IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 27, 2020, accepted July 31, 2020, date of publication August 7, 2020, date of current version August 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014979

Lossless Compression Using the Ramanujan Sums: Application to Hologram Compression

HARUTAKA SHIOMI, TOMOYOSHI SHIMOBABA, TAKASHI KAKUE, (Member, IEEE), AND TOMOYOSHI ITO
Graduate School of Engineering, Chiba University, Chiba 263-8522, Japan

Corresponding author: Harutaka Shiomi (h-shiomi@chiba-u.jp)

This work was supported in part by JSPS KAKENHI under Grant 19H04132 and Grant 19H01097.

ABSTRACT The Ramanujan sum, introduced by S. Ramanujan, has been utilized—among other applications—for signal processing. It has recently been suggested that transforms using the Ramanujan sums may also provide the benefit of data compression. This study presents a lossless hologram-compression method that employs transforms using the Ramanujan sums. In general, lossless compression of holograms is difficult, because the statistical properties of holograms are different from natural images. We compared the compression ratios of different hologram datasets, both with and without using Ramanujan-sums-based transforms. We found that the Ramanujan periodic transform improves the compression ratio of hologram data when using data having prime number dimensions.

INDEX TERMS Data compression, hologram, linear transform, Ramanujan sum.

I. INTRODUCTION

The Ramanujan sum was introduced in 1918 by Srinivasa Ramanujan to represent arithmetic functions [1], and it has been applied to de-noising [2], time-frequency analysis [3], the acceleration of discrete Fourier transforms [4], and multi-resolution analysis [5]. Linear transforms that use the Ramanujan sums have been studied as well [6], because the Ramanujan sum offers orthogonal and functional energy-conservation properties, like the Fourier transform. Reference [6] has suggested that linear transforms using the Ramanujan sums may be used for data compression; however, to our knowledge Ramanujan-sums-based transforms so far have not been shown to provide effective data compression using real data. This Letter is the first report of data compression using the Ramanujan sums. We found that transforms using the Ramanujan sums work well for lossless compression of holographic data, especially when the horizontal and vertical sizes of the data are prime numbers.

Holography is a well-known optical technique for recording and reproducing light waves reflected from a three-dimensional (3D) object to/from a two-dimensional medium, which is referred to as a hologram. By simulating the interference and diffraction of the light waves, we can generate holograms—or reproduce 3D images from optically recorded holograms—on a computer instead of using optical setups [7]. This approach has been studied for 3D displays [8], [9], and for 3D cameras and microscopy [10] because it can reproduce the original light field. The data capacity of a hologram tends to be large, because holography requires large spatial-bandwidth products. Holograms also have large dynamic ranges and contain random patterns, because holography utilizes the interference of light waves. It is therefore necessary to develop data compression techniques that match the characteristics of holograms.

Many methods have been proposed for both lossy and lossless compression of holograms [11]. In order to reduce the large dynamic range of a hologram, these techniques quantize the dynamic range into an 8-bit to 10-bit range in advance, and the quantized holograms are then compressed. Lossless compression techniques for quantized holograms have been proposed in [12], and [13]; however, Ref. [14] claims that the lossless compression of holograms is difficult. Although these studies have examined the compression of quantized holograms, in detail, the lossless compression of holograms in floating-point format has not been considered much. Due to the large dynamic ranges of holograms, lossless compression of floating-point holograms is important for hologram archiving applications.

In this Letter, we demonstrate that our proposed method, based on the Ramanujan sums, can improve the lossless compression ratio of floating-point holograms in comparison

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

with well-known compression techniques. This Letter is structured as follows. Section II presents three linear transforms using the Ramanujan sums. Section III shows the results of hologram compression, using holograms calculated for different conditions, both with and without using the three transforms based on the Ramanujan sums and with the discrete cosine transform (DCT). Finally, Section IV concludes the Letter and suggests future work.

II. PROPOSED METHOD

The Ramanujan sum is the sum of the nth powers of the qth primitive roots of unity, defined as

$$c_q(n) = \sum_{p=1;\gcd(p,q)=1}^{q} \exp\left(2\pi i \frac{p}{q} n\right), \quad (1)$$

where $\gcd(p, q) = 1$ means that the greatest common divisor is unity, and i is the imaginary unit. Various linear transforms using the Ramanujan sums have been proposed. In this Letter, to compress floating-point holograms, we use three transforms based on the Ramanujan sums. We summarize them below.

A. RAMANUJAN FIR TRANSFORM (RFT)

The RFT was introduced by Vaidyanathan - [15] -to represent finite-duration sequences with the Ramanujan sums. A one-dimensional (1D) input signal x (of length N) is represented by a_q as

$$x(n) = \sum_{q=1}^{N} a_q c_q(n). \quad (2)$$

In matrix and vector form, this can be written as

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} = F_N \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}, \quad (3)$$

where the qth column of F_N has the elements $c_q(n)$ repeated with period q until we get N rows: for example,

$$F_6 = \begin{bmatrix} 1 & 1 & 2 & 2 & 4 & 2 \\ 1 & -1 & -1 & 0 & -1 & 1 \\ 1 & 1 & -1 & -2 & -1 & -1 \\ 1 & -1 & 2 & 0 & -1 & -2 \\ 1 & 1 & -1 & 2 & -1 & -1 \\ 1 & -1 & -1 & 0 & 4 & 1 \end{bmatrix}. \quad (4)$$

An $N \times N$ matrix F_N always has full rank [15]. The two-dimensional (2D) RFT y of an input 2D signal x with dimensions $N \times M$ is defined as

$$y = F_N^{-1} x (F_M^{-1})^T, \quad (5)$$

where F^{-1} means the inverse matrix of F, and F^T means the transposed matrix of F.

B. RAMANUJAN PERIODIC TRANSFORM (RPT)

The RPT was also introduced by Vaidyanathan [15]. A 1D input signal x (of length N) is represented by a linear combination of c_{q_i} and its shifted version, where q_i is a divisor of the input signal of length N.

In matrix and vector form, the RPT can be written as

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} = P_N \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(N-1) \end{bmatrix}, \quad (6)$$

where $d(\cdot)$ is the transformed data of x. The matrix P_N is

$$\begin{bmatrix} c_{q_1}^{(0)}(0) & \cdots & c_{q_1}^{(\phi(q_1)-1)}(0) & \cdots & c_{q_K}^{(\phi(q_K)-1)}(0) \\ c_{q_1}^{(0)}(1) & \cdots & c_{q_1}^{(\phi(q_1)-1)}(1) & \ddots & c_{q_K}^{(\phi(q_K)-1)}(1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{q_1}^{(0)}(N-1) & \vdots & c_{q_1}^{(\phi(q_1)-1)}(N-1) & \cdots & c_{q_K}^{(\phi(q_K)-1)}(N-1) \end{bmatrix},$$

where $c_{q_i}^{(l)}$ is l^{th} shifted version of c_{q_i}:

$$c_{q_i}^{(l)}(n) = c_{q_i}(n-l), \quad (7)$$

K means the number of divisors in N, and $\phi(n)$ is Euler's totient function. For example, P_6 is given by

$$P_6 = \begin{bmatrix} 1 & 1 & 2 & -1 & 2 & 1 \\ 1 & -1 & -1 & 2 & 1 & 2 \\ 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 2 & -1 & -2 & -1 \\ 1 & 1 & -1 & 2 & -1 & -2 \\ 1 & -1 & -1 & -1 & 1 & -1 \end{bmatrix}. \quad (8)$$

The $N \times N$ matrix P_N always has full rank [15]. A 2D RPT of an input 2D signal x represented by an $N \times M$ matrix into the output 2D signal $N \times M$ matrix y is defined as

$$y = P_N^{-1} x (P_M^{-1})^T. \quad (9)$$

C. RAMANUJAN SUMS TRANSFORM (RST)

The RST was introduced by Chen et al. [6]. The RST from a 1D input signal x (of length N) to the 1D output signal y is

$$y = A_N x. \quad (10)$$

The $N \times N$ matrix A_N is defined by

$$A_N(q, j) = \frac{1}{\phi(q)N} c_q(\mathrm{mod}(j-1, q) + 1), \quad (11)$$

where $q, j \in [1, M]$ and mod$(\ldots)$ means the modular operation. The $N \times N$ matrix A_N always has full rank [6]. The RFT from a 2D input signal $N \times M$ matrix x to the output 2D signal $N \times M$ matrix y is defined as

$$y = A_N x A_M^T. \quad (12)$$

D. COMPRESSION METHOD USING RAMANUJAN-SUMS-BASED TRANSFORMS

The abovementioned Ramanujan-sums-based transforms can decrease the standard deviation of the data to be compressed. Therefore, the data-compression ratio may be improved by applying these transforms to the data before compression. Our simple method is as follows:

1) We transform the data to be compressed using Ramanujan-sums-based transforms.
2) We compress the transformed data with commonly used compression programs (fpzip [16], zfp [17], bzip2, gzip, zip), where fpzip and zfp are dedicated compression algorithms for floating-point format. We expect to boost the compression ratio by combining of the Ramanujan-sums-based transforms and the dedicated floating-point compression algorithms.

III. RESULTS

To verify the effectiveness of our proposed method, we compare the compression ratio for four kinds of hologram data, both with and without using the transforms RFT, RPT, RST, or DCT. We used the DCT as a comparison method. The four kinds of hologram data are as follows:

1) Complex hologram generated from point-cloud data
2) Amplitude hologram generated from point-cloud data
3) Phase hologram generated from point-cloud data
4) Light waves propagated from a natural image with random phases

These hologram datasets are detailed in [18]. The compression ratio is defined by

$$\textit{Compression ratio} = \frac{\textit{Compressed data size}}{\textit{Original data size}} \times 100\%. \quad (13)$$

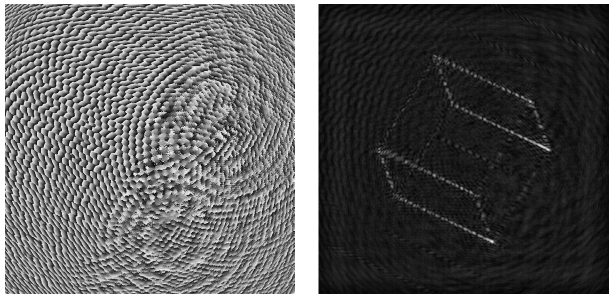
The transformation matrices are determined by the number of rows and columns of data. We used the data sizes 1024 × 1024 and 512 × 512. In addition, we used the closest prime numbers to these data sizes: 1021 × 1021 and 509 × 509. As discussed below, a data size that is a prime number boosts the compression capability of RPT-based compression.

A. COMPLEX HOLOGRAM

A complex hologram is numerical complex 2D data, which is calculated from light propagating from a 3D object to a hologram. Many algorithms have been proposed for complex hologram calculations [18]. In this study, we used the point-cloud method, in which a 3D object is represented by point-cloud data. A complex hologram is calculated as

$$u(x, y) = \sum_{j=1}^{N} \exp\left(\frac{2\pi i}{\lambda} r\right), \quad (14)$$

where N is the number of object points, λ is the wavelength of light, and $r = \sqrt{(x_j - x)^2 + (y_j - y)^2 + z_j^2}$ is the distance between the hologram pixel $(x, y, 0)$ and the jth object point (x_j, y_j, z_j). The calculation conditions are as follows: the wavelength is 520 nm, the pixel pitch of the holograms is 8 µm, the distance between the hologram and the object points is 1.0 m, and the resolution of the hologram is 1024 × 1024 or 1021 × 1021. Figures 1a and 1b show an example of the phase of a complex hologram and of the corresponding reconstructed image. We separate the complex hologram into its real and imaginary parts, and then we compress them individually. The compression results are shown in Tables 1 and 2; in these and subsequent tables, red text highlights the highest compression ratio in each row, and the asterisk (*) denotes the highest compression ratio in the table. Although in Table 1 the best compression ratio varies for each combination, Table 2 shows that the RPT is most effective when used with data having prime-sized dimensions. The highest compression ratio is obtained using the RPT combined with fpzip.



(a) Complex hologram (phase part). (b) Reconstructed image.

FIGURE 1. (a) Example of a complex hologram and (b) the reconstructed image.

TABLE 1. Compression results for a complex hologram with dimensions 1024 × 1024.

	RAW	RFT	RPT	RST	DCT
fpzip	85.673	91.343	82.551*	90.971	94.904
zfp	95.019	97.836	90.759	98.966	95.951
bzip2	95.538	95.695	95.213	95.829	95.028
gzip	93.218	93.374	92.910	93.547	92.906
zip	93.221	93.378	92.914	93.551	92.910

TABLE 2. Compression results for a 1021 × 1021 complex hologram.

	RAW	RFT	RPT	RST	DCT
fpzip	85.653	91.303	81.448*	90.977	94.911
zfp	95.270	97.962	89.841	99.274	96.267
bzip2	95.531	95.631	94.589	95.830	95.030
gzip	93.219	93.326	92.164	93.556	92.909
zip	93.223	93.330	92.168	93.559	92.912

B. AMPLITUDE HOLOGRAM

An amplitude hologram is calculated by taking the real part of a complex hologram. We used the same object and calculation conditions as in Fig. 1a. The compression results are shown in Tables 3 and 4. Although in Table 3 the best compression ratio again varies with the combination, Table 4 shows that the RPT is again most effective when used with prime-sized data.

TABLE 3. Compression results for a 1024 × 1024 amplitude hologram.

	RAW	RFT	RPT	RST	DCT
fpzip	85.650	91.366	82.565*	90.961	94.911
zfp	94.997	97.823	90.743	98.965	95.952
bzip2	95.532	95.692	95.214	95.827	95.026
gzip	93.209	93.369	92.912	93.546	92.905
zip	93.213	93.373	92.916	93.550	92.909

TABLE 4. Compression results for a 1021 × 1021 amplitude hologram.

	RAW	RFT	RPT	RST	DCT
fpzip	85.628	91.321	80.790*	90.965	94.897
zfp	95.258	97.954	89.159	99.261	96.273
bzip2	95.527	95.626	94.307	95.829	95.030
gzip	93.217	93.323	91.820	93.552	92.904
zip	93.221	93.327	91.824	93.556	92.908

The highest compression ratio is obtained for RPT combined with fpzip.

C. PHASE HOLOGRAM

A phase hologram is generated by calculating the argument of a complex amplitude. We used the same object and calculation conditions as in Fig. 1a. The compression results are shown in Tables 5 and 6. In this case, the compression ratios for the raw data in Table 5 are superior to all the linear transforms. In contrast, for the case of a prime-sized hologram (Table 6), we find that the RPT is superior to all the others, except for the raw data combined with bzip2. The highest compression ratio is obtained using RPT combined with fpzip.

TABLE 5. Compression results for a 1024 × 1024 phase hologram.

	RAW	RFT	RPT	RST	DCT
fpzip	84.165*	92.210	89.154	91.549	94.742
zfp	91.322	97.101	92.537	97.869	95.790
bzip2	93.039	95.544	94.953	95.555	94.847
gzip	92.335	93.230	92.775	93.173	92.599
zip	92.339	93.234	92.779	93.177	92.603

TABLE 6. Compression results for a 1021 × 1021 phase hologram.

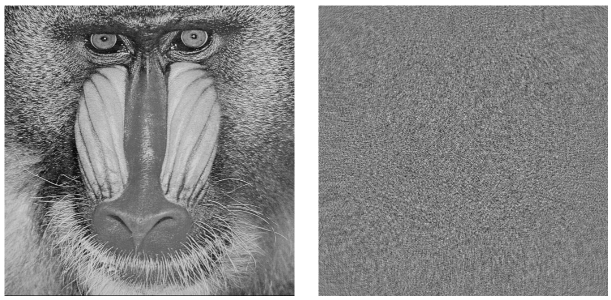
	RAW	RFT	RPT	RST	DCT
fpzip	84.155	92.193	79.213*	91.581	94.759
zfp	91.581	97.282	88.785	98.215	96.095
bzip2	93.022	95.487	93.492	95.564	94.868
gzip	92.337	93.209	90.868	93.183	92.598
zip	92.341	93.213	90.872	93.187	92.602

D. LIGHT WAVES PROPAGATED FROM A NATURAL IMAGE WITH RANDOM PHASES

In computer holography, we often use random phases to spread the object light over the entire hologram [18]. Before calculating the hologram, we multiply the object data with random phases. In this study, we used a 2D natural image, $g(x, y)$, as the object data, as shown in Fig. 2a. The light waves on the hologram plane are calculated using the angular spectrum method [18]:

$$u(x, y) = FT^{-1}[FT[g(x, y)\exp(in(x, y))] \times H(u, v)], \quad (15)$$

where FT and FT^{-1} denote the 2D Fourier and inverse Fourier transforms, respectively; $n(x, y) \in [0, 2\pi)$ is a random distribution; $H = \exp(2\pi iz\sqrt{\frac{1}{\lambda^2} - u^2 - v^2})$ is the transfer function; and z is the propagation distance. The calculation conditions are as follows: the wavelength is 520 nm, the pixel pitch of the holograms is 8 µm, the propagation distance is 1.0 m, and the resolution of the natural image and propagated light waves are 512 × 512 and 509 × 509, which is a prime size. Figure 2b shows the real part of the propagated light waves. The compression results are shown in Tables 7 and 8. Although in Table 7 the compression ratios vary for the different combination, Table 8 shows that the RPT is the most effective transform when used with prime-sized data. The highest compression ratio is obtained using the RPT combined with fpzip.



(a) Natural image used. (b) The real part image of propagated light waves.

FIGURE 2. (a) The natural image used and (b) the real part of the image of the propagated light waves.

TABLE 7. Compression results for 512 × 512 propagated light waves.

	RAW	RFT	RPT	RST	DCT
fpzip	92.051*	94.270	92.140	93.792	94.442
zfp	98.985	97.086	93.962	96.599	94.948
bzip2	95.049	95.785	95.305	95.659	94.812
gzip	92.554	93.283	92.867	93.146	92.522
zip	92.570	93.299	92.882	93.161	92.538

TABLE 8. Compression results for 509 × 509 propagated light waves.

	RAW	RFT	RPT	RST	DCT
fpzip	92.038	94.155	87.903*	93.757	94.412
zfp	99.517	97.419	94.361	97.179	95.523
bzip2	95.061	95.710	94.465	95.675	94.832
gzip	92.557	93.210	92.428	93.146	92.532
zip	92.573	93.226	92.443	93.161	92.547

IV. CONCLUSION

In this Letter, we have demonstrated a lossless compression method using Ramanujan-sums-based transforms and holographic datasets. Some previous work has suggested that the Ramanujan sum may be useful for data compression, but did not demonstrate its effectiveness using real data. To our knowledge, this study is the first report of data compression using the Ramanujan sums. We found that with prime-sized-data we can obtain higher compression ratios by using the combination of RPT and fpzip.

REFERENCES

[1] S. Ramanujan, "On certain trigonometric sums and their applications in the theory of numbers," *Trans. Cambridge Philos. Soc.*, vol. 22, no. 13, pp. 259–276, 1918.

[2] A. M. Solomon, D. Abraham, A. George, and M. Manuel, "Signal denoising using Ramanujan periodic transform," in *Proc. 2nd Int. Conf. Trends Electron. Informat. (ICOEI)*, May 2018, pp. 51–56.

[3] L. Sugavaneswaran, S. Xie, K. Umapathy, and S. Krishnan, "Time-frequency analysis via Ramanujan sums," *IEEE Signal Process. Lett.*, vol. 19, no. 6, pp. 352–355, Jun. 2012.

[4] S. Samadi, M. O. Ahmad, and M. N. S. Swamy, "Ramanujan sums and discrete Fourier transforms," *IEEE Signal Process. Lett.*, vol. 12, no. 4, pp. 293–296, Apr. 2005.

[5] D. K. Yadav, G. Kuldeep, and S. D. Joshi, "Orthogonal Ramanujan sums, its properties, and applications in multiresolution analysis," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5789–5798, Nov. 2018.

[6] G. Chen, S. Krishnan, and T. D. Bui, "Matrix-based Ramanujan-sums transforms," *IEEE Signal Process. Lett.*, vol. 20, no. 10, pp. 941–944, Oct. 2013.

[7] P. W. M. Tsang and T.-C. Poon, "Review on the state-of-the-art technologies for acquisition and display of digital holograms," *IEEE Trans. Ind. Informat.*, vol. 12, no. 3, pp. 886–901, Jun. 2016.

[8] F. Yaraş, H. Kang, and L. Onural, "State of the art in holographic displays: A survey," *J. Display Technol.*, vol. 6, no. 10, pp. 443–454, Oct. 2010.

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

[10] T. Tahara, X. Quan, R. Otani, Y. Takaki, and O. Matoba, "Digital holography and its multidimensional imaging applications: A review," *Microscopy*, vol. 67, no. 2, pp. 55–67, Apr. 2018.

[11] F. Dufaux, Y. Xing, B. Pesquet-Popescu, and P. Schelkens, "Compression of digital holographic data: An overview," *Proc. SPIE*, vol. 9599, pp. 163–173, Sep. 2015.

[12] T. J. Naughton, Y. Frauel, B. Javidi, and E. Tajahuerce, "Compression of digital holograms for three-dimensional object reconstruction and recognition," *Appl. Opt.*, vol. 41, no. 20, pp. 4124–4132, 2002.

[13] D. Blinder, T. Bruylants, H. Ottevaere, A. Munteanu, and P. Schelkens, "JPEG 2000-based compression of fringe patterns for digital holographic microscopy," *Opt. Eng.*, vol. 53, no. 12, Dec. 2014, Art. no. 123102.

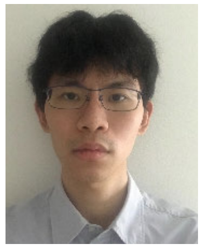
[14] R. Corda, D. Giusto, A. Liotta, W. Song, and C. Perra, "Recent advances in the processing and rendering algorithms for computer-generated holography," *Electronics*, vol. 8, no. 5, p. 556, May 2019.

[15] P. P. Vaidyanathan, "Ramanujan sums in the context of signal processing—Part II: FIR representations and applications," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4158–4172, Aug. 2014.

[16] P. Lindstrom and M. Isenburg, "Fast and efficient compression of floating-point data," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 5, pp. 1245–1250, Sep. 2006.

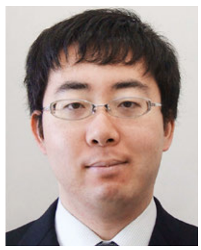
[17] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2674–2683, Dec. 2014.

[18] T. Shimobaba and T. Ito, *Computer Holography: Acceleration Algorithms and Hardware Implementations*. Boca Raton, FL, USA: CRC Press, 2019.



HARUTAKA SHIOMI received the B.E. degree in electrical and electronic engineering from Chiba University, Japan, in 2020. His research interests include signal processing and single-pixel imaging.

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



TAKASHI KAKUE (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electronics and information science from the Kyoto Institute of Technology, Japan, in 2006, 2008, and 2012, respectively. Since 2012, he has been an Assistant Professor at the Graduate School of Engineering, Chiba University, Japan. His research interests include holography, digital holography, computer holography, holographic interferometry, 3D imaging, high-speed imaging, and ultrafast optics. He is a member of OSA, OSJ, and SPIE.

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