Optics & Laser Technology 85 (2016) 30–34

Contents lists available at ScienceDirect

Optics & Laser Technology

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

Full length article

Reducing the variability in random-phase initialized Gerchberg-Saxton Algorithm

Francisco Javier Salgado-Remacha

Departamento de Física Aplicada, Facultad de Ciencias, Universidad de Zaragoza, 50009 Zaragoza, Spain

A R T I C L E I N F O

Article history:
Received 18 January 2016
Received in revised form
29 April 2016
Accepted 31 May 2016
Available online 8 June 2016

Keywords:
Digital holography
Optical information
Gerchberg-Saxton Algorithm
Optical computing

A B S T R A C T

Gerchberg-Saxton Algorithm is a common tool for designing Computer Generated Holograms. There exist some standard functions for evaluating the quality of the final results. However, the use of randomized initial guess leads to different results, increasing the variability of the evaluation functions values. This fact is especially detrimental when the computing time is elevated. In this work, a new tool is presented, able to describe the fidelity of the results with a notably reduced variability after multiple attempts of the Gerchberg-Saxton Algorithm. This new tool results very helpful for topical fields such as 3D digital holography.

1. Introduction

The synthesis of Computer Generated Holograms (CGH) represents an active and topical field of research. Some of the applications of CGHs are placed within fields like data storage, optical data processing, testing or interferometry [1]. One of the most potential application of CGHs is the ability of shaping a diffraction pattern at a certain plane in the space. The decade of the 90's was specially productive due to the development of Personal Computers and Liquid Crystal Displays (LCDs) [2]. In the recent years, this application has become an increasing interest motivated, specially, by the display industry and 3D imaging systems, leading to an increment of the number of published works about this topic [3–10]. These fields, compared to planar optics, require an elevated number of pixels increasing notably the computation time.

The CGH design methods fall into two main groups: global optimization methods (i.e., genetic algorithms, simulated annealing or direct search) [11], and iterative design methods [4]. The iterative methods are based on the Gerchberg-Saxton Algorithm (GSA) [12,13]. In the following we will center on GSA, since it is the most used algorithm. This technique was firstly proposed for phase retrieval problems and, due to the use of a Fast-Fourier transform (FFT), it results computationally efficient. The original algorithm successively transforms between the spatial and spatial-frequency domains, and imposes the respective constraints. For displaying purposes, the spatial-frequencies and the spatial domains represent the input and output planes respectively. By transforming the input and output plane with constraints iteratively, the diffraction pattern of the CGH becomes closer to the output target. The algorithm ends after a certain number n of iterations. In this work (as in the original GSA) the constraint for the input plane forces the CGH to be a pure-phase element, whereas the output constraint is that the intensity pattern matches the desired target regardless the output phase. Fig. 1 summarizes the flow chart of the standard GSA.

There are many ways to start the GSA and, probably, the most popular is to begin with a randomized phase map. Although it is always interesting to allow some degree of freedom at the initial state, GSA is strongly dependent on the starting guess. For this reason, the correct procedure should be to perform successive attempts (m iterations in Fig. 1) of the GSA with random initial estimates in order to average the quality of the results [14]. In order to illustrate the performance of GSA, Fig. 2 collects six gray-scale 8-bits images used as sample test in this work. Running a Matlab implementation of GSA with $n = 100$ iterations we can obtain the reconstruction I_n for each target I_T. In addition, Fig. 2 also shows the numerical differences between each target and its reconstruction. As can be seen, although reconstructions and targets are visually very similar, there still exits some kind of difference. In Section 2 we will see some standard functions used for the evaluation of the reconstruction error.

The use of several averaging iterations can be very expensive in terms of computing time, specially in applications such as 3D data storage or volumetric beam shaping. In this work, a new tool for the evaluation of GSA is presented, reducing the dependence with the random initial guess. The study is concerned in this work with 2D digital holograms, but can be extended to 3D digital

E-mail address: fjsalgado@unizar.es

http://dx.doi.org/10.1016/j.optlastec.2016.05.021

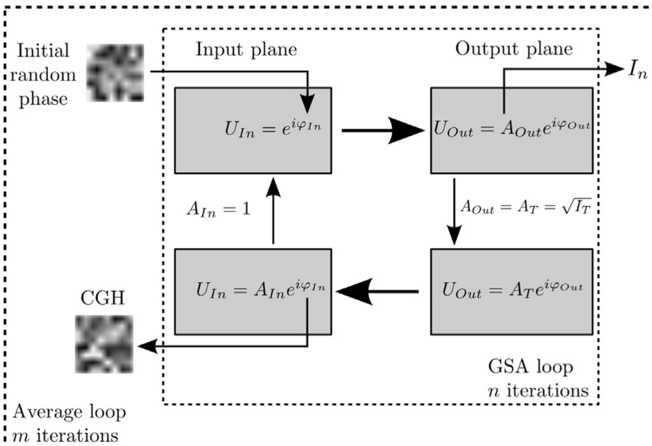

Fig. 1. Schematic representation of the GSA with random initial phase.

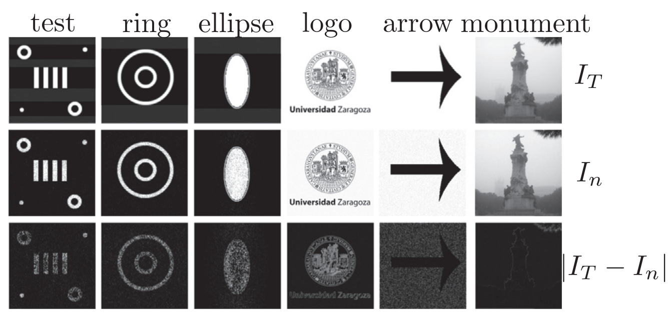

Fig. 2. Targets (up), examples of reconstructions (middle) and differences between reconstructions and targets (down) after $n=100$ iterations of GSA. The error values for each optimization, computed with different evaluation functions, are plotted in Figs. 6 and 7.

holography.

2. Common evaluation functions

The performance of the GSA (as well as other CGHs design algorithms) is commonly evaluated using different error functions. There exit a set of three image quality metrics widely applied in this field [1]. Probably the most usual function is the Root Mean Square Error (RMS). This function describes the fidelity of the output intensity compared to the desired target intensity I_T,

$$\mathrm{RMS} = \sqrt{\frac{\sum_{x,y} |\sqrt{I_n(x,y)} - \sqrt{I_T(x,y)}|^2}{\sum_{x,y} I_T(x,y)}} \tag{1}$$

where x and y denote the coordinates at the reconstruction plane, I_n is the intensity distribution produced by GSA after n iterations and I_T is the target intensity distribution. Thus, RMS quantifies the similarity between I_n and I_T. Another useful error function is the efficiency, which can be defined as the amount of intensity within a region of interest (ROI) divided by the total amount of intensity. Commonly, the ROI is defined using I_T as a mask [1], resulting,

$$\mathrm{efficiency} = \frac{\sum_{x,y} \sqrt{I_n(x,y) \cdot I_T(x,y)}}{\sum_{x,y} I_T(x,y)}. \tag{2}$$

It should be noticed that this definition of the efficiency function differs from the diffraction efficiency function, and it is designed for image formation rather than for beam shaping contexts. Based on similar concepts, the Signal-to-Noise Ratio (SNR) computes the ratio between light intensity in a Signal window and the amount of intensity in a Noise window. The Signal window mask can be defined using I_T, whereas the negative image of I_T constitutes the Noise window [15]. Thus,

$$\mathrm{SNR} = \frac{\sum_{x,y} \sqrt{I_n(x,y) \cdot I_T(x,y)}}{\sum_{x,y} \sqrt{I_n(x,y) \cdot I_{NT}(x,y)}}, \tag{3}$$

where I_{NT} denotes the negative image of I_T. Although this definition is specially designed for binary images, it also works with gray-scale images. Note that Eqs. 1 and 2 take values between 0 and 1, whereas Eq. (3) can take any positive value. For the ideal case $I_n = I_T$, Eq. (1) is minimum and the value of the efficiency and SNR are maxima. Thus, in a GSA loop, RMS follows a decreasing function and the efficiency and SNR are increasing functions.

As example, we perform an optimization using the target "ring" and $n=100$ GSA loops. If we repeat this procedure $m=200$ times, we obtain 200 optimized solutions. Using the terms presented in Fig. 1, we have $m=200$ values of RMS for any of the $n=100$ GSA loops. Fig. 3(a) shows the mean RMS averaged over the $m=200$ trials at any of the $n=100$ iterations, using the target "ring". Fig. 3 (b) shows a histogram with the resulting 200 optimized solutions. With these values it is possible to calculate the variance using the standard deviation, defined as

$$\Delta\epsilon = \sqrt{\frac{\sum_m |\epsilon_m - \bar{\epsilon}|^2}{m}}, \tag{4}$$

being ϵ_m the resulting error value for each averaging iteration, and $\bar{\epsilon}$ is the mean error,

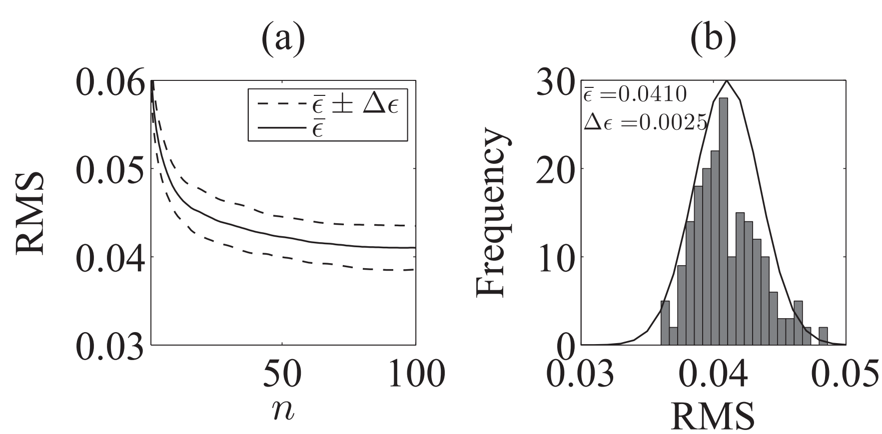

Fig. 3. RMS error function, averaged over $m=200$ and $n=100$ iterations of GSA, using the image "ring" as target; (a) evolution of the mean RMS (solid line). Dotted lines marks the interval where we can find the 68.3% of the trials; (b) histogram of the final RMS values after $m=200$ averaging loops, and values of the Gaussian distribution.