The Implicit Values of A Good Hand Shake: Handheld Multi-Frame Neural Depth Refinement

Ilya Chugunov1† Yuxuan Zhang1 Zhihao Xia2 Xuaner Zhang2 Jiawen Chen2 Felix Heide1
1Princeton University 2Adobe

Abstract

Modern smartphones can continuously stream multi-megapixel RGB images at 60 Hz, synchronized with high-quality 3D pose information and low-resolution LiDAR-driven depth estimates. During a snapshot photograph, the natural unsteadiness of the photographer’s hands offers millimeter-scale variation in camera pose, which we can capture along with RGB and depth in a circular buffer. In this work we explore how, from a bundle of these measurements acquired during viewfinding, we can combine dense micro-baseline parallax cues with kilopixel LiDAR depth to distill a high-fidelity depth map. We take a test-time optimization approach and train a coordinate MLP to output photometrically and geometrically consistent depth estimates at the continuous coordinates along the path traced by the photographer’s natural hand shake. With no additional hardware, artificial hand motion, or user interaction beyond the press of a button, our proposed method brings high-resolution depth estimates to point-and-shoot “tabletop” photography – textured objects at close range.

1. Introduction

The cell-phone of the 90s was a phone, the modern cell-phone is a handheld computational imaging platform [9] that is capable of acquiring high-quality images, pose, and depth. Recent years have witnessed explosive advances in passive depth imaging, from single-image methods that leverage large data priors to predict structure directly from image features [40, 41] to efficient multi-view approaches grounded in principles of 3D geometry and epipolar projection [50, 47]. At the same time, progress has been made in the miniaturization and cost-reduction [3] of active depth systems such as LiDAR and correlation time-of-flight sensors [29]. This has culminated in their leap from industrial and automotive applications [45, 11] to the space of mobile phones. Nestled in the intersection of high-resolution imaging and miniaturized LiDAR we find modern smartphones, such as the iPhone 12 Pro, which offer access to high frame-rate, low-resolution depth and high-quality pose estimates.

As applications of mixed reality grow, particularly in industry [30] and healthcare [17] settings, so does the demand for convenient systems to extract 3D information from the world around us. Smartphones fit this niche well, as they boast a wide array of sensors – e.g. cameras, magnetometer, accelerometer, and the aforementioned LiDAR system – while remaining portable and affordable, and consequently ubiquitous. Image, pose, and depth data from mobile phones can drive novel problems in view synthesis [35, 39], portrait relighting [38, 49], and video interpolation [2] that either implicitly or explicitly rely on depth cues, as well as more typical 3D understanding tasks concerning salient object detection [59, 13], segmentation [46], localization [61], and mapping [44, 37].

Although 3D scene information is essential for a wide array of 3D vision applications, today’s mobile phones do not offer accurate high-resolution depth from a single snapshot. While RGB image data is available at more than 100 megapixels (e.g. Samsung ISOCCELL HP1), the most successful depth sensors capture at least three orders of magnitude fewer measurements, with pulsed time-of-flight sensors [36] and modulated correlation time-of-flight imagers [28, 20, 27] offering kilopixel resolutions. Passive approaches can offer higher spatial resolution by exploiting RGB data; however, existing methods relying on stereo [7, 4, 25] depth estimation require large baselines, monocular depth methods [6, 40] suffer from scale ambiguity, and structure-from-motion methods [43] require diverse
poses that are not present in a single snapshot. Accurate high-resolution snapshot depth remains an open challenge.

For imaging tasks, align and merge computational photography approaches have long exploited subtle motion cues during a single snapshot capture. These take advantage of the photographer’s natural hand tremor during viewfiding to capture a sequence of slightly misaligned images, which are fused into one super-resolved image [55, 51]. These misaligned frames can also be seen as mm-baseline stereo pairs, and works such as [57, 24] find that they contain enough parallax information to produce coarse depth estimates. Unfortunately, this micro-baseline depth is not enough to fuel mixed reality applications alone, as it lacks the ability to segment clear object borders or detect cm-scale depth features. In tandem with high-quality poses from phone-based SLAM [12] and low-resolution LiDAR depth maps, however, we can use the high-resolution micro-baseline depth cues to guide the reconstruction of a refined high-resolution depth map. We develop a pipeline for recording LiDAR depth, image, and pose bundles at 60 Hz, with which we can conveniently record 120 frame bundles of measurements during a single snapshot event.

With this hand shake data in hand, we take a test-time optimization approach to distill a high-fidelity depth estimate from hand tremor measurements. Specifically, we learn an implicit neural representation of the scene from a bundle of measurements. Depth represented by a coordinate multilayer perceptron (MLP) allows us to query for depth at floating point coordinates, which matches our measurement model, as we effectively traverse a continuous path of camera coordinates during the movement of the photographer’s hand. We can, during training, likewise conveniently incorporate parallax and LiDAR information as photometric and geometric loss terms, respectively. In this way we search for an accurate depth solution that is consistent with low-resolution LiDAR data, aggregates depth measurements across frames, and matches visual features between camera poses similar to a multi-view stereo approach. Specifically, we make the following contributions

- A smartphone app with a point-and-shoot user interface for easily capturing synchronized RGB, LiDAR depth, and pose bundles in the field.
- An implicit depth estimation approach that aggregates this data bundle into a single high-fidelity depth map.
- Quantitative and qualitative evaluations showing that our depth estimation method outperforms existing single and multi-frame techniques.

The smartphone app, training code, experimental data, and trained models are available at github.com/princeton-computational-imaging/HNDR .

2. Related Work

Active Depth Imaging. Active depth methods prove a scene with a known illumination pattern and use the returned signal to reconstruct depth. Structured light approaches use this illumination to improve local image contrast [58, 42] and simplify the stereo-matching process. Time-of-Flight (ToF) technology instead uses the travel time of the light itself to measure distances. Indirect ToF achieves this through measuring the phase differences in the returned light [28], whereas direct ToF methods time the departure and return of pulses of light via avalanche photodiodes [8] or single-photon detectors (SPADs) [34]. The low spatial-resolution depth stream we use in this work relies on a LiDAR direct ToF sensor. While its mobile implementation comes with the caveats of low-cost SPADs [3] and vertical-cavity surface-emitting lasers [53], with limited spatial resolution and susceptibility to surface reflectance, this sensor can provide robust metric depth estimates, without scale ambiguity, on even visually textureless surfaces. We use this LiDAR depth data to regularize our network’s outputs, avoiding local minima solutions for low-texture regions.

Multi-View Stereo. Multi-view stereo (MVS) algorithms are passive depth estimation methods that infer the 3D shape of a scene from a bundle of RGB views and, optionally, associated camera poses. COLMAP [43] estimates both poses and sparse depths by matching visual features across frames. The apparent motion of each feature in image space is uniquely determined by its depth and camera pose. Thus there exists an important relationship that, for a noiseless system, any pixel movement (disparity) not caused by a change in pose must be caused by a change in depth. While classical approaches typically formulate this as an explicit photometric cost optimization [48, 15, 16], more recent learning-based approaches bend the definition of cost with learned visual features [56, 50, 32], which aid in dense matching as they incorporate non-local information into otherwise textureless regions and are more robust to variations in lighting and noise that distort RGB values. In our setup, with little variation in lighting or appearance and free access to reliable LiDAR-based depth estimates in textureless regions, we look towards photometric MVS to extract parallax information from our images and poses.

Monocular Depth Prediction. Single-image monocular approaches [41, 40, 18] offer visually reasonable depth maps, where foreground and background objects are clearly separated but may not be at a correct scale, with minimal data requirements – just a single image. Video-based methods such as [14, 33, 54] leverage structure-from-motion cues [52] to extract additional information on scene scale and geometry. Works such as [19, 23] use video data with small (decimeter-scale) motion, and [24, 57] explore micro-baseline (mm-scale) motion. As the baseline decreases, the
depth estimation problem gradually devolves from MVS to effectively single-image prediction. Our work resides in the micro-baseline domain, using only millimeters of baseline information, but leverages metric LiDAR depth and pose information to bypass the noisy search for affine depth solutions—the cycle of identifying and matching sparse image features—that previous works were forced to contend with.

3. Neural Micro-baseline Depth

Overview. When capturing a “snapshot photograph” on a modern smartphone, the simple interface hides a significant amount of complexity. The photographer typically composes a shot with the assistance of an electronic viewfinder, holding steady before pressing the shutter. During composition, a modern smartphone streams the recent past, consisting of synchronized RGB, depth, and six degrees of freedom pose (6DoF) frames into a circular buffer at 60 Hz.

In this setting, we make the following observations: (1) A few seconds is sufficient for a typical snapshot of a static object. (2) During composition, the amount of hand shake is small (mm-scale). (3) Under small pose changes viewpoint-dependent lighting effects are minor. (4) Our data shows that current commercial devices have excellent pose estimation, likely due to well-calibrated sensors (IMU, LiDAR, RGB camera) collaborating to solve a smooth low-dimensional problem. Concretely, at each shutter press, we capture a “data bundle” of time-synchronized frames, each consisting of an RGB image \( I_r \), 3D poses \( P_r \), camera intrinsics \( K \), and a depth map \( Z \). In our experiments, the high-resolution RGB is \( 1920 \times 1440 \), while the LiDAR depth is \( 256 \times 192 \)\(^1\). To save memory, we restrict bundles to \( N = 120 \) frames (two seconds) in all our experiments.

Micro-Baseline parallax. We specialize classical multi-view stereo [21] for our small motion scenario. Without loss of generality, we denote the first frame in the bundle the reference \((r)\) and represent all other query \((q)\) poses using the small angle approximation relative to the reference

\[
P_q = \begin{bmatrix} R(r) & \mathbf{t} \end{bmatrix} \approx \begin{bmatrix} 1 & -r^z & r^y & t^x \\ r^z & 1 & -r^x & t^y \\ -r^y & r^x & 1 & t^z \end{bmatrix}.
\]

Let \( X \) the homogeneous coordinates of a 3D point \((x, y, z, 1)^T\), the geometric consistency constraint is

\[
X_r = P_q X_q. 
\]

In other words, the known pose should transform any 3D point in the query frame to its corresponding 3D location in the reference frame. Given camera intrinsics \( K_r, K_q \), with

\[
K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix},
\]

and a point \( X \) perspective projection yields continuous pixel coordinates \( x_r, x_q \) via

\[
x = \pi(KX) = \begin{bmatrix} u = f_x x/z + c_x \\ v = f_y y/z + c_y \end{bmatrix}. 
\]

Using these pixel coordinates to sample from images \( I_r \) and \( I_q \), we arrive at our second constraint

\[
I_r(x_r) = \pi(K_r P_r X_q)) = I_q(x_q). 
\]

Corresponding 3D points should be photometrically consistent: with small motion, they should have the same color in both views. These two constraints are visualized in Fig. 2. To generate a 3D point \( X(x, z) \) we can “unproject” a pixel coordinate \( x \) at depth \( z = Z(x) \) by way of

\[
X(x, z) = \pi^{-1}(x, z; K) = \begin{bmatrix} z(u - c_x)/f_x \\ z(v - c_y)/f_y \\ z \\ 1 \end{bmatrix}.
\]

Our objective is to find a refined depth representation \( Z' \) such that for all \( z \in Z' \) any unprojected point \( X(x) \) best satisfies our geometric (2) and photometric (5) constraints for all query views.

Implicit Depth Representation. There are numerous ways with which we can represent \( Z' \). For example, we can represent it explicitly with a discrete depth map from the reference view, or as a large 3D point cloud. While explicit representations have many advantages (fast data retrieval, existing processing tools), they are also challenging to optimize. Depth maps are discrete arrays and merging multiple

\[ \text{Figure 2. Visualization of how we project points } x_r, x_q \text{ with corresponding camera poses } P_r, P_q \text{ to 3D space and bilinearly sample image points } I_r(x_r) \text{ and } I_q(x_q). \text{ Note that the pose change here is enlarged for ease of illustration, the real misalignment in views from hand shake is on the scale of millimeters.} \]
views at continuous coordinates requires resampling. This blurs fine-scale information and is non-trivial at occlusion boundaries. Point cloud representations trade off this adaptive filtering problem with one of scale. Not only is a two second sequence with 120 million points unwieldy for conventional tools \[1\], the points are almost entirely redundant.

Thus, we choose an implicit depth representation in the form of a coordinate multi-layer perceptron (MLP) \[22\], where its learnable parameters automatically adapt to the scene structure. Recent work have used MLPs to great success in neural rendering \[35, 5, 39\] and depth-estimation, where a continuous representation is of interest \[60\]. In our application, the MLP is a differentiable function

\[
    z' = Z'(x) = f(\text{inputs}; \theta)
\]

returning a continuous \(z'\) given an encoding of position, camera pose, color, and other features. In our implementation, inputs is a positionally encoded 3D colored point

\[
    \hat{X} = [\hat{x}, \gamma(y), \gamma(z), r, g, b]^{\top}.
\]

We follow the encoding of \[35\] with

\[
    \gamma(p) = \left[\sin(2^0 \pi p), \cos(2^0 \pi p), \ldots, \cos(2^{L-1} \pi p)\right],
\]

where \(L\) is a selected number of encoding functions, and \(r, g, b\) are color values scaled to \([0, 1]\). As the size of this MLP is fixed, the large dimensionality of our measurements does not affect the calculation of \(z'\), and instead becomes a large training dataset from which to sample. Translating (2) and (5) into a regularized loss function on \(z'\), and backpropagating through \(f\), our implicit depth representation \(\theta\) can be learned with stochastic gradient descent.

### Backward-Forward Projection Model

Fig. 3 illustrates how we combine geometric and photometric constraints to optimize our MLP to produce a refined depth. At each training step we sample a query view \((I_q, P_q, Z_q)\) and generate \(M\) randomly sampled colored points \(X_q\) via Eq. 6

\[
    X_q = \left[\begin{array}{c} x, y, z, 1 \\ r, g, b \end{array}\right]^{\top} = \left[\begin{array}{c} X_q(x, z = Z_q(x)) \\ I_q(x) \end{array}\right].
\]

\(x = [u, v]^{\top}, \quad u \sim \mathcal{U}(0, W), \quad v \sim \mathcal{U}(0, H)\), \hspace{1cm} (10)

where \(H\) and \(W\) are the image height and width, respectively. Here, \(x\) is a continuous coordinate and \(I(x), Z(x)\) represent sampling with a bilinear kernel. Following (2) we transform these points to the reference frame as

\[
    X_{q \to r} = \left[\begin{array}{c} \hat{x}, \hat{y}, \hat{z}, 1 \\ I_q(x) \end{array}\right] = \left[\begin{array}{c} P_q X_q(x, z) \\ I_q(x) \end{array}\right].
\]

\(\hat{x}, \hat{y}, \hat{z} + \Delta z\)

Then, rather than directly predicting a refined depth \(z'\), we ask our MLP to predict a depth correction \(\Delta z\), that is

\[
    \Delta z = f(\hat{X}_{q \to r}) \hspace{1cm} (12)
\]

\[X_f(x, z') = X_f(x, \hat{z} + \Delta z) = [\hat{x}, \hat{y}, \hat{z} + \Delta z, 1]^{\top}.
\]

As we show in Section 5, this parameterization allows us to avoid local minima in poorly textured regions. We transform these refined points \(X_f\) back to the query frame and resample the query image at the updated coordinates

\[
    I_q(x_f) = I_q(\pi(P_q^{-1} X_f(x, \hat{z} + \Delta z))).
\]

Finally, our photometric loss is

\[
    \mathcal{L}_P = |I_q(x_f) - I_r(x_r)|^2 \hspace{1cm} (13)
\]

\[
    x_r = \pi(\hat{K}_r P_q X_q(x, z)),
\]

which attempts to satisfy (5) by encouraging the colors of the refined 3D points to be the same in both the query and

---

**Figure 3.** An illustrated pipeline of our proposed model. The query depth is used to project and sample a patch from our reference image \(I_r\) for input into the MLP. This is weighed by a sample patch from our confidence \(C_r\) to produce a depth offset \(C_r(x_r) \Delta z\), which is used to project back to our query image \(I_q\) and sample an image patch for loss calculation.
reference frames. While (5) works well in well-textured areas, it is fundamentally underconstrained in flat regions. Therefore, we augment our loss with a weighted geometric regularizer term based on (2) that pushes the solution towards an interpolation of LiDAR depth in these regions

$$R_G = |X_f(x, z + Δz) - P_yX_g(x, z)| ≈ |Δz|.$$ (15)

Our final loss is a weighted combination of these two terms

$$\mathcal{L} = \mathcal{L}_P + \alpha R_G,$$ (16)

where by tuning $\alpha$ we adjust how strongly our reconstruction adheres to the LiDAR depth initialization.

**Patch Sampling.** In practice, we cannot rely on single-pixel samples as written in (6) for photometric optimization. Our two megapixel input images $I$ will almost certainly contain color patches that are larger than the depth-induced motion of pixels within them. With single-pixel samples, there are many incorrect depth solutions that yield $\mathcal{L}_P$ of zero. To combat this, we replace each sample $I(x)$ in (10) with Gaussian-weighted patches

$$G(I(x)) = \left[ N(\sqrt{\delta_u^2 + \delta_v^2}, \mu, \sigma^2)I(x - [\delta_u, \delta_v]^\top) \right],$$

for $\delta_u = \{-K \ldots K\}$, $\delta_v = \{-K \ldots K\}.$ (17)

Fig. 4 illustrates this for $K = 3$: the increased receptive field discourages false color matches. Adjusting $K$, we trade off the ability to reconstruct fine features for robustness to noise and low-contrast textures (see supplement).

**Explicit Confidence.** Another augmentation we make is to introduce a learned explicit $H \times W$ confidence map $C_r$ to weigh the MLP outputs. That is, we replace $Δz$ with $C_r(x, r)Δz$ in (12). This additional degree of freedom allows the network push $Δz$ toward zero in color-ambiguous regions, rather than forcing it to first learn a positional mapping of where these regions are located in the image. As $C_r(x, r)$ only adds an additional sampling step during point generation, the overhead is minimal. Once per epoch we apply an optional $5 \times 5$ median filter to $C_r$ to minimize the effects of sampling noise during training.

**Final Reconstruction.** After training, to recover a refined depth map $Z^*$ we begin by reprojecting all low-resolution depth maps $Z_0$ to the reference frame following (2). We then average and bilinearly resample this data to produce a $H \times W$ depth map $Z_{avg}$. We query the MLP at $H \times W$ grid-spaced points, using $I_r$ and $Z_{avg}$ to generate points as in (10). Finally, we extract and re-grid the depth channel from the MLP outputs $R_f$ to produce $Z^*$.

### 4. Data Collection

**Recording a Bundle.** We built a smartphone application for recording bundles of synchronized image, pose, and depth maps. Our app, running on an iPhone 12 Pro using ARKit 5, provides a real-time viewfinder with previews of RGB and depth (Fig. 5). The user can select bundle sizes of $[15, 30, 60, 120]$ frames ($[0.25, 0.5, 1, 2]$ seconds of recording time) and we save all data, including nanosecond-precision timestamps to disk. We will publish the code for both the app and our offline processing pipeline (which does color-space conversion, coordinate space transforms, etc.).

**Natural Hand Tremor Analysis.** To analyze hand motion during composition, we collected fifty 2-second pose-only bundles from 10 volunteers. Each was instructed to act as if they were capturing photos of objects around them, to hold the phone naturally in their dominant hand, and to keep focus on an object in the viewfinder. We illustrate our aggregate findings in Fig. 6 and individual measurements in the supplemental material. We focus on in-plane displacement—they are the dominant contribution to observed parallax. We find that natural hand tremor appears similar to paths traced by 2D Brownian motion, with some paths traveling far from the initial camera position as in Fig. 6 (b), and others forming circles around the initial position.
sequently, while the median effective baseline from a two second sequence is just under 6mm, some recordings exhibit nearly 1cm of displacement, while others appear almost static. We suspect that the effects of breathing and involuntary muscle twitches are greatly responsible for this variance. Herein lies the definition of good hand shake: it is one that produces a useful micro-baseline. Fig. 7 (c) illustrates, given our smartphone’s optics, what a 6mm baseline translates to in pixel disparity and therefore the depth feature precision. We intentionally limit ourselves to a depth range of approximately 50 cm, beyond which image noise and errors in pose estimation overpower our ability to estimate subpixel displacement.

5. Assessment

Implementation Details. We use $N = 120$ frame bundles for our main experiments. Images are recorded in portrait orientation, with $H = 1920$, $W = 1440$. Our MLP is a 4 layer fully connected network with ReLU activations and a hidden layer size of 256. For training we use inputs of $M = 4096$ colored points, a kernel size of $K = 11$, and $L = 6$ encoding functions. We use the Adam optimizer [26] with an initial learning rate of $10^{-5}$, exponentially decreased over 200 epochs with a decay rate of 0.985. We apply the geometric regularization $R_C$ with weight $\alpha = 0.01$. We provide ablation experiments on the effects of many of these parameters in the supplement. Training takes about 45 minutes on an NVIDIA Tesla P100 GPU, or 180 minutes on an Intel Xeon Gold 6148 CPU.

Comparisons. We compare our method (Proposed) to the input LiDAR data and several recent baselines. Namely, we reproject all the depth maps in our bundle to the reference frame and resample them to produce a $1920 \times 1440$ LiDAR baseline $Z_{avg}$. For the depth reconstruction methods we look to Consistent Video Depth Estimation [33] (CVD), which similarly uses photometric loss between frames in a video to refine a consistent depth; Depth Supervised NeRF [10] (DSNeRF), which also features an MLP for depth prediction; and Structure from Small Motion [19] (SfSM), which investigates a closed-form solution to depth estimation from micro-baseline stereo. Both DSNeRF and CVD rely on COLMAP [43] for poses or depth inputs. However, when input our micro-baseline data, COLMAP fails to converge and returns neither. For fair comparison, we substitute our high-accuracy LiDAR poses and depths.

Experimental Results. We present our results visually in Fig. 8 and quantitatively in Table 1 in the form of photometric error (PE). To compute PE, we take the final depth map $Z^*$ output by each method, use the phone’s poses and intrinsics to project each color point in $I_r$ to all other frames, and compare their sampled RGB values:

$$PE = \sum | I_q(x_q) - I_r(x)|, \quad x_q = \pi(P_q^{-1}X^*)$$

$$X^* = \pi^{-1}(x, Z^*(x); K), \quad x = [u, v]^T. \quad (18)$$

We exclude points $x$ that transform outside the image bounds. Similar to traditional camera calibration or stereo methods, in the absence of ground truth depth, PE serves as a measure of how consistent our estimated depth is with the observed RGB parallax. Table 1 summarizes the relative performance between these methods on 10 geometrically diverse scenes. Our method achieves the lowest PE for all scenes. Note that neither CVD nor DSNeRF achieve significantly lower PE as compared to the LiDAR depth $Z_{avg}$ even though both contain explicit photometric loss terms in their objective. We speculate that our micro-baseline data is out of distribution for these methods, and that the large loss gradients induced by small changes in pose results in unstable reconstructions. DSNeRF also has the added complexity of being a novel view synthesis method and is therefore encouraged to overfit to the scene RGB content in the presence of only small motion. We see this confirmed in Fig. 8, as DSNeRF produces an edge-aligned depth map but incorrect surface texture. SfSM successfully reconstructs textured regions close to the camera (<20cm), but fails for smaller disparity regions, textureless spaces, and parts of the image suffering from lens blur. The reprojected LiDAR depth produces well edge-aligned results but lacks intra-

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>castle</td>
<td>4.88/62.9</td>
<td>4.89/87.3</td>
<td>4.70/55.4</td>
<td>4.51/48.1</td>
<td>3.83/30.0</td>
</tr>
<tr>
<td>thinker</td>
<td>4.73/115</td>
<td>4.59/109</td>
<td>5.78/170</td>
<td>4.50/107</td>
<td>4.26/89.4</td>
</tr>
<tr>
<td>rocks</td>
<td>6.68/212</td>
<td>5.83/180</td>
<td>6.43/190</td>
<td>5.52/163</td>
<td>4.59/97.3</td>
</tr>
<tr>
<td>ganesha</td>
<td>4.93/56.5</td>
<td>5.29/103</td>
<td>4.45/41.3</td>
<td>4.41/36.8</td>
<td>3.51/23.4</td>
</tr>
<tr>
<td>ground</td>
<td>4.25/92.6</td>
<td>4.00/108</td>
<td>3.72/70.7</td>
<td>3.77/92.8</td>
<td>3.14/52.9</td>
</tr>
<tr>
<td>eagle</td>
<td>3.93/63.1</td>
<td>4.09/93.4</td>
<td>6.60/219</td>
<td>3.98/33.2</td>
<td>3.32/49.0</td>
</tr>
<tr>
<td>double</td>
<td>4.77/68.6</td>
<td>4.87/91.3</td>
<td>4.30/39.1</td>
<td>4.00/31.6</td>
<td>3.57/22.1</td>
</tr>
<tr>
<td>elephant</td>
<td>5.12/101</td>
<td>5.20/124</td>
<td>5.49/120</td>
<td>5.09/127</td>
<td>4.46/78.8</td>
</tr>
<tr>
<td>embrace</td>
<td>4.92/54.5</td>
<td>5.13/88.7</td>
<td>3.81/260</td>
<td>3.27/23.1</td>
<td>2.85/15.5</td>
</tr>
<tr>
<td>frog</td>
<td>3.44/68.2</td>
<td>3.11/58.3</td>
<td>3.05/51.5</td>
<td>2.80/49.9</td>
<td>2.58/34.1</td>
</tr>
</tbody>
</table>
Figure 8. Qualitative comparison of depth reconstruction methods for tabletop scenes. Normals are shown for $Z_{\text{avg}}$ and our proposed method to highlight how we recover centimeter-scale features absent from the input depths. See the supplement for additional results.

Figure 9. Frame count ablation with corresponding network train times. Note that due to various overheads, training time is not linear in the number of frames.

object structure, as it is relying on ambiguous mono-depth cues. Contrastingly, our proposed method reconstructs the castle’s towers, the hand under the thinker’s head, the depth disparity between stones in rocks, the trunk and arms of ganesha, and the smooth undulations of gourd. For visually complex scenes such as ganesha our method is able to cleanly separate the foreground object from its background.

Frame Ablation. Though the average max baseline in a recorded bundle is 6mm, the baseline between neighboring frames is on the order of 0.1mm. This means that we need not use every frame for effective photometric refinement. This tradeoff between frame count, training time, and reconstruction detail is illustrated in Fig. 9. We retain most of the castle detail by skipping every other frame, and as we discard more data we see the reconstruction gradually lose fine edges and intra-object features.

Role of LiDAR Supervision. To determine the contribution of the low-resolution LiDAR data in our pipeline, we perform an ablation where we set $Z = 1m$ and disable our geometric regularizer by setting $\alpha = 0$. Fig. 10 shows that, as expected, it fails to reconstruct the textureless background. It does, however, correctly reconstruct the gourd, and produces a result similar to the pipeline with full supervision. This demonstrates that our method extracts most
of its depth details from the micro-baseline RGB images. LiDAR depth, however, is an effective regularizer: in regions without strong visual features, our learned confidence map $C_r$ is nearly zero and our reconstruction gracefully degrades to the LiDAR data. Finally, we note that even though this ablation discards depth supervision, the LiDAR sensor is still used by the phone to produce high-accuracy poses.

**Comparison to Dedicated ToF Camera.** We record several scenes with a high-resolution time-of-flight depth camera (LucidVision Helios Flex). Given the differences in optics and effective operating range, the viewpoints and metric depths do not perfectly match. We offset and crop (but don’t rescale) the depth maps for qualitative comparison. Fig. 11 validates that our technique can reconstruct centimeter-scale features matching that of the ToF sensor, but smooths finer details corresponding to subpixel disparities. Since we rely on passive RGB rather than direct illumination, our technique can reconstruct regions the ToF camera cannot such as specular surfaces on the back of frog and the top of gourd. In these cases, multi-path interference leads to incorrect amplitude modulated ToF measurements.

**Offsets over Direct Depth.** Rather than directly learn $Z^*$, we opt to learn offsets to the collected LiDAR depth data. We effectively start at a coarse, albeit smooth, depth estimate and for each location in space, and search the local neighborhood for a more photometrically consistent depth solution. This allows us to avoid local minima solutions that overpower the regularization $\alpha R_G$ – e.g. accidentally matching similar image patches. This proves essential for objects with repetitive textures, as demonstrated in Fig. 12. Additional experiments can be found in the supplement.

**6. Discussion and Future Work**

We show that with a modern smartphone, it is possible to reconstruct a high-fidelity depth map from just a snapshot of a textured “tabletop” object. We quantitatively validate that our technique outperforms several recent baselines and qualitatively compare to a dedicated depth camera.

**Rolling Shutter.** There is a delay in the tens of milliseconds between when we record the first and last row of pixels from the camera sensor [31], during which time the position of the phone could slightly shift. Given accurate shutter timings, one may incorporate a model of rolling shutter similar to [23] directly into the implicit depth model.

**Training Time.** Although our training time is practical for offline processing and opens the potential for the easy collection of a large-scale training corpus, our method may be further accelerated with an adaptive sampling scheme which takes into account pose, color, and depth information to select the most useful samples for network training.

**Additional Sensors.** We hope in the future to get access to raw phone LiDAR samples, whose photon time tags could provide an additional sparse high-trust supervision signal. Modern phones now also come with multiple cameras with different focal properties. If synchronously acquired, their video streams could expand the overall effective baseline of our setup and provide additional geometric information for depth reconstruction – towards snapshot smartphone depth imaging that exploits all available sensor modalities.

**Acknowledgments.** Ilya Chugunov was supported by an NSF Graduate Research Fellowship. Felix Heide was supported by an NSF CAREER Award (2047359), a Sony Young Faculty Award, and a Project X Innovation Award.
References


[23] Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo, and In So Kweon. Accurate 3d reconstruction from small motion clip for rolling shutter cameras. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):775–787, 2018. 2, 8


[28] Robert Lange. 3d time-of-flight distance measurement with custom solid-state image sensors in cmos/ccd-technology. 2000. 1, 2


