IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 September 2023, accepted 16 October 2023, date of publication 23 October 2023, date of current version 27 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3326723

RESEARCH ARTICLE

Approximated Wavefront Composition for Computer-Generated Holograms Based on Tiny Logic Operations

TAKASHI NISHITSUJI[1], TOMOYOSHI SHIMOBABA[2], ATSUSHI SHIRAKI[2], AND TOMOYOSHI ITO[2]

[1]Faculty of Science, Toho University, Funabashi, Chiba 274-0851, Japan
[2]Graduate School of Engineering, Chiba University, Chiba 263-8522, Japan

Corresponding author: Takashi Nishitsuji (takashi.nishitsuji@is.sci.toho-u.ac.jp)

This work was supported in part by the Japan Society of Promotion Science under Grant 22H03616 and Grant 19H01097, and in part by the Tokyo Metropolitan University (TMU) Local 5G Support and Institute for Advanced Academic Research (IAAR) Support Program Chiba Uniersity.

ABSTRACT Accelerating the calculation of computer-generated holograms (CGHs) is an urgent requirement for realizing practical electro-holography for three-dimensional (3D) displays. Although many algorithms have been developed for fast CGH calculations, sufficient acceleration of the CGH calculations is yet to be achieved. Simplifying CGH calculations using the redundancy of a hologram while recording a 3D image is a promising way for CGH acceleration. That is, the feature that a hologram can reproduce the 3D image almost correctly even if it was recorded with deviations originating from the approximation, etc., can be used to perform the CGH calculations rapidly. Therefore, this study proposes an approximation method for trigonometric functions, which have a heavy computational load and are heavily used in the composition process of the wavefronts emitted from a virtual 3D object such as point-light sources. The proposed method substitutes sin or cos functions with tiny logic operations that are expected to have lower computational loads than the normal functions for fixed or floating precision numbers. Numerical and optical experiments showed that the proposed method can calculate CGH 2.20 times faster on a central processing unit (CPU) and 8.43 times faster on a graphics processing unit (GPU) than the normal implementation, while maintaining sufficient image quality. The proposed method is not only effective in CPU and GPU implementations but also is expected to make a significant contribution to future circuit implementations for CGH-specific computers.

INDEX TERMS Computer-generated hologram, electro-holography, three-dimensional display.

I. INTRODUCTION

Electro-holography is a promising technology for three-dimensional (3D) display because it can replay 3D images consistent with the visual cognitive principles of human perception. Several notable advances were made in the field of 3D displays with holographic technologies (e.g., near-eye display [1], [2], [3], contact-lens-embedded display [4], metasurface [5], [6]), practical 3D displays, namely, wide-viewing, large, and colored, still have not been realized because of many technological challenges. One of these challenges is the extremely high requirement of computational performance for calculating computer-generated holograms (CGHs), which are a digital medium of 3D images in electro-holography.

A CGH is calculated by simulating the light wave propagating from a digital 3D object to be displayed. The calculation of the light propagation is determined by the minimum elements of the 3D objects, e.g., polygon [7], [8], [9], wireframes [10], [11], and point-light source (PLSs) [12], [13], [14]. Further, CGH is also generated from the wavefront information obtained with special optics, e.g., a light-field

The associate editor coordinating the review of this manuscript and approving it for publication was Joewono Widjaja.

camera [15], [16]. Among them, the PLS is one of the most commonly used 3D models to generate CGH owing to the simplicity of its calculations. The proposed method focuses on CGH calculations from a 3D PLS object.

The PLS-based CGH calculation is a superposition of spherical waves emanating from the PLSs; thus, the calculation includes many square-roots and trigonometric functions, both of which are computationally more intensive than the basic functions (e.g., addition and multiplication). To reduce the computational load, approximations should made during CGH calculations; for example, Fresnel approximation [17] and recurrence relation [18], [19], [20] are both used for square root calculation, and table look-up [21], [22] and simple approximation [23], [24] are applied both for sine and cosine functions. Those approximations are employed in practical calculation systems such as dedicated computers [25], [26], [27], [28] or graphics processing units (GPUs) [16], [29].

This study focused on simplifying the computation of trigonometric functions in CGH calculation, considering that future implementation will be performed in dedicated circuits. Generally, holography is a method to record 3D information with high redundancy: it is robust to noise, missing data, and the approximation used in the calculation process. However, the scale of a parallel computation on a dedicated computer depends on how dense is the calculation unit implemented on a circuit chip (e.g., field programmable gate arrays (FPGAs) and application-specific integrated circuits); thus, designing a smaller calculation unit is crucial to improve the performance of the dedicated computer. Therefore, approximating the CGH calculation to be calculated in the smaller steps and with shorter bit-width is an important way to enhance the performance of the dedicated computer.

This study proposes an approximation method of the sine and cosine functions in CGH calculations mostly based on a simple logical operations whose outputs are only two bits in total. A CGH is obtained by accumulating the two-bits outputs of the proposed method; thus, the proposed method reduces not only the computational complexity of the trigonometric functions but also the computational resources required for the superposition of wavefronts.

Thus, the contributions of this study are as follows:

- We propose an effective approximation of the sine and cosine functions for CGH calculation.
- We validated the computational efficiency of the proposed method on a central processing unit (CPU) and a GPU for validating the possibility of implementing a dedicated circuit.
- We validated the image quality of proposed method via numerical and optical experiments.

The purpose of this paper was to clarify both the effectiveness and limitation of the proposed method in terms of computational speed and image quality in a feasibility study; thus, we only tested the proposed method on a CPU and GPU. We did not design a dedicated computer with the proposed method. Since we validated the effectiveness of the proposed method in this study, in future works, we will report on the implementation of the proposed method in a dedicated circuit.

The reminder of this paper is organized as follows: Section II overviews the PLS-based CGH calculation, and Section III describes the details of the proposed method. Section IV presents and discusses the experimental results, and Section V concludes this study.

II. CGH CALCULATION FOR 3D PLS MODEL

CGH calculation based on a 3D PLS model is described as

$$U(x_\alpha, y_\alpha) = \sum_{j=1}^{N_{\mathrm{obj}}} a_j \exp\left\{ i\frac{2\pi}{\lambda}\left(\frac{x_{\alpha j}^2 + y_{\alpha j}^2}{2|z_j|}\right)\right\}, \tag{1}$$

$$= \sum_{j=1}^{N_{\mathrm{obj}}} a_j \exp\left(i2\pi\theta_{\alpha j}\right), \tag{2}$$

where (x_α, y_α) are the coordinates of a hologram; U is a complex amplitude; N_{obj} is the number of PLSs; i is an imaginary unit; λ is the wavelength of the light source; (x_j, y_j, z_j) is a coordinate of j-th PLS; $x_{\alpha j} = x_\alpha - x_j$, $y_{\alpha j} = y_\alpha - y_j$; and a_j is the amplitude of j-th PLS. Taking $a_j = 1$ for simplicity, (2) can be regarded to be composed of the unit vectors with argument $2\pi\theta_{\alpha j}$ on a complex plane. Ideally, the PLS to be computed for each pixel is selective for prevent aliasing noise; in other words, the PLS far from the pixels in the $x - y$ direction can be ignored. However, for simplicity, this point is not considered in this study because the computer architecture can be simplified if the number of PLS to be computed is the same for all pixels.

Although (2) calculates the complex amplitude distribution of a hologram, the current spatial light modulator (SLM), which is a display device to modulate the incident light, cannot physically replay $U(x_\alpha, y_\alpha)$ directly. That is, the SLM can modulate either the amplitude or phase of $U(x_\alpha, y_\alpha)$. This study targets the CGH of phase-only modulation (the kinoform [30]) since it can be used to reconstruct a clearer 3D image than amplitude-only modulation. The kinoform is obtained using (2) as

$$c(x_\alpha, y_\alpha) = \arg\{U(x_\alpha, y_\alpha)\} \cdot \frac{2^m}{2\pi}, \tag{3}$$

where $\arg\{\cdot\}$ is an operator for obtaining the argument of the complex wave, and m is the bit-depth of a CGH image. The CGH is usually output as a digital image and displayed on an SLM.

Considering (2), the composition of the complex wavefront on (x_α, y_α) for all PLSs can be regarded as the composition of unit vectors with argument $2\pi\theta_{\alpha j}$ on the complex field. That is, obtaining $c(x_\alpha, y_\alpha)$ in (3) is equal to finding the quantized argument of the synthesized vector. The CGH is robust to approximations in the calculation process, implying that the result obtained by using a rough estimate of the argument or the position of the synthesized vector will be acceptable.

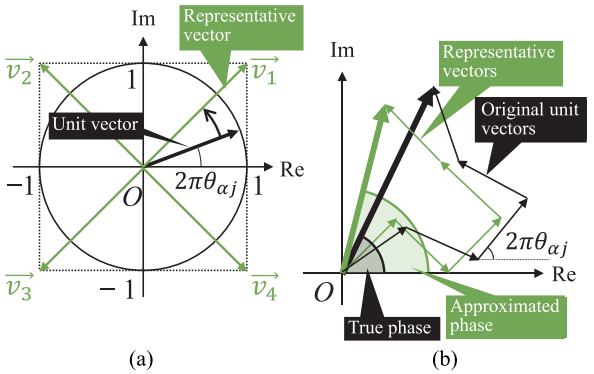


FIGURE 1. Overview of the proposed method: (a) Approximating a unit vector for a representative vector, and (b) an example of the approximation of complex vectors (wave front) compositions.

Therefore, the proposed method approximates each unit vector to four orthogonal vectors to obtain a trend that is approximately equivalent to the direction of the synthesized vector points.

III. PROPOSED METHOD

Figure 1 provides an overview of the proposed method. First, we define the representative vectors $\vec{v}_q$ of each quadrant as follows:

$$\vec{v}_1 = e^{i\frac{\pi}{4}} = (1, 1), \tag{4}$$
$$\vec{v}_2 = e^{i\frac{3\pi}{4}} = (-1, 1), \tag{5}$$
$$\vec{v}_3 = e^{i\frac{5\pi}{4}} = (-1, -1), \tag{6}$$
$$\vec{v}_4 = e^{i\frac{7\pi}{4}} = (1, -1). \tag{7}$$

In the proposed method, all the unit vectors are approximated to each representative vector. For example, a unit vector $(0.809, 0.588)$ (for $\theta_{\alpha j} = 0.1$ in (2)) is approximated to $\vec{v}_1$ since it belongs in the first quadrant. Given $\vec{v}_{(q,j)}$ is a representative vector approximated from the unit vector of j-th PLS, where q is the quadrant, (2) can be approximated as follows:

$$U(x_\alpha, y_\alpha) \approx \sum_{j=1}^{N_{\text{obj}}} \vec{v}_{(q,j)}. \tag{8}$$

Finally, we can obtain the approximated CGH using (3).

We modified the above process as to obtain $\vec{v}_{(q,j)}$ from $\theta_{\alpha j}$ easily. First, the representative vectors are redefined as follows for ignoring negative values since the accumulation of the negative values can be substituted by subtraction after the synthesis is completed.

$$\vec{v'}_1 = (1, 1), \tag{9}$$
$$\vec{v'}_2 = (0, 1), \tag{10}$$
$$\vec{v'}_3 = (0, 0), \tag{11}$$
$$\vec{v'}_4 = (1, 0). \tag{12}$$

TABLE 1. Truth table of the bits in $\theta_{\alpha j}(b_{0.5}, b_{0.25})$ and the representative vector $\vec{v'}_{(q,j)}$.

	$b_{0.5}$	$b_{0.25}$	$\Re(\vec{v'}_{(q,j)})$	$\Im(\vec{v'}_{(q,j)})$
1st quad. $(q = 1)$	0	0	1	1
2nd quad. $(q = 2)$	0	1	0	1
3rd quad. $(q = 3)$	1	0	0	0
4th quad. $(q = 4)$	1	1	1	0

Then, we rewrite (8) as

$$U(x_\alpha, y_\alpha) = 2\sum_{j=1}^{N_{\text{obj}}} \vec{v'}_{(q,j)} - (N_{\text{obj}}, N_{\text{obj}}). \tag{13}$$

Next, we calculate $\vec{v'}_{(q,j)}$ from $\theta_{\alpha j}$ via simple logical operations. Since $\theta_{\alpha j}$ is multiplied by 2π in (2), we can determine $\vec{v'}_{(q,j)}$ by the range of the fractional part of $\theta_{\alpha j}$ as follows:

$$\vec{v'}_{(q,j)} = \begin{cases} \vec{v'}_{(1,j)} & when \quad 0.00 \le \theta_{\alpha j} < 0.25, \\ \vec{v'}_{(2,j)} & when \quad 0.25 \le \theta_{\alpha j} < 0.50, \\ \vec{v'}_{(3,j)} & when \quad 0.50 \le \theta_{\alpha j} < 0.75, \\ \vec{v'}_{(4,j)} & when \quad 0.75 \le \theta_{\alpha j} < 1.00. \end{cases} \tag{14}$$

Equation (14) can be implemented without the branch and floor operations but with simple logical operations. Given $b_{0.5}$ and $b_{0.25}$ are the bits in $\theta_{\alpha j}$ that represents the bit flag of 2^{-1} and 2^{-2}, we obtain the truth table between the elements of each $\vec{v'}_{(q,j)}$ and those bits as summarized in Table 1.

Therefore, $\vec{v'}_{(q,j)}$ is obtained as

$$\Re(\vec{v'}_{(q,j)}) = \text{NOT}(\text{XOR}(b_{0.25}, b_{0.5})) \tag{15}$$
$$\Im(\vec{v'}_{(q,j)}) = \text{NOT}(b_{0.5}) \tag{16}$$

where $\Re(\cdot)$ and $\Im(\cdot)$ describes the function of extracting the real and imaginary parts; $\text{NOT}(\cdot)$ is the function of logical negation; and $\text{XOR}(\cdot)$ is the function of exclusive disjunction.

Fractional values can be expressed mainly in two ways: fixed-point and floating point. In the fixed-point approach, $b_{0.5}$ and $b_{0.25}$ are easily extracted since the length of both the integer and fractional parts are fixed. In the floating-point approach, considering IEEE754 [31], the positions of $b_{0.5}$ and $b_{0.25}$ are dynamically changed but can be determined by the value of the exponential part. In other words, $b_{0.5}$ is located at the $(2^{\text{exp}} - b + le + ls)$-th bit from the most significant bit, where exp is the recorded value of the exponent part; o is the bias value; and le and ls are the lengths of the exponent and sign parts, respectively. In the case of 32-bit floating point, $b = 127, le = 8, ls = 1$. Therefore, we obtain $\vec{v'}_{(q,j)}$ by simple logical operations.

In summary, the proposed method substitutes the accumulation of the complex values that comprised sine and cosine functions into simple computations based on few logical operations.

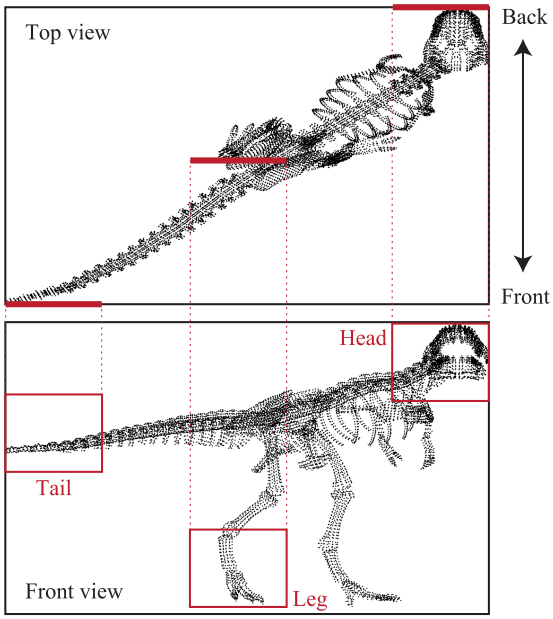


FIGURE 2. Three-dimensional (3D) model composed of point-light sources (PLSs).

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

We investigated the validity of the proposed method w.r.t. both the computational speed and reconstructed image quality. Since the purpose of this study was to conduct an initial investigation of the feasibility of the proposed method, we implemented the method in a CPU and GPU. That is, we did not implement the proposed method on a circuit (e.g., FPGA); this implementation will be considered in a future study. The results obtained by the proposed method are compared to those of four reference methods used for implementing sin/cos functions:

- Method using normal sin/cos functions
- Method using a look-up table (LUT) of sin/cos functions
- Method using a simple approximation of sin/cos functions [23]
- Method using the built-in fastmath operation of sin/cos function for a GPU

In the LUT method, we implemented the table on constant memory, and the number of the elements in the table was set as 256 for both the CPU and GPU because of the memory limitation of the constant memory. Further, the implementations out of the sin/cos functions were the same for all methods.

The computational environment used in this study was as follows: Microsoft Windows 11 Professional operating system, Intel Core i9-12900KF 3.20 GHz, 64-GB DDR5-38400 memory, Microsoft Visual C++ 2022 compiler, and a NVIDIA GeForce 3090Ti with CUDA 11.8. The CGH resolution was set to 4,096 × 2,400 pixels, corresponding to the SLM we used. The program in the CPU is parallelized with OpenMP, and the number of the active threads was 24, the maximum value for the CPU we used. The computational precision used in the experiment is a 32-bit floating point, whereas the proposed method partially used a 16-bit integer in accumulating the complex value according to (8).

TABLE 2. Results of calculation time.

	CPU		GPU	
	Time[s]	Ratio	Time[ms]	Ratio
Normal sin/cos	331.9	1.00	1252	1.00
Proposed	150.6	2.20	148.5	8.43
With [23]	213.6	1.55	147.9	8.46
LUT	339.8	0.977	2576	0.49
Fastmath	-	-	145.3	8.62

TABLE 3. Result of image quality assessment.

	CPU		GPU	
	PSNR	SSIM	PSNR	SSIM
Proposed	29.8	0.880	30.1	0.882
With [23]	45.4	0.992	45.0	0.983
LUT	33.1	0.803	35.2	0.816
Fastmath	-	-	67.8	1.000

We used a 3D object composed of 11,646 PLSs, as shown in Fig. 2. The coordinates of all the PLSs were distributed as follows: $0\,[\mathrm{m}] \le x_j < 4,096 \times p\,[\mathrm{m}], 0\,[\mathrm{m}] \le y_j < 2,400 \times p\,[\mathrm{m}], 0.14\,[\mathrm{m}] \le z_j \le 0.16\,[\mathrm{m}]$, where p is the pixel pitch of SLM (3.74 [μm] in this study).

A. COMPUTATIONAL SPEED

Table 2 presents a comparison of the calculation time and speeding-up ratio with the CPU and GPU. Here, the speeding-up ratio was calculated by setting the result of normal sin/cos functions as a reference. The table shows that the proposed method is the fastest for CPU implementation and is 2.20 times faster than the reference method. However, in GPU implementation, fastmath implementation is the fastest, followed by the method reported in [23]; the proposed method ranks the third in terms of the computational speed. These results indicate that the proposed method can reduce the computational complexity of the trigonometric functions compared to the ordinal functions. Further, the results of GPU implementation show that the proposed method performs calculations at a speed only slightly lower speed than that of [23] or built-in fastmath operations in the NVIDIA GPU. This is because the proposed method requires the use of logic operations for 32-bit integers or floating point numbers, whereas the other two methods can use specially designed high-speed arithmetic circuits. In addition, additional processing such as reinterpretation of bit strings and bit shifting is necessary when implementing the bit operations included in the proposed method. Because of these multiple factors, we assume that the proposed method does not provide a large advantage in GPU implementation, especially in terms of computation speed. However, the proposed method is suitable for the dedicated computer since the proposed method involves only simple logic and arithmetic operations. Therefore, the

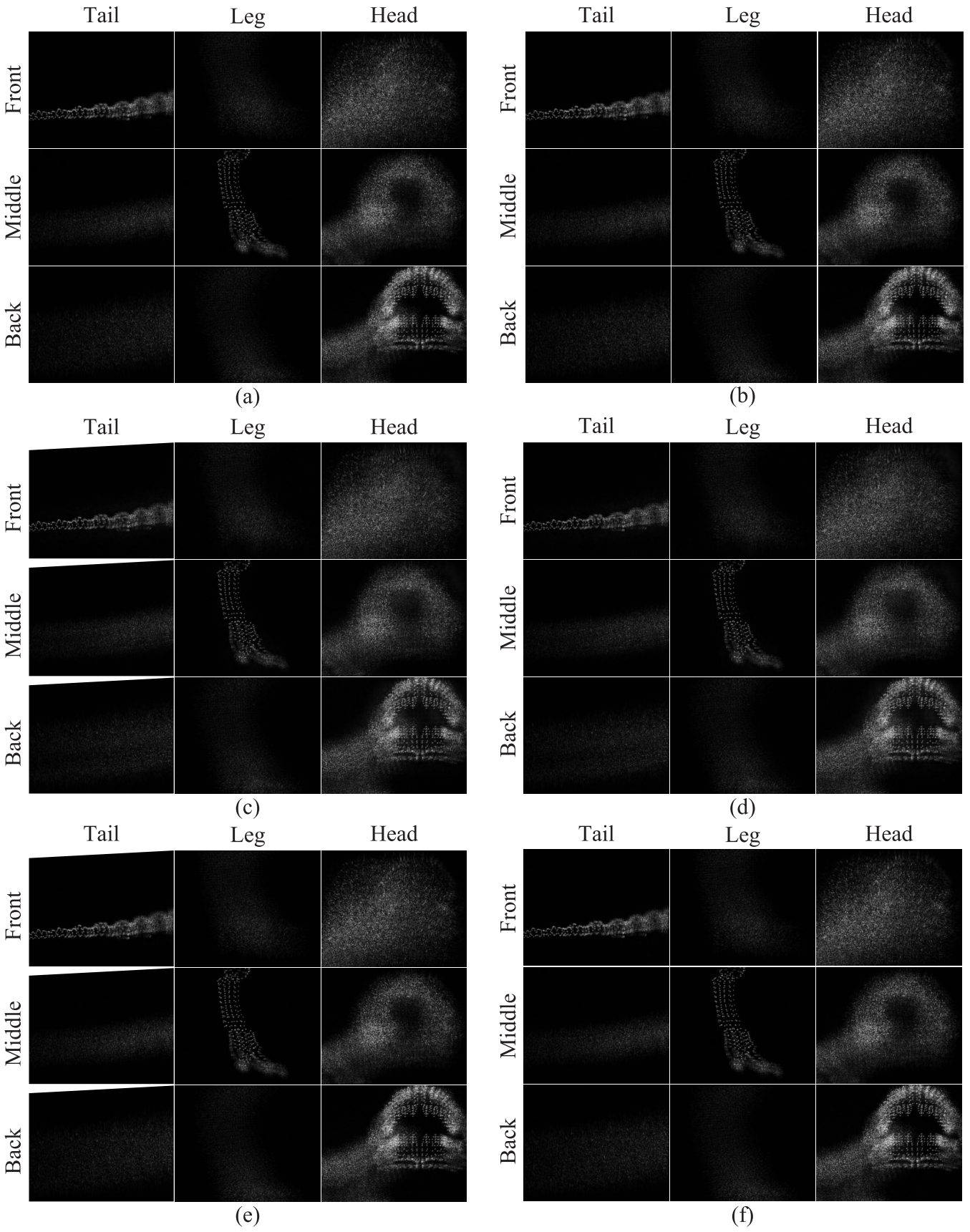


FIGURE 3. Numerically reconstructed images: (a) normal sin/cos functions on a central processing unit (CPU), (b) normal sin/cos function on a graphics processing unit (GPU), (c) the proposed method on the CPU, (d) the proposed method on the GPU, (e) the method of [23] on the CPU, and (f) the method of [23] on the GPU.

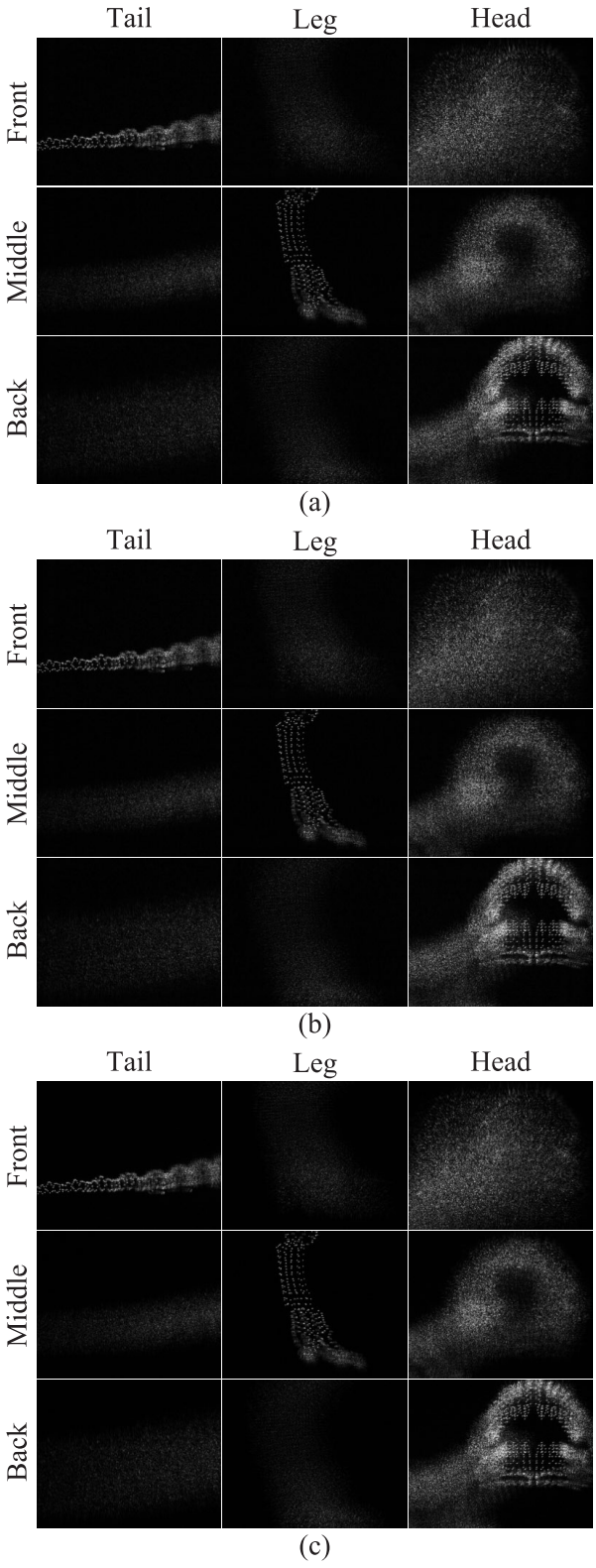


FIGURE 4. Numerically reconstructed images: (a) look-up table (LUT) of the sin/cos functions on the CPU, (b) LUT of the sin/cos functions on the GPU, (c) built-in fastmath operations on the GPU.

proposed method is expected to be advantageous regarding computational speed and resource consumption compared to other calculation methods when implemented on a dedicated computer.

B. IMAGE QUALITY

We investigated the image quality via numerical and optical experiments. In the numerical experiment, we obtained the image of the intensity distribution propagated from the CGH using the angular spectrum method and calculated the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). Here, the bit depth of the reconstructed image was 8 (grayscale). Further, the reconstructed image was obtained from the intensity distributions of floating point precision with a histogram-based threshold filter to avoid the effect of peaky signals that does not affect visual recognition. We obtained 100 images of the intensity distribution over the distance from the CGH, equally divided into 100 sections from the frontmost to the backmost of z_j. Table 3 summarizes the PSNR and SSIM results of the proposed and reference methods; these results provide the average values for 100 intensity images that were numerically obtained. The PSNR and SSIM of the proposed method are less than those of the reference method; however, they exceeds the standard of fair quality images (PSNR is 25 dB and SSIM is 0.88 higher) [32]. Thus, from the quantitative results, we confirmed that the proposed method can reconstruct the 3D images with sufficient image quality.

Figures 3 and 4 provide examples of the numerically reconstructed image from CGHs. The figures are cropped according to Fig.2 to show the focusing performance of the CGH. As seen clearly in the figures, all the reconstructed images can focus light according to the original PLS model with sufficient resolution, and there are no drastic differences among the images reconstructed by all the methods and at all depths; thus, these figures show the effectiveness of the proposed method.

Figure 5 provides the image of the absolute error distribution between the numerically reconstructed images of each method and the method with normal sin/cos functions run with the CPU. The figures show that the proposed method's error degree is relatively high compared to the other methods, and the error degree of the fastmath method is almost none. These intuitive observations are consistent with the numerical evaluation of PSNR and SSIM.

Lastly, we conducted the optical experiment. Figure 6 shows the optical system used in in this study. This system comprises a phase-modulation type SLM (JD7714, Jasper Display Corp.) with a resolution of $4,096 \times 2,400$ pixels and a pixel pitch of 3.74 μm, green laser with a wavelength of 532 nm (CPS532-C2, Thorlabs Inc.), half-wave plate (HWP) (WPH10M-523, Thorlabs Inc.), polarizer (WP25M-VIS, Thorlabs Inc.), beam expander (GBE-10A, Thorlabs Inc.), quarter-wave plate (QWP) (WPQ10M-532, Thorlabs Inc.), polarized beam splitter (PBS) (CCM1-PBS251/M, Thorlabs Inc.), achromatic doublet lenses (AC254-150A-ML,

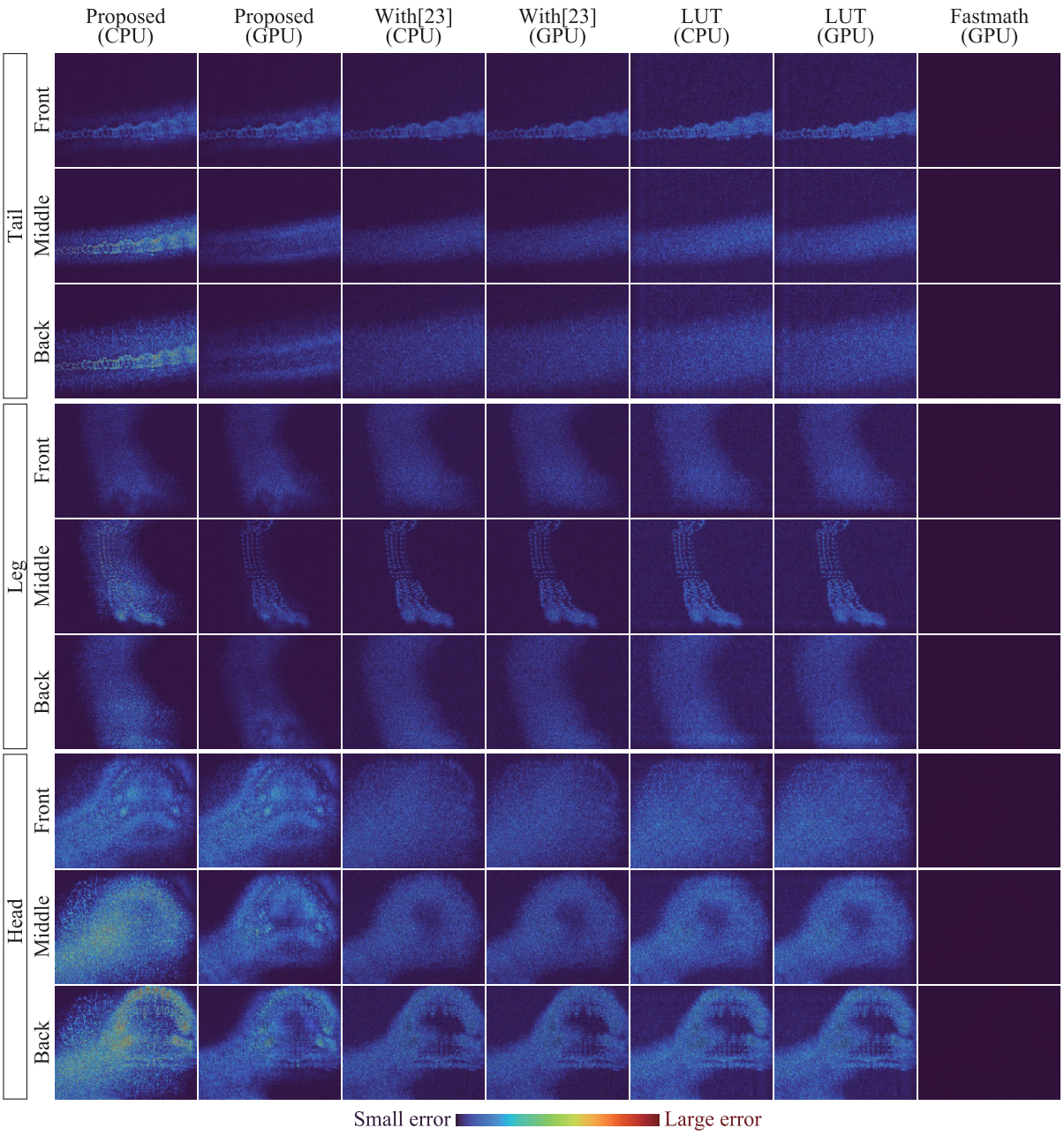


FIGURE 5. Absolute error distribution of numerically reconstructed images.

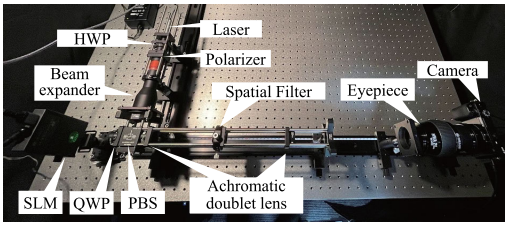


FIGURE 6. Annotated photograph of the optical setup on which the holographic image is shown. HWP: Half-wave plate; QWP: quarter-wave plate; PBS: polarized beam splitter; and SLM: spatial light modulator.

Thorlabs Inc.), eyepiece (602416, MEADE), and a camera (ILCE-6000, Sony). The images show that the we can obtain the physically reconstructed 3D image that is almost identical to the numerically obtained image.

Figure 7 provides examples of the optically reconstructed CGH images created by the proposed method and by using the normal sin/cos functions on the CPU and with the built-in fastmath operations on the GPU. As the pictures show, there are no major differences between the methods, which are valid for all methods that are not depicted in the figure.

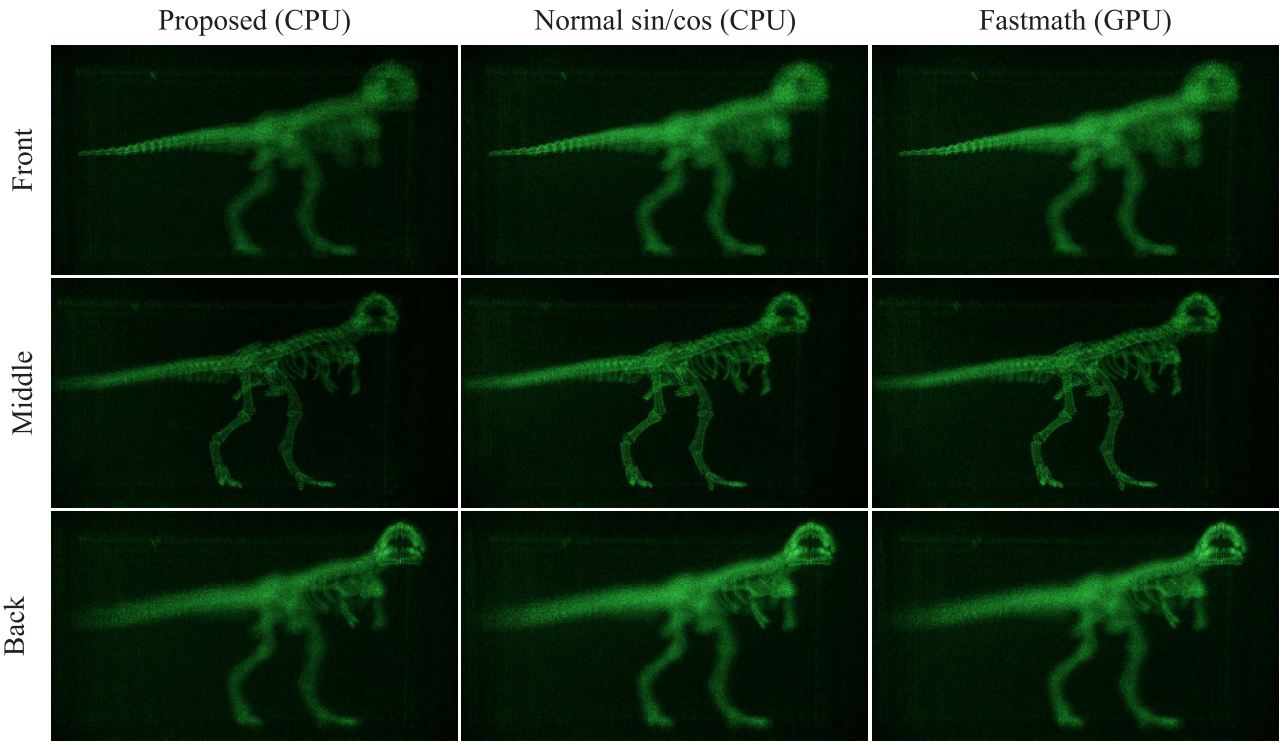


FIGURE 7. Example of optically reconstructed images.

Therefore, we conclude that the proposed method provides satisfactory image quality.

V. CONCLUSION

This paper proposed an approximation method for the wavefront composition of a CGH. The proposed method substitutes the trigonometric functions, which are usually required for calculating the complex amplitude of a wavefront, by simple logical operations. This substitution reduces the computational load greatly compared to the load of ordinal trigonometric functions, at the cost of the image quality of the reconstructed image. Although the proposed method accelerates CGH computation, the acceleration factor is limited, especially in the GPU implementation. This is because we used the existing arithmetic calculation unit for 32-bit integer or floating point numbers unless the proposed method only required two bits. Thus, the proposed method will be more effective when implemented on a dedicated circuit. This research direction will be considered in the future.

ACKNOWLEDGMENT

The authors would like to thank Dr. Hirotaka Nakayama who created the 3D model and allowed them to use it.

REFERENCES

[1] A. Maimone, A. Georgiou, and J. S. Kollin, "Holographic near-eye displays for virtual and augmented reality," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–16, Jul. 2017.

[2] T. Kozacki, M. Chlipala, J. Martinez-Carranza, R. Kukołowicz, and M. S. Idicula, "LED near-eye holographic display with a large non-paraxial hologram generation," *Opt. Exp.*, vol. 30, no. 24, p. 43551, Nov. 2022.

[3] 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*, vol. 7, no. 11, p. 1563, Nov. 2020.

[4] J. Sano and Y. Takaki, "Holographic contact lens display that provides focusable images for eyes," *Opt. Exp.*, vol. 29, no. 7, p. 10568, Mar. 2021.

[5] H. Gao, Y. Wang, X. Fan, B. Jiao, T. Li, C. Shang, C. Zeng, L. Deng, W. Xiong, J. Xia, and M. Hong, "Dynamic 3D meta-holography in visible range with large frame number and high frame rate," *Sci. Adv.*, vol. 6, no. 28, Jul. 2020, Art. no. eaba8595.

[6] R. Izumi, S. Ikezawa, and K. Iwami, "Metasurface holographic movie: A cinematographic approach," *Opt. Exp.*, vol. 28, no. 16, p. 23761, Aug. 2020.

[7] F. Wang, T. Ito, and T. Shimobaba, "High-speed rendering pipeline for polygon-based holograms," *Photon. Res.*, vol. 11, no. 2, p. 313, Feb. 2023.

[8] Y. Zhang, H. Fan, F. Wang, X. Gu, X. Qian, and T.-C. Poon, "Polygon-based computer-generated holography: A review of fundamentals and recent progress [invited]," *Appl. Opt.*, vol. 61, no. 5, p. 363, Feb. 2022.

[9] Y.-P. Zhang, F. Wang, T.-C. Poon, S. Fan, and W. Xu, "Fast generation of full analytical polygon-based computer-generated holograms," *Opt. Exp.*, vol. 26, no. 15, p. 19206, Jul. 2018.

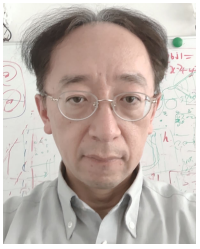
[10] T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Fast calculation of computer-generated hologram of line-drawn objects without FFT," *Opt. Exp.*, vol. 28, no. 11, p. 15907, May 2020.

[11] D. Blinder, T. Nishitsuji, and P. Schelkens, "Three-dimensional spline-based computer-generated holography," *Opt. Exp.*, vol. 31, no. 2, p. 3072, Jan. 2023.

[12] T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Review of fast calculation techniques for computer-generated holograms with the point-light-source-based model," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2447–2454, Oct. 2017.

[13] T. Sugawara, Y. Ogihara, and Y. Sakamoto, "Fast point-based method of a computer-generated hologram for a triangle-patch model by using a graphics processing unit," *Appl. Opt.*, vol. 55, no. 3, p. 160, 2016.
[14] H. Shiomi, T. Shimobaba, T. Kakue, and T. Ito, "Reducing the computational complexity of high-resolution hologram calculations using polynomial approximation," *Opt. Exp.*, vol. 31, no. 11, p. 18576, May 2023.
[15] Z. Wang, G. Lv, Q. Feng, A. Wang, and H. Ming, "Simple and fast calculation algorithm for computer-generated hologram based on integral imaging using look-up table," *Opt. Exp.*, vol. 26, no. 10, p. 13322, May 2018.
[16] Y. Ichihashi, R. Oi, T. Senoh, K. Yamamoto, and T. Kurita, "Real-time capture and reconstruction system with multiple GPUs for a 3D live scene by a generation from 4K IP images to 8K holograms," *Opt. Exp.*, vol. 20, no. 19, p. 21645, Sep. 2012.
[17] Y. Ichihashi, "Computer-generated hologram," in *Hardware Acceleration of Computational Holography*, T. Shimobaba and T. Ito, Eds. Singapore: Springer, 2023, pp. 25–29.
[18] H. Yoshikawa, "Fast computation of Fresnel holograms employing difference," *Opt. Rev.*, vol. 8, no. 5, pp. 331–335, Sep. 2001.
[19] T. Shimobaba and T. Ito, "An efficient computational method suitable for hardware of computer-generated hologram with phase computation by addition," *Comput. Phys. Commun.*, vol. 138, no. 1, pp. 44–52, Jul. 2001.
[20] K. Matsushima and M. Takai, "Recurrence formulas for fast creation of synthetic three-dimensional holograms," *Appl. Opt.*, vol. 39, no. 35, p. 6587, 2000.
[21] T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Fast calculation of computer-generated hologram using run-length encoding based recurrence relation," *Opt. Exp.*, vol. 23, no. 8, p. 9852, Apr. 2015.
[22] S. Jiao, Z. Zhuang, and W. Zou, "Fast computer generated hologram calculation with a mini look-up table incorporated with radial symmetric interpolation," *Opt. Exp.*, vol. 25, no. 1, p. 112, Jan. 2017.
[23] T. Nishitsuji, T. Shimobaba, T. Kakue, D. Arai, and T. Ito, "Simple and fast cosine approximation method for computer-generated hologram calculation," *Opt. Exp.*, vol. 23, no. 25, p. 32465, Dec. 2015.
[24] D. Blinder and P. Schelkens, "Fast low-precision computer-generated holography on GPU," *Appl. Sci.*, vol. 11, no. 13, p. 6235, Jul. 2021.
[25] T. Nishitsuji, Y. Yamamoto, T. Sugie, T. Akamatsu, R. Hirayama, H. Nakayama, T. Kakue, T. Shimobaba, and T. Ito, "Special-purpose computer HORN-8 for phase-type electro-holography," *Opt. Exp.*, vol. 26, no. 20, p. 26722, Oct. 2018.
[26] J. An, K. Won, Y. Kim, J.-Y. Hong, H. Kim, Y. Kim, H. Song, C. Choi, Y. Kim, J. Seo, A. Morozov, H. Park, S. Hong, S. Hwang, K. Kim, and H.-S. Lee, "Slim-panel holographic video display," *Nature Commun.*, vol. 11, no. 1, p. 5568, Nov. 2020.
[27] H. Kim, Y. Kim, H. Ji, H. Park, J. An, H. Song, Y. T. Kim, H.-S. Lee, and K. Kim, "A single-chip FPGA holographic video processor," *IEEE Trans. Ind. Electron.*, vol. 66, no. 3, pp. 2066–2073, Mar. 2019.
[28] Y.-H. Seo, Y.-H. Lee, and D.-W. Kim, "ASIC chipset design to generate block-based complex holographic video," *Appl. Opt.*, vol. 56, no. 9, p. 52, Mar. 2017.
[29] Y. Pan, X. Xu, S. Solanki, X. Liang, R. B. A. Tanjung, C. Tan, and T.-C. Chong, "Fast CGH computation using S-LUT on GPU," *Opt. Exp.*, vol. 17, no. 21, p. 18543, 2009.
[30] L. B. Lesem, P. M. Hirsch, and J. A. Jordan, "The kinoform: A new wavefront reconstruction device," *IBM J. Res. Develop.*, vol. 13, no. 2, pp. 150–155, Mar. 1969.
[31] *IEEE Standard for Floating-Point Arithmetic*, Standard IEEE 754-2008, Aug. 2008.
[32] A.-N. Moldovan, I. Ghergulescu, and C. H. Muntean, "A novel methodology for mapping objective video quality metrics to the subjective MOS scale," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, Jun. 2014, pp. 1–7.

TAKASHI NISHITSUJI received the B.E., M.E., and Ph.D. degrees in fast calculation algorithm of holography from Chiba University, Chiba, Japan, in 2007, 2013, and 2016, respectively. From 2013 to 2018, he was a Researcher with the Information Technology Research and Development Center, Mitsubishi Electric Corporation, Kanagawa, Japan. From 2018 to 2023, he was an Associate Professor with Tokyo Metropolitan University, Tokyo, Japan. He is currently a Junior Associate Professor with Toho University, Chiba. His current research interests include computer holography and its application. He is a member of Optica, the Optical Society of Japan, the International Society for Optical Engineering (SPIE), and the Information Processing Society of Japan (IPSJ).



TOMOYOSHI SHIMOBABA received the B.E. and M.E. degrees in electrical engineering from Gunma University, Japan, in 1997 and 1999, respectively, and the Ph.D. degree from Chiba University, Japan, in 2002. From 2002 to 2005, he was a Postdoctoral Researcher with RIKEN. From 2005 to 2009, he was an Associate Professor with the Graduate School of Science and Engineering, Yamagata University. From 2009 to 2019, he was an Associate Professor with the Graduate School of Engineering, Chiba University, where he has been a Professor, since 2019. His current research interests include computer holography and its various applications. He is a member of ITE, IEICE, Optica, OSJ, and SPIE.

ATSUSHI SHIRAKI received the B.E. degree in engineering from Yamaguchi University, Japan, in 2003, and the M.E. and Ph.D. degrees in engineering from Chiba University, Japan, in 2005 and 2008, respectively. From 2008 to 2011, he was an Assistant Professor with the Kisarazu College, National Institute of Technology, where he was a Lecturer, from 2011 to 2014, and an Associate Professor, from 2014 to 2015. Since 2015, he has been an Associate Professor with the Digital Transformation Enhancement Council, Chiba University. His current research interests include 3D systems, educational technology, and their applications. He is a member of ITE, IEICE, IPSJ, and JSET.

TOMOYOSHI ITO received the B.S. degree in pure and applied sciences and the M.S. and Ph.D. degrees in earth science and astronomy from The University of Tokyo, Japan, in 1989, 1991, and 1994, respectively. He was a Research Associate with Gunma University, Japan, from 1992 to 1994, where he was an Associate Professor, from 1994 to 1999. From 1999 to 2004, he was an Associate Professor with the Graduate School of Engineering, Chiba University, Japan, where he has been a Professor, since 2004. His current research interests include high-performance computing and its applications, such as electronic holography for 3D TV. He is a member of ACM, ASJ, ITE, IEICE, IPSJ, and Optica.