Neural Spline Fields for Burst Image Fusion and Layer Separation

Ilya Chugunov David Shustin Ruyu Yan Chenyang Lei Felix Heide
Princeton University

Abstract

Each photo in an image burst can be considered a sample of a complex 3D scene: the product of parallax, diffuse and specular materials, scene motion, and illuminant variation. While decomposing all of these effects from a stack of misaligned images is a highly ill-conditioned task, the conventional align-and-merge burst pipeline takes the other extreme: blending them into a single image. In this work, we propose a versatile intermediate representation: a two-layer alpha-composited image plus flow model constructed with neural spline fields – networks trained to map input coordinates to spline control points. Our method is able to, during test-time optimization, jointly fuse a burst image capture into one high-resolution reconstruction and decompose it into transmission and obstruction layers. Then, by discarding the obstruction layer, we can perform a range of tasks including seeing through occlusions, reflection suppression, and shadow removal. Tested on complex in-the-wild captures we find that, with no post-processing steps or learned priors, our generalizable model is able to outperform existing dedicated single-image and multi-view obstruction removal approaches.

1. Introduction

Over the last decade, as digital photos have increasingly been produced by smartphones, smartphone photos have increasingly been produced by burst fusion. To compensate for less-than-ideal camera hardware – typically restricted to a footprint of less than 1cm^3 [7] – smartphones rely on their advanced compute hardware to process and fuse multiple lower-quality images into a high-fidelity photo [11]. This proves particularly important in low-light and high-dynamic-range settings [23, 40], where a single image must compromise between noise and motion blur, but multiple images afford the opportunity to minimize both [27]. But even as mobile night- and astro-photography applications [17, 18] use increasingly long sequences of photos as input, their output remains a static single-plane image. Given the typically non-static and non-planar nature of the real world, a core problem in burst image pipelines is thus the alignment [33, 46] and aggregation [6, 65] of pixels into an image array – referred to as the *align-and-merge* process.

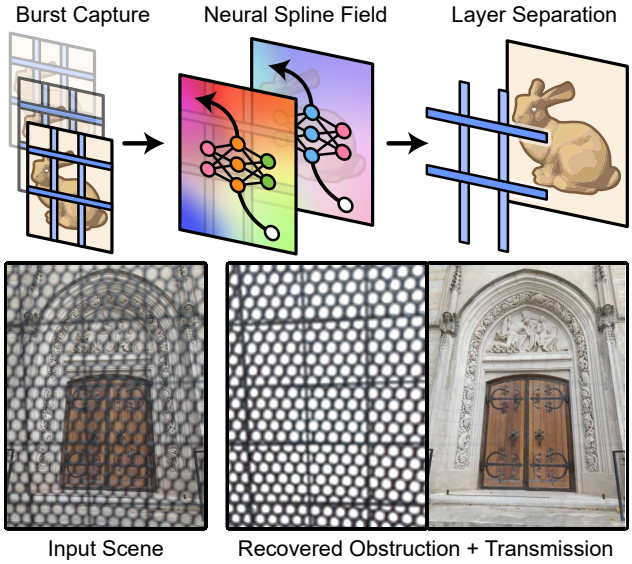


Figure 1. Fitting our two-layer neural spline field model to a stack of images we're able to directly estimate and separate even severe, out-of-focus obstructions to recover hidden scene content.

While existing approaches treat pixel motion as a source of noise and artifacts, a parallel direction of work [10, 21, 71] attempts to extract useful parallax cues from this pixel motion to estimate the geometry of the scene. Recent work by Chugunov et al. [9] finds that maximizing the photometric consistency of an RGB plus depth neural field model of an image sequence is enough to distill dense depth estimates of the scene. While this method is able to jointly estimate high-quality camera motion parameters, it does not perform high-quality image reconstruction, and rather treats its image model as "a vehicle for depth optimization" [9]. In contrast, work by Nam et al. [51] proposes a neural field fitting approach for multi-image fusion and layer separation which focuses on the quality of the reconstructed "canonical view". By swapping in different motion models, they can separate and remove layers such as occlusions, reflections, and moiré patterns during image reconstruction – as opposed to in a separate post-processing step [20, 55]. This approach, however, does not make use of a realistic cam-

era projection model, and relies on regularization penalties to discourage its motion models from representing non-physical effects – e.g., pixel tearing or teleportation.

In this work, we propose a versatile layered neural image representation [51] with a projective camera model [9] and novel neural spatio-temporal spline [69] parametrization. Our model takes as input an unstabilized 12-megapixel RAW image sequence, camera metadata, and gyroscope measurements – available on all modern smartphones. During test-time optimization, it fits to produce a high-resolution reconstruction of the scene, separated into *transmission* and *obstruction* image planes. The latter of which can be extracted to perform occlusion removal, reflection suppression, and other layer separation applications. To this end, we decompose pixel motion between burst frames into planar motion, from the camera's pose change in 3D space relative to the image planes, and a generic flow component which accounts for depth parallax, scene motion, and other image distortions. We model these flows with neural spline fields (NSFs): networks trained to map input coordinates to spline control points, which are then interpolated at sample timestamps to produce flow field values. As their output dynamics are strictly bound by their spline parametrization, these NSFs produce *temporally* consistent flow with no regularization, and can be controlled *spatially* through the manipulation of their positional encodings.

In summary, we make the following contributions:

- An end-to-end neural scene fitting approach which fits to a burst image sequence to distill high-fidelity camera poses, and high-resolution two layer transmission plus occlusion image decomposition.
- A compact, controllable neural spline field model to estimate and aggregate pixel motion between frames.
- Qualitative and quantitative evaluations which demonstrate that our model outperforms existing single image and multi-frame obstruction removal approaches.

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

2. Related Work

Burst Photography. A large body of work has explored methods for burst image processing [11] to achieve high image quality in mobile photography settings. During burst imaging, the device records a sequence of frames in rapid succession – potentially a *bracketed sequence* with varying exposure parameters [45] – and fuses them post-capture to produce a demosaiced [59], denoised [16, 46], superresolved [33, 65], or otherwise enhanced reconstruction. Almost all modern smartphone devices rely on burst photography for low-light [23, 40] and high dynamic range reconstruction from low dynamic range sensors [14, 23]. While existing methods typically use sequences of only 2-8 frames as input, a parallel field of micro-video [26, 71] or "long-burst photography" [9] research – which also encompasses widely deployed Apple Live Photos, Android Motion Photos, and night photography [17, 18] – consumes sequences of images up to several seconds in length, acquired naturally during camera viewfinding. Though not limited to long-burst photography, we adopt this setting to leverage the parallax [67] and pixel motion cues in these extended captures for separation of obstructed and transmitted scene content.

Obstruction Removal and Layer Separation. While their use of visual cues is diverse – e.g., identifying reflections from "ghosting" cues on thick glass [55] or detecting lattices for fence deletion [53] – single-image obstruction removal is fundamentally a segmentation [32, 42] and image recovery [15, 25] problem. In the most severe cases, with fully opaque occluders, this image recovery problem becomes an in-painting task [12, 66] to synthesize missing content. This is in contrast to approaches which rely on multiple measurements such as multi-focal stacks [1, 54], multi-view images [43, 52], flash no-flash pairs [34, 36], or polarization data [35]. These methods typically treat obstruction removal as an inverse problem [5], estimating a model of transmitted and occluded content consistent with observed data [37]. This can also be generalized to an image layer separation problem, an example of which is intrinsic decomposition [8], where the separated layer is the obstruction. These methods typically rely on learned priors [15] and pixel motion [51] to decompose images into multiple components. Our work explores the layer separation problem in the burst photography setting, where pixel motion is on a much smaller scale than in video sequences [44], and a high-resolution unobstructed view is desired as an output. Rather than tailor to a single application, however, we propose a unified model with applications to reflection, occlusion, and shadow separation.

Neural Scene Representations. A growing body of work investigating novel view synthesis has demonstrated that coordinate-based neural representations are capable of reconstructing complex scenes [3, 4] without an explicit structural backbone such as a pixel array or voxel grid. These networks are typically trained from scratch, through *test-time optimization*, on a single scene to map input coordinate encodings [60] to outputs such as RGB [56], depth [10], or x-ray data [57]. While neural scene representations require many network evaluations to generate outputs, as opposed to explicit representations which can be considered "pre-evaluated", recent works have shown great success in accelerating training [49] and inference [70] of these networks. Furthermore, this per-output network evaluation is what lends to their versatility, as they can be optimized through auto-differentiation with no computational penalties for sparse or non-uniform sampling of the scene [31]. Several recent approaches make use of neural scene rep-

resentations in tandem with continuous motion estimation models to fit multi-image [9] and video [39] data, potentially decomposing it into multiple layers in the process [28, 51]. Our work proposes a novel neural spline field continuous flow representation with a projective camera model to separate effects such as occlusions, reflections, and shadows. In contrast to existing approaches, our flow model does not require regularization to prevent overfitting, as its representation power is controlled directly through encoding and spline hyperparameters.

3. Neural Spline Fields for Burst Photography

We begin with a discussion of the proposed neural spline field model of optical flow. We then continue with our full two-layer projective model of burst photography, its loss functions, training procedure, and data collection pipeline.

3.1. Neural Spline Fields.

Motivation. To recover a latent image, existing burst photography methods *align and merge* [11] pixels in the captured image sequence. Disregarding regions of the scene that spontaneously change – e.g., blinking lights or digital screens – pixel differences between images can be decomposed into the products of scene motion, illuminant motion, camera rotation, and depth parallax. Separating these sources of motion has been a long-standing challenge in vision [62, 63] as this is a fundamentally ill-conditioned problem; in typical settings, scene and camera motion are geometrically equivalent [22]. One response to this problem is to disregard effects other than camera motion, which can yield high-quality motion estimates for static, mostly-lambertian scenes [9, 26, 71]. This can be represented as

$$I(u, v, t) = [R, G, B] = f(\boldsymbol{\pi}\boldsymbol{\pi}_t^{-1}(u, v)), \tag{1}$$

where $I(u, v, t)$ is a frame from the burst stack captured at time t and sampled at image coordinates $u, v \in [0, 1]$. Operators $\boldsymbol{\pi}$ and $\boldsymbol{\pi}_t$ perform 3D reprojection on these coordinates to transform them from time t to the coordinates of a reference image model $f(u, v) \rightarrow [R, G, B]$. To account for other sources of motion, layer separation approaches such as [28, 51] estimate a generic flow model $\Delta u, \Delta v = g(u, v, t)$ to re-sample the image model

$$I(u, v, t) = f(u + \Delta u, v + \Delta v). \tag{2}$$

However, this parametrization introduces an overfitting risk, the consequences of which are illustrated in Fig. 2, as $g(u, v, t)$ and $f(u, v)$ can now act as a generic video encoder [39]. To combat this, methods often employ a form of gradient penalty such as total variation loss [51]. That is

$$\mathcal{L}_{\text{TVFlow}} = \sum \|J_g(u, v, t)\|_1,$$

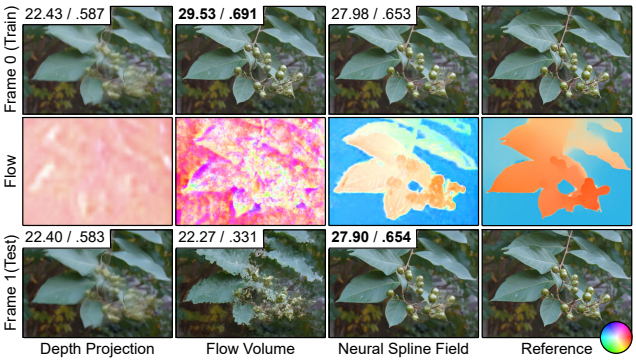


Figure 2. Image and flow estimates for different representations of a short video sequence of a swinging branch; PSNR/SSIM values inset top-left. Depth projection alone is unable to represent both parallax and scene motion, mixing reconstructed content, and an un-regularized 3D flow volume $g(u, v, t)$ trivially overfits to the sequence. With an identical network, spatial encoding, loss function, and training procedure as $g(u, v, t)$, our neural spline field $S(t; \mathbf{P} = h(u, v))$ produces temporally consistent flow estimates well-correlated with a conventional optical flow reference [41].

where $J_g(u, v, t)$ is the Jacobian of the flow model. During training, this can prove computationally expensive, however, as now each sample requires its local neighborhood to be evaluated to numerically estimate the Jacobian, or a second gradient pass over the model. In both cases, a large number of operations are spent to limit the reconstruction of high frequency spatial and temporal content.

Formulation. We propose a neural spline field (NSF) model of flow, a learned spatio-temporal spline [69] representation which provides strong controls on reconstruction directly through its parametrization. This model splits flow evaluation into two components

$$\Delta u, \Delta v = g(u, v, t) = S(t; \mathbf{P} = h(u, v)). \tag{3}$$

Here $h(u, v)$ is the NSF, a network which maps image coordinates to a set of spline control points $\mathbf{P}$. Then, to estimate flow for a frame at time t in the burst stack, we evaluate the spline at $S(t; \mathbf{P})$. We select a cubic Hermite spline

$$\begin{aligned} S(t, \mathbf{P}) = &(2t_r^3 - 3t_r^2 + 1)\mathbf{P}_{\lfloor t_s \rfloor} + (-2t_r^3 + 3t_r^2)\mathbf{P}_{\lfloor t_s \rfloor + 1} \\ &+ (t_r^3 - 2t_r^2 + t_r)(\mathbf{P}_{\lfloor t_s \rfloor} - \mathbf{P}_{\lfloor t_s \rfloor - 1})/2 \\ &+ (t_r^3 - t_r^2)(\mathbf{P}_{\lfloor t_s \rfloor + 1} - \mathbf{P}_{\lfloor t_s \rfloor})/2 \\ t_r = t_s - \lfloor t_s \rfloor, &\quad t_s = t \cdot |\mathbf{P}|, \end{aligned} \tag{4}$$

as it guarantees continuity in time with respect to its zeroth, first, and second derivatives and allows for fast local evaluation – in contrast to Bézier curves [9] which require recursive calculations. We emphasize that the use of splines in graphics problems is extensive [13], and that there are many alternate candidate functions for $S(t, \mathbf{P})$. E.g., if the motion is expected to be a straight line, a piece-wise linear spline with $|\mathbf{P}| = 2$ control points would insure this constraint is satisfied irrespective of the outputs of $h(u, v)$.

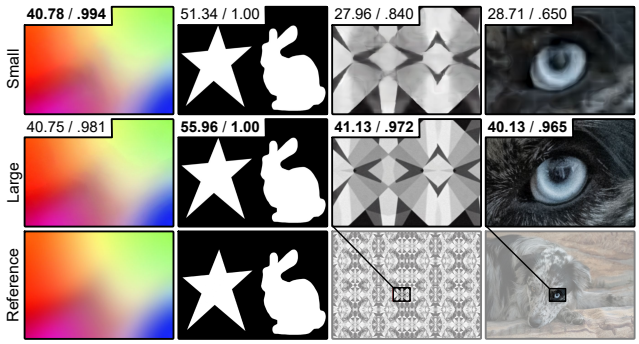


Figure 3. Image fitting results for coordinate networks with *Small* (L^γ=8) and *Large* (L^γ=16) multi-resolution hash encodings and identical other parameters; PSNR/SSIM values inset top-left. Unlike a traditional band-limited representation [68], the *Small* resolution network is able to fit both low-frequency smooth gradients and sharp edge mask images, but fails to fit a high density of either. This makes it a promising candidate representation for scene flow and alpha mattes which are comprised of smooth gradients and a limited number of object edges.

Where the choice of $S(t, \mathbf{P})$ and $|\mathbf{P}|$ determines the temporal behavior of flow, $h(u, v)$ controls its spatial properties. While our method, in principle, is not restricted to a specific spatial encoding function, we adopt the multi-resolution hash encoding $\gamma(u, v)$ presented in Müller et al. [49]

$$\begin{aligned} h(u, v) &= \mathbf{h}(\gamma(u, v;\, \mathrm{params}_\gamma);\, \theta) \\ \mathrm{params}_\gamma &= \{\mathrm{B}^\gamma, \mathrm{S}^\gamma, \mathrm{L}^\gamma, \mathrm{F}^\gamma, \mathrm{T}^\gamma\}, \end{aligned} \tag{5}$$

as it allows for fast training and strong spatial controls given by its encoding parameters params_γ: base grid resolution B^γ, per level scale factor S^γ, number of grid levels L^γ, feature dimension F^γ, and backing hash table size T^γ. Here, $\mathbf{h}(\gamma(u, v); \theta)$ is a multi-layer perceptron (MLP) [24] with learned weights θ. Illustrated in Fig. 3 with an image fitting example, the number of grid levels L^γ – which, with a fixed S^γ, sets the maximum grid resolution – provides controls on the maximum "spatial complexity" of the output while still permitting accurate reconstruction of image edges.

3.2. Projective Model of Burst Photography

Motivation. With a flow model $g(u, v, t)$, and a canonical image representation $f(u, v)$ in hand, we theoretically have all the components needed to model an arbitrary image sequence [28,51]. However, handheld burst photography does *not* produce arbitrary image sequences; it has well-studied photometric and geometric properties [9,10,21,65]. This, in combination with the abundance of physical metadata such as gyroscope values and calibrated intrinsics available on modern smartphone devices [9], provides strong support for a physical model of image formation.

Formulation. We adopt a forward model similar to traditional multi-planar imaging [22]. We note that this departs from existing work [9, 10], which employs a backward projection camera model – "splatting" points from a canonical representation to locations in the burst stack. A multi-plane imaging model allows for both simple composition of multiple layers along a ray – a task for which backward projection is not well suited – and fast calculation of ray intersections without the ray-marching needed by volumetric representations like NeRF [48]. For simplicity of notation, we outline this model for a single projected ray below. We also illustrate this process in Fig. 4. Let

$$c = [\mathrm{R}, \mathrm{G}, \mathrm{B}]^\top = I(u, v, t) \tag{6}$$

be a colored point sampled at time t in the burst stack at image coordinates $u, v \in [0, 1]$. Note that these coordinates are relative to the camera pose at time t; for example $(u, v) = (0, 0)$ is always the bottom-left corner of the image. To project these points into world space we introduce camera translation $T(t)$ and rotation $R(t)$ models

$$\begin{aligned} T(t) &= S(t, \mathbf{P}^\mathrm{T}), \quad R(t) = R^\mathrm{D}(t) + \eta_\mathrm{R} S(t, \mathbf{P}^\mathrm{R}) \\ \mathbf{P}^\mathrm{T}_i &= \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{P}^\mathrm{R}_i = \begin{bmatrix} 0 & -r^z & r^y \\ r^z & 0 & -r^x \\ -r^y & r^x & 0 \end{bmatrix}. \end{aligned} \tag{7}$$

Here $S(t, \mathbf{P})$ is the same cubic spline model from Eq. (4), evaluated element-wise over the channels of $\mathbf{P}$. We note there are *no coordinate networks* employed in these models. Translation $T(t)$ is learned from scratch, $\mathbf{P}^\mathrm{T}$ initialized to all-zeroes. Rotation $R(t)$ is learned as a small-angle approximation offset [26] to device rotations $R^\mathrm{D}(t)$ recorded by the phone's gyroscope – or alternatively, the identity matrix if such data is not available. With these two models, and calibrated intrinsic matrix K from the camera metadata, we now generate a ray with origin O and direction D as

$$O = \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} = T(t), \quad D = \begin{bmatrix} D_x \\ D_y \\ 1 \end{bmatrix} = \frac{R(t) K^{-1}}{D_z} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{8}$$

where D is normalized by its z component. We define our transmission and obstruction image planes as Π^T and Π^O, respectively. As XY translation of these planes conflicts with changes in the camera pose, we lock them to the z-axis at depth Π_z with canonical axes Π_u and Π_v. Thus, given ray direction D has a z-component of 1, we can calculate the ray-plane intersection as $Q = O + (\Pi_z - O_z)D$ and project to plane coordinates

$$u^\Pi, v^\Pi = \langle Q, \Pi_u \rangle / (\Pi_z - O_z),\ \langle Q, \Pi_v \rangle / (\Pi_z - O_z), \tag{9}$$

scaled by ray length to preserve uniform spatial resolution. Let $u^\mathrm{T}, v^\mathrm{T}$ and $u^\mathrm{O}, v^\mathrm{O}$ be the intersection coordinates for the transmission and obstruction plane, respectively. We alpha composite these layers along the ray as

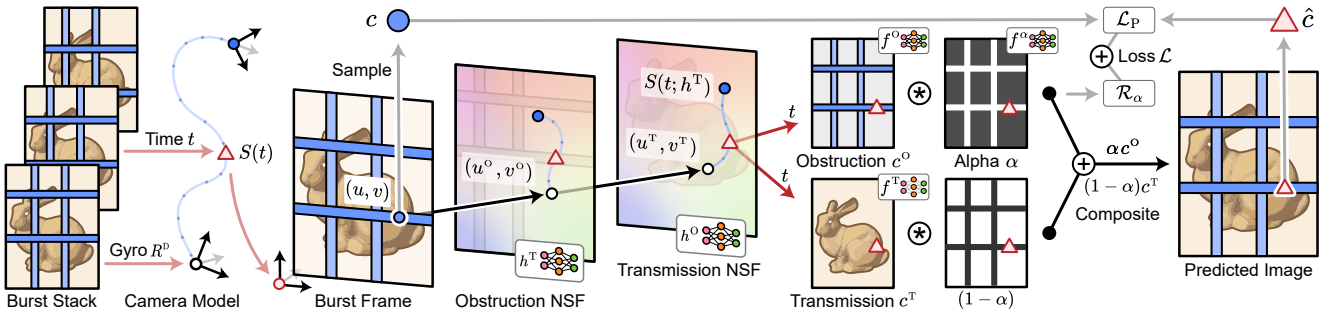


Figure 4. We model an input image sequence as the alpha composition of a *transmission* and *obstruction* plane. Motion in the scene is expressed as the product of a rigid camera model, which produces global rotation and translation, and two neural spline field models, which produce local flow estimates for the two layers. Trained to minimize photometric loss, this model separates content to its respective layers.

$$
\begin{aligned}
\hat{c} &= (1-\alpha)c^{\mathrm{T}} + \alpha c^{\mathrm{O}} \\
c^{\mathrm{T}} &= f^{\mathrm{T}}(u^{\mathrm{T}} + \Delta u^{\mathrm{T}}, v^{\mathrm{T}} + \Delta v^{\mathrm{T}}),\ \Delta u^{\mathrm{T}}, \Delta v^{\mathrm{T}} = S(t; h^{\mathrm{T}}(u^{\mathrm{T}}, v^{\mathrm{T}})) \\
c^{\mathrm{O}} &= f^{\mathrm{O}}(u^{\mathrm{O}} + \Delta u^{\mathrm{O}}, v^{\mathrm{O}} + \Delta v^{\mathrm{O}}),\ \Delta u^{\mathrm{O}}, \Delta v^{\mathrm{O}} = S(t; h^{\mathrm{O}}(u^{\mathrm{O}}, v^{\mathrm{O}})) \\
\alpha &= \sigma(\tau_{\sigma} f^{\alpha}(u^{\mathrm{O}} + \Delta u^{\mathrm{O}}, v^{\mathrm{O}} + \Delta v^{\mathrm{O}})),
\end{aligned}
\tag{10}
$$

where $\hat{c}$ is the composite color point, the weighted sum by α of the transmission color c^{T} and obstruction color c^{O}. Each is the output of an image coordinate network $f(u, v)$ sampled at points offset by flow from an NSF $h(u, v)$. The sigmoid function $\sigma = 1/(1+e^{-x})$ with temperature τ_{σ} controls the transition between opaque $\alpha = 1$ and partially translucent $\alpha = 0.5$ obstructions. This proves particularly helpful for learning hard occluders – e.g., a fence – where large τ_{σ} creates a steep transition between $\alpha = 0$ and $\alpha = 1$, which discourages $f^{\alpha}(u, v)$ from mixing content between layers.

3.3. Training Procedure

Losses. Given all the components of our model are fully differentiable, we train them end-to-end via stochastic gradient descent. We define our loss function $\mathcal{L}$ as

$$
\begin{aligned}
\mathcal{L} &= \mathcal{L}_{\mathrm{P}} + \eta_{\alpha} \mathcal{R}_{\alpha} \\
\mathcal{L}_{\mathrm{P}} &= |(c - \hat{c})/(\mathrm{sg}(c) + \epsilon)|, \quad \mathcal{R}_{\alpha} = |\alpha|,
\end{aligned}
\tag{11}
$$

where $\mathcal{L}_{\mathrm{P}}$ is a relative photometric reconstruction loss [9, 47], and sg is the stop-gradient operator. Shown in Fig. 5, when combined with linear RAW input data this loss proves robust in noisy imaging settings [47], appropriate for in-the-wild scene reconstruction with unknown lighting conditions. Regularization term $\mathcal{R}_{\alpha}$ with weight η_{α} penalizes content in the obstruction layer, discouraging it from duplicating features from the transmission layer.

Training. Given the high-dimensional problem of jointly solving for camera poses, image layers, and neural spline field flows, we turn to coarse-to-fine optimization to avoid low-quality local minima solutions. We mask the multi-resolution hash encodings $\gamma(u, v)$ input into our image, alpha, and flow networks, activating higher resolution grids during later epochs of training:

$$
\gamma_i(u, v) = \begin{cases} \gamma_i(u, v) & \text{if } \ i/|\gamma| < 0.4 + 0.6(\mathrm{sin\_epoch}) \\ 0 & \text{if } \ i/|\gamma| > 0.4 + 0.6(\mathrm{sin\_epoch}) \end{cases}
$$

$$
\mathrm{sin\_epoch} = \sin(\mathrm{epoch}/\mathrm{max\_epoch}), \tag{12}
$$

This strategy results in less noise accumulated during early training as spurious high-resolution features do not need to be "unlearned" [9, 38] during later stages of refinement.

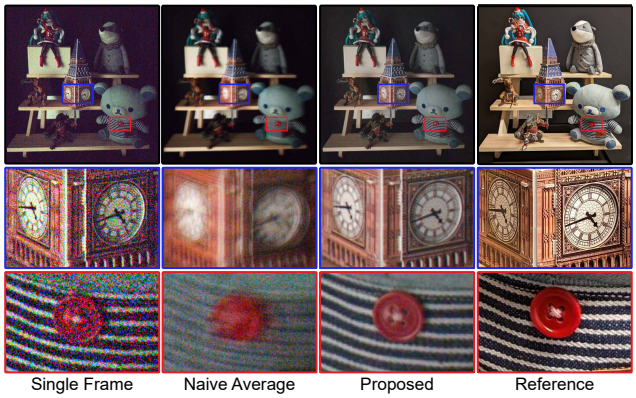


Figure 5. Reconstruction results for noisy, low-light conditions; exposure time 1/30, ISO 5000. The proposed model is able to robustly merge frames into a denoised image representation.

4. Applications

Data Collection. To collect burst data we modify the open-source Android camera capture tool Pani to record continuous streams of RAW frames and sensor metadata. During capture, we lock exposure and focus settings to record a 42 frame, two-second "long-burst" of 12-megapixel images, gyroscope measurements, and camera metadata. We refer the reader to Chugunov et al. [9] for an overview of the long-burst imaging setting and its geometric properties. We capture data from a set of Pixel 7, 7-Pro, and 8-Pro devices, with no notable differences in overall reconstruction quality or changes in the training procedure required. We train our networks directly on Bayer RAW data, and apply device color-correction and tone-mapping for visualization.

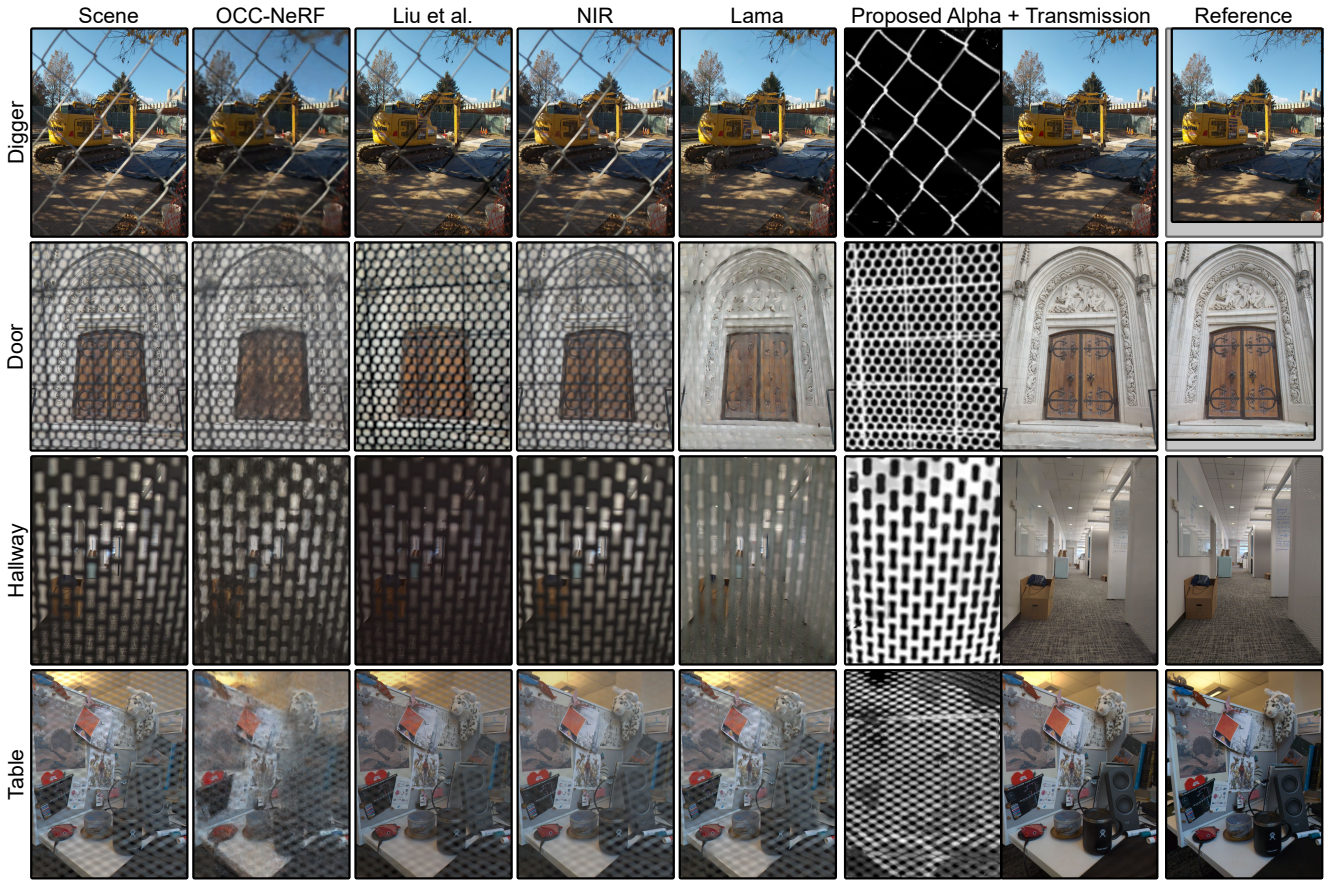


Figure 6. Occlusion removal results and estimated alpha maps for a set of captures with reference views; comparisons to single image, multi-view, and NeRF fitting approaches. See video materials for visualization of input data and scene fitting.

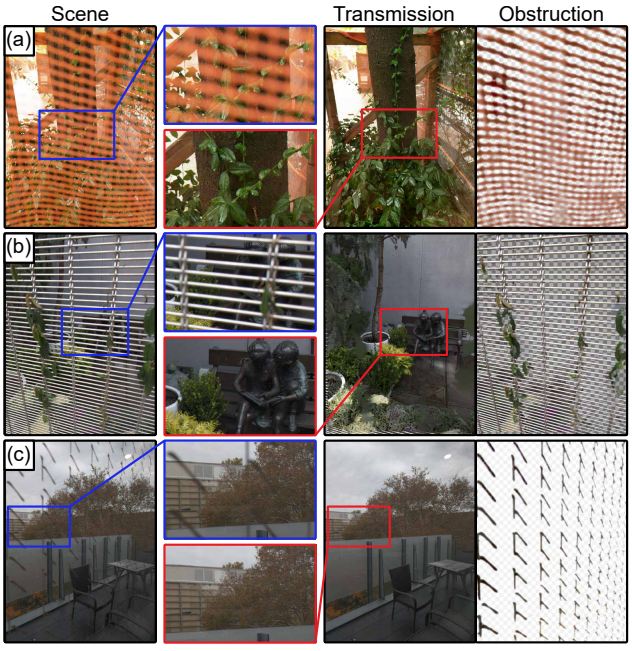


Figure 7. Layer separation results in unique real-world cases enabled by our generalizable two-layer image model: (a) orange planter, (b) fenced garden, (c) stickers on balcony glass.

OCC-NeRF	Liu et al.	NIR	Lama	Proposed
19.58/0.625	19.52/0.823	19.55/0.685	20.82/0.694	**41.20/0.982**

NeRFReN	Liu et al.	NIR	DSR-Net	Proposed
19.26/0.708	19.82/0.815	21.86/0.829	18.55/0.760	**33.84/0.985**

Figure 8. Qualitative and quantitative obstruction removal results for a set of synthetic scenes with paired ground truth, camera motion simulated from real measured hand shake data [10]. Evaluation metrics formatted as PSNR/SSIM.

Figure 9. Reflection removal results and estimated alpha maps for a set of captures with reference views; comparisons to single image, multi-view, and NeRF fitting approaches. See video materials for visualization of input data and scene fitting.

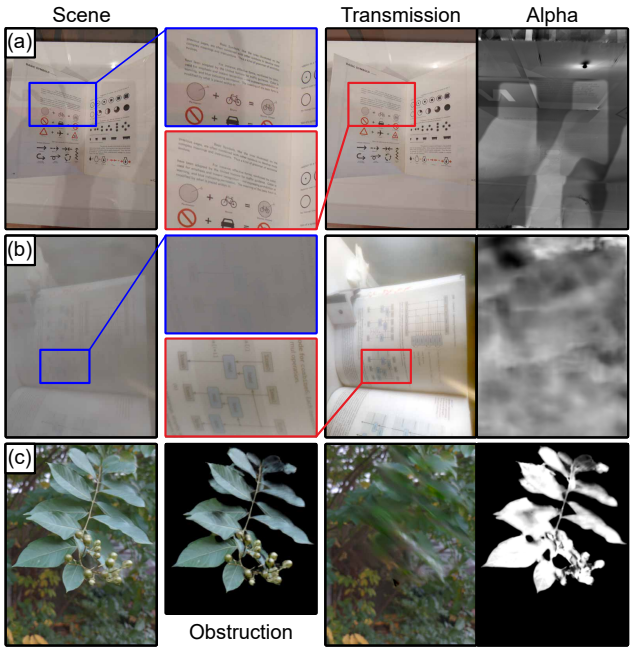


Figure 10. Layer separation results for additional example applications: (a) shadow removal, (b) image dehazing, and (c) video motion segmentation (see video materials for visualization).

Implementation Details. During training, we perform stochastic gradient descent on $\mathcal{L}$ for batches of 2^{18} rays per step for 6000 steps with the Adam optimizer [29]. All networks use the multi-resolution hash encoding described in Eq. (5), implemented in tiny-cuda-nn [50]. Trained on a single Nvidia RTX 4090 GPU, our method takes *approximately 3 minutes* to fit a full 42-frame image sequence. All networks have a base resolution B^γ=4, and scale factor S^γ=1.61, but while flow networks h^{T} and o are parameterized with a low number of grid levels L^γ=8, networks which represent high frequency content have L^γ=12 or L^γ=16 levels. These settings are task-specific, and full implementation details and results for short (4-8 frame) image bursts are included in the Supplementary Material.

Occlusion Removal. Initializing the obstruction plane closer to the camera than the transmission plane, that is $\Pi_z^{\mathrm{O}} < \Pi_z^{\mathrm{T}}$, we find that the $f^{\mathrm{O}}(u, v)$ naturally reconstructs foreground content in the scene. Given a scene with content hidden behind a foreground occluder – e.g., imaging through a fence – we can then perform occlusion removal with the proposed method by setting $\alpha = 0$. We report results in Fig. 6 for a set of captures collected with reference views using a tripod-mounted occluder. We compare here

to the multiview plus learning method presented in *Liu et al.* [43], the neural radiance field approach *OCC-NeRF* [72], the flow + homography neural image model *NIR* [51], and the single image inpainting method *Lama* [58] as these methods demonstrate a broad range of techniques for occlusion detection and removal with varying assumptions on camera motion. We find that in this small baseline burst photography setting, existing multi-view methods fail to achieve meaningful occlusion removal; as the occluder maintains a high level of self-overlap for the whole image sequence. While the single-image method, *Lama* is able to in-paint occluded regions based on un-occluded content, it cannot faithfully recover lost details such as the carvings in the *Door* scene. Furthermore, *Lama* does not produce an alpha matte, and rather *requires a hand-annotated mask as input*. Illustrated in Fig. 11, even otherwise robust mask segmentation networks such as the Segment Anything Model (SAM) [30] fail to correctly detect complex occluders. In contrast, our approach distills information from all input frames to accurately recover temporarily occluded content, and jointly produces a high-quality alpha matte. In Fig. 10 we present additional layer separation results for real in-the-wild scenes with complex occluders, which demonstrate the versatility of the obstruction image model $f^{\mathrm{O}}(u, v)$.

Reflection Removal. We show in Fig. 9 how by flipping the plane depths $\Pi_z^{\mathrm{O}} > \Pi_z^{\mathrm{T}}$, our model is also able to separate reflected from transmitted content. Here, we compare again to *Liu et al.* [43] and *NIR* [51], as well as the reflection-specific neural radiance approach *NeRFReN* [19] and single-image reflection removal network *DSR-Net* [25]. Similarly to occlusion removal, we observe that given small-baseline inputs the multi-view methods fail to achieve meaningful layer separation, and *NeRFRen* struggles to converge on a sharp reconstruction. Only *DSR-Net* is able to suppress even small parts of the reflection such as the car in the *Hydrant* scene. In contrast, the proposed method not only estimates nearly reflection-free transmission layers, but is also able to recover hidden content – such as the flowerpot highlighted in *Pinecones* – in the reflection layer.

Synthetic Validation. Given in-the-wild captures do not have perfectly aligned reference images, to further validate our method we construct a set of rendered scenes with paired ground truth data. Quantitative and qualitative results in Fig. 8 and the Supplementary Material align with our findings from real-world captures, with significant PSNR and SSIM improvements across all scenes.

Image Enhancement through Layer Separation. In addition to occlusion and reflection removal, a wide range of other computational photography applications can be viewed through the lens of layer separation. We showcase several example tasks in Fig. 7, including shadow removal, image dehazing, and video motion segmentation. The key relationship between all these tasks is that the two effects

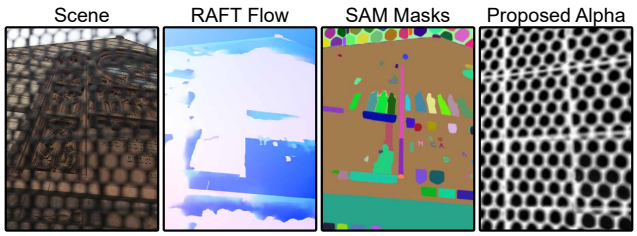


Figure 11. Learned flow estimator RAFT [61] and segmentation model SAM [30] struggle to produce meaningful outputs for a small-motion scene with an out-of-focus occluder. SAM successfully segments some objects behind the occluder (e.g., the statues on the building) but does not correctly segment the occluder itself.

undergo different motion models – e.g., photographer-cast shadows move with the cellphone, while the paper target stays static. By grouping color content with its respective motion model, $f^{\mathrm{T}}(u, v)$ with $h^{\mathrm{T}}(u, v)$ and $f^{\mathrm{O}}(u, v)$ with $h^{\mathrm{O}}(u, v)$, just as in the occlusion case, we can remove the effect by removing its image plane. Fig. 7 (c), which fits our two-layer model for an image sequence of a moving tree branch, also highlights that our method does not rely solely on camera motion. Scene motion itself can also be used as a mechanism for layer separation in image bursts, similar to approaches in video masking [28, 44].

5. Discussion and Future Work

In this work, we present a versatile representation of burst photography built on a novel neural spline field model of flow, and demonstrate image fusion and obstruction removal results under a wide array of conditions. In future work, we hope this generalizable model can be tailored to specific layer separation and image fusion applications:

Learned Features. Video layer separation works [28, 44, 69] make use of pre-trained segmentation networks and optical flow estimators to help guide reconstruction. However, shown in Fig. 11, we found these could not be directly applied to small-motion data with large obstructions, as this is far outside the domain of their training data. Adapting these models to complex burst photography settings could potentially help disambiguate image layers in areas without reliable parallax or motion information.

Physical Priors. Our generic image plus flow representation can accommodate task-specific modules for applications where there are known physical models, such as chromatic aberration removal or refractive index estimation.

Beyond Burst Data. There exist many other sources of multi-image data to which the method can potentially be adapted – e.g., microscopes, telescopes, and light field, time-of-flight, or hyperspectral cameras.

Acknowledgements. Ilya Chugunov was supported by 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).

References

[1] Hannan Adeel, Muhammad Mohsin Riaz, and Syed Sohaib Ali. De-fencing and multi-focus fusion using markov random field and image inpainting. *IEEE Access*, 10:35992–36005, 2022. 2

[2] Yuval Bahat and Tomer Michaeli. Explorable super resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2716–2725, 2020. 19

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023. 2

[5] Mario Bertero, Patrizia Boccacci, and Christine De Mol. *Introduction to inverse problems in imaging*. CRC press, 2021. 2

[6] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9209–9218, 2021. 1

[7] Vladan Blahnik and Oliver Schindelbeck. Smartphone imaging technology and its applications. *Advanced Optical Technologies*, 10(3):145–232, 2021. 1

[8] Qifeng Chen and Vladlen Koltun. A simple model for intrinsic image decomposition with depth cues. In *Proceedings of the IEEE international conference on computer vision*, pages 241–248, 2013. 2

[9] Ilya Chugunov, Yuxuan Zhang, and Felix Heide. Shakes on a plane: Unsupervised depth estimation from unstabilized photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13240–13251, 2023. 1, 2, 3, 4, 5, 21

[10] 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. 1, 2, 4, 6, 11, 12

[11] Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Mobile computational photography: A tour. *arXiv preprint arXiv:2102.09000*, 2021. 1, 2, 3

[12] Muhammad Shahid Farid, Arif Mahmood, and Marco Grangetto. Image de-fencing framework with hybrid inpainting algorithm. *Signal, Image and Video Processing*, 10:1193–1201, 2016. 2

[13] Gerald E Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002. 3

[14] Orazio Gallo, Alejandro Troccoli, Jun Hu, Kari Pulli, and Jan Kautz. Locally non-rigid registration for mobile hdr photography. In *Proceedings of the IEEE conference on computer vision and pattern recognition Workshops*, pages 49–56, 2015. 2

[15] Yosef Gandelsman, Assaf Shocher, and Michal Irani. " double-dip": unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11026–11035, 2019. 2

[16] Clément Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *Proceedings of the European conference on computer vision (ECCV)*, pages 538–554, 2018. 2

[17] Google. See in the dark with night sight. https://blog.google/products/pixel/see-light-night-sight/, 2018. Accessed: 2023-10-24. 1, 2

[18] Google. Astrophotography with night sight on pixel phones. https://blog.research.google/2019/11/astrophotography-with-night-sight-on.html, 2019. Accessed: 2023-10-24. 1, 2

[19] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18409–18418, June 2022. 8, 14, 18

[20] Divyanshu Gupta, Shorya Jain, Utkarsh Tripathi, Pratik Chattopadhyay, and Lipo Wang. Fully automated image defencing using conditional generative adversarial networks, 2019. 1

[21] Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. High-quality depth from uncalibrated small motion clip. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5413–5421, 2016. 1, 4

[22] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3, 4

[23] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 1, 2

[24] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 4

[25] Qiming Hu and Xiaojie Guo. Single image reflection separation via component synergy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13138–13147, October 2023. 2, 8, 14, 18

[26] Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo, and In So Kweon. High quality structure from small motion for rolling shutter cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 837–845, 2015. 2, 3, 4

[27] Nima Khademi Kalantari, Ravi Ramamoorthi, et al. Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph.*, 36(4):144–1, 2017. 1

[28] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing, 2021. 3, 4, 8

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[30] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 8

[31] Fredrik Kjolstad, Shoaib Kamil, Stephen Chou, David Lugato, and Saman Amarasinghe. The tensor algebra compiler. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA):1–29, 2017. 2

[32] Keitaro Kume and Masaaki Ikehara. Single image fence removal using fast fourier transform. In *2023 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–5, 2023. 2

[33] Bruno Lecouat, Thomas Eboli, Jean Ponce, and Julien Mairal. High dynamic range and super-resolution from raw image bursts. *arXiv preprint arXiv:2207.14671*, 2022. 1, 2

[34] Chenyang Lei and Qifeng Chen. Robust reflection removal with reflection-free flash-only cues. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[35] Chenyang Lei, Xuhua Huang, Mengdi Zhang, Qiong Yan, Wenxiu Sun, and Qifeng Chen. Polarized reflection removal with perfect alignment in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1750–1758, 2020. 2

[36] Chenyang Lei, Xudong Jiang, and Qifeng Chen. Robust reflection removal with flash-only cues in the wild, 2022. 2

[37] Yu Li and Michael S. Brown. Exploiting reflection change for automatic reflection removal. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013. 2

[38] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 5

[39] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023. 3

[40] Orly Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, Jonathan T Barron, Dillon Sharlet, Ryan Geiss, et al. Handheld mobile photography in very low light. *ACM Trans. Graph.*, 38(6):164–1, 2019. 1, 2

[41] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. *arXiv preprint arXiv:2109.07547*, 2021. 3

[42] Yunfei Liu, Yu Li, Shaodi You, and Feng Lu. Semantic guided single image reflection removal, 2022. 2

[43] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Learning to see through obstructions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 8, 13, 14, 18

[44] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T Freeman, and Michael Rubinstein. Omnimatte: Associating objects and their effects in video. In *CVPR*, 2021. 2, 8

[45] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. In *Computer graphics forum*, volume 28, pages 161–171. Wiley Online Library, 2009. 2

[46] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2502–2510, 2018. 1, 2

[47] 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. 5

[48] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 4, 12

[49] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 2, 4

[50] Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Real-time neural radiance caching for path tracing. *arXiv preprint arXiv:2106.12372*, 2021. 7

[51] Seonghyeon Nam, Marcus A. Brubaker, and Michael S. Brown. Neural image representations for multi-image fusion and layer separation, 2022. 1, 2, 3, 4, 8, 13, 14, 18, 19

[52] Simon Niklaus, Xuaner Cecilia Zhang, Jonathan T. Barron, Neal Wadhwa, Rahul Garg, Feng Liu, and Tianfan Xue. Learned dual-view reflection removal, 2020. 2

[53] Minwoo Park, Kyle Brocklehurst, Robert T Collins, and Yanxi Liu. Image de-fencing revisited. In *Computer Vision–ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part IV 10*, pages 422–434. Springer, 2011. 2

[54] Zeqi Shen, Shuo Zhang, and Youfang Lin. Light field reflection and background separation network based on adaptive focus selection. *IEEE Transactions on Computational Imaging*, 9:435–447, 2023. 2

[55] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T. Freeman. Reflection removal using ghosting cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2

[56] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[57] Yu Sun, Jiaming Liu, Mingyang Xie, Brendt Wohlberg, and Ulugbek S Kamilov. Coil: Coordinate-based internal learn-

ing for tomographic imaging. *IEEE Transactions on Computational Imaging*, 7:1400–1412, 2021. 2

[58] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 8, 13, 14, 18, 19

[59] Hanlin Tan, Xiangrong Zeng, Shiming Lai, Yu Liu, and Maojun Zhang. Joint demosaicing and denoising of noisy bayer images with admm. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2951–2955. IEEE, 2017. 2

[60] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 2

[61] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 8

[62] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8375–8384, 2021. 3

[63] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1384, 2013. 3

[64] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8178–8187, 2019. 12

[65] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (TOG)*, 38(4):1–18, 2019. 1, 2, 4

[66] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5840–5848, 2019. 2

[67] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T Freeman. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015. 2

[68] Guandao Yang, Sagie Benaim, Varun Jampani, Kyle Genova, Jonathan Barron, Thomas Funkhouser, Bharath Hariharan, and Serge Belongie. Polynomial neural fields for subband decomposition and manipulation. *Advances in Neural Information Processing Systems*, 35:4401–4415, 2022. 4

[69] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2657–2666, 2022. 2, 3, 8

[70] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2

[71] Fisher Yu and David Gallup. 3d reconstruction from accidental motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3986–3993, 2014. 1, 2, 3

[72] Chengxuan Zhu, Renjie Wan, Yunkai Tang, and Boxin Shi. Occlusion-free scene recovery via neural radiance fields. 2023. 8, 13, 18

Supplementary Material

In this supplementary material, we provide implementation details, additional results, ablation studies, and experimental analysis in support of the findings of the main text. The structure of this document is as follows:

- Section A: Details on data generation, model implementation, and training procedure.
- Section B: Additional obstruction removal results with comparison methods and synthetic validation. Analysis of challenging reconstruction settings.
- Section C: Additional analysis on manipulating model and training parameters. Includes reconstruction results for subsampled and short burst sequences.

A. Implementation Details

Data Acquisition To acquire paired obstructed and unobstructed captures, we construct two tripod-mounted rigs as illustrated in Fig. 1 (a-b). We begin by capturing a still of the scene without the obstruction, before rotating the tripod into position to capture a 42-frame obstructed long-burst [10] of 12-megapixel RAW frames. As the rig is only used to hold the obstruction – i.e., the smartphone is not attached to it – it does not affect natural hand motion during capture. For accessible natural occluders, such as the fences in Fig. 3, we acquire reference views by positioning the phone at a gap in the occluder – though this sometimes cannot perfectly remove the occluder as in the case of Fig. 3 *Pipes*. We collect data with our modified Pani capture app, illustrated in Fig. 1 (c), built on the Android camera2 API. During capture, we also record metadata such as camera intrinsics, exposure settings, channel color correction gains, tonemap curves, and other image processing and camera information during capture. We stream gyroscope and accelerometer measurements from on-board sensors as $\approx$100Hz, though we find accelerometer values to be highly unreliable for motion on the scale of natural hand tremor,

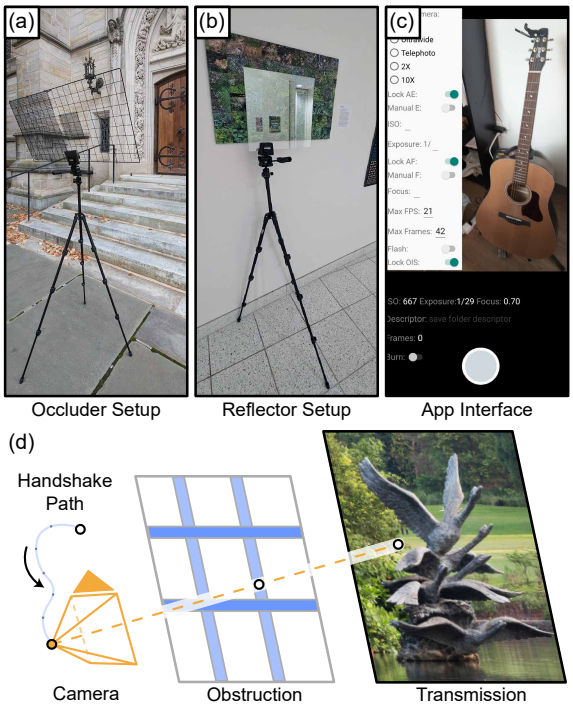


Figure 1. (a) Tripod-mounted occluder setup for capturing paired occlusion removal data. (b) Tripod-mounted reflector setup for capturing paired reflection removal data. (c) Capture app interface with the extended settings menu. (d-e) Example 3D scene with simulated occluder, camera frustum highlighted in orange.

and so disregard these measurements for this work. We apply minimal processing to the recorded 10-bit Bayer RAW frames – only correcting for lens shading and BGGR color channel gains – before splitting them into a 3-plane RGB color volume. We do not perform any further demosaicing on this volume, as these processes correlate local signal values, and instead input it directly into our model for scene fitting. For visualization, we apply the default color correction matrix and tone-curve supplied in the capture metadata.

Synthetic Data Generation Capturing aligned ground-truth data for obstruction removal is a long-standing problem in the field [64], greatly exacerbated by the requirement in our setting of *a sequence* of unstabilized frames with its base frame aligned to an unobstructed image. Thus, to help validate our method, we turn to synthetic captures created through image reprojection. We use 61-megapixel digital camera (Sony A7RIV) captures to simulate the transmission layer, and either hand-segmented occluders or a second 61-megapixel "reflection" image to simulate the obstruction. These are simulated as 3D planes in space at depths Π_z^{O} and Π_z^{T} respectively – $\Pi_z^{\mathrm{O}} < \Pi_z^{\mathrm{T}}$ for occluders and $\Pi_z^{\mathrm{O}} > \Pi_z^{\mathrm{T}}$ for reflectors – and apply a random tilt to the planes with angle $\theta \in [-20°, 20°]$. To generate realistic camera motion, we record samples of natural hand tremor with a pose-capture application built on the Apple ARKit library [10]. We then apply this motion path to a projective camera model, resample the image planes, and alpha-composite the outputs to produce the simulated burst stack. We emphasize that this data does not capture all the imaging effects present in real burst photography – e.g., lens distortion, scene deformation, motion blur, chromatic aberrations, or sensor and microlens defects – and use it as a tool for validating correct layer separation rather than a benchmark for overall performance. Reconstruction results for these simulated bursts are shown in Fig. 7 and Fig. 8.

Implementation Details While the overarching model structure is held constant between all applications – identical projection, image generation, and flow models for all tasks – elements such as the neural spline field $h(u, v)$ encoding parameters params_γ can be tuned for specific tasks:

$$h(u, v) = \mathbf{h}(\gamma(u, v;\, \mathrm{params}_\gamma);\, \theta)$$
$$\mathrm{params}_\gamma = \{\mathrm{B}^\gamma, \mathrm{S}^\gamma, \mathrm{L}^\gamma, \mathrm{F}^\gamma, \mathrm{T}^\gamma\}. \tag{13}$$

By manipulating the parameters of Eq. 13 as defined in Tab. 1 we construct four different "sizes" of network encodings: *Tiny*, *Small*, *Medium*, and *Large*. Image fitting results in Fig. 2 illustrate what scale of features each of these configurations is able to reconstruct, with larger encoding reconstructing denser and higher-frequency content. Then, assembling together multiple image and flow networks with varying encoding sizes as defined in Tab. 1, we are able to leverage this feature scale control for layer separation tasks such as occlusion, reflection, or shadow removal.

For tasks such as video segmentation, it is important that both the transmission layer and obstruction layer are able to represent high-resolution images, as the purpose here is to divide and compress video content into two canonical views, alpha matte, and optical flow. Hence for the video segmentation task in Tab. 1 both layers have *Large* network encodings. Conversely, for a task such as shadow removal we want to minimize the amount of color and alpha information the shadow obstruction layer is able to represent – as shadows, like the mask example in Fig. 2, are comprised of mostly low-resolution image features. Correspondingly, the shadow removal task in Tab. 1 has a *Tiny* image color encoding and only a *Medium* size alpha encoding. We keep these *parameters constant between all tested scenes* for clarity of presentation, however we emphasize that these model configurations are not prescriptive; all neural scene fitting approaches [48] have per-scene optimal parameters. Given the relatively fast training speed of our approach, approximately 3mins on a single Nvidia RTX 4090 GPU, in settings where data acquisition is costly – e.g., scientific imaging settings such as microscopy – it may even be tractable to sweep model parameters to optimally reconstruct each individual capture.

Size	base B^γ	scale S^γ	levels L^γ	feat. F^γ	table T^γ
Tiny (T)	4	1.61	**6**	4	**12**
Small (S)	4	1.61	**8**	4	**14**
Medium (M)	4	1.61	**12**	4	**16**
Large (L)	4	1.61	**16**	4	**18**

Table 1. Multi-resolution hash-table encoding parameters for different "sizes" of network, with larger encodings intended to fit higher-resolution data. Note that we only vary the number of grid levels L^γ, and match the backing table size T^γ accordingly to avoid hash collisions. The base grid resolution B^γ, grid per-level scale S^γ, and feature encoding size F^γ are kept constant.

occlusion removal:

	flow h	$\lvert h\rvert$	rgb f	f^α	depth Π_z	$\eta_\alpha \mathcal{R}$
Tr:	T	11	L		1.0	0.02
Ob:	T	11	M	M	0.5	

reflection removal:

	flow h	$\lvert h\rvert$	rgb f	f^α	depth Π_z	$\eta_\alpha \mathcal{R}$
Tr:	T	11	L		1.0	0.0
Ob:	T	11	T	L	2.5	

video segmentation:

	flow h	$\lvert h\rvert$	rgb f	f^α	depth Π_z	$\eta_\alpha \mathcal{R}$
Tr:	S	15	L		1.0	0.002
Ob:	S	15	L	M	2.0	

shadow removal:

	flow h	$\lvert h\rvert$	rgb f	f^α	depth Π_z	$\eta_\alpha \mathcal{R}$
Tr:	T	11	L		1.0	0.0
Ob:	T	11	T	M	2.0	

***dehazing*:**

	flow h	$\lvert h\rvert$	rgb f	f^α	depth Π_z	$\eta_\alpha \mathcal{R}$
Tr:	T	11	L		1.0	0.01
Ob:	T	11	T	S	0.5	

image fusion:

	flow h	$\lvert h\rvert$	rgb f	f^α	depth Π_z	$\eta_\alpha \mathcal{R}$
Tr:	S	31	L		1.0	0.0

Table 2. Network encoding, flow, and loss configurations used for several layer-separation applications, separated into rows individually defining transmission Tr and obstruction Ob layers. Encoding parameters are defined by the corresponding (T,S,M,L) row of Tab. 1. Flow size $|h|$ indicates the number of spline control points used for interpolation of the corresponding neural spline field $S(t, h(u, v))$.

B. Additional Reconstruction Results

In this section, we provide additional quantitative and qualitative obstruction removal results, comparing our proposed model against a range of multi-view and single-image methods. We include discussion of challenging imaging settings and potential directions of future work to address them.

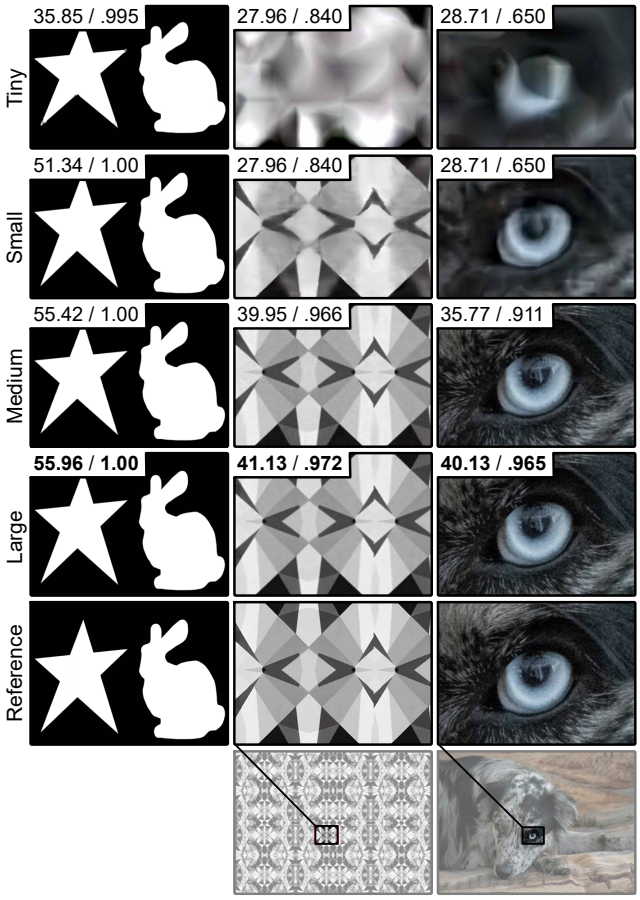


Figure 2. Image fitting results for network encoding configurations as described in Tab. 1, other training and network parameters held constant: 5-layer MLP coordinate networks, hidden dimension 64, ReLU activations. PSNR/SSIM values inset top-left.

Occlusion Removal We include a set of additional occlusion removal results in Fig. 3 with natural environmental occluders such as fences and grates. We evaluate our results against the multi-image learning-based obstruction removal method Liu et al. [43], the NeRF-based method OCC-NeRF [72], the flow plus homography neural image representation NIR [51], and the single image inpainting approach Lama [58] – to which we provide hand-drawn masks of the occlusion. We find that, as observed in the main text, the multi-image methods struggle to remove significant parts of the obstruction. Though in some scenes, the multi-image baselines are able to decrease the opacity of the occluder to reveal details behind it. Nevertheless, in all cases the obstruction is still clearly visible after applying each baseline. Given the small camera baseline setting of our input data, the volumetric OCC-NeRF approach struggles to converge on a cohesive 3D scene representation, producing blurred or otherwise inconsistent image re-

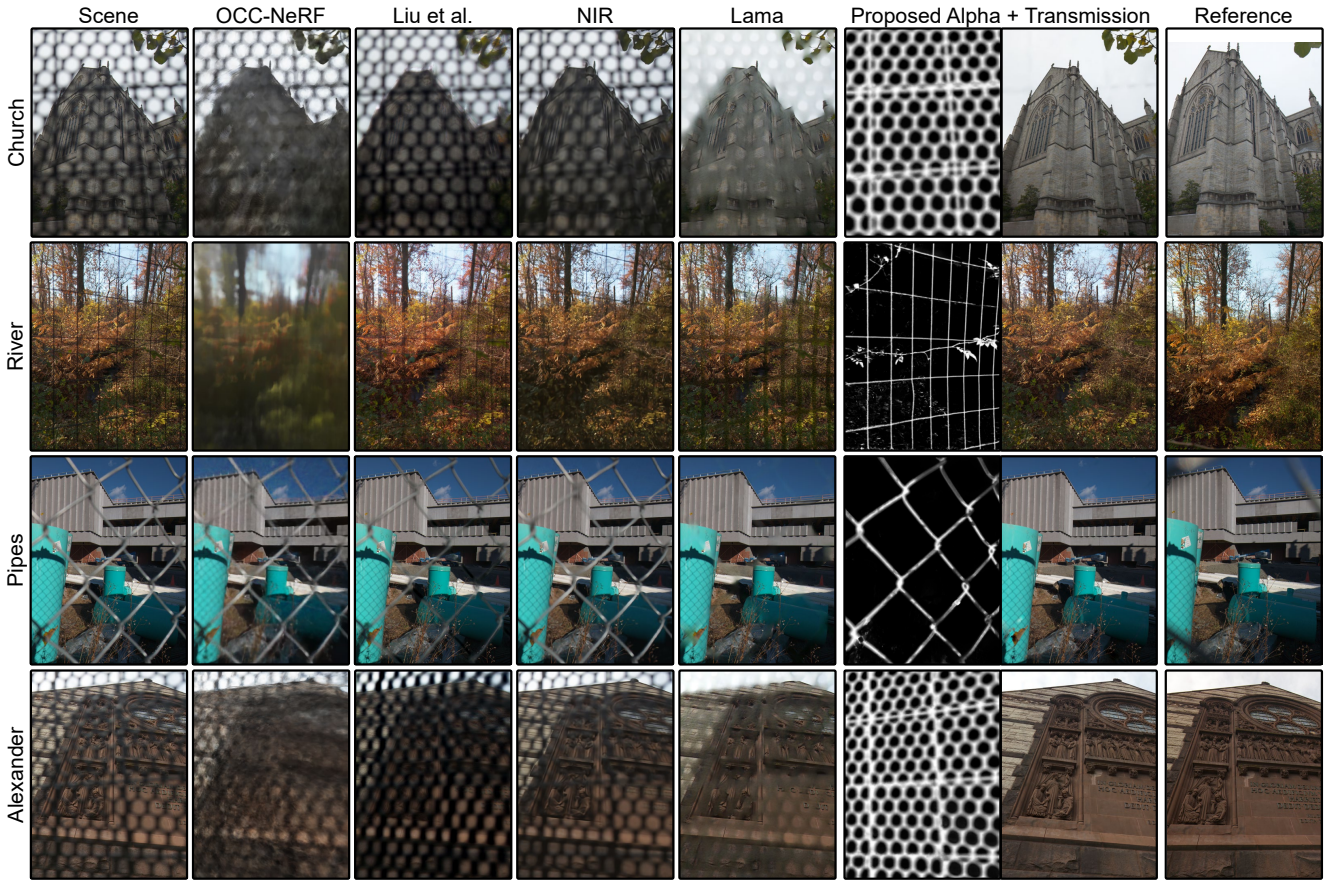


Figure 3. Occlusion removal results and estimated alpha maps for a set of captures with reference views, with comparisons to single image, multi-view, and NeRF fitting approaches. See video materials for visualization of input data and scene fitting.

constructions – as is the case for the *Church* scene. We find that the the homography-based NIR method also struggles in this small baseline setting, often identifying the entire scene as the canonical view rather than partly obstructed. Given hand annotated masks, single image methods such as DALL·E and Lama [58] can successfully inpaint sparse occluders such as the fence in the *Pipes* scene, but struggle to recover content behind dense occluders such as in *Alexander* and *Church* in Fig. 3. As they have no way to aggregate content between frames, they "recover" hidden content from visual priors on the scene, which may not be reliable when the scene is severely occluded.

In contrast, our method automatically distills a high-quality alpha matte for the obstruction and reconstructs the underlying transmission layer using information from multiple views. This mask is of similar quality regardless of whether the scene is obstructed by a dense occluder or a sparse occluder, so long as there is sufficient parallax between the two layers. The depth-separation properties of our alpha estimation are showcased in the *River* example, where the obstruction layer isolated not only the grid of the fence, but also the branches and leaves weaved through the fence. Our method reconstructs the transmitted layer behind the occlusion with favorable results compared to all baseline methods.

Reflection Removal For reflection removal, we compare with the reflection-aware NeRF-based method NeRFReN [19] in addition to NIR [51], Liu et al. [43], and the single-image reflection removal method DSRNet [25]. We show reflection removal results in Fig. 4. We observe results with a similar trend to those in the obstruction removal task. The volumetric method NeRFReN struggles to reconstruct a high-fidelity scene representation, as Liu et al. and NIR also struggle with the small baseline of the camera motion. The single-image method DSRNet performs best among the baselines, as it has no priors on image motion. However, without the ability to draw information from multiple views, DSRNet uses learned priors to disambiguate reflected and transmitted content. This appears not to be very effective for high opacity reflections, such as the *Leaves* example and the phone in the *Plaque* scene. Our method achieves the highest-quality reconstruction and layer separation among all methods tested, across all scenes, with our estimated obstruction revealing the detailed structure of the scene being reflected. In Fig. 6 we also showcase our model's performance on challenging, in-the-wild scenes where we do not

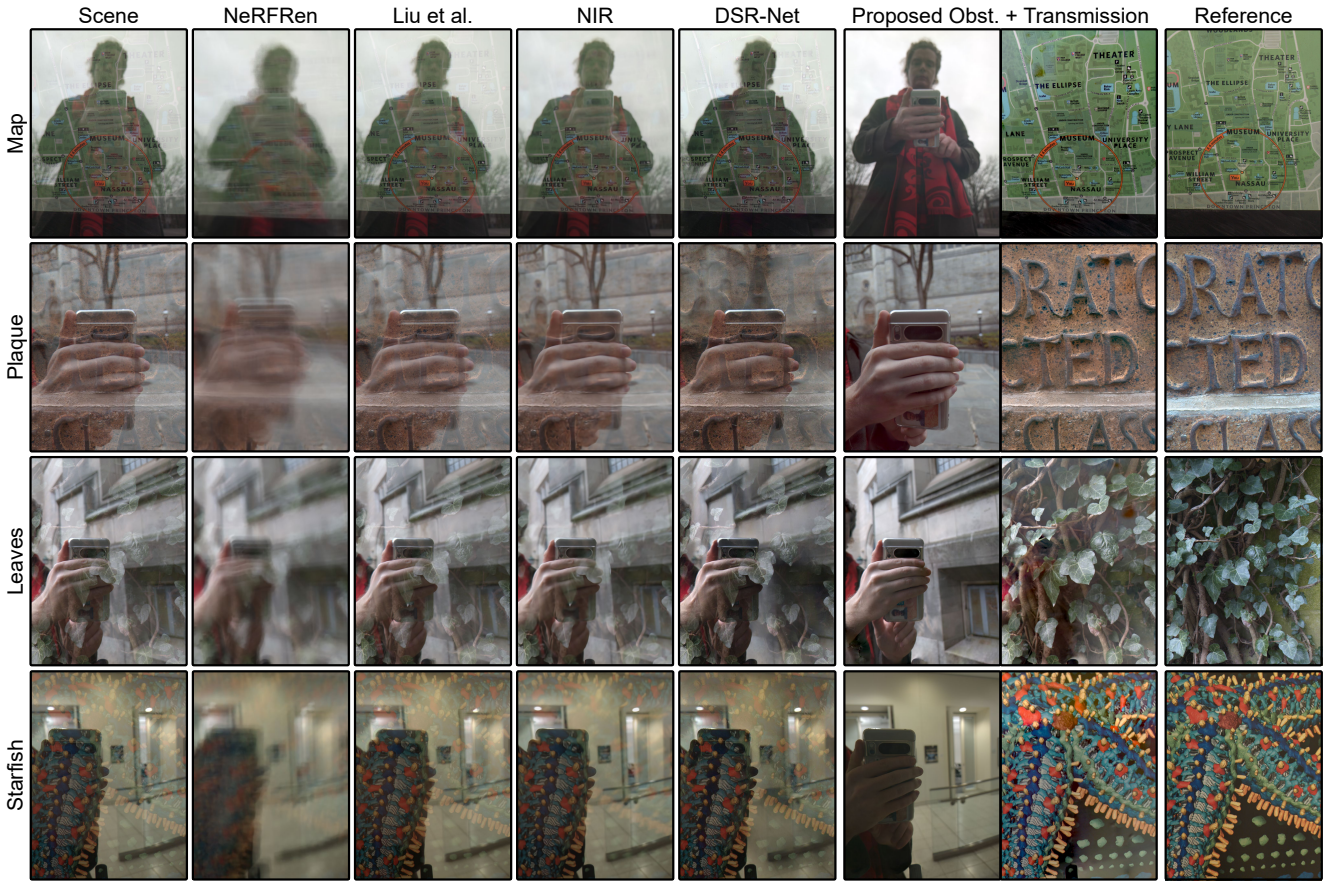


Figure 4. Reflection removal results and estimated alpha maps for a set of captures with reference views, with comparisons to single image, multi-view, and NeRF fitting approaches. See video materials for visualization of input data and scene fitting.

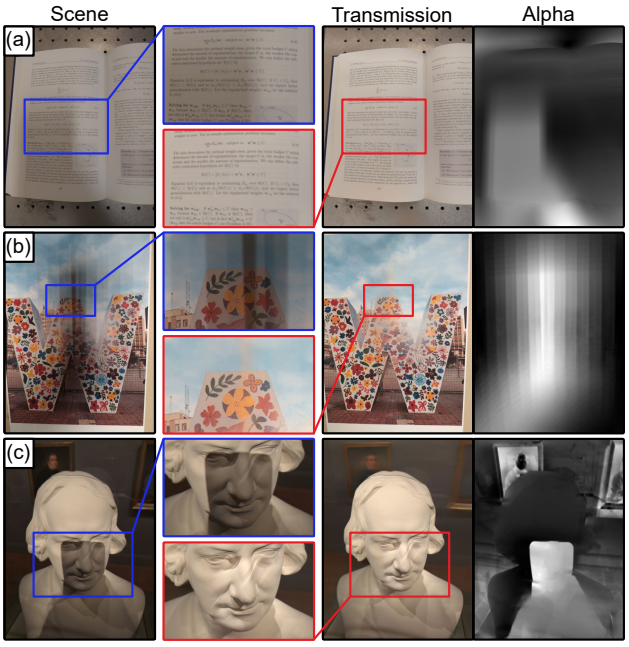
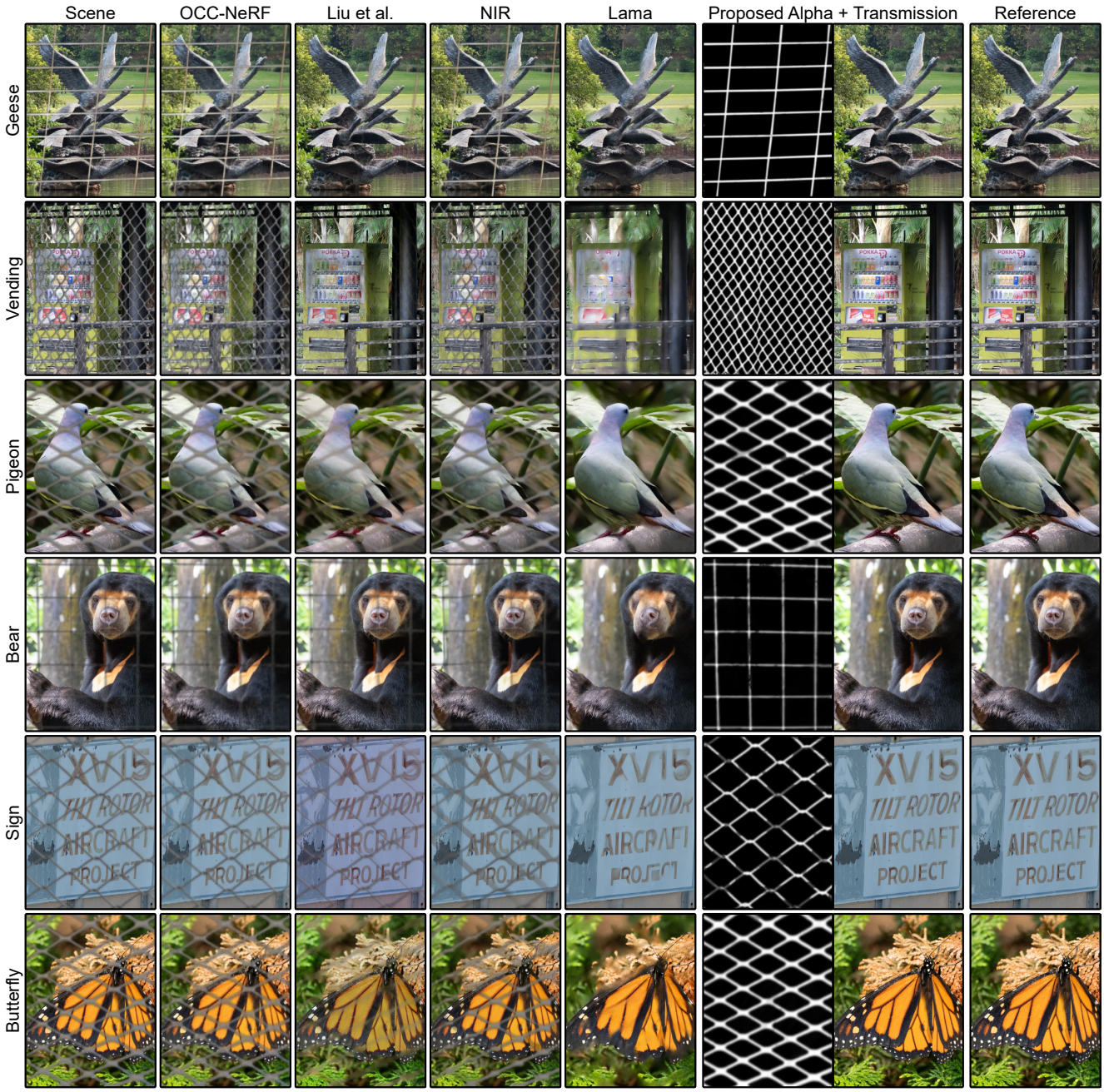


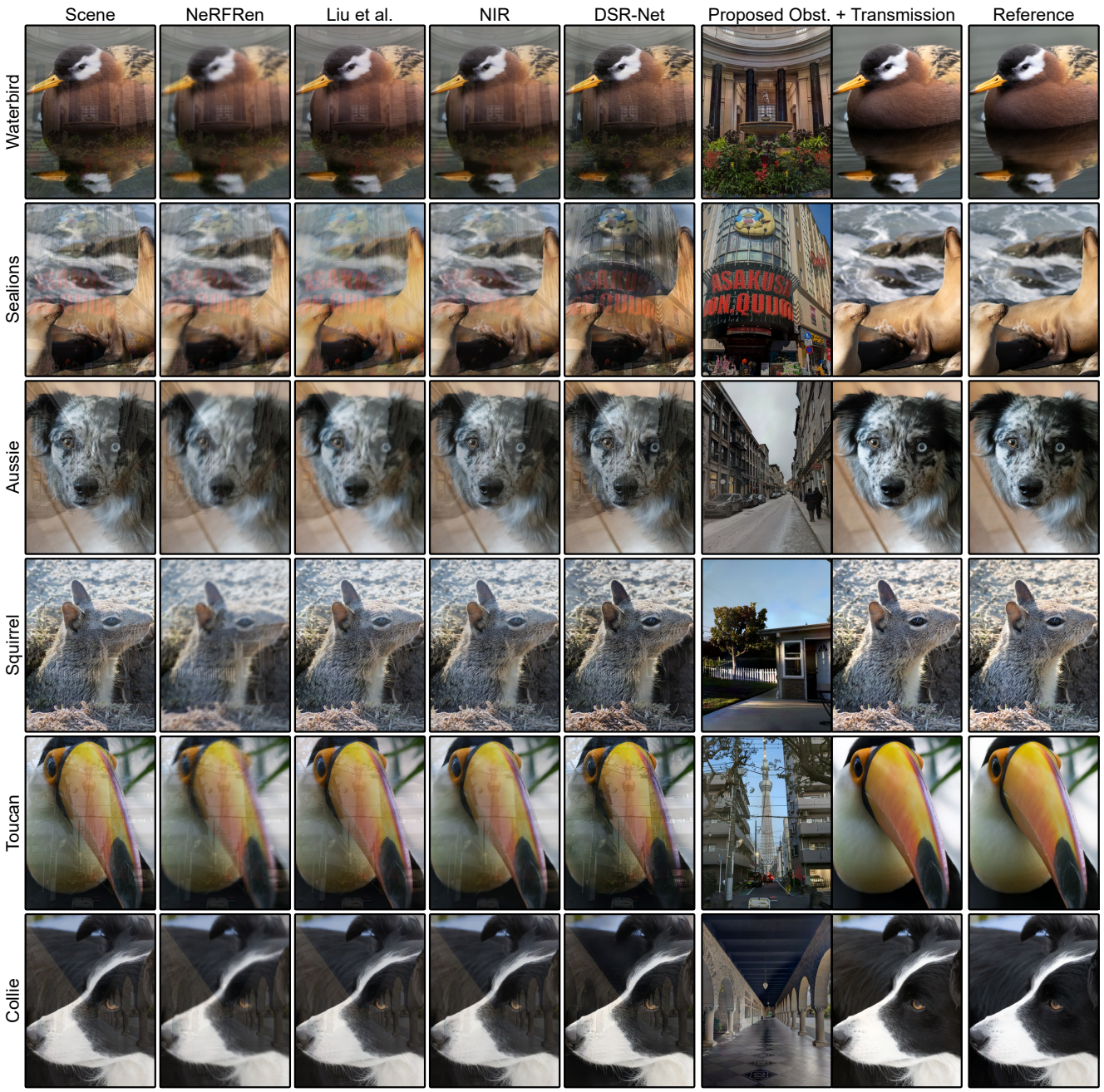
Figure 5. Shadow removal results under different lighting conditions: (a) partially diffuse, (b) multiple point, (c) single point.

Figure 6. Reflection removal results for challenging in-the-wild scenes: (a) storefront window, (b) poster, (c) museum painting.



Occlusion	OCC-NeRF	Liu et al.	NIR	Lama	Proposed	**Occlusion**	OCC-NeRF	Liu et al.	NIR	Lama	Proposed
Geese	19.49/0.578	32.24/0.970	20.89/0.696	21.96/0.760	**41.80**/**0.986**	*Vending*	18.05/0.550	15.10/0.754	17.96/0.625	17.42/0.591	**39.62**/**0.981**
Pigeon	18.60/0.691	15.17/0.725	18.74/0.691	21.55/0.753	**40.33**/**0.965**	*Bear*	23.72/0.696	26.32/0.930	23.28/0.746	23.84/0.815	**40.88**/**0.980**
Sign	24.34/0.870	24.11/0.952	22.84/0.905	28.57/0.932	**48.63**/**0.994**	*Butterfly*	17.67/0.674	15.43/0.828	18.25/0.750	17.89/0.722	**39.53**/**0.980**

Figure 7. Qualitative and quantitative occlusion removal results for a set of 3D rendered scenes with paired ground truth. Evaluation metrics formatted as PSNR/SSIM.



Reflection	NeRFReN	Liu et al.	NIR	DSR-Net	Proposed	**Reflection**	NeRFReN	Liu et al.	NIR	DSR-Net	Proposed
Waterbird	21.94/0.695	23.68/0.811	24.08/0.751	19.95/0.753	**39.16**/**0.982**	*Sealion*	20.28/0.811	11.45/0.726	22.36/0.899	13.27/0.657	**32.31**/**0.993**
Aussie	18.88/0.561	18.09/0.634	20.54/0.665	19.56/0.738	**30.90**/**0.971**	*Squirrel*	17.15/0.431	23.55/0.950	22.04/0.789	19.05/0.860	**33.34**/**0.988**
Toucan	19.98/0.817	21.14/0.837	21.67/0.873	17.63/0.717	**36.00**/**0.985**	*Collie*	18.60/0.706	22.34/0.862	22.08/0.801	21.96/0.847	**32.98**/**0.978**

Figure 8. Qualitative and quantitative reflection removal results for a set of 3D rendered scenes with paired ground truth. Evaluation metrics formatted as PSNR/SSIM.

have the ability to acquire reference views. We observe robust reflection removal, matching the reconstruction quality observed for scenes acquired with our tripod setup.

Validation on Synthetic Scenes We generate synthetic scenes as described in Sec. A, and compare our obstruction removal results to the same baselines outlined in the previous sections, including: OCC-NeRF [72], NeRFReN [19], Liu et al. [43], NIR [51], Lama [58] and DSRNet [25]. We show quantitative and qualitative results for occlusion removal and reflection removal in Fig. 7 and Fig. 8 respectively. We also provide NeRF-based methods with ground truth camera poses, which results in higher fidelity NeRF-based reconstruction than on real-world data. Overall, we observe similar trends to the real-world examples, with most multi-image based methods failing to remove the majority of the obstructions for the majority of scenes. This is with the exception of Liu et al. [43] for the *Geese*, *Vending* and *Butterfly* scenes in Fig. 7, where it succeeds at removing a large portion of the fence occluders. We believe this is a strong indication that this method relies heavily on visual cues to identify the occluder (e.g., gray mostly-in-focus fences), and helps to explain its failure to identify and remove other categories of obstructions such as the black hexagonal grids in Fig. 3. Lama [58], when provided with a ground-truth occlusion mask, is able to reconstruct a relatively coherent transmission layer. However, upon closer inspection the results are missing details in the ground-truth transmission layer, such as the distorted text in *Sign* and missing beak of *Pigeon* in Fig. 7. We observe that both multi-image methods and DSRNet [25] fail to effectively remove reflections in Fig. 8, with DSRNet [25] accidentally enhancing the reflected content in the *Sealions* scene. These observations are supported by quantitative results, with our method achieving the highest PSNR and SSIM across all scenes tested. We observe an average PSNR increase of more than 10db, with near-perfect reconstruction of both obstructions and obstructed content; though emphasize that these results represent a validation of the models in a simplified imaging setting, and are not fully representative of performance across diverse in-the-wild scenarios.

Shadow Removal In Fig. 5 we demonstrate shadow removal results for scenes with disparate lighting conditions: (a) a book illuminated by a diffuse overhead lamp, (b) a poster illuminated by an array of LEDs, and (c) a bust illuminated by a strong point light source. We note that the grid of LEDs act as a set of point light sources, producing multiple copies of the shadow to be overlayed on the scene. In all settings we are able to extract the shadow with the same obstruction network defined in the *shadow removal* application in Tab. 2, further reinforcing the our image fitting findings from Fig. 2. Namely that coordinate networks with low-resolution multi-resolution hash encodings are able to effectively fit both scenes comprised of smooth gradients,

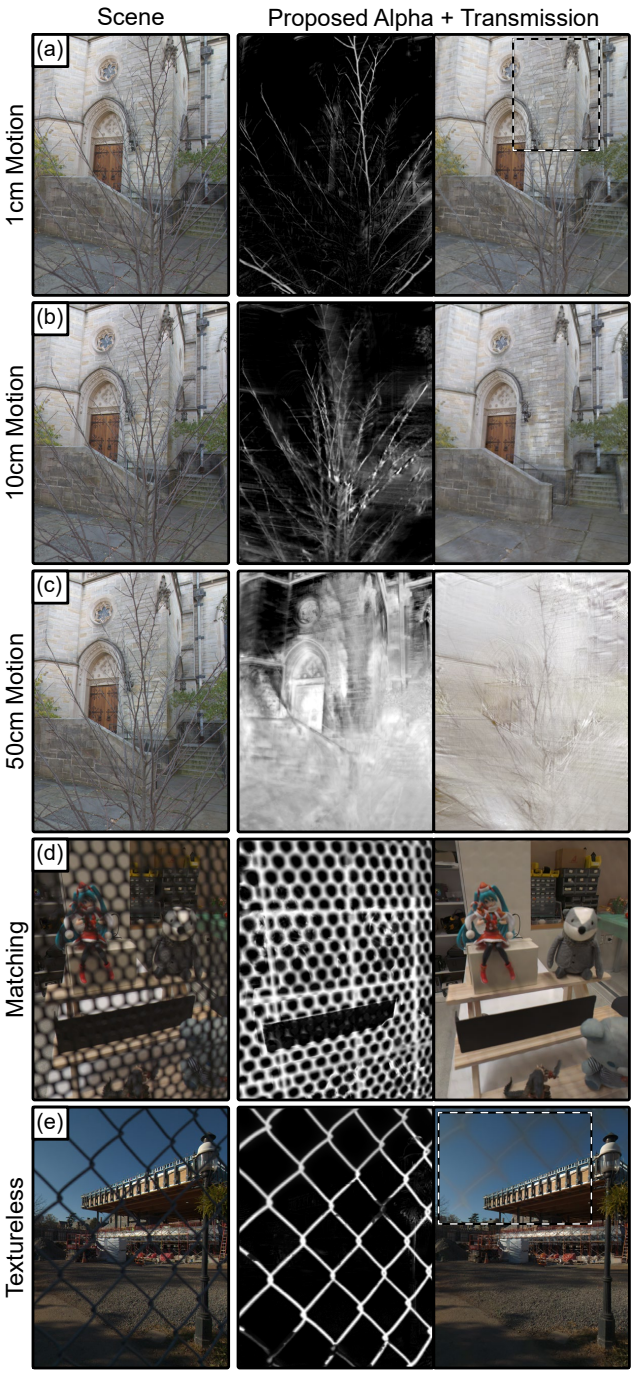


Figure 9. Challenging image reconstruction cases including varying scales of camera motion, overlap between occluder and transmission colors, and residual signal left on scene content in low-texture regions. Areas of interest highlighted with dashed border.

as in the diffuse shadow case, and limited numbers of image discontinuities, as in the multiple point source case. In (c) we furthermore see that while the photographer-cast shadow is successfully removed from the bust, the shadows cast by other light sources are left intact. This reinforces

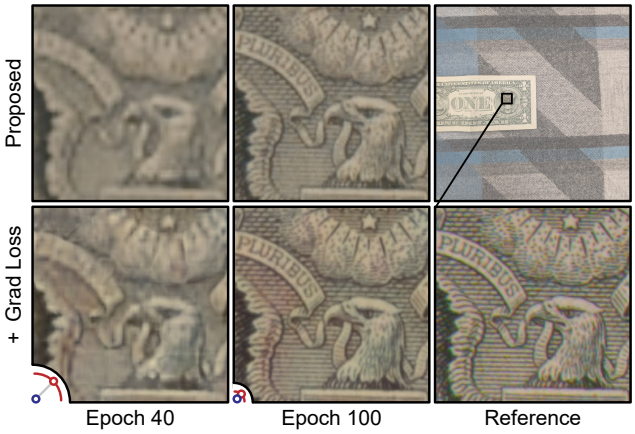


Figure 10. Visualization of the effects of gradient loss $\mathcal{L}_{\text{G}}$ on image reconstruction at 25x zoom. Inset bottom left is the radius of perturbation at epoch 40 and epoch 100, the end of training.

that our proposed model is separating shadows based not only on their color, but on the motion they exhibit in the scene; as the other shadows cast on the bust undergo the same parallax motion as the bust itself.

Challenging Settings We compile a set of challenging imaging settings in Fig. 9 which highlight areas where our proposed approach could be improved. One limitation of our work is that it cannot generate unseen content. While this means it cannot hallucinate features from unreliable image priors, it also means that it is highly parallax-dependent for generating accurate reconstructions. This is highlighted in Fig. 9 (a-c), where with hand motion on the scale of 1cm is only enough to separate and remove the topmost branch of the occluding plant. Motion on the scale of 10cm is enough to remove most of the branches, but larger motion on the scale of half a meter in diameter causes the reconstruction to break down. This is likely due to the small motion and angle assumptions in our camera model, as it is not able to successfully jointly align the input image data and learn its multi-layer representation. Thus work on large motion or wide-angle data for large obstruction removal – e.g., removing telephone poles blocking the view of a building – remains an open problem. Fig. 9 (d) demonstrates the challenge of estimating an accurate alpha matte when the transmitted and obstructed content are matching colors. In this case, although the obstruction is "removed", we see that the alpha matte is missing a gap around the black object in the scene behind the occluder. In this region the model does not need to use the obstruction layer to represent pixels that are already black in the transmission layer – in fact, the alpha regularization term $\mathcal{R}_\alpha$ would penalize this. Thus the alpha matte is actually a produce of both the actual alpha of the obstruction and its relative color difference with what it is occluding. Fig. 9 (e) highlights a related problem. In regions where the transmission layer is low-texture, and lacks parallax cues, it is ambiguous what is being obstructed and where the border of the obstruction lies. Thus ghosting artifacts are left behind in areas such as the sky of the *Textureless* scene. What is noteworthy, however, is that these are also exactly the regions in which in-painting methods such as Lama [58] are most successful, as there are no complex textures that need to be recovered from incomplete data, leaving a hybrid model as an interesting direction for future work.

C. Additional Experiments and Analysis

Gradient Loss A significant challenge posed by the task of aggregating long-burst data is the so-called problem of "regression to the mean". When minimizing a metric such as relative mean-square error, which penalizes small color differences significantly less than large discrepancies, the final reconstruction is encouraged to be smoother than the original input data [2]. Thus, in developing our approach we explored – but ultimately did not use – a form of gradient penalty loss:

$$\mathcal{L}_{\text{G}} = |(\Delta c - \Delta \hat{c})/(\text{sg}(\Delta c) + \epsilon)|^2.$$

Rather than sample a grid of points around $u^{\text{O}}, v^{\text{O}}$ and $u^{\text{T}}, v^{\text{T}}$ or perform a second pass over the image networks [51] to compute Jacobians, we compute color gradients Δc by pairing each ray with an input perturbed in a random direction

$$\begin{aligned} \Delta c &= I(u, v, t) - I(\tilde{u}, \tilde{v}, t) \\ \tilde{u}, \tilde{v} &= u + r\cos(\phi),\ v + r\sin(\phi), \quad \phi \sim \mathcal{U}(0, 2\pi), \end{aligned} \tag{14}$$

where r determines the magnitude of the perturbation. The estimated color gradient $\Delta \hat{c}$ is similarly calculated for the output colors of our model. Illustrated in Fig. 10, by reducing radius r from multi-pixel to sub-pixel perturbations during training, we are able to improve fine feature recovery in the final reconstruction via gradient loss $\mathcal{L}_{\text{G}}$ without significantly impacting training time – as perturbed samples are also re-used for regular photometric loss calculation $\mathcal{L}_p$. However, as we do not apply any demosaicing or post-processing to our input Bayer array data, we find this loss can also lead to increased color-fringing artifacts – the red tint in the bottom row of Fig. 10. For these reasons, and poor convergence in noisy scenes, we did not include this loss in the final model. However, there may be potentially interesting avenue of future research into a jointly trained demosaicing module to robustly estimate real color gradient directly from quantized and discretized Bayer array values.

Alpha Regularization Ablation In Fig. 12, we visualize the effects of alpha regularization weight η_α on reconstruction. The primary function of this regularization is remove low-parallax content from the obstruction layer, as there is no alpha penalty for reconstructing the same content via the transmission layer. As seen in the *Pipes* example, without

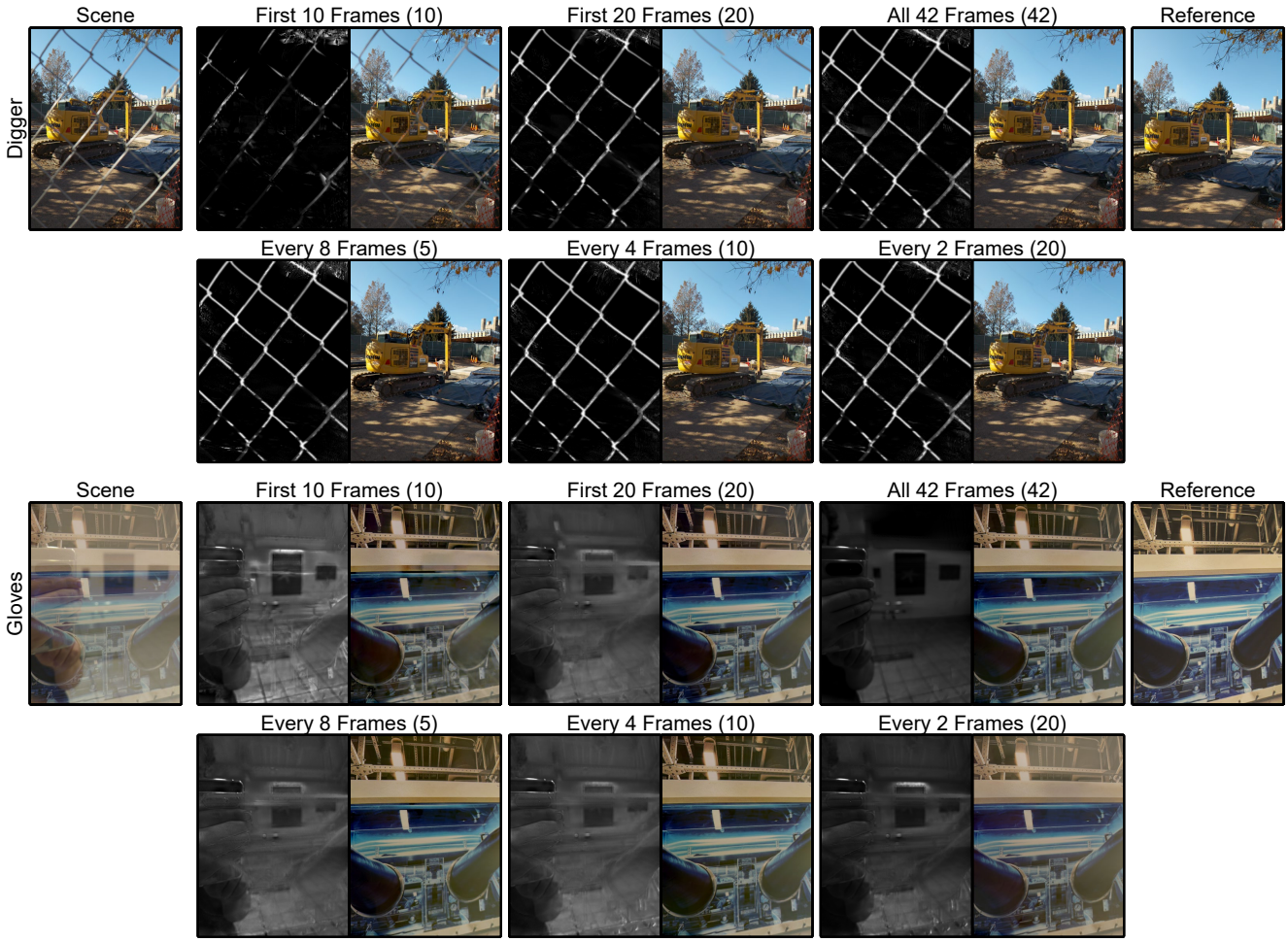


Figure 11. Ablation study on the effects of the number of input frames or duration of capture on transmission layer reconstruction and estimated alpha matte. Total number of frames input into the model denoted by the number in parentheses– e.g., (10) = ten frames.

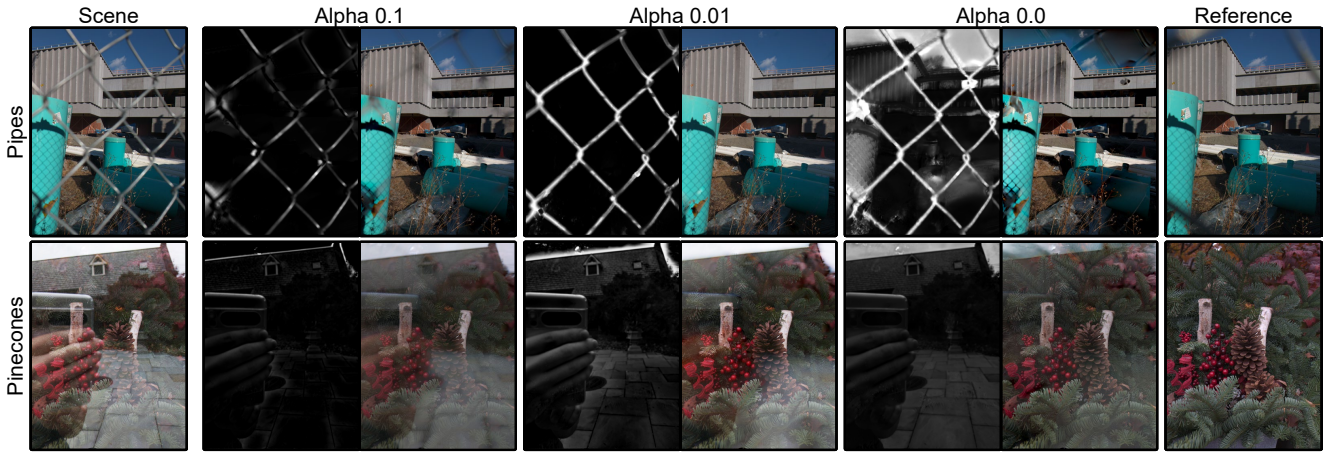


Figure 12. Ablation study on the effects of alpha regularization weight η_α on transmission layer reconstruction and estimated alpha matte.

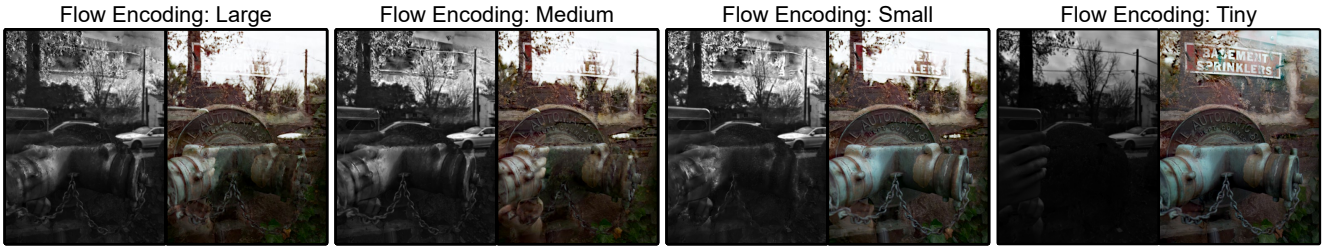


Figure 13. Ablation study on the effects of flow encoding size (Tab. 1) on transmission layer reconstruction and estimated alpha matte.

Figure 14. Demonstration of user-interactive scene editing facilitated by layer separation. Only the user-selected region of the obstruction, highlighted in red, is removed without affecting surrounding scene content, see text.

alpha regularization the obstruction layer is able to freely reconstruct part of the transmitted scene content such as the sky, the pipes, and the walls of the occluded buildings. A small penalty of $\eta_\alpha = 0.01$ is enough to remove this unwanted content from the obstruction layer, while $\eta_\alpha = 0.1$ is enough to also start removing parts of the actual obstruction. Contrastingly, in the case of reflection scenes such as *Pinecones*, even a relatively small alpha regularization weight of $\eta_\alpha = 0.01$ removes part of the actual reflection – leaving behind a grey smudge in the bottom right corner of the reconstruction. As reflections are typically partially transparent obstructions, and can occupy a large area of the scene, removing them purely photometrically is ill-conditioned. There is no visual difference between a gray reflector covering the entire view of the camera and the scene actually being gray. Thus η_α can also be a user-dependent parameter tuned to the desired "amount" of reflection removal.

Frame Count Ablation Thusfar we have used all 42 frames in each long-burst capture as input to our method, but we highlight that this is not a requirement of the approach. The training process can be applied to any number of frames – within computational limits. In Fig. 11 we showcase reconstruction results for both subsampled captures, where only every k-th frame of the image sequence is kept for training, and shortened captures, where only the first n frames are retained. Similar to the problem of depth reconstruction [9], we find that obstruction removal performance directly depends on the total amount of parallax in the input. Sampling the *first* 10 frames – approximately 0.5 seconds of recording – results in diminished obstruction removal for both the *Digger* and *Gloves* scenes as the obstruction exhibits significantly less motion during the capture. In contrast, given *a five frame input sampled evenly across the full two-second capture*, our proposed approach is able to successfully reconstruct and remove the obstruction. This subsampled scene also *trains considerably faster*, converging in only 3 minutes as less frames need to be sampled per batch – or equivalently more rays can be sampled from each frame for each iteration. This further validates the benefit of a long burst capture.

Flow Encoding Size Ablation A key model parameter which controls layer separation, as discussed in Section A, is the size of the encoding for our neural spline flow fields. In Fig. 13 we illustrate the effects on obstruction removal of over-parameterizing this flow representation. When the two layers are undergoing simple motion caused by parallax from natural hand tremor, a *Tiny* flow encoding is able to represent and pull apart the motion of the reflected and transmitted content. However, high-resolution neural spline fields, just like a traditional flow volume $h(u, v, t)$, can quickly overfit the scene and mix content between layers. We can see this clearly in the *Large* flow encoding example where the reflected phone, trees, and parked car appear in both the obstruction alpha matte and transmission image. Thus it is critical to the success of our method to construct a task-specific neural spline field representation appropriate for the expected amount and density of scene motion.

Applications to Scene Editing In Fig. 14 we showcase the scene editing functionality facilitated by our proposed methods layer separation. As we estimate an image model for both the transmission and obstruction, we are not limited to only removing a layer but can independently manipulate them. In this example we rasterize both layers to RGBA im-

ages and input them into an image editor. The user is then able to highlight and delete a portion of the occlusion while retaining its other content. Thus we can create physically unrealizable photographs such as only the fence appearing to be behind the *Digger*, or selectively remove the photographer's hand and parked car from the *Hydrant* scene.