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 [13]. Modern cellphones offer double-digit megapixel image streams at high fram rates; 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 [21,55,79], simulated depth-of-field effects [1,71,72], and frame interpolation [3]. Beyond helping improve or understand RGB content, depth itself is a valuable output for simulating objects in augmented reality [6,14,44,64] and interactive experiences [26,36].

Just as there are many ways to use depth, there are also many ways to produce it. Active depth methods such as pulsed time-of-flight [46] (e.g. LiDAR), correlation time-of-flight [38] (e.g. Kinect v2), and structured light [61,83] (e.g. Kinect v1) use controlled illumination to help with depth reconstruction. While this can help these methods operate more independent of image content, it also comes with complex circuitry and power demands [28]. Thus, miniaturization for mobile applications results in very low-resolution sub-kilopixel sensors [9,27,74]. 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.

Existing passive monocular depth estimation methods leverage training data to learn shape priors [7,30,59] – 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 [17,69] 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.

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 [12]. 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 [12] 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 for visualization and comparison methods.
- 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.

We provide additional materials on: our project page.

2. Related Work

Depth estimation can be broadly divided into active and passive 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 first, before discussing neural scene representations.

Active Depth Reconstruction. Structured light and active stereo method rely on patterned illumination to directly infer object shape [16, 83] and/or improve stereo feature matching [61]. In contrast, time-of-flight (ToF) depth 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 [9] 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 [4, 77], focus cues [78], or generic learned priors [7, 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 [18, 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 [12] 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 [5, 11, 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 unstabilized 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. [12] 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 [13]. 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 [19, 25], and image superresolution, demosaicing, and denoising tasks [75, 76]. On the other end of the imaging spectrum we have video processing literature, which operates on sequences hundreds 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 scenarios. 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. [12].

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 and exposure and record a two-second, 42 frame long-burst to the device. Our method uses the 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.1 A modern mobile image signal processing pipeline can have more than a dozen steps between light hitting the CMOS sensor and a photo appearing on screen: denoising, demosaicing, and gamma correction to name a few [13]. 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 image files) [8]. In our work we are concerned with the track-

1RAW here refers to sensor data after basic corrections such as compensating for broken and non-uniform pixels, not “raw-raw” data [57]
Figure 3. The Bayer color filter array on a camera sensor produces a spatially "mosaicked" RAW image, where each 2×2 block contains a blue, red, and two green pixels. Rather than mix channel content to "demosaic" the image, we separate these channels into three planes and only linearly interpolate gaps between measured pixels, preserving the original RAW values.

ing and reconstruction of small image features undergoing small continuous motion from natural hand tremor, and so apply minimal processing to our image data, using linear interpolation to only fill the gaps between Bayer measurements as shown in Fig. 3. 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 projective 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

$$C = [R, G, B]^T = I(u, v), \quad d = D(u, v)$$

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

$$
\begin{bmatrix}
  x \\
  y \\
  z \\
  1 
\end{bmatrix} = \pi^{-1}
\begin{bmatrix}
  u \\
  v \\
  d 
\end{bmatrix} ; K
= \begin{bmatrix}
  d(u - c_x)/f_x \\
  d(v - c_y)/f_y \\
  d \\
  1
\end{bmatrix},
$$

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

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

Here, $P(N)$ is decomposed into a $3 \times 3$ rotation matrix $R(N)$ and $3 \times 1$ translation vector $T(N) = [t_x, t_y, t_z]^T$. 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

$$
\begin{bmatrix}
  u^N \\
  v^N \\
  z^N \\
  1 
\end{bmatrix} = \pi
\begin{bmatrix}
  x^N \\
  y^N \\
  z^N \\
  1 
\end{bmatrix} ; K(N)
= \begin{bmatrix}
  (f_x x^N)/z^N + c_x \\
  (f_y y^N)/z^N + c_y \\
  1
\end{bmatrix},
$$

where $K(N)$ are the frame intrinsics. We can now use these coordinates to sample from the full image volume, that is

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

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 multiview 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. 4. *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.
Camera Coords. Multiresolution Hash Encoding Implicit Image $I(u, v)$ Implicit RGBD Model Motion Model

<table>
<thead>
<tr>
<th>Camera Coords.</th>
<th>Multiresolution Hash Encoding</th>
<th>Implicit Image $I(u, v)$</th>
<th>Implicit RGBD Model</th>
<th>Motion Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$Y$</td>
<td>$I(u, v)$</td>
<td>$T(N)$</td>
<td>$P(u, v)$</td>
</tr>
<tr>
<td>$M$</td>
<td>$E$</td>
<td>$D(u, v)$</td>
<td>$R(N)$</td>
<td>$L(u, v, N)$</td>
</tr>
</tbody>
</table>

**Figure 5.** We model a long-burst capture as a single, fully-differentiable forward model comprised of an implicit image $I(u, v)$ projected through implicit depth $D(u, v)$ with motion model $[R(N)T(N)]$. Calculating reprojection loss with respect to the the captured image stack $I(u, v, N)$, we train this model end-to-end and distill high quality depth and camera motion estimates directly from the burst data.

Under small camera motion, many depth solutions for a pixel can map 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 [12, 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

$$ I(u, v) = f_1(\gamma_1(u, v; \text{params}_{\gamma_1}); \theta) $$

$$ \text{params}_{\gamma_1} = \{ N_{\text{min}}, N_{\text{max}}, L^3, F^3, T^3 \} $$

where $f_1$ is a multi-layer perceptron (MLP) [29] with learned weights $\theta$. This MLP learns a mapping from $\gamma_1(u, v)$, a positional encoding of sampled camera coordinates, to image color. Specifically, we borrow the multi-resolution hash encoding from Müller et al. [53] for its spatial aggregation properties. The parameters in $\text{params}_{\gamma_1}$ determine the minimum $N_{\text{min}}$, and maximum $N_{\text{max}}$, grid resolutions, number of grid levels $L^3$, number of feature dimensions per level $F^3$, and overall hash table size $T^3$.

**Implicit Depth on a Plane Model.** Our depth model is a similar implicit representation with a learned planar offset

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

$$ d_p = D_p(u, v) = au + bv + c, $$

where $\{a, b, c\}$ are the learned plane coefficients, and $\gamma$ 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, $f_1$ 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 [12], we opt for a low-parameter Bézier curve motion model

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

$$ B(t; P, N_c) = \sum_{i=0}^{N_c} \left( \frac{N_c}{i} \right) (1-t)^{N_c-i} t^i P_i, $$

with $N_c$ number of control points $P_i$. Translation estimates $T(N)$ are learned from scratch, whereas rotations $R(N)$ are initialized with 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

$$ P_i^R = \begin{bmatrix} 0 & -r^z & r^y \\ r^z & 0 & -r^x \\ -r^y & r^x & 0 \end{bmatrix}. $$

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. 5. 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

$$ L = \mathcal{L}_d + \alpha_p(\mathcal{L}_r/\mathcal{L}_d) \mathcal{R}, \ \ \ \ \ \alpha_p > 0, \ \beta_p \geq 1 $$

$$ \mathcal{L}_d = \left| (C - C_p) / \left( \text{sg}(C) + \epsilon_c \right) \right|^2 $$

$$ \mathcal{L}_r = \left| (C - C_p) / \left( \text{sg}(C) + \epsilon_c \right) \right|^2 $$

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

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

5
scene representations is the difficulty of performing spatial

dimensionality of performing spatial

consistency [10,15]. However, one typical caveat of implicit

such as edges and internal structures is a tried-and-true tech-

\[ \alpha \]

meter penalty for incorrect and noisy depth estimates. The param-

otherwise, in low-texture regions, there is no photometric

First estimating low-

ulation rather than spurious depth predictions from

mation are pulled towards a spatially consistent plane so-

textureless, or otherwise have no meaningful parallax infor-

Figure 6. Ablation experiment on the effect of masked encod-

ings levels. Using 4 and 6 encoding levels for depth leads to ex-

pected lower-resolution depth reconstructions. However, in 8 Lev-

els, when weights are not swept from coarse-to-fine the recon-

struction acquires sharp depth artifacts due to a positive feedback

loop during training: high-frequency image gradients from \( I(u, v) \) produce discontinuities in estimated depth \( D(u, v) \), which in turn produce high-frequency images gradients in \( I(u, v) \).

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 \( d = d_p \), and \( C^y \) is the point sampled following Eqs. (1)–(5) for the plane depth \( d = d_p \). The regularization term (13) penalizes the magnitude of \( f_d \), pulling the depth output towards the plane model. Losses (11) and (12) are relative square photometric errors between sampled colored points, where \( sg \) is the stop-gradient operator preventing the denominator \( C \)'s gradient from being back-propagated. This normalization by the approximate luminance of sampled points is effective in aiding the unbiased reconstruction of HDR images [39], and we refer the reader to derivations in Mildenhall et. al [49] on its relation to tone-mapping. In (10), we combine the photometric loss term \( L_d \), which seeks to maximize overall image reconstruction quality, with a weighted regularization \( R \) which penalizes divergence from the planar model. When \( L_d \approx L_\text{L} - \text{i.e. the depth offset from } f_d \), our model is not improving reconstruction quality over a simple plane – the model is strongly penalized. As \( L_d \) decreases relative to \( L_\text{L} \) – the implicit depth model \( f_d \) improves reconstruction quality – this penalty falls off. In this way, regions that are blurred, textureless, or otherwise have no meaningful parallax information are pulled towards a spatially consistent plane solution rather than spurious depth predictions from \( f_d \). As otherwise, in low-texture regions, there is no photometric penalty for incorrect and noisy depth estimates. The parameter \( \alpha \) 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 tech-

mique for improving depth reconstruction quality and consist-

ency [10,15]. 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 neigh-

bors. Rather than try to aggregate outputs, we recog-

ize that the multi-resolution hash encoding \( \gamma_\theta(u, v) \) gives

us direct control over the scale of reconstructed features. By

masking the encoding \( u_\theta \gamma_\theta(u, v) \) with weights \( w_\theta \in [0, 1] \)

we can restrict the effective 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 train-

ing, we evolve this weight vector as

\[ w_\theta = 1/(1 + \exp(-ki)) \]

\[ k = -k_{\text{min}} + (\text{epoch} \cdot k_{\text{max}})/\text{max epochs} \] (14)

which smoothly sweeps from passing only low-resolution grid encodings to all grid encodings during training, with \( k_{\text{min}} \) and \( k_{\text{max}} \) controlling the rate of this sweep. The effects of this masking are visualized in Fig. 6.

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 implementa-
tion is built on tiny-cuda-dnn [54] and trained on a sin-
gle Nvidia A100 GPU with a total runtime of \(~15\) minutes per scene. Our encoding parameters are \( N_\gamma_{\text{min}} = 8, N_\gamma_{\text{max}} = 2048, L_\gamma = 16, L^\gamma = 4, T_\gamma = 2^{22} \) and \( N_\gamma = 8, N_\gamma_{\text{max}} = 128, L_\gamma = 8, L^\gamma = 4, T_\gamma = 2^{14} \), as depth has significantly less high-frequency features than \( I(u, v) \). The networks \( f \) and \( f_d \) are both 5-layer 128 neuron MLPs with ReLU activations. For the rotation offset weight we choose \( \eta_k = 10^{-4} \); regularization weight \( \alpha = 10^{-4} \) and \( \epsilon = 10^{-3} \); encoding weight parameters \( k_{\text{min}} = -100, k_{\text{max}} = 200 \); and number of control points \( N^y = N^x = 21 \), 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 AVFounda-
tion 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 sim-
ilar purely multi-view methods BARF [41] and Structure

From Small Motion (SfSM) [31], both of which estimated
depth and camera motion directly from an input image

stack. We note that BARF also has a similar implicit im-
Figure 7. Qualitative comparison of reconstruction results of indoor and outdoor scenes for a range of learned, purely multi-view, and mixed depth estimation methods. Given the mix of depth scales, results are re-scaled by minimizing relative mean square error.

age 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) [12], 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 which 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. 8, 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 loss and log loss as described and commonly used in monocular depth literature [70]; see the Supplemental Document for details.

Reconstruction Quality. Tested on a variety of scenes, illustrated in Fig. 7, 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. Compared to other purely multi-view methods, our plane plus depth offset approach avoids spurious depth solutions in low-parallax regions around the objects, cleanly segmenting them from their background. This plane seg-
Figure 8. Object reconstructions visualized as rendered meshes, with associated depth error metrics formatted as relative absolute difference/log difference. Edges over $10 \times$ the length of their neighbors were culled to avoid connecting mesh features in occluded regions. Additional renders are available in the Supplemental Document and associated video materials.

Figure 9. If we fix the image representation $I(u, v)$ to be the first long-burst frame $I(u, v, 0)$, or we replace the RAW data in $I(u, v, N)$ with processed 8-bit RGB, we see a reduction in overall depth reconstruction quality. Note the loss in detail around the star-shaped leaf in the center of the depth map zooms.

In this work, we demonstrate that from only a stack of images acquired during long-burst photography, with parallax information from natural hand-tremor, it is possible to recover high-quality, geometrically-accurate object depth. Forward Models. While we focus on static scenes modeled by a single implicit RGB-D representation, our approach could potentially be modified to include differentiable models of object motion, deformation, or occlusion.

Forward Models. While we focus on static scenes modeled by a single implicit RGB-D representation, our approach could potentially be modified to include differentiable models of object motion, deformation, or occlusion.

Image Refinement. We primarily use the learned image $I(u, v)$ as a vehicle for depth optimization, but is possible to flip this. The learned depth $D(u, v)$ can potentially be used as a vehicle for aggregating RGB information between frames for image deblurring and denoising tasks.

From Pixels to Features. Low-texture or distant image content provides insufficient parallax information for ray-based depth estimation. In these regions larger-scale features such as patches or learned embeddings could help aggregate parallax information and improve reconstruction.

Acknowledgements. We thank Jinglun Gao and Jun Hu of Meta Reality Labs for their inputs and support in the design of the data capture app.

6. Discussion and Future Work

In this work, we demonstrate that from only a stack of images acquired during long-burst photography, with parallax information from natural hand-tremor, it is possible to recover high-quality, geometrically-accurate object depth.

Value of an Implicit RAW Image Model. In Fig. 9 we observe how RAW long-burst data can improve depth reconstruction compared to using 8-bit RGB, as it offers higher effective resolution for finding matching pixels between frames. Similarly, a learned $I(u, v)$ outperforms using the first long-burst frame as a static reference as in [12]. Updating $I(u, v)$ during training might allow the model to avoid local minima solutions, as it first matches larger –but blurred or misaligned– image features before jointly refining the image and depth to match higher frequency details.

6. Discussion and Future Work

In this work, we demonstrate that from only a stack of images acquired during long-burst photography, with parallax information from natural hand-tremor, it is possible to recover high-quality, geometrically-accurate object depth.

Forward Models. While we focus on static scenes modeled by a single implicit RGB-D representation, our approach could potentially be modified to include differentiable models of object motion, deformation, or occlusion.

Image Refinement. We primarily use the learned image $I(u, v)$ as a vehicle for depth optimization, but is possible to flip this. The learned depth $D(u, v)$ can potentially be used as a vehicle for aggregating RGB information between frames for image deblurring and denoising tasks.

From Pixels to Features. Low-texture or distant image content provides insufficient parallax information for ray-based depth estimation. In these regions larger-scale features such as patches or learned embeddings could help aggregate parallax information and improve reconstruction.

Acknowledgements. We thank Jinglun Gao and Jun Hu of Meta Reality Labs for their inputs and support in the design of the data capture app.
References


In this document, we present supporting material including additional results, implementation details, ablation experiments, and further analysis in support of the findings from the main text. We organize this material as follows:

- Section A: Data, training, and evaluation details.
- Section B: Additional ablation studies on manipulating network and encoding parameters.
- Section C: Additional reconstruction results and examination of challenging imaging settings.
- Section D: Additional validation on simulated data with evaluation of motion estimates.
- Section E: Applications to depth and image matting.

A. Implementation Details

Long-Burst Data. We acquire long-burst data through a custom app built on the AVFoundation camera framework for iOS 16. While the vanilla AVFoundation framework offered a default method to capture burst or bracketed sequences, it was limited to only four frame sequences with significant overhead between captures, necessitating custom streaming code to save a longer continuous sequence of RAW data. A restriction we could not lift, however, is the inability to stream RAW captures from multiple cameras simultaneously. If this were possible, you could potentially use parallax and focus cues between two synchronized camera streams – for example the wide and ultra-wide cameras – to further improve reconstruction in the overlap of their fields of view. During capture, we record the following: Bayer CFA RAWs (42 frames $4032 \times 3024$px), processed RGB images (42 frames $1920 \times 1440$px), depth maps (42 frames $320 \times 240$px), frame timestamps, ISO, exposure time, brightness estimates, black level, white level, camera intrinsics, lens distortion tables, device acceleration estimates ($\sim 200$ measurements at $100$Hz), device rotation estimates ($\sim 200$), and motion data timestamps ($\sim 200$). To account for lens shading effects in bright scenes, as illustrated in Fig. 1, we estimate a shade map with the help of a simple diffuser and uniform light source. We note that neglecting to compensate for lens shading can disrupt depth estimation in the corners of the image as matching pixels no longer have uniform brightness between frames.

Training. We sample 1024 points $(u, v)$ per iteration of training, projecting these to $42 \times 1024$ points in the image stack $\hat{I}(u, v, N)$, corresponding to 1024 points per frame. We perform 256 iterations per epoch, for 100 epochs of training with the Adam optimizer [34] with betas $(0.9, 0.99)$ and epsilon $10^{-15}$. We exponentially decay learning rate during training with a factor of 0.98 per epoch. Training on a single Nvidia A100 takes approximately 15 minutes.

Evaluation. To generate depth maps we sample $D(u, v)$ at a grid of $(h, w) = (1920, 1440)$ points $(u, v) \in [0, 1]$. To reduce noise introduced by the stochastic training process we median filter this result with kernel size 13 before visualization. For depth evaluation, we use relative absolute error $L1$-rel and scale invariant error $sc-inv$ metrics, that is

$$L1\text{-rel}(\hat{d}, \hat{d}) = \frac{1}{HW} \sum_{u, v} \frac{|d(u, v) - \hat{d}(u, v)|}{\hat{d}(u, v)},$$
Figure 2. Ablation study on the effect of the number of levels $L^D$, and effective max resolution $N^D_{\text{max}}$, in the multiresolution hash encoding $\gamma^D$ on reconstruction. Here, given a scale factor of $\sqrt{2}$ between levels, $L^D = 2, 4, 6, 8, 16$ correspond to $N^D_{\text{max}} = 16, 32, 128, 2048$. The qualitatively best reconstructions are highlighted with a dashed border.

Figure 3. Ablation study on the effect of hash table size $T^D$ on reconstruction quality. Selected $T^D$ is highlighted with a dashed border.
Scenes such as the scale factors in an image pyramid \([2]\). As we see in we reconstruct, and presents a similar problem to choosing training. This in turn controls the scale of depth features our implicit depth representation \(f\) encoding \(\gamma\) Encoding.

### B. Additional Ablation Experiments

#### Encoding.

In this work we use the multiresolution hash encoding \(\gamma_D\) to directly control what spatial information our implicit depth representation \(f_d\) has access to during training. This in turn controls the scale of depth features we reconstruct, and presents a similar problem to choosing the scale factors in an image pyramid \([2]\). As we see in Fig. 2, increasing the number of levels \(L_D\) and effective max resolution \(N_{\text{max}}\) increases the spatial frequency of reconstructed depth features. Scenes such as Branch contain both high-frequency image and depth content, thin textured needles, and are best reconstructed by a fine resolution grid with \(L_D = 16\). The Desk Gourds, however, have small image features in the patterns on the gourds, but relatively low-frequency depth features. Setting \(L_D = 16\) allows the network to overfit to these features and bleed image texture into the depth reconstruction. We select \(L_D = 8\) as a compromise between these imaging settings, but in practice, like other neural scene fitting task \([53]\), different scenes have different optimal encoding parameters for maximum reconstruction quality. We find hash table size \(T^\gamma_D\) significantly easier to tune, as choosing an overly large table size primarily affects model storage size, rather than reconstruction quality. We thus choose \(T^\gamma_D = 2^{14}\), the smallest table size which does not lower the detail of depth reconstruction, as shown in Fig. 3.

#### Depth Model.

The main adjustable parameter in our forward model is the plane regularization weight \(\alpha_p\). This plane regularization affects depth reconstruction bidirectionally, regions with little parallax information are pulled towards the plane to remove spurious depth estimates, but in order to minimize depth offset, the plane is also pulled towards the reconstructed foreground objects. The effect of this can be seen in Fig. 5, where for very low \(\alpha_p \leq 10^{-5}\) this plane does not align with the foreground depth, and instead drifts into the background, causing a discontinuity in the reconstruction. Conversely, for large \(\alpha_p \geq 10^{-3}\), this regularization is so strong that the plane begins to cut into the foreground objects, flattening regions with low parallax information. We find \(\alpha_p = 10^{-4}\) to work well for a wide range of scenes, gluing the depth plane to the limit of reconstructed objects. We note that in scenes such as Desk Gourds, and as we will see later with synthetic data, this plane accurately reconstructs the real geometry of the background. However, for many settings it is more akin to a segmentation mask than depth, designating the area which we cannot reconstruct using parallax information.

#### Motion Model.

We use a Bézier curve model to represent translation between frames, as natural hand-tremor draws a continuous low-velocity path during capture. By limiting the number of control points \(N_c\) in this model we can enforce smoothness constraints on this motion, the effects of which are illustrated in Fig. 6. Not surprisingly, using too few control points does not allow us to faithfully model camera motion and results in blurry image reconstruction and inconsistent depth estimates. We thus choose the smallest number of control points which leads to successful image and depth reconstruction. We note that while for Desk Gourds reconstruction succeeded with \(N_c = 42\), for many scenes setting \(N_c \geq 42\) leads to very unstable training as the over-defined motion model can generate erratic high-velocity motion between frames.

Frames in a long-burst contain >90% overlapping scene content, and so many non-physical solutions for depth will produce identical reprojection error as compared to more geometrically plausible depth maps. This is illustrated in Fig. 4, where by “tearing” the image – compressing patches of similar colored pixels from the reference frame – the non-physical depth incurs no additional photometric penalty, and so results in an identical reprojection error to a far more qualitatively reasonable depth reconstruction.

\[
\text{reprojection error} = \frac{1}{\text{HW}} \sum_{u,v,N} |I(u,v) - I(u^N, v^N, N)|.
\]

![Figure 4. In this example both Depth A and B produce identical reprojection error, but where Depth A models smooth geometry which warps the image between frames to model parallax, Depth B performs a brute-force mapping of individual pixels in the reference frame to points with similar values in the image stack.](image)

\[
s^{-1}(d, \hat{d}) = \sqrt{\frac{1}{\text{HW}} \sum_{u,v} \delta(u,v)^2 - \frac{1}{(\text{HW})^2} \left(\sum_{u,v} \hat{d}(u,v)\right)^2}
\]

\[
\delta(u,v) = \log(d(u,v)) - \log(\hat{d}(u,v)),
\]

which are often used in the monocular depth estimation literature \([70]\) to compare approaches with varying scales and representations of depth. For methods such as MiDaS and RCVD we first convert inverse depth to depth before applying these metrics. We purposely avoid using photometric loss or reprojection error as comparison metrics \([12]\) for similar arguments as discussed in Gao et al. \([20]\). Where

\[
\text{sc-inv}(d, \hat{d}) = \sqrt{\frac{1}{\text{HW}} \sum_{u,v} \delta(u,v)^2 - \frac{1}{(\text{HW})^2} \left(\sum_{u,v} \hat{d}(u,v)\right)^2}
\]

\[
\delta(u,v) = \log(d(u,v)) - \log(\hat{d}(u,v)),
\]

### B. Additional Ablation Experiments

#### Encoding.

In this work we use the multiresolution hash encoding \(\gamma_D\) to directly control what spatial information our implicit depth representation \(f_d\) has access to during training. This in turn controls the scale of depth features we reconstruct, and presents a similar problem to choosing the scale factors in an image pyramid \([2]\). As we see in Fig. 2, increasing the number of levels \(L_D\) and effective max resolution \(N_{\text{max}}\) increases the spatial frequency of reconstructed depth features. Scenes such as Branch contain both high-frequency image and depth content, thin textured needles, and are best reconstructed by a fine resolution grid with \(L_D = 16\). The Desk Gourds, however, have small image features in the patterns on the gourds, but relatively low-frequency depth features. Setting \(L_D = 16\) allows the network to overfit to these features and bleed image texture into the depth reconstruction. We select \(L_D = 8\) as a compromise between these imaging settings, but in practice, like other neural scene fitting task \([53]\), different scenes have different optimal encoding parameters for maximum reconstruction quality. We find hash table size \(T^\gamma_D\) significantly easier to tune, as choosing an overly large table size primarily affects model storage size, rather than reconstruction quality. We thus choose \(T^\gamma_D = 2^{14}\), the smallest table size which does not lower the detail of depth reconstruction, as shown in Fig. 3.

#### Depth Model.

The main adjustable parameter in our forward model is the plane regularization weight \(\alpha_p\). This plane regularization affects depth reconstruction bidirectionally, regions with little parallax information are pulled towards the plane to remove spurious depth estimates, but in order to minimize depth offset, the plane is also pulled towards the reconstructed foreground objects. The effect of this can be seen in Fig. 5, where for very low \(\alpha_p \leq 10^{-5}\) this plane does not align with the foreground depth, and instead drifts into the background, causing a discontinuity in the reconstruction. Conversely, for large \(\alpha_p \geq 10^{-3}\), this regularization is so strong that the plane begins to cut into the foreground objects, flattening regions with low parallax information. We find \(\alpha_p = 10^{-4}\) to work well for a wide range of scenes, gluing the depth plane to the limit of reconstructed objects. We note that in scenes such as Desk Gourds, and as we will see later with synthetic data, this plane accurately reconstructs the real geometry of the background. However, for many settings it is more akin to a segmentation mask than depth, designating the area which we cannot reconstruct using parallax information.

#### Motion Model.

We use a Bézier curve model to represent translation between frames, as natural hand-tremor draws a continuous low-velocity path during capture. By limiting the number of control points \(N_c\) in this model we can enforce smoothness constraints on this motion, the effects of which are illustrated in Fig. 6. Not surprisingly, using too few control points does not allow us to faithfully model camera motion and results in blurry image reconstruction and inconsistent depth estimates. We thus choose the smallest number of control points which leads to successful image and depth reconstruction. We note that while for Desk Gourds reconstruction succeeded with \(N_c = 42\), for many scenes setting \(N_c \geq 42\) leads to very unstable training as the over-defined motion model can generate erratic high-velocity motion between frames.
Figure 5. Ablation study on the effects of regularization weight $\alpha_P$ on reconstruction quality. Selected $\alpha_P$ highlighted with a dashed border.

Figure 6. Ablation study on the effect of the number of chosen control points $N_T = N_R$ on reconstruction quality, with image reconstructions $I(u, v)$ and estimated motion paths plotted below. The selected number of control points is highlighted with a dashed border.

Figure 7. Depth reconstruction results for a set of challenging imaging scenarios. Not visible is the large motion of leaves in the Dynamic scene, captured during high wind. Areas of interest are highlighted with a dashed border.
Figure 8. Reconstruction on 7 additional scenes for our method and a mix of learned, purely multi-view, and mixed depth estimation methods. Given the mix of depth representations, results are re-scaled by minimizing relative mean square error.
C. Additional Reconstruction Results

**Challenging Imaging Scenarios.** Given the fundamental building blocks of our approach, namely that it performs multi-view depth estimation through ray reprojection, some scenes will naturally be more difficult to reconstruct than others. As shown in Fig. 7, each of these scenarios presents its own set of challenges and direction of study. In the Dynamic scene, we fail to reconstruct accurate depth for the majority of the plant leaves as they undergo deformation far larger than the parallax effects we observe in the long-burst. Our forward model has no way to model this deformation, and it is notoriously difficult to separate the effects of object motion from camera motion. The Textureless and Distant scenes present two sides of a similar problem, insufficient parallax information. While we are able to reconstruct the plant in Textureless, the textureless planter provides no multi-view information from which to estimate depth except for along its edges, which we can track relative to the motion of the background. The church in Distant is so far from the camera that it exhibits only fractions of pixel in disparity over the entire long-burst. In both these scenarios we need a mechanism to aggregate information in image space to make up for the lack of parallax. In Textureless this would be in-painting the planters depth based on its edges, and in Distant we would need to look at the deformation of larger image patches to estimate sub-pixel motion. The Thin Structures reconstruction is partially successful, as in the foreground region we are able to track and reconstruct the depth of the thin orange mesh, but breaks down when it begins to overlap with the traffic cone. We suspect this is because our forward model is a single-layer RGB-D representation, with no explicit way to model for occlusions. In the region of the traffic cone is has to decide between reconstructing the cone or the mesh in front of it in the long-burst $I(u, v, N)$ data, not both. Here, a layered depth representation could potentially solve this, but greatly increases the complexity of the problem as we would now need to learn an alpha map for each frame $N$ to sample these layers. For the Very High Dynamic Range scene, we have specular reflections three orders of magnitude brighter than the shadowed portions of the statue. While using the fixed auto exposure and ISO settings we are able to reconstruct a large portion of the statue body with our RAW data. To reconstruct all the regions of the scene, including the dimly-lit body, our model could potentially be augmented to incorporate bracketed image data with varying exposure, similar to Mildenhall et al. [49], and perform joint HDR image volume and depth reconstruction. Lastly, the Lens Blur scene shows a loss in depth reconstruction performance due to portions of the scene being blurred by a shallow depth of field from the camera. Depth-from-defocus cues [78] could potentially help regularize reconstruction in these areas.

**Additional Comparisons.** Fig. 8 provides additional qualitative comparisons of our proposed approach to a wide set of baseline methods. This includes the four target objects used to demonstrate object reconstruction in the main text, prefixed with **Obj-**. The visualizations also reflect the challenges in evaluating methods purely from depth maps, as geometric inconsistencies that are apparent in the mesh projections – such as the distorted arms of **Obj-Ganesha** – are much harder to identify in these 2D visualizations. In addition to these objects, we include 3 scenes **Leopardy, Bush, and Houseplant**, which demonstrate successful reconstruction with deceptive image features, small depth features, and large field of view respectively. Of particular note is how we are able to reconstruct the needles of the **Bush** scene and individual leaves of **Houseplant**, where other methods blend features at different depth levels together.

D. Synthetic Evaluation

**Setup.** To further validate our approach we use the high-fidelity structured light object scans we acquired for quantitative evaluation to generate simulated long-burst captures. Illustrated in Fig 9, we apply a Voronoi color texture to the surface of these meshes, and place them in front of a tilted background plane with an outdoor image texture. We add depth-of-field effects and match camera intrinsics to our real captures – using the ARKit poses captured by the software from Chugunov et al. [12] to generate realistic hand tremor motion paths – and render frames at 16-bit color depth with Blender’s Eevee engine. This synthetic data allows us to not only validate the fidelity of our object reconstructions, but also our estimated camera motion paths, for which we cannot otherwise get ground truth during ordinary captures.

**Assessment on Synthetic Data.** We find that for this synthetic data, in the absence of noise, lighting changes, and other imaging non-idealities, we are able to recover nearly ground truth reconstructions of both the objects and background planes. This supports our plane plus offset depth model, which fits the simple plane to the out-of-focus background content instead of generating spurious depth estimates for regions without reliable parallax information. Though the colorful object textures make single-view depth estimation visually difficult, as illustrated by artifacts in the MiDaS reconstructions, these high-contrast cues allow our method to reconstruct even tiny features such as the tusks of the **Synth-Ganesha**. This validates that even with small camera motion, given sufficient image texture we converge on geometrically correct solutions. In Fig. 9 we also see how the camera motion estimates converge to nearly ground truth as our method jointly refines depth and camera trajectory estimates during training.
Figure 9. Depth reconstruction and motion estimation results for a set of simulated textured objects with realistic hand-tremor motion. Motion estimates are re-normalized and overlaid to demonstrate the accuracy in estimated camera trajectory to ground truth data.

E. Depth and Image Matting

Forward Model Decomposition. In the proposed plane plus offset depth model, regions that do not generate sufficient parallax information are pulled towards the plane by the regularization term $R$. While we cannot recover meaningful depth from multiview in these regions, they prove useful for scene segmentation and editing. Illustrated in Fig. 10 (a), by masking what parts of the image produce negligible depth offset, we are able to cleanly segment the tiger statue in Scene A from its background. In Fig. 10 (b) we then superimpose this masked image over Scene B, a separately captured tree-covered street. We run Scene B through MiDaS to hallucinate the depth of the background trees, and overlay this with our geometrically-estimated depth of the tiger to produce a fused depth representation. In this way we leverage multiview information where we have it, and learned image priors where we do not. In Fig. 10 (c) we see an advantage of using this plane separation technique for segmentation over depth thresholding. As the floor under the dragon figure extends both in front of and behind the figure itself, setting a depth cutoff will always either miss a part of the figure, or include the area around it. Whereas as our plane here represents the depth of the floor, we can threshold the depth offset just like in Fig. 10 (a) to recover a high-quality mask of the object. Thanks to being based on depth rather than image features, this approach has no problems with the visual ambiguity of the dragon and its background, which both contain high-frequency black and white textures.
Figure 10. Image and Depth Matting. Example of scene editing enabled by our plane plus offset forward model. We can (a) threshold the depth offset component $d(u,v) - d_p$ to recover a mask of the object in focus and then (b) superimpose it over a new scene. (c) This works even for visually ambiguous scenes where simple depth thresholding fails.