ELSEVIER

Diffraction field computation from arbitrarily distributed data points in space ☆

G. Bora Esmer[a],*, Vladislav Uzunov[b], Levent Onural[a],
Haldun M. Ozaktas[a], Atanas Gotchev[b]

[a]*Department of Electrical and Electronics Engineering, Bilkent University, TR-06800 Bilkent, Ankara, Turkey*
[b]*Institute of Signal Processing, Tampere University of Technology, FIN-33101 Tampere, Finland*

Abstract

Computation of the diffraction field from a given set of arbitrarily distributed data points in space is an important signal processing problem arising in digital holographic 3D displays. The field arising from such distributed data points has to be solved simultaneously by considering all mutual couplings to get correct results. In our approach, the discrete form of the plane wave decomposition is used to calculate the diffraction field. Two approaches, based on matrix inversion and on projections on to convex sets (POCS), are studied. Both approaches are able to obtain the desired field when the number of given data points is larger than the number of data points on a transverse cross-section of the space. The POCS-based algorithm outperforms the matrix-inversion-based algorithm when the number of known data points is large.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Scalar optical diffraction; Plane wave decomposition; Pseudo-matrix inversion; Projection onto convex sets

1. Introduction

Holographic three-dimensional television (3DTV) systems will consist of several parts fulfilling different functions: capturing the 3D scene, representing it in abstract form, its transmission, and finally display. Once the abstract representation of the scene arrives at the display end, it is necessary to first compute the diffraction field that the scene would have created, in order to drive the display device in a manner so that the same field will be recreated. This paper deals with the solution of the problem of computing the diffraction field from the sampled abstract representation of the scene. This sampled representation consists of known values of the optical field over an irregularly distributed, arbitrary array of discrete points.

Diffraction problems are most commonly formulated such that the optical field at one plane is computed in terms of the field at another plane. This paper deals with computation of the field over all of space in terms of a set of given data points distributed arbitrarily over the space. It is not correct to calculate the optical field arising from such distributed data by means of straightforward superposition. This fact is often ignored and straightforward superposition is employed, which

amounts to assuming that each data point represents a source [12,10]. Here we properly formulate the problem and consider its solution using two approaches. The first utilizes a direct matrix inversion while the second one is iterative and utilizes the projections onto convex sets (POCS) method [1,9], which has been successfully used for various restoration problems in image processing and holography [7,20,5].

2. Review of diffraction theory

The computation of the light field in a desired region, based on the knowledge of the field in some other region, is the subject of diffraction theory, which is a mature area of knowledge [8,3,17,16]. Despite this, there are gaps in the efficient numerical application of the theory and there seems to be considerable scope for application of signal processing techniques such as sampling, numerical linear algebra, fast transformations, iterative projections, and decomposition algorithms towards obtaining better computation methods.

In most cases, monochromatic (single wavelength) light is used in holography. Moreover, the medium we are interested in is linear, isotropic, and homogeneous. Under these conditions, the optical field on one plane can be accurately related to that on another plane through the Rayleigh–Sommerfeld diffraction integral, which is a linear and shift-invariant relationship [8,3,17].

In this study, we work with fields consisting of propagating waves; hence the input and output fields do not contain any evanescent wave components. Moreover, we assume that the distances involved are $r \gg \lambda$, where λ is the wavelength. Also, we use plane wave decomposition to compute scalar optical diffraction field, because plane wave decomposition and Rayleigh-Sommerfeld diffraction integral are equivalent [18,11]. For simplicity, we restrict our discussions to only one transverse dimension; extension to two transverse dimensions is straightforward. The diffraction integral over 2D space arising from the plane wave decomposition approach is

$$u_a(x,z) = \int_{-2\pi/\lambda}^{2\pi/\lambda} A(k_x) \exp[j(k_x x + k_z z)] \, dk_x, \qquad (1)$$

where $u_a(x,z)$ is the field over 2D space, and $A(k_x)$ gives the complex amplitudes of the harmonic components (Fourier coefficients) of the field

$u_a(x,0)$ over the input line. (The relationship between $(2\pi)A(k_x)$ and $u_a(x,0)$ is the 1D Fourier transform (FT) relation.) The variables k_x and k_z are the spatial frequencies of the propagating plane waves along the x- and z-axis, respectively. The x-axis is the transverse axis and the z-axis is the optical axis along which the field propagates. The variable k_z is related to k_x by $k_z = \sqrt{k^2 - k_x^2}$ where $k = 2\pi/\lambda$. The expression in Eq. (1) can be rewritten as

$$u_a(x,z) = \mathscr{F}^{-1}\{\mathscr{F}[u_a(x,0)] \exp(j\sqrt{k^2 - k_x^2} \, z)\}, \qquad (2)$$

where $\mathscr{F}$ denotes the FT and $\mathscr{F}^{-1}$ is the inverse FT.

The input field is a bandlimited spatial function, whose bandlimit is within $\pm k$. This is because a propagating monochromatic wave with wavelength λ cannot have a harmonic component in the transverse plane with higher frequency. In certain cases, it may be desirable to further restrict the bandwidth along the transverse direction x. For instance, we may restrict k_x such that $-B \leqslant k_x < B$, where $B \leqslant k$. So far, k_x may assume any real value in this interval. To arrive at a feasible numerical framework, we must restrict ourselves to a finite number N of possible values of k_x. These may be chosen as $k_x = l(2B/N)$ where $l = -N/2, \ldots, N/2 - 1$ for even N and a similar formula for odd N. Discretizing the transverse frequency will result in a field which is periodic along the transverse direction x, with a fundamental period $X = \pi N/B$. Therefore, a careful choice of simulation parameters is necessary if the consequences of this periodicity effect are to be minimized. Therefore, the field becomes

$$u_a(x,z) = \sum_{m=0}^{N-1} A_m \exp(j\sqrt{k^2 - k_x^2} \, z) \exp\left(j\frac{2B}{N}mx\right), \qquad (3)$$

where

$$k_x = \begin{cases} 2\pi\dfrac{m}{X}, & m = \left[0, \dfrac{N}{2}\right), \\ 2\pi\dfrac{(m-N)}{X}, & m = \left[\dfrac{N}{2}, N\right). \end{cases} \qquad (4)$$

Sampling this periodic and bandlimited function with a sampling period $X_s = \pi/B$, which is its Nyquist rate, yields N samples per period, as expected. Restricting our attention to these samples, $x = nX_s$ and $z = pX_s$, where n is an integer

spanning one period, and p is the distance between the lines along the longitudinal direction z, we can write

$$u_a(nX_s, pX_s)$$
$$= \sum_{m=0}^{N-1} A_m \exp(j\sqrt{k^2 - k_x^2}\, pX_s) \exp\left(j\frac{2\pi}{N}mn\right). \quad (5)$$

Each frequency component m, determines the propagation direction of the corresponding plane wave. The angle ϕ, which is shown in Fig. 1, denotes the angle between the z-axis and the propagation direction:

$$\phi_m = \begin{cases} \sin^{-1}\left(\dfrac{\lambda m}{NX_s}\right), & m = \left[0, \dfrac{N}{2}\right), \\ \sin^{-1}\left(\dfrac{\lambda(m-N)}{NX_s}\right), & m = \left[\dfrac{N}{2}, N\right). \end{cases} \quad (6)$$

Thus $m \in [0, N/2)$ corresponds to the positive ϕ angles, and $m \in [N/2, N)$ corresponds to the negative ϕ angles. The relation between k_z and m is

$$k_z = \begin{cases} \dfrac{2\pi}{NX_s}\sqrt{\beta^2 - m^2}, & m = \left[0, \dfrac{N}{2}\right), \\ \dfrac{2\pi}{NX_s}\sqrt{\beta^2 - (m-N)^2}, & m = \left[\dfrac{N}{2}, N\right), \end{cases} \quad (7)$$

where $\beta = NX_s/\lambda$. Therefore, the discrete kernel corresponding to the plane wave decomposition becomes,

$$H_p(m) = \begin{cases} \exp\left(j\dfrac{2\pi}{N}\sqrt{\beta^2 - m^2}\, p\right), & m = \left[0, \dfrac{N}{2}\right), \\ \exp\left(j\dfrac{2\pi}{N}\sqrt{\beta^2 - (m-N)^2}\, p\right), & m = \left[\dfrac{N}{2}, N\right) \end{cases} \quad (8)$$
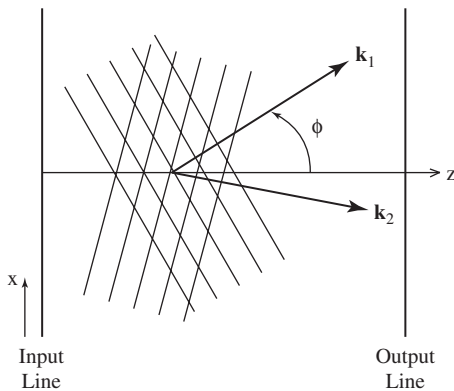

Fig. 1. The vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ are the wave vectors of the plane waves.

and the resultant form of the discrete representation of the plane wave decomposition is

$$u(n, p) = \sum_{m=0}^{N-1} A_m H_p(m) \exp\left(j\frac{2\pi}{N}nm\right), \quad (9)$$

where $N \cdot A_m$ is the DFT of the sampled input field. The variable n is restricted to the range $[0, N)$. Therefore, the discrete diffraction field can be expressed as

$$u(n, p) = \mathrm{DFT}^{-1}\{\mathrm{DFT}\{u(n, 0)\}H_p(m)\}. \quad (10)$$

This completes the proper discretization of the diffraction formula.

3. Computation of diffraction patterns from distributed data points over 2D space

Computation of the diffraction pattern at the output arising from a distributed set of points can be time-consuming, due to the z dependency of the kernel. The diffraction relation between the distributed data set and a given line cannot be represented as a shift-invariant system. As already described, calculation of the diffraction field from one line to another is relatively straightforward since the corresponding system is linear and shift-invariant. The distributed points we wish to deal with may represent samples over a curved line, a tilted line, or another shape in 2D space. An illustration is given in Fig. 2. There are several fast methods which can be used to compute the diffraction field between parallel and tilted lines [15,19,4,14,13,6]. However, these methods cannot be used to compute the diffraction field from arbitrarily-distributed data points. In such a case, a simple and naive approach, i.e., calculation of the diffraction field by direct superposition assuming each
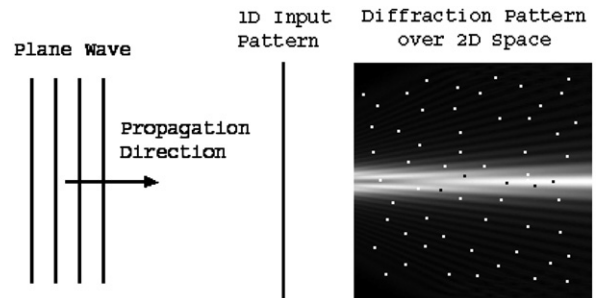

Fig. 2. An illustration of 1D object illumination and the diffraction pattern of the object over 2D space. Dots on the corresponding diffraction pattern represent the locations of a set of distributed known data points over 2D space.
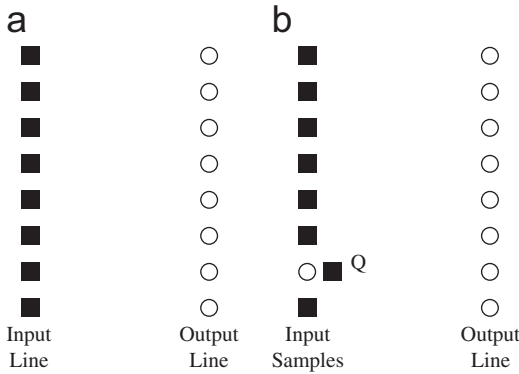
Fig. 3. (a) Parallel input and output lines. (b) Example involving a single displaced known data point.

data point as a source of light, would yield erroneous results. Two simple examples clarify this fact: In Fig. 3, black squares represent given samples while circles represent the missing ones. In the first example, the input and output lines are parallel to each other with no missing points (Fig. 3(a)). The relationship between the fields on these lines can be represented as a linear shift-invariant system, given by Eq. (10). In the second example (Fig. 3(b)), the field at one of the sample points on the input line is not known. Instead, the field at another sample point Q, not located on the input line, is given in order to compensate the missing data. Each point on the input line contributes to the field at Q, and in turn, the field at Q affects the field in other places. Therefore, it is not possible to write a proper superposition including the point Q for the field at unknown regions.

Let us assume a hypothetical straight line, referred to as the reference line, and compute the diffraction field on it. We rewrite Eq. (10) in matrix form:

$$\mathbf{g} = \mathbf{A}\mathbf{f}, \tag{11}$$

where the vectors $\mathbf{f}$ and $\mathbf{g}$ represent the discrete diffraction fields $u(n,0)$ and $u(n,p)$ of the reference line and some other line, respectively. The rows of $\mathbf{A}$ are obtained from Eq. (10) as

$$\mathbf{A} = \mathbf{W}^{-1}\mathbf{H}_p\mathbf{W}, \tag{12}$$

where the matrix $\mathbf{W}$ is the N-point DFT matrix. The matrix $\mathbf{H}_p$ is the diagonal matrix

$$\mathbf{H}_p = \begin{bmatrix} H_p(0) & 0 & \dots & 0 \\ 0 & H_p(1) & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & H_p(N-1) \end{bmatrix}. \tag{13}$$

The matrix $\mathbf{A}$ is unitary. Furthermore, $\mathbf{A}^k$ represents the diffraction field at a distance pk. Proofs of these properties are given in the Appendix.

3.1. Calculation by matrix inversion

Our first approach calculates the diffraction field on the reference line by directly solving the system of linear equations (direct matrix inversion).

Let the vector $\mathbf{g}'$ denote the diffraction data over s distributed data points,

$$\mathbf{g}' = \begin{bmatrix} g_1' \\ g_2' \\ \vdots \\ g_s' \end{bmatrix}. \tag{14}$$

Each g_i' is a function of both p_i, the index in the z direction, and n_i, the index in the x direction. The relation between the vector $\mathbf{g}'$ and the reference vector $\mathbf{f}$ is given by

$$\mathbf{g}' = \mathbf{A}_{\mathbf{BF}}\mathbf{f}, \tag{15}$$

where $\mathbf{A}_{\mathbf{BF}}$ is an s by N matrix:

$$\mathbf{A}_{\mathbf{BF}} = \begin{bmatrix} \mathbf{r}(p_1, n_1) \\ \mathbf{r}(p_2, n_2) \\ \vdots \\ \mathbf{r}(p_s, n_s) \end{bmatrix}, \tag{16}$$

where $\mathbf{r}(p_i, n_i)$ is a 1 by N row vector from the matrix $\mathbf{A}$ with $p = p_i$. This row vector provides the diffraction field relation between the field on the reference line and the field on the point specified by p_i and n_i. The vector $\mathbf{f}$ is obtained by

$$\mathbf{f} = \mathbf{A}_{\mathbf{BF}}^+\mathbf{g}', \tag{17}$$

where $\mathbf{A}_{\mathbf{BF}}^+$ is the pseudo-inverse of the matrix $\mathbf{A}_{\mathbf{BF}}$:

$$\mathbf{A}_{\mathbf{BF}}^+ = \begin{cases} (\mathbf{A}_{\mathbf{BF}}^H\mathbf{A}_{\mathbf{BF}})^{-1}\mathbf{A}_{\mathbf{BF}}^H, & s > N, \\ \mathbf{A}_{\mathbf{BF}}^H(\mathbf{A}_{\mathbf{BF}}\mathbf{A}_{\mathbf{BF}}^H)^{-1}, & s < N. \end{cases} \tag{18}$$

Here $\mathbf{A}_{\mathbf{BF}}^H$ is the conjugate transpose of $\mathbf{A}_{\mathbf{BF}}$. Knowing the diffraction field on the reference line, we can compute the diffraction field on any other line in the 2D space using Eq. (11).

3.2. Calculation by using projections onto convex sets

Our second approach utilizes an iterative technique, based on the POCS method [1,9]. POCS has

been applied to various problems in holography and image restoration where a priori information is used to constrain the size of the feasible solution set [7,20,5]. It is a computational approach for finding an element of a feasible region defined by the intersection of a number of convex constraints, starting with an arbitrary infeasible point [1,9]. Fig. 4 shows how convergence to the intersection is achieved by iterative projections onto the individual convex sets. In our problem, the constraints are the known data points on consecutive lines. In addition, the data points are known to belong to the same diffraction field and therefore Eq. (10) has to be satisfied among all lines. Therefore, the convex set $\mathbf{C}_l, l = 1, \ldots, M$ (number of lines), can be defined as all possible diffraction fields having the given data points on a certain line $z = z_l$:

$$\mathbf{C}_l = \{\forall f(x,z) : f(x_{\mathbf{i}_l}, z_l) = \mathbf{v}_l,$$
$$f(x, z_j) = \mathbf{A}^{j-l} f(x, z_l), \forall j, x\}, \tag{19}$$

where $\mathbf{v}_l$ is the vector of known values on the line $z = z_l$ and $\mathbf{i}_l$ is the vector with the indices of their positions. $\mathbf{A}$ is the diffraction matrix given by Eq. (12) written for two consecutive lines, and its $(j - l)$th power $\mathbf{A}^{j-l}$, is the diffraction matrix from the line $z = z_l$ to the line $z = z_j$. A closed form expression for $f(x, z_j)$ can be written with the help of an arbitrary function $q(x)$:

$$f(x, z_j) = \mathbf{A}_{\mathbf{i}_l}^{j-l} \mathbf{v}_l + \mathbf{A}_{\bar{\mathbf{i}}_l}^{j-l} q(x_{\bar{\mathbf{i}}_l}), \tag{20}$$

where $\bar{\mathbf{i}}_l$ is the vector with the indices of the unknown values on the line $z = z_l$, and $\mathbf{A}_{\mathbf{i}_l}^{j-l}$ and $\mathbf{A}_{\bar{\mathbf{i}}_l}^{j-l}$ are submatrices of $\mathbf{A}^{j-l}$ obtained by taking columns with indices $\mathbf{i}_l$ and $\bar{\mathbf{i}}_l$.

It is straightforward to show that the sets $\mathbf{C}_l$ defined by Eq. (19) are convex. Let us assume that the functions $f_1(x,z)$ and $f_2(x,z)$ belong to the set $\mathbf{C}_l$, and $F(x,z) = \alpha f_1(x,z) + (1-\alpha)f_2(x,z), 0 < \alpha < 1$ is their convex combination. Then for every line $z =$
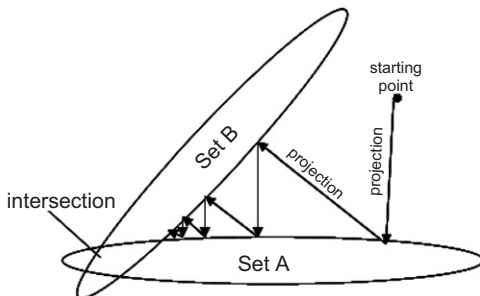

Fig. 4. Projections onto convex sets (POCS).

z_j we have

$$F(x, z_j) = \alpha \mathbf{A}_{\mathbf{i}_l}^{j-l} \mathbf{v}_l + \alpha \mathbf{A}_{\bar{\mathbf{i}}_l}^{j-l} q_1(x_{\bar{\mathbf{i}}_l})$$
$$+ (1-\alpha) \mathbf{A}_{\mathbf{i}_l}^{j-l} \mathbf{v}_l + (1-\alpha) \mathbf{A}_{\bar{\mathbf{i}}_l}^{j-l} q_2(x_{\bar{\mathbf{i}}_l})$$
$$= \mathbf{A}_{\mathbf{i}_l}^{j-l} \mathbf{v}_l + \mathbf{A}_{\bar{\mathbf{i}}_l}^{j-l} Q(x_{\bar{\mathbf{i}}_l}), \tag{21}$$

where $Q(x) = \alpha q_1(x) + (1-\alpha)q_2(x)$. Since $q(x)$ in Eq. (20) can be an arbitrary function, the last line of Eq. (21) becomes of the form of Eq. (20) and therefore $F(x,z)$ belongs to the set $\mathbf{C}_l$. This fact shows that $\mathbf{C}_l$ is convex, because a convex combination of any two functions which belong to $\mathbf{C}_l$ also belongs to $\mathbf{C}_l$.

A POCS-based algorithm requires iterating from set to set using projections. A straightforward strategy is to propagate from line to line using Eq. (10). Unknown points are generated by the field from the previous line and the known points are kept. Let us assume $g(x,z) \in \mathbf{C}_{l-1}$ and $f(x,z) \in \mathbf{C}_l$. Then,

$$f(x_{\bar{\mathbf{i}}_l}, z_l) = g(x_{\bar{\mathbf{i}}_l}, z_l)$$

and

$$f(x_{\mathbf{i}_l}, z_l) = \mathbf{v}_l, \tag{22}$$

while projecting from the set $\mathbf{C}_{l-1}$ to the set $\mathbf{C}_l$. With this choice, $f(x, z_l)$ differs from $g(x, z_l)$ in the restricted values at positions $x_{\mathbf{i}_l}$ only, and therefore the distance from $g(x, z_l)$ to $f(x, z_l)$ is minimized with respect to all functions in $\mathbf{C}_l$. Moreover, the distance taken for any other line $z = z_j$ is minimized since

$$d = \|f(x, z_j) - g(x, z_j)\|^2$$
$$= \|\mathbf{A}^{j-l}(f(x, z_l) - g(x, z_l))\|^2$$
$$= \|f(x, z_l) - g(x, z_l)\|^2 \tag{23}$$

and $\mathbf{A}^{j-l}$ is unitary for any j and l. Hence, the squared distance between $g(x,z)$ and $f(x,z)$ is also minimal for all the functions from the set $\mathbf{C}_l$ because

$$\|f(x,z) - g(x,z)\|^2 = \sum_j \|f(x, z_j) - g(x, z_j)\|^2.$$

Therefore, the choice for $f(x,z)$ given by Eq. (22) corresponds to the orthogonal projection of $g(x,z)$ onto the set $\mathbf{C}_l$ and hence the iterative method will converge [20].

At the first line $z = 0$, iterations are initiated by setting the missing points to arbitrary values. After passing through all lines, the unknown points on the first line are computed using the data on the last line.

The algorithm can be summarized as follows:

1. initialize the first line of the desired field
    $f(x_{i_l}, 0) = \mathbf{v}_0, f(x_{\bar{i}_l}, 0) = q(x_{\bar{i}_l})$, for any $q(x)$
2. for $i = 1$ to n_{it}
        (a) for $l = 2$ to M
            i. $f(x, z_l) = \mathbf{A}f(x, z_{l-1})$
            ii. $f(x_{\mathbf{i}_l}, z_l) = \mathbf{v}_l$
        (b) end
        (c) $f(x, 0) = \mathbf{A}^{-M+1}f(x, z_M)$
        (d) $f(x_{\mathbf{i}_l}, 0) = \mathbf{v}_0$;
3. end

where n_{it} is the number of iteration.

3.3. Results

For purposes of illustrating and evaluating the two approaches outlined above, we choose a simple function as the optical field on the reference line. Then, the diffraction field at other lines is calculated using Eq. (10). The computation is conducted with $M = 256$ lines with $N = 256$ samples per line and the range of p is [129, 384]. The function on the reference line is a unit-magnitude square pulse of 32 samples located at the center of the line. The generated initial data pattern is depicted in Fig. 5(a).

Assessment of the results is based on the normalized error between the original and reconstructed diffraction patterns $u(n, p)$ and $u'(n, p)$, respectively,

$$\frac{\sum_{n=0}^{N-1}\sum_{p=129}^{384}|u'(n,p) - u(n,p)|^2}{\sum_{n=0}^{N-1}\sum_{p=129}^{384}|u(n,p)|^2}. \tag{24}$$

The error has been tabulated for different numbers s of known data points, as shown in Tables 1 and 2. For each value of s, 15 diffraction patterns have been reconstructed using s randomly selected data points from the initial simulated field. Each reconstructed diffraction pattern corresponds to a different random choice of the positions of the s known data points within the field. The final error reported for each value of s is the average of the error over all 15 choices.

The numerical implementations of the proposed algorithms utilize the complex arithmetic operations such as multiplication and addition. Computation times of these operations depend on many implementation details including hardware properties and operating system behavior. Implementations of the described digital algorithms need the execution of large amount of complex arithmetic operations; furthermore, large amount of data fetch and write
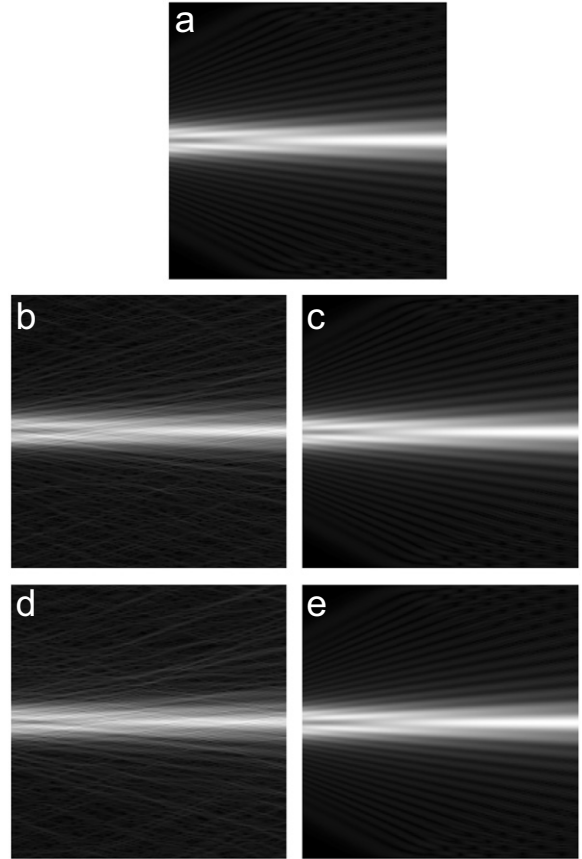
Fig. 5. Initial diffraction field over the entire 2D space; $N = 256$ samples per line (a) and reconstructed diffraction fields from s known data points (b)–(e); (b) matrix inversion method with $s = 230$; (c) matrix inversion method with $s = 282$; (d) POCS algorithm with $s = 230$; (e) POCS algorithm with $s = 282$.

operations are needed. The actual resultant computation time naturally depends on the specifics of the computer architecture and the operating system behavior. Though incomplete, a comparison of the required total number of multiplications may give an idea about the computational complexity of the algorithms.

The numerical results for the matrix inversion method are summarized in Table 1. Higher number of initial given samples yields better reconstruction of the original diffraction pattern, as expected. When this number is equal to or higher than N, the diffraction field is reconstructed perfectly. This approach involves computation of the diffraction field on a reference line by taking the pseudo-inverse of $\mathbf{A_{BF}}$ and computation of the entire diffraction field from that line. Forward and inverse DFTs are required to calculate the diffraction from between

consecutive lines. We chose to implement the DFT using common FFT algorithms. The diffraction field computation on a line requires two FFT algorithms and to implement FFT algorithm, $(N/2)\log_2 N$ complex multiplications are used. As a result of this, the total number of complex multiplications are needed to calculate the diffraction pattern over 2D space from the input field on a line is $MN\log_2 N + MN$, where M is the number of lines and the additional term MN is the multiplication of the discrete kernel by the DFT

coefficients of the input field. An efficient method for computing the pseudo-inverse of $\mathbf{A_{BF}}$ should be used; we chose the pseudo-inversion based on Householder transformations. According to [2], its number of complex multiplications is estimated as

$$sN^2 - N^3/3, \tag{25}$$

and therefore, the total number of complex multiplications for the matrix inversion method is

$$.sN^2 - N^3/3 + MN\log_2 N + MN. \tag{26}$$

For the POCS-based algorithm, the impact of the number of iterations has been investigated. For different values s of the number of known data points, the algorithm is applied for different numbers of iterations n_{it}. Again, 15 different random selections of s known data points have been used and the errors averaged. The numerical results are summarized in Table 2. As expected, the error decreases with increasing number of given points. A sufficient number of given points is crucial for the performance of the algorithm. For a certain number of iterations, there is a number of given points (drop-off value) providing the desired accuracy (error below a small threshold). Fig. 6 illustrates this behavior for 200 iterations. One would expect that the drop-off val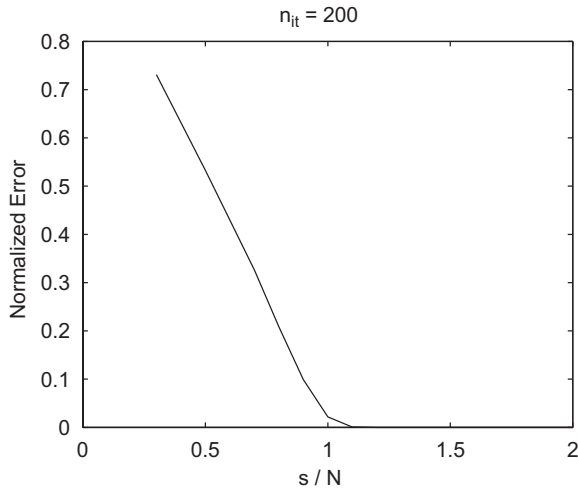ue is reached for values of s equal or higher than N. For such values, the sets $\mathbf{C}_l$ should intersect at a single point. When s is lower than N, the error takes higher values and cannot be reduced much by undertaking more iterations. This is expected since we have an

Table 1
Normalized error of the matrix inversion method for different numbers of given data points over 2D space

Number of given samples $= s$	Ave. norm. error
77	0.6928
128	0.5112
179	0.3104
205	0.2176
230	0.2240
256	0.0000
282	0.0000
307	0.0000
333	0.0000
512	0.0000
1024	0.0000
2048	0.0000
4096	0.0000
8192	0.0000
16 384	0.0000

Each normalized error is obtained by averaging the results of 15 simulations.

Table 2
Normalized error for different numbers of iterations n_{it} and given known data points s

s	Number of iterations n_{it}									
	10	20	30	50	100	200	300	500	1000	3000
77	0.7312	0.7312	0.7312	0.7312	0.7312	0.7312	0.7312	0.7312	0.7312	0.7312
128	0.5336	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328
179	0.3392	0.3288	0.3272	0.3264	0.3264	0.3264	0.3264	0.3264	0.3264	0.3264
205	0.2357	0.2183	0.2133	0.2101	0.2087	0.2084	0.2084	0.2083	0.2083	0.2083
230	0.1590	0.1304	0.1194	0.1098	0.1023	0.0991	0.0984	0.0981	0.0980	0.0980
256	0.1022	0.0708	0.0575	0.0441	0.0307	0.0216	0.0177	0.0138	0.0099	0.0053
282	0.0606	0.0311	0.0200	0.0104	0.0033	0.0007	0.0002	0.0000	0.0000	0.0000
307	0.0369	0.0127	0.0058	0.0017	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000
333	0.0201	0.0048	0.0015	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
512	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1024	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2048	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4096	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8192	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16 384	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

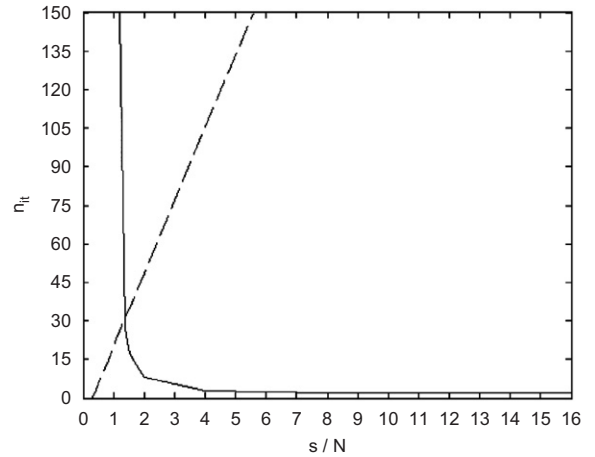Fig. 6. Normalized errors for different numbers of known samples at 200 iterations.

Fig. 7. Evaluation of POCS computational efficiency for different numbers of known samples s. Solid line: number of POCS iterations n_{it} needed to achieve normalized error <0.0005. Dashed lines: number of iterations for which POCS and matrix inversion methods give the same computational costs.

underdetermined system where the intersection of the sets C_l is a region rather than a point. Therefore, successive projections yield a solution rather far from the original diffraction field. In general, the error decreases by the number of iterations and after a certain number of iterations is reached, the error saturates. The algorithm converges much faster for larger values of s.

The computational complexity of the POCS-based algorithm is determined by the number of iterations n_{it}, the number of parallel lines M, and the number of sample points per line N. For each iteration, M lines are computed by Fourier domain operations (cf. Eq. (10)). $N \log_2 N + N$ complex multiplications are required for calculation of the diffraction field on a line when FFT is used, because FFT algorithm is used twice. This results in a total number of complex multiplications:

$$n_{it}(MN \log_2 N + MN). \tag{27}$$

Note that the number of given points s is not present in the complexity measure given by (27). However, this number influences the complexity indirectly, as it determines the number of iterations needed for achieving the desired accuracy. As shown in Fig. 7, more given points result in less iterations to achieve the same accuracy. The curve drops right after $s = N$ and after that point the desired error can be reached in a reasonable number of iterations. For example, less than 10 iterations are sufficient if starting with $s = 2N$ given points. In contrast, the matrix inversion algorithm gets more complex for higher s. To relate the computational

complexity of the POCS-based algorithm with that of direct matrix inversion, a second curve (the dashed line) is constructed in Fig. 7. For each s, it gives the number of iterations so that the computational complexity of the two algorithms is the same. Any number of iterations below that curve positions the POCS approach as the more preferable method. In Fig. 7, these iterations become less for $s \geqslant 1.4N$. Direct comparison of the number of the multiplications that both approaches require for different values of s are presented in Table 3. The values for the matrix inversion method are calculated using Eq. (26). The values for POCS method are found from Eq. (27). The parameter n_{it}, in Eq. (27), is determined by the POCS method that yields normalized error below 0.0005. Again, Table 3 shows that the POCS algorithm is less computationally costly when $s \geqslant 1.4N$. An additional peculiarity of the POCS approach is that, being an iterative algorithm, it is more robust to computational errors than direct matrix inversion, especially for high values of N.

Fig. 5 shows simulations for values of s close to N, with no visual difference between the results of the two methods.

4. Conclusion

In this work, the computation of the diffraction field from a set of distributed data points which may represent the abstract structure of an object has

Table 3
Comparison between the number of complex multiplications of POCS method and pseudo-inversion method (PINV) for different numbers of known samples s

s	256	282	358	384	410	512	1024	2048	4096	8192	16 384
PINV	1.18e + 7	1.35e + 7	1.85e + 7	2.02e + 7	2.19e + 7	2.86e + 7	6.21e + 7	1.29e + 8	2.63e + 8	5.32e + 8	1.07e + 9
POCS	8.55e + 9	1.29e + 8	1.61e + 7	1.13e + 7	9.17e + 6	4.72e + 6	1.77e + 6	1.18e + 6	1.18e + 6	1.18e + 6	1.18e + 6

The costs are shown in number of complex multiplications needed to achieve error below 0.0005.

been investigated. First, a discrete formulation of the relevant diffraction theory has been presented. In this formulation, fictionally periodic fields have been used where the period of the field is determined by the size of the field along the transverse direction. Two approaches for obtaining simultaneous solution of the diffraction field have been studied. The first attacks the problem by a direct matrix inversion approach while the second utilizes the POCS method to recover the desired diffraction field iteratively. Both algorithms converge to the desired field when the number of given samples is equal to or larger than the period but they may not converge to the desired field if the number of given data points is lower than the period. Computational complexity which is determined by the number of complex multiplications is another issue addressed in this work. The matrix inversion enjoys lower computational complexity for small numbers of initially given points while POCS is beneficial when there are more given points, when a few number of iterations becomes sufficient to achieve the desired accuracy.

Appendix

The proof that the matrix $\mathbf{A}$ is unitary is given below by showing that $\mathbf{A}^H \mathbf{A} = \mathbf{I}$ as follows:

$$
\begin{aligned}
\mathbf{A}^H \mathbf{A} = \mathbf{A}\mathbf{A}^H &= \mathbf{I} \\
&= (\mathbf{W}^{-1}\mathbf{H}_p\mathbf{W})(\mathbf{W}^{-1}\mathbf{H}_p\mathbf{W})^H \\
&= (\mathbf{W}^{-1}\mathbf{H}_p\mathbf{W})(\mathbf{W}^H\mathbf{H}_p^H\mathbf{W}^{-H}) \\
&= \mathbf{W}^{-1}\mathbf{H}_p N\mathbf{H}_p^H\mathbf{W}^{-H} \\
&= \mathbf{W}^{-1}N\mathbf{W}^{-H} \\
&= \mathbf{I}.
\end{aligned}
\tag{28}
$$

Another property of $\mathbf{A}$ is that

$$
\begin{aligned}
\mathbf{A}^l \mathbf{A}^j &= (\mathbf{W}^{-1}\mathbf{H}_{lp}\mathbf{W})(\mathbf{W}^{-1}\mathbf{H}_{jp}\mathbf{W}) \\
&= \mathbf{W}^{-1}\mathbf{H}_{lp}\mathbf{H}_{jp}\mathbf{W} \\
&= \mathbf{W}^{-1}\mathbf{H}_{(l+j)p}\mathbf{W},
\end{aligned}
\tag{29}
$$

where $\mathbf{H}_{(l+j)p}$ represents the kernel of the discrete system which is used to calculate the diffraction field at $p(l+j)$.

References

[1] R. Aharoni, Y. Censor, Block iterative projection methods for parallel computation of solutions to convex feasibility problems, Linear Algebra Appl. 120 (1989) 165–175.

[2] Å. Björck, Numerical Methods for Least Squares Problems, SIAM, Amsterdam, Holland, 1990.

[3] M. Born, E. Wolf, Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light, Cambridge University Press, New York, 1980.

[4] N. Delen, B. Hooker, Free-space beam propagation between arbitrarily oriented planes based on full diffraction theory: a fast Fourier transform approach, J. Opt. Soc. Am. A 15 (1998) 857–867.

[5] R.G. Dorsch, A.W. Lohmann, S. Sinzinger, Fresnel ping-pong algorithm for two-plane computer-generated hologram display, Appl. Opt. 33 (1994) 869–875.

[6] G.B. Esmer, Computation of holographic patterns between tilted planes, Master's Thesis, Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey, 2004.

[7] R.W. Gerchberg, W.O. Saxton, A practical algorithm for the determination of phase from image and diffraction plane pictures, Optik 35 (1972) 237–246.

[8] J.W. Goodman, Introduction to Fourier Optics, McGraw-Hill, New York, 1996.

[9] L.G. Gubin, B.T. Polyak, E.V. Raik, The method of projections for finding the common point of convex sets, USSR Comput. Math. Math. Phys. 7 (1967) 1–24.

[10] M.L. Huebschman, B. Munjuluri, H.R. Garner, Dynamic holographic 3-d image projection, Opt. Exp. 11 (2003) 437–445.

[11] E. Lalor, Conditions for the validity of the angular spectrum of plane waves, J. Opt. Soc. Am. 58 (1968) 1235–1237.

[12] M. Lucente, Diffraction-specific fringe computation for electro-holography, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1994 (online), ⟨http://www.lucente.us/pubs/PhDthesis/contents.html⟩.

[13] D. Mas, J. Pérez, C. Hernández, C. Vázquez, J.J. Miret, C. Illueca, Fast numerical calculation of fresnel patterns in convergent systems, Opt. Commun. 227 (2003) 245–258.

[14] K. Matsushima, H. Schimmel, F. Wyrowski, Fast calculation method for optical diffraction on tilted planes by use of

the angular spectrum of plane waves, J. Opt. Soc. A 20 (2003) 1755–1762.

[15] L. Onural, P.D. Scott, Digital decoding of in-line holograms, Opt. Eng. 26 (1987) 1124–1132.

[16] H.M. Ozaktas, A. Koc, I. Sari, Efficient computation of quadratic-phase integrals in optics, Opt. Lett. 31 (2006) 35–37.

[17] B.E.A. Saleh, M.C. Teich, Fundamentals of Photonics, Wiley, New York, 1991.

[18] G.C. Sherman, Application of the convolution theorem to Rayleigh's integral formulas, J. Opt. Soc. Am. 57 (1967) 546–547.

[19] T. Tommasi, B. Bianco, Computer-generated holograms of tilted planes by a spatial frequency approach, J. Opt. Soc. Am. A 10 (1993) 299–305.

[20] D.C. Youla, H. Webb, Image restoration by the method of convex projections: part I-theory, IEEE Trans. Med. Imag. TMI-1 (1982) 81–94.