Neural Spline Fields for Burst Image Fusion and Layer Separation (Supplementary Material)

Ilya Chugunov David Shustin Ruyu Yan Chenyang Lei Felix Heide

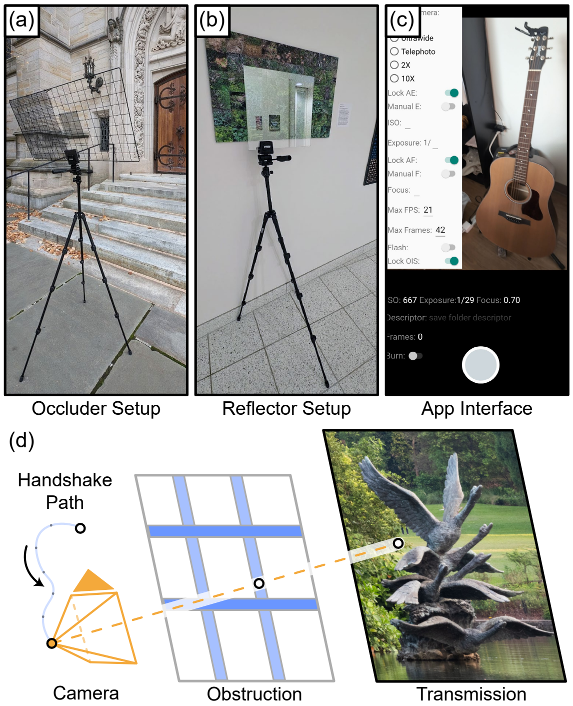
Princeton University

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



mera:
○ Ultrawide
○ Telephoto
○ 2X
○ 10X
Lock AE: ●
Manual E: ○
ISO: _
Exposure: 1/ _
Lock AF: ●
Manual F: ○
Focus: _
Max FPS: 21
Max Frames: 42
Flash: ○
Lock OIS: ●
SO: 667 Exposure:1/29 Focus: 0.70
Descriptor: save folder descriptor
Frames: 0
Burn: ○

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.

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 [3] 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 ≈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 [10], 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 [3]. 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:

$$
\begin{aligned}
h(u, v) &= \mathbf{h}(\gamma(u, v; \mathrm{params}_\gamma); \theta) \\
\mathrm{params}_\gamma &= \{\mathrm{B}^\gamma, \mathrm{S}^\gamma, \mathrm{L}^\gamma, \mathrm{F}^\gamma, \mathrm{T}^\gamma\}.
\end{aligned}
\tag{1}
$$

By manipulating the parameters of Eq. 1 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.

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 $S(t, h(u, v))$.

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 [7] have per-scene optimal parameters. Given

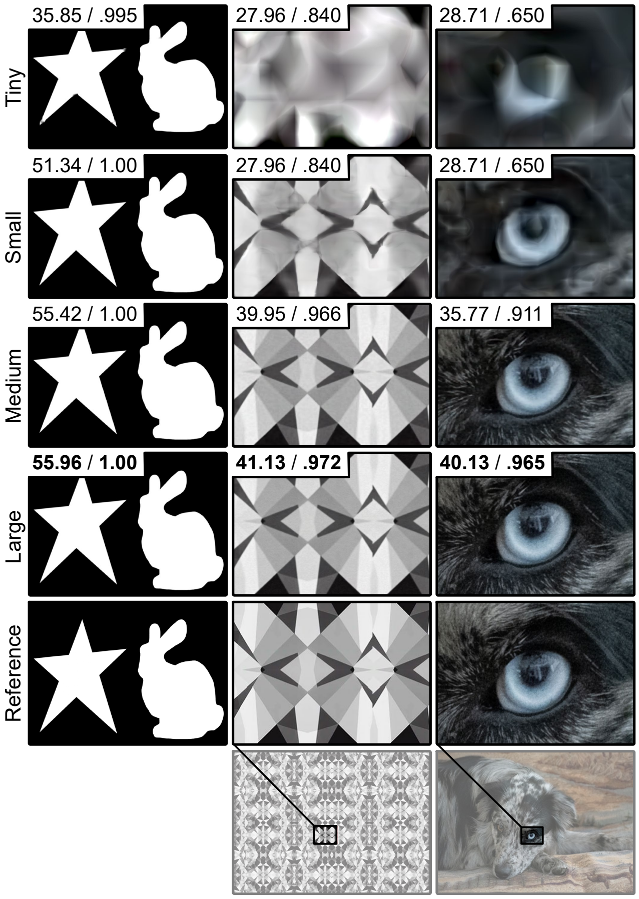


Figure 2. Image fitting results for network encoding configurations as described in Tab. 1, other training and network parameters held constant: 5-layer MLP coordinate networks, hidden dimension 64, ReLU activations. PSNR/SSIM values inset top-left.

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.

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 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. [6], the NeRF-based method OCC-NeRF [11], the flow plus homography neural image representation NIR [8], and the single image inpainting approach Lama [9] – 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 *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 [9] 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 [4] in addition to NIR [8], Liu et al. [6], and the single-image reflection removal method DSRNet [5]. 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

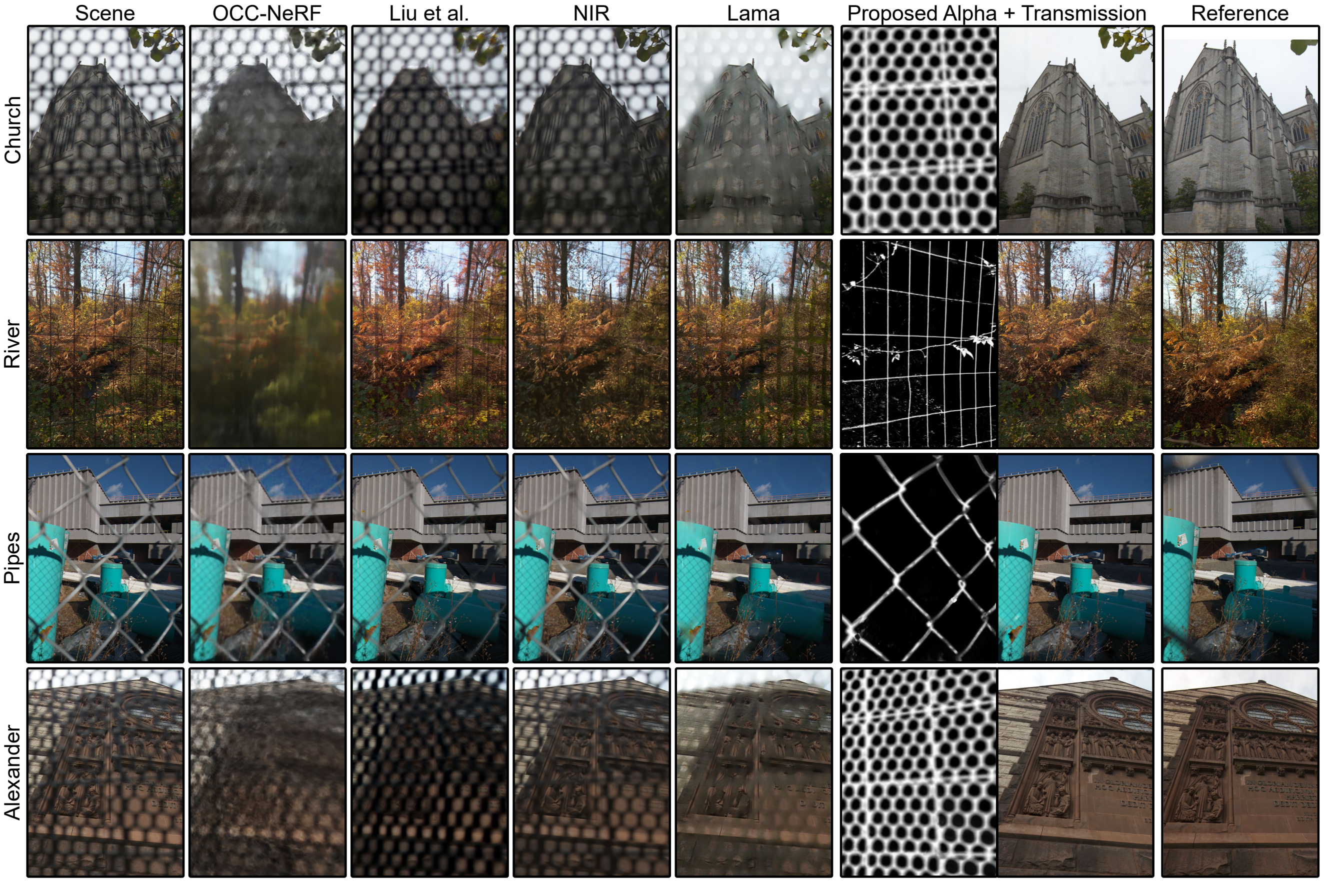


Figure 3. Occlusion removal results and estimated alpha maps for a set of captures with reference views, with comparisons to single image, multi-view, and NeRF fitting approaches. See video materials for visualization of input data and scene fitting.

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 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 [11], NeRFReN [4], Liu et al. [6], NIR [8], Lama [9] and DSRNet [5]. 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. [6] 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 [9], 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 [5] fail to effectively remove reflections in Fig. 8, with DSRNet [5] 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:

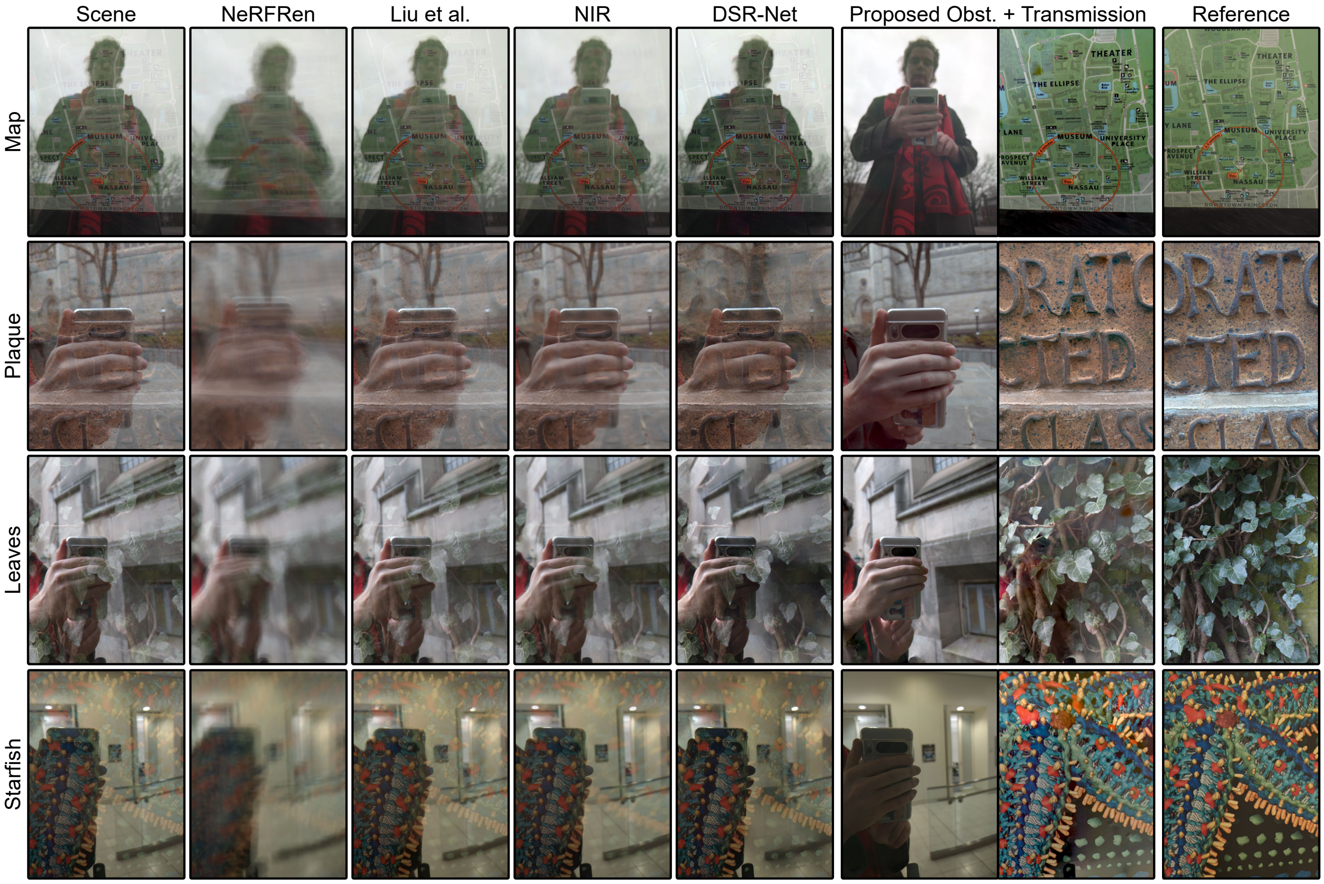


Figure 4. Reflection removal results and estimated alpha maps for a set of captures with reference views, with comparisons to single image, multi-view, and NeRF fitting approaches. See video materials for visualization of input data and scene fitting.

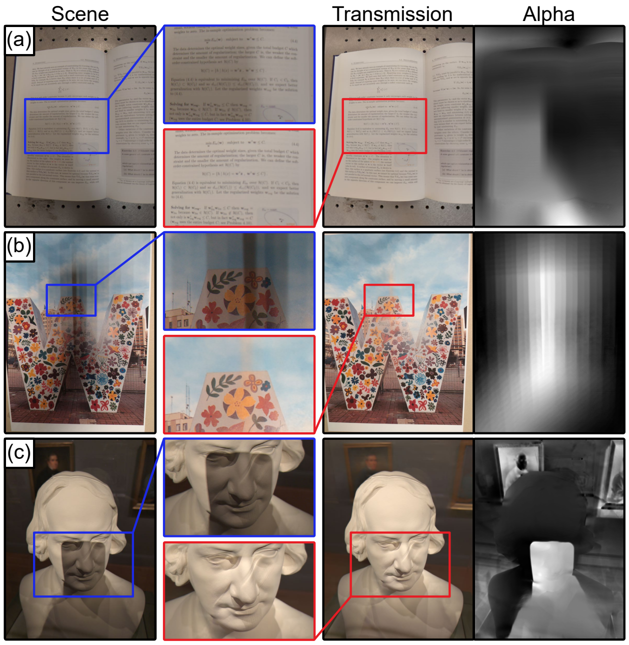


Figure 5. Shadow removal results under different lighting conditions: (a) partially diffuse, (b) multiple point, (c) single point.

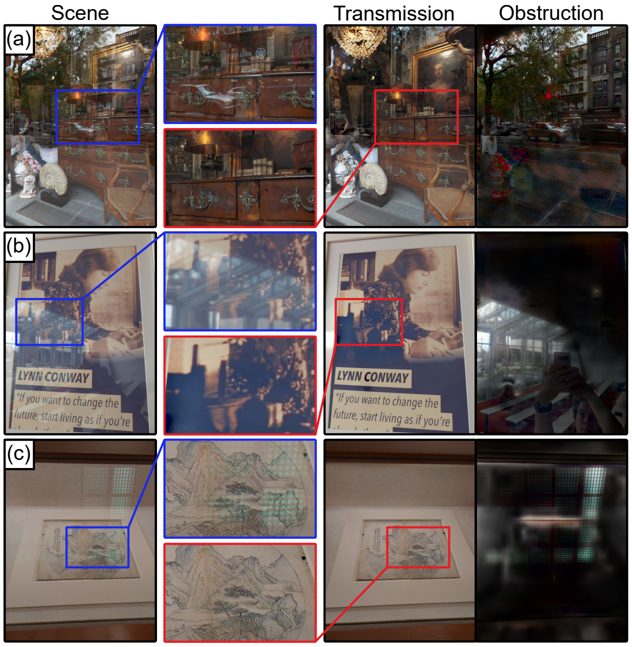
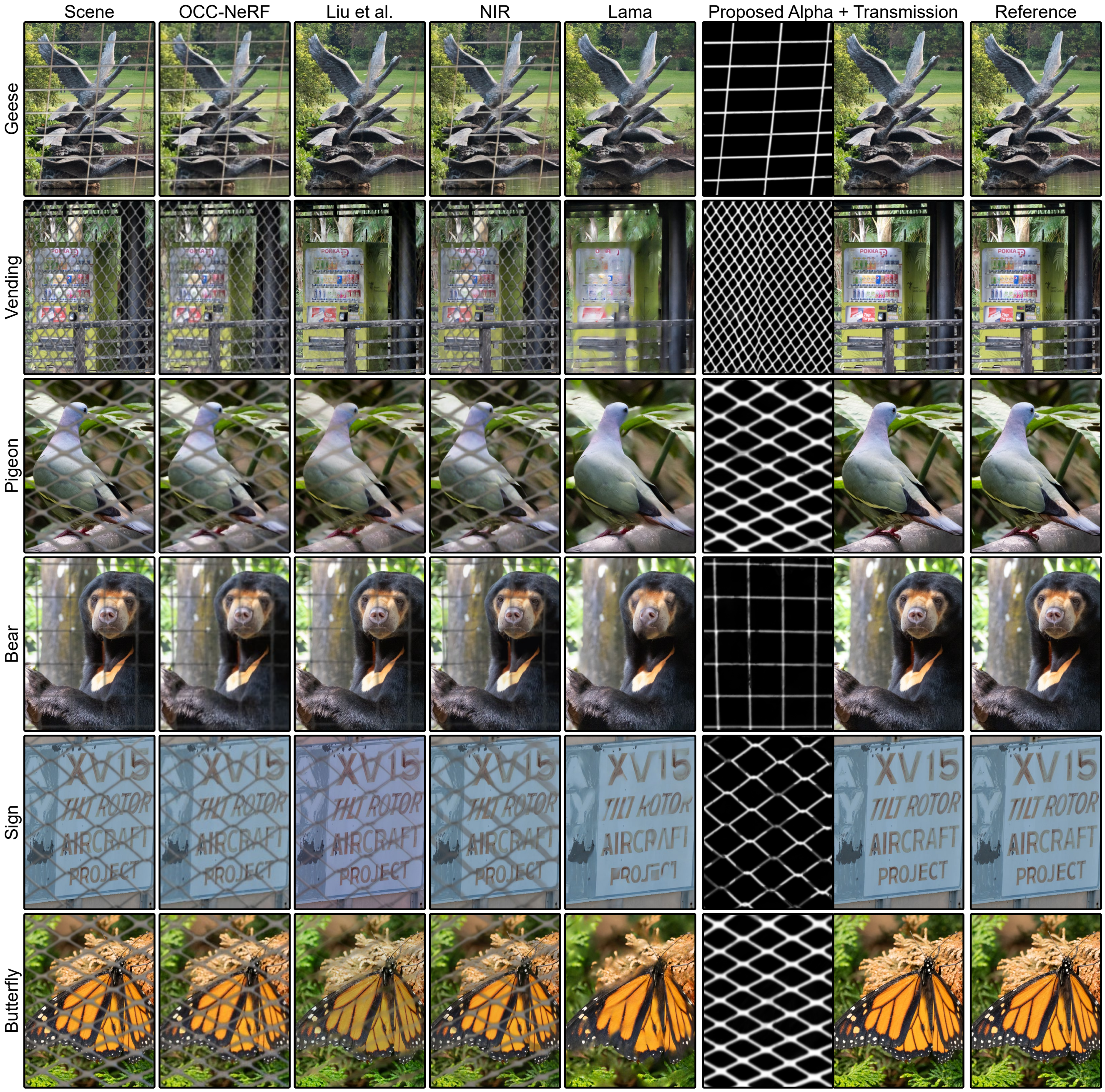
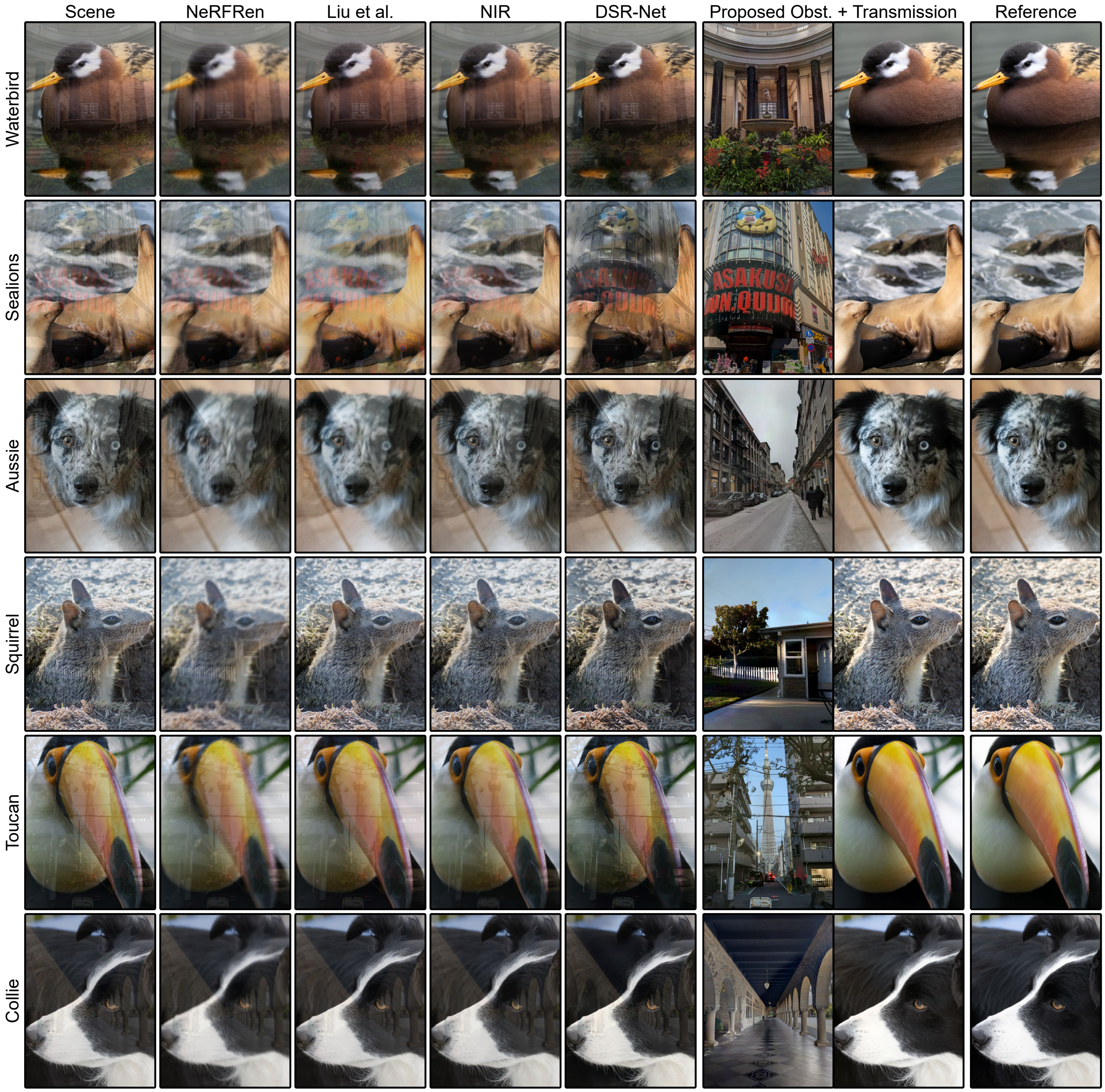


Figure 6. Reflection removal results for challenging in-the-wild scenes: (a) storefront window, (b) poster, (c) museum painting.



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.

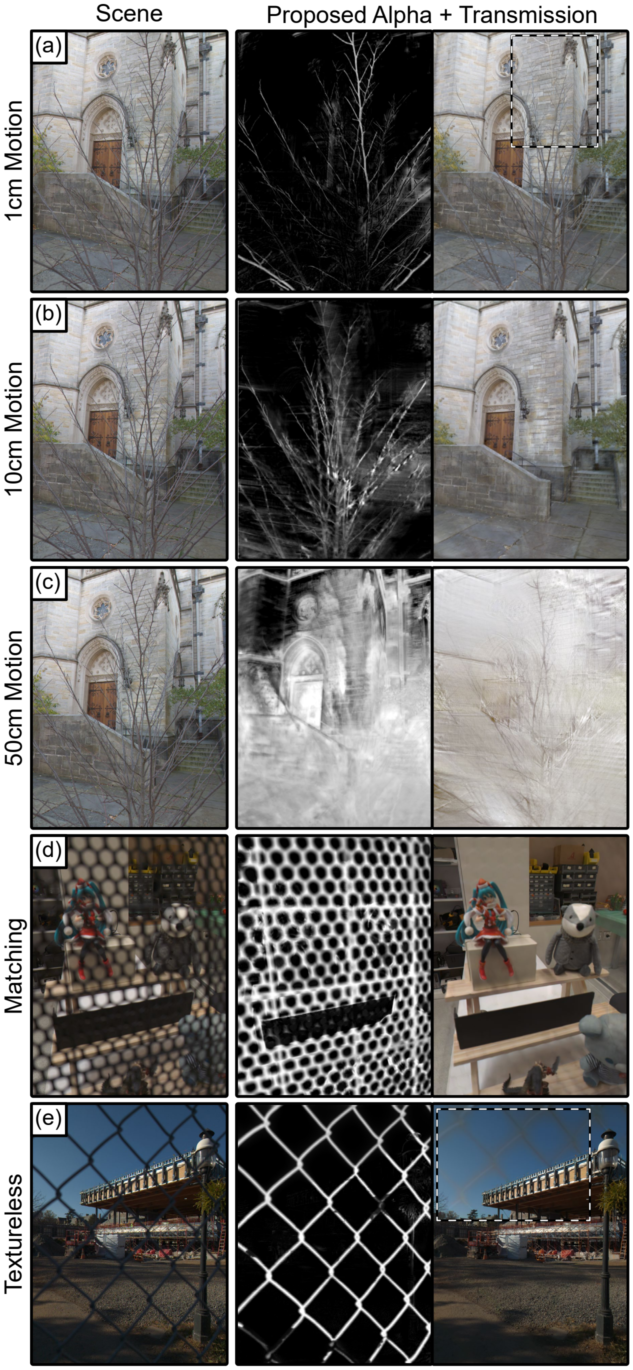


Figure 9. Challenging image reconstruction cases including varying scales of camera motion, overlap between occluder and transmission colors, and residual signal left on scene content in low-texture regions. Areas of interest highlighted with dashed border.

(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, 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.

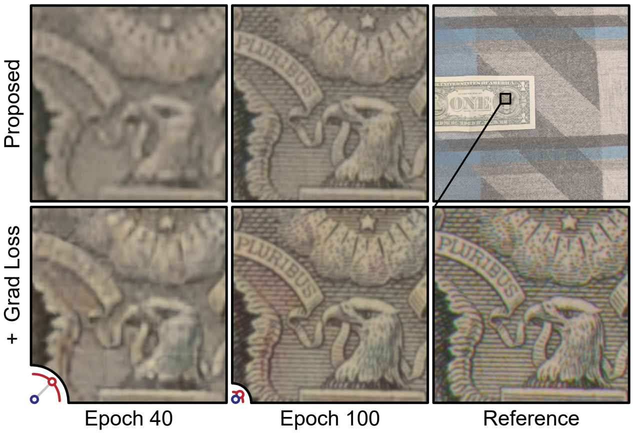


Figure 10. Visualization of the effects of gradient loss $\mathcal{L}_{\text{G}}$ on image reconstruction at 25x zoom. Inset bottom left is the radius of perturbation at epoch 40 and epoch 100, the end of training.

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 [9] 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 [1]. 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 [8] to compute Jacobians, we compute color gradients Δc by pairing each ray with an input perturbed in a random direction

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

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 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 [2], 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

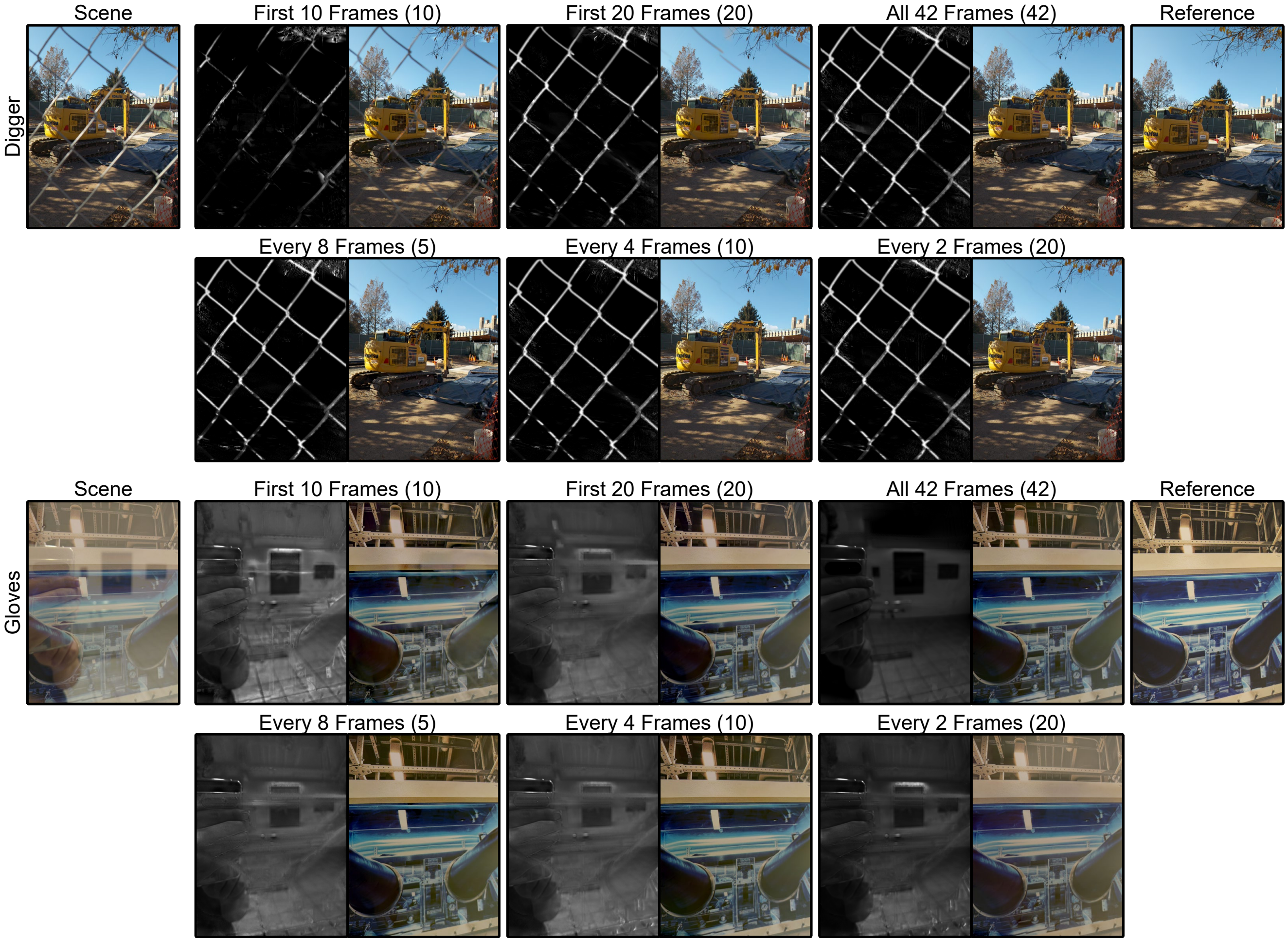


Figure 11. Ablation study on the effects of the number of input frames or duration of capture on transmission layer reconstruction and estimated alpha matte. Total number of frames input into the model denoted by the number in parentheses– e.g., (10) = ten frames.

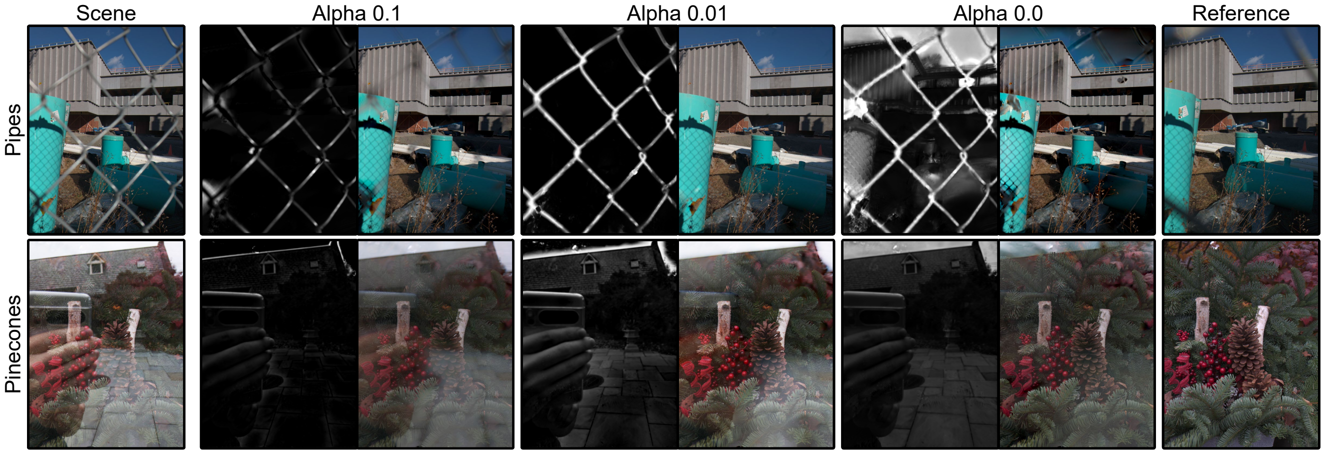


Figure 12. Ablation study on the effects of alpha regularization weight η_α on transmission layer reconstruction and estimated alpha matte.

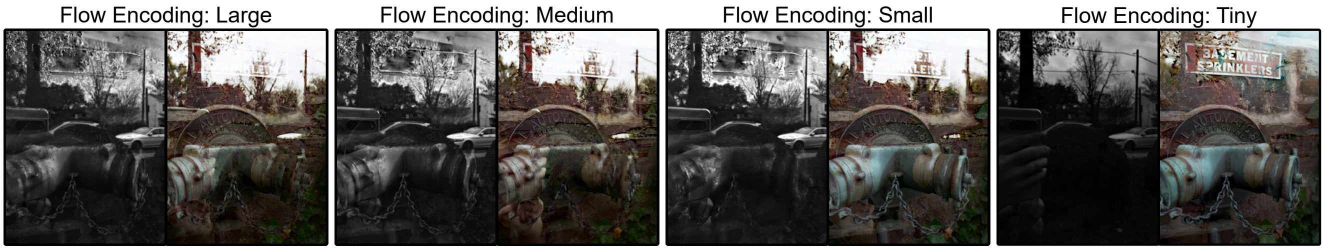


Figure 13. Ablation study on the effects of flow encoding size (Tab. 1) on transmission layer reconstruction and estimated alpha matte.

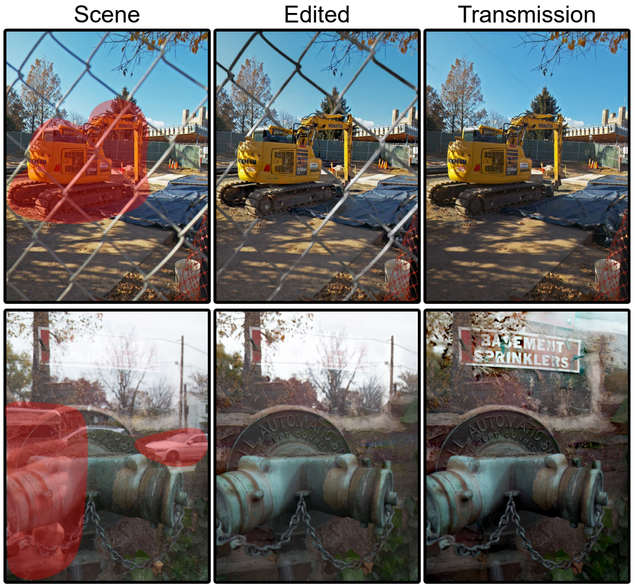


Figure 14. Demonstration of user-interactive scene editing facilitated by layer separation. Only the user-selected region of the obstruction, highlighted in red, is removed without affecting surrounding scene content, see text.

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 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.

References

[1] 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. 9

[2] 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. 9

[3] 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] 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. 3, 4

[5] 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. 3, 4

[6] 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. 3, 4

[7] 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. 2

[8] Seonghyeon Nam, Marcus A. Brubaker, and Michael S. Brown. Neural image representations for multi-image fusion and layer separation, 2022. 3, 4, 9

[9] 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. 3, 4, 9

[10] 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. 1

[11] Chengxuan Zhu, Renjie Wan, Yunkai Tang, and Boxin Shi. Occlusion-free scene recovery via neural radiance fields. 2023. 3, 4