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)

Abstract

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.

Keywords: Digital holography, Optical Information, Gerchberg-Saxton Algorithm, Optical Computing

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, 4, 5, 6, 7, 8, 9, 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.

*Corresponding author

Email address: fjsalgado@unizar.es (Francisco Javier Salgado-Remacha)

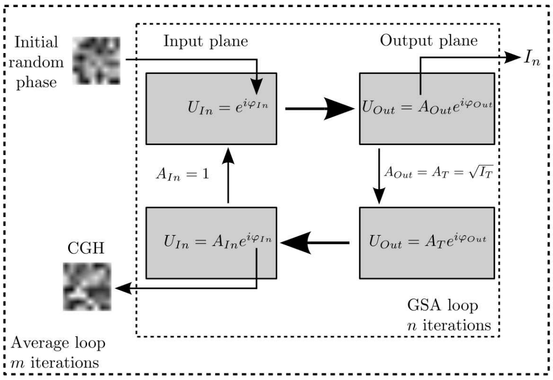
Initial random phase
Input plane
Output plane
I_n
$U_{In} = e^{i\varphi_{In}}$
$U_{Out} = A_{Out} e^{i\varphi_{Out}}$
$A_{In} = 1$
$A_{Out} = A_T = \sqrt{I_T}$
CGH
$U_{In} = A_{In} e^{i\varphi_{In}}$
$U_{Out} = A_T e^{i\varphi_{Out}}$
GSA loop
n iterations
Average loop
m iterations

Figure 1: Schematic representation of the GSA with random initial phase.

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 grey-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.

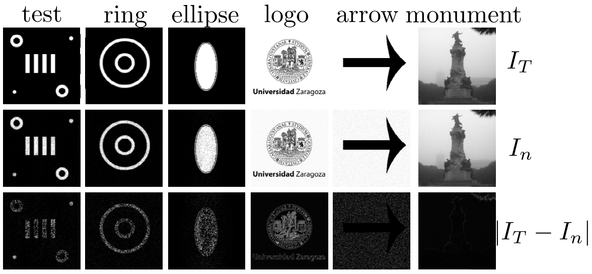

Figure 2: Targets (up), examples of reconstructions (middle) and differences between reconstructions and targets (down) after n=100 iterations of GSA. The error values for each optimization, computed with different evaluation functions, are plotted in Fig. 6 and 7..

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 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,

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

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,

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

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,

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

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

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

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

$$\overline{\epsilon} = \frac{\sum_m (\epsilon_m - \overline{\epsilon})}{m}. \tag{5}$$

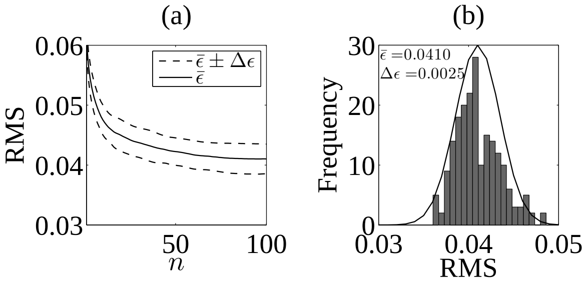

Figure 3: RMS error function, averaged over $m = 200$ and $n = 100$ iterations of GSA, using the image "ring" as target; (a) evolution of the mean RMS (solid line). Dotted lines marks the interval where we can find the 68.3% of the trials; (b) histogram of the final RMS values after $m = 200$ averaging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

Thus, the 68.3% of the trials fals in the interval $\overline{\epsilon} \pm \Delta\epsilon$. Dotted lines in Fig. 3(a) marks this interval. Note that Eqs. 4 and 5 can be applied not only for the ending solution, but also for any number of iteration n. In addition, they can be computed using any other error function. Thus, Fig. 4 shows the variance in the efficiency and SNR error function after $m = 200$ averaging loops with $n = 100$ iterations. Now we can deduce some facts. First of all, RMS and efficiency functions are strongly dependent on the random initial phase. Both functions present an elevated variability after several averaging attempts. On the contrary, the Signal-to-Noise Ratio present a reduced variability compared with the two other functions. However, RMS and efficiency are defined between 0 and 1, whereas SNR reach different maximum value depending on the target image. This fact results very important in order to compare the performance using different targets, limiting the applicability of SNR.

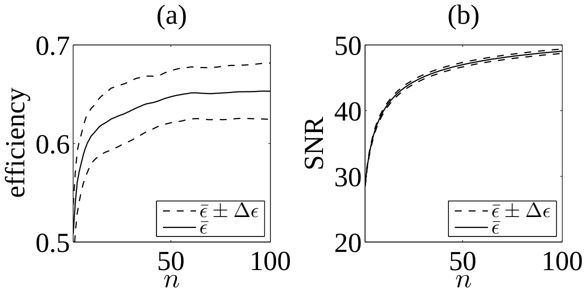

Figure 4: Mean value and variance of the error function, averaged over $m = 200$ and $n = 100$ iterations of GSA, using the image "ring" as target; (a) efficiency; (b) SNR.

3. Signal-to-Noise Contrast

At this point, we are able to define a new function combining the reduced variability of SNR within a normalized range of operation. First of all, we define a Signal window as the pixels in the target image with non-zero value. At the same time, the zero-valued pixels in the target image constitute the Noise window. In other words, we will consider that the black pixels in the target should not receive light at the output plane. Consequently, this amount of light at the Noise window is computed as noise. Thus,

$$I_{Signal}(x, y) = \begin{cases} I_n(x_i, y_i), & \text{if } I_T(x_i, y_i) > 0 \\ 0, & \text{if } I_T(x_i, y_i) = 0 \end{cases} \tag{6}$$

and

$$I_{Noise}(x, y) = \begin{cases} I_n(x_i, y_i), & \text{if } I_T(x_i, y_i) = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

In order to get a normalized function, we will use a new function computing the visibility of the amount of signal over the amount of noise, that is, the Signal-to-Noise Contrast (SNC). This new function is defined as,

$$\text{SNC} = \frac{|I_{Signal}(x, y) - I_{Noise}(x, y)|}{I_{Signal}(x, y) + I_{Noise}(x, y)}. \tag{8}$$

Thus, if I_n coincides exactly with I_T, SNC reaches a maximum value of 1. On the contrary, If $I_n = I_{NT}$, the value of SNC is 0. The performance of Eq. 8 is shown Fig. 5 for GSA with $n = 100$, $m = 200$ and the image "ring" as target. We can appreciate that SNC follows an increasing curve as GSA runs. At the same time, it is also noticed that the variability of the results has been notably reduced, comparing with Figs. 3 and 4. Specifically, SNC reduces more than 20 times the variability comparing with the results of Eq. 1 (RMS) shown in Fig. 3.

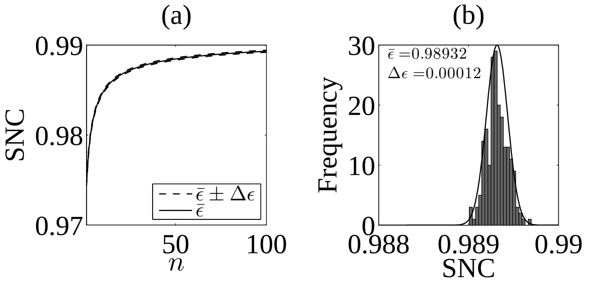

Figure 5: Performance of SNC error function, averaged over $m = 200$ and $n = 100$ iterations of GSA, using the image "ring" as target; (a) evolution of the mean SNC (solid line) and its standard deviation (dashed lines) with a confidence interval of 68.3%; (b) histogram of the final SNC values after $m = 200$ averaging loops, and values of the Gaussian distribution.

This definition of Noise window (Eq. 7) is, to some extent, arbitrary. First of all, we should note that Eq. 7 requires the existence of black pixels in the target. Althoug the definitions in 2 and 3 (efficiency and SNR) also have this problem, this requirement can be easily solved using the minimum value of intensity in the target, instead of $I = 0$. We only need a reference window where we can compute the amount of noise. However, since the most usual applications of CGHs looks for addressing the light into some particular directions (and no others), the definition of Noise window in Eq. 7 results more convenient for our interests. Second, Eq. 8 depends strongly on the size of the Noise window. For this reason, targets images collected in Fig. 2 have been carefully selected. Images "test", "ring" and "ellipse" are almost binary images (there are some few grey gray pixels), with more amount of black pixels. In other words, the Noise windows for these images, following Eq. 7, are bigger than the Signal window. On the contrary, images "logo" and "arrow" show more amount of white pixels. Finally, the image "monument" results very interesting for this work. It was taken with a dense fog and, consequently, it is a gray-scale image with very low number of 0-valued pixels.

4. Evaluation

We are ready now to perform a comparison between SNC and the other three error functions. In a general way, it is always interesting to allow some dynamic range during the optimization and, at the same time, reduce the variability depending on the random-initialized guess. We center firstly on the dynamic range, defined as,

$$\text{Range} = \epsilon_n - \epsilon_0, \tag{9}$$

being ϵ_n the final error value after n iterations and ϵ_0 the initial error value. The performance of each error function has been evaluated using the six image targets of Fig. 2 with $n = 100$ and averaging over $m = 200$. Fig 6 summarizes the mean Range value for each target image, as well as the total mean Range for each error method (in this figure, the SNR has been normalized to the maximum value reached). Although the efficiency and NSR show a better performance in terms of Range value, the Range of SNC and RMS are similar. This Range values are enough for performance evaluation.

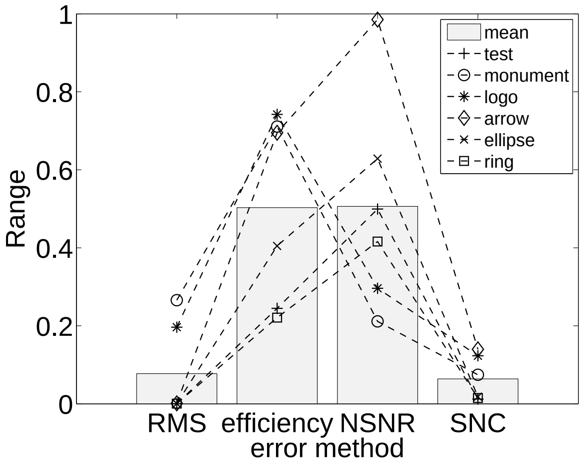

Figure 6: Range value for the six target images obtained with the studied error methods (Root Mean Square, Efficiency, Normalized Signal-to-Noise Ratio and Signal-to-Noise Contrast), with $n = 100$ and $m = 200$. The bars point out the mean Range for each error method.

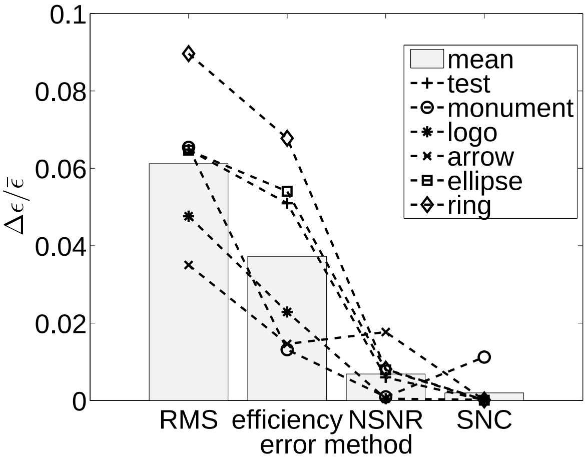

Figure 7: Relative variance of the error function for the six target images, obtained with the studied error methods (Root Mean Square, Efficiency, Normalized Signal-to-Noise Ratio and Signal-to-Noise Contrast), with $n = 100$ and $m = 200$. The bars point out the mean error value for each error method.

Our next performance text evaluates the variability among averaging iterations. Thus, we repeat the same analysis but using Eq. 4. The variance is then divided by the mean error, obtaining a relative variance. The results are collected in Fig. 7.

As can be seen, SNC shows the best performance, in terms of variability, among the four error methods. We can also note that, although the variability of SNC is notably reduced, it depends on the image target. Thus, the maximum SNC value has been obtained with the "arrow" target. This image, in fact, has a low number of zero-valued pixels. As a consequence, a low performance of SNC was expected. In spite of this fact, the variability of SNC with the "arrow" target is lower that the variability of RMS with any target.

Table 1: **Summary of Error Functions Performances**

Error method	Mean Range	Mean variance
RMS	0.0773	0.06120
efficiency	0.5030	0.03720
NSNR	0.5060	0.00685
SNC	0.0637	0.00197

For a better understanding, the mean relative variances of each error function have been collected in Table 1. The difference is specially outstanding compared with RMS. In this case, the relative variance of SNC function is up to 30 times lower. Thus, the results show that the use of SNC, instead of other error metrics, allows to reduce the number of averaging iterations in order to evaluate a solution of GSA, which is very beneficial in order to test variations over GSA particularly when the computing time becomes significant.

5. Conclusions

For summarizing, it is clear that the use of GSA with random initial guess produces a certain amount of variability in the results. Thus, it is needed to average the results over multiple attempts, which reduces the efficiency in terms of computation time. This fact becomes significant when the number of pixels of the target is elevated, as is the case of 3D Digital Holography, for example. For this reason, a new error metric has been introduced. This function, called Signal-to-Noise Contrast, is specially designed for minimizing the dependence with the initial random state in a GSA loop. Thus, the final error value depends principally on the image target, and not so much on the initial guess. The performance of this function has been tested using six targets with different characteristics. A comparison with three error functions (Root Mean Square, Efficiency and Signal-to-Noise Ratio) has been also carried out, showing that SNC allows enough dynamic Range among GSA iteration whereas reduces significantly the variance of the final results. This new error function is specially indicated for the evaluation of variations and improvements upon GSA with random-initialized states and elevated number of pixels.

References

[1] B. C. Kress, P. Meyrueis, Applied Digital Optics: from micro-optics to nanophotonics, John Wiley and Sons, Chichester, West Sussex, UK, 2009.

[2] O. Ripoll, V. Kettunen, H. P. Herzig, Review of iterative fourier-transform algorithms for beam shaping applications, Opt. Eng. 43 (11) (2004) 2549–2556. doi:10.1117/1.1804543.

[3] C.-Y. Chen, Q.-L. Deng, P.-J. Wu, B.-S. Lin, H. T. Chang, H.-E. Hwang, G.-S. Huang, Speckle reduction by combination of digital filter and optical suppression in a modified gerchberg-saxton algorithm computer-generated hologram, Appl. Opt. 53 (27) (2014) 163–168. doi:10.1364/AO.53.00G163.

[4] C. Guo, S. Liu, J. T. Sheridan, Iterative phase retrieval algorithms. i: optimization, Appl. Opt. 54 (15) (2015) 4698–4708. doi:10.1364/AO.54.004698.

[5] C. Guo, J. Tan, Z. Liu, Precision influence of a phase retrieval algorithm in fractional fourier domains from

position measurement error, Appl. Opt. 54 (22) (2015) 6940–6947. doi:10.1364/AO.54.006940.

[6] P. Netrapalli, P. Jain, S. Sanghavi, Phase retrieval using alternating minimization, IEEE Trans. Signal Process. 63 (18) (2015) 4814–4826. doi:10.1109/TSP.2015.2448516.

[7] P.-J. Wu, C.-Y. Chen, Q.-L. Deng, H. T. Chang, B.-S. Lin, G.-S. Huang, Integral computer-generated hologram via a modified gerchberg-saxton algorithm, J. Opt. 17 (1) (2015) 015701.

[8] T. Shimobaba, T. Kakue, Y. Endo, R. Hirayama, D. Hiyama, S. Hasegawa, Y. Nagahama, M. Sano, M. Oikawa, T. Sugie, T. Ito, Improvement of the image quality of random phase-free holography using an iterative method, Opt. Commun. 355 (2015) 596 – 601. doi:http://dx.doi.org/10.1016/j.optcom.2015.07.030.

[9] Y. Xu, Z. Ren, K. K. Y. Wong, K. Tsia, Overcoming the limitation of phase retrieval using gerchberg-saxton-like algorithm in optical fiber time-stretch systems, Opt. Lett. 40 (15) (2015) 3595–3598. doi:10.1364/OL.40.003595.

[10] Z. Liu, C. Guo, J. Tan, Q. Wu, L. Pan, S. Liu, Iterative phase-amplitude retrieval with multiple intensity images at output plane of gyrator transforms, Journal of Optics 17 (2) (2015) 025701.

[11] M. A. Seldowitz, J. P. Allebach, D. W. Sweeney, Synthesis of digital holograms by direct binary search, Appl. Opt. 26 (14) (1987) 2788–2798. doi:10.1364/AO.26.002788.

[12] R. Gerchberg, W. Saxton, Practical algorithm for the determination of phase from image and diffraction plane pictures., Optik 35 (2) (1972) 237–250.

[13] J. R. Fienup, Phase retrieval algorithms: a comparison, Appl. Opt. 21 (15) (1982) 2758–2769. doi:10.1364/AO.21.002758.

[14] A. Weiss, J. Picard, Maximum-likelihood position estimation of network nodes using range measurements, Signal Processing 2 (4) (2008) 394–404. doi:10.1049/iet-spr:20070161.

[15] F. J. Salgado-Remacha, L. M. Sanchez-Brea, E. Bernabeu, Micromachining of diffractive optical elements embedded in bulk fused silica by nanosecond pulses, J. Lightwave Technol. 29 (6) (2011) 850–855. doi:10.1109/JLT.2011.2104939.