A new DCT-based algorithm
for numerical reconstruction of electronically recorded holograms

Leonid Bilevich*[a] and Leonid Yaroslavsky**[a]
[a]Department of Physical Electronics, Faculty of Engineering,
Tel Aviv University, 69978, Tel Aviv, Israel[1]

ABSTRACT

A new universal low computational complexity algorithm for numerical reconstruction of holograms recorded in near diffraction zone is presented. The algorithm implements digital convolution in DCT domain, which makes it virtually insensitive to boundary effects. It can be used for reconstruction of holograms for arbitrary ratios of hologram size to the object-to-hologram distance and wavelength to camera pitch and allows image reconstruction in arbitrary scale.

1. INTRODUCTION

Numerical reconstruction of images from digital holograms is a fundamental task in digital holography. There exist several methods for hologram reconstruction. The most important methods for reconstruction of holograms, recorded in Fresnel zone, are Fourier reconstruction algorithm and Convolutional reconstruction algorithm [1]. If the image, obtained by hologram reconstruction, has to be scaled by scaling factor σ, then a certain scaling algorithm has to be applied to the reconstructed image.

A new reconstruction algorithm with simultaneous scaling is suggested. This algorithm is based on the Inverse Scaled Discrete Fresnel Transform (IScDFrT) that converts the problem of reconstruction and scaling to the problem of convolution. The convolution, in its turn, is computed through the boundary effects free DCT domain convolution consisting of DCT and IDCT transforms only.

The IScDFrT reconstruction algorithm is universal. We'll demonstrate that Fourier reconstruction algorithm and "Central Part" Convolutional reconstruction algorithm are special cases of the IScDFrT reconstruction algorithm.

2. HOLOGRAM RECONSTRUCTION WITH SCALING

2.1 Reconstruction of holograms recorded in near zone [1]

Images are reconstructed from near-zone recorded holograms using the Canonical Inverse Discrete Fresnel Transform (IDFrT):

$$\boldsymbol{a}_k = \frac{\mathbf{1}}{\sqrt{N}} \sum_{r=0}^{N-1} \alpha_r \exp\left(-i\pi \frac{\left(k\mu - r/\mu\right)^2}{N}\right), \tag{1}$$

where μ is a focusing parameter defined in terms of the wavelength λ, the hologram-to-observation plane distance Z, the number of hologram samples N and the hologram sampling interval Δf as:

$$\mu^2 = \lambda Z \Big/ \left[N\left(\Delta f\right)^2\right]. \tag{2}$$

There exist several algorithms implementing this reconstruction method, among them Fourier reconstruction algorithm, Convolution Discrete Fresnel Transform (ConvDFrT) and its modification "Central Part" Convolutional reconstruction algorithm. The formulae for these algorithms are listed below.

---

[1] *bilevich@eng.tau.ac.il; http://www.eng.tau.ac.il/~bilevich/
**yaro@eng.tau.ac.il; http://www.eng.tau.ac.il/~yaro/

- **Fourier reconstruction algorithm:**

$$\begin{aligned}
a_k &= \frac{1}{\sqrt{N}} \sum_{r=0}^{N-1} \alpha_r \exp\left[-i\pi \frac{\left(k\mu - r/\mu + \tilde{w}\right)^2}{N}\right] \\
&= \frac{1}{\sqrt{N}} \exp\left[-i\pi \frac{k\mu\left(k\mu + 2\tilde{w}\right)}{N}\right] \sum_{r=0}^{N-1} \left\{\alpha_r \exp\left[-i\pi \frac{\left(r/\mu - \tilde{w}\right)^2}{N}\right]\right\} \exp\left[i2\pi \frac{kr}{N}\right] \\
&= \exp\left[-i\pi \frac{k\mu\left(k\mu + 2\tilde{w}\right)}{N}\right] DFT\left\{\alpha_r \exp\left[-i\pi \frac{\left(r/\mu - \tilde{w}\right)^2}{N}\right]\right\},
\end{aligned} \tag{3}$$

where $\tilde{w}$ is the shift parameter defined as:

$$\tilde{w} = \frac{N}{2\mu}. \tag{4}$$

This algorithm is used for $\mu \geq 1$.

- **Convolution Discrete Fresnel Transform (ConvDFrT):**

$$\begin{aligned}
a_k &= \frac{1}{N} \sum_{s=0}^{N-1} \left[\sum_{r=0}^{N-1} \alpha_r \exp\left(-i2\pi \frac{\left(k - r + w\right)s}{N}\right)\right] \exp\left(i\pi \frac{\mu^2 s^2}{N}\right) \\
&= \sum_{r=0}^{N-1} \alpha_r \left[\frac{1}{N} \sum_{s=0}^{N-1} \exp\left(-i2\pi \frac{\left(k - r + w\right)s}{N}\right) \exp\left(i\pi \frac{\mu^2 s^2}{N}\right)\right] = \sum_{r=0}^{N-1} \alpha_r \,\mathrm{frincd}\left(N; \mu^2; k - r + w\right) \\
&= \alpha_r * \mathrm{frincd}\left(N; \mu^2; r + w\right),
\end{aligned} \tag{5}$$

where **frincd** is a discrete-frinc function defined by:

$$\begin{aligned}
\mathrm{frincd}\left(N; \mu^2; r + w\right) &= \frac{1}{N} \sum_{s=0}^{N-1} \exp\left(i\pi \frac{\mu^2 s^2}{N}\right) \exp\left(-i2\pi \frac{\left(r + w\right)s}{N}\right) = \frac{1}{N} \sum_{s=0}^{N-1} \exp\left(i\pi \frac{\left(\mu^2 s - 2w\right)s}{N}\right) \exp\left(-i2\pi \frac{rs}{N}\right) \\
&= \frac{1}{\sqrt{N}} IDFT\left[\exp\left(i\pi \frac{\left(\mu^2 s - 2w\right)s}{N}\right)\right],
\end{aligned} \tag{6}$$

and w is the shift parameter defined as:

$$w = \mu\tilde{w}. \tag{7}$$

This algorithm is used for $\mu \leq 1$.

- **“Central Part” Convolutional reconstruction algorithm:**

$$a_k = \frac{1}{\sqrt{N}} \exp\left[-i\pi \frac{\tilde{w}\left(2k/\mu + \tilde{w}\right)}{N}\right] \sum_{r=0}^{N-1} \left\{\alpha_r \exp\left[i2\pi \frac{r\tilde{w}}{\mu N}\right]\right\} \exp\left[-i\pi \frac{\left(k - r\right)^2}{\mu^2 N}\right]$$

$$= \frac{1}{\sqrt{N}} \exp\left[-i\pi \frac{\tilde{w}\left(2k/\mu + \tilde{w}\right)}{N}\right]\left[\left\{\alpha_r \exp\left[i2\pi \frac{r\tilde{w}}{\mu N}\right]\right\} * \exp\left[-i\pi \frac{r^2}{\mu^2 N}\right]\right]. \quad (8)$$

This algorithm is used for $\mu \le 1$.

Now let's present the new algorithm:

- **Inverse Scaled Discrete Fresnel Transform (IScDFrT):**

$$a_k = \frac{1}{\sqrt{N}} \sum_{r=0}^{N-1} \alpha_r \exp\left[-i\pi \frac{\left(k/\sigma - r + w\right)^2}{\mu^2 N}\right], \quad (9)$$

where α_r is the hologram, a_k is the image reconstructed from the hologram, σ is the scaling factor and w is the shift parameter. This algorithm is used for any μ.

The reconstructed image a_k can be represented as a convolution:

$$a_k = \frac{1}{\sqrt{N}} \exp\left(-i\pi \frac{k\left(k\left(1/\sigma - 1\right) + 2w\right)}{\mu^2 \sigma N}\right) \sum_{r=0}^{N-1} \left\{\alpha_r \exp\left[-i\pi \frac{\left(r\left(1 - 1/\sqrt{\sigma}\right) - w\right)\left(r\left(1 + 1/\sqrt{\sigma}\right) - w\right)}{\mu^2 N}\right]\right\} \exp\left[-i\pi \frac{\left(k - r\right)^2}{\mu^2 \sigma N}\right]$$

$$= \frac{1}{\sqrt{N}} \exp\left(-i\pi \frac{k\left(k\left(1/\sigma - 1\right) + 2w\right)}{\mu^2 \sigma N}\right)$$

$$\times \left[ZP_{\lceil \sigma N \rceil}\left\{\alpha_r \exp\left[-i\pi \frac{\left(r\left(1 - 1/\sqrt{\sigma}\right) - w\right)\left(r\left(1 + 1/\sqrt{\sigma}\right) - w\right)}{\mu^2 N}\right]\right\} * \exp\left[-i\pi \frac{r^2}{\mu^2 \sigma N}\right]\right], \quad (10)$$

where the zero-padding operator **ZP** is defined as follows: in case of signal enlargement ($\sigma > 1$) the hologram α_r is zero-padded to size $\lceil \sigma N \rceil$, in case of signal reduction ($\sigma < 1$) the hologram α_r is truncated to size $\lceil \sigma N \rceil$. The convolution, in its turn, is implemented by the complex "centered" DCT convolution algorithm described below.

Eq. (10) defines the IScDFrT reconstruction algorithm, and both Fourier and "Central Part" Convolutional reconstruction algorithms represent its special cases. Indeed, the case $\sigma = 1/\mu^2$ corresponds to the Fourier reconstruction algorithm:

$$a_k = \frac{1}{\sqrt{N}} \exp\left[-i\pi \frac{k\mu\left(k\mu + 2\tilde{w}\right)}{N}\right] \sum_{r=0}^{N-1} \left\{\alpha_r \exp\left[-i\pi \frac{\left(r/\mu - \tilde{w}\right)^2}{N}\right]\right\} \exp\left[i2\pi \frac{kr}{N}\right]$$

$$= \exp\left[-i\pi \frac{k\mu\left(k\mu + 2\tilde{w}\right)}{N}\right] DFT\left\{\alpha_r \exp\left[-i\pi \frac{\left(r/\mu - \tilde{w}\right)^2}{N}\right]\right\}, \quad (11)$$

while the case $\sigma = 1$ corresponds to the "Central Part" Convolutional reconstruction algorithm:

$$a_k = \frac{1}{\sqrt{N}} \exp\left[-i\pi \frac{\tilde{w}\left(2k/\mu + \tilde{w}\right)}{N}\right] \sum_{r=0}^{N-1} \left\{\alpha_r \exp\left[i2\pi \frac{r\tilde{w}}{\mu N}\right]\right\} \exp\left[-i\pi \frac{\left(k - r\right)^2}{\mu^2 N}\right]$$

$$= \frac{1}{\sqrt{N}} \exp\left[-i\pi \frac{\tilde{w}\left(2k/\mu + \tilde{w}\right)}{N}\right]\left[\left\{\alpha_r \exp\left[i2\pi \frac{r\tilde{w}}{\mu N}\right]\right\} * \exp\left[-i\pi \frac{r^2}{\mu^2 N}\right]\right]. \tag{12}$$

Now let's describe the DCT-domain convolution algorithm that is used in the implementation of the IScDFrT reconstruction algorithm.

3. BOUNDARY EFFECT SAFE DIGITAL CONVOLUTION IN DCT DOMAIN

The DCT-domain boundary-effects free convolution algorithm [2-4] is given by the following formula:

$$\tilde{a}_k = \sqrt{2N}\left[\mathbf{IDCT}\left(\mathbf{DCT}\left[a_k\right]\mathbf{Re}\left\{\mathbf{DFT}\left[\mathbf{ZP}_{2N}\left(h_k\right)\right]\right\}\right) + \mathbf{IDcST}\left(\mathbf{DCT}\left[a_k\right]\mathbf{Im}\left\{\mathbf{DFT}\left[\mathbf{ZP}_{2N}\left(h_k\right)\right]\right\}\right)\right], \tag{13}$$

where DFT, DCT, IDCT and IDcST are the Discrete Fourier Transform, Discrete Cosine Transform, Inverse Discrete Cosine Transform and Inverse Discrete Cosine-Sine Transform defined by:

$$\alpha_r^{(DFT)} = \mathbf{DFT}\left(a_k\right) \equiv \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} a_k \exp\left(i2\pi \frac{kr}{N}\right), \tag{14}$$

$$\alpha_r^{(DCT)} = \mathbf{DCT}\left(a_k\right) \equiv \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} a_k \cos\left(\pi \frac{\left(k+1/2\right)r}{N}\right), \tag{15}$$

$$a_k = \mathbf{IDCT}\left(\alpha_r^{(DCT)}\right) \equiv \frac{1}{\sqrt{2N}}\left\{\alpha_0^{(DCT)} + 2\sum_{r=1}^{N-1} \alpha_r^{(DCT)} \cos\left(\pi \frac{\left(k+1/2\right)r}{N}\right)\right\}, \tag{16}$$

$$a_k = \mathbf{IDcST}\left(\alpha_r^{(DST)}\right) \equiv \frac{1}{\sqrt{2N}}\left\{\left(-1\right)^k \alpha_N^{(DST)} + 2\sum_{r=1}^{N-1} \alpha_r^{(DST)} \sin\left(\pi \frac{\left(k+1/2\right)r}{N}\right)\right\}; \tag{17}$$

$\mathbf{Re}(\bullet)$ denotes the real part, $\mathbf{Im}(\bullet)$ denotes the imaginary party, and $\mathbf{ZP}_{2N}\left(h_k\right)$ denotes a zero-padding of $\{h_k\}$ to double length. This convolution, consisting of transforms of 4 different types, can be converted to the form consisting of DCT and IDCT transforms only. Let's present this convolution.

Let's start with the 1D convolution.

The "DCT-only" form of DCT convolution (1D case)

The "DCT-only" form of DCT convolution is given by:

$$\tilde{a}_k = \sqrt{\frac{N}{2}}\left\{IDCT\left[\alpha_r^{(C)}\eta_r^{(CI)}\right] + \left(-1\right)^k IDCT\left[\left\{\alpha_r^{(C)}\eta_r^{(SI)}\right\}_{N-r}\right]\right\}, \tag{18}$$

where

$$\alpha_r^{(C)} = DCT\left(a_k\right) \tag{19}$$

$$\eta_r^{(CI)} = \cos\left(\pi \frac{r}{2N}\right) DCT\left(h_k\right) + \sin\left(\pi \frac{r}{2N}\right)\left\{DCT\left(\left(-1\right)^k h_k\right)\right\}_{N-r} \tag{20}$$

$$\eta_r^{(SI)} = \cos\left(\pi \frac{r}{2N}\right)\left\{DCT\left(\left(-1\right)^k h_k\right)\right\}_{N-r} - \sin\left(\pi \frac{r}{2N}\right) DCT\left(h_k\right). \tag{21}$$

"Centered" DCT convolution (1D case)

The DCT convolution provides a result shifted by half a kernel size from its "true" position. The "centered" DCT convolution will compensate this shift. Suppose that the kernel h_k is of the size N_h not exceeding N (the size of signal a_k), that is, $N_h \le N$. Suppose that the size of the input image a_k is N and the size of the kernel h_k is N_h ($N_h \le N$). Let's denote by $\tilde{h}_k$ the zero-padded version of h_k:

$$\tilde{h}_k \equiv \begin{cases} h_k, & k = 0,\ldots,N_h - 1 \\ 0, & k = N_h,\ldots,N-1 \end{cases}. \tag{22}$$

Then we'll obtain the centered result $\tilde{a}_k$ using the previously described algorithm with the following modifications of the terms $\eta_r^{(CI)}$ and $\eta_r^{(SI)}$:

- Use $\tilde{h}_k$ instead of h_k.
- For odd N_h: change $\cos\left(\pi \frac{r}{2N}\right)$ to $\cos\left(\pi \frac{N_h r}{2N}\right)$, $\sin\left(\pi \frac{r}{2N}\right)$ to $\sin\left(\pi \frac{N_h r}{2N}\right)$.
- For even N_h: change $\cos\left(\pi \frac{r}{2N}\right)$ to $\cos\left(\pi \frac{(N_h+1) r}{2N}\right)$, $\sin\left(\pi \frac{r}{2N}\right)$ to $\sin\left(\pi \frac{(N_h+1) r}{2N}\right)$.

Complex convolution (1D case)

If the input signal, the kernel and the output signal consist of complex-valued samples:

$$a_k = a_k^{re} + i a_k^{im}, \quad h_k = h_k^{re} + i h_k^{im}, \quad \tilde{a}_k = \tilde{a}_k^{re} + i \tilde{a}_k^{im}, \tag{23}$$

then the complex convolution

$$\tilde{a}_k = \sum_{n=0}^{N-1} a_n h_{k-n} \tag{24}$$

can be computed in the following way:

$$\begin{aligned} \tilde{a}_k = {} & \sqrt{\frac{N}{2}} \left\{ IDCT\left[\alpha_r^{re,(C)} \eta_r^{re,(CI)} - \alpha_r^{im,(C)} \eta_r^{im,(CI)} \right] + (-1)^k \, IDCT\left[\left\{ \alpha_r^{re,(C)} \eta_r^{re,(SI)} - \alpha_r^{im,(C)} \eta_r^{im,(SI)} \right\}_{N-r} \right] \right\} \\ & + i \sqrt{\frac{N}{2}} \left\{ IDCT\left[\alpha_r^{re,(C)} \eta_r^{im,(CI)} + \alpha_r^{im,(C)} \eta_r^{re,(CI)} \right] + (-1)^k \, IDCT\left[\left\{ \alpha_r^{re,(C)} \eta_r^{im,(SI)} + \alpha_r^{im,(C)} \eta_r^{re,(SI)} \right\}_{N-r} \right] \right\}, \end{aligned} \tag{25}$$

where $\alpha_r^{re,(C)}, \alpha_r^{im,(C)}, \eta_r^{re,(CI)}, \eta_r^{im,(CI)}, \eta_r^{re,(SI)}, \eta_r^{im,(SI)}$ denote transforms of real/imaginary parts of the complex signal and kernel:

$$\alpha_r^{re,(C)} = DCT\left(a_k^{re}\right) \tag{26}$$

$$\alpha_r^{im,(C)} = DCT\left(a_k^{im}\right) \tag{27}$$

$$\eta_r^{re,(CI)} = \cos\left(\pi \frac{r}{2N}\right) DCT\left(h_k^{re}\right) + \sin\left(\pi \frac{r}{2N}\right) \left\{ DCT\left((-1)^k h_k^{re}\right) \right\}_{N-r} \tag{28}$$

$$\eta_r^{im,(CI)} = \cos\left(\pi \frac{r}{2N}\right) DCT\left(h_k^{im}\right) + \sin\left(\pi \frac{r}{2N}\right) \left\{ DCT\left((-1)^k h_k^{im}\right) \right\}_{N-r} \tag{29}$$

$$\eta_r^{re,(SI)} = \cos\left(\pi \frac{r}{2N}\right)\left\{DCT\left((-1)^k h_k^{re}\right)\right\}_{N-r} - \sin\left(\pi \frac{r}{2N}\right) DCT\left(h_k^{re}\right) \tag{30}$$

$$\eta_r^{im,(SI)} = \cos\left(\pi \frac{r}{2N}\right)\left\{DCT\left((-1)^k h_k^{im}\right)\right\}_{N-r} - \sin\left(\pi \frac{r}{2N}\right) DCT\left(h_k^{im}\right). \tag{31}$$

Now let's consider the 2D convolution that is a generalization of a 1D convolution.

The "DCT-only" form of DCT convolution (2D case)

The "DCT-only" form of DCT convolution is given by:

$$\tilde{a}_{k,l} = \frac{\sqrt{N_1 N_2}}{2}\left\{ IDCT\left(\alpha_{r,s}^{(C,C)}\eta_{r,s}^{(CI,CI)}\right) + (-1)^{k+l}\, IDCT\left[\left\{\alpha_{r,s}^{(C,C)}\eta_{r,s}^{(SI,SI)}\right\}_{N_1-r,N_2-s}\right]\right.$$

$$\left. + (-1)^k\, IDCT\left[\left\{\alpha_{r,s}^{(C,C)}\eta_{r,s}^{(SI,CI)}\right\}_{N_1-r,s}\right] + (-1)^l\, IDCT\left[\left\{\alpha_{r,s}^{(C,C)}\eta_{r,s}^{(CI,SI)}\right\}_{r,N_2-s}\right]\right\}, \tag{32}$$

where

$$\alpha_{r,s}^{(C,C)} = DCT\left(a_{k,l}\right) \tag{33}$$

$$\eta_{r,s}^{(CI,CI)} = \cos\left(\pi \frac{r}{2N_1}\right)\cos\left(\pi \frac{s}{2N_2}\right) DCT\left(h_{k,l}\right) + \sin\left(\pi \frac{r}{2N_1}\right)\cos\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^k h_{k,l}\right)\right\}_{N_1-r,s} \tag{34}$$

$$+ \cos\left(\pi \frac{r}{2N_1}\right)\sin\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^l h_{k,l}\right)\right\}_{r,N_2-s} + \sin\left(\pi \frac{r}{2N_1}\right)\sin\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^{k+l} h_{k,l}\right)\right\}_{N_1-r,N_2-s}$$

$$\eta_{r,s}^{(SI,SI)} = \sin\left(\pi \frac{r}{2N_1}\right)\sin\left(\pi \frac{s}{2N_2}\right) DCT\left(h_{k,l}\right) - \cos\left(\pi \frac{r}{2N_1}\right)\sin\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^k h_{k,l}\right)\right\}_{N_1-r,s} \tag{35}$$

$$- \sin\left(\pi \frac{r}{2N_1}\right)\cos\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^l h_{k,l}\right)\right\}_{r,N_2-s} + \cos\left(\pi \frac{r}{2N_1}\right)\cos\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^{k+l} h_{k,l}\right)\right\}_{N_1-r,N_2-s}$$

$$\eta_{r,s}^{(SI,CI)} = -\sin\left(\pi \frac{r}{2N_1}\right)\cos\left(\pi \frac{s}{2N_2}\right) DCT\left(h_{k,l}\right) + \cos\left(\pi \frac{r}{2N_1}\right)\cos\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^k h_{k,l}\right)\right\}_{N_1-r,s} \tag{36}$$

$$- \sin\left(\pi \frac{r}{2N_1}\right)\sin\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^l h_{k,l}\right)\right\}_{r,N_2-s} + \cos\left(\pi \frac{r}{2N_1}\right)\sin\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^{k+l} h_{k,l}\right)\right\}_{N_1-r,N_2-s}$$

$$\eta_{r,s}^{(CI,SI)} = -\cos\left(\pi \frac{r}{2N_1}\right)\sin\left(\pi \frac{s}{2N_2}\right) DCT\left(h_{k,l}\right) - \sin\left(\pi \frac{r}{2N_1}\right)\sin\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^k h_{k,l}\right)\right\}_{N_1-r,s} \tag{37}$$

$$+ \cos\left(\pi \frac{r}{2N_1}\right)\cos\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^l h_{k,l}\right)\right\}_{r,N-s} + \sin\left(\pi \frac{r}{2N_1}\right)\cos\left(\pi \frac{s}{2N_2}\right)\left\{DCT\left((-1)^{k+l} h_{k,l}\right)\right\}_{N_1-r,N_2-s}$$

"Centered" DCT convolution (2D case)

Suppose that the size of the input image $a_{k,l}$ is $N_1 \times N_2$ and the size of the kernel $h_{k,l}$ is $N_{h1} \times N_{h2}$ ($N_{h1} \le N_1, N_{h2} \le N_2$). The zero-padded kernel is defined as:

$$\tilde{h}_{k,l} \equiv \begin{cases} h_{k,l}, & k = 0,\dots,N_{h1}-1, \quad l = 0,\dots,N_{h2}-1 \\ 0, & k = N_{h1},\dots,N_1-1, \quad l = 0,\dots,N_{h2}-1 \\ 0, & k = 0,\dots,N_{h1}-1, \quad l = N_{h2},\dots,N_2-1 \\ 0, & k = N_{h1},\dots,N_1-1, \quad l = N_{h2},\dots,N_2-1 \end{cases} . \tag{38}$$

Then we'll obtain the centered result $\tilde{a}_{k,l}$ using the previously described algorithm with the following modifications of the terms $\eta_{r,s}^{(CI,CI)}$, $\eta_{r,s}^{(SI,SI)}$, $\eta_{r,s}^{(SI,CI)}$ and $\eta_{r,s}^{(CI,SI)}$:

- Use $\tilde{h}_{k,l}$ instead of $h_{k,l}$.
- For odd N_h: change $\cos\left(\pi \frac{r}{2N}\right)$ to $\cos\left(\pi \frac{N_h r}{2N}\right)$, $\sin\left(\pi \frac{r}{2N}\right)$ to $\sin\left(\pi \frac{N_h r}{2N}\right)$.
- For even N_h: change $\cos\left(\pi \frac{r}{2N}\right)$ to $\cos\left(\pi \frac{(N_h+1)r}{2N}\right)$, $\sin\left(\pi \frac{r}{2N}\right)$ to $\sin\left(\pi \frac{(N_h+1)r}{2N}\right)$.

  (For rows use N_{h1} instead of N_h and N_1 instead of N. For columns use N_{h2} instead of N_h and N_2 instead of N.)

Complex convolution (2D case)

If the input signal, the kernel and the output signal consist of complex-valued samples:

$$a_{k,l} = a_{k,l}^{re} + i a_{k,l}^{im}, \quad h_{k,l} = h_{k,l}^{re} + i h_{k,l}^{im}, \quad \tilde{a}_{k,l} = \tilde{a}_{k,l}^{re} + i \tilde{a}_{k,l}^{im}, \tag{39}$$

then the complex convolution

$$\tilde{a}_{k,l} = \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} a_{m,n} h_{k-m,l-n} \tag{40}$$

can be computed in the following way:

$$\begin{aligned} \tilde{a}_{k,l} = \frac{\sqrt{N_1 N_2}}{2} \Big\{ & IDCT\left(\left(\alpha_{r,s}^{re,(C,C)} + i\alpha_{r,s}^{im,(C,C)}\right)\left(\eta_{r,s}^{re,(CI,CI)} + i\eta_{r,s}^{im,(CI,CI)}\right)\right) \\ & + (-1)^{k+l}\, IDCT\left[\left\{\left(\alpha_{r,s}^{re,(C,C)} + i\alpha_{r,s}^{im,(C,C)}\right)\left(\eta_{r,s}^{re,(SI,SI)} + i\eta_{r,s}^{im,(SI,SI)}\right)\right\}_{N_1-r,N_2-s}\right] \\ & + (-1)^{k}\, IDCT\left[\left\{\left(\alpha_{r,s}^{re,(C,C)} + i\alpha_{r,s}^{im,(C,C)}\right)\left(\eta_{r,s}^{re,(SI,CI)} + i\eta_{r,s}^{im,(SI,CI)}\right)\right\}_{N_1-r,s}\right] \\ & + (-1)^{l}\, IDCT\left[\left\{\left(\alpha_{r,s}^{re,(C,C)} + i\alpha_{r,s}^{im,(C,C)}\right)\left(\eta_{r,s}^{re,(CI,SI)} + i\eta_{r,s}^{im,(CI,SI)}\right)\right\}_{r,N_2-s}\right]\Big\} \\ = \frac{\sqrt{N_1 N_2}}{2} \Big\{ & IDCT\left(\alpha_{r,s}^{re,(C,C)}\eta_{r,s}^{re,(CI,CI)} - \alpha_{r,s}^{im,(C,C)}\eta_{r,s}^{im,(CI,CI)}\right) \\ & + (-1)^{k+l}\, IDCT\left[\left\{\alpha_{r,s}^{re,(C,C)}\eta_{r,s}^{re,(SI,SI)} - \alpha_{r,s}^{im,(C,C)}\eta_{r,s}^{im,(SI,SI)}\right\}_{N_1-r,N_2-s}\right] \\ & + (-1)^{k}\, IDCT\left[\left\{\alpha_{r,s}^{re,(C,C)}\eta_{r,s}^{re,(SI,CI)} - \alpha_{r,s}^{im,(C,C)}\eta_{r,s}^{im,(SI,CI)}\right\}_{N_1-r,s}\right] \end{aligned}$$

$$+(-1)^l\ IDCT\left[\left\{\alpha_{r,s}^{re,(C,C)}\eta_{r,s}^{re,(CI,SI)}-\alpha_{r,s}^{im,(C,C)}\eta_{r,s}^{im,(CI,SI)}\right\}_{r,N_2-s}\right]\Bigg\}$$

$$+i\frac{\sqrt{N_1N_2}}{2}\Bigg\{IDCT\left(\alpha_{r,s}^{re,(C,C)}\eta_{r,s}^{im,(CI,CI)}+\alpha_{r,s}^{im,(C,C)}\eta_{r,s}^{re,(CI,CI)}\right)$$

$$+(-1)^{k+l}\ IDCT\left[\left\{\alpha_{r,s}^{re,(C,C)}\eta_{r,s}^{im,(SI,SI)}+\alpha_{r,s}^{im,(C,C)}\eta_{r,s}^{re,(SI,SI)}\right\}_{N_1-r,N_2-s}\right]$$

$$+(-1)^k\ IDCT\left[\left\{\alpha_{r,s}^{re,(C,C)}\eta_{r,s}^{im,(SI,CI)}+\alpha_{r,s}^{im,(C,C)}\eta_{r,s}^{re,(SI,CI)}\right\}_{N_1-r,s}\right]$$

$$+(-1)^l\ IDCT\left[\left\{\alpha_{r,s}^{re,(C,C)}\eta_{r,s}^{im,(CI,SI)}+\alpha_{r,s}^{im,(C,C)}\eta_{r,s}^{re,(CI,SI)}\right\}_{r,N_2-s}\right]\Bigg\}, \tag{41}$$

where the transforms of real/imaginary parts of signal and kernel are denoted by subscripts $(\bullet)^{re}/(\bullet)^{im}$ (similarly to the complex convolution for the 1D case).

4. RESULTS

Let's define the modified version of the IScDFrT with the scaling factor:

$$\tilde{\sigma}=\mu^2\sigma \tag{42}$$

and the shift parameter:

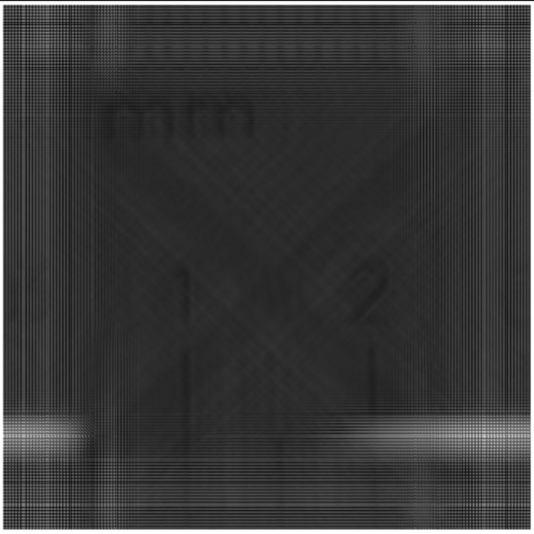
$$\tilde{w}=\frac{w}{\mu}. \tag{43}$$

Substituting $\sigma=\tilde{\sigma}/\mu^2$ and $w=\mu\tilde{w}$ into Eq. (9), we obtain the modified version of the IScDFrT:

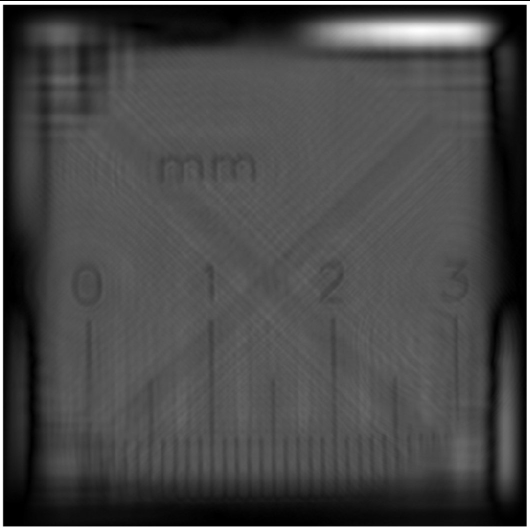
$$a_k=\sum_{r=0}^{N-1}\alpha_r\exp\left[-i\pi\frac{\left(k\mu/\tilde{\sigma}-r/\mu+\tilde{w}\right)^2}{N}\right]. \tag{44}$$

Let's illustrate the relative sizes of different images that are reconstructed from holograms with different methods and then show methods for removing "ringing artifacts".

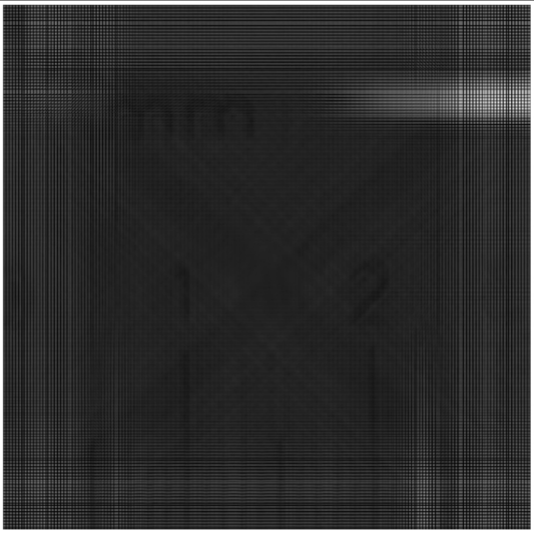
The modified IScDFrT with scaling factor $\tilde{\sigma}=1$ corresponds to the Fourier reconstruction algorithm (Figure 1(a,c)), and with scaling factor $\tilde{\sigma}=\mu^2$ it corresponds to the "Central Part" Convolutional reconstruction algorithm (Figure 1(c,d)). For reference, the ConvDFrT-reconstructed image (of size N) is shown in (Figure 1(b)). The central part (of size μ^2N) of the ConvDFrT reconstructed image is approximated by the "Central Part" Convolutional reconstruction algorithm. For comparison, the ConvDFrT reconstruction with scaling $\tilde{\sigma}=1/\mu^2$ is shown in Figure 1(d). The reconstruction and scaling were done in two separate steps; the scaling was done using sinc-interpolation [4].



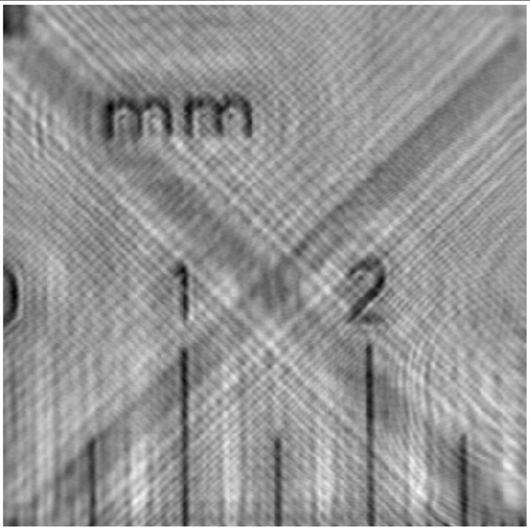
(a) Fourier reconstruction algorithm.



(b) ConvDFrT reconstruction algorithm.



(c) IScDFrT reconstruction algorithm ($\tilde{\sigma} = 1/\mu^2$).



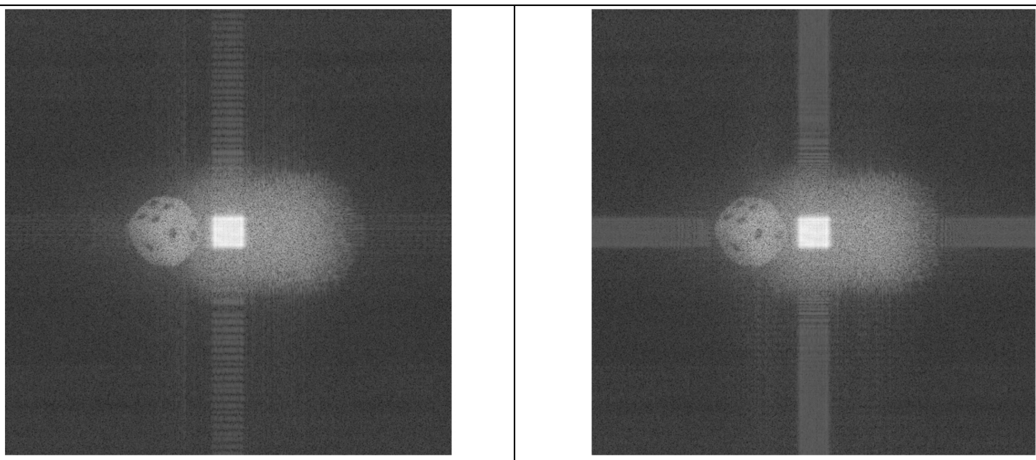
(d) ConvDFrT reconstruction algorithm followed by scaling by factor $\tilde{\sigma} = 1/\mu^2$.

Figure 1. Comparison of “raw” images reconstructed from holograms using Fourier, ConvDFrT and IScDFrT reconstruction algorithms ($\mu^2 = 0.6580$). (a) - Fourier reconstruction algorithm. (b) - ConvDFrT reconstruction algorithm. (c) – IScDFrT reconstruction algorithm ($\tilde{\sigma} = 1/\mu^2$). (d) - ConvDFrT reconstruction algorithm followed by scaling by factor $\tilde{\sigma} = 1/\mu^2$.

The sizes of image features in (a),(c) and (d) are identical. The large “ringing artifacts” are evident.

In Figure 1 the “raw” images with “ringing artifacts” were shown. The choice of method for elimination of “ringing” artifacts depends on the value of the focusing parameter μ .

Case 1: $\mu > 1$. In this case there are no “ringing artifacts” for both the IScDFrT and the Fourier reconstruction algorithms. The images reconstructed with IScDFrT with $\tilde{\sigma} = 1$ and with the Fourier reconstruction algorithm are shown in Figure 2.



(a) Fourier reconstruction algorithm. (b) IScDFrT reconstruction algorithm ($\tilde{\sigma} = 1$).

Figure 2. Comparison of images reconstructed from holograms for the case $\mu > 1$. The focusing parameter is: $\mu^2 = 14.0907$. (a) - Fourier reconstruction algorithm. (b) – IScDFrT reconstruction algorithm ($\tilde{\sigma} = 1$).

Case 2: $\mu < 1$. In this case the “ringing artifacts” are present in the “raw” reconstructed images, as was demonstrated in Figure 1. In order to eliminate these “ringing artifacts”, it is recommended to retain the central part of size $\lceil \mu^2 \tilde{\sigma} N \rceil$ of the reconstructed image. If the required size of the output image is N, then the reconstructed image has to be scaled by the factor $1/\mu^2$ and then truncated (pruned) to the size N. Also, it is recommended to eliminate the near-border pixels of the image that contain the residual “ringing artifacts”.

- For *Fourier reconstruction algorithm*: the image reconstruction is followed by scaling ($\tilde{\sigma} = 1/\mu^2$) and then pruning to size N:

$$a_k = ZM_\Delta \left\{ PRUN_N \left[SC_{1/\mu^2} \left(FourierRecon\left(\alpha_r\right)\right)\right]\right\}, \tag{45}$$

  where ***FourierRecon*** denotes the Fourier reconstruction algorithm, SC_{1/μ^2}, denotes scaling by the factor $1/\mu^2$, $PRUN_N$ denotes the extraction of N central samples of its argument and ZM_Δ denotes the zero-masking of the Δ leftmost and the Δ rightmost samples. The parameter Δ is small with respect to N (for example, in Figure 2 the following values were used: $\Delta = 50$ and $N = 512$).

- For *ConvDFrT reconstruction algorithm*: the image reconstruction is followed by scaling ($\tilde{\sigma} = 1/\mu^4$) and then pruning to size N:

$$a_k = PRUN_N \left[SC_{1/\mu^4} \left(ConvRecon\left(\alpha_r\right)\right)\right], \tag{46}$$

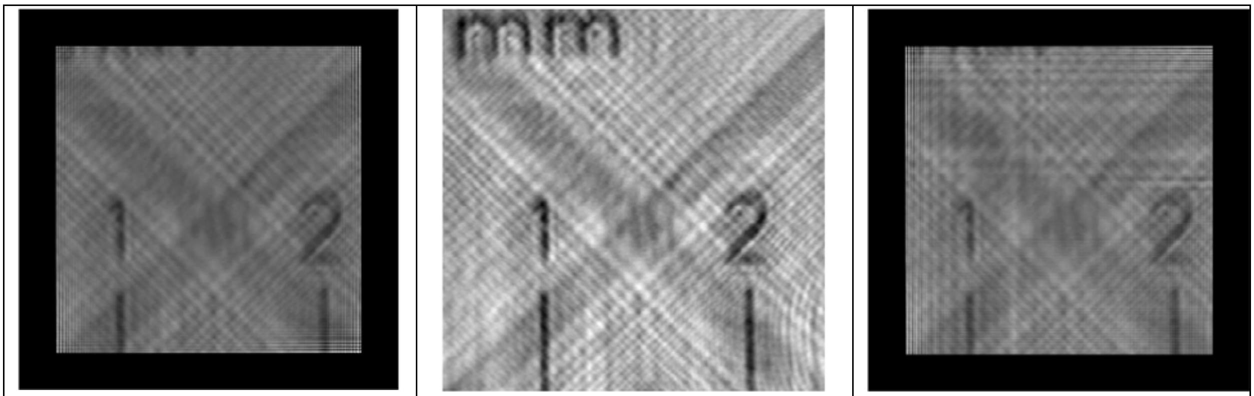
  where ***ConvRecon*** denotes the ConvDFrT reconstruction algorithm. The zero-masking is not needed. The scaling factor is $1/\mu^4$ and not $1/\mu^2$ due to the difference between the “scales” of the ConvDFrT reconstruction and the Fourier reconstruction, as was demonstrated in Figure 1.

- For *IScDFrT*: the image reconstruction is followed by pruning to size N:

$$a_k = ZM_\Delta \left\{ PRUN_N \left[IScDFrT\left(\alpha_r\right)\right]\right\}. \tag{47}$$

For this method (unlike the previous two) the "stand-alone" scaling operation SC_{1/μ^2} is not needed, because the scaling is embedded into the IScDFrT.

The results of reconstruction with algorithms Eqs. (45), (46), (47) are shown in Figure 3. For all 3 methods we obtain similar results. The Fourier reconstruction and the IScDFrT results are almost identical, while the ConvDFrT has larger field of view.



(a) Fourier reconstruction algorithm followed by scaling by factor $\tilde{\sigma} = 1/\mu^2$.

(b) ConvDFrT reconstruction algorithm followed by scaling by factor $\tilde{\sigma} = 1/\mu^4$.

(c) IScDFrT reconstruction method ($\tilde{\sigma} = 1/\mu^2$).

Figure 3. Comparison of images reconstructed from holograms for the case $\mu < 1$. The focusing parameter is: $\mu^2 = 0.6580$. (a) - Fourier reconstruction algorithm followed by scaling by factor $\tilde{\sigma} = 1/\mu^2$. (b) - Convolutional reconstruction algorithm followed by scaling by factor $\tilde{\sigma} = 1/\mu^4$. (c) - IScDFrT reconstruction algorithm ($\tilde{\sigma} = 1/\mu^2$).

5. CONCLUSIONS

The algorithm for the hologram reconstruction with simultaneous scaling is presented. The universality of this algorithm and its relation to the Fourier and "Central Part" Convolutional reconstruction algorithms is shown. The algorithm is implemented through the boundary effect-free DCT-domain convolution. Thanks to the availability of fast FFT-type algorithms for computing DCT and IDCT transforms involved in the reconstruction algorithm, the suggested IScDFrT reconstruction method represents a valuable alternative to DFT-domain reconstruction methods in real-time video processing applications.

6. REFERENCES

[1] Yaroslavsky, L., "Introduction to digital holography," in [*Digital Signal Processing in Experimental Research*], Yaroslavsky, L. and Astola, J., eds., *Bentham E-book Series*, 1–187 (2009).

[2] Yaroslavsky, L., "Discrete transforms, fast algorithms, and point spread functions of numerical reconstruction of digitally recorded holograms," in [*Advances in Signal Transforms: Theory and Applications*], Astola, J. and Yaroslavsky, L., eds., *EURASIP Book Series on Signal Processing and Communications* **7**, 93–141, Hindawi (2007).

[3] Yaroslavsky, L., "Boundary effect free and adaptive discrete signal sinc-interpolation algorithms for signal and image resampling," *Appl. Opt.* **42**, 4166–4175 (July 2003).

[4] Yaroslavsky, L., [*Digital Holography and Digital Image Processing: Principles, Methods, Algorithms*], Kluwer Academic Publishers (2004).