3D-Kernel Foveated Rendering for Light Fields

Xiaoxu Meng, *Student Member, IEEE,* Ruofei Du, *Member, IEEE,*
Joseph F. JaJa, *Fellow, IEEE,* and Amitabh Varshney, *Fellow, IEEE*

Abstract—Light fields capture both the spatial and angular rays, thus enabling free-viewpoint rendering and custom selection of the focal plane. Scientists can interactively explore pre-recorded microscopic light fields of organs, microbes, and neurons using virtual reality headsets. However, rendering high-resolution light fields at interactive frame rates requires a very high rate of texture sampling, which is challenging as the resolutions of light fields and displays continue to increase. In this paper, we present an efficient algorithm to visualize $4D$ light fields with 3D-kernel foveated rendering (3D-KFR). The 3D-KFR scheme coupled with eye-tracking has the potential to accelerate the rendering of $4D$ depth-cued light fields dramatically. We have developed a perceptual model for foveated light fields by extending the KFR for the rendering of $3D$ meshes. On datasets of high-resolution microscopic light fields, we observe $3.47 \times -7.28 \times$ speedup in light field rendering with minimal perceptual loss of detail. We envision that 3D-KFR will reconcile the mutually conflicting goals of visual fidelity and rendering speed for interactive visualization of light fields.

Index Terms—light fields, foveated rendering, microscopic light fields, eye tracking, visualization.

✦

1 INTRODUCTION

Classic light field rendering is limited to low-resolution images because the rendering process of large-scale, high-resolution light field image arrays requires a great amount of texture sampling, thus increasing the latency of rendering and streaming.

With the rapid advances in optical microscopy imaging, several technologies have been developed for interactively visualizing or reconstructing microscopy volumes [1]–[3]. Recently, light field microscopy [4] has emerged, which allows one to capture light fields of biological specimens in a single shot. Afterwards, one could interactively explore the microscopic specimens with a light-field-renderer, which automatically generates novel perspectives and focal stacks from the microscopy data [5]. Unlike regular images, light field microscopy enables natural-to-senses stereoscopic visualization. Users may examine the high-resolution microscopy light fields with the inexpensive commodity virtual reality head-mounted displays (HMDs) as a natural stereoscopic tool. The method of rendering light field microscopy can be applicable to high-resolution light fields from other sources.

To the best of our knowledge, the interactive visualization of high-resolution light fields with low latency and high-quality remains a challenging problem.

Human vision spans 135° vertically and 160° horizontally, but the highest-resolution foveal vision only covers a 5° central region of the vision [6]. The peripheral vision are fast in perception [7] but has lower visual acuity. As estimated by Patney *et al.* [8], only 4% of the pixels in a modern HMD are mapped on the fovea. Therefore, foveated rendering techniques that allocate more computational resources for the foveal pixels and fewer resources elsewhere can dramatically speed up light field visualization.

In this paper, we present 3D-kernel foveated rendering (3D-KFR), a novel approach to extend the kernel foveated rendering (KFR) [9] framework to light fields. In 3D-KFR, we parameterize the foveation of light fields by embedding polynomial kernel functions in the classic log-polar mapping [10], [11] for each slice. This formulation allows us to alter both the sampling density and distribution, and match the perceptual acuity of virtual reality HMDs. Next, we optimize 3D-KFR by adjusting the weight of each slice in the light fields, so that it automatically selects the

optimal foveation parameters for different images according to the gaze position and achieves higher speedup. In this way, 3D-KFR further accelerates the rendering process of high-resolution light fields while preserving the perceptually accurate foveal detail.

We validated our approach on the rendering of light fields by carrying out both quantitative experiments and user studies. Our method achieves speedups of $3.47 \times -7.28 \times$ on different levels of foveation and different rendering resolutions. Moreover, our user studies suggest optimal parameters for rendering of foveated light fields on modern HMDs.

In summary, our contributions include:

1) designing 3D-KFR, a new visualization method to observe the light fields, which provides similar visual results as the original light fields, but at a higher rendering frame rate;
2) conducting user studies to identify the 3D-KFR parameters governing the density of sampling to maximize perceptual realism and minimize computation for foveated light fields in HMDs;
3) implementing the 3D-KFR light fields pipeline on a GPU, and achieving speedups of up to $7.28 \times$ for the light fields with a resolution of $25 \times 25 \times 1024 \times 1024$ px with minimal perceptual loss of detail.

We organize the paper as follows: Section 2 summarizes the development of foveated 3D graphics and light field rendering. The detailed algorithm for kernel foveated rendering for light fields follows in Section 3 and the user study follows in Section 4. Results of the user study, performance evaluation for the rendering acceleration, and discussions are presented in Section 5. We conclude the paper and propose future work in Section 6.

2 RELATED WORK

Our work is inspired by the prior art in foveated 3D graphics and light field microscopy.

2.1 Foveated 3D Graphics

Approaches for foveated rendering, including mesh simplification in the areas of lower acuity [12]–[14], are reviewed by Weier *et al.* [15]. Texture sampling and shading have gradually come to dominate the costs of rendering sophisticated scenes on modern rendering pipelines [16], [17]. Ragan-Kelley *et al.* [18] use decoupled sampling for stochastic super-sampling of motion and defocus blur at a reduced shading cost. Guenter *et al.* [6] present a three-pass pipeline for foveated 3D rendering by using three eccentricity layers around the tracked gaze point. Vaidyanathan *et al.* [17] perform foveated rendering by sampling coarse pixels (2×2 pixels and 4×4 pixels) in the peripheral regions. Clarberg *et al.* [19] propose a modification to the current hardware architecture, which enables flexible control of shading rates and automatic shading reuse between triangles in tessellated primitives. He *et al.* [16] introduce multi-rate GPU shading to support more shading samples near regions of specular highlights, shadows, edges, and motion blur regions, helping achieve a $3\times$ to $5\times$ speedup. Bektas *et al.* [20] present a testbed featuring gaze-contingent displays, which is able to manage the visual level of detail. With multiple models of the human visual system combined, the system can respond to the viewer's gaze in real-time and rendering a space-variant visualization. Patney *et al.* [8], [21] optimize foveated rendering [17] by addressing temporal artifacts through pre-filters and temporal anti-aliasing. They also show that contrast preservation greatly enhances the image quality by reducing the tunneling effect. Swafford *et al.* [22] implement four independent foveated renderers. The first method reduces peripheral resolution. The second varies per-pixel depth-buffer samples in the fovea and periphery for screen-space ambient occlusion. The third implements a terrain renderer using GPU-level tessellation for the fovea. The final method varies the per-pixel ray-casting steps across the field of view (FoV). Stengel *et al.* [23] use adaptive sampling from fovea to peripheral regions in a gaze-contingent deferred shading pipeline. Meng *et al.* [9] present kernel foveated rendering (KFR) for meshes, which parameterizes foveated rendering by embedding polynomial kernel functions in the classic log-polar mapping, thus allowing users to vary the sampling density and distribution that matches human perception. Ritschel *et al.* [24] also rasterize the rendered image with spatially varying pixel density to produce foveated images. Turner *et al.* [25] propose an algorithm to reduce motion artifacts in the periphery of foveated rendering by aligning the rendered pixel grid to the virtual scene content during rasterization and upsampling. In addition to the application of rendering, the foveation technique could also be applied to accelerate video streaming [26], path tracing [27], and depth-of-field filtering [28].

2.2 Light Field Rendering

4D light fields [29], [30] represent an object or a scene from multiple camera positions. Chai *et al.* [31] determine the minimum sampling rate for light field rendering by spectral analysis of light field signals using the sampling theorem. Ng [32] contributes to a Fourier-domain algorithm for fast digital focusing for light fields. Lanman and Luebke [33] propose near-eye light field displays supporting continuous accommodation of the eye throughout a finite depth of field, thus providing a means to address the accommodation-convergence conflict occurring with existing stereoscopic displays. Huang *et al.* [34] analyze the lens-distortion in light field rendering and correct it, thus improving the resolution and blur quality. Zhang *et al.* [35] propose a unified mathematical model for multilayer-multiframe compressive light field displays that significantly reduces artifacts compared with attenuation-based multilayer-multiframe displays. Lee *et al.* [36] propose foveated retinal optimization (FRO), which has tolerance for pupil movement without gaze tracking while maintaining image definition and accurate focus cues. The system achieves $38° \times 19°$ FoV, continuous focus cues, low aberration, small form factor, and clear see-through property. However, FRO adopts the idea of foveation to improve the display performance of the multilayer displays rather than the rendering speed of 3D content. Sun *et al.* [37] design a real-time foveated 4D light field rendering and display system. Their work analyzes the bandwidth bound for perceiving 4D light fields and proposes a rendering method with importance sampling and a sparse reconstruction scheme. Their prototype renders only $16\% - 30\%$ of the rays without compromising the perceptual quality. The algorithm is designed for the desktop screen. In contrast to the previous work, our approach focused on foveated visualization of large light fields in virtual reality HMDs. Mildenhall *et al.* [38] propose an algorithm to render novel views from an irregular grid of sampled views by expanding each sampled view into a local light field via a multiplane image (MPI) scene representation and blending adjacent local light fields.

2.3 Light Field Microscopy

Weinstein and Descour [39] use lens arrays for single-view-point array microscope with ultra-wide FOV instead of light fields with perspective views. Levoy *et al.* [4] propose using light fields to produce microscopy with perspective views and focal stacks. Wilt *et al.* [40] confirm the importance of observing cellular properties by using light microscopy for neuroscientists. The advances include enabling new experimental capabilities and permitting functional imaging at faster speeds. Prevedel *et al.* [5] implement a light field deconvolution microscopy and demonstrate its ability to simultaneously capture the neuronal activity of the entire nervous system.

3 3D-KERNEL FOVEATED RENDERING FOR 4D LIGHT FIELDS

In this section, we first introduce KFR for 4D light field rendering. Next, we generalize KFR to 3D-KFR. Finally, we discuss the resulting rendering acceleration that 3D-KFR achieves over KFR.

3.1 KFR for 4D Light Field Rendering

In the $k \times k$ light fields with image resolution of $W \times H$, the total number of texture samples for rendering the original light fields N_{original} is:

$$N_{\text{original}} = k^2 \cdot WH \tag{1}$$

KFR accelerates the rendering process of light fields by reducing texture sampling. In the first pass, we perform kernel log-polar transformation for each slice and render to a reduced resolution buffer with dimensions of $k \times k \times w \times h$. In the second pass, we perform the inverse log-polar transformation to map the pixels back to the screen. As defined in Meng *et al.* [9], the kernel function $\mathbf{K}(x)$ can be any monotonically increasing function with $\mathbf{K}(0) = 0$ and $\mathbf{K}(1) = 1$, such as a polynomial,

$$\mathbf{K}(x) = \sum_{i=0}^{\infty} \beta_i x^i, \quad \text{where} \sum_{i=0}^{\infty} \beta_i = 1. \tag{2}$$
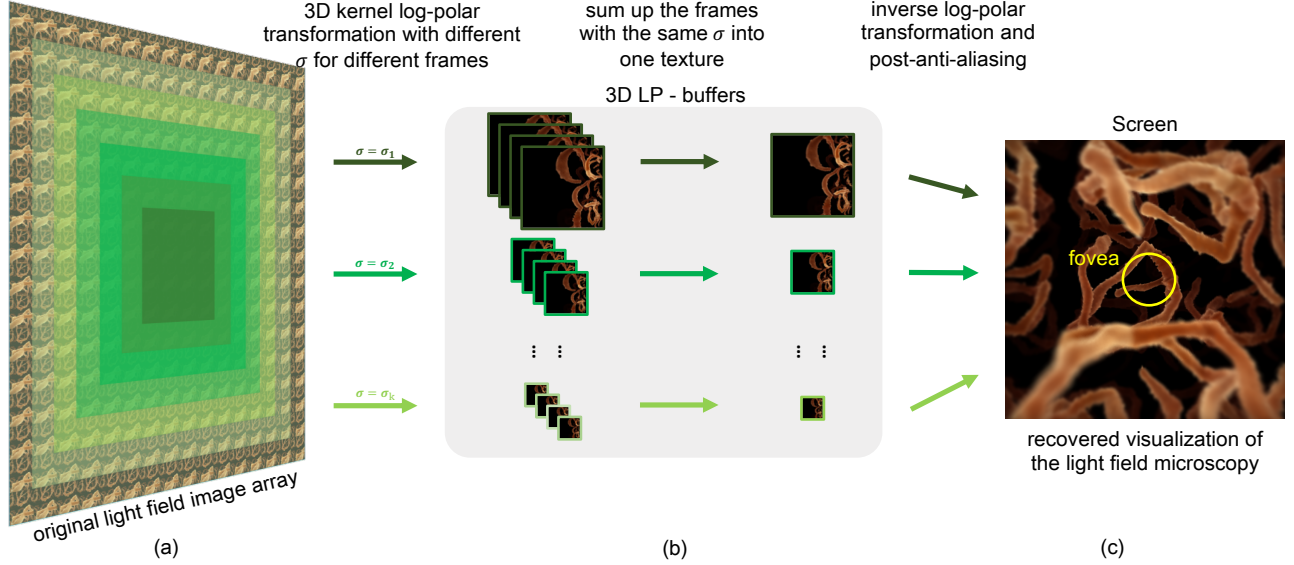
Fig. 1. The pipeline of the foveated light field. Part (a) represents the light fields image array, the region with dark-green mask represents the *foveal region* that the local center camera position of the frames are around fovea. The *peripheral region* masked by light-green are the regions that the local center camera position of the frames is far from the fovea. We apply kernel log-polar transformation for each image with different σ (σ is determined by the gaze position) to get the image sub-arrays as shown in the left part of (b). Then we average the image sub-arrays to get textures as shown in the right part of (b). Finally, we apply the inverse log-polar transformation for each sub-array, calculate the weighted sum of pixel values and perform anti-aliasing to get the final image displayed on-screen as shown in part (c).

We next present the two passes of the KFR algorithm.

In the first pass, we transform the image from Cartesian coordinates to kernel log-polar coordinates. For each pixel in screen space with coordinates (x, y), foveal point $F(\mathring{x}, \mathring{y})$ in Cartesian coordinates, we define x', y' as

$$x' = x - \mathring{x}, \quad y' = y - \mathring{y}. \tag{3}$$

Then, we transform point (x', y') to (u, v) in kernel log-polar coordinates using Equation 4,

$$
\begin{aligned}
u &= \mathbf{K^{-1}} \left(\frac{\log \| x', y' \|_2}{L} \right) \cdot w \\
v &= \left(\arctan \left(\frac{y'}{x'} \right) + \mathbf{1} \left[y' < 0 \right] \cdot 2\pi \right) \cdot \frac{h}{2\pi}
\end{aligned}
\tag{4}
$$

$\mathbf{K^{-1}}(\cdot)$ is the inverse of the kernel function, and L is the log of the maximum distance from fovea to one of the four corners of the screen as shown in Equation 5,

$$L = \log \left(\max \left(\max \left(l_1, l_2 \right), \max \left(l_3, l_4 \right) \right) \right). \tag{5}$$

Here,

$$
\begin{aligned}
l_1 &= \| \mathring{x}, \mathring{y} \|_2 \\
l_2 &= \| W - \mathring{x}, H - \mathring{y} \|_2 \\
l_3 &= \| \mathring{x}, H - \mathring{y} \|_2 \\
l_4 &= \| W - \mathring{x}, \mathring{y} \|_2
\end{aligned}
\tag{6}
$$

We define $\sigma = \frac{W}{w} = \frac{H}{h}$ as the ratio between the full-resolution screen width (or height) and the reduced-resolution buffer width (or height). The number of texture samples for the first pass $N_{\text{KFR pass 1}}$ can be theoretically inferred as:

$$N_{\text{KFR pass 1}} = k^2 \cdot \frac{W}{\sigma} \cdot \frac{H}{\sigma} = \frac{k^2}{\sigma^2} \cdot WH \tag{7}$$

In the second pass, a pixel with kernel log-polar coordinates (u, v) is transformed back to (x'', y'') in Cartesian coordinates. Let

$$A = \frac{L}{w}, \quad B = \frac{2\pi}{h}, \tag{8}$$

then the inverse transformation can be formulated as Equation 9,

$$
\begin{aligned}
x'' &= \exp \left(A \cdot \mathbf{K}(u) \right) \cdot \cos \left(Bv \right) + \mathring{x} \\
y'' &= \exp \left(A \cdot \mathbf{K}(u) \right) \cdot \sin \left(Bv \right) + \mathring{y}
\end{aligned}
\tag{9}
$$

The number of texture samples for the second pass $N_{\text{KFR pass 2}}$ is

$$N_{\text{KFR pass 2}} = WH \tag{10}$$

The total number of texture samples for rendering the light fields with KFR is:

$$
\begin{aligned}
N_{\text{KFR}} &= N_{\text{KFR pass 1}} + N_{\text{KFR pass 2}} \\
&= \left(\frac{k^2}{\sigma^2} + 1 \right) \cdot WH
\end{aligned}
\tag{11}
$$

The parameter σ controls the total number of pixels of the reduced-resolution buffer, thus controlling the foveation rate and the amount of sampling. Comparing Equations 1 and 11, the number of texture samples can be greatly reduced in KFR with $\sigma > 1.0$. Kernel function controls the distribution of pixels in the whole image. By adjusting kernel functions, we can determine the pixel distribution and choose one that mimics the photo-receptor distribution of human eyes. The kernel log-polar mapping algorithm allows us to mimic the fall-off of photo-receptor density of human visual system with different σ and different kernel functions.

3.2 3D-KFR for 4D Light Fields

The rendering of 4D light fields is different from the rendering of 3D meshes because the center camera position of each slice is different. Since the foveation level of a pixel is affected by the distance to the center camera, the foveation parameter can be different for different slices in a light-field image array. We optimize the KFR algorithm into 3D-KFR by adjusting the weight of each slice in the light fields, so that it automatically selects the optimal foveation parameters for different images according to the
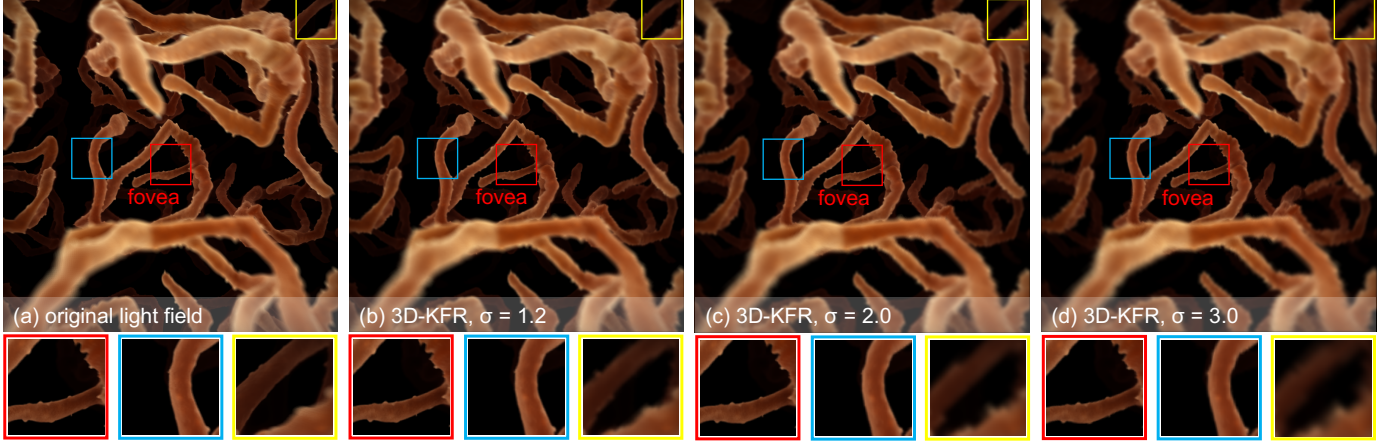
Fig. 2. The result comparison of the foveated light fields with fovea on the center of the screen. (b) - (d) are the application of 3D-KFR on light fields with (b) $\sigma_0 = 1.2$, (c) $\sigma_0 = 2.0$, (d) $\sigma_0 = 3.0$. The left zoomed-in views show that the application of 3D-KFR doesn't make changes in the fovea; the middle zoomed-in views and the right zoomed-in views show that larger σ_0 causes detail loss in the peripheral region.
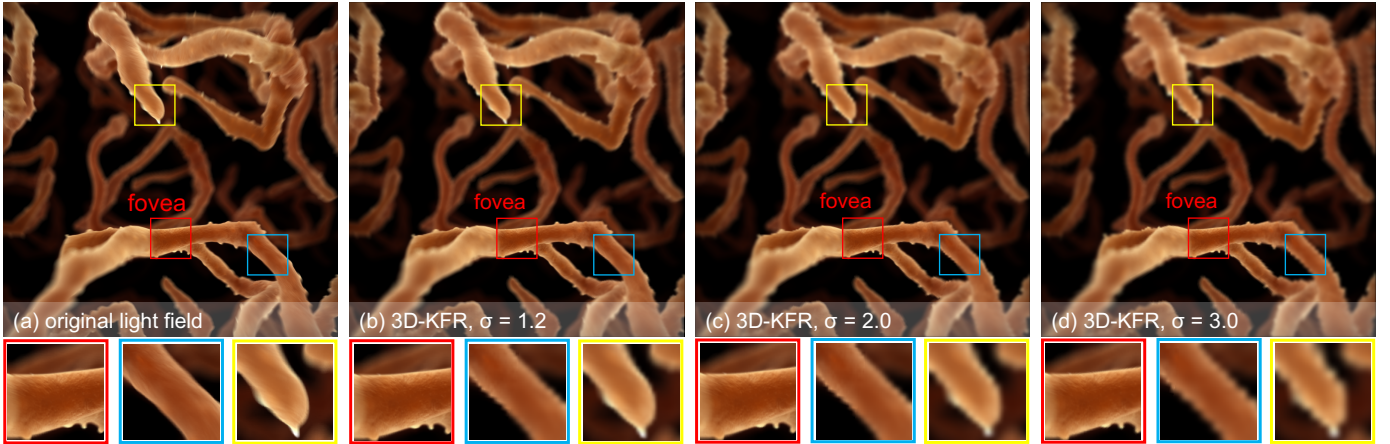

Fig. 3. The result comparison of the foveated light fields with fovea on the peripheral region of the screen. (b) - (d) are the application of 3D-KFR on light fields with (b) $\sigma_0 = 1.2$, (c) $\sigma_0 = 2.0$, (d) $\sigma_0 = 3.0$. The left zoomed-in views show that the application of 3D-KFR doesn't make changes in the fovea; the middle zoomed-in views and the right zoomed-in views show that larger σ_0 causes detail loss in the peripheral region.

gaze position, thereby achieving greater speedup. Our algorithm consists of two passes as shown in Figure 1.

We define d as the distance between the local center camera of the frame $X_{\text{cam } ij}(x_{ij}, y_{ij})$ and the gaze position $X_{\text{cam } 0}(x_0, y_0)$,

$$d = \left\| X_{\text{cam } 0} - X_{\text{cam } ij} \right\|_2 \tag{12}$$

We partition the original dataset into multiple progressive regions: the inner *foveal region* (highlighted in dark green) indicates the fovea, *i.e.*, where the user is currently looking at; as d increases, the *peripheral regions* (highlighted in lighter green and white) are rendered in smaller framebuffers with less texture sampling.

We classify the frame of the i-th row and j-th column I_{ij} to *foveal region* or *peripheral region* with different framebuffers by d as shown in Equation 13.

$$I_{ij} \in \begin{cases} \textit{foveal region} & d < r_0 \\ \textit{peripheral region 1} & r_0 \leq d < r_1 \\ \textit{peripheral region 2} & r_1 \leq d < r_2 \\ ... & ... \\ \textit{peripheral region N} & r_{N-1} \leq d < r_N \end{cases} \tag{13}$$

In the first pass, assume the *foveal region* covers k_0 frames and the *peripheral region i* covers k_i frames. Our approach reduces the framebuffer size for the *foveal region* by σ_0^2, and reduces the framebuffer size for the *peripheral region i* by $\sigma_1^2, ..., \sigma_N^2$, respectively. Then the number of total texture samples in the first pass $N_{\text{3D-KFR pass 1}}$ can be theoretically inferred as:

$$N_{\text{3D-KFR pass 1}} = k_0 \cdot WH \cdot \frac{1}{\sigma_0^2} + k_1 \cdot WH \cdot \frac{1}{\sigma_1^2} + ... + k_N \cdot WH \cdot \frac{1}{\sigma_N^2}$$

$$= \left(\frac{k_0}{\sigma_0^2} + \frac{k_1}{\sigma_1^2} + ... + \frac{k_N}{\sigma_N^2} \right) \cdot WH \tag{14}$$

We can also write N_{original} as:

$$N_{\text{original}} = k^2 \cdot WH = (k_0 + k_1 + ... + k_N) \cdot WH \quad (15)$$

So the total number for texture sampling for the *foveal region* and *peripheral region* are reduced by $\frac{1}{\sigma_0^2}\times$, $\frac{1}{\sigma_1^2}\times$, ..., and $\frac{1}{\sigma_N^2}\times$, respectively. We choose smaller σ with small d in order to keep more details. And we choose larger σ for frames with larger distance in order to reduce rendering cost (*i.e.*, $\sigma_0 \leq \sigma_1 \leq ... \leq \sigma_n$).

The algorithm of light field rendering combined with kernel log-polar transformation is shown in Algorithm 1.

ALGORITHM 1: 3D-KFR: Pass I

Input:
> Aperture size: a,
> focal point ratio: f,
> fovea coordinate in screen space: X_{fovea} $(\mathring{x},\mathring{y})$,
> pixel coordinate in LP-Buffer: X_{buffer} (u,v),
> $k \times k$ light fields $\{I\}$ with image resolution of $n \times n$.

Output:
> Pixel value $C_{\text{buffer}, \sigma}$ for the coordinate X_{buffer}.

1: acquire the coordinate for the center camera $X_{\text{cam } 0}$
2: acquire the coordinate for the fovea X_{fovea}
3: initialization: $C_{\text{buffer}, \sigma} \leftarrow \mathbf{0}, N_{\text{buffer}, \sigma} \leftarrow 0$
4: **for** row index $i \in [0,k]$ **do**
5:   **for** column index $j \in [0,k]$ **do**
6:     calculate σ with X_{fovea} with Equation 13
7:     update L with X_{fovea} with Equation 5
8:     let $A = \frac{L}{w}$, $B = \frac{2\pi}{h}$
9:     acquire $X_{\text{cam } ij}$ for frame I_{ij}
10:     $d_{ij} \leftarrow \|X_{\text{cam } 0} - X_{\text{cam } ij}\|_2$
11:     **if** $d_{ij} < a$ **then**
12:       $x' \leftarrow \exp(A \cdot \mathbf{K}(u,\sigma)) \cdot \cos(Bv) + \mathring{x}$
13:       $y' \leftarrow \exp(A \cdot \mathbf{K}(v,\sigma)) \cdot \sin(Bv) + \mathring{y}$
14:       $X'_{\text{Sample}} \leftarrow (x',y')$
15:       $X_{\text{Sample}} \leftarrow X_{\text{cam } ij} + (X'_{\text{Sample}} - X_{\text{cam } ij}) \cdot f$
16:       **if** X_{Sample} in the range of the screen **then**
17:         $C_{\text{buffer}, \sigma} \leftarrow C_{\text{buffer}, \sigma} + I_{ij} \cdot Color(X_{\text{Sample}})$
18:         $N_{\text{buffer}, \sigma} \leftarrow N_{\text{buffer}, \sigma} + 1$
19:       **end if**
20:     **end if**
21:   **end for**
22: **end for**
23: return $C_{\text{buffer}, \sigma} \leftarrow \frac{C_{\text{buffer}, \sigma}}{N_{\text{buffer}, \sigma}}$

In the second pass, we carry out the inverse-log-polar transformation with anti-aliasing to map the reduced-resolution rendering to the full-resolution screen, the algorithm is shown in Algorithm 2. We use a Gaussian filter with a 5×5 kernel to reduce artifacts in the peripheral regions of the recovered image in the screen.

The number of texture samples for the second pass $N_{\text{3D-KFR pass 2}}$ is:

$$N_{\text{3D-KFR pass 2}} = (1+N) \cdot WH \quad (16)$$

The total number of texture samples for rendering the light fields with 3D-KFR is

$$N_{\text{3D-KFR}} = N_{\text{3D-KFR pass 1}} + N_{\text{3D-KFR pass 2}}$$
$$= \left(\frac{k_0}{\sigma_0^2} + \frac{k_1}{\sigma_1^2} + ... + \frac{k_N}{\sigma_N^2} + 1 + N \right) \cdot WH \quad (17)$$

In the light field rendering, we commonly have $k \geq 16$. In 3D-KFR, we commonly have $1.0 < \sigma \leq 3.0$, and we choose $N = 2$ as the number of peripheral regions and $\mathbf{K}(x) = x^4$ as the kernel function. Therefore, we have

$$N_{\text{KFR pass 2}} = WH \ll \frac{k^2}{\sigma^2} \cdot WH = N_{\text{KFR pass 1}}, \quad (18)$$

and

$$N_{\text{3D-KFR pass 2}} = (N+1)WH$$
$$\ll \left(\frac{k_0}{\sigma_0^2} + \frac{k_1}{\sigma_1^2} + ... + \frac{k_N}{\sigma_N^2} \right) \cdot WH = N_{\text{3D-KFR pass 1}}. \quad (19)$$

Equations 18 and 19 show that the extra time consumed by the *Pass II* can be omitted. We then have

$$N_{\text{KFR}} \approx \left(\frac{k^2}{\sigma_0^2} \right) \cdot WH \quad (20)$$

$$N_{\text{3D-KFR}} \approx \left(\frac{k_0}{\sigma_0^2} + \frac{k_1}{\sigma_1^2} + ... + \frac{k_N}{\sigma_N^2} \right) \cdot WH \quad (21)$$

Comparing Equations 1, 20 and 21, we have

$$N_{\text{3D-KFR}} \ll N_{\text{KFR}} \ll N_{\text{original}}, \quad (22)$$

which shows that the 3D KFR scheme can accelerate the rendering of the light fields beyond a simple KFR approach. The resulting comparison of the original light fields rendering and the 3D-KFR for light fields is shown in Figure 2 and Figure 3. With 3D-KFR applied, pixel density decreases from the fovea to the periphery. We do not notice any differences in the fovea with different σ_0 between the left zoomed-in views because 3D-KFR uses a weighted-sum which strengthens the frames with small d. For the same reason, we can notice the loss of detail from the right zoomed-in views of the periphery. Next, we determine what parameters ensure that the peripheral loss and the peripheral blur are not noticeable by conducting user studies.

ALGORITHM 2: 3D-KFR: Pass II

Input:
> Fovea coordinate in screen space: X_{fovea} $(\mathring{x},\mathring{y})$,
> pixel coordinate in screen space: X_{Display} (x,y),

Output:
> Pixel value C_{display} for coordinate X_{Display}.

1: initialization: $C_{\text{display}} \leftarrow \mathbf{0}, N_{\text{display}} \leftarrow 0$
2: acquire the coordinate for the fovea X_{fovea}
3: update L with X_{fovea} with Equation 5
4: **for** each attachment I_σ in LP-Buffer **do**
5:   $x' \leftarrow x - \mathring{x}$
6:   $y' \leftarrow y - \mathring{y}$
7:   $u \leftarrow \mathbf{K}^{-1}\left(\frac{\log\|x',y'\|}{L} \right) \cdot w$
8:   $v \leftarrow \arctan\left(\frac{y'}{x'} \right) \frac{h}{2\pi} + \mathbf{1}[y' < 0] \cdot h$
9:   $X_{\text{Sample}} \leftarrow (u,v)$
10:   $C_{\text{display}} \leftarrow C_{\text{display}} + I_\sigma \cdot Color(X_{\text{Sample}})$
11:   $N_{\text{display}} \leftarrow N_{\text{display}} + 1$
12: **end for**
13: return $C_{\text{display}} \leftarrow \frac{C_{\text{display}}}{N_{\text{display}}}$

---

4 USER STUDY

We have carried out user studies to find the largest foveation parameter values for σ_0 that results in visually acceptable foveated rendering.

4.1 Apparatus

Our user study apparatus is shown in Figure 4. We used an *Alienware* laptop with an NVIDIA GTX 1080, a *FOVE* HMD, and an *XBOX* controller. The *FOVE* display has a 100° field of view, a resolution of 2560×1440, and a 120 Hz infrared eye-tracking system with a precision of 1° and a latency of 14 ms.

Since public datasets on large-scale and high-resolution microscopy light fields datasets are not yet available, we have rendered and open sourced 30 microscopy light field datasets[1]. We synthesized the microscopy dataset on *Cell* and *Cellular Lattice* for the user study.

Fig. 4. Our user study set up with gaze-tracker integrated into the *FOVE* head-mounted display.

4.2 Participants

We recruited a total of 22 participants via campus email lists and flyers. All participants are at least 18 years old with normal or corrected-to-normal vision (with contact lenses).

4.3 Procedure

We conducted three different and independent experiments to test the parameters for which 3D-KFR produces acceptable quality to non-foveated rendering: a **Pair Test**, a **Random Test**, and a **slider test**. Inspired by prior user studies [8], [23], we instructed the participants to maintain their gaze at the center of the screen so the participants experienced a similar visual field across all the trials.

In the **Pair Test**, we presented each participant with pairs of foveated and full-resolution light field renderings. We presented the two renderings in each pair in a random order and separated by a short interval of black screen (0.75 seconds). The foveation parameter ranged between $\sigma_0 = 1.2$ to $\sigma_0 = 3.0$. Pairs at all quality levels in this range were presented twice (monotone increasing then monotone decreasing) for each dataset, i.e. σ_0 increased from 1.2 to 3.0 then decreased from 3.0 to 1.2. At the end of each comparison, the participant responded upon the similarity between the two rendering results by the *XBOX* controller. The answer contains 5 confidence levels: 5 represents *perceptually identical*, 4 represents *minimal perceptual difference*, 3 represents *acceptable*

1. Simulated HD Light Fields: https://foveation.umiacs.umd.edu/3D_KFR/index.html

perceptual difference, 2 represents *noticeable perceptual difference* and 1 represents *significant perceptual difference*.

In the **Random Test**, we presented each participant with pairs of foveated and full-resolution light field renderings. We presented the two renderings in each pair in a random order and separated by a short interval of black (0.75 seconds). The foveation parameter ranged between $\sigma_0 = 1.2$ to $\sigma_0 = 3.0$. Pairs at all quality levels in the range were presented once for each dataset in random order. At the end of each comparison, the participant responded upon the similarity between the two rendering results by the *XBOX* controller. The answer contains 5 confidence levels: 5 represents *perceptually identical*, 4 represents *minimal perceptual difference*, 3 represents *acceptable perceptual difference*, 2 represents *noticeable perceptual difference* and 1 represents *significant perceptual difference*.

The **slider test** lets the participants navigate the foveation quality space themselves. First, the participant observed the full-resolution rendering result as a reference. Next, we presented the participant with the lowest level of foveation quality ($\sigma_0 = 3.0$) while the participant could progressively increase the foveation level (with a step size of 0.1). The participant switched between the foveated rendering result and the reference image back and forth, until they found out the lowest foveation level which is visually equivalent to the non-foveated reference. We recorded the first quality level index at which the participant stopped as the final response for the slider test.

While the eye-tracking data from the FOVE HMD could be used at a high level for ensuring that the users gaze points are at the foveal location, there are internal errors in the current generation eye-tracking HMDs, *e.g.*, mismatch of left and right eye positions and jittering of the gaze positions, especially around eye blinks. Therefore, it is not satisfactory to depend on the current FOVE HMD's eye-tracking results for validating the participants' focus. We therefore instruct the participants to focus on the center of the rendered light fields and assist them to focus with a tiny dot at the center of the screen. Additionally, we evaluate the participants' performance with respect to the ground truth data for the determination of the participants' attention.

To ensure the validity of our study, we randomly inserted 30% of the trials to be control or validation trials, that had identical full-resolution results for both choices in the **Pair Test** and the **Random Test**. If the participant declared these validation renderings to have a low score for similarity (3 or lower): *mostly the same with acceptable difference*, *noticeable difference* or *totally different*, we would ask the participant to stop, take some rest, and then continue. Meanwhile, we recorded this choice as an *error*. If *error* ≥ 5 in the **Pair Test** and the **Random Test**, we would stop the user study and discard the data of the user. We discarded two participants according to this rule.

5 RESULTS AND ANALYSIS

Let S_σ be the average score of all the users for a specific σ_0, and let P_σ be the percentage of responses of rated foveated and non-foveated renderings as *perceptually identical (5)* and *minimal perceptual difference (4)* for a specific $\sigma = \sigma_0$.

The result of S_σ for the **Pair Test** is shown in Figure 5. Generally, S_σ decreases with the increase of σ_0. A Friedman test revealed a significant effect of the users' responses on foveation parameter σ $\left(\chi^2(20) = 104.3, p < 8.9 \times 10^{-14}\right)$. The result of P_σ for the **Pair Test** is shown in Figure 6. We have identified a threshold of $P_\sigma = 90\%$ for $\sigma_{pair} = 2.4$ as the foveation parameter
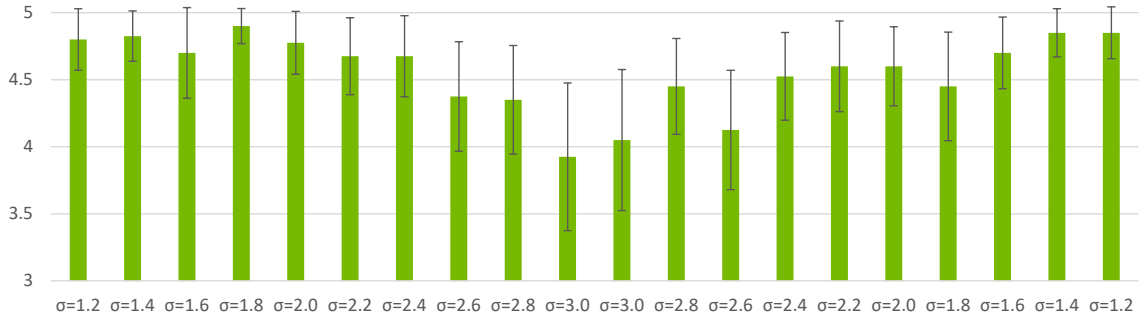
Fig. 5. The **Pair Test** responses of S_σ across sliding foveation parameters σ. S_σ decreases with the increase of σ. 5 represents perceptually identical, 4 represents minimal perceptual difference, 3 represents acceptable perceptual difference, 2 represents noticeable perceptual difference, and 1 represents significant perceptual difference (2 and 1 are not shown).
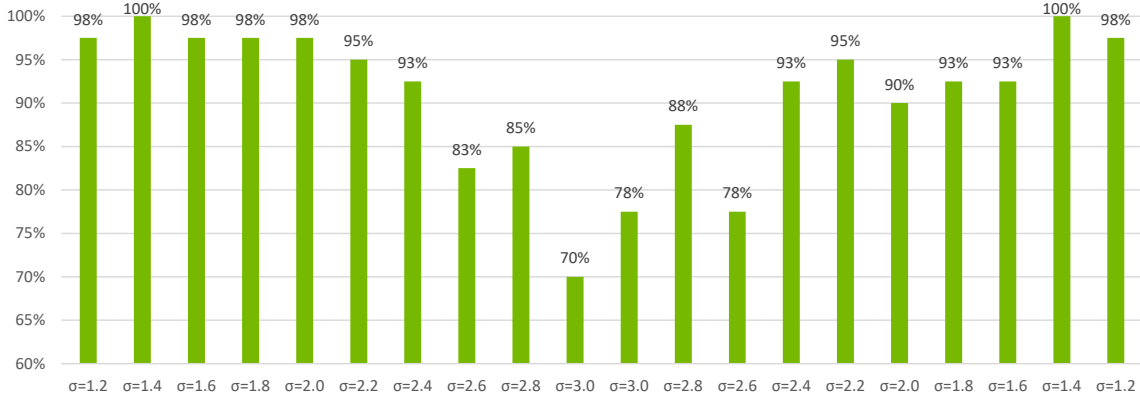

Fig. 6. The **Pair Test** responses of P_σ across sliding foveation parameters σ. P_σ decreases with the increase of σ.

Fig. 7. The **Random Test** responses of S_σ across gradually varied foveation parameters σ. S_σ decreases with the increase of σ. 5 represents perceptually identical, 4 represents minimal perceptual difference, 3 represents acceptable perceptual difference, 2 represents noticeable perceptual difference, and 1 represents significant perceptual difference (2 and 1 are not shown).
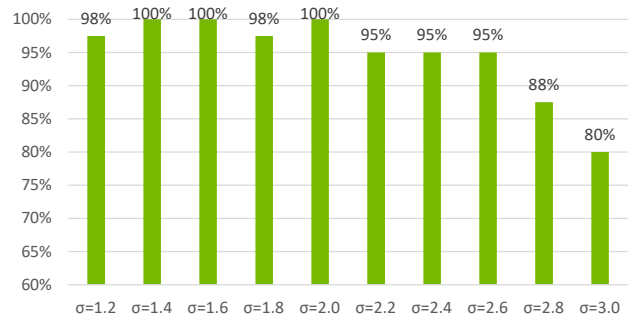

Fig. 8. The **Random Test** responses of P_σ across sliding foveation parameters σ. P_σ decreases with the increase of σ.

that provides minimal perceptual differences based on the **Pair Test**.

The result of S_σ for the **Random Test** is shown in Figure 7. The trend that S_σ decreases with an increase of σ matches our expectation. A Friedman test revealed a significant effect of the users' responses on foveation parameter σ $\left(\chi^2(20) = 29.2, p < 0.0006\right)$. The result of P_σ for the **Random Test** is shown in Figure 8. We have identified a threshold of $P_\sigma = 90\%$ for $\sigma_{random} = 2.6$ as the foveation parameter that provides minimal perceptual differences based on the **Random Test**.

The histogram of the user-chosen thresholds in the **Slider Test** is shown in Figure 9. For instance, the histogram shows that 25% of the users found that $\sigma = 3.0$ or lower is acceptable; 75% of the users found that $\sigma = 1.8$ or lower is acceptable. With $\sigma_0 = 1.6$, 80% of the users considered that the foveated rendering is visually indistinguishable from full-resolution rendering. We chose threshold $\sigma_{slider} = 1.6$.

Note that $\sigma_{slider} = 1.6$ is smaller than $\sigma_{pair} = 2.4$ and $\sigma_{random} = 2.6$. We speculate that the reason for a smaller sigma in the **Slider Test** is: if the users are free to choose the threshold, they tend to choose the best quality they can achieve, instead of the lower bound of the perceptually indistinguishable quality.

Using the three parameters, one could think of building a foveated rendering system where the saccades are implemented with $\sigma = 2.6$ and the fixation implemented with $\sigma = 1.6$.

Resolution	Ground Truth	$\sigma = 1.6$		$\sigma = 2.4$		$\sigma = 2.6$	
		3D KFR	3D-KFR Speedup	3D KFR	3D-KFR Speedup	3D KFR	3D-KFR Speedup
$20 \times 20 \times 1024 \times 1024$	66.83 ms	19.27 ms	$3.47\times$	10.22 ms	$6.54\times$	9.39 ms	$7.11\times$
$21 \times 21 \times 1024 \times 1024$	74.17 ms	22.39 ms	$3.31\times$	11.90 ms	$6.24\times$	10.39 ms	$7.14\times$
$22 \times 22 \times 1024 \times 1024$	92.33 ms	28.26 ms	$3.27\times$	14.65 ms	$6.30\times$	12.64 ms	$7.30\times$
$23 \times 23 \times 1024 \times 1024$	100.26 ms	30.64 ms	$3.27\times$	16.30 ms	$6.15\times$	13.95 ms	$7.18\times$
$24 \times 24 \times 1024 \times 1024$	122.29 ms	35.92 ms	$3.40\times$	19.09 ms	$6.41\times$	16.79 ms	$7.28\times$
$25 \times 25 \times 1024 \times 1024$	138.93 ms	41.42 ms	$3.35\times$	21.96 ms	$6.33\times$	19.09 ms	$7.28\times$

TABLE 1
The average timings and the corresponding speedups of 3D-KFR at different light field dimensions and foveation parameters σ. We used the *Tissue* datasets for the performance evaluation.
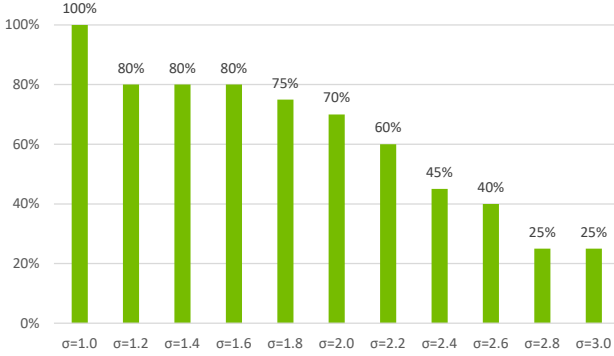

Fig. 9. The histogram of the optimal foveation parameter σ selected by each user in the **Slider Test**. For instance, the histogram shows that 80% of the users found that $\sigma = 1.6$ or lower is acceptable.
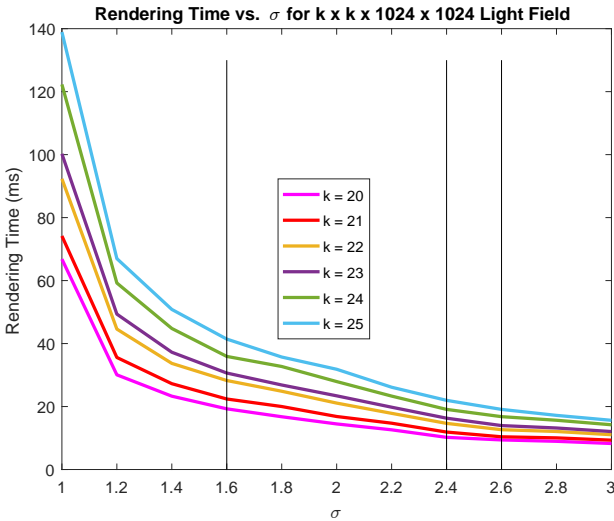

Fig. 10. The rendering time for light fields with different dimension and different σ.

Performance Evaluation and Discussion We have implemented the 3D kernel foveated rendering pipeline in C++ 11 and OpenGL 4 on NVIDIA GTX 1080. We report the results of our rendering acceleration for the *tissue* dataset at the resolution of $k \times k \times 1024 \times 1024$. We tested the rendering time for different light field dimensions ($20 \leq k \leq 25$) and different σ_0 ($1.2 \leq \sigma_0 \leq 3.0$) with $\sigma_1 = 1.6\sigma_0$, $\sigma_2 = 2.0\sigma_1$. We used the kernel function $\mathbf{K}(x) = x^4$. The evaluations are shown in Figure 10, where $\sigma_0 = 1.0$ corresponds to the rendering time of the original fields, and $\sigma_0 > 1.0$ corresponds to the 3D-KFR rendering time.

We further tested the rendering time comparison and the speedup for $\sigma_{slider} = 1.6$, $\sigma_{pair} = 2.4$ and $\sigma_{random} = 2.6$ as shown in Table 1. With $\sigma_{pair} = 2.4$, the rendering time is less than 21.96 ms (45.54 fps); with $\sigma_{random} = 2.6$, the rendering time is less than 19.09 ms (52.38 fps). $\sigma_{pair} = 2.4$ and $\sigma_{random} = 2.6$ meets the real-time requirement of 30 fps. With $\sigma_{slider} = 1.6$, the rendering times for $k = 20, 21, 22$, or 23 are less than 30.64 ms (32.64 fps), which meets the real-time requirement of 30 fps. While the rendering times for $k = 24$ and 25 are less than 41.42 ms (24.14 fps), they are still able to achieve reasonably interactive frame rates.

The comparisons of the visualization of the original light field rendering and the 3D-KFR rendering of different datasets are shown in Figure 11 - Figure 13 (please see the supplemental material for all the 30 datasets). We use structural dissimilarity (DSSIM) [41] [42] between the 3D-KFR and the original light field approaches as the metric to evaluate the quality of 3D-KFR results. DSSIM can be derived from structural similarity index (SSIM) [43] [44]. The measurement of SSIM and DSSIM between the two images Ψ and Ω with size $N \times N$ is shown in Equations 23 and 24.

$$SSIM(\Psi, \Omega) = \frac{(2\mu_\Psi\mu_\Omega + c_1)(2\sigma_{\Psi\Omega} + c_2)}{(\mu_\Psi^2 + \mu_\Omega^2 + c_1)(\sigma_\Psi^2 + \sigma_\Omega^2 + c_2)} \quad (23)$$

$$DSSIM(\Psi, \Omega) = \frac{1 - SSIM(\Psi, \Omega)}{2} \quad (24)$$

where μ_Ψ and μ_Ω are the average pixel values for images Ψ and Ω, respectively; σ_Ψ and σ_Ω are the pixel variances for images Ψ and Ω, respectively; $\sigma_{\Psi\Omega}$ is the covariance between images Ψ and Ω; c_1, c_2 are two constants used to stabilize the division with a weak denominator.

SSIM is a perception-based model that considers image degradation as perceived change in structural information. A pair of images with low DSSIM indicates better structural similarity. We measure the average DSSIM of the RGB channels, and we show the results in Figure 11 - Figure 13. The DSSIM measurement of the zoomed-in views for the fovea regions are small, which indicates high visual similarity. With an increase in distance between the fovea position and the pixel position, the DSSIM increases because of the foveation effect.

6 CONCLUSIONS AND FUTURE WORK

We have presented 3D-KFR, a novel approach to accelerate the interactive visualization of high-resolution light fields. We conduct user studies to determine the optimal foveation parameters to validate the 3D-KFR pipeline in practice. According to the quantitative experiments, our methods accelerate the rendering process of large-scale, high-resolution light fields by a factor of up to $7.28\times$ at the resolution of $25 \times 25 \times 1024 \times 1024$.
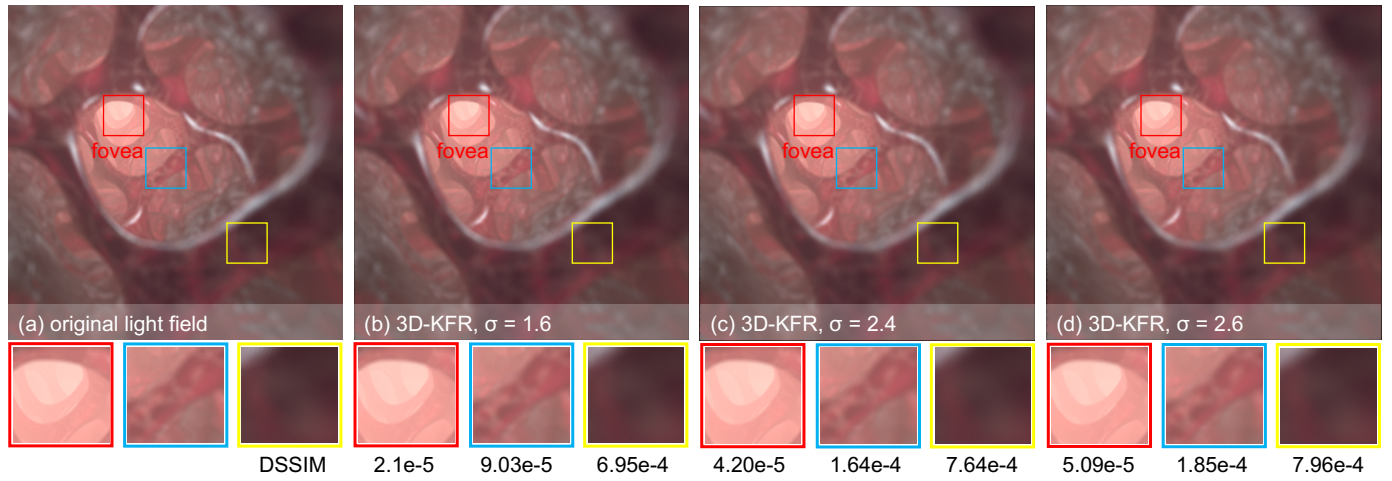
Fig. 11. The result comparison of the foveated light field *Biomine II*. (b) - (d) are the application of 3D-KFR on light field with (b) $\sigma_{slider} = 1.6$, (c) $\sigma_{pair} = 2.4$, (d) $\sigma_{random} = 2.6$. The measured DSSIM (lower is better) is shown for each zoomed-in view.
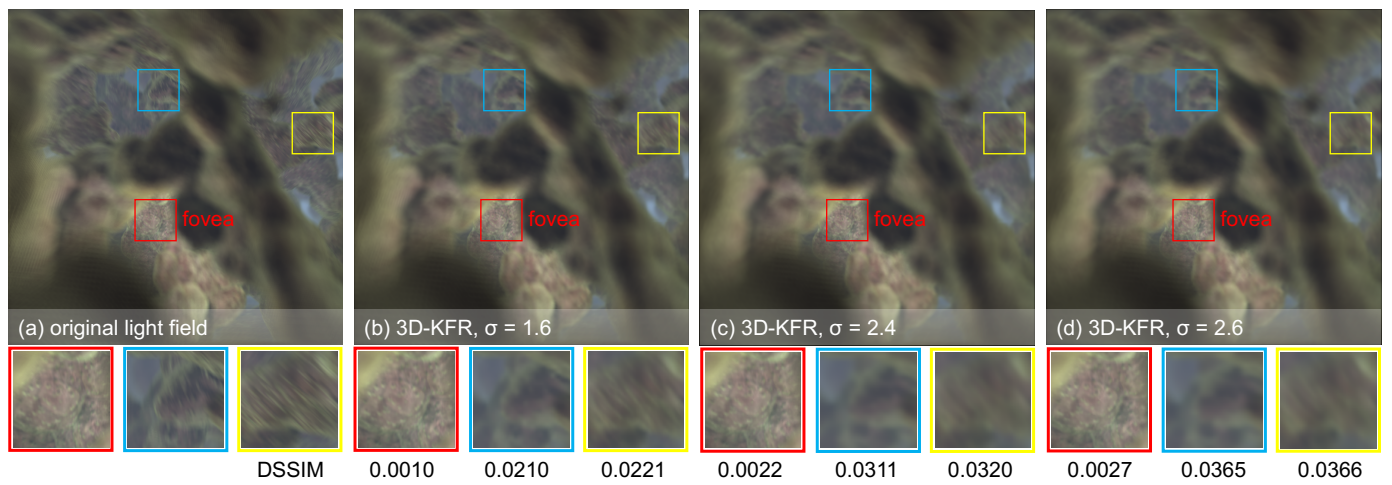

Fig. 12. The result comparison of the foveated light field *Cellular Lattice IV*. (b) - (d) are the application of 3D-KFR on light field with (b) $\sigma_{slider} = 1.6$, (c) $\sigma_{pair} = 2.4$, (d) $\sigma_{random} = 2.6$. The measured DSSIM (lower is better) is shown for each zoomed-in view.
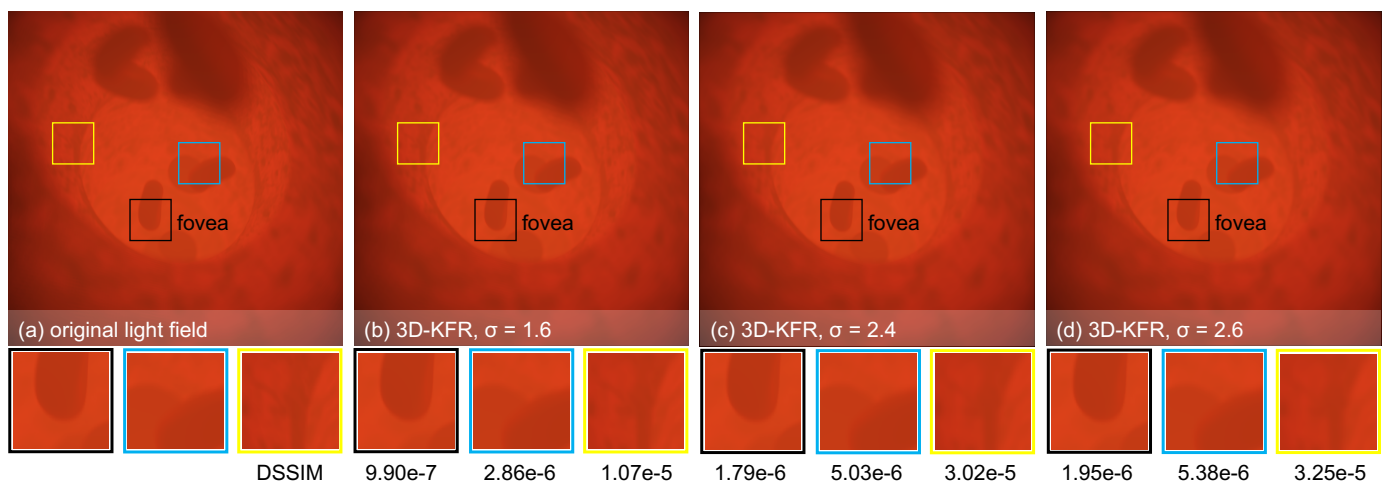

Fig. 13. The result comparison of the foveated light field *Red Cells IV*. (b) - (d) are the application of 3D-KFR on light field with (b) $\sigma_{slider} = 1.6$, (c) $\sigma_{pair} = 2.4$, (d) $\sigma_{random} = 2.6$. The measured DSSIM (lower is better) is shown for each zoomed-in view.

Our algorithm offers an effective approach to render the high resolution light fields using foveation for virtual reality HMD with low latency, low power consumption and minimal perceptual differences. With the increase of VR headset resolution and the growth of the VR market, we envision that 3D-KFR may inspire further research in the foveated rendering of high-resolution light fields.

Potential improvements could be gained with the choice of the relationship between σ_0, σ_1, and σ_2, where other sigma arrays may provide a higher speedup. However, the trade-off between rendering quality and foveation parameter σ always exists. We plan on exploring this relationship between rendering quality and σ in future.

Our current user study only evaluates static foveation with fixed gaze position. Extending it to adapt with changes in gaze is desirable. This will involve taking into account different levels of temporal sensitivity across various subjects. We hope to further explore this in the future.

ACKNOWLEDGEMENT

REFERENCES

[1] Y. Wan, H. Otsuna, C. Chien, and C. Hansen, "An Interactive Visualization Tool for Multi-Channel Confocal Microscopy Data in Neurobiology Research," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1489–1496, Nov 2009.

[2] M. Hadwiger, J. Beyer, W. Jeong, and H. Pfister, "Interactive Volume Exploration of Petascale Microscopy Data Streams Using a Visualization-Driven Virtual Memory Approach," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2285–2294, Dec 2012.

[3] K. Mosaliganti, L. Cooper, R. Sharp, R. Machiraju, G. Leone, K. Huang, and J. Saltz, "Reconstruction of Cellular Biological Structures From Optical Microscopy Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 863–876, July 2008.

[4] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, "Light Field Microscopy," *ACM Trans. Graph*, vol. 25, no. 3, pp. 924–934, 2006.

[5] R. Prevedel, Y.-G. Yoon, M. Hoffmann, N. Pak, G. Wetzstein, S. Kato, T. Schrödel, R. Raskar, M. Zimmer, E. S. Boyden, and A. Vaziri, "Simultaneous Whole-Animal 3D Imaging of Neuronal Activity Using Light-Field Microscopy," *Nature Methods*, vol. 11, pp. 727 – 730, 05/18/2014 2014.

[6] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder, "Foveated 3D Graphics," *ACM Trans. Graph*, vol. 31, no. 6, pp. 164:1–164:10, 2012.

[7] X. Sun and A. Varshney, "Investigating Perception Time in the Far Peripheral Vision for Virtual and Augmented Reality," in *ACM Symposium on Applied Perception (SAP)*, ser. Perception. ACM, Aug 2018.

[8] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn, "Towards Foveated Rendering for Gaze-Tracked Virtual Reality," *ACM Trans. Graph*, vol. 35, no. 6, pp. 179:1–179:12, 2016.

[9] X. Meng, R. Du, M. Zwicker, and A. Varshney, "Kernel Foveated Rendering," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 1, no. 1, pp. 5:1–5:20, May 2018.

[10] H. Araujo and J. M. Dias, "An Introduction to the Log-Polar Mapping," in *Proc. II Workshop on Cybernetic Vision*. New York, NY, USA: ACM, Dec 1996, pp. 139–144.

[11] M. Antonelli, F. D. Igual, F. Ramos, and V. J. Traver, "Speeding Up the Log-Polar Transform With Inexpensive Parallel Hardware: Graphics Units and Multi-Core Architectures," *J. Real-Time Image Process*, vol. 10, no. 3, pp. 533–550, 2015.

[12] T. Ohshima, H. Yamamoto, and H. Tamura, "Gaze-Directed Adaptive Rendering for Interacting With Virtual Space," in *Proc. of the IEEE Virtual Reality Annual International Symposium*, 1996, pp. 103–110, 267.

[13] H. Hoppe, "Smooth View-Dependent Level-of-Detail Control and Its Application to Terrain Rendering," in *Proc. of IEEE Visualization*, Los Alamitos, CA, USA, 1998, pp. 35–42.

[14] L. Hu, P. V. Sander, and H. Hoppe, "Parallel View-Dependent Level-of-Detail Control," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 5, pp. 718–728, Sept 2010.

[15] M. Weier, M. Stengel, T. Roth, P. Didyk, E. Eisemann, M. Eisemann, S. Grogorick, A. Hinkenjann, E. Kruijff, M. Magnor, K. Myszkowski, and P. Slusallek, "Perception-Driven Accelerated Rendering," *Comput. Graph. Forum*, vol. 36, no. 2, pp. 611–643, 2017.

[16] Y. He, Y. Gu, and K. Fatahalian, "Extending the Graphics Pipeline With Adaptive, Multi-Rate Shading," *ACM Trans. Graph*, vol. 33, no. 4, pp. 142:1–142:12, 2014.

[17] K. Vaidyanathan, M. Salvi, R. Toth, T. Foley, T. Akenine-Möller, J. Nilsson, J. Munkberg, J. Hasselgren, M. Sugihara, P. Clarberg, T. Janczak, and A. Lefohn, "Coarse Pixel Shading," in *Proc. of High Performance Graphics*. Goslar, Germany: Eurographics Association, 2014, pp. 9–18.

[18] J. Ragan-Kelley, J. Lehtinen, J. Chen, M. Doggett, and F. Durand, "Decoupled Sampling for Graphics Pipelines," *ACM Trans. Graph*, vol. 30, no. 3, pp. 17:1–17:17, 2011.

[19] P. Clarberg, R. Toth, J. Hasselgren, J. Nilsson, and T. Akenine-Möller, "AMFS: Adaptive Multi-Frequency Shading for Future Graphics Processors," *ACM Trans. Graph*, vol. 33, no. 4, pp. 141:1–141:12, 2014.

[20] Bektas, Kenan and Cöltekin, Arzu and Krüger, Jens and Duchowski, Andrew T., "A Testbed Combining Visual Perception Models for Geographic Gaze Contingent Displays," in *Eurographics Conference on Visualization (EuroVis) - Short Papers*, E. Bertini, J. Kennedy, and E. Puppo, Eds. The Eurographics Association, 2015, pp. 67–71.

[21] A. Patney, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and D. Luebke, "Perceptually-Based Foveated Virtual Reality," in *ACM SIGGRAPH 2016 Emerging Technologies*. New York, NY, USA: ACM, 2016, pp. 17:1–17:2.

[22] N. T. Swafford, J. A. Iglesias-Guitian, C. Koniaris, B. Moon, D. Cosker, and K. Mitchell, "User, Metric, and Computational Evaluation of Foveated Rendering Methods," in *Proc. of the ACM Symposium on Applied Perception*. New York, NY, USA: ACM, 2016, pp. 7–14.

[23] M. Stengel, S. Grogorick, M. Eisemann, and M. Magnor, "Adaptive Image-Space Sampling for Gaze-Contingent Real-Time Rendering," *Computer Graphics Forum*, vol. 35, no. 4, pp. 129–139, 2016.

[24] S. Friston, T. Ritschel, and A. Steed, "Perceptual rasterization for head-mounted display image synthesis," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019. [Online]. Available: https://doi.org/10.1145/3306346.3323033

[25] E. Turner, H. Jiang, D. Saint-Macary, and B. Bastani, "Phase-Aligned Foveated Rendering for Virtual Reality Headsets," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. Los Alamitos, CA, USA: IEEE Computer Society, mar 2018, pp. 1–2. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/VR.2018.8446142

[26] P. Lungaro and R. Sjöberg and A. J. F. Valero and A. Mittal and K. Tollmar, "Gaze-Aware Streaming Solutions for the Next Generation of Mobile VR Experiences," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1535–1544, April 2018.

[27] M. Koskela, A. Lotvonen, M. Mäkitalo, P. Kivi, T. Viitanen, and P. Jääskeläinen, "Foveated Real-Time Path Tracing in Visual-Polar Space," in *Eurographics Symposium on Rendering - DL-only and Industry Track*, T. Boubekeur and P. Sen, Eds. The Eurographics Association, 2019.

[28] M. Weier, T. Roth, A. Hinkenjann, and P. Slusallek, "Foveated depth-of-field filtering in head-mounted displays," *ACM Trans. Appl. Percept.*, vol. 15, no. 4, Sep. 2018. [Online]. Available: https://doi.org/10.1145/3238301

[29] M. Levoy and P. Hanrahan, "Light Field Rendering," in *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: ACM, 1996, pp. 31–42.

[30] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," in *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. New Orleans, LA, USA: ACM, 1996, pp. 43–54.

[31] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, "Plenoptic Sampling," in *Proc. of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 307–318.

[32] R. Ng, "Fourier Slice Photography," *ACM Trans. Graph*, vol. 24, no. 3, pp. 735–744, 2005.

[33] D. Lanman and D. Luebke, "Near-Eye Light Field Displays," in *ACM SIGGRAPH 2013 Emerging Technologies*.   New York, NY, USA: ACM, 2013, pp. 11:1–11:1.

[34] F.-C. Huang, K. Chen, and G. Wetzstein, "The Light Field Stereoscope: Immersive Computer Graphics Via Factored Near-Eye Light Field Displays With Focus Cues," *ACM Trans. Graph*, vol. 34, no. 4, pp. 60:1–60:12, 2015.

[35] J. Zhang, Z. Fan, D. Sun, and H. Liao, "Unified Mathematical Model for Multilayer-Multiframe Compressive Light Field Displays Using LCDs," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.

[36] 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*, vol. 6, pp. 2170–2180, 2018.

[37] Q. Sun, F.-C. Huang, J. Kim, L.-Y. Wei, D. Luebke, and A. Kaufman, "Perceptually-Guided Foveation for Light Field Displays," *ACM Trans. Graph*, vol. 36, no. 6, pp. 192:1–192:13, 2017.

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

[39] R. S. Weinstein, M. R. Descour, C. Liang, G. Barker, K. M. Scott, L. Richter, E. A. Krupinski, A. K. Bhattacharyya, J. R. Davis, A. R. Graham, M. Rennels, W. C. Russum, J. F. Goodall, P. Zhou, A. G. Olszak, B. H. Williams, J. C. Wyant, and P. H. Bartels, "An Array Microscope for Ultrarapid Virtual Slide Processing and Telepathology. Design, Fabrication, and Validation Study," *Human Pathology*, vol. 35, no. 11, pp. 1303 – 1314, 2004.

[40] B. Wilt, L. Burns, E. T. W. Ho, K. Ghosh, E. Mukamel, and M. Schnitzer, "Advances in Light Microscopy for Neuroscience," *Annual Review of Neuroscience*, pp. 435–506, 9.

[41] B. Sheng, P. Li, Y. Jin, P. Tan, and T. Lee, "Intrinsic image decomposition with step and drift shading separation," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.

[42] Q. Chen and V. Koltun, "A simple model for intrinsic image decomposition with depth cues," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 241–248.

[43] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[44] W. Feng, Y. Yang, L. Wan, and C. Yu, "Tone-mapped mean-shift based environment map sampling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 9, pp. 2187–2199, Sep. 2016.

Xiaoxu Meng Xiaoxu Meng is a Ph.D. candidate and a graduate research assistant in Computer Science Department at the University of Maryland, College Park, advised by Prof. Amitabh Varshney. She holds a M.S. from University of Maryland, College Park and a B.S. from Shanghai Jiao Tong University. Her research focuses on foveated rendering (Kernel Foveated Rendering) and 3D mesh reconstruction from 2D images using deep learning.

Ruofei Du Ruofei Du is a Research Scientist at Google and works on creating novel interactive technologies for virtual and augmented reality. He holds a Ph.D. in Computer Science from the University of Maryland, College Park, advised by Prof. Amitabh Varshney. His research focuses on interactive 3D technologies, spanning mixed-reality social platforms (Geollery.com), 4D video-based rendering (Montage4D), foveated rendering, and deep learning in graphics. He served as the Associate Chair on the Late Breaking Work committee of ACM CHI 2020. He chaired the "Making the Virtual Physical" session at ACM CHI 2019 and the "Augmented and Virtual Reality" session at ACM Web3D 2019. He has received the Best Paper Award in ACM Web3D 2016 for Social Street View and the Best Student Poster Award in ACM l3D 2018.

Joseph F. JaJa Dr. Joseph JaJa is currently serving as the Interim Chair of the Department of Electrical and Computer Engineering at the University of Maryland. He received his Ph.D. degree in Applied Mathematics from Harvard University after which he joined Penn State University as Assistant Professor of Computer Science. He moved to the University of Maryland in 1983 with a joint appointment in ECE and the Institute for Advanced Computer Studies (UMIACS). He served as Director of UMIACS from 1994-2004. He also served as Interim Vice President and Chief Information Officer of the University of Marylands Office of Information Technology from 2010-2011, Director of Cyberinfrastructure of the National Socio-Environmental Synthesis Center (SESYNC) from 2012-2014,and as co-Technical Director of the NIST FFRDC in Cybersecurity during 2014-2015. Prof. JaJas current research interests involve big data, statistical machine learning, computational neuroscience, and high-performance computing. He has published extensively in several areas, including parallel and distributed computing, combinatorial optimization, VLSI design, and data-intensive computing. He has received numerous awards including the IEEE Fellow Award, the ACM Fellow Award, the Internet2 IDEA Award, as well as several best paper awards.

Amitabh Varshney Amitabh Varshney is Dean of the College of Computer, Mathematical, and Natural Sciences and a Professor of Computer Science at the University of Maryland, College Park. In his research, Varshney explores applications of high-performance computing and visualization in engineering, science, and medicine. He has worked on a number of research areas including visual saliency, summarization of large visual datasets, and visual computing for big data. He is currently exploring applications of virtual and augmented reality in several applications, including healthcare and telemedicine. Varshney is a Fellow of IEEE, an inaugural Fellow of the IEEE Visualization Academy, and has served in various roles on the IEEE Visualization and Graphics Technical Committee, including as its Chair (2008–2012). He also received the IEEE Visualization Technical Achievement Award in 2004.