Video See-Through Mixed Reality with Focus Cues

Christoph Ebner, Shohei Mori, Peter Mohr, Yifan Peng, Dieter Schmalstieg, Gordon Wetzstein, Denis Kalkofen

Fig. 1. Examples of video see-through with focus cues. We introduce a gaze-contingent layered display driven by mixed reality focal stacks, which consist of captured and rendered images that have been focused behind and in front of the measured user's gaze. This enables diminishing the impact of imprecise and inaccurate eye tracking while supporting a large workspace in a small device form factor. Here we show the perceived image when focusing on (a) the back and (b) the front of a scene. The illustrations on the left and the right indicate the location of the MR focal stacks and the corresponding display planes.

Abstract— This work introduces the first approach to video see-through mixed reality with full support for focus cues. By combining the flexibility to adjust the focus distance found in varifocal designs with the robustness to eye-tracking error found in multifocal designs, our novel display architecture reliably delivers focus cues over a large workspace. In particular, we introduce gaze-contingent layered displays and mixed reality focal stacks, an efficient representation of mixed reality content that lends itself to fast processing for driving layered displays in real time. We thoroughly evaluate this approach by building a complete end-to-end pipeline for capture, render, and display of focus cues in video see-through displays that uses only off-the-shelf hardware and compute components.

Index Terms—Mixed reality, Video see-through, Focus cues

1 Introduction

Mixed reality (MR) offers interactive computer graphics within the user's physical environment. The potential of MR has been demonstrated in numerous applications [53]. With a head-worn display (HWD), MR experiences become mobile and hands-free, which is important to engage the user. Recent research has concentrated on optical see-through (OST) display technology, which allows seeing the real world in full detail in places where no augmentation exists [15]. However, OST displays still suffer from many shortcomings, including a limited field of view, lack of proper occlusion of physical objects, as well as poor color and contrast reproduction.

Since digital cameras have become more powerful, video see-through (VST) displays are able to provide a serious alternative to OST solutions. Using live video to represent the physical world in MR enables using opaque HWD designs based on virtual reality (VR) displays, which do not suffer from the restrictions of transparent optical elements. A VST HWD supports full control over light reaching the user, it enables a large field of view, and naturally supports transitions to VR [61] as well as sharing the captured environment with remote collaborators [38].

- *Christoph Ebner, Shohei Mori, Peter Mohr, Dieter Schmalstieg and Denis Kalkofen are with Graz University of Technology. E-mail: christoph.ebner@icg.tugraz.at*
- *Yifan Peng and Gordon Wetzstein are with Stanford University. E-mail: evanpeng, gordonwz@stanford.edu.*

Manuscript received 6 Sept. 2021; revised 3 Dec. 2021; accepted 7 Jan. 2022. Date of publication 15 Feb. 2022; date of current version 29 Mar. 2022. Digital Object Identifier no. 10.1109/TVCG.2022.3150504

Unfortunately, existing OST and VST HWD designs suffer from the vergence–accommodation conflict (VAC) [23]. With a display at a fixed focus distance, the user cannot accommodate to the depth where vergence is driven to by stereo disparity. Consequently, the VAC causes a blurred perception, resulting in eye strain and other symptoms of cybersickness [25]. Researchers from the optics and graphics communities have developed several display solutions for delivering natural defocus blur, including gaze-contingent [30], holographic [55], and multifocal [71] displays, which are commonly implemented using a layered setup [40]. Although these approaches have mitigated the VAC to some extent, none of them has considered the problem of delivering live video-based MR with focus cues for both virtual and physical elements.

In this work, we introduce the first approach to VST MR with full support for focus cues. To deliver high resolution within a compact form factor, we developed *gaze-contingent VST displays*, which adjusts the focus distance of the camera and the display simultaneously based on real-time measurements of the user's vergence.

Varifocal displays can present focus cues across a large workspace by shifting the display plane according to the measured vergence distance. However, this requires precise eye-tracking. In contrast, dual layer displays offer focus cues at multiple focus distances simultaneously over a small workspace without any need for eye-tracking. To compensate for inaccuracies of the eye-tracker, we introduce *gaze-contingent layered* VST displays, which capture a focal stack and adjust the focus distance of multiple display layers at once (illustrated in Fig. 1). Thus, our approach combines the advantages of varifocal and layered displays, thereby delivering high-quality focus cues over a large workspace for limited eye-tracking precision and accuracy.

Our display is driven by a real-time software pipeline built around *mixed reality focal stacks*, an efficient representation for capturing,

rendering, and display of MR with focus cues. All pipeline steps are designed to run in real-time on commodity GPUs. Frames of a focal stack video are captured by a high-speed camera. To enable fast focus tuning, we use a focus-tunable lens (FTL), an electrically addressable liquid lens. The captured image sequence is aligned into a focal stack using a novel real-time motion compensation technique that reuses the defocus blur of previous captured frames. The FTL timing is calibrated using a novel laser-based calibration routine. The captured focal stack is combined with a synthetically rendered focal stack. The focal stack rendering relies on a multi-plane image (MPI) representation [73] of multiple views. The combined virtual and real focal stack is decomposed into images for each of the display layers.

In summary, our approach significantly adds to the state of the art of MR displays. For the first time, focus cues are supported in MR on a VST HWD. Our work makes the following technical contributions:

- We introduce *gaze-contingent layered VST displays*, a novel MR display architecture that can deliver natural focus cues across large workspaces. Our design inherently enables the first varifocal VST display.
- We introduce *mixed reality focal stacks* for efficient capturing, processing, and presentation of MR content on layered displays in real-time frame rates.
- We present an *end-to-end pipeline* for capturing, rendering, and display of MR focal stacks that uses only off-the-shelf hardware and compute components.

Our prototype overcomes several limitations of previous VST displays, leaving as restrictions to visual fidelity mainly the resolution, dynamic range, and color gamut of the available hardware. We expect that cameras and displays will make rapid progress and any remaining fidelity gaps will be mitigated soon. Section 8 gives a detailed discussion.

2 Background

Designing a VST display with focus cues requires three key components: capturing the real environment with focus cues, rendering the MR scene with focus cues, and displaying the MR scene with focus cues on an HWD. We review the most relevant work in these areas in the following sections.

2.1 Capturing of focus cues

Light fields have often been captured for applications that require refocusing images to a given focus distance [17]. Thus, several approaches for capturing light fields have been proposed, including camera arrays [65, 68], lenslet arrays [42] and cameras with coded aperture [29,35,62]. However, since light fields consist of many views, they typically demand high bandwidth. Thus, systems that aim for real-time capturing either reduce the resolution per view or the number of views per light field. However, either reduction will decrease the quality of generated focus cues.

As an alternative to capturing the entire light field at once, one may adjust the optical components to capture focus cues for one focus distance at a time, for instance using a mechanical lens [58], a shifting image sensor [24], or an electrically tunable liquid lens [37]. A moving sensor or lens requires sensitive mechanics. Therefore, changing the shape of the lens is more practical for wearable systems.

However, when continuously capturing a focal stack of a dynamic scene, pixel motion may occur between consecutive focal stack images. To compensate for such motion, aligning corresponding pixels across images of a focal stack is necessary, typically via depth estimation, deblurring, pixel flow, and pixel remapping [20, 57]. Such existing approaches are promising but computationally expensive. Therefore, we develop a more efficient approach, which re-uses previously captured focal images and pixel flow between them to achieve real-time performance on a single GPU.

2.2 Rendering with focus cues

Traditionally, ray tracing [48] and multi-view rendering into an accumulation buffer [14] have been applied for rendering high-quality defocus blur. However, both approaches require many samples for high-quality results [41]. Therefore, real-time applications often apply post-processing in image space to a single rendering [51, 52] as a faster alternative. Unfortunately, the lack of information about occluded structures means that, in a single-view rendering, such approaches to defocus blur rendering commonly suffer from artifacts at occlusion boundaries [11]. Thus, Liu and Rokne [32] propose a technique that uses multiple renderings. They warp a sparse multi-view image into a layered depth image (LDI) [54], centered around the view direction. Our technique is inspired by this idea, but we replace the LDI with an MPI [73], which lends itself to high-performance filtering.

Rendering high-quality defocus blur in VR has also been approached using a neural network [67]. While this approach can possibly be extended to render real environments, mixing real and virtual blur has not been demonstrated in this context yet. Mandl et al. [34] use neural networks to apply characteristics of a camera to renderings. For generating depth of field (DOF) effects, they apply post-processing in image space. While this approach is able to support coherent rendering on monocular MR devices, for driving accommodation in head-mounted stereoscopic displays we aim for high-quality defocus blur, which includes considering information about occluded areas.

2.3 Near-eye displays with focus cues

Varifocal displays. Conventional displays can be extended with focus cues by continuously adjusting the virtual focus plane to match the user's vergence distance [21]. Several approaches have been proposed to drive the virtual focus plane, including mechanical setups [1, 43], deformable mirrors [13], as well as electrically tunable lenses [31, 50]. In all cases, precise eye-tracking is mandatory to correctly shift the virtual image plane. As pointed out by Dunn [12], current state-of-the-art methods for eye-tracking usually fail to provide the necessary accuracy. Despite promising attempts to improve the accuracy by adding additional depth sensors [44], precise depth-sensing remains challenging, especially in the presence of high-frequency depth variations and transparent or reflective surfaces.

Multifocal displays. Layered displays can naturally deliver focus cues at several distances in-between the display panels [63]. Existing approaches may be categorized by how they combine the pixels of multiple display panels. Additive displays often use a set of optical combiners to redirect the light rays originating from the display panels into the user's line of sight. Retinal [36, 40] and tomographic [28, 63] optimization methods are used to compute the image decomposition either from a light field or a focal stack as input [27, 39]. However, the limited light efficiency and the conic volume of rays emitting from off-the-shelf display panels prevent additive panels to be separated further than about 0.6diopter (D). Therefore, several panels and optical combiners have to be used, limiting current installations in common workspaces to large bench-top prototypes.

The alternative to spatially combining additive layers is to use time-multiplexing. Recent methods [9, 49] use high-speed projectors in conjunction with an FTL to generate several depth layers which are temporally fused into a volume by the human eye. However, since each layer is displayed only for a small amount of time, the time-multiplexed approaches commonly suffer from a low brightness.

To optimize the contrast in layered displays, Wu et al. [66] proposed adjusting the layers to the scene content. However, their approach requires a depth map of the scene, which may not be available for the real-world part of an MR environment. In comparison, measuring the user's focus shows several advantages. First, the display quality may adapt to the user rather than the scene. Second, fewer display layers are required, as layers can be shifted towards the inferred focus distance, rather than naïvely processing the entire scene.

Multiplicative (i.e., light attenuating) layered displays [63, 64] allow stacking display panels, and thus support a more compact form factor compared to additive layered displays [16]. However, diffraction of light limits the resolution that can be obtained in practice, and light efficiency is low.

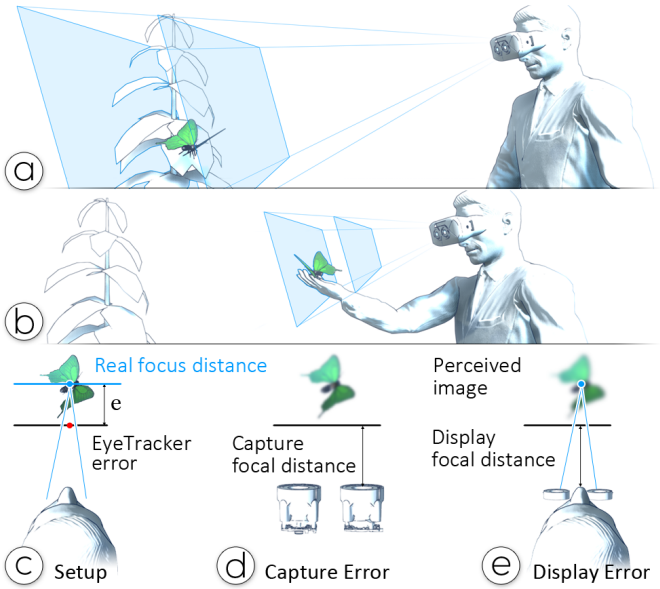

Fig. 2. Gaze-contingent layered display. Illustration of display panel location when focusing (a) to the back and (b) to the front. (c-e) Illustration of captured and displayed scene content in the event of erroneous eye-tracking data in single-layer varifocal displays. (c) The user is focusing on the butterfly, whereas inaccurate eye-tracking data causes an incorrect estimation of the gaze point (the red dot). (d) Scene content (the butterfly) appears blurry when captured or rendered with an erroneous focus distance. (e) Displaying the blurred rendering on a display plane that is offset from the actual gaze will further increase the perceived blur.

Holographic displays. Holographic displays make use of diffraction and naturally support focus cues. Although research in the optics and graphics communities has made remarkable progress over the last few years [7, 18, 45–47], holographic displays are still in their infancy. The fundamental limiting factor is the space–bandwidth product of available spatial light modulators [8], severely limiting the depth of field holographic near-eye displays can provide. Although emerging computational approaches, such as neural networks [47], can compensate for certain hardware limitations, today's holographic displays still suffer from severe issues, such as limited resolution, poor color fidelity, a compromised field of view, small eye-box size, bulky form factor, and high computational cost. To date, they cannot be used to build the kind of MR display we desire.

3 System Design

We present a complete *end-to-end* design for VST MR with focus cues. In contrast to prior displays that support focus cues, our system is built for VST and, thereby, consists of components for capturing, rendering, and display. Our approach is based on MR focal stacks (Fig. 3), which are composed of live video capturings and focal stack renderings. MR focal stacks can be presented on a layered display to support focusing at any distance in-between the display panels.

3.1 Display considerations

While we can achieve high-quality results with additive layered displays, their value is limited by the small working volume that can be encoded between two layers [33]. If we use a varifocal display instead, the limiting factor is the accuracy of the eye-tracking. State-of-the-art solutions can compute the user's gaze at an error of 0.5—1 ° [3, 22, 44], which translates to an offset of ±0.3-0.6D from the user's actual vergence distance. While this error is small enough to not jeopardize eye comfort [56], it does reduce the perceived contrast, diminishing the perceived quality (see Section 7.3 for examples and an evaluation).

Erroneous eye-tracking information is especially problematic in varifocal displays, as two undesirable effects will accumulate. The measured vergence distance will be used to synthesize (or capture) images with wrong depth of field, leading to blurry representations of in-focus objects. When the already blurry objects are shown on a screen that is moved to an erroneous vergence distance, an additional degradation of contrast will occur (see Fig. 2(c-e) for an illustration of this effect).

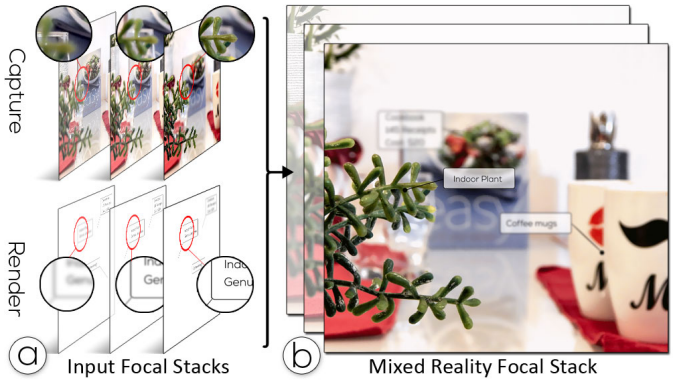

Fig. 3. Mixed reality focal stacks consist of (a) a captured focal stack, which is motion-compensated, and a focal stack rendering of the virtual scene. (b) The MR focal stack is generated by blending captured and rendered focal images and subsequently used to compute the display decomposition of a layered display.

We avoid these problems by introducing gaze-contingent layered displays, which inherit the large workspace of gaze-contingent approaches and the robustness against erroneous eye-tracking of multifocal approaches. We make use of an additive layered setup for building a high-resolution display. Our gaze-contingent approach overcomes the limited workspace of small numbers of additive layers in a compact form factor. Fig. 2(a-b) illustrates the position of the working volume for a near and a far gaze point. Extending the single display plane of varifocal approaches to a volume in which the user can naturally focus, diminishes the impact of erroneous eye-tracking.

3.2 Pipeline for mixed reality focal stacks

In order to deliver MR experiences, the display must not only be wearable but also present images with focus cues at real-time update rates. We meet this performance requirement by relying on an MR focal stack representation (Fig. 3(b)). A focal stack is sufficient for creating high-quality focus cues, but it is much less data-intensive than a full light field. Using a focal stack simplifies the video capturing since we can use temporal rather than spatial multiplexing. Even more importantly, the stack is small enough so that its processing runs on a single GPU.

Capturing (Section 4). We capture a focal stack using an off-the-shelf video camera through an FTL (Fig. 3(a), top). To ensure high frame rates, we continuously change the focal power of the FTL. To capture images at known focus distances, we calibrate the focal power of the lens to the shutter of the camera. Motion within a captured focal stack is computationally compensated.

Rendering (Section 5). We present a novel focal stack renderer (Fig. 3(a), bottom). Our renderer uses multi-view information for resolving occlusions and introduces a novel MPI filtering method to generate the focal stack at high frame rates. Furthermore, we develop a blending scheme that considers the mixture of focus cues from captured and rendered images. As an image of the captured focal stack provides blur corresponding to a specific focal length, we can easily blend the acquired image with the rendered focal stack.

Display processing (Section 6). Aiming at high in-focus contrast and high-quality defocus blur in a mobile form factor, we explore a combined gaze-contingent and additive layered display. The layers are combined via a conventional beam splitter, while gaze contingency is established with an FTL. The volumetric nature of the resulting multifocal display enables compensating for inaccurate and imprecise eye-tracking measurements in a form factor that is still compact.

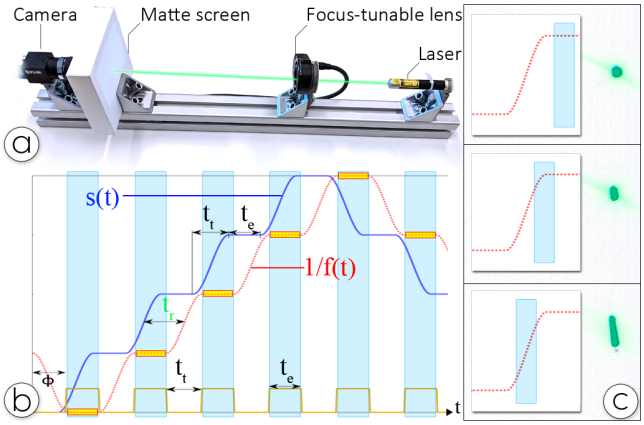

Fig. 4. Focal stack capture. (a) We calibrate the temporal offset between the point in time where the signal is sent and when the FTL is focused accordingly. This offset is calibrated offline using a green laser beam. (b) The blue plot indicates the lens signal, while the dotted red line indicates the lens response. Images of the focal stack are captured during the flat segments of the trigger signal (yellow bars). (c) During changes in focal length, the beam moves in image space and, therefore, appears as a line in the captured image. We adjust the offset between the point in time when the signal is sent and the point in time when the shutter is opened until the beam stops moving during exposure.

4 Capturing

To quickly capture images with different focal lengths, we use an FTL whose focal power depends on its input current. The prototype used throughout this paper relies on a stereo setup, where each channel consists of one FTL, one additional fixed-focus lens, and one camera.

The driving current of the FTL can either be set in software, by sending commands to the driver, or by providing the current directly, using an external signal generator. Since the quality of an MR experience relies on high update rates, we connect the FTL and the camera directly to a signal generator. As we must avoid FTL oscillations resulting from sudden changes of the focal power, the lens is driven with a signal $s(t)$ that smoothly changes from one constant electrical current to another (illustrated by the blue signal in Fig. 4(b)). The flat segments of the signal are modeled with Rect functions. The length of each Rect function matches the exposure time of the camera t_e. Smooth transitions between adjacent Rect functions are modeled as Sigmoid functions, which are scaled and shifted to connect neighboring Rect functions during the transition time t_t. See the blue line plot in Fig. 4(b) for an illustration of the resulting function $s(t)$. Note the offset between the driving signal $s(t)$ and the resulting focal length $f(t)$. The dotted red line represents the response time t_r of the FTL, which is the time between emitting a current and the lens settling at the corresponding focal power.

4.1 Camera and lens synchronization

Each image of the focal stack should be focused to one specific distance. To make sure the camera shutter is opened only when the focal power of the FTL is constant, we synchronize the shutter of the camera with the electric current that is sent to the FTL. Precise alignment of the camera shutter with the focal power of the lens (the dotted red line in Fig. 4(b–c)) is obtained through the calibration of the response time t_r. Measuring t_r is challenging because it is not explicitly known when the lens receives a new current or when it settles at a focal power. A common means is to constantly read out the focal power from the driver and derive the offset from the difference between the point in time when the current is sent to the lens and when the focal power reading stays constant. However, such an approach suffers from severe latency when polling in software.

Instead of measuring the response time t_r directly, we estimate the change in focal power within the captured image using a laser-based calibration approach. Specifically, we measure the length of the line that appears on a flat surface behind the FTL when a laser beam is sent through. Thus, we make use of the insight that the line will converge into a point when the lens stops changing its focal power. A photograph of our calibration setup is shown in Fig. 4(a).

The exposure time of the camera t_e is set to a time interval where the input current is constant. This enables changing the phase ϕ of the shutter signal until the focal power of the FTL is constant when the shutter is triggered. We find the offset t_r by shifting ϕ until the length of the line is minimized. The length is determined from the endpoints of an ellipse fitted to the captured image (Fig. 4(c)).

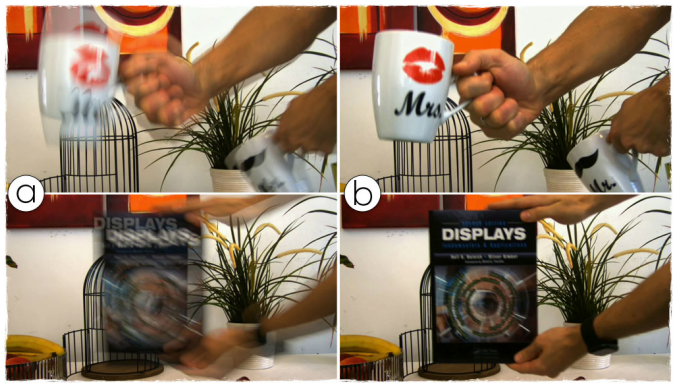

Fig. 5. Motion compensation. Two example focal stacks with seven images (a) without, and (b) with motion compensation.

4.2 Motion compensation

Cumulative capture of focal stack images can incur objects in motion, which would further lead to noticeable ghosting artifacts when the focal stack is displayed (Fig. 5(a)). Unfortunately, aligning moving pixels between focal stack images is complicated by the changes in focal lengths between images [2]. Existing approaches rely on a combination of several operations, such as depth estimation, synthetic de-blurring, and optical flow [20, 57, 59, 60]. These methods are unsuitable for real-time applications, as processing a single frame usually requires several seconds or even minutes using rather low resolution.

To provide faster update rates, we reuse the captured blur for compensating motions in the images of a focal stack. Our approach is capable of aligning structure in the images of a focal stack while providing a good compromise between speed and quality (Fig. 5(b)). Our approach consists of two steps. First, we determine how pixels move between frames. To this end, we calculate the pixel flow between the current and previously captured images. Second, we use the calculated flow maps to remap previously captured pixels to the current image, which results in a focal stack with aligned images,. Note that blur information is retained in the remapping step.

In the following, we denote previously captured images as $\mathscr{I}_i, i \in [0;F)$ and motion compensated images as $\mathscr{I}'_i, i \in [0;F)$, where F is the number of images in a focal stack.

1. Obtaining pixel flow maps. In order to obtain all necessary pixel flow maps, we first compute the pixel flow from the current captured image $\mathscr{I}'_i$ to its direct predecessor $\mathscr{I}_{i\pm1}$ (the index is either $i-1$ for near-to-far focus capturing or $i+1$ vice versa), which uses a slightly different focus distance, and to the predecessor $\mathscr{I}_i$, which uses the same focus distance (illustrated by the two solid black arrows in Fig. 5(c)). We specifically use these two images because of their similarity to the current captured frame: $\mathscr{I}_{i\pm1}$ is the closest image in time, i.e. the image has a very similar structure, and $\mathscr{I}_i$ is the last captured image with the same focus distance, i.e. this image is likely to contain similar blur information.

We use PatchMatch [4] to compute the flow maps since it is more stable in the presence of blur than conventional approach to computing optical flow [20]. To support a fast look-up during patch similarity calculation, we use a sum of squared differences (SSD) implementation via GPU jump flooding [70] with four neighbors. We use PatchMatch for computing the flow maps between $\mathscr{I}'_i$ to $\mathscr{I}_{i\pm1}$, and $\mathscr{I}'_i$ to $\mathscr{I}_i$. To obtain flow maps for the remaining images of the stack, we concatenate

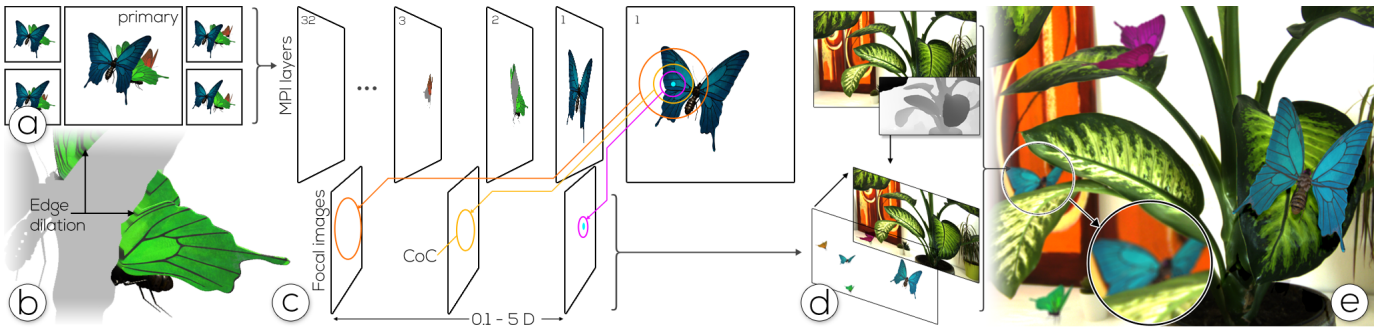

Fig. 6. Rendering MR focal stacks. (a) We leverage a multiplanar image (MPI) representation and fill it with multi-view information. (b) Subsequently, we dilate the edges to compensate for missing information. (c) We process the MPI from back to front. The contribution from each layer to all focal stack images is computed before proceeding to the next layer. We sort the CoCs that will appear in a layer to reuse pixels from a small CoC in a larger CoC. (d) We augment the captured focal stack images with rendered pixels. If the scene depth is available we furthermore resolve occlusions using depth sorting. (e) Thus, rendered pixels are used only where they occlude captured pixels.

previously estimated flow maps with the currently computed maps [59]. Note that flow estimations to images with the same focus distance can be omitted for images at maximal and minimal focus distance because such flow will not be used during focal image reconstruction.

2. Reconstructing focal images. Once the pixel flow is calculated, we use it to reconstruct a focal stack at the current point in time. We use the flow maps to remap pixels of previously captured frames so they are aligned with the current captured frame. The remaining difference between images of a stack is the blur due to different focus distances.

5 Rendering

The MR focal stack is composed of captured and rendered focal images. Therefore, we render virtual scene elements at focus distances that match the focus distances of the images in the captured focal stack. Rendering high-quality defocus blur is an essential requirement for presenting focus cues [36]. MR applications additionally require support for quick update cycles. Thus, we aim at real-time rendering of virtual content onto the captured focal stack, $\mathfrak{I}'$, so that defocus blur of virtual and real pixels provides high-quality focus cues. However, rendering blur at sufficient quality is expensive due to the large number of required samples [6] and increases for focal stacks, as we have to render several images at varying focus distances. Thus, our approach uses sparse multi-view rendering and inexpensive image dilations to resolve occlusions and a novel MPI filtering method to generate the focal stack. In Section 7 we show runtime and quality measures for several configurations.

5.1 Focal stack rendering

As blurred foreground objects might reveal structure that is occluded from a single point of view, we start by rendering a sparse light field, i.e., multiple views of the scene. For each view, we render color, alpha, and depth information. Per-pixel alpha enables us to use alpha compositing for merging the rendered pixels with the captured focal stack. Via the depth channel, the rendered views are transformed into a layered scene representation, which we implement using one MPI per focal stack. We partition the scene into layers that are equally spaced in diopters and parallel to the image plane of the camera used to capture the corresponding focal stack. We use a volume of 0.1-5D with 32 layers. Each layer in the MPI has the same resolution as an image in the captured focal stack.

1. MPI generation. We distinguish between a primary view and secondary views. The primary view is aligned with the center of the corresponding physical camera, e.g., the left primary view is aligned with the left camera on the HWD. All other views that contribute to an MPI are considered secondary views. The information in the primary view is copied into the corresponding MPI first. We use the rendered depth information to assign each pixel to its closest layer in the MPI. Subsequently, we warp the secondary views into the primary view and distribute their pixel information to the MPI layers. To prioritize information from the primary view, we add information from the secondary views only if a layer pixel is empty and occluded by at least one other pixel. See Fig. 6(a) for an illustration of five input views and the corresponding MPI.

2. Edge dilation. To account for missing information caused by the sparsity of the viewpoint sampling, we dilate edge information in each layer for occluded pixels. Thus, for every empty and occluded pixel, we run three iterations of a filter kernel averaging pixel colors and alpha values within a 5×5 neighborhood (Fig. 6(b)). The kernel size and the number of repetitions have been empirically evaluated on 1024×602 layer resolution, which corresponds to the processing resolution in our display prototype.

3. Focal image rendering. We add defocus information to the MPI by blurring each pixel, in each layer, with a circular kernel that corresponds to the circle of confusion (CoC) of the pixel. The size of the CoC is proportional to the difference between the distance of the layer (a) to the virtual camera and (b) to the focus distance of the image in diopter space.

A naïve method for rendering a focal stack from the MPI would adjust the defocus blur of each slice in the MPI stack individually. This rendering method must (1) identify the focus distance of the focal image to render, (2) generate the blur of all pixels in all layers, and (3) blend the layers of the MPI from back to front using the *over* operator. Such a naïve method redundantly computes the blur of each pixel in each image of the focal stack. However, the convolutions of the largest CoC already include pixels required for the convolutions with a smaller CoC. Hence, we can significantly increase the efficiency by reusing previously computed blur (see Section 7 for an evaluation).

We generate the focal stack by processing the MPI layers from back to front, computing, for each layer, its contribution to *all* focal stack images, before proceeding to the next layer. We start by determining the size of all CoCs of a layer with respect to the focal stack images. Note that the CoC is constant for a layer and a single focal image. We sort the estimated CoCs by size and start by blurring the layer with the smallest CoC. The result is written to the corresponding focal image, i.e., the image with the focus distance most similar to the distance of the layer.

Now, for the next CoC, we can reuse the pixels that have already been gathered, so we only have to include the additional pixels contained in the enlarged CoC. The result is written to the corresponding focal stack image, and the procedure repeats until the largest CoC has been processed. After processing all layers, the whole focal stack is processed. Fig. 6(c) illustrates our sorted blur computation for one layer of the MPI.

5.2 Mixing rendered and captured focal stacks

We mix the rendered $\mathfrak{I}_v$ and the captured $\mathfrak{I}'$ focal stacks into a target $\mathfrak{I}_{\mathrm{MR}}$. The captured defocus blur of $\mathfrak{I}'$ remains untouched; $\mathfrak{I}_v$ is alpha-blended into $\mathfrak{I}'$ using the opacity determined during the filtering process (Fig. 6(d–e)). To handle occlusions of real over virtual pixels, we apply phantom rendering [5] to generate a depth map. For pixels where the real depth is closer to the camera, we discard the rendered pixel and use the video pixel instead. Additionally, we add blur originating from

real objects, as it potentially spreads over the rendered scene elements. Hence, for each real pixel, we scatter its color according to its point spread function (PSF) and the current focal length across rendered pixels. However, as the blur from a real object only spreads to virtual objects behind the real pixel, we only update rendered pixels that have a larger depth than the one of the real source pixel. Fig. 6(e) shows how the green leaf occludes the blue butterfly.

6 Display

Our display combines varifocal and additive layered designs. Adaptively adjusting the placement of a multifocal display, instead of only a single image plane, allows us to cover a large workspace while tolerating some amount of error from eye-tracking.

6.1 Weighted multilayer decomposition

The effort required to drive multifocal displays can only be justified if high-quality images are generated at high update rates. Our choice of an MR focal stack representation is advantageous in that the focal stack images can be spatially related to the additive display layers. This allows us to weight the input focal images based on their distance to each display panel, so we can use fewer images with a higher impact on the result per display panel. The goal is to speed up the decomposition significantly while maintaining good results. The decomposition is usually formulated as an optimization problem [36, 67] of the form

$$x_k = \underset{x_k}{\operatorname{argmin}} \sum_{i}^{F-1} ||\mathscr{I}_i - \sum_{k}^{N-1} x_k * c_{i,k}||, \tag{1}$$

where $*$ denotes convolution; $\mathscr{I}_i \in \mathfrak{I}_{\mathrm{MR}}$, is the i^{th} image of the focal stack; $x_k, k \in [0;N)$, is the k^{th} layer of the display, and $c_{i,k}$ is the PSF of a pixel on x_k, when focusing to $\mathscr{I}_i$. Several approaches to resolving this problem have been previously investigated. We apply the image weighting to the iterative Simultaneous Algebraic Reconstruction Technique (SART), a common approach for solving Equation 1 [26, 69]. Conventional SART updates a current estimate using the rule

$$x_j^{(n+1)} = x_j^{(n)} + \frac{1}{FN} \sum_{i}^{F-1} \left(\mathscr{I}_i - \sum_{j}^{N-1} x_k^{(n)} * c_{i,k} \right) * c_{i,j}. \tag{2}$$

To introduce focal image weighting, we add $w_{i,j}$, which denotes the weight of focal image $\mathscr{I}_i$ when updating the display panel x_j. Specifically, we define the weight as

$$w_{i,j} = 1 - \min(|\frac{d}{f_i} - \frac{d}{p_j}|, 1), \tag{3}$$

where p_j is the location of the j^{th} layer, f_i is the focus distance, and d is the distance between two adjacent layers. Since computing the images of an additive display can be computationally intensive [40], we also seek to optimize the runtime of the decomposition. Similar to Mercier et al. [36], we pre-compute the convolved focal stack and the convolution of the kernels. Note that, even though the entire volume shifts according to the user's gaze, the relative distances between the input focal stack and the panels do not change. Thus, the kernel convolution is computed only once. This leads to an overall update rule of

$$\begin{aligned} x_j^{(n+1)} &= x_j^{(n)} + \frac{1}{FN} \left(\hat{\mathscr{I}}_j - \sum_{j}^{N-1} x_k^{(n)} * \tilde{c}_{k,j} \right), \\ \hat{\mathscr{I}}_j &= \sum_{i}^{F-1} w_{i,j} \cdot \mathscr{I}_i * c_{i,j} \quad \tilde{c}_{k,j} = \sum_{i}^{F-1} c_{i,k} * c_{i,j}. \end{aligned} \tag{4}$$

6.2 Prototype

We discuss our two-layer gaze-contingent display prototype and compare it against a conventional varifocal display. Our prototype consists of two 2.9 inch LCD panels per eye with a resolution of 1440×1440 each, and an update rate of 120Hz. The LCD panels are optically aligned and combined with a beam splitter, enabling users to observe both screens simultaneously in an additive manner. The virtual images are shifted with an FTL pair (Optotune EL-16-40-TC-VIS-20D). This configuration allows placing the center of the virtual 0.6D volume spanned by the screens anywhere between 0.67 to 4D. For eye-tracking, we use a PupilLabs [19] binocular eye-tracker, which is modified to fit into the housing.

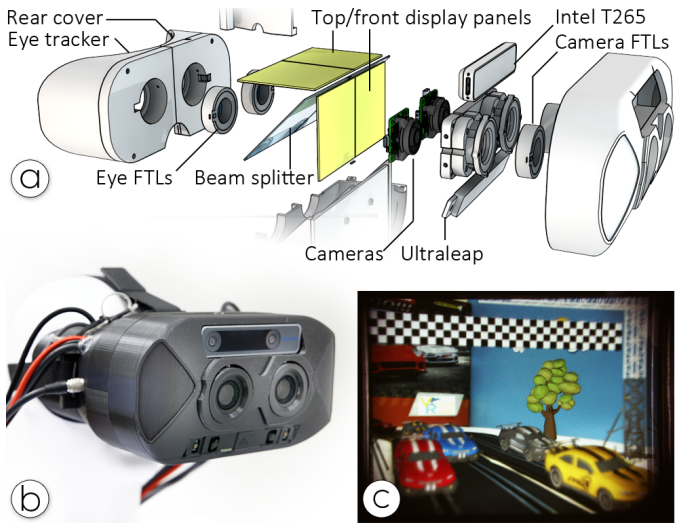

Fig. 7. Display prototype. We have implemented a two-panel configuration of our approach. (a) Our implementation uses focus-tunable lenses for adapting the focus planes of the cameras and the two displays, which we combine into the user's line of sight using an off-the-shelf 50/50 half-silvered mirror. (b) Our device is small enough to be wearable, while it provides high-quality images. (c) A photograph through the display.

For capturing, we use two board-level cameras (IDS UI-3861LE), each of which has a fixed-focus lens (Lensagon B5M6018) and an FTL (Optotune EL-16-40-TC-VIS-5D) attached. The cameras were set to an exposure time of 10ms, and a frame rate of 30Hz (limited by the software pipeline). The response time of the FTLs were measured to be 4ms. The focal stack captured by the camera can be shifted anywhere between 0D and 4D. The field of view is about $50 \times 30°$per eye. We use an Intel Realsense T265 for 6DOF head tracking and an Ultraleap Stereo IR 170 for hand tracking. All components of capture, tracking, and display are synchronized in software via the host computer. Fig. 7 shows (a) an explosion diagram and (b) a photo of our display, and (c) a view through the display.

7 Evaluation

To assess the performance of our system, we provide an evaluation and details about the configuration of individual components. Subsequently, we discuss its end-to-end performance and the configuration chosen to drive our prototype using current state-of-the-art hardware.

7.1 Motion compensation

To evaluate the resulting quality of our approach, we generated a dataset of synthetic focal stacks, consisting of over 134,640 frames in five scenes with two, three, five, and seven different focus distances, and with six different intervals between focal stack images. We tested our approach on an NVIDIA GeForce RTX 2080Ti GPU using a 9×9 pixels patch size and 16 jump steps for JFA. We measured the resulting image quality using *peak signal-to-noise ratio* (PSNR), *structured similarity image metric* (SSIM), and the *learned perceptual image patch similarity* (LPIPS) [72].

Fig. 8(a) shows the runtime measurements. It is evident that the major bottleneck is the iterative patch search. In a 2-focal image configuration, the system only searches the pixels in the previous frame during the flow map calculation, and therefore, it is the fastest regardless of pixel resolutions. Other configurations with more focal images require more time, depending on the number of images. However,

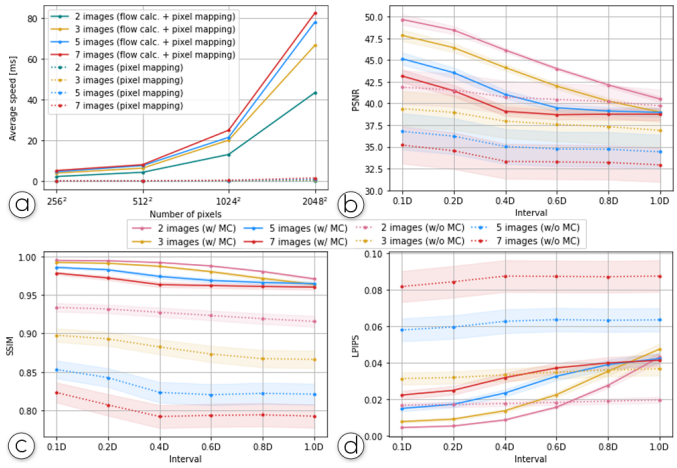

Fig. 8. Motion compensation evaluation. (a) Average processing time per focal stack, and (b) PSNR (c) SSIM, and (d) LPIPS per focal stack image. Standard deviations shown as semitransparent ribbons.

the relative difference between configurations decreases with higher numbers of focal stack images because every additional image causes only two additional flow calculation passes.

Fig. 8(b–d) show the results of per focal stack image quality in PSNR, SSIM, and LPIPS, respectively. The data shows that the quality has been improved in all conditions except for those with large intervals in LPIPS. The improvement becomes smaller as the intervals become larger, because the defocus differences become larger. We observe that for distances above approximately 0.6D only small improvements appear in the configuration with five images per stack. We also note that the standard deviation is smaller when applying the motion compensation, which demonstrates that our approach successfully suppresses inhomogeneities in the original focal stacks. The standard deviation has been scaled by 0.1 in all plots to visualize this trend.

Limitations. Our approach searches for captured structures in previously observed images to preserve the structure of the reference frame (i.e., the current frame) in several focal images. However, if objects do not exist or have been occluded in previous frames our approach can only find the best matching similar colors. This may cause artifacts when new structures with entirely different colors and texture appears. With more powerful hardware we could extend the search to more previously captured frames to resolve occlusions of structures that were visible before. Compensating for motion of entirely unseen structures remains a topic for future work.

7.2 Rendering

To gain insight into the performance of our rendering approach, we measure its quality and runtime and we compare it to a common approach to DOF rendering. Note that we do not compare to approaches for rendering blur in VR applications as they commonly do not provide mixed captured and rendered results. We test the approaches by rendering three MR scenes on an NVIDIA GeForce RTX 2080Ti GPU (Table 1). A screenshot from each scene is shown in Fig. 9. All three scenes show objects in the front and the back, providing large and fine structures.

To generate the ground truth for each scene, we render a focal stack with seven images of 1024×602px which are focused at equal distances in dioptric space within a range of 0.67 to 4D. The resolution has been chosen to match our prototype. Each focal stack image is rendered by aggregating 961 views into an offline accumulation buffer, where each view is rendered on top of a capturing with the same focus distance. We compare it to a common depth-dependent convolution (DDC) [11], where each pixel is blurred with a kernel corresponding to its CoC.

We evaluate our renderer in two different configurations. We render five views within the pupil, a primary view, and four corner views, as illustrated in Fig. 6, and we compare it to a stereo pair as input, using renderings at the pupil center at each eye. We compute PSNR,

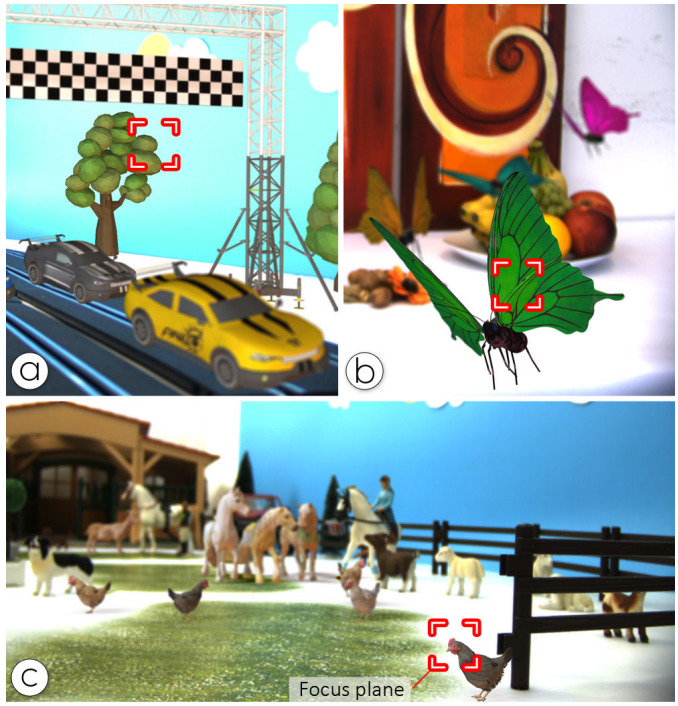

Fig. 9. Mixed reality focal stack images from the three scenes used to evaluate the rendering. We added a red box to indicate the focus plane. (a) Cars, finish line construction, and trees are virtual. (b) Butterflies are virtual. (c) All hens, the three horses in the center, and the brown goat who stands next to a white horse in the back are virtual.

Table 1. Render evaluation. The performance has been measured from rendering focal stacks which show the scenes depicted in Fig. 9. Runtimes consists of all-in-focus + without/with sorted CoC filtering.

		DDC	Five views	Stereo
Runtime ms	Scene A	1.2 + 15.4	6 + 22.4/**2.4**	2.4 + 22.9/**2.4**
	Scene B	1.7+ 9.4	8.5 + 12.3/**1.6**	3.4 + 8.8/2.1
	Scene C	2.2 + 11.2	10.5 + 8.5/**1.3**	4.0 + 12.9/**1.3**
PSNR dB	Scene A	25.27	**32.35**	32.12
	Scene B	29.95	**35.39**	35.33
	Scene C	35.27	**42.45**	**42.45**
SSIM	Scene A	0.889	**0.966**	0.965
	Scene B	0.958	**0.988**	**0.988**
	Scene C	0.984	**0.996**	**0.996**
LPIPS	Scene A	0.1263	0.0323	**0.0321**
	Scene B	0.0614	0.0134	**0.0133**
	Scene C	0.0333	**0.0053**	**0.0053**

SSIM, and LPIPS values using average values across all images in the focal stack (Table 1). The runtime measurement contains three values. The first value shows the time to render the all-in-focus input, and, following the + sign, two values showing the render times to generate the entire focal stack without and with our approach of sorted CoC filtering. Note that the DDC approach does not support sorted CoC filtering, and thus, shows only one value.

The results in Table 1 show that our approach outperforms DDC in terms of quality and runtime (better quality and faster processing is marked in bold font). The results also demonstrate the effectiveness of the sorted CoC filtering. Runtimes improve by a factor of 7.9 on average, compared to sequentially processing each focal image.

The stereo configuration and the configuration with five views lead to very similar qualitative results. We believe this is the case because the additional views in both configurations reveal similar occluded

structures. A thorough evaluation using a larger scene database is planned as future work to discern which configuration leads to superior results in which scene. However, differences in runtime between stereo and the configuration with five views are larger, due to the number of additional images that need to be generated for five input views. Since an all-in-focus stereo pair is always rendered to provide images for both eyes, we use the stereo pair in our current prototype configuration.

7.3 Display

Decomposition. We start with an evaluation of the weighted decomposition. We assess PSNR and SSIM of the weighted SART compared to those of a non-weighted version, subject to the number of iterations, for a two-layer setup with 0.6D volume size and an eye-tracker error of 0.3D (see Fig. 10). The results show that the approach for weighting reaches a higher quality than the non-weighted SART. It effectively reaches the quality of the converged non-weighted solution already after approximately four iterations. We believe this quality gain is owed to only a subset of focal images being taken into account when computing the decomposition of a certain panel. Essentially, the weighting method takes into account the reduction of a single pixel's contribution with increasing blur kernel sizes.

Runtime. Table 2 shows the performance of our implementation for display configurations using two and three panels spaced to span a volume of 0.6D and 1.2D, respectively. In each configuration, we use a dense focal stack with images spaced at 0.1D distance. Each image has a resolution of 1024×602 to match the processing resolution in our prototype. Runtimes are measured using an NVIDIA GeForce RTX 2080Ti GPU.

The measurements indicate that larger volumes require more time for precomputing $\hat{\mathscr{J}}_j$ than smaller volumes. This is caused by processing more focal images within the stack. In addition, when a higher number of focal images is used in the decomposition step, the size of the kernel $\tilde{c}_{k,j}$ is increased and, thus, the computation time increases. We believe this is why the non-weighted decomposition requires more time per iteration. Here, each panel is updated with images of the entire focal stack, leading to large kernels $\tilde{c}_{k,j}$.

Image quality. We evaluate the effectiveness of gaze-contingent layered displays. For this purpose, we compute the perceived contrast and image similarity to ground truth measured by PSNR, SSIM, and LPIPS, while gradually decreasing the accuracy of the eye-tracker. We assess the quality for a conventional varifocal display and a layered display comprising two display panels spaced 0.6D apart, and to two displays with three panels, which span over a volume of 0.6D and 1.2D. Eye-tracker errors are simulated up to 0.6D.

Fig. 11(a) shows results from simulating the perceived contrast on a varifocal display and several configurations of gaze-contingent layered displays. Each value represents the mean perceived contrast of gratings between 1-20 cpd. The left plot shows perceived contrast for different vergence offsets, while the right plot shows the average contrast in the range defined by the eye-tracking error. For example, with an eye-tracking error of 0.3D, the erroneous vergence distance might be anywhere within $\pm$0.3D to the true vergence distance. Thus, for an eye-tracking error of 0.3D, we measure the average contrast within a volume $\pm$0.3D around the gratings.

The results show that the perceived contrast of the varifocal display decreases steadily, as the distance to the panel increases, while the gaze-contingent layered displays maintain high contrast values for larger error values. Contrast is worst in the middle between two layers, and, best when focusing directly on the panel. The three-layer setup outperforms all other display configurations within the 1.2D volume for vergence offsets $\geq$ 0.5D, but the average contrast for eye-tracking errors of up to 0.5D is still equal or higher in the three-layer setup with 0.6D volume. The two-layer setup has low average contrast for small eye-tracking errors. However, for current eye-tracking solutions, it remains a feasible option, as the angular accuracy of currently available eye-trackers translates to an offset of up to 0.3D [12].

Fig. 12 provides a qualitative comparison to a conventional varifocal display design in the event of eye-tracking errors. Due to erroneous

Table 2. We assess the runtime of weighted SART and compare it against the uniform weighting of conventional SART.

		2 layers 0.6D	3 layers 0.6D	3 layers 1.2D
Weighted	Precomputation	0.52 ms	0.73 ms	1.32 ms
	Single iteration	0.54 ms	0.65 ms	0.76 ms
Non-Weighted	Precomputation	0.54 ms	0.77 ms	1.35 ms
	Single iteration	0.58 ms	0.85 ms	1.34 ms

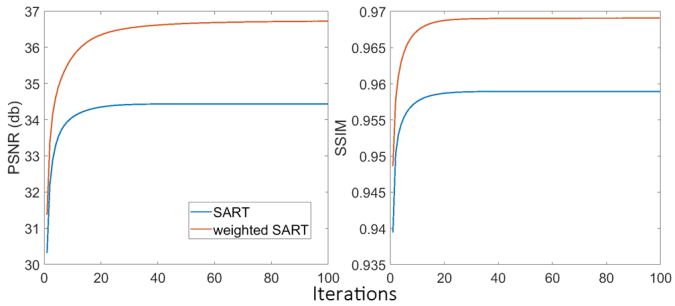

Fig. 10. Qualitative evaluation of weighted multilayer decomposition. We show PSNR and SSIM metrics over the number of iterations for the conventional SART and our approach to weighted SART.

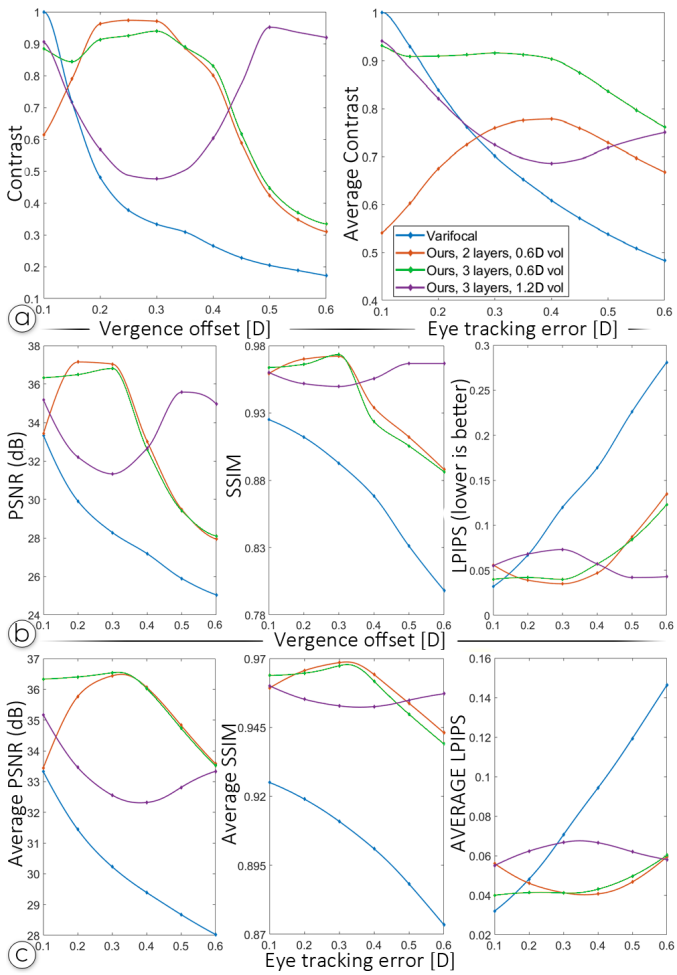

Fig. 11. We evaluate the quality for different vergence offsets and eye tracking errors in several display configurations. For each eye tracking error, we also compute the average quality over all vergence offsets within the $\pm$ error range. We measure (a) perceived contrast, and (b–c) PSNR, SSIM, and LPIPS values compared to ground-truth focus distance.

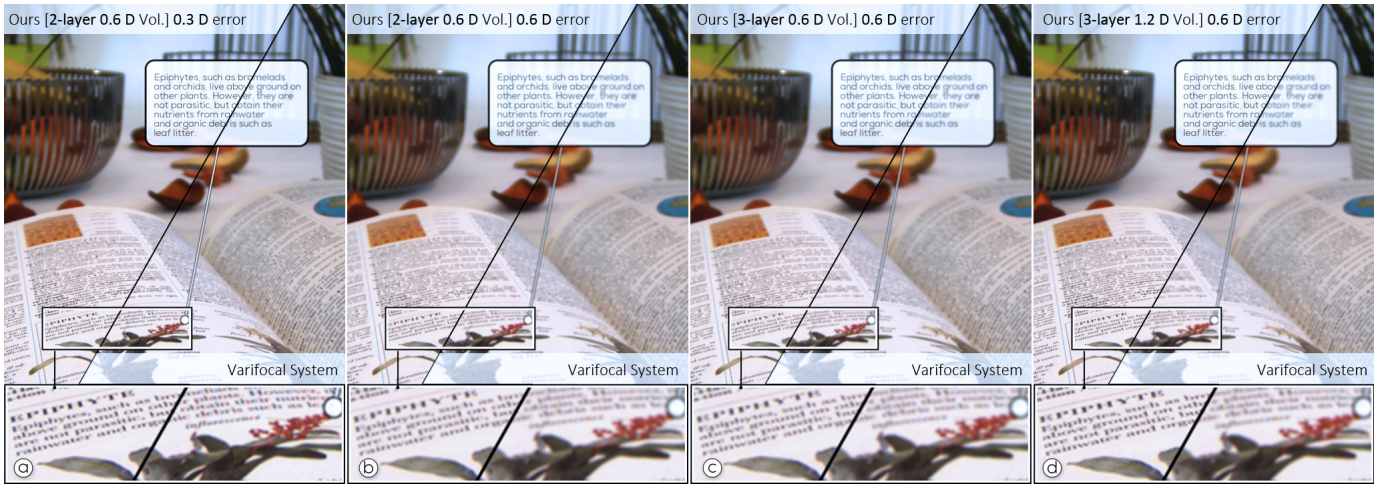

Fig. 12. Comparison to varifocal displays in the event of erroneous eye-tracking. For each pair, the image on the right shows the result achieved with a conventional single-layer display. Results of each pair are obtained with the same focus distance and gaze error. Note that there is significantly larger blur in all configurations of the counterpart varifocal display. We vary the vergence offset, number of display panels, and spacing between panels. Specifically, we apply (a) an error of 0.3D, and 2 panels spaced 0.6D apart. (b) error of 0.6D, and 2 panels spaced 0.6D apart. (c) error of 0.6D, and 3 panels spaced 0.3D apart, spanning a volume of 0.6D. (d) error of 0.6D, and 3 panels, spaced 0.6D apart, spanning a volume of 1.2D.

eye tracking, only blurry representations of the augmented label and the highlighted text in the book are perceived in the varifocal display. In contrast, our gaze-contingent layered design retains more crisp results, even for higher eye-tracking errors. To verify these findings, we perform a quantitative analysis, comparing our results to those of the varifocal display in PSNR, SSIM, and LPIPS metrics for eye-tracker errors from 0.1 to 0.6D. In Fig. 11(b-c)), the top row shows quality metrics for a certain vergence offset, while the bottom row shows an average of these metrics for a certain eye-tracking error. Similar to the average contrast, mean values for PSNR, SSIM, and LPIPS are calculated from measurements within the volume that is bounded by adding the error in front of and behind the ground-truth distance.

The quality improves with decreasing distance to a panel because blur kernels get smaller. The setup with three layers within a 1.2D volume performs exceptionally well for vergence offsets of about 0.6D. In terms of average quality with a 0.6D eye-tracking error, the three-layer setup with 0.6D volume and even the two-layer setup perform equally well regarding PSNR and LPIPS. However, they outperform the three-layer setup with 1.2D volume for smaller eye-tracking errors of about 0.3D. The three-layer setup with 0.6D increases the quality compared to the two-layer setup for very small eye-tracking errors. The quality of the conventional varifocal display decreases steadily with increasing eye-tracker error and matches the quality of gaze-contingent layered displays only for minimal eye-tracking errors. These results confirm the viability of gaze-contingent layered displays. Notably, the two-layer setup seems to exhibit a good trade-off between hardware effort and achievable quality.

7.4 Discussion

The performance of the proposed system depends on many variables. Finding a good compromise between render time and quality is a key challenge in building real-time MR systems. Our prototype is reconstructing a 2-layer 0.6D volume using a focal stack of 7 images. In this configuration, we achieve an update rate of approximately 30ms per eye for a processing resolution of 1024×602px on an NVIDIA GeForce RTX 2080Ti GPU, i.e. we use one GPU per eye in our system. The latency between capturing a frame and the subsequent screen update is mainly affected by the exposure time of the camera and the runtime of the motion compensation. Focal stack rendering and capturing has been implemented in parallel, i.e. while capturing an image of the focal stack we render the corresponding image using the same focal distance. Decreasing the latency between capture and display introduced by the individual components of the processing pipeline is desired, especially in scenarios, where movements of the user's own body is displayed. While the latency introduced by the motion compensation can be decreased with more powerful GPUs, decreasing the exposure time of the cameras affects the image quality of the captured focal stack.

Note that increasing the volume, or capturing a more dense focal stack does not affect capturing exposure time of our system. Instead, the whole stack can be captured over a longer time period where intra-stack motion is accounted for by our motion compensation approach. However, increasing the number of focal images has a performance impact. While the DOF rendering approach and decomposition remains largely invariant to the number of focal images, the motion compensation performance decreases as the number of focal images increases. With this in mind, our design choice of having 7 images per stack within a volume of 0.6D provides real-time interaction while keeping the focal stack dense, which leads to high contrast results.

8 Conclusion and Future Work

We introduce the design and evaluation of gaze-contingent layered displays for robustly supporting focal cues in video see-through MR displays. Our evaluations show that gaze-contingent layered displays can reliably compensate for erroneous eye-tracking of approximately 1°, which translates to an error of the estimated vergence distance of up to 1.2D. As a reference, research prototypes [3, 22] and commercial products [1,2] report an accuracy of 0.5–1°. Extending the volume which is spanned by the display layers is conceptually simple by adding more display panels at the cost of more data processing and a bulkier form factor. However, given the accuracy of available eye-tracking solutions, we believe that two or three panels are sufficient for most setups.

Though we were able to demonstrate the benefits of our design, we see several directions for future work. For example, we can improve the prototype by utilizing higher-quality components, such as faster displays and cameras with higher dynamic range and resolution. In addition, providing physiologically correct chromatic aberrations [10] at varying focal distances in video see-through MR displays remains a compelling topic for future research.

Acknowledgments

This work was enabled by the Austrian Science Fund FWF (grant no. P30694) and the Competence Center VRVis, which is funded by BMK, BMDW, Styria, SFG, Tyrol and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (879730) which is managed by FFG.

[1] https://vr.tobii.com/integrations/htc-vive-pro-eye/
[2] https://pupil-labs.com/products/vr-ar/tech-specs/

REFERENCES

[1] K. Akşit, W. Lopes, J. Kim, P. Shirley, and D. Luebke. Near-eye varifocal augmented reality display using see-through screens. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.

[2] E. Alexander, Q. Guo, S. Koppal, S. Gortler, and T. Zickler. Focal flow: Measuring distance and velocity with defocus and differential motion. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 667–682. Springer, 2016.

[3] A. N. Angelopoulos, J. N. Martel, A. P. Kohli, J. Conradt, and G. Wetzstein. Event-based near-eye gaze tracking beyond 10,000 Hz. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 27(5):2577–2586, 2021. doi: 10.1109/TVCG.2021.3067784

[4] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3), 2009.

[5] D. E. Breen, R. T. Whitaker, E. Rose, and M. Tuceryan. Interactive occlusion and automatic object placement for augmented reality. *Computer Graphics Forum*, 15(3):11–22, 1996.

[6] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Proc. Conference on Computer Graphics and Interactive Techniques (Siggraph)*, pp. 307–318, 2000.

[7] P. Chakravarthula, Y. Peng, J. Kollin, H. Fuchs, and F. Heide. Wirtinger holography for near-eye displays. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[8] C. Chang, K. Bang, G. Wetzstein, B. Lee, and L. Gao. Toward the next-generation vr/ar optics: a review of holographic near-eye displays from a human-centric perspective. *Optica*, 7(11):1563–1578, 2020.

[9] J.-H. R. Chang, B. V. K. V. Kumar, and A. C. Sankaranarayanan. Towards multifocal displays with dense focal stacks. *ACM Transactions on Graphics (TOG)*, 37(6), 2018. doi: 10.1145/3272127.3275015

[10] S. A. Cholewiak, G. D. Love, P. P. Srinivasan, R. Ng, and M. S. Banks. ChromaBlur: Rendering chromatic eye aberration improves accommodation and realism. *ACM Transactions on Graphics (TOG)*, 36(6), 2017.

[11] J. Demers. Depth of field: A survey of techniques. In *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics*, chap. 23, pp. 375–390. Addison-Wesley Professional, 2004.

[12] D. Dunn. Required accuracy of gaze tracking for varifocal displays. In *Proc. IEEE Virtual Reality (VR)*, pp. 1838–1842, 2019.

[13] D. Dunn, C. Tippets, K. Torell, P. Kellnhofer, K. Akşit, P. Didyk, K. Myszkowski, D. Luebke, and H. Fuchs. Wide field of view varifocal near-eye display using see-through deformable membrane mirrors. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 23(4):1322–1331, 2017.

[14] P. Haeberli and K. Akeley. The accumulation buffer: Hardware support for high-quality rendering. In *Proc. Conference on Computer Graphics and Interactive Techniques (Siggraph)*, pp. 309–318, 1990. doi: 10.1145/97879.97913

[15] R. R. Hainich and O. Bimber. *Displays: Fundamentals & Applications*. AK Peters/CRC Press, 2017.

[16] F. Huang, K. Chen, and G. Wetzstein. The light field stereoscope: Immersive computer graphics via factored near-eye light field displays with focus cues. *ACM Transactions on Graphics (TOG)*, 34(4), 2015.

[17] A. Isaksen, L. McMillan, and S. J. Gortler. Dynamically reparameterized light fields. In *Proc. Conference on Computer Graphics and Interactive Techniques (Siggraph)*, pp. 297–306, 2000.

[18] C. Jang, K. Bang, G. Li, and B. Lee. Holographic near-eye display with expanded eye-box. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

[19] M. Kassner, W. Patera, and A. Bulling. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Adjunct Proc. ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*, pp. 1151–1160, 2014.

[20] H. Kim, C. Richardt, and C. Theobalt. Video depth-from-defocus. In *Proc. International Conference on 3D Vision*, pp. 370–379, 2016.

[21] J. Kim, Y. Jeong, M. Stengel, K. Akşit, R. Albert, B. Boudaoud, T. Greer, J. Kim, W. Lopes, Z. Majercik, P. Shirley, J. Spjut, M. McGuire, and D. Luebke. Foveated ar: Dynamically-foveated augmented reality display. *ACM Transactions on Graphics (TOG)*, 38(4), 2019.

[22] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke. Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation. In *Proc. ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 1–12, 2019.

[23] F. Kooi and A. Toet. Visual comfort of binocular and 3D displays. *Displays*, 25(2-3):99–108, 2004.

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

[25] J. J. LaViola. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32(1):47–56, 2000. doi: 10.1145/333329.333344

[26] S. Lee, J. Cho, B. Lee, Y. Jo, C. Jang, D. Kim, and B. Lee. Foveated retinal optimization for see-through near-eye multi-layer displays. *IEEE Access*, 6:2170–2180, 2018.

[27] S. Lee, C. Jang, S. Moon, J. Cho, and B. Lee. Additive light field displays: Realization of augmented reality with holographic optical elements. *ACM Transactions on Graphics (TOG)*, 35(4):60, 2016.

[28] S. Lee, Y. Jo, D. Yoo, J. Cho, D. Lee, and B. Lee. Tomographic near-eye displays. *Nature Communications*, 10:2497, 2019.

[29] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen. Programmable aperture photography: Multiplexed light field acquisition. *ACM Transactions on Graphics (TOG)*, 27(3):55:1–55:10, 2008.

[30] S. Liu, D. Cheng, and H. Hua. An optical see-through head mounted display with addressable focal planes. In *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 33–42, 2008. doi: 10.1109/ISMAR.2008.4637321

[31] S. Liu, D. Cheng, and H. Hua. An optical see-through head mounted display with addressable focal planes. In *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 33–42, 2008. doi: 10.1109/ISMAR.2008.4637321

[32] X. Liu and J. G. Rokne. Depth of field synthesis from sparse views. *Computers and Graphics*, 55:21–32, 2016. doi: 10.1016/j.cag.2015.10.015

[33] K. J. MacKenzie, R. A. Dickson, and S. J. Watt. Vergence and accommodation to multiple-image-plane stereoscopic displays: 'Real world' responses with practical image-plane separations? In *Stereoscopic Displays and Applications XXII*, vol. 7863, pp. 363 – 373. SPIE, 2011.

[34] D. Mandl, P. M. Roth, T. Langlotz, C. Ebner, S. Mori, S. Zollmann, P. Mohr, and D. Kalkofen. Neural cameras: Learning camera characteristics for coherent mixed reality rendering. In *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 508–516, 2021. doi: 10.1109/ISMAR52148.2021.00068

[35] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar. Compressive light field photography using overcomplete dictionaries and optimized projections. *ACM Transactions on Graphics (TOG)*, 32(4), 2013.

[36] O. Mercier, Y. Sulai, K. Mackenzie, M. Zannoli, J. Hillis, D. Nowrouzezahrai, and D. Lanman. Fast gaze-contingent optimal decompositions for multifocal displays. *ACM Transactions on Graphics (TOG)*, 36(6):237, 2017.

[37] D. Miau, O. Cossairt, and S. K. Nayar. Focal sweep videography with deformable optics. In *Proc. IEEE Int. Conference on Computational Photography (ICCP)*, pp. 1–8, 2013.

[38] P. Mohr, S. Mori, T. Langlotz, B. H. Thomas, D. Schmalstieg, and D. Kalkofen. Mixed reality light fields for interactive remote assistance. In *Proc. ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 1–12, 2020. doi: 10.1145/3313831.3376289

[39] S. Moon, C. Lee, D. Lee, C. Jang, and B. Lee. Layered display with accommodation cue using scattering polarizers. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1223–1231, 2017. doi: 10.1109/JSTSP.2017.2738614

[40] R. Narain, R. A. Albert, A. Bulbul, G. J. Ward, M. S. Banks, and J. F. O'Brien. Optimal presentation of imagery with focus cues on multi-plane displays. *ACM Transactions on Graphics (TOG)*, 34(4):59, 2015.

[41] R. Ng. Fourier slice photography. *ACM Transactions on Graphics (TOG)*, 24(3):735—744, 2005. doi: 10.1145/1073204.1073256

[42] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Technical report, Stanford Tech Report CTSR 2005-02 Light, 2005.

[43] N. Padmanaban, R. Konrad, T. Stramer, E. A. Cooper, and G. Wetzstein. Optimizing virtual reality for all users through gaze-contingent and adaptive focus displays. *Proc. the National Academy of Sciences*, 114(9):2183–2188, 2017. doi: 10.1073/pnas.1617251114

[44] N. Padmanaban, R. Konrad, and G. Wetzstein. Autofocals: Evaluating gaze-contingent eyeglasses for presbyopes. *Science Advances*, 5(6), 2019.

[45] N. Padmanaban, Y. Peng, and G. Wetzstein. Holographic near-eye displays based on overlap-add stereograms. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[46] Y. Peng, S. Choi, J. Kim, and G. Wetzstein. Speckle-free holography

with partially coherent light sources and camera-in-the-loop calibration. *Science Advances*, 7(46), 2021.

[47] Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein. Neural holography with camera-in-the-loop training. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[48] M. Potmesil and I. Chakravarty. Synthetic image generation with a lens and aperture camera model. *ACM Transactions on Graphics (TOG)*, 1(2):85–108, 1982. doi: 10.1145/357299.357300

[49] K. Rathinavel, H. Wang, A. Blate, and H. Fuchs. An extended depth-of-field volumetric near-eye augmented reality display. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 24(11):2857–2866, 2018.

[50] K. Rathinavel, G. Wetzstein, and H. Fuchs. Varifocal occlusion-capable optical see-through augmented reality display based on focus-tunable optics. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 25(11):3125–3134, 2019.

[51] G. Riguer, N. Tatarchuk, and J. Isidoro. Real-time depth of field simulation. In *ShaderX2: Shader Programming Tips and Tricks with DirectX*, vol. 9, pp. 529–556. Wordware Publishing, Inc., 2004.

[52] T. Scheuermann and N. Tatarchuk. Improved depth of field rendering. In *ShaderX3: Advanced Rendering with DirectX and OpenGL (Shaderx Series)*. Charles River Media, 2004.

[53] D. Schmalstieg and T. Höllerer. *Augmented Reality - Principles and Practice*. Addison-Wesley Professional, 2016.

[54] J. Shade, S. Gortler, L.-w. He, and R. Szeliski. Layered depth images. In *Proc. Conference on Computer Graphics and Interactive Techniques (Siggraph)*, pp. 231–242, 1998. doi: 10.1145/280814.280882

[55] L. Shi, F.-C. Huang, W. Lopes, W. Matusik, and D. Luebke. Near-eye light field holographic rendering with spherical waves for wide field of view interactive 3D computer graphics. *ACM Transactions on Graphics (TOG)*, 36(6), 2017.

[56] T. Shibata, J. Kim, D. M. Hoffman, and M. S. Banks. The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of Vision*, 11(8):11, 2011. doi: 10.1167/11.8.11

[57] N. Shroff, A. Veeraraghavan, Y. Taguchi, O. Tuzel, A. Agrawal, and R. Chellappa. Variable focus video: Reconstructing depth and video for dynamic scenes. *Proc. IEEE Int. Conference on Computational Photography (ICCP)*, 2012. doi: 10.1109/ICCPhot.2012.6215219

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

[59] S. Suwajanakorn, C. Hernandez, and S. M. Seitz. Depth from focus with your mobile phone. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3497–3506, 2015.

[60] H. Tang, S. Cohen, B. Price, S. Schiller, and K. N. Kutulakos. Depth from defocus in the wild. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2740–2748, 2017.

[61] M. Tatzgern, R. Grasset, D. Kalkofen, and D. Schmalstieg. Transitional augmented reality navigation for live captured scenes. In *Proc. IEEE Virtual Reality (VR)*, pp. 21–26, 2014. doi: 10.1109/VR.2014.6802045

[62] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics (TOG)*, 26(3):69–es, 2007.

[63] G. Wetzstein, D. Lanman, W. Heidrich, and R. Raskar. Layered 3D: Tomographic image synthesis for attenuation-based light field and high dynamic range displays. *ACM Transactions on Graphics (TOG)*, 30(4), 2011. doi: 10.1145/2010324.1964990

[64] G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar. Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012.

[65] B. Wilburm, M. Smulski, K. Lee, and M. A. Horowitz. The light field video camera. In *SPIE Electronic Imaging*, pp. 29–36, 2002.

[66] W. Wu, P. Llull, I. Tosic, N. Bedard, K. Berkner, and N. Balram. Content-adaptive focus configuration for near-eye multi-focal displays. In *IEEE Int. Conf. on Multimedia and Expo*, pp. 1–6. IEEE, 2016.

[67] L. Xiao, A. Kaplanyan, A. Fix, M. Chapman, and D. Lanman. DeepFocus: Learned image synthesis for computational displays. *ACM Transactions on Graphics (TOG)*, 37(6), 2018.

[68] J. C. Yang, M. Everett, C. Buehler, and L. McMillan. A real-time distributed light field camera. In *Proc. Eurographics Workshop on Rendering*, pp. 77–86, 2002.

[69] H. Yu, M. Bemana, M. Wernikowski, M. Chwesiuk, O. T. Tursun, G. Singh, K. Myszkowski, R. Mantiuk, H. Seidel, and P. Didyk. A perception-driven hybrid decomposition for multi-layer accommodative displays. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 25(5):1940–1950, 2019.

[70] P. Yu, X. Yang, and L. Chen. Parallel-friendly patch match based on jump flooding. In *Advances on Digital Television and Wireless Multimedia Communications*, pp. 15–21. Springer, Berlin, Heidelberg, 2012.

[71] T. Zhan, J. Xiong, J. Zou, and S.-T. Wu. Multifocal displays: Review and prospect. *PhotoniX*, 1:1–31, 2020.

[72] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[73] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4), 2018. doi: 10.1145/3197517.3201323