How fast can one numerically reconstruct digitally 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

Results of comparative study of the computational complexity of different algorithms for numerical reconstruction of electronically recorded holograms are presented and discussed. The following algorithms were compared: different types of Fourier and convolutional algorithms and a new universal DCT-based algorithm, in terms of the number of operations. Based on the comparison results, the feasibility of real-time implementation of numerical reconstruction of holograms is evaluated.

1. INTRODUCTION

Numerical reconstruction of digital holograms is a fundamental subject in digital holography. The choice of a certain reconstruction algorithm from a "toolbox" of available algorithms depends on the physical properties of the hologram, on the desired level of quality of the reconstructed image and on the amount of computational power that can be dedicated for the reconstruction process.

We review the different reconstruction algorithms, list their constituting computational units and compare their computational complexity.

2. IMAGE RECONSTRUCTION FROM HOLOGRAMS – REVIEW OF ALGORITHMS

2.1 Reconstruction of holograms recorded in far zone [1]

Images are reconstructed from near-zone recorded holograms using the Discrete Fourier Transform (DFT). There exist several algorithms implementing this reconstruction method, among them Canonical Discrete Fourier Transform and Scaled Discrete Fourier Transform. The formulae for these algorithms are listed below.

- **Canonical Discrete Fourier Transform (DFT):**

$$\boldsymbol{a_k = \frac{1}{\sqrt{N}}\sum_{r=0}^{N-1}\alpha_r \exp\left(i2\pi\frac{kr}{N}\right) = DFT\left(\alpha_r\right)}. \tag{1}$$

- **Scaled Discrete Fourier Transform (ScDFT):**

$$\boldsymbol{a_k = \frac{1}{\sqrt{N}}\sum_{r=0}^{N-1}\alpha_r \exp\left(i2\pi\frac{kr}{\sigma N}\right)}$$

[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/

$$= \frac{1}{\sqrt{N}} \exp\left(i\pi\frac{k^2}{\sigma N}\right) \sum_{r=0}^{N-1} \exp\left(i\pi\frac{r^2}{\sigma N}\right) \alpha_r \exp\left(-i\pi\frac{(k-r)^2}{\sigma N}\right)$$

$$= \frac{1}{\sqrt{N}} \exp\left(i\pi\frac{k^2}{\sigma N}\right) \sum_{r=0}^{\lceil \sigma N \rceil -1} ZP_{\lceil \sigma N \rceil}\left[\exp\left(i\pi\frac{r^2}{\sigma N}\right) \alpha_r\right] \exp\left(-i\pi\frac{(k-r)^2}{\sigma N}\right)$$

$$= \frac{1}{\sqrt{N}} \exp\left(i\pi\frac{k^2}{\sigma N}\right) \left\{ ZP_{\lceil \sigma N \rceil}\left[\exp\left(i\pi\frac{r^2}{\sigma N}\right) \alpha_r\right] * \exp\left(-i\pi\frac{r^2}{\sigma N}\right)\right\}, \qquad (2)$$

where ZP is a zero-padding operator defined as follows:

$$ZP_{\lceil \sigma N \rceil}\left[\alpha_r\right] = \begin{cases} \alpha_r, & r = 0,\ldots,N-1 \\ 0, & r = N,\ldots,\lceil \sigma N \rceil - 1 \end{cases}. \qquad (3)$$

(In case of signal reduction ($\sigma < 1$) the hologram α_r is truncated to size $\lceil \sigma N \rceil$.)

2.2 Reconstruction of holograms recorded in near zone [1]

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

$$a_k = \frac{1}{\sqrt{N}} \sum_{r=0}^{N-1} \alpha_r \exp\left(-i\pi\frac{(k\mu - r/\mu)^2}{N}\right), \qquad (4)$$

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:

$$\mu^2 = \lambda Z \Big/ \left[N(\Delta f)^2\right]. \qquad (5)$$

There exist several algorithms implementing this reconstruction method, among them Fourier reconstruction algorithm, Convolution Discrete Fresnel Transform, Convolution reconstruction algorithm, “Central Part” Convolution reconstruction algorithm and Inverse Scaled Discrete Fresnel Transform. The formulae for these algorithms are listed below.

- **Fourier reconstruction algorithm:**

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

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

$$= \exp\left[-i\pi\frac{k\mu(k\mu + 2\tilde{w})}{N}\right] DFT\left\{\alpha_r \exp\left[-i\pi\frac{(r/\mu - \tilde{w})^2}{N}\right]\right\}, \qquad (6)$$

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

$$\tilde{w} = \frac{N}{2\mu}. \tag{7}$$

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

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

$$a_k = \frac{1}{N}\sum_{s=0}^{N-1}\left[\sum_{r=0}^{N-1}\alpha_r \exp\left(-i2\pi\frac{(k-r+w)s}{N}\right)\right]\exp\left(i\pi\frac{\mu^2 s^2}{N}\right)$$
$$= \sum_{r=0}^{N-1}\alpha_r\left[\frac{1}{N}\sum_{s=0}^{N-1}\exp\left(-i2\pi\frac{(k-r+w)s}{N}\right)\exp\left(i\pi\frac{\mu^2 s^2}{N}\right)\right] = \sum_{r=0}^{N-1}\alpha_r\,\mathbf{frincd}\left(N;\mu^2;k-r+w\right)$$
$$= \alpha_r * \mathbf{frincd}\left(N;\mu^2;r+w\right), \tag{8}$$

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

$$\mathbf{frincd}\left(N;\mu^2;r+w\right) = \frac{1}{N}\sum_{s=0}^{N-1}\exp\left(i\pi\frac{\mu^2 s^2}{N}\right)\exp\left(-i2\pi\frac{(r+w)s}{N}\right) = \frac{1}{N}\sum_{s=0}^{N-1}\exp\left(i\pi\frac{\left(\mu^2 s-2w\right)s}{N}\right)\exp\left(-i2\pi\frac{rs}{N}\right)$$
$$= \frac{1}{\sqrt{N}}IDFT\left[\exp\left(i\pi\frac{\left(\mu^2 s-2w\right)s}{N}\right)\right], \tag{9}$$

and w is the shift parameter defined as:

$$w = \mu\tilde{w}. \tag{10}$$

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

- **Convolution reconstruction algorithm:**

$$a_k = \frac{1}{N}\sum_{s=0}^{N-1}\left[\sum_{r=0}^{N-1}\alpha_r \exp\left(-i2\pi\frac{(k-r+w)s}{N}\right)\right]\exp\left(i\pi\frac{\mu^2 s^2}{N}\right)$$
$$= \frac{1}{\sqrt{N}}\sum_{s=0}^{N-1}\left\{\left[\frac{1}{\sqrt{N}}\sum_{r=0}^{N-1}\alpha_r\exp\left(i2\pi\frac{rs}{N}\right)\right]\exp\left(i\pi\frac{\left(\mu^2 s-2w\right)s}{N}\right)\right\}\exp\left(-i2\pi\frac{ks}{N}\right)$$
$$= IDFT\left\{DFT\left(\alpha_r\right)\exp\left(i\pi\frac{\left(\mu^2 s-2w\right)s}{N}\right)\right\}. \tag{11}$$

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

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

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

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

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

Now let's present the new algorithm:

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

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

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

$$\times \left[ZP_{\lceil \sigma N \rceil}\left\{\alpha_r \exp\left[-i\pi \frac{\left(r\left(1 - 1/\sqrt{\sigma}\right) - w\right)\left(r\left(1 + 1/\sqrt{\sigma}\right) - w\right)}{\mu^2 N}\right]\right\} * \exp\left[-i\pi \frac{r^2}{\mu^2 \sigma N}\right]\right]. \tag{13}$$

This algorithm is used for any μ.

3. BASIC COMPUTATIONAL UNITS USED IN RECONSTRUCTION ALGORITHMS

We'll assume that the signal size N is a power of 2. All reconstruction algorithms consist of the following basic computational units:

- Complex exponent matrix generator,
- Complex matrix multiplier,
- DFT unit,
- Convolution unit.

The complex matrix multiplier needs N complex multiplications, that is, $4N$ real multiplications and $2N$ real additions, in total $6N$ real floating-point operations (flops).

The DFT unit operates on complex signals, producing complex result; it is implemented by the Fast Fourier Transform (FFT) algorithm that needs $3\frac{7}{9} N \log_2 N - 4\frac{16}{27} N$ flops [2]. For real-only input the real-data FFT algorithm with $1\frac{8}{9} N \log_2 N - 3\frac{8}{27} N$ flops is used. The benchmark comparing execution time of different implementations of the FFT algorithms running on different CPUs is presented in [3].

The Convolution unit can be implemented either in DFT domain by cyclic convolution or in DCT domain by the complex "centered" "all-DCT" convolution that is the version of the DCT convolution [4-6].

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

$$\tilde{a}_k = \sqrt{\frac{N}{2}} \left\{ IDCT\left[\alpha_r^{(C)} \eta_r^{(CI)}\right] + (-1)^k \, IDCT\left[\left\{\alpha_r^{(C)} \eta_r^{(SI)}\right\}_{N-r}\right] \right\}, \tag{14}$$

where:

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

$$\eta_r^{(CI)} = \cos\left(\pi\frac{r}{2N}\right) DCT\left(h_k\right) + \sin\left(\pi\frac{r}{2N}\right)\left\{DCT\left(\left(-1\right)^k h_k\right)\right\}_{N-r}, \tag{16}$$

$$\eta_r^{(SI)} = \cos\left(\pi\frac{r}{2N}\right)\left\{DCT\left(\left(-1\right)^k h_k\right)\right\}_{N-r} - \sin\left(\pi\frac{r}{2N}\right) DCT\left(h_k\right), \tag{17}$$

and IDCT is given by:

$$a_k = \mathrm{IDCT}\left(\alpha_r^{(DCT)}\right) \equiv \frac{1}{\sqrt{2N}}\left\{\alpha_0^{(DCT)} + 2\sum_{r=1}^{N-1}\alpha_r^{(DCT)}\cos\left(\pi\frac{\left(k+1/2\right)r}{N}\right)\right\}. \tag{18}$$

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

$$a_k = a_k^{re} + i a_k^{im}, \quad h_k = h_k^{re} + i h_k^{im}, \quad \tilde{a}_k = \tilde{a}_k^{re} + i\tilde{a}_k^{im}, \tag{19}$$

then the complex convolution

$$\tilde{a}_k = \sum_{n=0}^{N-1} a_n h_{k-n} \tag{20}$$

can be computed in the following way:

$$\begin{aligned}\tilde{a}_k = &\sqrt{\frac{N}{2}}\left\{IDCT\left[\alpha_r^{re,(C)}\eta_r^{re,(CI)} - \alpha_r^{im,(C)}\eta_r^{im,(CI)}\right] + \left(-1\right)^k IDCT\left[\left\{\alpha_r^{re,(C)}\eta_r^{re,(SI)} - \alpha_r^{im,(C)}\eta_r^{im,(SI)}\right\}_{N-r}\right]\right\}\\ &+ i\sqrt{\frac{N}{2}}\left\{IDCT\left[\alpha_r^{re,(C)}\eta_r^{im,(CI)} + \alpha_r^{im,(C)}\eta_r^{re,(CI)}\right] + \left(-1\right)^k IDCT\left[\left\{\alpha_r^{re,(C)}\eta_r^{im,(SI)} + \alpha_r^{im,(C)}\eta_r^{re,(SI)}\right\}_{N-r}\right]\right\},\end{aligned} \tag{21}$$

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:

$$\alpha_r^{re,(C)} = DCT\left(a_k^{re}\right), \tag{22}$$

$$\alpha_r^{im,(C)} = DCT\left(a_k^{im}\right), \tag{23}$$

$$\eta_r^{re,(CI)} = \cos\left(\pi\frac{r}{2N}\right) DCT\left(h_k^{re}\right) + \sin\left(\pi\frac{r}{2N}\right)\left\{DCT\left(\left(-1\right)^k h_k^{re}\right)\right\}_{N-r}, \tag{24}$$

$$\eta_r^{im,(CI)} = \cos\left(\pi\frac{r}{2N}\right) DCT\left(h_k^{im}\right) + \sin\left(\pi\frac{r}{2N}\right)\left\{DCT\left(\left(-1\right)^k h_k^{im}\right)\right\}_{N-r}, \tag{25}$$

$$\eta_r^{re,(SI)} = \cos\left(\pi\frac{r}{2N}\right)\left\{DCT\left(\left(-1\right)^k h_k^{re}\right)\right\}_{N-r} - \sin\left(\pi\frac{r}{2N}\right) DCT\left(h_k^{re}\right), \tag{26}$$

$$\eta_r^{im,(SI)} = \cos\left(\pi\frac{r}{2N}\right)\left\{DCT\left(\left(-1\right)^k h_k^{im}\right)\right\}_{N-r} - \sin\left(\pi\frac{r}{2N}\right) DCT\left(h_k^{im}\right). \tag{27}$$

The DCT operates on real signals, producing real output, it is implemented with $2N\log_2 N - N$ flops [7]. The fast implementation of the DCT convolution algorithm uses also two modified DFT units (real input, scaled twiddle-factors) with $1\frac{8}{9}N\log_2 N - 3\frac{8}{27}N$ flops each. The first modified DFT unit computes the spectra $\eta_r^{re,(CI)}$ and $\eta_r^{re,(SI)}$:

$$\eta_r^{re,(CI)} = 2\,\mathrm{Re}\left[PRUN_N\left\{DFT\left[ZP_{2N}\left(h_k^{re}\right)\right]\right\}\right] = \sqrt{2}\,\mathrm{Re}\left(\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} h_k^{re}\exp\left(i\pi\frac{kr}{N}\right)\right), \tag{28}$$

$$\eta_r^{re,(SI)} = 2\,\mathrm{Im}\left[PRUN_N\left\{DFT\left[ZP_{2N}\left(h_k^{re}\right)\right]\right\}\right] = \sqrt{2}\,\mathrm{Im}\left(\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} h_k^{re}\exp\left(i\pi\frac{kr}{N}\right)\right), \tag{29}$$

and the second modified DFT unit computes the spectra $\eta_r^{im,(CI)}$ and $\eta_r^{im,(SI)}$:

$$\eta_r^{im,(CI)} = 2\,\mathrm{Re}\left[PRUN_N\left\{DFT\left[ZP_{2N}\left(h_k^{im}\right)\right]\right\}\right] = \sqrt{2}\,\mathrm{Re}\left(\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} h_k^{im}\exp\left(i\pi\frac{kr}{N}\right)\right), \tag{30}$$

$$\eta_r^{im,(SI)} = 2\,\mathrm{Im}\left[PRUN_N\left\{DFT\left[ZP_{2N}\left(h_k^{im}\right)\right]\right\}\right] = \sqrt{2}\,\mathrm{Im}\left(\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} h_k^{im}\exp\left(i\pi\frac{kr}{N}\right)\right), \tag{31}$$

where $\mathrm{Re}(\bullet)$ denotes the real part, $\mathrm{Im}(\bullet)$ denotes the imaginary party, $ZP_{2N}(\bullet)$ denotes a zero-padding of its argument to double length and $PRUN_N(\bullet)$ denotes the pruning of its argument to size N.

The computational complexity $NumFlopsConv(N)$ of the Convolution unit for different types of convolution is listed in Table 1.

Table 1. Comparison of computational complexities of the fast convolution algorithms.

Type of the convolution	**Number of flops** $NumFlopsConv(N)$
Real cyclic convolution	$5\frac{2}{3}N\log_2 N - 3\frac{8}{9}N$
Real "centered" "all-DCT" convolution	$7\frac{8}{9}N\log_2 N - 5\frac{8}{27}N$
Complex cyclic convolution	$11\frac{1}{3}N\log_2 N - 7\frac{7}{9}N$
Complex "centered" "all-DCT" convolution	$15\frac{7}{9}N\log_2 N + 2\frac{11}{27}N$

4. COMPARISON OF DIFFERENT RECONSTRUCTION ALGORITHMS IN TERMS OF COMPUTATIONAL COMPLEXITY

The computational complexity of the reconstruction algorithms for different types of convolution is listed in Table 2. Notably, the computational complexities of the Inverse Scaled Discrete Fresnel Transform and of the Scaled Discrete Fourier Transform are equal.

Table 2. Comparison of computational complexities of the algorithms for image reconstruction from holograms.

Type of the reconstruction algorithm	**Number of flops**
Canonical Discrete Fourier Transform	$\mathbf{3\frac{7}{9}N\log_2 N - 4\frac{16}{27}N}$
Scaled Discrete Fourier Transform	$\boldsymbol{NumFlopsConv\left(\lceil \sigma N \rceil\right) + 6\lceil \sigma N \rceil + 6N}$
Fourier reconstruction algorithm	$\mathbf{3\frac{7}{9}N\log_2 N + 7\frac{11}{27}N}$
Convolution Discrete Fresnel Transform	$\boldsymbol{NumFlopsConv(N) + 3\frac{7}{9}N\log_2 N - 4\frac{16}{27}N}$
Convolution reconstruction algorithm	$\mathbf{7\frac{5}{9}N\log_2 N - 3\frac{5}{27}N}$
“Central Part” Convolution reconstruction algorithm	$\boldsymbol{NumFlopsConv\left(N\right) + 12N}$
Inverse Scaled Discrete Fresnel Transform	$\boldsymbol{NumFlopsConv\left(\lceil \sigma N \rceil\right) + 6\lceil \sigma N \rceil + 6N}$

5. CONCLUSIONS

Computational complexities of different hologram reconstruction algorithms are compared. The Inverse Scaled Discrete Fresnel Transform, performing hologram reconstruction with simultaneous scaling, presents an attractive alternative to other reconstruction methods that perform reconstruction and scaling in two consecutive steps. Thanks to the availability of fast FFT-type algorithms for computing DFT, DCT and IDCT transforms involved in the IScDFrT reconstruction algorithm, the latter represents a valuable alternative to DFT-domain no-scale hologram reconstruction algorithms in real-time video processing applications.

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] Johnson, S. G. and Frigo, M., “A modified split-radix FFT with fewer arithmetic operations,” *IEEE Trans. Signal Process.* **55**, 111–119 (Jan. 2007).

[3] Frigo, M. and Johnson, S. G., “The design and implementation of FFTW3,” *Proc. IEEE* **93**, 216–231 (Feb. 2005).

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

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

[6] 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).

[7] Plonka, G. and Tasche, M., “Fast and numerically stable algorithms for discrete cosine transforms,” *Linear Algebra Appl.* **394**, 309–345 (Jan. 1, 2005).