Fast DCT-based algorithm for signal and image accurate scaling

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 DCT-based algorithm for signal and image scaling by arbitrary factor is presented. The algorithm is virtually free of boundary effects and implements the discrete sinc-interpolation, which preserves the spectral content of the signal, and therefore is free from interpolation errors. Being implemented through the fast FFT-type DCT algorithm, the scaling algorithm has computational complexity of $\mathbf{O}(\log\lfloor\sigma N\rfloor)$ operations per output sample, where N and $\lfloor\sigma N\rfloor$ are number of signal input and output samples, correspondingly.

Keywords: scaling, DCT, convolution, fast algorithm

1. INTRODUCTION

Scaling is one of basic image processing tasks with wide range of applications. Presently, for image scaling available are spline methods[1-3] and discrete sinc-interpolation methods implemented as a convolution in DFT domain[4-6]. The spline methods work in signal domain and tend to introduce signal blurring. The discrete sinc-interpolation methods are potentially perfectly accurate but heavily suffer from boundary effects. We propose a novel DCT-based scaling algorithm that implements the ideal discrete sinc-interpolation and at the same time is very substantially less vulnerable to boundary effects. The algorithm is derived and described in Section 2. The results of its experimental verification are reported in Section 3 and conclusions are outlined in Section 4.

2. DCT-DOMAIN SCALING ALGORITHM

2.1 Inverse Scaled Discrete Cosine Transform (IScDCT)

Discrete sinc-interpolation methods are based on inverse Scaled DFT of DFT spectrum of the original signal[4,5]:

$$\tilde{a}_k = \frac{1}{\sqrt{\lfloor\sigma N\rfloor}} \sum_{r=0}^{\lfloor\sigma N\rfloor-1} \alpha_r^{ZP} \exp\left(-i2\pi\frac{kr}{\sigma N}\right), \qquad (1)$$

where $\tilde{a}_k$ is a σ-times scaled signal, $\lfloor\bullet\rfloor$ is a rounding operator, which rounds real numbers off:

$\sigma \ge 1$	$\sigma < 1$
$\lfloor x\rfloor = \mathrm{CEIL}(x)$	$\lfloor x\rfloor = \mathrm{FLOOR}(x)$

, (2)

α_r^{ZP} is the zero-padded or truncated DFT spectrum α_r:

$\sigma \ge 1$	$\sigma < 1$
$\alpha_r^{ZP} = \begin{cases} \alpha_r & \text{if} \quad r = 0,\ldots,N-1, \\ 0 & \text{if} \quad r = N,\ldots,\lfloor\sigma N\rfloor-1 \end{cases}$	$\alpha_r^{ZP} = \{\alpha_r \quad \text{if} \quad r = 0,\ldots,\lfloor\sigma N\rfloor-1$

(3)

[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 /

and α_r is the DFT spectrum of the input signal a_n:

$$\alpha_r = \mathrm{DFT}(a_n) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} a_n \exp\left(i2\pi \frac{nr}{N} \right). \tag{4}$$

The resulting scaled signal is shifted with respect to the original signal. In order to correct this shift Eq. (1) has to be modified to map the central sample of the input signal to the central sample of the output signal. The resulting ***Centered Inverse Scaled Discrete Fourier Transform*** is defined as follows:

$$\tilde{a}_k = \frac{1}{\sqrt{\lfloor \sigma N \rfloor}} \sum_{r=0}^{\lfloor \sigma N \rfloor -1} \alpha_r^{ZP} \exp\left\{ -i2\pi \frac{\left[k-(\sigma-1)/2-\left(\lfloor \sigma N \rfloor - \sigma N\right)/2\right] r}{\sigma N} \right\}. \tag{5}$$

The Centered Inverse Scaled DFT (Eq. (5)) can be computed through a convolution in DFT domain using Fast Fourier Transforms[4-6]. This implementation, however, suffers from border artifacts caused by the periodic nature of the DFT. A simple method to eliminate these artifacts (resulting from the mismatch between left and right borders of the input signal a_n) is to extend the signal a_n to double length by means of its mirror-reflection (guaranteeing the continuity of the resulting signal a_n^{MR} at the boundary points)[7] and to use this mirror-reflected signal a_n^{MR} as input to the Centered Inverse Scaled Discrete Fourier Transform (Eq. (5)). In this way we arrive to the following algorithm:

$$\tilde{a}_k = \frac{1}{\sqrt{2\lfloor \sigma N \rfloor}} \sum_{r=0}^{2\lfloor \sigma N \rfloor -1} \hat{\alpha}_r^{ZP} \exp\left\{ -i2\pi \frac{\left[k-(\sigma-1)/2-\left(\lfloor \sigma N \rfloor - \sigma N\right)/2\right] r}{2\sigma N} \right\}, \tag{6}$$

where $\lfloor \bullet \rfloor$ is a rounding operator (Eq. (2)), $\hat{\alpha}_r^{ZP}$ is the zero-padded or truncated DFT spectrum $\hat{\alpha}_r$:

$\sigma \ge 1$	$\sigma < 1$
$\hat{\alpha}_r^{ZP} = \begin{cases} \hat{\alpha}_r & \text{if} \quad r = 0,\ldots,2N-1, \\ 0 & \text{if} \quad r = 2N,\ldots,2\lfloor \sigma N \rfloor -1 \end{cases}$	$\hat{\alpha}_r^{ZP} = \left\{ \hat{\alpha}_r \quad \text{if} \quad r = 0,\ldots,2\lfloor \sigma N \rfloor -1 \right.$

, (7)

$\hat{\alpha}_r$ is the DFT spectrum of the signal a_n^{MR}:

$$\hat{\alpha}_r = \mathrm{DFT}\left(a_n^{MR}\right) = \frac{1}{\sqrt{2N}} \sum_{n=0}^{2N-1} a_n^{MR} \exp\left(i2\pi \frac{nr}{2N} \right). \tag{8}$$

and a_n^{MR} is extended by mirror-reflection input signal a_n:

$$a_n^{MR} = \begin{cases} a_n & \text{if} \quad n = 0,\ldots,N-1, \\ a_{2N-1-n} & \text{if} \quad n = N,\ldots,2N-1 \end{cases}. \tag{9}$$

One can show that, under constraint that the scaled signal $\tilde{a}_k$ has to be real-valued, Eq. (6) is reduced to the ***Inverse Scaled Discrete Cosine Transform (IScDCT)*** defined as follows:

$$\tilde{a}_k = \frac{1}{\sqrt{2\lfloor \sigma N \rfloor}} \left\{ \alpha_0^{C,ZP} + 2 \sum_{r=1}^{\lfloor \sigma N \rfloor -1} \alpha_r^{C,ZP} \cos\left\{ \pi \frac{\left[k+1/2-\left(\lfloor \sigma N \rfloor - \sigma N\right)/2\right] r}{\sigma N} \right\} \right\}, \tag{10}$$

where $\tilde{a}_k$ is a σ-times scaled signal, $\lfloor \bullet \rfloor$ is a rounding operator (Eq. (2)), $\alpha_r^{C,ZP}$ is the zero-padded or truncated DCT spectrum α_r^C:

$\sigma \ge 1$	$\sigma < 1$
$\alpha_r^{C,ZP} = \begin{cases} \alpha_r^C & \text{if} \quad r = 0,\ldots,N-1, \\ 0 & \text{if} \quad r = N,\ldots,\lfloor \sigma N \rfloor -1 \end{cases}$	$\alpha_r^{C,ZP} = \left\{ \alpha_r^C \quad \text{if} \quad r = 0,\ldots,\lfloor \sigma N \rfloor -1 \right.$

(11)

and α_r^C is the DCT spectrum of the input signal a_n:

$$\alpha_r^C = \mathbf{DCT}(a_n) = \sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} a_n \cos\left[\pi\frac{(n+1/2)r}{N}\right]. \qquad (12)$$

For $\sigma = 1$ the Eq. (10) coincides with the IDCT:

$$\tilde{a}_k = \mathbf{IDCT}(\alpha_r^C) = \frac{1}{\sqrt{2N}}\left\{\alpha_0^C + 2\sum_{r=1}^{N-1}\alpha_r^C \cos\left[\pi\frac{(k+1/2)r}{N}\right]\right\}. \qquad (13)$$

One can show that point spread function (PSF) of signal scaling algorithm defined by Eq. (10) is the discrete sinc function. In order to improve the speed of convergence of the scaling PSF to zero, it is recommendable to modify the scaling algorithm (Eq. (10)) by halving the last non-zero coefficient in Eq. (11)[4]. The modified in this way ***"Convergent" Inverse Scaled Discrete Cosine Transform*** is given by the following formula:

$$\tilde{a}_k = \sqrt{\frac{2}{N}}\sum_{r=0}^{\lfloor\sigma N\rfloor-1}\alpha_r^{C(1/2),ZP}\cos\left\{\pi\frac{[k+1/2-(\lfloor\sigma N\rfloor-\sigma N)/2]r}{\sigma N}\right\}, \qquad (14)$$

where $\alpha_r^{C(1/2),ZP}$ is the zero-padded or truncated DCT spectrum α_r^C:

$\sigma \geq 1$	$\sigma < 1$
$\alpha_r^{C(1/2),ZP} = \begin{cases} \alpha_r^C/2 & \text{if} \quad r = 0, \\ \alpha_r^C & \text{if} \quad r = 1,\ldots,N-2, \\ \alpha_r^C/2 & \text{if} \quad r = N-1, \\ 0 & \text{if} \quad r = N,\ldots,\lfloor\sigma N\rfloor - 1 \end{cases}$	$\alpha_r^{C(1/2),ZP} = \begin{cases} \alpha_r^C/2 & \text{if} \quad r = 0, \\ \alpha_r^C & \text{if} \quad r = 1,\ldots,\lfloor\sigma N\rfloor-2, \\ \alpha_r^C/2 & \text{if} \quad r = \lfloor\sigma N\rfloor - 1 \end{cases}$

(15)

and α_r^C is the DCT spectrum of the input signal a_n given by Eq. (12).

Note that for integer σ the Eq. (14) represents the DCT-domain zero-padding scaling algorithm[4,8].

2.2 Interpolation kernel and frequency response

In order to evaluate the resampling quality of scaling algorithm, consider the interpolation kernel (i.e., interpolation filter PSF) and frequency response of the DFT-based and of the suggested DCT-based scaling algorithms.

DFT-based scaling. One can show that from Eq. (5) it follows that that the scaled signal $\tilde{a}_k$ is related to the input signal a_n according to the following formula[4-6]:

$$\tilde{a}_k = \sum_{n=0}^{N-1} a_n\,\mathbf{sincd}\left(N^0-1, N, \frac{k+1/2}{\sigma} - n - 1/2 - \frac{\lfloor\sigma N\rfloor-\sigma N}{2\sigma}\right)\cos\left[\frac{\pi}{N}\left(\frac{k+1/2}{\sigma} - n - 1/2 - \frac{\lfloor\sigma N\rfloor-\sigma N}{2\sigma}\right)\right], \qquad (16)$$

where:

$$N^0 = \min(N, \lfloor\sigma N\rfloor) \qquad (17)$$

and the interpolation PSF is represented by the discrete sinc function $\mathbf{sincd}$:

$$\mathbf{sincd}(M, N, x) = \frac{\sin\left(\dfrac{\pi M}{N}x\right)}{N\sin\left(\dfrac{\pi}{N}x\right)}. \qquad (18)$$

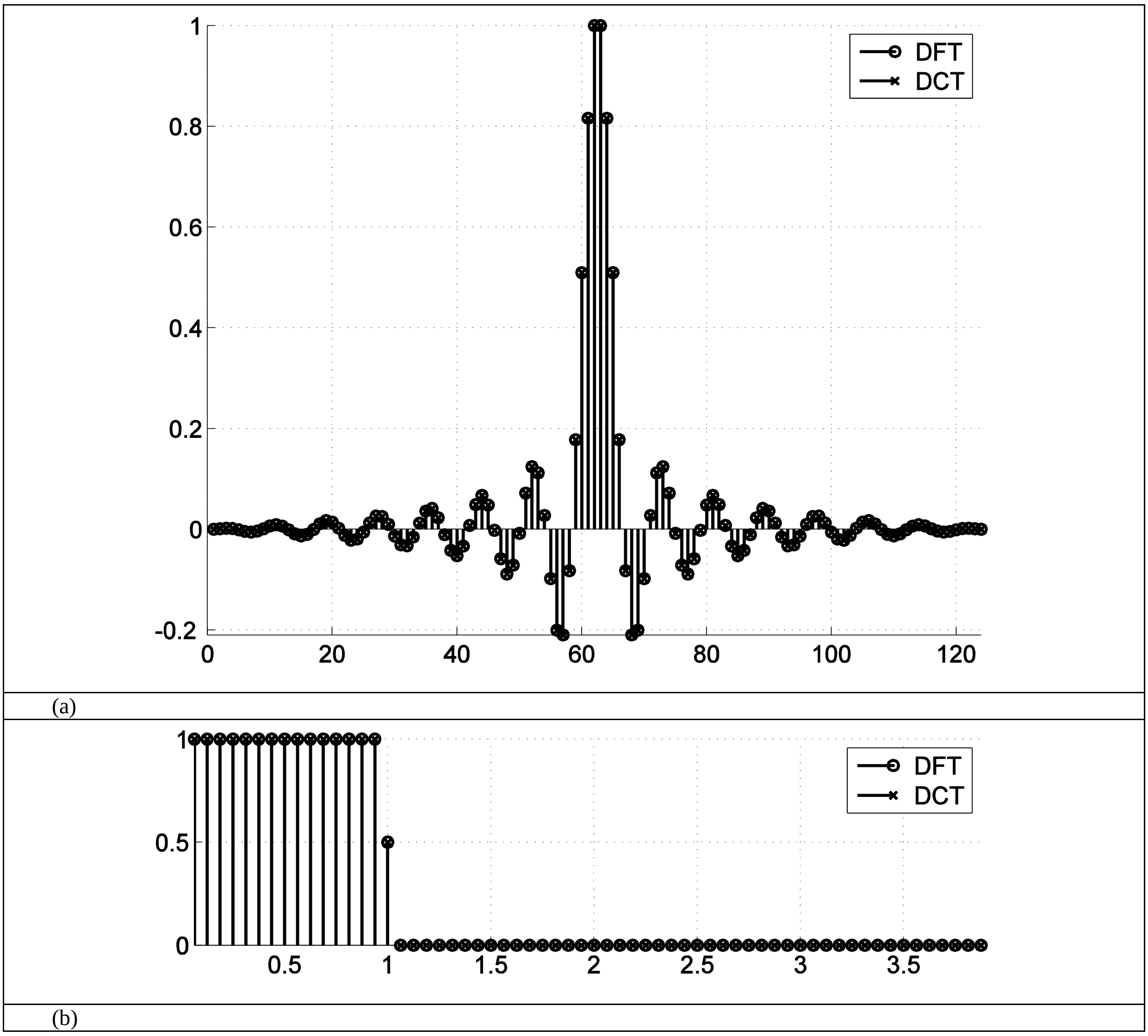
<u>Notes:</u> For computation of Eq. (16) the coefficients α_0 and α_{N^0-1} were halved. The exponential factor was discarded from Eq. (16).

DCT-based scaling. One can show that from Eq. (14) it follows that the scaled signal $\tilde{a}_k$ is related to the input signal a_k according to the following formula:

$$\begin{aligned}\tilde{a}_k = \sum_{n=0}^{N-1} a_n \Bigg\{ & \mathrm{sincd}\left(2N^0 - 2, 2N, \frac{k+1/2}{\sigma} - n - 1/2 - \frac{\lfloor \sigma N \rfloor - \sigma N}{2\sigma}\right) \cos\left[\frac{\pi}{2N}\left(\frac{k+1/2}{\sigma} - n - 1/2 - \frac{\lfloor \sigma N \rfloor - \sigma N}{2\sigma}\right)\right] \\ & + \mathrm{sincd}\left(2N^0 - 2, 2N, \frac{k+1/2}{\sigma} + n + 1/2 - \frac{\lfloor \sigma N \rfloor - \sigma N}{2\sigma}\right) \cos\left[\frac{\pi}{2N}\left(\frac{k+1/2}{\sigma} + n + 1/2 - \frac{\lfloor \sigma N \rfloor - \sigma N}{2\sigma}\right)\right]\Bigg\}, \quad (19)\end{aligned}$$

in which the interpolation PSF is represented by the term in curly brackets.

Both PSFs are compared in Figure 1(a) for the case $N = 31$, $\sigma = 4$, their DFTs (or interpolation filter frequency responses) are presented in Figure 1(b). As one can see from the figures, the PSFs and frequency responses of DFT-based and DCT-based interpolations are identical, so both preserve frequency content of signals.



(a)

(b)

Figure 1. Scaling of 1D signal by DCT-domain and DFT-domain scaling algorithms. PSFs (a) and frequency responses (b).

2.3 Implementation through DCT-domain convolution

For efficient computation, Eq. (14) can be converted into four convolutions:

$$\begin{aligned}\tilde{a}_k = \sqrt{\frac{2}{N}}\cos\left[\frac{\pi}{2\sigma N}k^2\right]&\left\{\begin{array}{l}\sum\limits_{r=0}^{\lfloor\sigma N\rfloor-1}\left\{\alpha_r^{C(1/2),ZP}\cos\left[\frac{\pi}{2\sigma N}\left(r^2+\left(1-\lfloor\sigma N\rfloor+\sigma N\right)r\right)\right]\right\}\cos\left[\frac{\pi}{2\sigma N}(k-r)^2\right]\\ +\sum\limits_{r=0}^{\lfloor\sigma N\rfloor-1}\left\{\alpha_r^{C(1/2),ZP}\sin\left[\frac{\pi}{2\sigma N}\left(r^2+\left(1-\lfloor\sigma N\rfloor+\sigma N\right)r\right)\right]\right\}\sin\left[\frac{\pi}{2\sigma N}(k-r)^2\right]\end{array}\right\}\\ +\sqrt{\frac{2}{N}}\sin\left[\frac{\pi}{2\sigma N}k^2\right]&\left\{\begin{array}{l}\sum\limits_{r=0}^{\lfloor\sigma N\rfloor-1}\left\{\alpha_r^{C(1/2),ZP}\cos\left[\frac{\pi}{2\sigma N}\left(r^2+\left(1-\lfloor\sigma N\rfloor+\sigma N\right)r\right)\right]\right\}\sin\left[\frac{\pi}{2\sigma N}(k-r)^2\right]\\ -\sum\limits_{r=0}^{\lfloor\sigma N\rfloor-1}\left\{\alpha_r^{C(1/2),ZP}\sin\left[\frac{\pi}{2\sigma N}\left(r^2+\left(1-\lfloor\sigma N\rfloor+\sigma N\right)r\right)\right]\right\}\cos\left[\frac{\pi}{2\sigma N}(k-r)^2\right]\end{array}\right\}\end{aligned}. \qquad (20)$$

In order to avoid boundary effects which are characteristic for cyclic convolution performed using FFT, one can perform convolution in DCT domain rather than in DFT domain[7]. The convolutions in Eq. (20) can be computed in DCT domain using the formula:

$$b_k = \sqrt{\frac{\lfloor\sigma N\rfloor}{2}}\left\{\mathrm{IDCT}\left[\xi_p^C\eta_p^{CI}\right]+\mathrm{IDcST}\left[\xi_p^S\eta_p^{CI}\right]\right\}, \qquad (21)$$

where ξ_p^C is the DCT (Discrete Cosine Transform) of α_r:

$$\xi_p^C = \mathrm{DCT}(\alpha_r) = \sqrt{\frac{2}{\lfloor\sigma N\rfloor}}\sum_{r=0}^{\lfloor\sigma N\rfloor-1}\alpha_r\cos\left[\pi\frac{(r+1/2)p}{\lfloor\sigma N\rfloor}\right], \qquad (22)$$

ξ_p^S is the DcST (Discrete Sine Transform) of α_r:

$$\xi_p^S = \mathrm{DcST}(\alpha_r) = \sqrt{\frac{2}{\lfloor\sigma N\rfloor}}\sum_{r=0}^{\lfloor\sigma N\rfloor-1}\alpha_r\sin\left[\pi\frac{(r+1/2)p}{\lfloor\sigma N\rfloor}\right], \qquad (23)$$

η_p^{CI} is the so called Discrete Cosine Transform – type I ($\mathrm{DCT}^{\mathrm{I}}$) of h_r (assuming that $h_{\lfloor\sigma N\rfloor}=0$):

$$\eta_p^{CI} = \mathrm{DCT}^I(h_r) = \frac{1}{2\sqrt{\lfloor\sigma N\rfloor}}\left[h_0+(-1)^p h_{\lfloor\sigma N\rfloor}+2\sum_{r=1}^{\lfloor\sigma N\rfloor-1}h_r\cos\left(\pi\frac{r\,p}{\lfloor\sigma N\rfloor}\right)\right], \qquad (24)$$

IDCT is the Inverse Discrete Cosine Transform:

$$a_k = \mathrm{IDCT}(\alpha_p) = \frac{1}{\sqrt{2\lfloor\sigma N\rfloor}}\left\{\alpha_0+2\sum_{p=1}^{\lfloor\sigma N\rfloor-1}\alpha_p\cos\left[\pi\frac{(k+1/2)p}{\lfloor\sigma N\rfloor}\right]\right\} \qquad (25)$$

and IDcST is the Inverse Discrete Sine Transform:

$$a_k = \mathrm{IDcST}(\alpha_p) = \frac{1}{\sqrt{2\lfloor\sigma N\rfloor}}\left\{(-1)^k\alpha_{\lfloor\sigma N\rfloor}+2\sum_{p=1}^{\lfloor\sigma N\rfloor-1}\alpha_p\sin\left[\pi\frac{(k+1/2)p}{\lfloor\sigma N\rfloor}\right]\right\}. \qquad (26)$$

This convolution (Eq. (21)) can be converted into the “all-DCT” form containing only – DCT (Eq. (22)) and IDCT (Eq. (25)):

$$b_k = \sqrt{\frac{\lfloor\sigma N\rfloor}{2}}\left\{\mathrm{IDCT}\left[\xi_p^C\eta_p^{CI}\right]+(-1)^k\,\mathrm{IDCT}\left[\left\{\xi_p^S\eta_p^{CI}\right\}_{\lfloor\sigma N\rfloor-p}\right]\right\}, \qquad (27)$$

where

$$\xi_p^C = \mathbf{DCT}(\alpha_r), \tag{28}$$

$$\xi_p^S = \left\{ \mathbf{DCT}\left[(-1)^r \alpha_r\right] \right\}_{\lfloor \sigma N \rfloor - p} \tag{29}$$

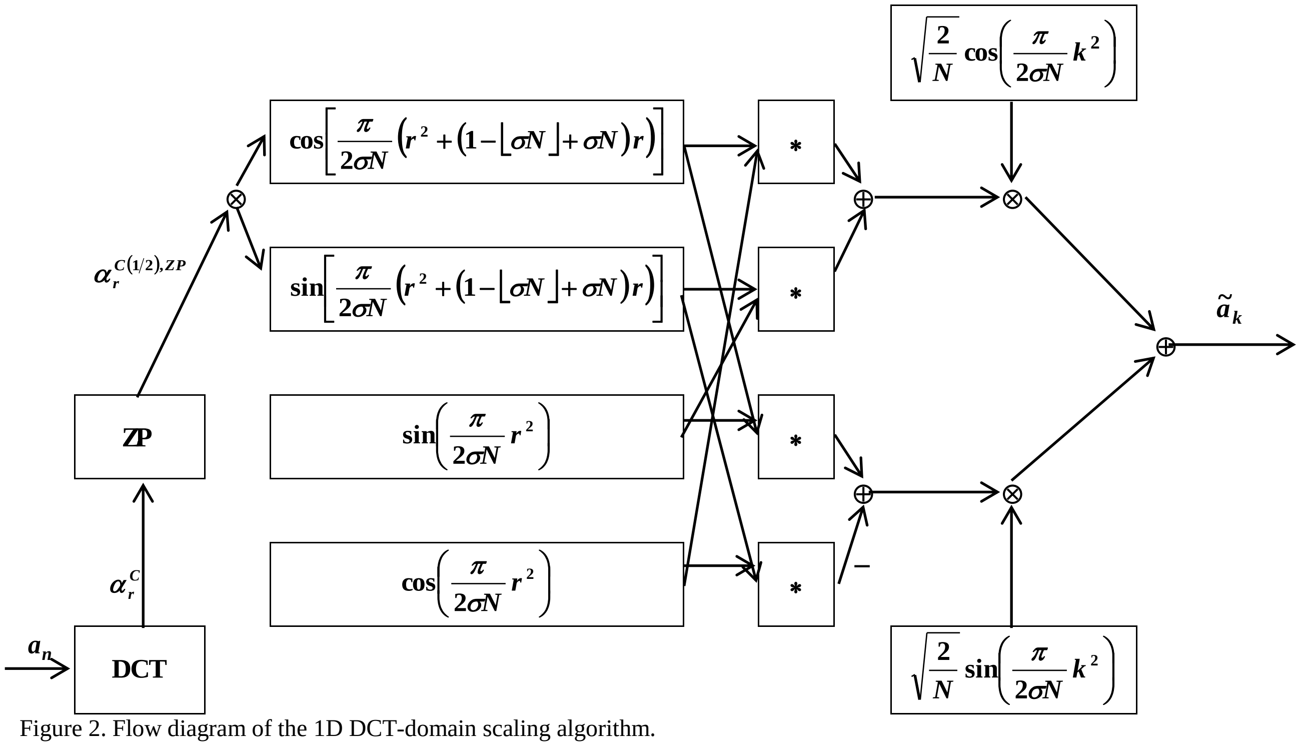
and

$$\eta_p^{CI} = -\frac{\mathbf{1}}{\sqrt{\mathbf{2}\lfloor \sigma N \rfloor}} h_0 + \mathbf{cos}\left(\frac{\pi}{\mathbf{2}\lfloor \sigma N \rfloor} p\right)\mathbf{DCT}(h_r) + \mathbf{sin}\left(\frac{\pi}{\mathbf{2}\lfloor \sigma N \rfloor} p\right)\left\{\mathbf{DCT}\left[(-1)^r h_r\right]\right\}_{\lfloor \sigma N \rfloor - p}. \tag{30}$$

Equation (20) describes the proposed DCT based signal scaling algorithm. The flow diagram of this algorithm is shown in Figure 2. The DCT unit implements Discrete Cosine Transform (Eq. (12)) and the ZP unit implements zero-padding or truncation (Eq. (15)). The $*$ unit implements DCT-domain convolution (Eqs. (27)-(30)). Both r and k are vectors of integers $(0,\ldots,\lfloor \sigma N \rfloor - 1)$; a_k denotes the input signal and $\tilde{a}_k$ denotes the scaled signal.

From Figure 2 and Eqs. (27)-(30) it is evident that the DCT-based scaling algorithm amounts to 16 DCTs of length $\lfloor \sigma N \rfloor$, one DCT of length N, $9\lfloor \sigma N \rfloor$ real additions and $14\lfloor \sigma N \rfloor + 2N$ real multiplications. Recalling that the DCT of length N can be implemented with $2\frac{1}{3} N \log_2 N - 2\frac{2}{9} N$ flops[9], the computational complexity of the DCT-based scaling algorithm is equal to $32\frac{2}{3}\log_2 \lfloor \sigma N \rfloor + 2\frac{1}{3}\frac{N}{\lfloor \sigma N \rfloor}\log_2 N$ flops per output sample.

The 2D scaling is defined as separable and therefore it is implemented through two consecutive applications of 1D DCT-based scaling. At the first step the 1D vertical scaling is applied to the columns of the input image $a_{m,n}$, resulting in the intermediate column-scaled image $a_{k,n}^{(col)}$. At the second step the 1D horizontal scaling is applied to the rows of the intermediate image $a_{k,n}^{(col)}$, resulting in the 2D-scaled image $\tilde{a}_{k,l}$.



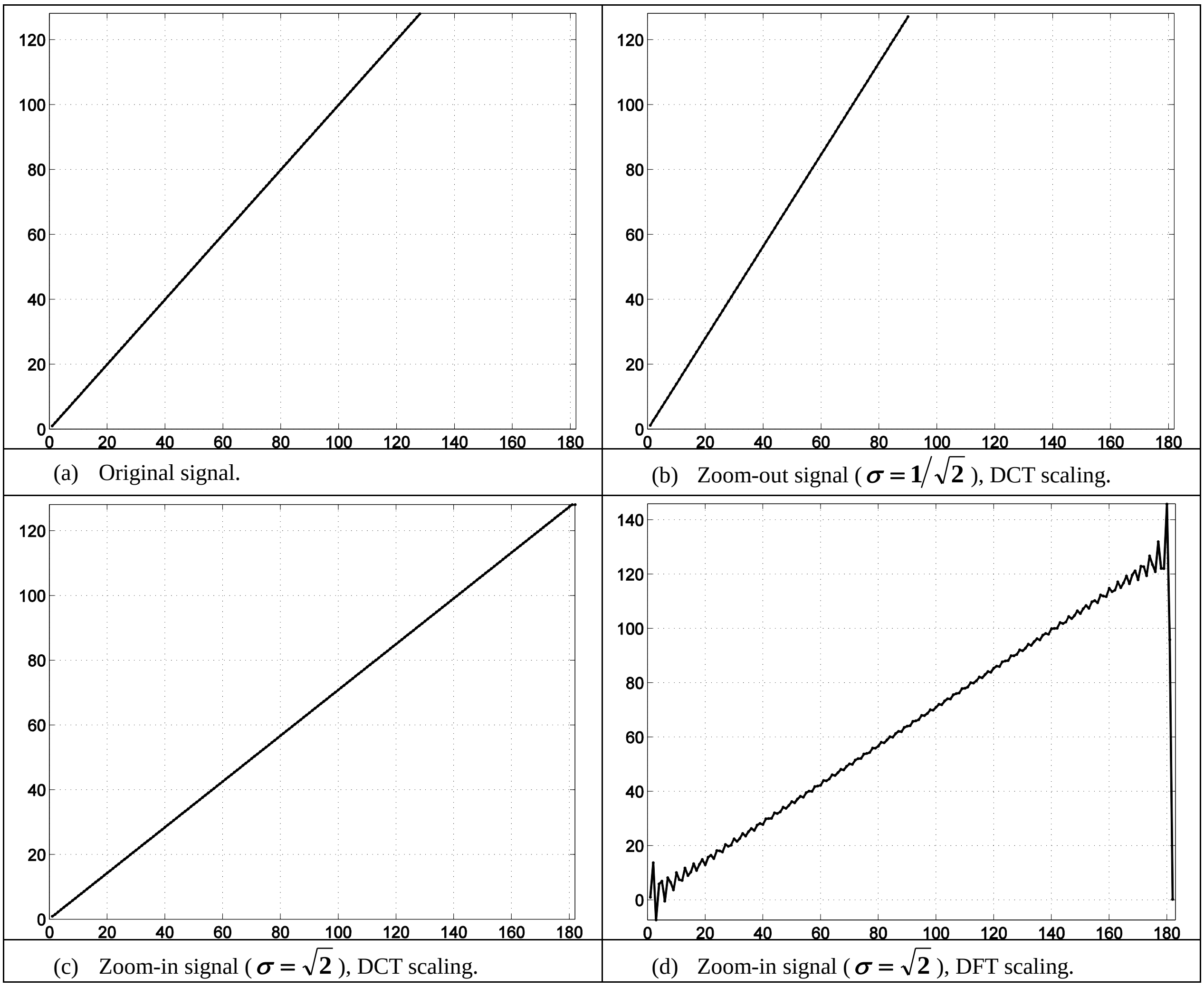
$\sqrt{\frac{2}{N}}\cos\left(\frac{\pi}{2\sigma N}k^2\right)$
$\cos\left[\frac{\pi}{2\sigma N}\left(r^2 + \left(1 - \lfloor \sigma N \rfloor + \sigma N\right) r\right)\right]$
$*$
$\alpha_r^{C(1/2),ZP}$
$\sin\left[\frac{\pi}{2\sigma N}\left(r^2 + \left(1 - \lfloor \sigma N \rfloor + \sigma N\right) r\right)\right]$
$*$
$\tilde{a}_k$
ZP
$\sin\left(\frac{\pi}{2\sigma N}r^2\right)$
$*$
α_r^C
$\cos\left(\frac{\pi}{2\sigma N}r^2\right)$
$*$
$-$
a_n
DCT
$\sqrt{\frac{2}{N}}\sin\left(\frac{\pi}{2\sigma N}k^2\right)$

Figure 2. Flow diagram of the 1D DCT-domain scaling algorithm.

3. EXPERIMENTAL VERIFICATION

For experimental verification of the proposed algorithm we checked boundary effects of scaling and compared the accuracy of the suggested DCT algorithm with that of conventional scaling methods using bilinear/ bicubic[10]/ and cubic spline[1-3] interpolation and DFT-domain discrete sinc-interpolation methods[4-6].

As a 1D test signal, a "ramp" signal of length $N = 128$ was used to check the boundary effects of signal scaling. Shown in Figure 3 are results of scaling of this test signal by factor $\sigma = 1/\sqrt{2}$ for zooming-out and by factor $\sigma = \sqrt{2}$ for zooming-in. One can clearly see that the DFT-domain scaling algorithm[4-6] suffers from heavy boundary artifacts, while the described DCT-domain scaling algorithm doesn't virtually introduce any artifacts at the borders of the signal.



(a) Original signal.

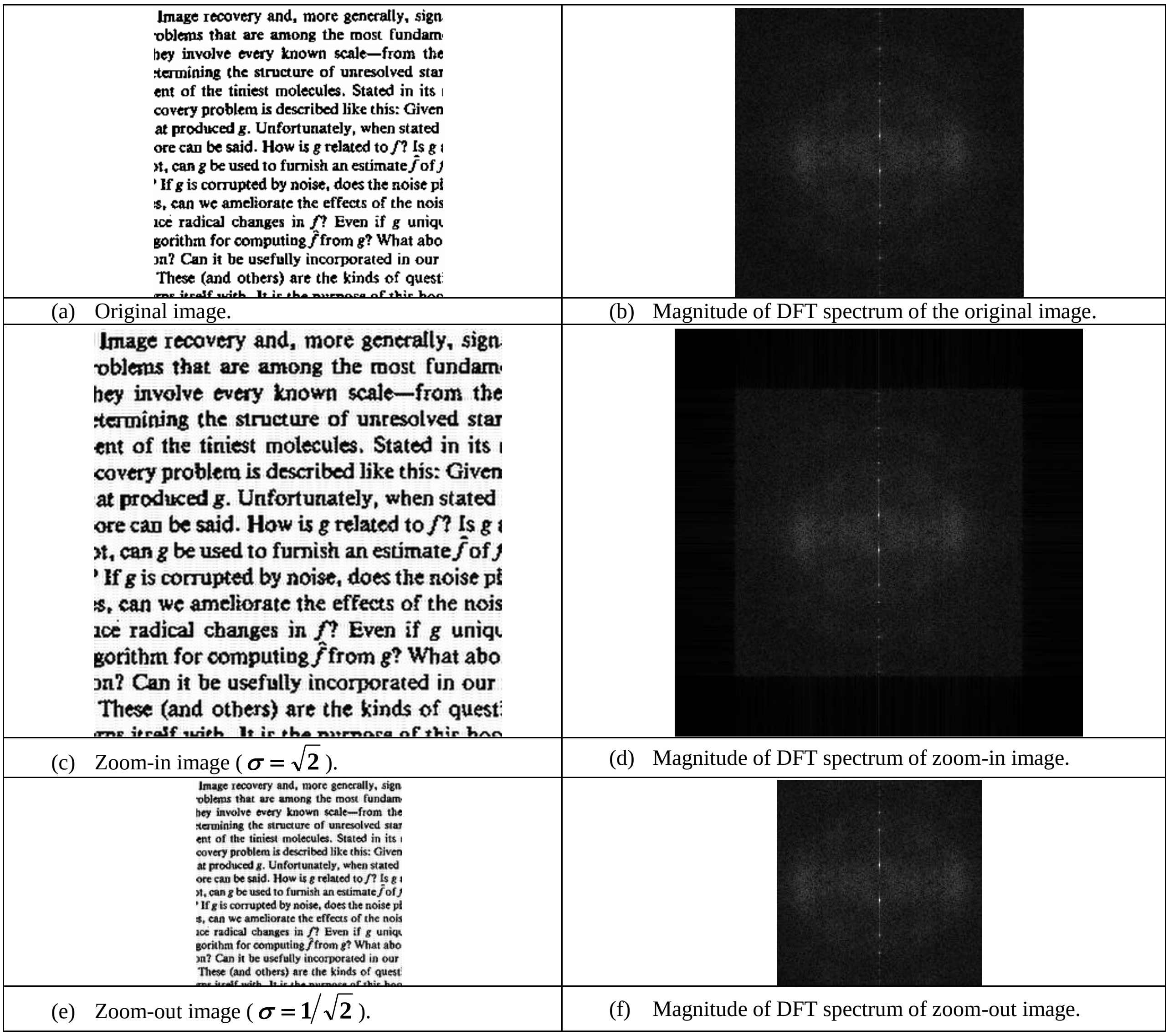
(b) Zoom-out signal ($\sigma = 1/\sqrt{2}$), DCT scaling.

(c) Zoom-in signal ($\sigma = \sqrt{2}$), DCT scaling.

(d) Zoom-in signal ($\sigma = \sqrt{2}$), DFT scaling.

Figure 3. Scaling of the 1D signal by the DCT-domain scaling algorithm. (a)-The original signal. (b)-The zoom-out signal ($\sigma = 1/\sqrt{2}$), DCT scaling. (c)-The zoom-in signal ($\sigma = \sqrt{2}$), DCT scaling. (d)-The zoom-in signal ($\sigma = \sqrt{2}$), DFT scaling.

As test images for checking 2D scaling, an image “text” and a pseudo-random image with uniform spectrum of size $\mathbf{256\times256}$ were used. The results of zoom-in by factor $\sigma=\sqrt{2}$ and zoom-out by factor $\sigma=1/\sqrt{2}$ of the “text” image and magnitudes of corresponding DFT spectra are shown in Figure 4.



(a) Original image.

(b) Magnitude of DFT spectrum of the original image.

(c) Zoom-in image ($\sigma=\sqrt{2}$).

(d) Magnitude of DFT spectrum of zoom-in image.

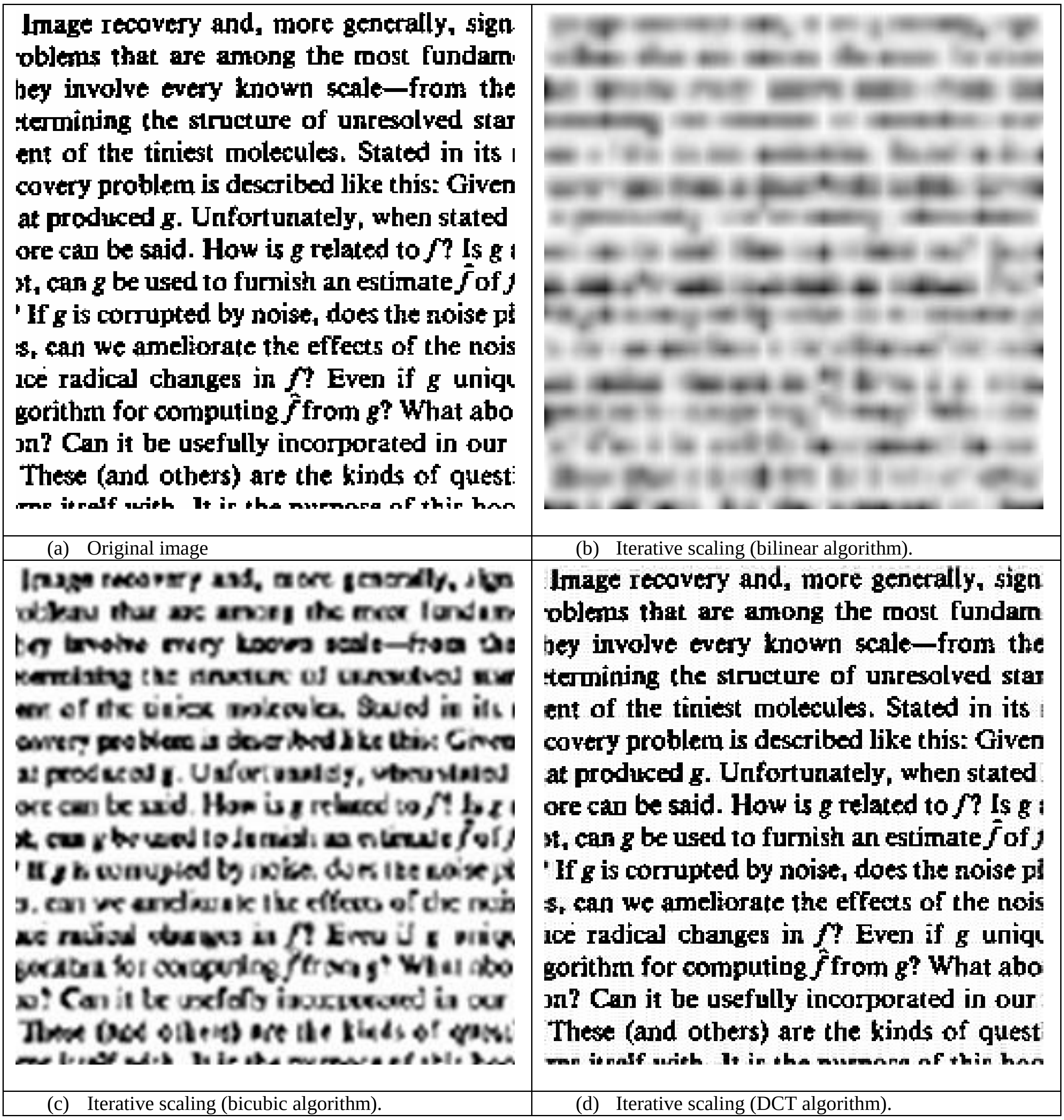
(e) Zoom-out image ($\sigma=1/\sqrt{2}$).

(f) Magnitude of DFT spectrum of zoom-out image.

Figure 4. (a)-The original test image. (b)-Magnitude of DFT spectrum of the original image shown (centered around DC component; (the top, bottom, left and right borders correspond to the highest signal frequency $\mathbf{FLOOR}(N/2)$. (c)-The zoom-in image (DCT method, $\sigma=\sqrt{2}$). (d)-Magnitude of DFT spectrum of the zoom-in image. (e)-The zoom-out image (DCT method, $\sigma=1/\sqrt{2}$). (f)-Magnitude of DFT spectrum of the zoom-out image. (The DFT spectra were normalized and raised to power 0.5 for display purposes.)

In order to evaluate the accuracy of image resampling by the suggested DCT-domain scaling algorithm in comparison with conventional bilinear/ bicubic[10]/ cubic spline[1-3]-based scaling algorithms, we employed iterative zooming-in & zooming-out sequence of operations (zooming-in of the original image by factor σ followed by zooming-out by the reciprocal factor $1/\sigma$). For the ideal resampling procedure the resulting “zoom-back” image has to be identical to the original image.

The results of 75-step iterative zoom-back of “text” image by factor $\sigma = \sqrt{2}$ by bilinear/ bicubic/ DCT-based scaling algorithms are shown in Figure 5.



(a) Original image

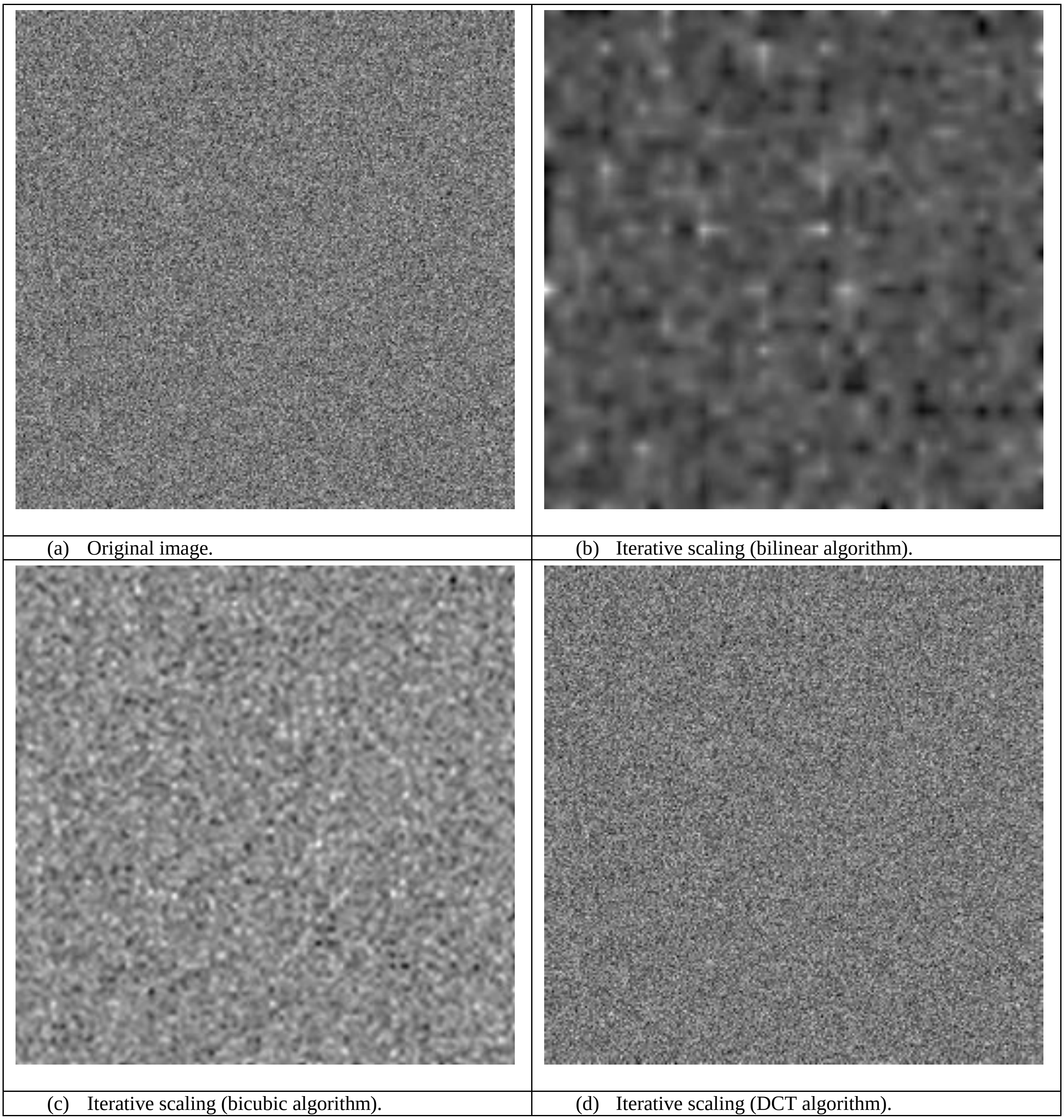
(b) Iterative scaling (bilinear algorithm).

(c) Iterative scaling (bicubic algorithm).

(d) Iterative scaling (DCT algorithm).

Figure 5. Iterative zoom-in&zoom-out scaling of the “text” image (a) by the bilinear algorithm (b), by the bicubic algorithm (c) and by the DCT-domain scaling algorithm (d) after 75 iterations.

The results of 75-step iterative zoom-back of “random” image by factor $\sigma = \sqrt{2}$ by bilinear/ bicubic/ DCT-based scaling algorithms are shown in Figure 6.



(a) Original image.

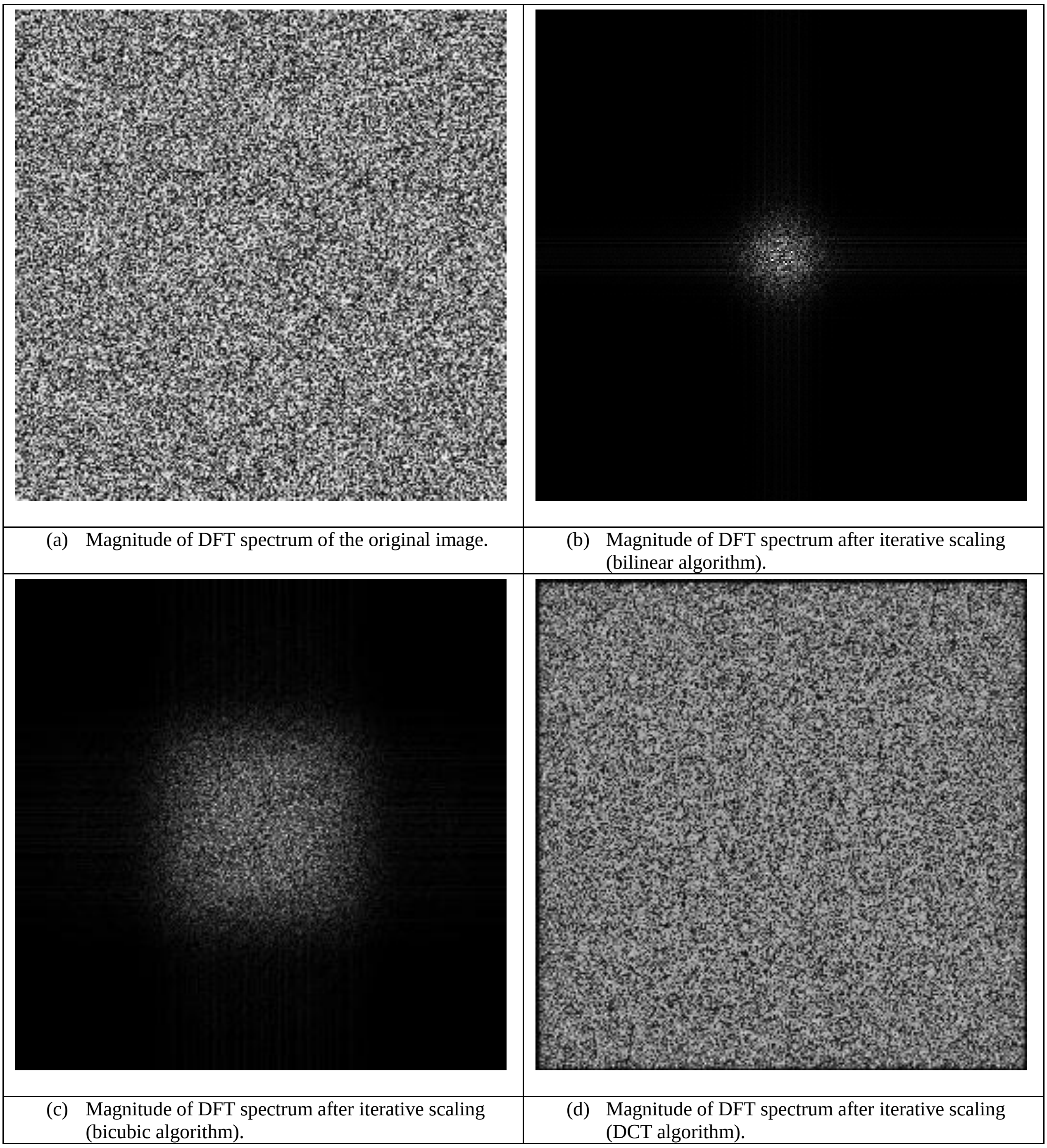
(b) Iterative scaling (bilinear algorithm).

(c) Iterative scaling (bicubic algorithm).

(d) Iterative scaling (DCT algorithm).

Figure 6. Iterative zoom-in&zoom-out scaling of the “random” image (a) by the bilinear algorithm (b), by the bicubic algorithm (c) and by the DCT-domain scaling algorithm (d) after 75 iterations.

Evolution of image spectra in such an iterative zooming-in&zooming-out test one can evaluate from comparison of spectra of “zoom-back”-iterated images. The magnitudes of DFT-spectra after 75-step iterative zoom-back of “random” image by factor $\sigma = \sqrt{2}$ by bilinear/ bicubic/ DCT-based scaling algorithms are shown in Figure 7.



(a) Magnitude of DFT spectrum of the original image.

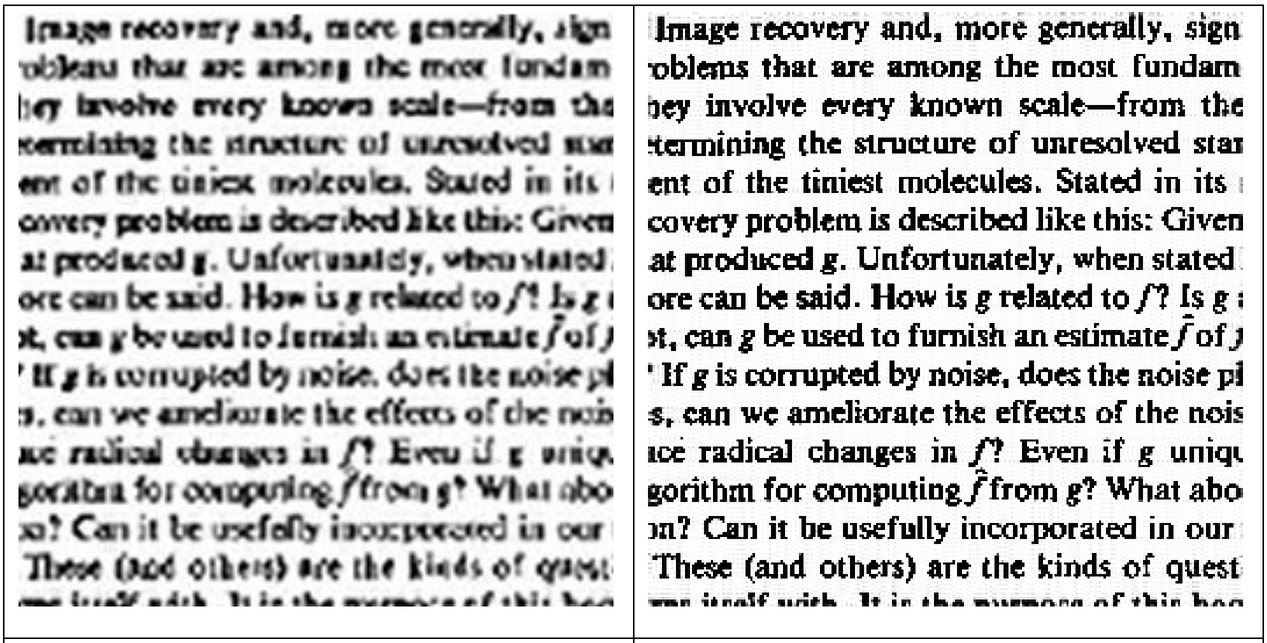
(b) Magnitude of DFT spectrum after iterative scaling (bilinear algorithm).

(c) Magnitude of DFT spectrum after iterative scaling (bicubic algorithm).

(d) Magnitude of DFT spectrum after iterative scaling (DCT algorithm).

Figure 7. Magnitude of DFT spectrum of iterative zoom-in&zoom-out scaling of the random image (a) by the bilinear algorithm (b), by the bicubic algorithm (c) and by the DCT-domain scaling algorithm (d) after 75 iterations.

Cubic spline is known to perform better than simple bicubic interpolation. However it is also not as perfect as discrete sinc-interpolation. The results of 200-step iterative zoom-back of “text” image by factor $\sigma = \sqrt{2}$ by cubic spline and by DCT-based scaling algorithms are shown in Figure 8.



(a) Iterative scaling (cubic spline algorithm). (b) Iterative scaling (DCT algorithm).

Figure 8. Iterative zoom-in&zoom-out scaling of the “text” image by the cubic spline scaling algorithm (a) and by the DCT domain scaling algorithm (b) after 200 iterations.

Comparing the presented results one can see that the bilinear/ bicubic/ cubic spline scaling methods tend to blur scaled images and the DFT-based scaling methods suffer from boundary artifacts. The suggested DCT-based scaling method produces perfectly sharp images and is virtually free of boundary effects, demonstrating virtually perfect accuracy of resampling.

4. CONCLUSION

A novel DCT-domain algorithm for signal and image scaling by arbitrary factors is presented. It was demonstrated that the proposed algorithm is virtually free of boundary effects and ensures virtually perfect reconstruction accuracy. Thanks to the availability of fast FFT-type algorithms for computing DCT and IDCT transforms, the algorithm has reasonably low computational complexity and represents a valuable alternative to known scaling algorithms in image and video processing applications.

REFERENCES

[1] Moler, C. B., [*Numerical Computing with MATLAB*], Society for Industrial and Applied Mathematics (2004).

[2] Wolberg, G., [*Digital Image Warping*], IEEE Computer Society Press (1990).

[3] de Boor, C., [*A Practical Guide to Splines*], Springer-Verlag (1978).

[4] Yaroslavsky, L., “Fast discrete sinc-interpolation: a gold standard for image resampling,” in [*Advances in Signal Transforms: Theory and Applications*], Astola, J. and Yaroslavsky, L., eds., *EURASIP Book Series on Signal Processing and Communications* **7**, ch. 8, 337–405, Hindawi (2007).

[5] 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**, ch. 3, 93–141, Hindawi (2007).

[6] Rabiner, L. R., Schafer, R. W., and Rader, C. M., “The chirp *z*-transform algorithm and its application,” *Bell Syst. Tech. J.* **48**, 1249–1292 (May-June 1969).

[7] Yaroslavsky, L., “Boundary effect free and adaptive discrete signal sinc-interpolation algorithms for signal and image resampling,” *Appl. Optics* **42**, 4166–4175 (July 10, 2003).

[8] Agbinya, J. I., “Interpolation using the discrete cosine transform,” *Electron. Lett.* **28**, 1927–1928 (Sept. 24, 1992).

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

[10] Keys, R. G., “Cubic convolution interpolation for digital image processing,” *IEEE Trans. Acoust., Speech, Signal Process.* **29**, 1153–1160 (Dec. 1981).