# Boundary effect free and adaptive discrete signal sinc-interpolation algorithms for signal and image resampling

**L. Yaroslavsky**

Dept. of Interdisciplinary Studies, Faculty of Engineering,

Tel Aviv University, 69978 Tel Aviv, Israel

yaro@eng.tau.ac.il

ABSTRACT

The problem of digital signal and image resampling with discrete sinc-interpolation is addressed. Discrete sinc-interpolation is theoretically the best one among the digital convolution based signal resampling methods since it does not distort the signal as defined by its samples and is completely reversible. However, sinc-interpolation is frequently not considered in applications since it suffers from boundary effects, tends to produce signal oscillations at the image edges and has relatively high computational complexity when irregular signal resampling is required. In the paper, a solution that allows to eliminate these limitations of the discrete sinc-interpolation is suggested. Two flexible and computationally efficient algorithms for boundary effects free and adaptive discrete sinc-interpolation are presented: frame-wise (global) sinc-interpolation in DCT domain and local adaptive sinc-interpolation in DCT domain of a sliding window. The latter offers options not available with other interpolation methods: interpolation with simultaneous signal restoration/enhancement and adaptive interpolation with “super resolution”.

*OCIS codes* ***100.0100, 100.200, 110.6980***

## 1. INTRODUCTION

Signal and image resampling is required in many signal and image processing applications. It is a key issue in audio signal spectral analysis and fractional delay, signal and image differentiating and integrating, image geometrical transformations and rescaling, target location and tracking with sub-pixel accuracy, Radon Transform and tomographic reconstruction, 3-D image volume rendering and volumetric imaging. Signal/image resampling assumes one or another method of interpolation between available signal/image samples. By virtue of the sampling theorem, sinc-interpolation is the best interpolation of continuous signals. Given samples $\{a_k\}$ of a continuous signal $a(x)$ sinc-interpolated approximation $\tilde{\tilde{a}}(x)$ to this signal is defined as

$$\tilde{\tilde{a}}(x) = \sum_{-\infty}^{\infty} a_k \,\mathbf{sinc}\left[\pi\left(x - k\Delta x\right)/\Delta x\right], \tag{1}$$

where $\mathbf{sinc}\, x = (\sin x)/x$ and $\Delta x$ is the signal discretization interval. Provided unlimited number of signal samples obtained as

$$a_k = \frac{1}{\Delta x}\int_{-\infty}^{\infty} a(x)\,\mathbf{sinc}\left[\pi\left(x - k\Delta x\right)\right] dx \ , \tag{2}$$

sinc-interpolation restores the continuous signal with the least mean squared error. For band-limited signals with spectrum bandwidth $\mathbf{1/2\Delta x}$, sinc-interpolation provides exact restoration of signals from their samples. In other words, signal sampling according to Eq. 2 is in this case completely reversible. However exact sinc-interpolation can not be implemented in reality since it requires unlimited number of signal samples and interpolation function with infinite support.

In digital signal processing with a finite number of available signal samples a discrete analog of the continuous sinc-interpolation is discrete sinc-interpolation

$$\tilde{\tilde{a}}(x) = \sum_{k=0}^{N-1} a_k \,\mathbf{sincd}(N\,; N; x/\Delta x - k), \tag{3}$$

where

$$\mathbf{sincd}(N; N; x) = \frac{\sin(\pi x)}{N \sin(\pi x/N)} \tag{4}$$

is a discrete sinc-function. It approximates the continuous sinc-function $\mathbf{sinc(x)} = \sin x / x$ for $x/\Delta x << N$. As it will be shown in Sect. 2, given finite number of signal samples, discrete sinc-interpolation is the only convolution based fully reversible discrete signal resampling method. This feature defines the attractiveness of the discrete sinc-interpolation for signal/image resampling.

Conventionally, discrete sinc-interpolation is implemented by means of signal spectrum zero-padding algorithm[1-3]. A more efficient and flexible discrete sinc-interpolation algorithm was described in[4]. However despite the attractiveness of discrete sinc-interpolation it is frequently not regarded appropriate in applications. First, discrete sinc-interpolation tends to produce heavy artifacts in form of oscillations (ripples) at signal borders. This property may be especially restrictive in image processing since image dimensions are usually relatively small numbers (256-1024), and noticeable ripples from image left, right, upper and bottom borders may occupy substantial part of the image. Second, signal oscillations caused by sinc-interpolation may also be observed in the vicinity of signal/image sharp edges. These oscillations are absolutely normal as soon as reversible discrete sinc-interpolation is required. However they are frequently considered undesirable artifacts that worsen visual image quality. In addition, the above available discrete sinc-interpolation methods are not well suited to irregular (not equidistant) resampling. Due to these reasons, discrete sinc-interpolation is quite rarely practiced in digital signal and image processing.

In the paper, we introduce two new discrete sinc-interpolation algorithms that eliminate above-mentioned drawbacks of the discrete sinc-interpolation and offer additional useful capabilities not available with other methods. In Sect. 3, a computationally efficient and flexible algorithm of the discrete sinc-interpolation is described that is free of oscillation phenomena at signal borders. In distinction to the known discrete sinc-interpolation algorithms, the algorithm computes and modifies Discrete Cosine Transform (DCT) rather than Discrete Fourier Transform (DFT) signal spectra. It is referred to as global DCT domain discrete sinc-interpolation algorithm. In Sect. 4, a sliding window signal resampling algorithm is introduced that also works in the domain of DCT. Being a good approximation to the ideal global discrete sinc-interpolation, the algorithm is, in addition, capable of simultaneous signal denoising and of local adaptation of the convolution kernel. The latter feature allows, in particular, to eliminate, whenever it is required by application, oscillations at signal/image sharp edges and at the same time to avoid smoothing the edges. In this way, an increased signal resolution with respect to that defined by signal's initial sampling rate can be obtained.

## 2. DISCRETE SINC-INTERPOLATION: ERROR FREE INTERPOLATION OF SAMPLED DATA

Let a discrete signal of $\boldsymbol{N}$ samples $\{\boldsymbol{a}_k\}$ (Fig. 1, a) be interpolated to a signal with ($\boldsymbol{L}$-1) interpolated intermediate samples per each initial one. For convolution based interpolation, the interpolation process is a digital convolution,

$$\tilde{\tilde{\boldsymbol{a}}}_k = \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{L-1} \boldsymbol{a}_{k_1}\delta\left(\boldsymbol{k}_2\right)\boldsymbol{h}_{\mathbf{int}}\left(\boldsymbol{k}-\boldsymbol{k}_1\boldsymbol{L}+\boldsymbol{k}_2\right),\ \boldsymbol{k}=\mathbf{0},\mathbf{1},...,\boldsymbol{LN}-\mathbf{1} \tag{5}$$

with interpolation kernel $\left\{\boldsymbol{h}_{\mathbf{int}}\left(\boldsymbol{k}\right)\right\}$, of signal $\left\{\tilde{\boldsymbol{a}}_k = \boldsymbol{a}_{k_1}\delta\left(\boldsymbol{k}_2\right)\right\}$, $\boldsymbol{k}=\boldsymbol{k}_1\boldsymbol{L}+\boldsymbol{k}_2$; $\boldsymbol{k}_1=\mathbf{0},...,\boldsymbol{N}-\mathbf{1};$ $\boldsymbol{k}_2=\mathbf{0},...,\boldsymbol{L}-\mathbf{1}$; $\delta\left(\boldsymbol{x}\right)=\mathbf{0}^{\boldsymbol{x}}$, obtained from the initial signal $\{\boldsymbol{a}_k\}$ by placing ($\boldsymbol{L}$-**1**) zeros between its samples as it is illustrated in Fig. 1, b). Compute Discrete Fourier Transform (DFT) of signal $\{\tilde{\boldsymbol{a}}_k\}$:

$$\tilde{\alpha}_r = \frac{\mathbf{1}}{\sqrt{\boldsymbol{LN}}}\sum_{k=0}^{LN-1}\tilde{\boldsymbol{a}}_k\,\mathbf{exp}\left(\boldsymbol{i}\mathbf{2}\pi\,\frac{\boldsymbol{kr}}{\boldsymbol{LN}}\right) = \frac{\mathbf{1}}{\sqrt{\boldsymbol{LN}}}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{L-1}\boldsymbol{a}_{k_1}\delta\left(\boldsymbol{k}_2\right)\mathbf{exp}\left[\boldsymbol{i}\mathbf{2}\pi\,\frac{\left(\boldsymbol{k}_1\boldsymbol{L}+\boldsymbol{k}_2\right)}{\boldsymbol{LN}}\boldsymbol{r}\right] =$$

$$\frac{\mathbf{1}}{\sqrt{\boldsymbol{LN}}}\sum_{k_1=0}^{N-1}\boldsymbol{a}_{k_1}\,\mathbf{exp}\left(\boldsymbol{i}\mathbf{2}\pi\,\frac{\boldsymbol{k}_1}{\boldsymbol{N}}\boldsymbol{r}\right) = \frac{\mathbf{1}}{\sqrt{\boldsymbol{L}}}\alpha_{(r)\mathbf{mod}\,N}\,, \tag{6}$$

where $\{\alpha_r\}$ is DFT of signal $\{\boldsymbol{a}_k\}$. Eq. (6) shows that sampling discrete signal by placing zeros between its samples results in periodical replication of its DFT spectrum with the number of replicas equal to the number of zeros plus one as it is illustrated in Figs. 1, c) and d), respectively**.** If interpolation (Eq.5) is computed as a cyclic (periodical) convolution, it will correspond, in DFT domain, to multiplying spectrum $\{\tilde{\alpha}_r\}$ with DFT of the interpolation kernel:

$$\boldsymbol{DFT}\left\{\tilde{\tilde{a}}_k\right\}=\tilde{\tilde{\alpha}}_r=\left(\frac{1}{\sqrt{L}}\alpha_{(r)\mathrm{mod}\,N}\right)\cdot \boldsymbol{DFT}\left\{h_{\mathrm{int}}(k)\right\}. \tag{7}$$

One can see from this equation that the only way to secure reversibility of the interpolation and to avoid, in the interpolation, distorting initial signal spectrum and introducing into the interpolated signal aliasing spectral components is signal ideal low pass filtering, when $DFT\{h_{\mathrm{int}}(k)\}$ is a rectangle function of $N$ samples. Such a filtering is graphically illustrated in Fig. 1 e). Since the interpolation kernel should be a real valued function, complex conjugacy symmetry property $\tilde{\tilde{\alpha}}_r=\tilde{\tilde{\alpha}}^*_{LN-r}$ of its DFT spectrum should be observed in zeroing aliasing spectral components. One can meet this requirement only for odd $N$, in which case $DFT\{h_{\mathrm{int}}(k)\}$ will be:

$$DFT\left\{h_{\mathrm{int}}(k)\right\}=1-rect\frac{r-(N+1)/2}{LN-N-1}, \tag{8}$$

where $r=0,1,...,LN-1$; $rect(x)=\begin{cases}1, & 0<x<1\\ 0, & otherwise\end{cases}$. Interpolated signal is then

$$\tilde{\tilde{a}}_k=IDFT\left\{\left[1-rect\frac{r-(N+1)/2}{LN-N-1}\right]\frac{\alpha_{(r)\mathrm{mod}\,N}}{\sqrt{L}}\right\}=\frac{1}{\sqrt{L}}\sum_{n=0}^{N-1}a_{n_1}\mathrm{sincd}\left(\mathrm{N};\mathrm{N};\left(\mathrm{k}-\mathrm{n}_1 L\right)\right), \tag{9}$$

where $IDFT(.)$ is inverse Discrete Fourier Transform and

$$\mathrm{sincd}(K;N;x)=\frac{\sin(\pi Kx/N)}{N\sin(\pi x/N)} \tag{10}$$

is the discrete sinc-function. Since no signal spectrum components are distorted in the convolution, discrete sinc-interpolation of signals with odd number of samples described by Eq. 9 is completely reversible.

For even ***N***, interpolation described by Eq. (9) can't be implemented since term with index $(\boldsymbol{N}+\mathbf{1})/\mathbf{2}$ required by Eq. (9) does not exist. Two immediate options in this case are

$$\tilde{\tilde{a}}_k = IDFT\left\{\left[1 - rect\frac{r-N/2}{LN-N}\right]\frac{\alpha_{(r)\mathrm{mod}\,N}}{\sqrt{L}}\right\} = \frac{1}{\sqrt{L}}\sum_{n_1=0}^{N-1} a_{n_1}\mathbf{sincd}\left[\mathrm{N-1;N;}\left(\mathrm{k-n_1}L\right)\right]; \tag{11}$$

$$\tilde{\tilde{a}}_k = IDFT\left\{\left[1 - rect\frac{r-N/2-1}{LN-N-2}\right]\frac{\alpha_{(r)\mathrm{mod}\,N}}{\sqrt{L}}\right\} = \frac{1}{\sqrt{L}}\sum_{n_1=0}^{N-1} a_{n_1}\mathbf{sincd}\left\{\mathrm{N+1;N;}\left(\mathrm{k-n_1}L\right)\right\}. \tag{12}$$

Both interpolation functions $\mathbf{sincd}(\mathrm{N-1;N;}k)$ and $\mathbf{sincd}(\mathrm{N+1;N;}k)$ converge to zero relatively slowly and therefore tend to produce severe boundary effects. A practical compromise in the case of even ***N*** is to halve the (***N***/2)-th spectral coefficient which corresponds to the following interpolation formula:

$$\tilde{\tilde{a}}_k = \frac{1}{\sqrt{L}}\sum_{n_1=0}^{N-1} a_{n_1}\mathbf{sincd}\left[\pm 1\mathrm{;N;}\left(\mathrm{k-n_1}L\right)\right], \tag{13}$$

where

$$\mathbf{sincd}(\pm 1\mathrm{;N;}k) = \left[\mathbf{sincd}(N+1\mathrm{;N;}k) + \mathbf{sincd}(N-1\mathrm{;N;}k)\right]/2\,. \tag{14}$$

It follows from Eqs. 11-13 that, for even ***N,*** one can't avoid distorting the signal. Its highest frequency coefficient with index ***N***/2 is either zeroed (Eq. 11) or repeated twice (Eq. 12) or halved (Eq. 13) in the interpolation process.

Direct convolution of signal samples with discrete sinc-interpolation function according to right hand parts of Eqs. 9, 11, 12 and Eq. 13 requires $\boldsymbol{O}(\boldsymbol{N})$ operations per output signal sample or $\boldsymbol{O}(\boldsymbol{N}^2\boldsymbol{L})$ operations for the entire sinc-interpolated output signal of $\boldsymbol{NL}$ samples. The computational complexity of the discrete sinc-interpolation can be substantially reduced if signal convolution is computed in the domain of DFT with the use of Fast Fourier Transform algorithms. Two methods of such an implementation are available. The first method, spectrum zero padding one[1-3], literally reproduces manipulations with signal DFT spectrum described by middle parts of Eqs. 9, 11, and 12. The algorithm computes signal's DFT spectrum, pads it with $\boldsymbol{N}(\boldsymbol{L}$-1) zeros and then computes inverse DFT of the obtained *NL* spectrum coefficients. Thanks to the use of Fast Fourier Transform for computing DFT, the computational complexity of the algorithm is $\boldsymbol{O}(\boldsymbol{N}\,\boldsymbol{log}\,\boldsymbol{N}+\boldsymbol{NL}\,\boldsymbol{log}\,\boldsymbol{NL})$ operations for the entire output signal of $\boldsymbol{NL}$ samples or $\boldsymbol{O}[(\mathbf{1}+\mathbf{1}/\boldsymbol{L})\mathbf{log}\,\boldsymbol{NL}]$ operations per output signal sample. The second algorithm described in[4] allows, for a signal $\boldsymbol{N}$ samples, to generate its discrete sinc interpolated copy of N samples shifted with respect to initial samples by an arbitrary interval. Such a shifted signal $\{\boldsymbol{a}_k^{(p)}\}$ is obtained as

$$\{\boldsymbol{a}_k^p\}=\boldsymbol{IDFT}\{\alpha_r\varphi_r(\boldsymbol{p})\}, \tag{15}$$

where $\{\varphi_r(\boldsymbol{p})\}$ is DFT of the discrete sinc-function:

$$\varphi_r(\boldsymbol{p})=\frac{\mathbf{1}}{\sqrt{\boldsymbol{N}}}\sum_{\boldsymbol{k}=\mathbf{0}}^{\boldsymbol{N}-\mathbf{1}}\mathbf{sincd}(\boldsymbol{K};\boldsymbol{N};\boldsymbol{k}-\boldsymbol{p})\mathbf{exp}\left(\boldsymbol{i}\mathbf{2}\pi\frac{\boldsymbol{kr}}{\boldsymbol{N}}\right) \tag{16}$$

and $\boldsymbol{p}$ is a shift parameter (measured in units of the signal discretization interval). One can show that

$$\varphi_r(p) = \begin{cases} \frac{1}{\sqrt{N}} \exp(i2\pi pr / N); & r = 0,1,...,(N-1)/2-1 \\ \varphi^*_{N-r}; & r = (N-1)/2+1,...,N-1 \end{cases} \tag{17}$$

for odd $N$ ($K$=$N$) and

$$\varphi_r(p) = \begin{cases} \frac{1}{\sqrt{N}} \exp(i2\pi pr / N); & r = 0,1,...,N/2-1 \\ \frac{1}{\sqrt{N}} \cos(2\pi pr / N); & r = N/2 \\ \varphi^*_{N-r}; & r = N/2+1,...,N-1 \end{cases} \tag{18}$$

for even $N$ ( $K = \pm 1$ ).

As it follows from Eq. 15 the algorithm has computational complexity of $O(2\, log\, N)$ operations per output signal sample when one shifted signal copy is required or $O((1+1/L) log\, N)$ per sample operations for obtaining $L$ differently shifted copies. The algorithm is well suited for signal arbitrary translation that is required in many signal/image processing applications, such as, for instance, signal fractional delay and image rotation[5].

## 3. GLOBAL DISCRETE SINC-INTERPOLATION IN DCT DOMAIN

Discrete sinc-interpolation described in Sect. 2 suffers from boundary effects caused, in particular, by its implementation as a cyclic convolution. The simplest and one of the most efficient ways to minimize boundary effects in digital filtering is signal extension by its “mirror reflection” from its boundaries. Such an extension completely eliminates signal discontinuities at the boundaries. For such signals, one still can retain advantages of computing convolution with the use of FFT if Shifted DFT ( $SDFT_{u,v}$ )[6,7]

$$\alpha_r^{(u,v)} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} a_k \exp\left[ i2\pi \frac{(k+u)}{N} r \right] \exp\left( i2\pi \frac{kv}{N} \right) \qquad (19)$$

with shift parameters $u = 1/2$ (half discretization interval in signal domain) and $v = 0$ (no shift in Fourier domain) is used instead of DFT. For signals extended to its double length by mirror reflection, $SDFT_{1/2,0}$ is reduced to the Discrete Cosine Transform[6,7]. Filtering such signals in SDFT domain of is also a cyclic convolution with a period of 2*N*, where *N* is the number of samples of the initial signal. Therefore, interpolation function for the extended signal should be obtained from that for the initial signal by padding it with zeros to the double length of 2*N* samples as it is illustrated in Fig. 2. Zero padding prevents convolution results from the influence of boundary effects of the cyclic convolution. Specifically, for generating a discrete sinc-interpolated copy of the initial signal of $N$ samples shifted by interval ***p***, interpolation function $\left\{h_{\mathrm{int}}^{(p)}(k)\right\}$ should be

$$h_{\mathrm{int}}^{(p)}(k) = \begin{cases} \mathrm{sincd}(K; N; k-p); & k = 0,1,...N-1 \\ 0; & k = N, N+1,...2N-1 \end{cases}, \qquad (20)$$

where $K = N$ for odd *N*, $K = \pm 1$ for even *N*.

With the use of $SDFT_{1/.2,0}$, the algorithm for generating *p*-shifted signal $\left\{a_k^{(p)}\right\}$ described by Eq. 15 is modified to

$$\left\{a_k^{p}\right\} = ISDFT_{1/2,0}\left\{\alpha_r^{DCT} \cdot \eta_r(p)\right\}, \qquad (21)$$

where $ISDFT_{1/2,0}$ is inverse $SDFT_{1/.2,0}$, $\left\{\alpha_r^{DCT}\right\}$ are DCT transform coefficients of signal $\{a_k\}$ and $\{\eta_r(p)\}$ are DFT coefficients of the interpolation function $\left\{h_{\mathrm{int}}^{(p)}(k)\right\}$:

$$\eta_r(p) = \eta_r^{re}(p) + i\eta_r^{im}(p) = \frac{1}{\sqrt{2N}} \sum_{k=0}^{2N-1} h_{\mathrm{int}}^{(p)}(k) \exp\left( i2\pi \frac{kr}{2N} \right). \qquad (22)$$

As DCT spectral coefficients $\left\{\alpha_r^{DCT}\right\}$ exhibit odd symmetry:

$$\alpha_r^{DCT} = \begin{cases} \alpha_r^{DCT} = DCT\{a_k\}, & r = 0,1,\ldots,N-1; \\ 0, & r = N \\ -\alpha_{2N-1-r}^{DCT}, & r = N+1, N+2,..,2N-1 \end{cases}, \qquad (23)$$

inverse SDFT $ISDFT_{1/2,0}$ for generating the interpolated signal according to Eq. 21 is reduced to inverse DCT and DST (Discrete Sine Transform):

$$a_k^{(p)} = \frac{1}{\sqrt{2N}} \sum_{r=0}^{2N-1} \alpha_r^{DCT} \eta_r(p) \exp\left( -i2\pi \frac{(k+1/2)r}{2N} \right) =$$

$$\frac{1}{\sqrt{2N}} \left\{ \alpha_0^{DCT} \eta_0 + \sum_{r=1}^{N-1} \alpha_r^{DCT} \left[ \eta_r \exp\left( -i\pi \frac{(k+1/2)}{N} r \right) + \eta_r^* \exp\left( i\pi \frac{(k+1/2)}{N} r \right) \right] \right\} =$$

$$\frac{1}{\sqrt{2N}} \left\{ \alpha_0^{DCT} \eta_0 + 2\sum_{r=1}^{N-1} \alpha_r^{DCT} \eta_r^{re} \cos\left( \pi \frac{(k+1/2)}{N} r \right) - 2\sum_{r=1}^{N-1} \alpha_r^{DCT} \eta_r^{im} \sin\left( \pi \frac{(k+1/2)}{N} r \right) \right\}, \qquad (24)$$

where $\eta_r^*$ is complex conjugate to $\eta_r$. For each $p$, coefficients $\eta_{2r}(p)$ with even indices can be found directly from the definition of $\{\eta_r\}$ (Eqs. 20, 22 and 17, 18):

$$\eta_{2r}(p) = \frac{1}{\sqrt{N}} \exp\left( i2\pi \frac{pr}{N} \right). \qquad (25)$$

Therefore one needs to additionally compute only terms $\{\eta_{2r+1}(p)\}$ with odd indices:

$$\eta_{2r+1}(p) = \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1} \mathrm{sincd}(K; N; k-p) \exp\left( i2\pi \frac{k(2r+1)}{2N} \right) \tag{26}$$

where, as in Eqs. 20, $K = N$ for odd $N$, $K = \pm 1$ for even $N$.

Flow diagram of this algorithm for generating $p$-shifted sinc-interpolated copy of signal is shown in Fig. 3. Fig. 4 illustrates the described discrete sinc interpolation in DCT domain applied for image zooming and compares it with that implemented in DFT domain. One can see that the algorithm does solve the problem of boudary effects charachteristic for discrete sinc-interpolation in DFT domain. It is, however, computationally efficient only if regular (equiduistant) resamplimg signal is required. In this case it allows to compute $p$-shifted sinc-interpolated copy of the signal of $N$ samples with the complexity of $O(N \log N)$ operations or $O(\log N)$ operations per signal sample. This complexity is, by the order of magnitude, the same as that of the above DFT domain sinc-interpolation algorithm. However fast algorithms for DCT, IDCT and IDST exist[8-15] that require even less computations than FFT and IFFT involved in the DFT domain algorithm.

## 4. SINC INTERPOLATION IN DCT DOMAIN IN SLIDING WINDOW

When, as it frequently happens in signal and image resampling tasks, required signal sample shifts are different for different samples, the above global discrete sinc-interpolation algorithm in DCT domain has no efficient computational implementation. However, in these cases it can be implemented in sliding window. In processing signal in sliding window, only those shifted and interpolated signal samples that correspond to the window central sample have to be computed in each window position from signal samples within the window. Interpolation function in this case is a windowed discrete sinc-function whose extent is equal to the window size rather to the signal size required for the perfect discrete sinc-interpolation. Fig. 5 illustrates frequency responses of the corresponding low pass filters for different window size. As one can see from the figure they deviate from a rectangle function, a frequency response of the ideal low pass filter (Eq. 8).

Such an implementation of the discrete sinc-interpolation can be regarded as a variety of direct convolution interpolation methods. In terms of the interpolation accuracy it has no special advantages before other direct convolution interpolation methods such as spline oriented ones[16,17]. However it offers features that are not available with other methods. These are: (i) signal resampling with arbitrary shifts and simultaneous signal restoration and enhancement and (ii) local adaptive interpolation with "super resolution".

For signal resampling with simultaneous restoration/enhancement, sliding window discrete sinc-interpolation should be combined with local adaptive filtering. Local adaptive filters that work in sliding window in transform domain such as that of DFT or DCT have shown their high potentials in signal and image restoration and enhancement[18,19]. The filters, in each position $\boldsymbol{k}$ of the window of $\boldsymbol{W}$ samples (usually an odd number), compute transform coefficients $\{\beta_r = \boldsymbol{T}\{\boldsymbol{b}_n\}\}$of the signal $\{\boldsymbol{b}_n\}$ in the window ($\boldsymbol{n},\boldsymbol{r} = \mathbf{1,2,...,}\boldsymbol{W}$) and nonlinearly modify them to obtain coefficients $\{\hat{\alpha}_r(\beta_r)\}$. These coefficients are used to generate an estimate $\hat{\boldsymbol{a}}_k$ of the window central pixel by inverse transform $\boldsymbol{T}_k^{-1}\{\ \}$ computed for the window central pixel as

$$\hat{\boldsymbol{a}}_k = \boldsymbol{T}_k^{-1}\{\hat{\alpha}_r(\beta_r)\}, \tag{27}$$

For instance, for filtering additive noise, "soft thresholding" (Empirical Wiener filter)

$$\hat{\alpha}_r(\beta_r) = \mathbf{max}\left(\mathbf{0}, \frac{|\beta_r|^2 - \boldsymbol{Thr}}{|\beta_r|^2}\right)\beta_r \tag{28}$$

or “hard thresholding”

$$\hat{\alpha}_r = \begin{cases} \beta_r, \ |\beta_r|^2 > \boldsymbol{Thr} \\ \mathbf{0}, \quad \boldsymbol{otherwise} \end{cases} \tag{29}$$

are used with where ***Thr*** is a certain threshold level associated with the noise variance. Such a filtering can be implemented in the domain of any transform though DCT has proved to be one of the most efficient. Therefore one can, in a straightforward way, combine sliding window DCT domain discrete sinc-interpolation signal resampling (Eq. 21) and filtering for signal restoration and enhancement (Eq. 27):

$$\left\{\boldsymbol{a}_k^p\right\} = \boldsymbol{ISDFT}_{1/2,0}\left\{\hat{\alpha}_r^{DCT}\left(\beta_r^{DCT}\right)\cdot\eta_r(\boldsymbol{p})\right\}. \tag{30}$$

Fig. 6 shows flow diagram of such a combined algorithm for signal restoration/enhancement and fractional ***p***-shift. It is assumed in the diagram that signal ***p***-shift is implemented according to the flow diagram of Fig. 3. Fig. 7 illustrates application of the combined filtering/interpolation for image irregular-to regular resampling combined with denoising. In this example, left image is distorted by known displacements of pixels with respect to regular equidistant positions and by additive noise. In right image, these displacements were compensated and noise was substantially reduced with the above-described sliding window algorithm.

One can further extend the applicability of this method to make interpolation kernel transform coefficients $\{\eta_r(\boldsymbol{p})\}$ in Eq. (30) to be adaptive to signal local features that exhibit themselves in signal local DCT spectra:

$$\left\{\boldsymbol{a}_k^p\right\} = \boldsymbol{ISDFT}_{1/2,0}\left\{\hat{\alpha}_r^{DCT}\left(\beta_r^{DCT}\right)\cdot\eta_r\left(\boldsymbol{p},\left\{\beta_r^{DCT}\right\}\right)\right\}. \tag{31}$$

The adaptivity may be desired in such applications as, for instance, resampling images that contain gray tone images in a mixture with graphical data. While discrete sinc-interpolation is completely perfect for gray tone images, it may produce undesirable oscillating artifacts in graphics.

The principle of local adaptive interpolation is schematically presented on Fig. 8. It assumes that modification of signal local DCT spectra for signal resampling and restoration in the above-described algorithm is supplemented with the spectrum analysis for generating a control signal. This signal is used to select, in each sliding window position, discrete sinc-interpolation or another interpolation method such as, for instance, nearest neighbor one. Fig. 9 compares non adaptive and adaptive sliding window sinc-interpolation on an example of a shift, by an interval equal 16.54 of discretization intervals, of a test signal composed of a sinusoidal wave and rectangle impulses. As one can see from the figure, non adaptive sinc-interpolated resampling of such a signal results in oscillations at the edges of rectangle impulses. Adaptive resampling implemented in this example switches between sinc-interpolation and nearest neighbor interpolation whenever energy of high frequency components of signal local spectrum is higher than a certain threshold level. As a result, rectangle impulses are re-sampled with “super-resolution”. Fig. 10 illustrates, for comparison, zooming a test signal by means of nearest neighbor, linear and bi-cubic spline interpolations and the above-described adaptive sliding window DCT sinc-interpolation. One can see from this figure that interpolation artifacts seen in other interpolation methods are absent when the adaptive sliding widow interpolation was used. Non adaptive and adaptive sliding window sinc-interpolation are also illustrated and compared in Fig. 11 for rotation of an image that contains gray tone and graphic components.

Computational complexity of sliding window DCT sinc-interpolation evaluated in terms of the number of multiplication and summation operations per signal sample is proportional to the window size thanks to the availability of recursive algorithms for computing DCT in sliding window[20-25]. It is comparable with that of other convolution based interpolation methods. Note that, for reconstruction of only window central sample, inverse DCT and DST of the modified window spectrum required by the interpolation algorithm is reduced to simple summations of the modified spectral coefficients:

$$\boldsymbol{b}_k = \frac{\mathbf{1}}{\sqrt{\mathbf{2}N}}\left[\alpha_{\mathbf{0}}^{DCT} + \sum_{r=\mathbf{1}}^{(N-\mathbf{1})/\mathbf{2}} \left(-\mathbf{1}\right)^r \left[\alpha_{\mathbf{2}r}^{DCT}\eta_{\mathbf{2}r}^{re}(p) - \alpha_{\mathbf{2}r-\mathbf{1}}^{DCT}\eta_{\mathbf{2}r-\mathbf{1}}^{re}(p)\right]\right], \qquad (32)$$

## 5. CONCLUSION

Two new methods of signal discrete sinc-interpolation are described. The first method implements, by means of signal processing in the domain of Discrete Cosine Transform, discrete sinc-interpolation that is practically free of boundary effects. The second method implements discrete sinc-interpolation in sliding window and allows arbitrary irregular signal resampling with simultaneous signal and image restoration and local adaptive interpolation with super resolution.

## REFERENCES

1. L. R. Rabiner, B. Gold, *Theory and application of Digital Signal Processing*, Englewood Cliffs, N J: Prentice Hall (1975)
2. D. Fraser, “Interpolation by the FFT Revisited – An Experimental Investigation”, IEEE Trans. , **ASSP- 37,** 665-675 (1989)
3. T. Smith, M.S. Smith, and S.T. Nichols, “Efficient sinc function interpolation technique for center padded data”, *IEEE Trans. ,* **ASSP- 38**, 1512-1517 (1990)
4. L. Yaroslavsky, “Efficient algorithm for discrete sinc-interpolation”, Applied Optics, **36**, 460-463 (1997)
5. M. Unser, P. Thevenaz, L. Yaroslavsky, “Convolution-based Interpolation for Fast, High-Quality Rotation of Images”, IEEE Trans. on Image Processing, 4, No. 10, 1371-1382 (1995)
6. L. Yaroslavsky, *Digital Picture Processing. An Introduction*, Springer Verlag , Berlin, Heidelberg, New York, Tokyo (1985).
7. L. Yaroslavsky, M. Eden, *Fundamentals of Digital Optics*, Birkhauser, Boston, (1996).
8. Z. Wang, “Fast algorithms for the discrete W transform and for the Discrete Fourier Transform”, IEEE Trans. ASSP, vol. ASSP-32, pp. 803-816, Aug. 1984
9. H.S. Hou, “A fast recursive algorithm for computing the Discrete Cosine Transform”, IEEE Trans. on ASSP, vol. ASSP-35, No. 10, Oct. 1987, p. 1455-1461
10. Z. Wang, “A simple structured algorithm for the DCT, in *Proc 3rd Ann. Conf. Signal Process. Xi’an*, China, Nov 1988, pp.28-31 (in Chinese).
11. A. Gupta, K. R. Rao, “A fast recursive algorithm for the discrete sine transform”, IEEE Trans. on ASSP, vol. ASSP-38, No. 3, March 1990, pp. 553-557.

12. Z. Wang, “Pruning the Fast Discrete Cosine Transform”, IEEE Transactions on communications, Vol. 39, No. 5, May 1991.

13. Z. Cvetkovic, M. V. Popovic, “New fast recursive algorithms for the computation of Discrete Cosine and Sine Transforms”, IEEE Trans. Signal Processing, vol. 40, No. 8, August 1992, pp. 2083-2086.

14. J. Ch. Yao, Ch-Y. Hsu, “Further results on “ New fast recursive algorithms for the discrete cosine and sine transforms”, IEEE Trans. on Signal Processing, vol. 42, No. 11, Nov. 1994, pp. 3254-3255

15. L. Yaroslavsky, A. Happonen, Y. Katyi, “Signal discrete sinc-interpolation in DCT domain: fast algorithms”, *SMMSP 2002, Second International Workshop on Spectral Methods and Multirate Signal Processing*, Toulouse (France), 07.09.2002 - 08.09.2002.

16. M. Unser, “Splines: A perfect fit for signal and image processing,” IEEE Trans. Signal Processing, 16, 22-38, (1999)

17. Ph. Thevenaz, Thierry Blu, M. Unser, “Interpolation Revisited”, IEEE Trans., MI-19, 739-758 (2000)

18. L. Yaroslavsky, “Image restoration, enhancement and target location with local adaptive filters”, In: *International Trends in Optics and Photonics, ICOIV*, T. Asakura, ed., (Springer Verlag, 1999), pp. 111-127

19. L.P. Yaroslavsky, K.O. Egiazarian, J.T. Astola, “Transform domain image restoration methods: review, comparison and interpretation”, In: *Photonics West, Conference 4304, Nonlinear Processing and Pattern Analysis*, (Proceedings of SPIE, v. 4304). (2001)

20. R. Yu. Vitkus, L.P. Yaroslavsky, “Recursive Algorithms for Local Adaptive Linear Filtration”, In: *Mathematical Research. Computer Analysis of Images and Pattern*s,ed. by

L.P. Yaroslavsky, A. Rosenfeld, W. Wilhelmi, Band 40, (Academie Verlag, Berlin, 1987), p. 34-39.

21. N. Rama Murthy, N.S. Swamy, "On Computation of Running Discrete Cosine and Sine Transform", IEEE Trans. on Signal Processing, vol. 40, No. 6, June 1992
22. K.J.R. Liu, C.T. Chiu, R.K. Kolagotla, and J.F. Jaja, "Optimal Unified Architectures for the Real-Time Computation of Time-Recursive Discrete Sinusoidal Transforms", IEEE, Transactions on Circuits and Systems for Video Technology, Vol. 4, No. 2, April 1994
23. J. A. R. Macias, and A. Exposito, "Recursive Formulation of Short-Time Discrete Trigonometric Transforms", IEEE Transactions on Circuits and Systems – II Analog and Digital Signal Processing, Vol. 45, No. 4, April 1998
24. V. Kober, and G. Cristobal, "Fast recursive algorithms for short-time discrete cosine transform", IEEE, Electronics Letters, 22nd July 1999, Vol. 35, No. 15
25. Jiangao Xi, Joe F. Chicharo, "Computing running DCT's and DST's based on their second order shift properties", IEEE Trans. on Circuits and Systems - I, Fundamental Theory and Applications, vol. 47, No. 5, May 2000

**Figure captions**

Fig. 1 Graphical illustration of the discrete sampling theorem: a) initial signal; b) initial signal with zeros placed between its samples; c) spectrum of signal (a); d) spectrum of signal (b): periodical replication of the initial signal spectrum; e) removing spectrum replicas that may cause aliasing by low pass filter; f) signal sinc-interpolated between samples of signal (b).

Fig. 2 Principle of signal convolution in DCT domain with signal extension by its mirror reflection

Fig. 3. Flow diagram of discrete sinc-interpolation in DCT domain for generating a p-shifted copy of a signal

Fig. 4. Zooming a fragment of an image (left) by sinc-interpolation in DFT domain (right upper image) and in DCT domain (right bottom image). Oscillations due to boundary effects that are clearly seen in DFT-interpolated image completely disappear in DCT-interpolated image.

Fig. 5. Windowed discrete sinc-functions with window size 11 and 15 samples (left) and their DFT spectra for $\mathbf{3}\times$ signal zooming(right).

Fig. 6 Flow diagram of simultaneous signal sliding window sinc-interpolation and restoration/enhancement in DCT domain

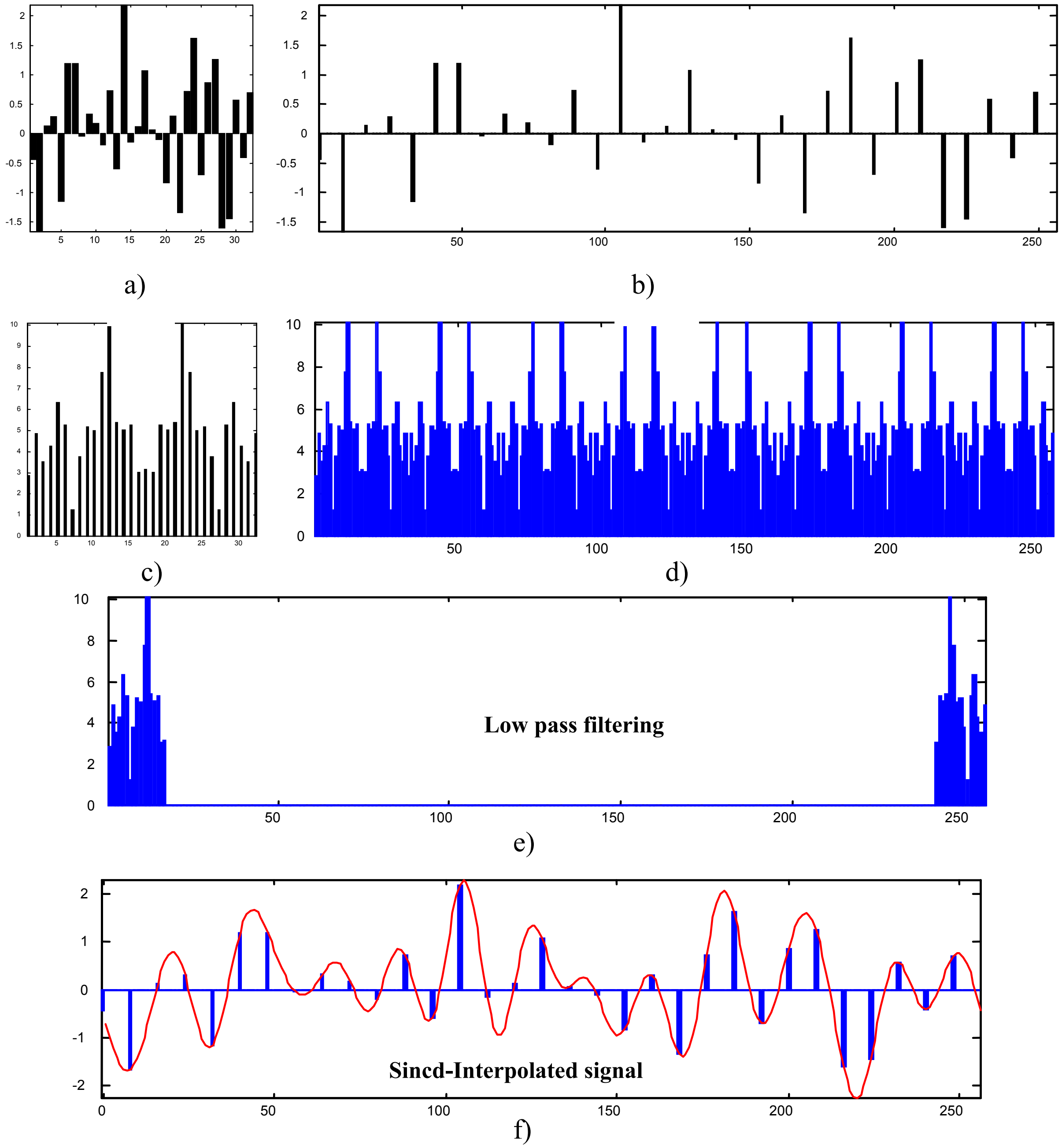
Fig. 7. Image rectification and denoising by resampling with sinc-interpolation in sliding window in DCT domain.

Fig. 8. Principle of local adaptive interpolation

Fig. 9. Signal (upper plot) shift by non-adaptive (middle plot) and adaptive (bottom plot) sliding window DCT sinc-interpolation. One can notice disappearance of oscillations at the edges of rectangle impulses when interpolation is adaptive.

Fig. 10. Comparison of nearest neighbor, linear, bicubic spline and adaptive sliding window sinc interpolation methods for zooming a digital signal (From left to right, from top to bottom: Continuous signal; initial sampled signal; nearest neighbor $\mathbf{8}\times$-interpolated signal; linearly $\mathbf{8}\times$-interpolated signal; cubic spline $\mathbf{8}\times$-interpolated signal; sliding window $\mathbf{8}\times$ sinc-interpolated signal).

Fig. 11. Image (upper) rotation with sliding window non-adaptive (left) and adaptive DCT sinc-interpolation (right). Note disappearance of oscillations at sharp edges thanks to switching between sinc-interpolation and nearest neighbor interpolation at the boundaries of black and white squares



a) b) c) d) e) f)

Fig. 1. Graphical illustration of the discrete sampling theorem: a) initial signal; b) initial signal with zeros placed between its samples; c) spectrum of signal (a); d) spectrum of signal (b): periodical replication of the initial signal spectrum; e) removing spectrum replicas that may cause aliasing by low pass filter; f) signal sinc-interpolated between samples of signal (b).

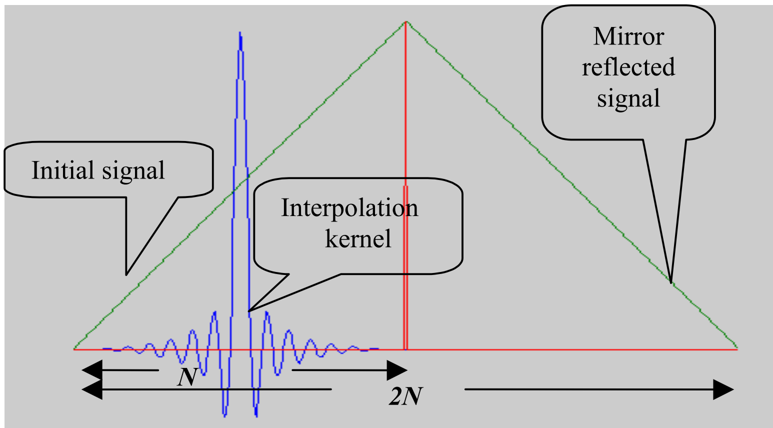


Fig. 2 Principle of signal convolution in DCT domain with signal extension by its mirror reflection

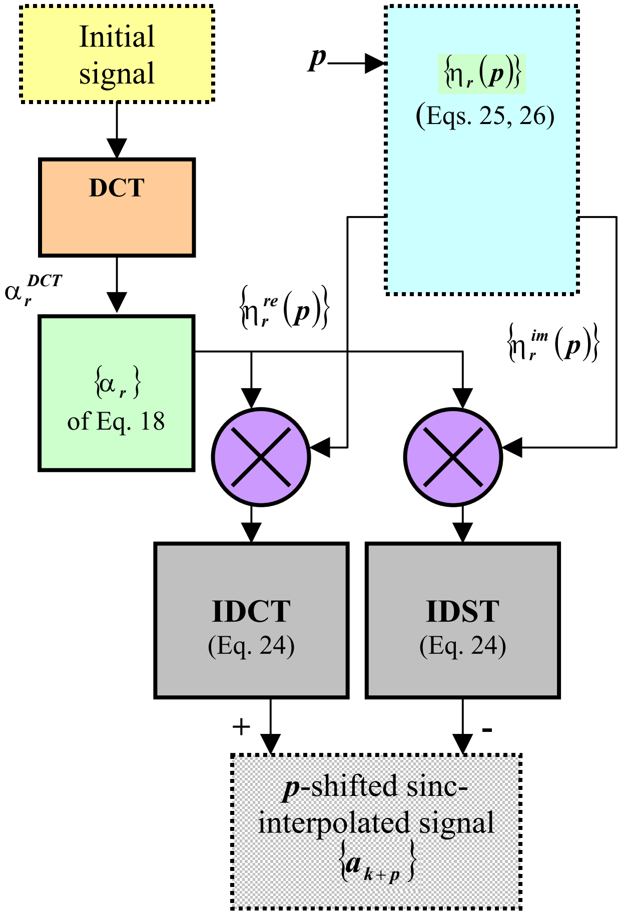


Fig. 3. Flow diagram of discrete sinc-interpolation in DCT domain for generating a p-shifted copy of a signal

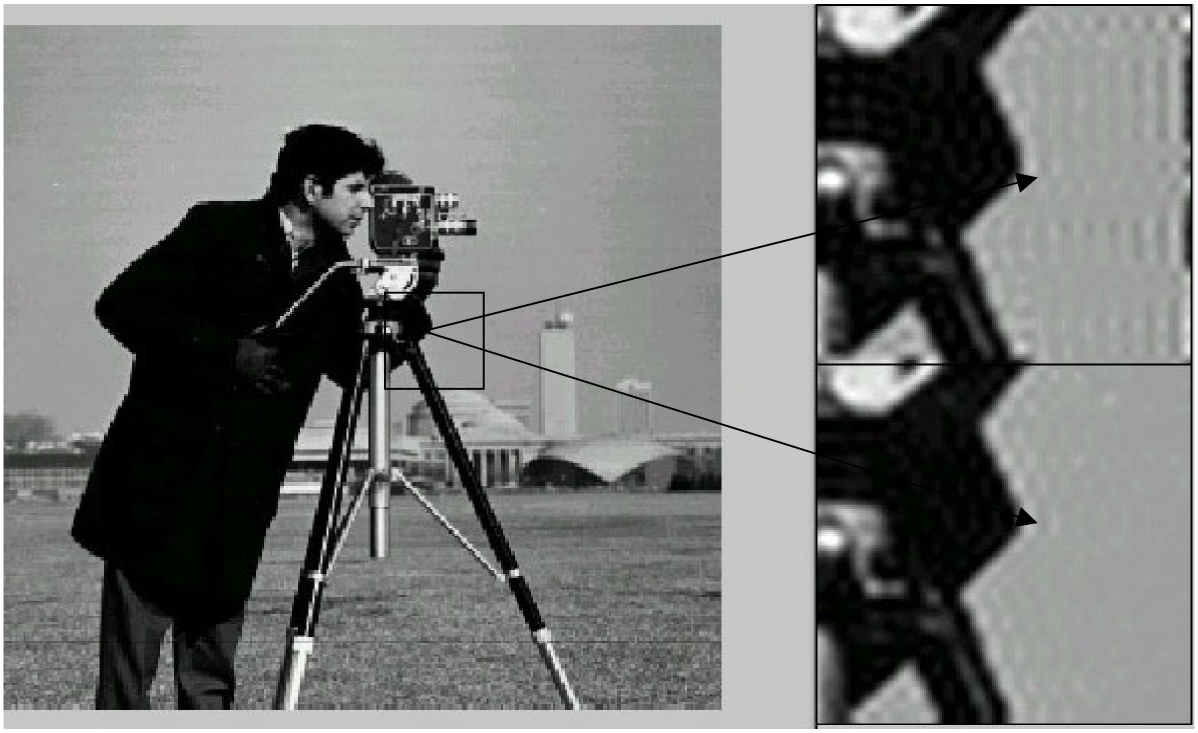


Fig. 4 Zooming a fragment of an image (left) by sinc-interpolation in DFT domain (right upper image) and in DCT domain (right bottom image). Oscillations due to boundary effects that are clearly seen in DFT-interpolated image completely disappear in DCT-interpolated image.

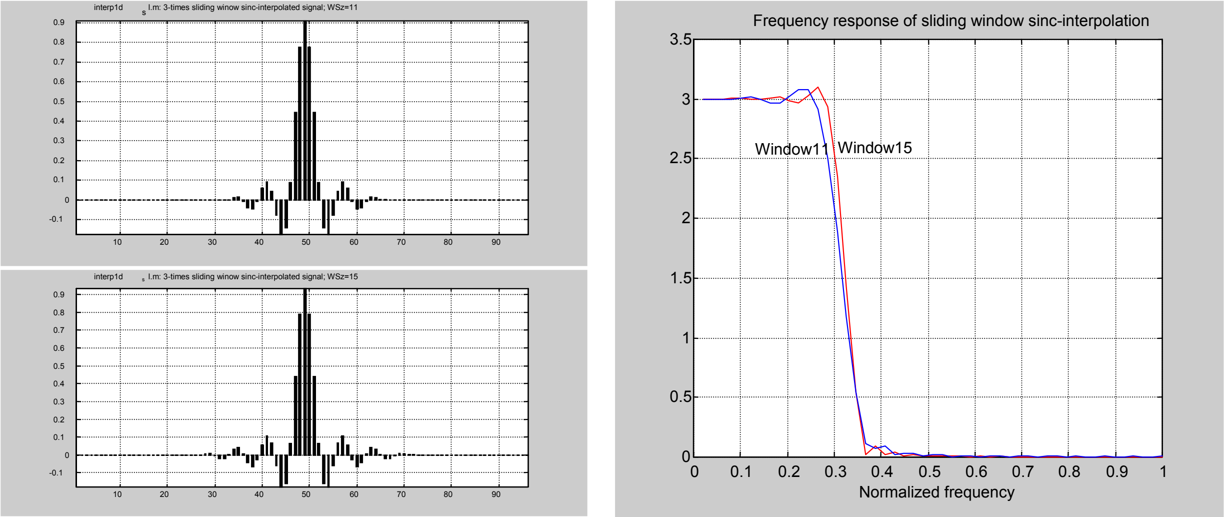


Fig. 5 Windowed discrete sinc-functions with window size 11 and 15 samples (left) and their DFT spectra for **3**× signal zooming(right).

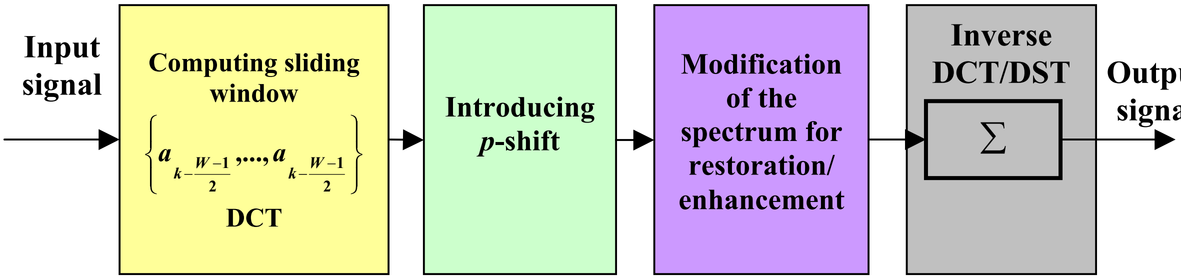
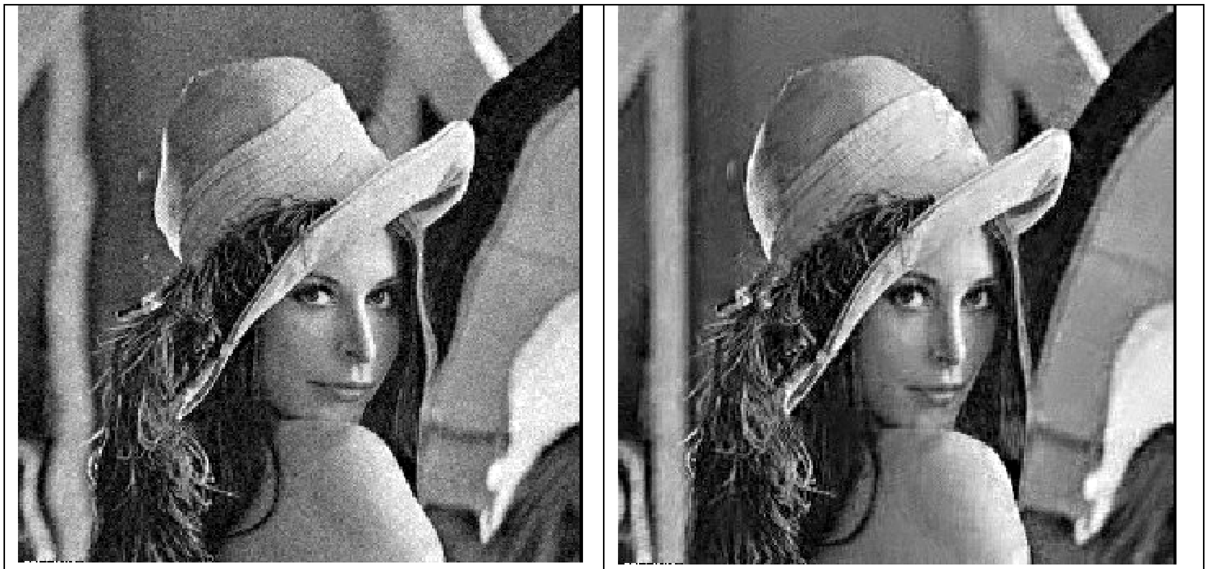

Input signal
Computing sliding window
$\left\{a_{k-\frac{W-1}{2}},\ldots,a_{k-\frac{W-1}{2}}\right\}$
DCT
Introducing *p*-shift
Modification of the spectrum for restoration/ enhancement
Inverse DCT/DST
Σ
Output signal

Fig. 6 Flow diagram of simultaneous signal sliding window sinc-interpolation and restoration/enhancement in DCT domain



Noisy image sampled in a irregular raster

Denoised and resampled (rectified) image

Fig. 7. Image rectification and denoising by resampling with sinc-interpolation in sliding window in DCT domain.

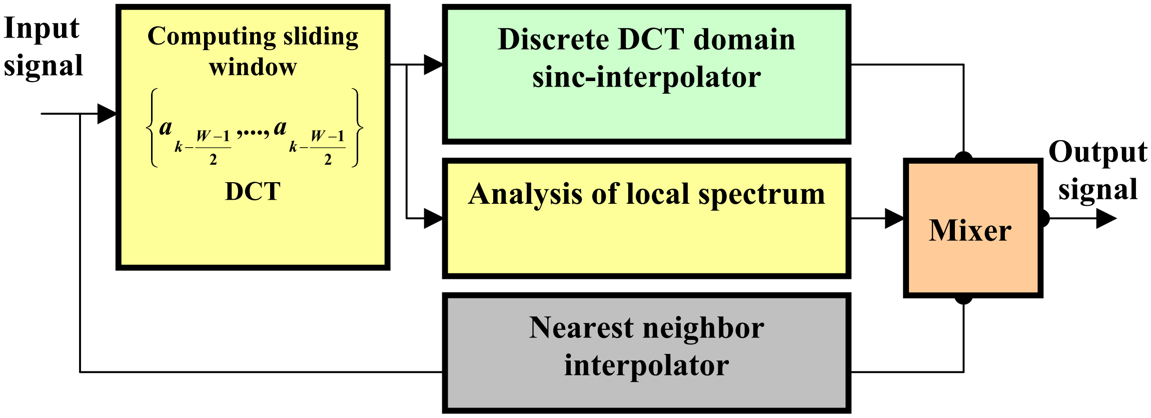

Input signal
Computing sliding window
$\left\{a_{k-\frac{W-1}{2}},\ldots,a_{k-\frac{W-1}{2}}\right\}$
DCT
Discrete DCT domain sinc-interpolator
Analysis of local spectrum
Nearest neighbor interpolator
Mixer
Output signal

Fig. 8. Principle of local adaptive interpolation

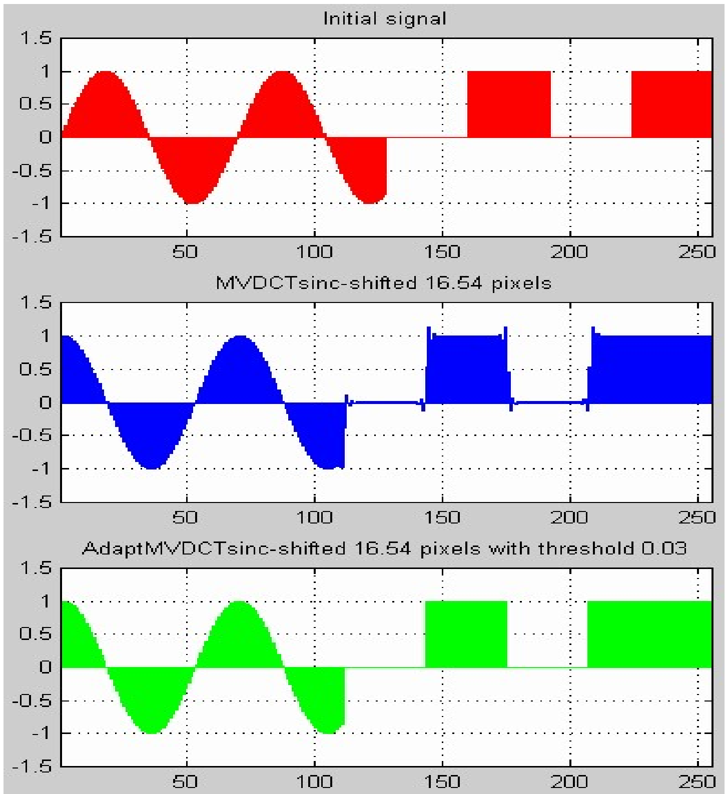


Fig. 9 Signal (upper plot) shift by non-adaptive (middle plot) and adaptive (bottom plot) sliding window DCT sinc-interpolation. One can notice disappearance of oscillations at the edges of rectangle impulses when interpolation is adaptive.

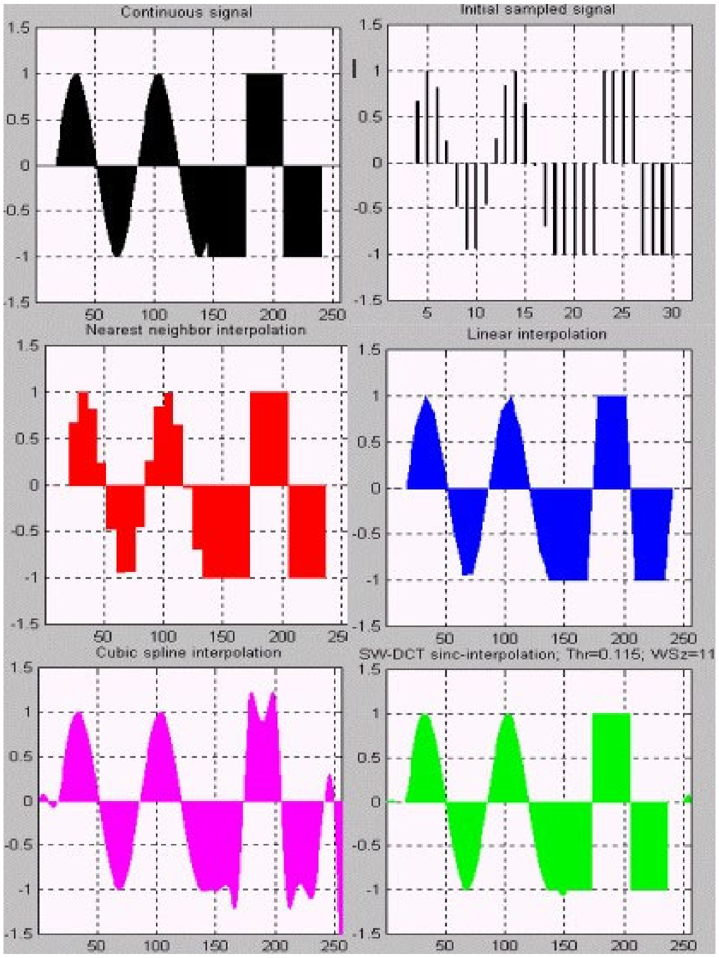


Fig. 10 Comparison of nearest neighbor, linear, bicubic spline and adaptive sliding window sinc interpolation methods for zooming a digital signal (From left to right, from top to bottom: Continuous signal; initial sampled signal; nearest neighbor **8**× -interpolated signal ; linearly **8**× -interpolated signal; cubic spline **8**× -interpolated signal; sliding window **8**× sinc-interpolated signal ).

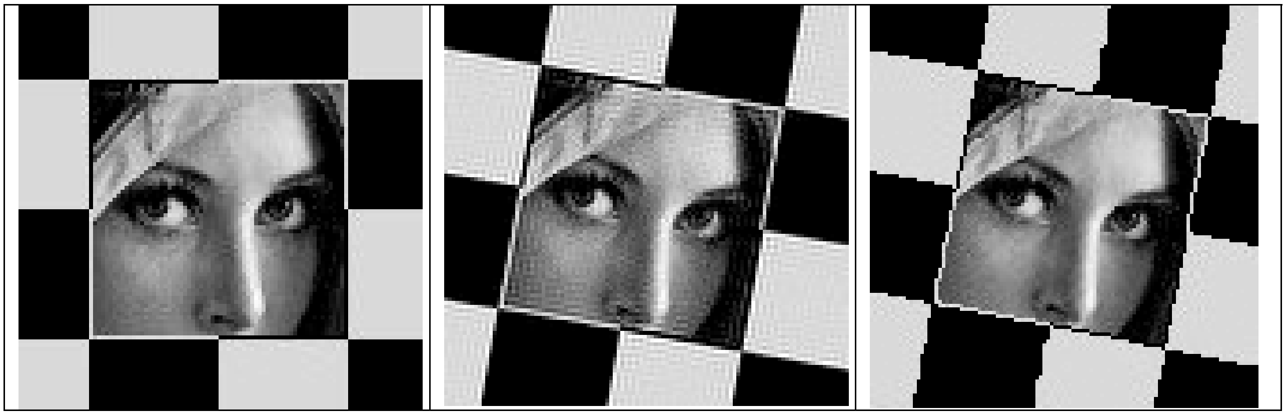


Fig. 11. Image (upper) rotation with sliding window non-adaptive (left) and adaptive DCT sinc-interpolation (right). Note disappearance of oscillations at sharp edges thanks to switching between sinc-interpolation and nearest neighbor interpolation at the boundaries of black and white squares