Toward Multi-Plane Image Reconstruction from a Casually Captured Focal Stack

Shiori Ueda[1][a], Hideo Saito[1][b], and Shohei Mori[2,1][c]
[1]*Faculty of Science and Technology, Keio University, Yokohama, Kanagawa, Japan*
[2]*Institute of Computer Graphics and Vision, Graz University of Technology, Graz, Styria, Austria*
{*shiori.ueda, hs*}*@keio.jp, s.mori.jp@ieee.org*

Keywords: Focal Stack Imaging, Casual Photography, Multi-Plane Image, View Synthesis.

Abstract: 3D imaging combining a focal stack and multi-plane images (MPI) facilitates various real-time applications, including view synthesis, 3D scene editing, and augmented and virtual reality. Building upon the foundation of MPI, originally derived from multi-view images, we introduce a novel pipeline for reconstructing MPI by casually capturing a focal stack *optically* using a handheld camera with a manually modulated focus ring. We hypothesized two distinct strategies for focus ring modulation that users could employ, to sample defocus images along the front-facing axis uniformly. Our quantitative analysis using a synthetic dataset suggests tendencies in possible simulated errors in focus modulations, while qualitative results illustrate visual differences. We further showcase applications utilizing the resultant MPI, including depth rendering, occlusion-aware defocus filtering, and de-fencing.

1 INTRODUCTION

A focal stack is a series of images optically focused at different distances. A focal stack is known as an approximated representation of a set of multi-view images on a 2D grid, or a light field, that can reproduce ambient surfaces from the captured range (Pérez et al., 2016). As such, focal stack imaging has been applied to light field displays (Takahashi et al., 2018), all-in-focus image generation (Kim et al., 2016), and free-viewpoint image synthesis (Ishikawa et al., 2023). Taking a focal stack often involves systematic focus changes to enable such useful applications and thus requires synthetic approaches or controllable optics.

One method is to synthesize approximated blur over an all-in-focus image using depth-dependent blur kernels (Kim et al., 2016) or through neural rendering techniques (Wu et al., 2022). Another method involves synthetic aperture photography, achieving synthetic yet optically accurate blur from multi-view observations (Vaish et al., 2004). Optical solutions include mechanical lenses (Subbarao and Choi, 1995), focus tunable lenses (Ebner et al., 2022), and shifting image sensors (Kuthirummal et al., 2011). However, the majority of cameras on the market does not have such functionalities, while finding kernels for blurring all-in-focus images is not trivial (Abuolaim et al., 2021).

We focus more on the user who manually twists a lens focus ring to take a focal stack. We investigate two different strategies (1) *continuous-rotation*, where the user holds and rotates the ring as linearly as possible and (2) *delta-rotation*, where the user repeats stop and rotate like a clock tick, during a video recording. To select a better approach, we create a synthetic dataset to simulate potential errors from the two different strategies. Spatial misalignment from handshakes is corrected using direct image alignments.

Based on the multi-plane image (MPI) from synthetic aperture photography (Ishikawa et al., 2023), we generate an MPI from a casually photographed focal stack. Therefore, our applications include ones that MPI can support, such as all-in-focus view synthesis, depth rendering, per-layer defocus, and defencing, as we demonstrate in this paper. In summary, we present the following contributions to MPI generation from an optically captured focal stack:

- We present a pipeline for generating an MPI from a casually captured focal stack (Figure 1(a–d)),
- investigate how two different focal stack imaging strategies with a handheld camera (i.e., *continuous-rotation* and *delta-rotation*) can impact the MPI rendering qual-

[a] https://orcid.org/0009-0007-3820-6919
[b] https://orcid.org/0000-0002-2421-9862
[c] https://orcid.org/0000-0003-0540-7312

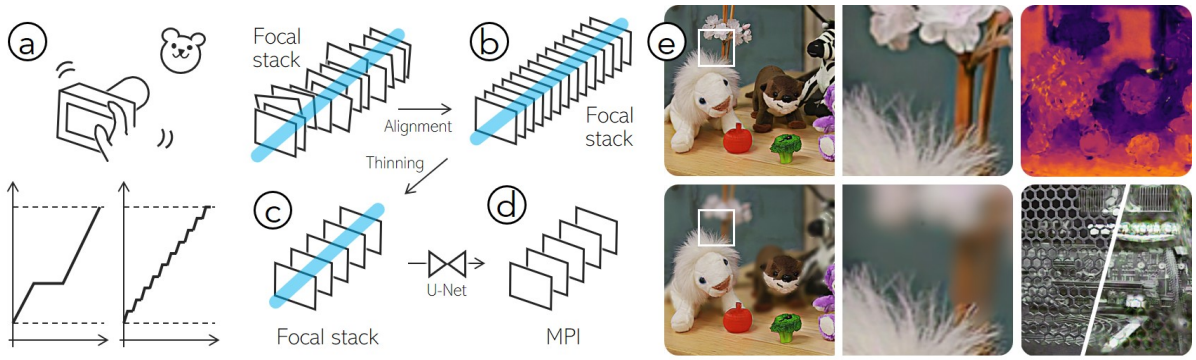

Figure 1: Pipeline of MPI generation from a casual focal stack with a manually controlled camera. (a) The user takes a focal stack either by rotating the focus ring at once in a sequence (*continuous-rotation*) or by repeating rotating and stopping the focus ring for a short interval (*delta-rotation*). We discuss which approach is better regarding MPI rendering quality in our experiment. (b) A vision technique corrects spatial misalignment over the focal stack images. (c) A fixed number of focal stack images is selected. (d) A U-Net that is trained by analysis by synthesis generates MPI. (e) Applications include depth rendering, per-layer defocus for occlusion-aware depth of field effect, and de-fencing.

ity on a synthetic dataset (Figure 1(a)),

- and demonstrate real use cases using a camera in real scenes (Figure 1(e)).

2 RELATED WORK

We overview three related domains to our research to provide a rationale for employing casual focal stack imaging in the context of MPI generation.

2.1 Casual Photography

Casual 3D photography (Hedman et al., 2017) has been an attractive research topic that aims to reduce users' workload and mental demands. A typical example is panorama image capture with a mobile phone, which presents instant visual feedback and guidance, such as intermediate image stitching results and the rectangular representing the current field of view on the screen (Wagner et al., 2010).

For light field imaging, 3D annotations with prominent colors are often used. 3D annotations include 3D axes floating in the air as reference camera poses (Mildenhall et al., 2019), a 3D plane suggesting a free-form capture area (Ishikawa et al., 2023), a 3D dome visualizing the current coverage (Mohr et al., 2020; Davis et al., 2012), a 1D trajectory for a rotation camera motion like panorama imaging (Tomoto et al., 2020), and hitting a dodging virtual ball to gamify the capturing (Birklbauer and Bimber, 2015).

No specific visual guidance is needed if the capture motion is simple. This can include rotating a hand-held camera with a stretched arm (Baker et al., 2020), moving around with a 360 camera that captures a wide range of the scene at once (Bertel et al., 2020), or when only a single shot (Han et al., 2022) or a few images (Khan et al., 2023) are required.

We also employ an approach without visual guidance. The task for the user in our focal stack photography is to continuously and uniformly rotate a camera focus ring to cover the entire scene. We examine two different approaches to see their impact to MPI generation and rendering.

2.2 Focal Stack Imaging

A focal stack is a series of images focused at different distances. Since multiple rays pass through a camera lens to form an image, a structurally captured focal stack is potentially an approximated light field (Pérez et al., 2016). A focal stack can be recorded with a modulated varifocal lens (Ebner et al., 2022) and a shifting image sensor (Kim et al., 2016), or synthetically with a synthetic aperture photography technique (Ishikawa et al., 2023).

All of the above works employ pre-calibrated hardware (Ebner et al., 2022; Kim et al., 2016) or a synthetically fully-controlled approach (Ishikawa et al., 2023). In contrast, our approach utilizes a handheld camera equipped with a manually modulated lens that introduces novel challenges in algorithmic design decisions and capture guidelines. We investigate two different focal stack capture strategies to derive an improved solution that ensures uniform scene coverage, thereby suppressing artifacts in MPI.

2.3 Multi-layer Scene Representation

Multi-layer scene representation employs a collection of RGB+α images mapped onto meshes to slice over

the scene. This technique is commonly referred to as MPI (Szeliski and Golland, 1999) or multi-sphere images (MSI) (Broxton et al., 2020). MPI represents the camera view frustum, and MSI envelops the viewer (i.e., a panoramic view). The data structure is explicit and thus directly editable (Mori et al., 2023) and fast to render with less capable graphics hardware. These characteristics are considered preferable for augmented reality (AR) and virtual reality (VR) applications (Ishikawa et al., 2023).

A deep neural network can infer MPI or MSI from a perspective (Han et al., 2022), stereo (Khakhulin et al., 2022), multi-view images (Mildenhall et al., 2019), and a focal stack (Ishikawa et al., 2023). We derive MPI from a focal stack similar to Ishikawa et al. (Ishikawa et al., 2023), while they focus on theoretical bounds to form a focal stack from multi-view images and evaluation of denoising aspects. Contrary to the work, our research interest lies in the recording process of a focal stack using a handheld camera with a manually modulated lens.

3 MULTI-PLANE IMAGE FROM A FOCAL STACK PIPELINE

Figure 1 illustrates our proposed pipeline of MPI generation from a casual focal stack. Given camera parameters, we calculate the supported minimum and maximum range. The user, therefore, takes a focal stack within that range, either by rotating the focus ring at once in a sequence (*continuous-rotation*, Figure 1(a) bottom-left) or by repeating rotating and stopping the focus ring for a short interval (*delta-rotation*, Figure 1(a) bottom-right). The captured focal stack images are aligned by homography warping between adjacent frames (Figure 1(a, b)). Only a necessary number of focal stack images is selected (Figure 1(b, c)) to generate an MPI using a U-Net-like network that is trained by analysis by synthesis (Figure 1(c, d)).

3.1 Focal Stack Photography Strategies

Cameras do not save focus distances at every frame while recording focus-modulated videos. Therefore, we record only the first and the last focus distances and calculate the in-between focus distances by uniformly dividing the minimum and maximum inverse depth difference by the number of required focal stack images. However, we have observed that the strategies employed in photography can impact the linearity of focus modulation. Consequently, we developed two strategies to ensure evenly spaced focus distances.

Continuous-rotation. One straightforward strategy is photographing a focal stack by rotating the focus ring from the minimum to maximum distances in one continuous motion. Major concerns of this approach include user-dependent accuracy in linearity and several separate modulations when the focus range is vast and the user fails to rotate the ring in a single motion (Figure 2(a, b)).

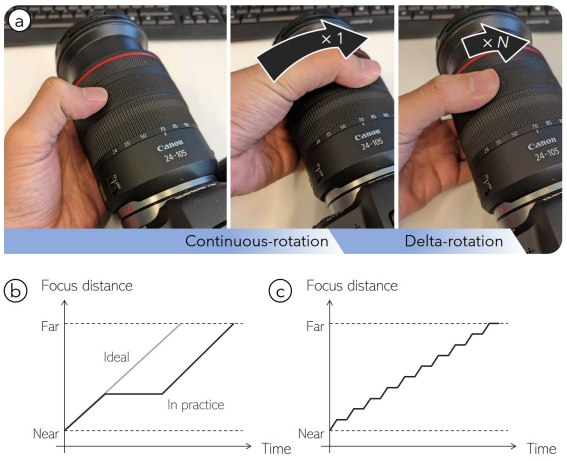

Figure 2: Two focal stack photography strategies. (a) The user may photograph a focal stack either by rotating the focus ring from the minimum to the maximum distances or by M times. (b) When the focus range is vast, the user may fail to rotate the ring at once and repeat rotating it with large intervals (*continuous-rotation*). (c) To avoid such a large interval, the user may rotate the ring by small steps until the maximum focus distance is reached (*delta-rotation*).

Delta-rotation. The second strategy is designed to overcome the first approach. The focus ring is rotated in small steps until the maximum distance is reached from the minimum distance. Such an approach would introduce variations in the rotation speed and steps and quantization errors depending on the size of the steps.

3.2 Metadata-Based Thinning

We use N defocus images from a focal stack video as the input of our network. Such images must comprehensively traverse the scene depth so that every pixel appears sharp in focus at least one of the images. Ishikawa et al. derived theoretical bounds for the synthetic aperture size using camera parameters and depth range (Ishikawa et al., 2023). Conversely, we derive the depth range from a fixed lens aperture size and camera parameters. Given an aperture size A with camera field of view θ_{fov}, camera image width in pixels W_{px}, and the number of layers of the focal stack N, the minimum focus distance d_{min} and the maximum focus distance d_{max} are constrained by the following equation:

$$\frac{1}{d_{min}} - \frac{1}{d_{max}} \leq \frac{4C_{px}\tan(\theta_{fov}/2)(N-1)}{AW_{px}}, \qquad (1)$$

where C_{px} is the maximum circle of confusion in pixels, which is 1 for the highest quality. When the shooting settings are fixed, the aperture size and camera parameters are obtained. Once the user sets the minimum distance, the maximum distance can be calculated accordingly (and vice versa). Figure 3 shows the supported depth ranges with $N = 32, W_{px} = 1920$ depending on the shooting settings (focal length and f-number).

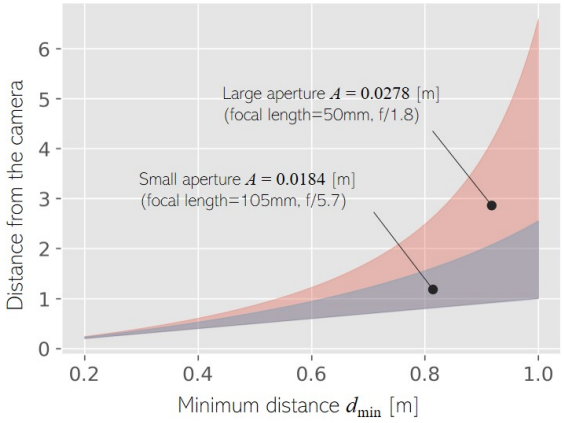

Figure 3: Supporting depth ranges for $N = 32$ and $W_{px} = 1920$ pixels depending on the aperture sizes.

We sample N images from the focal stack video so that the inverse distances are as equally spaced as possible based on the assumption that the focus distance of a series of frames moves at equal intervals between the determined d_{min} and d_{max}.

3.3 Focal Stack Alignment

Taking a focal stack in a sequence induces camera shaking and leads to misaligned images over time. Such misalignment must be corrected to avoid visual artifacts in the resultant MPI (Figure 4). Since differences in depth of field blur between two images are significant, we implement the direct alignment method that estimates Homography warping between two images, I_i and I_{i+1}, so that the overall difference in pixel colors gets minimum after applying the warping, $I_i(H(\cdot))$, (Baker and Matthews, 2004).

$$\arg\min_{\mathbf{p}} \sum_{\mathbf{x}} [I_i(H(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - I_{i+1}(\mathbf{x})], \tag{2}$$

where $\mathbf{x} = (x,y)^\top$ is a 2D pixel location, $\mathbf{p} = (p_1, p_2, ..., p_8)^\top$ and $\Delta\mathbf{p} = (\delta p_1, \delta p_2, ..., \delta p_8)^\top$ are a vector of Homography parameters and incremental parameters to be estimated, respectively. We repeat this process for every pair of i and $i+1$ with the minimum appearance differences in depth of field effects.

We focused on static scenes for brevity. For dynamic scenes, however, pixel-wise alignment is necessary. For such cases, we refer to the approaches (Ebner et al., 2022; Kim et al., 2016) based on the PatchMatch algorithm (Barnes et al., 2009).

3.4 Multi-Plane Image Generation

We used the same network architecture as Ishikawa et al. (Ishikawa et al., 2023) but trained it with a different loss function for efficiency. Given a light field (i.e., multi-view) dataset, we train our network with analysis by synthesis at five views (Figure 5). We first synthesize a focal stack from the center view at the N calculated focus distances. The synthesized N-layer focal stack is provided to the U-Net-like CNN. The network outputs N-layer MPI at the center view.

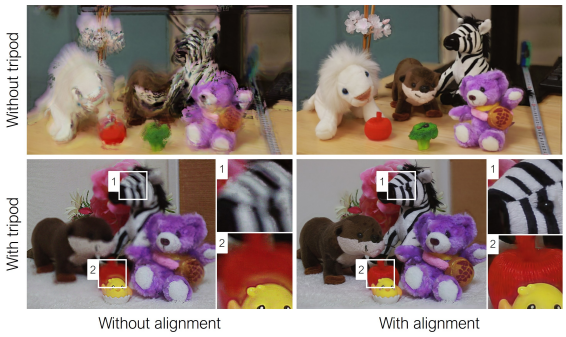

Figure 4: Misalignment and correction. All results here show MPI rendering. A shaky focal stack results in erratic MPI (top left), while alignment effectively suppresses the artifacts (top right). Stably rotating a focus ring is practically challenging even with a tripod (bottom left), and thus, alignment is still superior to have (bottom right).

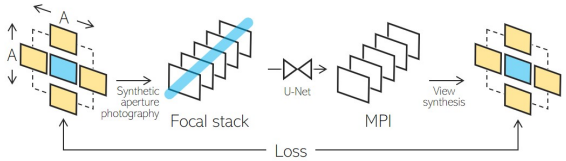

Figure 5: Training by synthesis. We compose a focal stack from a light field and generate an MPI from the focal stack via a U-Net like network. A differentiable renderer renders five views to calculate the loss. As illustrated in the figure, we use the center, top, bottom, right, and left views.

Each MPI layer consists of an RGBα image with the same height and width as the focal stack images. We assume the output MPI layers are evenly spaced in inverse depth between d_{min} and d_{max}. A differentiable renderer synthesizes a novel view by over alpha composition from back to the front layers (Porter and Duff, 1984).

We designed a loss function, $\mathcal{L}$, and minimized the overall loss at five viewpoints inside the aperture, V_{ref} of center, top, bottom, right, and left views,

$$\arg\min_{W} \sum_{v \in V_{ref}} (\mathcal{L}(\mathcal{R}^v(I^{\text{MPI}}), I^v_{\text{gt}}) + \mathcal{L}(\mathcal{R}^v(I'^{\text{MPI}}), I^v_{\text{gt}})), \tag{3}$$

where $\mathcal{R}^v(I^{\text{MPI}})$ and $\mathcal{R}^v(I'^{\text{MPI}})$ represent a rendered image from a viewpoint v with an MPI and from a viewpoint v with an MPI whose colors are replaced with those of the original focal stack input, respectively. I^v_{gt} represents ground-truth image from a viewpoint v.

$\mathcal{L}$ consists of the L1 loss $\mathcal{L}_{L1}$ and the perceptual loss with the backbone of VGG16 (Liu and Deng, 2015) $\mathcal{L}_{\text{Percept}}$ as

$$\mathcal{L} = \mathcal{L}_{\text{Percept}} + \lambda \mathcal{L}_{L1}, \tag{4}$$

with $\lambda = 0.1$ for our experiment.

3.5 View Synthesis

View synthesis, $\mathcal{R}^v(I'^{\text{MPI}})$, is available within the aperture range (i.e., the $A/2$ radius range from the center viewpoint) after generating an MPI.

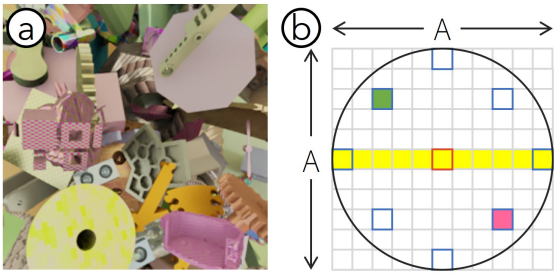

Figure 6: An example frame from the synthetic dataset and our synthetic aperture setup. (a) We place Thingi10K objects randomly to fill in the field of view. (b) We place 11×11 light field views depicted as grids. The red and blue rectangles represent views for testing. In particular, the blue rectangles filled in green and pink represent the reference viewpoints (green for top-left view, pink for bottom-right view) in Figure 9. The yellow band shows views used to create the Epipolar images in Figure 7.

4 EVALUATIONS

We evaluated how much the potential errors in two focal stack photography strategies affects the quality of generated MPI on both synthetic focal stacks and real focal stacks.

4.1 Dataset

We created a 11×11 light field dataset rendered using Blender (Community, 2018), from which we synthesize focal stacks using synthetic aperture photography (Vaish et al., 2004; Ishikawa et al., 2023). We relied on a synthetic dataset to simulate different errors independently and evaluate them quantitatively. The synthetic camera had 256×256 pixels with 56.2475°of the horizontal and vertical field of view. We took a similar way to that in (Xiao et al., 2018). Namely, each scene consists of randomly placed 3D objects from the Thingi10K dataset (Zhou and Jacobson, 2016). We created 120 scenes in total and separated them into 80, 20, and 20 scenes for training, validating, and testing the network, respectively.

We also prepared a real-scene dataset for qualitative evaluations. We used a DSLR (Canon EOS 6D equipped with SIGMA 50mm F1.4 DG HSM) to capture real-scene focal stack videos. During the capture, the f-number, image resolution, and frame rate were 1.8, 1920×1080 pixels, and 30fps, respectively. We photographed three scenes with $(d_{min}, d_{max}) \in \{(0.4, 0.5), (0.6, 1.2), (0.8, 1.5)\}$.

4.2 Metrics

We calculated the peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018) between the ground truth images and MPI renderings.

4.3 Training Details

We implemented our deep neural network using the PyTorch framework (Paszke et al., 2019) v2.1.0. To train our network, we used a desktop computer consisting of Intel(R) Xeon(R) W-3235 CPU @ 3.30GHz, 128GB RAM, and NVIDIA Titan RTX 24GB VRAM. To generate all the results in this paper, we used a desktop computer consisting of Intel(R) Core(TM) i7-6950X CPU @ 3.00GHz, 128GB RAM, and NVIDIA GeForce RTX 3080 10GB VRAM.

We trained our network using the RMSprop optimizer with a learning rate of 10^{-4}, a weight decay 10^{-8}, momentum 0.9, 4 batch size, and 614 epochs. We fixed the number of depth layers to $N = 32$, d_{min} to 1.0 meters, and d_{max} to 10.0 meters.

4.4 Designing Ring Rotation Errors

To quantitatively evaluate the impact of ring rotation errors on MPI rendering quality, we modeled the ideal rotations for both rotation strategies and those with potential errors associated with each strategy. The focus distances for all frames during a sweep from d_{min} to d_{max} were obtained through simulation. We then generated a synthetic focal stack by sampling N focus distances from the focus distances of all frames, which was used as input for the network. The first rows of Figure 7 show the simulated focus distances of all frames and sampled distances.

Continuous-rotation. With an ideal continuous rotation, the focus distance is expected to step linearly by $\delta v = 0.003$ in inverse depth per frame from the minimum focus distance d_{min} until it reaches the maximum focus distance d_{max}. For variations, we redefine the increment with a Gaussian noise as $\delta v = N(v_{mean}, v_{std})$, where $v_{mean} = 0.003$ and $v_{std} = \{0.001, 0.002, 0.003\}$. The increment never went below 0 m. We further introduce a pause at $i = \{50, 150, 250\}$ frame for early, middle, and late stops to simulate a scenario where the user stops rotating the ring for a short interval, $p = 50$ (i.e., the focus remains the same between i and $i + p$ frames). The ideal ring rotation ends at 300 frames (= 10 s).

Delta-rotation. The ideal delta rotation increases the focus distance by a step $\delta v = 0.003$ or 0.045 in inverse depth at every $s = 15$ frame from the minimum focus distance d_{min} until it reaches the maximum focus distance d_{max}. Similarly to the continuous rotation, we redefine the increment as a Gaussian noise, $\delta v = N(v_{mean}, v_{std})$, with $v_{mean} = \{0.003, 0.045\}$ and $v_{std} = \{v_{mean} \times 1/3, v_{mean} \times 2/3, v_{mean} \times 1\}$. The increment never went below 0 m.

4.5 Results

Synthetic-scene results. Figure 7 shows MPI rendering results in the synthetic dataset. Table 1 summarizes the metric values of all variants evaluated in the dataset. The continuous rotation exhibits fewer errors than the delta rotation, which takes quantized rotation steps and focus distances, especially when the step size is large, $v_{mean} = 0.045$, but as fast as the continuous rotation approach. Both approaches are affected more by the larger inserted noises.

The stops in the continuous rotation affected the quality regardless of the frame at which the pause was introduced. The rendering results from the center viewpoint (Figure 7(a), the fourth column) reveal that the quality of

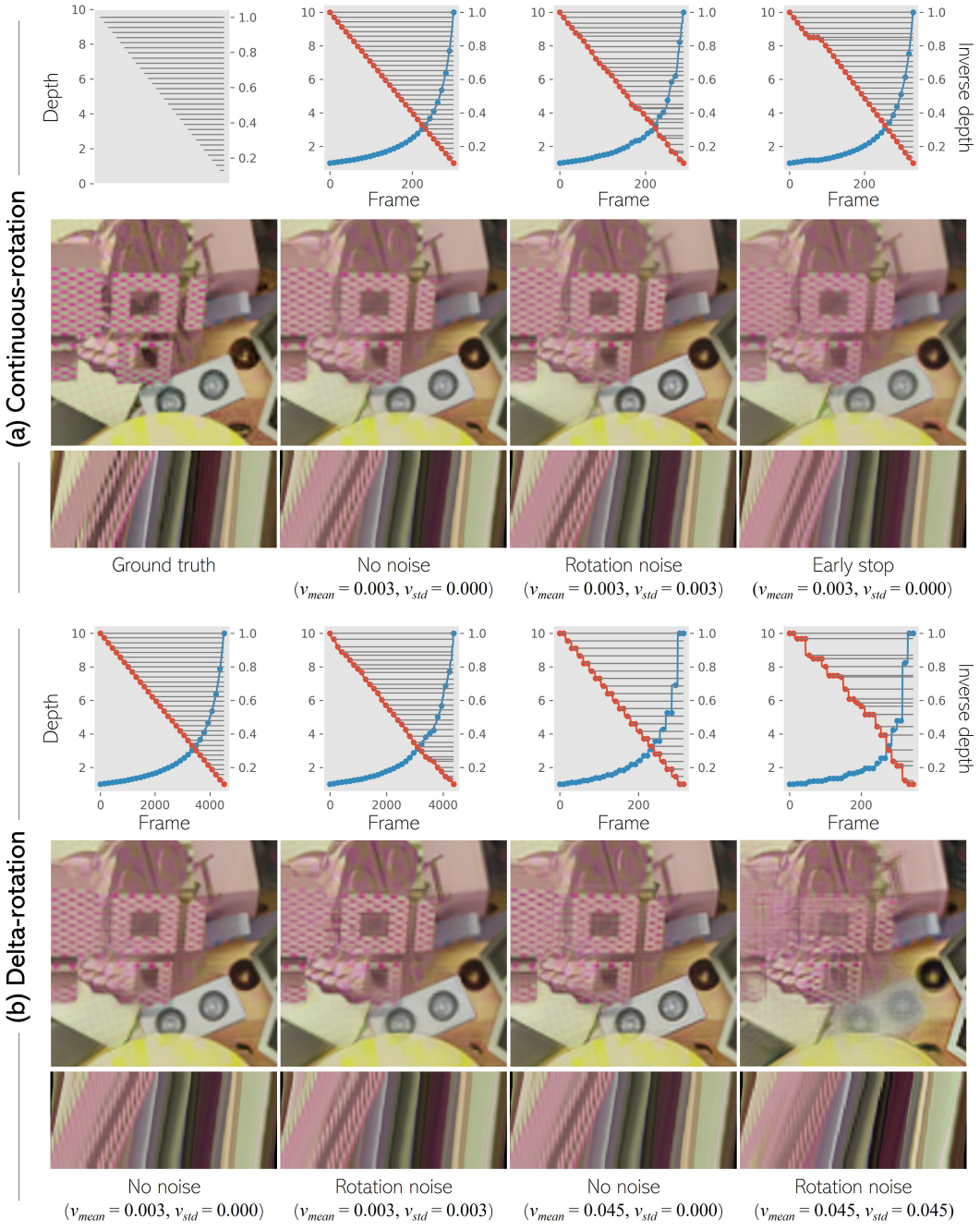

Figure 7: MPI rendering results under different rotation strategies. (a) The results of the continuous rotation. (b) The results of the delta rotation. (1st rows) Plots of focus distances. The red and blue plots show metric and inverse depth distances, respectively. (2nd rows) MPI rendering results from the center view. Cropped areas from Figure 6(a). (3rd rows) Epipolar images. 11 views on the horizontal line through the center view (the yellow rectangles in Figure 6(b)) were used.

the stop variants is inferior to those with rotation noises (Figure 7(a), the third column), exhibiting more blurry areas and low-contrast colors.

The delta rotation shows a similar quality to that of the continuous rotation if a long enough time is given to photograph (4000 frames as in Figure 7(b), the first column). However, the quality significantly decreases when the time to photograph is limited (300 frames as in Figure 7(b), the third column).

The Epipolar images represent disparities and show successful reconstructions of disparities with no noise and quality degradation under noise.

Table 1: Quantitative evaluation in PSNR, SSIM, and LPIPS over the MPI renderings at nine viewpoints shown as the red and blue rectangles in Figure 6(b). Mean and standard deviation values are calculated for reference.

	Focus ring modulation	v_{mean}	v_{std}	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Continuous-rot.	No noise	0.003	0.000	21.748 (2.856)	0.8422 (0.0406)	0.0992 (0.0269)
	Rotation noise	0.003	0.001	21.715 (2.839)	0.8385 (0.0413)	0.1006 (0.0272)
		0.003	0.002	21.609 (2.802)	0.8282 (0.0454)	0.1059 (0.0273)
		0.003	0.003	21.549 (2.716)	0.8221 (0.0439)	0.1116 (0.0270)
	Early stop	0.003	0.000	20.702 (2.870)	0.7372 (0.0822)	0.1230 (0.0305)
	Middle stop	0.003	0.000	21.026 (2.649)	0.7754 (0.0629)	0.1282 (0.0288)
	Late stop	0.003	0.000	19.733 (2.582)	0.6568 (0.0902)	0.1592 (0.0289)
Delta-rot.	No noise	0.003	0.000	21.741 (2.846)	0.8408 (0.0408)	0.0999 (0.0271)
	Rotation noise	0.003	0.001	21.679 (2.847)	0.8340 (0.0426)	0.1017 (0.0271)
		0.003	0.002	21.690 (2.787)	0.8378 (0.0406)	0.1040 (0.0267)
		0.003	0.003	21.398 (2.746)	0.8103 (0.0506)	0.1134 (0.0266)
	No noise	0.045	0.000	21.609 (2.718)	0.8269 (0.0419)	0.1088 (0.0268)
	Rotation noise	0.045	0.015	21.257 (2.815)	0.7897 (0.0533)	0.1191 (0.0270)
		0.045	0.030	20.410 (2.236)	0.7139 (0.0656)	0.1780 (0.0264)
		0.045	0.045	17.832 (2.557)	0.4662 (0.1369)	0.2221 (0.0315)

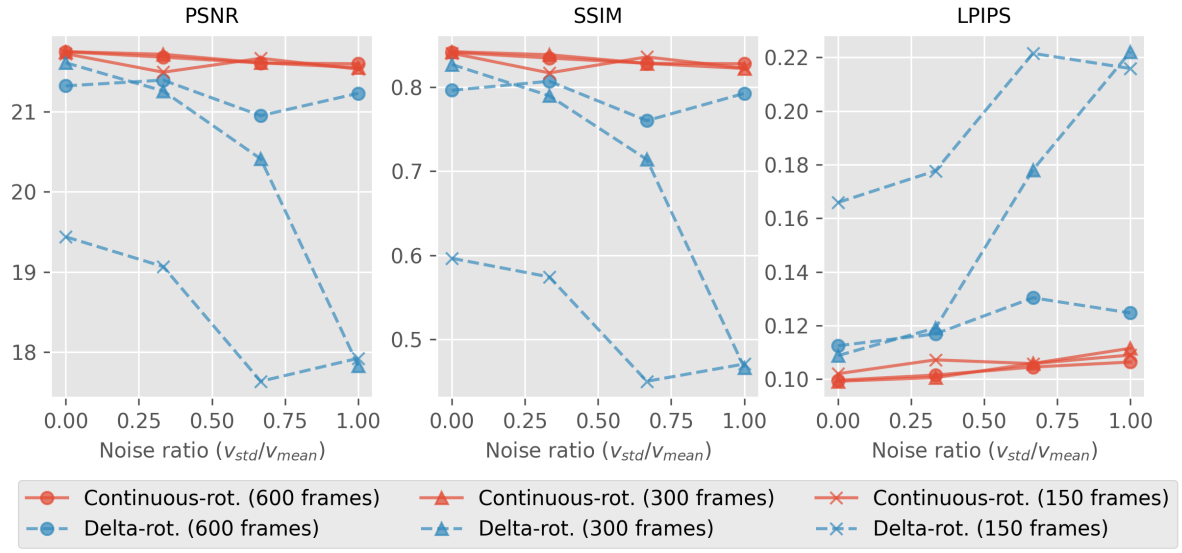

Figure 8: Varying noise ratios (v_{std} / v_{mean}) in PSNR, SSIM, and LPIPS for continuous rotation and delta rotation.

Figure 8 shows the comparisons of the changes in metrics over varying noise ratios (v_{std}/v_{mean}), which indicate the degree of noise over the step size. Each line in the continuous rotation (red line) has a different step mean of $v_{mean} = \{0.0015, 0.003, 0.006\}$, resulting in 600, 300, and 150 frames. Each line in the delta rotation (blue line) has the same number of frames as the continuous rotation with the same marker for $s = 15$ and $v_{mean} = \{0.0225, 0.045, 0.090\}$. Continuous rotations (red lines) are more stable than delta rotations (blue lines) for any v_{mean} and noise ratios.

Real-scene results. Figure 9 shows real-scene results. In this evaluation, the camera was fixed on a tripod. The setup was configured to ensure that d_{min} and d_{max} of rotation remained consistent for both continuous and delta rotations. For the delta rotation, we used a metronome to ensure uniformity in the duration of rotations.

Note that these experiments are based on simulated noises and do not explore actual usage by users. We plan to conduct a user study, but it remains our future work.

5 APPLICATIONS

We demonstrate applications utilizing resultant MPIs from real-scene focal stacks. The applications include depth rendering, occlusion-aware defocus filtering, and de-fencing.

Depth rendering. Since the alpha value of each layer represents the object certainties at the depth, assigning the depth value to each layer pixels results in free-viewpoint depth map rendering. Figure 10(a) demonstrates such a depth rendering and the original color rendering.

Figure 9: Real-scene MPI rendering results from top-left views (the green rectangle with blue outlines in in green in Figure 6) and bottom-right views (the pink rectangle with blue outlines in Figure 6) inside the lens aperture. We fixed the camera on a tripod. With this setup, the results are nearly identical.

Occlusion-aware defocusing. Since the layers are explicitly separated along the camera forward-facing axis, we can apply different blur kernels to different layers (i.e., depth-dependent blur kernels). This results in occlusion-

Figure 10: Three applications using MPI. (a) *Depth rendering*. Assigning depth values to individual layers, instead of colors (top), results in a depth rendering (bottom). (b) *Occlusion-aware defocusing*. Since the scene is decomposed into layers at depths, applying layer-dependent blur kernels can make occlusion-aware defocusing. The layers at the purple bear and the layers except for them are blurred on the left and right, respectively. (c) *De-fencing*. Removing frontal layers can remove objects that lay in the depths. One useful example is de-fencing, with which fences or lattices are removed. The computer internal components are more visible, and the RTX logo is more legible after de-fencing (bottom) than the original (top).

aware defocus blur without any special care such as edge-aware blur kernels. We validated two types of defocus: foreground defocus, which blurs out the near layers, and background defocus, which blurs out the far layers. The target layers were blurred with a Gaussian filter with a kernel size of 25 pixels. As demonstrated in Figure 10(b), the results show that even after applying the blur kernels the object boundaries are distinctly delineated.

De-fencing. Removing frontal objects is easy with MPI. Removing frontal layers can remove objects that lay in the depths. One useful example is de-fencing, with which fences or lattices occluding the backgrounds are removed. We took a focal stack from outside the computer to cover the depth of the computer. Though the computer internal components and the RTX logo were partially occluded by the cover, these became more visible after removing the frontal layers as shown in Figure 10.

6 CONCLUSION

This paper demonstrated a pipeline to generate MPI from a casually photographed focal stack. We proposed two possible focal stack photography strategies and quantitatively and qualitatively evaluated them using a synthetic dataset. We also demonstrated our system using real-scene focal stacks. Our results suggest that the continuous rotation is advantageous over the delta-rotation, especially when the time to photograph is limited. Finally, we showed three applications utilizing generated MPI.

One of the major limiting factors of our approach is the non-local image alignment in a focal stack that omits moving objects and strongly appearing disparities caused by camera shakes from the assumption. Another point is the validity of our simulated errors in focal stack photography. Therefore, future work includes user-involved studies where we collect various participants who actually control a camera focus ring. We are also interested in applying pixel-wise focal stack image alignment and designing an efficient network architecture for faster MPI inference.

ACKNOWLEDGEMENTS

This work was partly supported by JST SPRING, Grant Number JPMJSP2123, JSPS KAKENHI Grant Number JP23H03422, and the Austrian Science Fund FWF (grant no. P33634).

REFERENCES

Abuolaim, A., Delbracio, M., Kelly, D., Brown, M. S., and Milanfar, P. (2021). Learning to reduce defocus blur by realistically modeling dual-pixel data. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 2289–2298.

Baker, L., Mills, S., Zollmann, S., and Ventura, J. (2020). Casualstereo: Casual capture of stereo panoramas with spherical structure-from-motion. In *Proc. IEEE Virtual Reality (VR)*, pages 782–790. IEEE.

Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *Int. J. of Computer Vision (IJCV)*, 56:221–255.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. on Graphics (TOG)*, 28(3):24.

Bertel, T., Yuan, M., Lindroos, R., and Richardt, C. (2020). Omniphotos: casual 360°vr photography. *ACM Trans. on Graphics (TOG)*, 39(6):1–12.

Birklbauer, C. and Bimber, O. (2015). Active guidance for

light-field photography on smartphones. *Computers and Graphics (C&G)*, 53:127–135.

Broxton, M., Flynn, J., Overbeck, R., Erickson, D., Hedman, P., Duvall, M., Dourgarian, J., Busch, J., Whalen, M., and Debevec, P. (2020). Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4):86–1.

Community, B. O. (2018). *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

Davis, A., Levoy, M., and Durand, F. (2012). Unstructured light fields. In *Computer Graphics Forum*, volume 31, pages 305–314. Wiley Online Library.

Ebner, C., Mori, S., Mohr, P., Peng, Y., Schmalstieg, D., Wetzstein, G., and Kalkofen, D. (2022). Video see-through mixed reality with focus cues. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, 28(5):2256–2266.

Han, Y., Wang, R., and Yang, J. (2022). Single-view view synthesis in the wild with learned adaptive multiplane images. In *Proc. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 1–8.

Hedman, P., Alsisan, S., Szeliski, R., and Kopf, J. (2017). Casual 3d photography. *ACM Trans. on Graphics (TOG)*, 36(6):1–15.

Ishikawa, R., Saito, H., Kalkofen, D., and Mori, S. (2023). Multi-layer scene representation from composed focal stacks. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, 29(11):4719–4729.

Khakhulin, T., Korzhenkov, D., Solovev, P., Sterkin, G., Ardelean, A.-T., and Lempitsky, V. (2022). Stereo magnification with multi-layer images. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8687–8696.

Khan, N., Xiao, L., and Lanman, D. (2023). Tiled multiplane images for practical 3d photography. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 10454–10464.

Kim, H., Richardt, C., and Theobalt, C. (2016). Video depth-from-defocus. In *Proc. Int. Conf. on 3D Vision*, pages 370–379.

Kuthirummal, S., Nagahara, H., Zhou, C., and Nayar, S. K. (2011). Flexible depth of field photography. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1):58–71.

Liu, S. and Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. In *Proc. Asian Conf. on Pattern Recognition (ACPR)*, pages 730–734.

Mildenhall, B., Srinivasan, P. P., Ortiz-Cayon, R., Kalantari, N. K., Ramamoorthi, R., Ng, R., and Kar, A. (2019). Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. on Graphics (TOG)*.

Mohr, P., Mori, S., Langlotz, T., Thomas, B. H., Schmalstieg, D., and Kalkofen, D. (2020). *Mixed Reality Light Fields for Interactive Remote Assistance*, page 1–12. Association for Computing Machinery, New York, NY, USA.

Mori, S., Schmalstieg, D., and Kalkofen, D. (2023). Exemplar-based inpainting for 6dof virtual reality photos. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, 29(11):4644–4654.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pérez, F., Pérez, A., Rodríguez, M., and Magdaleno, E. (2016). Lightfield recovery from its focal stack. *Journal of Mathematical Imaging and Vision*, 56(3):573–590.

Porter, T. and Duff, T. (1984). Compositing digital images. In *Proc. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, SIGGRAPH '84, page 253–259, New York, NY, USA. Association for Computing Machinery.

Subbarao, M. and Choi, T. (1995). Accurate recovery of three-dimensional shape from image focus. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(3):266–274.

Szeliski, R. and Golland, P. (1999). Stereo matching with transparency and matting. *Int. J. of Computer Vision (IJCV)*, 32(1):45–61.

Takahashi, K., Kobayashi, Y., and Fujii, T. (2018). From focal stack to tensor light-field display. *IEEE Trans. on Image Processing*, 27(9):4571–4584.

Tomoto, Y., Rao, S., Bertel, T., Chande, K., Richardt, C., Holzer, S., and Ortiz-Cayon, R. (2020). Casual real-world vr using light fields. In *Proc. SIGGRAPH Asia 2020 Posters*, pages 1–2.

Vaish, V., Wilburn, B., Joshi, N., and Levoy, M. (2004). Using plane+ parallax for calibrating dense camera arrays. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE.

Wagner, D., Mulloni, A., Langlotz, T., and Schmalstieg, D. (2010). Real-time panoramic mapping and tracking on mobile phones. In *Proc. IEEE Virtual Reality (VR)*, pages 211–218. IEEE.

Wu, Z., Li, X., Peng, J., Lu, H., Cao, Z., and Zhong, W. (2022). Dof-nerf: Depth-of-field meets neural radiance fields. In *ACM Int. Conf. on Multimedia*, pages 1718–1729.

Xiao, L., Kaplanyan, A., Fix, A., Chapman, M., and Lanman, D. (2018). Deepfocus: Learned image synthesis for computational displays. 37(6).

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595.

Zhou, Q. and Jacobson, A. (2016). Thingi10k: A dataset of 10, 000 3d-printing models. *CoRR*, abs/1605.04797.