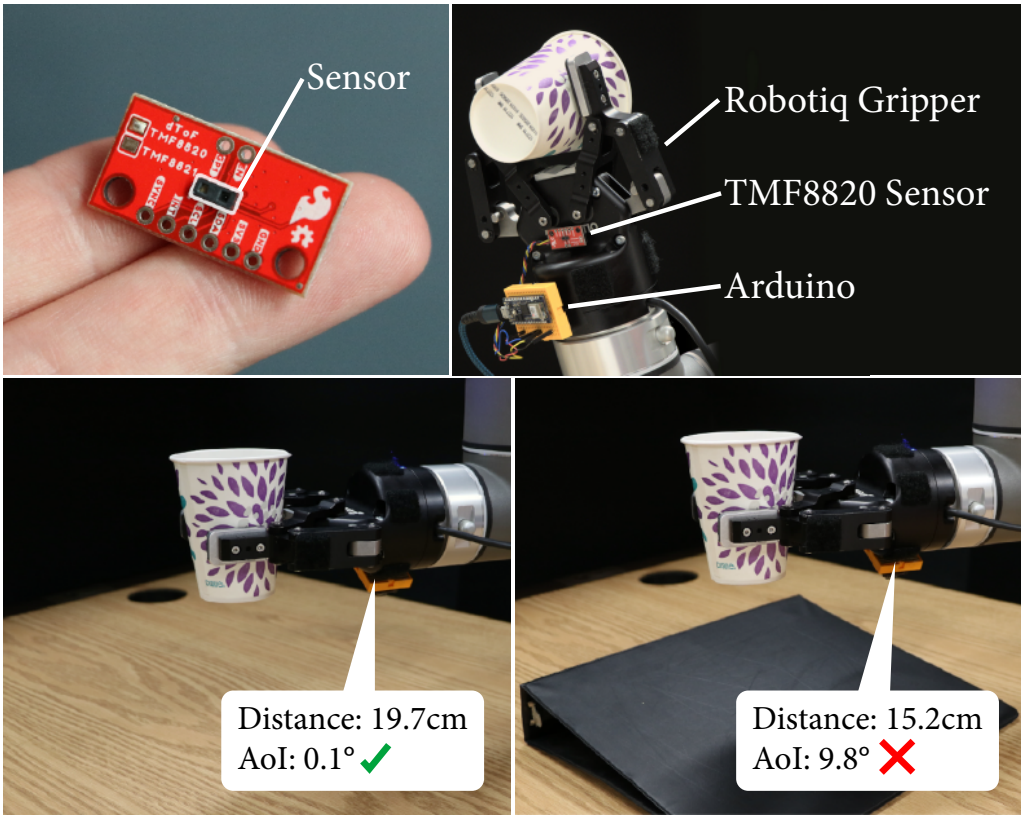


Figure 5.6: **Plane Recovery Demo**: We mount a proximity sensor on a robot gripper (top). The sensor detects when the surface below the gripper is level and safe to place a cup full of liquid (bottom left) or is not level and therefore unsafe (bottom right).

One challenge for future work is the low bandwidth available on commodity sensors. In our test setup, histograms are read from the sensor at 4.5 frames per second (where one "frame" consists of nine histograms) despite the sensor generating proximity estimates at 150Hz. This is not a limitation of the sensor technology, but of the I^2C interface over which it transmits data. We hope that commodity SPAD sensors will in the future come packaged with high bandwidth interfaces to enable granular and high-speed sensing. Algorithms will also need to be optimized to perform inference quickly enough to keep pace with higher bandwidth sensors.

Lastly, we provide only a basic demonstration of utilizing transient histograms in a robotics setting. It is yet to be shown that utilizing these histograms leads to improvement in performance of downstream robotics tasks. An important next step is to build a complete robotics system which utilizes transient histogram data, and evaluate the system performance compared to alternative sensing modalities. We are hopeful that future robotics systems will harness the power of transient histograms to be highly aware of their environment on a low size, weight, and power budget.

6 RECONSTRUCTING PARAMETRIC 3D SCENES

In this chapter, we aim to recover the geometry of 3D parametric scenes using very few depth measurements from low-cost, commercially available time-of-flight sensors. The time-of-flight data captured by these sensors 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.

Project website: cpsiff.github.io/recovering_parametric_scenes/

This work was completed under the supervision of Yin Li, Mohit Gupta, and Michael Gleicher. This is joint work with Yiquan Li, with additional help from Yiming Li and Fangzhou Mu. Yiquan led feedforward network training and architecture, and ran many experiments. Yiming led the development of the differentiable mesh renderer. Fangzhou contributed to the development of the simulator used to generate training data. Carter led data capture, overall ideation and experiment design, sensor characterization, baseline methods, writing, figures, and presentation.

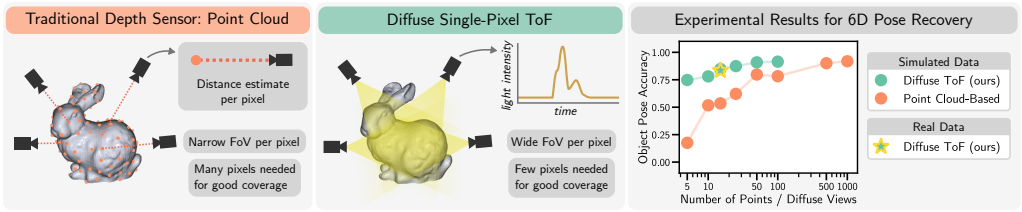


Figure 6.1: **Overview of Our Method for Recovering Parametric Scenes**: We introduce a method for recovering the geometry of parametric 3D scenes, such as the 6D pose of a known object, from a distributed set of **very few** (*e.g.*, 15), diffuse (*i.e.*, wide field-of-view) single-pixel ToF Sensors. 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.

6.1 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 [109, 2]. These sensors, already deployed in mobile [114] and wearable applications [82], are often implemented with a single photon avalanche diode (SPAD) array [79, 73], 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 [75, 7, 58, 42], 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 conditions, *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. 6.1), *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 practical for widespread use. Moreover, we rely on currently available consumer hardware, which has a restricted sensing range, limiting our experiments to tabletop scenes.

6.2 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) [73]. This setup has been successfully applied to fluorescence lifetime imaging [102], novel view synthesis [67], and

non-line-of-sight (NLOS) imaging [128, 39, 49, 48, 29]. 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 [16, 131], shape reconstruction [75], human pose estimation [101], and SLAM [61]. 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*. [107] introduce a simple model for miniature ToF sensors. Recent works model the SPAD image formation process with differentiable functions for laboratory-grade [68, 65] or commodity [75] sensors, enabling gradient-based optimization and facilitating 3D tasks such as pose estimation and shape reconstruction. Our work modifies the sensor model in [75] to accommodate differentiable mesh rendering.

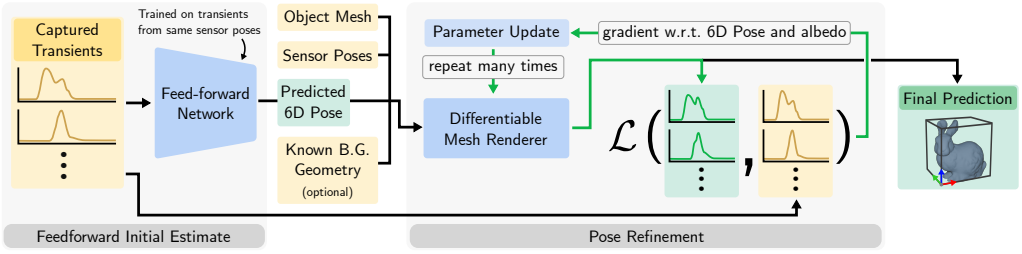


Figure 6.2: **Overview of our method** as applied to 6D pose estimation. Our method consists of two components: 1) a pose prediction module, where a feedforward network estimates initial object pose from a sparse set of input transient histograms; and 2) a pose refiner, where a differentiable renderer 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.

3D Vision with Low-Cost SPADs. Prior works have explored feedforward neural networks for inference from transient histograms. Pixels2Pose [101] proposes a learning-based method to estimate whole-body human pose from a single 4×4

pixel transient histogram. DELTAR [58] and Jungerman *et al*. [42] 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*. [107] design an AbS pipeline to recover 3DoF plane pose and albedo from a set of 3×3 transient measurements. Mu *et al*. [75] present a method to recover arbitrary 3D scenes from a distributed set of $>$100 single-pixel transient measurements. Behari *et al*. [7] leverage AbS to reconstruct arbitrary 3D scenes from a miniature ToF sensor plus an RGB camera. Liu *et al*. [61] build a neural radiance field for dense SLAM by integrating data from a miniature ToF sensor with an RGB camera. Luo *et al*. [65] 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 [125, 127, 84, 53, 15]. The predicted pose can be further refined via an AbS pipeline [125, 116, 53, 50]. 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.

6.3 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 $\boldsymbol{h}$ 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 chapter. *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 [97, 57, 62, 129]. 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 estimate $\mathbf{P}$ from the set of measured histograms $\{\boldsymbol{h}_i\}_{i=1}^n$.

Method Overview. Our method consists of two steps: 1) given a set of input transient histograms $\{\boldsymbol{h}_i\}_{i=1}^n$, 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 $\{\boldsymbol{h}_i\}_{i=1}^n$ 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. 6.2. In what follows we introduce the SPAD image formation process and present each component of our method.

Background: Transient Formation Model

We utilize physics-based sensor modeling to accurately render the transient histograms $\{\boldsymbol{h}_i\}_{i=1}^n$. For each captured histogram, the laser source emits N_{emit} photons. Assuming that the source is co-located with the sensor at the origin

$\boldsymbol{o}$, the rays of the emitted photons can be parametrized by a direction $\boldsymbol{\omega}$. As in [75, 42, 107], we ignore high-order light paths and consider one-bounce paths only. Therefore, each photon travels from $\boldsymbol{o}$ to a point on the scene $\boldsymbol{x}$ and then back to $\boldsymbol{o}$, where $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\omega}, \mathcal{M})$ is the first intersection between the ray in the direction of $\boldsymbol{\omega}$ and the scene $\mathcal{M}$.

Following prior work [75, 42], the expected number of photons $N[i]$ received by the sensor within the i-th bin, in its angular integral form, is

$$N[i] = N_{\text{emit}} \int_{\Omega} I(\boldsymbol{\omega}) \frac{\rho(\boldsymbol{x})}{\pi} \frac{\left\langle -\boldsymbol{\omega}, \hat{\boldsymbol{n}}(\boldsymbol{x}) \right\rangle}{\|\boldsymbol{x}\|^2} W(\frac{2\|\boldsymbol{x}\|}{c}, t_i) \, \mathrm{d}\boldsymbol{\omega} \tag{6.1}$$

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

$$W(t, t_i) = \begin{cases} 1 & \text{if } \; t \in [t_i, t_i + \Delta t), \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

bins the photons. Due to hardware effects, the binning of photons is not perfect in reality. $\boldsymbol{h}[i]$ 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 $\boldsymbol{s}$ to account for this effect

$$\boldsymbol{h}[i] = \Sigma_j N[i + \Delta i - j] \boldsymbol{s}[j], \tag{6.3}$$

where Δi is sensor-specific, constant temporal offset to the histogram, which needs to be calibrated.

Pile-up Correction. Previous works [36, 75] 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 [20]. 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.

Differentiable Rendering

We numerically integrate Eq. (6.1) using the weighted sum

$$N[i] \approx \frac{N_{\text{emit}}}{\pi} \sum_{\boldsymbol{\omega} \in \mathcal{W}} Q(\boldsymbol{\omega}) I(\boldsymbol{\omega}) \rho(\boldsymbol{x}) \frac{\left\langle -\boldsymbol{\omega}, \hat{\boldsymbol{n}}(\boldsymbol{x}) \right\rangle}{\|\boldsymbol{x}\|^2} W(\frac{2\|\boldsymbol{x}\|}{c}, t_i) \tag{6.4}$$

where $\mathcal{W}$ is a specified set of rays, and $Q(\boldsymbol{\omega})$ is the associated quadrature. To make the rendering differentiable, we calculate $\partial \boldsymbol{x} / \partial \mathcal{M}$ and $\partial \hat{\boldsymbol{n}} / \partial \mathcal{M}$ 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' $\boldsymbol{x}, \hat{\boldsymbol{n}}$, 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 $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$, the quadrature $Q(\boldsymbol{\omega})$ transforms the square pixel areas to solid angle differentials by

$$Q(\boldsymbol{\omega}) = \frac{A_{\text{pixel}} \langle \omega, \hat{z} \rangle}{(1/\omega_z)^2} = \frac{4\tan^2(FoV/2)\omega_z^3}{h \cdot w}. \tag{6.5}$$

The binning function W in Eq. (6.2) is discontinuous. We thus approximate it with the sigmoid function $\sigma(x)$ by

$$W(t, t_i) = \sigma\left(k(t - t_i)\right) - \sigma\left(k(t - t_i - \Delta t)\right), \tag{6.6}$$

where k is a hand-picked constant to balance smoothness and realism. Further, the intensity map $I(\boldsymbol{\omega})$ 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 $I(\boldsymbol{\omega})$ using a differentiable, spatially-varying function. We fit this function using real-world sensor properties in our experiments (see Sec. 6.4).

Feedforward Estimation of Scene Parameters

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

$$f_\theta(\{\boldsymbol{h}_i\}_{i=1}^n) \rightarrow \mathbf{P}_{\text{FF}}, \tag{6.7}$$

where θ is the network weights learned from data, $\{\boldsymbol{h}_i\}_{i=1}^n$ 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_θ is a standard Transformer model [122]. 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 (Sec. 6.7).

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. 6.3 and availability of 3D models [17]. 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 [56].

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 $\{\boldsymbol{h}_i\}_{i=1}^n$ and rendered histograms $\mathcal{R}(\mathbf{P})$, given by

$$\arg\min_{\mathbf{P}} \ \Sigma_{i=1}^{n} \|\mathcal{R}(\mathbf{P})_i - \boldsymbol{h}_i\|. \tag{6.8}$$

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 gradients steps until convergence. Since the rendering process $\mathcal{R}$ is highly nonlinear, the quality of the solution depends on the accuracy of initial estimate $\mathbf{P}_{\text{FF}}$.

6.4 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 transform 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.

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 ± 0.1mm [120], to record sensor pose.

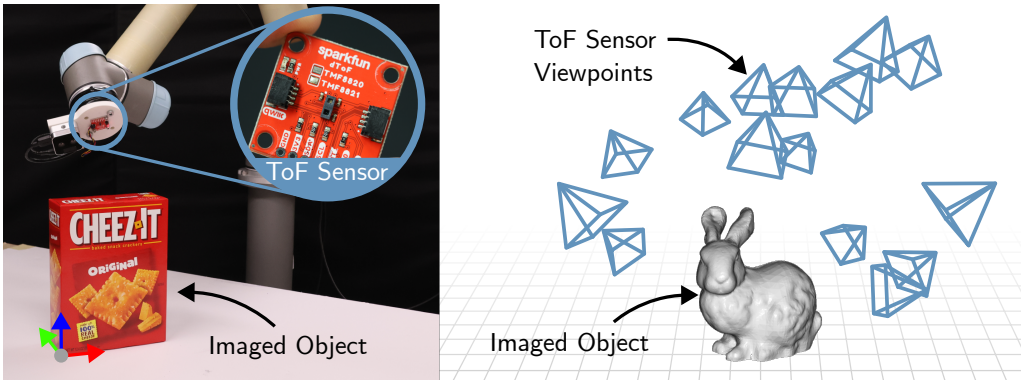


Figure 6.3: **Overview of 6D Pose Capture Setup**: **Left**: Illustration of our capture setup, where a ToF sensor is mounted on a robot arm and moved between a set of positions. **Right**: 15 sensor poses used for our experiments.

We use the AMS TMF8820 SPAD-based ToF sensor. Like other sensors of its class, the sensor is very small (12.8mm^3, $< 1\text{g}$) and low-power ($< 100\text{mW}$) [2]. 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 ($\pm 15°$). 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. 6.4). 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 [8] 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. 6.3.

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.

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 [107, 7]. 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 [75, 106].

Further, in the TMF8820 datasheet, the laser illumination is reported as non-uniform across the FoV [2] $I(\boldsymbol{\omega})$. We therefore approximate the intensity map using

$$I(\boldsymbol{\omega}) = K_1 \exp\left(-K_2\left(\omega_x^2 + \omega_y^2\right) - K_3\left(\omega_x^4 + \omega_y^4\right)\right). \tag{6.9}$$

We calibrate the constants K to match the intensity map shown in the TMF8820 datasheet. A visualization of $I(\boldsymbol{\omega})$ is included in (Fig. 6.10). 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 $\boldsymbol{s}$ 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 [107, 75]. We resample the reference histogram by a factor of $\boldsymbol{s}_{\text{scale}}$ before use in our imaging model. Additionally, the temporal resolution Δt and a constant temporal offset to the histogram Δi are not known. Following prior work, we recover the parameters $\boldsymbol{s}_{\text{scale}}$, Δt, and Δi 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. (6.8).

Feedforward Network. We train the network described in Sec. 6.3 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 [127]. See the Sec. 6.7 for details of the loss functions and training parameters. A single-instance forward pass takes ~ 4.6ms on Nvidia RTX 4080.

Pose Refinement. We use adaptive gradient descent following Adam [47] to solve Eq. (6.8). 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 (ρ_{obj}) and the planar surface (ρ_{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 [135]. Differentiable rendering is implemented via Nvdiffrast [54] and PyTorch [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 [17] — a standard benchmark for 6D pose recognition. See Fig. 6.15 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 [115]; 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.

Data	Total Pixels	Method	AUC-ADD-S (↑) Crackers	Mustard	Chips	SPAM	Basketball	Tennis Ball	Softball	Mean
Sim.	15	1 Px Point Cloud†	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	**90.04**	95.15	**96.26**	**94.95**	91.83
Real	15	Ours: FF + Refiner	**90.04**	**90.07**	**88.50**	90.00	**95.76**	96.06	**94.95**	**92.20**
Sim.	3840	16^2 Point Cloud†	95.17	97.23	97.23	97.19	97.67	97.57	97.37	97.06
Real	407K	Single-View RGB [138]	60.71	87.93	40.12	58.95	65.46	77.68	72.42	66.18
Real	407K	Single-View RGB-D* [125]	90.49	92.10	92.54	93.80	94.24	86.67	94.24	92.01

Table 6.1: **6D Pose Estimation of Symmetric Objects from the YCB Object Set**. gray: methods using additional pixels; †methods using oracle ground-truth pose; *methods using simulated high-resolution point cloud data. See details in Sec. 6.4.

Evaluation Metrics. We follow standard metrics [127] 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

Data	Total Pixels	Method	AUC-ADD (↑) Two	P	L	Bunny	Armadillo	Mean
Sim.	15	1 Px Point Cloud†	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	**83.47**	**84.94**	**77.75**	**77.68**	**79.71**	**80.71**
Sim.	3840	16^2 Point Cloud†	97.55	96.18	95.17	96.18	96.90	96.39
Real	407K	Single-View RGB [138]	56.99	65.40	51.51	64.63	86.28	64.96
Real	407K	Single-View RGB-D* [125]	86.57	85.51	82.72	88.30	87.58	86.14

Table 6.2: **6D Pose Estimation of (Non-Symmetric) 3D Printed Objects**. gray: methods using additional pixels; † methods using oracle ground-truth pose; * methods using simulated high-resolution point cloud data. See details in Sec. 6.4.

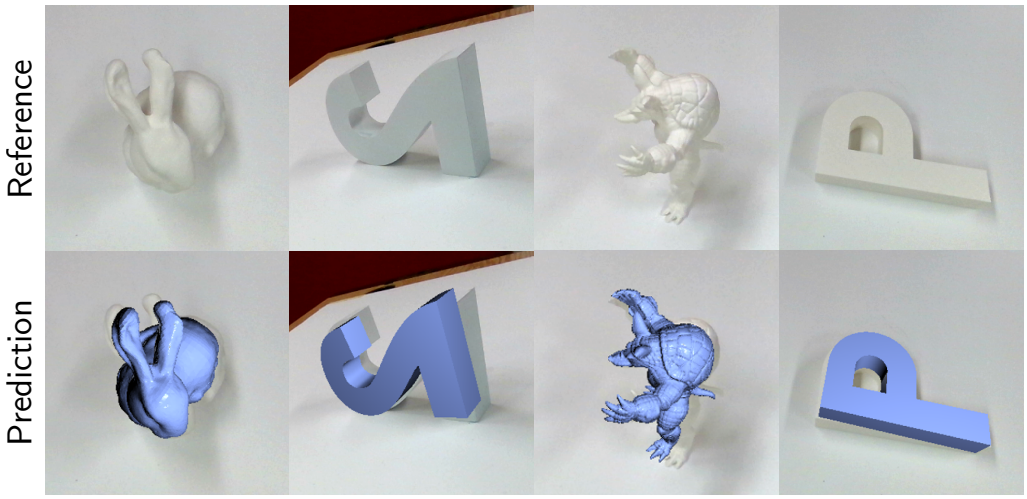


Figure 6.4: **6D Pose Recovery of 3D Printed Objects**: Using our method (feedforward + refiner). For each object, the pose prediction with the median pose error (ADD) over the 25-capture dataset is shown.

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 measure-

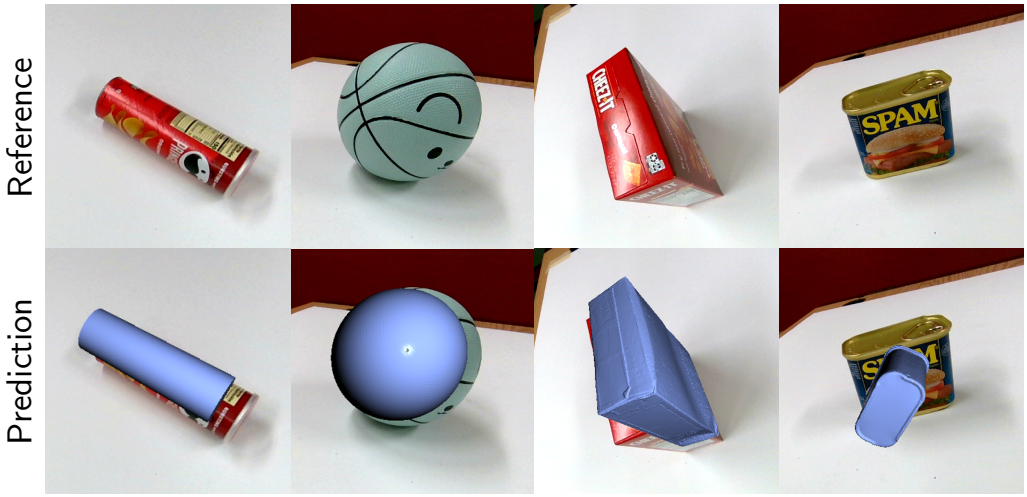


Figure 6.5: **6D Pose Recovery of Objects from the YCB Object Set**: Using our method (feedforward + refiner). For each object, the pose prediction with the median pose error (ADD-S) over the 25-capture dataset is shown.

ments; 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 [8] 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 16×16 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 [138], which uses single-view RGB input, and FoundationPose [125], 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.

6D Pose Estimation Results

Main Results. Our main results are presented in Tab. 6.1 and Tab. 6.2. Visualization of sample results with median pose errors are shown in Fig. 6.4 and Fig. 6.5. 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. 6.5) 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. 6.4. We evaluate on synthetic data of the 25 real "2" poses. Camera poses are sampled via the same process as real poses (Sec. 6.4). Results are shown in the rightmost panel of Fig. 6.1. See Tab. 6.8 for full results. 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.

6.5 Exploration Beyond 6D Pose Estimation

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. 6.4), 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. 6.3. We find that our method can recover the position and diameter of the sphere with an average error of < 1cm in all cases, with often < 0.5cm of error, despite the temporal resolution of an individual SPAD being ~ 1.4cm.

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. 6.6. 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 [97].

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 6.3: **Recovering Position and Size of Spherical Objects.**

Data Capture. As an initial feasibility study, we gather 250 measurements of a single individual's hand from the ring of sensors. To sample hand poses, we

Method	PA-MPJPE (mm) (↓)
ToF-based Prior Work† [25]	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† [85]	6.0

Table 6.4: **Hand Pose and Shape Estimation**: †Related works are provided for context only; metrics are over a different dataset and should not be directly compared.

prompt the user to match their hand pose to a random hand pose selected from the DART dataset [33]. RGB-D cameras are mounted above and below the hand to capture ground truth, which is provided by the RGB-based method HaMeR [85], and aligned to a fused point cloud from the two depth maps via ICP [8]. We reserve 50 captures for testing.

Results. We report Procrutes aligned mean per joint position error (PA-MPJPE), a standard metric for hand tracking [25, 85]. 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. 6.4. 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

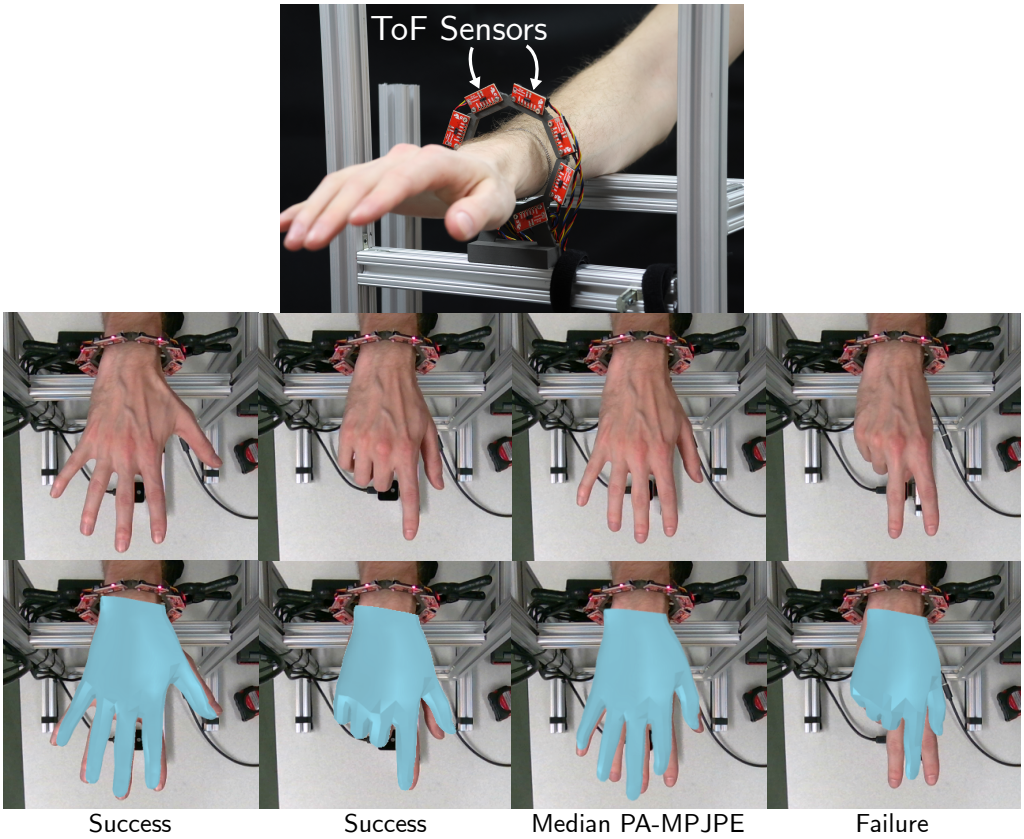


Figure 6.6: **Visualization of Hand Pose Recovery Results**: **Top**: Setting for hand pose capture, in which eight ToF sensors encircle the wrist. **Bottom**: Visualization of results of applying our method to hand pose estimation, corresponding to method "Pretrained on Sim., F.T. on Real" in Tab. 6.4.

comparisons to related works in Tab. 6.4. However, results from prior works are based on different datasets and experimental conditions, and thus are not directly comparable.

6.6 Discussion

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

6.7 Supplementary Materials

In this supplementary material, we provide (1) a description of loss functions for training our feedforward model (Sec. 6.7); (2) additional results on 6D pose estimation (Sec. 6.7); (3) an analysis of runtime and complexity (Sec. 6.7); (4) experiments and discussion on sensor interference (Sec. 6.7); and (5) additional visualization of our results on 6D pose estimation (Sec. 6.7).

Training Loss of Feedforward Models

6D Pose Estimation

As described in Sec. 6.4, 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 $\mathbf{R}_{\text{gt}}$ (represented by the 6D representation proposed by [135]) and translation $\mathbf{t}_{\text{gt}}$. 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

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

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 [127], where $\mathcal{X}$

represents the set of object points:

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

Spherical Object Recovery

For spherical object recovery (Sec. 6.5), 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:

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

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

Human Hand Pose Estimation

For hand pose estimation (Sec. 6.5), we predict the MANO model [97] shape parameters $\boldsymbol{\beta}$, pose parameters $\boldsymbol{\theta}$, global 3D rotation $\mathbf{R}$ (represented by the 6D representation proposed by [135]), and global 3D translation $\mathbf{t}$. The loss for a given prediction is given by:

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

where the loss terms are given by

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

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.

Additional 6D Pose Estimation Experiments

Data Visualization

We visualize the transient histograms captured by multiple, distributed ToF sensors across two different 3D scenes in Fig. 6.7. The measurement has a complex relationship with scene geometry. We aim to solve the inverse problem (multi-view transient histogram → geometry) for simple parametric scenes.

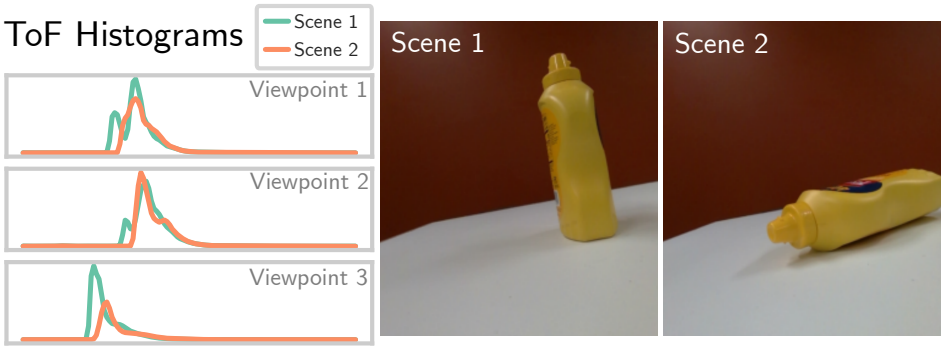


Figure 6.7: **Transient histograms from multiple viewpoints alongside corresponding 3D scenes.**

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 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. 6.5. We see a significant improvement in the performance of the feedforward network. We also

Training Data	AUC-ADD (↑) Feedforward	FF + Refiner
Fully Sim.	74.67	83.47
Finetune on Real	**86.16**	**90.36**

Table 6.5: **Results of fine-tuning the 6D pose estimation method on real data, over 25 measurements of the "2" object.**

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.

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. 6.8. We test two table materials: matte white and matte black.

The results of varying surface properties are presented in Tab. 6.6. 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.

Varying Ambient Light

We evaluate the performance of our method under varying levels of ambient lighting in Tab. 6.7, on a new set of 10 captures of the "2" object at each light level. We see consistent performance in darkness (<0.1 lux) and the same indoor

Figure 6.8: **"2" objects with different reflectance properties used in the varying scene reflectance experiment** (Sec. 6.7. From left to right: matte white, glossy white, and spotted black and white.

		AUC-ADD (↑)	
Obj. Material	**Table Material**	**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 6.6: **6D Pose Estimation of the "2" object with varying object and tabletop surface reflectance.**

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.

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

Table 6.7: **6D Pose Estimation of the "2" object under varying levels of ambient illumination.**

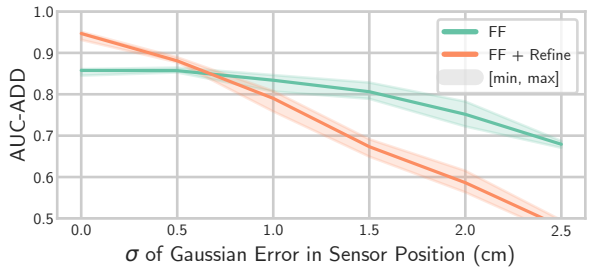


Figure 6.9: **Effect of adding Gaussian error to sensor poses on the "2" pose estimation task before feeding into our method.**

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. 6.9. 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.

Sensor Model Ablation Study

We perform an ablation study over key components of our sensor model as described in Sec. 6.3. We consider the following variants:

1. **Full:** The full sensor model as described in Sec. 6.3 and used for all

Number of Views	AUC-ADD (AUC-ADD-S) (↑) 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 6.8: **6D Pose Estimation with Different Numbers of Views.**

previous experiments.

2. **Idealized Jitter Kernel:** The jitter kernel $\boldsymbol{s}$ is replaced by a Dirac delta function at the location of the peak of $\boldsymbol{s}$.

3. **Inaccurate Bin Size:** The temporal bin size Δt 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° to 38°. Additionally, the intensity map $I(\boldsymbol{\omega})$ 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. 6.9. The results demonstrate that each of these aspects of sensor modeling are important to achieve good performance.

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

Ablation	AUC-ADD (↑) 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 6.9: Results of 6D Pose Estimation under varying sensor model ablations, over a dataset of 25 captures of the "2" object.

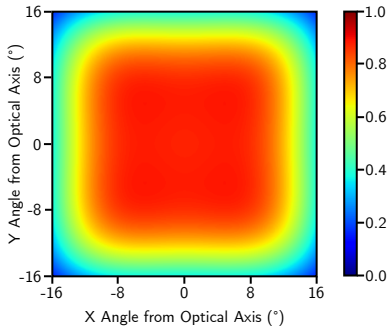
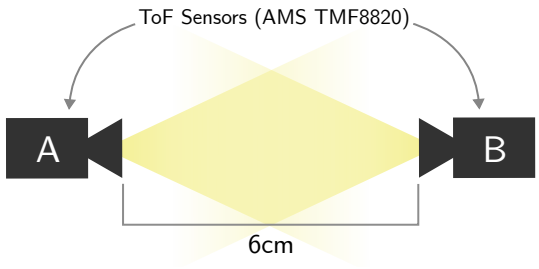


Figure 6.10: **Visualization of the laser intensity function $I(\omega)$ that we use for the TMF8820 sensor**, as given by Eq. (6.9). We set $K_1 = 0.88$, $K_2 = -3.16$, $K_3 = 250.51$.

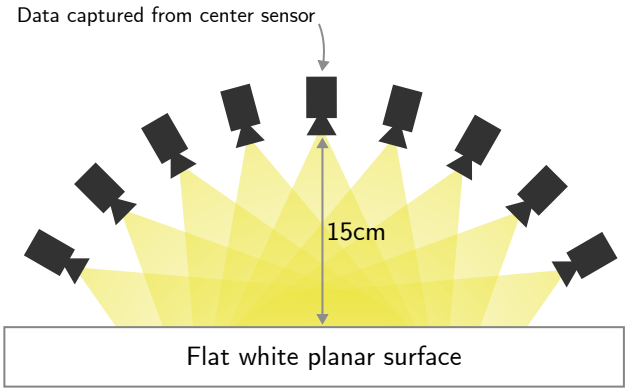
of our network takes ∼4.8 ms. The (unoptimized) refiner takes ∼2 seconds. With attention paid to efficiency, refiner speed could likely be increased. The costs of both the forward pass and optimization scale linearly with the number of viewpoints.

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.



(a) Sensor configuration for interference experiment 1.

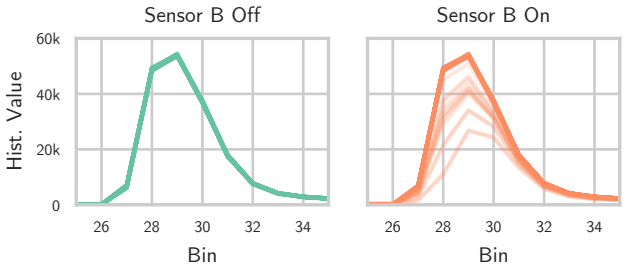


(b) Sensor configuration for interference experiment 2.

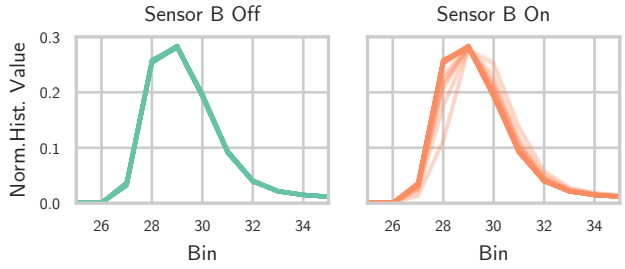
Figure 6.11: **Sensor configurations used for interference experiments.**

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. 6.11a. We compare measurements captured by sensor A between two conditions: sensor B on and sensor B off. The raw and normalized histograms for both conditions are shown in Fig. 6.12. 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 [36], 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



(a) Raw histograms



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

Figure 6.12: **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.

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 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. 6.13. In this case we can clearly see interference manifest as a DC offset in the captured histogram, again matching the signature of ambient light.

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

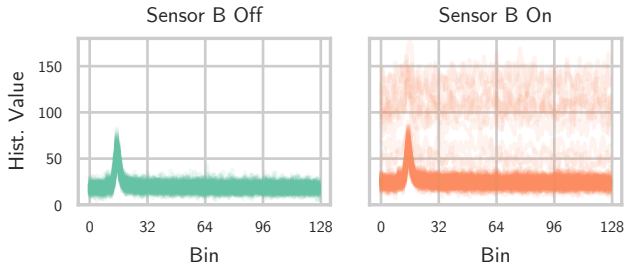
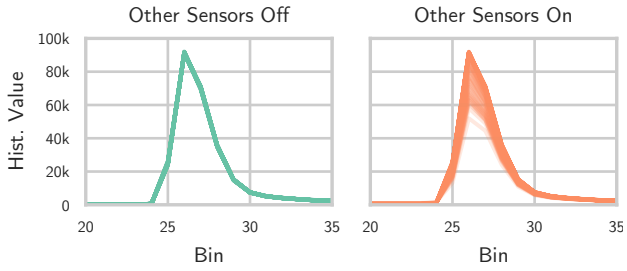


Figure 6.13: **Comparison of the histograms captured in interference 1, with the light source of sensor A covered.** Each plot shows 128 sensor measurements overlaid. About 90% of the samples in the right column exhibit no interference artifacts, comprising the dark orange line.

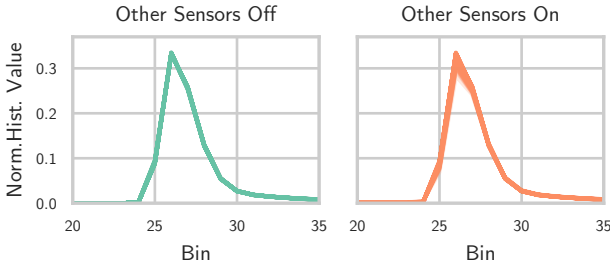
setup is illustrated in Fig. 6.11b. We position the sensors such that the centers of their optical axes each intersect with a planar surface at the same point, and record data only from the center sensor. Again, we compare between two conditions: the other 8 sensors on, and the other 8 sensors off. The results of this experiment are shown in Fig. 6.14. We see the same effect as in the previous experiment, but with a slightly higher occurrence rate of $\sim 25\%$.

Discussion: Between-Sensor Interference

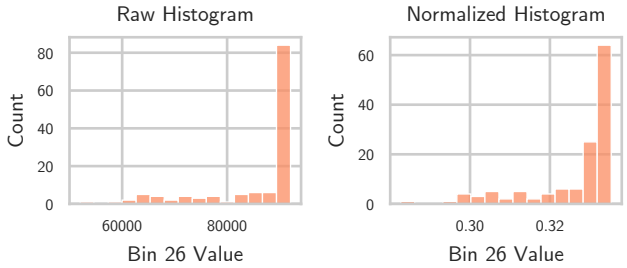
We have demonstrated that, at least for the AMS TMF8820 sensor, the effect of interference between sensors happens only occasionally even in the worst case. In practical scenarios, the rate of interference is likely to be quite low (*i.e.* $< 10\%$). Further, the effect of interference on the histogram appears to be similar to the effect of ambient light. Adjusting captured histograms to account for ambient light is a well-studied problem [36], 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.



(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 6.14: **Comparison of the histograms captured in interference experiment 2.**

Visualization of 6D Pose Results

We provide visualization of our results on 6D pose estimation in Figures 6.16 to 6.24.

Figure 6.15: **Objects used for 6D pose estimation experiments.**

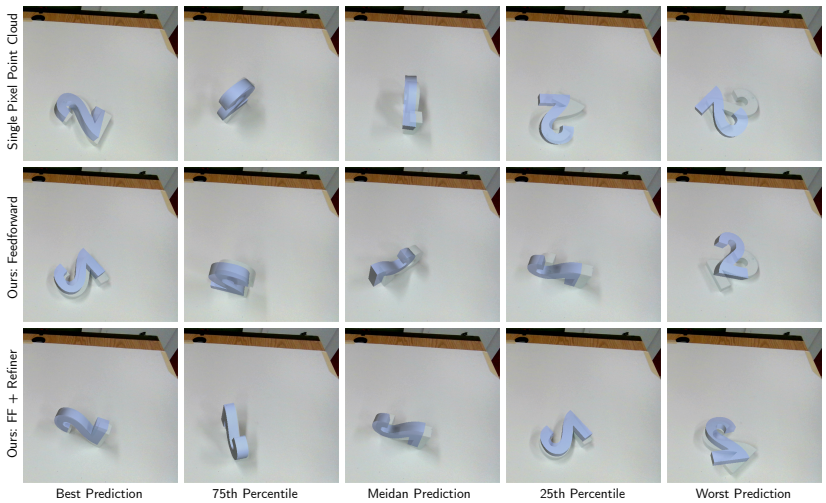


Figure 6.16: **Visualization of results on the 3D printed "two" object.**

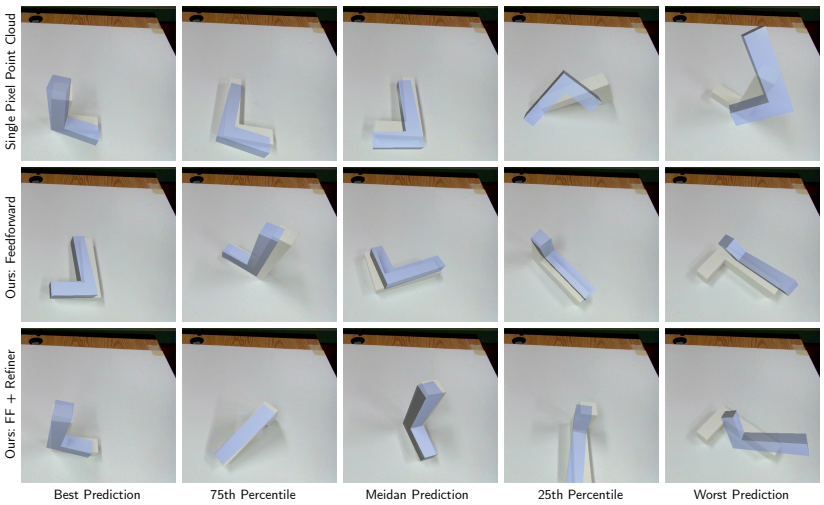


Figure 6.17: **Visualization of results on the 3D printed "L" object.**

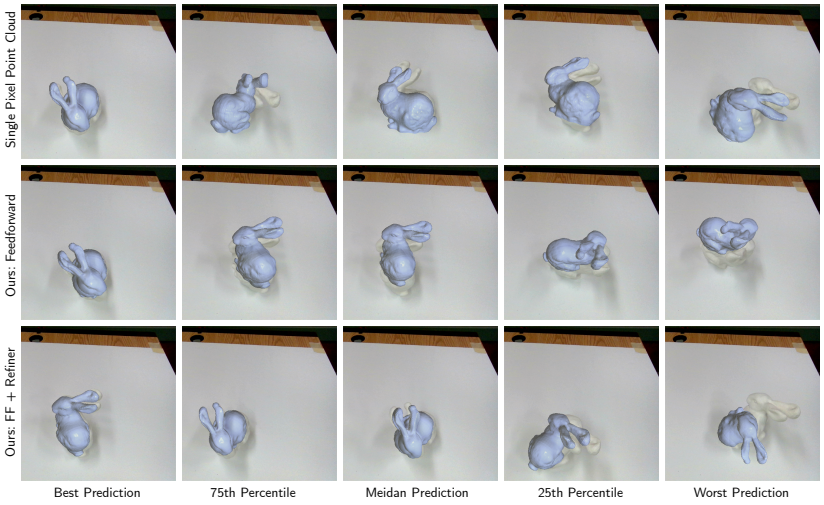


Figure 6.18: **Visualization of results on the 3D printed "bunny" object.**

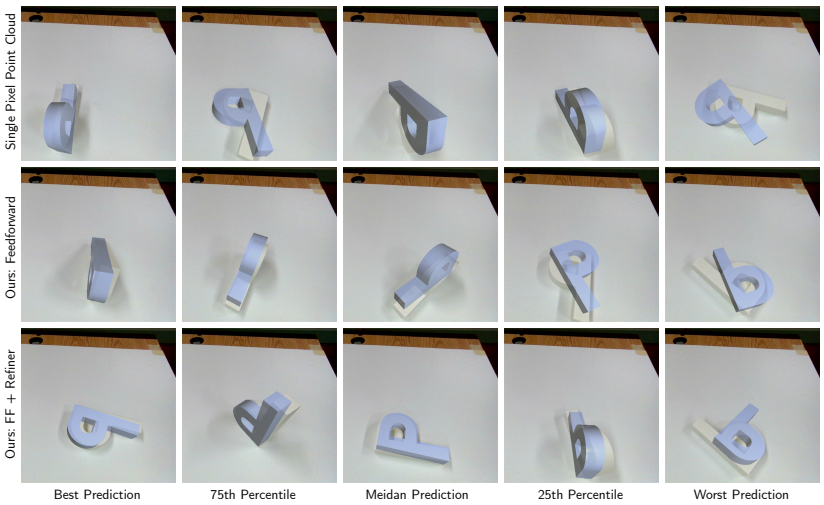


Figure 6.19: **Visualization of results on the 3D printed "P" object.**

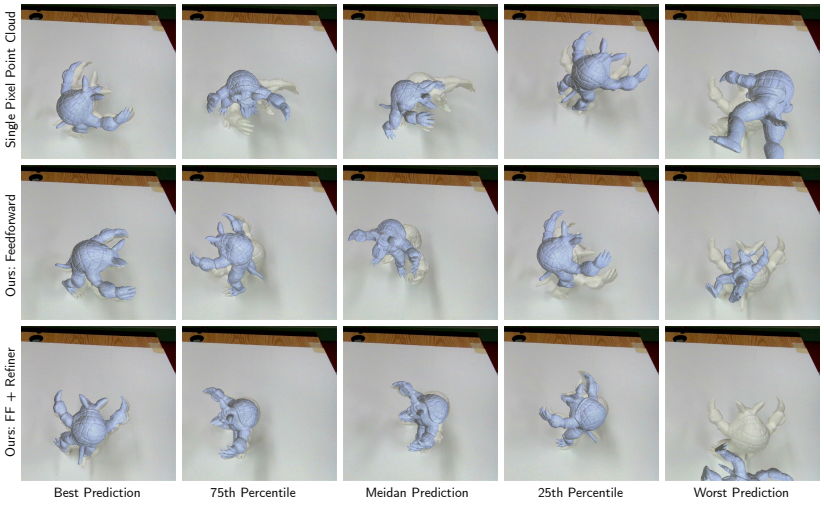


Figure 6.20: **Visualization of results on the 3D printed "armadillo" object.**

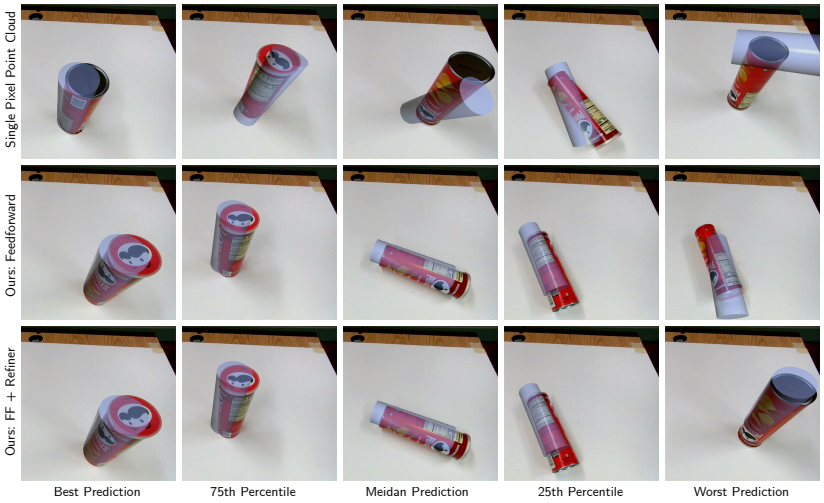


Figure 6.21: **Visualization of results on the "chips" object from the YCB dataset.**

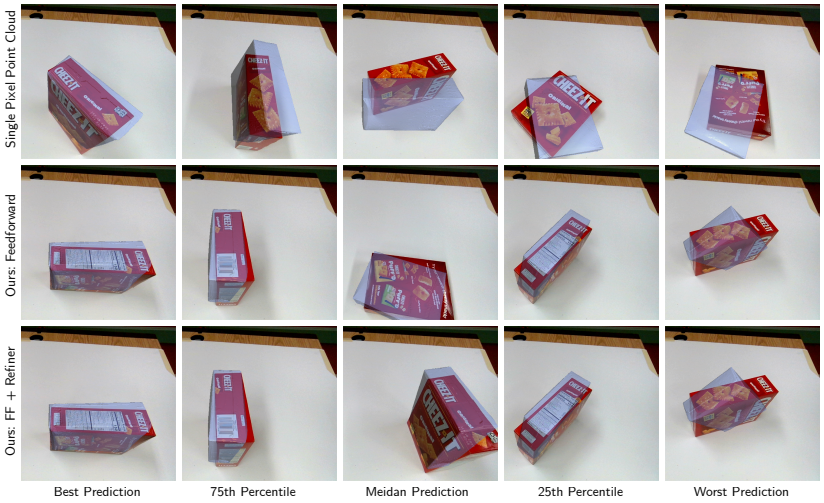


Figure 6.22: **Visualization of results on the "crackers" object from the YCB dataset.**

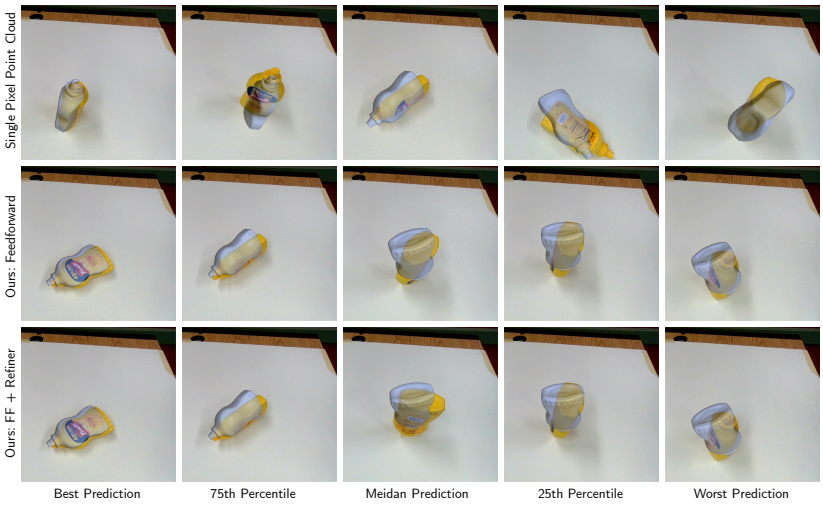


Figure 6.23: **Visualization of results on the "mustard" object from the YCB dataset.**

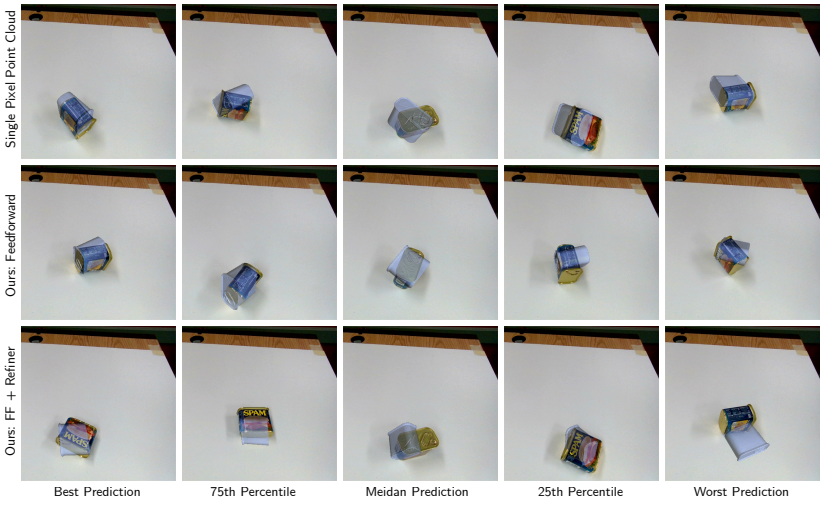


Figure 6.24: **Visualization of results on "SPAM" object from the YCB dataset.** The SPAM is a failure case for our method due to its specular surface, small size, and many near-symmetries which make optimization difficult.

7 CONCLUSION AND FUTURE OUTLOOK

In this dissertation, we demonstrated that miniature time-of-flight sensors can be used for demanding 3D sensing tasks despite their very limited spatial resolution. Along the way we modeled and characterized real sensor hardware, built systems for data capture, and real-time demos which incorporate sensors into control for robot arms and mobile robots. While the systems demonstrated in this work are far from complete products, they represent a first step towards systems which utilize a distributed set of miniature sensors to enable sensing from the "inside out", sensing the world with a minimal set of modular, highly configurable sensors.

Limitations and Open Problems

Geometric-Photometric Ambiguity: In Chapter 3, we demonstrate a fundamental ambiguity in a single transient histogram measurement: because the sensor measures incident light, a small, bright patch will appear the same as a large, dark patch. Using only a single sensor, this ambiguity became a limiting factor in the performance of our method for detecting deviations in planar surfaces. In theory, this ambiguity can be resolved in the multi-view setting. Future work should aim to develop general multi-view algorithms that resolve this ambiguity, and characterize the theoretical limits of such methods dependent on sensor configuration.

Multimodal Sensing & Deep Learning: The works presented in this dissertation utilize miniature time-of-flight sensors as the sole sensing modality. However, miniature time-of-flight sensors nicely compliment RGB cameras; RGB has high spatial resolution, while ToF has high depth resolution. RGB cameras work best in bright ambient light, while ToF sensors work best in no ambient light. Future work should aim to utilize the low-level techniques that we developed in this dissertation to make the most of multimodal RGB + ToF data. This direction has already been explored in existing work [81, 42], but these works do not

utilize the detailed sensor model and algorithms (*e.g.* background subtraction, analysis-by-synthesis) presented in this dissertation. Utilizing deep learning is a promising direction for solving multimodal problems, and doing so will likely require scaling up both real and synthetic data collection.

Distributed Sensing: A primary advantage of miniature time-of-flight sensors over traditional high-resolution LiDARs is that they can be distributed around a device, for example a robot or wearable device. This set of sensors can give an "inside-out" view of the world which is more robust to occlusions than a single high resolution sensor. Additionally, sensors can be positioned to customize coverage, *e.g.* providing high-density coverage in crucial areas while ignoring less important areas. In this dissertation, we largely demonstrate algorithms on a single sensor. We develop a capture setup with 8 simultaneous sensors in Chapter 6, but that setup resembles a prototype more than a product. Future work should aim to create miniature hardware prototypes which take full advantage of the form factor and modularity of miniature sensors. Such prototypes may require new algorithmic approaches to *e.g.* support movement of relative sensor positions or run on embedded hardware.

Hardware Limitations: Because miniature time-of-flight sensors are typically used for only coarse sensing, they are typically outfitted with only low-bandwidth data interfaces like I^2C. This proved a limiting factor for the frame rate of our end-to-end system in Chapter 3, Chapter 4, and Chapter 5. Additionally, many sensors provide limited software support for capture of transient histograms, and may artificially limit the histogram temporal resolution. We hope that the work presented in this dissertation demonstrates that access to high-speed, full quality transient histograms will unlock a range of applications for miniature time-of-flight sensors, thus encouraging extra attention from sensor manufacturers.

BIBLIOGRAPHY

[1] Odysseus Alexander Adamides, Anmol Saiprasad Modur, Shitij Kumar, and Ferat Sahin. 2019. A time-of-flight on-robot proximity sensing system to achieve human detection for collaborative robots. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, IEEE, Vancouver, BC, Canada, 1230–1236. 33, 55

[2] ams OSRAM AG. 2023. *TMF882X Datasheet*. ams OSRAM AG. https://ams.com/tmf8820#tab/documents 3, 8, 29, 35, 52, 58, 60, 75, 85, 86

[3] 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 Puhrsch, 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 Chanan, Peng Wu, and Soumith Chintala. 2024. 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* (La Jolla, CA, USA) *(ASPLOS '24)*. Association for Computing Machinery, New York, NY, USA, 929–947. https://doi.org/10.1145/3620665.3640366 87

[4] Apple Inc. 2023. *Apple Unveils iPhone 15 Pro and iPhone 15 Pro Max*. Apple, Inc. https://www.apple.com/newsroom/2023/09/apple-unveils-iphone-15-pro-and-iphone-15-pro-max/ Press Release. 1, 3

[5] Samira Badrloo, Masood Varshosaz, Saied Pirasteh, and Jonathan Li. 2022. Image-based obstacle detection methods for the safe navigation of unmanned vehicles: A review. *Remote Sensing* 14, 15 (2022), 3824. 9

[6] Cienna N Becker and Lucas J Koerner. 2023. Plastic Classification Using Optical Parameter Features Measured with the TMF8801 Direct Time-of-Flight Depth Sensor. *Sensors* 23, 6 (2023), 3324. 10, 56

[7] Nikhil Behari, Aaron Young, Siddharth Somasundaram, Tzofi Klinghoffer, Akshat Dave, and Ramesh Raskar. 2024. Blurred LiDAR for Sharper 3D: Robust Handheld 3D Scanning with Diffuse LiDAR and RGB. https://doi.org/10.48550/arXiv.2411.19474 arXiv:2411.19474 [eess]. 76, 79, 86

[8] P.J. Besl and Neil D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (Feb. 1992), 239–256. https://doi.org/10.1109/34.121791 Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. 85, 90, 93

[9] Liheng Bian, Haoze Song, Lintao Peng, Xuyang Chang, Xi Yang, Roarke Horstmeyer, Lin Ye, Tong Qin, Dezhi Zheng, and Jun Zhang. 2022. Large-scale single-photon imaging. (2022). arXiv preprint arXiv:2212.13654. 56

[10] Robert Bogue. 2013. Robotic vision boosts automotive industry quality and productivity. *Industrial Robot: An International Journal* 40, 5 (2013), 415–419. 1

[11] Alberto Broggi, Michele Buzzoni, Mirko Felisa, and Paolo Zani. 2011. Stereo obstacle detection in challenging environments: the VIAC experience. In *IROS*. IEEE, IEEE, San Francisco, USA, 1599–1604. 9

[12] Christopher Brown. 1976. *Principal Axes and Best-Fit Planes, with Applications*. Technical Report TR7. Department of Computer Science, University of Rochester, Rochester, New York. 63

[13] Giovanni Buizza Avanzini, Nicola Maria Ceriani, Andrea Maria Zanchettin, Paolo Rocco, and Luca Bascetta. 2014. Safety Control of Industrial Robots Based on a Distributed Distance Sensor. *Transactions on Control Systems Technology* 22, 6 (Nov. 2014), 2127–2140. https://doi.org/10.1109/TCST.2014.2300696 32

[14] Mauro Buttafava, Jessica Zeman, Alberto Tosi, Kevin Eliceiri, and Andreas Velten. 2015. Non-line-of-sight imaging using a time-gated single photon avalanche diode. *Optics express* 23, 16 (2015), 20997–21011. 55

[15] Dingding Cai, Janne Heikkia, and Esa Rahtu. 2022. OVE6D: Object Viewpoint Encoding for Depth-based 6D Object Pose Estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 6793–6803. https://doi.org/10.1109/CVPR52688.2022.00668 79

[16] Clara Callenberg, Zheng Shi, Felix Heide, and Matthias B. Hullin. 2021. Low-cost SPAD sensing for non-line-of-sight tracking, material classification and depth imaging. *ACM Transactions on Graphics* 40, 4 (Aug. 2021), 1–12. https://doi.org/10.1145/3450626.3459824 8, 10, 29, 33, 55, 78

[17] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. 2015. The YCB object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, Istanbul, Turkey, 510–517. https://doi.org/10.1109/ICAR.2015.7251504 83, 88

[18] Ted Camus, David Coombs, Martin Herman, and Tsai-Hong Hong. 1996. Real-time single-workstation obstacle avoidance using only wide-field flow divergence. In *CVPR*, Vol. 3. IEEE, IEEE, San Francisco, USA, 323–330. 9

[19] Yangyu Chen, Jiayuan Lin, and Xiaohan Liao. 2022. Early detection of tree encroachment in high voltage powerline corridor using growth model and UAV-borne LiDAR. *International Journal of Applied Earth Observation and Geoinformation* 108 (2022), 102740. 1

[20] PB Coates. 1968. The correction for photon pile-up in the measurement of radiative lifetimes. *Journal of Physics E: Scientific Instruments* 1, 8 (1968), 878. 15, 82

[21] D. Conrad and G. N. DeSouza. 2010. Homography-based ground plane detection for mobile robot navigation using a Modified EM algorithm. In *International Conference on Robotics and Automation*. IEEE, Anchorage, USA, 910–915. https://doi.org/10.1109/ROBOT.2010.5509457 9

[22] Sergio Cova, Massimo Ghioni, Andrea Lacaita, Carlo Samori, and Franco Zappa. 1996. Avalanche photodiodes and quenching circuits for single-photon detection. *Applied optics* 35, 12 (1996), 1956–1976. 53

[23] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Vol. 1. Ieee, IEEE, San Diego, USA, 886–893. 4

[24] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39, 1 (1977), 1–22. 17

[25] Nathan Devrio and Chris Harrison. 2022. 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 (UIST '22)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3526113.3545634 93

[26] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. 2021. Pixel-wise anomaly detection in complex driving scenes. In *CVPR*. IEEE, Virtual, 16918–16927. 10

[27] David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth map prediction from a single image using a multi-scale deep network. *NIPS* 27 (2014), 1–6. 14

[28] Caleb Escobedo, Matthew Strong, Mary West, Ander Aramburu, and Alessandro Roncone. 2021. Contact anticipation for physical human–robot interaction with robotic manipulators using onboard proximity sensors. In *IROS*. IEEE, IEEE, Prague, Czech Republic, 7255–7262. 6, 10, 28, 33, 50, 55

[29] Daniele Faccio, Andreas Velten, and Gordon Wetzstein. 2020. Non-line-of-sight imaging. *Nature Reviews Physics* 2, 6 (2020), 318–327. 78

[30] Xiaoran Fan, Riley Simmons-Edler, Daewon Lee, Larry Jackel, Richard Howard, and Daniel Lee. 2021. AuraSense: Robot Collision Avoidance by Full Surface Proximity Detection. In *International Conference on Intelligent Robots and Systems*. IEEE, Prague, Czech Republic, 1763–1770. https://doi.org/10.1109/IROS51168.2021.9635919 31

[31] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. 2018. Deep ordinal regression network for monocular depth estimation. In *CVPR*. IEEE, Salt Lake City, USA, 2002–2011. 14

[32] Mark N. Gamadia, Abhishek Dhanda, Gregory Guyomarc'h, Andrew D. Fernandez, and Moshe Lalfenfeld. 2023. Camera autofocus using time-of-flight assistance. US Patent 11,856,293. https://patents.google.com/patent/US11856293B2/en Issued December 26, 2023. Assignee: Apple Inc.. 1

[33] Daiheng Gao, Yuliang Xiu, Kailin Li, Lixin Yang, Feng Wang, Peng Zhang, Bang Zhang, Cewu Lu, and Ping Tan. 2022. DART: Articulated Hand Model with Diverse Accessories and Rich Textures. https://doi.org/10.48550/arXiv.2210.07650 arXiv:2210.07650 [cs]. 93

[34] Francesco Giovinazzo, Francesco Grella, Marco Sartore, Manuela Adami, Riccardo Galletti, and Giorgio Cannata. 2024. From CySkin to ProxySKIN: Design, Implementation and Testing of a Multi-Modal Robotic Skin for Human–Robot Interaction. *Sensors* 24, 4 (2024), 1334. 31

[35] Anant Gupta, Atul Ingle, and Mohit Gupta. 2019. Asynchronous single-photon 3D imaging. In *International Conference on Computer Vision*. IEEE, Los Angeles, USA, 7909–7918. 54

[36] Anant Gupta, Atul Ingle, Andreas Velten, and Mohit Gupta. 2019. Photon-Flooded Single-Photon 3D Cameras. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Long Beach, CA, USA, 6763–6772. https://doi.org/10.1109/CVPR.2019.00693 15, 36, 49, 57, 81, 104, 106

[37] Krishnam Gupta, Syed Ashar Javed, Vineet Gandhi, and K Madhava Krishna. 2018. Mergenet: A deep net architecture for small obstacle discovery. In *ICRA*. IEEE, IEEE, Brisbane, Australia, 5856–5862. 10

[38] Felipe Gutierrez-Barragan, Atul Ingle, Trevor Seets, Mohit Gupta, and Andreas Velten. 2022. Compressive single-photon 3D cameras. In *CVPR*. IEEE, New Orleans, USA, 17854–17864. 3, 11, 35

[39] Felix Heide, Matthew O'Toole, Kai Zang, David B Lindell, Steven Diamond, and Gordon Wetzstein. 2019. Non-line-of-sight imaging with partial occluders and surface normals. *ACM Transactions on Graphics (ToG)* 38, 3 (2019), 1–10. 78

[40] Urban B Himmelsbach, Thomas M Wendt, Nikolai Hangst, and Philipp Gawron. 2019. Single pixel time-of-flight sensors for object detection and self-detection in three-sectional single-arm robot manipulators. In *International Conference on Robotic Computing*. IEEE, IEEE, Naples, Italy, 250–253. 32

[41] Minjie Hua, Yibing Nan, and Shiguo Lian. 2019. Small Obstacle Avoidance Based on RGB-D Semantic Segmentation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, Los Angeles, USA, 886–894. https://doi.org/10.1109/ICCVW.2019.00117 9, 10

[42] Sacha Jungerman, Atul Ingle, Yin Li, and Mohit Gupta. 2022. 3D Scene Inference from Transient Histograms. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Vol. 13667. Springer Nature Switzerland, Cham, 401–417. https://doi.org/10.1007/978-3-031-20071-7_24 Series Title: Lecture Notes in Computer Science. 3, 8, 10, 11, 13, 34, 35, 36, 56, 60, 76, 79, 81, 113

[43] Samer Karam, Francesco Nex, Bhanu Teja Chidura, and Norman Kerle. 2022. Microdrone-based indoor mapping with graph slam. *Drones* 6, 11 (2022), 352. 1, 6, 10, 33

[44] Mohammad Amin Karimi, David Cañones Bonham, Esteban Lopez, Ankit Srivastava, and Matthew Spenko. 2023. SLAM and Shape Estimation for Soft Robots. In *IROS*. IEEE, IEEE, Detroit, USA, 9133–9138. 6

[45] Samantha Murphy Kelly. 2024. *A billion laser points helped bring Notre Dame back to life*. CNN Business. https://www.cnn.com/2024/12/23/tech/ai-models-tech-notre-dame-cathedral/index.html 1

[46] Daehwa Kim, Mario Srouji, Chen Chen, and Jian Zhang. 2024. ARMOR: Egocentric Perception for Humanoid Robot Collision Avoidance and Motion Planning. https://doi.org/10.48550/arXiv.2412.00396 arXiv:2412.00396 [cs]. 31, 33, 50

[47] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980 arXiv:1412.6980 [cs]. 61, 87

[48] Ahmed Kirmani, Tyler Hutchison, James Davis, and Ramesh Raskar. 2009. Looking around the corner using transient imaging. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, IEEE, Kyoto, Japan, 159–166. 78

[49] Tzofi Klinghoffer, Xiaoyu Xiang, Siddharth Somasundaram, Yuchen Fan, Christian Richardt, Ramesh Raskar, and Rakesh Ranjan. 2024. 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*. IEEE, Seattle, USA, 14565–14574. 78

[50] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. 2015. Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Santiago, Chile, 954–962. https://doi.org/10.1109/ICCV.2015.115 79

[51] Suryansh Kumar, M Siva Karthik, and K Madhava Krishna. 2014. Markov random field based small obstacle discovery over images. In *ICRA*. IEEE, IEEE, Hong Kong, 494–500. 9

[52] Kiriakos N Kutulakos and Steven M Seitz. 2000. A theory of shape by space carving. *IJCV* 38 (2000), 199–218. 14

[53] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. 2022. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. http://arxiv.org/abs/2212.06870 arXiv:2212.06870 [cs]. 79

[54] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. 2020. Modular Primitives for High-Performance Differentiable Rendering. *ACM Transactions on Graphics* 39, 6 (2020), 1–14. 87

[55] Przemyslaw A Lasota, Gregory F Rossano, and Julie A Shah. 2014. Toward safe close-proximity human-robot interaction with standard industrial robots. In *International Conference on Automation Science and Engineering*. IEEE, IEEE, Taipei, Taiwan, 339–344. 28

[56] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. 2025. Cameras as Relative Positional Encoding. (2025). arXiv preprint arXiv:2507.10496. 83

[57] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (2017), 194:1–194:17. https://doi.org/10.1145/3130800.3130813 80

[58] Yijin Li, Xinyang Liu, Wenqi Dong, Han Zhou, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. 2022. DELTAR: Depth Estimation from a Light-Weight ToF Sensor and RGB Image. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Vol. 13661. Springer Nature Switzerland, Cham, 619–636. https://doi.org/10.1007/978-3-031-19769-7_36 Series Title: Lecture Notes in Computer Science. 10, 34, 76, 79

[59] Jiacai Liao, Libo Cao, Xiaole Luo, Xu Sun, Cong Duan, Jianhua Li, and Feng Yuan. 2022. Road Garbage Segmentation With Deep Supervision and High Fusion Network for Cleaning Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2022), 11190–11204. https://doi.org/10.1109/TITS.2021.3101400 10

[60] Fengyuan Liu, Sweety Deswal, Adamos Christou, Yulia Sandamirskaya, Mohsen Kaboli, and Ravinder Dahiya. 2022. Neuro-inspired electronic skin for robots. *Science robotics* 7, 67 (2022), eabl7344. 31

[61] Xinyang Liu, Yijin Li, Yanbin Teng, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. 2023. Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor. In *ICCV*. IEEE, Paris, France, 1–11. 10, 34, 78, 79

[62] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16. 80

[63] David G Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, Vol. 2. Ieee, IEEE, Kerkrya, Greece, 1150–1157. 4

[64] Vladimir Lumelsky, Michael S Shur, Sigurd Wagner, and Mingzhou Ding. 2000. *Sensitive skin*. Vol. 18. World Scientific, Singapore. 31

[65] Weihan Luo, Anagh Malik, and David B. Lindell. 2024. Transientangelo: Few-Viewpoint Surface Reconstruction Using Single-Photon Lidar. https://doi.org/10.48550/arXiv.2408.12191 arXiv:2408.12191. 78, 79

[66] Natalia Lyubova, David Filliat, and Serena Ivaldi. 2013. Improving object learning through manipulation and robot self-identification. In *International Conference on Robotics and Biomimetics*. IEEE, Shenzhen, China, 1365–1370. https://doi.org/10.1109/ROBIO.2013.6739655 32

[67] Anagh Malik, Noah Juravsky, Ryan Po, Gordon Wetzstein, Kiriakos N Kutulakos, and David B Lindell. 2024. Flying with photons: Rendering novel views of propagating light. In *European Conference on Computer Vision*. Springer, Springer, Milan, Italy, 333–351. 77

[68] Anagh Malik, Parsa Mirdehghan, Sotiris Nousias, Kyros Kutulakos, and David Lindell. 2023. Transient Neural Radiance Fields for Lidar View Synthesis and 3D Reconstruction. *Advances in Neural Information Processing Systems* 36 (Dec. 2023), 71569–71581. https://proceedings.neurips.cc/paper_files/paper/2023/hash/e261e92e1cfb820da930ad8c38d0aead-Abstract-Conference.html 3, 14, 78

[69] D Marko and D Hrubỳ. 2020. Distance measuring in vineyard row using ultrasonic and optical sensors. In *Proceedings of 22nd International Conference of Young Scientists. Praha: Česká zemědělská univerzita*. Czech University of Life Sciences Prague, Prague, Czech Republic, 194–204. 55

[70] Jessica Mathews. 2025. *Waymo experimenting with a generative AI frontier model, but exec says LiDAR and radar sensors important to self-driving safety 'under all conditions'*. Fortune. https://fortune.com/2025/08/15/waymo-srikanth-thirumalai-interview-ai4-conference-las

-vegas-lidar-radar-self-driving-safety-tesla/ Accessed: 2026-01-08. 1

[71] P. Michel, K. Gold, and B. Scassellati. 2004. Motion-based robotic self-recognition. In *International Conference on Robotics and Systems*, Vol. 3. IEEE, Sendai, Japan, 2763–2768 vol.3. https://doi.org/10.1109/IROS.2004.1389827 32

[72] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106. 71

[73] Kazuhiro Morimoto, Andrei Ardelean, Ming-Lo Wu, Arin Can Ulku, Ivan Michel Antolovic, Claudio Bruschini, and Edoardo Charbon. 2020. Megapixel time-gated SPAD image sensor for 2D and 3D imaging applications. *Optica* 7, 4 (April 2020), 346. https://doi.org/10.1364/OPTICA.386574 75, 77

[74] Fangzhou Mu, Sicheng Mo, Jiayong Peng, Xiaochun Liu, Ji Hyun Nam, Siddeshwar Raghavan, Andreas Velten, and Yin Li. 2025. Physics to the Rescue: Deep Non-Line-of-Sight Reconstruction for High-Speed Imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47, 8 (2025), 6146–6158. https://doi.org/10.1109/TPAMI.2022.3203383 60

[75] Fangzhou Mu, Carter Sifferman, Sacha Jungerman, Yiquan Li, Mark Han, Michael Gleicher, Mohit Gupta, and Yin Li. 2024. Towards 3D Vision with Low-Cost Single-Photon Cameras. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 5302–5311. https://doi.org/10.1109/CVPR52733.2024.00507 10, 14, 34, 35, 39, 76, 78, 79, 81, 86, 87

[76] Hanna Müller, Vlad Niculescu, Tommaso Polonelli, Michele Magno, and Luca Benini. 2023. Robust and efficient depth-based obstacle avoidance for autonomous miniaturized uavs. *Transactions on Robotics* 39, 6 (2023), 4935–4951. 33

[77] Stefan Escaida Navarro, Stephan Mühlbacher-Karrer, Hosam Alagi, Hubert Zangl, Keisuke Koyama, Björn Hein, Christian Duriez, and Joshua R Smith. 2021.

Proximity perception in human-centered robotics: A survey on sensing systems and applications. *Transactions on Robotics* 38, 3 (2021), 1599–1620. 31

[78] John A. Nelder and Roger Mead. 1965. A Simplex Method for Function Minimization. *Comput. J.* 7 (1965), 308–313. 63

[79] C. Niclass, A. Rochas, P.-A. Besse, and E. Charbon. 2005. Design and characterization of a CMOS 3-D image sensor based on single photon avalanche diodes. *IEEE Journal of Solid-State Circuits* 40, 9 (Sept. 2005), 1847–1854. https://doi.org/10.1109/JSSC.2005.848173 Conference Name: IEEE Journal of Solid-State Circuits. 3, 11, 35, 75

[80] Vlad Niculescu, Tommaso Polonelli, Michele Magno, and Luca Benini. 2024. NanoSLAM: Enabling Fully Onboard SLAM for Tiny Robots. *IEEE Internet of Things Journal* 11, 8 (April 2024), 13584–13607. https://doi.org/10.1109/JIOT.2023.3339254 3, 8, 10, 33

[81] Mark Nishimura, David B Lindell, Christopher Metzler, and Gordon Wetzstein. 2020. Disambiguating monocular depth estimation with a single transient. In *ECCV*. Springer, Springer, Glasgow, UK, 139–155. 10, 34, 113

[82] Dan O'Shea. 2025. UK start-up Singular Photonics touts SPAD sensors, Meta collab. https://www.fierceelectronics.com/sensors/uk-start-singular-photonics-touts-spad-sensors-meta-collab Accessed: 2025-03-07. 75

[83] Ouster, Inc. 2023. Ouster Awarded Production Win by Motional to be the Exclusive Supplier of Long-Range Lidar for its Autonomous Vehicles. Press Release. https://www.ouster.com SAN FRANCISCO–(BUSINESS WIRE). 3

[84] Kiru Park, Timothy Patten, and Markus Vincze. 2019. Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Seoul, South Korea, 7667–7676. https://doi.org/10.1109/ICCV.2019.00776 arXiv:1908.07433 [cs]. 79

[85] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. 2024. Reconstructing Hands in 3D with Transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 9826–9836. https://doi.org/10.1109/CVPR52733.2024.00938 93

[86] Adithya K Pediredla, Aswin C Sankaranarayanan, Mauro Buttafava, Alberto Tosi, and Ashok Veeraraghavan. 2018. Signal processing based pile-up compensation for gated single-photon avalanche diodes. (2018). arXiv preprint arXiv:1806.07437. 54

[87] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *JMLR* 12 (2011), 2825–2830. 19

[88] Sara Pellegrini, Gerald S Buller, Jason M Smith, Andrew M Wallace, and Sergio Cova. 2000. Laser-based distance measurement using picosecond resolution time-correlated single-photon counting. *Measurement Science and Technology* 11, 6 (2000), 712. 53

[89] Bui Tuong Phong. 1975. Illumination for computer generated pictures. *Commun. ACM* 18, 6 (1975), 311–317. 57

[90] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. 2016. Lost and found: detecting small road hazards for self-driving vehicles. In *IROS*. IEEE, IEEE, Daejeon, South Korea, 1099–1106. 9

[91] pmdtechnologies ag and Infineon Technologies AG. 2025. *Sensor technology from pmd and Infineon enables industry-first 5-axis robot arm for next-level cleaning robots*. Roborock. https://pmdtec.com/en/company/news/pmd-and-infineon-sensor-technology-enables-industry-first-5-axis-robot-arm-for-next-level-cleaning-robots/ Press Release. 1

[92] Shyam Nandan Rai, Fabio Cermelli, Dario Fontanel, Carlo Masone, and Barbara Caputo. 2023. Unmasking anomalies in road-scene segmentation. In *ICCV*. IEEE, Paris, France, 4037–4046. 10

[93] Daniel Rakita, Haochen Shi, Bilge Mutlu, and Michael Gleicher. 2021. CollisionIK: A Per-Instant Pose Optimization Method for Generating Robot Motions with Environment Collision Avoidance. In *2021 International Conference on Robotics and Automation*. IEEE, Xi'an, China, 9995–10001. https://doi.org/10.1109/ICRA48506.2021.9561505 50

[94] Panjawee Rakprayoon, Miti Ruchanurucks, and Ada Coundoul. 2011. Kinect-based obstacle detection for manipulator. In *International Symposium on System Integration*. IEEE, Kyoto, Japan, 68–73. https://doi.org/10.1109/SII.2011.6147421 32

[95] Sebastian Ramos, Stefan Gehrig, Peter Pinggera, Uwe Franke, and Carsten Rother. 2017. Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling. In *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, IEEE, Redondo Beach, USA, 1025–1032. 10

[96] Roborock. 2023. *Roborock Introduces S7 Max Ultra and Q Revo With Unique Roborock App Features For Seamless Cleaning Experiences*. Roborock. https://newsroom.roborock.com/us/news/roborock-introduces-s7-max-ultra-and-q-revo-with-unique-roborock-app-features-for-seamless-cleaning-experiences Press Release. 1

[97] Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017), 1–17. 80, 92, 97

[98] Stéphen Rostain, Antoine Dorison, Geoffroy de Saulieu, Heiko Prümers, Jean-Luc Le Pennec, Fernando Mejía Mejía, Ana Maritza Freire, Jaime R. Pagán-Jiménez, and Philippe Descola. 2024. Two thousand years of garden urbanism in the Upper Amazon. *Science* 383, 6679 (2024), 183–189. https://doi.org/10.1126/science.adi6317 arXiv:https://www.science.org/doi/pdf/10.1126/science.adi6317 1

[99] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*. Ieee, IEEE, Barcelona, Spain, 2564–2571. 4

[100] Alice Ruget, Max Tyler, Germán Mora Martín, Stirling Scholes, Feng Zhu, Istvan Gyongy, Brent Hearn, Steve McLaughlin, Abderrahim Halimi, and Jonathan Leach. 2021. Real-time, low-cost multi-person 3D pose estimation. (2021). arXiv preprint arXiv:2110.11414. 56

[101] Alice Ruget, Max Tyler, Germán Mora-Martín, Stirling Scholes, Feng Zhu, Istvan Gyongy, Brent Hearn, Steve McLaughlin, Abderrahim Halimi, and Jonathan Leach. 2022. Pixels2Pose: Super-Resolution Time-of-Flight Imaging for 3D Pose Estimation. In *Imaging and Applied Optics Congress 2022 (3D, AOA, COSI, ISA, pcAOP)*. Optica Publishing Group, Vancouver, British Columbia, ITh5D.5. https://doi.org/10.1364/ISA.2022.ITh5D.5 10, 34, 56, 78

[102] David Eric Schwartz, Edoardo Charbon, and Kenneth L Shepard. 2008. A single-photon avalanche diode array for fluorescence lifetime imaging microscopy. *IEEE journal of solid-state circuits* 43, 11 (2008), 2546–2557. 77

[103] Malarvizhi Selvaraj, Vasilios Baltzopoulos, Andy Shaw, Constantinos N Maganaris, Jeff Cullen, Thomas O'Brien, and Patryk Kot. 2018. Stair fall risk detection using wearable sensors. In *2018 11th International Conference on Developments in eSystems Engineering (DeSE)*. IEEE, IEEE, Cambridge, UK, 108–112. 55

[104] Siyuan Shen, Zi Wang, Ping Liu, Zhengqing Pan, Ruiqian Li, Tian Gao, Shiying Li, and Jingyi Yu. 2021. Non-line-of-sight imaging via neural transient fields. *Transactions on Pattern Analysis and Machine Intelligence* 43, 7 (2021), 2257–2268. 55

[105] Carter Sifferman, Dev Mehrotra, Mohit Gupta, and Michael Gleicher. 2022. Geometric Calibration of Single-Pixel Distance Sensors. *IEEE Robotics and Automation Letters* 7, 3 (2022), 6598–6605. 10, 33, 55

[106] Carter Sifferman, William Sun, Mohit Gupta, and Michael Gleicher. 2024. Using a Distance Sensor to Detect Deviations in a Planar Surface. *IEEE Robotics and*

Automation Letters 9, 10 (Oct. 2024), 8515–8522. https://doi.org/10.1109/LRA.2024.3445665 Conference Name: IEEE Robotics and Automation Letters. 33, 34, 36, 39, 40, 86

[107] Carter Sifferman, Yeping Wang, Mohit Gupta, and Michael Gleicher. 2023. Unlocking the Performance of Proximity Sensors by Utilizing Transient Histograms. *IEEE Robotics and Automation Letters* 8, 10 (Oct. 2023), 6843–6850. https://doi.org/10.1109/LRA.2023.3313069 18, 29, 33, 34, 35, 38, 39, 78, 79, 81, 86, 87

[108] ST Microelectronics. 2023. *VL53L8CH Datasheet*. ST Microelectronics. https://www.st.com/resource/en/datasheet/vl53l8ch.pdf 3, 8, 29, 35

[109] ST Microelectronics. 2023. *VL6180X Proximity and Ambient Light Sensing Module Datasheet*. ST Microelectronics. https://www.st.com/resource/en/datasheet/vl6180x.pdf 3, 35, 52, 75

[110] Chris Stauffer and W Eric L Grimson. 1999. Adaptive background mixture models for real-time tracking. In *CVPR*, Vol. 2. IEEE, IEEE, Fort Collins, USA, 246–252. 16

[111] NO Stoffler, Tim Burkert, and G Farber. 2000. Real-time obstacle avoidance using an MPEG-processor-based optic flow sensor. In *International Conference on Pattern Recognition (ICPR)*, Vol. 4. IEEE, IEEE, Barcelona, Spain, 161–166. 9

[112] Wantana Sukmanee, Miti Ruchanurucks, and Panjawee Rakprayoon. 2012. Obstacle modeling for manipulator using iterative least square (ILS) and iterative closest point (ICP) base on Kinect. In *International Conference on Robotics and Biomimetics*. IEEE, Ghangzhou, China, 672–676. https://doi.org/10.1109/ROBIO.2012.6491044 32

[113] Petr Svarny, Michael Tesar, Jan Kristof Behrens, and Matej Hoffmann. 2019. Safe physical HRI: Toward a unified treatment of speed and separation monitoring together with power and force limiting. In *International Conference on Intelligent Robots and Systems*. IEEE, Montreal, Canada, 7580–7587. https://doi.org/10.1109/IROS40897.2019.8968463 28

[114] TechInsights. 2023. iPhone 15 Pro Max Rear LiDAR Camera Process Flow Analysis. https://www.techinsights.com/blog/iphone-15-pro-max-rear-lidar-camera-process-flow-analysis Accessed: 2025-03-07. 75

[115] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. 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)*. IEEE, IEEE, Vancouver, Canada, 23–30. 88

[116] Jonathan Tremblay, Bowen Wen, Valts Blukis, Balakumar Sundaralingam, Stephen Tyree, and Stan Birchfield. 2023. Diff-DOPE: Differentiable Deep Object Pose Estimation. http://arxiv.org/abs/2310.00463 arXiv:2310.00463 [cs]. 79

[117] S. Tsuji and T. Kohama. 2019. Proximity Skin Sensor Using Time-of-Flight Sensor for Human Collaborative Robot. *IEEE Sensors Journal* 19, 14 (2019), 5859–5864. https://doi.org/10.1109/JSEN.2019.2905848 10, 31, 55

[118] Satoshi Tsuji and Teruhiko Kohama. 2022. Omnidirectional Proximity Sensor System for Drones Using Optical Time-of-Flight Sensors. *IEEJ Transactions on Electrical and Electronic Engineering* 17, 1 (2022), 19–25. 3, 8, 10, 33, 55

[119] Dugan Um, Brooke Stankovic, Kendall Giles, Troy Hammond, and V Lumelsky. 1998. A modularized sensitive skin for motion planning in uncertain environments. In *International Conference on Robotics and Automation*, Vol. 1. IEEE, IEEE, Leuven, Belgium, 7–12. 31

[120] Universal Robots. 2016. UR5 Technical Specifications. Online. https://www.universal-robots.com/media/50588/ur5_en.pdf Accessed: 2025-02-28. 84

[121] Carlos Ureña, Marcos Fajardo, and Alan King. 2013. An Area-Preserving Parametrization for Spherical Rectangles. *Computer Graphics Forum* 32,

4 (2013), 59–66. https://doi.org/10.1111/cgf.12151 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12151 60

[122] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017), 6000–6010. 83

[123] Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G Bawendi, and Ramesh Raskar. 2012. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications* 3, 1 (2012), 745. 55

[124] Xinyu Wang, Chenguang Yang, Zhaojie Ju, Hongbin Ma, and Mengyin Fu. 2017. Robot manipulator self-identification for surrounding obstacle detection. *Multimedia Tools and Applications* 76, 5 (March 2017), 6495–6520. https://doi.org/10.1007/s11042-016-3275-8 32

[125] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. 2024. FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 17868–17879. https://doi.org/10.1109/CVPR52733.2024.01692 79, 88, 89, 90

[126] Fan Xia, Behraad Bahreyni, and Fabio Campi. 2016. Multi-functional capacitive proximity sensing system for industrial safety applications. In *Sensors*. IEEE, IEEE, Orlando, USA, 1–3. 31

[127] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. 2018. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. https://doi.org/10.48550/arXiv.1711.00199 arXiv:1711.00199 [cs]. 79, 87, 88, 96

[128] Shumian Xin, Sotiris Nousias, Kiriakos N. Kutulakos, Aswin C. Sankaranarayanan, Srinivasa G. Narasimhan, and Ioannis Gkioulekas. 2019. 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*. IEEE, Los Angeles,

USA, 6800–6809. https://openaccess.thecvf.com/content_CVPR_2019/html/Xin_A_Theory_of_Fermat_Paths_for_Non-Line-Of-Sight_Shape_Reconstruction_CVPR_2019_paper.html 78

[129] Wenqiang Xu, Zhenjun Yu, Han Xue, Ruolin Ye, Siqiong Yao, and Cewu Lu. 2023. Visual-Tactile Sensing for In-Hand Object Reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Vancouver, BC, Canada, 8803–8812. https://doi.org/10.1109/CVPR52729.2023.00850 80

[130] Nivasan Yogeswaran, Wenting Dang, William Taube Navaraj, Dhayalan Shakthivel, Saleem Khan, Emre Ozan Polat, Shoubhik Gupta, Hadi Heidari, Mohsen Kaboli, Leandro Lorenzelli, et al. 2015. New materials and advances in making electronic skin for interactive robots. *Advanced Robotics* 29, 21 (2015), 1359–1373. 31

[131] Aaron Young, Nevindu M. Batagoda, Harry Zhang, Akshat Dave, Adithya Pediredla, Dan Negrut, and Ramesh Raskar. 2024. Enhancing Autonomous Navigation by Imaging Hidden Objects using Single-Photon LiDAR. https://doi.org/10.48550/arXiv.2410.03555 arXiv:2410.03555 [cs]. 78

[132] Xiaoyan Yu and Marin Marinov. 2020. A study on recent developments and issues with obstacle detection systems for automated vehicles. *Sustainability* 12, 8 (2020), 3281. 9

[133] Franco Zappa, Simone Tisa, Alberto Tosi, and Sergio Cova. 2007. Principles and features of single-photon avalanche diode arrays. *Sensors and Actuators A: Physical* 140, 1 (2007), 103–112. 3, 11, 35

[134] Jin Zhou and Baoxin Li. 2006. Homography-based ground detection for a mobile robot platform using a single camera. In *ICRA*. IEEE, IEEE, Orlando, USA, 4100–4105. 9

[135] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the Continuity of Rotation Representations in Neural Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Los

Angeles, USA, 5738–5746. https://doi.org/10.1109/CVPR.2019.00589 ISSN: 2575-7075. 87, 96, 97

[136] Yanmin Zhou, Jiangang Zhao, Ping Lu, Zhipeng Wang, and Bin He. 2023. TacSuit: A wearable large-area, bioinspired multi-modal tactile skin for collaborative robots. *Transactions on Industrial Electronics* 71, 2 (2023), 1708–1717. 31

[137] Nicky Zimmerman, Hanna Müller, Michele Magno, and Luca Benini. 2023. Fully Onboard Low-Power Localization with Semantic Sensor Fusion on a Nano-UAV using Floor Plans. (2023). arXiv preprint arXiv:2310.12536. 33

[138] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. 2025. FoundPose: Unseen Object Pose Estimation with Foundation Features. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Vol. 15084. Springer Nature Switzerland, Cham, 163–182. https://doi.org/10.1007/978-3-031-73347-5_10 Series Title: Lecture Notes in Computer Science. 88, 89, 90