Neural Light Spheres for Implicit Image Stitching and View Synthesis

ILYA CHUGUNOV*, Princeton University, USA
AMOGH JOSHI, Princeton University, USA
KIRAN MURTHY, Google Inc., USA
FRANCOIS BLEIBEL, Google Inc., USA
FELIX HEIDE, Princeton University, USA

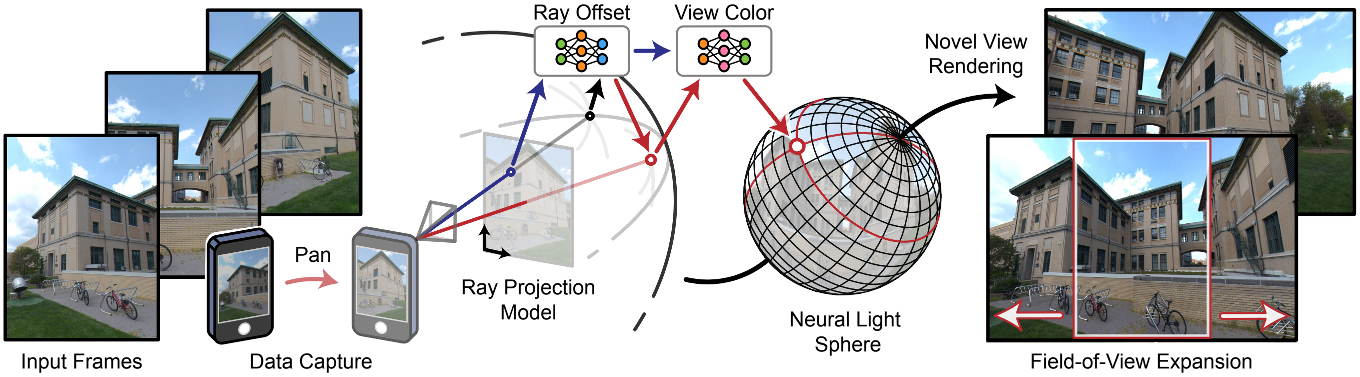


Fig. 1. Fit during test-time directly to an input panoramic video capture, with no pre-processing steps, our neural light sphere model produces a parallax, lighting, and motion-tolerant reconstruction of the scene. Placing a virtual camera into the sphere, we can generate high-quality wide field-of-view renders of the environment, turning what would otherwise be a static panorama into an interactive viewing experience.

Challenging to capture, and challenging to display on a cellphone screen, the panorama paradoxically remains both a staple and underused feature of modern mobile camera applications. In this work we address both of these challenges with a spherical neural light field model for implicit panoramic image stitching and re-rendering; able to accommodate for depth parallax, view-dependent lighting, and local scene motion and color changes during capture. Fit during test-time to an arbitrary path panoramic video capture – vertical, horizontal, random-walk – these neural light spheres jointly estimate the camera path and a high-resolution scene reconstruction to produce novel wide field-of-view projections of the environment. Our single-layer model avoids expensive volumetric sampling, and decomposes the scene into compact view-dependent ray offset and color components, with a total model size of 80 MB per scene, and real-time (50 FPS) rendering at 1080p resolution. We demonstrate improved reconstruction quality over traditional image stitching and radiance field methods, with significantly higher tolerance to scene motion and non-ideal capture settings.

CCS Concepts: • **Computing methodologies** → **Computational photography**; **Computer vision representations**.

Additional Key Words and Phrases: Neural Fields, Panorama, Image Stitching, View Synthesis

ACM Reference Format:
Ilya Chugunov, Amogh Joshi, Kiran Murthy, Francois Bleibel, and Felix Heide. 2024. Neural Light Spheres for Implicit Image Stitching and View Synthesis. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3680528.3687660

1 INTRODUCTION

The *panorama* of the 19th century was typically a commissioned collection of paintings in a cylindrical arrangement with a dedicated viewing platform to maximize observers' immersion in the work [Trumpener and Barringer 2020]. The digital panorama of the 21st century is typically a long rectangle left un-shared – or un-*viewed* – in the storage space of the cellphone used to capture it. Yet, arguably the most common form of digital panorama might be the one that is un-*taken*, where the user decides that the hassle of acquisition – e.g., slowly and carefully sweeping the camera in a level arc across a scene – is not worth the final product.

To address this imbalance, we can simplify acquisition, increase the appeal of the final product, or (preferably) do both. Moving from cylindrical warping [Szeliski and Shum 1997] and seam matching [Zomet et al. 2006] approaches to more parallax-tolerant image stitching processes [Zaragoza et al. 2013; Zhang and Liu 2014] allows the photographer to take a less restricted camera path and still produce a high-quality panorama. However, the end result remains a single static image. Work on multi-layer depth panoramas [Lin et al. 2020; Zheng et al. 2007] and panoramic mesh reconstruction [Hedman et al. 2017] offer a more interactive experience than

[*] Part of this work was done during an internship at Google Inc.

Authors' addresses: Ilya Chugunov, chugunov@princeton.edu, Princeton University, USA; Amogh Joshi, aj0699@princeton.edu, Princeton University, USA; Kiran Murthy, murthykk@google.com, Google Inc., USA; Francois Bleibel, fbleibel@google.com, Google Inc., USA; Felix Heide, Princeton University, USA, fheide@princeton.edu.

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687660

a traditional panorama, able to use parallax information to render novel views of the scene. The recent explosion in radiance field methods [Kerbl et al. 2023; Mildenhall et al. 2021] can be seen as an evolution of "interactive panoramas", with a line of connected works from image-based rendering [Chen and Williams 1993] to direct view synthesis [Flynn et al. 2016] and hybrid 3D and image feature approaches [Sitzmann et al. 2019]. Neural radiance field (NeRF) methods can produce fast [Muller et al. 2022] scene reconstructions which model for both parallax and view-dependent lighting effects with high visual quality [Barron et al. 2023] and from unstructured and unknown poses [Lin et al. 2021]. However, outward or front-facing panoramas present a major challenge for these volumetric representations, as large parts of the scene are only observed for a few frames before falling out of view, turning scene reconstruction into a collection of sparse view problems [Niemeyer et al. 2022].

In this work we explore a compact neural light field [Attal et al. 2022] model for panoramic image stitching and view synthesis; capable of encoding depth parallax, view-dependent lighting, and local scene motion and color changes. We represent the scene as a color-on-a-sphere model decomposed into two components: a view-dependent ray offset model for parallax, lens distortion, and smooth motion; and a view-dependent color model for occluded content, reflections, refraction, and color changes. Taking as input an arbitrary path panorama – vertical, horizontal, random-walk – we fit our model at *test-time* to jointly estimate the camera path, and produce a high-resolution stitched representation of the scene. We demonstrate how this model enables geometrically consistent field-of-view expansion, transforming portrait-mode panoramas into immersive, explorable wide-view renders.

Specifically, we make the following contributions:

- A compact and efficient (80 MB model size per scene, 50 FPS rendering at 1080p resolution) two-stage neural light sphere model of panoramic photography.
- Validation of panoramic image stitching and view synthesis performance under varying imaging settings, including low-light conditions, with comparisons to traditional image stitching and radiance field approaches.
- An Android-based data collection tool for streaming and recording full-resolution RAW image arrays, camera and system metadata, and on-board device measurements such as gyroscope and accelerometer values.
- A diverse collection of 50 indoor and outdoor handheld panoramic scenes recorded from all three on-device cameras with full 10-bit color depth, 12-megapixel resolution.

We make our code, data, and data collection app available open-source on our project website: light.princeton.edu/NeuLS

2 RELATED WORK

Image Stitching. There is a rich history of methods for stitching or *mosaicing* [Burt and Adelson 1983] multiple images into one, with demand for the task long pre-dating the invention of digital photography [Shepherd 1925]. A common approach is to first extract image features, either directly calculated [Brown and Lowe 2007; Lowe 2004] or learned [Sarlin et al. 2020], which are matched to position and warp images together [Gao et al. 2011]. Allowing for image transforms beyond simple homographies [Hartley and Zisserman 2003] can allow for parallax-tolerant image warping and stitching [Shum and Szeliski 2002; Zhang and Liu 2014], reducing blur from pixel disparity between views. Seam-carving approaches dynamically adjust the stitching boundaries to better match visual features [Agarwala et al. 2004; Gao et al. 2013], helping to avoid artifacts from mismatched content on image boundaries. Inspired by local deformation image stitching [Zaragoza et al. 2013] and panoramic video texture [Agarwala et al. 2005] work, we develop a neural field model which can accommodate for both parallax and scene motion during reconstruction. However, rather than use sparse pre-computed features and break the reconstruction pipeline into multiple discrete steps, we leverage a neural scene representation and fast ray sampling to optimize our model end-to-end over dense pixel-wise photometric loss.

Layered and Depth Panoramas. Concentric mosaic [Shum and He 1999] and layered depth map [Shade et al. 1998] representations offer a compact way to model the effects of parallax and occlusion in a scene. Layered depth panoramas [Zheng et al. 2007] make use of a layered representation to produce an interactive image stitching reconstruction, able to render novel views through trigonometric reprojection. Follow-on work extends this reconstruction to mesh representations [Hedman et al. 2017; Hedman and Kopf 2018] and learned features [Lin et al. 2020], offering improved reconstruction of object surfaces which are otherwise occluded between depth layers. Work in this space often targets VR applications [Attal et al. 2020; Bertel et al. 2020; Lai et al. 2019], as they drive demand for high-quality immersive and interactive user experiences in 3D environments. Also related are video mosaic approaches [Kasten et al. 2021; Rav-Acha et al. 2008], which forgo re-rendering to decompose a video into a direct 2D-to-2D pixel mapping onto a set of editable atlases. In this work, we target reconstructions that can provide an interactive user experience with minimal hardware or camera motion requirements [Bertel et al. 2020], and which are able to tolerate moderate scene motion and color changes.

Light Field Methods. Modeling ray color as a product of three dimensional spatial and angular components, a light field can fully represent effects of depth parallax, reflections, and refraction in a scene [Levoy and Hanrahan 1996; Ng et al. 2005] at the cost of high data, storage, and computational requirements [Wilburn et al. 2005]. Lumigraphs [Gortler et al. 1996] make use of a simpler geometric proxy – e.g., the crossing points of a ray intersecting with two planes – to represent the spatial and angular components of a light field, greatly lowering data and computational requirements for reconstruction and rendering [Chai et al. 2000]. Motivated by recent work in neural light field representations [Attal et al. 2022; Suhail et al. 2022], we develop a compact spherical representation which decomposes the scene into view-dependent ray offset – for effects such as parallax and local motion – and view-dependent color for occlusions and time-dependent content.

Neural Scene Representations. Recent work in neural scene representations, particularly in the area of neural radiance fields (NeRFs) [Barron et al. 2023; Mildenhall et al. 2021], has demonstrated that high quality scene reconstruction can be achieved without pixel

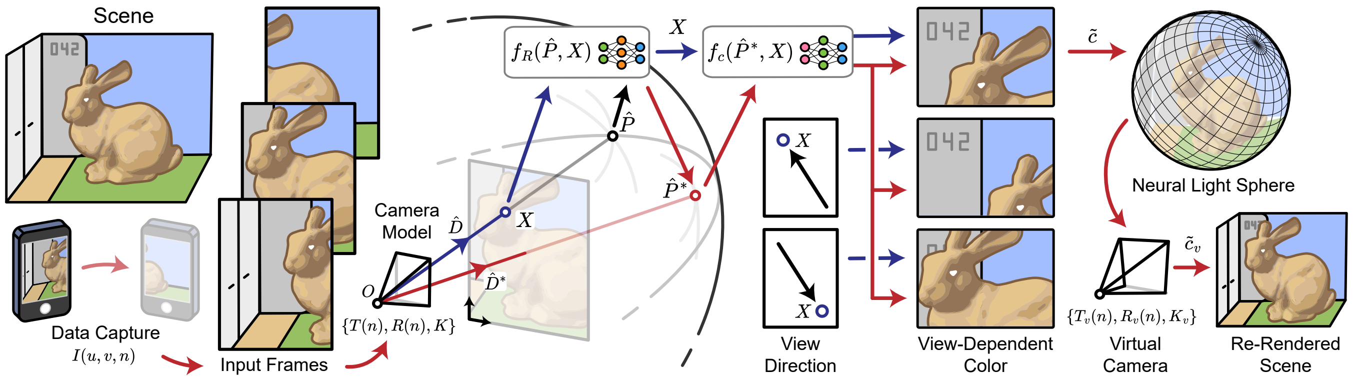


Fig. 2. **Neural Light Sphere Model.** Taking as input panoramic video capture $I(u,v,n)$, we perform backward camera projection from a point $X=(u,v)$ into a spherical hull to estimate an initial intersection point P. Ray offset model $f_R(\hat{P},X)$ then bends this ray to a corrected point $\hat{P}^*$, which is used to sample the view-dependent color model $f_c(\hat{P}^*,X)$. Simulating a new virtual camera with our desired position and FOV, we use this neural light sphere model to re-render the scene to novel views.

arrays, voxel grids, or other explicit backing representations. These approaches train a neural network at *test time* – starting with an untrained network, overfit to a single scene – to map from encoded [Tancik et al. 2020] coordinates to output parameters such as color [Nam et al. 2022], opacity [Martin-Brualla et al. 2021], density [Corona-Figueroa et al. 2022], depth [Chugunov et al. 2023], camera lens parameters [Xian et al. 2023], and surface maps [Morreale et al. 2021]. While they are not neural scene representations, forward projection "Gaussian Splatting" [Kerbl et al. 2023] models have recently exploded in popularity as an alternative to NeRF scene representations, offering increased rendering speed by avoiding costly volume sampling operations. However, outward panoramic captures with largely rotational motion present a challenge for these methods, which rely on large view disparity to localize content in 3D space. We instead propose a view-dependent ray offset and color model to reconstruct local parallax and view-dependent effects from minimal view disparity. By embedding this representation on a spherical surface, we also substitute costly NeRF volume sampling with efficient ray-sphere crossings, resulting in a compact 80 MB model capable of real-time 1920x1080px rendering at 50 FPS.

3 NEURAL LIGHT SPHERE RECONSTRUCTION

In this section we describe our proposed neural light sphere model for implicit image stitching and re-rendering. We begin with an overview of our backward projection model for unstructured panoramic captures. We then discuss the neural field representations backing this model, its loss and training procedure, how we collect scene data for reconstruction, and implementation details.

3.1 Projective Model of Panoramic Imaging

In this work, we adopt a spherical backward projection model [Szeliski et al. 2007] for our scene representation. That is, we model each image in the input video as the product of rays originating at the camera center intersecting with the inner surface of a sphere. To simplify notation, we outline this process for a single ray below, illustrated in Fig. 2, and later generalize to batches of rays. Let

$$c = [\mathrm{R}, \mathrm{G}, \mathrm{B}]^\top = I(u,v,n) \tag{1}$$

be a colored point sampled at image coordinates $u,v \in [0,1]$ from a frame $n \in [0, N{-}1]$ in a video $I(u,v,n)$, where N is the total number of captured video frames. To project this point to a camera ray, we introduce camera rotation $R(n)$ and translation $T(n)$ models

$$T(n) = \mathbf{T}_n, \quad R(n) = \mathrm{rot}(\eta_{\mathrm{R}}\mathbf{R}_n)\mathbf{G}_n$$

$$\mathbf{T}_n = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \ \mathbf{R}_n = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}, \ \mathrm{rot}(\mathbf{R}_n) = \begin{bmatrix} 1 & -r_z & r_y \\ r_z & 1 & -r_x \\ -r_y & r_x & 1 \end{bmatrix}. \tag{2}$$

Here, we model translation for frame n as three dimensional motion, initialized at zero. $R(n)$ is a small-angle approximation [Boas 2006] offset $\mathbf{R}_n$ to device rotation $\mathbf{G}_n$ recorded from the phone onboard gyroscope, weighted by η_{R}. With calibrated intrinsics matrix K, sourced from device camera metadata, we project the point at u,v sampled from frame n to a ray with origin O and direction D as

$$O = \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} = T(n), \qquad D = \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = R(t)K^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{3}$$

We normalize the direction vector $\hat{D} = D/\|D\|$ to simplify reprojection steps. Next, we define our image model to lie on the surface of a sphere, and calculate its intersection point P with this ray as

$$\hat{P} = P/\|P\|, \quad P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = O + t\hat{D}$$

$$t = -\left(O \cdot \hat{D}\right) + \sqrt{(O \cdot \hat{D})^2 - (\|O\|^2 - 1)}, \tag{4}$$

assuming a sphere of radius 1, centered at $[0,0,0]^\top$, with the ray originating within its radius ($\|O\|^2 < 1$). However, as this sphere model, in general, does not match the true scene geometry, we introduce a ray offset model $f_{\mathrm{R}}(\hat{P},X)$ to offset the ray direction as

$$\hat{D}^* = D^*/\left\|D^*\right\|, \quad D^* = \mathrm{rot}\left(\mathbf{R} = f_{\mathrm{R}}(\hat{P},X)\right)\hat{D}, \tag{5}$$

where $X = [u,v]^\top$ is the ray's originating image coordinates, and $\mathrm{rot}(\mathbf{R})$ is the small-angle rotation model from Eq. 2. We can observe that this model generalizes effects such as parallax (deflecting rays

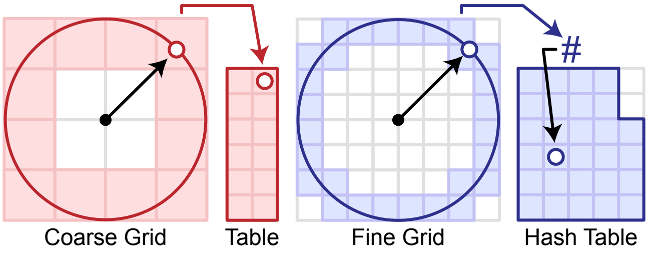


Fig. 3. **Hash Grid Spheres.** In this 2D example we can observe how, for points on a circle, the number of accessed elements in the backing grid roughly doubles for a squaring of grid elements. Given an efficient mapping from grid location to element – e.g., hash table lookup – this forms a compact representation even at high resolutions, where storing a dense grid would be computationally intractable.

as a function of position via $\hat{P}$) and lens distortion (deflecting rays as a function of their angle relative to the camera center via X). With this corrected ray $(O, \hat{D}^*)$, we re-sample our sphere via Eq. 14 to generate a new intersection point $\hat{P}^*$. To map this point to estimated scene color $\tilde{c}$, we introduce a view-dependent color model f_{C}, where

$$\tilde{c} = [\tilde{\mathrm{R}}, \tilde{\mathrm{G}}, \tilde{\mathrm{B}}]^\top = f_{\mathrm{C}}(\hat{P}^*, X). \tag{6}$$

This model takes as input the camera coordinate X, which allows for modeling of view-dependent effects such as occlusions, reflections, motion, and generated content (e.g., flashing lights), and maps it together with the ray intersection on the sphere $\hat{P}^*$ to an output estimated RGB value $\tilde{c}$. To generate novel views we take as input virtual camera intrinsics K_v, translation $T_v(n)$ and rotation $R_v(n)$, and repeat Eq. (2)–(16) with these new parameters to generate a colored point $\tilde{c}_v$.

3.2 Neural Field Representations

In the section above, we introduce, but do not *define*, our two core models: f_{R} for ray offset, and f_c for view-dependent color estimation. Much of the diversity in image stitching and view synthesis approaches can be seen as design choices for these models. For example, f_{R} could be a layered depth model [Shade et al. 1998] or cylindrical projection [McMillan and Bishop 1995], and f_{C} could be an explicit color blending [Buehler et al. 2001] or implicit radiance field [Mildenhall et al. 2021], each with tradeoffs in representation power, extrapolation, and input data requirements. With this in mind, we aim to design f_{R} and f_{C} to produce a system which is:

(1) **Compact**: such that that model is simple to train and has low memory and disk space usage. Thus we minimize the number of components, networks, loss and regularization functions, and avoid pre-processing steps (such as COLMAP [Schonberger and Frahm 2016]).
(2) **Robust**: able to reconstruct a wide range of capture settings (indoor, outdoor, night-time), capture paths, and scene dynamics (e.g., moving clouds, blinking lights). Failing gracefully for hard-to-model effects, with localized reconstruction errors.

Neural scene representations, particularly with high-level hardware-optimized implementations [Muller et al. 2022], offer compelling solutions to this design challenge. By implicitly representing the scene in the weights of a multi-layer perceptron (MLP) [Hornik et al. 1989], we can effectively turn data storage and retrieval into a component of our inverse imaging model. Correspondingly, we represent ray offset f_{R} as

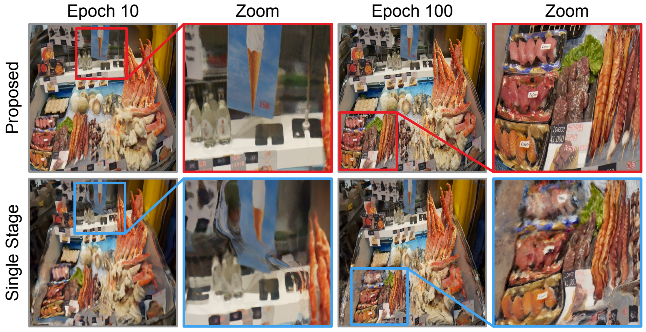


Fig. 4. **Two Stage Training.** Breaking training into two stages allows the camera pose and static image model to first fit an approximation of the scene before view-dependent effects are introduced via h_{R} and h_{D}. This helps avoid artifacts during early training, like the discontinuities around the sign in the *Single Stage* example, which result in poor final reconstruction quality.

$$f_{\mathrm{R}}(\hat{P}, X) = h_{\mathrm{R}}(\gamma_1(\hat{P}) \oplus \gamma_1(X);\, \theta_{\mathrm{R}}), \tag{7}$$

where $\oplus$ denotes concatenation. Here, h_{R} is an MLP with learned weights θ_{R}, and γ_1 is the multi-resolution hash grid encoding from [Muller et al. 2022], sampled with 3D normalized ray intersection $\hat{P}$ and 2D camera coordinate X. During training, h_{R} learns a mapping between these encoded vectors and the offset applied to $\hat{D} \to \hat{D}^*$. We similarly construct the view-dependent color model f_{C} as

$$f_{\mathrm{C}}(\hat{P}^*, X) = h_{\mathrm{C}}\left(h_{\mathrm{P}}(\gamma_2(\hat{P}^*);\, \theta_{\mathrm{P}}) + h_{\mathrm{D}}(\gamma_1(X);\, \theta_{\mathrm{D}});\, \theta_{\mathrm{C}}\right), \tag{8}$$

The network h_{D} takes as input camera coordinate X and outputs a vector encoding of view direction; network h_{P} similarly encodes the corrected position of the sphere crossing. This combined encoding is then mapped to color via h_{C}. Of note is that γ_2, the multi-resolution hash encoding applied to $\hat{P}$, and γ_1, the encoding applied to $\hat{P}^*$, operate in 3D world space on the surface of the unit sphere. That is, *we never convert intersections to spherical coordinates*, and avoid the associated non-linear projection [Zelnik-Manor et al. 2005] and singularity problems. While it would be exceedingly inefficient to store a sphere in a dense representation of sufficient resolution for high-quality image synthesis (e.g., 4000^3 voxels for 12-megapixels images, the majority of which would be empty), this is made possible thanks to the hash-grid backing of γ. Illustrated in Fig. 3, as the majority of the grid locations inside in the unit cube are never sampled, since they do not intersect with the unit sphere's surface, the corresponding stored entries in γ are never queried. Thus the size of the hash table for γ – which determines its latency, memory usage, and storage requirements – can be on the order of magnitude of the sphere's surface area rather than its volume.

3.3 Loss and Training Procedure

With the rotation model $R(n)$ initialized with the device's onboard gyroscope measurements, and the translation model $T(n)$ initialized as all zeroes, we train the networks $\{h_{\mathrm{R}}, h_{\mathrm{C}}, h_{\mathrm{P}}, h_{\mathrm{D}}\}$ from scratch via

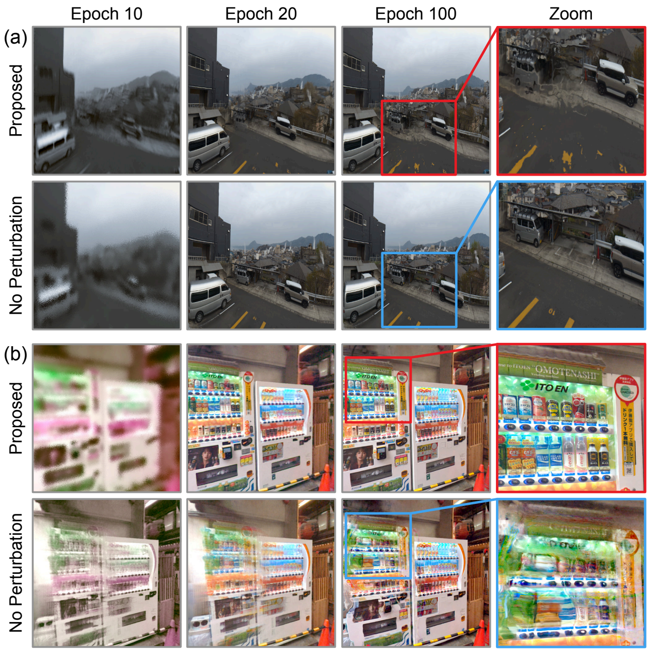


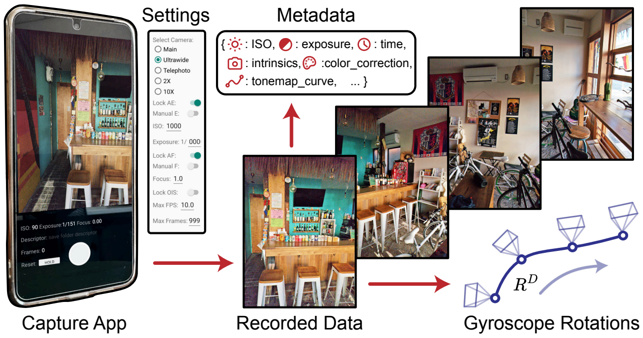
Fig. 5. **Ray Perturbations.** By applying small perturbations to ray origins O we are able to avoid hard-to-escape local minima solutions during early training epochs. In (a) we see how for the road, a region with low image texture, the *No Perturbation* example duplicates content; creating two copies of the #10 parking spot. In (b) we see how for repeated textures, perturbations can also help avoid "crunching" content in early training, where the repeated cans in the vending machine are accidentally aligned on top of each other.

stochastic gradient descent to fit an input scene. We break training into two stages: in the first, we freeze the ray offset and view-dependent color networks $h_\mathrm{R}, h_\mathrm{D}$ to allow the model to learn initial camera pose estimates and spherical color map, and in the second stage we unlock all networks to let them jointly continue training. Illustrated in Fig. 4, this helps prevent image artifacts caused by $h_\mathrm{R}, h_\mathrm{D}$ from accumulating during early training, where it is uncertain if parts of the scene are undergoing view-dependent color changes or simply stereo parallax. A similar problem also occurs for training the sphere color networks $h_\mathrm{C}, h_\mathrm{P}$, where the multi-resolution hash encoding γ allows the network to fit image content *undesirably* fast. This leads to artifacts, as seen in Fig. 5, where the image model learns duplicated or overlapping content faster than the motion model can correct for. We find that an effective and computationally inexpensive way of combating this behavior, shown in Eq. (9), is to add small perturbations to rays generated via Eq. (3) as

$$\tilde{O} = O + \eta_p \mathcal{N}(0,1), \tag{9}$$

where $\mathcal{N}(0,1)$ is zero-mean standard Gaussian noise. The weight term η_p is gradually decayed to zero over the first stage of training. Similar to prior work [Chugunov et al. 2023; Li et al. 2023] we also mask the highest frequency grids in γ_1 and γ_2 to reduce the amount of accumulated noise during early training.

Given linear RAW inputs, we find L_1 to be an effective training loss, particularly for high noise reconstruction where zero-mean Gaussian read noise [Brooks et al. 2019] can be averaged out:



Select Camera:
○ Main
● Ultrawide
○ Telephoto
○ 2X
○ 10X
Lock AE:
Manual E:
ISO: 1000
Exposure: 1/ 000
Lock AF:
Manual F:
Focus: 1.0
Lock OIS:
Max FPS: 10.0
Max Frames: 999

ISO: 90 Exposure:1/151 Focus: 0.00
Descriptor: save folder descriptor
Frames: 0
Reset: HOLD

Fig. 6. **Data Capture.** We develop an open-source Android-based mobile application to facilitate in-the-wild capture of scenes. The app's settings allow for camera selection (main, ultrawide, or telephoto) and to either use the device's auto-focus and auto-exposure features for capture, or set their respective values. During capture, we record full resolution Bayer RAW images, device accelerometer and gyroscope measurements, and all exposed camera and frame metadata including: ISO, exposure, timestamps, camera intrinsics, and color and tone correction values.

$$\mathcal{L} = |c - \tilde{c}|. \tag{10}$$

We find that, with careful selection of encoding parameters for γ_1 and γ_2, *no additional explicit regularization penalties are required* to constrain scene reconstruction [Chugunov et al. 2024].

3.4 Data Collection

To record in-the-wild panorama video captures in unknown imaging conditions – ranging from broad daylight to night-time photography – we developed an Android-based data capture application, illustrated in Fig. 6. The app records a stream of RAW images along with metadata, enabling us to leverage linear sensor data for noise-robust reconstruction. While there exist other RAW video and image recording apps, we found they were paid and closed-source, missing desired functionality (e.g., specifying ISO, exposure, and recording frame-rate), and/or failed to record desired data (e.g., gyroscope measurements). In contrast, our app records full-resolution full bit-depth RAW images at the hardware maximum of 30 frames per second, accelerometer and gyroscope measurements, and nearly all camera and image metadata exposed by the Android APIs – a list of which is included in the supplementary material. We make this app available open-source at: github.com/Ilya-Muromets/Pani

We used a handheld Google Pixel 8 Pro cellphone to record a set of 50 scenes, a selection of which are presented in Fig. 7, which cover a wide span of both imaging settings and capture paths. These include traditional 360° and 180° panoramas, as well as linear horizontal and vertical pans, back-and-forth pans, and random-walk paths. We use the device's auto-exposure settings for recording, with sensor sensitivity varying from ISO ≈ 20 in daylight to ISO ≈ 10,000 for night-time scenes. Though we restrict exposure time to ≤ 1/100s to minimize motion blur during the relatively fast capture process (3-10 seconds depending on the length of the capture path). Recorded image sequences range between 30 and 100 frames depending, and

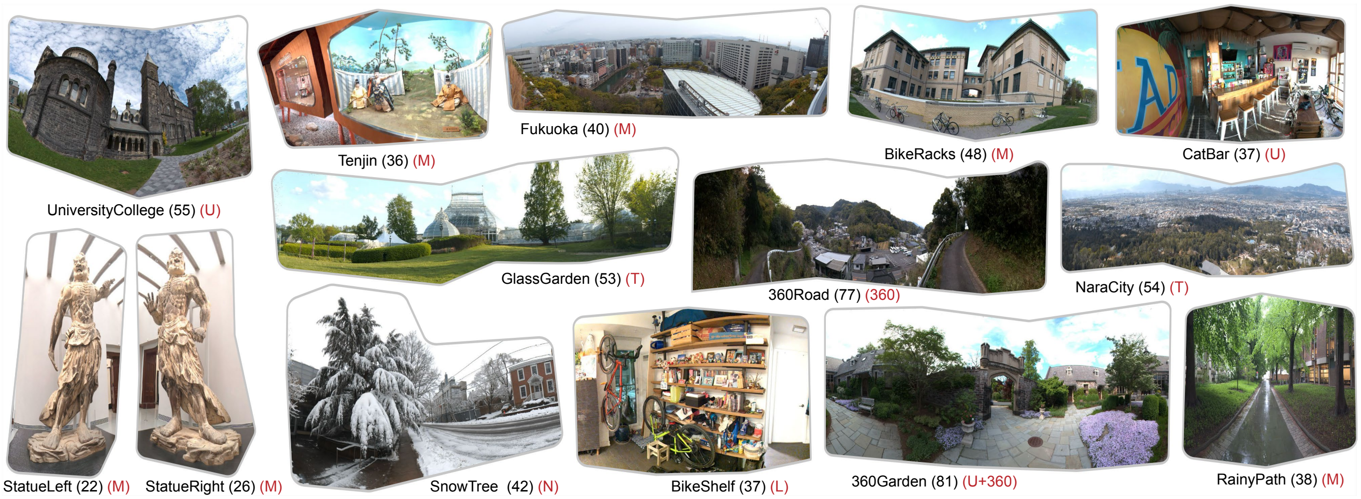


Fig. 7. **Scene Diversity.** Shown above are spherical re-projections of reconstructions for a representative subset of scenes from our collected dataset. These include: (M) 1x main lens, (U) 0.5x ultrawide, (T) 5x telephoto, (L) low-light, (N) non-linear, and (360) full 360 degree captures. Scene titles are formatted as: *Scene Name (Number of Captured Frames in Input)*.

include captures with the main (1x), ultrawide (0.5x), and telephoto (5x) cameras available on the device.

3.5 Implementation Details

We implement our model in PyTorch with the tiny-cuda-nn framework [Müller et al. 2021]. It is trained via stochastic gradient descent with the Adam [Kingma and Ba 2014] optimizer ($\beta = [0.9, 0.99]$, $\epsilon = 10^{-9}$, weight decay 10^{-5}, learning rate 10^{-3}) for 100 epochs, with 200 batches of 2^{18} rays per epoch. Rotation weight $\eta_R = 10^{-3}$. Networks h_R, h_P, h_D are all identical 5 layer 128 hidden unit MLPs; h_C is single 32×3 linear layer to discourage blending of view-dependent and static color. Encoding γ_2 is a 15-level hash grid, with grid resolutions spanning 4 to 3145 by powers of 1.61 for each encoded dimension, and with a backing table size of 2^{19}. To constrain the spatial frequency of the view-dependent color and ray models, encoding γ_1 is a significantly lower-resolution grid, with 8 levels spanning resolutions of 4 to 112. Trained on a single Nvidia RTX 4090 GPU, our method takes approximately *12 minutes* to fit a 40 frame 12-megapixel sequence, though we include results in the supplementary material for how this can be further accelerated to under 30 seconds for generating "preview-quality" reconstructions from 3-megapixel inputs. The image model takes 80 MB of disk space, and can render 1920×1080px frames at 50 FPS. Critical to our core design goals discussed in Sec. 3.2, *all parameters and training procedures are identical for all captures tested in all settings* (daytime, night-time, ultrawide, telephoto, etc.).

4 ASSESSMENT

In this section we compare our method to traditional image stitching and radiance field approaches. We then analyze the contributions of core model components, and confirm its applicability to the reconstruction of night-time scenes with noisy captures. For each scene we render views at 3x their original captured FOV, and include 3 input frames spanning the same FOV.

4.1 Comparisons To Traditional Image Stitching

While a large stitched image canvas is not the primary intended output of our neural light sphere model, as we focus on wide-view video rendering, comparisons to traditional image stitching methods help illustrate the challenges of this setting.

Presented in Fig. 8, we compare our approach to As-Projective-As-Possible (APAP) image stitching [Zaragoza et al. 2013], a robust parallax-tolerant cell-warping approach, and the Microsoft Image Composite Editor (ICE) [Microsoft Research 2015], a polished software suite which performs globally projective warping and seam-blending to hide stitched image borders. APAP is able to warp and average multiple noisy measurements into a cleaner reconstruction, while ICE is restricted to stitching the borders of images together. However, ICE is significantly more resilient to motion-blur, freezing a sharp still frame of moving scene content. Our neural light sphere model offers both of these capabilities, averaging rays for better signal-to-noise ratio in static regions of the scene, while also more faithfully reconstructing dynamic content.

4.2 Comparisons To Radiance Field Approaches

In Fig. 14 we compare our hash-grid based, non-volume-sampling neural light sphere approach to several related radiance field methods including: *K-Planes* [Fridovich-Keil et al. 2023], an explicit representation that also avoids volume sampling by representing the scene as a product of two-dimensional planar features; *Gaussian Splatting* [Kerbl et al. 2023], which also avoids volume sampling through its forward-projection model; *Instant-NGP* [Muller et al. 2022], which makes use of the same multi-resolution hash-grid backing as our approach; and the *Nerfacto* [Tancik et al. 2023] model,

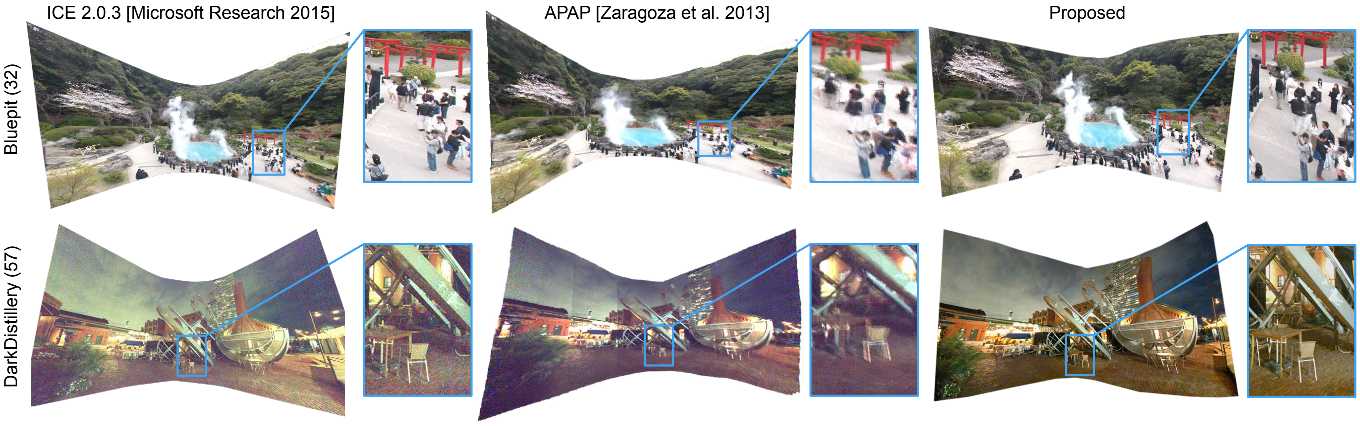

ICE 2.0.3 [Microsoft Research 2015]
APAP [Zaragoza et al. 2013]
Proposed
Bluepit (32)
DarkDistillery (57)

Fig. 8. **Image Stitching Comparisons.** Visualizing rectilinear projections of the stitched panoramas, we see that APAP [Zaragoza et al. 2013] averages multiple frames in *DarkDistillery* to reduce noise, while ICE [Microsoft Research 2015] segments and freezes the motion of pedestrians in *Bluepit*. Our proposed approach aims to do both, averaging multiple rays to reduce noise when possible while also preserving content in areas with local scene motion.

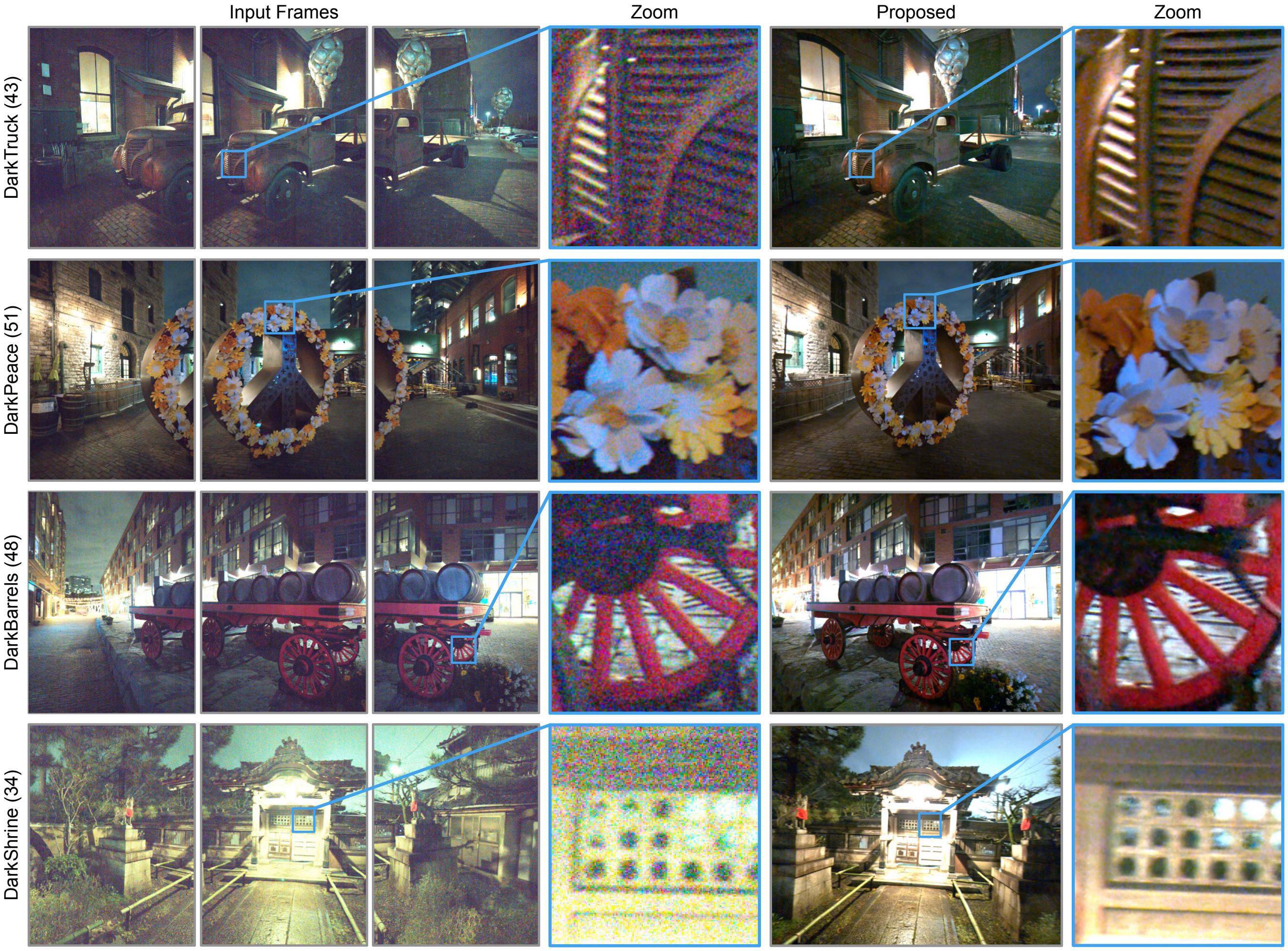


Fig. 9. **Low-light Reconstruction.** Under low-light conditions, with sensor sensitivity at ISO 10,000 and exposure between 1/60s and 1/120s, our proposed model is able to not only successfully reconstruct but also considerably denoise the captured scene. We recommend the reader to view the associated video materials to see the effects of this denoising for interactive rendering.

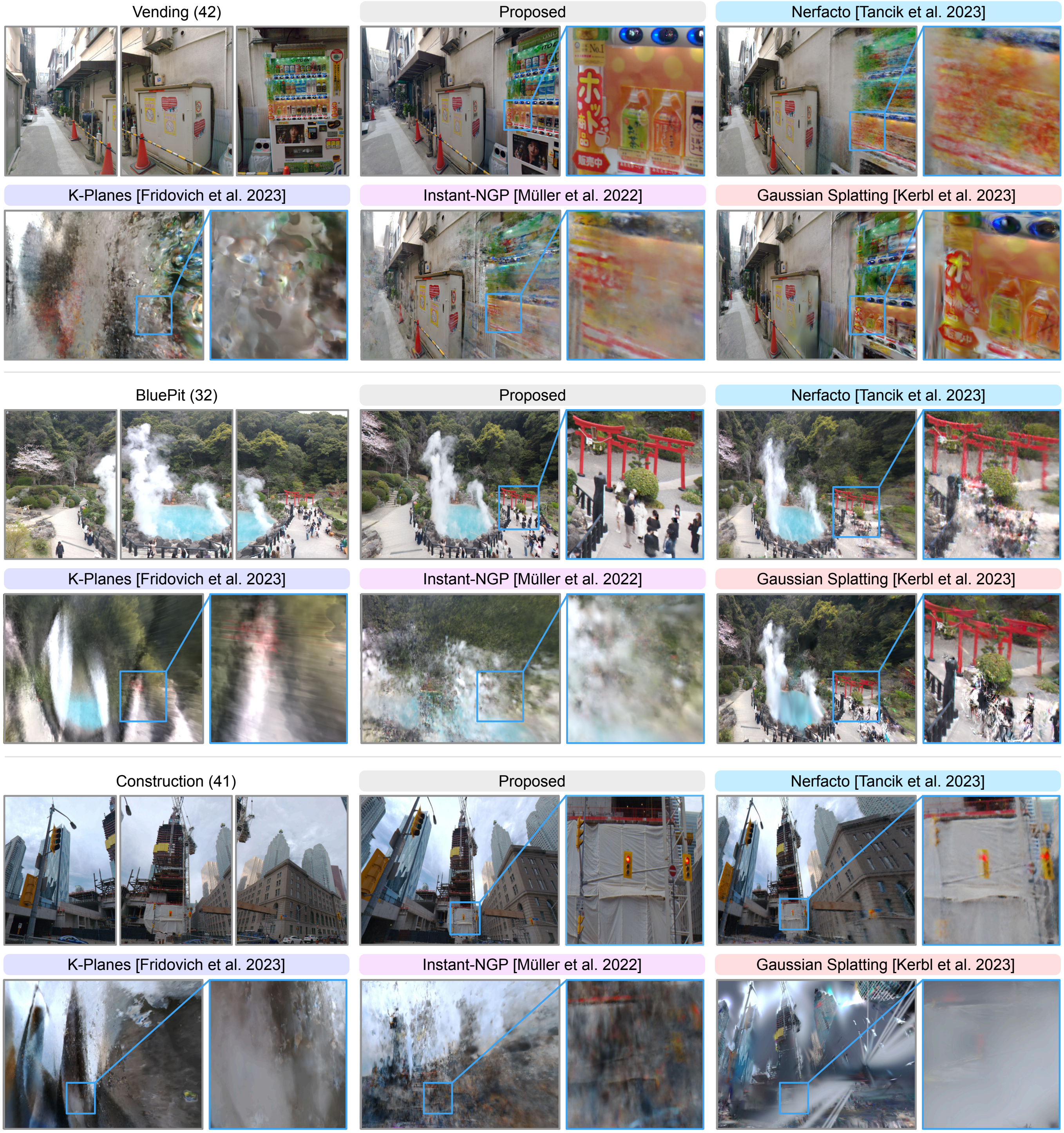

Vending (42)
Proposed
Nerfacto [Tancik et al. 2023]
K-Planes [Fridovich et al. 2023]
Instant-NGP [Müller et al. 2022]
Gaussian Splatting [Kerbl et al. 2023]
BluePit (32)
Proposed
Nerfacto [Tancik et al. 2023]
K-Planes [Fridovich et al. 2023]
Instant-NGP [Müller et al. 2022]
Gaussian Splatting [Kerbl et al. 2023]
Construction (41)
Proposed
Nerfacto [Tancik et al. 2023]
K-Planes [Fridovich et al. 2023]
Instant-NGP [Müller et al. 2022]
Gaussian Splatting [Kerbl et al. 2023]

Fig. 10. **Radiance Field Comparisons.** Compared to radiance field approaches, including other multi-resolution hash-based [Muller et al. 2022; Tancik et al. 2023] and non-volume-integrating [Fridovich-Keil et al. 2023; Kerbl et al. 2023] methods, we achieve significantly higher reconstruction quality over a range of settings. While Gaussian Splatting and Nerfacto are able to successfully overfit the center of most scenes (observed content), when the FOV is expanded to sample rays at wide angles they fail to correctly reconstruct fine images textures like the bottle labels in *Vending*. In contrast, our neural light sphere model is able to reconstruct content in motion, like the pedestrians in *BluePit* and fine parallax effects as in the traffic lights in *Construction*. We recommend the reader to view the associated video materials to better visualize these effects. Scene titles are formatted as: *Scene Name (Number of Captured Frames in Input)*.

a robust combined approach with a hash-grid backing and per-image appearance conditioning. Unfortunately, given the largely rotational motion of panorama captures, even using exhaustive feature matching both COLMAP [Schonberger and Frahm 2016] and HLOC [Sarlin et al. 2019] failed to reconstruct poses for a significant portion of our tested scenes – including virtually all telephoto and ultrawide captures. We thus limit the comparison scenes to ones where COLMAP produced valid poses, and enable camera pose optimization in baseline methods which support it. In contrast, we emphasize that, beyond selecting a directory to load from, *there is no human interaction required between capture and reconstruction for our proposed pipeline.*

Despite tuning feature grid and regularization parameters, we were unable to achieve high-quality reconstructions with *K-Planes*, which appears to produce noisy low-dimensional approximations of the scene. We find that the other baseline methods tend to overfit input captures by placing content a large distance away from the estimated camera position, producing an effect similar to traditional image stitching [Brown and Lowe 2007]. We suspect this is in large part due to inaccurate initial camera pose estimates, which cause content to be incorrectly localized in 3D space, and cause the reconstructions to settle in geometrically inaccurate local-minima solutions. When the FOV of the simulated camera is expanded, and we simulate rays at steeper angles relative to the camera axis as compared to the input data, we see these overfitting artifacts as texture quality on the edges of the baseline renders significantly degrades. *Instant-NGP* in particular struggles to extrapolate from data with low parallax or significant scene motion, such as the billowing steam clouds in *Bluepit*. Conversely, our proposed approach is able to recover fine texture content in these areas, including readable text on the drink labels in *Vending*.

4.3 Applications to Low-light Photography

Illustrated in Fig. 9, we find that, when trained on 10-bit linear RAW data, our neural light sphere model is robust to sensor noise as experienced in high ISO ($\geq 10,000$) settings during low-light photography. Similar to the findings of [Mildenhall et al. 2022], we find that by averaging rays that converge to identical scene points during training, our model learns a mean photometric solution for scene reconstruction, averaging out zero-mean Gaussian read noise. This also proves beneficial for non-light-limited settings, as we can lower exposure time for a single image to reduce motion blur during capture without risking failed reconstruction. Based on these initial findings, we expect a neural neural light sphere-style model could potentially be tailored for applications such as video denoising and astrophotography.

5 DISCUSSION AND FUTURE WORK

In this work we present a compact and robust neural light sphere model for handheld panoramic scene reconstruction. We demonstrate high-quality texture reconstruction in expanded field-of-view renders, with high tolerance to adverse imaging effects such as noise and localized pixel motion.

Future Work. We hope that this work, and the accompanying metadata- and measurement-rich dataset, can encourage follow-on research into scene reconstruction under adverse imaging conditions. Many of the scenes, such as those illustrated in Fig. 7, purposely contain effects such as lens flare, snow, clouds, smog, reflections, sensor noise, and saturated high-dynamic range content. During in-the-wild data collection we found these effects unavoidable, highlighting the importance of robust reconstruction methods for practical computational photography.

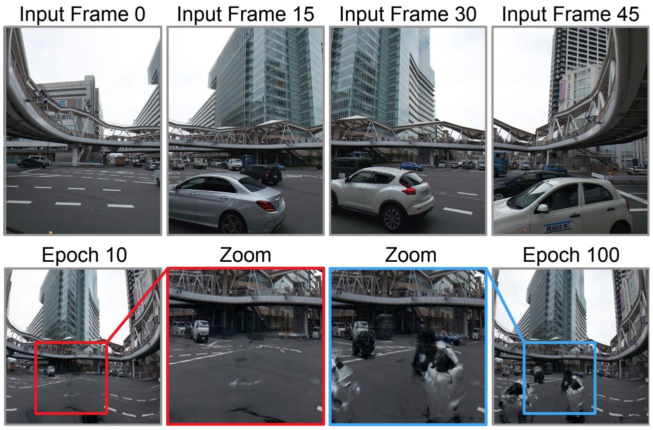


Fig. 11. **Fast Occluders.** Objects such as bikes and cars, which quickly enter and exit the field-of-view of the camera, pose a challenge for scene reconstruction as they cannot be compactly modeled as a view-dependent effect. Shown in the example above, during early training the fast-moving cars are effectively erased from the reconstruction, which fits quickly to the median static pixel color. However, during later training stages, the view-dependent h_R and h_D models attempt (and fail) to reconstruct the content in motion, leading to transient car-shaped artifacts in the reconstruction.

Beyond conventional photography, we believe this approach can be extended to industrial and scientific imaging settings such as satellite and telescope-based photography, scanning and array microscopes, and infrared or hyperspectral imaging. In particular, with a hardware-optimized hash-grid backing, our model design makes it computationally tractable to fit petapixel-and-larger data produced by these imaging modalities by breaking it into smaller ray batches – e.g., a hash table size of 2^{22} reliably trains with batch size 2^{13} on a single Nvidia RTX 4090 GPU.

Limitations. Although the proposed method is robust to localized pixel motion and color changes – e.g., swaying tree branches, flowing water – it is not capable of reconstructing large fast-moving obstructions such as vehicles driving through a scene as shown in Fig. 11. This setting has posed a long-standing challenge for image-stitching and panoramic reconstruction works [Szeliski et al. 2007], as when there are few observations of these occluders, this becomes a segmentation and tracking problem that is difficult to solve with a purely photometric approach such as ours. Similarly, without the ability to generate novel content, the camera path of the input capture strongly determines view synthesis performance – e.g., a purely horizontal pan does not provide enough view information to simulate the effects of large vertical camera motion.

ACKNOWLEDGMENTS

Ilya Chugunov was supported by an NSF GRFP (2039656). Felix Heide was supported by an Amazon Science Research Award, Packard Foundation Fellowship, Sloan Research Fellowship, Sony Young Faculty Award, the Project X Fund, and NSF CAREER (2047359). We thank Richard Szeliski, Jon Barron, and Ben Mildenhall for their valuable insights and discussions during this work's development.

– Supplementary Material –

Contents

A IMPLEMENTATION DETAILS

We compile a list of data recorded by our capture app and its uses in Tab. 1. Our image processing pipeline follows the following sequence:

1. Rearrange RAW data to BGGR format with *color filter arrangement*
2. Re-scale color channels as: *(channel - black level)/(white level - black level)*
3. Multiply by *color correction gain*
4. Multiply by inverse of *shade map*
5. Linearly interpolate gaps in mosaic (i.e., three interpolated values per red or blue, two interpolated values per green)
6. Input into dataloader for training

To render final output images we then:

1. Multiply RGB by the 3 × 3 *color correction matrix*
2. Re-scale color values with the *tonemap curve*

Or, optionally, skip this color correction to maximize render speed.

During training, we also use *lens distortion* and *rolling shutter skew* values to correct measurements on the ray level. Specifically we apply the lens distortion model as:

$$\begin{aligned} x_{\text{dist}} &= x(1+\kappa_1 r^2+\kappa_2 r^4+\kappa_3 r^6+\kappa_4 r^8+\kappa_5 r^{10}) \\ y_{\text{dist}} &= y(1+\kappa_1 r^2+\kappa_2 r^4+\kappa_3 r^6+\kappa_4 r^8+\kappa_5 r^{10}) \end{aligned} \tag{11}$$

where ($r^2 = x^2 + y^2$) is the squared radius from the optical center given by the camera *intrinsics.* We also shift the time n at which rays are sampled – linearly interpolating translation *T(n)* and rotation *R(n)* – by the row the ray was sampled from multiplied by the row rolling shutter delay given by *rolling shutter skew/image height.* We note that this rolling shutter delay had negligible effect on the overall reconstruction, possibly due to view-dependent ray offset model $f_{\text{R}}(\hat{P}, X)$ already able to compensate for it (introducing a row-dependent skew to the rays).

Data	Purpose
intrinsics	ray projection (K)
color correction matrix	render output images
tonemap curve	render output images
shade map	correct RAWs (lens shading)
color filter arrangement	correct RAWs (BGGR)
lens distortion	correct RAWs (distortion)
color filter gains	correct RAW (color)
whitelevel	scale RAW data (max)
blacklevel	scale RAW data (min)
gyroscope values	rotation initialization (G)
timestamps	synchronize measurements
rolling shutter skew	rolling shutter correction
accelerometer values	**unused**
ISO	**unused**
exposure time	**unused**
focus distance	**unused**
focal length	**unused**
lens extrinsics	**unused**
lens aperture	**unused**
neutral color point	**unused**
noise profile	**unused**

Table 1. **Recorded Data.** A non-exhaustive list of data, both used and unused in this project, recorded by our capture app.

While we do not use data such as *accelerometer values*, which give poor localization performance after double integration for pans, or *ISO* and *exposure time*, we hope that these may be of use in follow-on work. For example, while we keep exposure and ISO locked during our captures, it could be possible to combine bracketing [?] with panoramic capture to reconstruct ultra-HDR scenes.

B MODEL COMPONENT ANALYSIS

In Fig. 12 we visualize the independent contributions of the view-dependent color $f_{\text{C}}(\hat{P}^*, X)$ and ray offset models $f_{\text{R}}(\hat{P}, X)$ to our neural light sphere reconstructions. We train the model with both of these components active, and *during inference time* we remove the output of the ray offset model

$$\hat{D}^* = D^*/\|D^*\|, \quad D^* = \text{rot}\left(\cancelto{1}{\mathbf{R} = f_{\text{R}}(\hat{P}, X)}\right)\hat{D} = \hat{D}, \tag{12}$$

remove the view-dependent color model

$$\begin{aligned} f_{\text{C}}(\hat{P}^*, X) &= h_{\text{C}}\left(h_{\text{P}}\left(\gamma_2(\hat{P}^*);\, \theta_{\text{P}}\right) + \cancelto{0}{h_{\text{D}}\left(\gamma_1(X);\, \theta_{\text{D}}\right)};\, \theta_{\text{C}}\right) \\ f_{\text{C}}(\hat{P}^*, X) &= h_{\text{C}}\left(h_{\text{P}}\left(\gamma_2(\hat{P}^*);\, \theta_{\text{P}}\right);\, \theta_{\text{C}}\right), \end{aligned} \tag{13}$$

or remove both. From the resultant reconstructions, we can see how effects in the scene are modeled by one, both, or neither of these models. Static content on the surface of the sphere, such as the background folliage in *BluePit* and *ShinySticks* remains nearly identical in all reconstructions, which is entirely expected as this content exhibits almost no parallax and view-dependent color changes. In contrast, scene elements such as the reflections on the surface of

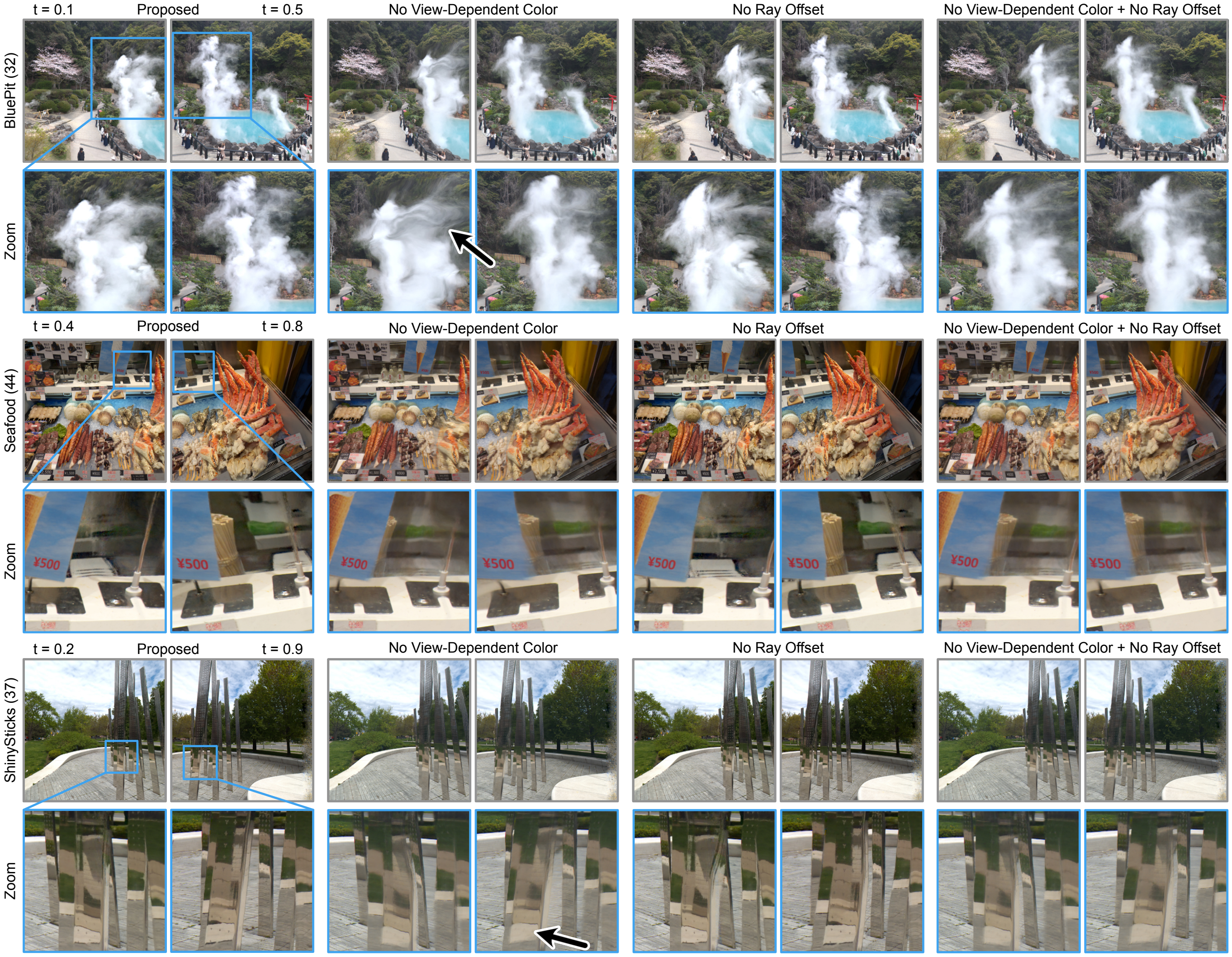


Fig. 12. **Model Component Analysis.** Shown above are the effects on reconstruction of zeroing out the contribution of the view-dependent color model $h_{\mathrm{D}}(\gamma_1(X);\,\theta_{\mathrm{D}})$, ray offset model $f_{\mathrm{R}}(\hat{P}, X)$, or both models. We can observe that complex dynamic effects such as the steam clouds in *BluePit* are produced by a combination of view-dependent color effects for the cloud texture, and ray offset for bulk motion. This is in contrast to the chopstick canister hidden behind the blue sign in *Seafood*, which is almost entirely reconstructed with view-dependent color alone. In *ShinySticks*, we observe how the sharp content and dots on the surface of the statue disappear when view-dependent color is removed, and large distortions in geometry appear when ray offset is omitted.

ShinySticks and the steam clouds in *BluePit* require both the ray offset and view-dependent color models to work in tandem in order to produce these complex visual effects. This separability of our neural light sphere model also points towards a potentially interesting direction of future work, editing both content and its dynamics after reconstruction similar to a video mosaic [Kasten et al. 2021] (e.g., turning the motion of the steam clouds into billowing smoke from a fire).

C ALTERNATIVE RAY OFFSET MODELS

During the development of this work, we experimented with different ray offset models to model parallax and scene motion. This includes a *Depth* model where we modify Eq. 4 of the main work to individually offset the radius of the sphere by $f_{\mathrm{R}}(\hat{P}, X)$ for each ray

$$\hat{P}^* = P/\|P\|, \quad P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = O + t\hat{D}$$

$$t = -\left(O \cdot \hat{D}\right) + \sqrt{(O \cdot \hat{D})^2 - (\|O\|^2 - (1 + f_{\mathrm{R}}(\hat{P}, X)))}, \qquad (14)$$

simulating a depth map stretched across the inside surface of the sphere model. Another model we tested was a *Multiplicative* ray offset

$$\hat{D}^* = D^*/\left\|D^*\right\|, \quad D^* = (1 + f_{\mathrm{R}}(\hat{P}, X)) \circ \hat{D}, \qquad (15)$$

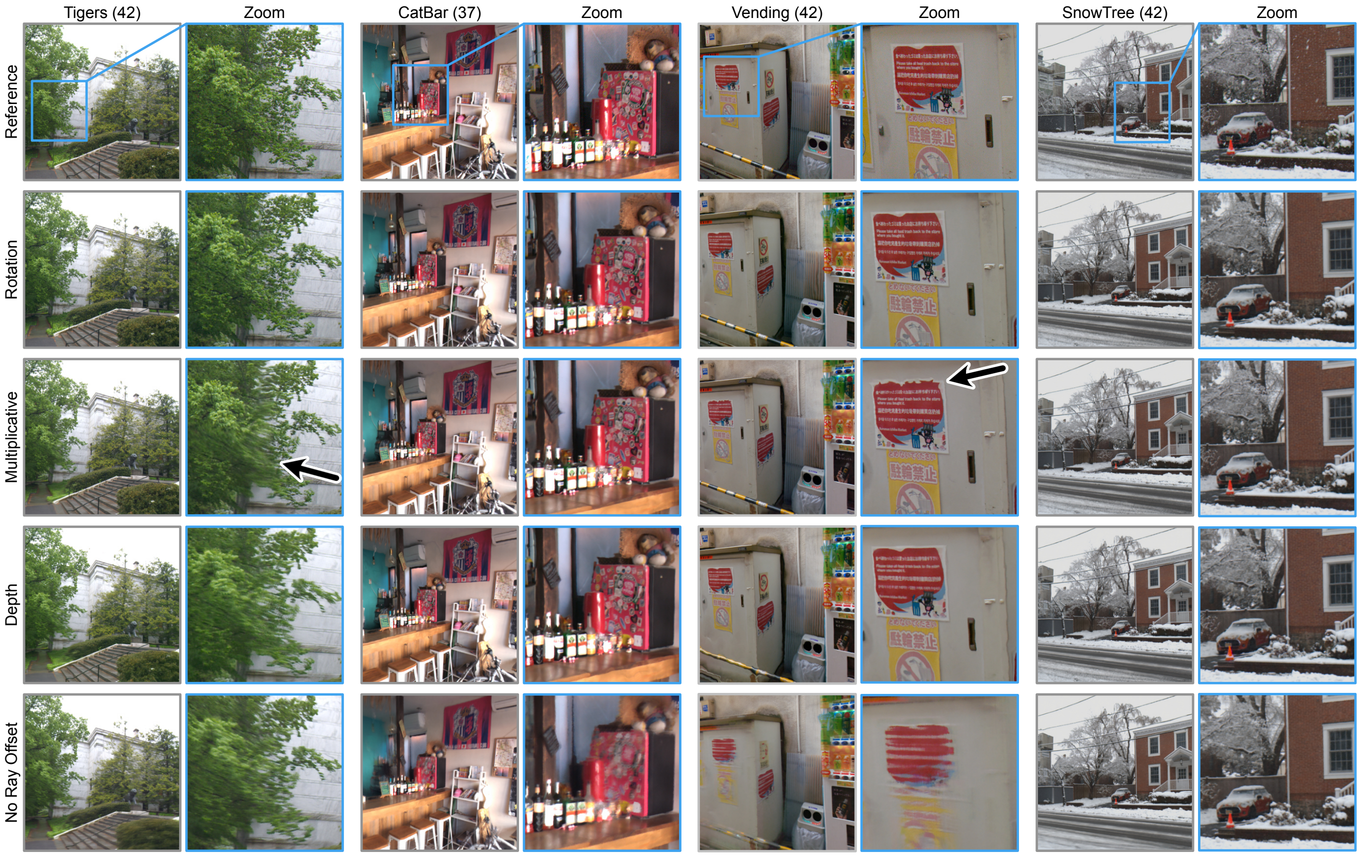


Fig. 13. **Ray Offset Models.** Comparing scene reconstruction results for various ray offset models, it's clear from the *No Ray Offset* results that many scenes such as *CatBar* and *Vending* contain significant parallax effects that a sphere projection model alone cannot compensate for. The *Depth* and *Multiplicative* models significantly improves reconstruction quality, albeit some regions in the *Multiplicative* reconstructions suffer from distortions. The linearized *Rotation* model avoids these artifacts while maintaining high reconstruction quality, recovering legible text in the *Vending* scene.

where ∘ denotes element-wise multiplication. In Fig. 13 we can see how this model further sharpens content when compared to the *Depth* model, but leads to blur and distortions in the scene where a large multiplicative offset causes rays to be "pushed" out of a region in the scene. The final ray offset model we chose was a linearized *Rotation* model

$$\hat{D}^* = D^*/\left\|D^*\right\|, \quad D^* = \text{rot}\left(\mathbf{R} = f_{\mathrm{R}}(\hat{P}, X)\right)\hat{D}, \tag{16}$$

which we observed to lead to high reconstruction quality without the distortions observed in the *Multiplicative* model. Here a larger $f_{\mathrm{R}}(\hat{P}, X)$ rotates a region of rays together a larger distance, rather than pushing them out of a region on the sphere.

To compare these models, we remove the view-dependent color model $h_{\mathrm{D}}\,(\gamma_1(X);\, \theta_{\mathrm{D}})$ as outlined in Sec. B *during training*, not just during inference. As otherwise this $h_{\mathrm{D}}\,(\gamma_1(X);\, \theta_{\mathrm{D}})$ can compensate for content that was not correctly reconstructed by the ray offset model. We compare reconstruction results for these offset models in Fig. 13, noting that for scenes such as *Vending* and *CatBar* with large amount of parallax the choice of offset model significantly affects reconstruction quality. Conversely, for *SnowTree*, where content is far from the camera, all models produce similar reconstructions, emphasizing the importance of collecting a diverse set of scenes to holistically evaluate in-the-wild image stitching.

D ADDITIONAL RECONSTRUCTION RESULTS

In Fig. 14 we showcase additional reconstruction results and comparisons to radiance field baselines: *K-Planes* [Fridovich-Keil et al. 2023], *Gaussian Splatting* [Kerbl et al. 2023], *Instant-NGP* [Muller et al. 2022], and *Nerfacto* [Tancik et al. 2023]. Noteably, we see in *Bridge* the high resolution reconstruction enabled by our method, which is able to correctly resolve the cross-hatch bars in the bridge's support structure. In *DarkPeace* we see that while *Nerfacto* and *Gaussian Splatting* successfully reconstruct the left side of the scene, the area of maximum overlap where the capture started, they produce extremely noisy reconstructions at the end of the capture sequence, with *Instant-NGP* failing to reconstruct any of the scene. In *CityCars* we can observe how, while our neural light sphere model is not able to reconstruct the fast-moving cars, the reconstruction artifacts only disrupt local content. Zooming into the background, we can still

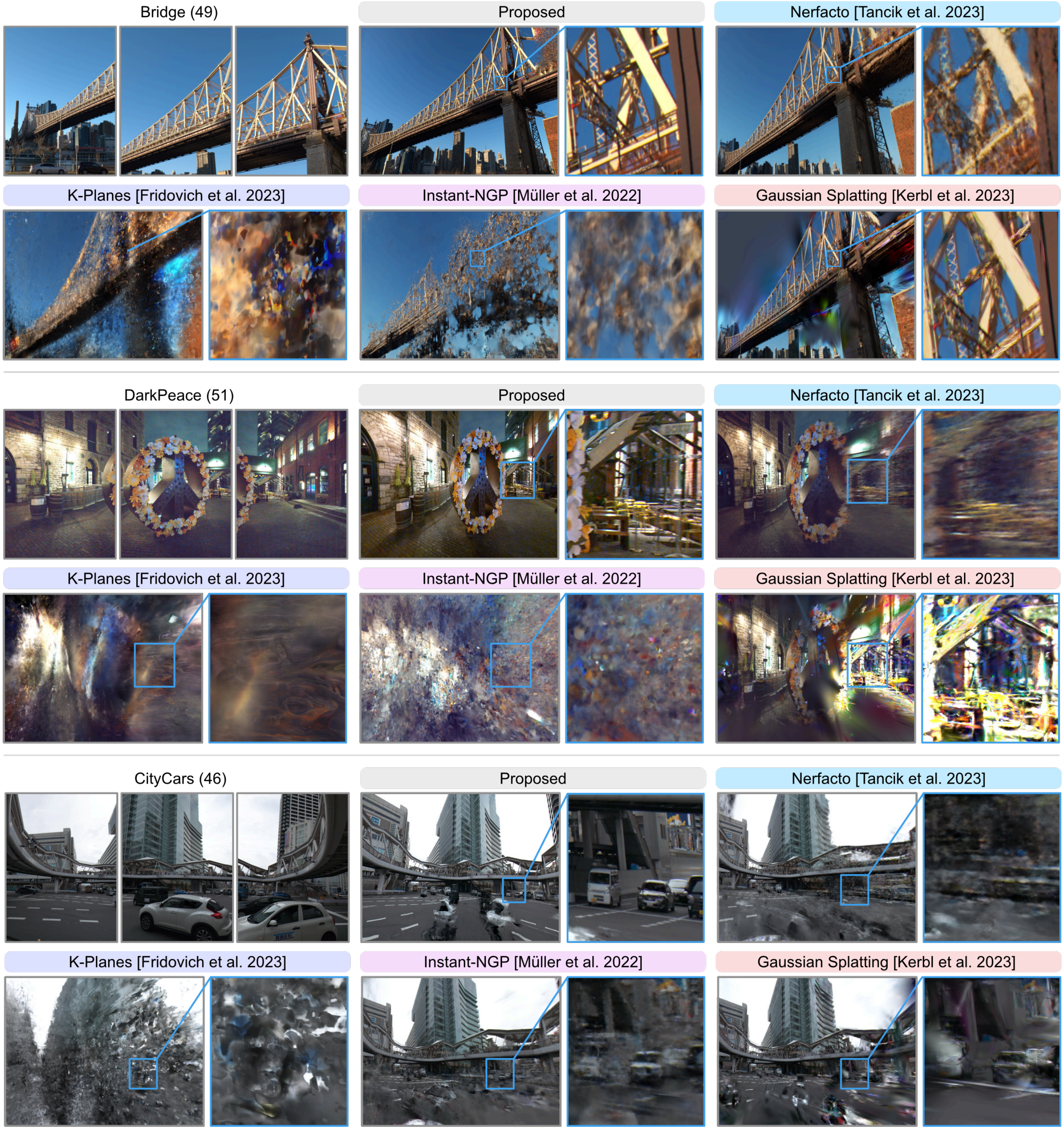

Bridge (49)
Proposed
Nerfacto [Tancik et al. 2023]
K-Planes [Fridovich et al. 2023]
Instant-NGP [Müller et al. 2022]
Gaussian Splatting [Kerbl et al. 2023]
DarkPeace (51)
Proposed
Nerfacto [Tancik et al. 2023]
K-Planes [Fridovich et al. 2023]
Instant-NGP [Müller et al. 2022]
Gaussian Splatting [Kerbl et al. 2023]
CityCars (46)
Proposed
Nerfacto [Tancik et al. 2023]
K-Planes [Fridovich et al. 2023]
Instant-NGP [Müller et al. 2022]
Gaussian Splatting [Kerbl et al. 2023]

Fig. 14. **Additional Radiance Field Comparisons.** Reconstruction results for a highly detailed back-and-forth *Bridge* capture, night-time *DarkPeace*, and *CityCars* with fast-moving occluders. Scene titles are formatted as: *Scene Name (Number of Captured Frames in Input)*

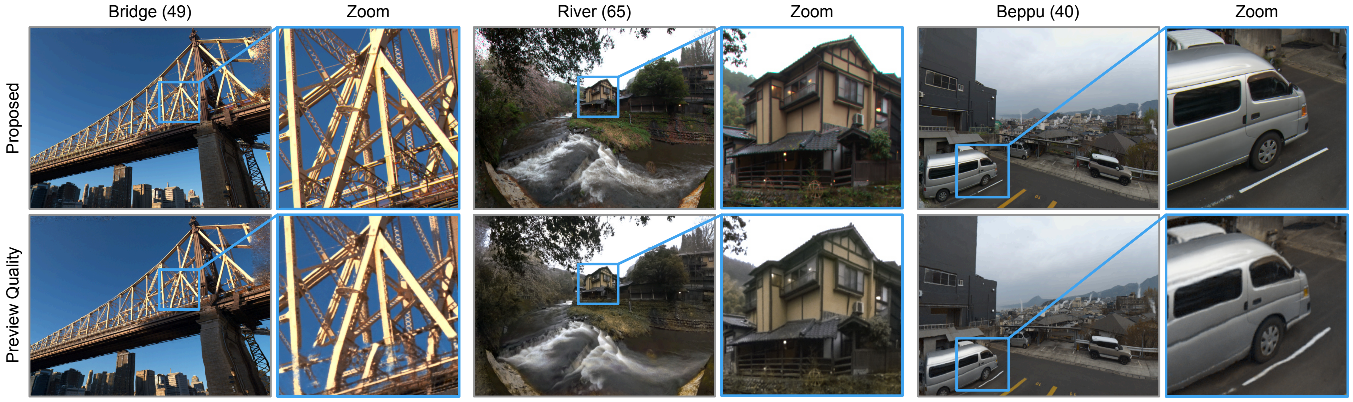


Fig. 15. **Preview Quality Reconstructions.** Trained on 1/4 resolution inputs for 1/10th of the number of epochs, while they don't reach the full reconstruction quality of the proposed method, these "Preview Quality" reconstructions take less than 30 seconds of training time per scene.

resolve the static cars, unlike the baseline methods, which produce reconstructions corrupted by motion artifacts.

E PREVIEW-SCALE RENDERING

While the reconstructions shown in the main text are relatively fast to train compared to the average neural radiance field approach, during model exploration and development we found it extremely beneficial to be able to quickly test large collections of scenes. By down-sampling the input data from full resolution 12-megapixel images to 1/4 resolution 3-megapixel images, dividing the max epochs by 10, and removing tensorboarding operations we are able to render "preview quality" scenes in less than 30 seconds. While there is a notable drop in quality for some scene content, as seen in the deformation of the grey car in the *Beppu* example shown in Fig. 15, other scenes reach high reconstruction quality even in this short training time. Even zooming into the *River* scene it is difficult to see a change in quality between the two reconstructions; suggesting that with some training augmentation, near-instant reconstruction could be possible for some subset of panoramic video captures.

REFERENCES

Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. 2004. Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers*. 294–302.

Aseem Agarwala, Ke Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Richard Szeliski. 2005. Panoramic video textures. In *ACM SIGGRAPH 2005 Papers*. 821–827.

Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. 2022. Learning neural light fields with ray-space embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19819–19829.

Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. 2020. MatryODShka: Real-time 6DoF video view synthesis using multi-sphere images. In *European Conference on Computer Vision*. Springer, 441–459.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19697–19705.

Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. 2020. Omniphotos: casual 360 vr photography. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–12.

Mary L Boas. 2006. *Mathematical methods in the physical sciences.* John Wiley & Sons.

Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. 2019. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11036–11045.

Matthew Brown and David G Lowe. 2007. Automatic panoramic image stitching using invariant features. *International journal of computer vision* 74 (2007), 59–73.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 425–432.

Peter J Burt and Edward H Adelson. 1983. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)* 2, 4 (1983), 217–236.

Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. 2000. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 307–318.

Shenchang Eric Chen and Lance Williams. 1993. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 279–288.

Ilya Chugunov, David Shustin, Ruyu Yan, Chenyang Lei, and Felix Heide. 2024. Neural Spline Fields for Burst Image Fusion and Layer Separation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024).

Ilya Chugunov, Yuxuan Zhang, and Felix Heide. 2023. Shakes on a plane: Unsupervised depth estimation from unstabilized photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13240–13251.

Abril Corona-Figueroa, Jonathan Frawley, Sam Bond-Taylor, Sarath Bethapudi, Hubert PH Shum, and Chris G Willcocks. 2022. Mednerf: Medical neural radiance fields for reconstructing 3d-aware ct-projections from a single x-ray. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 3843–3848.

John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5515–5524.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. 2023. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12479–12488.

Junhong Gao, Seon Joo Kim, and Michael S Brown. 2011. Constructing image panoramas using dual-homography warping. In *CVPR 2011*. IEEE, 49–56.

Junhong Gao, Yu Li, Tat-Jun Chin, and Michael S Brown. 2013. Seam-driven image stitching.. In *Eurographics (Short Papers)*. 45–48.

Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. 1996. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 43–54.

Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision.* Cambridge university press.

Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. 2017. Casual 3D photography. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–15.

Peter Hedman and Johannes Kopf. 2018. Instant 3d photography. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. 2021. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–12.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* 42, 4 (2023), 1–14.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Po Kong Lai, Shuang Xie, Jochen Lang, and Robert Laganière. 2019. Real-time panoramic depth maps from omni-directional stereo images for 6 dof videos in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 405–412.

Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, New York, NY, USA, 31–42.

Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. 2023. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8456–8465.

Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. 2021. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5741–5751.

Kai-En Lin, Zexiang Xu, Ben Mildenhall, Pratul P Srinivasan, Yannick Hold-Geoffroy, Stephen DiVerdi, Qi Sun, Kalyan Sunkavalli, and Ravi Ramamoorthi. 2020. Deep multi depth panoramas for view synthesis. In *European Conference on Computer Vision*. Springer, 328–344.

David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60 (2004), 91–110.

Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7210–7219.

Leonard McMillan and Gary Bishop. 1995. Plenoptic modeling: an image-based rendering system. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 39–46.

Microsoft Research. 2015. Microsoft Image Composite Editor. https://www.microsoft.com/en-us/research/product/computational-photography-applications/image-composite-editor. Version 2.0.3, released in 2015.

Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. 2022. 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*. 16190–16199.

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

Luca Morreale, Noam Aigerman, Vladimir G Kim, and Niloy J Mitra. 2021. Neural surface maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4639–4648.

Thomas Muller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)* 41, 4 (2022), 1–15.

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time neural radiance caching for path tracing. *arXiv preprint arXiv:2106.12372* (2021).

Seonghyeon Nam, Marcus A Brubaker, and Michael S Brown. 2022. Neural image representations for multi-image fusion and layer separation. In *European conference on computer vision*. Springer, 216–232.

Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. 2005. *Light field photography with a hand-held plenoptic camera*. Ph. D. Dissertation. Stanford university.

Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. 2022. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5480–5490.

Alex Rav-Acha, Pushmeet Kohli, Carsten Rother, and Andrew Fitzgibbon. 2008. Unwrap mosaics: A new representation for video editing. In *ACM SIGGRAPH 2008 papers*. 1–11.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 2019. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12716–12725.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4938–4947.

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

Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. 1998. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 231–242.

GJV Shepherd. 1925. The Interpretation of Aerial Photographs. *Royal United Services Institution. Journal* 70, 478 (1925), 279–287.

Heung-Yeung Shum and Li-Wei He. 1999. Rendering with concentric mosaics. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 299–306.

Heung-Yeung Shum and Richard Szeliski. 2002. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision* 48, 2 (2002), 151–152.

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. 2019. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2437–2446.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. 2022. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8269–8279.

Richard Szeliski et al. 2007. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision* 2, 1 (2007), 1–104.

Richard Szeliski and Heung-Yeung Shum. 1997. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 251–258.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. 2020. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems* 33 (2020), 7537–7547.

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. 2023. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–12.

Katie Trumpener and Tim Barringer. 2020. *On the Viewing Platform: The Panorama Between Canvas and Screen*. Yale University Press.

Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. 2005. High performance imaging using large camera arrays. In *ACM siggraph 2005 papers*. 765–776.

Wenqi Xian, Aljaž Božič, Noah Snavely, and Christoph Lassner. 2023. Neural Lens Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8435–8445.

Julio Zaragoza, Tat-Jun Chin, Michael S Brown, and David Suter. 2013. As-projective-as-possible image stitching with moving DLT. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2339–2346.

Lihi Zelnik-Manor, Gabriele Peters, and Pietro Perona. 2005. Squaring the circle in panoramas. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2. IEEE, 1292–1299.

Fan Zhang and Feng Liu. 2014. Parallax-tolerant image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3262–3269.

Ke Colin Zheng, Sing Bing Kang, Michael F Cohen, and Richard Szeliski. 2007. Layered depth panoramas. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

Assaf Zomet, Anat Levin, Shmuel Peleg, and Yair Weiss. 2006. Seamless image stitching by minimizing false edges. *IEEE transactions on image processing* 15, 4 (2006), 969–977.