Image Synthesis With Efficient Defocus Blur for Stereoscopic Displays

YI-CHUN CHEN AND TIAN-SHEUAN CHANG, (Senior Member, IEEE)
Department of Electronics Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C
Corresponding author: Yi-Chun Chen (onegroup.ee03g@g2.nctu.edu.tw)

ABSTRACT Image synthesis for stereoscopic displays shall carefully control its depth of field to enhance perception while reduce visual fatigue due to accommodation-vergence mismatch. Existing approaches suffer from either lower quality due to occluding contour error, or high computational complexity. To meet above demands, this paper proposes a perceptual oriented defocus blur that reduces complexity with the depth layer processing and improves quality with transparency degree oriented superposition and edge enhancement. The simulation results show that the proposed approach has better quality than the conventional defocus blur methods and achieves similar quality compared to the deep learning based methods but with much lower complexity.

INDEX TERMS Focusing, image filtering, image reconstruction, image texture, reconstruction algorithms, visual effects.

I. INTRODUCTION
Defocus blur as an important cue of depth perception has been widely used in image synthesis or rendering to enhance depth perception of stereoscopic 3D displays [1]–[5] and reduce the visual discomfort and fatigue [6], [7] due to mismatch between accommodation and vergence. The vergence is a depth perception process for binocular vision that brain compares and processes two images with slightly different distances from two eyes to form a single stereo image. The depth perception also happens in the monocular vision that the eye can change its focus from distant to near objects and vice versa, denoted as accommodation. For stereoscopic 3D displays, eyes will try to converge on created virtual object due to vergence, but they will also try to locate on the surface of displays due to the accommodation. Such mismatch can be inhibited by limit the depth of field (DoF) in the scenes [2]. DoF is caused by finite aperture where light rays cannot focus to a point but in a circle, denoted circle of confusion (CoC), and thus results in blur effect to limit clear vision range [21], [26]. Therefore, the artificial defocus blur is a popular method to control DoF that keeps the pixels inside DoF clear and the pixels outside DoF blur [8]–[12] while constrains DoF within a comfortable viewing zone [2], [3], [13]. However, the comfortable zone is highly user dependent, which demands an adaptable defocus blur method to adjust DoF for each kind of stereoscopic display.

Another approach is to use novel optical elements and joint algorithm design on rendering and optimization [14]–[17]. But they still suffer two problems: synthetically rendered blur instead of the optical blur, and high computational complexity due to various blur filter size. The synthetically rendered blur only reduces the texture detail in the blur area and leaves DoF clear, which results in unnatural boundary error. To simulate real vision, blur of an occluding contour should help determine whether an adjoining blurred region perceived as near or far, which cannot be achieved by blur filters alone because the blur filter processing does not indicate the sign of relative distance in any obvious way [18], [19]. Without the correctly rendered defocus blur, these displays create incorrect cues that diminish depth perception [19], [20].

On the other hand, the complexity computation problem of enhancement defocus blur quality is not only in conventional defocus filter approaches but also in learning based approaches. The defocus blur filter needs to be changed to fit regions with complex occlusion such as when in-focus and out-of-focus objects contribute to the same pixel, and to fit regions with different depth. This prevents efficient separable filter formulations and greatly impacts performance. Although end to end deep learning based approaches [39]–[46] can overcome occluding contour error and complete the sign of depth in defocus blur processing,
the huge training database and heavy convolution computational complexity increase the processing cost. To address the tradeoff between quality and computational cost, this paper introduces an efficient defocus blur algorithm to control DoF easily, which can significantly improve the reconstruction quality similar to the result of deep learning approaches with much lower complexity. To reduce computational complexity, our approach partitions an image into layers according to their depth values, and thus can share the same defocus filter in each layer. Furthermore, we accumulate each layer result to propagate the blur effect instead of buffer them all to reduce the buffer size. To improve quality, we classify image pixels into three transparency types after defocus blur filter, and reconstruct the defocus image with back to front superposing order to resolve complex occluded reconstruction cases and reconstruct the sign of depth for perceptual enhancement. The final output is further enhanced on the edges to imitate real depth perception. The simulation results show better quality than conventional deep focus methods or similar quality to deep learning based approaches.

This rest of the paper is organized as follows. Section II first reviews the related work. Then Section III presents our proposed approach. The experimental results and comparisons are shown in Section IV. Finally, Section V concludes this paper.

II. RELATED WORK
The defocus blur is a popular technique to increase depth perception in photography and image processing [22]–[24], which can fill in the parts of visual space where disparity is imprecise [25]. Rendering defocus blur consists of two types of algorithms: conventional image processing approach and the deep learning based approach as discussed below.

A. CONVENTIONAL DEFOCUS RENDERING
Reference [27] first proposes a post-processing to simulate defocus blur from a single image with color and depth (RGB-D). These types of approaches ignore transparency information and thus often produce artifacts. However, they are commonly used in real-time rendering due to their high performance and simplicity [28]–[31].

Reconstruction defocus blur based on frequency analysis and sheared filtering tightly bounds the sheared frequency spectrum of the light field. The spectrum of the light field is proved bounded by the minimum and maximum depth values in [32]. They determine a minimum sampling rate and a reconstruction filter for light field rendering. The light field frequency bounds become narrower when the scene is partitioned into depth layers, and hence fewer samples are required for reconstruction. However, they do not address occlusion between layers and ignore transparency degree in their analysis. As observed by [33], most sheared filtering approaches revert to a less effective axis-aligned filter near occlusion boundaries, producing a noisy result. Reference [34] further proposes independent depth layer filters and then combine them together by including inter-layer transparency. But frequency analysis still increases total cost.

Some researches apply defocus effects before compositing the final image [35]–[37]. Reference [35] uses per-pixel arrays for collection, progressive filtering and alpha blending to simulate large and near-field blurs due to foreground is blurry and transparent. Reference [36] renders defocus blur by ray tracing into a layered color and depth buffer, but requires depth peeling to render the multilayer frame buffer and cause visible errors due to some occluded contours. A similar approach in Reference [37] renders some sharp layers, splats and gathers samples on image plane to simulate defocus blur for a particular focus depth. Reference [38] develops an interesting fixed viewpoint volumetric display that maintains view-dependent effects such as occlusion, specular reflection. They introduces a “linear blending” method to superimpose different decomposed images layers. These kinds of approaches need more computational cost including hardware support to store different layer’s result, and more computational time to get high performance.

In summary, to faithfully rendering defocus blur in current approaches are either prohibitively computationally expensive or fail to approximate the blur effects at partially occluded regions, and no research considers the effect of sign of depth while rendering.

B. DEFOCUS RENDERING BASED ON LEARNING-BASED APPROACH
With the popularity of deep learning, more and more researches render or synthesize images by learning-based approaches. A span screen-space image rendering method includes defocus blur [40] that uses a U shape neural networks [39] and end to end training. Reference [43] uses deep networks to synthesize novel views from images with wide baselines. Reference [44] takes advantage of the texture structure of the epipolar plane image for light field view synthesis. Reference [45] demonstrates view reconstruction in a light field camera using only the four corner views, with follow up work [12] using only the central image. This method uses two sequential networks in a pipelined approach that the first network predicts disparity and the second network predicts color images from this disparity and pre-warped versions of the input images. Reference [46] introduces the high quality DeepFocus deep neural net that is the state of the art method for virtual simulated focus. They use an end-to-end pipeline from RGB-D images to accommodation. Meanwhile, they adopt a coded-aperture camera as the input device. Although learning based approaches can meet high quality requirement, they demand large training data and expensive computational cost.

III. PROPOSED METHOD
Fig. 1 shows the proposed algorithm, which consists of a depth layer processing step to compute per layer defocus blur, a pixel processing step to superpose per pixel blur result from
different depth layers based on the pixel types, and finally a refinement step to refine the supposed images. The inputs are one image of a scene and corresponding depth map. The depth map is found by depth camera or stereo matching. The stereo matching finds the corresponding points in both left and right views that are projections of the same scene with different view angles. The difference of coordinate points between projections of left and right view can be converted to the depth values.

Contrast to the traditional image based approach, the proposed approach adopts a depth layer processing because pixels in the same depth level share the same size of CoC, and thus avoid the convolution kernel variation problem. This results in a regular processing flow because of only one kind of the convolution kernel at a time, which is also suitable for hardware design.

In the first depth layer processing step, we first separate image I to several layers as I_l according to its depth level l. Then, we simulate the defocus blur by convolving I_l with the corresponding convolution kernel G_l. This kernel uses the diameter of CoC as its size and 2D Gaussian distribution as its point weighting to represent transparency degree of each pixel for our proposed algorithm.

Based on above convolution results, the pixel processing step classifies each pixel into three transparency types: replacement, mixture and linkage, and superposes them accordingly. The replacement type is a pixel to be replaced by the convolutional result of the current depth layer. The mixture type pixels will add their convolutional results to last layer’s result to simulate illumination accumulation. The linkage type is specifically for a case of solid objects across multiple depth layers. In this case, the defocus blur result of this object is not transparent enough to see the background. This object would be segmented to different depth layer in the previous step. Therefore, we have to reconnect them together to recover the solid surface, which is indicated by the feedback arrow from mixture to linkage block as in Fig. 1 because this type needs results from different layers.

In the final refinement step, we first normalize illumination of each pixel after the superposition step because the illumination of each pixel will be different due to the accumulation of the irregular convolutional result. Then we refine edges between foreground and background with linear interpolation to simulate heavy defocus blur due to mixed transparent foreground and background. The details of each step will be discussed below.

A. DEPTH LAYER CONVOLUTION BASED ON COC

This subsection shows how to derive the convolutional filter for defocus blur. To recover the real blur effect, one common way is to use a point spread function, which is defined by its range and response intensity. The range is relative to diameter of CoC in a human eye or camera system. The response intensity usually represents by a bell curve distribution such as a 2-D Gaussian function. For a 2-D Gaussian function to approximate real blur response, its standard deviation (σ) determines the degree of de-focus blur that is proportional to the CoC diameter [47]. Thus, in the following, we will first derive the diameter of CoC to decide the filter size, and then uses this diameter as a range to obtain the corresponding Gaussian filter.

1) SIZE OF THE CONVOLUTIONAL KERNEL

Fig. 2 demonstrates how the blur effect depends on the diameter of CoC in a human eye, and how to simulate equivalent effect of defocus blur on a display. In which, the diameter of CoC, C in user’s eye, is relative to the distance of the focal plane and the object in a scene on the upper side of Fig. 2. The optics of an eye can be modeled by the thin-lens equation [48]:

\[
\frac{1}{v_f} + \frac{1}{D_f} = \frac{1}{f}
\]  

(1)

where \(f\) is the focal length, \(v_f\) is the distance between lens and retina, and \(D_f\) is the distance from the fixation point to lens. As shown in Fig. 2 (a), the retinal blur will occur when fixating at a point F since a peripheral point P will not come into perfect focus. In this case, the blur strength can be characterized by the CoC based on the thin-lens model:

\[
C_{eye} = A \cdot v_f \left| \frac{1}{D_f} - \frac{1}{D_p} \right|
\]  

(2)

where \(C_{eye}\) is the diameter of CoC on the retina, \(D_p\) is the distance of the peripheral point and A is the pupil diameter.
deviation of the Gaussian filter kernel can be defined by (5)

\[
\sigma = \frac{C_{\text{pixel}}}{\sqrt{2\ln2}}
\]

The intensity of the filter kernel is defined by (6) where the mean value is zero and the two dimensions of the filters are independent to each other due to symmetrical and uncorrelated features of the point spread function.

\[
f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)
\]

3) DEPTH LAYER CONVOLUTION

Fig. 3 shows how to convolve different depth layers with different filter kernels, where \(I\) is input image, and \(I_1, I_2\) are part of the image separated according to their depth values indicated by binary index maps, \(B_1\) and \(B_2\). The output, \(O_1\) and \(O_2\), are the convolutional results of different image parts. \(W_1\) and \(W_2\) are the convolutional results of the binary index map, which represents the distribution intensity of each pixel on the binary index map, or it can be understood as weighting of illumination before the layer superposition. We will discuss how to superposition each pixel in the next section.

In summary of this section, to recover the defocus blur various based on the depth map, we first define the diameter of CoC by diopter difference as in (4), and use this parameter to derive the standard deviation of the Gaussian filter as in (6). Then this depth specific filter is applied to dedicated images parts based on their depth values.

B. PIXEL CLASSIFICATION AND SUPERPOSITION

To recover sign of depth in defocus blur, our proposed method superposes the defocus result of each depth layer from the farthest to nearest with the nearer layers on farther ones. How to superpose or merge them depends on their transparency which is defined by the light gathering power. If the light gathering power is strong enough, image should focus in one or a few pixels, and those pixels are opaque and replaced by the nearer depth ones. The farther depth one is always too small to replace the image pixels at the farther depth. Based on above observations, we classify pixels to three transparency types: replacement, mixture and linkage, according to the corresponding the light gathering power at the depth layer \(l\), denoted by \(W_l\).

\[
W_{A_l} = \begin{cases} 
W_l, & W_l > \text{Thr1} \\
W_{l-1} + W_l, & W_{l-1} + W_l > \text{Thr2} \\
m \ast W_{A_{l-1}} + W_l, & \text{others}
\end{cases}
\]

\[
O_{A_l} = \begin{cases} 
O_l, & W_l > \text{Thr1} \\
O_{l-1} + O_l, & W_{l-1} + W_l > \text{Thr2} \\
m \ast O_{A_{l-1}} + O_l, & \text{others}
\end{cases}
\]
The superposition method for the defocus image is shown in (7) and (8), where $WA_1$ and $OA_1$ are weighting accumulation and output image accumulation, respectively. We set a weighting threshold, $Thr_1$, between zero and one to find the replacement type pixels. If $W_1$ of a pixel is larger than $Thr_1$, its weighting and output value will be replaced by new values, $W_1$ and $O_1$, respectively. On the other hand, if $W_1$ is smaller than $Thr_1$, the pixel may be mixture or linkage type depending on the sum of $W_1-1$ and $W_1$. If the sum is larger than another threshold, $Thr_2$, it is a linkage type where its weighting and value will be replaced by $W_1-1+W_1$ and $O_1-1+O_1$, respectively. In which, the $Thr_2$ value is linearly increased from a defined minimum value (larger than zero) to $Thr_1$ from farthest to nearest depth. All the other case belongs to the mixture type. In such case, the nearer image pixels will be diffused on the farther one and be mixed together. The mix method uses a decayed weighting for further layers accumulation, $m \times WA_{l-1}$ because the foreground ones are more conspicuous than the background ones. In which the constant $m$ is a value between zero to one to keep pixel relative order for better performance. The same operation is also applied to the output accumulation image $OA_1$.

Take Fig. 3 as an example. In which, we assume the depth layer 1 (including $I_1$, and $B_1$) is farther than the layer 2 (including $I_2$, and $B_2$). Therefore, we first calculate $W_1$ and $O_1$ and save them to the weighting accumulation $WA_1$ and output image accumulation, $OA_1$. Second, we calculate $O_2$ and $W_2$ and then find replacement type pixels by comparing them with $Thr_1$. For other pixels beyond the replacement type, we will add $W_2$ to $W_1$ and compare the results with $Thr_2$ to distinguish mixture and linkage type pixels. Based on above pixel types, we compute new accumulation result, $WA_2$ and $OA_2$, by (7) and (8).

C. THE REFINEMENT STEP

The post refinement step includes two parts, illumination normalization and interpolation.

1) ILLUMINATION NORMALIZATION

After above convolution and superposition for all layers, the accumulation weighting will be totally different for each pixel. However, since these weightings indicate the light gathering power in our supposition, the different weighting accumulation leads to different illumination. To overcome this problem, we divide the final result $OA_{nearest}$ by $WA_{nearest}$ to normalize illumination and recover the same illumination as the input image. For the example in Fig. 3, the illumination normalization result is $OA_2$ divided by $WA_2$.

2) INTERPOLATION

Above steps recover blur effect for image, but neglect the diffraction effect since we only recover the aberration effect in the defocus blur. The most obvious diffraction effect in the defocus blur is that the foreground is too transparent to cover the background. However, such case does not exist in the input image, which needs to be restored by the Laplace interpolation method.
This leads to the question of which pixels need this interpolation, and how to find them. Fortunately, we can solve this question by our weighting accumulation and diameter of CoC. For such case, the weighting has a huge drop due to no background data during accumulation. However, there are other conditions leading to the weighting drop but does not need interpolation. To automatically differentiate out the area for interpolating, we also consider the CoC of each pixel because the interpolation area belongs to foreground of the focal plane. Therefore, we set the interpolation area by finding the intersection of big weighting drop and foreground. In which, the foreground pixels can be found by (9) which is similar to (4) but without absolute value computation. In which, the pixels with negative $C_{\text{pixel}}$ are foreground.

$$C_{\text{pixel}} = T \left( \frac{1}{D_f} - \frac{1}{D_p} \right)$$  \hspace{1cm} (9)

After the interpolation region is defined, we use the mean value theorem to solve the Laplace equation with harmonic functions to restore the missing background data. We use the above normalized output $O_n$, set interpolation region as zero, and take one pixel from its north, east, and south, and west neighbor respectively for all the pixels in the interpolation region to compute the partial differential equation for vertical and horizontal direction. We also transform the partial differential operators to a sparse matrix $A$ that becomes:

$$A \cdot O = O_n$$  \hspace{1cm} (10)

The interpolation output $O$ can be calculated by

$$O = A^{-1} \cdot O_n$$  \hspace{1cm} (11)

The interpolation step creates the mixture image between the foreground and background because the boundary of the interpolation region is surrounded by our simulated defocus blur result of each layer.

### IV. EXPERIMENTAL RESULT

The proposed method has been implemented and tested on the image dataset from [51] with comparison to other approaches. For quantitative evaluation, we use PSNR for pixel-wise accuracy, and structural similarity index (SSIM) [50] for perceptual image quality. The image datasets from [51] provides images and their depth map and multifocal ground truth defocus images. In which, the ground truth images are generated by the off-line accumulation buffer method implemented in Unity.

Fig. 4 and Table 1 show the simulation results on the test image ‘living room’ and comparisons with the conventional method [38] and learning based method, DeepFocus [46]. Compared to [38], the proposed method has much better results on PSNR, SSIM and resulted images. Compared to the state-of-the-art learning based method [46], the proposed method has similar SSIM results but with better PSNR for images on the farthest (0.1D) and nearest (2.2D) depth.

Fig. 5 shows the comparison on another test image “robot” with zoom in on occluded contours. The zoom in area of

---

**TABLE 1. Simulation results for the test image 'living room'.**

|         | PSNR (dB) | SSIM  |
|---------|-----------|-------|
| Akeley et al. [38] | 39.1 | 0.984 | 0.984 |
| DeepFocus [46]    | 42.7 | 0.994 | 0.997 |
| Proposed          | 43.2 | 0.994 | 0.998 |
FIGURE 5. Zoom in for different focus point with test image "robot". This test image has more broad focus range from 0.1D to 3.0D, and (c) is the zoom in area of input image, and (d) and (f) are the ground truth and our results are (e), (g).

FIGURE 6. Ablation experimental results of the zoom in the area of robot hand for the input image "robot" after our 'Depth layer processing step'. All are focusing on furthest place (0.1D). (a) is the result of 'Depth layer processing step' without any post processing. (b) to (f) are results of different combinations.

FIGURE 7. Simulation results for the random scene case. Each scene contains 40 defocus images with focal distances between 0.1D and 4.0D. (a) an input example, (b) our result of focusing on background, and (c) focusing on foreground.

Table 2: Ablation experimental results for the test image ‘robot’.

|                        | 0.1D | 0.5D | 1.0D | 1.5D |
|------------------------|------|------|------|------|
| Classification         | ✓    | ✓    | ✓    | ✓    |
| and superposition      | ✓    | ✓    | ✓    | ✓    |
| Illumination           | ✓    | ✓    | ✓    | ✓    |
| normalzation           | ✓    | ✓    | ✓    | ✓    |
| Interpolation          | ✓    | ✓    | ✓    | ✓    |
| PSNR (dB)              | 33.4 | 37.8 | 36.1 | 39.3 |
| SSIM                   | 0.996 | 0.995 | 0.995 | 0.995 |

Table 3: Quantitative comparison with different algorithm for defocus blur rendering for test image ‘robot’.

|                 | PSNR (dB) | SSIM |
|-----------------|-----------|------|
| Unity [53]      | 25.3      | 0.887|
| Nuke [52]       | 40.1      | 0.993|
| Nalbach et al. [40] | 37.0      | 0.989|
| DeepFocus [46]  | 45.6      | 0.999|
| Proposed        | 44.5      | 0.997|

We have also analyzed the three parameters in the superposition stage, Thr1, Thr2, and m, for replacement, linkage, and mixture respectively. According to our hypothesis the
weighting sum should be between 0 and 1, we simulate these three parameters in this range for test image ‘robot’ in Fig. 8. In Fig. 8 (a) to (c), we calculate the average quality for all focusing depth while fixing two parameters and one is variable, and the best setting for three parameters are nearby but does not equal to 1. In general, different test images should have different settings. Take different focusing depth for example, the best setting are totally different for three parameters in Fig 8 (d) to (f). However, the difference is quite small and best settings are around 0.9, even for different test images the analysis result is quite similar and stable. Thus, we choose 0.94, 0.8, and 1 as \( \text{Thr1} \), \( \text{Thr2} \), \( m \) respectively for all focusing depth.

Our result is much better than Nalbach et al. [40] and offline DoF rendering software, Nuke [52]. Compared to the learning based approach, our result has similar performance as those based U-Net [39], Dilated-Net [41] but is inferior compared to DeepFocus [46] as shown in Table 3. The difference is less than one dB in PSNR and 0.01 in SSIM. The DeepFocus is built specially for creating defocus blur with computation intensive neural network, and its performance almost better than the proposed one by one dB in PSNR. On the contrary, our proposed approach needs much lower computational complexity, but provide almost high quality as DeepFocus without training, which will be beneficial for low cost real time implementation.

V. CONCLUSION

This paper proposes an efficient defocus blur method for image synthesis to increase depth perception or reduce visual fatigue on stereoscopic displays. Our depth layer based processing can recover sign of depth in the defocus blur correctly and also has low computational complexity and buffer requirement with layer based accumulation. The results show that our approach can solve occluding contour error through sign of depth to achieve better quality than conventional defocus blur methods and similar quality than the learning based approach with much lower complexity.

REFERENCES

[1] A. Smolic and P. Kauff, “Interactive 3-D video representation and coding technologies,” Proc. IEEE, vol. 93, no. 1, pp. 33–36, Jan. 2005.
[2] M. Lambooij, W. J. IJsselsteijn, M. Fortuin, and I. Heynderickx, “Visual discomfort and visual fatigue of stereoscopic displays: A review,” J. Clin. Imag. Sci. Technol., vol. 53, no. 3, pp. 1–14, May 2009.
[3] W. J. Tam, F. Speranza, S. Yano, K. Shimono, and H. Ono, “Stereoscopic 3D-TV: Visual comfort,” IEEE Trans. Broadcast., vol. 57, no. 2, pp. 335–346, Jun. 2011.
[4] K. Lee, A. K. Moorthy, S. Lee, and A. C. Bovik, “3D visual activity assessment based on natural scene statistics,” IEEE Trans. Image Process., vol. 23, no. 1, pp. 450–465, Jan. 2014.
[5] H. Oh and S. Lee, “Visual presence: Viewing geometry visual information of UHD S3D entertainment,” IEEE Trans. Image Process., vol. 25, no. 7, pp. 3358–3371, Jul. 2016.
[6] J. Park, H. Oh, S. Lee, and A. C. Bovik, “3D visual discomfort predictor: Analysis of disparity and neural activity statistics,” IEEE Trans. Image Process., vol. 24, no. 3, pp. 1101–1114, Mar. 2015.
[7] T. Kim, S. Lee, and A. C. Bovik, “Transfer function model of physiological mechanisms underlying temporal visual discomfort experienced when viewing stereoscopic 3D images,” IEEE Trans. Image Process., vol. 24, no. 11, pp. 4335–4347, Nov. 2015.
[8] Y. Okada, K. Ukai, J. S. Wolffsohn, B. Gilmartin, A. Iijima, and T. Bando, “Target spatial frequency determines the response to conflicting defocus- and convergence-driven accommodative stimuli,” Vis. Res., vol. 46, no. 4, pp. 475–484, Feb. 2006.
[9] L. Leroy, P. Fuchs, and G. Moreau, “Visual fatigue reduction for immersive stereoscopic displays by disparity, content, and focus-point adapted blur,” IEEE Trans. Ind. Electron., vol. 59, no. 10, pp. 3998–4004, Oct. 2012.

[10] Y. J. Jung, H. Sohn, S.-I. Lee, F. Speranza, and Y. M. Ro, “Visual importance- and discomfort region-selective low-pass filtering for reducing visual discomfort in stereoscopic displays,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 8, pp. 1408–1421, Aug. 2013.

[11] T. Shibata, J. Kim, D. M. Hoffman, and M. S. Banks, “The zone of comfort: Predicting visual discomfort with stereo displays,” J. Vis., vol. 11, pp. 1–28, Jul. 2011.

[12] G. Kramida, “Resolving the vergence-accommodation conflict in head-mounted displays,” IEEE Trans. Vis. Comput. Graphics, vol. 22, no. 7, pp. 1912–1931, Jul. 2016.

[13] D. Dunn, C. Tippett, K. Torell, P. Kellnhofer, K. Aksit, P. Didyk, K. Myszkowski, D. Luebke, and H. Fuchs, “Wide field of view varifocal near-eye display using see-through deformable membrane mirrors,” IEEE Trans. Vis. Comput. Graphics, vol. 23, no. 4, pp. 1322–1331, Apr. 2017.

[14] F. C. Huang, K. Chen, and G. Wetzstein, “The light field stereoscope: Immersive computer graphics with factored near-eye light field displays with focus cues,” ACM Trans. Graph., vol. 34, no. 4, pp. 60, Jul. 2015.

[15] A. K. M. Sami, K. Selgrad, C. Reintges, D. Penk, P. Wagner, and M. Stamminger, “Real-time depth of field using multi-layer filtering,” in Proc. 19th Symp. Interact. 3D Graph. Games, 2015, pp. 121–127.

[16] S. Lee, E. Eisenmann, and H.-P. Seidel, “Real-time lens blur effects and focus control,” ACM Trans. Graph., vol. 29, no. 4, pp. 7–65, 2010.

[17] K. Lei and J. F. Hughes, “Approximate depth of field effects using few samples per pixel,” in Proc. ACM SIGGRAPH Symp. Interact. 3D Graph. Games, 2013, pp. 119–128.

[18] K. Akeley, S. J. Watt, A. R. Girshick, and M. S. Banks, “A stereo display prototype with multiple focal distances,” ACM Trans. Graph., vol. 23, no. 3, pp. 804–813, Aug. 2004.

[19] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Assist. Intervent., 2015, pp. 234–241.

[20] O. Naibach, E. Arabadzhyska, D. Mehta, H.-P. Seidel, and T. Ritschel, “Deep shading: Convolutional neural networks for screen space shading,” Comput. Graph. Forum, vol. 36, no. 4, pp. 65–78, Jul. 2017.

[21] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in Proc. Int. Conf. Learn. Represent., 2016, pp. 1–4.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[23] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, “Deepstereo: Learning to predict new views from the world’s imagery,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Dec. 2015, pp. 5515–5524.

[24] W. Wu, P. Llull, I. Tosic, N. Bedard, K. Berkner, and N. Balram, “Content-adaptive focus configuration for near-eye multi-focal displays,” in Proc. IEEE Int. Conf. Multimedia Expo. (ICME), Jul. 2016, pp. 1–6.

[25] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, “Learning-based view synthesis for light field cameras,” ACM Trans. Graph., vol. 35, no. 6, p. 15, 2016.

[26] L. Xiao, A. Kaplanany, A. Fix, M. Chapman, and D. Lannan, “DeepFocus: Learned image synthesis for computational displays,” ACM Trans. Graph., vol. 37, no. 6, pp. 1–13, Jan. 2019.

[27] S. Zhuo and T. Sim, “Defocus map estimation from a single image,” Pattern Recognit., vol. 44, no. 9, pp. 1852–1858, Sep. 2011.

[28] A. P. Pentland, “A new sense for depth of field,” IEEE Trans. Pattern Anal. Mach. Intell., vols. PAMI–9, no. 4, pp. 523–531, Jul. 1987.

[29] A. C. Bovik, The Essential Guide to Image Processing. New York, NY, USA: Academic, 2009, pp. 498–500.

[30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[31] (2018). DeepFocus. [Online]. Available: https://research.fb.com/publications/deepfocus-siggraph-asia-2018/

[32] (2018). Nuke. [Online]. Available: https://www.foundry.com/products/nuke

[33] (2005). Unity Technologies. [Online]. Available: http://unity3d.com

YI-CHUN CHEN received the B.S. degree in electronics engineering from Chung Yuan Christian University, Taoyuan, in 2005, and the M.S. degree from the National Taiwan University of Technology, Taipei City, Taiwan, in 2009. He is currently pursuing the Ph.D. degree with National Chiao Tung University, Hsinchu, Taiwan. His current research interests include image processing, stereoscopic vision, system-on-a-chip design, and VLSI signal processing.

TIAN-SHEUAN CHANG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1993, 1995, and 1999, respectively. From 2000 to 2004, he was the Deputy Manager of Global Unichip Corporation, Hsinchu. In 2004, he joined the Department of Electronics Engineering, NCTU, where he is currently a Professor. In 2009, he was a Visiting Scholar with IMEC, Belgium. His current research interests include system-on-a-chip design, VLSI signal processing, and computer architecture.

Dr. Chang has been actively involved in many international conferences as an organizing committee or technical program committee member. He has received the Excellent Young Electrical Engineer from the Chinese Institute of Electrical Engineering, in 2007, and the Outstanding Young Scholar from the Taiwan IC Design Society, in 2010.