Shakes on a Plane: Unsupervised Depth Estimation from Unstabilized Photography

Ilya Chugunov  Yuxuan Zhang  Felix Heide
Princeton University

Abstract

Modern mobile burst photography pipelines capture and merge a short sequence of frames to recover an enhanced image, but often disregard the 3D nature of the scene they capture, treating pixel motion between images as a 2D aggregation problem. We show that in a “long-burst”, forty-two 12-megapixel RAW frames captured in a two-second sequence, there is enough parallax information from natural hand tremor alone to recover high-quality scene depth. To this end, we devise a test-time optimization approach that fits a neural RGB-D representation to long-burst data and simultaneously estimates scene depth and camera motion. Our plane plus depth model is trained end-to-end, and performs coarse-to-fine refinement by controlling which multi-resolution volume features the network has access to at what time during training. We validate the method experimentally, and demonstrate geometrically accurate depth reconstructions with no additional hardware or separate data pre-processing and pose-estimation steps.

1. Introduction

Over the last century we saw not only the rise and fall in popularity of film and DSLR photography, but of standalone cameras themselves. We’ve moved into an era of ubiquitous multi-sensor, multi-core, multi-use, mobile-imaging platforms [12]. Modern cellphones offer double-digit megapixel image streams at high framerates; optical image stabilization; on-board motion measurement devices such as accelerometers, gyroscopes, and magnetometers; and, most recently, integrated active depth sensors [43]. This latest addition speaks to a parallel boom in the field of depth imaging and 3D reconstruction [22, 84]. As users often photograph people, plants, food items, and other complex 3D shapes, depth can play a key role in object understanding tasks such as detection, segmentation, and tracking [32, 63, 80]. 3D information can also help compensate for non-ideal camera hardware and imaging settings through scene relighting [20, 55, 79], simulated depth-of-field effects [1, 71, 72], and frame interpolation [2]. Beyond helping improve or understand RGB content, depth itself is a valuable output for simulating objects in augmented reality [5, 13, 44, 64] and interactive experiences [26, 36].

Depth reconstruction can be broadly divided into passive and active approaches. Passive monocular depth estimation methods leverage training data to learn shape priors [6, 30, 59] – e.g., what image features imply curved versus flat objects or occluding versus occluded structures – but have a hard time generalizing to out-of-distribution scenes [48, 60]. Multi-view depth estimation methods lower this dependence on learned priors by leveraging parallax information from camera motion [16, 69] or multiple cameras [45, 67] to recover geometrically-guided depth. The recent explosion in neural radiance field approaches [49, 50, 66, 81] can be seen a branch of multi-view stereo where a system of explicit geometric constraints is swapped for a more general learned scene model. Rather than classic feature extraction and matching, these models are fit directly to image data to distill dense implicit 3D information.
Active depth methods such as pulsed time-of-flight [46] (e.g., LiDAR), correlation time-of-flight [38], and structured light [61, 83] use controlled illumination to help with depth reconstruction. While these methods are less reliant on image content than passive ones, they also come with complex circuitry and increased power demands [28]. Thus, miniaturization for mobile applications results in very low-resolution sub-kilopixel sensors [8, 27, 74]. The Apple iPhone 12-14 Pro devices, which feature one of these miniaturized sensors, use depth derived from RGB, available at 12 mega-pixel resolution, to recover scene details lost in the sparse LiDAR measurements. While how exactly they use the RGB stream is unknown, occluding camera sensors reveals that the estimated geometry is the result of monocular RGB-guided depth reconstruction.

Returning to the context of mobile imaging, even several seconds of continuous mode photography, which we refer to as a “long-burst”, contain only millimeter-scale view variation from natural hand tremor [11]. While these micro-baseline [33] shifts are effectively used in burst superresolution and denoising methods [58, 76] as indirect observations of content between sensor pixels, 3D parallax effects on pixel motion are commonly ignored in these models as the depth recovered from this data is too coarse for sub-pixel refinement [31, 33, 82]. A recent work [11] demonstrates high-quality object reconstructions refined with long-burst RGB data, but relies on the iPhone 12 Pro LiDAR sensor for initial depth estimates and device poses, not available on many other cellphones. They treat these poses as ground truth and explicitly solve for depth through minimization of photometric reprojection loss.

In this work, we devise an unsupervised end-to-end approach to jointly estimate high-quality object depth and camera motion from more easily attainable unstabilized two-second captures of 12-megapixel RAW frames and gyroscope data. Our method requires no depth initialization or pose inputs, only a long-burst. We formulate the problem as an image synthesis task, similar to neural radiance methods [50], decomposed into explicit geometric projection through continuous depth and pose models. In contrast to recent neural radiance methods, which typically estimate poses in a pre-processing step, we jointly distill relative depth and pose estimates as a product of simply fitting our model to long-burst data and minimizing photometric loss. In summary, we make the following contributions:

- An end-to-end neural RGB-D scene fitting approach that distills high-fidelity affine depth and camera pose estimates from unstabilized long-burst photography.
- A smartphone data collection application to capture RAW images, camera intrinsics, and gyroscope data for our method, as well as processed RGB frames, low-resolution depth maps, and other camera metadata.
- Evaluations which demonstrate that our approach outperforms existing single and multi-frame image-only depth estimation approaches, with comparisons to high-precision structured light scans to validate the accuracy of our reconstructed object geometries.

Code, data, videos, and additional materials are available on our project website: https://light.princeton.edu/soap

2. Related Work

There exist a wide array of both active and passive depth estimation methods, ones that recover depth with the help of a controlled illumination source, and ones that use only naturally collected light. We review related work in both categories before discussing neural scene representations.

Active Depth Reconstruction. Structured light and active stereo method rely on patterned illumination to directly infer object shape [15, 83] and/or improve stereo feature matching [61]. In contrast, time-of-flight (ToF) sensors use the round trip time of photons themselves – how long it takes light to reach and return from an object – to infer depth. Indirect ToF does this by calculating phase changes in continuously modulated light [23, 35, 38], and direct ToF times how long a pulse of light is in flight to estimate depth [46, 52]. The LiDAR system found in the iPhone 12-14 Pro devices is a type of direct ToF sensor built on low-cost single-photon detectors [8] and solid-state vertical-cavity surface-emitting laser technology [74]. While active LiDAR depth measurements can help produce metric depth estimates, without scale ambiguity, existing mobile depth sensors have very limited sub-kilopixel spatial resolution, are sensitive to surface reflectance, and are not commonly found on other mobile devices.

Passive Depth Reconstruction. Single-image passive methods leverage the correlation between visual and geometric features to estimate 3D structure. Examples include depth from shading [3, 77], focus cues [78], or generic learned priors [6, 30, 59]. Learned methods have shown great success in producing visually coherent results, but rely heavily on labeled training data and produce unpredictable outputs for out-of-distribution samples. Multi-view and structure from motion works leverage epipolar geometry [24], the relationship between camera and image motion, to extract 3D information from multiple images. Methods typically either directly match RGB features [17, 65], or higher-level learned features [42, 67], in search of depth and/or camera parameters which maximize photometric consistency between frames. COLMAP [62] is a widely adopted multi-view method which many neural radiance works [49, 50] rely on for camera pose estimates. In the case of long-burst photography, this problem becomes significantly more challenging as many different...
depth solutions produce identical images under small view variations. Work in this space either relies on interpolation between sparse feature matches [31, 33, 82] or additional hardware [11] to produce complete depth estimates. Our work builds on these methods to produce both dense depth and high-accuracy camera motion estimates from long-burst image data alone, with a single model trained end-to-end rather than a sequence of disjoint data processing steps.

**Neural Scene Representations.** Recent work in the area of novel view synthesis has demonstrated that explicit models – e.g. voxel grids, point clouds, or depth maps – are not a necessary backbone to generate high-fidelity representations of 3D space. Rather, the neural radiance family of works, including NeRF [50] and its extensions [4, 10, 53], learn an implicit representation of a 3D scene by fitting a multi-layer perceptron (an MLP) [29] to a set of input images through gradient descent. Similar to multi-view stereo, these methods optimize for photometric loss, ensuring output colors match the underlying RGB data, but they typically don’t produce depth maps or camera poses as outputs. On the contrary, most neural radiance methods require camera poses as inputs obtained in a separate pre-processing step from COLMAP [62]. Our setting of long-burst unstructured photography not only lacks ground truth camera poses, but also provides very little view variation from which to estimate them. While neural scene representation works exist which learn camera poses [41, 73], or operate in the burst photography setting [56], to our best knowledge this is the first work to jointly do both. The most similar recent work by Chugunov et al. [11] uses poses derived from the iPhone 12 Pro ARKit library to learn an implicit representation of depth, but does not have an image generation model, and is functionally closer to a direct multi-view stereo approach. In contrast, our work uses a neural representation of RGB as an optimization vehicle to distill high quality continuous representations of both depth and camera poses, with loss backpropogated through an explicit 3D projection model.

3. Long-Burst Photography

**Problem Setting.** Burst photography refers to the imaging setting where for each button press from the user the camera records multiple frames in rapid succession, sometimes varying parameters such as ISO and exposure time during capture to create a bracketed sequence [47]. Burst imaging pipelines investigate how these frames can be merged back into a single higher-fidelity image [12]. These pipelines typically operate with 2-8 frame captures and have proven key to high-quality mobile imaging in low-light [25, 40], high dynamic range imaging with low dynamic range sensors [18, 25], and image superresolution, demosaicing, and denoising [75, 76]. On the other end of the imaging spectrum we have video processing literature, which operates on sequences hundreds or thousands of frames in length [51] and/or large camera motion [37]. Between these two settings we have what we refer to as “long-burst” photography, several seconds of continuous capture with small view variation. Features built into default mobile camera applications such as Android Motion Photos and Apple Live Photos, which both record three seconds of frames around a button press, demonstrate the ubiquity of long-burst data, as they are captured spontaneously without user interaction during natural handheld photography. In this work we capture two-second long-bursts, which result in 42 recorded frames with an average 6mm maximum effective stereo baseline. This produces on the order of several dozen pixels of disparity for close-range objects (<0.5m), see Fig. 2 (b). For an in-depth discussion of motion from natural hand tremor we refer the reader to Chugunov et al. [11].

**Data Collection.** As there were no commodity mobile applications that allowed for continuous streaming of Bayer RAW frames and metadata, we designed our own data collection tool for long-burst recording. Shown in Fig. 2 (a), it features a live viewfinder with previews of RGB, device depth, and auto-adjusted ISO and exposure values. On a button press, we lock ISO, exposure, and focus, and record a two-second, 42 frame long-burst to the device. Our method uses recorded timestamps, camera intrinsics, gyroscope-driven device rotation estimates, and 12-megapixel RAW frames. However, our app also records processed RGB frames, low-resolution depth maps, and other metadata which we use for validation and visualization.

**RAW Images.** A modern mobile image signal processing pipeline can have more than a dozen steps between light

---

1RAW here refers to sensor data after basic corrections such as compensating for broken and non-uniform pixels, not “raw-raw” data [57].
hitting the CMOS sensor and a photo appearing on screen: denoising, demosaicing, and gamma correction to name a few [12]. While these steps, when finely-tuned, can produce eye-pleasing results, they also pose a problem to downstream computer vision tasks as they break linear noise assumptions, correlate pixel neighborhoods, and lower the overall dynamic range of the content (quantizing the 10- to 14-bit sensor measurements down to 8-bit color depth images) [7]. In our work we are concerned with the tracking and reconstruction of small image features undergoing small continuous motion from natural hand tremor, and so we apply minimal processing to our image data, using linear interpolation to only fill the gaps between Bayer measurements. We preserve the full 14-bit color depth and fit our depth plus image model directly to this 4032px×3024px×3 channel×42 frame volume.

4. Unsupervised Depth Estimation

In this section, we propose a method for depth estimation from long-burst data. We first lay out the projection model our method relies on, before introducing the scene model, loss functions, and training procedure used to optimize it.

**Projection Model.** Given an image stack \(I(u, v, N)\), where \(u, v \in [0, 1]\) are continuous image coordinates and \(N \in [0, 1, \ldots, 41]\) is the frame number, we aim to condense the information in \(I(u, v, N)\) to a single compact projection model. Given that the motion between frames is small, and image content is largely overlapping, we opt for an RGB-D representation which models each frame of \(I(u, v, N)\) as the deformation of some reference image \(I(u, v)\) projected through depth \(D(u, v)\) with a change in camera pose \(P(N)\). We expand this process for a single point at coordinates \(u, v\) in the reference frame. Let

\[
C = [R, G, B]^\top = I(u, v), \quad d = D(u, v) \tag{1}
\]

be a sampled colored point \(C\) at depth \(d\). Before we can project this point to new frame, we must first convert it from camera \((u, v)\) to world \((x, y, z)\) coordinates. We assume a pinhole camera model to un-project this point via

\[
\begin{bmatrix}
x
y
z
1
\end{bmatrix}^T = \pi^{-1}\left(\begin{bmatrix} u \\ v \\ d \end{bmatrix}; K\right) = \frac{d(u - cx)/fx}{d(v - cy)/fy} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{2}
\]

where \(K\) are the corresponding camera intrinsics with focal point \((f_x, f_y)\) and principal point \((c_x, c_y)\). We transform this point from the reference frame to target frame \(N\), with camera pose \(P(N)\), via

\[
\begin{bmatrix}
x^N
y^N
z^N
1
\end{bmatrix} = [R(N) | T(N)] \begin{bmatrix}
x
y
z
1
\end{bmatrix} = [P(N)] \begin{bmatrix}
x
y
z
1
\end{bmatrix}. \tag{3}
\]

Here, \(P(N)\) is decomposed into a \(3 \times 3\) rotation matrix \(R(N)\) and \(3 \times 1\) translation vector \(T(N) = [tx, ty, tz]^\top\). Reverse of the process in Eq. (2), we now project this point from the world coordinates \((x^N, y^N, z^N)\) in frame \(N\) to camera coordinates \((u^N, v^N)\) in the same frame as

\[
[u^N, v^N] = K^{-1} \begin{bmatrix} x^N \\ y^N \\ z^N \end{bmatrix}, \quad K(N) = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}
\]

where \(K(N)\) are the frame intrinsics. We can now use these coordinates to sample a point from the full image stack

\[
C^N = I(u^N, v^N, N), \quad \mathcal{L}_{\text{photo}} = |C - C^N|. \tag{5}
\]

Here \(\mathcal{L}_{\text{photo}}\) is photometric loss, the difference in color between the point we started with in the reference frame and what we sampled from frame \(N\). Given ideal multi-view imaging conditions — no occlusions, imaging noise, or changes in scene lighting — if depth \(d\) and pose change \(P(N)\) are correct, we will incur no photometric loss \(\mathcal{L}_{\text{photo}} = 0\) as we sample matching points in both frames. This is visualized in Fig. 3. Inverting this observation, we can solve for unknown \(D(u, v)\) and \(P(N)\) by finding ones that minimize photometric loss [62].

**Implicit Image Model.** In our problem setting, we are given a long-burst image stack \(I(u, v, N)\) and device rotation values \(R(N)\), supplied by an on-board gyroscope, and are tasked with recovering depth \(D(u, v)\) and translation \(T(N)\) which make these observations consistent. Given the sheer number of pixels in \(I(u, v, N)\), in our case about 500 million, exhaustively matching and minimizing pixel-to-pixel loss is both computationally intractable and ill-posed. Under small camera motion, many depth solutions for a pixel can map it to identical-colored pixels in the image, especially in textureless parts of the scene. Traditional multi-view stereo (MVS) and bundle adjustment methods tackle this problem with feature extraction and matching [68], optimizing over a cost-volume orders of magnitude smaller than the full image space. Here we strongly diverge from...
previous small motion works [11, 31, 82]. Rather than divide the problem into feature extraction and matching, or extract features at all, we propose a single fully differentiable forward model trained end-to-end. Depth is distilled as a product of fitting this neural scene model to long-burst data. We start by redefining $I(u, v)$ from a static reference image to a learned implicit representation

$$I(u, v) = f_i(\gamma_i(u, v; \text{params}_{\gamma_i}); \theta_i)$$

where $f_i$ is a multi-layer perceptron (MLP) [29] with learned weights $\theta$. This MLP learns a mapping from $\gamma_i(u, v)$, a positional encoding of sampled camera coordinates, to image color. Specifically, we borrow the multiresolution hash encoding from Müller et al. [53] for its spatial aggregation properties. The parameters in $\text{params}_{\gamma_i}$ determine the minimum $N_{\min}^{\gamma_i}$ and maximum $N_{\max}^{\gamma_i}$ grid resolutions, number of grid levels $L^{\gamma_i}$, number of feature dimensions per level $F^{\gamma_i}$, and overall hash table size $T^{\gamma_i}$.

### Implicit Depth on a Plane Model.

Our depth model is a similar implicit representation with a learned planar offset

$$d = D(u, v) = D_p(u, v) + f_0(\gamma_0(u, v; \text{params}_{\gamma_0}); \theta_0)^+$$

where $\{a, b, c\}$ are the learned plane coefficients, and $^+$ is the ReLU operation $\max(0, x)$. Here $D_p(u, v)$ acts as the depth of the scene background – the surface on or in front of which objects are placed – which is often devoid of parallax cues. Then $f_0$ reconstructs the depth of the scene foreground content recovered from parallax in $I(u, v, N)$. While it may seem that we are increasing the complexity of the problem, as we now have to learn $I(u, v)$ in addition to $D(u, v)$, this model actually simplifies the learning task when compared to a static $I(u, v)$. Rather than solving for a perfect image from the get-go, $f_i$ can move between intermediate representations of the scene with blurry, noisy, and misaligned content, and is gradually refined during training.

### Camera Motion Model.

Given the continuous, smooth, low-velocity motion observed in natural hand tremor [11], we opt for a low-parameter Bézier curve motion model

$$T(N) = B(N; P^T, N^T), \quad R(N) = R_d(N) + \eta_k B(N; P^R, N^R)$$

with $N_c$ number of control points $P$. Translation estimates $T(N)$ are learned from scratch, whereas rotations $R(N)$ are initialized as device values $R_d(N)$ with learned offsets weighted by $\eta_k$. Under the small angle approximation [31], we parameterize the rotational offsets $P^R$ as

$$P^R_i = \begin{bmatrix} 0 & -r_z & r_y \\ -r_z & 0 & -r_x \\ r_y & r_x & 0 \end{bmatrix}.$$  

The choice of $N_c$ controls the dimensionality of the curve on which motion lies – e.g. $N_c = 1$ restricts motion to be linear, $N_c = 2$ is quadratic, and $N_c = 42$ trivially overfits the data with a control point for each frame.

### Loss and Regularization.

Putting all of the above together we arrive at the full forward model, illustrated in Fig. 4. Given that all of our operations – from re-projection to Bézier interpolation – are fully differentiable, we train all these components simultaneously, end-to-end, through stochastic gradient descent. But to do this, we need an objective to minimize. We employ a weighted composite loss

$$\mathcal{L} = \mathcal{L}_D + \alpha_p(\mathcal{L}_p/\mathcal{L}_D)\mathcal{R}, \quad \alpha_p > 0, \beta_p \geq 1$$

$$\mathcal{L}_D = \left|I - C^D\right|^2 / (sg(C) + \epsilon_c)^2$$

$$\mathcal{L}_p = \left|I - C^P\right|^2 / (sg(C) + \epsilon_c)^2$$

$$\mathcal{R} = \left|1 - d/d_p\right|^2$$

$$C = I(u, v), \quad C^N = I(u, v, N), \quad C^P = I(u, v, N)_p$$

Here $d$ is the depth output by our combined depth model, and $d_p$ is the depth of only the planar component as in Eq. (7). $C$ is a colored point sampled from our implicit image model, $C^N$ is the point sampled from the image stack $I(u, v, N)$ following Eqs. (1)–(5) for depth $d = d$, and $C^P_p$ is the point sampled following Eqs. (1)–(5) for the plane
the multi-resolution hash encoding & controls the strength of this regularization.

Coarse-to-Fine Reconstruction. First estimating low-resolution depths for whole objects before refining features such as edges and internal structures is a tried-and-true technique for improving depth reconstruction quality and consistency [9, 14]. However, one typical caveat of implicit scene representations is the difficulty of performing spatial aggregation – an image pyramid is not well-defined for a continuous representation with no concept of pixel neighborhoods. Rather than try to aggregate outputs, we recognize that the multi-resolution hash encoding $\gamma_d(u, v)$ gives us control over the scale of reconstructed features. By masking the encoding $w^\alpha \gamma_d(u, v)$ with weights $w^\alpha \in [0, 1]$ we can restrict the effective spatial resolution of the implicit depth network $f_d$, as two coordinates that map to the same masked encoding are treated as identical points by $f_d$. During training, we evolve this weight vector as

$$w^\alpha = 1/(1 + exp(-k \bar{i}))$$

$$k = -k_{min} + (epoch \cdot k_{max})/max\_epochs$$

which smoothly sweeps from passing only low-resolution grid encodings to all grid encodings during training, with $k_{min}$ and $k_{max}$ controlling the rate of this sweep. The effects of this masking are visualized in Fig. 5.

Training and Implementation Details. For simplicity of notation we have so far only worked with a single projected point. In practice, during a single forward pass of the model we perform one-to-all projection of a batch of 1024 points at a time from the reference $I(u, v)$ to all 42 frames in $I(u, v, N)$. We perform stochastic gradient descent on $L$ with the Adam optimizer [34]. Our implementation is built on tiny-cuda-dnn [54], and on a single Nvidia A100 GPU has a training time of approximately 15 minutes per scene. Our encoding parameters are $N_{\gamma_{\min}} = 8, N_{\gamma_{\max}} = 2048, L_{\gamma_{\max}} = 16, F_{\gamma_{\max}} = 2^{22}$ and $N_{\gamma_{\min}} = 8, N_{\gamma_{\max}} = 128, L_{\gamma_{\max}} = 8, F_{\gamma_{\max}} = 4, T_{\gamma_{\max}} = 2^{14}$, as depth has significantly less high-frequency features than $I(u, v)$. The networks $f_i$ and $f_c$ are both 5-layer 128 neuron MLPs with ReLU activations. For the rotation offset weight we choose $\eta_r = 10^{-4}$; regularization weight $\alpha_p = 10^{-4}$ and $\epsilon_c = 10^{-3}$; encoding weight parameters $k_{\min} = -100, k_{\max} = 200$; and number of control points $N_{\gamma_{\max}} = 21$, one for every two frames. We provide additional training details, and an extensive set of ablation experiments in the Supplemental Document to illustrate the effects of these parameters and how the above values were chosen. Our data capture app is built on the AVFoundation library in iOS 16 and tested with iPhone 12-, 13-, and 14-Pro devices. For consistency, a single 14 Pro device was used for all data captured in this work. RAW capture is hardware/API limited to ~21 FPS, hence a two-second long-burst contains 42 frames.

5. Assessment

Evaluation. We compare our approach to the most similar purely multi-view methods BARF [41] and Depth From Uncalibrated Small Motion Clip (DfUSMC) [21], both of which estimate depth and camera motion directly from an input image stack. We note that BARF also has a similar implicit image generation model. We also compare to learned monocular methods: iPhone’s 14 Pro’s native depth output and MiDaS [60], a robust single-image approach. Lastly, we compare to Robust Consistent Video Depth Interpolation (RCVD) [37] and Handheld Multi-frame Neural Depth
Refinement (HNDR) [11], which both use multi-view information to refine initial depth estimates initialized from a learned monocular approach. The latter of which is most directly related to our approach as it targets close-range objects imaged with multi-view information from natural hand tremor, but relies on iPhone LiDAR hardware for depth initialization and pose estimation. All baselines were run on processed RGB data synchronously acquired by our data capture app, except for HNDR which required its own data capture software that we ran in tandem to ours. We note that other neural scene volume methods such as [49] require COLMAP as a pre-processing step, which fails to find pose solutions for our long-burst data. To assess absolute performance and geometric consistency, we scan a select set of complex 3D objects, illustrated in Fig. 7, with a commercial high-precision turntable structured light scanner (Einscan SP). We use this data to generate ground truth object meshes, which we register and render to depth with matching camera parameters to the real captures. For quantitative depth assessment, we use relative absolute error and scale invariant error, commonly used in monocular depth literature [70]; see the Supplemental Document for details.

Reconstruction Quality. Tested on a variety of scenes, illustrated in Fig. 6, we demonstrate high-quality object depth reconstruction outperforming existing learned, mixed, and multi-view only methods. Of particular note is how we are able to reconstruct small features such as Dragon’s tail, Harold’s scarf, and the ear of the Tiger statue consistent to the underlying scene geometry. This is in contrast to methods such as RCVD or HNDR which either neglect to reconstruct the Tiger’s ear or reconstruct it behind its head. Our coarse-to-fine approach also allows us to reconstruct scenes with larger low-texture regions, such as Harold’s head, which produces striped depth artifacts for HNDR as it can only refine depth within a patch-size of high-contrast edges. Our depth on a plane decomposition avoids spurious depth solutions in low-parallax regions around objects, cleanly segmenting them from their background. This plane segmentation, and it’s applications to image and depth matting, are further discussed in the Supplemental Document. In contrast to DfUSMC, which relies on sparse feature matches and RGB-guided filtering to in-paint contiguous depth regions, our unified end-to-end model directly produces complete and continuous depth maps.
relative absolute error / scale invariant error
metrics formatted as removing device initial rotation estimates from our model. Met-
complex objects. While from a single image
various key method components. For the No Gyro tests we replace device rotations $R_d(N)$ with identity rotation for all
frames N and learn offsets as usual. We find that while the use of a fixed reference image, 8-bit RGB, or no gyro measurements can reduce our model’s average reconstruction quality, all these experiments still converge to acceptable depth solutions. This is especially true of the No Gyro experiments, which for many scenes results in near identical reconstructions. This further validates our model’s ability to independently learn high quality camera pose estimates, and demonstrates its modularity with respect to input data and optimization settings – applicable even to settings where RAW images and device motion data are not available.

6. Discussion and Future Work

In this work, we demonstrate that from only a stack of images acquired during long-burst photography, with paral-
xiaction from natural hand-tremor, it is possible to recover high-quality, geometrically-accurate object depth. **Forward Models.** Our static single-plane RGB-D representation could potentially be modified to include differentiable models of object motion, deformation, or occlusion. **Image Refinement.** We use the learned image $I(u, v)$ as a vehicle for depth optimization, but it could be possible to flip this and use the learned depth $D(u, v)$ as a vehicle for aggregating RGB content (e.g., denoising or deblurring). **From Pixels to Features.** Low-texture or distant image regions have insufficient parallax cues for ray-based depth estimation. Learned local feature embeddings could help aggregate spatial information for improved reconstruction. **Acknowledgements.** We thank Jinglun Gao and Jun Hu for their support in developing the data capture app. Ilya Chugunov was supported by NSF GRFP (2039656). Felix Heide was supported by a Packard Foundation Fellowship, an NSF CAREER Award (2047359), a Sloan Research Fellowship, a Sony Young Faculty Award, a Project X Innovation Award, and an Amazon Science Research Award.
References


