



Binary hologram compression using context based Bayesian tree models with adaptive spatial segmentation

RAEES KIZHAKKUMKARA MUHAMAD,^{1,2,*}  TOBIAS BIRNBAUM,^{1,2} 
DAVID BLINDER,^{1,2}  COLAS SCHRETTTER,^{1,2} AND PETER
SCHELKENS^{1,2} 

¹Vrije Universiteit Brussel (VUB), Dept. of Electronics and Informatics (ETRO), Belgium

²imec, Kapeldreef 75, B- 3001 Leuven, Belgium

*raees.kizhakkumkara.muhamad@vub.be

Abstract: With holographic displays requiring giga- or terapixel resolutions, data compression is of utmost importance in making holography a viable technique in the near future. In addition, since the first-generation of holographic displays is expected to require binary holograms, associated compression algorithms are expected to be able to handle this binary format. In this work, the suitability of a context based Bayesian tree model is proposed as an extension to adaptive binary arithmetic coding to facilitate the efficient lossless compression of binary holograms. In addition, we propose a quadtree-based adaptive spatial segmentation strategy, as the scale dependent, quasi-stationary behavior of a hologram limits the applicability of the advocated modelling approach straightforwardly on the full hologram. On average, the proposed compression strategy produces files that are around 12% smaller than JBIG2, the reference binary image codec.

© 2022 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

1. Introduction

In the last decades, technological advances in the field of optical nano-electronics and the exponential growth in computing power have increased the viability of holographic displays. Particularly binary holographic displays appear to be a viable short-term solution for high-end display. While the reduction in the number of the levels implies a loss of information [1], non-linearities in the output medium make greyscale modulation difficult [2,3]. For example, the first wave of computer generated holography (CGH) algorithms [4–6] employed binary modulation, as only microscopic printing of dots was possible in computer guided plotters and laser printers. These computer generated holograms offered not only a theoretically and experimentally verifiable higher signal-to-noise ratio (SNR) over analog equivalents at the time, but also a much higher diffraction efficiency [7]. Binary digital micro-mirror devices (DMD) based spatial light modulators that permit thousands of refreshes per second, allowing for spatial/temporal multiplexing [8,9], were proposed for holographic displays. These techniques were used to maximize space-bandwidth product (SBP) such that they could be used in prototype tabletop displays offering 360-degree vision [10–12]. Furthermore, commercially viable devices would need gigapixel or even terapixel resolution, increasing further the signal processing costs to accommodate greyscale levels. It is interesting to mention at this point that binary compression can also be used as a component in a hologram compression scheme where a binary image is utilized by algorithms to predict a greyscale phase [13]. Though compression solutions for binary holograms can further reduce the associated data bandwidth costs, achieving efficient compression for binary holographic content remains a challenge.

Binary signals vary step-wise between the two states, which makes them unsuitable for transformation with classic sparsifying transforms such as the Discrete Cosine Transform (DCT)

or the Discrete Wavelet Transform (DWT). The lack of smoothness results in a less sparse signal that has multi-valued transform coefficients, and as a consequence will reduce compressibility of the signal. In this case, local neighborhood (context) prediction based compression techniques maybe more advantageous depending on the spatial correlation exhibited by the signal. The efficacy of utilizing spatial correlation for compression can be quantified in terms of entropy – the mathematical lower bound on the minimum number of bits needed to encode a random variable given its corresponding probability distribution [14]. As the uncertainty regarding the signal decreases, the entropy will come down. Compression techniques try to reduce this uncertainty, while using entropy coding techniques that strive to achieve the entropy rate while minimizing computational overhead.

For binary images, initial compression standards like G3/G4 [15] represented the image as a series of scanlines, where each scanline is assigned a variable length codeword using the distribution of 0/1 symbols within the scanline. While G4 also looks at vertical features to find similarities between scanlines, these techniques were not able to fully exploit the spatial correlation found in the signal. The Joint Bi-level Image Experts Group standard (JBIG) [16], introduced in 1993, with its context based “MQ arithmetic coder”, can adapt uniquely to each possible combination of a predefined region of neighboring pixels. JBIG2 [17], the state of the art bi-level image codec is similar to JBIG, but utilizes larger context models along with a more efficient QM arithmetic coder to achieve better compression efficiency.

In this paper, we will focus on an alternative context modelling approach with arithmetic coding, designed for the stochastic properties of holograms. The paper is structured as follows: in section 2 adaptive arithmetic coding with fixed template context models are discussed, while in section 3 tree-based context models for improving compression performance are described. In section 4, the limitations of these statistical modelling techniques for holograms are discussed along with ways to mitigate its compression impact. Section 5 describes the experimental results and the work is concluded in section 6.

2. Arithmetic coding using fixed template context models

2.1. Arithmetic coding

Arithmetic coders [18] belong to the family of entropy coders that can generate variable length codewords that directly exploit probability relationships for achieving the entropy of the underlying statistical model. For encoding a binary sequence, the encoding (and decoding) takes place in a per symbol manner, where the encoding is equivalent to obtaining a probability estimate ratio $\{1 - \hat{p}, \hat{p}\}$ of the current symbol to be $\{0, 1\}$. To ensure successful decoding, the probability estimate ratio used at the encoder should also be known at the decoder. As the length of the sequence tends to ∞ and the probability estimate ratio tends to the actual probability. The arithmetic coder obtains the entropy bitrate. The present work utilizes the fixed point arithmetic implementation as described in [19]. We refer to [19] for the intricacies on the practical design of arithmetic coding.

Arithmetic coders also enable separation of the signal model design from the coding process [19], allowing the development of flexible and tailor-made compression solutions. In many cases, the required probability estimate ratio is not available and adaptive arithmetic coding is required. Thereby, the probability estimate for the i^{th} symbol x_i to be 1 is calculated using the previous observations $\mathbf{x}^{i-1} = x_1 x_2 \dots x_{i-1}$ with any arbitrary function. As more of a statistically stationary signal gets processed, the model adapts better to the input statistics resulting in improved compression.

Arithmetic coding is defined for 1D data, while holograms are 2D. To serialize 2D data, the hologram is read in raster scan order, which maps some binary pixel $x_{k,l}$ belonging to row k and column l as $i = (k - 1)A + l$. If a row-wise raster scan is used and the starting point is in the top left corner, any pixel in a row above or to the left and within the same row as the current pixel is

accessible to the decoder. This knowledge can be used for adaptively determining the probability estimate. As such the causality condition is fulfilled.

2.2. Fixed template context model

The conditional entropy of a general pixel considered a random variable now and denoted by X given some k neighboring pixels X_{nei} is calculated by Eq. (1) as

$$H(X|X_{\text{nei}}) = - \sum_{x \in \{0,1\}} \sum_{x_{\text{nei}} \in \{0,1\}^k} p(x, x_{\text{nei}}) \log_2 \left(\frac{p(x, x_{\text{nei}})}{p(x_{\text{nei}})} \right). \quad (1)$$

The entropy is minimized when the pixel X can be predicted with high certainty from its neighboring pixels X_{nei} and attains the maximum value of 1 bit per pixel (bpp) when an equiprobable distribution is induced on X by X_{nei} , i.e., when 0 and 1 would be equally likely.

For a diverse set of holograms, the calculation of entropy indicates that neighboring pixels within the same color channel are correlated and can be utilized for data compression. Examples for the entropy of a pixel given its immediate neighbors are shown in Fig. 1(a) for two holograms. The holograms have equal physical dimensions and record the same CGH scene but at different pixel pitches. As it can be seen, the uncertainty regarding the pixel can be reduced significantly by utilizing the information present in neighboring pixels, where the uncertainty decreases with smaller pixel pitches. For pixels belonging to different color channels no such correlation is observed. Thus in this work, the color channels of a hologram are compressed independently. This does not necessarily imply that the different color channels are completely uncorrelated, but that no directly exploitable probabilistic relationship exists with nearby spatial neighborhoods across different color channels.

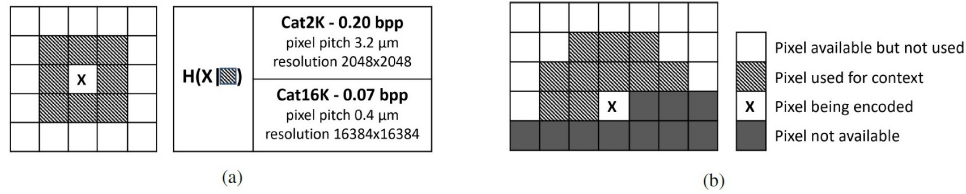


Fig. 1. (a) Entropy of a pixel given its immediate neighbors for two holograms - Cat2K (2048 × 2048 px, 3.2 μm) and Cat16K (16384 × 16384 px, 0.4 μm). Both holograms record the same CGH scene at 633 nm. Entropy obtained by averaging over the individual entropies of sub-holograms of size 512x512 from the hologram. (b) The 3L10P (3 line 10 pixel) fixed template context model used in JBIG.

Bi-level compression standards like JBIG and JBIG2 make use of the fixed template context model for adaptive arithmetic coding. Here, the probability for a specific pixel $x_{k,l}$ of a specific hologram is conditioned explicitly only with respect to a subset F of pixels in the close neighborhood of $x_{k,l}$ — available to the decoder. The number and spatial location of these neighboring pixels with respect to the current pixel being encoded is fixed as shown in Eq. (2), and is also referred to as the context in which the pixel occurs. An example for a fixed template context model used in JBIG is shown in Fig. 1(b).

$$x^{k,l,F} = [..x_{k-k',l-l'}..] \text{ for all } (k',l') \in F. \quad (2)$$

We will now discuss the probability estimation function that is used for adaptive arithmetic coding in our proposed scheme.

We chose a probability estimation function which assumes a statistically stationary signal, i.e. where the probability distribution of any pixel given its context remains the same irrespective

to its location. For a specific instance of a hologram this means, that some context $\mathbf{c} = \mathbf{x}^{k,l,F}$ of the current pixel, the probability estimate $\hat{p}(\mathbf{c})$ is estimated by using histogram counts $n(\mathbf{c})$ and $n_1(\mathbf{c})$. Here, $n(\mathbf{c})$ is how many times the context $\mathbf{c}$ has occurred until now and $n_1(\mathbf{c})$ is in how many of these occurrences the current pixel was 1. The expected rate of the arithmetic coder for an estimate e will be given by Eq. (3), where $f(p|\mathbf{c})$ is the conditional probability distribution function (PDF) of the probability of the current pixel to be 1 given the context information.

$$R(e, \mathbf{c}) = - \int_0^1 \left(\log_2(e) \cdot f(p|\mathbf{c}) + \log_2(1-e) \cdot (1-f(p|\mathbf{c})) \right) dp. \quad (3)$$

For an unknown prior distribution for the PDF, it can be shown using Bayes theorem [20] that

$$f(p|\mathbf{c}) = \frac{(n(\mathbf{c}) + 1)!}{n_1(\mathbf{c})! (n(\mathbf{c}) - n_1(\mathbf{c}))!} \cdot p^{n_1(\mathbf{c})} \cdot (1-p)^{n(\mathbf{c})-n_1(\mathbf{c})}. \quad (4)$$

Using Eq. (4) and the Beta function property $\int_0^1 t^{x-1} \cdot (1-t)^{y-1} dt = \frac{(x-1)!(y-1)!}{(x+y-1)!}$, the probability estimate that minimizes the expected rate in Eq. (3) is given by

$$\hat{p}(\mathbf{c}) = \frac{n_1(\mathbf{c}) + 1}{n(\mathbf{c}) + 2}. \quad (5)$$

After encoding or decoding the current pixel the histogram counts (initialized as 0 for all contexts) of the encountered context are updated accordingly.

This estimation method was chosen, as the analysis of a selection of holograms from different sources indicated a high degree of diversity in statistical properties, which makes design of a more tailored model difficult. The best, general statistical assumption that can be made is that for holograms is quasi-stationarity, which requires a small detour to explain.

Any full hologram may be analyzed with on a smaller scope, within which the spatial frequency changes slower than the variations of the underlying signal. This is true, as the rate of change of the spatial frequencies is almost always (except for image-plane holograms) small compared to the wavelength of light which is an approximate scale for the oscillations within the hologram.

Therefore, provided such a limited scope, the dominant spatial frequency is approximately constant. This means, that if analyzed with any smaller window, the spatial frequency is approximately translation-invariant within the scope. Coming back to our model from above, within such a scope the conditional probability of a pixel and its context are approximately independent of the spatial position of the pixel within said scope. Of course, the bigger such a scope can be chosen, the greater the number of observations for a context and the better the probability estimate in Eq. (5) converges towards the actual probability. However, the bigger the scope the worse is typically the quasi-stationarity assumption. Therefore, there exists a tradeoff which is regulated by the choice of the context size and the size of said limited scopes. How, exactly we construct said scopes we will discuss in section 4. Before, we will describe how to replace fixed with adaptive template models.

3. Tree-based context models

3.1. Statistical modelling

In the previous sections, we discussed adaptive arithmetic coding using a fixed template model - a simple yet practical model used in context-based compression. In this section, we will discuss the statistical modelling in more detail and elucidate the dependence of compression performance on model complexity.

In case of a fixed template context - assuming a statistically stationary signal and neglecting the impact of computational complexity - one might be tempted to pick a template set F

that contains more neighboring pixels. This would be a valid choice once the probability estimates have all converged to the true probabilities. Indeed, information theory postulates that $H(X|Y_1, Y_2) \leq H(X|Y_1)$, which implies that the entropy regarding any random variable decreases as the number of conditioning states increases. However, due to the finite number of hologram pixels (or pixels within a scope), the true probability induced by some context and the probabilities estimated using Eq. (5) will rarely be the same case. As the size of the fixed template set F increases, the uncertainty may decrease but will also require more observations for reliable probability estimates. This is because the increase in the number of context states due to the bigger template dilutes the available statistical information used for probability estimation.

Mathematically, this is formulated by Rissanen's lower bound [21] for the expected code length for finite state models as a function of the model cardinality. This lower bound places fundamental implications on the selection of an ideal statistical model. Increasing the cardinality for the statistical model is the only way to reduce the conditional entropy of the stochastic process, but will increase the bit(s) per symbol by a value proportional to $\frac{\log n}{n}$ per additional parameter [22].

3.2. Tree-based context modelling algorithms

The Universal Context (UC) tree algorithm introduced in [22] takes into account the implications of model complexity on compression performance. As the penalty for every additional parameter is proportional to $\frac{\log n}{n}$ bits – which decreases as n increases – the algorithm uses a tree model that adaptively increases the template size as more pixels are processed. Instead of having only one fixed template model, the tree algorithm employs many fixed template models of different sizes, out of which the model that encodes the current pixel being encoded with the least bits needs to be determined.

Another aspect of tree-based algorithms is tree construction, which determines the child nodes of some parent node. Free tree models [16] can provide more flexibility by permitting dynamic ordering dependent on the context, but come at the cost of additional computational complexity. On the other hand, a static ordering of M template pixels starting from the pixel with the most predictive power to the least, is preferable for a simpler design. Furthermore, the static ordering of M pixels can be communicated with at most $\log_2(M!)$ bits, which is marginal overhead for typical values of M . In contrast, dynamic ordering would require a not insignificant amount of $\sum_{i=0}^{M-1} 2^i \log_2(M-i)$ bits.

The static ordering $(k_1, l_1) \rightarrow (k_2, l_2) \cdots \rightarrow (k_M, l_M)$ will determine the parent-child order of the fixed template models contained in the tree. For the current pixel $x_{k,l}$, there shall be M contexts $\mathbf{x}^{k,l,d} = [x_{k-k_1, l-l_1}, x_{k-k_2, l-l_2}, \dots, x_{k-k_d, l-l_d}]$, where $1 \leq d \leq M$. Each possible context $\mathbf{c}^d$ forms a node of depth d in the tree, and stores the histograms counts $n(\mathbf{c}^d)$ and $n_1(\mathbf{c}^d)$. After selecting the best context model of depth d_o , the calculation of the probability estimate for the current pixel $x_{k,l}$ is done with the fixed template model of depth d_o for the context $\mathbf{x}^{k,l,d_o}$ using Eq. (5). After encoding the pixel, the histogram counts for the contexts $\mathbf{x}^{k,l,d}$ are updated for all depths $1 \leq d \leq M$.

For single pass schemes, the template pixel order can be determined with a suitable distance metric with respect to the current pixel, such as the ℓ_1 or ℓ_2 metrics. Alternatively, given a template of size M , a two pass scheme in which the first pass finds a suitable ordering for compression can be employed [16]. Thereby, the ordering is determined by iteratively minimizing the conditional entropy in Eq. (1). In the first step, the pixel with the lowest conditional entropy of the pixels in the template is selected as the first pixel in the order. Subsequent steps then determine the i^{th} pixel in the order by calculating the conditional entropy of all the remaining $M-i+1$ pixels given the candidate pixel and the previously $i-1$ selected pixels.

For the determination of the best context depth d_o , the tree algorithms perform successive child-parent coding efficiency comparisons from the deepest node to the root [23]. The best depth d_o is chosen as the first child whose coding efficiency is better than its parent.

We will introduce an Instantaneous Bayesian Context (IBC) tree algorithm in the next section, that utilizes the same principle as the UC algorithm, which is to pick a larger sized model as more pixels are processed. The differences lie in the rules deployed to decide for the best template model.

3.3. Instantaneous Bayesian Context algorithm

The central problem to be solved by the tree-based context modelling algorithms is to compare the coding efficiency of the parent node context $\mathbf{c}^d$ with the child node context $\mathbf{c}^{d+1}$. In IBC, we employ Bayesian inference for this purpose, under the assumption of no *a priori* information about the stationary statistical model generating the hologram.

Under these conditions, the minimal expected bitrate incurred at some node is obtained by substituting the Bayesian probability estimate in Eq. (5) in Eq. (3) and is given by

$$R_{\text{bay}}(\mathbf{c}^d) = h_b \left(\frac{n_1(\mathbf{c}^d) + 1}{n(\mathbf{c}^d) + 2} \right). \quad (6)$$

The difference in the expected bitrate of using the parent node over its child nodes is given by

$$\Delta(\mathbf{c}^d) = R_{\text{bay}}(\mathbf{c}^d) - \int_0^1 f(q|\mathbf{c}^d) \cdot R_{\text{bay}}([\mathbf{c}^d \ 1]) + (1 - f(q|\mathbf{c}^d)) \cdot R_{\text{bay}}([\mathbf{c}^d \ 0]) dq. \quad (7)$$

where $f(q|\mathbf{c}^d)$ is the PDF of the (conditional) probability of occurrence of the child context $[\mathbf{c}^d \ 1]$ given $\mathbf{c}^d$ has occurred and given the histogram counts of the contexts. Without a priori information, the PDF becomes

$$f(q|\mathbf{c}^d) = \frac{(n(\mathbf{c}^d) + 1)!}{n([\mathbf{c}^d \ 1])! (n(\mathbf{c}^d) - n([\mathbf{c}^d \ 1]))!} \cdot q^{n([\mathbf{c}^d \ 1])} \cdot (1 - q)^{n(\mathbf{c}^d) - n([\mathbf{c}^d \ 1])}. \quad (8)$$

Substituting Eq. (6) and Eq. (8) in Eq. (7), we get

$$\Delta(\mathbf{c}^d) = h_b \left(\frac{n_1(\mathbf{c}^d) + 1}{n(\mathbf{c}^d) + 2} \right) - \frac{n([\mathbf{c}^d \ 0]) + 1}{n(\mathbf{c}^d) + 2} \cdot h_b \left(\frac{n_1([\mathbf{c}^d \ 0]) + 1}{n([\mathbf{c}^d \ 0]) + 2} \right) - \frac{n([\mathbf{c}^d \ 1]) + 1}{n(\mathbf{c}^d) + 2} \cdot h_b \left(\frac{n_1([\mathbf{c}^d \ 1]) + 1}{n([\mathbf{c}^d \ 1]) + 2} \right). \quad (9)$$

If $\Delta(\mathbf{c}^d) > 0$, then the expected bitrate using the children is smaller than the parents. As mentioned previously, the comparison starts at the deepest node and stops when the first children that are better than the parent are obtained. Note that here the parent context $\mathbf{c}^d$ is compared with the information in both child contexts $[\mathbf{c}^d \ 0]$ and $[\mathbf{c}^d \ 1]$, irrespective of the child context $\mathbf{c}^{d+1}$ that has actually occurred.

In the case of the UC algorithm and its variants, the child-parent comparisons are done in terms of an additional parameter stored per node. It is called the efficiency index [24] and is based on a simulated coding metric called Predictive Minimum Description Length principle (PMDL), which compares the cumulative cost of encoding a pixel, when using the statistics of the child node, versus the statistics of the parent node. After encoding the current pixel, the difference between the two code lengths is accumulated in the efficiency index initialized with an

experimentally determined bias parameter Q . Once the efficiency index becomes negative it is determined that the child has become better than its parent. In contrast to the IBC algorithm, the next depth of the tree is not chosen immediately when the child becomes better than the parent node, but only when the cumulative cost of encoding all the previously encountered pixels with the child node is better than using the parent node. This means there exists a “lag” between the evaluation point where the child node becomes better than its parent node and the point when this is indicated by the efficiency index.

The compression performance obtained by the fixed template and tree models is shown in Fig. 2 as a function of template size, along with the JBIG and JBIG2 implementations as baselines. The evaluation was performed for a selection of binary sub-holograms of size 1024×1024 px.

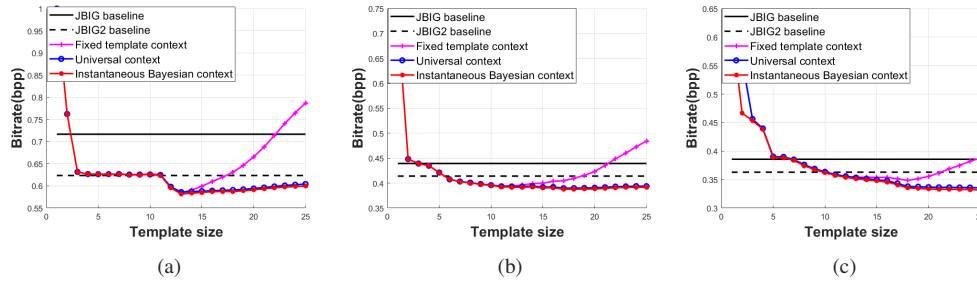


Fig. 2. Lossless compression bitrate as a function of template size for various lossless compression schemes for 1024×1024 px sub-hologram from (a) Sphere, (b) Aeroplane, (c) Dragons. The order of template pixels is determined using the ℓ_1 metric. The bitrates obtained for JBIG and JBIG2 implementations are also provided as baselines.

In agreement with the trade-off imposed by Rissanen’s lower bound for finite state models discussed in section 3.1, the compression performance of the fixed template model in Fig. 2 resembles a U-shape curve with respect to the template size. In contrast, for both the IBC and UC tree algorithms the relationship is almost monotonic, where IBC and UC perform similarly well. With the Bayesian probability estimation with unknown priors used in both fixed template and tree models, it is possible to improve the compression performance over codecs like JBIG and JBIG2.

For both fixed template and tree algorithms, we observe that the best performing template size increases with the size of the processed hologram segments, as more information increases the viability of having larger state models. However as mentioned in section 2.2, this analysis is confounded by the non-stationary properties of holograms over relatively large segments, which will also impair the compression performance of these models. The non-stationary behavior and the ways to overcome this issue will be discussed in the next section.

There are differences in computational complexity between the three techniques. For some depth d , 2^{d+2} histogram counts with integer precision are required for the UC and IBC tree algorithms, while 2^{d+1} counts are needed for FT. In the case of UC, additional 2^{d+1} efficiency indexes each representing a real number need also to be stored. In IBC and UC, the primary contributors to the performed operations are the binary entropy function and logarithms computed respectively for determining the optimal depth in the tree. Both these are transcendental functions requiring several iterations to converge to an acceptable value. With IBC the accuracy for correctly determining the sign of Eq. (9) is only required at a given point, which makes it possible to leverage digit-by-digit algorithms like coordinate rotation digital computer (CORDIC) [25]. In the case of UC, the required accuracy is undefined due to the cumulative nature of the efficiency index. Even if we were to consider both transcendental functions to have the identical number of operations, IBC still requires fewer computations in practical cases. This is due to

the fact that in IBC, the binary entropy function (2 for each depth) needs to be calculated only till the optimal depth is reached. In case of UC, the efficiency indexes needs to be updated by computing logarithms (1 for each depth) for all depths belonging to the tree. As more of the hologram is processed fewer computations need to be done for IBC with respect to UC. The number of transcendental functions computed for both IBC and UC for a selection of binary sub-holograms and template sizes is shown in Fig. 3. We see that in the region of interest, UC requires approximately 2-3 times more transcendental function computations when compared to IBC for holograms of size 1024×1024 px.

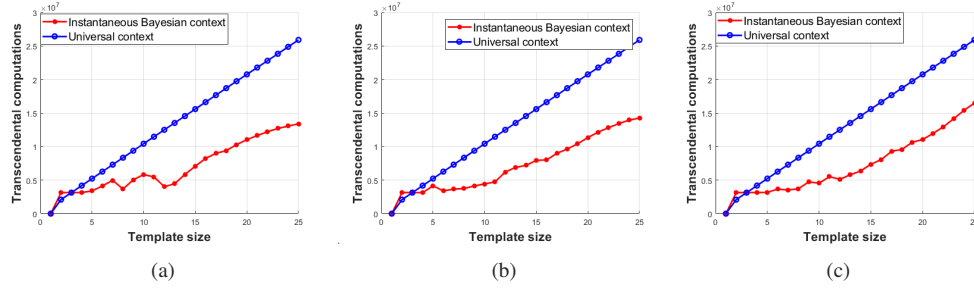


Fig. 3. Total transcendental function computations as a function of template size for IBC and UT for 1024×1024 px sub-hologram from (a) Sphere, (b) Aeroplane, (c) Dragons. The order of template pixels is determined using the ℓ_1 metric.

4. Hologram segmentation

The context-based adaptive arithmetic coding techniques discussed in the previous sections assume a statistically stationary source generating the input data. The notion of stationarity is however also frequently transferred to deterministic signals whose spectral content is (translation-)invariant over space (in our case). Moreover, the notion is often relaxed using the quasi-stationary assumption, which states that the rate of change of the spatial frequency in a given scope is slow compared to the frequency itself. This notion hinges therefore on two dimensions, the maximal size of the analysis window, used to evaluate a frequency and the scope for which quasi-stationarity holds [26].

In general practical terms, holograms are known to be almost never stationary [27]: in the sense that when the spatial frequency is analyzed with short windows across the entire hologram it drifts with spatial position of the analysis window. The same holds true for the dominant spatial frequency of the corresponding binary pattern. Or in other words, for a given short window and hologram size, the conditional probability of the value of one pixel varies not only as a function of its neighborhood, but also through the absolute position of the pixel in the hologram.

In our case, we consider the adaptive contexts to be the frequency analysis windows and establish a limited scope within which the signal is approximately stationary, through a quadtree segmentation which we will discuss promptly.

Empirically, we confirmed for a larger dataset (see Table 2) that when the segment size increases, the average bitrate will initially decrease due to the improved convergence of the probability estimate by the statistical model and then increase due to the drift in the statistical behavior. We demonstrate this behavior by dividing the “Aeroplane” hologram of resolution 32768×32768 px into segments of size 256×256 px. The probability estimate and number of occurrences of the context [0000000000] after processing each block independently are shown in Fig. 4. The quasi-stationary behavior is universally observed for all contexts and holograms tested.

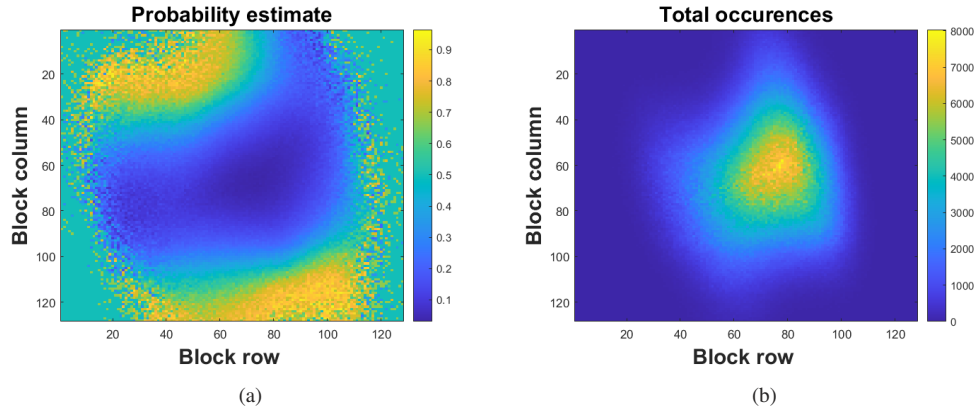


Fig. 4. Quasi-stationary behavior shown by (a) probability estimate of the occurrence of the [0000000000] context conditioned by the number of its (b) total occurrences. The context belongs to the 3L10P fixed template model. These parameters are calculated for spatial blocks of size 256×256 px from the Aeroplane hologram. While (a) shows a big difference in suitability of a fixed context for various regions of the hologram, (b) shows that the chosen context is still very useful in the central region where its most occurrences are.

While all compression codecs used in this work employ segmentation schemes, these are not designed for holographic content. For example, the JBIG2 [28] standard can segment the input into three region types, namely of text, halftone and others — a solution not applicable for holography.

We propose to use a quadtree to determine the segmentation producing the smallest bitrate. Quadrees allow for an image to be decomposed into variable sized tiles permitting more degrees of freedom [29] for the compression solution to adapt to the properties of the hologram. A single quadtree decomposition of some two-dimensional image I divides it into four equally large sub-images as shown in Eq. (10), where each child can be further be split in a recursive manner.

$$I = \begin{bmatrix} I_{NW} & I_{NE} \\ I_{SW} & I_{SE} \end{bmatrix}. \quad (10)$$

To find a *compression-aware* decomposition for a given hologram, we pursue a data-agnostic approach for the lack of existing heuristics and the great variability among holograms. Starting from a chosen the smallest sized segment size possible, we increase the segment size as needed. Each child quadtree segment is fully encoded, as is its parent. The bitrates are obtained and compared. If the cumulative bitrate of the children is smaller for some node than their parents the decomposition is maintained, otherwise the decomposition level is reduced and the procedure starts again. The entire procedure stops once no more parent nodes report a smaller bitrate than their children.

In the next section, we study the influence of compression performance of all the compression methods with and without the quadtree segmentation. For hologram compression, spatial segmentation into independent units not only ensures separation of the statistical behavior from various regions of the hologram, but also enables random access and parallel processing - key features for any practical codec to be used in hologram compression [30]. Without segmentation, these features are not possible for adaptive entropy coding schemes due to their sequential nature.

5. Results

In this section, we will evaluate the compression performance of the algorithms listed in Table 1 on a diverse set of holograms. This set features both computer generated and optically captured holograms and also includes a selection of holograms from the JPEG Pleno database. The numerical reconstructions obtained from these holograms are shown in Fig. 5. The bitrates obtained by the compression solutions for the various holograms are presented in Table 2, where the percentual reduction in bitrate obtained by the adaptive quadtree segmentation is provided in brackets.

Table 1. Compression solutions evaluated in section 5.^a

Compression algorithms	Short description	Template size
IBC (Entropy), IBC (ℓ_1)	Adaptive fixed point arithmetic coder with a tree-based context algorithm. Performs Bayesian estimation assuming unknown prior probability distributions. Pixel order used in tree determined by entropy estimation and ℓ_1 distance metric respectively.	up to 25 pixels
FT (16 pel), FT (10 pel)	Adaptive fixed point arithmetic coder using fixed template context algorithm. Performs Bayesian inference assuming unknown prior probability distributions.	16 and 10 pixels respectively
JBIG2	Implementation of ISO/IEC 14492 standard. Adaptive MQ coder with a fixed template context algorithm.	16 with optional 4 adaptive pixels
JBIG	Implementation of ISO/IEC 11544 standard. Adaptive QM coder with a fixed template context algorithm.	9 with optional 1 adaptive pixel
PNG	Implementation of ISO/IEC 15948 standard. Uses a simple predictor filter with 5 states followed by entropy coding of the prediction residual using the DEFLATE [31] algorithm.	4 pixels

^aIBC: Instantaneous Bayesian Context; FT: Fixed Template; PNG: Portable Network Graphics format

It can be seen that the tree-based IBC outperforms the fixed template (FT) context algorithms in compression performance, where the best performer is the two pass IBC (Entropy) solution. For the average bitrate calculated over all holograms, it was able to obtain a percentual reduction of 0.4%, 2.4%, 6.3%, 6.5%, 19.7% and 38.1% with respect to IBC (L1), FT(16), FT(10), JBIG2, JBIG and PNG, respectively, with all methods employing adaptive quadtree segmentation. In the case of fixed template algorithms, we saw that a template of size 16 outperformed the size 10. FT(16) requires a bitrate that was around 4.0% lower on average when compared to FT(10), while JBIG2 was around 14.0% lower compared to JBIG.

Comparing the algorithms with similar template sizes – FT(16) with JBIG2 and FT(10) with JBIG – we see percentual reductions of 4.3% and 14.3%, respectively. This indicates that in the case of holograms, the unknown a priori assumption based probability estimation used by FT(16) and FT(10) can offer improved compression performance over the estimation used in bi-level image compression standards. For JBIG, the compression performance is further affected as it uses the QM coder - an entropy coding solution that sacrifices compression performance for a low(er) computational burden.

The compression gain of the adaptive quadtree segmentation is hologram dependant, and is somewhat marginal for smaller resolutions. The percentual reduction obtained by the quadtree for the various compression methods, calculated over the average bitrate of all holograms, is given in Fig. 6(a). It can be seen that the quadtree gain for the various image compression standards is moderate compared to IBC and FT, as the standards implement spatial tiling as part of their default operation. In general, for context based solutions, we observe that the larger

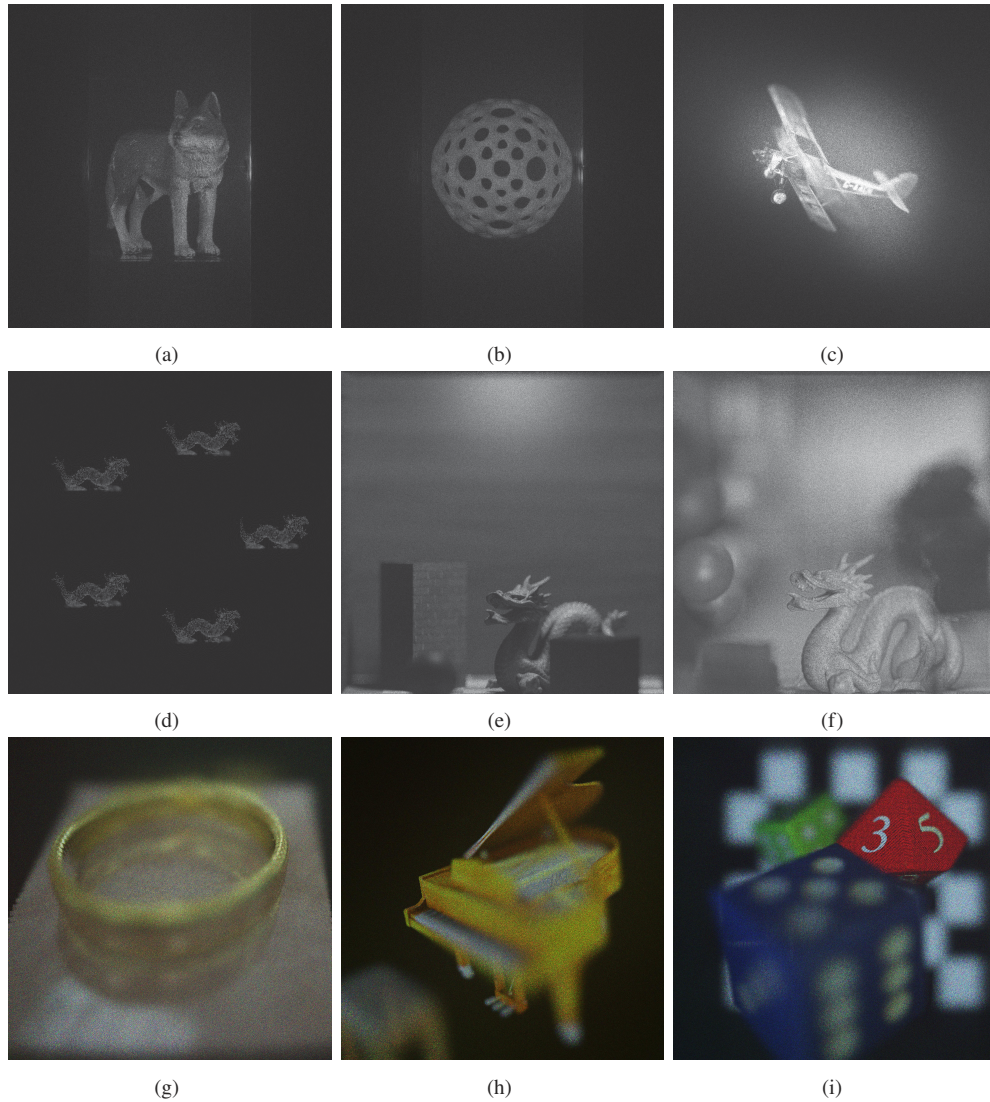


Fig. 5. Numerical object plane reconstructions of a diverse hologram set used for evaluating compression techniques: (a) **Wolf** - OCH, 2048×32768 px, $3.45 \mu\text{m}$, 633 nm; (b) **Sphere** - OCH, 2048×32768 px, $3.45 \mu\text{m}$, 532 nm; (c) **Aeroplane** - CGH, 32768×32768 px, $1 \mu\text{m}$, 638 nm; (d) **Dragons** - CGH, 16384×16384 px, $2 \mu\text{m}$, 475 nm; (e) **Cornell box** - CGH, 32768×16384 px, $2 \mu\text{m}$, 633 nm; (f) **Deep Cornell box** - CGH, 32768×16384 px, $2 \mu\text{m}$, 633 nm; (g) **Ring** - CGH, 32768×16384 px, $0.4 \mu\text{m}$, 473/532/640 nm; (h) **Piano** - CGH, 32768×16384 px, $0.4 \mu\text{m}$, 473/532/640 nm; (i) **Dices** - CGH, 32768×16384 px, $0.4 \mu\text{m}$, 473/532/640 nm. The operating parameters of the holograms are listed in the following order - source, resolution, pixel pitch, and wavelength/s. *CGH*: computer-generated hologram; *UCH*: optically captured hologram.

Table 2. Bitrates in bpp obtained by various compression methods with adaptive quadtree segmentation for a diverse set of holograms.^a

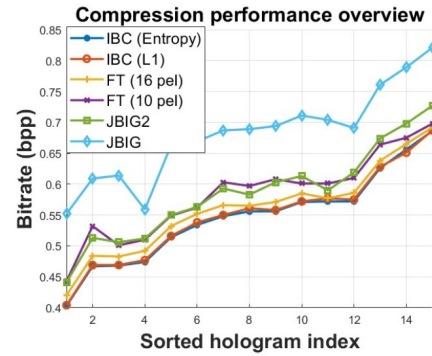
Hologram	Wavelength	IBC (Entropy)	IBC (L1)	FT (16 pel)	FT (10 pel)	JBIG2	JBIG	PNG
Wolf	633 nm	0.556 (0%)	0.561 (0%)	0.565 (0%)	0.597 (0%)	0.583 (0%)	0.689 (0%)	0.779 (0%)
Sphere	532 nm	0.572 (0%)	0.577 (0%)	0.577 (0%)	0.601 (0%)	0.590 (0%)	0.704 (0%)	0.810 (0%)
Aeroplane	638 nm	0.534 (17.6%)	0.538 (17.0%)	0.552 (23.4%)	0.562 (48.5%)	0.563 (10.5%)	0.669 (13.7%)	0.886 (8.1%)
Dragons	475 nm	0.474 (10.9%)	0.477 (10.3%)	0.492 (13.4%)	0.510 (18.4%)	0.512 (5.2%)	0.559 (6.7%)	1.170 (0%)
Cornell box	633 nm	0.572 (11.0%)	0.575 (10.6%)	0.586 (18.8%)	0.610 (39.5%)	0.619 (4.3%)	0.691 (6.4%)	0.864 (1.5%)
Deep Cornell box	633 nm	0.548 (11.9%)	0.550 (11.6%)	0.566 (18.6%)	0.603 (39.5%)	0.593 (3.9%)	0.687 (5.8%)	0.861 (1.6%)
Ring	640 nm	0.571 (10.2%)	0.572 (10.1%)	0.585 (17.0%)	0.601 (38.4%)	0.613 (3.6%)	0.711 (4.4%)	0.897 (0.9%)
	532 nm	0.656 (8.8%)	0.657 (9.5%)	0.665 (13.6%)	0.675 (36.1%)	0.698 (4.0%)	0.789 (4.6%)	0.944 (1.0%)
	472 nm	0.686 (8.9%)	0.687 (8.8%)	0.692 (13.1%)	0.698 (34.4%)	0.727 (4.2%)	0.821 (3.9%)	0.971 (0.9%)
Piano	640 nm	0.403 (19.9%)	0.404 (19.7%)	0.420 (31.9%)	0.443 (56.4%)	0.441 (5.4%)	0.552 (7.5%)	0.817 (2.5%)
	532 nm	0.468 (16.0%)	0.469 (15.8%)	0.483 (26.9%)	0.501 (52.8%)	0.506 (5.8%)	0.614 (8.4%)	0.870 (2.6%)
	472 nm	0.514 (15.3%)	0.516 (15.0%)	0.532 (24.0%)	0.549 (52.6%)	0.550 (7.3%)	0.663 (9.9%)	0.902 (3.6%)
Dices	640 nm	0.467 (15.7%)	0.469 (15.5%)	0.484 (23.3%)	0.532 (34.2%)	0.513 (3.9%)	0.609 (3.3%)	0.783 (1.4%)
	532 nm	0.556 (12.6%)	0.558 (12.3%)	0.571 (19.9%)	0.608 (35.8%)	0.603 (4.3%)	0.694 (3.6%)	0.828 (1.4%)
	472 nm	0.626 (9.8%)	0.628 (9.5%)	0.638 (16.5%)	0.664 (36.3%)	0.674 (4.1%)	0.761 (4.2%)	0.868 (1.6%)

^aThe percentual decrease in bitrate through the use of the quadtree is stated in brackets.

sized templates are more resilient to non-stationary effects. This is because the larger number of unique contexts inherently offers some level of separation for the differing behavior.

Algorithm	Reduction (%)
IBC (Entropy)	11.1
IBC (L1)	10.9
FT (16 pel)	17.6
FT (10 pel)	37.7
JBIG2	4.4
JBIG	5.5
PNG	1.8

(a)



(b)

Fig. 6. Summary of compression results. (a) Average percentual reduction in bitrate through the use of quadtree segmentation for the various compression solutions. (b) Bitrate achieved by the various compression solutions (PNG excluded) with quad tree segmentation for all holograms. Holograms sorted from left to right in order of increasing bitrate obtained by IBC (Entropy) compression solution.

The compression performance for all codecs and all holograms is summarized in Fig. 6(b).

6. Conclusion

In this work, the application of context based adaptive arithmetic coding for the compression of binary hologram was explored. The limitations caused by the quasi-stationary behavior of holograms was demonstrated for these approaches, as they are valid only for a statistically stationary signal. To mitigate this, an adaptive quadtree-based spatial segmentation scheme was proposed, which not only demonstrated improvements in compression performance but also facilitates spatial random access.

The tree-based IBC (Instantaneous Bayesian Context) algorithm was described, which utilizes Bayesian inference under the assumption of unknown priors. For all 9 tested holograms generated from various different techniques, IBC obtained the smallest file sizes. In total, 7 different compression configurations were evaluated and the compression results are visually summarized in 6(b). Overall, when compared to direct application of the JBIG2, JBIG and PNG compression standards, our IBC based compression solution was able to obtain percentual reductions of 12%, 31.7% and 64.5% respectively. It was submitted as part of the INTERFERE compression proposal [32], in response to the JPEG Pleno call for proposals for holographic content coding [33] and selected as the solution for encoding of binary holograms.

Funding. Fonds Wetenschappelijk Onderzoek (12ZQ220N, VS07820N).

Acknowledgments. Some of the holograms used for testing made use of point-cloud objects provided by the Stanford Computer Graphics Laboratory. Holograms “Sphere” and “Wolf” were generated in a collaboration [34] between Vrije Universiteit Brussel-imec and Warsaw University of Technology, while holograms “Ring,” “Piano,” and “Dices” [35] are at the courtesy of b<>.com.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. P. S. Naidu, "Quantization noise in binary holograms," *Opt. Commun.* **15**(3), 361–365 (1975).
2. E. Zhang, S. Nohte, C. H. Dietrich, and R. Männer, "Gradual and random binarization of gray-scale holograms," *Appl. Opt.* **34**(26), 5987–5995 (1995).
3. R. Li and L. Cao, "Progress in Phase Calibration for Liquid Crystal Spatial Light Modulators," *Appl. Sci.* **9**(10), 2012 (2019).
4. A. W. Lohmann and D. P. Paris, "Binary Fraunhofer Holograms, Generated by Computer," *Appl. Opt.* **6**(10), 1739–1748 (1967).
5. B. R. Brown and A. W. Lohmann, "Computer-generated Binary Holograms," *IBM J. Res. Dev.* **13**(2), 160–168 (1969).
6. W.-H. Lee, "Computer-Generated Holograms: Techniques and Applications," *Prog. Opt.* **16**, 119–232 (1978).
7. T. C. Strand, "Signal/Noise In Analog And Binary Holograms," *Opt. Eng.* **13**(3), 219–227 (1974).
8. Y. Takaki and N. Okada, "Hologram generation by horizontal scanning of a high-speed spatial light modulator," *Appl. Opt.* **48**(17), 3255–3260 (2009).
9. B. Lee, D. Yoo, J. Jeong, S. Lee, D. Lee, and B. Lee, "Wide-angle speckleless DMD holographic display using structured illumination with temporal multiplexing," *Opt. Lett.* **45**(8), 2148–2151 (2020).
10. Y. Lim, K. Hong, H. Kim, H.-E. Kim, E.-Y. Chang, S. Lee, T. Kim, J. Nam, H.-G. Choo, J. Kim, and J. Hahn, "360-degree tabletop electronic holographic display," *Opt. Express* **24**(22), 24999–25009 (2016).
11. J. Kim, Y. Lim, K. Hong, H. Kim, H.-E. Kim, J. Nam, J. Park, J. Hahn, and Y.-J. Kim, "Electronic Tabletop Holographic Display: Design, Implementation, and Evaluation," *Appl. Sci.* **9**(4), 705 (2019).
12. H.-G. Choo, T. Kozacki, W. Zaperty, M. Chlipala, Y. Lim, and J. Kim, "Fourier digital holography of real scenes for 360° tabletop holographic displays," *Appl. Opt.* **58**(34), G96–G103 (2019).
13. T. Shimobaba, D. Blinder, P. Schelkens, Y. Yamamoto, I. Hoshi, T. Kakue, and T. Ito, "Deep-learning-assisted Hologram Calculation via Low-Sampling Holograms," in *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI)* (2019), pp. 936–941.
14. C. E. Shannon, "A mathematical theory of communication," *The Bell Syst. Tech. J.* **27**(3), 379–423 (1948).
15. CCITT G4, "Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus," ITU-T Recommendation T.6.
16. B. Martins and S. Forchhammer, "Bi-level image compression with tree coding," in *Proceedings of Data Compression Conference - DCC '96* (1996), pp. 270–279.
17. H. Hampel, R. B. Arps, C. Chamzas, D. Dellert, D. L. Duttweiler, T. Endoh, W. Equitz, F. Ono, R. Pasco, I. Sebestyen, C. J. Starkey, S. J. Urban, Y. Yamazaki, and T. Yoshida, "Technical features of the JBIG standard for progressive bi-level image compression," *Signal Process. Image Commun.* **4**(2), 103–111 (1992).
18. J. Rissanen and G. G. Langdon, "Arithmetic Coding," *IBM J. Res. Dev.* **23**(2), 149–162 (1979).
19. I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," *Commun. ACM* **30**(6), 520–540 (1987).
20. P. M. Lee, *Bayesian Statistics: An Introduction* (Wiley Publishing, 2012), 4th ed.
21. J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inf. Theory* **30**(4), 629–636 (1984).
22. M. J. Weinberger, J. J. Rissanen, and R. B. Arps, "Applications of universal context modeling to lossless compression of gray-scale images," *IEEE Trans. Image Process.* **5**(4), 575–586 (1996).
23. R. Nohre, Some Topics in Descriptive Complexity (Linköping Univ., 1994).
24. G. Furlan, C. Galand, and E. Lancon, "Etude comparative des systèmes de compression de données sans perte d'information et codage arithmétique," in *Colloques sur le Traitement du Signal et des Images* (1989).
25. J. Volder, "The CORDIC Computing Technique," in *Managing Requirements Knowledge, International Workshop*, vol. 1 (IEEE Computer Society, 1959), p. 257.
26. P. Flandrin, *Explorations in Time-Frequency Analysis* (Cambridge University Press, 2018).
27. T. Birnbaum, T. Kozacki, and P. Schelkens, "Providing a visual understanding of holography through phase space representations," *Appl. Sci.* **10**(14), 4766 (2020).
28. F. Ono, W. Rucklidge, R. Arps, and C. Constantinescu, "JBIG2-the ultimate bi-level image coding standard," in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 1 (2000), pp. 140–143.
29. J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Process.* **40**(8), 2040–2060 (1992).
30. P. Schelkens, T. Ebrahimi, A. Gilles, P. Gioia, K.-J. Oh, F. Pereira, C. Perra, and A. M. G. Pinheiro, "JPEG Pleno: Providing representation interoperability for holographic applications and devices," *ETRI J.* **41**(1), 93–108 (2019).
31. P. Deutsch, "RFC1951: DEFLATE Compressed Data Format Specification Version 1.3" (RFC Editor, 1996).
32. ISO/IEC JTC1/SC29/WG1, "Response to Call for Proposals on JPEG Pleno Holography by Vrije Universiteit Brussel – imec," WG1M93005, 93rd JPEG Meeting (2021).
33. ISO/IEC JTC1/SC29/WG1, "Final Call for Proposals on JPEG Pleno Holography," WG1N91022, 91st JPEG Meeting (2021).
34. 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).

35. 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).