Shakes on a Plane: Unsupervised Depth Estimation from Unstabilized Photography (Supplemental Document)

Ilya Chugunov Yuxuan Zhang Felix Heide

Princeton University

1. Supplemental Document

In this supplementary document, we provide 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, one 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 4032×3024px), processed RGB images (42 frames 1920×1440px), depth maps (42 frames 320×240px), frame timestamps, ISO, exposure time, brightness estimates, black level, white level, camera intrinsics, lens distortion tables, device acceleration estimates ($\sim$200 measurements at 100Hz), device rotation

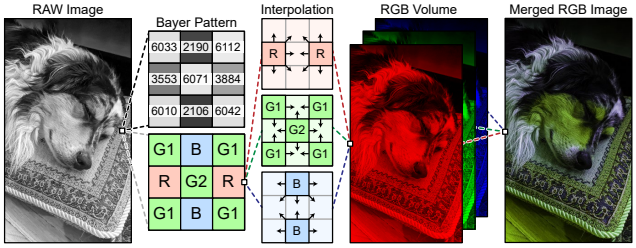

Figure 1. The bayer color filter array on a camera sensor produces a spatially "mosaicked" RAW image, where each 2×2 block contains a blue, red, and two green pixels. Rather than mix channel content to "demosaick" the image, we separate these channels into three planes and only linearly interpolate gaps between measured pixels, preserving the original RAW values.

estimates ($\sim$200), and motion data timestamps ($\sim$200). To preserve measured RAW values, we convert the single channel Bayer CFA images to three channel RGB volumes as shown in Fig. 1, linearly interpolating to fill missing values. To account for lens shading effects in bright scenes we also estimate a shade map with the help of a simple diffuser and uniform light source, illustrated in Fig. 2. 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 42×1024 points in the image stack $I(u, v, \text{N})$, corresponding to 1024 points per frame. We perform 256 iterations per epoch, for 100 epochs of training with the Adam optimizer [4] 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 $(\text{H}, \text{W}) = (1920, 1440)$ 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

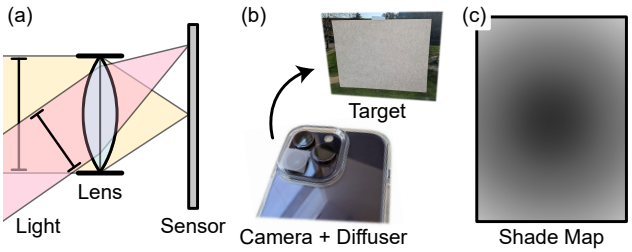

Figure 2. (a) Lens shading is an effect caused by the geometry of the camera lens assembly, where regions close to the edge of the image sensor receive less total light than the center. (b) Camera manufacturers calibrate for this by capturing what should be a uniformly bright scene and (c) generating a *shade map* to compensate for the observed fall-off in brightness in the image.

L1-rel and scale invariant error *sc-inv* metrics, that is

$$\text{L1-rel}(d, \hat{d}) = \frac{1}{\text{HW}} \sum_{u.v} \frac{|d(u,v) - \hat{d}(u,v)|}{\hat{d}(u,v)},$$

and

$$\text{sc-inv}(d, \hat{d}) = \sqrt{\frac{1}{\text{HW}} \sum_{u,v} \delta(u,v)^2 - \frac{1}{(\text{HW})^2} (\sum_{u,v} \hat{\delta}(u,v))^2}$$
$$\delta(u,v) = \log(d(u,v)) - \log(\hat{d}(u,v)),$$

which are often used in the monocular depth estimation literature [7] 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 [2] for similar arguments as discussed in Gao et al. [3], where

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

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. 3, 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 plausible depth reconstruction.

B. Additional Ablation Experiments

Encoding. In this work we use the multiresolution hash encoding γ_{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

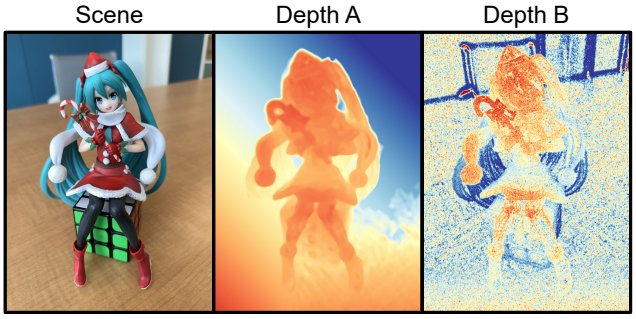

Figure 3. In this example both Depth A and B produce *identical* reprojection error, but where Depth A models smooth geometry which warps the image between frames to model parallax, Depth B performs a brute-force mapping of individual pixels in the reference frame to points with similar values in the image stack.

the scale factors in an image pyramid [1]. As we see in Fig. 4, increasing the number of levels $L^{\gamma\text{D}}$ and effective max resolution $N^{\gamma\text{D}}_{max}$ 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 $L^{\gamma\text{D}} = 16$. The *Desk Gourds*, however, have small image features in the patterns on the gourds, but relatively low-frequency depth features. Setting $L^{\gamma\text{D}} = 16$ allows the network to overfit to these features and bleed image texture into the depth reconstruction. We select $L^{\gamma\text{D}} = 8$ as a compromise between these imaging settings, but in practice, different scenes have different optimal encoding parameters for maximum reconstruction quality. We find hash table size $T^{\gamma\text{D}}$ significantly easier to tune, as choosing an overly large table size primarily affects model storage size, rather than reconstruction quality. We thus choose $T^{\gamma\text{D}} = 2^{14}$, the smallest table size which does not lower the detail of depth reconstruction, as shown in Fig. 5.

Depth Model. The main adjustable parameter in our forward model is the plane regularization weight α_{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. 6, where for very low $\alpha_{\text{P}} \leq 10^{-5}$ this plane does not align with the foreground depth, and instead drifts into the background, causing a discontinuity in the reconstruction. Conversely, for large $\alpha_{\text{P}} \geq 10^{-3}$, this regularization is so strong that the plane begins to cut into the foreground objects, flattening regions with low parallax information. We find $\alpha_{\text{P}} = 10^{-4}$ 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 back-

ground. 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. 7. 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.

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. 8, 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 $I(u, v, \text{N})$ 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. [5], and perform joint HDR image volume and depth reconstruction. The *Lens Blur* scene shows a loss in reconstruction quality due to portions of the scene being blurred by a shallow depth of field from the camera. Depth-from-defocus cues [8] could potentially help regularize reconstruction in these areas which are otherwise devoid of fine image features. Lastly, the *Translucent* and *Highly Reflective* settings both violate view consistency. Namely, changes in pixel colors can no longer be attributed solely to parallax or camera motion, and can be caused by seeing through or around the objects. We further discuss the reconstruction of non-lambertian objects in the next section.

Non-Lambertian Reconstruction. While we focus on the reconstruction of primarily lambertian scenes – matte, diffusely-reflective objects – non-lambertian scenes provide an interesting set of both imaging challenges and opportunities. We first divide this setting into two categories: *local reflections* and *distant light sources*, illustrated in Fig. 10. In the first setting, sampled light paths and colors can drastically change for even small view variations. As photometric matching tries to match reflected content, which does not follow the parallax motion of the reflective object itself, this produces erroneous depth estimates for objects such as the copper pot in Fig. 10 (a). With a distant light source, however, small changes in view angle result in the same apparent specularities as the path from the camera center to the illuminator remains connected. These specularities thus act as image texture, and exhibit the same parallax effects as the surface of the object. As seen in Fig. 10 (b) and the *Tiger* scene, this does not disrupt depth reconstruction. This finding, that specularities from distant light sources act as object texture and local reflections do not, points towards an avenue of work in lighting separation and reflection removal. Regions which do not fit a static RGB-D model and incur large photometric penalties regardless of their depth, could be separated into view-dependent texture plus reflection components for later manipulation.

Small Camera Motion. As hand shake is a naturally random process, long-burst captures have varying effective stereo baseline. While on average we can expect 5-6 millimeters of baseline [2], if we are unlucky – e.g. the user is

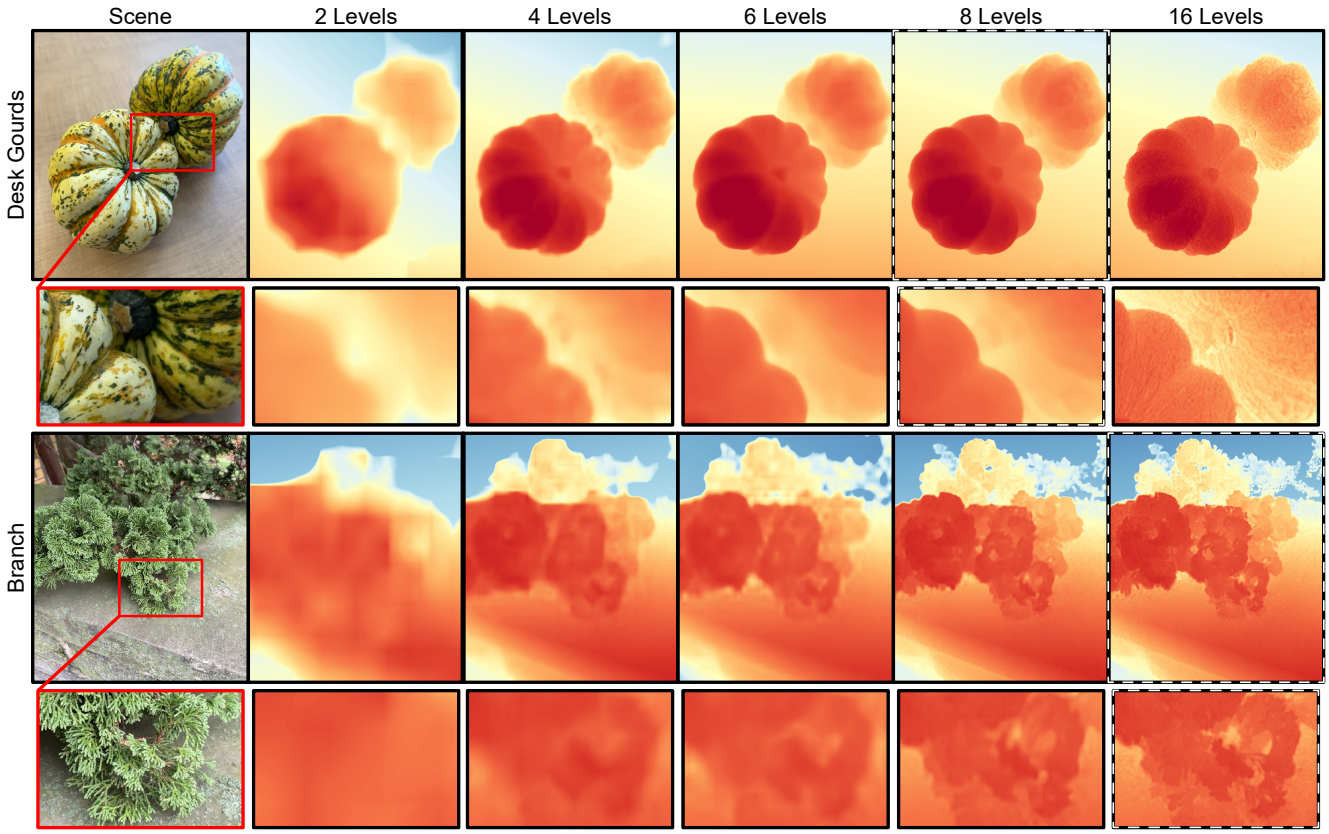

Figure 4. Ablation study on the effect of the number of levels L^{γ_D}, and effective max resolution $N_{max}^{\gamma_D}$, in the multiresolution hash encoding γ_D on reconstruction. Here, given a scale factor of $\sqrt{2}$ between levels, $L^{\gamma_D} = 2, 4, 6, 8, 16$ correspond to $N_{max}^{\gamma_D} = 16, 32, 128, 2048$. The qualitatively best reconstructions are highlighted with a dashed border.

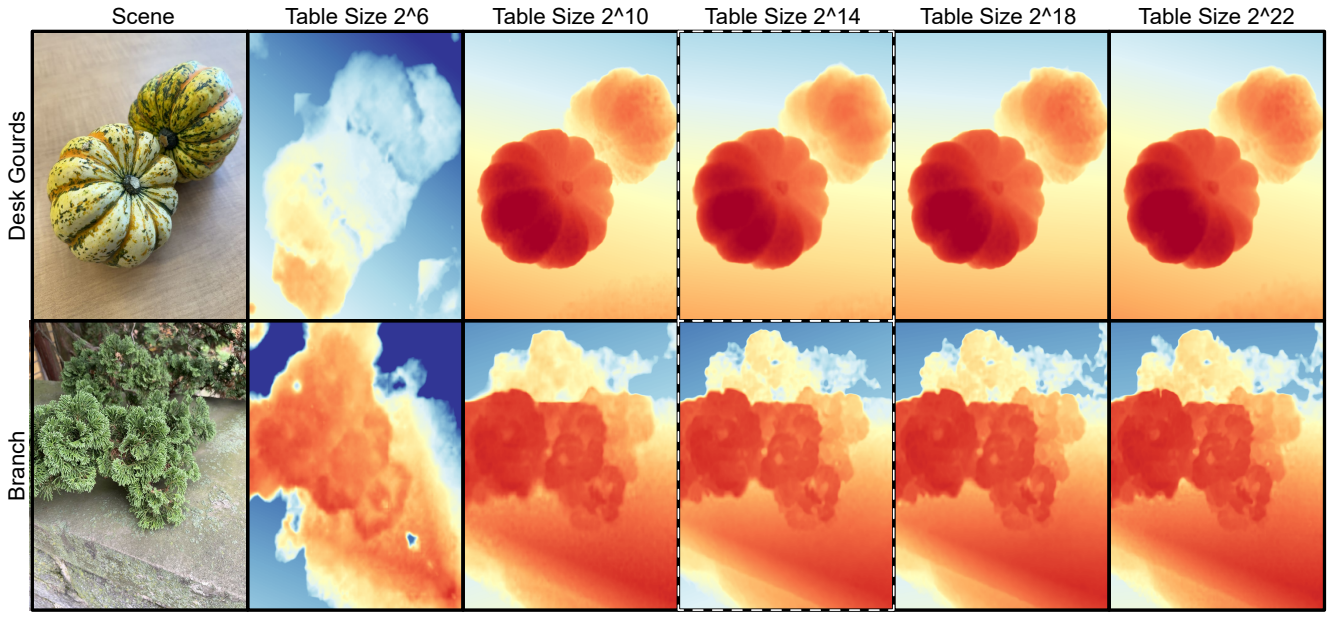

Figure 5. Ablation study on the effect of hash table size T^{γ_D} on reconstruction quality. Selected T^{γ_D} is highlighted with a dashed border.

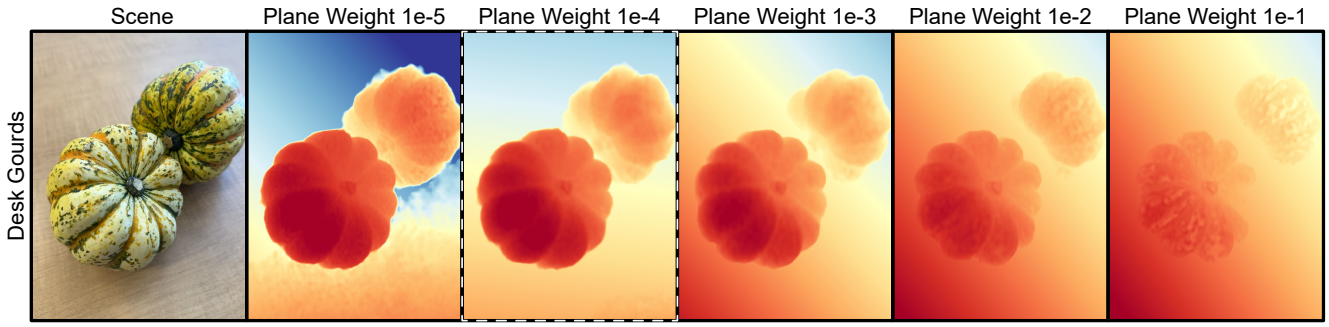

Figure 6. Ablation study on the effects of regularization weight α_{P} on reconstruction quality. Selected α_{P} highlighted with a dashed border.

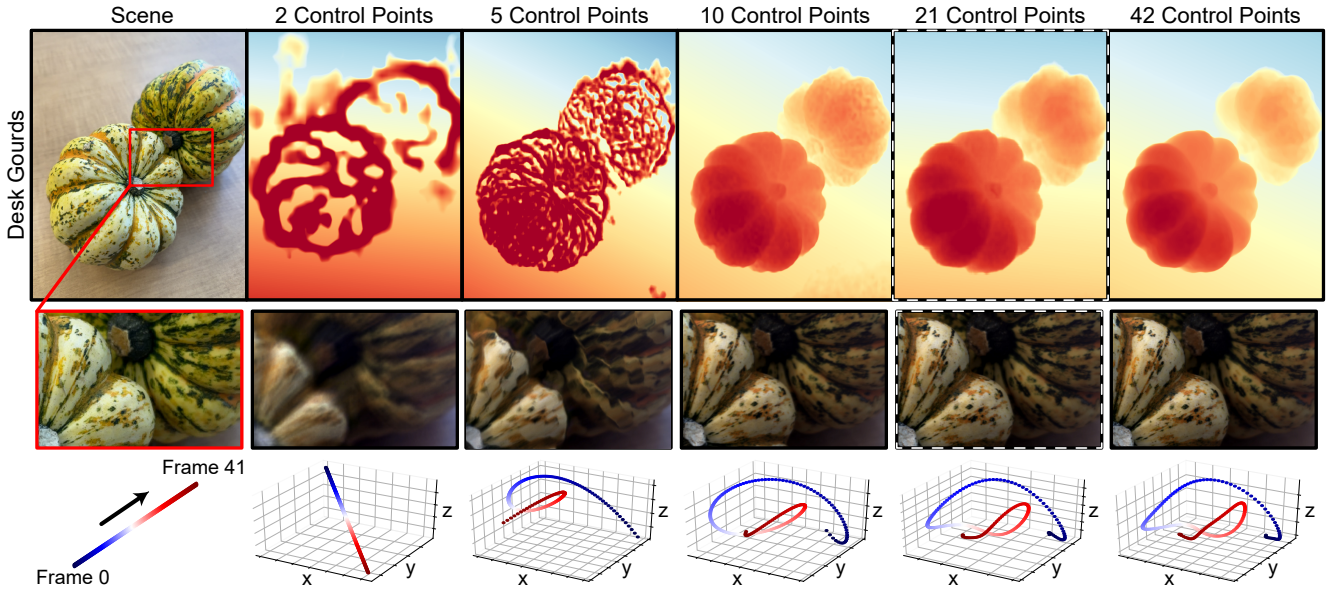

Figure 7. Ablation study on the effect of the number of chosen control points $N_c^{\text{T}} = N_c^{\text{R}}$ on reconstruction quality, with image reconstructions $I(u, v)$ and estimated motion paths plotted below. The selected number of control points is highlighted with a dashed border.

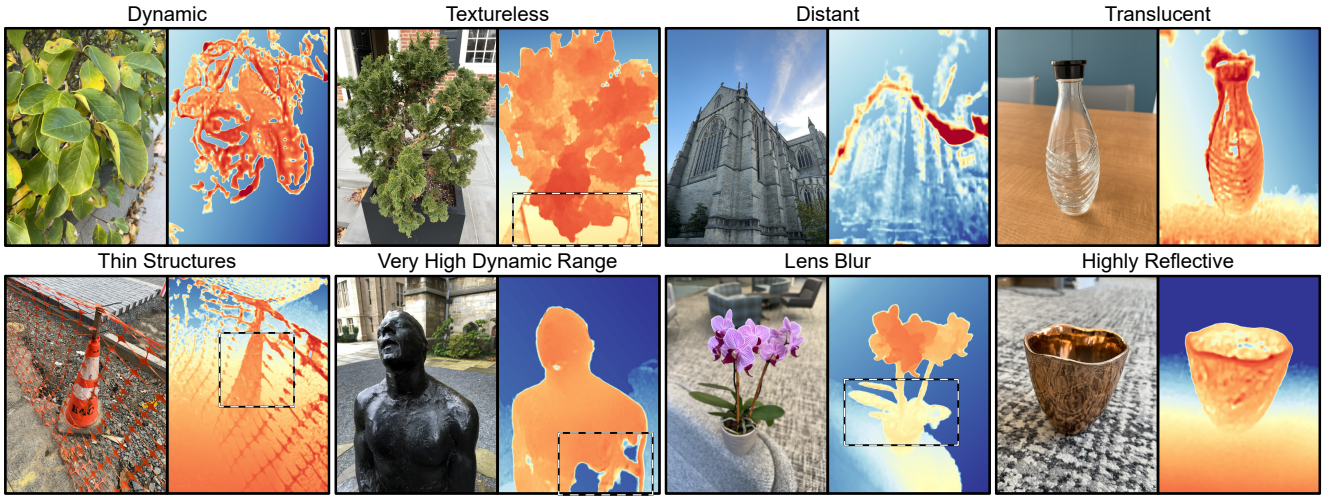

Figure 8. Depth reconstruction results for a set of challenging imaging scenarios. Not visible is the large motion of leaves in the *Dynamic* scene, captured during high wind. Areas of interest are highlighted with a dashed border.

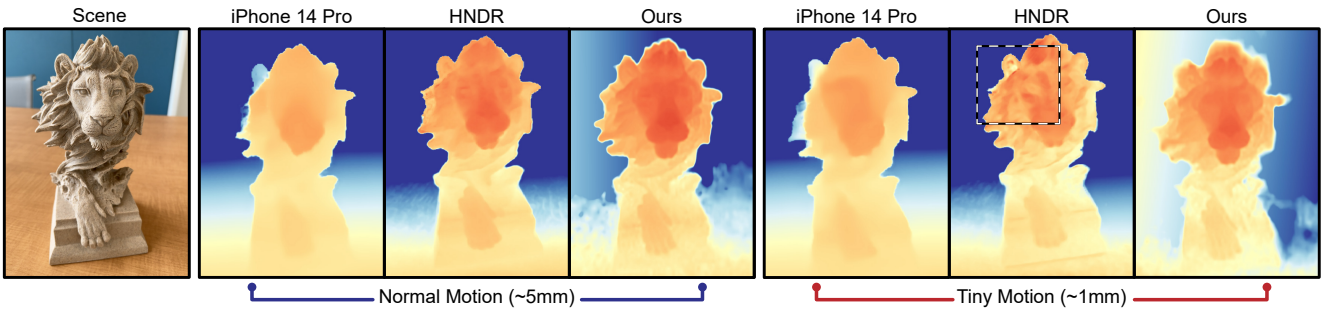

Figure 9. Depth reconstruction results for long-bursts captured with normal (approximately 5 millimeter maximum effective stereo baseline) and minimal (on the scale of a millimeter) hand shake motion. Major depth artifacts are highlighted with a dashed border.

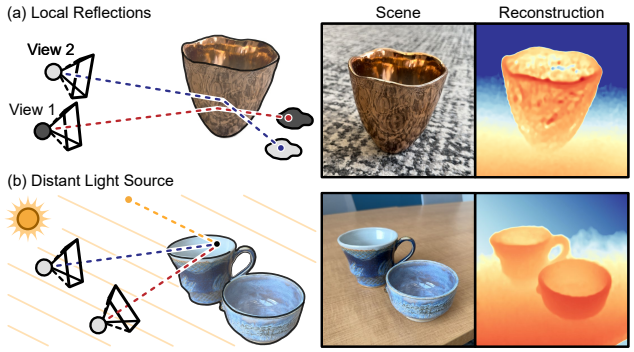

Figure 10. (a) Objects which reflect local scene content, in this example a mirror-finish copper pot reflecting the carpet around it, can completely break view consistency assumptions used for depth reconstruction. Even small view angle changes result in light paths which sample completely inconsistent colors in the surrounding environment. (b) In contrast, objects with specularities caused by a light source at effectively infinity, in this example polished ceramic reflecting sunlight, do not fully break view consistency.

not taking a breath and is rigidly holding the phone with two hands close to their body – this motion can be as small as a millimeter. Illustrated in Fig. 9, we see how our end–to-end camera pose estimation still converges in the minimal baseline setting, and how we are able to produce useable – albeit blurrier – depth estimates. This is in contrast to HNDR [2], which uses the imperfect ARKit pose estimates for reprojection and produces major artifacts because of it, mapping incorrect pixel matches to spurious depths solutions. This demonstrates the value of continuous pose refinement, as mobile SLAM algorithms and COLMAP [6] *do not produce ground truth poses*.

Additional Results. Fig. 11 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 12, 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. [2] 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

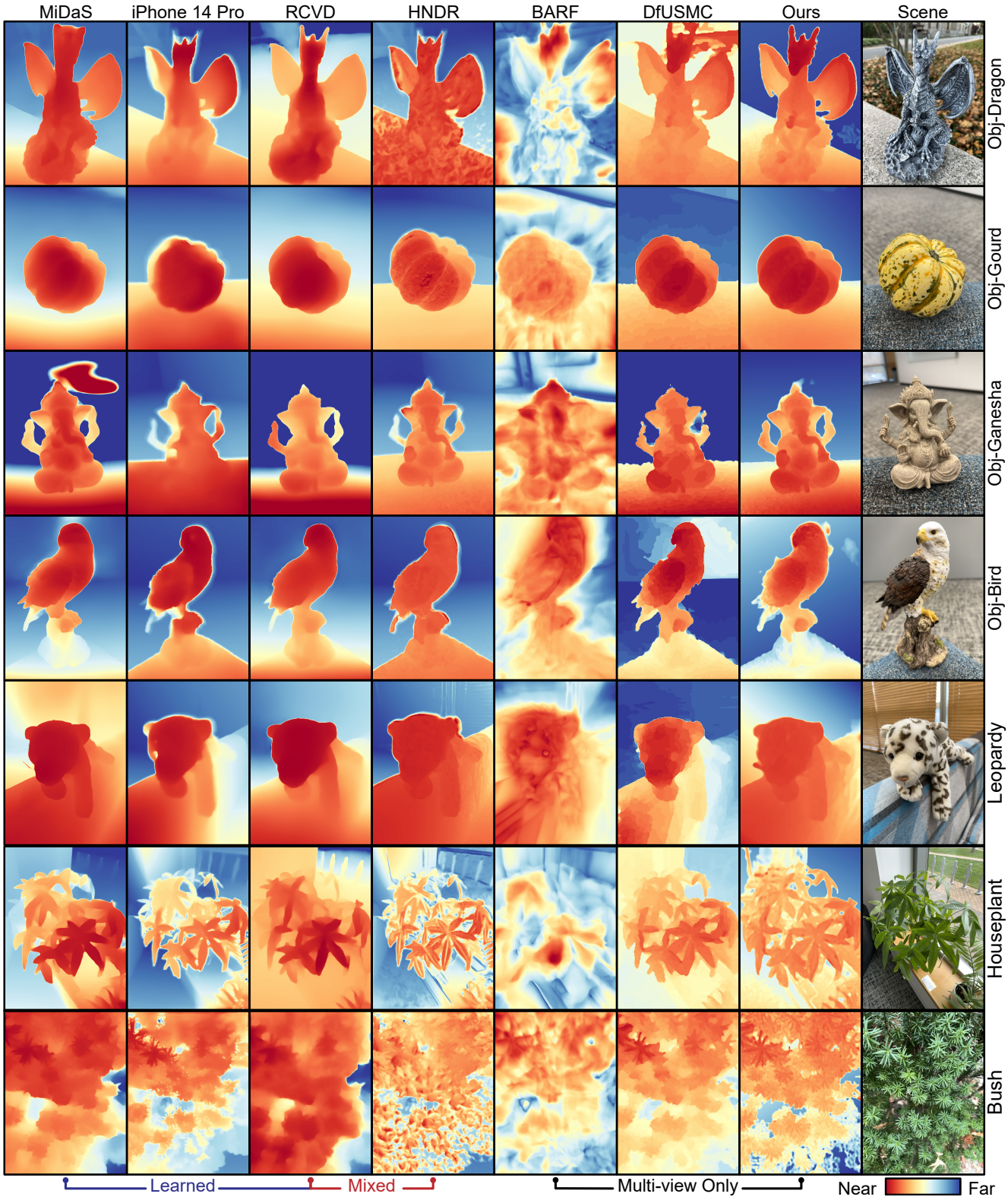

Figure 11. Reconstruction on 7 additional scenes for our method and a mix of learned, purely multi-view, and mixed depth estimation methods. Given the mix of depth representations, results are re-scaled by minimizing relative mean square error.

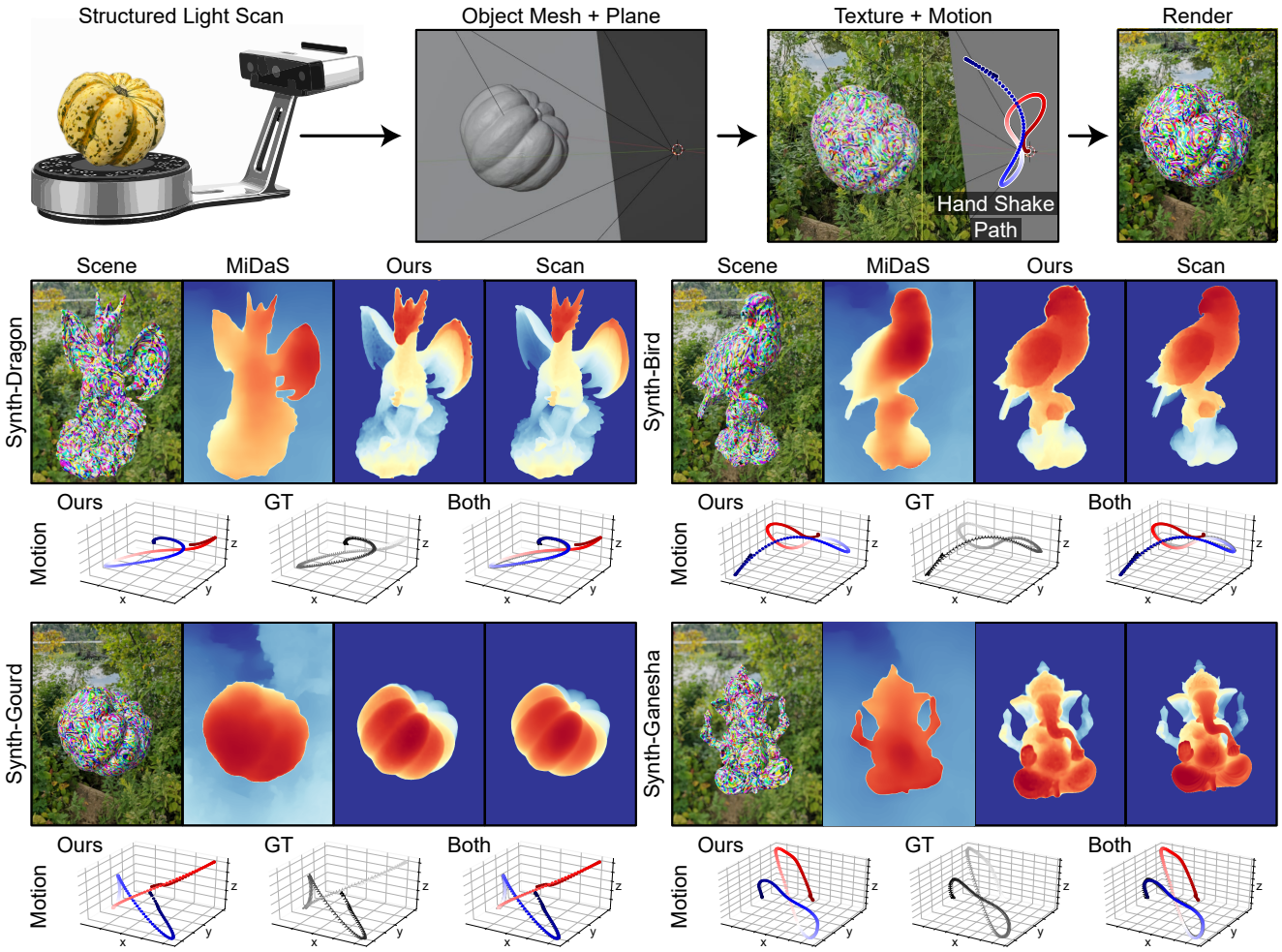

Figure 12. Depth reconstruction and motion estimation results for a set of simulated textured objects with realistic hand-tremor motion. Motion estimates are re-normalized and overlaid to demonstrate the accuracy in estimated camera trajectory to ground truth data.

of the *Synth-Ganesha*. This validates that even with small camera motion, given sufficient image texture we converge on geometrically correct solutions. In Fig. 12 we also see how the camera motion estimates converge close to ground truth as our method jointly refines depth and camera trajectory estimates during training.

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. 13 (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. 13 (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. 13 (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. 13 (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.

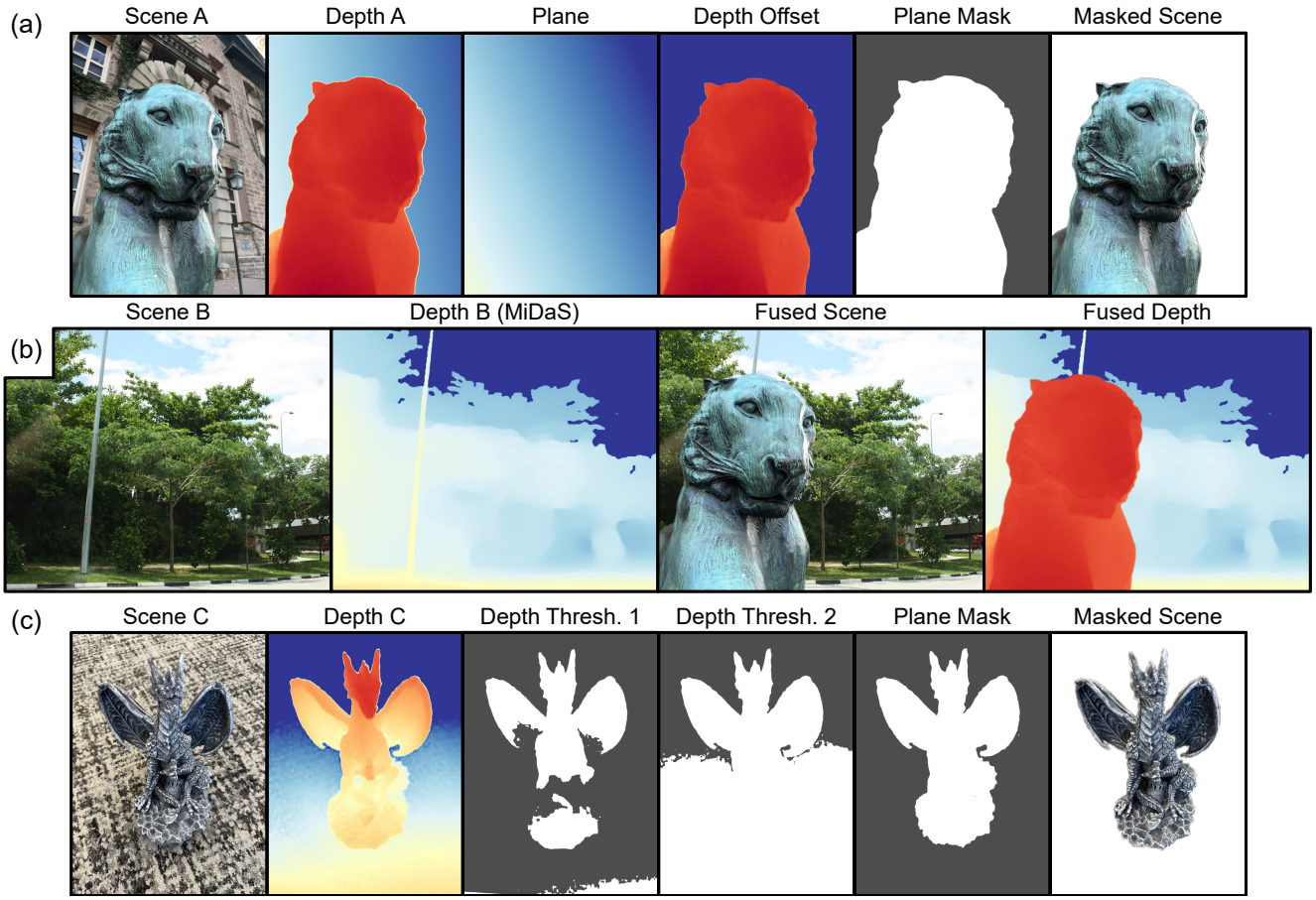

Figure 13. Image and Depth Matting. Example of scene editing enabled by our plane plus offset forward model. We can (a) threshold the depth offset component $d(u, v) - d_{\text{P}}$ to recover a mask of the object in focus and then (b) superimpose it over a new scene. (c) This works even for visually ambiguous scenes where simple depth thresholding fails.

References

[1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984. 2

[2] Ilya Chugunov, Yuxuan Zhang, Zhihao Xia, Xuaner Zhang, Jiawen Chen, and Felix Heide. The implicit values of a good hand shake: Handheld multi-frame neural depth refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2852–2862, 2022. 2, 3, 6

[3] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *arXiv preprint arXiv:2210.13445*, 2022. 2

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[5] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022. 3

[6] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 6

[7] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. DeMoN: Depth and motion network for learning monocular stereo. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[8] Yalin Xiong and Steven A Shafer. Depth from focusing and defocusing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 68–73. IEEE, 1993. 3