Research Article | Vol. 31, No. 23 / 6 Nov 2023 / Optics Express 38589
Optics EXPRESS

Autoregressive modeling for lossless compression of holograms

RAEES KIZHAKKUMKARA MUHAMAD,[1,2,*] COLAS SCHRETTER,[1,2] DAVID BLINDER,[1,2,3] AND PETER SCHELKENS[1,2]

[1]Vrije Universiteit Brussel (VUB), Dept. of Electronics and Informatics (ETRO), Pleinlaan 2, B- 1050 Brussels, Belgium
[2]imec, Kapeldreef 75, B- 3001 Leuven, Belgium
[3]Graduate School of Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba, Chiba, Japan
[*]raees.kizhakkumkara.muhamad@vub.be

Abstract: The large number of pixels to be processed and stored for digital holographic techniques necessitates the development of effective lossless compression techniques. Use cases for such techniques are archiving holograms, especially sensitive biomedical data, and improving the data transmission capacity of bandwidth-limited data transport channels where quality loss cannot be tolerated, like display interfaces. Only a few lossless compression techniques exist for holography, and the search for an efficient technique well suited for processing the large amounts of pixels typically encountered is ongoing. We demonstrate the suitability of autoregressive modeling for compressing signals with limited spatial bandwidth content, like holographic images. The applicability of such schemes for any such bandlimited signal is motivated by a mathematical insight that is novel to our knowledge. The devised compression scheme is lossless and enables decoding architecture that essentially has only two steps. It is also highly scalable, with smaller model sizes providing an effective, low-complexity mechanism to transmit holographic data, while larger models obtain significantly higher compression ratios when compared to state-of-the-art lossless image compression solutions, for a wide selection of both computer-generated and optically-acquired holograms. We also provide a detailed analysis of the various methods that can be used for determining the autoregressive model in the context of compression.

1. Introduction

Digital holographic techniques spatially sample a 2D interference pattern produced by coherent light waves reflected or transmitted from the different objects in the 3D scene, from which the scene can be recreated at a later desired stage. The 2D pattern is to be sampled at an interval with a similar order of magnitude as the wavelength of coherent illumination. As the sampling rate increases, a larger field of view can be made visible without aliasing, as dictated by the grating equation [1]. This results in holograms that are relatively data-dense. To provide an example of a high-end use case, recording/displaying the blue spectrum on a 1.0 cm^2 patch without aliasing/loss of FOV requires close to 2 terapixels. For many applications like video displays [2], holographic microscopy, and tomography [3], this could mean several terabytes of data or more that must be processed and ultimately stored or transmitted. Data processing and transmission/storage of digital holograms (DH) are among its most pressing challenges, preventing widespread adoption in many applications where available computational power and transmission bandwidth is limited [2,4,5]. Compression techniques digitally represent the data using fewer bits and can help mitigate this bottleneck. With lossless techniques, the data can be decoded without any errors and is the primary focus of this work.

Lossless compression can be used in the archival of holograms for posterity. While allowing for errors can significantly increase the compressibility of many signals, lossless compression is

#502545 https://doi.org/10.1364/OE.502545
Journal © 2023 Received 3 Aug 2023; revised 13 Sep 2023; accepted 14 Sep 2023; published 31 Oct 2023

Optics EXPRESS

applicable when data fidelity is paramount. For many sensitive applications, lossy compression may introduce inherent risks that are unacceptable in the data processing pipeline. In biomedical imaging use cases, lossy techniques may smooth over small structures present in tissue samples, which oftentimes is the target of measurement. Due to the proliferation of high-resolution displays, compression techniques are also being deployed on data transport channels used in display interfaces [6]. Here they can effectively increase the resolution and/or bit-depth than what is possible without it, all while consuming less power. However, there should not be any visible quality loss which can only be guaranteed with lossless compression. Lossless compression also attracts interest as a technique to increase the data transfer capabilities between processing units in bandwidth-heavy applications [7,8]. Over the past decades, the raw compute performance of CPUs/GPUs, etc., has improved much faster than the data bandwidth available to them [9,10]. For example, when comparing the flagship NVIDIA GPUs between 2014-2022, single-precision floating-point operations that could be performed per second improved by a factor of 6.9×, while the memory bandwidth of its RAM (random access memory) only improved by a factor of 1.8× [11,12]. In this context, a lightweight compression scheme can be beneficial, increasing the available bandwidth while adding (local) operations between two points. Such schemes can be applicable at different memory hierarchy levels, where voluminous data transfer can take place [8]. Due to these reasons, designing efficient lossless compression algorithms for digital holography can improve its viability as a more accessible technique across application modalities.

Now we give a brief overview of what lossless compression for multimedia content typically entails. Given a signal expressed as a random variable, the mathematical lower bound of the achievable compression rate (entropy) is related to its probability distribution [13] - a signal that can be predicted with high certainty will have low entropy and be efficiently coded with entropy coding techniques like arithmetic coding [14,15] and Huffman coding [16]. The most efficient lossless compression mechanisms, as seen for various types of multimedia, utilize a two-step approach [17]. In the first step, the data to be compressed is initially decorrelated in some manner into a low entropy representation. The second step involves applying entropy coding techniques on the decorrelated data.

For decorrelation, two approaches are in use. The first is transform-based coding - in which, for encoding, the input data is transformed into a domain where most of the signal energy is contained in a few coefficients. Transform coding is particularly effective for lossy compression, as many of the low energy coefficients can be simply skipped without affecting the perceptual fidelity of the data. For compressing holographic content, applying an STFT (short-time Fourier transform) followed by quantization with an adaptive bit-depth and range was shown to bring significant gains over any known lossy compression solution [18,19].

In the case of lossless coding of natural imagery, transform-based coding offers noticeable gains when compared to just zero-order entropy coding and has been used for defining the lossless parts of popular image codecs. For example, JPEG 2000 utilizes a reversible Le Gall 5/3 wavelet transform with the Embedded Block Coding and Optimized Truncation (EBCOT) scheme being driven by a context-adaptive binary MQ-coder for lossless compression [20]. Integer Fresnel transforms, which are reversible propagation operators that can be implemented using Fast Fourier Transforms (FFTs), were also proposed for the lossless compression of holograms [21]. Here, the hologram is initially propagated to an object plane such that most of its energy is contained in a few spatial regions. Then, it is losslessly compressed with JPEG 2000 using a R/I representation, improving the bitrate noticeably. A quincunx embedded zero-tree wavelet coder was proposed in [22] for the lossless compression of inline digital holograms with support for progressive transmission.

Local neighborhood (context) based predictors represent the other category of decorrelators and can be geared towards lightweight compression solutions. In such mechanisms, the image is en/decoded in a pixel-by-pixel manner, such that context information available from the previous

Optics EXPRESS

pixels is utilized and only the non-redundant information of the current pixel (i.e., the pixel being coded)is stored. To keep the computational load low, only a few neighboring pixels of the current pixel are explicitly used for compression. In JPEG-LS, the current pixel is predicted as one of its neighbours using a median edge style filter. The residuals are subject to context modeling with either Golomb-rice or run-length coding [23]. Some compression solutions, like HEVC Intra mode used for compressing still images, utilize transforms for lossy compression but forgo them for lossless compression [24]. Instead, HEVC Intra solely utilizes prediction modes leveraging information from similar-looking regions in the same image, where it can even orient regions angularly. The prediction residual is afterwards subject to the CABAC (context-adaptive binary arithmetic coding procedure) procedure. JPEG XL [25], the latest image compression standard from JPEG, also proposes using context-based lossless compression. It utilizes multiple predictors, each designed to be efficient for commonly encountered image structures. In the case of lossless compression, a weighted average of these predictors is obtained as the final prediction. For losslessly compressing binary holographic data, [26] proposed to use a quad-tree decomposition to split the data into stationary regions and then use a Markovian context model-based arithmetic coder. Markovian models generally permit a wide class of stationary signals. However, as the signal becomes more multi-valued, i.e. as its bit precision increases, Markovian models become not only unfeasibly large to store in memory but also will exhibit poor compression performance due to a slow convergence speed.

To compress such multi-valued holographic signals, we propose to use an alternative context-based compression strategy called linear predictive coding that constrains the context model to be linear. Here, the prediction is obtained as a weighted sum of the context elements, i.e. as an autoregressive prediction. Linear predictive coding was first proposed in [27] for compressing 1D audio signals and is still used. The state-of-the-art FLAC lossless audio codec utilizes autoregressive prediction with up to 32 pre-determined weights. These weights can be either constant, i.e., a general set of weights applicable for any audio signal, or statically defined. The motivation here is many types of sounds, like vocals, can be described as impulses passing through a linear filter and hence be efficiently decorrelated by such models. For losslessly compressing holograms, 1D linear predictive coding based on the directionality of interference patterns of a hologram was proposed in [28]. They decompose a hologram into smaller blocks and determine the best-fit direction for the fringes present in each block from 8 preset directions (-60°, -45° , -30°, 0°, 30°, 60° and 90°) based on an entropy minimization criterion. After determining the best direction and from it the corresponding predictive 1D model, the prediction residuals are compressed by applying Huffman coding. A key limitation in [28] is the 1D model cannot perform decorrelation orthogonally to the chosen direction. Improved compression can be obtained by utilizing 2D autoregressive models, as they can decorrelate both directions jointly, as proposed in this work.

Additionally, the methods used to determine the model weights do not necessarily transfer from 1D to 2D. For example, the weight determination method used in [28] and the publicly available encoder of FLAC available from Xiph.com utilize the Levinson-Durbin recursion, which is not applicable for 2D models.

For linear predictive coding with a causal 2D autoregressive model, one is interested in a non-symmetric half-plane (NSHP) support [29], which does not place any restriction on the neighbour positions. Most of the 2D autoregressive models proposed in the context of image processing, particularly texture analysis and synthesis [30], are based on Gaussian-Markov random field models [31]. These models cannot be NSHP as their construction requires the use of a symmetrical neighbour set. For lossless [32] and lossy compression [33] of medical images, parametric 2D NSHP autoregressive models have also been proposed. Here, they assume the autoregressive polynomial equation to be from a specific parametric class from which two templates are generated - one having six neighbours and three free parameters, while the other has

Optics EXPRESS

11 neighbours and four free parameters. The parameter estimates are obtained by the recursive least squares (RLS) algorithm, updated by scanning through the image on a pixel-by-pixel. One advantage of the construction is that the autoregressive polynomial is stable. This is beneficial when lossy compression is applied on the residuals but not necessary for lossless compression. On the other hand, being constrained to a subset of all possible weights may exclude solutions that would have performed better.

Instead, we propose more generalized approaches applicable to determine the weights of any given 2D neighbour positions. With the weight estimation strategies proposed in this work, we show that in the case of holograms, it is possible to attain nearly the same compression performance as the weights that produce predictions with the minimal least square error (L2 error) but with significantly less computation. Moreover, the compression framework discussed in section 3 has been designed and parameterized from the ground up for holographic content. It features a simple decoding procedure that is essentially comprised of only two stages. The first stage utilizes an autoregressive model to predict the signal, while the second step performs zero-order entropy coding using arithmetic coding on the residual data. An ablation study of the framework is conducted in section 4.1. Compression benchmarks with respect to conventional image codecs are conducted on a wide corpus of holograms in section 4.2.

Our compression framework based on 2D autoregressive modeling obtained the highest compression performance of any known solution for all tested non-binary holograms. Moreover, we also provide a motivation about the applicability of autoregressive modeling for holograms as well as any bandlimited signal in the frequency domain, based on a novel mathematical insight, in the next section.

2. Motivation - band limitation and autoregression

The motivation for autoregressive modeling to be used in hologram compression is predicated on two propositions. The first proposition elaborates on how holograms can be viewed as bandlimited signals. The second proposition posits that efficient decorrelation can be achieved with small-sized autoregressive models for bandlimited signals.

Consider a 2D surface with a line representing a discrete 1D hologram $\boldsymbol{h}$ of length N. The hologram at position x, h_x, can be represented using the inverse DFT as shown by (1), where H_k is the 1D-DFT at position k calculated on $\boldsymbol{h}$.

$$h_x = \sum_{k=0}^{N-1} H_k \cdot e^{\frac{j2\pi kx}{N}} \tag{1}$$

Note that the Fourier transform has a physical interpretation in that each of the basis functions $e^{\frac{j2\pi kx}{N}}$, maps to the component of light received in the sub-hologram corresponding to a wave traveling in the direction represented by $(\sin^{-1}(\frac{k}{N}), \cos^{-1}(\frac{k}{N}))$. A typical hologram (chosen at an appropriate scale) is composed of only K dominant plane waves given by $\boldsymbol{K}$, depending on the geometrical position of the sub-hologram with respect to the objects emitting light in the scene; see Fig. 1 for visualization and examples. Therefore, the hologram can be then written as:

$$h_x = \boldsymbol{H} \cdot \boldsymbol{P}(x) = \sum_{k \in \boldsymbol{K}} H_k \cdot e^{\frac{j2\pi kx}{N}} \tag{2}$$

Using (2) we can write K equations, as shown by (3).

$$\boldsymbol{h}^{\text{obs}} = \boldsymbol{V} \cdot \boldsymbol{He} \cdot \boldsymbol{P}(x) \tag{3}$$

Optics EXPRESS

where $\boldsymbol{h}^{\text{obs}}$ is a vector containing the past observations, $\boldsymbol{V}$ is a Vandermonde matrix and $\boldsymbol{He}$ is a diagonal matrix as given in (4).

$$\boldsymbol{h}^{\text{obs}} = \begin{bmatrix} h_{x-1} \\ h_{x-2} \\ \vdots \\ h_{x-K} \end{bmatrix}, \boldsymbol{V} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{\frac{-j2\pi k_1}{N}} & e^{\frac{-j2\pi k_2}{N}} & \cdots & e^{\frac{-j2\pi k_K}{N}} \\ \vdots & & & \\ e^{\frac{-j2\pi k_1 K}{N}} & e^{\frac{-j2\pi k_2 K}{N}} & \cdots & e^{\frac{-j2\pi k_K K}{N}} \end{bmatrix}, \boldsymbol{He} = \text{diag}\left[H_{k_1} e^{\frac{-j2\pi k_1}{N}} \quad H_{k_2} e^{\frac{-j2\pi k_2}{N}} \quad \cdots H_{k_K} e^{\frac{-j2\pi k_K}{N}} \right] \tag{4}$$

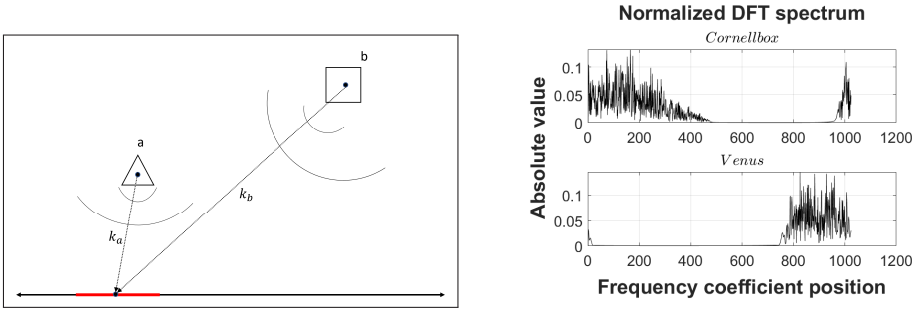


Fig. 1. Visual aid on the left side shows that depending on scene geometry, only a few dominant plane waves clustered around k_a and k_b are received by a sub-hologram (shown in red) chosen at an appropriate scale. The normalized DFT spectrum from randomly selected lines of size 1024 from representative holograms is shown on the right side, indicating bandlimited behavior.

Using (2) and (4), we can write the current observation h_x as

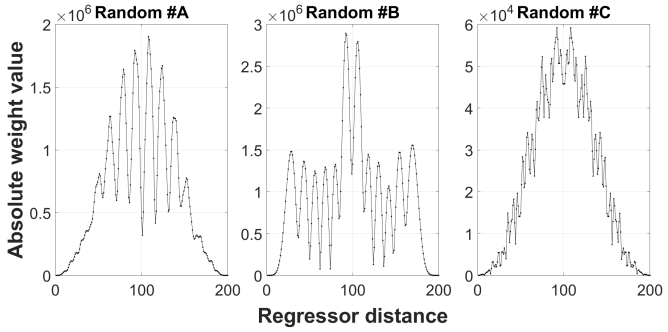
$$\begin{aligned} h_x &= \boldsymbol{H} \cdot \boldsymbol{He}^{-1} \cdot \boldsymbol{V}^{-1} \cdot \boldsymbol{h}^{\text{obs}} \\ &= \boldsymbol{w} \cdot \boldsymbol{h}^{\text{obs}} \quad \text{where} \quad w = \left[e^{\frac{j2\pi k_1}{N}} \quad e^{\frac{j2\pi k_2}{N}} \quad \cdots \quad e^{\frac{j2\pi k_K}{N}} \right] \cdot \boldsymbol{V}^{-1}. \end{aligned} \tag{5}$$

Only if $e^{\frac{-j2\pi k_a}{N}} = e^{\frac{-j2\pi k_b}{N}}$ for $a \neq b$, the Vandermonde matrix is not invertible, which will not be the case for DFT basis functions. Thus any current instance of any periodic signal with K non-zero DFT coefficients can be written as a linear sum of K previous instances multiplied by some weighting factor, i.e., it can be written as an autoregression with zero error.

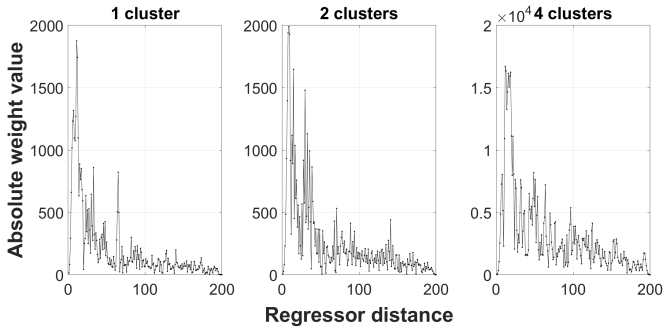
More remarkably, the result in (5) is constructive and implies that for the same frequency positions, the weights are independent of the value of DFT at these positions and can be calculated directly from the DFT positions as $\boldsymbol{w} = \left[e^{\frac{j2\pi k_1}{N}} \quad e^{\frac{j2\pi k_2}{N}} \quad \cdots \quad e^{\frac{j2\pi k_K}{N}} \right] \cdot \boldsymbol{V}^{-1}$. The same was also validated in our testing, where (5) was observed to hold independent of the statistical distribution of the DFT values at the activated frequencies and the weight estimation method used. Now we come to the second part of the motivation, where we utilize (5) to contrast two cases, (a) one when the frequency coordinates are uniformly and randomly distributed versus (b) when they are clustered in contiguous positions, i.e., bandlimited signals. The weights obtained for signals with $K = 200$ non-zero DFT positions for these two different classes are shown in Fig. 2. When the signal belongs to the first class where the activated coordinates are random, the weight coefficients tend to have the same magnitude across positions.

In contrast, for the bandlimited case, it can be seen that the regressors that are close to the current pixel have significantly larger magnitudes, where the regressor values taper away with distance. We compare the prediction (perfect decorrelation) between this linear model of size K for the bandlimited case with another one that shares the same weights, except the furthest

Optics EXPRESS



(a)



(b)

Fig. 2. Absolute values of auto-regressive weight (Y-axis) obtained by (5) as a function of regressor distance (X-axis) for two classes of 1D signals both having size $N = 1024$ and $K = 200$ non-zero N point DFT coefficients (a) Uniformly and randomly selected non-zero DFT positions, where Random #A, Random #B and Random #C are three instantiations of these K positions. (b) Bandlimited signals where the non-zero DFT positions are chosen as clusters of contiguous positions. In 1 cluster, 2 clusters and 4 clusters, the positions of the N-point DFT were chosen were {800:999}, {500:599, 900:999} and {500:549, 600:649, 700:749, 900:949} respectively. It can be seen that significant valued weights are distributed among the closest regressors in the bandlimited cases.

regressors are set to zero and posit that their predictions are close. This is because the prediction difference arises from the multiplicative sum involving regressor values with smaller absolute values. Alternatively formulated, by utilizing an autoregressive model with only a few close neighbors, we can decorrelate a bandlimited signal efficiently. See Fig. 3 for the Monte-Carlo investigation of this claim, where the SNR of the prediction error is obtained as a function of regression length for these two cases. For this experiment, the real and imaginary components of the non-zero DFT coefficients were selected from independent and identical zero-mean Gaussian distributions with unit variance. This selection is shown because typical holograms tend to have a Gaussian distribution profile for their DFT coefficients.

It can be seen that the L2 prediction error for the randomly distributed cases vanishes to zero only when the number of regressors is approximately K, while for the bandlimited cases, it vanishes at point $\ll K$. While we have shown the predictive performance for Gaussian-distributed coefficients, it was also observed to hold true for other distributions like uniform, Laplacian, etc. This is to be expected since (5) holds independent of the value of the coefficients.

Note that this discussion does not straightforwardly extend to the 2D domain. When solving the weights in the 2D domain, it can be shown that if they exist, they are also independent of the value of the non-zero DFT positions. However, in this case, the Vandermonde matrix is

Optics EXPRESS

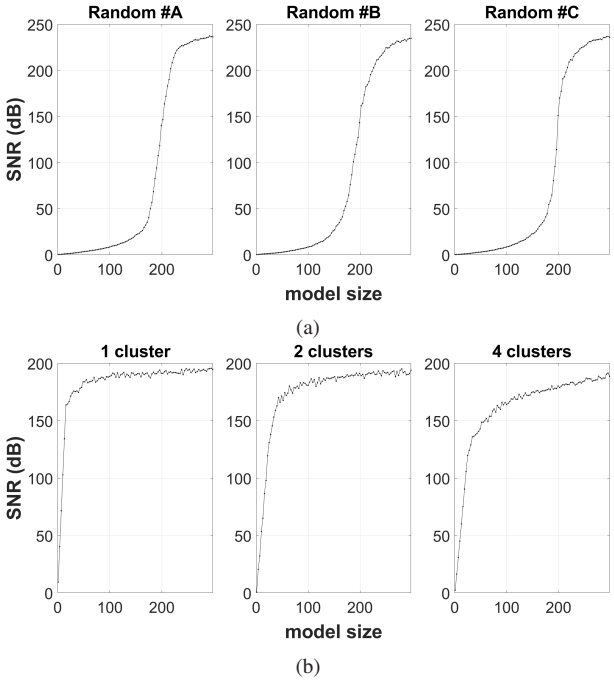


Fig. 3. Monte-Carlo validation of proposition that bandimited signals are efficiently decorrelated by autoregressive models. The L2 prediction error measured by SNR (Y-axis) is shown as a function of the model size (X-axis) for the 2 classes of signal in Fig. 2(a) Random #A, Random #B and Random #C with uniformly and randomly selected non-zero DFT positions (b) Bandlimited signals with 1 cluster, 2 clusters and 4 clusters of contiguous non-zero DFT positions. The real and imaginary values of the DFT coefficients were selected from independent and identical zero-mean Gaussian distributions with unit variance. The figures show the average SNR of 50 runs. Observe that for the random case, SNR transitions to low prediction error (decorrelation) only with model size ≈ 200, while SNR transitions for lengths $\ll 200$ in the bandlimited case.

not necessarily invertible. Consider a simple counter-example for a signal with two activated 2D-DFT coordinates $\{\{k_1, k_2\}, \{k_3, k_4\}\}$ for a 2D signal of size $N \times N$. We use a causal template with two immediate neighbors along each axis and attempt to solve (3) for the 2D case in (6). The 2×2 square matrix is not invertible if $k_1 + k_4 = k_2 + k_3$.

$$\begin{bmatrix} h_{y-1,x} \\ h_{y,x-1} \end{bmatrix} = \begin{bmatrix} H_{k_1,k_2} e^{-\frac{j2\pi k_1}{N}} & H_{k_3,k_4} e^{-\frac{j2\pi k_3}{N}} \\ H_{k_1,k_2} e^{-\frac{j2\pi k_2}{N}} & H_{k_3,k_4} e^{-\frac{j2\pi k_4}{N}} \end{bmatrix} \cdot \begin{bmatrix} e^{\frac{j2\pi(k_1 x + k_2 y)}{N}} \\ e^{\frac{j2\pi(k_3 x + k_4 y)}{N}} \end{bmatrix} \tag{6}$$

Nevertheless, 2D compression frameworks are more powerful as they enable us to decorrelate the hologram across both dimensions jointly, which leads to demonstrably higher compression performance. The hologram compression framework used in this work is now presented.

3. Autoregressive lossless hologram compression

We start with a given hologram of $N_1 \times N_2$ pixels divided into tiles where a tile has a size of $T_1 \times T_2$ pixels as shown by the left side of Fig. 4. Each of these tiles can be encoded and decoded independently. The encoding and decoding steps performed for a 2D hologram tile $\boldsymbol{h}$ of size $T_1 \times T_2$ are now described. A pixel belonging to row y and column x is denoted as $h_{y,x} = h_{\boldsymbol{x}}$.

Optics EXPRESS

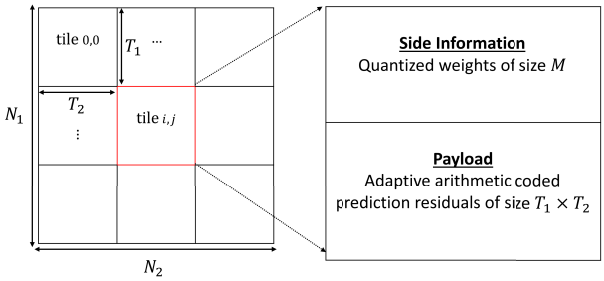


Fig. 4. Data organization of a hologram with $N_1 \times N_2$ pixels into tiles of size $T_1 \times T_2$ pixels. Each tile is assigned its own causal autoregressive model of size M, which is stored after quantization (side information). Residuals obtained after subtracting each of the $T_1 \times T_2$ pixels in the tile from their corresponding model prediction are stored after applying arithmetic coding (payload).

Each tile is assigned its own 2D causal autoregressive model represented by weights $\boldsymbol{w}$ of size M. For predicting any given pixel with the autoregressive model, we use as regressors (previous instances) all pixel neighbours whose row and column positions are both within a D pixel distance to the row and column positions of the current pixel. The row and position offset $\boldsymbol{o}$ of all these neighbours with respect to the current pixel is contained in $\boldsymbol{O}$, also known as the 2D template of the model, as given below.

$$\boldsymbol{O} = \left\{ \{o_y, o_x\} \mid o_y \in \{0:D\}, o_x \in \left\{ \begin{array}{lll} \{-D:D\} & , & \text{if} o_y > 0 \\ \{0:D\} & , & \text{if} o_y = 0 \end{array} \right\} \right\} \tag{7}$$

The model size, i.e. the number of regressors M is related to the distance D as $M = 2D \cdot (D+1)$. By taking the multiplicative sum of the weights of the autoregressive model and the neighbouring pixel values, the encoder obtains the prediction of each of the pixels $h_{y,x}$ belonging to a tile, as also shown in Fig. 5. Since neighbouring pixels that have already been decoded are exclusively utilized in this prediction, the decoder can make the same prediction for $h_{y,x}$ if it has access to the same model weights. Thus, for losslessly transmitting the pixel $h_{y,x}$ to the decoder, only the residual obtained after subtracting the pixel from its corresponding prediction has to be stored after applying adaptive arithmetic coding. If the model successfully decorrelates the hologram, then the entropy of the residuals is lower than that of the original hologram pixels, resulting in effective compression. The finally stored entities for each tile, as shown by the right side of Fig. 4 are

- **Side information** - The M weights of the autoregressive model after undergoing quantization.
- **Payload** - The $T_1 \times T_2$ prediction residuals of each pixel belonging to the tile after undergoing adaptive arithmetic coding.

This compression system is primarily parameterized by the model size M, the tile size $T_1 \times T_2$ and the quantization parameters used for the model weights. Large model sizes can potentially provide better decorrelation but will induce a bitrate penalty by increasing the number of the weights that must be stored. For fixed parametric models like ours, the template's appropriate size is matched to the stochastic complexity [34] of the data and the length of the sequence being compressed. It can be shown that the lower bound of the penalty of incrementing the model size is approximately $\frac{logN}{N}$ [34], where N is the length of the sequence for any given fixed parametric

Optics EXPRESS

model. Thus, the reduction in entropy due to the larger model must be larger than the penalty to justify the increase in complexity and needs to be adapted to obtain the best compression performance possible. A detailed analysis of the influence of tile size, model size and weight quantization bitdepth on compression performance is given in section 4.1.

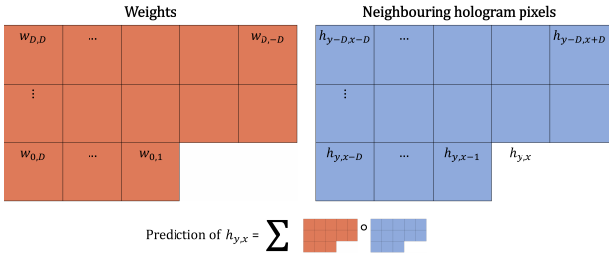


Fig. 5. Visualization of the autoregressive prediction obtained for a pixel $h_{y,x}$ by using the 2D template in (7). The prediction is obtained as the multiplicative sum of the model weights with the neighbouring pixels in the 2D template.

The compression steps for the encoder and decoder are now elucidated. The sequence of steps to be performed at the encoder as shown in Fig. 6(a) are:

E.1 weight determination using some specified causal template;

E.2 weight quantization using the MRQ (mid-rise quantizer);

E.3 prediction of each pixel from neighbors using the causal autoregressive model with the dequantized weights;

E.4 fixed-point arithmetic encoding of the prediction residual.

The quantized weights and the bitstream from the arithmetic coder constitute the codestream. The decoder shown in Fig. 6(b), performs the following steps in sequence:

D.1 dequantizing the quantized weights stored in the codestream to obtain the same model as the encoder;

D.2 fixed-point arithmetic decoding of the bitstream in the file to recover the prediction residual of the current pixel;

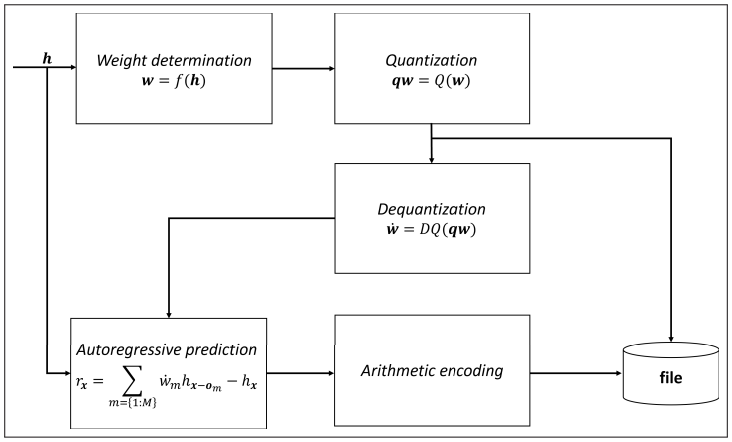
D.3 recovering the current pixel as the additive sum of the residual and the prediction obtained from the previous pixels using the autoregressive model.

Step E.1 is discussed in section 3.1. Steps E.2 & D.1 are discussed in section 3.2, while steps E.3 & D.3 and steps E.4 & D.2 are in sections 3.3 and 3.4 respectively.

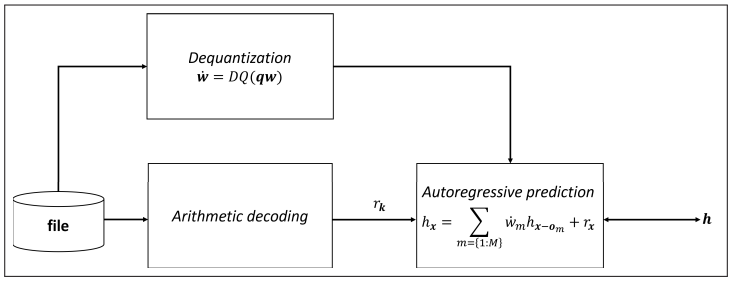
3.1. Weight determination

The most suitable weights for a given predictive model (neighbor positions) are determined in this step. Various methods exist that obtain weights with different levels of prediction accuracy and associated tradeoffs in computational complexity. Note that the weights must be determined before the subsequent prediction and residual encoding steps can occur. Now we look at some techniques that may be used to estimate the weights.

Optics EXPRESS

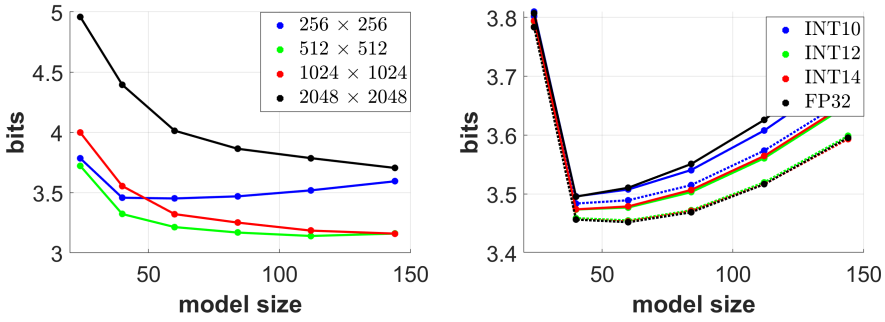


(a) Encoder (overview)



(b) Decoder (overview)

Fig. 6. Overview of the autoregressive model-based compression framework proposed in this work.



(a) bitrate without weight transmission

(b) bitrate with weight transmission

Fig. 7. Relationship between model size (X-axis), tile size, and quantization on bitrate (Y-axis). The figure on the left shows bitrates for different tile sizes without accounting for weight quantization and assumes weights are known at the receiver. The figure on the right shows bitrates for a tile size of 512×512 for different levels of quantization precision and weight transmission. For a given color (quantization precision), the dashed line shows the bitrate contribution of the payload, while the solid line shows the total bitrate. The gap between these lines represents the bitrate contribution of the weights.

Optics EXPRESS

3.1.1. Ordinary least squares (OLS)

In this method [35], available instances of the observation (current pixel) and its corresponding regressor observations (previous pixels) are expressed as shown below.

$$\boldsymbol{h}^{\text{obs}} = \boldsymbol{h}^{\text{reg}} \cdot \boldsymbol{w}^{\mathsf{T}} + \boldsymbol{r}$$

$$\begin{bmatrix} \vdots \\ h_{\boldsymbol{x}} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & & & & \\ h_{\boldsymbol{x}-\boldsymbol{o}_1} & \cdots & h_{\boldsymbol{x}-\boldsymbol{o}_m} & \cdots & h_{\boldsymbol{x}-\boldsymbol{o}_M} \\ \vdots & & & & \end{bmatrix} \cdot \begin{bmatrix} w_{\boldsymbol{o}_1} \\ \vdots \\ w_{\boldsymbol{o}_m} \\ \vdots \\ w_{\boldsymbol{o}_M} \end{bmatrix} + \begin{bmatrix} \vdots \\ r_{\boldsymbol{x}} \\ \vdots \end{bmatrix} \tag{8}$$

The weights $\boldsymbol{w}$ that minimize the L2 norm of the residuals $\boldsymbol{r}$ as given in (9) are found. If the residuals are Gaussian distributed, then the OLS estimator is the same as the MLE(maximum likelihood estimator).

$$\boldsymbol{w} = \text{argmin}_{\boldsymbol{w}} \sum\nolimits_{\boldsymbol{x}} \left| r_{\boldsymbol{x}} \right|^2 = \text{argmin}_{\boldsymbol{w}} \sum\nolimits_{\boldsymbol{x}} \left| h_{\boldsymbol{x}} - \sum_{m=1}^{M} w_m \cdot h_{\boldsymbol{x}-\boldsymbol{o}_m} \right|^2 \tag{9}$$

The unique solution(or one of the non-unique solutions) will be given by $\boldsymbol{w}^{\mathsf{T}} = (\boldsymbol{h}^{\text{reg}})^{+}\boldsymbol{h}^{\text{obs}}$, where $(.)^{+}$ is the pseudoinverse operator [36]. The Appendix also describes a simple proof for the claim that the OLS is one of the minimizing solutions. In our work, the ordinary least squares solution is found as $\boldsymbol{w} = mldivide(\boldsymbol{h}^{\text{reg}}, \boldsymbol{h}^{\text{obs}})$ with MATLAB.

If we were to use all the available information in the hologram tile to find the weights, this would result in an observation matrix $\boldsymbol{h}^{\text{obs}}$ of size $T_1T_2 \times 1$ and a regressor matrix $\boldsymbol{h}^{\text{reg}}$ of size $T_1T_2 \times M$. Solving the pseudoinverse would then have a computational complexity of around $O(T_2^2 \cdot T_1^2 \cdot M)$, making computation impractical for reasonably sized tiles. Thus, only $S \times 1$ randomly selected rows from observations $\boldsymbol{h}^{\text{obs}}$ and their corresponding $S \times M$ regressor values $\boldsymbol{h}^{\text{reg}}$ are used. S is expressed in the form of a percentage called sampling rate (%), which is defined as

$$\text{Sampling rate} = \frac{S \times 100}{T_1T_2} \tag{10}$$

Only a small percentage of the total available equations are used for weight determination. Furthermore, we exclude pixels near the edge of the tile whose regressors are outside the tile.

3.1.2. Yule Walker equations

The Yule-Walker equations [37] are an alternative formulation for determining the weights by obtaining the autocorrelation function of the hologram. Assuming a zero-mean wide-sense statistically stationary signal, the autocorrelation of a signal measures its statistical correlation with itself at different lags $\boldsymbol{o}$ and is defined in terms of its mathematical expectation as

$$\tau_{\boldsymbol{o}} = \mathbb{E}(h^*_{\boldsymbol{x}} \cdot h_{\boldsymbol{x}+\boldsymbol{o}}) \tag{11}$$

The Yule-Walker equations shown in (12) relate the autocorrelation function and the autoregressive weights by assuming the current observation is obtained as an additive sum of previous

Optics EXPRESS

LTI-filtered observations of size M and uncorrelated process noise.

$$\tau = \eta \cdot \boldsymbol{w}^{\mathsf{T}}$$
$$\begin{bmatrix} \tau_{\boldsymbol{o}_1} \\ \vdots \\ \tau_{\boldsymbol{o}_m} \\ \vdots \\ \tau_{\boldsymbol{o}_M} \end{bmatrix} = \begin{bmatrix} \tau_{\boldsymbol{o}_1-\boldsymbol{o}_1} & \cdots & \tau_{\boldsymbol{o}_1-\boldsymbol{o}_m} & \cdots & \tau_{\boldsymbol{o}_1-\boldsymbol{o}_M} \\ \vdots & & & & \\ \tau_{\boldsymbol{o}_m-\boldsymbol{o}_1} & \cdots & \tau_{\boldsymbol{o}_m-\boldsymbol{o}_m} & \cdots & \tau_{\boldsymbol{o}_m-\boldsymbol{o}_M} \\ \vdots & & & & \\ \tau_{\boldsymbol{o}_M-\boldsymbol{o}_1} & \cdots & \tau_{\boldsymbol{o}_M-\boldsymbol{o}_m} & \cdots & \tau_{\boldsymbol{o}_M-\boldsymbol{o}_M} \end{bmatrix} \begin{bmatrix} w_{\boldsymbol{o}_1} \\ \vdots \\ w_{\boldsymbol{o}_m} \\ \vdots \\ w_{\boldsymbol{o}_M} \end{bmatrix} \tag{12}$$

In general, the Yule-Walker equations do not necessarily lead to optimal solutions. However, they are widely used in practice because they enable lower complexity, especially in those cases where OLS obtains only a marginally smaller L2 error and does not offer any practically significant advantage, like improved compression performance. Given the autocorrelation function, one can then obtain the weights using (12) as $\boldsymbol{w}^{\mathsf{T}} = (\sigma)^{-1}\tau$, where $(.)^{-1}$ is the matrix inverse operator. In this work, the weights are obtained using $\boldsymbol{w}^{\mathsf{T}} = mldivide(\sigma, \tau)$ in MATLAB, which has complexity $O(M^3)$. Alternatively, had we used a 1D template, σ is a Toeplitz matrix and can be inverted using the Levinson recursion [38] with complexity $O(M^2)$. However, this procedure is not used in this work as it is numerically more sensitive to errors and not applicable to the 2D case.

There are myriad ways in which one could obtain the autocorrelation function [39]. Firstly, the autocorrelation at an offset $\boldsymbol{o}$ could be obtained by directly using (11) on measurements with known regressors as

$$\tau_{\boldsymbol{o}} = \bar{\sum} h^*_{\boldsymbol{x}} \cdot h_{\boldsymbol{x}+\boldsymbol{o}} \text{ where} \bar{\sum}(.) \text{ returns the arithmetic mean} \tag{13}$$

This requires a complexity of $O(T_1T_2M^2)$, and is an efficient way to compute the autocorrelation function when M is low. Here, the values of $h_{\boldsymbol{x}+\boldsymbol{o}}h^*_{\boldsymbol{x}}$ where $h_{\boldsymbol{x}+\boldsymbol{o}}$ lies outside the given hologram tile is skipped in the computation, which makes the estimate **unbiased**. However, this usually results in poorer estimates in the mean squared error sense. One solution would be to use all available measurements and set $h_{\boldsymbol{x}+\boldsymbol{o}}$ as zero (**zero-padding**) if they are outside the tile. Alternatively, the Wiener-Kinchine theorem [40], stating that a signal's power spectral density is obtained as the Fourier transform of its autocorrelation function, can also be utilized as shown in (14).

$$\tau = \text{IDFT}(|\boldsymbol{H}|^2) \text{ where} \boldsymbol{H} = \text{DFT}(\boldsymbol{h}) \tag{14}$$

In the DFT domain (any dimensional), it can be shown that the autocorrelation function calculated in (14) is equivalent to the autocorrelation function calculated in (13) with periodic padding, i.e., **circular convolution**. Estimating the autocorrelation function in this manner has complexity $O(T_1T_2\log(T_1T_2))$. When M is large, it may be computationally beneficial to compute the autocorrelation function via this route than the previously mentioned unbiased and zero-padded estimates.

Furthermore, more advanced spectral estimation methods for the periodogram can be used. We note that the **Welch** [41] method of computing the periodogram (spectral estimate) is particularly effective. In this method, the signal is initially grouped into overlapping regions. Then each such region is multiplied by a window that is zero outside the region, and its periodogram is taken. The final periodogram is obtained as the average of the individual periodograms calculated for each region. We use a Hann window $\boldsymbol{G}(L_1, L_2)$, defined for a 2D rectangle of size $L_1 \times L_2$ as

$$G_{l_1,l_2}(L_1, L_2) = 0.25 \cdot \big(1 - \cos(2\pi l_1/L_1)\big) \cdot \big(1 - \cos(2\pi l_2/L_2)\big) \tag{15}$$

Optics EXPRESS

3.2. Weight quantization

The chosen weights must be transmitted as side information so the decoder can access the same predictive model. This represents the overhead of utilizing the autoregressive model. Transmitting the determined model weights at an arbitrarily high precision could cause a substantial overhead when the payload size is small. Choosing to quantize the representation can reduce the overhead. However, predictive models using such a quantized representation can be sensitive to errors, so care must be taken to ensure that the predictive model is not affected [42]. Put another way, the bitrate savings from the quantization must not exceed the increase arising from the incorrect quantized model. Detailed breakdowns of the influence of the quantization procedure on both the payload and side-information bitrate are provided in section 4.1.

We shall use a simple uniform mid-rise quantizer for this purpose. In our case, the mid-rise quantizer and dequantizer are parameterized by its bit-depth $b \in \mathbb{Z}^+$, and quantization ranges $X_{\max} \in \mathbb{C}$ and quantization offset $X_{\text{off}} \in \mathbb{C}$. Applying the quantization procedure on an input $x \in \mathbb{C}$ is given by

$$Q^{\mathbb{C}}(x, b, X^{\max}, X^{\text{off}}) = Q(\text{Re}(x - X^{\text{off}}), b, \text{Re}(X^{\max})) + j \cdot Q(\text{Im}(x - X^{\text{off}}), b, \text{Im}(X^{\max})) \text{ where}$$
$$Q(x, b, X) = \left\{ \begin{array}{lll} -2^{b-1} & , & \text{if} x < -X \\ \lfloor \frac{2^{b-1}x}{X} \rfloor & , & \text{if} -X \le x \le X \\ 2^{b-1} - 1 & , & \text{otherwise} \end{array} \right\}. \tag{16}$$

Applying dequantization on a quantized output q using its corresponding uniform mid-rise dequantizer (MRDQ) yields

$$\begin{aligned} DQ^{\mathbb{C}}(q, b, X^{\max}, X^{\text{off}}) &= DQ(\text{Re}(q), b, \text{Re}(X^{\max})) + DQ(\text{Im}(q), b, \text{Im}(X^{\max})) + X^{\text{off}} \\ \text{where } DQ(q, b, X) &= (q + 0.5)\frac{X}{2^{b-1}}. \end{aligned} \tag{17}$$

The bit-depth b is kept constant for all weight coefficients, while the quantizer parameters are found as

$$X^{\text{off}} = \left(\max_{m \in \{1:M\}} w_{\boldsymbol{o}_m} + \min_{m \in \{1:M\}} w_{\boldsymbol{o}_m} \right)/2 \quad \text{and} \quad X^{\text{off}} = \left(\max_{m \in \{1:M\}} w_{\boldsymbol{o}_m} - \min_{m \in \{1:M\}} w_{\boldsymbol{o}_m} \right)/2 \tag{18}$$

For real-valued holograms like off-axis measurements, the complex channel is omitted.

3.3. Autoregressive prediction

The autoregressive model's core prediction step is executed at both the encoder and decoder. The prediction is based on a rounded, weighted sum of the preceding pixel values followed by a clipping procedure. The clipping is performed such that the prediction has the same input range as the original signal. This guarantees that the range of the prediction residual shall be not more than or equal to twice the range of the input signal, which can be used to bind the operating parameters of the arithmetic coder. The prediction is obtained as

$$\dot{h}_{\boldsymbol{x}} = \left\{ \begin{array}{lll} \text{round}(\sum_{i=1}^{M} h_{\boldsymbol{x}-\boldsymbol{o}_i} w_{\boldsymbol{o}_i}) & , & \text{if} \boldsymbol{x} \in \textbf{regular pixels} \\ 0 & , & \text{if} \boldsymbol{x} \in \textbf{border pixels} \end{array} \right\}. \tag{19}$$

Predictions are impaired when regressors (border pixels) are unavailable and set to zero, offering little to no discernable compression improvements. Thus for simplicity, we have utilized autoregressive prediction only when all the regressors are available, which shall be termed

Optics EXPRESS

"regular pixels". Alternatively, smaller, appropriately shaped templates can deal with the special case of border pixels.

Since this is one of the two core steps to be done at the decoder, the accuracy of the computation performed will have implications for the hardware and power consumption for practical applications. The prediction step must also be identical at both the decoder and encoder, i.e., the same numerical precision must be maintained. We use a 32-bit IEEE 754 floating point representation.

3.4. Residual compression using entropy coding

The prediction residual is given by

$$r_{\boldsymbol{x}} = h_{\boldsymbol{x}} - \dot{h_{\boldsymbol{x}}} \tag{20}$$

They are to be compressed using entropy coding. Since the real and imaginary components of the residual are identically distributed, we propose to encode them as a single interleaved stream. We use fixed-point arithmetic coding [43], which en/decodes a stream of integer-valued symbols into a binary bitstream and utilizes a real-time adaptive zero-order model.

For encoding a stream of integers belonging to the set $\boldsymbol{S}$, both the encoder and decoder use a histogram vector $\boldsymbol{C}$ of the same size as $\boldsymbol{S}$. Each element in the vector corresponds to the occurrence of some integer in $\boldsymbol{S}$ encountered in the stream till that time. At the start of encoding/decoding a code block, all values in the vectors are set to one. After encoding/decoding the current integer $s \in \boldsymbol{S}$, its count is incremented as shown in (21).

$$\boldsymbol{C}_s = \boldsymbol{C}_s + 1 \tag{21}$$

This implies that the probability assigned to the occurrence of some integer s shall be $\frac{C_s}{\sum_{s\in S} C_s}$, and if it had occurred incurs a bitrate of $\approx -\log_2(\frac{C_s}{\sum_{s\in S} C_s})$ bits. For unknown prior knowledge of the probability distribution of the integers, it can be shown that initializing the counts to 1 minimizes the expected rate of the arithmetic coder. It is also a consistent estimator, i.e., as more of the stream is coded, the estimated probability converges to the actual probability. When the predictive power of the autoregressive model is high, it produces a low entropy zero-order distribution with most symbols clustered around zero, thereby leading to efficient compression. When the linear autoregressive model cannot fully "decorrelate" the hologram, there may still exist a spatial correlation between the pixels and/or cross correlation between the real and imaginary components in the residuals. In this case, adaptive context-based entropy coding [44] and vector quantization [45] may bring further reductions. However, the added complexity may not be proportional to the bitrate reductions such strategies may bring about.

The only contextualization we perform on the residual is based on its position at the border. This is achieved by creating two histogram vectors $\boldsymbol{C}^{\text{regular}}$ and $\boldsymbol{C}^{\text{border}}$ respectively. Depending on the residual status, only the corresponding histogram vector is used and updated for arithmetic coding.

The motivation stems from the fact that the entropy of any distribution is concave to its probability mass function. For any two dissimilar probability distributions $\boldsymbol{p}^a$ and $\boldsymbol{p}^b$, the entropy $H(\lambda \boldsymbol{p}^a + (1-\lambda)\boldsymbol{p}^b)$ when considering them together as a single distribution in any proportion will be greater than the entropy $\lambda H(\boldsymbol{p}^a) + (1-\lambda)H(\boldsymbol{p}^b)$ when they are considered separately.

4. Results

This section will analyze the compression performance using the autoregressive modeling framework described in this work. In section 4.1, we will analyze the influence of various stages of the compression pipeline. The compression performance is then benchmarked with state-of-the-art image codecs on a wide selection of optically acquired and computer-generated holograms in section 4.2.

Optics EXPRESS

Benchmarking compression performance for the different lossless methods can be done by comparing the size of the bitstreams produced by the respective methods. The primary metric used is the bits required to encode a R/I hologram pixel which is given by (22), also referred to as the bitrate in short.

$$\text{bitrate} = \frac{\text{bits in bitstream}}{\text{number of pixels}} = \frac{\text{bits in payload + bits in side-information}}{\text{number of pixels}} \tag{22}$$

4.1. Ablation study

In this section, we will investigate the design choices possible within the autoregressive framework. Initially, we look at a few weight estimation methods that the encoder may use. Afterward, we look at the influence of the quantitative parameters of the compression framework - the model size, the tile size and the quantization precision for the weights.

4.1.1. Weight determination method

Determining the autoregressive weights at the encoder is the most computationally expensive step in our compression framework. In section 3.1, we discussed two approaches: the Ordinary Least Squares (OLS) and the Yule-Walker equations that may be used for the weight determination problem. OLS finds the weights that minimize the L2 error of the residuals but is computationally expensive, which is mitigated by random sampling.

The Yule-Walker equations lead to a more compact weight determination problem in the form of the autocorrelation function of the hologram, as given by (12). The computational complexity here is dominated by the determination of the autocorrelation function, which can be done using a variety of methods as discussed in section 3.1.

Table 1 gives the bitrates achieved by these different weight estimation methods on a selection of sub-holograms. Overall, OLS obtains the best compression performance. As the sampling rate increases, the bitrate converges, indicating diminishing returns for the higher sampling rates. For example, when using a sampling rate of 0.5% versus 20%, the bitrate increases by less than 2%, but the number of floating point multiplications performed is reduced by a factor of around 1600.

Table 1. Bitrates obtained by the different weight estimation methods for the proposed compression pipeline on complex-valued sub-holograms of size 512×512 with 16 bits of information per pixel split equally among the R/I channels.[a]

Hologram	Ordinary Least Squares (Sampling rate)				Yule-Walker (Autocorrelation determination method)			
	0.5%	2%	10%	20%	Unbiased	Zero padded	Circular convolution	Welch
Venus	3.27	3.25	3.20	**3.20**	4.41	4.20	4.20	3.21
Deep chess	6.24	6.21	6.19	**6.19**	7.01	6.86	6.88	6.19
Cornellbox 3	5.16	5.12	5.11	**5.11**	6.18	5.97	5.98	5.12
Squirrel	7.86	7.82	7.81	**7.80**	8.15	8.14	8.14	7.80

[a]The model size M was 60, and the weight quantization bit-depth was 14 bits. For the Welch method, the Hann window $\boldsymbol{G}(256, 256)$ was used with a 50% overlap.

In the case of the Yule-Walker equations, the worst performance was when using the unbiased estimator for the autocorrelation function. The performance improves by utilizing either zero-padding or circular padding (computed via the DFT and IDFT) in the autocorrelation function estimation but still is noticeably behind densely sampled OLS. However, when computing the autocorrelation via the Welch periodogram, the performance is substantially improved, with only negligible differences over densely sampled OLS. These trends hold across different tile sizes, model sizes and holograms.

Optics EXPRESS

For the remainder of this work, we shall use OLS with a 5% sampling rate as our weight determination method.

4.1.2. Parameterization - tile and model size

The influence of adjusting various quantitative parameters for the codec is discussed in this section in detail.

First, we investigate the relation between tile size and model size on compression performance, assuming that the model weights are known at the decoder without applying quantization. As the model size increases, the obtained compression performance is shown in Fig. 7(a) for different tile sizes on the *Venus* hologram. When the model size is small, a substantial reduction in bitrate can be obtained by increasing it. However, after a certain point, the gains offered by increasing the model size do not translate into improving the compression performance. This is because the larger model size will necessarily increase the proportion of border pixels, which cannot be properly predicted, increasing the total payload size. The penalty of having a larger-than-necessary model is the greatest for the smallest tiles. On the other hand, for the same model size, the smallest tile sizes tend to have the lowest bitrate cost per predicted (regular) pixel. As tile size increases, the signal's sparsity in the frequency domain decreases due to the collection of planar waves traveling in different directions (see Fig. 1), which reduces the efficacy of the autoregressive model. When compared to smaller ones, very large tile sizes may require much larger model sizes to provide the same compression performance or obtain worse compression for the same model size, and should be avoided as well.

4.1.3. Parameterization - weight transmission and quantization bit-depth

We will now consider weight transmission and motivate the need for quantization. The bitrates achieved by increasing model size are shown in Fig. 7(b) for a tile size of 512×512 at different quantization precisions. On the one hand, as the quantization precision increases, the payload size decreases due to better prediction but only to an extent. Increasing the quantization bit-depth beyond 12 bit does not appreciably decrease the payload size(shown by dotted lines) when compared to floating point representation. On the other hand, the bit number required to represent the quantized weight rises by $(\frac{M}{T_1 \cdot T_2})$ bits per quantization bit precision.

It can be observed that the total bitrate required for the floating-point representation of the weights exceeds what is possible with fixed-point quantized versions.

Furthermore, comparing compression performance with and without weight transmission, it can be seen that by a judicious selection of parameters, the added bitrate cost of weight transmission is only marginally larger than the best possible bitrate without transmission. This means that decoding architectures, like ours, do not require adaptive determination of the weights.

4.2. Benchmarks

In this section, we compare the performance obtained by best-performing (global minima) codec parameters with weight quantization and transmission against the performance of conventional image compression tools. The search space for the codec parameters is as follows - the tile size can be a maximum of 2048×2048, while the model size is limited to be less than 500. The quantization precision can be up to 14 bits. We utilize a wide selection of optically recorded and computer-generated holograms. These holograms and bitrates obtained by the different methods are given in Table 2.

As anchor codecs, JPEG XL, JPEG 2000 and JPEG LS —discussed in section 1.— were used. HEVC and PNG were also tested for lossless compression, but they are not shown in Table 2 due to their poor compression performance. Among the anchor codecs, the best performer was JPEG XL, with JPEG 2000 and JPEG LS requiring around 6.6% and 10.9% more bits on average, respectively.

Optics EXPRESS

Table 2. Bitrates of lossless codecs on optically recorded and computer-generated holograms.[a]

Hologram					Compression solutions				
Name	Resolution (pels)	Bit-depth	Pixel pitch (µm)	Wavelength (nm)	JPEG XL	JPEG 2000	JPEG LS	AR framework	
					Bitrate (bpp)				Parameters
Venus	4096×4096	2×8	1	633	8.53	10.07	11.18	**3.14**	1024, 120, INT14
CornellBox 3 [46]	16384×16384	2×8	2	633	11.02	11.43	12.28	**4.15**	1024, 364, INT14
Chess [47]	2048×16384	2×8	3.45	532	10.91	11.57	12.56	**4.89**	2048, 480, INT14
Deep Chess [47]	2048×16384	2×8	3.45	532	13.18	13.03	13.58	**6.43**	1024, 480, INT14
Deep Cornell Box [46]	16384×16384	2×8	2	633	11.32	11.90	12.48	**4.98**	1024, 312, INT14
DeepDices2K [48]	2048×2048	2×8	4.8	640	12.25	12.29	12.70	**9.78**	128, 84, INT10
				532	12.16	12.23	12.5	**10.00**	128, 84, INT8
				473	12.37	12.42	12.74	**10.46**	128, 84, INT8
Dices4K [48]	4096×4096	2×8	0.4	640	11.04	11.76	12.05	**5.57**	512, 312, INT14
				532	11.93	12.36	12.62	**6.74**	512, 220, INT14
				473	12.53	12.76	13.14	**7.56**	512, 312, INT14
Piano8K [48]	8192×8192	2×8	0.4	640	10.31	11.33	12.01	**4.16**	1024, 420, INT14
				532	11.21	12.06	12.69	**4.90**	1024, 480, INT14
				473	11.85	12.42	13.02	**5.15**	512, 364, INT14
DiffuseCar8K [49]	8192×8192	2×8	0.4	640	10.59	11.30	12.18	**5.54**	1024, 364, INT14
				532	12.15	12.70	13.25	**6.74**	512, 312, INT12
				473	11.73	12.48	13.09	**6.38**	512, 264, INT12
SpecularCar8K [49]	8192×8192	2×8	0.4	640	9.29	10.36	11.21	**4.41**	512, 220, INT13
				532	11.29	11.96	12.65	**5.46**	512, 220, INT12
				473	11.15	11.90	12.51	**5.47**	512, 180, INT13
Ring8K [49]	8192×8192	2×8	0.4	640	11.64	12.10	12.75	**7.01**	1024, 312, INT14
				532	12.86	13.09	13.47	**8.33**	1024, 264, INT14
				473	13.37	13.52	13.77	**9.04**	1024, 220, INT13
Single Cell [50]	2048×2048	8	3.45	532	5.35	6.28	5.95	**4.76**	2048, 40, INT14
Microspheres [50]	2048×2048	8	3.45	532	6.27	6.71	6.88	**5.88**	2048, 60 INT 12
Ball_ roughness [51]	1024×1024	8	-	-	4.82	5.59	5.78	**4.11**	1024, 144, INT14
Neuron [51]	1024×1024	8	-	-	4.16	4.58	4.47	**3.72**	512, 40, INT10
MDCK_ wild_053 [52]	1024×1024	8	6.45	660	4.41	5.02	5.16	**3.53**	256, 40, INT10
MDCK_ KRasV12_211 [52]	1024×1024	8	6.45	660	4.02	4.47	4.56	**3.17**	256, 40, INT8
Sphere [53,54]	2048×16384	2×8	3.45	532.8	11.45	12.31	12.65	**7.72**	2048,480, INT14
Mermaid [53,54]	2048×16384	2×8	3.45	532.8	12.49	13.05	13.38	**7.85**	2048,480, INT14
Squirrel [53,54]	1792×26880	2×8	3.45	632.8	11.38	12.04	12.98	**7.38**	1792,480, INT14
Astronaut [55]	2048×2048	2×8	2.2	632.8	10.20	10.98	11.04	**9.72**	112, 12, INT12
Lowiczanka Doll [54]	2016×58464	2×8	3.45	637	10.53	11.42	12.11	**6.34**	2016, 480, INT14
				532	10.47	11.28	11.87	**6.65**	2016, 480, INT14
				457	10.58	11.19	11.69	**6.85**	2016, 480, INT14
Average percentual increase in bitrate over AR framework (%)					**+72.1.%**	**+83.5%**	**+92.1%**		

[a]The listed parameters for the AR framework indicate the size of the side of one squared tile, model size and weight quantization precision. File size of uncompressed hologram (bits) = Resolution × Bitdepth. File size of compressed hologram (bits) = Resolution × Bitrate.

The autoregressive framework proposed in this work achieves the best compression performance for all holograms. For computer-generated holograms, JPEG XL, JPEG 2000, and JPEG LS required around 93.4%, 104.4% and 115.4% more times the bitrate of our framework, respectively. For optically recorded content, the average percentual increases were 34.1%, 46.4%, and 50.5%.

Optics EXPRESS

It can be seen that the compression gain of the framework for optically recorded content, while significant, is still lower than the gain of computer-generated content. This is due to the presence of measurement noise in the recorded content, which is fundamentally incompressible. Among the computer-generated content, the smallest bitrates were observed for the *Venus*, which was computed analytically from a point cloud. All other computed holograms tested utilize some approximations that can be seen as an injection of noise. It can also be seen that the best-performing model quantization precision of optical content is smaller than that of computed content, suggesting — not surprisingly — that more accurate autoregressive models are less effective in the presence of strong measurement noise.

The encoding/decoding times of all the compression solutions are given in Table 3. Note that our software has not been optimized, and the numbers do not fully indicate the achievable speed. Nevertheless, leveraging parallel processing across tiles improves the encoding/decoding times.

Table 3. The encoding and decoding times in µs for a single pixel when using lossless compression software tested in section 4.2.[a]

	JPEG LS		JPEG 2000		JPEG XL		AR Model		
Hologram resolution	Enc. time	Dec. time	Enc. time	Dec. time	Enc. time	Dec. time	Enc. time	Dec. time	Processors utilized
1024 × 1024	0.32	0.33	0.34	0.36	1.88	0.44	2.67	1.81	1
2048 × 2048	0.17	0.19	0.09	0.09	1.06	0.14	1.29	0.81	4
4096 × 4096	0.14	0.13	0.03	0.03	0.96	0.08	0.84	0.48	16

[a]The autoregressive framework was used with a model size of 144 and a tile size of 1024 × 1024. The tests were run using an Intel Core i7 11800H@4GHZ equipped with DDR4-3200 RAM.

5. Conclusion

In this work, we have showcased the suitability of causal autoregressive modeling for the lossless compression of digital holograms. When a typical hologram is analyzed locally in the Fourier domain, only a few clusters of frequency coefficients are activated depending on the position of the objects illuminating the scene. As discussed in section 2, a given pixel from such a sequence can be stochastically predicted as a weighted sum of only a few previously decoded neighboring pixels, i.e.using a causal autoregressive model. The compression framework brings significant compression gain over standard image compression tools, especially for computer-generated content. Moreover, the decoder depicts only linear complexity with respect to the data volume and is highly scalable, i.e., its operating parameters can be adapted to deal with different types of holograms and desired computational footprint. Different approaches were used at the encoder to determine the autoregressive model from the given hologram. Here, we demonstrated that it is possible to attain nearly the same compression performance as the autoregressive model minimizing the L2-norm, but with less computation. We also evaluated the implications of varying the tile size, model size, and weight quantization precision on the achieved compression performance.

Optics EXPRESS

6. Appendix: OLS minimizes L2 error

This appendix gives a proof for the claim that $\boldsymbol{w} = (\boldsymbol{h}^{\text{reg}})^{+}\boldsymbol{h}^{\text{obs}}$ satisfies (9). $\boldsymbol{w}$ is expressed as a column vector for ease of notation. Consider $\boldsymbol{x} \neq \boldsymbol{w}$ and let $||\ (.)\ ||_2$ represent L2 error. Then

$$\begin{aligned} |\ |\boldsymbol{h}^{\text{reg}}\boldsymbol{x} - \boldsymbol{h}^{\text{obs}}\ ||_2 &= ||\boldsymbol{h}^{\text{reg}}\boldsymbol{w} - \boldsymbol{h}^{\text{obs}} + \boldsymbol{h}^{\text{reg}}\boldsymbol{x} - \boldsymbol{h}^{\text{reg}}\boldsymbol{w}\ ||_2 \\ &= (\boldsymbol{h}^{\text{reg}}\boldsymbol{w} - \boldsymbol{h}^{\text{obs}} + \boldsymbol{h}^{\text{reg}}(\boldsymbol{x} - \boldsymbol{w}))^{*}(\boldsymbol{h}^{\text{reg}}\boldsymbol{w} - \boldsymbol{h}^{\text{obs}} + \boldsymbol{h}^{\text{reg}}(\boldsymbol{x} - \boldsymbol{w})) \\ &= ||\boldsymbol{h}^{\text{reg}}\boldsymbol{w} - \boldsymbol{h}^{\text{obs}}\ ||_2 + ||\boldsymbol{h}^{\text{reg}}(\boldsymbol{x} - \boldsymbol{w})\ ||_2 + 2\,\text{Re}((\boldsymbol{x} - \boldsymbol{w})^{*}(\boldsymbol{h}^{\text{reg}})^{*}(\boldsymbol{h}^{\text{reg}}\boldsymbol{w} - \boldsymbol{h}^{\text{obs}}) \end{aligned} \tag{23}$$

Here

$$\begin{aligned} (\boldsymbol{h}^{\text{reg}})^{*}(\boldsymbol{h}^{\text{reg}}\boldsymbol{w} - \boldsymbol{h}^{\text{obs}}) = (\boldsymbol{h}^{\text{reg}})^{*}\boldsymbol{h}^{\text{reg}}(\boldsymbol{h}^{\text{reg}})^{+}\boldsymbol{h}^{\text{obs}} - (\boldsymbol{h}^{\text{reg}})^{*}\boldsymbol{h}^{\text{obs}} = 0 \\ \text{as } (\boldsymbol{A})^{*}\boldsymbol{A}(\boldsymbol{A})^{+} = (\boldsymbol{A})^{*} \text{for any } \boldsymbol{A} \end{aligned} \tag{24}$$

Therefore

$$\begin{aligned} |\ |\boldsymbol{h}^{\text{reg}}\boldsymbol{x} - \boldsymbol{h}^{\text{obs}}\ ||_2 &= ||\boldsymbol{h}^{\text{reg}}\boldsymbol{w} - \boldsymbol{h}^{\text{obs}}\ ||_2 + ||\boldsymbol{h}^{\text{reg}}(\boldsymbol{x} - \boldsymbol{w})\ ||_2 \\ &\geq |\ |\ \boldsymbol{h}^{\text{reg}}\boldsymbol{w} - \boldsymbol{h}^{\text{obs}}\ ||_2 \end{aligned} \tag{25}$$

Funding. Fonds Wetenschappelijk Onderzoek (12ZQ223N, G0B3521N); Japan Society for the Promotion of Science (P22752).

Disclosures. The authors declare no conflicts of interest.

Data availability. Some holograms used for evaluating lossless compression performance in this work are available in [56].

References

1. A. W. Lohmann, R. G. Dorsch, D. Mendlovic, Z. Zalevsky, and C. Ferreira, "Space–bandwidth product of optical signals and systems," J. Opt. Soc. Am. A **13**(3), 470–473 (1996).
2. P. Schelkens, A. Ahar, A. Gilles, R. Muhamad, T. Naughton, C. Perra, A. Pinheiro, P. Stepien, and M. Kujawinska, "Compression strategies for digital holograms in biomedical and multimedia applications," Light: Adv. Manuf. **3**(3), 1 (2022).
3. R. K. Muhamad, P. Stepień, D. Blinder, P. Schelkens, and M. Kujawińska, "Holographic data compression for holographic microscopy and tomography in biomedical applications," in *Imaging and Applied Optics Congress* (Optica Publishing Group, 2020), paper HTh5D.1.
4. D. Blinder, A. Ahar, S. Bettens, T. Birnbaum, A. Symeonidou, H. Ottevaere, C. Schretter, and P. Schelkens, "Signal processing challenges for digital holographic video display systems," Signal Process.: Image Commun. **70**, 114–130 (2019).
5. 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).
6. F. Walls and A. MacInnis, "VESA Display Stream Compression for Television and Cinema Applications," IEEE J. Emerging Sel. Top. Circuits Syst. **6**(4), 460–470 (2016).
7. J.-S. Lee, W.-K. Hong, and S.-D. Kim, "Design and evaluation of a selective compressed memory system," in *International Conference on Computer Design: VLSI in Computers and Processors* (1999), pp. 184–191.
8. N. R. Mahapatra, J. Liu, and K. Sundaresan, "The performance advantage of applying compression to the memory system," in *Workshop on Memory System Performance* (Association for Computing Machinery, 2002), pp. 86–96.
9. T. P. Morgan, "Compute is Easy, Memory is Harder and Harder".
10. H. Levinson, "High-NA EUV lithography: current status and outlook for the future," Jpn. J. Appl. Phys. **61**(SD), SD0803 (2022).
11. Wikipedia, "Geforce 10 series," https://en.wikipedia.org/wiki/GeForce_10_series.
12. Wikipedia, "Geforce 40 series," https://en.wikipedia.org/wiki/GeForce_40_series.
13. C. E. Shannon, "A mathematical theory of communication," The Bell Syst. Tech. J. **27**(3), 379–423 (1948).
14. P. Elias, "Universal codeword sets and representations of the integers," IEEE Trans. Inf. Theory **21**(2), 194–203 (1975).
15. J. J. Rissanen, "Generalized Kraft Inequality and Arithmetic Coding," IBM J. Res. Dev. **20**(3), 198–203 (1976).
16. D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proc. IRE **40**(9), 1098–1101 (1952).
17. P. Roos, M. Viergever, M. van Dijke, and J. Peters, "Reversible intraframe compression of medical images," IEEE Trans. Med. Imaging **7**(4), 328–336 (1988).
18. R. K. Muhamad, T. Birnbaum, D. Blinder, C. Schretter, and P. Schelkens, "INTERFERE: A Unified Compression Framework for Digital Holography," in *Digital Holography and 3-D Imaging* (Optica Publishing Group, 2022), paper Th4A.2.

Optics EXPRESS

19. R. K. Muhamad, D. Blinder, A. Symeonidou, T. Birnbaum, O. Watanabe, C. Schretter, and P. Schelkens, "Exact global motion compensation for holographic video compression," Appl. Opt. **58**(34), G204–G217 (2019).
20. D. Taubman, E. Ordentlich, M. Weinberger, and G. Seroussi, "Embedded block coding in JPEG 2000," Signal Process.: Image Commun. **17**, 49–72 (2002).
21. D. Blinder and P. Schelkens, "Integer Fresnel Transform for Lossless Hologram Compression," in *Data Compression Conference* (2019), pp. 389–397.
22. S. Belaid, J. Hattay, and M. Machhout, "Tele-Holography: a new concept for lossless compression and transmission of inline digital holograms," Signal, Image and Video Processing **16**(6), 1659–1666 (2022).
23. M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," IEEE Trans. on Image Process. **9**(8), 1309–1324 (2000).
24. M. Zhou, W. Gao, M. Jiang, and H. Yu, "HEVC Lossless Coding and Improvements," IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1839–1843 (2012).
25. J. Alakuijala, R. van Asseldonk, S. Boukortt, M. Bruse, I.-M. Comşa, M. Firsching, T. Fischbacher, E. Kliuchnikov, S. Gomez, R. Obryk, K. Potempa, A. Rhatushnyak, J. Sneyers, Z. Szabadka, L. Vandevenne, L. Versari, and J. Wassenberg, "JPEG XL next-generation image compression architecture and coding tools," Proc. SPIE **11137**, 111370K (2019).
26. R. K. Muhamad, T. Birnbaum, D. Blinder, C. Schretter, and P. Schelkens, "Binary hologram compression using context based Bayesian tree models with adaptive spatial segmentation," Opt. Express **30**(14), 25597–25611 (2022).
27. B. Atal and M. Schroeder, "Predictive Coding of Speech Signals," The Bell Syst. Tech. J. **49**(8), 1973–1986 (1970).
28. K. D. Hyun, K. Tamagawa, and Y. Sakamoto, "Lossless compression applying linear predictive coding based on the directionality of interference patterns of a hologram," Appl. Opt. **58**(18), 5018–5028 (2019).
29. B. Choi and D. N. Politis, "Modeling 2-D AR Processes With Various Regions of Support," IEEE Trans. Signal Process. **55**(5), 1696–1707 (2007).
30. J. Bennett and A. Khotanzad, "Multispectral random field models for synthesis and analysis of color images," IEEE Trans. Pattern Anal. Machine Intell. **20**(3), 327–332 (1998).
31. R. Chellappa, Y.-H. Hu, and S.-Y. Kung, "On two-dimensional Markov spectral estimation," IEEE Trans. Acoust., Speech, Signal Process. **31**(4), 836–841 (1983).
32. M. Das and S. Burgett, "Lossless compression of medical images using two-dimensional multiplicative autoregressive models," IEEE Trans. Med. Imaging **12**(4), 721–726 (1993).
33. S. Burgett and M. Das, "Predictive image coding using two-dimensional multiplicative autoregressive models," Signal Processing **31**(2), 121–132 (1993).
34. J. Rissanen, "Stochastic Complexity and Modeling," The Ann. Stat. **14**(3), 1080–1100 (1986).
35. S. Puntanen and G. P. H. Styan, "The Equality of the Ordinary Least Squares Estimator and the Best Linear Unbiased Estimator," The Am. Stat. **43**(3), 153–161 (1989).
36. M. Planitz, "Inconsistent systems of linear equations," The Mathematical Gazette **63**(425), 181–185 (1979).
37. G. U. Yule, "On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers," Phil. Trans. R. Soc. A **226**(636-646), 267–298 (1927).
38. N. Wiener, *The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction* (1964), pp. 129–148.
39. M. Hayes, *Statistical Digital Signal Processing and Modeling* (Wiley India Pvt. Limited, 2009).
40. N. Wiener, "Generalized harmonic analysis," Acta Math. **55**(0), 117–258 (1930).
41. P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," IEEE Trans. Audio Electroacoust. **15**(2), 70–73 (1967).
42. W. B. Kleijn and K. K. Paliwal, *Speech Coding and Synthesis* (Elsevier Science Inc., 1995).
43. I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," Commun. ACM **30**(6), 520–540 (1987).
44. M. Weinberger, J. Rissanen, and R. Arps, "Applications of universal context modeling to lossless compression of gray-scale images," IEEE Trans. on Image Process. **5**(4), 575–586 (1996).
45. R. Gray, "Vector quantization," IEEE ASSP Mag. **1**(2), 4–29 (1984).
46. D. Blinder, M. Chlipala, T. Kozacki, and P. Schelkens, "Photorealistic computer generated holography with global illumination and path tracing," Opt. Lett. **46**(9), 2188–2191 (2021).
47. A. Symeonidou, D. Blinder, A. Munteanu, and P. Schelkens, "Computer-generated holograms by multiple wavefront recording plane method with occlusion culling," Opt. Express **23**(17), 22149–22161 (2015).
48. A. Gilles, P. Gioia, R. Cozot, and L. Morin, "Hybrid approach for fast occlusion processing in computer-generated hologram calculation," Appl. Opt. **55**(20), 5459–5470 (2016).
49. A. Gilles, P. Gioia, R. Cozot, and L. Morin, "Computer generated hologram from Multiview-plus-Depth data considering specular reflections," (2016), pp. 1–6.
50. P. Stěpień, R. K. Muhamad, D. Blinder, P. Schelkens, and M. Kujawińska, "Spatial bandwidth-optimized compression of image plane off-axis holograms with image and video codecs," Opt. Express **28**(19), 27873–27892 (2020).
51. T. Bruylants, D. Blinder, H. Ottevaere, A. Munteanu, and P. Schelkens, "Microscopic off-axis holographic image compression with JPEG 2000," Proc. SPIE **9138**, 91380F (2014).
52. T. Pitkäaho, T. J. Naughton, and A. Manninen, "Monitoring MDCK cell vesicles by digital holographic microscopy and image processing," in *Imaging and Applied Optics* (2016).

Optics EXPRESS

53. A. Ahar, M. Chlipala, T. Birnbaum, W. Zaperty, A. Symeonidou, T. Kozacki, M. Kujawinska, and P. Schelkens, "Suitability analysis of holographic vs light field and 2D displays for subjective quality assessment of Fourier holograms," Opt. Express **28**(24), 37069–37091 (2020).
54. A. Goloś, W. Zaperty, G. Finke, P. Makowski, and T. Kozacki, "Fourier RGB synthetic aperture color holographic capture for wide angle holographic display," Proc. SPIE **9970**, 99701E (2016).
55. M. V. Bernardo, P. Fernandes, A. Arrifano, M. Antonini, E. Fonseca, P. T. Fiadeiro, A. M. Pinheiro, and M. Pereira, "Holographic representation: Hologram plane vs. object plane," Signal Process. Image Commun. **68**, 193–206 (2018).
56. "JPEG Pleno database," Available at https://plenodb.jpeg.org/.