

High-performance computer system dedicated to ray-wavefront conversion technique aimed to display holograms in real-time

Tatsuya Maruyama^{a,b}, Yasuyuki Ichihashi^{b,*}, Ikuo Hoshi^b, Takashi Kakue^a,
Tomoyoshi Shimobaba^a and Tomoyoshi Ito^a

^aChiba University, Graduate School of Engineering, Chiba, Japan

^bNational Institute of Information and Communications Technology, Radio Research Institute, Koganei, Japan

ABSTRACT. Holography is the ultimate three-dimensional (3D) imaging technology, and research is actively being conducted on generating at high-speed holograms with enormous amounts of information and improving image quality. Because computer holography is based on wave optics algorithms, reconstructing the texture of 3D images is difficult. On the other hand, in principle, reconstructing texture in integral photography based on geometric optics is easy and the method is well established. We developed a high-performance special-purpose computer, called holographic reconstruction for ultra-realistic imaging, dedicated to the ray-wavefront conversion method. The design and implementation of the circuit for the hologram generation were performed using field-programmable gate array technology. Parallelization was performed at each step for increasing the speed of the calculation process. Furthermore, by sending output data directly to a displaying device in the high-definition multimedia interface, the communication between the host personal computer (PC) and special-purpose computer was controlled in one direction, which significantly reduced communication time. The system was ~ 7.7 times faster than a PC alone and succeeded in the holographic reproduction of a textured 3D image in real time at 30 frames per second for a 1024×1024 -pixel hologram.

© The Authors. Published by SPIE under a Creative Commons Attribution 4.0 International License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.OE.62.8.085102](https://doi.org/10.1117/1.OE.62.8.085102)]

Keywords: holography; computer-generated hologram; field-programmable gate array; light-ray information; discrete Fourier transform

Paper 20230086G received Jan. 30, 2023; revised Jul. 5, 2023; accepted Jul. 27, 2023; published Aug. 9, 2023.

1 Introduction

Holography¹⁻³ is the ultimate technology for reconstructing natural three-dimensional (3D) images. The 3D image information is recorded in the hologram as interference fringes of the object light from the 3D object and reference light. When a hologram is irradiated with the reproduction illumination light, the 3D images can be reconstructed. In addition, computer-generated holograms (CGHs) are created by simulating the propagation and interference of objects and reference lights on a computer. A hologram can be displayed on a spatial light modulator (SLM) for reconstructing a 3D image. When an electronic device, such as a micro liquid crystal display (LCD) is used as an SLM, real-time video reproduction is also possible by rewriting holograms displayed on the SLM.

However, computer holography has not yet been practically used because of the large computational load of the hologram. In addition, reconstructing textures with high quality is possible

*Address all correspondence to Yasuyuki Ichihashi, y-ichihashi@nict.go.jp

for computer holography,⁴ but it is difficult to maintain high-speed computing. To speed-up hologram computation, many researchers have attempted developing algorithms, such as look-up table (LUT) methods⁵⁻⁸ and wavefront recording plane methods,^{9,10} graphics processing units (GPUs),¹¹⁻¹³ and dedicated hardware approaches.¹⁴⁻²¹ Holographic reconstruction (HORN-9), a special-purpose computer for holography, proved 600 times faster than a personal computer (PC).²²

In many cases, computer holography is based on computational wave optics. Wave optics-based methods include the spherical wave synthesis method, which considers a 3D object as a collection of point-light sources, and the polygon method,^{23,24} which calculates object lights from a 3D polygon model. In contrast, geometric optical methods based on light-ray information such as integral photography^{25,26} can be applied to existing computer graphics methods, which enable the expression of 3D images with high texture reproducibility. However, an inherent limitation of ray-based methods is that the resolution is reduced by geometric optics approximations. Therefore, a hologram generation method^{27,28} based on light-ray information that exploits both geometric and wave optics has been proposed. A lens array is commonly used to capture information about the intensity and direction of light rays from a 3D object. The lens array consists of a number of small elemental lenses, and the light ray information is recorded at the focal plane through the elemental lens. When the 3D images are reproduced using a light field display, the 3D images are reproduced through the lens array used for recording. On the other hand, the acquisition of light-ray information on the focal plane by the elemental lens can also be expressed by the Fourier transform. Therefore, by the Fourier transformation of the elemental images, the light-ray information can be treated as wavefront information. Therefore, in the processing of light-ray information, the processing is performed on the texture, shading, and occlusion of the 3D object, and then the light-ray information is converted into wavefront information by the Fourier transform. In this way, hologram data can be created and reproduced from light-ray information. One of these methods, a ray-wavefront conversion method^{29,30} using a ray-sampling (RS) plane, has been proposed. An RS plane is placed near the object, and light-rays from the 3D object are recorded on the RS plane, and hologram is obtained by calculating the propagation of wavefronts from the RS plane to the hologram. This reduces the degradation of the reconstructed images at deep depths. On the other hand, the conversion from light-ray information to wavefront information and the calculation of light propagation from the RS plane to the hologram requires extensive use of the fast Fourier transforms (FFTs), which increase the computational load. To solve this problem, some methods of reducing computational complexity by introducing orthogonal projections in RS plane have been proposed.³¹

In this study, we developed a special-purpose computer, holographic reconstruction for ultra-realistic imaging (UR-HORN) that implements the ray-wavefront conversion method on a field-programmable gate array (FPGA) incorporating parallelization and pipelining of each processing module. In addition, to make effective use of special-purpose computers, we designed and implemented a system that performed computation and data transfer without delay. The real-time computation of hologram enables the reproduction of 3D movies and faster printing in the field of hologram printers.³²

2 Ray-Wavefront Conversion Method

In this section, we describe a hologram calculation method based on RS planes. Figure 1 shows the flowchart of the proposed method.

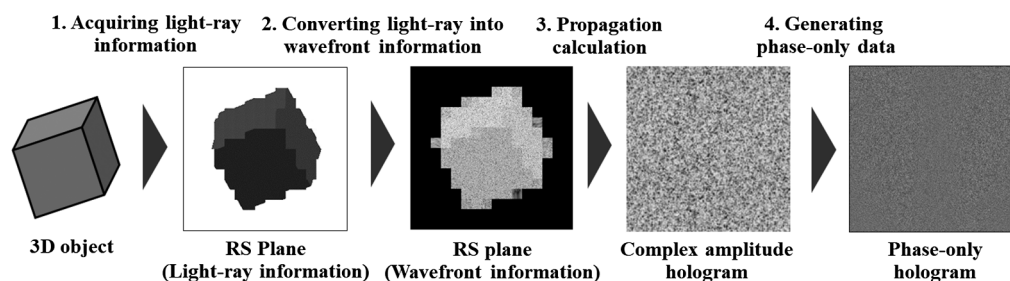


Fig. 1 Flow of ray-wavefront conversion method.

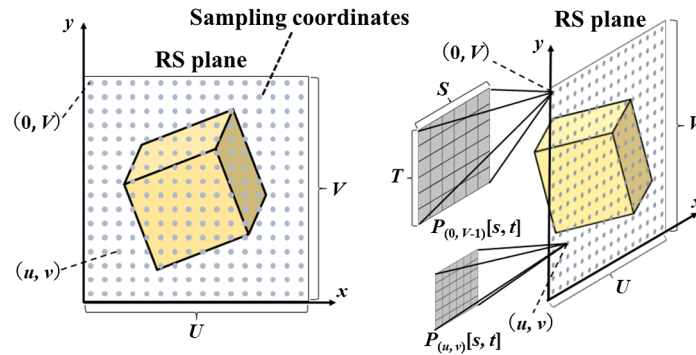


Fig. 2 Model of RS plane and elemental images.

2.1 Acquiring Light-Ray Information

First, an RS plane is set up in the vicinity of virtual 3D objects (Fig. 2). U and V are the horizontal and vertical sizes of the RS plane, S and T are the horizontal and vertical sizes of the elemental image. Then, the cameras are placed at the sampling coordinates (u, v) on the RS plane, and the elemental image $P_{(u,v)}[s, t]$ is acquired from each camera. Here, u and v represent the horizontal and vertical coordinates, which contain information of the direction of the light-rays. s and t represent the coordinates in the elemental images, and the $P_{(u,v)}[s, t]$ represents information of the intensity of the light-rays. This set of elemental images is called an RS image, and the number of elemental images represents the resolution of the reconstructed image.

2.2 Converting Light-Ray Information into Wavefront Information

Because of the distance between the RS plane and hologram, calculating wavefront propagation from the RS plane to the hologram is necessary. The acquired RS image retained information on the intensity and direction of the light-rays. To calculate the propagation from the RS plane to the hologram, converting into wavefront information, including phase information, is necessary. The method of the conversion from light-ray information into wavefront information was proposed in Ref. 29, and an acceleration method with multi-GPU was proposed in Ref. 30. In this study, an acceleration was achieved using an FPGA based on the method proposed in Ref. 29.

Let a complex amplitude distribution in the spatial domain be $a(x, y, 0)$, the coordinates in the frequency domain be $(f_x, f_y, 0)$, and j be the imaginary unit. The complex amplitude of the wavefront information $A(f_x, f_y, 0)$ can be expressed by using the Fourier transform as

$$A(f_x, f_y, 0) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} a(x, y, 0) \exp[-j2\pi(f_x x + f_y y)] dx dy. \quad (1)$$

Changing all the coordinates to the discrete coordinates, the above can be converted in the form of an FFT as follows:

$$A(f_x, f_y, 0) = \text{FFT}[a(x, y, 0)]. \quad (2)$$

Figure 3 shows a schematic of the procedure for converting light-ray information into wavefront information.

The elemental image $P_{(u,v)}[s, t]$ at coordinates (u, v) of the RS image is given a random phase $\varphi_{(u,v)}[s, t]$ to intentionally diffuse light-rays in various directions, and a two-dimensional (2D) FFT is performed to obtain the light wave distribution of the wavefront. The range of the random phase is $[0, 2\pi)$. The wavefront information $W_{(u,v)}[s, t]$ is calculated by

$$W_{(u,v)}[s, t] = \text{FFT}[P_{(u,v)}[s, t] \exp(j\varphi_{(u,v)}[s, t])]. \quad (3)$$

2.3 Propagation Calculation

We calculate the propagation from the RS plane to the hologram to obtain the complex amplitude on the hologram, as shown in Fig. 4. In this case, the propagation is calculated using the Fresnel diffraction.³³

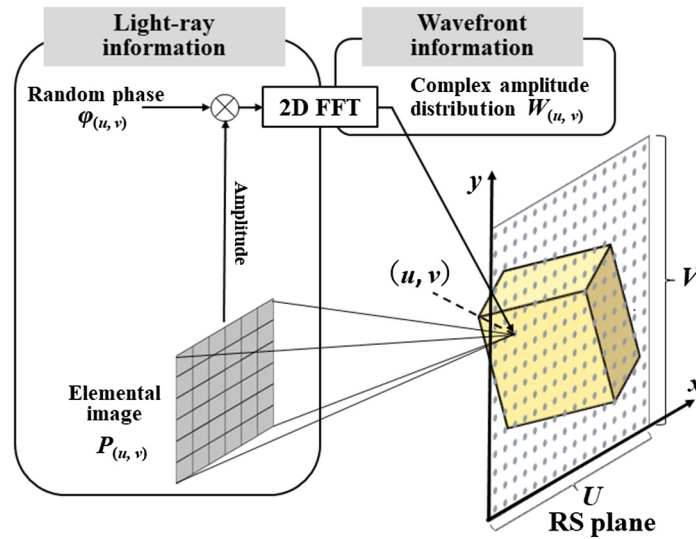


Fig. 3 Converting light-ray information into wavefront information.

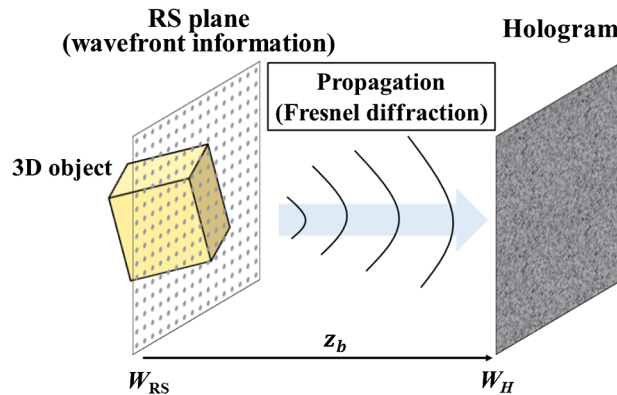


Fig. 4 Calculation of the complex hologram using the Fresnel diffraction.

$W_{(u,v)}[s, t]$ on the RS plane is again represented as $W_{RS}(x_a, y_a)$ using variables x_a and y_a . In this case, x_a and y_a are represented by the following equations:

$$x_a = s + u \times S, \quad y_a = t + v \times T. \quad (4)$$

The complex amplitude $W_H(x_b, y_b)$ of the hologram at a distance z_b can be given as

$$W_H(x_b, y_b) = \sum_{y_a = -\frac{N}{2}}^{\frac{N}{2}-1} \sum_{x_a = -\frac{M}{2}}^{\frac{M}{2}-1} W_{RS}(x_a, y_a) g(x_b - x_a, y_b - y_a), \quad (5)$$

$$g(x_b - x_a, y_b - y_a) = \frac{\exp(jkz_b)}{j\lambda z_b} \exp\left[\frac{jk}{2z_b} \{(x_b - x_a)^2 + (y_b - y_a)^2\}\right], \quad (6)$$

where x_b and y_b represent the coordinates in the hologram plane, and the resolution is the same as x_a and y_a , λ and k represent the wavelength and wavenumber of light, respectively, expressed as $k = 2\pi/\lambda$. M represents $S \times U$ and N represents $T \times V$. Assuming $M = N$, the 2D convolution integral regarding to the x - and y -axes can be transformed by a 2D FFT and Eq. (5) is expressed as

$$\text{FFT}[W_H(x_b, y_b)] = \text{FFT}[W_{RS}(x_a, y_a)]G(m, n), \quad (7)$$

where $G(m, n)$ represents the Fourier transform of $g(x_b, y_b)$, m and n are variables in the frequency domain. By performing an inverse 2D FFT on the results of the product calculation, the complex amplitude of the propagated plane can be obtained. $G(m, n)$ is

$$G(m, n) = \exp(jkz_b) \exp \left[2\pi j \left(\frac{-\lambda z_b}{2N^2(\Delta P)^2} \right) (m^2 + n^2) \right], \quad (8)$$

where ΔP denotes the pixel pitch of the hologram. The term $\exp(jkz_b)$ can be omitted because it does not affect the hologram generation; we can simplify it as

$$G(m, n) = \exp \left[2\pi j \left(\frac{-\lambda z_b}{2N^2(\Delta P)^2} \right) (m^2 + n^2) \right]. \quad (9)$$

In addition, the following equation is defined as the depth-related coefficient:

$$z_{\text{param}} = \frac{-\lambda z_b}{2N^2(2\Delta P)^2}. \quad (10)$$

$G(m, n)$ is

$$G(m, n) = \exp[2\pi j z_{\text{param}} (m^2 + n^2)]. \quad (11)$$

Finally, the complex amplitude at the hologram is

$$W_H(x_b, y_b) = \text{FFT}^{-1}[\text{FFT}[W_{\text{RS}}(x_a, y_a)]G(m, n)]. \quad (12)$$

In this study, the reference light is assumed to be in-line collimated light. The hologram is represented as interference fringes between the object light and the reference light, but the influence of the reference light can be ignored because it is ultimately normalized to 8-bit data in the computer simulation. Therefore, the complex amplitude distribution obtained in Eq. (12) can be treated as the hologram data as it is.

2.4 Generating Phase-Only Data

Since we expect to use a phased-type SLM, we need to make a phase-only hologram from the complex amplitude obtained in Eq. (12). The phase-only hologram equation is

$$W_{\text{phase}} = \arg(W_H). \quad (13)$$

3 Implementation of Calculation Unit

In this section, we describe the implementation of the ray-wavefront conversion method on an FPGA, including the process of converting light-ray information into wavefront information, propagation calculation, and generation of phase-only data. Furthermore, we describe the implementation of parallel processing at high speeds. In this study, we used the Virtex UltraScale+ VCU118 as the FPGA board. A schematic of the hologram calculation unit is shown in Fig. 5. The RS image, scaling parameters used in the calculation circuit, and the parameter defined in Eq. (10) were sent from the host PC. The scaling parameters were set to prevent the overflow of values in the FFT intellectual property (IP) core provided by Xilinx for one-dimensional (1D) FFT.

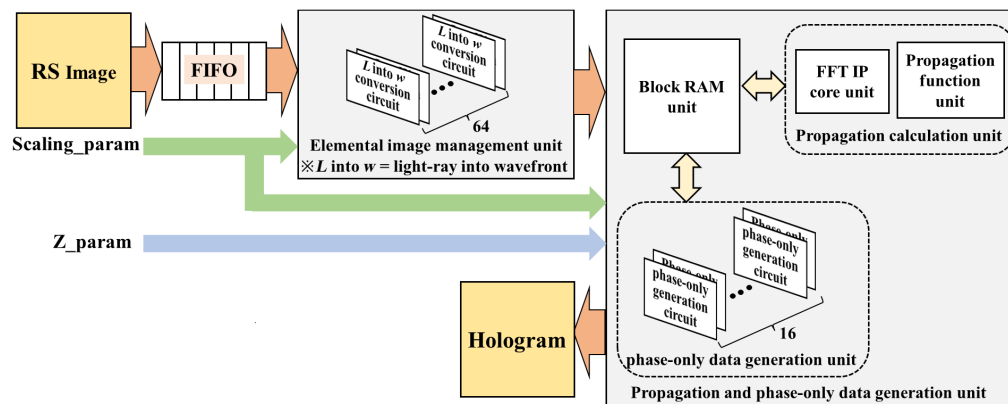


Fig. 5 The overall diagram of hologram calculation unit.

Table 1 Memory capacity in VCU118.

	LUT RAM	Block RAM	Ultra RAM
Capacity per block	64 bit	36 Kbit	288 Kbit
Total capacity	36.1 Mbit	75.9 Mbit	270 Mbit

FPGAs have three types of internal memory, LUT, Block, and Ultra RAMs. Table 1 lists the total capacity in VCU118, and Ultra RAM is available only in Xilinx's UltraScale series of FPGA devices.

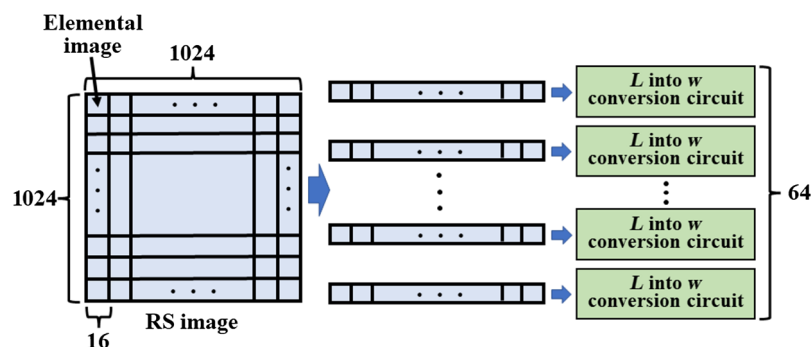
Ultra RAMs have only one RAM port, whereas Block RAMs have two ports and act as a true dual-port function that allows simultaneous access to the same RAM address from two ports. In this study, considering 1024×1024 pixels of RS images, we used Ultra RAMs for the light-ray into wavefront conversion circuit, whereas we used Block RAMs for the propagation and phase-only data generation unit.

In the hologram calculation unit, the RS image is first sent to the elemental image management unit through a first-in-first-out. This unit performs conversions of elemental images from light-ray information to wavefront information in 64 parallel. After processing all elemental images, the RS image is stored in the Block RAM unit in the propagation and phase-only data generation unit. This unit has a propagation calculation unit, including the FFT IP core unit, the propagation function unit for Eq. (11), and the phase-only data generation unit for Eq. (13). Sixteen calculation circuits in each of the propagation calculation unit and the phase-only data generation unit can be operated in parallel. After generating the phase-only data, the unit outputs the hologram.

3.1 Elemental Image Management Unit

In the light-ray into wavefront conversion circuit, random phase multiplication and 2D FFT are performed for each elemental image, as described in Sec. 2. Therefore, the units can be parallelized for each elemental image. In this case, the assumed sizes of RS and elemental images are 1024×1024 and 16×16 pixels, respectively, resulting in 64×64 elemental images. To implement parallel processing, as shown in Fig. 6, we introduced 64 light-ray into wavefront conversion circuits in parallel by horizontally dividing the RS image by 16 pixels, which is the vertical size of the elemental image.

Figure 7 shows the light-ray into wavefront conversion circuit. Sixty-four elemental images were stored in Ultra RAM, and each elemental image was retrieved and sequentially processed. Two LUT RAMs were used for storing each elemental image, one for FFT and the other for storage after quadrant transformation that swaps the high-frequency and low-frequency components of an image. The random phase was output by inputting the random number generated by the linear feedback shift register to pre-defined trigonometric tables. The 2D FFT was performed using an IP core for 1D FFT.

**Fig. 6** Method of parallelization in light-ray into wavefront conversion circuit.

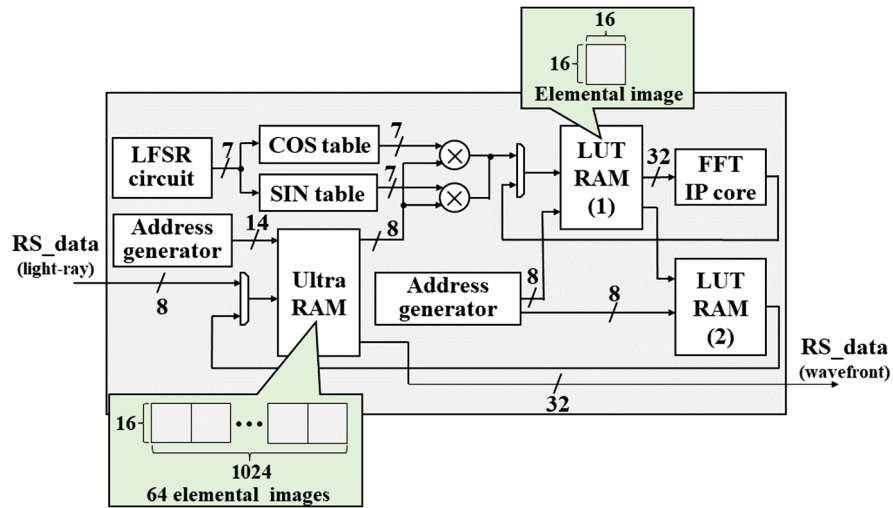


Fig. 7 Light-ray into wavefront conversion circuit.

3.2 Propagation Calculation Unit

This section describes the propagation calculation unit. The propagation calculation of Eq. (12) is performed in the following order: 2D FFT, complex multiplication with propagation function $G(m, n)$, and inverse 2D FFT. Because the Block RAM that manages data in this unit has two ports, one port is dedicated to sending data to the circuit and the other port is dedicated to receiving data from the circuit. If the target circuit can be processed in a pipeline, data can be continuously sent and received. For example, the FFT IP core has a pipeline mode; therefore, the data can be processed, as shown in Fig. 8.

The 2D FFT in this unit requires many clocks to perform 1D FFTs of 1024 points in the horizontal and vertical directions, and because the FFT is processed line-by-line, parallelizing the FFT processing for each row or column to increase the speed is possible. However, parallelization is difficult when storing RS images in a single Block RAM and using the two-port pipelined transmission/reception described above owing to the limited number of ports. Although the number of ports can be increased by increasing the number of Block RAMs, managing the addresses and data of each Block RAM becomes difficult. To solve this problem, we divided the RS image into small squares and stored each square in a Block RAM; thus, we were able to manage data while using bit conversion.

Figure 9 shows a case of a 4×4 pixels image divided into four parts. In the case of a horizontal 1D FFT, the red dashed frames in the figure indicate two parallel operations. The address in the upper red frame must be shifted to 0, 1, 0, 1, 2, 3, 2, and 3. As shown in Table 2, the address can be transitioned in the horizontal direction by setting both sides of the 3 bit sequence required for the count-up of the red frame to the address of the Block RAM. In addition, the center bit of the Block RAM can be used for the read/write enable signal (the RAM discrimination bit), easily reading/writing the left and right sides of RAM is possible as shown in the green frames. In the case of the vertical FFT, a bit conversion that swaps both sides of the 3 bit sequence, as shown in

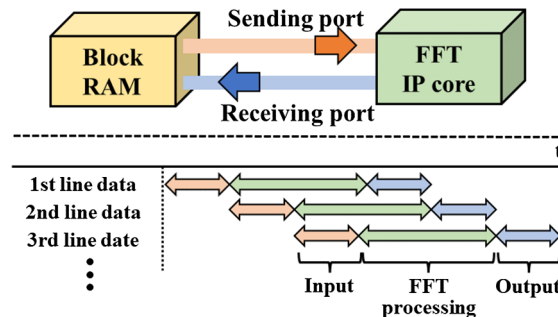


Fig. 8 Utilization of two ports of block RAM in FFT.

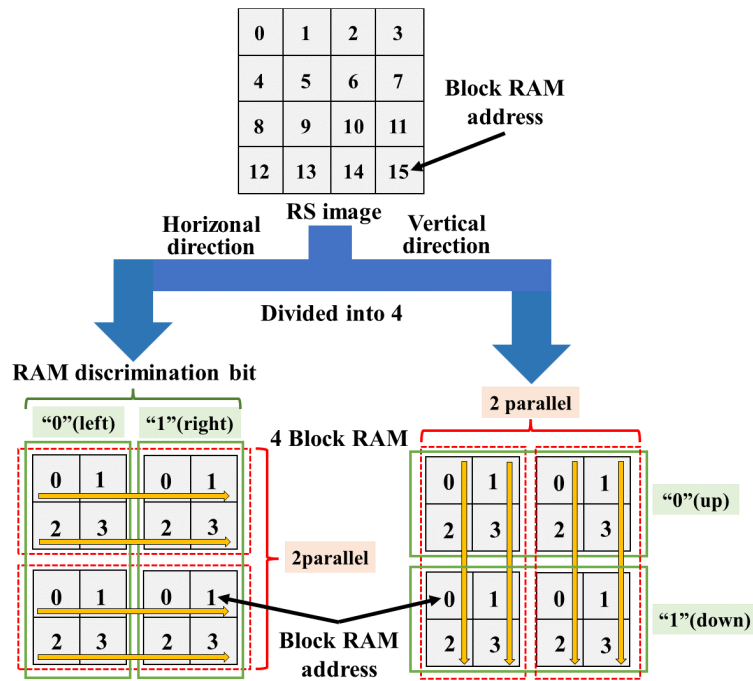


Fig. 9 Block RAM division (16 address RAM, four divisions).

Table 2 Example of 3 bit sequence in a 4 × 4 pixels image. In parentheses are decimal numbers.

3 bit	000	001	010	011	100	101	110	111
Both sides	00(0)	01(1)	00(0)	01(1)	10(2)	11(3)	10(2)	11(3)
Both sides (replace)	00(0)	10(2)	00(0)	10(2)	01(1)	11(3)	01(1)	11(3)
Central	0	0	1	1	0	0	1	1

Table 2, enables vertical address transitions 0, 2, 0, 2, 1, 3, 1, and 3. This enables address management of individual RAMs by the bit conversion.

This partitioning can be performed in 4^t (where t is a natural number), and the numbers of parallels and the number of Block RAM discriminators are the square numbers $\sqrt{(4^t)} = 2^t$. That is, the number of bits required to discriminate Block RAMs is t bits. If the address size of the RS image is 2^X bits and the number of divisions is $4^t = 2^{2t}$, the number of address sizes per Block RAM is $X - 2t$ bits. For example, in the case of an image with $1024 \times 1024 = 2^{20}$ pixels with 4^1 divisions, one bit is required to discriminate left/right or top/bottom, as described above, and the number of address bits in the Block RAM is $20 - 2 \times 1 = 18$ bits. In this study, Block RAMs was implemented with $256 = 4^4$ partitions. Therefore, the number of discrimination bits in Block RAM was 4, and the number of addresses in Block RAM was $20 - 2 \times 4 = 12$ bits. Figure 10 shows a schematic of the Block RAM unit and the propagation calculation unit.

Next, we describe the propagation function circuit that performs complex multiplication with propagation function $G(m, n)$. The propagation function circuit is shown in Fig. 11. As complex multiplication is a pixel-by-pixel process, the calculations are performed in a pipeline using the method shown in Fig. 8. Equation (11) can be expressed as $\exp(j\theta) = \cos \theta + j \sin \theta$ using Euler’s formula. After calculating θ , the real and imaginary parts of $G(m, n)$ were calculated using the trigonometric function tables prepared in advance. The complex exponential in Eq. (11) is normalized by 2π . Therefore, the value of the trigonometric function was obtained by inputting 11 bits to the tables after the decimal point of the value, which is the multiplication result of $m^2 + n^2$ and z_{param} .

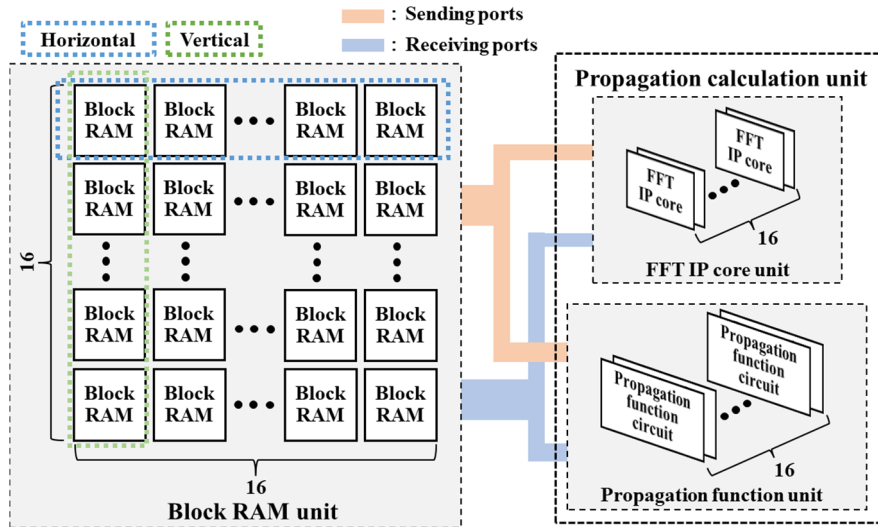


Fig. 10 Block RAM unit and propagation calculation unit.

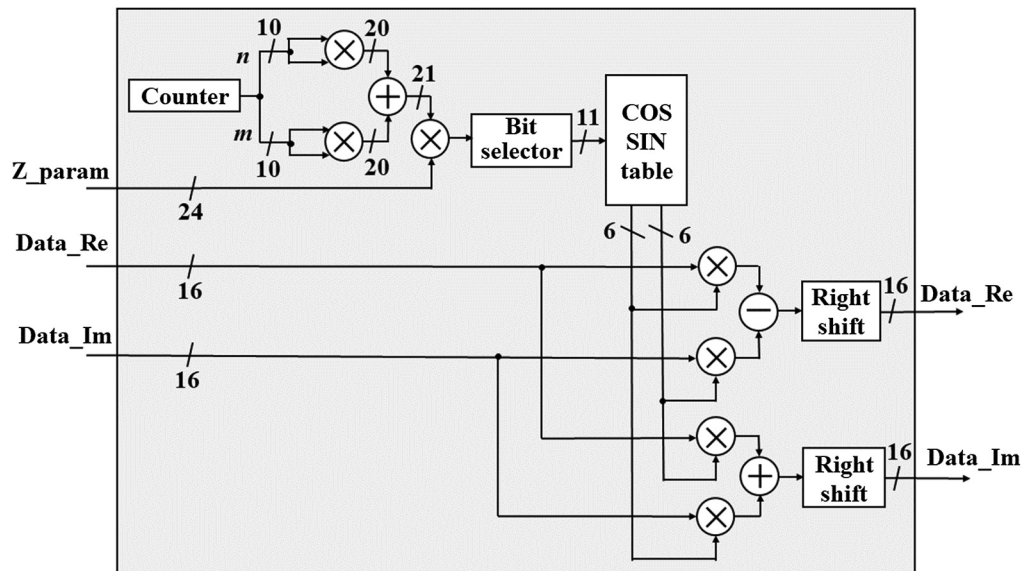


Fig. 11 Propagation function circuit.

3.3 Phase-Only Data Generation Unit

Finally, we describe the phase-only data-generation unit. In phase-only data generation, the real and imaginary parts of a pixel are normalized, as shown in Fig. 12, and the arctangent was calculated by inputting them into an LUT, as shown in Fig. 13. The phase-only generation module had a latency of 20 clocks (including 2 clocks for input and output of the circuit, 16 clocks for normalization, and 2 clocks for input and output of the table). To hide this latency, 20 phase-only generation modules were implemented. By switching the ports of the Block RAM unit shown in

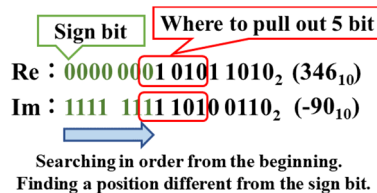


Fig. 12 Normalization method.

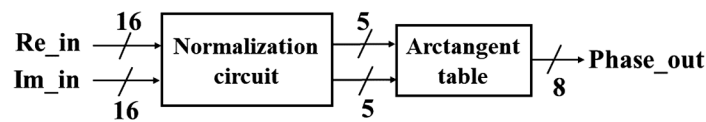


Fig. 13 Phase-only generation module.

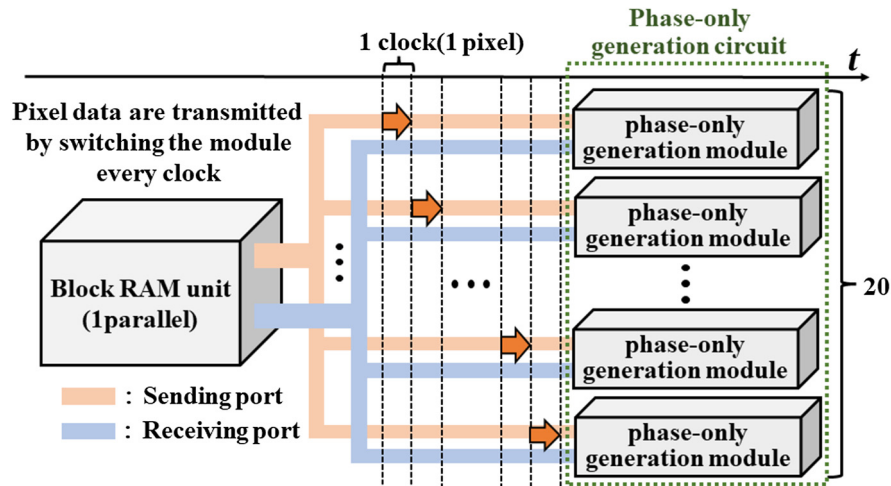


Fig. 14 Relationship between block RAM unit and phase-only generation circuit.

Table 3 Number of clocks for individual unit.

	L into w conversion	Unit-to-unit transfer	Propagation calculation	Phase-only data generation	Total
Clocks	256,641	65,542	337,135	65,560	724,878
Time (ms)	1.83	0.468	2.41	0.468	5.18

Fig. 14, generating phase-only data while continuously inputting and outputting the data is possible.

Table 3 lists the number of clocks for each unit in the entire circuit and the processing time when the operating frequency is 140 MHz and shows that the hologram calculation time was 5.18 ms.

4 Implementation of Data Transfer Function

This section describes the data transfer of the host PC to an output device. In this study, data was transferred via high-definition multimedia interface (HDMI) transfer using an HDMI 2.0 FMC card developed by the Tokyo Electron Device LCD. This product was connected to the FPGA mezzanine card (FMC) pin on the FPGA board and used the “HDMI 1.4/2.0 Transfer Subsystem” IP core provided by Xilinx for enabling data transfer. Figure 15 shows a schematic of the system using an HDMI 2.0 FMC card.

The transmission using the HDMI IP core was established based on the sample design provided by Xilinx. The transfer flow is described as follows:

1. The Rx and Tx sides of the FMC card are connected to the host PC and output device, respectively.
2. By displaying the RS image on a screen of the host PC, the RS image is sent from the host PC to the Rx side of the FMC card.

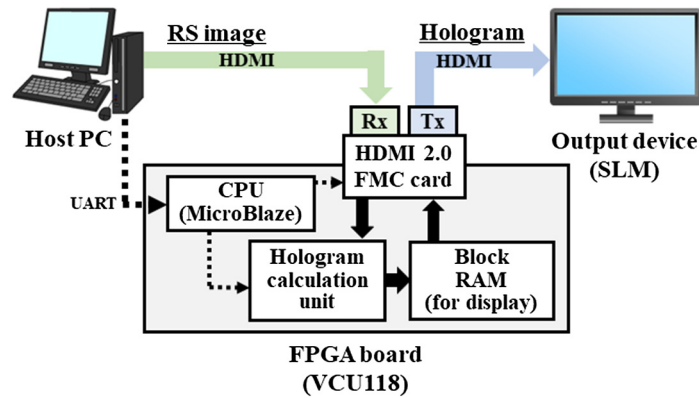


Fig. 15 Schematic of a system using HDMI 2.0 FMC card.

3. When the RS image is received by the FPGA, hologram calculation is performed in the FPGA, and the hologram is stored in Block RAM for displaying after the completion of calculation.
4. Hologram is transmitted from the Block RAM to the Tx side for displaying on the output device.

MicroBlaze, which is a soft macro central processing unit (CPU), is placed in the FPGA to set the resolution, frame rate, and output mode, which are parameters required for the hologram calculation, via universal asynchronous receiver transmitter communication.

Next, we describe video frame transmission. Frame data are transmitted within the FPGA using an advanced extensible interface (AXI) Stream protocol. The HDMI IP core used in this study had transmission specifications of 2 pixel per clock. Because one pixel is 24 bits of RGB, the data width of one transmission data is $24 \text{ bit} \times 2 = 48 \text{ bits}$. Therefore, if the frame size to be transmitted is 3140×2160 pixels and the hologram size to be displayed (displayed in the center) is 1024×1024 pixels, the frame image to be transmitted is as shown in Fig. 16. In this study, grayscale (8-bit) objects are used as subjects. Since data transmission and reception via HDMI are performed with 24-bit color data, the same grayscale data are stored in red, green, and blue channels.

We describe the timing chart for the video frames and hologram calculations. Consider the case in which a frame is displayed, as shown in Fig. 16, and the operating frequency of the hologram calculation unit is 140 MHz.

In the AXI Stream protocol used for transmission, “TUSER,” “TVALID,” “TDATA,” and “TLAST” signals indicate the first data in the frame to be transmitted, validity of the data, pixel data, and the last data in the horizontal line, respectively. However, owing to the specifications of the sample design, changing the timing of these signals to match that of the output of the hologram calculation unit was not possible. Therefore, while these signals were passed from Rx to Tx, only “TDATA” was rewritten to transmit the frame. Assuming a latency of 10 clocks between

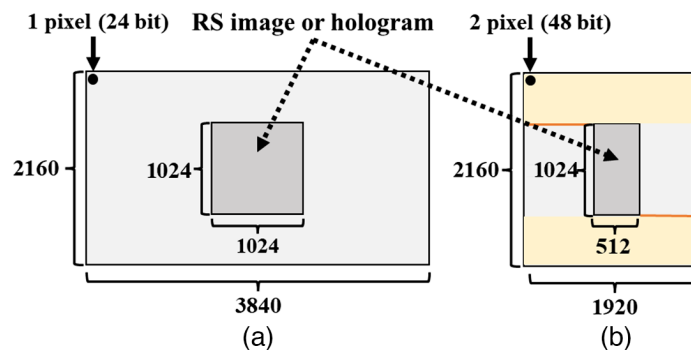


Fig. 16 Difference between frames on display and FPGA board: (a) frame when displaying and (b) frame in FPGA board.

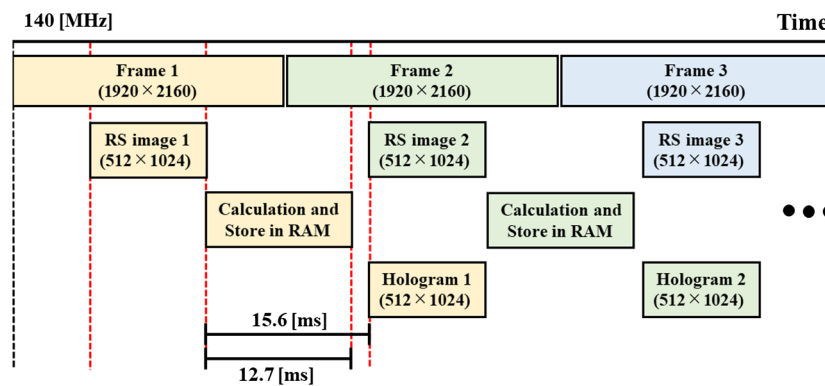


Fig. 17 The timing chart in data transfer.

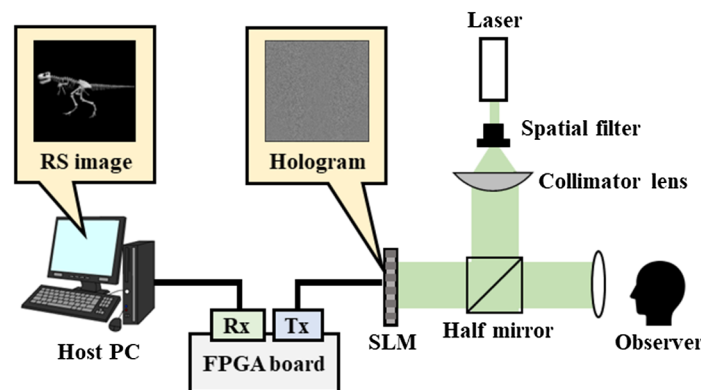


Fig. 18 Optical reconstruction system connected to FPGA board.

frames, after the transmission of the 512×1024 pixels area in the first frame until the arrival of 512×1024 pixels area in the second frame, a latency equal to the number of pixels in the yellow area and orange lines exist, as shown in Fig. 16(b). The number of clocks for these were calculated as follows:

$$2 \times \left\{ 1920 \times \frac{(2160 - 1024)}{2} \right\} + 2 \times \frac{(1920 - 512)}{2} + 10 = 2,182,538 \text{ clocks.} \quad (14)$$

This can be converted to a time of 15.6 ms. Therefore, the hologram calculation should be performed and stored in the RAM within 15.6 ms. Based on the number of clocks calculated in Table 3, the hologram calculation and storage in RAM for display require 5.18 and $(1024 \times 1024 \text{ clocks} / 140 \text{ MHz} =) 7.49$ ms, respectively, the total time is 12.7 ms. Therefore, the results are within 15.6 ms. Figure 17 shows a timing chart summarizing the above process.

When frame 1 (yellow) arrives, the first RS image (yellow) is acquired and the hologram calculation is performed after the completion of acquisition. When the next frame 2 (green) arrives, the first hologram (yellow) is output as soon as the second RS image (green) arrives. This operation is performed frame by frame. Figure 18 shows the optical reconstruction system connected to the FPGA board and SLM.

5 Implementation Result

5.1 Resource Usage

Table 4 shows the circuit size of the designed circuit in the FPGA board of VCU118. The operating frequency was 140 MHz, and the digital signal processor (DSP) slice was an intrinsic sum-of-products hardware accelerator that speeds up the execution of signal-processing functions.

Table 4 Hardware resource utilization of the designed circuit.

Resource name	Number of uses	Use rate (%)
Register	269,659	22.81
LUT RAM	4156 Kbit	10.97
Block RAM	51,246 Kbit	65.90
Ultra RAM	73,728 Kbit	26.67
DSP slice	384	5.610

Table 5 Overall processing time.

Device	Transfer RS image and store data in RAM for display	Hologram calculation	Total	Frame rate
CPU	—	211 ms	211 ms	4.74 fps
FPGA	22.1 ms	5.18 ms	27.3 ms	36.6 fps
Acceleration rate	—	40.7	7.73	—

5.2 Calculation Time

Table 5 lists the processing time results for the CPU (Intel(R) Core(TM) i9-9900K 3.6 GHz) and the FPGA board. The calculation time on the FPGA was measured by observing the counter value using a logic analyzer. Hologram calculation in the FPGA board was 40.7 times faster than that in the CPU, and the FPGA board was able to achieve 36.6 frames per second (fps) as an overall process.

5.3 Optical Reconstruction

We used three 3D objects (a cube, ring, and dinosaur). Figure 19 shows the 3D objects, RS images acquired from them, and optical reconstruction results obtained using the constructed system. RS images were obtained from the object data using OpenGL. The pixel pitch, wavelength, and propagation distance were set as $3.74 \mu\text{m}$, 532 nm , and 0.3 m , respectively.

Figure 19 confirms that the 3D objects can be reproduced optically. As described in Sec. 3, the special-purpose computer developed in this study can handle RS images of 1024×1024 pixels. However, because the number of elemental images was 64×64 , the resolution of the reconstructed image was $\sim 64 \times 64$ pixels. Therefore, although the quality of the reconstructed image

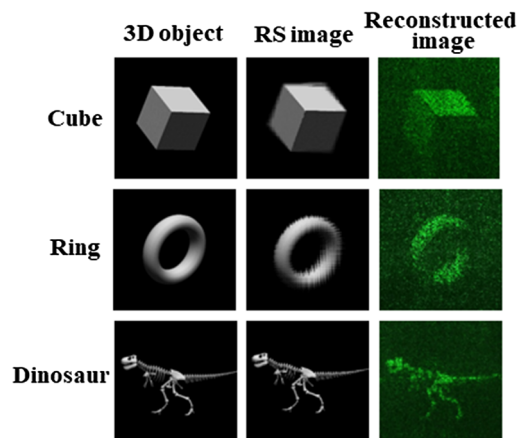


Fig. 19 3D objects, RS images, and reconstructed images (Video 1, MP4, 1.2 MB [URL: <https://doi.org/10.1117/1.OE.62.8.085102.s1>]).

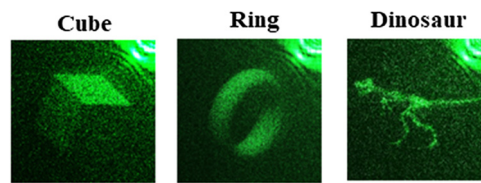


Fig. 20 Reconstructed images of 2048×2048 pixels by CPU.

was coarse, it succeeded in reproducing hidden planes and shadings, which are difficult to achieve using HORN based on the point cloud method.

To improve resolution while maintaining the shading of the reconstructed image, for example, the number of pixels of the RS image can be increased to 2048×2048 pixels. For the size of the elemental image (16×16 pixels), the resolution of the reconstructed image is $\sim 128 \times 128$ pixels, which is a fourfold improvement in the resolution. Figure 20 shows the actual reproduction results from a hologram created by the CPU simulation.

The special-purpose computer developed in this study can only handle RS images of 1024×1024 pixels, owing to the limited memory capacity of the FPGA. The reasons for this are as follows: the special-purpose computer developed in this study uses internal RAMs in the FPGA shown in Table 1. The propagation and phase-only data generation units use Block RAM, and the utilization of Block RAM on VCU118 was 65.9%. If the same circuit configuration is used to output a hologram with 2048×2048 pixels, a simple calculation shows requirement of four times the current hologram, resulting in a utilization rate of 263.6%. In the case of Ultra RAM in the light-ray into wavefront conversion circuit, the utilization rate was 26.67%. Multiplying by four, the utilization rate of Ultra RAM is 106.68%. The insufficiency of the largest Ultra RAM can be related to the bit width of RAM. Whereas Block RAM has a variable bit width of 16 or 36 bits, Ultra RAM has a fixed bit width of 72 bits. Therefore, a bit width of 32 bits for the real and imaginary parts, which is the bit width implemented in this study, resulted in a 40-bit surplus. The light-ray into wavefront conversion circuit using Ultra RAM has this surplus.

One solution is to allocate two-pixel data to one address to completely exploit the RAM's capacity. In this case, reconsidering address management such as bit conversion, which can be used by assigning one pixel of data to one address, is necessary. Other solutions include using boards with FPGA chips that have larger Block RAM or Ultra RAM, or using double data rate (DDR) memory on the FPGA board. DDR memory is more difficult to handle than Block and Ultra RAM, but the memory of VCU118 dealt with this time is ~ 5 GB and is suitable for parallel processing; therefore, we think that the parallelization and address management described in Sec. 3 can be applied. Furthermore, using the orthographic ray-wavefront conversion method reduces the computational load and creates large-scale holograms by dividing holograms into tiles.³⁴

6 Conclusion

We developed a special-purpose computer for holography, UR-HORN, using the ray-wavefront conversion method, which is superior in hidden plane processing, to accelerate the speed of hologram computations. We parallelized the processing of each elemental image when converting light-ray information into wavefront information. We also parallelized the propagation calculation and phase-only data generation by devising an address management method for the Block RAM. As a result, we successfully implemented a circuit that generated a 1024×1024 pixels hologram in 5.18 ms with an FPGA operating frequency of 140 MHz. In addition, by constructing a system that can transfer input/output data via HDMI, we successfully controlled the communication between the host PC, special-purpose computer system, and SLM in a single direction. Consequently, real-time video reproduction at 30 fps was achieved.

However, the maximum hologram size was 1024×1024 pixels, which is largely related to the limited memory capacity of the FPGA, and can be solved using following solutions: reconsideration of the data and address management in Ultra RAM, to use boards with FPGA chips equipped with large RAM and to use DDR memory on FPGA boards. Another solution is to divide holograms into tiles and use multiple FPGAs to compute the segmented regions. When the

resolution of the hologram data is 1024×1024 pixels, the pixel pitch of the hologram is $3.74 \mu\text{m}$, so the size of the hologram displayed is ~ 4 mm square. Therefore, it would normally be possible to observe the reproduced image from multiple viewpoints, but because the size of the hologram is insufficient, it is difficult to observe the image from multiple viewpoints. We would therefore like to increase the size of the holograms that can be calculated so that the reproduced image can be observed from multiple viewpoints.

Code, Data, and Materials Availability

The code, data, and materials that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Nos. JP17K18425 and JP19H01097.

References

1. P. S. Hilaire et al., "Electronic display system for computational holography," *Proc. SPIE* **1212**, 174–182 (1990).
2. N. Hashimoto et al., "Real-time holography using the high-resolution LCTV-SLM," *Proc. SPIE* **1461** (1991).
3. Y. Ichihashi and K. Yamamoto, "Integral photography capture and electronic holography display," *Proc. SPIE* **9117**, 91170P (2014).
4. T. Kurihara and Y. Takaki, "Shading of a computer-generated hologram by zone plate modulation," *Opt. Express* **20**(4), 3529–3540 (2012).
5. M. E. Lucente, "Interactive computation of holograms using a look-up table," *J. Electron. Imaging* **2**(1), 28–34 (1993).
6. S. C. Kim and E. S. Kim, "Effective generation of digital holograms of three-dimensional objects using a novel look-up table method," *Appl. Opt.* **47**(19), D55–D62 (2008).
7. T. Nishitsuji et al., "Fast calculation of computer-generated hologram using the circular symmetry of zone plates," *Opt. Express* **20**(25), 27496–27502 (2012).
8. Y. Pan et al., "Fast CGH computation using S-LUT on GPU," *Opt. Express* **17**(21), 18543–18555 (2009).
9. T. Shimobaba et al., "Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane," *Opt. Lett.* **34**(20), 3133–3135 (2009).
10. P. W. M. Tsang and T. C. Poon, "Fast generation of digital holograms based on warping of the wavefront recording plane," *Opt. Express* **23**(6), 7667–7673 (2015).
11. N. Masuda et al., "Computer generated holography using a graphics processing unit," *Opt. Express* **14**(2), 603–608 (2006).
12. Y. Ichihashi et al., "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. Express* **20**(19), 21645–21655 (2012).
13. H. Niwase et al., "Real-time spatiotemporal division multiplexing electroholography with a single graphics processing unit utilizing movie features," *Opt. Express* **22**(23), 28052–28057 (2014).
14. T. Ito et al., "Special-purpose computer HORN-1 for reconstruction of virtual image in three dimensions," *Comput. Phys. Commun.* **82**(2–3), 104–110 (1994).
15. T. Ito et al., "Special-purpose computer for holography HORN-2," *Comput. Phys. Commun.* **93**(1), 13–20 (1996).
16. T. Shimobaba et al., "Special-purpose computer for holography HORN-3 with PLD technology," *Comput. Phys. Commun.* **130**(1–2), 75–82 (2000).
17. T. Shimobaba et al., "Special-purpose computer for holography HORN-4 with recurrence algorithm," *Comput. Phys. Commun.* **148**(2), 160–170 (2002).
18. T. Ito et al., "Special-purpose computer HORN-5 for a real-time electroholography," *Opt. Express* **13**(6), 1923–1932 (2005).
19. Y. Ichihashi et al., "HORN-6 special-purpose clustered computing system for electroholography," *Opt. Express* **17**(16), 13895–13903 (2009).
20. N. Okada et al., "Special-purpose computer HORN-7 with FPGA technology for phase modulation type electro-holography," in *IDW/AD, Proc. Int. Display Workshops, '12, 3Dp-26* (2012).
21. T. Sugie et al., "High-performance parallel computing for next-generation holographic imaging," *Nat. Electron.* **1**(4), 254–259 (2018).
22. Y. Yamamoto et al., "HORN-9: special-purpose computer for electroholography with the Hilbert transform," *Opt. Express* **30**(21), 38115–38127 (2022).
23. K. Matsushima, "Computer-generated holograms for three-dimensional plane objects with shade and texture," *Appl. Opt.* **44**(22), 4607–4614 (2005).

24. K. Matsushima and S. Nakahara, "Extremely high-definition full-parallax computer-generated hologram created by the polygon-based method," *Appl. Opt.* **48**(34), H54–H63 (2009).
25. F. Okano et al., "Real-time pickup method for a three-dimensional image based on integral photography," *Appl. Opt.* **36**(7), 1598–1603 (1997).
26. J. Arai et al., "Gradient-index lens-array method based on real-time integral photography for three-dimensional images," *Appl. Opt.* **37**(11), 2034–2045 (1998).
27. T. Yatagai, "Stereoscopic approach to 3-D display using computer-generated holograms," *Appl. Opt.* **15**(11), 2722–2729 (1976).
28. T. Mishina et al., "Calculation of holograms from elemental images captured by integral photography," *Appl. Opt.* **45**(17), 4026–4036 (2006).
29. K. Wakunami and M. Yamaguchi, "Calculation for computer generated hologram using ray-sampling plane," *Opt. Express* **19**(10), 9086–9101 (2011).
30. H. Sato et al., "Real-time colour hologram generation based on ray-sampling plane with multi-GPU acceleration," *Sci. Rep.* **8**(1), 1500 (2018).
31. S. Igarashi et al., "Fast method of calculating a photorealistic hologram based on orthographic ray-wavefront conversion," *Opt. Lett.* **41**(7), 1396–1399 (2016).
32. K. Yamamoto et al., "Hologram printing for next-generation holographic display," *Proc. SPIE* **10557**, 105570O (2018).
33. Y. Yamamoto et al., "Special-purpose computer for digital holographic high-speed three-dimensional imaging," *Opt. Eng.* **59**(5), 054105 (2020).
34. S. Igarashi et al., "Efficient tiled calculation of over-10-gigapixel holograms using ray-wavefront conversion," *Opt. Express* **26**, 10773–10786 (2018).

Tatsuya Maruyama is an ME student at the Graduate School of Engineering, Chiba University. He is also a cooperating researcher at the National Institute of Information and Communications Technology (NICT). He received his BE degree from Chiba University in 2021. His current research interests include computer holography. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).

Yasuyuki Ichihashi is a senior researcher at NICT. He received his BE, ME, and DE degrees from Chiba University in 2005, 2007, and 2010, respectively. His current research interests include 3D imaging and reproducing systems and calculating holograms for a wavefront printer. He is a member of IEICE and the Institute of Image Information and Television Engineers (ITE).

Ikuo Hoshi is a researcher at NICT. He received his BE, ME, and DE degrees from Chiba University in 2018, 2020, and 2022, respectively. His research interests include computational imaging, machine-learning, and special-purpose computing. He is a member of IEICE.

Takashi Kakue is an associate professor at Chiba University. He received his BE, ME, and DE degrees from Kyoto Institute of Technology in 2006, 2008, and 2012, respectively. His current research interests include holography, 3D display, 3D measurement, and high-speed imaging. He is a member of SPIE, Optica, the Institute of Electrical and Electronics Engineers (IEEE), Optical Society of Japan (OSJ), and ITE.

Tomoyoshi Shimobaba is a professor at Chiba University. He received his PhD in fast calculation of holography using special-purpose computers from Chiba University in 2002. His current research interests include computer holography and its applications. He is a member of SPIE, Optica, IEICE, OSJ, and ITE.

Tomoyoshi Ito is a professor at Chiba University. He received his PhD in special-purpose computers for many-body systems in astrophysics and molecular dynamics from the University of Tokyo in 1994. His current research interests include high-performance computing and its applications for electro-holography. He is a member of ACM, Optica, OSJ, ITE, IEICE, the Information Processing Society of Japan, and the Astronomical Society of Japan.