

## Research Article

# Application of Mathematical Symmetrical Group Theory in the Creation Process of Digital Holograms

Agustín Pérez-Ramírez,<sup>1,2</sup> Julian Guerrero Juk,<sup>1</sup> Rafael Sanchez-Lara,<sup>2</sup>  
Joel Antonio Trejo-Sánchez,<sup>3</sup> and Lelio de la Cruz-May<sup>2</sup>

<sup>1</sup>Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Autopista del Sol Km 104, Real del Puente, 62790 Xochitepec, MOR, Mexico

<sup>2</sup>Facultad de Ingeniería, Universidad Autónoma del Carmen, 24180 Ciudad del Carmen, CAM, Mexico

<sup>3</sup>CONACyT-Centro de Investigación en Matemáticas, 97205 Mérida, YUC, Mexico

Correspondence should be addressed to Rafael Sanchez-Lara; rafaunacar@gmail.com

Received 13 March 2017; Revised 7 June 2017; Accepted 20 June 2017; Published 31 July 2017

Academic Editor: Pasquale Memmolo

Copyright © 2017 Agustín Pérez-Ramírez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents an algorithm to reduce the multiplicative computational complexity in the creation of digital holograms, where an object is considered as a set of point sources using mathematical symmetry properties of both the core in the Fresnel integral and the image. The image is modeled using group theory. This algorithm has multiplicative complexity equal to zero and an additive complexity  $(k - 1)N^2$  for the case of sparse matrices or binary images, where  $k$  is the number of pixels other than zero and  $N^2$  is the total of points in the image.

## 1. Introduction

The problem of reducing the computational time in the creation of digital holograms is becoming a demanding research topic due to the growing range of applications in different areas [1] such as optical microscopy, interferometry, entertainment, security, medicine, and education. The creation time of holograms has been reduced by using increasingly faster algorithms and hardware such as FPGA (Field Programmable Gate Array) and parallel programming techniques [2] and precomputed lookup tables [3, 4]. Nevertheless, another challenging approach seeks to reduce the computational complexity since it also can reduce the processing time. This work only focuses on reducing the multiplicative computational complexity in the creation of digital holograms since the time depends on the hardware of the computer and the programming techniques used (parallel programming).

Related work exists that has proposed the use of symmetry for reducing computational complexity and/or processing time. For example, run-length coding is used in [5]. Nevertheless, mathematical symmetry of the image is used only

partially, because only the redundant color of the image in adjacent pixels is used. In [6], Jiao et al. use radial symmetric interpolation to reduce the size of the precomputed lookup table by considering the redundancy in data; however they do not use the global mathematical symmetry to reduce the complexity in computing the hologram. This work considers the global mathematical symmetry to reduce such computational complexity.

Some authors use the symmetry of the principal fringe patterns [7], but the symmetry by reflection rather than the mathematical symmetry is considered. In [8], only a 45-degree diagonal of the zone plates is calculated, and the first quarter of the concentric circles quadrant of the zone plates is calculated from this diagonal using interpolation and data duplicity, and using the symmetry by reflection, the zone plates are fully calculated.

In this work, an algorithm using group theory is proposed, where MCC (multiplicative computational complexity) is zero in the generation of digital holograms for 2D images, two colors, and different sizes. Unlike other works, it successfully explores the use of the global mathematical

symmetry, a key consideration which allows reducing the MCC, which in turn contributes to increasing the speed of digital hologram generation. The memory size used to generate the hologram is the same as that reported [9]. The time to generate a hologram point is 0.009 ms compared to the N-LUT algorithm which takes 0.0170 ms. By this method, the computational complexity is reduced by approximately the half with respect to the N-LUT method.

The rest of the paper is divided as follows. Section 2 presents the development of the algorithm. Section 3 provides the results of computational complexity reduction of digital Fresnel holograms. Finally, Section 4 presents the concluding remarks.

## 2. Development

Group theory to reduce the computational complexity in the creation of digital holograms is used, by considering an object as a set of point sources [6, 10–12]. It is known that if the symmetry group of a problem is known, this can greatly simplify the process of creating digital holograms [13–15]. The problem of creating digital holograms depends on three components, namely: the image, the transform, and the hologram. Each one of these components has its own symmetry properties, which can be helpful to reduce the computational complexity [15, 16].

If there exists redundancy in a digital image, particularly in the core of the transform or in the hologram itself, then it is possible to use mathematical symmetry to reduce the MCC of the digital hologram generation [5, 13, 14].

A mathematical symmetry group is a set of permutations such that when they are applied to an object, the object remains invariant. It is worth mentioning that mathematical symmetry includes axial symmetry. Thus, a 2D image can be represented as vector where the same permutations of the same color pixel are applied without changing the original image. The total number of gray levels of an image is 256, whereas the image size in pixels is usually greater than 256. Thus, for a gray scale image (where colors are codified from 0 to 255), there will be some repetition of colors of pixels, based on the principle of the Dirichlet [17]. Therefore, there will always be a group of permutations acting over the same color pixels of the image without changing it, due to the fact that, in any 2D or 3D images, there are pixels with the same color; with processing purposes the pixels can be permuted without any change in the image; the same can be made for the hologram generated. Here the application of the core of the transform to an object is analyzed, so that the mathematical symmetry can be obtained.

To use the symmetry of an image in the process of creating holograms to reduce the computational complexity, it is necessary to calculate the maximum symmetric group of the image. Mathematically, images can be two-dimensional, three-dimensional, or  $n$ -dimensional. In fact, an  $n$ -dimensional image with  $m$  colors can be presented as a set of tuples  $A$ .

$$A = \{(x_1, x_2, \dots, x_n; c_1, c_2, c_3, \dots, c_m)\}, \quad (1)$$

where  $x_1, x_2, x_3, \dots, x_n$  are the pixel coordinates of an  $n$ -dimensional image and  $c_1, c_2, c_3, \dots, c_m$  are the components of the color vector for each pixel.

It is assumed that the image is transformed by an operator  $T$  into a set of tuples  $A$  and that the inverse transform is  $T^{-1}$ . The elements of the set  $A$  are the tuples that represent all pixels of the image. The place of the tuple in a set  $A$  represents the place of the corresponding pixel in the image.

The transforms used to generate holograms are linear, so the generation process for each color can be carried out separately, thus only one component  $c$  in the color vector can be assumed without loss of generality, as follows:

$$A = \{(x_1, x_2, \dots, x_n; c)\}. \quad (2)$$

Also, the next properties can be assumed for the digital images.

Suppose that  $N_1, N_2, \dots, N_n$  are the size of each dimension; then

$$\begin{aligned} x_1 &= 1, \dots, N_1, \\ x_2 &= 1, \dots, N_2, \\ &\vdots \\ x_n &= 1, \dots, N_n, \\ c &= 0, \dots, M - 1. \end{aligned} \quad (3)$$

Then, the set  $A$  can be partitioned according to the color  $c$  as follows:

$$A = A_0 \oplus A_1 \oplus A_2 \cdots \oplus A_{M-1}, \quad (4)$$

where

$$A_i = \{(x_1^i, x_2^i, \dots, x_n^i; i)\}, \quad i = 0, \dots, M - 1, \quad (5)$$

and  $\oplus$  is the operator for direct addition of groups.

Then factoring the color,  $A_i$  can be represented as

$$A_i = i * \{(x_1^i, x_2^i, \dots, x_n^i; 1)\} = i * B_i, \quad (6)$$

where  $*$  is the multiplication operation for real numbers

$$B_i = \{(x_1^i, x_2^i, \dots, x_n^i; 1)\}, \quad \text{for } i = 0, \dots, M - 1. \quad (7)$$

Thus, in (6), the factor  $i$  multiplies only the last element of the tuple, which is the color. The set  $B_i$  ( $i = 0, 1, \dots, M - 1$ ) can be considered as a vector whose dimension is  $N_1 \times N_2 \times \cdots \times N_n$ . The set of values  $B_i$  ( $i = 0, 1, \dots, M - 1$ ) can be considered as a basis to represent the image as follows:

$$A = B_0 + \sum_{i=1}^{M-1} i * B_i. \quad (8)$$

$B_0$  is the set of pixels of the image with a gray level of zero; this is because the color component of its tuples does not need to be multiplied for some  $i$ . This means less multiplication operations. Equation (8) allows defining a maximum symmetric group ( $G$ ) of the image:

$$G = S^0 (\text{card}(A_0)) \times S^1 (\text{card}(A_1)) \times \dots \times S^{M-1} (\text{card}(A_{M-1})), \quad (9)$$

where  $\text{card}(A_i)$  is the total number of elements of  $A_i$ .  $S^i(\text{card}(A_i)) = S^i$  is a symmetric group (permutation group) defined over  $A_i$  or  $B_i$  ( $i = 0, 1, \dots, M-1$ ).

It is evident that applying the group  $S^i$  to the sets  $A_i$  or  $B_i$  (considered as sets of pixels) modifies them, but it does not alter the corresponding images that are related to the set of tuples  $A_i$  or  $B_i$ . Applying group  $G$  to the image does not produce any changes to it. Therefore,

$$G(A) = S^0(B_0) + \sum_{i=1}^{M-1} i * S^i(B_i) = B_0 + \sum_{i=1}^{M-1} i * B_i = A. \quad (10)$$

According to (8), the image can be stated as the sum of the partial images  $B_i$  ( $i = 0, 1, \dots, M-1$ ). The values of the color of the pixels corresponding to the set of tuples  $B_i$  are equal and have a value of 1. Hence, the multiplicative computational complexity of creating digital holograms from images corresponding to the sets  $B_i$  ( $i = 0, 1, \dots, M-1$ ) is zero (there are no multiplications). On the other hand, it must be taken into account that images corresponding to the set of tuples  $B_i$  are sparse and this can also minimize the additive computational complexity. Digital holograms of partial images corresponding to the set of tuples  $B_i$  are named ‘‘partial holograms’’ and they are denoted by  $D(B_i)$  which can be obtained by (15) and the methodology of [9]. Since the transformed images to generate the corresponding holograms are linear, according to (8), the hologram corresponding to the set of tuples  $A$  can be presented as follows:

$$D(A) = D(B_0) + \sum_{i=1}^{M-1} i * D(B_i). \quad (11)$$

From (11), the multiplicative computational complexity (MCC) of the hologram creation process from the set of tuples  $A$  is given by

$$\text{MCC} = (M-1) * F_1 * F_2 * F_3 * \dots * F_k, \quad (12)$$

where  $k$  is the number of dimensions of the hologram  $D(A)$  and  $F_i$  is the size of each dimension. For example, for images and two-dimensional holograms presented with size  $N \times N$  and  $M = 256$ , the multiplicative computational complexity has a value of

$$\text{MCC} = (M-1) * N^2 = 255 * N^2. \quad (13)$$

This is so because each point in the hologram is obtained with 255 multiplications and the whole hologram has a total of  $N \times N$  points. In the extreme case that the image consists of two colors, for example, a blue cube with white background, the computational complexity is

$$\text{MCC} = 2N^2. \quad (14)$$

On the other hand, the equation for obtaining a hologram  $H(x, y)$  is given by

$$\begin{aligned} H(x, y) &= |R(x, y) + O(x, y)|^2 \\ &= |R(x, y)|^2 + |O(x, y)|^2 \\ &\quad + 2 |R(x, y)| |O(x, y)| \\ &\quad \times \cos(kr + kx \sin(\theta_R) + \varphi_p), \end{aligned} \quad (15)$$

where  $O(x, y)$  is the complex amplitude of the hologram plane and  $R(x, y)$  is the reference beam defined as follows:

$$O(x, y) = \sum_{p=1}^N \frac{a_p}{r_p} \exp(j(kr_p + \varphi_p)) \quad (16)$$

$$R(x, y) = a_R \exp(j(kx \sin(\theta_R))),$$

where

$$r_p = \sqrt{(x - x_p)^2 + (y - y_p)^2 + z_p^2}, \quad (17)$$

$k = 2\pi/\lambda$  is the wave number, and  $\lambda$  is the wavelength.

The N-LUT method which is like a convolution [18–21] can be applied only to the third element of (15) and the property of mathematical symmetry of the image; in this way the multiplicative computational complexity is reduced.

For example, a binary image (black and white) would make zero multiplications and  $(k-1)N^2$  sums, where  $k$  is the number of pixels other than zero. For the case of sparse and binary images, where  $k$  is a small number, the multiplicative complexity is kept at zero while the additive complexity decreases a lot.

To clarify the results above, let us consider the following example. For a hologram of one row  $1 \times 5$  and one image of the same size, the distances from each image point to each hologram point depict a symmetry distance vector ( $d_5, d_4, d_3, d_2, d_1, d_2, d_3, d_4$ ) as shown in Figure 1; each distance or its approximation obtained by using the MaClaurin polynomial is used in the core of the Fresnel integral.

Once the distances from the first point have been calculated, these can be used to calculate the distances of other points of the hologram, so it is no longer necessary to make redundant calculations of distances. This can be extended to two dimensions as shown from Figure 2 through Figure 5. Figure 2 shows the distance matrix from an image of  $3 \times 3$  to the first point of the hologram.

Similarly, Figure 3 shows the distance matrix can be obtained for all points of the hologram.

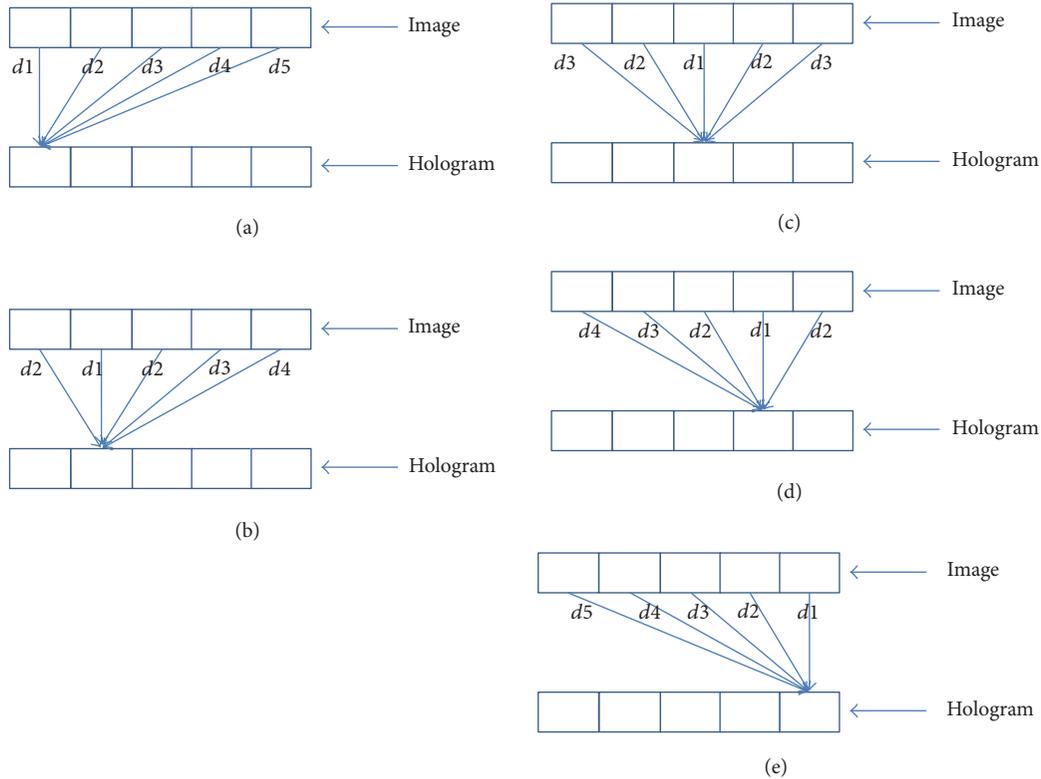


FIGURE 1: Redundant distances generated by symmetry for the 5 hologram points ((a), (b), (c), (d), and (e)) with respect to the image points.

$d_1$	$d_2$	$d_3$
$d_4$	$d_5$	$d_6$
$d_7$	$d_8$	$d_9$

FIGURE 2: Distance matrix of a  $3 \times 3$  image.

$d_9$	$d_8$	$d_7$	$d_8$	$d_9$
$d_6$	$d_5$	$d_4$	$d_5$	$d_6$
$d_3$	$d_2$	$d_1$	$d_2$	$d_3$
$d_6$	$d_5$	$d_4$	$d_5$	$d_6$
$d_9$	$d_8$	$d_7$	$d_8$	$d_9$

FIGURE 3: Distance matrix for all points of the hologram.

Figure 4 shows how Figure 3 is obtained. The matrix of Figure 4(b) is obtained by reflecting to the left the last two columns of the matrix of Figure 4(a). The matrix of Figure 4(c) is obtained by reflecting the last two rows of the matrix of Figure 4(b) upwards.

The hologram can be obtained from Figure 4(c). To obtain the first point of the hologram, a  $3 \times 3$  submatrix is used; in this first case, a  $3 \times 3$  submatrix located in the lower right corner in Figure 4(c) is considered. The dimensions of this submatrix are the same as the dimensions of the original image. For the next point of the hologram, it is considered again a  $3 \times 3$  submatrix but this time shifted one column to the left. This procedure is repeated until the row is completed (see Figure 5).

Each submatrix is applied to the image as a dot product between matrices so that all points of this row of the hologram are calculated.

To obtain the second row of the hologram, we use the  $3 \times 3$  submatrix shown in the lower left corner of Figure 5; the

same process is repeated as in the first row, and so on, until the whole hologram is generated.

Using this property of mathematical symmetry of the distances and image, the multiplicative computational complexity is reduced in the creation of digital holograms.

### 3. Results

Given a two-dimensional image, the hologram is created using the algorithm described in this work. It is worth mentioning that the algorithm does not improve the quality of the holograms of Fresnel using the N-LUT method and the peak signal-to-noise ratio (PSNR), and the structural similarity (SSIM) index is the same. This is because the methods of Ray-tracing, LUT, N-LUT, and the proposed method use the same equations to calculate the hologram. The difference is only the computational complexity of the algorithm.

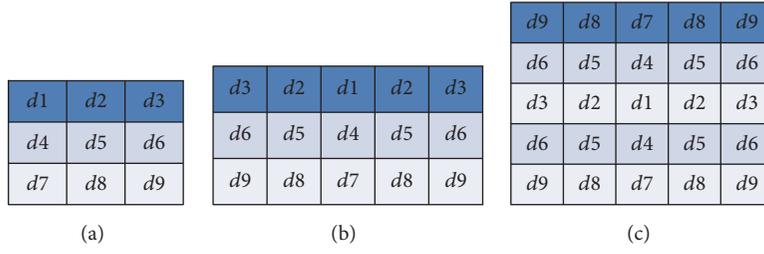


FIGURE 4: The sequence of (a), (b), and (c) shows how to obtain a distances symmetric matrix with respect to  $d_1$  in (c).

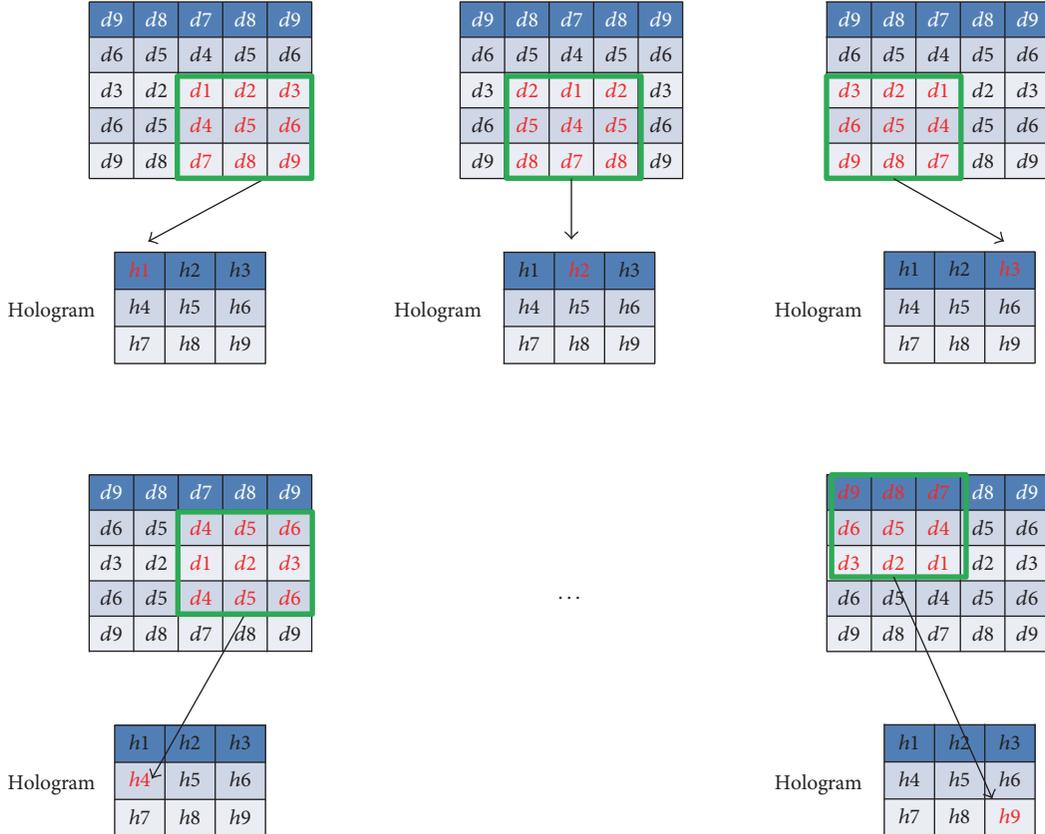


FIGURE 5: Each point of the hologram is computed by performing the dot product between the image and the distance matrix. In this example, the size of both, the hologram and the image, is  $3 \times 3$  pixels.

TABLE 1: Comparison of computation time and memory size for each method. For a grayscale image.

	N-LUT	Proposed algorithm
Computation time for 1 point (ms)	0.0170	0.00993
Total memory size for LUT (megabytes)	0.0625	0.0625

TABLE 2: Comparison of computation time and memory size for each method. For a binary image and sparse.

	N-LUT	Proposed algorithm
Computation time for 1 point (ms)	0.00207	0.00207
Total memory size for LUT (megabytes)	0.0625	0.0625

Let us now analyze the computational complexity through the image of Figure 6(a) and its binary form of Figure 6(d). The hologram and reconstruction are shown in Figures 6(b) and 6(c) and Figures 6(e) and 6(f), respectively.

The algorithm was programmed in Matlab and data from [22] was used.

The time used to obtain each point of the hologram as well as the amount of memory required is shown in Table 1. Table 2 shows the computational complexity of the N-LUT

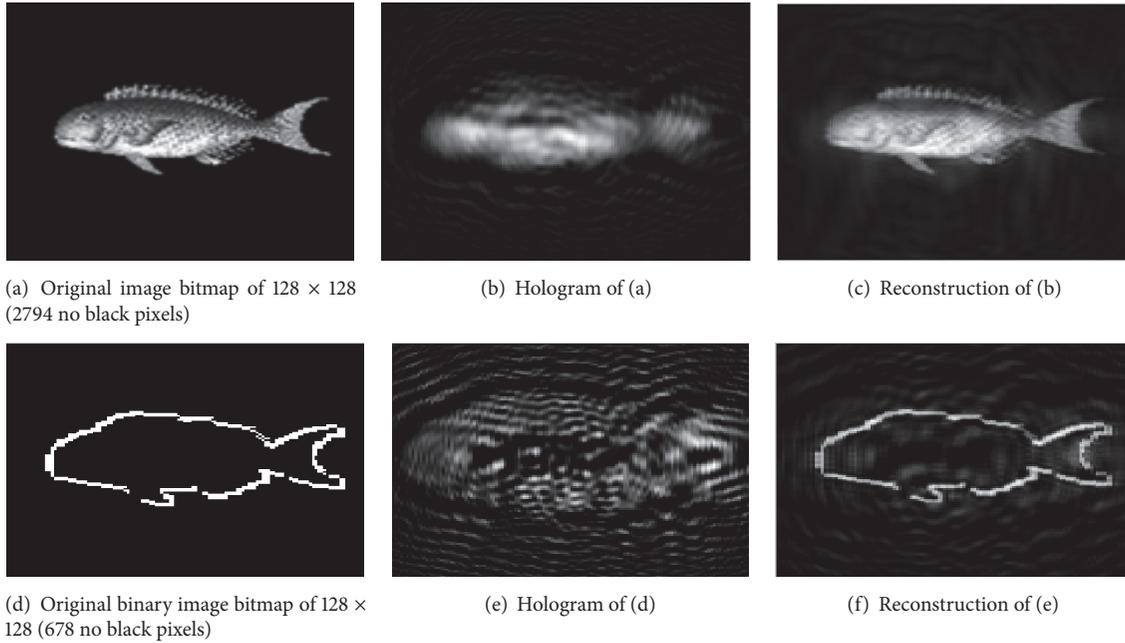


FIGURE 6: Different holograms and their reconstructions with  $\lambda = 632$  nm and  $z = 0.516$  (proposed algorithm). The image has a size of  $128 \times 128$  pixels.

TABLE 3: Computational complexity for a grayscale image.

	N-LUT	Proposed algorithm
Multiplicative computational complexity	$N^4$	$255 * N^2$
Additive computational complexity	$N^4$	$N^4$

TABLE 4: Computational complexity for a binary image.

	N-LUT	Proposed
Multiplicative computational complexity	0	0
Additive computational complexity, where $p$ is the number of pixels other than zero.	$N^2(p - 1)$	$N^2(p - 1)$

method and our method for obtaining the digital hologram of a grayscale image. The computational complexity of both methods for a digital hologram of a binary image is shown in Table 3.

Note in Table 4 that the computational complexity for a binary and sparse image is much lower than for a grayscale image.

#### 4. Conclusions

The use of group theory and image symmetry properties allow the reduction of the computational complexity in the creation of digital holograms.

Our algorithm applied on two colors images has a positive impact on multiplicative computational complexity, which is reduced from  $255N^2$  to 0.

It is relevant to mention that the algorithm for binary and sparse images has a very low or zero multiplicative computational complexity and it is independent of image size; however, the additive complexity has not been significantly reduced.

The computational complexity for a grayscale image is reduced approximately to the half with respect to N-LUT algorithm.

#### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

#### Acknowledgments

The authors gratefully acknowledge the PRODEP (UNACAR) and the CONACyT FOMIX (CIMAT Yucatan 2014) for their support.

#### References

- [1] U. Schnars, *Digital Holography*, Springer, New York, NY, USA, 2005.
- [2] Y. Ichihashi, H. Nakayama, T. Ito et al., "HORN-6 special-purpose clustered computing system for electroholography," *Optics Express*, vol. 17, no. 16, pp. 13895–13903, 2009.
- [3] M. Lucente, "Interactive computation of holograms using a look-up table," *Journal of Electronic Imaging*, vol. 2, no. 1, pp. 28–34, 1993.

- [4] J. Jia, Y. Wang, J. Liu et al., "Reducing the memory usage for effective computer-generated hologram calculation using compressed look-up table in full-color holographic display," *Applied Optics*, vol. 52, no. 7, pp. 1404–1412, 2013.
- [5] T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Fast calculation of computer-generated hologram using run-length encoding based recurrence relation," *Optics Express*, vol. 23, no. 8, pp. 9852–9857, 2015.
- [6] S. Jiao, Z. Zhuang, and W. Zou, "Fast computer generated hologram calculation with a mini look-up table incorporated with radial symmetric interpolation," *Optics Express*, vol. 25, no. 1, pp. 112–123, 2017.
- [7] D.-W. Kwon, S.-C. Kim, and E.-S. Kim, "Efficient digital hologram generation using reflection symmetry of principle fringe pattern," in *Proceedings of the ICTC 2010*, pp. 197–198, IEE, Jeju, South Korea, November 2010.
- [8] S. Lee, H. C. Wey, D. K. Nam, D. S. Park, and C. Y. Kim, "Fast hologram pattern generation by radial symmetric interpolation," in *Optics and Photonics for Information Processing VI*, Proceedings of SPIE, pp. 84980O-1–84980O-7, San Diego, Calif, USA, August 2012.
- [9] S.-C. Kim and E.-S. Kim, "Effective generation of digital holograms of three-dimensional objects using a novel look-up table method," *Applied Optics*, vol. 47, no. 19, pp. D55–D62, 2008.
- [10] S. Cho, B.-K. Ju, N.-Y. Kim, and M.-C. Park, "One-eighth look-up table method for effectively generating computer-generated hologram patterns," *Optical Engineering*, vol. 53, no. 5, Article ID 054108, 2014.
- [11] N. Okada, T. Shimobaba, Y. Ichihashi et al., "Fast calculation of a computer-generated hologram for RGB and depth images using a Wavefront recording plane method," *Photonics Letters of Poland*, vol. 6, no. 3, pp. 90–92, 2014.
- [12] H. Wei, G. Gong, and N. Li, "Improved look-up table method of computer-generated holograms," *Applied Optics*, vol. 55, no. 32, pp. 9255–9264, 2016.
- [13] Ye. A. Shelomov, "On  $m$  correlation matrices," *II journal communications Technology and electronics*, vol. 39, no. 13, pp. 79–91, 1994.
- [14] E. Shelomov, J. Seguel, A. Ortiz, J. Ruz, and G. Shelomova, "Análisis de Fourier en grupos finitos con utilidades de Matlab," in *Proceedings of the CINVESTAV, 2002*, pp. 90–116, CINVESTAV, Mexico City, Mexico, June 2002.
- [15] M. Hamermesh, *Group Theory and Its Applications to Physical Problems*, Addison-Wesley, New York, NY, USA, 1962.
- [16] D. K. Maslen and D. N. Rockmore, "The cooley–tukey FFT and group theory. Modern signal processing," *MSRI Publications*, vol. 46, 2003.
- [17] R. Johnsonbaugh, *Discrete Mathematics*, Prentice Halls, New York, NY, USA, 6th edition, 2005.
- [18] Poon T.-C. and P. P. Banerjee, *Contemporary Optical Image Processing with MATLAB*, Rastogi and Inaudi, Oxford, UK, 1st edition, 2001.
- [19] T. C. Poon, *Optical Scanning Holography with MATLAB*, Springer Science and Business Media, Berlin, Germany, 2007.
- [20] J. W. Goodman, *Introduction to Fourier Optics*, Roberts and Company Publishers, 2005.
- [21] R. P. Muffoletto, *Numerical techniques for Fresnel diffraction in computational holography [Ph.D. thesis]*, University of Louisiana, 2006.
- [22] T. F. Raúl, *Modulación completa del plano complejo mediante pantallas de cristal líquido. Aplicación a la reconstrucción de hologramas de Fresnel digitales [Ph.D. thesis]*, University of Barcelona, 2004.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

