



Variable-intensity line 3D images drawn using kinoform-type electroholography superimposed with phase error

TAKASHI NISHITSUJI,^{1,*}  NOBUYA SHIINA,² DAVID BLINDER,^{3,4} 
TOMOYOSHI SHIMOBABA,⁵  TAKASHI KAKUE,⁵  PETER
SCHELKENS,^{3,4}  TOMOYOSHI ITO,⁵  AND TAKUYA ASAKA¹

¹Faculty of Systems Design, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino, Tokyo, Japan

²Graduate school of Systems Design, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino, Tokyo, Japan

³Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussel, Belgium

⁴IMEC, Kapeldreef 75, B-3001 Leuven, Belgium

⁵Graduate School of Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba, Chiba, Japan

*nishitsuji@tmu.ac.jp

Abstract: Three-dimensional (3D) display using electroholography is a promising technology for next-generation television systems; however, its applicability is limited by the heavy computational load for obtaining computer-generated holograms (CGHs). The CG-line method is an algorithm that calculates CGHs to display 3D line-drawn objects at a very high computational speed but with limited expressiveness; for instance, the intensity along the line must be constant. Herein, we propose an extension for drawing gradated 3D lines using the CG-line method by superimposing phase noise. Consequently, we succeeded in drawing gradated 3D lines while maintaining the high computational speed of the original CG-line method.

© 2022 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

1. Introduction

Alongside the growing interest in virtual and augmented reality technology, applications and devices that visually realize telepresence have become increasingly popular. For example, telepresence is currently led by head-mounted displays [1,2] and volumetric displays [3–5], which are already commercialized. Although these devices provide users with three-dimensional (3D) visual perception, problems such as visual fatigue and limited realism have hindered their widespread adoption.

In this context, electroholography is a promising 3D display technology for realizing telepresence because, theoretically, 3D images reconstructed using electroholography perfectly reproduce the reflected light of the 3D object, thereby eliminating visual fatigue and providing high realism. However, substantial limitations have prevented the popularization of electroholography in everyday use. One problem is the heavy computational processing load for calculating computer-generated holograms (CGHs) in real time. CGHs are generated as two-dimensional (2D) complex-valued amplitude data representing the wavefield and encoding the 3D display information. They can be calculated by simulating the wave propagation of virtual 3D objects. The computational requirements of CGH calculations are exceptionally high; thus, high-performance computing systems and algorithms are required for practical applications of electroholography.

To date, several studies regarding high-performance computations for electroholography have been reported [6,7]. Among such studies are dedicated computers based on field programmable gate arrays [8–10], computational systems based on graphic processing units (GPUs) [11,12], and application-specific circuits [13]. Additionally, there are several sophisticated algorithms for fast CGH calculations. Many different solutions exist, trading-off speed, accuracy, and support visual

effects, as well as several acceleration techniques [14]. CGH techniques can be classified based on their constituents, such as point-based [15], polygon-based, ray-tracing, geometric primitives, or using layer-based techniques. We will elaborate on the latter.

The layer-based approach is for creating CGH of 3D objects partitioned into layers, where every component is assigned to its closest layer [16–18]. That way, the computation of the wavefield is spatially localized, benefiting calculation time. Because the layers are placed parallel to the hologram plane, one can use efficient numerical convolution calculation to propagate them, such as the angular-spectrum method (ASM). The calculation time is roughly proportional to the number of layers and the CGH resolution, but does not depend very much on the 3D object complexity in each layer; thus, the layer-based method is effective for calculating the 3D object composed of complex textures on a few layers. On the other hand, our proposed method could be said to belong to the geometric primitive category, decomposing objects into curve segments.

Moreover, multiple acceleration techniques exist to speed up the computation of these elements by leveraging their symmetries. Look-up table (LUT) based approaches are memory-based method calculations that pre-calculate wavefront segments created from the representative elements of the 3D object. The elements are, for example, the point-light sources [19–24], and polygons [25]. Although the LUT method reduces the heavy calculation of wave propagation by using stored data, the required memory space to store the pre-calculated data can be a severe problem, especially for memory caching. Therefore, it is of interest to reduce the LUT size in memory. This can be achieved e.g. by utilizing their geometric symmetries [19–22], signal separability [23,24], and principle component analysis [25]. Sparse CGH is named after the signal processing concept, where a signal (in this case, a wavefield) can be efficiently expressed by a few coefficients in the right transform space. Examples include wavefront recording planes, coefficient shrinking techniques and holographic stereogram approximations. In recent years, deep-learning-based methods [26] for CGH acceleration have emerged as well. Its success is attributable to many factors, such as its general-purpose nature and suitability for modern GPUs. They can serve as an accelerator or even a substitute for different algorithmic CGH components. Examples include computing numerical diffraction [27], improving low bit-depth [28] CGH quality, speckle denoising [29], neural camera-in-loop holography [30] or even a full RGB+D CGH deep learning pipeline [31]. While the deep-learning based method can offer high performance in terms of image quality and computation speed, it has some drawbacks such as requiring a large amount of training data, sensitivity to outlier input data, and poor explainability.

The present authors previously proposed the CG-line method for generating CGHs of 3D line-drawn objects such as outlined characters and wire-frame art [32–36]. Unlike the other related methods introduced in the above section, the CG-line method restricts the expressiveness of a 3D object to display only a 3D object comprising the lines with constant intensity. Since the CG-line method does not require signal conversion operation such as the fast Fourier transform (FFT) or pre-calculated data, which are required in most sparsity-based, layer-based, and LUT-based methods, it is suitable for hardware implementations (e.g., GPU) with a small computational load. Further, the CG-line method only calculates the wavefront according to the line shape; the computational performance does not directly depend on the CGH resolution, i.e., the CG-line method is more effective for high-resolution CGH with a simple 3D image. Consequently, the CG-line method greatly accelerates the speed of CGH calculation compared with the conventional method with a GPU implementation [33] while realizing an interactive display system with hand-drawn interfaces [34]. However, in practical applications such as the 3D displays of car navigation systems, enhancing the limited expressiveness is a significant challenge using the CG-line method. Some examples of such enhancements include adding gradation and thickening the lines. Therefore, the present study aims to enhance the expressiveness of the CG-line method by adding gradations on the line-drawn object.

Most of the current CGHs are either amplitude-modulated or phase-modulated. These types of CGHs are regularly used owing to limitations of the spatial light modulator (SLM), a modulating device that reconstructs the reflected light recorded in CGHs. In general, phase-modulation-type CGHs reconstruct brighter 3D images than amplitude-modulation-type CGHs because phase modulation is more efficient than amplitude modulation regarding the light utilization. Therefore, the present study focuses on the phase-modulation type, specifically, the kinoform CGH [37] obtained by extracting and quantizing the phase information from the complex wavefront of the hologram.

The remainder of this paper is organized as follows. Section 2 overviews the CG-line method and Section 3 describes the details of the proposed method. Section 4 presents and discusses the experimental results. Section 5 concludes the study.

2. CG-line method for calculating CGHs of line-drawn 3D objects

2.1. Theory

The CGH calculation performs a linear superposition of the spherical wavefronts emitted from point light sources (PLSs) on the virtual hologram plane. The wavefront created via a single PLS located at $(\delta, \epsilon, \zeta)$ is defined as

$$P(x, y) = \frac{a}{\zeta} \cdot \exp\left(\frac{i\pi}{\lambda\zeta} [(x - \delta)^2 + (y - \epsilon)^2]\right), \quad (1)$$

where (x, y) are the coordinates on the hologram plane, ζ is the distance between PLS and the hologram plane, i is the imaginary unit, λ is the wavelength of the incident light, and a is the amplitude of PLS.

As lines constitute a set of points, the CGH calculation of a line-drawn object can be interpreted as the superposition of spherical wavefronts emitted from PLSs aligned on the lines. In the special case of an infinitely long straight line on the x -axis at a certain depth from a hologram plane, the wavefronts form a simple pattern on the hologram plane. The wavefront created from the line is given by

$$L(x, y) = \frac{1}{\zeta} \int_{-\infty}^{\infty} a(u) \cdot \exp\left(\frac{i\pi}{\lambda\zeta} [(x - u)^2 + y^2]\right) du, \quad (2)$$

where $a(u)$ is the amplitude of PLS at u . When the amplitudes of all PLS are equal, i.e., $a(u) = c \in \mathbb{C}$, Eq. (2) can be transformed to a pure y -function using the Fresnel integral as

$$U(y) = a\sqrt{\frac{\lambda}{\zeta}} \exp\left\{\pi i \left(\frac{y^2}{\lambda\zeta} + \frac{1}{4}\right)\right\}. \quad (3)$$

As the constant phase shift in Eq. (3) does not affect the final hologram, we can ignore the term $1/4$. The equation then simplifies to

$$U(y) = a\sqrt{\frac{\lambda}{\zeta}} \exp\left(\frac{i\pi y^2}{\lambda\zeta}\right). \quad (4)$$

In practice, the effective range of y in $U(y)$ is defined according to the diffraction limit of SLM, i.e.,

$$U(y) = \begin{cases} \text{Eq. (4)} & (y < R_\zeta), \\ 0 & (\text{otherwise}) \end{cases}$$

where

$$R_\zeta = \frac{\zeta\lambda}{\sqrt{4p^2 - \lambda^2}}, \quad (5)$$

and p is the pixel pitch of the SLM. Therefore, the wavefronts from a straight-line object with infinite length converge in a manner similar to imageically duplicating a one-dimensional (1D) wavefront. Thus, the superposition of 2D spherical wavefronts of PLSs can be substituted by the 1D wavefront $U(y)$ along the line, which can drastically reduce the computational time of CGHs.

The CG-line method calculates the CGH of every line type drawn on the same depth plane by superimposing $U(y)$ s along the normal direction of the lines. Figure 1 provides an overview of the CG-line method. As shown in Fig. 1, the CG-Line method superimposes the 1D wavefront formulated in Eq. (4) along the normal direction of the lines on the hologram plane; thus, the aperture shape on the hologram plane corresponds to the line shapes. Note that the 3D object is decomposed into lines and curves parallel to the hologram plane (SLM), i.e., the 3D object partitioned into a stack of 2D line-drawn object layers all parallel to the SLM plane. For a line-drawn 3D object with N segments, we describe the k -th segment using parametric functions $\vec{g}_k(t) = (x_k(t), y_k(t))$, where $\{t|0 \leq t \leq 1, t \in \mathbb{R}\}$. The normal vector $\vec{h}_k(t)$ of the line passing through a certain point on the hologram plane $\vec{r} = (\alpha, \beta)$ then becomes

$$\vec{h}_k(t) = \vec{g}_k(t) - \vec{r}, \quad (6)$$

$$= (x_k(t) - \alpha, y_k(t) - \beta). \quad (7)$$

Here, $\vec{h}_k(t)$ should satisfy

$$\vec{h}_k(t) \cdot \vec{g}'_k(t) = 0, \quad (8)$$

$$\rightarrow \{x_k(t) - \alpha\}x'_k(t) + \{y_k(t) - \beta\}y'_k(t) = 0, \quad (9)$$

where the apostrophe denotes differentiation with respect to t . Assuming that $x_k(t)$ and $y_k(t)$ are polynomials, Eq. (9) can be analytically solved when the maximum degree of $x_k(t)$, $y_k(t)$ is two or lower [33]. Therefore, given that T_j is the j -th real-number solution of Eq. (9), the complex amplitude distribution at $\vec{r} = (\alpha, \beta)$ becomes

$$L_k(\vec{r}) = \sum_j U(|\vec{h}_k(T_j)|), \quad (10)$$

where

$$|\vec{h}_k(T_j)| = \sqrt{\{x_k(T_j) - \alpha\}^2 + \{y_k(T_j) - \beta\}^2}. \quad (11)$$

Because Eq. (10) can be calculated independently for each pixel on the hologram plane, it can be efficiently implemented on a massively parallel processor such as a GPU. Finally, the kinoform CGH is obtained as

$$c(\vec{r}) = \arg \left(\sum_k^N L_k(\vec{r}) \right) \cdot \frac{2^b - 1}{2\pi}, \quad (12)$$

where $\arg(\cdot)$ is an operator taking the argument and b is the quantization bit length of the SLM. Note that the line width of the reconstructed line-drawn object is almost the same as the SLM's pixel pitch, which was experimentally investigated in [32] for the original CG-line method.

Because the maximum degree of the line-drawing object polynomial $\vec{g}(t)$ is restricted to two or below, the CG-line method assumes three types of lines: quadratic Bezier curves, arcs, and straight lines. Figure 2 provides the overview of the CG-line method for the three types of lines. In the following subsections, we explain each model in detail.

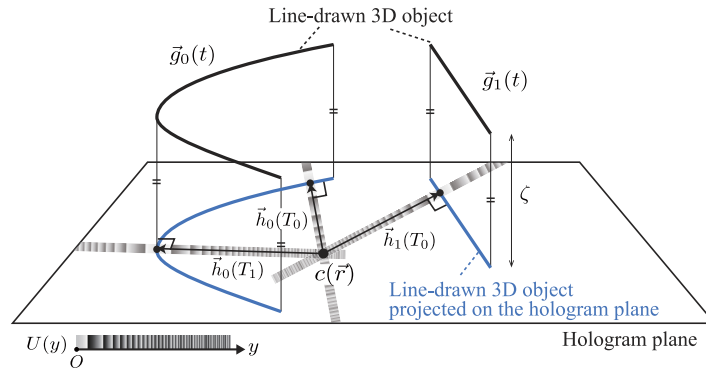


Fig. 1. Overview of the CG-line method for CGH calculations.

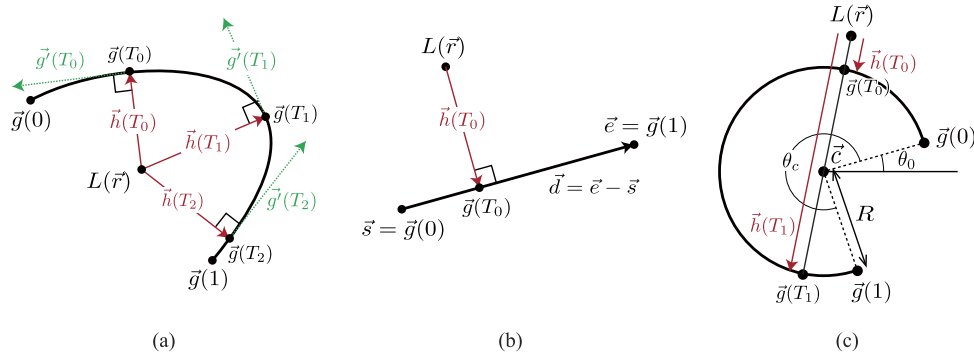


Fig. 2. Obtaining the CGH wavefront of: (a) a quadratic Bezier curve object, (b) a straight line, and (c) an arc.

2.1.1. Quadratic Bezier curve

For the quadratic Bezier curve, $\vec{g}(t) = (x(t), y(t))$ is defined as

$$x(t) = (1 - t)^2 x_0 + 2(1 - t)t x_1 + t^2 x_2, \quad (13)$$

$$y(t) = (1 - t)^2 y_0 + 2(1 - t)t y_1 + t^2 y_2, \quad (14)$$

where (x_0, y_0) and (x_2, y_2) are the start and end points of the curve, respectively, and (x_1, y_1) is the control point. In this case, Eq. (9) can be analytically solved using Cardano's formula [38], which yields at most three real-number solutions; i.e., up to three 1D wavefronts $U(y)$ are superimposed at $\vec{r} = (\alpha, \beta)$ at different positions for y .

2.1.2. Straight line

For a straight line, $\vec{g}(t) = (x(t), y(t))$ is defined as

$$x(t) = x_s + t(x_e - x_s), \quad (15)$$

$$y(t) = y_s + t(y_e - y_s), \quad (16)$$

where $(x_s, y_s) = \vec{s}$ and $(x_e, y_e) = \vec{e}$ are the start and end points of the straight line, respectively. The norm of the normal vector from a point on the line to a certain point is invariably the minimum

distance between the point and the line. Thus, $|\vec{h}(T_j)|$ can be obtained as

$$|\vec{h}(T_j)| = \frac{|d_y\alpha - d_x\beta + x_yd_x - x_sd_y|}{\sqrt{d_x^2 + d_y^2}}, \quad (17)$$

where $(d_x, d_y) = (x_e - x_s, y_e - y_s) = \vec{d}$. The following formula obtains at the most one solution if that solution satisfies $0 \leq t \leq 1$:

$$\vec{d} \cdot (\vec{s} + t\vec{d} - \vec{r}) = 0. \quad (18)$$

2.1.3. Arc

For an arc, $g(t) = (x(t), y(t))$ is defined as

$$x(t) = R \cos(t\theta_c + \theta_0) + x_c, \quad (19)$$

$$y(t) = R \sin(t\theta_c + \theta_0) + y_c, \quad (20)$$

where R is the radius, θ_c is the central angle, θ_0 is the argument of the starting point, and $\vec{c} = (x_c, y_c)$ are the coordinates of the arc center. Because the vector $\vec{r} - \vec{c}$ is always the normal vector of the arc, $|\vec{h}(T_j)|$ can be obtained using the following equation:

$$|\vec{h}(T_j)| = |\vec{r} - \vec{c}| \pm R. \quad (21)$$

The following formula affords two solutions at the most if those solutions satisfy $0 \leq t \leq 1$:

$$t\theta_c + \theta_0 = \arg(\vec{r} - \vec{c}), \arg(\vec{c} - \vec{r}). \quad (22)$$

2.2. Difficulty in realizing amplitude gradation along a line

Because Eq. (4) is derived assuming a constant line intensity, this version of the CG-line method cannot, in principle, project a 3D line-drawn object with amplitude gradation. Therefore, in this study, we approximately realize this gradation by modulating the 1D wavefront calculated using Eq. (4).

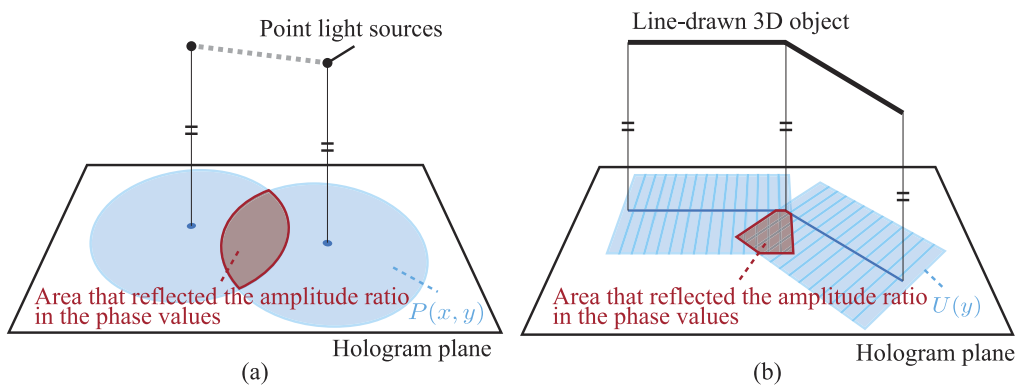


Fig. 3. Causes of difficulty in modulating the intensity along the line using the CG-line method. (a) Point-based method: sufficient overlap regions exist between the wavefronts created by each PLS so that the intensity ratio of the PLS is easily reflected in the phase information. (b) CG-line method: the wavefronts partially overlap only when the line curves; therefore, the intensity ratio on the line is not easily reflected in the phase information.

Figure 3 explains the difficulty of realizing gradation control using the CG-line method. As the kinoform only uses the phase information on the hologram plane, the amplitude ratio between the points on the line should be encoded in the phase information of the hologram; therefore, those wavefronts should overwrap each other on the hologram plane. Within this context, gradation can be controlled by directly applying Eq. (1) (i.e., using the point-based method) because the PLSs are closely spaced and all PLS wavefronts overlap, as depicted in Fig. 3(a). Expressing a variable intensity ratio between points on the line is more difficult with the CG-line method than the point-based method using the phase information of the kinoform CGH. As shown in Fig. 3(b), the CG-line method generates 1D wavefronts for each point of the line for superimposition and the wavefront-overlap areas are small or zero. The worst-case scenario is a straight line for which the wavefronts never overlap, meaning that any information of the amplitude variation is omitted. Therefore, the CG-line method must be modified to effectively encode the amplitude ratio in the phase information of the kinoform CGH.

3. Proposed method

3.1. Intensity control of the reconstructed image via imposing phase error

The proposed method introduces a controlled random phase error to the 1D wavefront of the CG-line method. The control is implemented by varying the degree of the error. Figure 4 provides the overview of the phase-error imposition of the proposed method and examples of the phase error-imposed zoneplates. Imposing a random phase error on the hologram is equivalent to setting a diffuser plate in front of the hologram. Therefore, it is expected to attenuate the intensity of the reconstructed image. Furthermore, as the error degree corresponds to the grid of the diffuser plate, the proposed method is expected to arbitrarily control the intensity on the projected line.

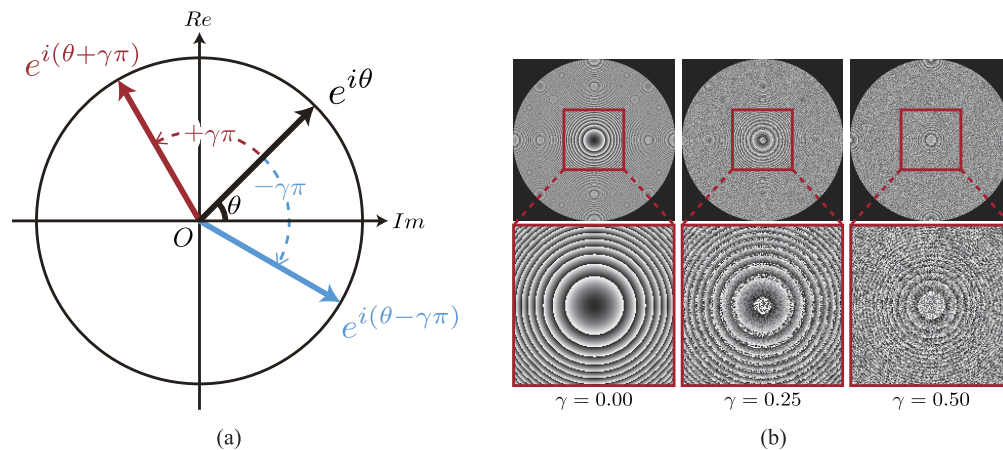


Fig. 4. Phase-error imposition in the proposed method: (a) diagram of phase-error imposition in the complex plane, where we set $\frac{a}{\lambda} = 1$ for simplicity, and (b) examples of phase error-imposed zoneplates.

To verify the abovementioned assumption, we investigated the relation between the error degree and intensity on the focal point of the zoneplate, which is a hologram created from one PLS using Eq. (1). Because this zoneplate is the most primitive hologram to focus the incident light to a point light like a lens; its response to the imposed phase error is valuable for our present purpose.

Applying Eq. (1), the error imposed on the zoneplate is calculated as

$$P(x, y) = \frac{a}{\zeta} \cdot \exp \{i(\theta \pm \gamma\pi)\}, \quad (23)$$

where $\theta = \frac{\pi}{\lambda z} [(x - \delta)^2 + (y - \epsilon)^2]$ and γ is the error rate. The sign of γ is randomly changed with equal probability. In the proposed method, the superposition of random phase errors is assumed to attenuate the intensity (like a diffuser plate) and the error degree is correlated with the degree of intensity attenuation.

The discrepancy $\gamma\pi$ of a binary random phase source will be directly related to the degree of intensity modulation. Locally, the signal will interfere with equal proportions of $e^{-i\pi\gamma}$ and $e^{+i\pi\gamma}$, causing an average relative intensity modulation of

$$I = \left| \frac{e^{-i\pi\gamma} + e^{+i\pi\gamma}}{2} \right|^2 = \cos(\pi\gamma)^2 = \frac{\cos(2\pi\gamma) + 1}{2}. \quad (24)$$

Inverting this expression, we can select the correct γ for a chosen intensity modulation I :

$$\gamma(I) = \frac{\arccos(2I - 1)}{2\pi}. \quad (25)$$

By varying γ , we impose a random phase error that corresponds to the desired intensity on the line to the wavefront.

Figure 5 shows the experimental results of the intensity attenuations of the error-imposed zoneplate with varying γ for three focal distances: 0.1, 0.2, and 0.3 m. These results were analyzed using a hand-crafted diffraction simulator based on the ASM. Here, the measured intensities were normalized to the same depth because the line-drawn objects used in the proposed method lie at the same depth. Therefore, the relationship between the imposed error and reconstructed intensity with the same depth is more relevant to the proposed method. In the experiment, we set $\lambda = 532\text{nm}$ and $p = 3.74\mu\text{m}$. The computer configuration was as follows: Microsoft Windows 11 Professional operating system, Intel Core i9-12900KF 3.20 GHz, 64-GB DDR5-38400 memory, and Microsoft Visual C++ 2019 compiler with single floating-point computational precision. The reconstructed intensities presented in Fig. 5 obeyed Eq. (24).

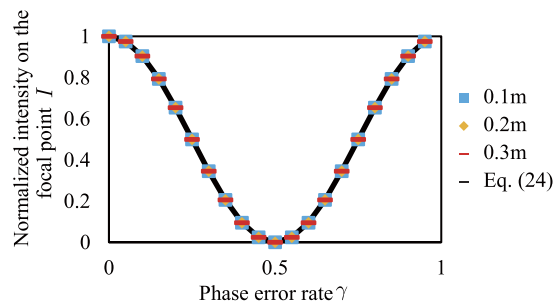


Fig. 5. Relation between the normalized intensity at the focal point and γ .

3.2. CG-line method with gradation control

As discussed in the Sec. 3.1, the reconstructed intensity of a hologram can be controlled by superimposing phase errors on the complex wavefront of a zoneplate. To apply these results to

the CG-line method, we modify Eq. (4) to include the phase-error imposition as follows:

$$U(y) = a\sqrt{\frac{\lambda}{\zeta}} \exp \left[i\pi \left\{ \frac{y^2}{\lambda\zeta} \pm \gamma(I) \right\} \right]. \quad (26)$$

To calculate the CGHs of the line-drawn objects, we substitute Eq. (26) into Eq. (10). To effectively transfer the gradation information on the line to the GPU, we represent the gradation using a Catmull–Rom spline curve [39], as demonstrated in Fig. 6. The Catmull–Rom spline curve is an interpolation function designed to pass through every control point. We include the feature points of the gradation curve (e.g., the extremum points of the curve) in a 3D model file and generate the gradation curve at each calculation process of the CGHs. Given that $I_k(t)$ is the intensity function of the k -th line segment of the 3D object that is formulated via the Catmull–Rom spline curve, $L_k(\alpha, \beta)$ in Eq. (12) can be rewritten in terms of Eq. (26) as follows:

$$L_k(\alpha, \beta) = \sum_j a\sqrt{\frac{\lambda}{\zeta}} \exp \left\{ i\pi \left(\frac{|\vec{h}(T_j)|^2}{\lambda\zeta} \pm \gamma \{I_k(T_j)\} \right) \right\}. \quad (27)$$

Finally, the CGHs of a line-drawn object with gradation expression are obtained using Eq. (12).

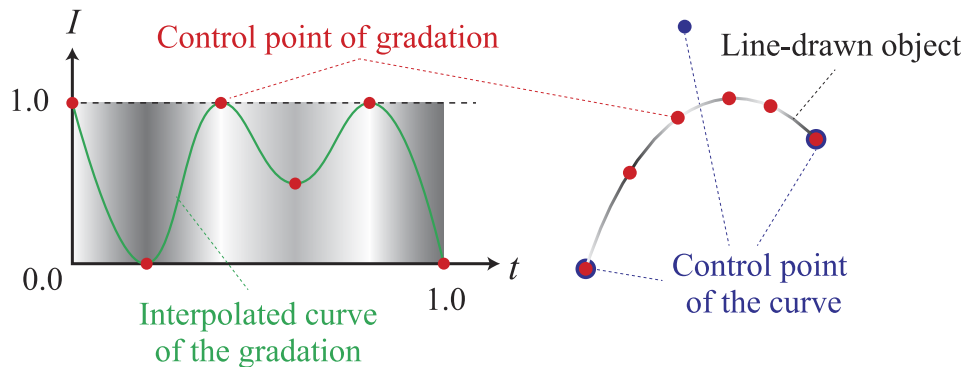


Fig. 6. Expression of gradation on the line.

Compared with the conventional approach of using a 2D mask image that describes the intensity distribution over the line, the proposed approach reduces the number of required memory accesses for the GPU. Because memory accesses in a GPU are often a bottleneck of the calculation speed, reducing their number and duration is essential for efficient implementation of the GPU.

4. Experimental results and discussion

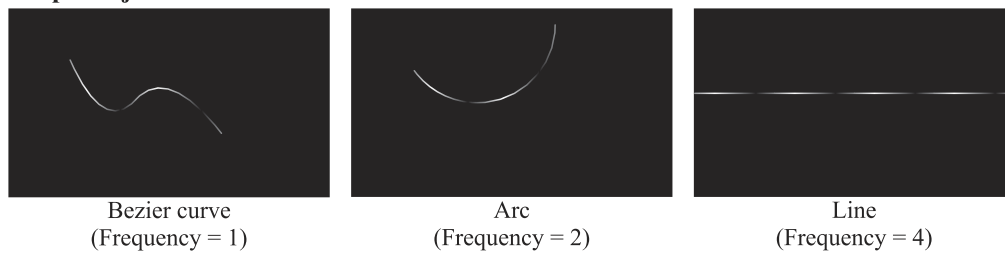
To verify the feasibility of the proposed method, we investigated the image quality and computational speed. The proposed method was implemented in a GPU, and its results were compared with those of three reference methods that were expected to control the reconstructed intensity on the line.

- **The conventional point-based method with amplitude modulation** directly applies Eq. (2) for calculating the wavefront of every PLS on the line obtained by quantizing the line objects into points with a density corresponding to the pixel pitch of SLM. The intensity on the line is encoded as $a(u)$ in Eq. (2).
- **The ASM** calculates the wave diffraction by applying the fast Fourier transform to the specific depth plane on which the PLS exists.

- **The CG-line method with amplitude modulation** superimposes $U(y)$ from Eq. (4) with an amplitude weight that corresponds to the desired intensity on the line.

We used three types of simple objects (a quadratic Bezier curve with two segments, a single straight line, and a single arc) and two types of complex objects (the “SimpleShape” model with twelve simple objects at different depths and the “Tokyo” model with five alphabetic characters). The models are overviewed in Fig. 7. For the simple objects and Tokyo model, we set the reconstructed depth to 0.1 and 0.3 m, respectively. Additionally, we sinusoidally modulated the intensity on a segment of the simple-object and SimpleShape models at different frequencies (1, 2, 4, and 8 wave/segment). Some examples of sinusoidal intensity modulation are depicted in Fig. 7. For the Tokyo model, we applied different relative intensity ratios to the characters, which is also depicted in Fig. 7.

Simple objects



Complex objects

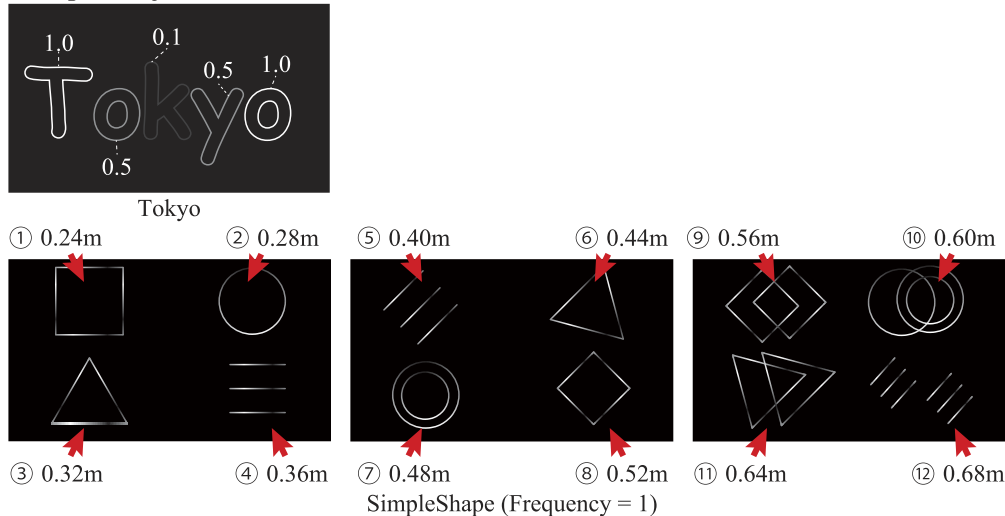


Fig. 7. Models. Numbers in the figure of the Tokyo model mean the relative intensity of each character, and that in the SimpleShape model means the depth of replaying each shape.

The CGH calculations and reconstruction simulations were both performed using the following computer configuration: Microsoft Windows 11 Professional operating system, Intel Core i9-12900KF 3.20 GHz, 64-GB DDR5-38400 memory, Microsoft Visual C++ 2019 compiler with single floating-point computational precision, and a NVIDIA Geforce RTX 3090 GPU with CUDA 11.6. The CGH resolution was set to $4,096 \times 2,400$ and $8,192 \times 4,800$ pixels, and the wavelength of the optical light source was assumed as 532 nm.

Figure 8 provides an overview of the optical system. We used an optical system comprising a phase-modulation-type SLM (JD7714, Jasper Display Corp., California, USA) with a resolution

of $4,096 \times 2,400$ pixels and a pixel pitch of $3.74 \mu\text{m}$, a green laser with a wavelength of 532 nm (CPS532, Thorlabs Inc., New Jersey, USA), a half-wave plate (HWP) (WPH10M-532, Thorlabs Inc.), a polarizer (WP25M-VIS, Thorlabs Inc.), a beam expander (GBE10-A, Thorlabs Inc.), a polarized beam splitter (#49-002, Edmund Optics, New Jersey, USA), a quarter-wave plate (QWP) (#48-489, Edmund Optics), a spherical achromatic lens (DLB-50-100PM, Sigma-koki, Saitama, Japan), a beam splitter (47-571, Edmund Optics), and a field lens (SLB-100B-300PM, Sigma-koki).

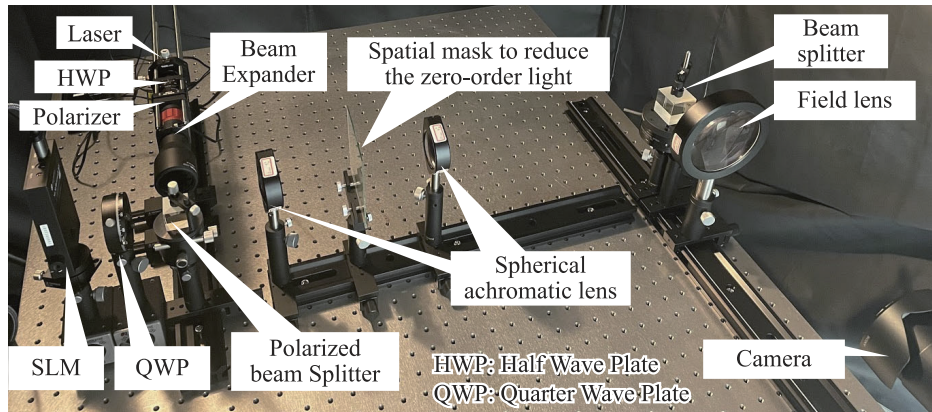


Fig. 8. Optical system.

4.1. Image quality

Here, we examine the image qualities of the numerically and optically reconstructed images of CGHs created using the proposed method and the comparative methods.

We evaluated the image quality using two evaluation indices: root mean square error (RMSE) on the line and the structural similarity index measure (SSIM). Both the indices were calculated by comparing the numerically reconstructed images of CGHs obtained using ASM with the desired reference images created via computer graphics. The RMSE can evaluate the precision of the proposed method and comparative methods for generating the intensity modulation on the line. The SSIM can evaluate the overall image quality of the reconstructed plane and indicate the noise intensity outside the line object. Because the random phase error imposed in the proposed method behaves like a diffuser plate in front of the CGH, the proposed method can induce unwanted noise. We therefore evaluated the precision of the intensity modulation using the RMSE and the unwanted noise via the SSIM.

Figure 9 displays the RMSE and SSIM results of each model. The RMSEs of the proposed method were smaller than that of the other reference methods for the simple models at each resolution, indicating the superior precision of intensity modulation by the proposed method for simple models. Furthermore, the RMSE tended to be lower at the higher resolution CGHs in the proposed method. For the simple models, the SSIMs were almost 1.0 and did not considerably differ among different methods, suggesting a small effect of unwanted noise on the proposed method.

Figure 10 shows an example of the numerically and optically reconstructed images using the proposed and comparative methods. The results of the Bezier and arc models were obtained from the CGH with $4,096 \times 2,400$ pixel resolution; thus, we show both the numerically and optically reconstructed images using the JD7714 SLM. For the straight-line model obtained from the CGH with $8,192 \times 4,800$ pixel resolution, we show only the numerically reconstructed image because the resolution between the SLM we used and the CGH is not match. Furthermore, we used a

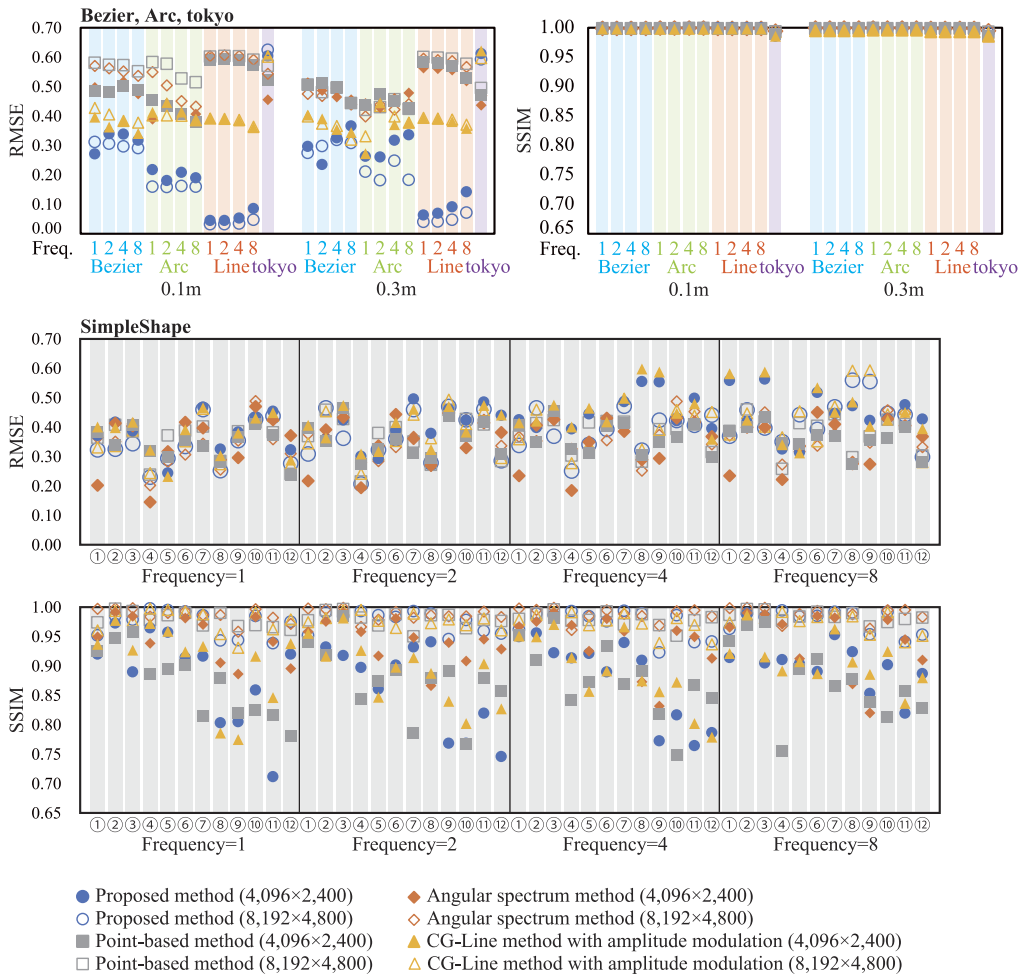


Fig. 9. RMSE and SSIM results of the compared methods.

histogram-based thresholding filter to enhance the visibility of the numerically reconstructed image; however, the filter did not considerably affect the overall gradation on the line. Notably, the RMSE and SSIM values were calculated from the reconstructed image without the filter. The figure shows that the proposed method can reconstruct the line with gradation effect that is similar or even superior to the ASM and the point-based method for the simple models. Moreover, it appears that the CG-line method with the amplitude modification cannot modulate the intensity on the line effectively.

Figure 11 provides an example of the intensity distribution on the line of the simple models that are presented in Fig. 10. In this figure, the Bezier curve model includes two segments: (1) $0 \leq t \leq 1$ and (2) $1 \leq t \leq 2$. The intensity distribution of the proposed method reproduces the sinusoidal intensity distribution equivalently to or more accurately than the comparative methods. Similarly accurate intensity distributions were obtained for the other cases of simple models.

Meanwhile, the modulation was more accurate on the straight line model than on the Bezier and arc curves, probably because the intensity along the line was imbalanced in the CG-line method, as already reported in [33]. The CG-line method computes the CGHs by synthesizing 1D wavefronts and duplicate them perpendicularly to the drawn line object. The synthesized

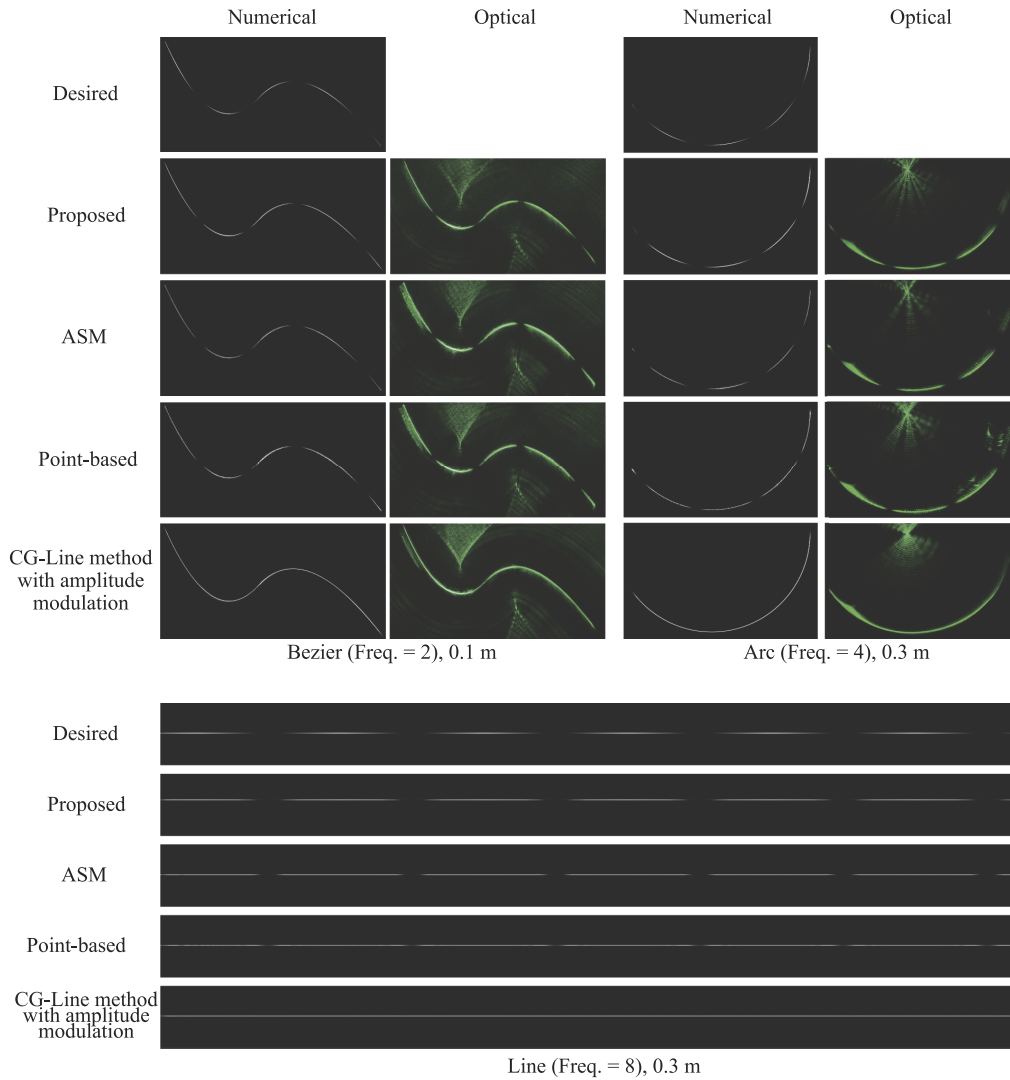


Fig. 10. Numerically and optically reconstructed images of the simple models.

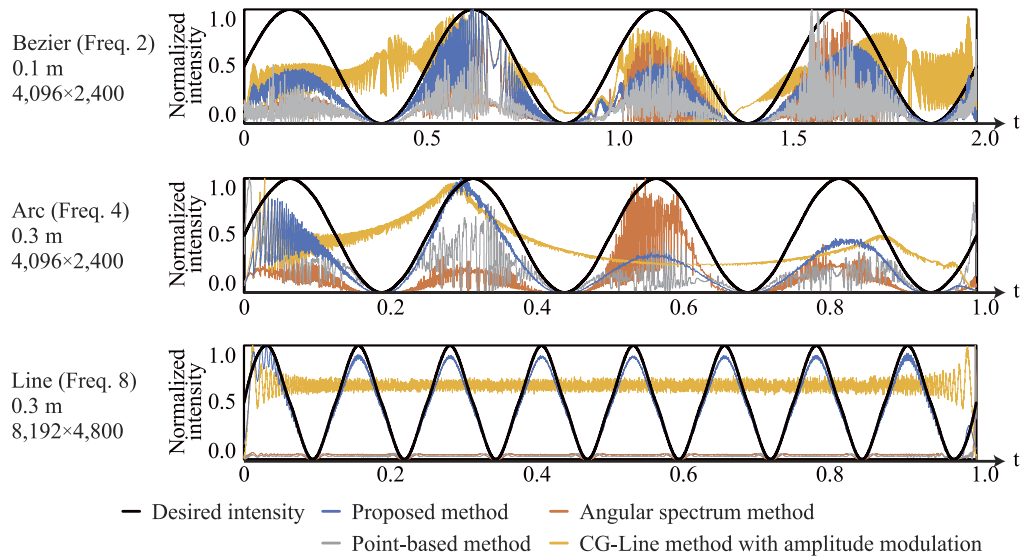


Fig. 11. Examples of intensity distributions on the line in the reconstructed plane.

area of the wavefronts depends on the radius of curvature of the line-drawn object, its playback distance and the pixel pitch of the SLM. Moreover, we found that the synthesized area of the wavefront was correlated with the intensity ratio at the playback plane. Therefore, in the proposed method, the intensity distribution along the line of the models with a curved part was affected by intensity imbalance. To enhance the accuracy of the modulation such as that of the straight-line case, we should modify Eq. (25) to consider the intensity imbalance. We are now investigating the theory regarding source of intensity imbalance. To overcome this problem, we will adjust the rate of the phase error imposed along a continuous line. The results will be reported in the next study.

The RMSE and SSIM results of the complex models (Fig. 9) reconstructed using the proposed method were sometimes inferior to those of the reference methods, probably because the noise was contributed by other layers. Figure 12 shows an example of the numerically reconstructed images of the SimpleShape model with frequency 1 wave/segment at both two reconstructed distance (0.24 and 0.52 m) obtained from the CGH with $8,192 \times 4,800$ pixel resolution. As the figure shows that high unwanted noises appeared outside the line-drawn object in both the proposed method and the CG-line method with amplitude modulation which are assumed to the contribution from the other layers' object because of their position and shapes. These noises are considered to explain the poorer evaluation indices of these methods than those of the comparative methods. However, the object reproduced using the proposed method showed sufficient visibility among the comparative methods.

Figure 13 shows a numerically and optically reconstructed image of the Tokyo model obtained using the CGH with a $4,096 \times 2,400$ pixel resolution. As shown in the figure, the proposed method can modulate the relative intensity between the characters in a manner similar to the comparative methods. It seems that the intensity unevenness on the line generated by the proposed method worsens the SSIM and RMSE. Compared to both numerical and optical reconstructed images in Fig. 13, the intensity is concentrated at the edges of characters, especially in "T," "k," and "y" in the proposed method. This is a known issue of the CG-line method that has been discussed in [33]. Since the intensity unevenness worsens the SSIM and RMSE and visibility of

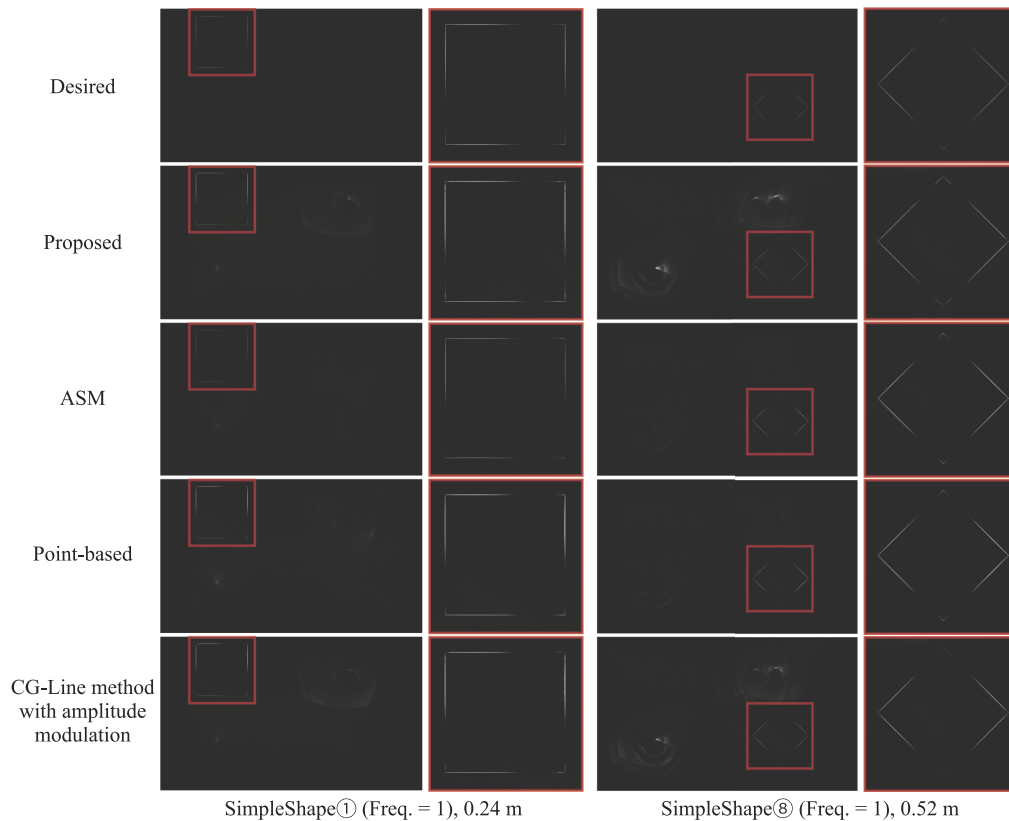


Fig. 12. Numerically and optically reconstructed images of simple models.

the reconstructed image, we are now studying the method to cancel out the intensity unevenness by adjusting the phase error, which will be reported shortly.

The CG-line method with amplitude modulation also modulated the intensity on the lines of the complex model, for which the SSIM and RMSE almost equaled those of the proposed method. This is because unlike the simple models, the 1D wavefronts sufficiently overlap with each other when processing the complex models.

4.2. Computational speed

Figure 14 provides a comparison of the computational speeds of the models. The proposed method was 3.7–7.8 times faster than the ASM method for all models except the Tokyo model, and 25–237 times faster than the point-based method. These results confirm the superior computational speed of the proposed method. The slower computational speed of the proposed method than the ASM method for the Tokyo model might be explained by the high complexity of the Tokyo model, which comprises a quadratic Bezier curve with 111 segments and two straight lines. This result implies that the calculation method should be switched between the ASM and proposed methods depending on the complexity of the 3D model.

The ratio of computational times of the proposed and CG-line methods with amplitude modification was 0.57–0.97. This is probably due to the generation of random numbers. Thus, the additional computational cost against the original CG-line method is insignificant.

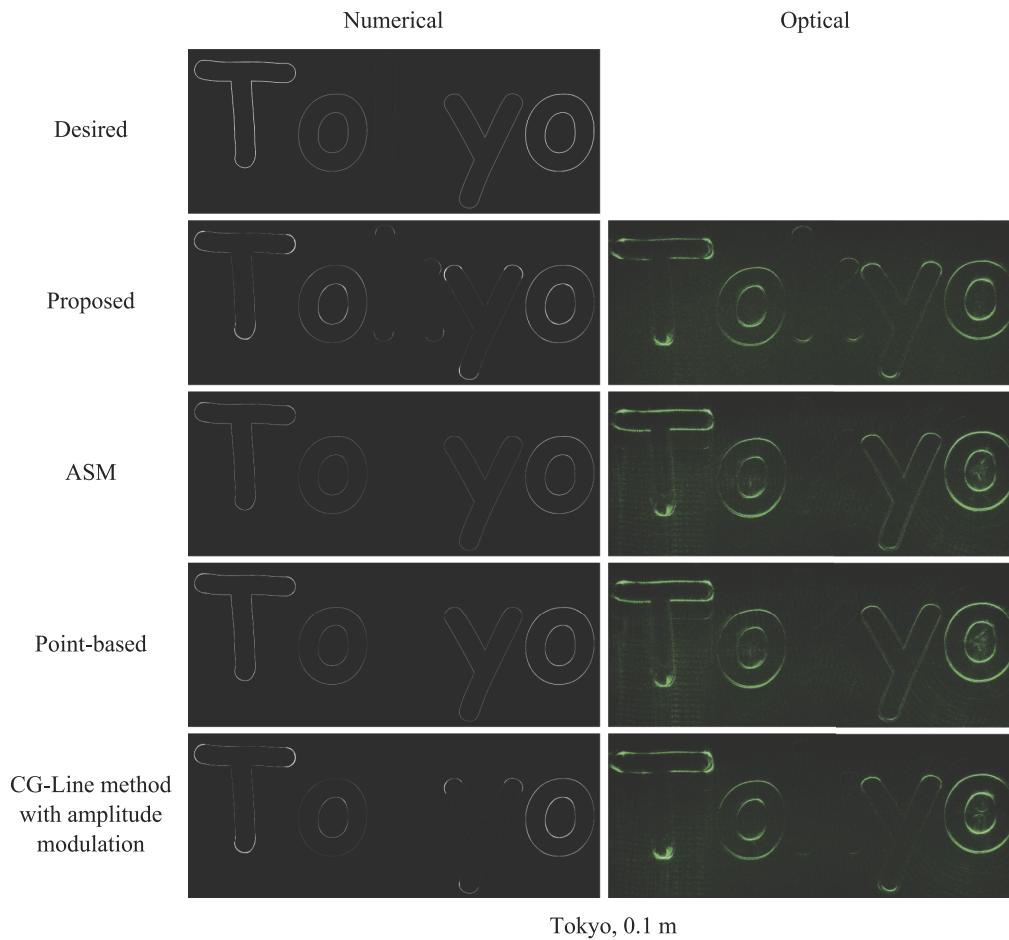


Fig. 13. Numerically and optically reconstructed images of simple models.

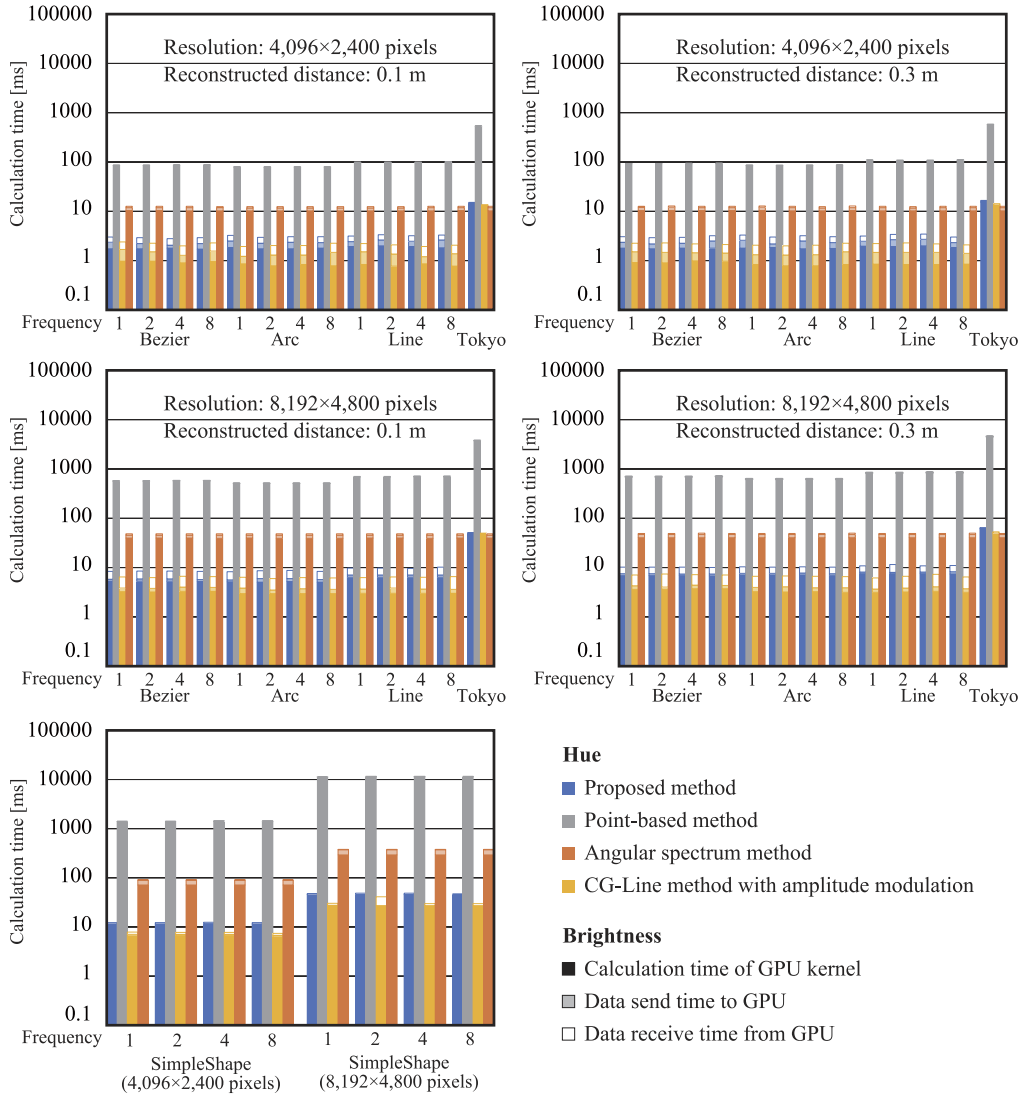


Fig. 14. Comparison results of computational time.

5. Conclusion

A novel method for modulating the intensity gradation of lines in kinoform-type CGHs of 3D line-drawn objects is proposed in this study. We controlled the intensity distributions along the lines by adding an adaptive phase error on the hologram without substantially degrading the image quality or reducing high computational speed of the original algorithm. The variable line distributions enhanced the expressiveness of 3D images using the CG-line method. In comparison with several reference methods, the proposed method was most effective when the 3D model comprised simple shapes. Images of more complex models could be reconstructed with arbitrarily modulated intensities using all calculation methods, although the relative image qualities and computational speeds differ among the methods. The proposed method achieved more stable modulation performance and lower computational speed than the comparison methods.

The proposed method still has some drawbacks that will be addressed in future work. Specifically, we will enhance the image quality by reducing the noise and adjust the phase error to remove the intensity imbalance on the line. Further, we will reduce the computational time of processing complex objects. The computational performance of the proposed method depends on the complexity of the 3D image. This is unavoidable given the computational scheme of the proposed method. However, it differs from other methods (e.g., the layer-based methods) in terms of the characteristics that cause fluctuations in computational performance. For example, the computational performance of the layer-based methods depends on the CGH resolution, while the proposed method is almost independent of the CGH resolution. Therefore, we believe that there should be a calculation method appropriate for different calculation conditions (CGH resolution, 3D image complexity) and that the method should be able to switch between different methods depending on the conditions. We are currently studying this method and expect to report on it shortly. Notably, the computational speed and expressiveness of the proposed method are already sufficient for potential applications such as heads-up displays.

Funding. Fonds Wetenschappelijk Onderzoek (12ZQ220N, VS07820N); Tokyo Metropolitan University (TMU local 5G research support); Takayanagi Kenjiro Foundation; Japan Society for the Promotion of Science (19H01097, 22H03616).

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. E. Murakami, Y. Oguro, and Y. Sakamoto, "Study on compact Head-Mounted display system using Electro-Holography for augmented reality," *IEICE Trans. Electron.* **E100.C**(11), 965–971 (2017).
2. Y. Liu, H. Dong, L. Zhang, and A. E. Saddik, "Technical evaluation of HoloLens for multimedia: A first look," *IEEE Multimedia* **25**(4), 8–18 (2018).
3. Y. Ochiai, K. Kumagai, T. Hoshi, J. Rekimoto, S. Hasegawa, and Y. Hayasaki, "Fairy lights in femtoseconds: Aerial and volumetric graphics rendered by focused femtosecond laser combined with computational holographic fields," *ACM Trans. Graph.* **35**(2), 1–14 (2016).
4. M. Parker, "Lumarca," in *ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation*, (Association for Computing Machinery, New York, NY, USA, 2009), SIGGRAPH ASIA '09, p. 77.
5. R. Hirayama, D. Martinez Plasencia, N. Masuda, and S. Subramanian, "A volumetric display for visual, tactile and audio presentation using acoustic trapping," *Nature* **575**(7782), 320–323 (2019).
6. 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. Inf.* **13**(5), 2447–2454 (2017).
7. E. Sahin, E. Stoykova, J. Mäkinen, and A. Gotchev, "Computer-Generated holograms for 3D imaging: A survey," *ACM Comput. Surv.* **53**(2), 1–35 (2021).
8. 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. Express* **26**(20), 26722–26733 (2018).
9. 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," *Nat. Commun.* **11**(1), 5568 (2020).
10. H. Kim, Y. Kim, H. Ji, H. Park, J. An, H. Song, Y. T. Kim, H. Lee, and K. Kim, "A Single-Chip FPGA holographic video processor," *IEEE Trans. Ind. Electron.* **66**(3), 2066–2073 (2019).

11. 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. Express* **20**(19), 21645–21655 (2012).
12. 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. Express* **17**(21), 18543–18555 (2009).
13. Y.-H. Seo, Y.-H. Lee, and D.-W. Kim, "ASIC chipset design to generate block-based complex holographic video," *Appl. Opt.* **56**(9), D52–D59 (2017).
14. D. Blinder, T. Birnbaum, T. Ito, and T. Shimobaba, "The state-of-the-art in computer generated holography for 3D display," *Light. Adv. Manuf.* **3**, 1 (2022).
15. M. E. Lucente, "Interactive computation of holograms using a look-up table," *J. Electron. Imaging* **2**(1), 28–34 (1993).
16. J.-S. Chen and D. P. Chu, "Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications," *Opt. Express* **23**(14), 18143–18155 (2015).
17. Y. Zhao, L. Cao, H. Zhang, D. Kong, and G. Jin, "Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method," *Opt. Express* **23**(20), 25440–25449 (2015).
18. H. G. Kim and Y. Man Ro, "Ultrafast layer based computer-generated hologram calculation with sparse template holographic fringe pattern for 3-D object," *Opt. Express* **25**(24), 30418–30427 (2017).
19. 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).
20. T. Nishitsuji, T. Shimobaba, T. Kakue, N. Masuda, and T. Ito, "Fast calculation of computer-generated hologram using the circular symmetry of zone plates," *Opt. Express* **20**(25), 27496–27502 (2012).
21. T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Fast calculation of computer-generated hologram using run-length encoding based recurrence relation," *Opt. Express* **23**(8), 9852–9857 (2015).
22. S.-C. Kim and E.-S. Kim, "Fast computation of hologram patterns of a 3D object using run-length encoding and novel look-up table methods," *Appl. Opt.* **48**(6), 1030–1041 (2009).
23. D. Pi, J. Liu, Y. Han, A. U. R. Khalid, and S. Yu, "Simple and effective calculation method for computer-generated hologram based on non-uniform sampling using look-up-table," *Opt. Express* **27**(26), 37337–37348 (2019).
24. D. Pi, J. Liu, R. Kang, Z. Zhang, and Y. Han, "Reducing the memory usage of computer-generated hologram calculation using accurate high-compressed look-up-table method in color 3D holographic display," *Opt. Express* **27**(20), 28410–28422 (2019).
25. F. Wang, T. Shimobaba, Y. Zhang, T. Kakue, and T. Ito, "Acceleration of polygon-based computer-generated holograms using look-up tables and reduction of the table size via principal component analysis," *Opt. Express* **29**(22), 35442–35455 (2021).
26. T. Shimobaba, D. Blinder, T. Birnbaum, I. Hoshi, H. Shiomi, P. Schelkens, and T. Ito, "Deep-learning computational holography: A review (invited)," *Front. Photonics* **3**, 1 (2022).
27. R. Horisaki, Y. Nishizaki, K. Kitaguchi, M. Saito, and J. Tanida, "Three-dimensional deeply generated holography [invited]," *Appl. Opt.* **60**(4), A323–A328 (2021).
28. T. Shimobaba, D. Blinder, M. Makowski, P. Schelkens, Y. Yamamoto, I. Hoshi, T. Nishitsuji, Y. Endo, T. Kakue, and T. Ito, "Dynamic-range compression scheme for digital hologram using a deep neural network," *Opt. Lett.* **44**(12), 3038–3041 (2019).
29. D.-Y. Park and J.-H. Park, "Hologram conversion for speckle free reconstruction using light field extraction and deep learning," *Opt. Express* **28**(4), 5393–5409 (2020).
30. Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," *ACM Trans. Graph.* **39**(6), 1–14 (2020).
31. L. Shi, B. Li, C. Kim, P. Kellnhofer, and W. Matusik, "Towards real-time photorealistic 3d holography with deep neural networks," *Nature* **591**(7849), 234–239 (2021).
32. T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Fast calculation of computer-generated hologram of line-drawn objects without FFT," *Opt. Express* **28**(11), 15907–15924 (2020).
33. T. Nishitsuji, D. Blinder, T. Kakue, T. Shimobaba, P. Schelkens, and T. Ito, "GPU-accelerated calculation of computer-generated holograms for line-drawn objects," *Opt. Express* **29**(9), 12849–12866 (2021).
34. T. Nishitsuji, T. Kakue, D. Blinder, T. Shimobaba, and T. Ito, "An interactive holographic projection system that uses a hand-drawn interface with a consumer CPU," *Sci. Rep.* **11**(1), 147 (2021).
35. D. Blinder, T. Nishitsuji, T. Kakue, T. Shimobaba, T. Ito, and P. Schelkens, "Analytic computation of line-drawn objects in computer generated holography," *Opt. Express* **28**(21), 31226–31240 (2020).
36. D. Blinder, T. Nishitsuji, and P. Schelkens, "Real-time computation of 3D wireframes in computer-generated holography," *IEEE Trans. Image Process.* **30**, 9418–9428 (2021).
37. L. B. Lesem, P. M. Hirsch, and J. A. Jordan, "The kinoform: A new wavefront reconstruction device," *IBM J. Res. Dev.* **13**(2), 150–155 (1969).
38. W. H. Beyer, *CRC handbook of mathematical sciences* 5th edition (CRC, 1978).
39. E. Catmull and R. Rom, "A Class of Local Interpolating Splines," in *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, eds. (Academic Press, 1974), pp. 317–326.