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

L. Yaroslavsky

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 because it does not distort the signal as defined by its samples and is completely reversible. However, sinc interpolation is frequently not considered in applications because 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. A solution that enables the elimination of 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 the discrete cosine transform (DCT) domain and local adaptive sinc interpolation in the 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. © 2003 Optical Society of America

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 subpixel accuracy, Radon transform and tomographic reconstruction, and three-dimensional (3-D) image volume rendering and volumetric imaging. Signal/image resampling procedes with the assumption that one or another method of interpolation between available signal/image samples is employed. By virtue of the sampling theorem, sinc interpolation is the best interpolation of continuous signals. Given samples $\{a_k\}$ of a continuous signal $a(x)$, the sinc-interpolated approximation $\tilde{\tilde{a}}\ (x)$ to this signal is defined as

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

where $\operatorname{sinc} x = (\sin x)/x$ and Δx is the signal discretization interval. If an unlimited number of samples obtained as

$$a_k = \frac{1}{\Delta x}\int_{-\infty}^{\infty} a(x)\operatorname{sinc}[\pi(x - k\Delta x)]\mathrm{d}x, \tag{2}$$

is available, sinc interpolation restores the continuous signal with the least mean squared error. For band-limited signals with a spectrum bandwidth $1/2\Delta x$, sinc interpolation provides an exact restoration of signals from their samples. In other words, signal sampling in accordance with Eq. (2) is in this case completely reversible. However, exact sinc interpolation cannot be implemented in reality because it requires an unlimited number of signal samples and an 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 a discrete sinc interpolation

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

where

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

The author is with the Faculty of Engineering, Interdisciplinary Studies Department, Tel Aviv University, Tel Aviv 69978, Israel. E-mail address for L. Yaroslavsky is yaro@eng.tau.ac.il.

Received 8 April 2003.
0003-6935/03/204166-10$15.00/0

is a discrete sinc function. It approximates the continuous sinc function $\mathrm{sinc}(x) = \sin x/x$ for $x/\Delta x \ll N$. As will be shown in Section 2, for a given finite number of signal samples, a 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 a signal spectrum zero-padding algorithm.[1–3] A more efficient and flexible discrete sinc-interpolation algorithm was described in Ref. 4. However, despite the attractiveness of discrete sinc interpolation it is frequently not regarded appropriate in some applications. First, the discrete sinc interpolation tends to produce heavy artifacts in the form of oscillations (ripples) at the signal borders. This property may be especially restrictive in image processing because image dimensions are usually relatively small numbers (256–1024), and noticeable ripples from the image left, right, upper, and bottom borders may occupy a 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. Owing to these reasons, discrete sinc interpolation is quite rarely practiced in digital signal and image processing.

In this paper, we introduce what are to our knowledge two new discrete sinc-interpolation algorithms that eliminate the above-mentioned drawbacks of the discrete sinc interpolation and offer additional useful capabilities not available with other methods. In Section 3, a computationally efficient and flexible algorithm of the discrete sinc interpolation is described that is free of oscillation phenomena at signal borders. In differentiation with the known discrete sinc-interpolation algorithms, this algorithm computes and modifies a discrete cosine transform (DCT) rather than a discrete Fourier transform (DFT) signal spectra. It is referred to as the global DCT domain discrete sinc-interpolation algorithm. In Section 4 a sliding window signal resampling algorithm is introduced that also works in the domain of the DCT. The algorithm is a good approximation to the ideal global discrete sinc interpolation and is capable of simultaneous signal denoising and of a local adaptation of the convolution kernel. The latter feature enables us, 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 the signal's initial sampling rate can be obtained.

2. Discrete Sinc Interpolation: Error-Free Interpolation of Sampled Data

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

$$\tilde{\tilde{a}}_k = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{L-1} a_{k_1} \delta(k_2) h_{\mathrm{int}}(k - k_1 L + k_2),$$
$$k = 0, 1, \ldots, LN - 1, \tag{5}$$

with the interpolation kernel $\{h_{\mathrm{int}}(k)\}$, of signal $\{\tilde{a}_k = a_{k_1}\delta(k_2)\}$, $k = k_1L + k_2$; $k_1 = 0, \ldots, N - 1$; $k_2 = 0, \ldots, L - 1$; $\delta(x) = 0^x$, obtained from the initial signal $\{a_k\}$ by placing $(L - 1)$ zeros between its samples as it is illustrated in Fig. 1(b). Compute the DFT of signal $\{\tilde{a}_k\}$:

$$\begin{aligned}
\tilde{\alpha}_r &= \frac{1}{\sqrt{LN}} \sum_{k=0}^{LN-1} \tilde{a}_k \exp\left(i2\pi \frac{kr}{LN}\right) \\
&= \frac{1}{\sqrt{LN}} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{L-1} a_{k_1} \delta(k_2) \exp\left[i2\pi \frac{(k_1L + k_2)}{LN} r\right] \\
&= \frac{1}{\sqrt{LN}} \sum_{k_1=0}^{N-1} a_{k_1} \exp\left(i2\pi \frac{k_1}{N} r\right) \\
&= \frac{1}{\sqrt{L}} \alpha_{(r) \bmod N},
\end{aligned} \tag{6}$$

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

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

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

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

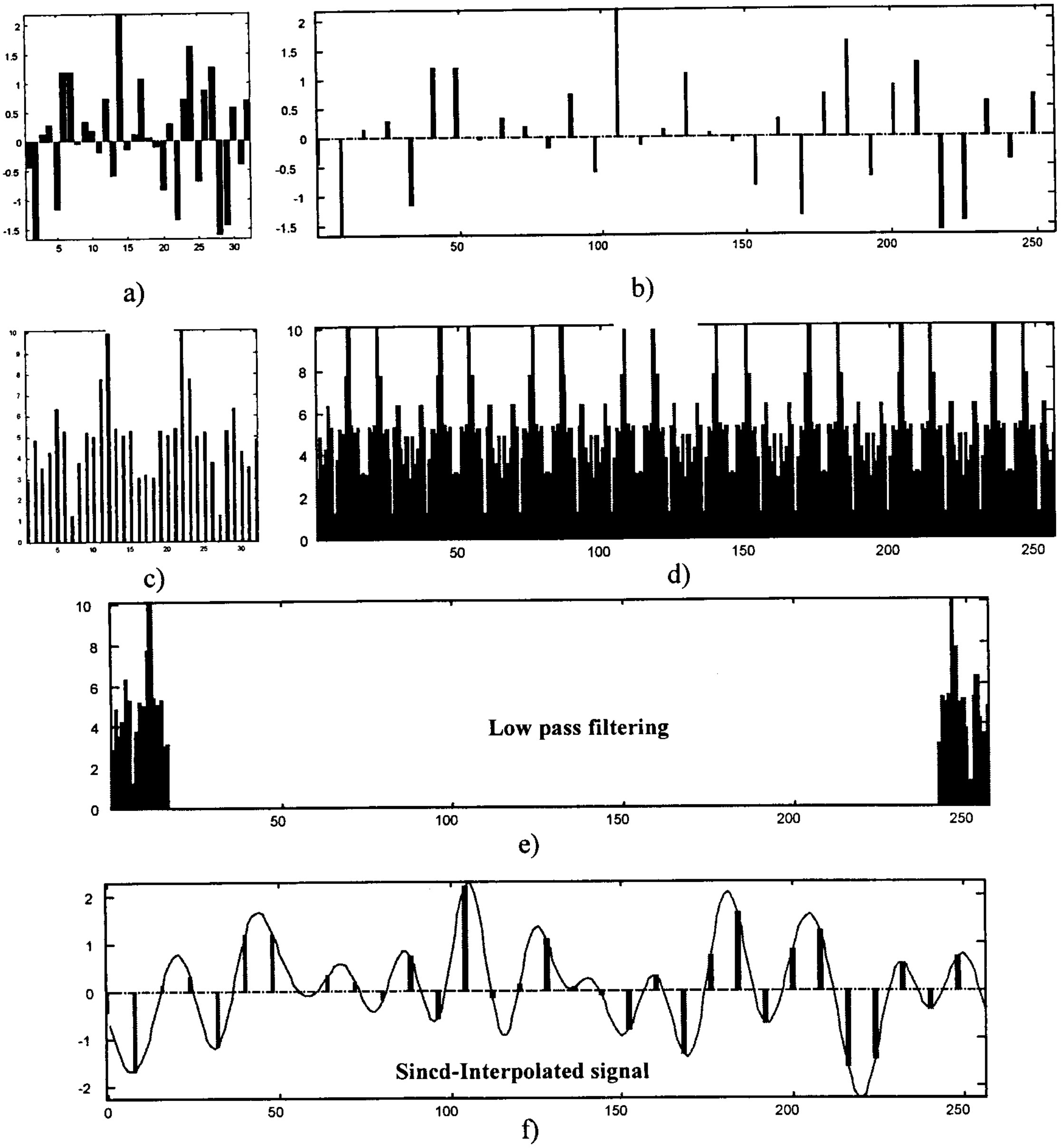

Fig. 1. Illustration of the discrete sampling theorem: (a) initial signal, (b) initial signal with zeros placed between its samples, (c) spectrum of signal (a), (d) spectrum of signal (b): periodical replication of the initial signal spectrum; (e) removing spectrum replicas that may cause aliasing by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

where

$$r = 0, 1, \ldots, LN - 1; \ \mathrm{rect}(x) = \begin{cases} 1, & 0 < x < 1 \\ 0, & \text{otherwise} \end{cases}.$$

The interpolated signal is then

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

where IDFT(·) is the inverse discrete Fourier transform and

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

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

For even-numbered N, interpolation described by Eq. (9) cannot be implemented since the term with index $(N+1)/2$ required by Eq. (9) does not exist. Two immediate options in this case are

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

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

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

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

where

$$\mathrm{sincd}(\pm 1; N; k) = [\mathrm{sincd}(N+1; N; k) + \mathrm{sincd}(N-1; N; k)]/2. \quad (14)$$

It follows from Eqs. (11)–(13) that, for even-numbered N, one cannot 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 the discrete sinc-interpolation function according to the right-hand sides of Eqs. (9), (11)–(13) requires $O(N)$ operations per output signal sample or $O(N^2L)$ operations for the entire sinc-interpolated output signal of NL samples. The computational complexity of the discrete sinc interpolation can be substantially reduced if the signal convolution is computed in the domain of the DFT with the use of fast Fourier transform (FFT) algorithms. Two methods of such an implementation are available. The first method, one with spectrum zero padding,[1–3] literally reproduces manipulations with the signal DFT spectrum described by the middle parts of Eqs. (9), (11), and (12). The algorithm computes the signal's DFT spectrum, pads it with $N(L-1)$ zeros, and then computes the inverse DFT of the obtained NL spectrum coefficients. Thanks to the use of the FFT for computing the DFT, the computational complexity of the algorithm is $O(N \log N + NL \log NL)$ operations for the entire output signal of NL samples or $O[(1 + 1/L)\log NL]$ operations per output signal sample. The second algorithm is described in Ref. 4. It enables a signal with N samples to generate its discrete sinc-interpolated copy of N samples shifted with respect to the initial samples by an arbitrary interval. Such a shifted signal $\{a_k^{(p)}\}$ is obtained as

$$\{a_k^{p}\} = \mathrm{IDFT}\{\alpha_r \varphi_r(p)\}, \quad (15)$$

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

$$\varphi_r(p) = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\mathrm{sincd}(K;\, N;\, k - p)\exp\left(i2\pi\frac{kr}{N}\right) \quad (16)$$

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

$$\varphi_r(p) = \begin{cases} \dfrac{1}{\sqrt{N}}\exp(i2\pi pr/N); & r = 0, 1, \ldots, (N-1)/2 - 1 \\ \varphi_{N-r}{}^{*}; & r = (N-1)/2 + 1, \ldots, N-1 \end{cases} \quad (17)$$

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

$$\varphi_r(p) = \begin{cases} \dfrac{1}{\sqrt{N}}\exp(i2\pi pr/N); & r = 0, 1, \ldots, N/2 - 1 \\ \dfrac{1}{\sqrt{N}}\cos(2\pi pr/N); & r = N/2 \\ \varphi_{N-r}{}^{*}; & r = N/2 + 1, \ldots, N-1 \end{cases} \quad (18)$$

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

It follows from Eq. (15) that the algorithm has a 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 arbitrary translation signal 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

The discrete sinc interpolation described in Section 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 the advantages of computing convolution with the use of an FFT if shifted DFT (SDFT$_{u,v}$)[6,7]

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

with shift parameters $u = 1/2$ (half discretization interval in the signal domain) and $v = 0$ (no shift in

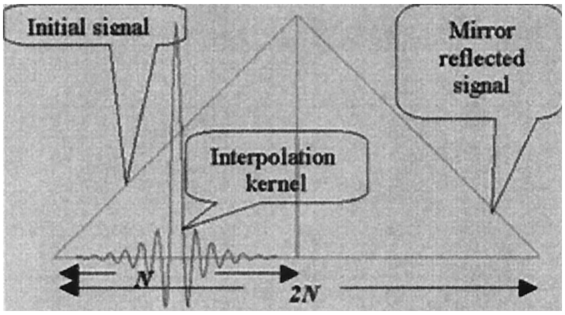

Fig. 2. Principle of signal convolution in the DCT domain with signal extension by its mirror reflection.

the Fourier domain) is used instead of the DFT. For a signal extended to its double length by mirror reflection, $\text{SDFT}_{1/2,0}$ is reduced to the DCT.[6,7] Filtering such signals in the SDFT domain is also of a cyclic convolution with a period of $2N$, where N is the number of samples of the initial signal. Therefore the 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 $2N$ samples as 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, the interpolation function $\{h_{\text{int}}^{(p)}(k)\}$ should be

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

where $K = N$ for odd-numbered N, and $K = \pm 1$ for even-numbered N.

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

$$\{a_k^{p}\} = \text{ISDFT}_{1/2,0}\{\alpha_r^{\text{DCT}} \cdot \eta_r(p)\}, \tag{21}$$

where $\text{ISDFT}_{1/2,0}$ is inverse $\text{SDFT}_{1/.2,0}$, $\{\alpha_r^{\text{DCT}}\}$ are DCT transform coefficients of signal $\{a_k\}$, and $\{\eta_r(p)\}$ are DFT coefficients of the interpolation function $\{h_{\text{int}}^{(p)}(k)\}$:

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

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

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

inverse SDFT $\text{ISDFT}_{1/2,0}$ for generating the interpolated signal according to Eq. (21) is reduced to inverse DCT and DST (discrete sine transform):

$$\begin{aligned} a_k^{(p)} &= \frac{1}{\sqrt{2N}} \sum_{r=0}^{2N-1} \alpha_r^{\text{DCT}} \eta_r(p) \exp\left[-i2\pi \frac{(k+1/2)r}{2N}\right] \\ &= \frac{1}{\sqrt{2N}} \Bigg(\alpha_0^{\text{DCT}}\eta_0 + \sum_{r=1}^{N-1} \alpha_r^{\text{DCT}} \\ &\quad \times \left\{\eta_r \exp\left[-i\pi \frac{(k+1/2)}{N} r\right] \right. \\ &\quad \left. + \eta_r^{*} \exp\left[i\pi \frac{(k+1/2)}{N} r\right]\right\}\Bigg) \\ &= \frac{1}{\sqrt{2N}} \Bigg\{\alpha_0^{\text{DCT}}\eta_0 \\ &\quad + 2\sum_{r=1}^{N-1} \alpha_r^{\text{DCT}} \eta_r^{\text{re}} \cos\left[\pi \frac{(k+1/2)}{N} r\right] \\ &\quad - 2\sum_{r=1}^{N-1} \alpha_r^{\text{DCT}} \eta_r^{\text{im}} \sin\left[\pi \frac{(k+1/2)}{N} r\right]\Bigg\}, \end{aligned} \tag{24}$$

where $\eta_r{}^{*}$ is a complex conjugate to η_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):

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

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

$$\begin{aligned}\eta_{2r+1}(p) = \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1} \text{sincd}(K;\, N;\, k \\ - p) \exp\left[i2\pi \frac{k(2r+1)}{2N}\right],\end{aligned} \tag{26}$$

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

A flow diagram of this algorithm for generating a p-shifted sinc-interpolated copy of the signal is shown in Fig. 3. Figure 4 illustrates the described discrete sinc interpolation in the DCT domain applied for image zooming and compares it with that implemented in the DFT domain. One can see that the algorithm does solve the problem of boundary effects characteristic for discrete sinc interpolation in the DFT domain. It is, however, computationally efficient only if a regular (equidistant) resampling signal is required. In this case it enables the computation of a 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

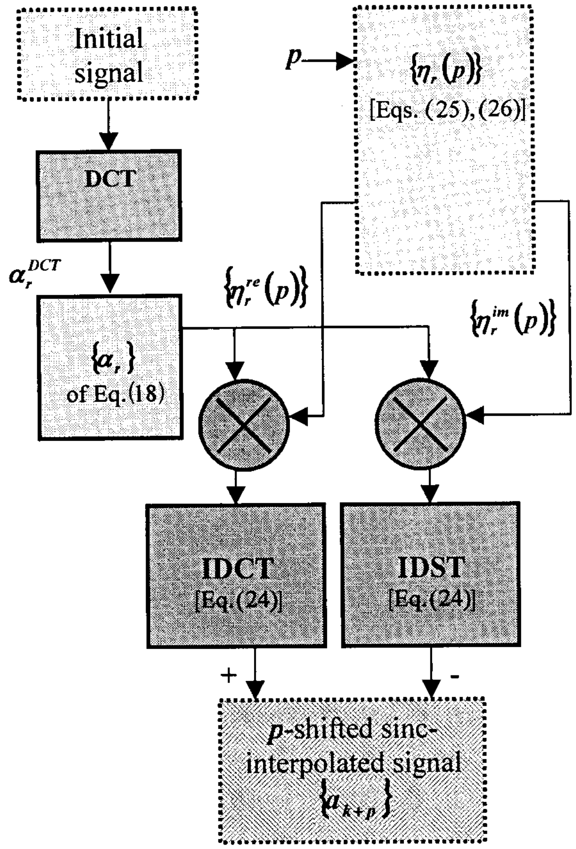
Initial signal
DCT
α_r^{DCT}
$\{\alpha_r\}$ of Eq.(18)
p
$\{\eta_r(p)\}$ [Eqs. (25),(26)]
$\{\eta_r^{re}(p)\}$
$\{\eta_r^{im}(p)\}$
IDCT [Eq.(24)]
IDST [Eq.(24)]
+
-
p-shifted sinc-interpolated signal $\{a_{k+p}\}$

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

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 a Sliding Window

When, as 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 the DCT domain has no efficient computational implementation. However, in these cases it can be implemented in a sliding window. In processing a signal in the 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. The interpolation function in this case is a windowed discrete sinc function whose extent is equal to the window size rather than to the signal size required for the perfect discrete sinc interpolation. Figure 5 illustrates frequency responses of the corresponding low-pass filters for different window sizes. As one can see from the figure, they deviate from a rectangular function as 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 over 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 a sliding window in the 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 k of the window of W samples (usually an odd number), compute transform coefficients $\{\beta_r = T\{b_n\}\}$ of the signal $\{b_n\}$ in the window ($n, r = 1, 2, \ldots, W$) and nonlinearly modify them to obtain the coefficients $\{\hat{\alpha}_r(\beta_r)\}$. These coefficients are used to generate an estimate $\hat{a}_k$ of the window

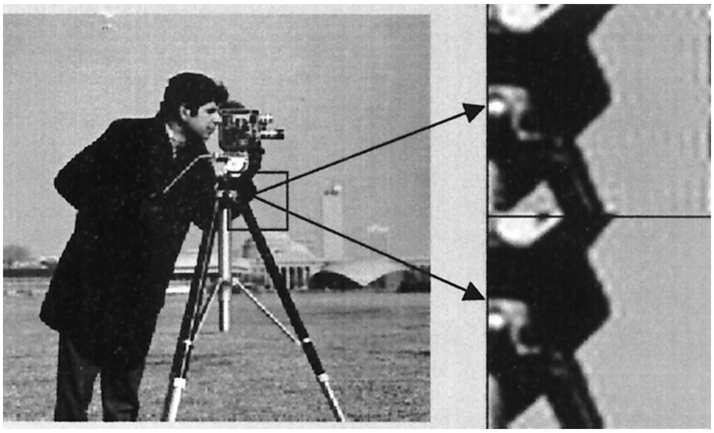

Fig. 4. Enlarging a fragment of an image (left) by sinc interpolation in the DFT domain (upper-right image) and in the DCT domain (bottom-right image). Oscillations due to boundary effects that are clearly seen in a DFT-interpolated image completely disappear in the DCT-interpolated image.

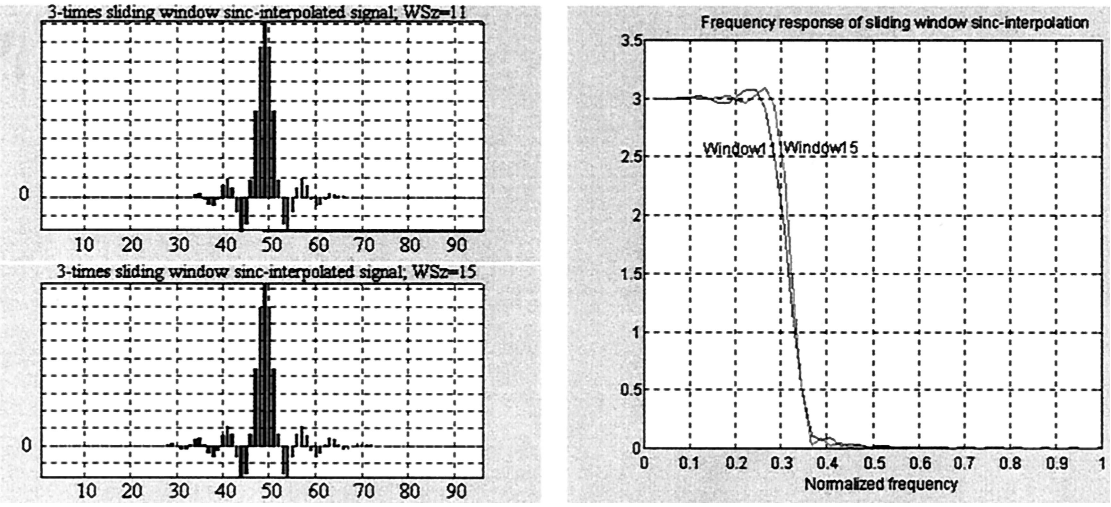

Fig. 5. Windowed discrete sinc functions with a window size of 11 and 15 samples (left) and their DFT spectra for 3× signal enlarging (right).

central pixel by inverse transform $T_k^{-1}\{\}$ computed for the window central pixel as

$$\hat{a}_k = T_k^{-1}\{\hat{\alpha}_r(\beta_r)\}. \tag{27}$$

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

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

or hard thresholding

$$\hat{\alpha}_r = \begin{cases} \beta_r, & |\beta_r|^2 > \mathrm{Thr} \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

are used 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

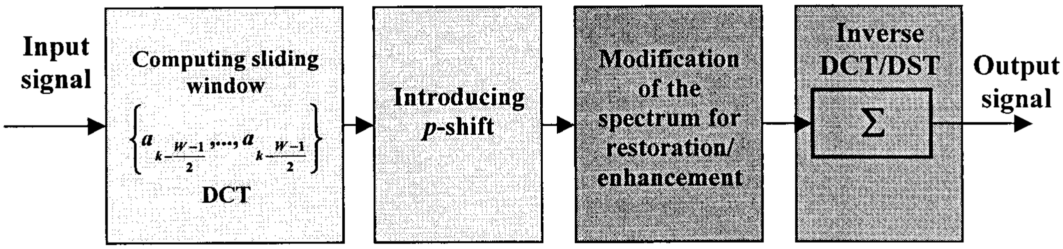
Input signal
Computing sliding window
$\left\{a_{k-\frac{W-1}{2}},\ldots,a_{k-\frac{W-1}{2}}\right\}$
DCT
Introducing p-shift
Modification of the spectrum for restoration/ enhancement
Inverse DCT/DST
Σ
Output signal

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

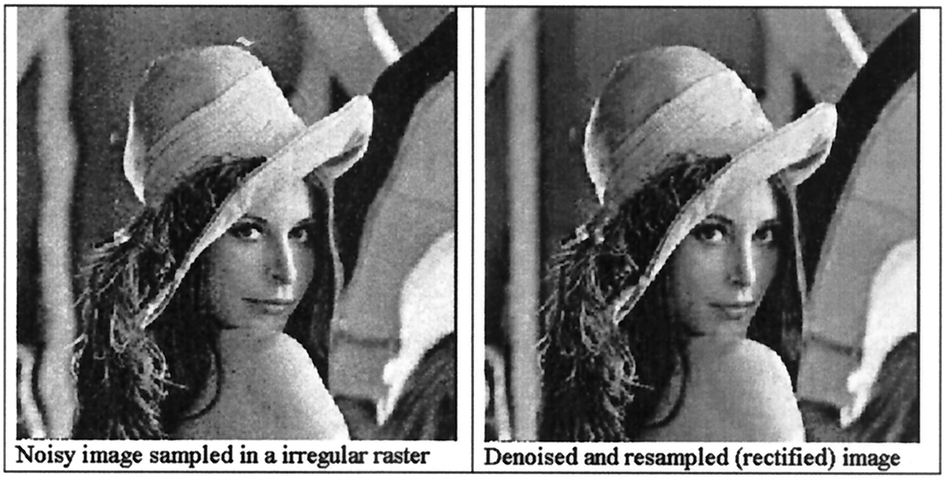
Noisy image sampled in a irregular raster
Denoised and resampled (rectified) image

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

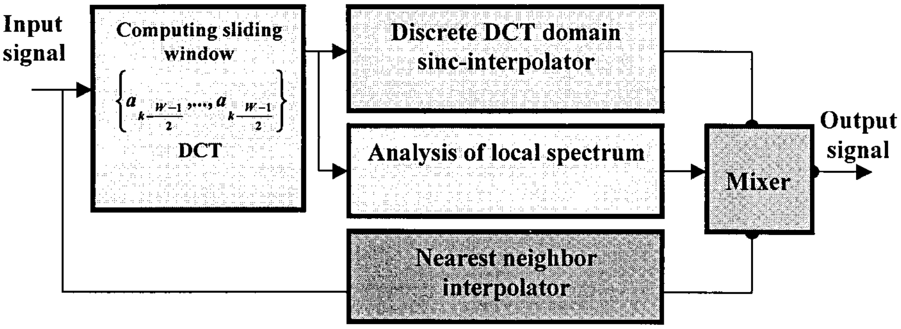
Input signal
Computing sliding window
$\left\{a_{k-\frac{W-1}{2}},\dots,a_{k\frac{W-1}{2}}\right\}$
DCT
Discrete DCT domain sinc-interpolator
Analysis of local spectrum
Mixer
Output signal
Nearest neighbor interpolator

Fig. 8. Principle of local adaptive interpolation.

way, combine sliding-window DCT domain discrete sinc-interpolation signal resampling (Eq. 21) and filtering for signal restoration and enhancement (Eq. 27):

$$\{a_k^p\} = \mathrm{ISDFT}_{1/2,0}\{\hat{\alpha}_r^{\mathrm{DCT}}(\beta_r^{\mathrm{DCT}})\eta_r(p)\}. \tag{30}$$

Figure 6 shows a 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. Figure 7 illustrates the application of the combined filtering/interpolation for image irregular-to-regular resampling combined with denoising. In this example, the left image is distorted by known displacements of pixels with respect to regular equidistant positions and by additive noise. In the 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 $\{\eta_r(p)\}$ in Eq. (30) to be adaptive to local signal features that exhibit themselves in local signal DCT spectra:

$$\{a_k^p\{ = \mathrm{ISDFT}_{1/2,0}\{\hat{\alpha}_r^{\mathrm{DCT}}(\beta_r^{\mathrm{DCT}})\eta_r(p, \{\beta_r^{\mathrm{DCT}}\})\}. \tag{31}$$

The adaptivity may be desired in such applications as, for instance, resampling images that contain gray-scale images in a mixture with graphical data. While discrete sinc interpolation is completely perfect for gray-scale 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 local signal 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, the nearest neighbor method. Figure 9 compares nonadaptive and adaptive sliding-window sinc interpolation on an example of a shift, by an interval equal to 16.54 of the discretization intervals, of a test signal composed of a sinusoidal wave and rectangular impulses. As one can see from Fig. 9, nonadaptive sinc-interpolated resampling of such a signal results in oscillations at the edges of rectangular impulses. Adaptive resampling implemented in this example switches between sinc interpolation and nearest-neighbor interpolation whenever energy of high-frequency components of local signal spectrum is higher than a certain threshold level. As a result, rectangular impulses are resampled with superresolution. Figure 10 illustrates, for comparison, enlarging a test signal by means of nearest neighbor, linear, and bicubic 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-window interpolation was used. Nonadaptive and adaptive sliding-window sinc interpolation are also illustrated and compared in Fig. 11 for rotation of an

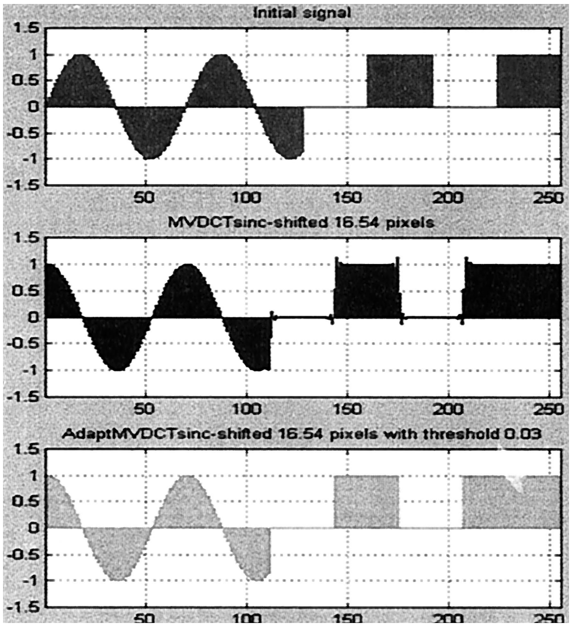

Fig. 9. Signal (upper plot) shift by nonadaptive (middle plot) and adaptive (bottom plot) sliding-window DCT sinc interpolation. One can notice the disappearance of oscillations at the edges of the rectangular impulses when interpolation is adaptive.

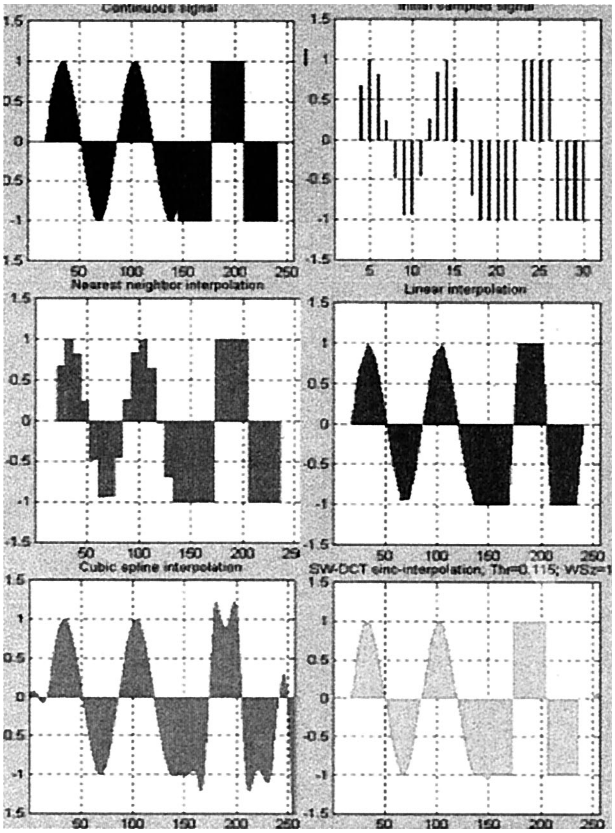

Fig. 10. Comparison of nearest neighbor, linear, bicubic spline, and adaptive sliding-window sinc-interpolation methods for enlarging a digital signal. (From left to right, from top to bottom: Continuous signal, initial sampled signal, nearest-neighbor 8×-interpolated signal; linearly 8×-interpolated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

image that contains gray-scale 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 because of the availability of recursive algorithms for computing DCT in sliding windows.[20–25] It is comparable with that of other convolution-based interpolation methods. Note that for reconstruction of only the window central sample, the inverse DCT and DST of the modified window spectrum required by the interpolation algorithm is reduced to simple summations of the modified spectral coefficients:

$$b_k = \frac{1}{\sqrt{2N}}\left\{\alpha_0^{\mathrm{DCT}} + \sum_{r=1}^{(N-1)/2} (-1)^r\left[\alpha_{2r}^{\mathrm{DCT}}\eta_{2r}^{\mathrm{re}}(p) - \alpha_{2r-1}^{\mathrm{DCT}}\eta_{2r-1}^{\mathrm{re}}(p)\right]\right\}. \qquad (32)$$

5. Conclusion

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

The author thanks J. Astola, K. Eguiazaryan, A. Gotchev, and A. Happonen from Tampere International Center for Signal Processing, Tampere Institute of Technology, Tampere, Finland, for their assistance and fruitful discussions.

References

1. L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, N.J., 1975).
2. D. Fraser, "Interpolation by the FFT Revisited—An Experimental Investigation," IEEE Trans. Acoust. Speech Signal Process. **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. Acoust. Speech Signal Process. **ASSP-38,** 1512–1517 (1990).
4. L. Yaroslavsky, "Efficient algorithm for discrete sinc interpolation," Appl. Opt. **36,** 460–463 (1997).
5. M. Unser, P. Thevenaz, and L. Yaroslavsky, "Convolution-based interpolation for fast, high-quality rotation of images," IEEE Trans. Image Process. **4,** 1371–1382 (1995).
6. L. Yaroslavsky, *Digital Picture Processing: An Introduction* (Springer-Verlag, Berlin, 1985).
7. L. Yaroslavsky and M. Eden, Fundamentals of Digital Optics (Birkhäuser, Boston, 1996).
8. Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," IEEE Trans. Acoust. Speech Signal Process. **ASSP-32,** 803–816 (1984).
9. H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," IEEE Trans. Acoust. Speech Signal Process. **ASSP-35,** 1455–1461 (1987).

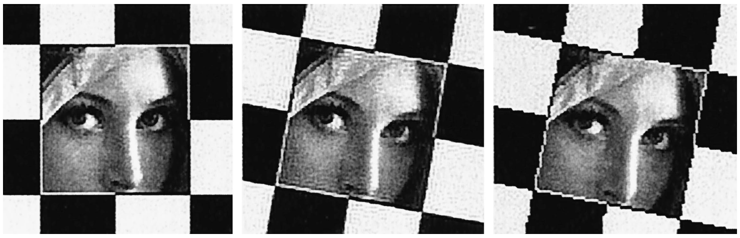

Fig. 11. Image (upper) rotation with sliding window nonadaptive (left) and adaptive DCT sinc interpolation (right). Note disappearance of oscillations at sharp edges thanks to switching between sinc interpolation and nearest-neighbor interpolation at the boundaries of black and white squares.

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 and K. R. Rao, “A fast recursive algorithm for the discrete sine transform,” IEEE Trans. Acoust. Speech Signal Process. **ASSP-38,** 553–557 (1990).

12. Z. Wang, “Pruning the Fast Discrete Cosine Transform,” IEEE Trans. Commun. **39,** 640–643 (1991).

13. Z. Cvetkovic and M. V. Popovic, “New fast recursive algorithms for the computation of discrete cosine and sine transforms,” IEEE Trans. Signal Process. **40,** 2083–2086 (1992).

14. J. Ch. Yao and Ch-Y. Hsu, “Further results on “New fast recursive algorithms for the discrete cosine and sine transforms,” IEEE Trans. Signal Process. **42,** 3254–3255 (1994).

15. L. Yaroslavsky, A. Happonen, and 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 Process. **16,** 22–38 (1999).

17. Ph. Thevenaz, Th. Blu, and M. Unser, “Interpolation Revisited,” IEEE Trans. Med. Imaging **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, Berlin, 1999), pp. 111–127.

19. L. P. Yaroslavsky, K. O. Egiazarian, and J. T. Astola, “Transform domain image restoration methods: review, comparison, and interpretation,” in Photonics West 2001: Electronic Imaging Nonlinear Processing and Pattern Analysis, Proc. SPIE **4304,** 155–170 (2001).

20. R. Yu. Vitkus and L. P. Yaroslavsky, “Recursive Algorithms for Local Adaptive Linear Filtration,” in *Mathematical Research, Computer Analysis of Images and Patterns*, L. P. Yaroslavsky, A. Rosenfeld, and W. Wilhelmi, eds. Band 40, (Academie Verlag, Berlin, 1987), pp. 34–39.

21. N. Rama Murthy, N. S. Swamy, “On computation of running discrete cosine and sine transform,” IEEE Trans. Signal Process. **40,** 1430–1437 (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 Trans. 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 Circuits Syst. II Analog and Digital Signal Processing, **45,** 525–527 (1998).

24. V. Kober and G. Cristobal, “Fast recursive algorithms for short-time discrete cosine transform,” Electron. Lett. **35,** 1236–1238 (1999).

25. J. Xi and J. F. Chicharo, “Computing running DCT’s and DST’s based on their second order shift properties,” IEEE Trans. Circuits Syst. - I, Fundamental Theory and Applications, **47,** 779–783 (2000).