



Cimmino simultaneously iterative holographic projection

MIGUEL DE V. EL GUENDY,^{1,2,*}  RALF MOUTHAN,¹ ANDREW KADIS,¹  YOUCHAO WANG,¹  OANA NICULESCU,¹ AMR SOLIMAN,¹ DARRAN MILNE,² ANDRZEJ KACZOROWSKI,² AND TIMOTHY D. WILKINSON¹

¹Centre for Molecular Materials, Photonics and Electronics, Department of Engineering, University of Cambridge, 9 JJ Thomson Ave, Cambridge CB3 0FF, UK

²VividQ Ltd, Cambridge, UK

*me431@cam.ac.uk

Abstract: The Cimmino algorithm is an iterative algorithm used for solving linear equations ($Ax = b$) and is part of a wider family of algebraic reconstruction algorithms. Despite being used in computed tomography and digital signal processing, we present an approach for using it in holographic projections in the Fraunhofer region. Following the work carried out by our group on Kaczmarz holography, an earlier algebraic linear equation solver, we compare the performance and speed of Cimmino against this algorithm. Three versions of the Cimmino algorithm, Cimmino full, Cimmino eye, and Cimmino FFT, are evaluated and compared to both Kaczmarz and Gerchberg-Saxton. Interestingly, the derivation and simplification of Kaczmarz and Cimmino leads to the input-output algorithm originally derived by Fienup and while the solution is well-known, the link between these algorithms is not and despite being around since the 1930s, it was originally used to solve linear equations and not phase retrieval. While the Cimmino full and Cimmino eye reflect fairly poorly against Cimmino FFT and Kaczmarz in terms of image quality and speed, the Cimmino FFT delivers better results in terms of image quality than all algorithms considered. It is significantly faster than Kaczmarz, being only a few milliseconds slower than Gerchberg-Saxton.

Published by Optica Publishing Group under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

1. Introduction

Holography has seen a significant development since its discovery in the 1940s by Dennis Gabor [1]. However, the advancements made in computer technology and its natural improvements in processing power contributed towards the evolution of computer-generated holography (CGH). Conventional holography records the interference pattern between a coherent reference light source and this coherent light scattered off an object onto a photographic plate. This interference pattern is known as a hologram and when a coherent light source shines through it, a replay field (projected image) is generated.

Computer-generated holography allows the user to digitally generate these interference patterns without previously recording the object onto a photographic plate [2]. The evolution of computer controlled spatial light modulators (SLMs) allowed these devices to be incorporated in optical tweezing [3] and telecommunications [4] as well as digital holographic imaging in the form of augmented reality and virtual reality systems (AR and VR respectively) [5–7].

The push to incorporate computer-generated holography into VR and AR systems comes from the desire to have full 3-dimensional images with true depth cues. Current "3D" alternatives, whether they are stereoscopic or light field displays, are not capable of delivering these images

with full depth cues without a loss in resolution. The main benefit of being capable of delivering these images with full depth cues is that it mitigates what is known as vergence-accommodation conflict (VAC) [8,9].

While computer-generated holography has notorious 3-dimensional benefits, this paper will primarily focus in projecting 2-dimensional objects since it emphasizes on the effectiveness of the algorithm in holographic displays as opposed to its direct applicability to VR and AR.

In this system, a spatial light modulator (SLM) is used to modulate the coherent light beam and allows either the amplitude or the phase of this incident light to be controlled and the interference pattern to be subsequently influenced. Its digital interface restricts the modulation of light into discrete energy levels. This device is pixelated and is in the diffraction field (H). As a result, the projected image or replay field (R) in the Fraunhofer region is also portrayed as discrete pixels [10]. This relationship is depicted in Fig. 1.

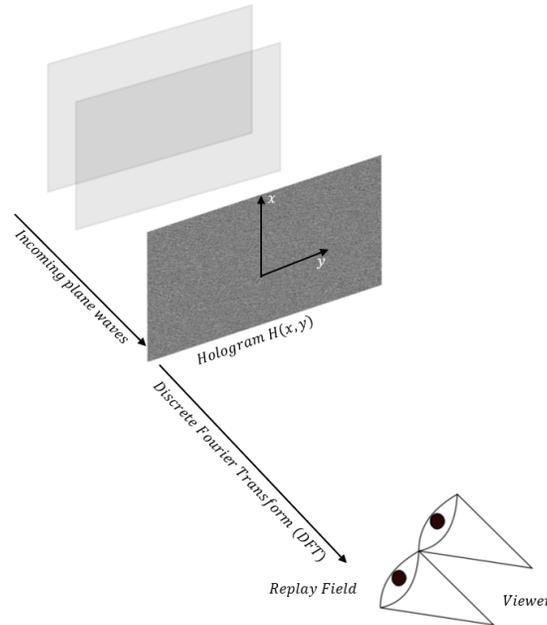


Fig. 1. Diffraction and replay fields coordinate system

From Goodman [11], we have proof that the discrete Fourier transform (DFT) of the aperture function of the SLM will recreate the desired replay field in the Fraunhofer region (far-field). Equations (1) and (2) demonstrate the relationship between the diffraction and replay fields. The Fourier transform is typically represented as an integral function, however, in this instance it is portrayed as a DFT sum.

$$R_{u,v} = \mathcal{F}\{H_{x,y}\} = \frac{1}{\sqrt{m_x n_y}} \sum_{x=0}^{m_x-1} \sum_{y=0}^{n_y-1} H_{x,y} e^{-2\pi i \left(\frac{ux}{m_x} + \frac{vy}{n_y} \right)} \quad (1)$$

$$H_{x,y} = \mathcal{F}^{-1}\{R_{u,v}\} = \frac{1}{\sqrt{m_x n_y}} \sum_{u=0}^{m_x-1} \sum_{v=0}^{n_y-1} R_{u,v} e^{2\pi i \left(\frac{ux}{m_x} + \frac{vy}{n_y} \right)} \quad (2)$$

The SLM used to modulate the coherent light source is restricted to modulating either the phase or the amplitude of this source. The challenge therefore lies in reconstructing the intended replay field in spite of these SLM restrictions.

Our earlier investigation, on Kaczmarz holographic projections in the Fraunhofer region [12] revealed that while very high quality replay fields are generated, it is still comparatively slower than the more widely used Gerchberg-Saxton. The Kaczmarz algorithm iterates the Ax component in a linear equation on a row-by-row basis and therefore explains the slower running of this phase retrieval method.

Error metrics used throughout this investigation are the same as Kaczmarz holography whereby mean squared error (MSE) and peak signal to noise ratio (PSNR) are used to compare the generated replay fields (R), to the target (T).

$$MSE(T, R) = \frac{1}{m_x n_y} \sum_{u=0}^{m_x-1} \sum_{v=0}^{n_y-1} |T_{u,v} - R_{u,v}|^2 \quad (3)$$

$$PSNR(T, R) = 20 \log_{10} \left(\frac{m_x n_y}{\sqrt{MSE}} \right) \quad (4)$$

2. Algebraic reconstruction techniques

The 1970s saw a remarkable progression in the image reconstruction field with the first successful implementation of the algebraic reconstruction technique (ART) by Gordon, Bender and Herman [13]. It is mainly used in computed tomography and reconstructs an image from a large number of projections at uniformly distributed angles. These are then used to reconstruct a two-dimensional beam density distribution [14]. The method used in this approach was the iterative algorithm developed by Polish mathematician, Stefan Kaczmarz [15] in the 1930s, aptly named the Kaczmarz algorithm. It is used to solve linear equations in the form of $Ax = b$ and has been used in a variety of field ranging from computed tomography to microscopy and metasurface antennas [16–19].

More recently, the Kaczmarz algorithm was shown to be effective in computer-generated holography, projecting holographic images in the Fraunhofer region [12]. It is therefore very suitable to resolving the phase retrieval problem associated with holographic projections.

The original simple Kaczmarz algorithm, seen in Eq. (5) solves overdetermined linear systems and works by sequentially iterating in a row-action manner, suggesting the overall solution to x is updated one row of matrix A at a time. The row iterates sequentially with $r = l + 1 \pmod{m}$. For that reason, this approach is referred to as a fully sequential method in the algebraic reconstruction field [20–22].

$$x_{l+1} = x_l + \frac{b_r - A_r x_l}{\|A_r\|_2^2} A_r^* \quad (5)$$

Wei [23] more recently adapted the Kaczmarz algorithm to solve phase retrieval problems where the magnitude of Ax is equated to b . The phase is then a free parameter and determined as $\theta_l = \angle A_r x_l$. The original approach can therefore be written as:

$$x_{l+1} = x_l + \frac{b_r e^{j\theta_l} - A_r x_l}{\|A_r\|_2^2} A_r^* \quad (6)$$

The r th row of A is A_r while A_r^* is the complex conjugate transpose of A_r .

Our previous work on using the Kaczmarz algorithm to project replay fields in the Fraunhofer region [12] demonstrates that while high image quality is obtainable, its row-action approach makes it much slower than alternative methods that compute the product of Ax simultaneously. Furthermore, iterative algorithms such as Gerchberg-Saxton use Fast Fourier Transforms (FFTs), a significantly more time-effective approach to resolve this phase retrieval problem, $O(N \log N)$ as opposed to $O(N^2)$.

This investigation, therefore, evaluates a mechanism that is still encompassed within the algebraic reconstruction group, however, rather than updating the overall solution of x one row of A at a time, it determines x using all rows of A simultaneously. This approach is known as the Cimmino algorithm and is part of the fully simultaneous family within the algebraic reconstruction mechanisms.

2.1. Cimmino algorithm

First published in 1938, after Kaczmarz, G. Cimmino developed a mechanism that resolves linear equations in the form of $Ax = b$, simultaneously updating solution x without sequentially iterating through the rows of matrix A [24].

Rather than updating the vector x_{l+1} after each row, it determines the average of all the previous iteration vectors [22,25]. The Kaczmarz approach is therefore adapted as follows:

$$x_{l+1} = x_l + \frac{1}{m} \sum_{r=1}^m \frac{b_r e^{j\theta_l} - A_r x_l}{\|A_r\|_2^2} A_r^*; \quad \text{where } \theta_l = \angle A_r x_l \quad (7)$$

This equation can therefore be expressed as follows:

$$x_{l+1} = x_l + A^* M (b e^{j\theta_l} - A x_l); \quad \text{where } M = \text{diag} \left(\frac{1}{m \|A_r\|_2^2} \right) \quad (8)$$

3. Adapting Cimmino to computer-generated holography

The Cimmino algorithm is widely recognized as being useful in computed tomography and digital signal processing, however, much like in our previous work involving the Kaczmarz algorithm we are adapting these algorithms to project 2-dimensional holographic images in the Fraunhofer region. Our motivation involves finding a mechanism that involves fewer computational steps as well as delivering high quality replay fields. Kaczmarz delivered higher quality projections but required significantly more time to update the hologram after each iteration.

Much like our previous Kaczmarz work, this problem has to be adapted to suit the nature of this algorithm. As opposed to Gerchberg-Saxton and Fienup, for instance, that solve this phase retrieval problem through fast Fourier transforms (FFTs) [26,27], Kaczmarz and Cimmino use discrete Fourier transform (DFT) matrices to generate the replay field in the far-field. This is achieved by multiplying the 2-dimensional DFT matrix with a vector of the hologram.

$$\begin{array}{ccc} \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} & \begin{pmatrix} x_{11} \\ \vdots \\ x_{m1} \end{pmatrix} & = & \begin{pmatrix} b_{11} \\ \vdots \\ b_{m1} \end{pmatrix} \\ \text{2D DFT} & \text{Holo} & & \text{Target} \end{array}$$

In this $Ax = b$ linear system, A is the 2D DFT matrix, x is the vector of the hologram and b is the target image. The hologram is configured with a constant phase at the start throughout this investigation and the target image will be iteratively compared to the updated replay fields. The Cimmino algorithm will be compared to Kaczmarz and Gerchberg-Saxton and the results discussed in the subsequent sections.

It is worth noting at this stage, however, that the main challenge faced when adapting Kaczmarz and Cimmino to computer-generated holography is adjusting to the constraints imposed by the spatial light modulator (SLM). When using these devices, one can modulate the phase or the amplitude of the coherent light source, not both, and in this case we constrained the amplitude of the hologram. This was accomplished by multiplying a mask of ones by the angle of the updated

iteration of the hologram. If $Illum$ below is a mask of ones and θ_l is the angle of the updated iteration of the hologram.

$$x_{l+1} = Illum.e^{j\theta_l}; \quad \text{where } \theta_l = \angle x_{l+1} \quad (9)$$

The Cimmino algorithm was divided into three sections with three different approaches to resolving this phase retrieval problem. These will be addressed as full Cimmino, Cimmino eye and Cimmino FFT.

3.1. Full Cimmino algorithm

The full Cimmino algorithm refers to the original algorithm initially developed by G. Cimmino [24] and seen in Per Christian Hansen's and Tommy Elfving's [20,22,25] work with algebraic reconstruction techniques (ART). The matrix version seen in Eq. (8) has been adopted to represent this version of Cimmino.

$$x_{l+1} = x_l + A^*M(be^{j\theta_l} - Ax_l); \quad \text{where } M = \text{diag}\left(\frac{1}{m\|A_r\|^2}\right) \quad (10)$$

3.2. Cimmino eye algorithm

The Cimmino eye algorithm replaces the diagonal matrix, represented as M , with an identity matrix, \mathbb{I} . Depending on the size of the matrices being computed, the diagonal matrix, M , can take some time to initially calculate and determine. Replacing this with an identity matrix should cut down this time and consequently the overall time to process this phase retrieval algorithm. The expression would therefore now read as:

$$x_{l+1} = x_l + A^*\mathbb{I}(be^{j\theta_l} - Ax_l) \quad (11)$$

Interestingly, this is also a well known Landweber iteration [28].

3.3. Cimmino FFT algorithm

Following the use of an identity matrix in the Cimmino eye equation, one can simplify and greatly increase the speed of this algorithm. If we interpret A^* as the complex conjugate transpose of the original DFT matrix A and similarly if we know that the inverse fast Fourier transform (IFFT) is the complex conjugate transpose of the fast Fourier transform (FFT), we can replace A^* in the equation with an IFFT.

The diagonal matrix is now an identity matrix, following the Cimmino eye approach, and finally we maintain the difference between the target b and the fast Fourier transform of the hologram from the previous iteration, x_l . By adopting this FFT approach, we add a weighting parameter, W , replicating Fienup's input-output algorithm [27].

$$x_{l+1} = x_l + W\mathcal{F}^{-1}(be^{j\theta_l} - \mathcal{F}(x_l)) \quad (12)$$

The significant reduction in matrix multiplications suggests this algorithm computes much faster than the alternatives.

Interestingly, as previously mentioned, from a completely different starting point and different method, James Fienup has achieved the same result by computing the difference between the target and current replay field, taking the inverse FFT of the product and adding a weighting parameter [27]. This is more commonly known as the input-output method he developed from the error-reduction algorithm, Gerchberg-Saxton.

4. Holographic image projection using the Cimmino algorithm

The following test images of 128x128 pixels, seen in Fig. 2, were used to test the aforementioned algorithms, as well as the Kaczmarz and Gerchberg-Saxton algorithms. The Kaczmarz, Cimmino full and eye algorithms are constrained by image size, since they heavily rely on matrix multiplications, however, the remaining two algorithms, Cimmino FFT and Gerchberg-Saxton do not have this issue. Larger 512x512 pixels of the same target images tested experimentally, seen in Section 5 to further compare these algorithms.

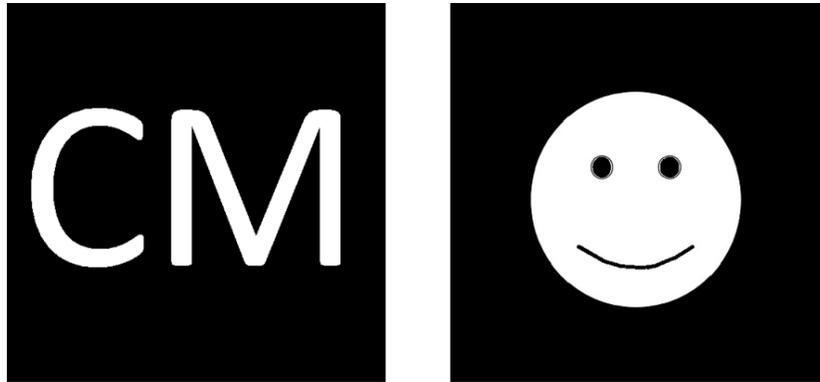


Fig. 2. Target images used throughout this investigation

All algorithms were evaluated in the far-field for ten iterations and their performances in terms of image quality, via mean squared error (MSE) and peak signal to noise ratio (PSNR), as well as overall algorithm runtime was assessed.

4.1. Algorithm replay field image quality

This section quantitatively assesses the image quality of the replay fields generated by the aforementioned algorithms. The mean squared error and peak signal to noise ratio metrics were used throughout to assess the quality of the replay fields. The performance of the algorithms in successfully retrieving the target images seen in Fig. 2 can be seen in Figs. 3 and 4.

The retrieved replay fields vary in performance. For consistency all algorithms have the same starting conditions, with the hologram having a constant phase, and the target image a random phase. Cimmino FFT/ input out algorithm, Kaczmarz, Cimmino eye are displaying better results when compared to Cimmino full and Gerchberg-Saxton. While all algorithm successfully retrieve the correct phases and neatly reconstruct the original target image, the variations in image quality is still apparent. It must be noted at this stage the weight parameter adopted on the Cimmino FFT method is $\frac{m^2}{2}$ where m^2 is the dimension of the image. The results are consistent irrespective of the target image. The metrics in Figs. 5 and 6 confirm this disparity in performances. It is noticeable that all Cimmino variations and Kaczmarz outperform Gerchberg-Saxton in terms of image quality metrics in 10 iterations, with Cimmino FFT displaying the clearest image.

From the work previously conducted on Kaczmarz holographic projections [12], it is noticeable the image quality for Kaczmarz outperforms Gerchberg-Saxton in 10 iterations. However, the main drawbacks of implementing Kaczmarz are its slow running time and ability to solely be adopted on smaller images 128x128 pixels since it relies on discrete Fourier transform (DFT) matrices. Cimmino FFT is shown to be of adequate usage on larger images as seen in Fig. 7 where the 512x512 pixel target image is successfully recreated in the far-field. In this case, 100 iterations were used rather than 10 since using fast Fourier transforms is both faster to run and not as heavy computationally as Kaczmarz, Cimmino full and Cimmino eye rely on matrix



Fig. 3. Simulated replay fields retrieved from target image CM; Top Left: Cimmino FFT (CM FFT); Top Middle: Cimmino Full (CM Full); Top Right: Cimmino Eye (CM Eye); Bottom Left: Gerchberg-Saxton (GS); Bottom Right: Kaczmarz (KZ)

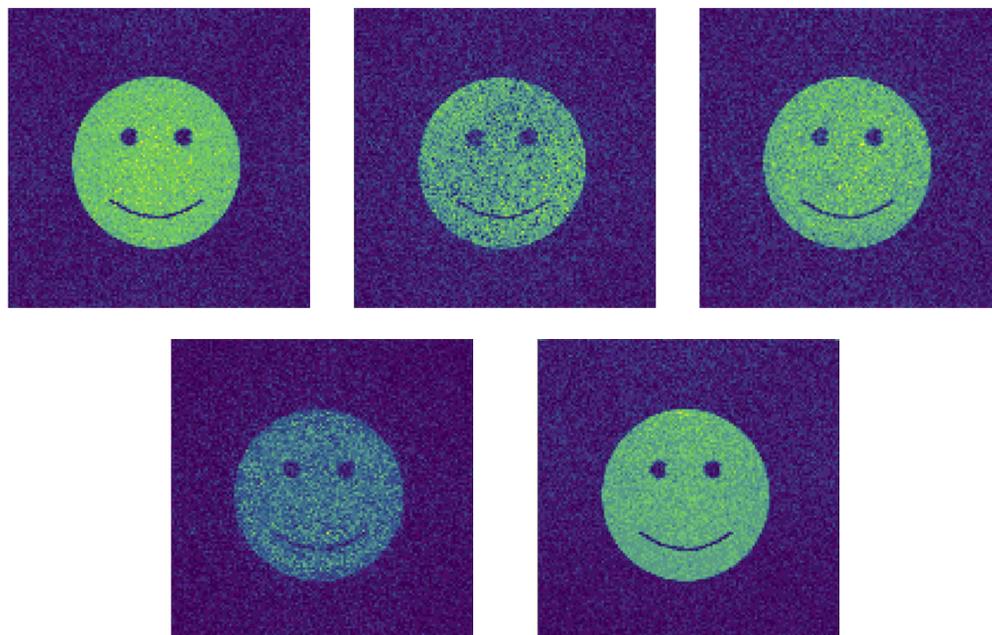


Fig. 4. Simulated replay fields retrieved from smiley face target image; Top Left: Cimmino FFT (CM FFT); Top Middle: Cimmino Full (CM Full); Top Right: Cimmino Eye (CM Eye); Bottom Left: Gerchberg-Saxton (GS); Bottom Right: Kaczmarz (KZ)

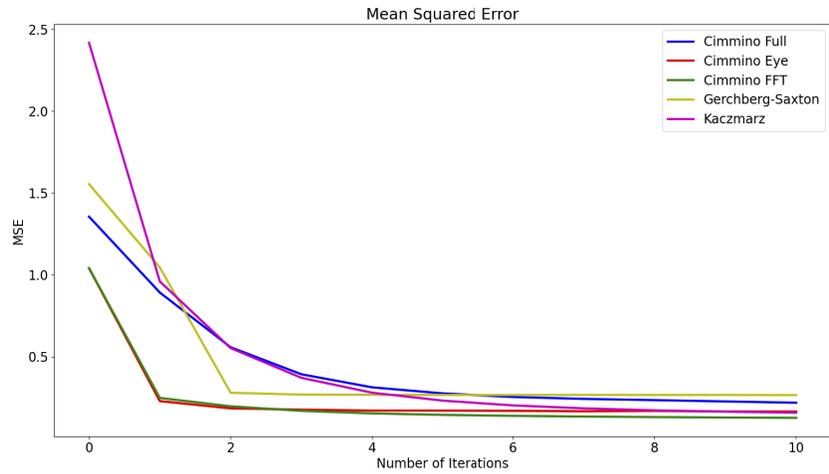


Fig. 5. Mean squared errors (MSE) of Cimmino FFT, Cimmino full, Cimmino eye, Gerchberg-Saxton, and Kaczmarz

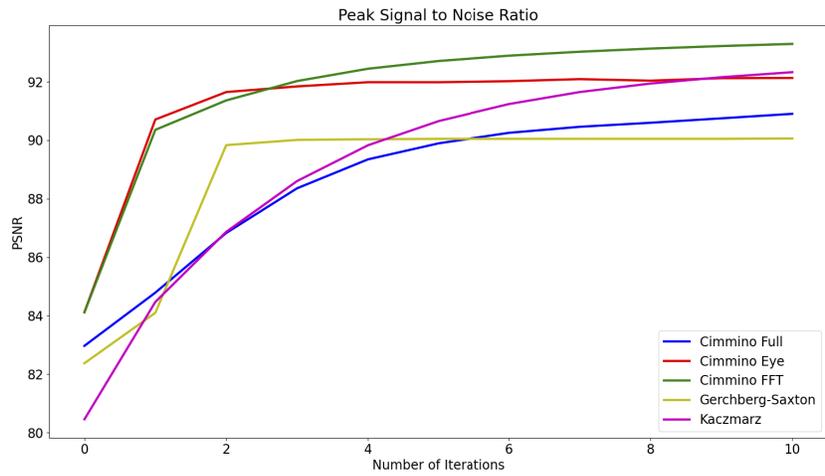


Fig. 6. Peak Signal to noise ratios (PSNR) of Cimmino FFT, Cimmino full, Cimmino eye, Gerchberg-Saxton, and Kaczmarz

calculations. Cimmino FFT still outperforms Gerchberg-Saxton in 100 iterations according to the metrics seen in Fig. 8 and 9.

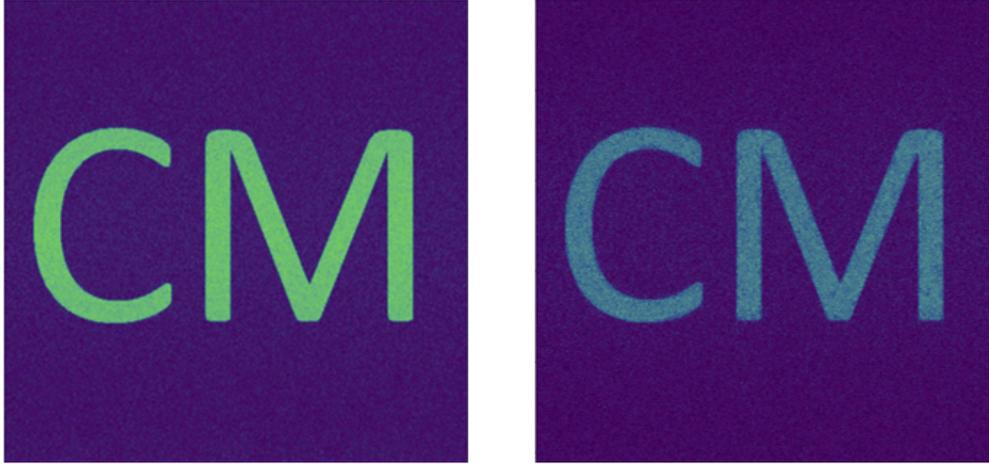


Fig. 7. Simulated replay fields on larger 512x512 pixel target images; Left: Cimmino FFT (CM FFT); Right: Gerchberg-Saxton (GS)

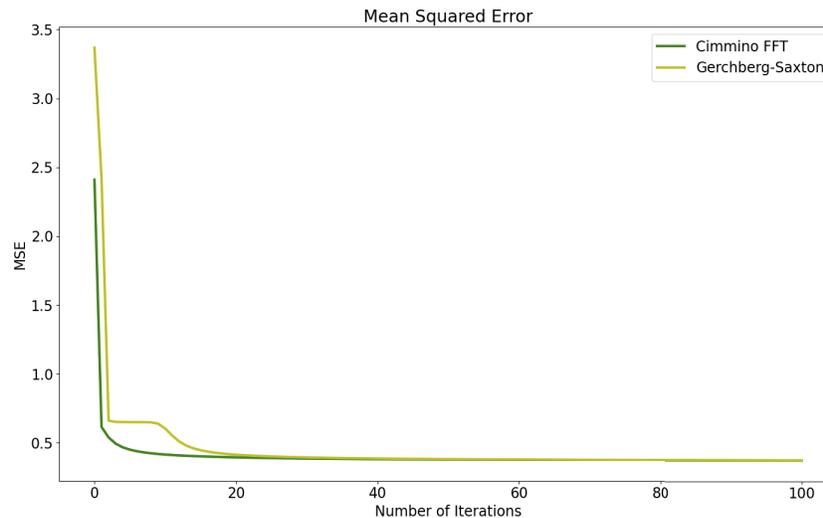


Fig. 8. MSE of 512x512 pixel replay fields

It must be noted, however, the stagnation seen in Gerchberg-Saxton is expected when applying an initial constant phase [29]. This does not happen with Cimmino FFT and while it could be argued an initial random phase is more suitable, when it comes to assessing image quality, an initial random phase may be counter-productive in obtaining a high quality image. By adding a random phase, we are essentially distributing and elevating the phase across the entire hologram plane [29]. On the other hand, a constant phase does not distribute this phase and it is concentrated at a single point in the hologram plane. This can be broadly extended to consequently distributing noise across the replay field using random phase while this does not happen using a constant phase. Similarly, using a constant phase will bandlimit the signal considerably when compared

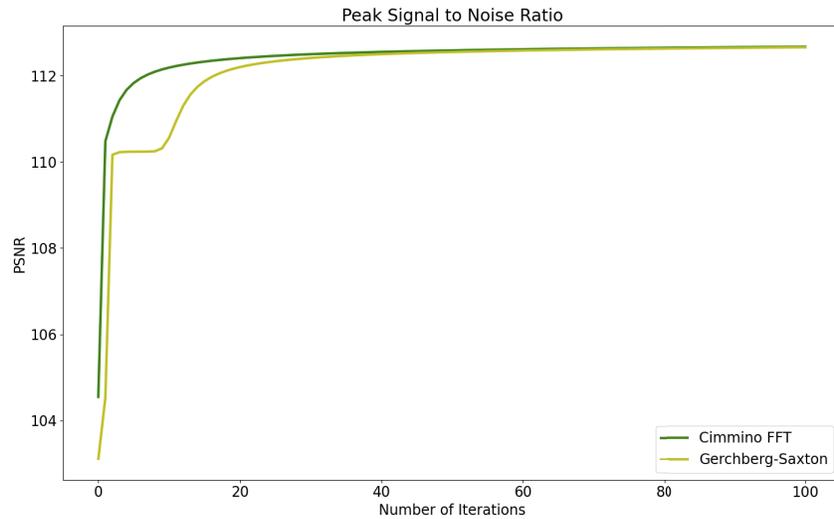


Fig. 9. PSNR of 512x512 pixel replay fields

to the random phase counterpart. Nevertheless, with algorithms such as Gerchberg-Saxton, randomizing the phase will avoid stagnation and convergence can be obtained significantly faster.

Wyrowski and Bryngdahl [29] also suggested using a quadratic phase as a means of improving the overall image quality of the holographic display. More recently, Wu et al., have also looked into adapting Gerchberg-Saxton with this technique, and have seen significant improvements in image quality with a PSNR improvement of 4.8dB on average [30].

Furthermore, related to the work carried out by Wu et al. [30], Chen et al. have used Stochastic gradient descent to demonstrate improvements in image quality [31]. Deep learning and neural networks in holography are additional separate methods for improving image quality, however, these are not covered in this work [7,32,33], since the purpose of this work is finding the link between the traditional linear algebraic solving algorithms such as Kaczmarz and Cimmino, with more modern holography display algorithms.

4.2. Algorithm running times

This section analyzes the time taken for each algorithm to complete 10 iterations. A comparison between the performance of each approach in terms of speed will subsequently be made.

As well as only being capable of reconstructing replay fields of smaller dimensions, the Kaczmarz algorithm is also slower than fast Fourier transform (FFT) based algorithms such as the Gerchberg-Saxton, as seen in our previous work [12]. As a result, alternative algorithms that compute the full image simultaneously rather than relying on row-by-row iteration, as seen by Kaczmarz, were evaluated. The family of simultaneously iterative algebraic reconstruction techniques, known as the Cimmino algorithms were therefore considered.

Despite simultaneously resolving the linear equation $Ax = b$, the Cimmino full and Cimmino eye mechanisms still rely on discrete Fourier transform matrices and as a result, are not comparable in terms of speed to Gerchberg-Saxton. The overall runtimes of all the algorithms for 128x128 target images is shown in Fig. 10.

Despite solving the linear equations simultaneously, both the Cimmino full and Cimmino eye are significantly slower than the Kaczmarz algorithm. The multiple large matrix multiplications required to successfully compute these Cimmino approaches explain this extended runtime.

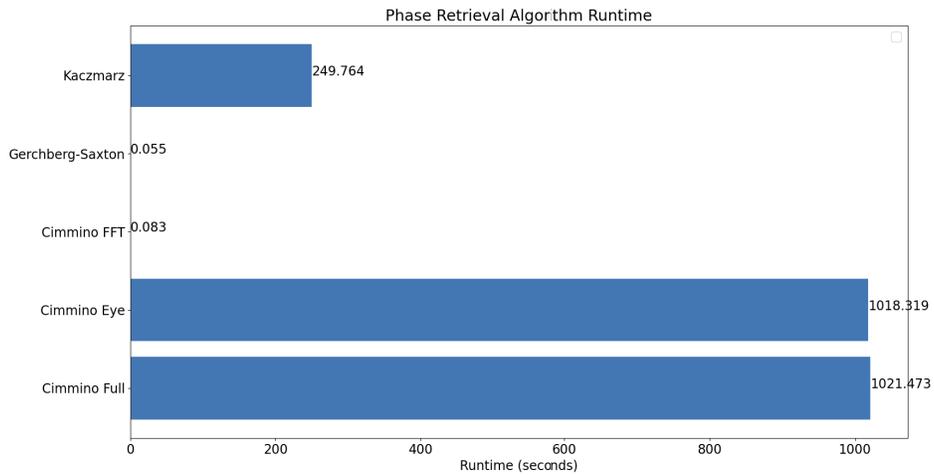


Fig. 10. Algorithm runtimes for 128x128 pixel target images

When smaller matrices are used, for instance, for target images 32x32 pixels, this runtime is considerably reduced, running faster than its Kaczmarz counterpart, Fig. 11.

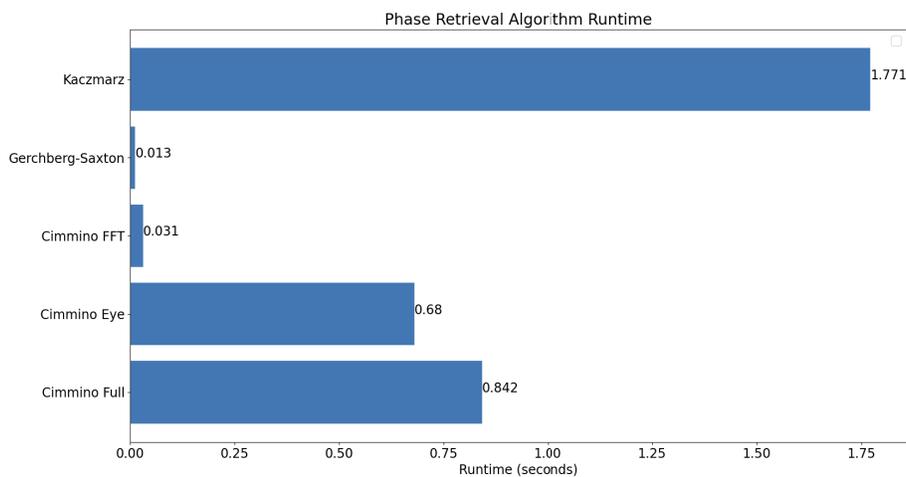


Fig. 11. Algorithm runtimes for 32x32 pixel target images

As expected, Cimmino FFT is slightly slower than Gerchberg-Saxton, since it relies on more computations than the latter. Nevertheless, the disparity in speed between both approaches is minimal.

5. Experimental analysis

This section is a validation that these algorithms effectively run on a far-field holographic projector. The main comparison drawn in this section is between Cimmino FFT and Gerchberg-Saxton since out of all these algorithms, these can be successfully implemented on larger images. For that reason, 512x512 pixel replay fields are displayed throughout this section.

The projector uses a 520nm laser and its light is modulated by a multiphase Himax 7322 spatial light modulator (SLM) with a 1920x1080 pixel resolution. The generated replay fields are single frame and the results can be seen in Fig. 12.

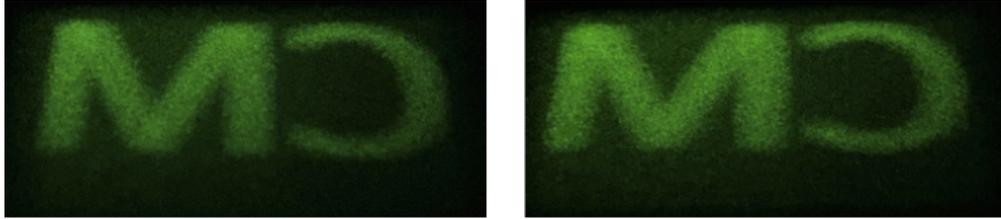


Fig. 12. Algorithm experimental analysis; Left: Cimmino FFT; Right: Gerchberg-Saxton

Comparing the mean squared error and peak signal to noise ratios of the retrieved experimental replay fields with the target image seen in Fig. 2, we obtained 114.58 and 80.53 (MSE and PSNR respectively) for Cimmino FFT, against 115.68 and 80.49 (MSE and PSNR) for Gerchberg-Saxton. There is some improvement when comparing Cimmino FFT to Gerchberg-Saxton experimentally.

6. Conclusion

Our previous work analyzing the performance of the Kaczmarz algorithm in projecting replay fields in the far-field, demonstrated this algorithm delivered higher quality images than Gerchberg-Saxton in 10 iterations [12]. Nevertheless, while it does outperform in terms of image quality, it relies on matrix computations, making it slower than the alternative and difficult to process on very large images. The Cimmino algorithm should therefore be a viable alternative to Kaczmarz since it computes the whole image simultaneously rather than on a row-by-row basis. In fact, as seen from this earlier analysis, as well as delivering poorer results, Cimmino full and Cimmino eye are also significantly slower than Kaczmarz when running on larger 128x128 images.

The Cimmino FFT, on the other hand, performs significantly faster than Kaczmarz, Cimmino full and Cimmino eye whilst solely being a few milliseconds slower than Gerchberg-Saxton. It delivers better image quality than all remaining algorithms, whilst also being successfully implemented on larger 512x512 images.

While these results validate Fienup's work [27], they also demonstrate a parallel between two fields that have co-existed without this link. Kaczmarz [15] and Cimmino [24] originally conceived this work in the 1930s to solve linear equations in the form of $Ax = b$ and not for phase retrieval. Fienup developed his algorithm in the 1980s specifically comparing input-output to Gerchberg-Saxton, or the error-reduction algorithm. While the motivations for these works were different, the link between these fields is evident and interesting to note.

Funding. VividQ; Engineering and Physical Sciences Research Council (EP/L015455/1).

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available at the University of Cambridge Apollo repository [34].

References

1. D. Gabor, "A new microscopic principle," *Nature* **161**(4098), 777–778 (1948).
2. Y. Wang, D. Dong, P. J. Christopher, A. Kadis, R. Mouthaan, F. Yang, and T. D. Wilkinson, "Hardware implementations of computer-generated holography: a review," *Opt. Eng.* **59**(10), 102413 (2020).
3. J. Grieve, A. Ulcinas, S. Subramanian, G. Gibson, M. Padgett, D. Carberry, and M. Miles, "Hands-on with optical tweezers: a multitouch interface for holographic optical trapping," *Opt. Express* **17**(5), 3595–3602 (2009).
4. W. Crossland, T. Wilkinson, I. Manolis, M. Redmond, and A. Davey, "Telecommunications applications of ICOS devices," *Mol. Cryst. Liq. Cryst.* **375**, 1–13 (2002).

5. T. Widjanarko, M. El Guendy, A. Spiess, D. Sullivan, T. Durrant, O. Tastemur, A. Newman, D. Milne, and A. Kaczorowski, "Clearing key barriers to mass adoption of augmented reality with computer-generated holography," in *Optical Architectures for Displays and Sensing in Augmented, Virtual, and Mixed Reality (AR, VR, MR)*, vol. 11310 (International Society for Optics and Photonics, 2020), p. 113100B.
6. A. Maimone, A. Georgiou, and J. S. Kollin, "Holographic near-eye displays for virtual and augmented reality," *ACM Transactions on Graphics (Tog)* **36**(4), 1–16 (2017).
7. L. Shi, B. Li, C. Kim, P. Kellnhofer, and W. Matusik, "Towards real-time photorealistic 3D holography with deep neural networks," *Nature* **591**(7849), 234–239 (2021).
8. T. Shibata, J. Kim, D. M. Hoffman, and M. S. Banks, "Visual discomfort with stereo displays: effects of viewing distance and direction of vergence-accommodation conflict," in *Stereoscopic Displays and Applications XXII*, vol. 7863 (International Society for Optics and Photonics, 2011), p. 78630P.
9. K. Ukai and P. A. Howarth, "Visual fatigue caused by viewing stereoscopic motion images: background, theories, and observations," *Displays* **29**(2), 106–116 (2008).
10. P. J. Christopher, R. Mouthaan, M. El Guendy, and T. D. Wilkinson, "Linear-time algorithm for phase-sensitive holography," *Opt. Eng.* **59**(08), 085104 (2020).
11. J. W. Goodman, *Introduction to Fourier Optics* (Roberts & Co., 2005).
12. M. d. V. El Guendy, R. Mouthaan, B. Wetherfield, A. Kadis, P. J. Christopher, A. Kaczorowski, and T. D. Wilkinson, "Kaczmarz holography: holographic projection in the Fraunhofer region," *Opt. Eng.* **60**(10), 103101 (2021).
13. R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography," *J. Theor. Biol.* **29**(3), 471–481 (1970).
14. D. Raparia, J. Alessi, and A. Kponou, "The algebraic reconstruction technique (ART)," in *Proceedings of the 1997 Particle Accelerator Conference (Cat. No. 97CH36167)*, vol. 2 (IEEE, 1997), pp. 2023–2025.
15. S. Karczmarz, "Angenaherte auflösung von systemen linearer gleichungen," *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.* **35**, 355–357 (1937).
16. T. Strohmer and R. Vershynin, "J. Fourier Analysis Appl.," *Journal of Fourier Analysis and Applications* **15**(2), 262–278 (2009).
17. Y. Jiao, B. Jin, and X. Lu, "Preasymptotic convergence of randomized kaczmarz method," *Inverse Probl.* **33**(12), 125012 (2017).
18. C. Schretter, D. Blinder, S. Bettens, H. Ottevaere, and P. Schelkens, "Regularized non-convex image reconstruction in digital holographic microscopy," *Opt. Express* **25**(14), 16491–16508 (2017).
19. Z. Luo, Y. Cheng, K. Cao, Y. Qin, and K. Liu, "A phase-calibrated computational imaging method based on programmable metasurface," in *2019 IEEE 2nd International Conference on Electronic Information and Communication Technology (ICEICT)*, (IEEE, 2019), pp. 778–780.
20. P. C. Hansen and M. Saxild-Hansen, "Air tools—a Matlab package of algebraic iterative reconstruction methods," *J. Comput. Appl. Math.* **236**(8), 2167–2178 (2012).
21. P. C. Hansen and J. S. Jørgensen, "Air tools II: algebraic iterative reconstruction methods, improved implementation," *Numer. Algorithms* **79**(1), 107–137 (2018).
22. T. Elfving, P. C. Hansen, and T. Nikazad, "Semi-convergence properties of Kaczmarz's method," *Inverse Probl.* **30**(5), 055007 (2014).
23. K. Wei, "Solving systems of phaseless equations via Kaczmarz methods: A proof of concept study," *Inverse Probl.* **31**(12), 125008 (2015).
24. G. Cimmino, "Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari," *La Ricerca Scientifica (Roma)* **1**, 326–333 (1938).
25. T. Elfving, P. C. Hansen, and T. Nikazad, "Semiconvergence and relaxation parameters for projected SIRT algorithms," *SIAM J. on Sci. Comput.* **34**(4), A2000–A2017 (2012).
26. R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik* **35**, 237–246 (1972).
27. J. R. Fienup, "Phase retrieval algorithms: a comparison," *Appl. Opt.* **21**(15), 2758–2769 (1982).
28. M. Hanke, A. Neubauer, and O. Scherzer, "A convergence analysis of the Landweber iteration for nonlinear ill-posed problems," *Numerische Mathematik* **72**(1), 21–37 (1995).
29. F. Wyrowski and O. Bryngdahl, "Iterative Fourier-transform algorithm applied to computer holography," *J. Opt. Soc. Am. A* **5**(7), 1058–1065 (1988).
30. Y. Wu, J. Wang, C. Chen, C.-J. Liu, F.-M. Jin, and N. Chen, "Adaptive weighted Gerchberg-Saxton algorithm for generation of phase-only hologram with artifacts suppression," *Opt. Express* **29**(2), 1412–1427 (2021).
31. C. Chen, B. Lee, N.-N. Li, M. Chae, D. Wang, Q.-H. Wang, and B. Lee, "Multi-depth hologram generation using stochastic gradient descent algorithm with complex loss function," *Opt. Express* **29**(10), 15089–15103 (2021).
32. Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," *ACM Transactions on Graphics (TOG)* **39**(6), 1–14 (2020).
33. S. Choi, M. Gopakumar, Y. Peng, J. Kim, and G. Wetzstein, "Neural 3D holography: learning accurate wave propagation models for 3D holographic virtual and augmented reality displays," *ACM Transactions on Graphics (TOG)* **40**(6), 1–12 (2021).
34. M. El Guendy and T. Wilkinson, "Cimmino simultaneously iterative holographic projection," University of Cambridge Apollo (2022), <https://doi.org/10.17863/CAM.83457>.