Optics and Lasers in Engineering 122 (2019) 312–318

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Non-iterative direct binary search algorithm for fast generation of binary holograms

Ji-Hoon Kang [a,1], Thibault Leportier [a,1], Myungha Kim [b], Min-Chul Park [a,c,*]

[a] Center for Opto-Electronic Materials and Devices, Post-Silicon Semiconductor Institute, Korea Institute of Science and Technology (KIST), Seoul 02792, Republic of Korea

[b] Department of Computer Science, Seokyeong University, Seoul 02173, Republic of Korea

[c] Division of Nano & Information Technology, KIST School, Korea University of Science and Technology, Seoul 02792, Republic of Korea

ARTICLE INFO

Keywords:
Holography
Binary hologram
Direct binary search

ABSTRACT

The direct binary search (DBS) algorithm is an efficient method for the generation of binary holograms, but it is also an iterative method involving lengthy computation. Thus, fast non-iterative approaches are more preferred in practice even though they yield poorer results. In this paper, we propose a strategy to drastically reduce the computational time of the DBS algorithm. First, we show that convergence of the conventional DBS algorithm can be significantly improved by optimizing the order in which the pixels are examined. Then, we demonstrate the efficiency of a design based on optimization of multiple small blocks of binary pixels through parallel computation. Since each block can be optimized in parallel utilizing platforms such as those offering cloud computing services, the time to compute the final pattern is determined by the computational time for a single block. The proposed block-partition strategy involves a trade-off between the computation time and the quality of the final hologram. However, it should be noted that simply randomizing the pixel examination order during the DBS procedure reduces the computational time by 67% even without parallel computation. In summary, our proposed method facilitates easier generation of high-quality binary holograms in less time than is required by the conventional DBS.

1. Introduction

The ability of digital holography to provide 3D information has allowed it to be applicable to multiple fields. To date, several methods have been developed for the acquisition of complex holograms, such as optical scanning holography (OSH) [1–3], phase shifting [4–6], and Fresnel incoherent correlation holography (FINCH) [7–9]. Continuous enhancement of computational power can further improve the possibility of digital hologram calculation via virtual model-based numerical methods [10,11].

Numerical reconstruction of complex holograms can be achieved by applying the Fresnel formula or the angular spectrum [12,13]. Since 3D information is included in the hologram, the focus of the reconstruction plane can be adjusted to visualize the desired plane. However, optical reconstruction is more difficult because the spatial light modulator (SLM) devices used for the display are not suitable for complex modulation. Several reports have described complex modulation with the use of a single SLM [14–20] or multiple SLMs [21,22]. The ideal way to display holographic information is to perform complex modulation. Complex modulation enables the reconstruction of a complex optical field and, therefore, helps avoid undesired twin image and zeroth order contributions. However, it usually reduces the resolution (the pixels are divided into amplitude and phase) or slows the process down (amplitude and phase are displayed on two time frames). Resolution and response time are currently strong limitation factors in the SLM technology that limits the total size of the viewing window where holograms can be displayed. The binary format offers many advantages in the field of optical reconstruction. One major challenge for holographic display is the increased size of 3D images, and the binary format is the best option for time multiplexing because it exploits the best bandwidth of the SLM. However, the conversion of a complex hologram into a binary hologram inevitably leads to loss of information. Since the proposal of the binary format, several methods have been proposed to minimize the loss-induced noise introduced in the image plane during the conversion process.

The easiest method known to convert a complex hologram into binary format requires performing a simple threshold operation based on

* Corresponding author at: Center for Opto-Electronic Materials and Devices, Korea Institute of Science and Technology (KIST), 5, Hwarang-ro 14-gil, Seongbuk-gu, Seoul, Republic of Korea.

E-mail address: minchul@kist.re.kr (M.-C. Park).

[1] Authors with equal contribution.

https://doi.org/10.1016/j.optlaseng.2019.06.017
Received 22 April 2019; Received in revised form 18 June 2019; Accepted 21 June 2019
Available online 27 June 2019

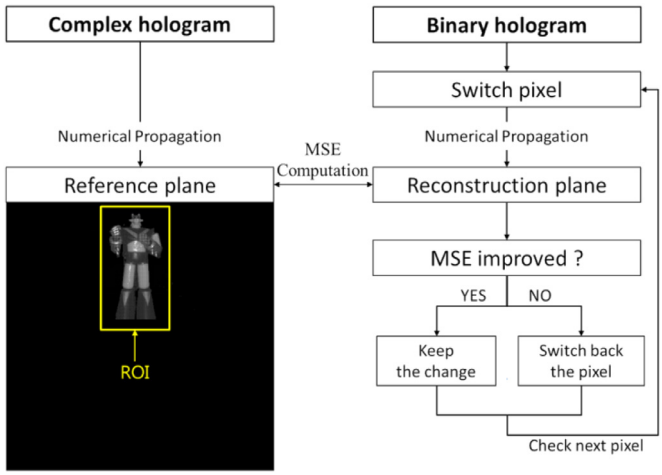

Fig. 1. Principle of the DBS algorithm.

the sign of the real part of the holographic data. This method is fast, but leaves a large amount of noise in the reconstruction plane. Another non-iterative approach based on bidirectional error diffusion (BERD) [23] has been shown to yield better results by spreading the quantification noise away from the center of the image.

It is generally accepted that the method that yields the best results for the generation of binary holograms is the direct binary search (DBS) algorithm [24,25]. However, the price to pay for implementing the DBS algorithm is prolonged computational time, since it is an iterative method that examines each pixel multiple times. Although a few improvements have been proposed to reduce the computational time [26,27], the DBS algorithm remains a slow iterative method. One major problem with the DBS algorithm is that the procedure cannot be implemented in parallel because every operation influences the succeeding operations.

In this study, therefore, we investigated the influence of the order in which the pixels are examined in order to reduce the number of iterations necessary to obtain a satisfying result. Then, we developed a method based on block partition to enable parallel processing of the DBS algorithm and thus substantially decrease the computational time.

2. Conventional direct binary search algorithm

2.1. Principle of the algorithm

The principle of the DBS algorithm is illustrated in Fig. 1. The image plane obtained via numerical reconstruction of the original complex hologram at a given distance z is computed and used as a reference. Then, the binary hologram is generated, beginning from a random binary pattern.

Each pixel of the pattern is tested individually, with the two possible pixel values being considered each time. After each pixel is tested, numerical reconstructions of the two binary patterns, which are exactly the same with the exception of the pixel under consideration, are computed and compared to the reference. The comparison is made by computing the mean square error (MSE) as follows:

$$\mathrm{MSE} = \frac{1}{M \cdot N} \frac{\sum_{m=\frac{-M}{2}}^{\frac{M}{2}-1} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \left(Im_{ref}(m,n) - k \cdot Im_{bin}(m,n) \right)}{\sum_{m=\frac{-M}{2}}^{\frac{M}{2}-1} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \left| Im_{ref}(m,n) \right|^2} \tag{1}$$

with

$$k = \frac{\sum_{m=\frac{-M}{2}}^{\frac{M}{2}-1} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \left(Im_{ref}(m,n) \cdot conj\left[Im_{bin}(m,n) \right] \right)}{\sum_{m=\frac{-M}{2}}^{\frac{M}{2}-1} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \left| Im_{bin}(m,n) \right|^2} \tag{2}$$

where Im_{ref} and Im_{bin} are the complex amplitudes computed using the Fresnel propagation formula from the complex hologram and the binary pattern, respectively; (M, N) refers to the number of rows and columns of the images; and $conj[.]$ is the operator that gives the conjugate of a complex number. The pixel value yielding the smallest MSE is selected, and then the next pixel is tested. One iteration is complete when all the pixels have been tested once; the next iteration then starts following the same procedure.

The MSE is computed only for a given region of interest (ROI) in the image plane containing the object, as shown in Fig. 1. This is a crucial advantage of the DBS algorithm over other methods as it enables optimization of the hologram for a selected area of the reconstruction plane. The size and position of the ROI affects the value of the MSE and, therefore, the final result. Similar to observations made when using phase retrieval algorithms, where the phase holograms are embedded within a large area with zero padding, optimization in this case gives a better result when the ROI is small with respect to the total size of the optimized pattern. A large ROI would enable optimization of the reconstruction of larger objects, but the remaining background noise would also be higher on average. It is also important to realize that even if the original hologram is complex, the pattern computed with the binary algorithm is not. The conjugate of the object will then be present in the plane reconstructed from the binary pattern (see Fig. 2, for instance). In order to ensure optimal convergence of the algorithm, it is usually best to ensure that the ROI does not include the area containing the twin image.

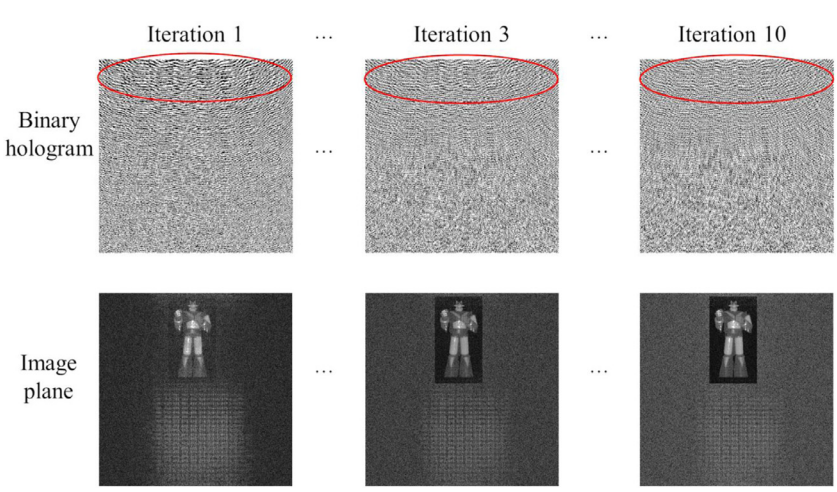

Fig. 2. Binary patterns and their corresponding reconstruction planes at different stages of computation of the DBS algorithm.

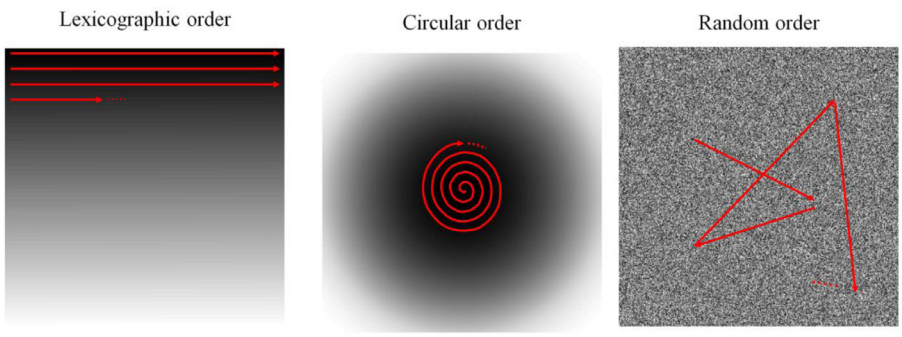

Fig. 3. Illustration of pixel order strategies for the DBS procedure.

2.2. Order of examination of pixels

One advantage of the DBS algorithm is that the optimization is not independently applied to each pixel, since the entire pattern is optimized. Thus, the order in which the pixels are examined is important. Conventional implementation of the DBS algorithm involves testing the pixels in lexicographic order, i.e., from the top-left to the bottom-right. This strategy is considered not very efficient, because the initial optimization of the patterns may be cancelled out in subsequent iterations by the newly computed binary patterns. This phenomenon is illustrated in Fig. 2. The complex hologram used here as an example is a hologram of 512 × 512 pixels computed from an amplitude image and depth map acquired with a Kinect system (Microsoft, Kinect sensor V2).

To prevent the generation of undesired patterns in the initial state, we investigated strategies that examine the pixels in a different order. The pixels in the center may contribute to the reconstruction plane more significantly because their diffraction pattern is scattered in all directions and thus overlap with the diffraction pattern of all the surrounding pixels. In contrast, the bordering pixels may be relatively less influential. Based on this assumption, we tested a case in which the pixels are examined in a circular order, beginning from the center of the hologram. We also tested a case in which the pixels are simply examined one by one in a randomized order. All three of the pixel examination orders are illustrated in Fig. 3: the pixels were ordered from 1 (black) to N (white), with N being the total number of pixels (262,144 pixels in this example).

We observed that the convergence of the circular order-based DBS algorithm was only slightly faster than that of the conventional lexicographic order-based DBS. Moreover, the undesirable pattern visible at the end of the first iteration with the lexicographic order was only partially attenuated. However, the random order-based DBS algorithm yielded significantly improved results. Fig. 4 shows the binary holograms and their corresponding reconstruction planes for each strategy.

The convergence behavior was examined for each case by computing the MSE between the ideal complex amplitude and the reconstructions obtained from the binary holograms at the end of each iteration. Fig. 5 presents the evolution of the MSE as a function of the iteration number. The MSE is normalized by the initial MSE value computed using the reconstruction obtained from the initial random binary pattern.

The empirical criterion of visual satisfaction is satisfied if the noise is almost visually imperceptible when the normalized MSE is less than 10%. It is a fact that approximately four iterations are necessary to reach the final state when the pixels are examined in a randomized order; however, in this case, the image quality was found to be very satisfying after a single iteration, as shown in Fig. 4. The computational time for each strategy is listed in Table 1. Since the order of pixel examination was the only differentiating factor, the computational time for a single iteration was the same for all three cases. The hardware used in this study was an AMD Ryzen 7 1700X CPU with 64GB of RAM, and the computational time for one iteration was 22 min.

As mentioned previously, the long computational time is the primary drawback of the DBS algorithm. Thus, although it is a very efficient technique, it is rarely used in practice because of this undesirable feature. We have demonstrated that, by changing the order in which the pixels are examined during the DBS procedure, the convergence time of the algorithm could be significantly reduced. Moreover, we have demonstrated that an acceptable result could even be obtained after only a single iteration.

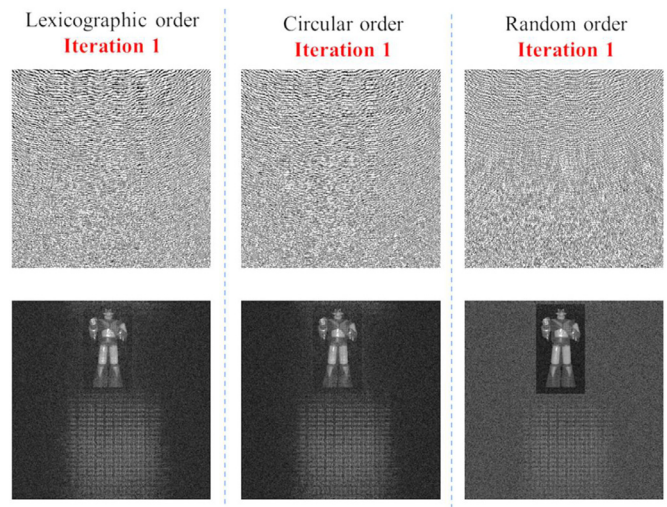

Fig. 4. Effect of the pixel examination order. The binary patterns and corresponding reconstruction planes obtained after a single iteration are shown for the three different strategies.

Table 1
Computational time required for each DBS algorithm.

	Visually satisfying	Final result
Lexicographic	3 iterations (~1 h 6 min)	6 iterations (~2 h 12 min)
Circular	3 iterations (~1 h 6 min)	6 iterations (~2 h 12 min)
Random	1 iteration (~22 min)	4 iterations (~1 h 28 min)

Since every pixel should be examined at least once, the only way to reduce the computational time beyond that required for a single iteration is to perform operations in parallel. However, when a pixel is tested, it is the entire pattern that is used to reconstruct the image plane. Therefore, the operations are not independently performed, and parallel implementation cannot be considered s feasible when a conventional DBS method is used to implement the algorithm.

3. Parallel implementation of DBS algorithm

3.1. Principle of our proposed block-partition-based method

When a single pixel is examined to determine is optimal value, numerical reconstruction of the entire pattern is computed, and only this

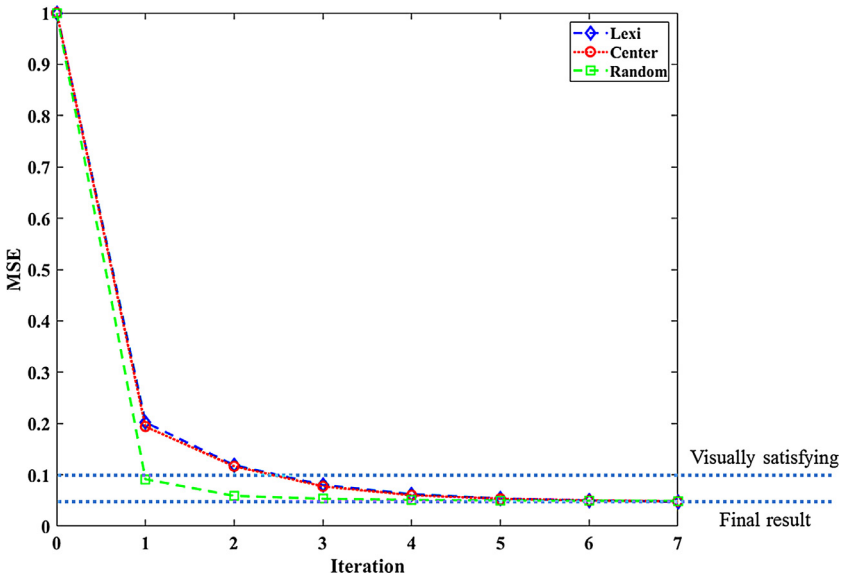

Fig. 5. Evolution of the normalized MSE as a function of iteration number.

specific pixel value is switched. At the beginning of the procedure, the surrounding pixels are considered to be a random pattern, as they do not actively contribute to object reconstruction. However, as the procedure advances, more pixels become optimized. Then, for the last pixels to be examined, the pixel value is determined as based on the cumulative contribution of the pattern. This means that there are differences in the significance of the examined pixels, because the first pixel values are predominantly determined based on their own contribution, whereas the last pixel values benefit from the active contribution of all of the previously examined pixels. However, this problem can be solved in the conventional DBS algorithm by successively examining the pixels over several iterations.

In this study, we modified the DBS algorithm by incorporating a block-partition strategy to optimize the process of pixel value selection. Instead of examining a pixel as a part of an initial random binary pattern, only a block of binary pixels surrounded by the original complex values are optimized, as illustrated in Fig. 6.

Our proposed method optimizes each block in a way similar to the conventional DBS algorithm, with a randomized order of pixel examination. In the case illustrated in Fig. 6, four blocks were independently computed and assembled at the end of the procedure to form the final binary hologram. Since the optimization of a block is performed within a hologram that contains complex reference information, even determination of the value of the first examined pixel is based on some constructive information.

One very important feature of our method is that the computation for each block is performed completely independent of that for all other blocks, thus enabling parallel processing. In addition, the main reason for performing several iterations is to confirm that the pixel values within a pattern were optimized during the first iteration. Furthermore, with our method, we also observed that subsequent iterations yielded negligible effect on the final result, as a satisfying result could be obtained with a single iteration.

3.2. Performance of the method

We tested our method, using blocks of varying sizes, against the threshold method, BERD, and the conventional DBS method that implements a single block. The proposed algorithm was implemented by operating CUDA (Compute Unified Device Architecture) on an NVIDIA

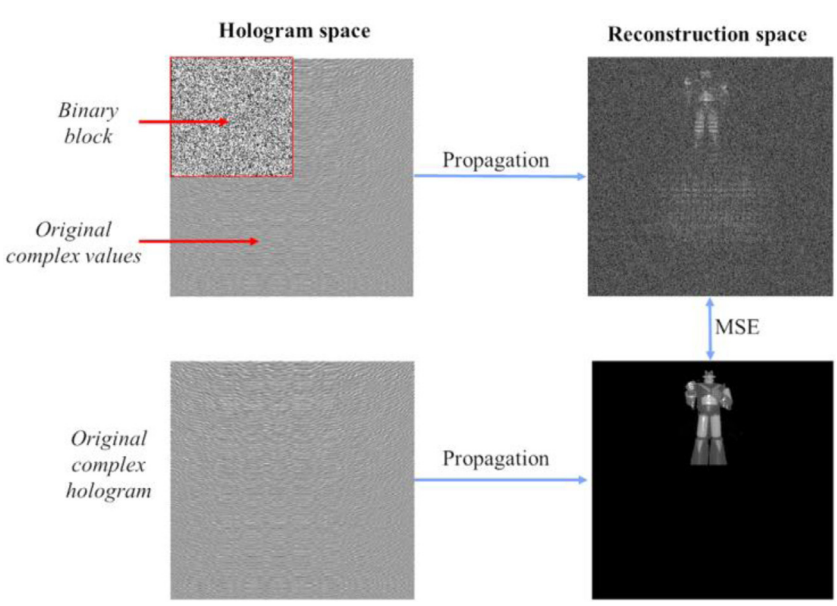

Fig. 6. Principle of the proposed block-partition strategy for the DBS algorithm.

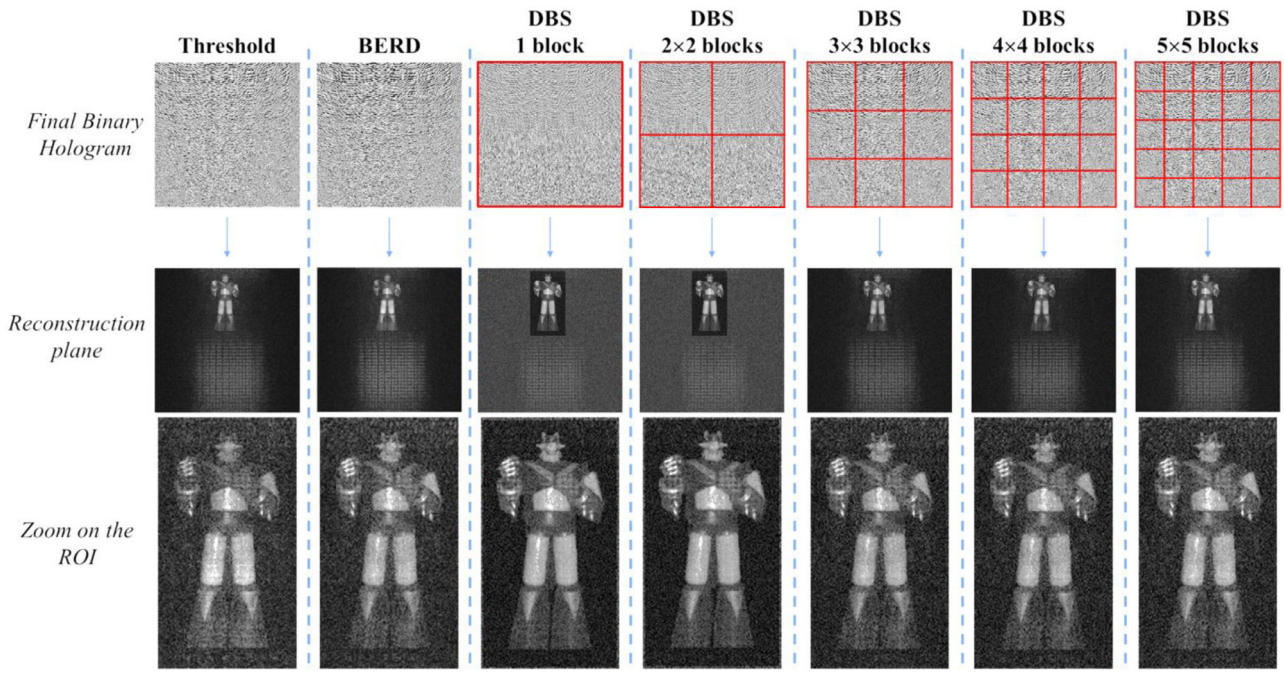

Fig. 7. Comparison of results obtained using the threshold method, BERD, and DBS algorithm using various block-partitioning strategies.

GPU parallel programming platform. The performance test was run on an AMD Ryzen 7 1700X CPU with 64GB of RAM that was equipped with an NVIDIA GeForce GTX1080 VGA graphics card. The comparison of the different methods is illustrated in Fig. 7. The conventional DBS algorithm yields a better result compared to the threshold method and BERD. In the case of the proposed method, although the computational time per block decreased, the reconstructed images showed only little degradation of quality as the number of blocks increased. In addition, as the number of blocks increases, the quality of the results becomes similar to that offered by the threshold method and BERD.

With the proposed strategy, the fact that the first pixels to be examined neighbor randomly distributed pixels is irrelevant. Moreover, the final result is obtained after a single iteration, and optimization of each block occurs independently of the other blocks. This strategy, as mentioned previously, enables parallel processing; however, the fact remains that blocks are not optimized as one all-encompassing pattern. Furthermore, the quality of the final pattern was observed to decrease as the block size decreased. Since the performance of the DBS algorithm relies on the optimization of each pixel, which is realized by taking into account the surrounding binary pattern, the performance of the block-partition strategy is limited by the relatively smaller number of binary pixels that are optimized simultaneously. We actually observed that the case in which a block was as small as a single pixel yielded a result equivalent to that offered by the threshold method. We can therefore describe our method as a bridge between the threshold method, in which every pixel value is determined based on its own contribution, and the conventional DBS method, in which the contribution of all pixels is taken into account.

Therefore, there must be a trade-off between the size of the block and the quality of the final result. The advantage of using small blocks is that computation is much faster because there are fewer pixels in each block. Since the optimization of each block can be performed in parallel by utilizing platforms such as those offering cloud computing, the time to compute the final pattern is determined by the computational time for a single block. To evaluate the performance of the proposed method, the MSE, peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) were employed. The MSE and PSNR are pixel difference-based metrics that are simple to calculate and mathematically facile. They take into account the complex amplitude and therefore the preservation of depth information. However, these metrics do not consider the way humans perceive images. Humans usually evaluate image quality based on the structural information rather than pixel intensities [28]. SSIM measures the similarity between two images by using information about luminosity, contrast, and structure [29]. The computational times, MSE, PSNR, and SSIM for the threshold method, BERD, and the proposed method with various block sizes are presented in Table 2. A lower MSE value and higher SSIM and PSNR values yielded better image quality.

The threshold method and BERD showed a much faster computational time (less than 1 s) compared to that of the conventional DBS method. However, they exhibited much worse MSE, PSNR, and SSIM

Table 2
Computational time, MSE, PSNR, and SSIM associated with the threshold method, BERD, and DBS algorithm for various block sizes.

Method	Threshold	BERD	DBS				
Number of pixels per block	512 × 512	512 × 512	512 × 512	256 × 256	170 × 170	128 × 128	102 × 102
Number of blocks in parallel	1 block	1 block	1 block	2×2 = 4 blocks	3×3 = 9 blocks	4×4 = 16 blocks	5×5 = 25 blocks
Computational time per block	~**0.58 s**	~0.78 s	~452 s	~112 s	~51 s	~29 s	~19 s
MSE	5.976×10^3	4.275×10^3	$\mathbf{1.609 \times 10^3}$	1.657×10^3	3.317×10^3	3.340×10^3	3.688×10^3
PSNR	20.366	21.821	**26.066**	25.937	22.923	22.892	22.463
SSIM	0.245	0.272	**0.390**	0.385	0.299	0.297	0.289

The values in bold highlight the best results.

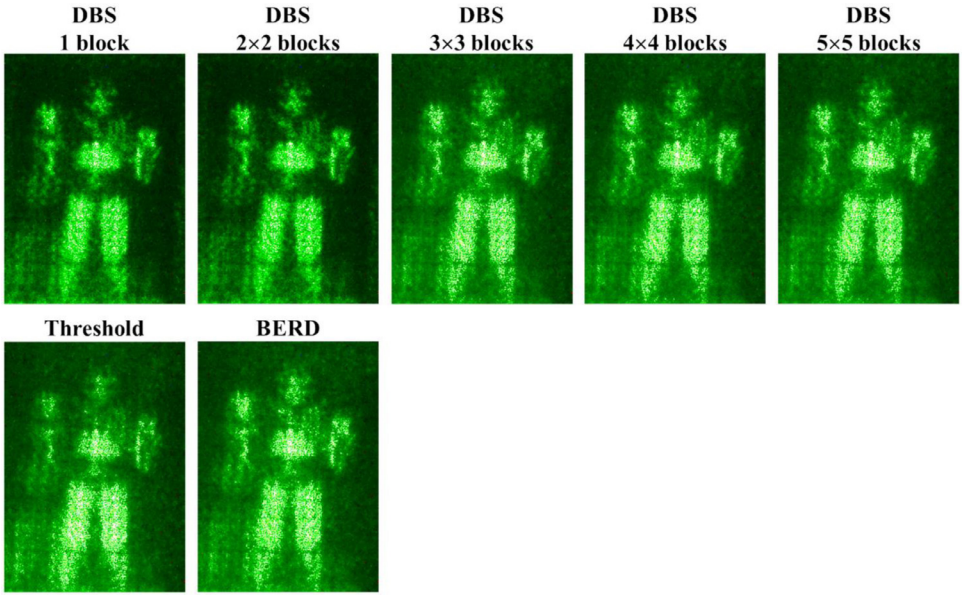

Fig. 8. Optical reconstructions of binary holograms obtained using the proposed DBS algorithm employing different block-partitioning strategies, the threshold method, and BERD.

values than those of the conventional DBS method. The conventional DBS method required one iteration to obtain a satisfying result, but the computational time exceeded 7 min. The block-partition strategy that we propose allows binary holograms to be generated during a single iteration, and has a computational time that ranges from approximately 19 s to less than 112 s, depending on the block size. However, an increase in the number of blocks results in a higher MSE and lower PSNR and SSIM, though it stays better than the BERD and threshold methods.

4. Optical reconstruction

Fig. 8 shows the optical reconstructions of the binary holograms generated using the proposed DBS algorithm with different block-partitioning strategies, the threshold method, and BERD. The resulting holograms are Fresnel holograms that were computed to have a reconstruction distance of 245 mm. The reconstruction was performed by using a laser emitting at 532 nm (CNI, MGL-III-532), and the SLM used to display the binary hologram was a 1366 × 768-pixel LCoS display with a pitch of 7 μm (CREMOTECH, C200).

The optical reconstruction achieved by the conventional DBS algorithm implementing a single block showed better quality and lower noise than those produced by the threshold method and BERD. As was observed in the simulation results, comparison of the optical reconstructions revealed a slow degradation of the image quality that was coincident with block size reduction. However, although the degradation was only barely perceptible in the case of the 2 × 2 block-partitioning strategy, as detailed in Table 2, the computational time was four times less than that required for the conventional single-block strategy. The proposed DBS algorithm with a larger number of blocks yields more noisy pixels in the reconstructed image. Noisy pixels could increase the brightness of the reconstructed image, which may cause poor contrast. MSE, PSNR, and SSIM were computed for the evaluation of the optical reconstruction. A reference image is necessary to compute the MSE, PSNR, and SSIM. However, in optical reconstruction, there is no reference image. Therefore, we used a black image as the reference image, and MSE, PSNR, and SSIM were computed on the black background of the reconstructed images. The intention is to give an indicator of the optical quality of the different reconstructed images by quantifying the noise level in a uniform dark region. The MSE, PSNR, and SSIM were computed only for a given ROI, illustrated in Fig. 9. The size of the ROI was equal to the size of the reference image. The MSE, PSNR, and SSIM of the optically reconstructed images are presented in Table 3.

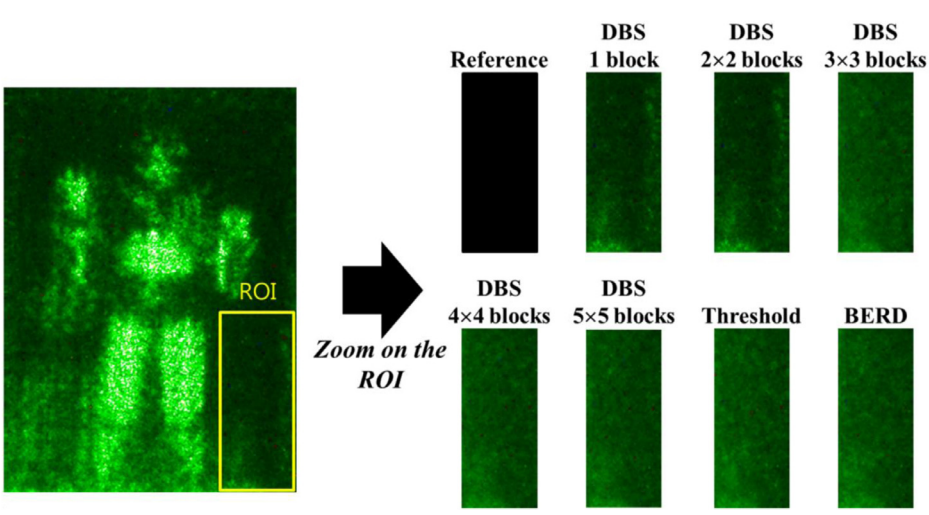

Fig. 9. Region of interest (ROI) of optically reconstructed images for computing MSE, PSNR, and SSIM.

Table 3
MSE, PSNR, and SSIM of the optically reconstructed images obtained using different algorithms.

	Threshold	BERD	DBS 1 block	DBS 2×2 block	DBS 3×3 blocks	DBS 4×4 blocks	DBS 5×5 blocks
MSE	6.457×10^3	6.371×10^3	$\mathbf{3.495 \times 10^3}$	3.533×10^3	5.795×10^3	5.820×10^3	5.873×10^3
PSNR	10.030	10.089	**12.697**	12.649	10.500	10.482	10.443
SSIM	7.064×10^{-4}	7.602×10^{-4}	$\mathbf{16.881 \times 10^{-4}}$	16.660×10^{-4}	8.446×10^{-4}	8.183×10^{-4}	8.133×10^{-4}

The values in bold highlight the best results.

5. Conclusions

We have demonstrated a method to significantly reduce the computational time associated with the DBS algorithm. The limitation of the proposed strategy is a trade-off between the computational time and the quality of the final hologram; however, it should be noted that simply randomizing the pixel examination order during the DBS procedure reduces the computational time by 67% even without parallel computation. As the computational time represents a major disadvantage in the DBS algorithm, a simple threshold method is typically used to generate binary holograms. However, we expect that our proposed method can make it easier to generate high-quality binary holograms in less time than is required for the conventional DBS operation.

Declaration of Competing Interest

The authors declare no competing financial interest.

Acknowledgments

This research was supported by the Ministry of Culture, Sports and Tourism (MCST) and the Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2017 (R2017060005, Development of AR Platform based on Hologram).

References

[1] Poon TC, Wu MH, Shinoda K, Suzuki Y. Optical scanning holography. Proc IEEE 1996;84:753–64.
[2] Poon TC. On the fundamentals of optical scanning holography. Am J Phys 2008;76:738–45.
[3] Kim YS, Kim T, Woo SS, Kang H, Poon TC, Zhou CH. Speckle-free digital holographic recording of a diffusely reflecting object. Opt Express 2013;21:8183–9.
[4] Yamaguchi I, Zhang T. Phase-shifting digital holography. Opt Lett 1997;22:1268–70.
[5] Zhou WJ, Zhang HB, Yu YJ, Poon TC. Experiments on a simple setup for two-step quadrature phase-shifting holography. IEEE Trans Ind Inform 2016;12:1564–70.
[6] Tahara T, Kanno T, Arai Y, Ozawa T. Single-shot phase-shifting incoherent digital holography. J Opt 2017;19:065705–12.
[7] Rosen J, Brooker G. Fresnel incoherent correlation holography (FINCH): a different way of 3D imaging. Laser Focus World 2013;49:49–51.
[8] Rosen J, Kelner R, Kashter Y, Vijayakumar A. Recent advances in FINCH technology. In: IEEE International conference on industrial informatics (INDIN); 2016. p. 582–7.
[9] Rosen J, Kelner R. Three-dimensional imaging by self-reference single-channel digital incoherent holography. IEEE Trans Ind Inf 2016;12:1571–83.
[10] Chen JS, Chu DP, Smithwick Q. Rapid hologram generation utilizing layer-based approach and graphic rendering for realistic three-dimensional image reconstruction by angular tiling. J Electron Imaging 2014;23:023016.
[11] Chen RHY, Wilkinson TD. Computer generated hologram from point cloud using graphics processor. Appl Opt 2009;48:6841–50.
[12] Schnars U, Juptner WPO. Digital recording and numerical reconstruction of holograms. Measur Sci Technol 2002;13:R85–R101.
[13] Kim MK, Yu LF, Mann CJ. Interference techniques in digital holography. J Opt A 2006;8:S518–23.
[14] Birch P, Young R, Chatwin C, Farsari M, Budgett D, Richardson J. Fully complex optical modulation with an analogue ferroelectric liquid crystal spatial light modulator. Opt Commun 2000;175:347–52.
[15] Goorden SA, Bertolotti J, Mosk AP. Superpixel-based spatial amplitude and phase modulation using a digital micromirror device. Opt Express 2014;22:17999–8009.
[16] Chen Y, Fang ZX, Ren YX, Gong L, Lu RD. Generation and characterization of a perfect vortex beam with a large topological charge through a digital micromirror device. Appl Opt 2015;54:8030–5.
[17] Fang ZX, Zhao HZ, Chen Y, Lu RD, He LQ, Wang P. Accelerating polygon beam with peculiar features. Sci Rep 2018;8:8593–600.
[18] Mirhosseini M, Magana-Loaiza OS, Chen CC, Rodenburg B, Malik M, Boyd RW. Rapid generation of light beams carrying orbital angular momentum. Opt Express 2013;21:30196–203.
[19] Fang ZX, Ren YX, Gong L, Vaveliuk P, Chen Y, Lu RD. Shaping symmetric airy beam through binary amplitude modulation for ultralong needle focus. J Appl Phys 2015;118:203102–2013107.
[20] Fang ZX, Chen Y, Ren YX, Gong L, Lu RD, Zhang AQ, Zhao HZ, Wang P. Interplay between topological phase and self-acceleration in a vortex symmetric airy beam. Opt Express 2018;26:7324–35.
[21] Leportier T, Park MC, Kim T. Numerical alignment of spatial light modulators for complex modulation in holographic displays. J Disp Technol 2016;12:1000–7.
[22] Tudela R, Martin-Badosa E, Labastida I, Vallmitjana S, Juvells I, Carnicer A. Full complex fresnel holograms displayed on liquid crystal devices. J Opt A 2003;5:S189–94.
[23] Tsang PWM, Poon TC. Novel method for converting digital fresnel hologram to phase-only hologram based on bidirectional error diffusion. Opt Express 2013;21:23680–6.
[24] Seldowitz MA, Allebach JP, Sweeney DW. Synthesis of digital holograms by direct binary search. Appl Opt 1987;26:2788–98.
[25] Takaki Y, Yokouchi M, Okada N. Improvement of grayscale representation of the horizontally scanning holographic display. Opt Express 2010;18:24926–36.
[26] Jennison BK, Allebach JP, Sweeney DW. Efficient design of direct-binary-search computer-generated holograms. J Opt Soc Am A 1991;8:652–60.
[27] Chhetri BB, Yang SY, Shimomura T. Stochastic approach in the efficient design of the direct-binary-search algorithm for hologram synthesis. Appl Opt 2000;39:5956–5964.
[28] Al-Najjar YA, Soong DC. Comparison of image quality assessment: PSNR, HVS, SSIM, UIQI. Int J Sci Eng Res 2012;3:1–5.
[29] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: From error visibility to structural similarity. IEEE Trans Image Process 2004;13:600–12.