Fast signal sinc-interpolation methods for signal and image resampling

L. Yaroslavsky, Dept. of Interdisciplinary Studies, Faculty of Engineering,
Tel Aviv University, 69978 Tel Aviv, Israel

ABSTRACT

Digital signal resampling is required in many digital signal and image processing applications. Among the digital convolution based signal resampling methods, sinc-interpolation is theoretically the best one since it does not distort the signal defined by its samples. Discrete sinc-interpolation is most frequently implemented by the "signal spectrum zero padding method". However, this method is very inefficient and inflexible. Sinc-interpolation badly suffers also from boundary effects. In the paper, a flexible and computationally efficient methods for boundary effects free discrete sinc-interpolation are presented in two modifications: frame (global) sinc-interpolation in DCT domain and sinc-interpolation in sliding window (local). In sliding window interpolation, interpolation kernel is a windowed sinc-function. Windowed sinc-interpolation offers options not available with other interpolation methods: interpolation with simultaneous local adaptive signal denoising and adaptive interpolation with super resolution. The methods outperform other existing discrete signal interpolation methods in terms of the interpolation accuracy and flexibility of the interpolation kernel design. Their computational complexity is *O(log(Size of the frame))* per output sample for frame interpolation and *O(Window Size)* per output sample for sliding window interpolation.

Keywords: Signal interpolation, Sinc-interpolation, Spline interpolation, Super resolution

1. INTRODUCTION

Signal and image resampling is required in many signal/image processing applications. It is a key issue in audio signal spectral analysis and fractional delay, target location and tracking with sub-pixel accuracy, image geometrical transformations and rescaling, Radon Transform and tomographic reconstruction, 3-D image volume rendering and volumetric imaging. Although it is almost commonly agreed that, in view of the sampling theorem, sinc-interpolation is theoretically the best one, it is rarely used in practice because of computational inefficiency and inflexibility of the most known sinc-interpolation, spectrum zero-padding algorithm and because sinc-interpolation tends to produce ringing at signal edges which is frequently considered undesirable artifact. In the paper, we advocate using discrete sinc-interpolation implemented via signal processing in DFT domain by proving, in Sect. 2, that discrete sinc-interpolation is the only interpolation method that does not introduce any distortions in the initial signal as defined by its samples, and by offering, in Sects. 3, a computationally efficient and flexible discrete sinc-interpolation algorithm and in Sect. 4, its modification free of Gibbs phenomena at signal boundaries. Finally, Sect. 5, we suggest sliding window signal resampling algorithms that, being good approximations to the ideal sinc-interpolation, are capable also of simultaneous signal denoising and of local adaptation. The latter allows, in particular, to eliminate oscillations at signal edges.

2. DISCRETE SAMPLING THEOREM. DISCRETE SINC INTERPOLATION AS A GOLD STANDARD FOR INTERPOLATION OF SAMPLED DATA

Let a signal of $\boldsymbol{N}$ samples $\{a_k\}$is to be interpolated to a signal with $\boldsymbol{L}$ interpolated samples per each initial one. For convolution based interpolation, the interpolation process is digital convolution, with interpolation kernel $\{h_{\text{int}}(k)\}$,

$$\tilde{a}_k = \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{L-1} a_{n_1}\delta(n_2)h_{\text{int}}(k - n_1L + n_2),\ k = 0,1,...,LN-1 \tag{1}$$

of signal $\tilde{a}_k = a_{k_1}\delta(k_2)$ obtained from initial signal $\{a_k\}$ by placing ($\boldsymbol{L}$-1) zeros between its samples ($\delta(x) = 0^x$). Compute DFT of this signal:

Image Processing: Algorithms and Systems, Edward R. Dougherty, Jaakko T. Astola, Karen O. Egiazarian, Editors, Proceedings of SPIE Vol. 4667 (2002) © 2002 SPIE · 0277-786X/02/$15.00

$$DFT\left(\tilde{a}_k\right)=\tilde{\alpha}_r=\frac{1}{\sqrt{LN}}\sum_{k_2}^{L-1}\sum_{k_1=0}^{N-1}a_{k_1}\delta\left(k_2\right)\exp\left[i2\pi\frac{\left(k_1L+k_2\right)}{LN}r\right]=\frac{1}{\sqrt{LN}}\sum_{k_1=0}^{N-1}a_{k_1}\exp\left[i2\pi\frac{k_1}{N}r\right]=\frac{1}{\sqrt{L}}\alpha_{(r)\bmod N}, \tag{2}$$

where $\{\alpha_r\}$ is DFT of signal $\{a_k\}$. Eq. (2) shows that sampling discrete signal (placing zeros between its samples) results in periodical replication of its spectrum with the number of replicas equal to the sampling interval (the number of zeros plus one). If digital convolution is computed as cyclic (periodical) one, in DFT domain it will correspond to masking spectrum $\{\tilde{\alpha}_r\}$ with DFT of the interpolation kernel:

$$DFT\left(\{\tilde{a}_k\}\right)=\tilde{\alpha}_r=\left(\alpha_{(r_1)\bmod K_1}\right)\cdot DFT\left(\{h_{\mathrm{int}}(k)\}\right). \tag{3}$$

Therefore, the only way to avoid, in the interpolation, aliasing and signal distortions is signal ideal low pass filtering, when $DFT\left(\{h_{\mathrm{int}}(k)\}\right)$ is a rectangle function. For ***N***-odd number, $DFT\left(\{h_{\mathrm{int}}(k)\}\right)$ should be:

$$DFT\left(h_{\mathrm{int}}(k)\right)=1-rect\frac{r-(N+1)/2}{LN-N-1}, \tag{5}$$

where $r=0,1,...,LN-1$; $rect(x)=\left\{\begin{matrix}1, & 0<x<1\\ 0, & otherwise\end{matrix}\right\}$. In this case, interpolated signal is

$$\tilde{\tilde{a}}_k=\frac{1}{\sqrt{L}}\sum_{n_1=0}^{N-1}a_n\,\mathrm{sincd}\left(\mathrm{N};\mathrm{N};\left(\mathrm{k}-\mathrm{n}_1L\right)\right)=IDFT\left(\left[1-rect\frac{r-(N+1)/2}{LN-N-1}\right]\alpha_{(r)\bmod N}\right), \tag{6}$$

where

$$\mathrm{sincd}(\mathrm{K};\mathrm{N};\mathrm{x})=\frac{\sin(\pi Kx/N)}{N\sin(\pi x/N)} \tag{7}$$

is discrete sinc-function. It approximates continuous sinc function $\mathrm{sinc}(\mathrm{x})=\sin x/x$ for ***x<<N.*** For even ***N***, similar relationships can be obtained[1,2]:

$$\tilde{\tilde{a}}_k=\frac{1}{\sqrt{L}}\sum_{n_1=0}^{N-1}a_n\,\mathrm{sincd}\left(\mathrm{N-1};\mathrm{N};\left(\mathrm{k}-\mathrm{n}_1L\right)\right)=IDFT\left(\left[1-rect\frac{r-N/2}{LN-N}\right]\alpha_{(r)\bmod N}\right); \tag{8}$$

$$\tilde{\tilde{a}}_k=\frac{1}{\sqrt{L}}\sum_{n_1=0}^{N-1}a_n\,\mathrm{sincd}\left(\mathrm{N+1};\mathrm{N};\left(\mathrm{k}-\mathrm{n}_1L\right)\right)=IDFT\left(\left[1-rect\frac{r-N/2-1}{LN-N-2}\right]\alpha_{(r)\bmod N}\right). \tag{9}$$

Equations (6-9) are formulations of the discrete sampling theorem. They prove that, given samples of a digital signal,

- the signal cyclic convolution based interpolation does not distort initial signal samples iff the number of initial signal samples ***N*** is odd number and interpolation kernel is discrete sinc function, (Eq. 7, ***K=N***)
- the signal cyclic convolution based interpolation distort only the most high frequency component $\alpha_{N/2}$ (zeros it or doubles it) of the initial signal DFT spectrum iff the number of initial signal samples ***N*** is even number and interpolation kernel is the discrete sinc function (Eq. 7, ***K=N***-1 or ***N+1***, respectively);
- Discrete sinc-interpolation approximates continuous sinc interpolation $a(x)=\sum_{n=-\infty}^{\infty}a_n\mathrm{sinc}(x/\Delta x-n)$ where Δx is continuous signal discretization interval and $\mathrm{sinc}(x)=\sin x/x$, to the accuracy of boundary effects and converges to it as $N\to\infty$.

Graphical illustration of these reasoning is shown in Fig. 1.

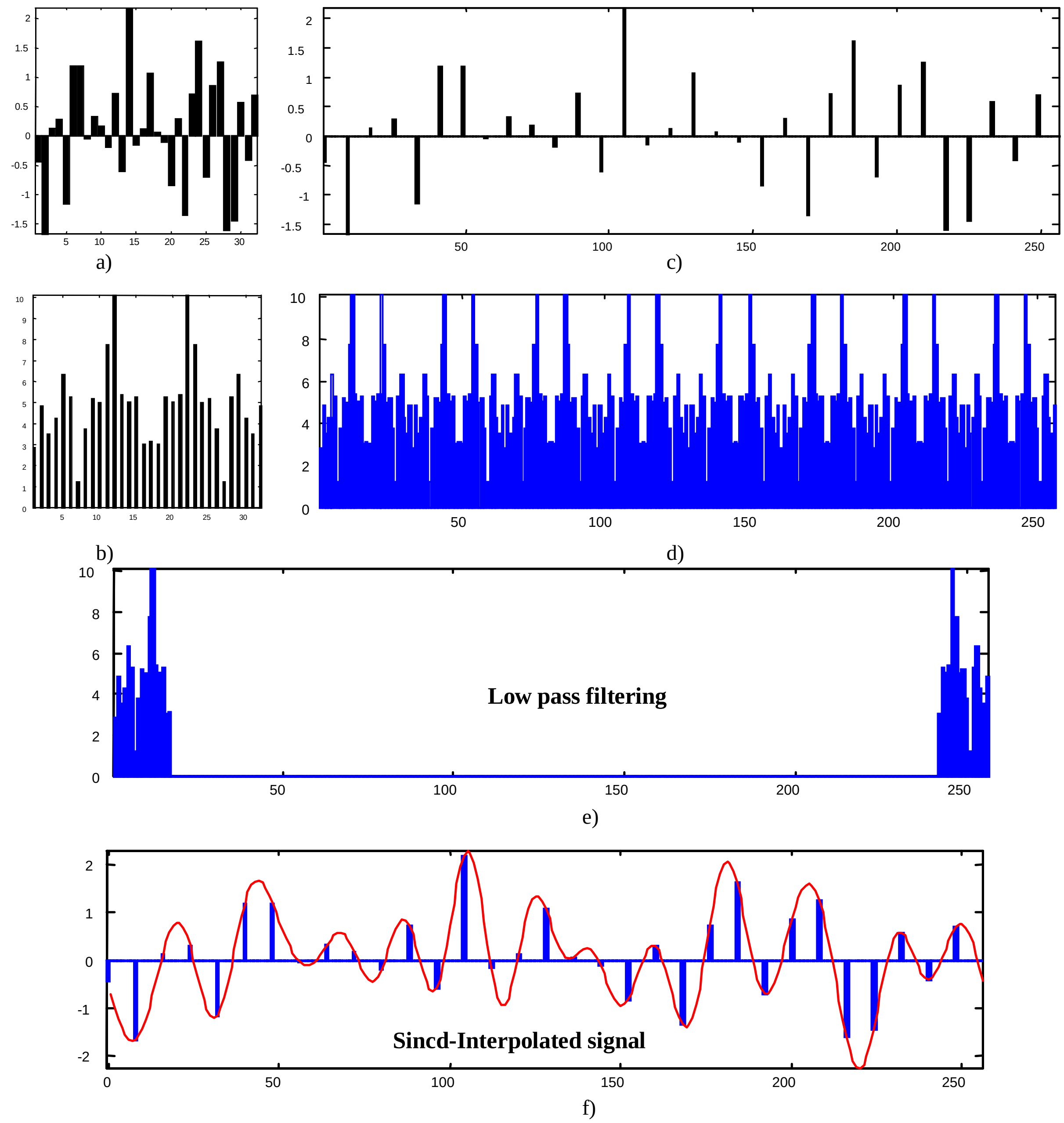


Fig. 1 Graphical illustration of discrete sampling theorem: a) initial signal; b) its spectrum; c) initial signal with zeros placed between its samples; d) its spectrum: periodical replication of the initial signal spectrum; e) removing replicas that may cause aliasing by low pass filter; f) Sinc-interpolated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

3. SDFT BASED DISCRETE SINC-INTERPOLATION ALGORITHM

A straightforward method for computer implementation of discrete sinc-interpolation that directly follows from the discrete sampling theorem is signal spectrum zero padding one[3-5]. An alternative and computationally more efficient and flexible method based on the notion of shifted DFTs (SDFT($\boldsymbol{p,q}$))[1,2]

$$\alpha_r^{p,q} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} a_k \exp\left(i2\pi \frac{kq}{N} \right) \exp\left(i2\pi \frac{(k+p)r}{N} \right); \quad \alpha_r^{p,q} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} a_k \exp\left(-i2\pi \frac{rp}{N} \right) \exp\left(i2\pi \frac{k(r+q)}{N} \right), \tag{10}$$

where p and q are arbitrary shift parameters in signal and spectral domain, respectively, was described in[6,7] .The method assumes computing each k-th set of (L-1) sets of N intermediate interpolated signal samples by inverse SDFT (Eq. 11), with correspondingly selected shift parameter $p=k/L$, of signal DFT spectrum (Eq.10, p=0;q=0) as it is illustrated in Fig. 2. Table 1 compares zero-padding and SDFT-based interpolation methods.

Table 1.

	Zero padding method	**SDFT based method**
Computational complexity of L-fold zooming signal of N samples with the use of FFT (operations per one output sample)	$O(\log NL)$	$O(\log N)$
Computational complexity of L-fold zooming signal of N samples in the vicinity of an individual sample	$O(NL \log NL)$, unless FFT pruned algorithms are used	$O(NL)$
Computational complexity (general operations) for signal shift by a fraction of the discretization interval (per one output sample)	$O(L \log NL)$, unless FFT pruned algorithms are used; shift only by (power of 2)-th fraction of the discretization interval are possible when radix 2 FFT algorithms are used.	$O(\log N)$; arbitrary shifts are possible
Zoom factor	Power of 2 for radix 2 FFT algorithms	Arbitrary
Memory usage	An intermediate buffer for NL samples is needed	No buffer is needed intermediate buffer

Yet another family of interpolation methods that has gained popularity is spline oriented, or direct convolution, interpolation one[8]. These methods assume direct convolution of the discrete signal with interpolation kernels that are build on the base of splines. Table 2 compares advantages and disadvantages of the described SDFT based and spline oriented interpolation methods.

Table 2.

	SDFT based discrete sinc interpolation	Spline oriented interpolation methods
Computational complexity (per output sample)	O(log(Size of the signal frame))	O(Interpolation function support)
Design	Simple: direct spectrum shaping in DFT domain	Requires analytical and numerical optimization
2-D interpolation	Separable & nonseparable	Separable
Interpolation accuracy	Zero interpolation error	Non-zero interpolation error: 1/ O(Interpolation function support)
Invertibility of resampling	Completely invertible	Non-invertible
Boundary effects	Severe due to cyclicity of convolution	Not substantial provided appropriate signal extension
Resampling arrangement	Only regular (equidistant) raster resampling is computationally efficient	Arbitrary resampling is possible

As one can see from the table, the main advantage of SDFT based resampling is that it does not distort initial data and is therefore completely invertible since it implements exact sinc-interpolation. Being implemented in DFT domain, SDFT based resampling easily allows signal spectrum shaping for modification of the interpolation kernel if it is required. Thanks to the use of FFT, SDFT based resampling has reasonably low computational complexity. As Table 1 shows, it is ideally suited for signal arbitrary translation that is required in many signal/image processing applications, such as, for instance, image rotation[9]. The method however has two major drawbacks: it suffers from boundary effects since it implements cyclic rather than arithmetic convolution, and it is not well suited to irregular (not equidistant) resampling. In what follows we describe two modifications of the method that overcome these drawbacks.

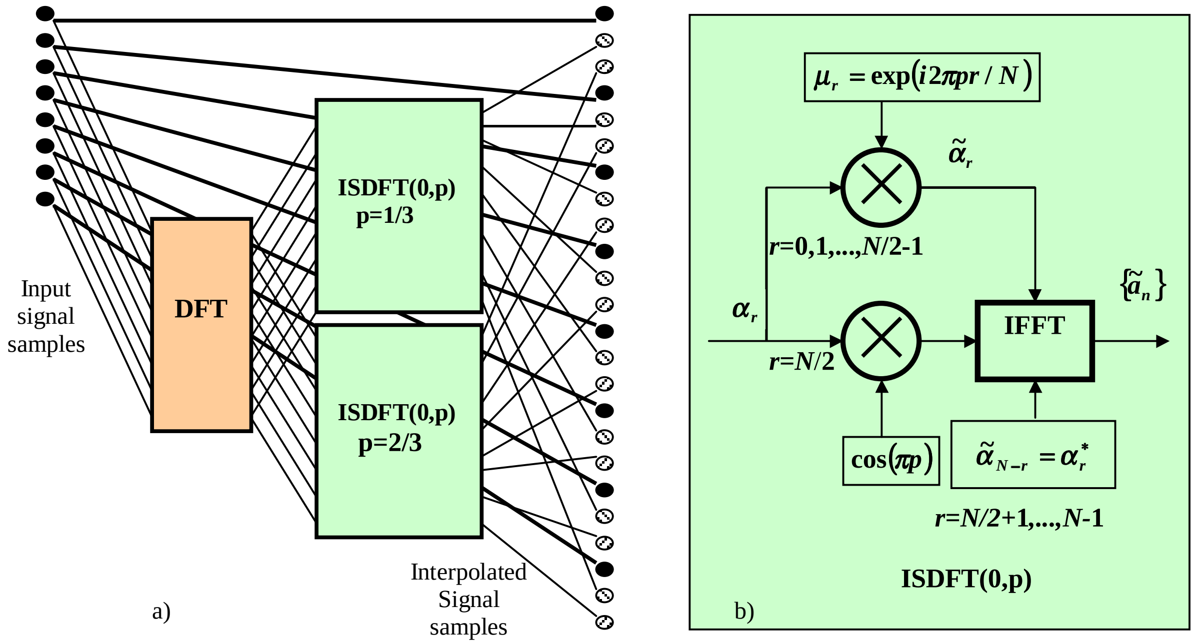


Fig. 2. Illustration of the principle (a) and flow diagram of SDFT based signal sinc-interpolation (b)

4. SINC INTERPOLATION IN DCT DOMAIN

The simplest and one of the most efficient ways to minimize boundary effects in digital filtering is signal extention by its mirror reflection from its boundaries. Such an extention complitely eliminates signal discontinuities at the boundaries.

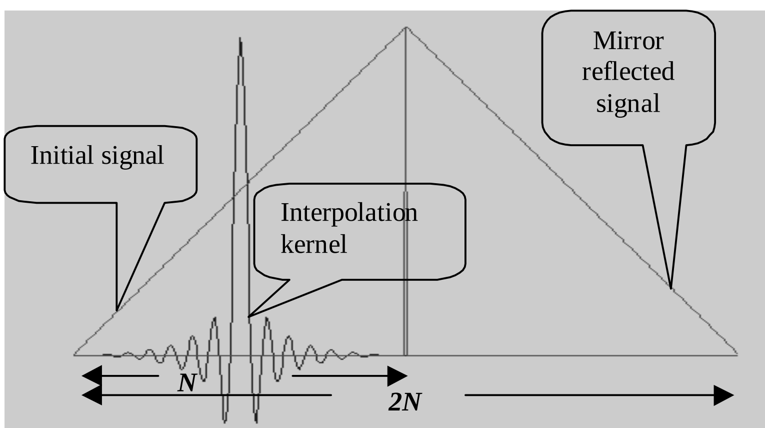


Fig. 3 Principle of sinc-interpolation with signal extension by its mirror reflection

Discrete Fourier Transform best suited to such a symmetrical signal is SDFT(1/2,0) which, in this particular case, coincides with Discrete Cosine Transform. Interpolation function for such an extended signal that has to be used to generate a ***p***-shifted sinc-interpolated copy of the initial signal should be discrete sinc-function of N samples pad by zeros to the double length of 2N samples:

$$h_k(p) = \begin{cases} \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} \varphi_s(p) \exp\left(-i2\pi \frac{ks}{N}\right); & k = 0,1,...N-1 \\ 0; & k = N, N+1,...2N-1 \end{cases} \tag{11}$$

where

$$\varphi_s(p) = \begin{cases} \exp(i2\pi ps/N); & s = 0,1,\ldots,N/2-1 \\ \cos(2\pi ps/N); & s = N/2 \\ \varphi^*_{N-s}; & s = N/2+1,\ldots,N-1 \end{cases} \tag{12}$$

One can show that inverse SDFT(1/2,0) for generating the interpolated signal is reduced to

$$b_k = \frac{1}{\sqrt{2N}} \sum_{r=0}^{2N-1} \beta_r \exp\left(-i2\pi \frac{(k+1/2)r}{2N}\right) = \frac{1}{\sqrt{2N}} \sum_{r=0}^{2N-1} \alpha_r \eta_r(p) \exp\left(-i2\pi \frac{(k+1/2)r}{2N}\right) =$$
$$\left\{\alpha_0^{DCT}\eta_0 + \sum_{r=1}^{N-1} \alpha_r^{DCT}\left[\eta_r \exp\left(-i\pi \frac{(k+1/2)}{N} r\right) + \eta_r^* \exp\left(i\pi \frac{(k+1/2)}{N} r\right)\right]\right\}, \tag{13}$$

where

$$\alpha_r = \begin{cases} \alpha_r^{DCT} = DCT\{a_k\}, & r = 0,1,\ldots,N-1; \\ 0, & r = N \\ -\alpha_{2N-1-r}^{DCT}, & r = N+1, N+2,\ldots,2N-1; \end{cases} \tag{14}$$

$$\eta_r(p) = \frac{1}{\sqrt{2N}} \sum_{k=0}^{2N-1} h_k(p) \exp\left(i2\pi \frac{kr}{2N}\right). \tag{15}$$

Note that one needs to compute only terms of $\{\eta_r(p)\}$ with odd indices:

$$\eta_{2r+1}(p) = \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1}\sum_{s=0}^{N-1} \varphi_s(p) \exp\left(-i2\pi \frac{ks}{N}\right) \exp\left(i2\pi \frac{k(2r+1)}{2N}\right) \tag{16}$$

since terms with even indices can be found directly from the definition of $\{\eta_r\}$ (Eqs. 13, 17):

$$\eta_{2r}(p) = \exp\left(i2\pi \frac{pr}{N}\right). \tag{17}$$

Flow diagram of the algorithm for generating ***p***-shifted sinc-interpolated copy of signal is shown in Fig. 4.

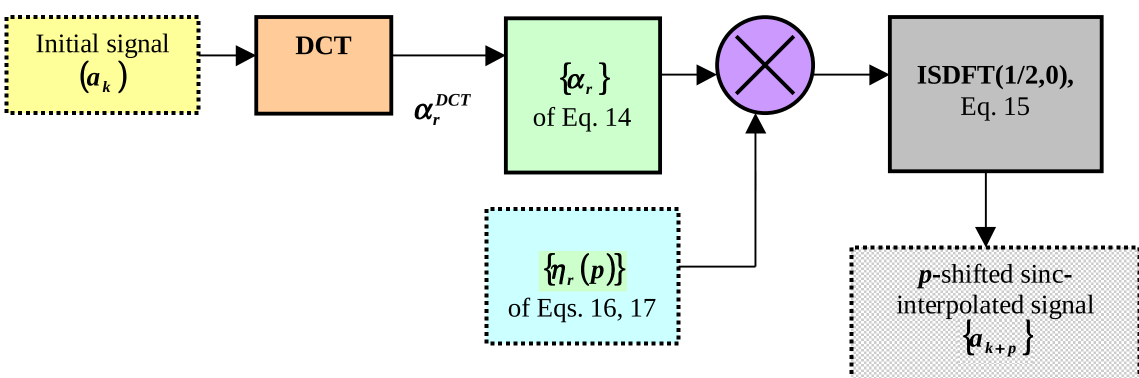
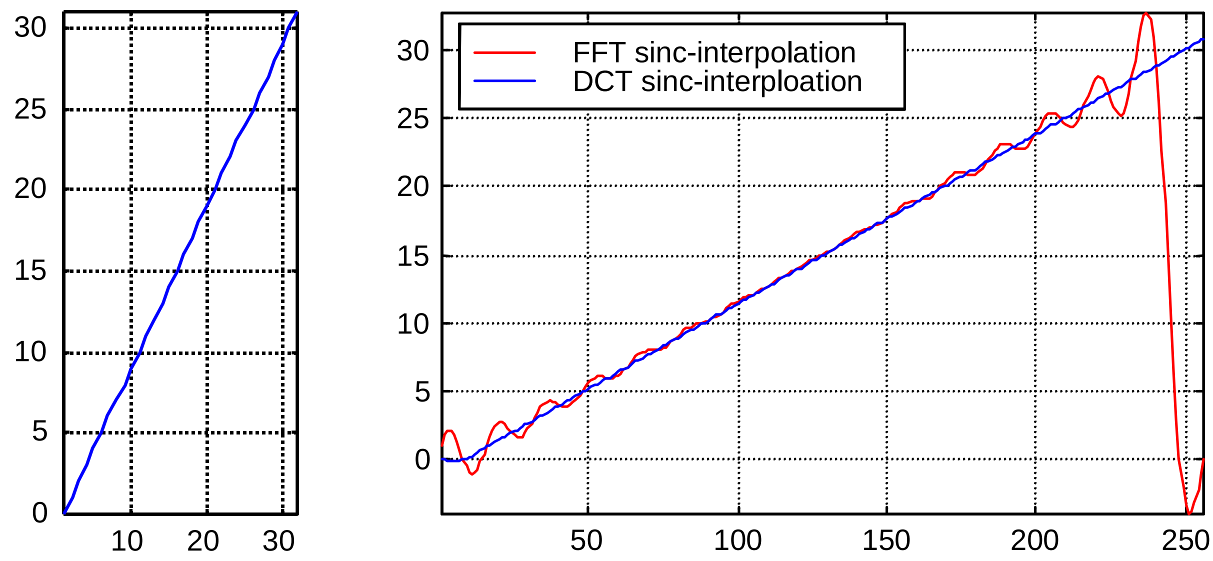


Fig. 4. Flow diagram of sinc-interpolation in DCT domain

Fig. 5 demonstrates that sinc interpolation in DCT domain is allmost completely free of oscillating Gibbs effects that are characteristic for sinc-interpolation in with DFT due to discontinuities on the signal borders.



a) b)

Fig. 5. $8\times$- signal zooming with sinc-interpolation: a) – initial “ramp” signal; b) – interpolated signals obtained by sinc-interpolation with DFT (oscillating curve) and with DCT (curve without oscillations).

5. SINC INTERPOLATION IN SLIDING WINDOW

Shifted DFT based sinc-interpolation algorithm described in Sect. 3 can be implemented in sliding window in which case, in each window position, only interpolated samples in the vicinity of the window central pixel are generated. Due to this fact, the method is free of boundary effects. However, it can not implement exact sinc-interpolation sinc interpolation function in this case is windowed discrete sinc-function, that is discrete sinc-function whose extent is equal to the window size. Fig. 6 illustrates frequency response of the corresponding low pass filter. The computational complexity of the interpolation in sliding window is ***O***(***WindowSize***) per output signal sample thanks to the existing recursive algorithm for computing signal DFT spectrum in sliding window[1,2].

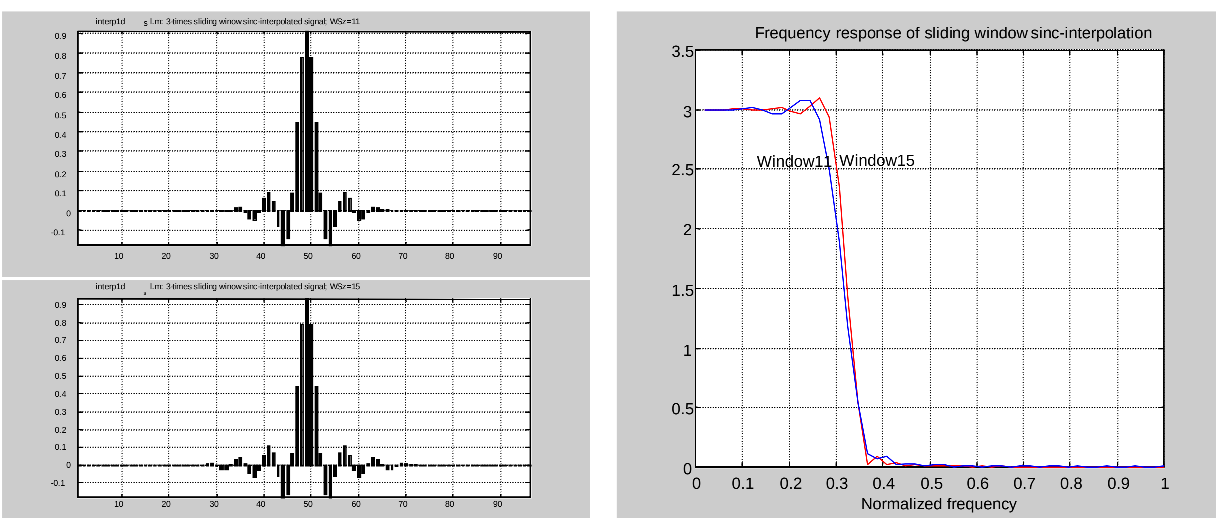


Fig. 6 Windowed discrete sinc-functions with window size 11 and 15 samples (left) and their DFT spectra (right).

Such an implementation of sinc interpolation can be regarded as a variety of direct convolution interpolation method. If applied directly, it shares all features of direct convolution interpolation methods such as the possibility of arbitrary, not necessarily regular (equidistant), arrangement of signal samples and absence of Gibbs effects but it has no special advantages before other direct convolution interpolation methods such as spline oriented one except for the simplicity of optimization of the interpolation kernel by shaping of its spectrum thanks to working in DFT domain. However, with an appropriate modification, SDFT based sinc-interpolation in sliding window offers features that are not available in other methods. These are signal resampling in an arbitrary raster with simultaneous signal denoising and local adaptive signal interpolation with "super resolution".

5.1 Signal sinc-interpolation and denoising by filtering in sliding window

Local adaptive filters that work in sliding window in transform domain have shown their high potentials in signal and image restoration[10,11]. One can, in a straightforward way, extended it to sliding window sinc-interpolation if signal spectrum denoising by empirical Wiener filter or by soft or hard thresholding in DFT domain is complemented with introducing signal ***p***-shift as it is shown in flow diagram in Fig. 6. Note that inverse SDFT of the signal modified window spectrum for reconstruction of only window central sample is reduced to simple summation of the modified spectral coefficients. Fig. 7 illustrates application of such combined filtering/interpolation for image irregular-to regular sampling raster resampling and denoising.

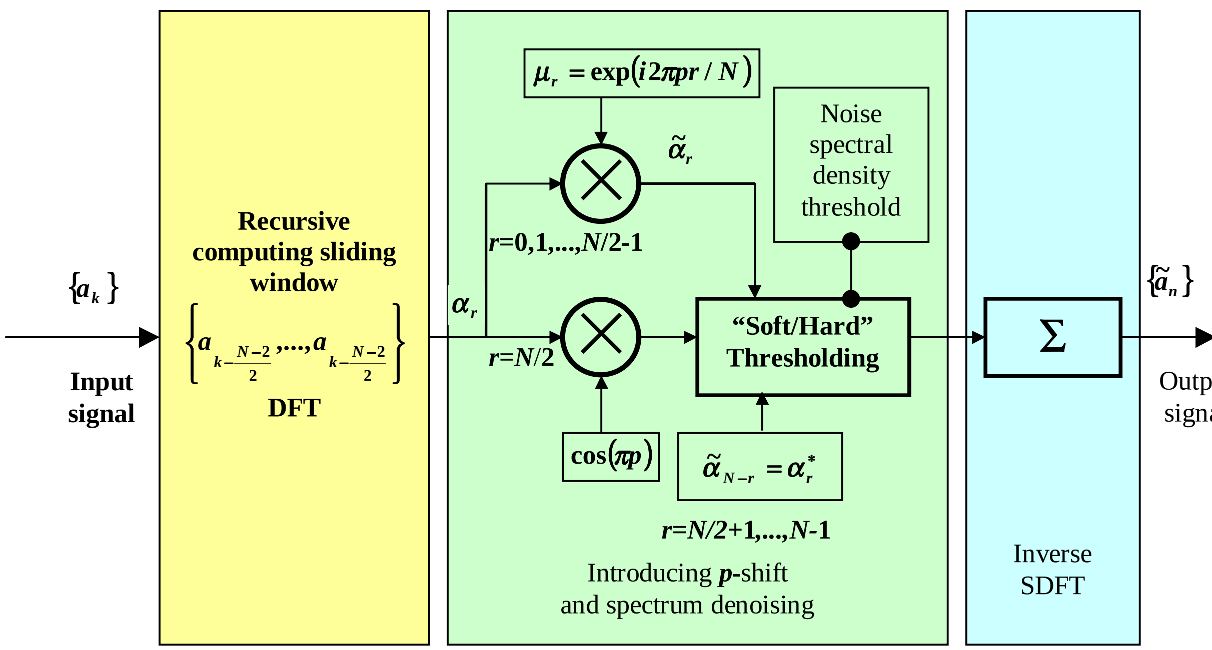


Fig. 7. Flow diagram of simultaneous the signal sliding window sinc-interpolation and denoising

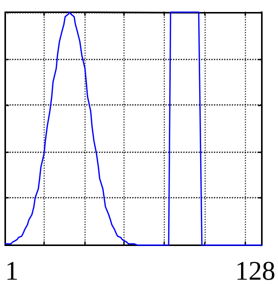
5.2 Adaptive interpolation in sliding window

Method of signal sinc-interpolation and denoising by filtering in sliding window can be further extended to interpolation adaptive to signal local features that exibit themselves in signal local spectra. For instance, energy of local spectra high frequency components can be used as an indicator of the presence or absence of sharp edges in the signal and serve as a control signal for modifying local interpolation kernel. Fig. 7 illustrates this idea on the example of zooming a signal composed of Gaussian shaped and rectangle impulses. Sinc-interpolation of such a signal results in severe oscillations at the borders of the rectangle impulse (Fig. 7b). If, however, it is known a priori that signal may

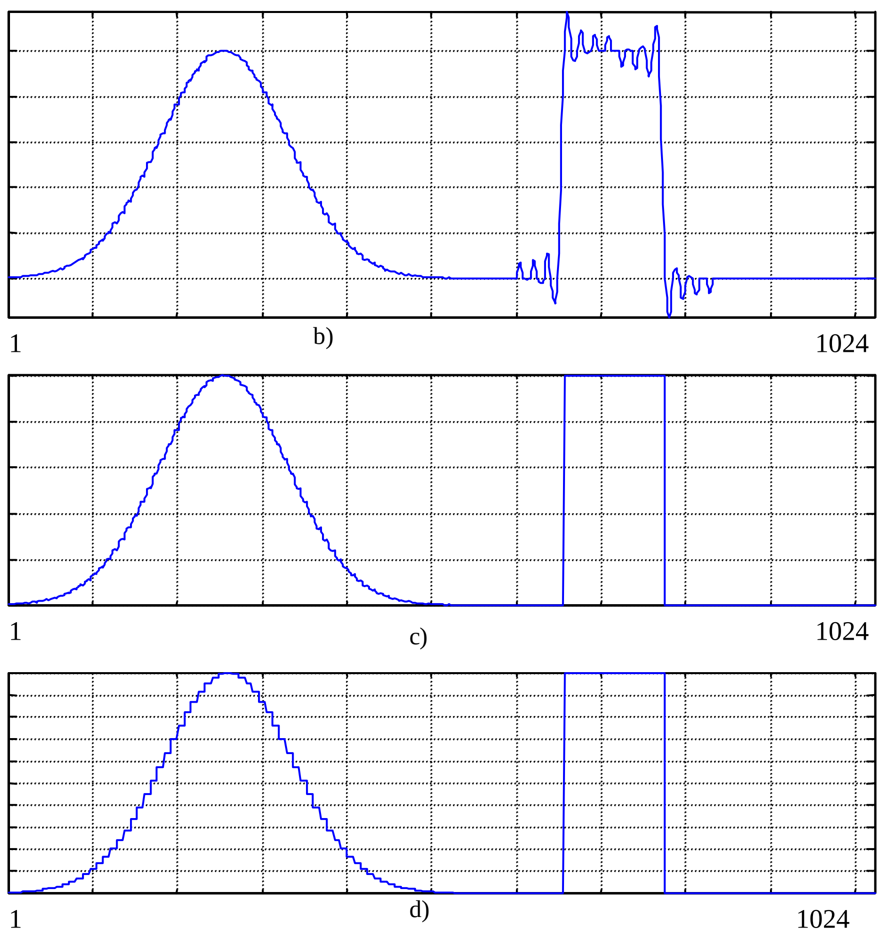
Noisy image sampled in a irregular raster

Denoised and resampled (rectified) image

Fig. 8. Image rectification and denoising by resampling with sinc-interpolation in sliding window.



a)



b)

c)

d)

Fig. 7. Comparison of non adaptive and local adaptive interpolation for signal zooming: a) – initial signal; b) - non adaptive sinc-interpolation is sliding window (Window Size=15 samples) ; c) – local adaptive interpolation with switching between two interpolation modes, sinc-interpolation and nearest neighbor interpolation; d) – zooming by nearest neighbor interpolation

contain edges with high frequencies outside the bandwidth defined by the discretization interval, interpolation should be capable of “super resolution” by keeping sharpness of such edges and not producing artifacts in form of oscillations. This can be achieved if, for instance, use nearest neighbor or similar interpolation in the vicinity of signal sharp edges rather than sinc-interpolation, most appropriate for smooth signals. In order to switch between these two modes of interpolation, energy of high frequency components of signal local spectrum can be used as it is illustrated in Fig. 7c. For comparison, regular nearest neighbor interpolation for signal zooming is shown in Fig. 7d.
.

ACKNOWLEDGMENT

The work was partially carried out at Tampere International Center for Signal Processing, Tampere University of Technology, Tampere, Finland

REFERENCES

1. L. Yaroslavsky, *Digital Picture Processing. An Introduction,* Springer Verlag, p. 95, Berlin, Heidelberg, New York, Tokyo, 1985.
2. L. Yaroslavsky, M. Eden, *Fundamentals of Digital Optics*, p. 85, Birkhauser, Boston, 1996
3. L. R. Rabiner, B. Gold, *Theory and application of Digital Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1975
4. D. Fraser, “Interpolation by the FFT Revisited – An Experimental Investigation”, *IEEE Trans.* ,**ASSP- 37,** pp. 665-675, 1989
5. T. Smith, M.S. Smith, and S.T. Nichols, “Efficient sinc function interpolation technique for center padded data”, *IEEE Trans.* , **ASSP- 38**, pp. 1512-1517, 1990
6. L. Yaroslavsky, “Signal sinc-interpolation: a fast computer algorithm”, *Bioimaging*, **4**, pp. 225-231, 1996
7. L.P. Yaroslavsky, “Efficient algorithm for discrete sinc-interpolation”, *Applied Optics*, **36**, p. 460-463, 1997
8. Ph. Thevenaz, Thierry Blu, M. Unser, Interpolation Revisited, *IEEE Trans.*, MI-19, pp. 758, 2000
9. M. Unser, P. Thevenaz, L. Yaroslavsky, Convolution-based Interpolation for Fast, High-Quality Rotation of Images, *IEEE Trans. on Image Processing*, Oct. 1995, v. 4, No. 10, p. 1371-1382
10. L. Yaroslavsky, “Image Restoration, enhancement and target location with local adaptive filters”, *International Trends in Optics and Photonics, ICOIV*, T. Asakura, ed., pp. 111-127, Springer Verlag, 1999,
11. L.P. Yaroslavsky, K.O. Egiazarian, J.T. Astola, “Transform domain image restoration methods: review, comparison and interpretation”, *Photonics West, Conference 4304, Nonlinear Processing and Pattern Analysis*, 22-23 January, 2001, *Proceedings of SPIE*, v. 4304.