Self-Supervised Video Defocus Deblurring with Atlas Learning

Lingyan Ruan
lyruan@mpi-inf.mpg.de
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Martin Bálint
mbalint@mpi-inf.mpg.de
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Mojtaba Bemana
mbemana@mpi-inf.mpg.de
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Krzysztof Wolski
kwolski@mpi-inf.mpg.de
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Hans-Peter Seidel
hpseidel@mpi-inf.mpg.de
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Karol Myszkowski
karol@mpi-inf.mpg.de
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Bin Chen*
binchen@mpi-inf.mpg.de
Max-Planck-Institut für Informatik
Saarbrücken, Germany
University of Melbourne
Melbourne, Australia

Figure 1: Systematic video defocus deblurring and editing. Taking a misfocused video as input (first row), we first reconstruct the latent sharp video (second row), parameterizing it into two atlases reconstructed from noise. We edit our background atlas in Adobe Photoshop, changing some objects (blue arrows) in the scene. These simple edits propagate to the entire video in a consistent and temporally stable manner. Finally, during compositing, we fix subject tracking and change the background bokeh. These edits are also consistent and temporally stable. We slice a piece of video data (yellow dash line) on the time dimension and present it in the last column, demonstrating motion quality. The only costly step of our method is learning the initial parametrization; edits can be made in real time.

*Corresponding author

SIGGRAPH Conference Papers '24, July 27–August 01, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07
https://doi.org/10.1145/3641519.3657524

ABSTRACT

Misfocus is ubiquitous for almost all video producers, degrading video quality and often causing expensive delays and reshoots. Current autofocus (AF) systems are vulnerable to sudden disturbances such as subject movement or lighting changes commonly present in real-world and on-set conditions. Single image defocus deblurring methods are temporally unstable when applied to videos and cannot recover details obscured by temporally varying defocus blur. In this paper, we present an end-to-end solution that allows users to correct misfocus during post-processing. Our method generates

and parameterizes defocused videos into sharp layered neural atlases and propagates consistent focus tracking back to the video frames. We introduce a novel differentiable disk blur layer for more accurate point spread function (PSF) simulation, coupled with a circle of confusion (COC) map estimation module with knowledge transferred from the current single image defocus deblurring (SIDD) networks. Our pipeline offers consistent, sharp video reconstruction and effective subject-focus correction and tracking directly on the generated atlases. Furthermore, by adopting our approach, we achieve comparable results to the state-of-the-art optical flow estimation approach from defocus videos.

CCS CONCEPTS

• **Computing methodologies** → **Computational photography**.

KEYWORDS

defocus deblur, implicit representation, neural atlas, video deblur, post refocus

ACM Reference Format:

Lingyan Ruan, Martin Bálint, Mojtaba Bemana, Krzysztof Wolski, Hans-Peter Seidel, Karol Myszkowski, and Bin Chen. 2024. Self-Supervised Video Defocus Deblurring with Atlas Learning. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27–August 01, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641519.3657524

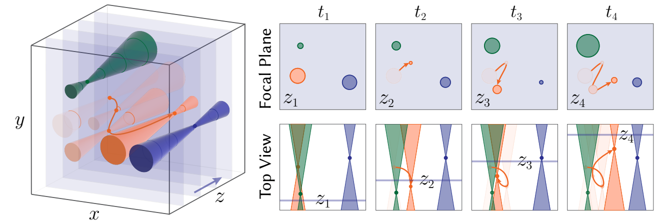

Figure 2: Spatially and temporally varying defocus blur. The focus distance moves from front to back ($z_1 \rightarrow z_4$) as the video capturing progresses ($t_1 \rightarrow t_4$). We represent the focal plane as a light blue rectangle for each frame. The green and blue points are static and placed at different spatial locations and depths, while the orange point moves in 3D space for each frame following the orange curve.

1 INTRODUCTION

Shallow focus initially emerged due to early filmmaking constraints such as limited ISO sensitivity. However, directors soon leveraged its storytelling potential to direct the audience's attention. Thus, shallow focus quickly became essential for achieving a cinematic look. Today, professional focus pullers meticulously plan for shoots, particularly long takes or "oners", involving extensive camera movement and complex blocking. Maintaining focus is crucial for a successful shot.

Conversely, amateur content creators rely on camera autofocus (AF). Modern AF systems have advanced from traditional Contrast Detection (CDAF) [Chen and van Beek 2015; Vuong and Lee 2013] and Phase Detection (PDAF) [Fontaine 2017; Inoue and Takahashi 2009] to smarter systems with face, eye, and object tracking capabilities [Wang et al. 2021]. Unfortunately, in real-world scenarios, abrupt changes in subject movement, lighting conditions, or the presence of multiple subjects can easily disrupt AF systems, resulting in misfocused footage.

Technical deficiencies and human mistakes frequently force video producers to either reshoot their misfocused footage or settle for lower-quality videos, ignoring the focus errors. Restoration of misfocused footage in post-production is emerging as a promising alternative. Our paper introduces a pipeline designed to reconstruct sharp content from videos distorted by defocus, enable focus editing to fix subject tracking, allow for convenient scene editing via our learned atlas, and output additional channels that enable complex effects during compositing.

In Fig. 2, we illustrate the complexity of our task by simulating a video featuring three spatial points: a stationary green point in the foreground, a stationary blue point in the background, and a dynamic orange point moving freely in 3D space. Each stationary point has a unique defocus level determined by its depth in the scene. As the focus distance shifts from front to back (light blue rectangles) over time, the defocus degree for each stationary point also changes. In real-world video capture scenarios, camera movement adds another layer of ambiguity. These complexities pose significant challenges to video defocus deblurring and influence other video processing tasks such as object tracking or optical flow estimation [Ruan et al. 2021; Teed and Deng 2020].

Correcting misfocus is a two-step procedure that includes eliminating defocus blur, also known as defocus deblurring, followed by defocus synthesis, which applies defocus blur following the scene's depth. Defocus synthesis is a forward problem and has been investigated for images [Srinivasan et al. 2018; Wadhwa et al. 2018] and videos [Zhang et al. 2019] in the past. Defocus deblurring, on the other hand, is an inverse problem and poses different challenges than defocus synthesis. Misfocus correction in video, unlike single image defocus deblurring (SIDD), requires an emphasis on maintaining temporal consistency. The key difficulty here is the temporally varying defocus blur caused by changes due to scene and camera motion and, potentially, autofocus camera functionality.

Inspired by recent implicit video representation approaches [Kasten et al. 2021], we apply the layered atlas to parameterize and deblur our video with good consistency and temporal stability. Recently, Abuolaim et al. [2021] synthesized a dataset based on the sharp images and depth maps in the SYNTHIA dataset [Hernandez-Juarez et al. 2019]. However, when applied to real-world videos, methods trained on synthetic datasets often suffer due to the domain gap. Our self-supervised network circumvents the need for such extensive datasets and performs deblurring directly on each individual video.

In this work, we make the following contributions: (1) We introduce the first end-to-end video defocus deblurring approach, allowing for extensive scene editing, including focus corrections and bokeh editing. (2) We implement a lens blur CUDA layer featuring a novel differentiable disk kernel capable of simulating the realistic fall-off boundary (soap bubble effect) of the Point Spread Function (PSF). (3) We introduce a COC map estimation network using a transfer learning approach. This strategy facilitates a more lightweight architecture while maintaining competitive performance.

(4) As a direct application of our pipeline, we show that focus correction and tracking can be conducted on our output foreground and background channels, our learned atlases allow for extensive editing, and we demonstrate that optical flow can also be estimated from our learned UV map.

Our code and the training data we rendered with synthetic defocus are available under https://neuralatlasvdd.mpi-inf.mpg.de.

2 RELATED WORK

In this section, we cover two main tracks of work that are most related to our approach: the image/video defocus deblurring and implicit image/video representations.

2.1 Image/Video Defocus Deblurring

Dealing with defocus blur has been a persistent challenge in image processing and computer vision, largely because of its inherent spatially varying nature in static and dynamic scenes compared to motion blur. Recovering the sharp latent details and information from blurred images or videos holds significant potential for various applications, including object detection [Redmon and Farhadi 2018] and text recognition [Liao et al. 2021]. The majority of prior research has concentrated on SIDD.

Single Image Defocus Deblurring. Traditional solutions typically follow a two-step process involving estimating a dense defocus map followed by non-blind deconvolution [D'Andrès et al. 2016; Karaali and Jung 2017; Park et al. 2017; Shi et al. 2015]. In this process, the quality and precision of the defocus map are critical factors significantly impacting the final outcome. Techniques utilizing deep neural networks to estimate defocus maps have been proposed [Karaali et al. 2022; Lee et al. 2019]. Recent works such as [Zhang and Sun 2021] and [Piché-Meunier et al. 2023] have advanced the estimation process by jointly deriving depth and defocus while adhering to consistency constraint, where deriving the lens parameters leads to further improvements [Piché-Meunier et al. 2023].

Our work aligns with their goals in accurately determining the circle of confusion (COC) [Potmesil and Chakravarty 1982] for realistic defocus synthesis. However, we diverge from these methods by omitting scene physical depth estimation and estimating defocus level in pixel units, which better suits our pipeline.

The advent of deep learning has significantly advanced SIDD, leading to the emergence of neural network-based solutions. Abuolaim et al. [2020] introduce a dual-pixel defocus deblurring dataset and an end-to-end network to recover a sharp image from its defocused counterpart. Ruan et al. [2021] introduce another dataset synthesized from light field images to address the defocus image and sharp ground truth mismatch problem. Other approaches have integrated spatially varying blur into network structures with per-pixel dynamic residual kernels [Ruan et al. 2022], iterative filter adaptive kernels [Lee et al. 2021], learnable recursive kernels [Quan et al. 2023], Gaussian kernel mixture [Quan et al. 2021] in an end-to-end manner, more recent studies, e.g., [Zamir et al. 2022] have incorporated transformer structures. Unfortunately, most of these methods present as a black box providing no straightforward ways to obtain COC maps, use temporal data available in other frames of the videos, or map to a temporally consistent space such as our multi-layer atlas.

Video Defocus deblurring. Unlike single image defocus deblurring or synthesis, blurry videos present additional challenges, particularly due to temporal variations discussed in Sec. 1. This area remains less explored but also holds significant practical appeal. RDPD [Abuolaim et al. 2021] proposes an RNN-based network structure to handle defocus deblurring for image sequences. Neucam [Huang et al. 2023] proposes an implicit camera model to simulate the image signal processing (ISP) process in a deep neural network and can recover all-in-focus images from multi-focus stacks. We compare our approach with theirs. Although no video defocus deblurring is demonstrated in the original Neucam paper, we adapt their video motion deblurring network to video defocus deblurring for a fair comparison. It is worth noting that video defocus deblurring is different from the problem of sharp image reconstruction from multi-focus, which is based on the assumption that the camera and scene are static. Several methods address videos involving dynamic scenes with static cameras using custom-designed camera systems [Zhou et al. 2012], specialized optical systems like deformable lenses [Miau et al. 2013], or rely on known focus distances [Kim et al. 2016]. We relax these constraints in our approach.

2.2 Implicit image/video representation

Emerging works integrate scene representation as neural radiance fields (NeRF) with multiple layer perceptrons (MLPs) in a continuous manner for 3D geometry [Mildenhall et al. 2021]. NeRF reconstruction quality degrades when input images contain defocus blur (shallow depth of field) [Wu et al. 2022]. To address this, Wu et al. [2022] incorporate explicit aperture modeling, coupled with a differentiable COC representation, while Li et al. [2022] instead incorporate a learned deformable kernel into the pipeline to compensate for defocus blur. A recent work [Wang et al. 2023] investigates reconstructing the all-in-focus frame from the image stack by simultaneously estimating depth constrained by an explicit physical camera model. Unfortunately, all these methods are limited to static scenes.

Motivated by recent success in dynamic video representation, such as view synthesis [Li et al. 2023], volumetric video rendering [Peng et al. 2023], video decomposition [Kasten et al. 2021], integrating MLPs to overfit to video content, we aim to fit a unique sharp content that could represent the whole sequence, thus achieving the spatially and temporally consistency for misfocused video. Specifically, we adopt a 2D atlas representation, which parameterizes the dynamic and static background separately with MLPs, originally used for video consistent editing, allowing alterations on the atlas to be reflected in the original video. This enables several applications such as text-driven video stylization [Bar-Tal et al. 2022], sketch face editing [Liu et al. 2022], video de-flickering [Lei et al. 2023], and others.

3 METHOD

Aiming for consistent video editing, layered neural atlases decompose an input video into a collection of 2D layered atlases [Kasten et al. 2021]. This representation effectively circumvents the need to

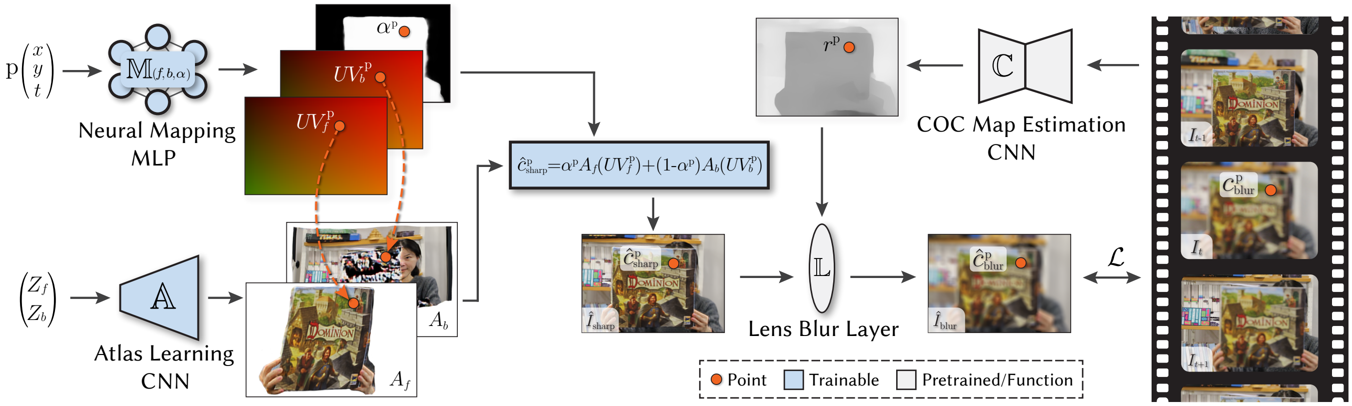

Figure 3: System Overview. Our mapping network learns foreground UV, background UV, and alpha maps from input coordinates. Simultaneously, the atlas network generates sharp foreground and background atlases from noise maps. We sample the background and foreground from the atlases following the UV maps and blend the resulting layers guided by the alpha map. Subsequently, we reblur the combined image using an estimated blur scale from our COC map estimation network. We self-supervise this process using the original video frames, estimating them with the reblurred output, learning the sharp prior in the process.

physically resolve the ambiguity between camera and object movement in our problem (Fig. 2) by efficiently mapping the motion of dynamic points onto a 2D plane. The multi-layer representation can be seamlessly adapted to our specific challenge of distinguishing between static and dynamic points, allowing for separate treatment.

Also, the layered neural atlases enable straightforward and intuitive editing using readily available image editing tools, which can be naturally propagated back to the video frames consistently. The distinct foreground, background, and mask layers enable us to conduct focus-tracking in post-production and are easy and quick to compose into video frames.

The efficient representation of video, intuitive editing space, and consistent editing quality make layered neural atlases excel in sharp video editing. However, it does not demonstrate strong performance when applied directly to video defocus deblurring tasks. We demonstrate and compare the performance of the original layered neural atlas and the combination of it and our reblurring module in Sec. 4.2.

Figure 3 provides an overview of our training pipeline, which consists of four main submodules: (1) The *Neural Mapping Network* employs three MLPs $\mathbb{M}_{(f,b,\alpha)}$ to transform input coordinates into foreground UV map, background UV map, and alpha map, respectively (Sec. 3.1). (2) The *Atlas Learning Network* ($\mathbb{A}$) adopts the deep image prior concept and utilizes a CNN network to learn foreground and background atlases from noise (Sec. 3.2). (3) The *Lens Blur Layer* ($\mathbb{L}$) reblurs the latent sharp image based on the estimated COC map (Sec. 3.3). (4) The *COC Map Estimation Network* ($\mathbb{C}$) estimates the COC radius for each pixel from an input defocus image (Sec. 3.4). We provide detailed explanations for each of these submodules in the following subsections.

3.1 Neural Mapping

Following the principles of coordinate-based neural representations, layered neural atlases commence the synthesis process at coordinates $\mathbf{p}_{x,y,t} \in \mathbb{R}^3$, with x and y representing spatial coordinates and t indicating the temporal coordinate (frame index). Three MLPs, referred to as $\mathbb{M}_b : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, $\mathbb{M}_f : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, and $\mathbb{M}_\alpha : \mathbb{R}^3 \rightarrow \mathbb{R}^1$, predict the background UV map $UV_b \in \mathbb{R}^2$, foreground UV map $UV_f \in \mathbb{R}^2$, and the alpha blending map $\alpha \in \mathbb{R}^1$, respectively. Each point $\mathbf{p}$ maps to $UV_f^{\mathbf{p}} = \mathbb{M}_f(\mathbf{p}) = (u_f^{\mathbf{p}}, v_f^{\mathbf{p}})$, $UV_b^{\mathbf{p}} = \mathbb{M}_b(\mathbf{p}) = (u_b^{\mathbf{p}}, v_b^{\mathbf{p}})$, and $\alpha^{\mathbf{p}}$. We share the same architecture with the original layered neural atlases with regard to this part. However, we find that directly applying their atlas estimation network results in inferior performance. Inspired by Deep Image Prior (DIP) [Lempitsky et al. 2018] and [Ye et al. 2022], we employ a CNN network for our *Atlas Learning Network*.

3.2 Atlas Learning

Deep Image Prior (DIP) uses the structure of a deep convolutional network to implicitly capture prior knowledge about natural image statistics and has proven to be particularly suitable for image restoration tasks [Lempitsky et al. 2018]. In general, a simple MLP structure tends to perform less effectively in capturing spatial features such as textures and edges when compared to CNN structures without specialized design [Tolstikhin et al. 2021; Tu et al. 2022]. Thus, we employ a lightweight CNN to generate and represent our layered atlas images from noise maps. We have observed that a decoder-only architecture efficiently learns our atlases. We applied pixel-shuffle [Shi et al. 2016] in the upsampling layers to better learn and preserve high-frequency details. The results indicate improved performance compared to the original MLP-based atlas representation, as elaborated in Sec. 4.2.

The *Atlas Learning Network*, denoted as $\mathbb{A}$, takes as input the noise map $(Z_f, Z_b) \in \mathbb{R}^{C' \times H' \times W'}$ and upscales it to produce our foreground and background atlas images $(A_f, A_b) \in \mathbb{R}^{3 \times H \times W}$. Note that the *Atlas Learning Network* shares weights between foreground and background atlas generation. Given a particular point

$\mathbf{p}$, we employ the acquired $UV_f^{\mathbf{p}}$ and $UV_b^{\mathbf{p}}$ coordinates to sample colors from A_f and A_b. Finally, we obtain the latent sharp color $\hat{c}^{\mathbf{p}}_{\text{sharp}}$ by combining these two colors, using the learned α value as the blending factor. This entire procedure can be summarized as follows:

$$\hat{c}^{\mathbf{p}}_{\text{sharp}} = \alpha A_f(\mathbb{M}_f(\mathbf{p})) + (1-\alpha)A_b(\mathbb{M}_b(\mathbf{p})) \tag{1}$$

where $A_f = \mathbb{A}(Z_f)$ and $A_b = \mathbb{A}(Z_b)$. Note that we utilize one foreground and one background layer to present the video content in this paper. However, additional layers can readily be implemented to accommodate more complex motion and scenes as indicated in [Kasten et al. 2021]. To encourage the network generating sharp images, we incorporate a reblur module to reproduce the blurred images, allowing the network to be optimized using the original input video frames as self-supervision.

3.3 Lens Blur Layer

To enable reblurring of the latent sharp image, we have developed a differentiable disk kernel CUDA layer that supports the continuous radius of PSF in our reblur module. While the disk kernel is considered a more accurate representation of realistic lens blur compared to the Gaussian kernel [Potmesil and Chakravarty 1982], it presents a challenge in terms of differentiability. Traditional disk kernels are typically represented in discrete sizes, making them unsuitable for accurately representing a continuous PSF radius. Existing methods for achieving differentiability with the disk kernel typically rely on functions like the hyperbolic tangent (tanh) as demonstrated in [Busam et al. 2019] and [Luo et al. 2023], or piecewise functions as discussed in [Gwosdek et al. 2011]. These approaches primarily focus on controlling the size of the PSF and result in uniform PSFs. However, the actual shape of the PSF is non-uniform, with sharply defined fall-off edges due to optical aberrations, as highlighted in [Tang and Kutulakos 2012]. Abuolaim et al. [2021] attempted to create the fall-off boundary of the PSF by combining a Butterworth filter with an undifferentiable disk kernel, resulting in a final kernel that remains undifferentiable. In this paper, we propose a straightforward and efficient solution—a differentiable disk kernel that inherently captures the sharply defined fall-off boundary without the need for additional modifications or complex functions. The value k inside our differentiable disk kernel can be calculated as:

$$k = \frac{e^{\beta d}}{e^{\beta d} - d}, \quad \text{with } d = r - \sqrt{x^2 + y^2}, \tag{2}$$

where $d \in \mathbb{R}$ represents the distance to the edge of disk, $r \in \mathbb{R}^+$ the radius of disk, and $x, y \in [-\lfloor \frac{s}{2} \rfloor - 1, \lfloor \frac{s}{2} \rfloor + 1]$ with $s \in \mathbb{N}$ representing the kernel size. We use $\beta \in \mathbb{R}^+$ to control the thickness of the fall-off boundary. Note that the value of β should be greater than $\frac{1}{e}$ to ensure a positive kernel. We used $\beta = 2$ in our lens blur CUDA layer. We applied per-pixel scatter-sum convolution similar to [Gur and Wolf 2019]. With our lens blur layer, the sharp image can be reblurred as:

$$\hat{c}^{\mathbf{p}}_{\text{blur}} = \mathbb{L}(\hat{c}^{\mathbf{p}}_{\text{sharp}}, r^{\mathbf{p}}) \tag{3}$$

For simplicity and efficiency, we construct the estimated sharp image using two layers. Optionally, multiple layers can be used, compositing from back to front, to better handle occluded areas around the subject's silhouette, as demonstrated in [Zhang et al., 2019]. As depicted in Fig. 4, we conduct a comparative analysis involving our differentiable disk kernel, the Gaussian kernel, an undifferentiable disk kernel, and methods outlined in [Gwosdek et al. 2011] and [Busam et al. 2019]. Our disk kernel presents a realistic fall-off boundary and approximates the real PSF more closely. To demonstrate our proposed disk kernel's advantages, we reblur a sharp image (GT sharp) and compare the outcomes with the other kernels. As expected, the Gaussian kernel fails to produce bokeh balls, while the undifferentiable disk kernel and the methods from [Gwosdek et al. 2011] and [Busam et al. 2019] yield similar reblurring outcomes. In contrast, our kernel replicates the subtle soap bubble effect, closely resembling the captured real bokeh ball (GT blur). Moreover, we present the effect of different kernel radius r and the effect of parameter β in the last two rows of Fig. 4 respectively.

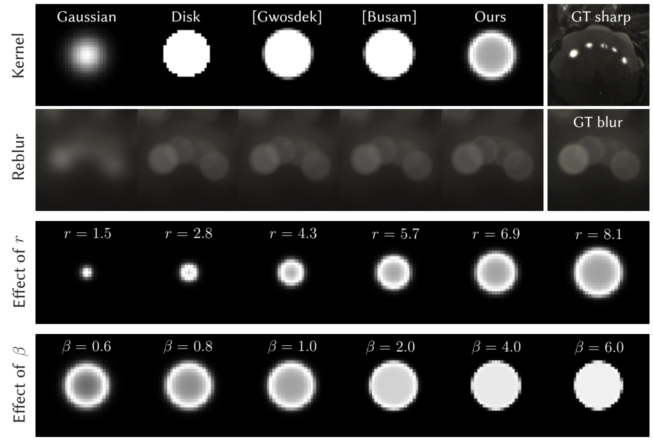

Figure 4: Our differentiable disk kernel. In the first two rows, we compare with the Gaussian kernel, an undifferentiable disk kernel, and methods from [Gwosdek et al. 2011] and [Busam et al. 2019]. We show real sharp and blurred images (GT sharp and GT blur) and their reblurred versions using these kernels. In the third row, we vary our differentiable disk kernel's radius r, and in the last row, we demonstrate the fall-off effect controlled by β.

3.4 COC Map Estimation

Our lens-blur CUDA layer requires per-pixel input regarding the radius of COC. Typically, a physical thin lens model is employed to compute COC from depth, which relies on camera-specific parameters when capturing this particular video, such as aperture size, focal length, and focus distance. Capturing depth and such metadata would greatly limit the applicability of our method. In our pipeline, we address this challenge by training a network $\mathbb{C}$ to directly estimate the COC radius $r^{\mathbf{p}} = \mathbb{C}(c^{\mathbf{p}}_{\text{blur}})$ in pixel units from each frame. To this end, we employ the concept of transfer learning and draw upon encoded defocus knowledge from existing defocus deblurring networks. Specifically, we train a defocus map estimation network structured similarly to [Ruan et al. 2023] due to its lightweight and efficient design, which aligns with our requirements. We fix the weights of their encoder, which encodes rich defocus information after being trained on large datasets, and exclusively train the modified decoder to estimate the defocus map rather than a sharp image. We show the effectiveness of this strategy in Sec 4.2 and generalize well on real image defocus estimation. To supervise the network training, we create a large defocus map estimation dataset, which includes rendered defocus images and corresponding ground truth COC maps. We select 22 animations from Blender Open Movies [Blender 2024] and render 25 distinct defocus variations for each frame, encompassing 5 different defocus levels (aperture sizes) and 5 focus distances. In total, we prepare a dataset comprising 27K pairs of defocus and COC maps, where COC map is obtained based on the focal distance, f-number and depth map.

3.5 Loss

In training our network, we incorporate a subset of loss functions from the original layered neural atlas approach. These include $\mathcal{L}_{rigid}$, which ensures a rigid mapping to the atlas for intuitive editing through the Jacobian matrix; $\mathcal{L}_{flow}$, which minimizes disparities between corresponding points in the video, ensuring consistency; and $\mathcal{L}_{sparsity}$, which prevents redundant representations in both foreground and background atlases. Please refer to [Kasten et al. 2021] for more details. We combine our losses as follows:

$$\mathcal{L} = \sum_{i=1}^{n}(\lambda_1\mathcal{L}_{reco} + \lambda_2\mathcal{L}_{flow} + \lambda_3\mathcal{L}_{sparsity} + \lambda_4\mathcal{L}_{rigid}), \quad (4)$$

Specifically, for $\mathcal{L}_{reco}$ between the blurry input video and reblurred frames as shown in Fig. 3, we use the L_1 loss, together with multi-scale structural similarity index (MS-SSIM) as suggested in [Zhao et al. 2016] for the image restoration. λ_n are our weights of losses during training, the choice of which we explain in the following section.

4 EXPERIMENTS

4.1 Implementation details

We evaluate our pipeline using videos with dimensions of 512×288, each consisting of approximately 60 frames. The training process was conducted on an NVIDIA Quadro RTX 8000, employing the Adam optimizer with a learning rate of 1e-4 over 50,000 iterations. To control the loss functions, we set the weights λ_1 to λ_4 to 0.3, 1.5, 100, and 0.5, respectively. We set the atlas resolution at 640×360, with the input noise dimension being eight times smaller than that of the atlas.

In line with the original layered neural atlas approach, we performed preliminary estimations of optical flow using RAFT [Teed and Deng 2020] and generated masks using SAM [Kirillov et al. 2023] for the first frame and XMem [Cheng and Schwing 2022] for subsequent frames. The training duration ranged from 6 to 8 hours. When training the *COC Map Estimation Network*, we start with a learning rate of 3e-4 for 300,000 iterations, followed by a linear decay to 1e-6 over another 300,000 iterations.

To assess our pipeline's effectiveness, we evaluated two types of videos: rendered and real. For rendered videos, we modified four clips sourced from Blender Open Movies [Blender 2024] (distinct from our COC training dataset) to simulate camera misfocus. We utilized five clips of real videos, two of which were captured using a Canon 6D II camera, while the remaining three were obtained from online sources.

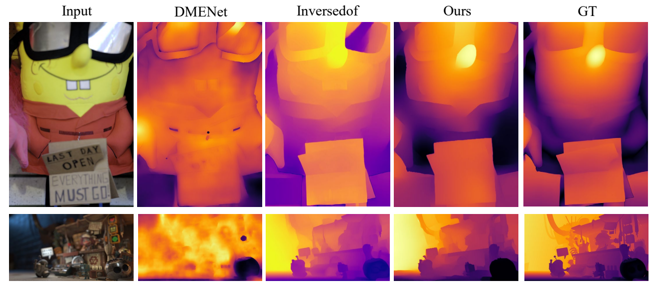

Figure 5: Qualitative comparison of COC map estimation on our semi-synthetic images and BLB dataset [Peng et al. 2022].

4.2 Results & Comparisons

We compare our approach to RDPD (RSDP+ variant) [Abuolaim et al. 2021] - the first framework that tackles defocus deblurring on image sequences, and Neucam (video deblur variant) [Huang et al. 2023] - video deblurring that uses implicit camera model. Note that while Neucam does not test video *defocus* deblurring in their paper, in our experiments, Neucam always converges and learns defocus deblurring.

We present in-depth qualitative comparisons for real captured video and rendered animation in Fig. 9. Among the methods evaluated, RDPD performs the least effectively in defocus deblurring, as the defocus blur remains unchanged in the original video frames, as evidenced in the insets of each frame and the time-dimensional data slice. Neucam, while capable of recovering sharp information, introduces noticeable artifacts. In contrast, our approach excels in both the quality of sharp content reconstruction and consistency. It's worth noting that small details, such as facial expressions, pose a challenge for all algorithms, including ours, as illustrated by the face of the truck driver in Fig. 9.

Table 1: Quantitative comparison on our rendered datasets.

Method	PSNR↑	SSIM↑	FovVideoVDP↑	Flip ↓	tPSNR↑
Neucam	25.17	0.86	5.66	0.13	28.64
RDPD	27.93	0.92	6.91	0.10	31.81
Ours	**29.23**	**0.95**	**7.20**	**0.07**	**33.15**

We add corresponding quantitative comparison only on rendered animation in Tab. 1, as the ground truth for real videos is not available. Besides the PSNR and SSIM, we also use perception-informed FovVideoVDP [Mantiuk et al. 2021], Flip [Andersson et al. 2020], and tPSNR [Banitalebi-Dehkordi et al. 2016] for capturing the distortions over large areas. Our approach achieves the best performance across all metrics.

COC map estimation. We assess our COC map estimation network in comparison to the top two approaches, namely, DMENet [Lee et al. 2019] and Inversedof [Piché-Meunier et al. 2023] (IDOF), as depicted in Fig. 5 and summarized in Table 2. For quantitative

Table 2: A quantitative analysis of our COC map estimation network, based on RMSE across two datasets, and we show both trainable parameters and total, denoted by a slash, along with the volume of the dataset used for training. We exclude DMENet from this quantitative comparison due to its differing representation (DMENet employs sigma, in contrast to our use of COC diameter) but include it in the qualitative analysis.

Method	BLB	SSD	Params (M)	Dataset (Train)
IDOF	8.30	2.30	156/156	1120K
Ours	8.77	2.80	10/37.5	37k

comparison, we perform the evaluation on two datasets: the BLB dataset [Peng et al. 2022], employing the Blender's Cycles renderer, allowing for a realistic representation of defocus effects while providing absolute scale COC values. Additionally, we construct a semi-synthetic dataset (SSD) using Bokehme [Peng et al. 2022], combining the traditional physics-based rendering approach with a neural rendering method, as recommended by [Piché-Meunier et al. 2023], and apply it to real RGB images. We randomly select 300 images from the FiveK Dataset [Bychkovsky et al. 2011] and generate 15 unique defocus variations for each image. These variations include 5 different defocus levels and 3 distinct focus distances, resulting in a test dataset of 4.5k images. Note that our neural network was not trained on these datasets; they were employed exclusively for testing purposes.

Table 2 illustrates our method's performance, which is comparable to IDOF, albeit with marginally lower numbers. IDOF utilizes three Transformer-based structures to estimate disparity, defocus, and per-pixel weights between them, enhancing accuracy at a cost approximately five times greater than our approach. Additionally, IDOF trained on a dataset of roughly 1120k images, whereas our strategy was trained on a more modest dataset of 37k images. This dataset comprised approximately 11k real images originally intended for single-image defocus deblurring and around 26k images rendered for training purposes.

Table 3: Ablation study on the COC network training strategy evaluated on the two datasets using RMSE.

Method	BLB	SSD
Fixed encoder	8.77	2.80
Train from scratch	12.26	3.43

Our method, illustrated in Fig. 5, roughly matches the visual quality of IDOF and also adeptly manages difficult areas, such as glass. This performance mainly stems from the proficiency of our pre-trained encoder in extracting unique features from defocused images. This capability enhances the accuracy of defocus map estimation and shows strong generalization in real image defocus map estimation tasks. The ablation study results, detailed in Table 3, further support the effectiveness of this approach. Notably, the performance diminished when we trained the encoder from scratch using our rendered dataset rather than utilizing a pre-trained encoder.

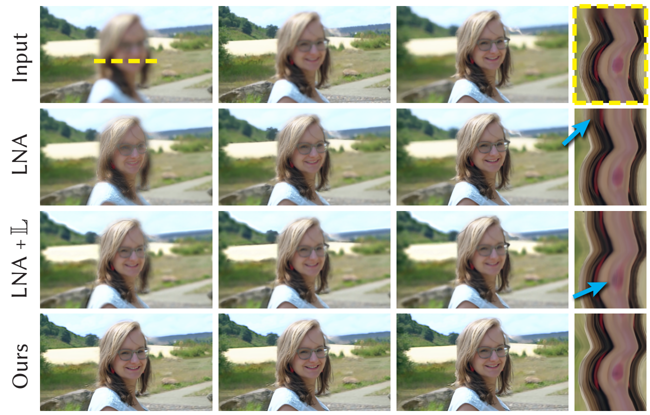

Figure 6: The comparison in video defocus deblurring involves using the original Layer Neural Atlas paper, adapting the Layer Atlas work with our reblur module, and our approach, which learns the atlas utilizing the deep image prior. © 2024 Cinecom Belgium BV

Effect of COC map estimation network $\mathbb{C}$. In addition to demonstrating the performance of the COC map itself with state-of-the-art (SOTA) algorithms, we also evaluate its impact on our pipeline. We compare it against alternative methods by either removing it entirely, identified as the baseline, or replacing it with existing COC estimation methods. We present results in terms of PSNR and SSIM for both reconstructed all-in-focus and reblurred images in Tab. 4.

The inaccurate COC map from DMENet leads to inferior performance, whereas our results are on par with IDOF regarding estimated COC (see Tab. 2) and translate to comparable restoration quality. This suggests that an accurate COC map is crucial for precise deblurring and reblurring performance.

Table 4: Ablation study on the impact of the COC map estimation network in our pipeline. We present the quantitative assessment of reblurred and deblurred images of the truck scene. Data are denoted in format Reblur/Sharp in the table.

Metric	Baseline	DMENet	IDOF	Ours
PSNR	35.20/29.55	38.18/31.18	39.14/32.07	39.76/32.19
SSIM	0.917/0.796	0.962/0.872	0.977/0.915	0.978/0.914

Comparison to [Kasten et al. 2021]. Figure 6 presents a qualitative comparison of the layered neural atlas, the same atlas equipped with our reblur module, and our present solution. The layer neural atlas, which is originally designed for sharp RGB images, effectively reconstructs video content. However, the presence of spatially and temporally varying blur introduces ambiguities across frames, leading to a ghosting effect on faces. Merely appending the reblur module, as described in Sec. 3.2, fails to recover sharp details. In contrast, our proposed methodology leverages the deep image prior and incorporates the COC map constraint, ultimately delivering superior visual quality.

Figure 7: The optical flow directly estimated from our UV maps achieves comparable performance when compared to [Teed and Deng 2020]. ***Agent 327*** **© 2024 Blender**

5 APPLICATIONS

In this section, we showcase two practical applications that directly leverage our pipeline's capabilities. Firstly, we illustrate how focus tracking for selected scene components can be seamlessly performed on the atlases generated by our system. Secondly, we demonstrate that our learned UV map enables the recovery of improved optical flow in defocused videos, achieving somewhat comparable performance of the state-of-the-art optical flow estimation algorithm described in [Teed and Deng 2020].

5.1 Focus Tracking

The layered neural atlas representation allows the refocusing process to be edited on a 2D atlas image, which is then mapped back to the original video frames. As demonstrated in Fig. 1, by keeping the foreground atlas and only adding blur to the background atlas, we can correct and simulate the focus tracking that failed during capturing. Please refer to the supplementary video.

5.2 Optical Flow Estimation

We approximate a linear transformation from an arbitrary atlas point (u, v) to an (x, y, t) pixel coordinate in an arbitrary frame t as the product with the inverse Jacobian of the UV transform learned by the mapping network $\boldsymbol{J}_{\mathbb{M}}^{-1}$ offset by a constant vector offset $\boldsymbol{o}$:

$$\begin{pmatrix} x \\ y \\ t \end{pmatrix} = \boldsymbol{J}_{\mathbb{M}}^{-1}(x', y', t) \begin{pmatrix} u \\ v \end{pmatrix} + \boldsymbol{o}(x', y', t) \; . \tag{5}$$

Using Pytorch's `jacrev` function, we compute and invert the Jacobian at a pixel (x', y', t). We choose (x', y', t) from all integer pixel coordinates in frame t such that $\mathbb{M}(x', y', t)$ is closest to (u, v) by evaluating $\mathbb{M}$ on frame t followed by a nearest neighbour lookup. We compute the offset $\boldsymbol{o}$ as:

$$\boldsymbol{o}(x', y', t) = \begin{pmatrix} x' \\ y' \\ t \end{pmatrix} - \boldsymbol{J}_{\mathbb{M}}^{-1}(x', y', t)\mathbb{M}(x', y', t) \; . \tag{6}$$

Linearly interpolating the nearest inverse Jacobians and offsets with `scipy.interpolate.LinearNDInterpolator` further improves results. We compute flow from frame t_0 to frame t_1 by mapping pixel coordinates (x, y, t_0) to (u, v) using $\mathbb{M}$, and, with the described inverse transform, computing (x, y, t_1) from the (u, v) atlas points.

Table 5 and Fig. 7 compare the estimated optical flow between our method and RAFT [Teed and Deng 2020], testing on various frame intervals across four scenes. Our method is comparable to RAFT on neighbor frames and outperforms it on long-range optical flow.

Table 5: Quantitative comparisons of optical flow: our method / RAFT [Teed and Deng 2020].

Interval	RMSE	EndPoint Error	Angular Error	Length Error
1	0.598/0.614	0.631/0.592	15.10/9.818	0.433/0.493
3	1.654/1.773	1.685/1.543	12.45/10.67	1.173/1.217
5	2.657/3.330	2.606/2.442	10.95/11.14	1.842/1.793
7	3.778/4.109	3.432/3.168	9.628/11.45	2.510/2.217
9	5.310/6.193	4.406/4.653	8.978/11.23	3.304/3.493

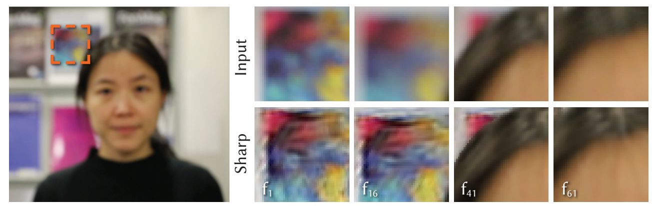

Figure 8: Our method exhibits limitation when objects experience only a few large defocus levels, as depicted in the inset shown here. The upper row indicates that it is heavily defocused and then occluded (see the frame index). While we could restore it, the restoration performance is limited.

6 CONCLUSION

In conclusion, we propose a systematic end-to-end solution to video defocus deblurring and editing. By generating and parameterizing defocused video into layered neural atlases through a differentiable thin lens model within a self-supervised network, our method enables achieves consistent, sharp video reconstruction and focus tracking in post-processing. Furthermore, we implement a lens blur CUDA layer featuring a novel differentiable disk kernel that accurately simulates the realistic fall-off boundary of the PSF.

Limitations. Similar to the original layered neural atlases [Kasten et al. 2021], our model could not handle videos with objects in large self-occlusion, which require more layers of atlases. Notably, our method excels when objects experience many distinct defocus blur levels. However, performance worsens when fewer defocus levels are present, as illustrated in Fig. 8. The inset is heavily defocused and then occluded by the person, we can restore it but to a limited extent. This is attributed to the high degrees of freedom inherent in our problem.

ACKNOWLEDGMENTS

We would like to thank Dominique Piché-Meunier for helping with the evaluation of their method [Piché-Meunier et al. 2023] on our dataset and for the valuable comments from anonymous reviewers.

REFERENCES

Abdullah Abuolaim and Michael S. Brown. 2020. Defocus deblurring using dual-pixel data. In *Proceedings of the European Conference on Computer Vision*. Springer, 111–126. https://doi.org/10.1007/978-3-030-58607-2_7

Abdullah Abuolaim, Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. 2021. Learning to reduce defocus blur by realistically modeling dual-pixel data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2269–2278. https://doi.org/10.1109/ICCV48922.2021.00229

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. 2020. FLIP: A difference evaluator for alternating images. In *Proceedings of the ACM on Computer Graphics and Interactive Techniques*. 1–23. https://doi.org/10.1145/3406183

Amin Banitalebi-Dehkordi, Maryam Azimi, Mahsa T. Pourazad, and Panos Nasiopoulos. 2016. Visual saliency aided High Dynamic Range (HDR) video quality metrics. In *Proceedings of the IEEE International Conference on Communications Workshops*. 486–491. https://doi.org/10.1109/ICCW.2016.7503834

Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. 2022. Text2live: Text-driven layered image and video editing. In *Proceedings of the European Conference on Computer Vision*. Springer, 707–723. https://doi.org/10.1007/978-3-031-19784-0_41

Blender. 2024. Blender Open Movies. https://studio.blender.org/films/.

Benjamin Busam, Matthieu Hog, Steven McDonagh, and Gregory Slabaugh. 2019. Sterefo: Efficient image refocusing with stereo vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop*. 3295–3304. https://doi.org/10.1109/ICCVW.2019.00411

Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 97–104. https://doi.org/10.1109/CVPR.2011.5995332

Rudi Chen and Peter van Beek. 2015. Improving the accuracy and low-light performance of contrast-based autofocus using supervised machine learning. *Pattern Recogn. Lett.* 56, C (2015), 30–37. https://doi.org/10.1016/j.patrec.2015.01.010

Ho Kei Cheng and Alexander G Schwing. 2022. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Proceedings of the European Conference on Computer Vision*. Springer, 640–658. https://doi.org/10.1007/978-3-031-19815-1_37

Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Süsstrunk. 2016. Non-parametric blur map regression for depth of field extension. *IEEE Transactions on Image Processing* 25, 4 (2016), 1660–1673. https://doi.org/10.1109/TIP.2016.2526907

Ray Fontaine. 2017. A survey of enabling technologies in successful consumer digital imaging products. In *Proceedings of the International Image Sensors Workshop*.

Shir Gur and Lior Wolf. 2019. Single image depth estimation trained via depth from defocus cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7683–7692. https://doi.org/10.1109/CVPR.2019.00787

Pascal Gwosdek, Sven Grewenig, Andrés Bruhn, and Joachim Weickert. 2011. Theoretical foundations of gaussian convolution by extended box filtering. In *Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 447–458. https://doi.org/10.1007/978-3-642-24785-9_38

Daniel Hernandez-Juarez, Lukas Schneider, Antonio Espinosa, David Vázquez, Antonio M López, Uwe Franke, Marc Pollefeys, and Juan C Moure. 2019. Slanted stixels: A way to represent steep streets. *International Journal of Computer Vision* 127 (2019), 1643–1658. https://doi.org/10.1007/s11263-019-01226-9

Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, and Qing Wang. 2023. Inverting the Imaging Process by Learning an Implicit Camera Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21456–21465. https://doi.org/10.1109/CVPR52729.2023.02055

Daisuke Inoue and Hidekazu Takahashi. US Patent 7,577,349, aug. 18, 2009. Focus detecting device and camera system using the same device.

Ali Karaali, Naomi Harte, and Claudio R Jung. 2022. Deep multi-scale feature learning for defocus blur estimation. *IEEE Transactions on Image Processing* 31 (2022), 1097–1106. https://doi.org/10.1109/TIP.2021.3139243

Ali Karaali and Claudio Rosito Jung. 2017. Edge-based defocus blur estimation with adaptive scale selection. *IEEE Transactions on Image Processing* 27, 3 (2017), 1126–1137. https://doi.org/10.1109/TIP.2017.2771563

Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. 2021. Layered neural atlases for consistent video editing. *ACM Trans. Graph.* 40, 6 (2021). https://doi.org/10.1145/3478513.3480546

Hyeongwoo Kim, Christian Richardt, and Christian Theobalt. 2016. Video Depth-from-Defocus. In *Proceedings of the International Conference on 3D Vision*. 370–379. https://doi.org/10.1109/3DV.2016.46

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3992–4003. https://doi.org/10.1109/ICCV51070.2023.00371

Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. 2019. Deep defocus map estimation using domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12214–12222. https://doi.org/10.1109/CVPR.2019.01250

Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. 2021. Iterative filter adaptive network for single image defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2034–2042. https://doi.org/10.1109/CVPR46437.2021.00207

Chenyang Lei, Xuanchi Ren, Zhaoxiang Zhang, and Qifeng Chen. 2023. Blind video deflickering by neural filtering with a flawed atlas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10439–10448. https://doi.org/10.1109/CVPR52729.2023.01006

Victor Lempitsky, Andrea Vedaldi, and Dmitry Ulyanov. 2018. Deep Image Prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9446–9454. https://doi.org/10.1109/CVPR.2018.00984

Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. 2023. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4273–4284. https://doi.org/10.1109/CVPR52729.2023.00416

Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. 2021. Mask textSpotter: an end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 2, 532–548. https://doi.org/10.1109/TPAMI.2019.2937086

Feng-Lin Liu, Shu-Yu Chen, Yukun Lai, Chunpeng Li, Yue-Ren Jiang, Hongbo Fu, and Lin Gao. 2022. Deepfacevideoediting: sketch-based deep editing of face videos. *ACM Trans. Graph.* 41, 4 (2022). https://doi.org/10.1145/3528223.3530056

Xianrui Luo, Juewen Peng, Ke Xian, Zijin Wu, and Zhiguo Cao. 2023. Defocus to focus: Photo-realistic bokeh rendering by fusing defocus and radiance priors. *Inf. Fusion* 89, C (2023), 320–335. https://doi.org/10.1016/j.inffus.2022.08.023

Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V. Sander. 2022. Deblur-NeRF: Neural Radiance Fields from Blurry Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12851–12860. https://doi.org/10.1109/CVPR52688.2022.01252

Rafał K. Mantiuk, Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney. 2021. FovVideoVDP: A visible difference predictor for wide field-of-view video. *ACM Trans. Graph.* 40, 4 (2021). https://doi.org/10.1145/3450626.3459831

Daniel Miau, Oliver Cossairt, and Shree K Nayar. 2013. Focal sweep videography with deformable optics. In *Proceedings of the IEEE International Conference on Computational Photography*. IEEE, 1–8. https://doi.org/10.1109/ICCPhot.2013.6528302

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106. https://doi.org/10.1145/3503250

Jinsun Park, Yu-Wing Tai, Donghyeon Cho, and In So Kweon. 2017. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2760–2769. https://doi.org/10.1109/CVPR.2017.295

Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. 2022. Bokehme: When neural rendering meets classical rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16262–16271. https://doi.org/10.1109/CVPR52688.2022.01580

Sida Peng, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2023. Representing volumetric videos as dynamic MLP maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4252–4262. https://doi.org/10.1109/CVPR52729.2023.00414

Dominique Piché-Meunier, Yannick Hold-Geoffroy, Jianming Zhang, and Jean-François Lalonde. 2023. Lens Parameter Estimation for Realistic Depth of Field Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 499–508. https://doi.org/10.1109/ICCV51070.2023.00052

Michael Potmesil and Indranil Chakravarty. 1982. A lens and aperture camera model for synthetic image generation. *ACM Trans. Graph.* 1, 2 (1982), 85–108. https://doi.org/10.1145/357299.357300

Yuhui Quan, Zicong Wu, and Hui Ji. 2021. Gaussian kernel mixture network for single image defocus deblurring. *Proceedings of the Advances in Neural Information Processing Systems* 34, 20812–20824.

Yuhui Quan, Zicong Wu, and Hui Ji. 2023. Neumann network with recursive kernels for single image defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5754–5763. https://doi.org/10.1109/CVPR52729.2023.00557

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint* (2018). https://doi.org/10.48550/arXiv.1804.02767

Lingyan Ruan, Mojtaba Bemana, Hans-peter Seidel, Karol Myszkowski, and Bin Chen. 2023. Revisiting Image Deblurring with an Efficient ConvNet. *arXiv preprint arXiv:2302.02234* (2023). https://doi.org/10.48550/arXiv.2302.02234

Lingyan Ruan, Bin Chen, Jizhou Li, and Miuling Lam. 2022. Learning to deblur using light field generated and real defocus images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16304–16313. https://doi.org/10.1109/CVPR52688.2022.01582

Lingyan Ruan, Bin Chen, Jizhou Li, and Miu-Ling Lam. 2021. Aifnet: All-in-focus image restoration network using a light field-based dataset. *IEEE Transactions on Computational Imaging* 7 (2021), 675–688. https://doi.org/10.1109/TCI.2021.3092891

Jianping Shi, Li Xu, and Jiaya Jia. 2015. Just noticeable defocus blur detection and estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 657–665. https://doi.org/10.1109/CVPR.2015.7298665

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1874–1883. https://doi.org/10.1109/CVPR.2016.207

Pratul P Srinivasan, Rahul Garg, Neal Wadhwa, Ren Ng, and Jonathan T Barron. 2018. Aperture supervision for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6393–6401. https://doi.org/10.1109/CVPR.2018.00669

Huixuan Tang and Kiriakos N Kutulakos. 2012. Utilizing optical aberrations for extended-depth-of-field panoramas. In *Proceedings of the Asian Conference on Computer Vision*. Springer, 365–378. https://doi.org/10.1007/978-3-642-37447-0_28

Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of European Conference on Computer Vision*. Springer, 402–419. https://doi.org/10.1007/978-3-030-58536-5_24

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Proceedings of the Advances in Neural Information Processing Systems* 34, 24261–24272.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. 2022. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5769–5780. https://doi.org/10.1109/CVPR52688.2022.00568

Quoc Kien Vuong and Jeong-won Lee. 2013. Initial direction and speed decision system for auto focus based on blur detection. In *Proceedings of the International Conference on Consumer Electronics*. IEEE, 222–223. https://doi.org/10.1109/ICCE.2013.6486867

Neal Wadhwa, Rahul Garg, David E Jacobs, Bryan E Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T Barron, Yael Pritch, and Marc Levoy. 2018. Synthetic depth-of-field with a single-camera mobile phone. *ACM Trans. Graph.* 37, 4 (2018), 1–13. https://doi.org/10.1145/3197517.3201329

Chengyu Wang, Qian Huang, Ming Cheng, Zhan Ma, and David J Brady. 2021. Deep learning for camera autofocus. *IEEE Transactions on Computational Imaging* 7 (2021), 258–271. https://doi.org/10.1109/TCI.2021.3059497

Chao Wang, Ana Serrano, Xingang Pan, Krzysztof Wolski, Bin Chen, Karol Myszkowski, Hans-Peter Seidel, Christian Theobalt, and Thomas Leimkühler. 2023. An Implicit Neural Representation for the Image Stack: Depth, All in Focus, and High Dynamic Range. *ACM Trans. Graph.* 42, 6 (2023). https://doi.org/10.1145/3618367

Zijin Wu, Xingyi Li, Juewen Peng, Hao Lu, Zhiguo Cao, and Weicai Zhong. 2022. DoF-NeRF: Depth-of-Field Meets Neural Radiance Fields. In *Proceedings of the ACM International Conference on Multimedia*. 1718–1729. https://doi.org/10.1145/3503161.3548088

Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. 2022. Deformable sprites for unsupervised video decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2647–2656. https://doi.org/10.1109/CVPR52688.2022.00268

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5728–5739. https://doi.org/10.1109/CVPR52688.2022.00564

Anmei Zhang and Jian Sun. 2021. Joint depth and defocus estimation from a single image using physical consistency. *IEEE Transactions on Image Processing* 30 (2021), 3419–3433. https://doi.org/10.1109/TIP.2021.3061901

Xuaner Zhang, Kevin Matzen, Vivien Nguyen, Dillon Yao, You Zhang, and Ren Ng. 2019. Synthetic defocus and look-ahead autofocus for casual videography. *ACM Trans. Graph.* 38, 4 (2019). https://doi.org/10.1145/3306346.3323015

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. 2016. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging* 3, 1 (2016), 47–57. https://doi.org/10.1109/TCI.2016.2644865

Changyin Zhou, Daniel Miau, and Shree K Nayar. 2012. Focal sweep camera for space-time refocusing. (2012). https://doi.org/10.7916/D8V69SZB

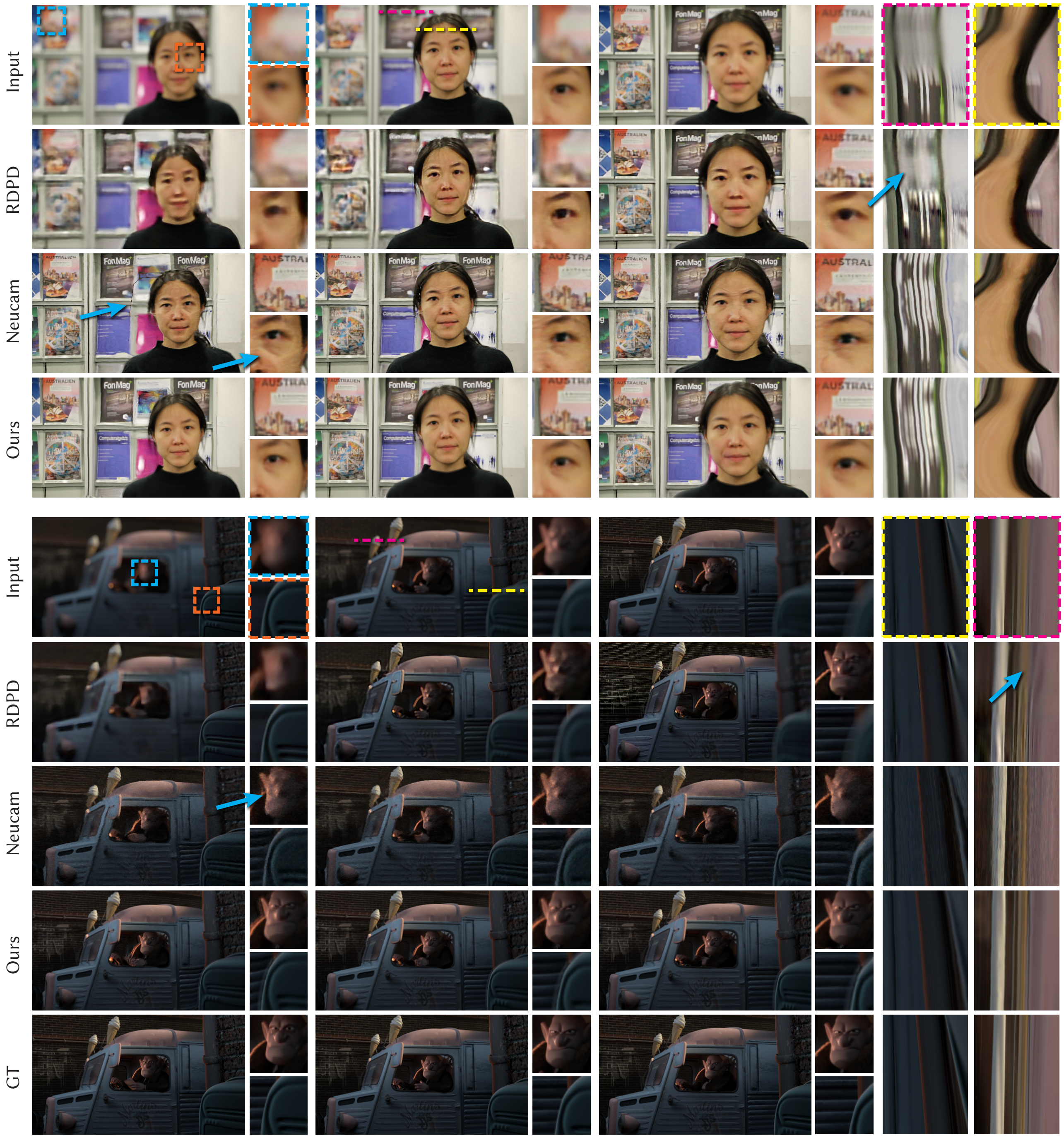

Figure 9: We compare our method to relate work on real captured video (up) and rendered animation (bottom). We crop and enlarge the small inserts for better comparison with region locations indicated by blue and orange dash line rectangle. We slice two pieces of data (pink and yellow dash line) on time dimension and visualize at the last two columns for temporal consistency comparison. Blue arrows have been used to highlight the artifacts and temporal inconsistency. ***Agent 327*** **© 2024 Blender**