

Article

Phase Retrieval of One-Dimensional Objects by the Multiple-Plane Gerchberg–Saxton Algorithm Implemented into a Digital Signal Processor

Varis Karitans ^{1,2,*}, Maris Ozolinsh ^{1,2}  and Sergejs Fomins ^{1,2}

¹ Institute of Solid State Physics, University of Latvia, LV-1063 Riga, Latvia; maris.ozolins@lu.lv (M.O.); sergejs.fomins@cfi.lu.lv (S.F.)

² Department of Optometry and Vision Science, Faculty of Science and Technology, University of Latvia, LV-1004 Riga, Latvia

* Correspondence: varis.karitans@cfi.lu.lv

Abstract: In the current study, we address the phase retrieval of one-dimensional phase objects from near-field diffraction patterns using the multiple-plane Gerchberg–Saxton algorithm, which is still widely used for phase retrieval. The algorithm was implemented in a low-cost digital signal processor capable of fast Fourier transform using Q15 arithmetic, which is used by the previously mentioned algorithm. We demonstrate similarity between one-dimensional phase objects, i.e., vectors cut out of a phase map of the tertiary spherical aberration retrieved by the multiple-plane Gerchberg–Saxton algorithm, and these vectors are measured with a non-contact profiler. The tertiary spherical aberration was induced by a phase plate fabricated using grayscale lithography. After subtracting the vectors retrieved by the algorithm from those measured with the profiler, the root mean square error decreased, while a corresponding increase in the Strehl ratio was observed. A single vector of size 64 pixels was retrieved in about 2 min. The results suggest that digital signal processors that are capable of one-dimensional FFT and fixed-point arithmetic in Q15 format can successfully retrieve the phase of one-dimensional objects, and they can be used for applications that do not require real-time operation, i.e., analyzing the quality of cylindrical micro-optics.



Citation: Karitans, V.; Ozolinsh, M.; Fomins, S. Phase Retrieval of One-Dimensional Objects by the Multiple-Plane Gerchberg–Saxton Algorithm Implemented into a Digital Signal Processor. *Optics* **2024**, *5*, 514–522. <https://doi.org/10.3390/opt5040038>

Academic Editor: Thomas Seeger

Received: 6 October 2024

Revised: 8 November 2024

Accepted: 13 November 2024

Published: 20 November 2024

Keywords: phase retrieval; aberrations; adaptive optics; diffraction

1. Introduction

Aberrometry plays a crucial role in characterizing the quality of optical systems and their performance. Aberrometry is a very essential part of adaptive optics; it uses either deformable mirrors or spatial light modulators to compensate for wavefront distortions. Adaptive optics is widely used in astronomy [1], biology [2], vision science [3], optical communications [4] and other fields of science. There are many techniques used to measure wavefront distortions, and a good review of these methods can be found in a review article by Hampson [5]. The most popular wavefront sensor remains the Shack–Hartmann wavefront sensor, while curvature and pyramid sensors are also widely used.

In recent years, phase retrieval based on coherent diffractive imaging has become a very valuable tool in many fields of science, including crystallography [6], astronomy [7], holography [8], biology [9], and many others. It is especially valuable in computational optics to retrieve wavefront aberrations [10]. Phase retrieval algorithms can be easily implemented into real systems; they are quite immune to noise and they can retrieve the object from sparse data. The first successful algorithm based on alternating projections was developed by Gerchberg and Saxton [10,11], and it is still widely used. Shechtman et al. also discuss many algorithms developed thereafter, including those based on semidefinite programming, e.g., PhaseLift and PhaseCut, and the sparsity of the object [10].



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Today, many phase retrieval algorithms are implemented into embedded systems [12–15]. Such systems enable high computational efficiency and are also very compact and suited for practical applications. In this paper, we demonstrate the applicability of low-cost digital signal processors for the phase retrieval of one-dimensional wavefront distortions using the multiple-plane Gerchberg–Saxton (MPGS) algorithm [16]. Multiple-plane algorithms have advantages compared to single-plane algorithms in terms of accuracy and numerical efficiency [17]. One-dimensional phase retrieval in optics may be important, for example, when inspecting the optical quality of cylindrical optical elements. Due to memory limitations and execution speed, these digital signal processors are not suited for two-dimensional objects and phase retrieval in real-time; however, if the retrieval time is not a critical issue, such low-cost digital signal processors allow us to achieve a good accuracy of phase retrieval in a time in the order of minutes.

2. Materials and Methods

2.1. MPGS Algorithm

The goal of the algorithm is to retrieve the phase $\varphi_0[n]$ of a one-dimensional object $x_0[n]$ at the central plane P_0 , having one and the same number of lateral planes on each side. The initial guess of the phase of the object $\varphi_0[n]$ is an object with the same phase at every point. In this study, the objects to be retrieved are one-dimensional distorted wavefronts, and it is possible to propagate them from one plane to another using the free-space transfer function H , thereby iteratively updating their phase. The total number of planes, including the central plane P_0 , is denoted by r , while the number of lateral planes on one side of the central plane P_0 is $\frac{r-1}{2}$. Neighboring planes are separated by distance Δz ; however, the distance of propagation is denoted by the symbol z . The distance separating the central plane P_0 and the last plane is equal to $\frac{(r-1) \cdot \Delta z}{2}$. At each plane, only information about the measured intensity I is available. The index z shows that the intensity is measured at the plane located at distance z from the central plane. The propagation is denoted by the free-space transfer function H_z , where z is the distance of propagation. It must be taken into account that the object has to be propagated in free space, i.e., empty bands have to be placed around the object.

Next, one iteration of the MPGS algorithm is described. First, the central object with an intensity I_0 that is located at the plane P_0 is propagated to the plane P_z , where its phase is preserved while the calculated intensity is replaced by the measured intensity I_z . The new object is propagated back to the central plane P_0 , and again the phase is preserved, while the calculated intensity is replaced with the measured one. The same procedure is repeated for all other planes on the same side of the central plane until the last plane. The value of the counter cnt is increased after the central object has been propagated to any lateral plane and back to the central plane P_0 . As there are $\frac{r-1}{2}$ planes on one side of the central plane P_0 , the counter cnt reaches a value $\frac{r-1}{2}$ on one side of the central plane P_0 . After the objects from all the planes located on the same side of the central plane P_0 have been propagated back to it, the average phase is calculated. Now, a new object with an average phase and intensity I_0 is formed at the central plane P_0 , and all the steps described previously are repeated for the planes that are located on the other side of the central plane P_0 . After all the objects have been propagated back to the central plane P_0 , the counter cnt has reached a value $r - 1$, and one iteration has been completed. At the end of one iteration, the counter cnt is set to zero. After all the iterations have been completed, i.e., $iter$ has reached the value $iter_{max}$, the algorithm is stopped, and the phase of the retrieved central object is extracted in MATLAB. The Algorithm 1 is explained step-by-step below.

Algorithm 1 MPGS Algorithm

- (1) $iter = 0; cnt = 0; \varphi_0[n] = 0$
 - (2) $iter = iter + 1$
 - (3) $cnt = cnt + 1$
 - (4) $z = \begin{cases} cnt \cdot \Delta z, & \text{if } cnt \leq \frac{r-1}{2} \\ -(cnt - \frac{r-1}{2}) \cdot \Delta z, & \text{if } cnt > \frac{r-1}{2} \end{cases}$
 - (5) $x_0[n] = \sqrt{I_0[n]} \cdot e^{i \cdot \varphi_0[n]}$
 - (6) $X_0[k] = \mathfrak{F}(x_0[n]) = |X_0[k]| \cdot e^{i \cdot \theta_0[k]}$
 - (7) $X_0[k] = |X_0[k]| \cdot e^{i \cdot \theta_0[k]} \Rightarrow X_z[k] = |X_0[k]| \cdot H_z[k] \cdot e^{i \cdot \theta_0[k]} = |X_z[k]| \cdot e^{i \cdot \theta_z[k]}$
 - (8) $x_z[n] = \mathfrak{F}^{-1}(X_z[k]) = |x_z[n]| \cdot e^{i \cdot \varphi_z[n]}$
 - (9) $x_z[n] = |x_z[n]| \cdot e^{i \cdot \varphi_z[n]} \Rightarrow x'_z[n] = \sqrt{I_z[n]} \cdot e^{i \cdot \varphi_z[n]}$
 - (10) $X'_z[k] = \mathfrak{F}(x'_z[n]) = |X'_z[k]| \cdot e^{i \cdot \theta'_z[k]}$
 - (11) $X'_z[k] = |X'_z[k]| \cdot e^{i \cdot \theta'_z[k]} \Rightarrow X'_0[k] = |X'_z[k]| \cdot H_{-z}[k] \cdot e^{i \cdot \theta'_z[k]} = |X'_0[k]| \cdot e^{i \cdot \theta'_0[k]}$
 - (12) $x'_0[n] = \mathfrak{F}^{-1}(X'_0[k]) = |x'_0[n]| \cdot e^{i \cdot \varphi'_0[n]}$
 - (13) $x'_0[n] = |x'_0[n]| \cdot e^{i \cdot \varphi'_0[n]} \Rightarrow x'_0[n] = \sqrt{I_0[n]} \cdot e^{i \cdot \varphi'_0[n]}$
 - (14) $\varphi_{0,temp}[n] = \varphi_{0,temp}[n] + \frac{2 \cdot \varphi'_0[n]}{r-1}$
 - (15) $cnt = \frac{r-1}{2}$ or $cnt = r - 1$?
 - (16) If NO in step #15, then go to Step #3
 - (17) If YES in step #15, then go to Step #18
 - (18) $\varphi_0[n] = \varphi_{0,temp}[n]$
 - (19) $\varphi_{0,temp}[n] = 0$
 - (20) $x_0[n] = \sqrt{I_0[n]} \cdot e^{i \cdot \varphi_0[n]}$
 - (21) $cnt = r - 1$?
 - (22) If NO in Step #21, then go to Step #3
 - (23) If YES in Step #21, then go to Step #24
 - (24) $iter = iter_{max}$?
 - (25) If NO in step #24, then $cnt = 0$ and go to step #2.
 - (26) If YES in step #24, then STOP
-

2.2. Hardware

The MPGS algorithm was implemented into a digital signal processor (DSP)—sPIC30F6010A. This DSP can carry out one-dimensional FFT, thus making it suited for the phase retrieval of vectorial objects. The DSP was run at a frequency of 7.37 MHz, which was provided by the internal RC oscillator, and a $16 \times$ phase-locked loop was used to speed up the execution of the code. The code was developed in software (MPLAB IDE 8.92) and was compiled using compiler XC16. The DSP was programmed in MPLAB IDE 8.92 using an in-circuit debugger—MPLAB ICD 4. For communication between the DSP and MPLAB ICD 4, only five lines are required, i.e., Vdd, Vss, PGC, PGD, and Vpp/MCLR. Communication between the MPLAB ICD 4 and computer occurs via

USB cable. The basic circuit is shown in Figure 1. The value of the resistor R1 was 50 k Ω . The integrated circuit used in the image was not the DSP used in this study and is shown only for illustrative purposes. The FFT function required the input data to be provided in fixed-point Q15 format. When performing operations in Q15 format, overflow must be avoided by operating in regions away from the limiting values 0×8000 and $0 \times 7FFF$. Also, the FFT function requires that data not fall outside the range of -0.5 to 0.5 . This was always guaranteed by using the scaling factor so that the input data were within the range of -0.25 to $+0.25$. When compiling the code, the following parameters were specified: the size of the step Δz ; the size of the pixel p ; the total number of images r , and the number of iterations $iter_{max} = 200$. In the current study, the following parameters were used: $\Delta z = 0.5$ cm, $p = 7.5$ μm , $r = 5$, and $c = 200$. The data from a camera had to be converted to Q15 format. When the execution of the code was completed, the variable was exported as an ASCII file and was processed in MATLAB. The retrieved phase was wrapped within the range from $-\pi$ to $+\pi$ and had to be unwrapped. Due to memory limitations, the maximum size of the object together with the empty bands was 128 pixels (the size of the object was 64 pixels) as several arrays had to be used for the temporary storage of the data, occupying a significant part of data memory.

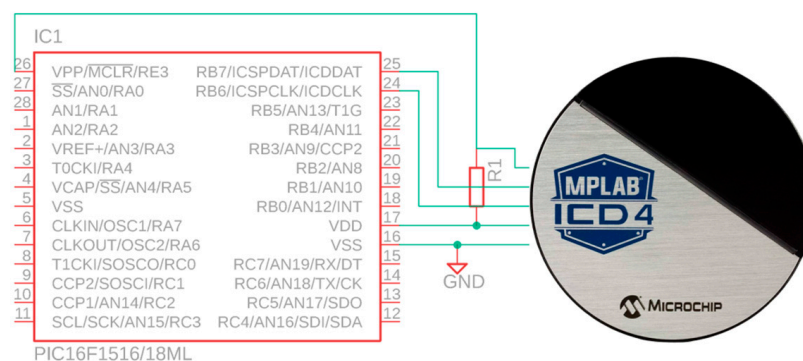


Figure 1. The basic circuit for programming a digital signal processor using MPLAB ICD 4. The resistance of the resistor is 50 k Ω . The actual digital signal processor used in the study was dsPIC30F6010A. Five lines are required for communication between MPLAB ICD 4 and the digital signal processor: V_{dd} , V_{ss} , $V_{pp}/MCLR$, PGC , and PGD .

2.3. Optical Setup

First, a two-dimensional phase object distorting a plane wavefront was fabricated using direct write lithography equipment $\mu\text{PG 101}$ (Heidelberg Instruments, Heidelberg, Germany). A square with a side equal to 480 μm was first etched in a copper layer. Next, a positive grayscale photoresist ma-P 1275G was applied over the etched area and was patterned to induce tertiary spherical aberration. One pixel (7.5 μm)-wide slit was fabricated in an identical way so that the one-dimensional objects to be retrieved could be cut out of the two-dimensional object. The two-dimensional object was measured using a non-contact surface profiler Zygo NewView 7100 (Middlefield, CT, USA), which is shown in the bottom right of Figure 2. The phase of the two-dimensional object is given in radians. The measurements with the non-contact profiler served as the ground truth against which the objects retrieved with the MPGS algorithm were compared.

The optical system is schematically shown in Figure 2. A beam coming from the laser is collimated by the fiber collimator C. Next, it is deviated by the mirror M1 followed by two aligned 4-f systems. A two-dimensional object was placed at the primary focal plane of the first lens L1 of the first 4-f system. A horizontally and vertically flipped copy of the original object was formed at the secondary focal plane of the second lens L2 of the first 4-f system. At this plane, a horizontally orientated slit was scanned vertically to cut out the vectors or one-dimensional phase objects to be retrieved. These objects were distorted one-dimensional wavefronts that were selected randomly. The size of the two-dimensional object was 64 by 64 pixels, and the 21st, 33rd, and the 55th (counting from the bottom)

vectors were selected as objects. As mentioned in Section 2.1, these one-dimensional wavefronts were the objects to be retrieved by the MPGS algorithm. The slit was followed by the mirror M2 that, in turn, was followed by the second 4-f system, which consisted of the lenses L3 and L4. A camera was moved around the secondary focal plane of the second lens L2 of the second 4-f system to capture the Fresnel diffraction patterns at all planes. The bit depth of the camera was 8 bits. Centroids were calculated from the diffraction patterns captured at the central plane P_0 . The data were leveled in software Gwyddion as there was tilt present in the retrieved vectors. The tilt was due to the shifts of centroids from one plane to another; meanwhile, during data processing, the coordinates of the centroid were fixed. During data processing, the centroid of the central image was always used.

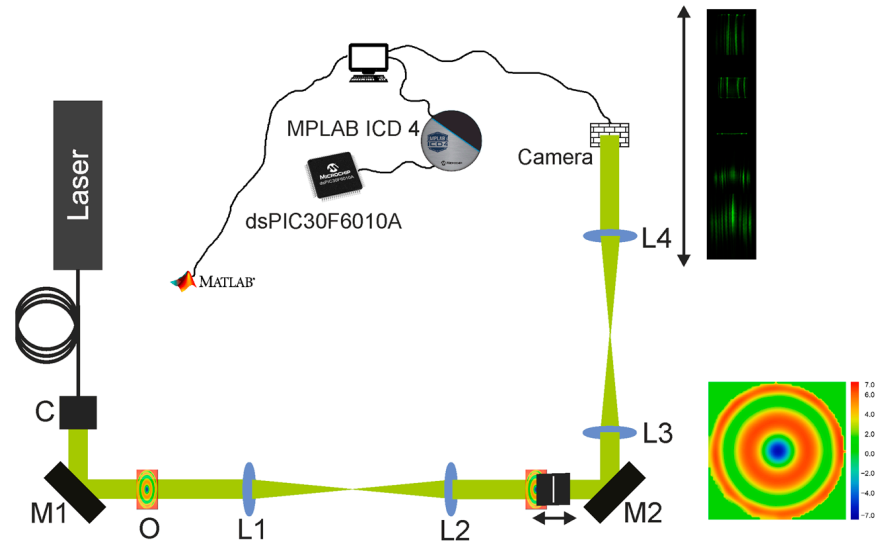


Figure 2. The experimental setup used in the study. L stands for lens; M stands for mirror. The two-dimensional object is shown in the bottom right, and the slit was used to cut out one-dimensional objects from it. These objects were then retrieved using the MPGS algorithm. The phase distortions are given in radians. Abbreviations: C—a fiber collimator, M—mirrors, L—lenses, O—the two-dimensional object.

2.4. Data Analysis

The root mean square error (*RMSE*) was calculated according to the following equation:

$$RMSE = \sqrt{\frac{\sum_{a=1}^n (\bar{x} - x_a)^2}{n}} \quad (1)$$

where \bar{x} is the mean phase of the object, x_a is the phase at an arbitrary point, and n is the length of the object.

The quality of the retrieved phase was estimated in terms of the *RMSE* of the residual phase, the corresponding Strehl ratio, and the peak-to-valley (*PV*) value. The residual phase was calculated as the difference between the phase of the ground truth and the retrieved objects. The Strehl ratio according to the Maréchal criterion was calculated using the following equation:

$$S \approx e^{-RMSE^2} \quad (2)$$

where *RMSE* is measured in radians.

3. Results

3.1. Diffraction Patterns

Figure 3 shows the Fresnel diffraction patterns of three vectors captured at all five planes. Planes change along the horizontal direction, while each row corresponds to a different vector. One can notice the asymmetric distribution of light intensity in the direction normal to the slit for the 55th vector. This indicates that the gradient of the phase was large, even across the width of the slit. When propagating away from the central plane, one can notice high frequencies beyond the highest DFT frequencies, extending into the free space around the object. The Fresnel number calculated for the most distant planes at the wavelength $\lambda = 520$ nm was $F = 11$, indicating that Fresnel diffraction patterns were observed and the method of angular spectrum for propagating the wave was valid [16].

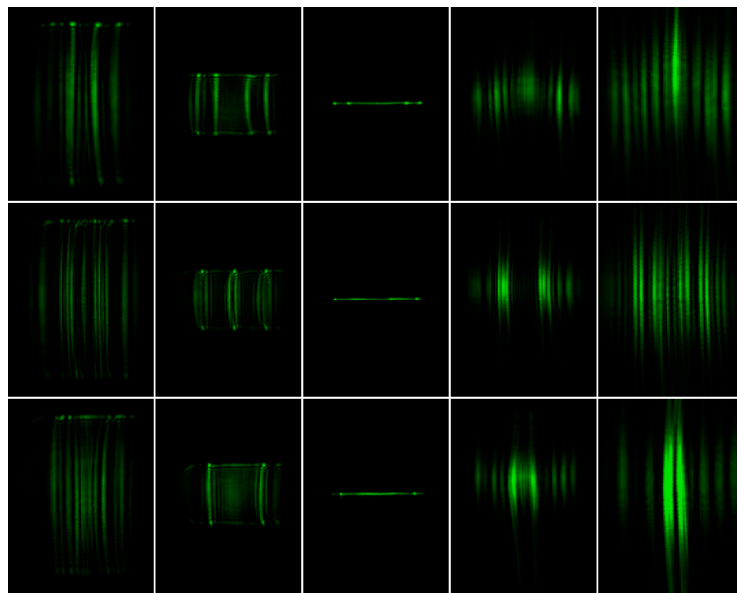


Figure 3. Diffraction patterns of the test objects, i.e., the 21st, the 33rd, and the 55th vectors (counting from the bottom). The vectors, i.e., the test objects, were cut out from the two-dimensional object via the slit. The planes change along the horizontal direction (from 1 to 5).

3.2. Phase Retrieval

The left column of Figure 4 shows the phase of the test objects measured with the non-contact profiler. These measurements serve as the ground truth. The mean plane was subtracted from the data to cancel out the prismatic effect. The phase is given in radians, while the units of the horizontal axis are pixels; one pixel is equal to $7.5 \mu\text{m}$. The *RMSE* values of the phase of the 21st, 33rd, and 55th (counting from below) vectors are 2.11 rad, 3.816 rad, and 2.504 rad, respectively. The corresponding *PV* values are 6.45 rad, 12.76 rad, and 7.45 rad. All Strehl ratios are practically zero.

The right column of Figure 4 shows the phase of the test objects retrieved with the MPGS algorithm. One iteration required about 2 s, and the total time needed for the retrieval of a single object was about 200 s. Like previously, the mean plane was subtracted from the data to cancel out the prismatic effect. The *RMSE* values of the phase of the retrieved objects are 2.08 rad, 3.51 rad, and 2.67 rad. The corresponding *PV* values are 6.50 rad, 11.79 rad, and 8.64 rad. All Strehl ratios are again practically zero.

The residual phase of the test objects calculated as the difference between the ground truth and the phase retrieved using the MPGS algorithm is shown in Figure 5. The *RMSE* values of the residual phase are 0.69 rad, 0.71 rad, and 1.23 rad, respectively. The corresponding *PV* values are 3.52 rad, 3.00 rad, and 5.32 rad. The respective values of the Strehl ratio were 0.62, 0.60, and 0.22, indicating a significant increase. In the top panel, one can notice a missed phase wrap of the 55th vector, which leads to the differences between the *RMSEs* of the original and the retrieved object.

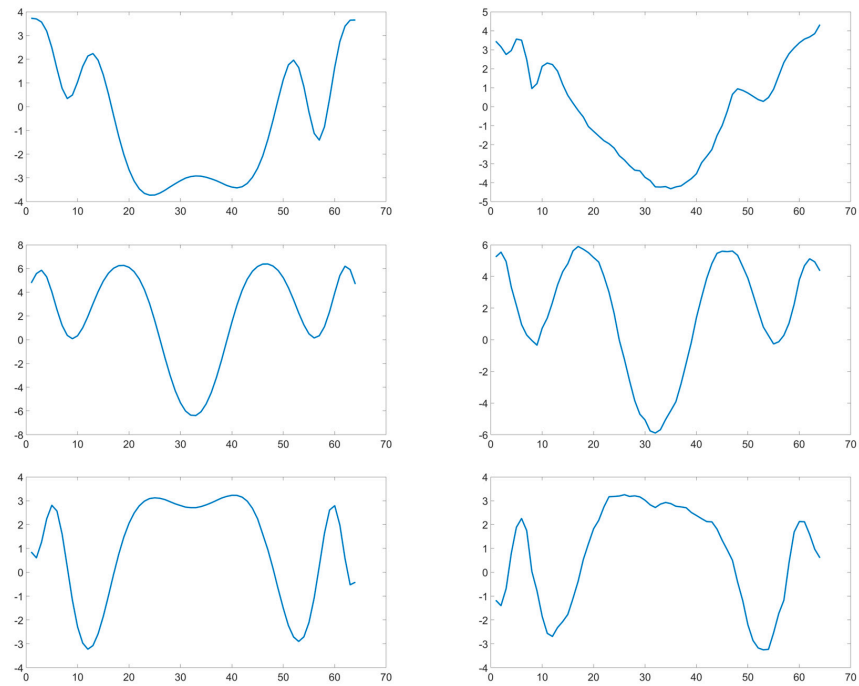


Figure 4. The left column shows the phase (in radians) of the test objects, i.e., the 55th (**top**), the 33rd (**middle**), and the 21st (**bottom**) vectors measured with the ground-truth method vs. the horizontal coordinate of the test object (in pixels). One pixel of the x -axis is equal to $7.5\ \mu\text{m}$. The right column shows the corresponding objects retrieved with the MPGS algorithm.

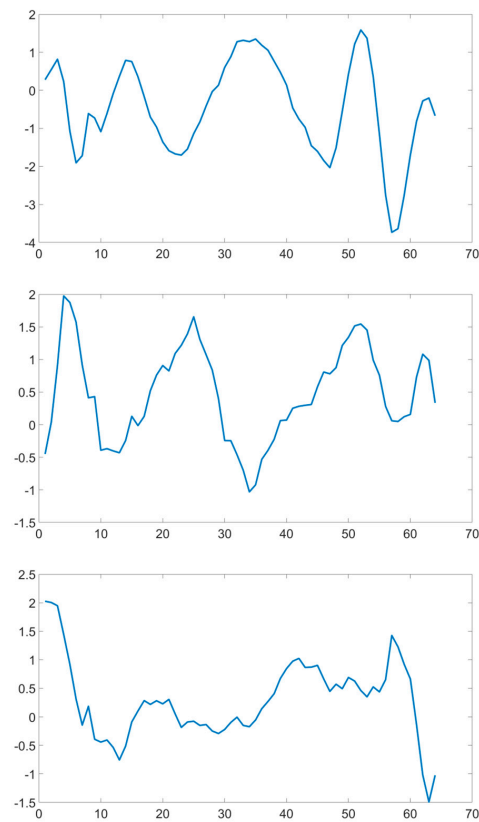


Figure 5. The residual phase (in radians) of the tests objects, i.e., the 55th (**top**), the 33rd (**middle**), and the 21st (**bottom**) vectors vs. the horizontal coordinate of the test object (in pixels). One pixel of the x -axis is equal to $7.5\ \mu\text{m}$. The residual phase was calculated by subtracting the retrieved phase from that measured with the non-contact profiler.

4. Discussion

The results of the current study suggest that a digital signal processor capable of one-dimensional FFT can be successfully used for one-dimensional phase retrieval. In addition, the results show that fixed-point arithmetic in Q15 format is sufficient, thereby reducing the computational load compared to floating-point precision and avoiding complex computational architectures [14] and optical systems. Cameras of low bit depth (8 bits in this study) can be successfully used for phase retrieval; in combination with Q15 fixed-point arithmetic, the proposed method seems computationally efficient. The efficiency of integer arithmetic is also supported by the results of a previous study [13]. The RMSE of the phase of the test object and the residual phase the respective Strehl ratios indicate successful reconstruction. While objects of size 64 by 64 pixels are smaller compared to objects studied elsewhere, they can still be useful in certain areas of optics. For example, the surface quality of microlenses can be estimated using such resolution, given that the size of the pixel is in the micron scale. This would result in the diameter of microlenses being a few hundred microns. For such applications, phase retrieval requiring several minutes is acceptable, and phase retrieval in real-time is not necessary.

The differences between the original and the retrieved object can most likely be attributed to imperfect phase unwrap and an insufficient number of iterations. Possibly, a bigger number of planes on each side of the central plane would also increase the accuracy of the retrieved objects. Another factor lowering the accuracy of phase retrieval is the insufficient width of the empty bands surrounding the object; however, increasing the width of the bands results in an increased memory consumption. The accuracy of the retrieved objects can also be affected by the sampling rate. It is known that the phase retrieval of one-dimensional objects benefits from oversampling [18]; however, this is at the cost of memory usage.

Two-dimensional phase retrieval is, of course, of bigger importance compared to one-dimensional objects, and therefore a question arises as to whether the proposed method could be applied to two-dimensional objects. The most straightforward method would be scanning the object under study; however, such a solution would make the whole procedure very time-consuming. The most efficient way to retrieve two-dimensional objects would be to split them into vectors using an array of optically separated waveguides, capturing an array of diffraction patterns, each of which would be processed using an independent DSP.

Author Contributions: Conceptualization, V.K. and M.O.; methodology, V.K.; software, V.K.; validation, V.K. and M.O.; formal analysis, V.K. and M.O.; investigation, V.K. and M.O.; resources, V.K.; data curation, V.K.; writing—original draft preparation, V.K.; writing—review and editing, V.K. and S.F.; visualization, S.F.; supervision, V.K.; project administration, V.K.; funding acquisition, V.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University of Latvia Foundation and company MikrtoTik; grant number 2257.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Schöck, M.; Mignant, D.L.; Chanan, G.A.; Wizinowich, P.L.; Dam, M.A.V. Atmospheric turbulence characterization with the Keck adaptive optics systems. I. Open-loop data. *Appl. Opt.* **2023**, *42*, 3705–3720. [[CrossRef](#)] [[PubMed](#)]
2. Girkin, J.M.; Poland, S.; Wright, A.J. Adaptive optics for deeper imaging of biological samples. *Curr. Opin. Biotechnol.* **2009**, *20*, 106–110. [[CrossRef](#)] [[PubMed](#)]
3. Artal, P.; Chen, L.; Fernández, E.J.; Singer, B.; Manzanera, S.; Williams, D.R. Neural compensation for the eye's optical aberrations. *J. Vis.* **2004**, *4*, 281–287. [[CrossRef](#)] [[PubMed](#)]
4. Toselli, I.; Gladysz, S. Improving system performance by using adaptive optics and aperture averaging for laser communications in oceanic turbulence. *Opt. Express* **2020**, *28*, 17347–17361. [[CrossRef](#)] [[PubMed](#)]
5. Hampson, K.M.; Turcotte, R.; Miller, D.T.; Kurokawa, K.; Males, J.R.; Ji, N.; Booth, M.J. Adaptive optics for high-resolution imaging. *Nat. Rev. Methods Primers* **2021**, *1*, 68. [[CrossRef](#)] [[PubMed](#)]

6. Harrison, R.W. Phase problem in crystallography. *J. Opt. Soc. Am. A* **1993**, *10*, 1045–1055. [[CrossRef](#)]
7. White, J.; Wang, S.; Eschen, W.; Rothhardt, J. Real-time phase-retrieval and wavefront sensing enabled by an artificial neural network. *Opt. Express* **2021**, *29*, 9283–9293. [[CrossRef](#)] [[PubMed](#)]
8. Barmherzig, D.A.; Sun, J.; Li, P.-N.; Lane, T.J.; Candès, E.J. Holographic phase retrieval and reference design. *Inverse Probl.* **2019**, *35*, 094001. [[CrossRef](#)]
9. Shevkunov, I.; Katkovnik, V.; Petrov, N.V.; Egiazarian, K. Super-resolution microscopy for biological specimens: Lensless phase retrieval in noisy conditions. *Biomed. Opt. Express* **2018**, *9*, 5511–5523. [[CrossRef](#)] [[PubMed](#)]
10. Shechtman, Y.; Eldar, Y.C.; Cohen, O.; Chapman, H.N.; Miao, J.; Segev, M. Phase retrieval with application to optical imaging: A contemporary overview. *IEEE Signal Process. Mag.* **2015**, *32*, 87–109. [[CrossRef](#)]
11. Gerchberg, R.W.; Saxton, W.O. Practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik* **1972**, *35*, 237–250.
12. Zhu, Y.; Xie, S. GPU acceleration for phase retrieval for electromagnetic interference source image. In Proceedings of the Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC), Shenzhen, China, 18–21 May 2016.
13. Rodríguez-Ramos, J.M.; Castelló, E.M.; Conde, C.D.; Valido, M.R.; Marichal-Hernández, J.G. 2D-FFT implementation on FPGA for wavefront phase recovery from the CAFADIS camera. In Proceedings of the Adaptive Optics Systems, Marseille, France, 23–28 June 2008.
14. Smith, J.S.; Dean, B.H.; Haghani, S. Distributed computing architecture for image-based wavefront sensing and 2D FFTs. In Proceedings of the Advanced Software and Control for Astronomy, Orlando, FL, USA, 24–31 May 2006.
15. Dean, B.H.; Zielinski, T.P. Heterogeneous processing architecture for phase-retrieval wavefront sensing. In Proceedings of the Frontiers in Optics 2012/Laser Science XXVIII, Rochester, NY, USA, 14–18 October 2012.
16. Hansen, A.K. Coherent laser phase retrieval in the presence of measurement imperfections and incoherent light. *Appl. Opt.* **2017**, *56*, 7341–7345. [[CrossRef](#)] [[PubMed](#)]
17. Bucu, R.L.; Almoró, P.F. Enhanced multiple-plane phase retrieval using adaptive support. *Opt. Lett.* **2019**, *44*, 6045–6048. [[CrossRef](#)] [[PubMed](#)]
18. Candès, E.J.; Eldar, Y.C.; Strohmer, T.; Voroninski, V. Phase retrieval via matrix completion. *SIAM J. Imaging Sci.* **2013**, *6*, 199–225. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.