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. Validated on complex synthetic and 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, 39], 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 [32, 45] and aggregation [6, 63] of pixels into an image array – referred to as the align-and-merge process.

While existing approaches treat pixel motion as a source of noise and artifacts, a parallel direction of work [10, 21, 69] 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. [50] 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, 54]. 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 [50] with a projective camera model [9] and novel neural spline flow 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 removed to perform occlusion removal, reflection suppression, or shadow removal. 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 generic flow components which account for depth parallax, scene motion, and other effects. 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 [44] – and fuses them post-capture to produce a demosaiced [58], denoised [16, 45], superresolved [32, 63], or otherwise enhanced reconstruction. Almost all modern smartphone devices rely on burst photography for low-light [23, 39] 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, 69] 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 [65] 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 [54] or detecting lattices for fence deletion [52] – all single-image obstruction removal is fundamentally a segmentation [31, 41] 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, 64] to synthesize missing content. This is in contrast to approaches which rely on multiple measurements such as multi-focal stacks [1, 53], multi-view images [42, 51], flash no-flash pairs [33, 35], or polarization data [34]. These methods typically treat obstruction removal as an inverse problem [5], estimating a model of transmitted and occluded content consistent with observed data [36]. 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 [50] 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 [67], 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 [59] to outputs such as RGB [55], depth [10], or x-ray data [56]. 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 [48] and inference [68] 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 [30]. Several recent approaches make use of neural scene rep-
representations in tandem with continuous motion estimation models to fit multi-image [9] and video [38] data, potentially decomposing it into multiple layers in the process [28, 50]. 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 [60, 61] 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, 69]. This can be represented as

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

(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 \( \pi \) and \( \pi ^{-1} \) 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 as [28, 50] estimate a generic flow model \( \Delta u, \Delta v = g(u, v, t) \) to re-sample the image model

\[ I(u, v, t) = f(u + \Delta u, v + \Delta v). \]  

(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 [38]. To combat this, methods often employ a form of gradient penalty such as total variation loss [50]. That is

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

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 which provides strong controls on reconstruction directly through its parametrization. This model splits flow evaluation into two components

\[ \Delta u, \Delta v = g(u, v, t) = S(t; P = h(u, v)). \]  

(3)

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

\[ S(t, P) = \left( 2t_r^3 - 3t_r^2 + 1 \right) P_{\lfloor t_s \rfloor} + \left( -2t_r^3 + 3t_r^2 \right) P_{\lfloor t_s \rfloor + 1} + \left( t_r^3 - 2t_r^2 + t_r \right) \left( P_{\lfloor t_s \rfloor} - P_{\lfloor t_s \rfloor - 1} \right)/2 + \left( t_r^3 - t_r^2 \right) \left( P_{\lfloor t_s \rfloor + 1} - P_{\lfloor t_s \rfloor} \right)/2 \]

\[ t_r = t_s - \lfloor t_s \rfloor, \quad t_s = t \cdot |P|, \]

(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 \( S(t; P) \). E.g., if the motion is expected to be a straight line, a piece-wise linear spline with \( |P| = 2 \) control points would insure this constraint is satisfied regardless of the outputs of the NSF.
Figure 3. Image fitting results coordinate networks with Small ($L^k=8$) and Large ($L^k=16$) multi-resolution hash encodings and identical other parameters; PSNR/SSIM values inset top-left. Unlike a traditional band-limited representation [66], 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 $S(t, P)$ and $|P|$ 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. [48]

$$h(u, v) = h(\gamma(u, v; \text{params}_r); \theta) \quad \text{params}_r = \{B^3, S^3, L^\gamma, F^\gamma, T^\gamma\},$$

as it allows for fast training and strong spatial controls given by its encoding parameters $\text{params}_r$: base grid resolution $B^3$, per level scale factor $S^3$, number of grid levels $L^\gamma$, feature dimension $F^\gamma$, and backing hash table size $T^\gamma$. Here, $h(\gamma(u, v); \theta)$ is a multi-layer perceptron (MLP) [24] with learned weights $\theta$. Illustrated in Fig. 3 with an image fitting example, the number of grid levels $L^\gamma$—which, with a fixed $S^3$, 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 [28,50]. However, handheld burst photography does not produce arbitrary image sequences; it has well-studied photometric and geometric properties [9,10,21,63]. 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 [47]. For simplicity of notation, we outline this model for a single projected ray below. We also illustrate this process in Fig. 4. Let

$$c = [R, G, B]^T = I(u, v, t)$$

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 top left corner of the image. To project these points into world space we introduce camera translation $T(t)$ and rotation $R(t)$ models

$$T(t) = S(t, P^T), \quad R(t) = R^0(t) + \eta_t S(t, P^R)$$

$$P^T_i = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad P^R_i = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}.$$  

Here $S(t, P)$ is the same cubic spline model from Eq. (4), evaluated element-wise over the channels of $P$. We note there are no coordinate networks employed in these models. Translation $T(t)$ is learned from scratch, $P^T_i$ initialized to all-zeroes. Rotation $R(t)$ is learned as a small-angle approximation offset [26] to device rotations $R^0(t)$ 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

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

where $D$ is normalized by its $z$ component. We define our transmission and obstruction image planes as $\Pi^T$ and $\Pi^0$, respectively. As XY translation of these planes conflicts with changes in the camera pose, we lock them to the $z$-axis at depth $\Pi_z$ with canonical axes $\Pi_u$ and $\Pi_v$. 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

$$u^0, v^0 = \langle Q, \Pi_u \rangle / (\Pi_z - O_z), \langle Q, \Pi_v \rangle / (\Pi_z - O_z),$$

scaled by ray length to preserve uniform spatial resolution. Let $u^T, v^T$ and $u^0, v^0$ be the intersection coordinates for the transmission and obstruction plane, respectively. We alpha
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.

composite these layers along the ray as

\[
\hat{c} = (1 - \alpha)c^T + \alpha c^O
\]

\[
c^T = f^T(u^T + \Delta u^T, v^T + \Delta v^T), \quad \Delta u^T, \Delta v^T = S(t; h^T(u^T, v^T))
\]

\[
c^O = f^O(u^O + \Delta u^O, v^O + \Delta v^O), \quad \Delta u^O, \Delta v^O = S(t; h^O(u^O, v^O))
\]

\[
\alpha = \sigma(\tau_\sigma f^\alpha(u^O + \Delta u^O, v^O + \Delta v^O)),
\]

where \(\hat{c}\) is the composite color point, the weighted sum by \(\alpha\) 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 \(\tau_\sigma\) 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 \(\tau_\sigma\) 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

\[
\mathcal{L} = \mathcal{L}_p + \eta_\alpha \mathcal{R}_\alpha
\]

\[
\mathcal{L}_p = |(c - \hat{c})/(\text{sgn}(c) + \epsilon)|, \quad \mathcal{R}_\alpha = |\alpha|,
\]

where \(\mathcal{L}_p\) is a relative photometric reconstruction loss [9, 46]. Demonstrated in Fig. 5, when combined with linear RAW input data, this loss proves robust in noisy imaging settings [46]. Thus, we select it for in-the-wild scene reconstruction with unknown conditions. The regularization \(\mathcal{R}_\alpha\) with weight \(\eta_\alpha\) controls the total contribution of the obstruction layer, discouraging it from duplicating content represented by 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. During training, we mask the multi-resolution hash encodings \(\gamma_i(u, v)\) input into our image, alpha, and flow networks as

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

\[
\text{sin\_epoch} = \sin(\text{epoch}/\text{max\_epoch}),
\]

activating higher resolution grids as training progresses. This strategy results in less noise accumulated during early training as spurious high-resolution features do not need to be “unlearned” [9, 37] during later stages of refinement.

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.

**Implementation Details.** During training, we perform stochastic gradient descent on $L$ 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 [49]. 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 $B^\gamma=4$, and scale factor $S^\gamma=1.61$, but while flow networks $h^T$ and $O$ are parameterized with a low number of grid levels $L^\gamma=8$, networks which represent high frequency content have $L^\gamma=12$ or $L^\gamma=16$ 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 $\Pi_0 < \Pi_T$, we find that the $f^O(u,v)$ 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 re-

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.
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. [42], the neural radiance field approach OCC-NeRF [70], the flow + homography neural image model NIR [50], and the single image inpainting method Lama [57] 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, it does not produce an alpha matte, and rather requires a hand-annotated mask as input. 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. 9 we present additional layer separation results for real in-the-wild scenes with complex occluders, which demonstrate the versatility of the obstruction image model $f^o(u, v)$. 

Figure 8. 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 9. Layer separation results for additional example applications: (a) shadow removal, (b) image dehazing, and (c) video motion segmentation (see video materials for visualization).
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 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, $f_i^\Pi(u, v)$ with $h^\Pi(u, v)$, and $f^\Omega(u, v)$ with $h^\Omega(u, v)$, 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, 43].

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. The proposed model relies entirely on photometric loss, with no visual priors on the scene. Learned features could help disambiguate content 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.
References


[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


[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


[28] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing, 2021. 3, 4, 8
[33] 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
[34] 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
[35] Chenyang Lei, Xudong Jiang, and Qifeng Chen. Robust reflection removal with flash-only cues in the wild, 2022. 2
[49] Seonghyeon Nam, Marcus A. Brubaker, and Michael S. Brown. Neural image representations for multi-image fusion and layer separation, 2022. 1, 2, 3, 4, 7, 8, 14, 18
[56] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov,


[70] Chengxuan Zhu, Renjie Wan, Yunkai Tang, and Boxin Shi. Occlusion-free scene recovery via neural radiance fields. 2023. 7, 14
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. 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 ≈100Hz, though we find accelerometer values to be highly unreliable for motion on the scale of natural hand tremor, 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 [62], 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 generate data for further validation of our method, we turn to synthetic captures created through the 3D rendering software Blender. We select a set of 3D scenes from the BlenderKit asset library and manually tune their lighting and materials to reduce non-physical rendering artifacts, such as the replacement of point light sources with small radius spherical sources. We add occluder and reflector objects to these scenes in the form of fences, window gratings, and glass enclosures containing textured meshes. 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 place cameras into the scene pointed at the synthetic obstructors, illustrated in Fig. 1, and offset their rotations and translations over time with the collected natural hand tremor data. We then render this data in the Cycles ray-tracing engine with 4000 samples per ray. Reconstruction results with this data are demonstrated in Fig. 5 and Fig. 6.

**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 $h(u, v)$ en-
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 

\[ \gamma \] 

weight 0.005 

\[ \gamma \] 

table 

<table>
<thead>
<tr>
<th>Size</th>
<th>base</th>
<th>scale</th>
<th>levels</th>
<th>feat.</th>
<th>table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiny</td>
<td>4</td>
<td>1.61</td>
<td>6</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Small</td>
<td>4</td>
<td>1.61</td>
<td>8</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>Medium</td>
<td>4</td>
<td>1.61</td>
<td>12</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Large</td>
<td>4</td>
<td>1.61</td>
<td>16</td>
<td>4</td>
<td>18</td>
</tr>
</tbody>
</table>

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 \( L^\gamma \), and match the backing table size \( T^\gamma \) accordingly to avoid hash collisions. The base grid resolution \( B^\gamma \), grid per-level scale \( S^\gamma \), and feature encoding size \( F^\gamma \) are kept constant.

\( \gamma \)

is defined 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.

coding parameters \( \text{params}_{\gamma} \) can be tuned for specific tasks:

\[
h(u, v) = h(\gamma(u, v; \text{params}_{\gamma}); \theta) \quad \text{params}_{\gamma} = \{B^\gamma, S^\gamma, L^\gamma, F^\gamma, T^\gamma\}.
\]

(13)

By manipulating the parameters of Eq. 13 as defined in Tab. 1 we construct four different “sizes” of network encodings to investigate the method behavior: 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 size separation 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 simple-to-fit 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 [47] have per-scene optimal parameters. Given the relatively fast training speed of our approach, approximately 10mins on a single Nvidia RTX 4090 GPU to fit a scene, 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.

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.

Occlusion Removal Focusing on natural environmental occluders such as fences and grates, we include a set of additional occlusion removal results in Fig. 3. We evaluate our results against a multi-image learning-based obstruction removal method Liu et al. [42], a NeRF-based method OCC-NeRF [70], the flow plus homography neural image representation NIR [50], and a single image inpainting approach Lama [57] – 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 reconstructions – as is the case for the Digger 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 [57] can successfully inpaint sparse occluders such as the fences in the Digger and Pipes scenes, but struggle to recover content behind dense occluders such as Hallway and Desk 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 [50], Liu et al. [42], and the single-image reflection removal method DSRNet [25]. We show reflection removal results in Fig. 7. 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 to fail for high opacity reflections, such as the Grass example, and regions like the top left of the Test Page scene where the reflection can be mistaken for a real lighting effect. Our method achieves the highest-quality reconstruction and layer separation among all methods tested, across all scenes, with our estimated alpha matte revealing the detailed structure of the scene being reflected. In Fig. 8 we also showcase our model’s performance on challenging, in-the-wild scenes where we do not 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 Next, we discuss evaluations on synthetic data with known ground truth. As described in Sec. A, we compare our obstruction and reflection removal method with several baselines, including OCC-NeRF [70], Liu et al. [42], and NIR [50], in Fig. 5 and Fig. 6 respectively. We provide NeRF-based methods with ground truth camera poses, which results in a higher fidelity NeRF-based reconstruction than on real-world data. We also include quantitative metrics measuring the quality of the reconstructed transmitted layer. In our evaluations of obstruction removal, we observe that just as in the real-world examples, the multi-image based methods almost entirely fail to remove occlusions. Single-image methods, when provided with a ground-truth occlusion mask, are able to construct a coherent transmission layer. However, upon closer inspection the single-image results are missing details in the ground-truth transmission layer, such as the...
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.

Figure 4. Shadow removal results under different lighting conditions: (a) partially diffuse, (b) multiple point, (c) single point.

string of flags in Station, in Fig. 5. Our reconstructions have the highest PSNR and SSIM across all methods and all scenes tested, both removing the obstruction and accurately recovering details hidden behind it. We also observe that most multi-image methods fail to remove reflections in Fig. 6, with the exception of Liu et al. on the scene Vase Outdoor. The single-image method DSRNet [25] once again outperforms the multi-view baselines, removing the reflection in front of the rightmost vase in Vase Outdoor, and improving the visibility into the glass case in Barrels. However, similarly to the real-world scenes, DSRNet struggles to fully disambiguate more opaque reflections like the cabinet doors in Mixer. Our proposed approach is able the only method that removes the reflection in front of the leftmost vase in Vase Indoor, and the entire reflection in Mixer. This qualitative performance is reflected in the quantitative metrics, where we achieve the highest PSNR and SSIM for all tested scenes.

Shadow Removal In Fig. 4 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
Figure 5. Qualitative and quantitative occlusion removal results for a set of 3D rendered scenes with paired ground truth. Evaluation metrics formatted as PSNR/SSIM.

<table>
<thead>
<tr>
<th>Occlusion</th>
<th>OCC-NeRF</th>
<th>Liu et al.</th>
<th>NIR</th>
<th>Lama</th>
<th>Proposed</th>
<th>Bookstore</th>
<th>21.34/0.467</th>
<th>24.57/0.716</th>
<th>20.44/0.459</th>
<th>28.72/0.824</th>
<th>35.59/0.929</th>
<th>Window</th>
<th>14.22/0.608</th>
<th>14.03/0.718</th>
<th>13.98/0.680</th>
<th>30.44/0.939</th>
<th>32.91/0.953</th>
</tr>
</thead>
<tbody>
<tr>
<td>Couch</td>
<td>13.37/0.658</td>
<td>13.09/0.708</td>
<td>13.09/0.690</td>
<td>26.86/0.927</td>
<td>28.72/0.824</td>
<td>Station</td>
<td>14.62/0.569</td>
<td>14.40/0.649</td>
<td>14.05/0.595</td>
<td>26.48/0.874</td>
<td>30.25/0.936</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. Qualitative and quantitative reflection removal results for a set of 3D rendered scenes with paired ground truth. Evaluation metrics formatted as PSNR/SSIM.

<table>
<thead>
<tr>
<th>Reflection</th>
<th>NeRFReN</th>
<th>Liu et al.</th>
<th>NIR</th>
<th>DSR-Net</th>
<th>Proposed</th>
<th>Vase</th>
<th>18.40/0.688</th>
<th>18.52/0.849</th>
<th>18.52/0.771</th>
<th>19.43/0.835</th>
<th>27.20/0.942</th>
<th>Mixer</th>
<th>20.59/0.736</th>
<th>20.76/0.850</th>
<th>20.57/0.752</th>
<th>20.39/0.811</th>
<th>26.07/0.921</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fish</td>
<td>20.05/0.744</td>
<td>24.32/0.904</td>
<td>19.78/0.783</td>
<td>23.51/0.873</td>
<td>25.47/0.824</td>
<td>Barrels</td>
<td>19.57/0.628</td>
<td>19.20/0.726</td>
<td>19.34/0.660</td>
<td>24.56/0.794</td>
<td>27.06/0.875</td>
<td>Barrels</td>
<td>19.57/0.628</td>
<td>19.20/0.726</td>
<td>19.34/0.660</td>
<td>24.56/0.794</td>
<td>27.06/0.875</td>
</tr>
</tbody>
</table>
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, 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 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 content 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 from in from – 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 $R_\alpha$ 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 [57] 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:

$$L_\alpha = |(\Delta c - \Delta \hat{c})/(sg(\Delta c) + \epsilon)|^2.$$  

Rather than sample a grid of points around $u^0, v^0$ and $u^1, v^1$ or perform a second pass over the image networks [50] to compute Jacobians, we compute color gradients $\Delta c$ by pair-

![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.](image)

$$\Delta c = I(u, v, t) - I(\tilde{u}, \tilde{v}, t)$$  

$$\tilde{u}, \tilde{v} = u + r\cos(\phi), \ v + r\sin(\phi), \ \phi \sim U(0, 2\pi),$$  

\[14\]
where \( r \) determines the magnitude of the perturbation. The estimated color gradient \( \hat{\Delta c} \) is similarly calculated for the output colors of our model. Illustrated in Fig. 13, 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 \( L_G \) without significantly impacting training time – as perturbed samples are also re-used for regular photometric loss calculation \( L_p \). 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. 13. 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 \( \eta_\alpha \) on reconstruction. The primary function of this regularization is remove
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. 10 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 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. 11 we illustrate the effects on obstruction removal of overparameterizing 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

Figure 12. Ablation study on the effects of alpha regularization weight \( \eta_\alpha \) on transmission layer reconstruction and estimated alpha matte.

Figure 13. Visualization of the effects of gradient loss \( L_c \) on image reconstruction at 25x zoom. Inset bottom left is the radius of perturbation at epoch 40 and epoch 100, the end of training.
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 images 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.