

**L. Yaroslavsky**

**FROM PHOTOGRAPHY TO \*.GRAPHIES:  
UNCONVENTIONAL IMAGING TECHNIQUES**

**A short course at Tampere University of Technology,  
Tampere, Finland, Sept. 3 – Sept. 14, 2001**

**Lecture 6.**

**SINC-INTERPOLATION IN DIGITAL IMAGING**

## Lecture 6

### SIGNAL FAST SINC-INTERPOLATION BY MEANS OF FFT

#### Digital signal/image geometrical transformations and signal resampling

The principle of image geometrical transformation is illustrated in Fig. From this figure one can see that for the transformation digital signal/image resampling (geometrical transformations) requires signal subsampling and interpolation.

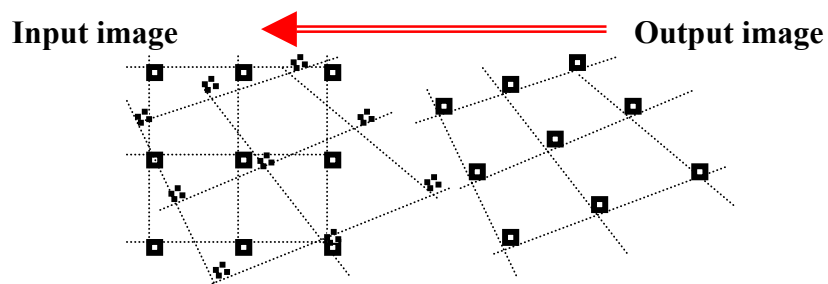


Fig. 1. The principle of image geometrical transformations:  
backward co-ordinate mapping

Illustrative examples of image geometrical transformations are given in Fig. 2.

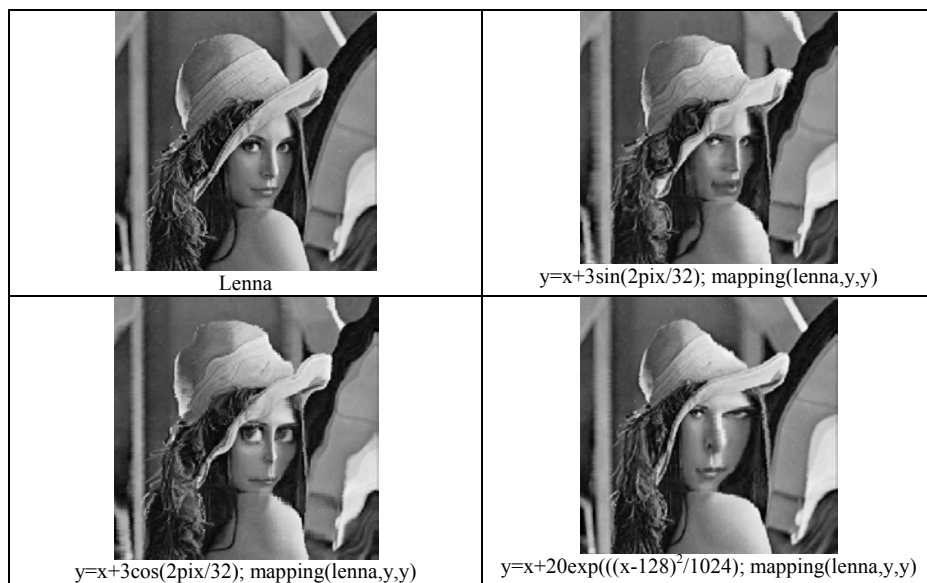


Fig. 2 Examples of image geometrical transformations

**Signal interpolation computational methods:**

- Nearest neighbor (zero order) interpolation:

$$\bar{a}_{int}^{(0)} = \text{conv}[\text{kron}(\bar{a}, \delta_L), \text{ones}(L)]; \quad (1)$$

$$\delta(L) = \underset{\leftarrow L \rightarrow}{[1, 0, 0, \dots, 0]} \quad (2)$$

- Linear (bilinear) interpolation:

$$\bar{a}_{int}^{(1)} = \text{conv}[\text{conv}[\text{kron}(\bar{a}, \delta_L), \text{ones}(L)], \text{ones}(L)]. \quad (3)$$

- $R$ -th order spline interpolation:

$$\bar{a}_{int}^{(R)} = \text{conv}[\text{conv} \cdots [\text{conv}[\text{kron}(\bar{a}, \delta_L), \text{ones}(L)], \text{ones}(L)]]]. \quad (4)$$

- Sinc-interpolation:

$$a(x) = \sum_{k=-\infty}^{\infty} \alpha_k \text{sinc}[\pi(x - k\Delta x) / \Delta x] \quad (5)$$

Two-dimensional interpolation is usually implemented as separable row-column interpolation.

Fig. 3 compares three methods of image interpolation (zooming): nearest neighbor, bilinear (2-D 1-st order spline), bicubic (2-D second order spline) and sinc-interpolation.

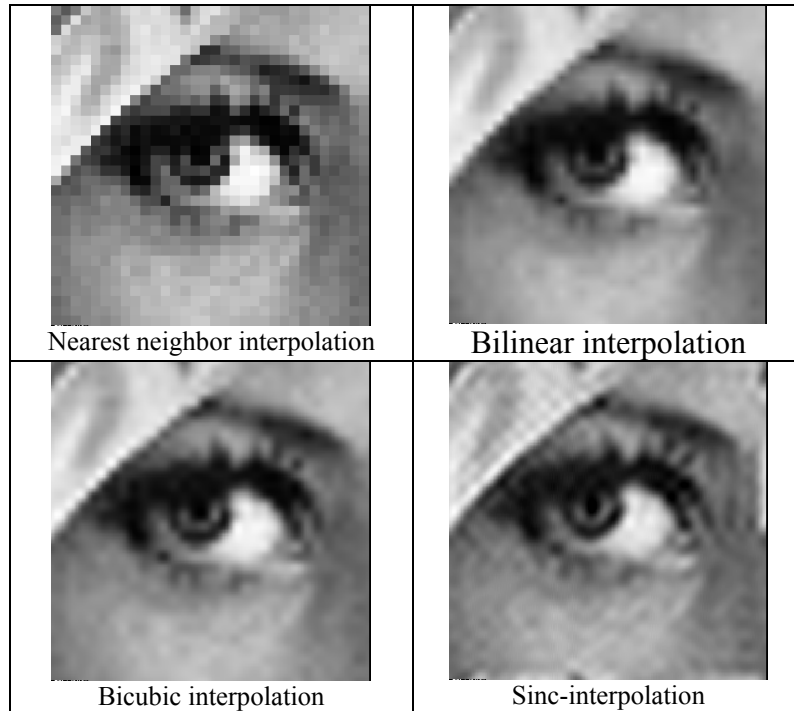


Fig. 3. 8x image zoom: comparison of interpolation methods.

**Sinc-interpolation is a gold standard for linear interpolation of sampled data.**

Proof for continuous signals:

Signal sampling assumes their representation over shifted bases:

$$a(x) = \sum_{k=-\infty}^{\infty} \alpha_k \varphi_{reconstr}(x - k\Delta x) = \sum_{k=-\infty}^{\infty} \alpha_k \delta(x - k\Delta x) \otimes \varphi_{reconstr}(x)$$

$\Downarrow$  *Fourier Transform*

$$\alpha(f) = \left\{ \sum_{k=-\infty}^{\infty} \alpha_k \exp(i2\pi k f \Delta x) \right\} \Phi_{reconstr}(f) = \tilde{\alpha}(f) \Phi_{reconstr}(f) \quad (5)$$

$\tilde{\alpha}(f)$  is a periodic function with period  $1/\Delta x$ :

$$\tilde{\alpha}(f) = \sum_{m=-\infty}^{\infty} \tilde{\alpha}(f - m/\Delta x), \quad (6)$$

where

$$\tilde{\alpha}(f) = \frac{\alpha(f)}{\Phi_r(f)}, \quad f \in [-1/2\Delta x, 1/2\Delta x]. \quad (7)$$

Ideal case, when complete separation of replicated copies of signal spectrum is possible:

$$\Phi_{reconstr}(f) = \text{rect}[(f + 1/2\Delta x)\Delta x], \quad (8)$$

which means signal sinc-interpolation

$$a(x) = \sum_{k=-\infty}^{\infty} \alpha_k \text{sinc}[\pi(x - k\Delta x)/\Delta x]. \quad (9)$$

Proof for discrete signals (Discrete Sampling Theorem):

Let a signal of  $LN$  samples is sampled with sampling interval of  $L$  samples. For  $k_1 = 0, \dots, N-1$ ;  $k_2 = 0, \dots, L-1$  the sampled signal can be written as  $a_{k_1} \delta(k_2)$ .

Compute its DFT:

$$\begin{aligned} a_{k_1} \delta(k_2) &\xrightarrow{\text{DFT}} \frac{1}{\sqrt{LN}} \sum_{k_2} \sum_{k_1=0}^{N-1} a_{k_1} \delta(k_2) \exp\left[i2\pi \frac{(k_1 L + k_2)}{LN} r\right] = \\ &\frac{1}{\sqrt{LN}} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{L-1} a_{k_1} \delta(k_2) \exp\left[i2\pi \frac{(k_1 L + k_2)}{LN} r\right] = \\ &\frac{1}{\sqrt{LN}} \sum_{k_1=0}^{N-1} a_{k_1} \exp\left[i2\pi \frac{k_1}{N} r\right] = \frac{1}{\sqrt{L}} \alpha_{(r) \bmod N} \end{aligned} \quad (10)$$

This means that sampling discrete signal results in periodical replication of its spectrum with the number of replicas equal to sampling interval.

For reconstruction of the initial signal one should remove all spectral replicas but one. One can do this by discrete low pass filtering. Its implementation depends on whether  $N$  is even or odd number since when removing extra spectral replicas one should keep spectrum symmetry property that reads that for real valued signals spectral samples symmetrical with respect to its end are complex conjugate:  $\alpha_r = \alpha_{LN-r}^*$ .

For even  $N$ , there are two options:

1. Low pass filtering (spectrum zero padding) with discarding signal spectrum sample  $\alpha_{N/2}$ :

$$\begin{aligned}
& \left[ 1 - \text{rect} \frac{r - N/2}{LN - N} \right] \alpha_{(r) \bmod N} \xleftrightarrow{IDFT} = \\
& \frac{1}{\sqrt{LN}} \sum_{r=0}^{LN-1} \left[ 1 - \text{rect} \frac{r - N/2}{LN - N} \right] \alpha_{(r) \bmod N} \exp\left(-i2\pi \frac{kr}{LN}\right) = \\
& \frac{1}{\sqrt{LN}} \left\{ \sum_{r=0}^{N/2-1} \alpha_{(r) \bmod N} \exp\left(-i2\pi \frac{kr}{LN}\right) + \sum_{r=LN-N/2+1}^{LN-1} \alpha_{(r) \bmod N} \exp\left(-i2\pi \frac{kr}{LN}\right) \right\} \\
& \frac{1}{\sqrt{LN}} \left\{ \sum_{r=0}^{N/2-1} \left[ \frac{1}{\sqrt{N}} \sum_{k_1=0}^{N-1} a_{k_1} \exp\left(i2\pi \frac{k_1 r}{N}\right) \right] \exp\left(-i2\pi \frac{kr}{LN}\right) + \right. \\
& \left. \sum_{r=LN-N/2+1}^{LN-1} \left[ \frac{1}{\sqrt{N}} \sum_{k_1=0}^{N-1} a_{k_1} \exp\left(i2\pi \frac{k_1 r}{N}\right) \right] \exp\left(-i2\pi \frac{kr}{LN}\right) \right\} = \\
& \frac{1}{N\sqrt{L}} \sum_{k_1=0}^{N-1} a_{k_1} \left[ \sum_{r=0}^{N/2-1} \exp\left(i2\pi \frac{(Lk_1 - k)r}{LN}\right) + \sum_{r=LN-N/2+1}^{LN-1} \exp\left(i2\pi \frac{(Lk_1 - k)r}{LN}\right) \right] = \\
& \frac{1}{N\sqrt{L}} \sum_{k_1=0}^{N-1} a_{k_1} \times \\
& \left[ \frac{\exp\left(i2\pi \frac{(Lk_1 - k)}{2L}\right) - 1 + \exp\left(i2\pi(Lk_1 - k)\right) - \exp\left(i2\pi \frac{(Lk_1 - k)(LN - N/2 + 1)}{NL}\right)}{\exp\left(i2\pi \frac{(Lk_1 - k)}{LN}\right) - 1} \right] \\
& = \frac{1}{N\sqrt{L}} \sum_{k_1=0}^{N-1} a_{k_1} \frac{\exp\left(i2\pi \frac{(Lk_1 - k)}{2L}\right) - \exp\left(i2\pi \frac{(Lk_1 - k)(1 - N/2)}{NL}\right)}{\exp\left(i2\pi \frac{(Lk_1 - k)}{LN}\right) - 1} = \\
& \frac{1}{N\sqrt{L}} \sum_{k_1=0}^{N-1} a_{k_1} \frac{\exp\left(i\pi \frac{N-1}{N} \frac{(Lk_1 - k)}{L}\right) - \exp\left(-i\pi \frac{N-1}{N} \frac{(Lk_1 - k)}{L}\right)}{\exp\left(i\pi \frac{(Lk_1 - k)}{LN}\right) - \exp\left(-i\pi \frac{(Lk_1 - k)}{LN}\right)} =
\end{aligned}$$

$$= \frac{1}{\sqrt{L}} \sum_{k_1=0}^{N-1} a_{k_1} \frac{\sin\left[\pi \frac{N-1}{N} (k - k_1 L)\right]}{N \sin[\pi(k - k_1 L)/LN]} = \frac{1}{\sqrt{L}} \sum_{k_1=0}^{N-1} a_{k_1} \text{sincd}(N-1; N; (k - k_1 L)), \quad (11)$$

where

$$\text{sincd}(K; N; x) = \frac{\sin(\pi K x / N)}{N \sin(\pi x / N)} \quad (12)$$

is a discrete sinc-function, a discrete analog of continuous sinc-function  $\text{sinc}(x) = \sin x / x$

2. Low pass filtering (spectrum zero padding) with doubling signal spectrum sample  $\alpha_{N/2}$ :

Similarly to the previous case one can show that

$$\left[1 - \text{rect} \frac{r - N/2 + 1}{LN - N}\right] \alpha_{(r) \bmod N} \xleftrightarrow{\text{IDFT}} = \frac{1}{\sqrt{L}} \sum_{k_1=0}^{N-1} a_n \text{sincd}(N+1; N; (k - k_1 L)) \quad (13)$$

For odd  $N$ , there is no symmetry problem and one can obtain:

$$\left[1 - \text{rect} \frac{r - (N+1)/2}{LN - N - 1}\right] \alpha_{(r) \bmod N} \xleftrightarrow{\text{IDFT}} = \frac{1}{\sqrt{L}} \sum_{k_1=0}^{N-1} a_n \text{sincd}(N; N; (k - k_1 L)). \quad (14)$$

### Computational methods of sinc-interpolation:

#### Sinc-interpolation by zero padding the DFT spectrum.

This method is based on the above sampling theorem for DFT. It is illustrated in Fig. 4.

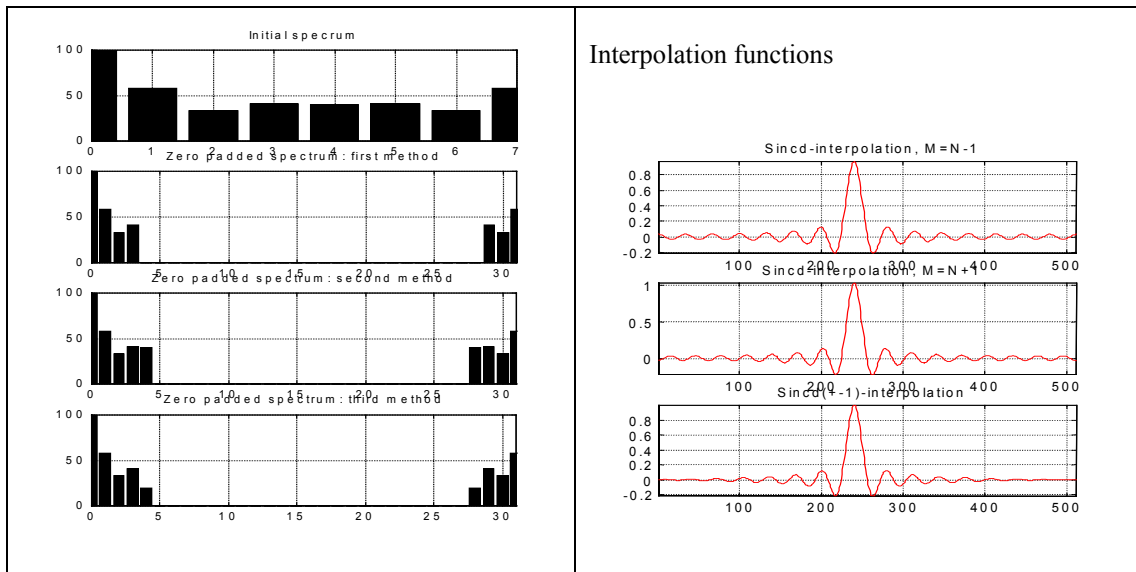


Fig. 4. Zero-padding interpolation method and interpolation functions for an even number of signal samples.

Upper plot from the left on Fig. 4 shows spectrum of a to be interpolated signal. Second from the top to the left is plot of the zero padded spectrum with discarding central sample  $\alpha_{N/2}$  of the initial spectrum. To the right from it the corresponding interpolation function is shown. Next plots illustrate zero padding with replication of the initial spectrum central sample  $\alpha_{N/2}$ . The bottom plots illustrate combination of the above two methods when central sample  $\alpha_{N/2}$  of the initial spectrum is halved before its replication. Such a combination results in the interpolation function

$$\text{sincd}(\pm 1; N; x) = [\text{sincd}(N-1; N; x) + \text{sincd}(N+1; N; x)]/2 \quad (15)$$

that features much faster decay to zero with rise of the interpolation distance and can be recommended as a good practical compromise between the need do not introduce too much signal distortions and do not have too much Gibbs' s ripples in the interpolated signal.

#### Fast sinc-interpolation algorithm: sinc-interpolation by SDFTs

Shifted direct and inverse discrete Fourier Transforms (SDFT) of a signal  $\{a_k\}$  of  $N$  samples are defined as (see Appendix):

$$\alpha_r^{u,v} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} a_k \exp\left(i2\pi \frac{kv}{N}\right) \exp\left(i2\pi \frac{(k+u)r}{N}\right) \quad (16)$$

$$\alpha_r^{u,v} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} a_k \exp\left(-i2\pi \frac{ru}{N}\right) \exp\left(i2\pi \frac{k(r+v)}{N}\right) \quad (17)$$

with  $u$  and  $v$  as signal and its spectrum sample shift parameters.

Computing first SDFT with certain shift parameters  $(u, v)$  of a signal and then making inverse SDFT with shift parameters  $(p, q)$  one can obtain a sinc-interpolated copy of the signal shifted with respect to the initial signal by  $(u - p)$  (see Lecture 5):

$$\begin{aligned} \tilde{a}_n^{u/p, v/q} &= \frac{1}{\sqrt{N}} \sum_{r=0}^{K-1} \left\{ \alpha_r^{u,v} \exp\left(-i2\pi \frac{rp}{N}\right) \right\} \exp\left(-i2\pi \frac{n(r+q)}{N}\right) = \\ & \left\{ \sum_{k=0}^{N-1} a_k \exp\left(i\pi k \left(\frac{K-1}{N} + 2v\right)\right) \text{sincd}(K; N; (k - n + u - p)) \right\} \times \end{aligned}$$

$$\exp\left(-i\pi\left(\frac{K-1}{N} + 2q\right)n\right)\exp\left(i\pi\frac{K-1}{N}(u-p)\right). \quad (18)$$

For zero initial shift parameters ( $u = 0$  and  $v = 0$ ) and  $q = 0$  obtain:

$$\begin{aligned} \tilde{a}_n^{0/p,0/0} &= \frac{1}{\sqrt{N}} \sum_{r=0}^{K-1} \left\{ \alpha_r^{u,v} \exp\left(-i2\pi\frac{rp}{N}\right) \right\} \exp\left(-i2\pi\frac{nr}{N}\right) = \\ & \left\{ \sum_{k=0}^{N-1} a_k \exp\left(i\pi\frac{K-1}{N}k\right) \text{sincd}(K; N; (k-n-p)) \right\} \exp\left(-i\pi\frac{K-1}{N}(n+p)\right). \end{aligned} \quad (19)$$

In this way we arrive at an algorithm for sinc-interpolation shown in Fig. 5.

In practice, however, one does not need to implement this general case. For initial shift parameters  $\{u = 0; v = 0\}$ , a substantial simplification is possible that can be derived from the observation that the interpolation formulas (11 – 13) are digital convolutions that can be implemented in DFT domain by multiplying signal spectrum by DFT of the interpolation functions and computing inverse DFT. One can show (see Appendix 3) that the following relationships hold:

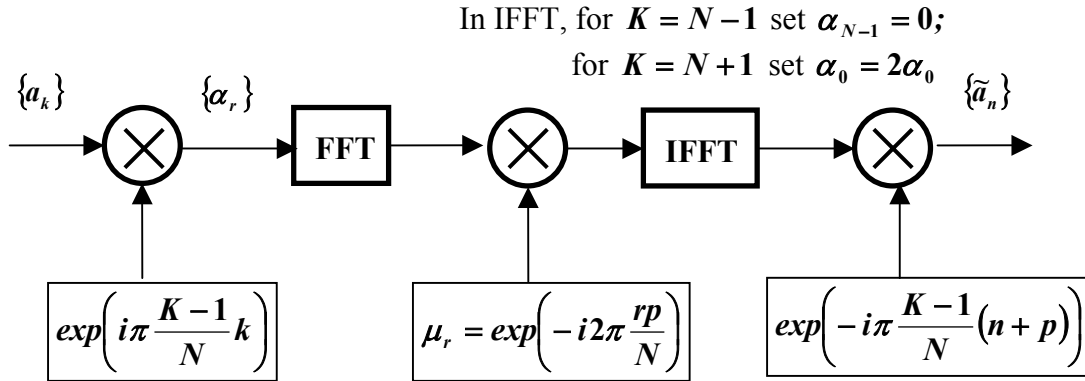


Fig. 5. An algorithm for fast discrete signal sinc-interpolation

$$\text{sincd}(N-1; N; (k-u)) \xleftrightarrow{DFT} \varphi_r^{(0)} = \begin{cases} \exp(i2\pi ur / N), & r = 0, 1, \dots, N/2 - 1 \\ 0, & r = N/2 \\ \varphi_{N-r}^*, & r = N/2 + 1, \dots, N-1 \end{cases}; \quad (20)$$

$$\text{sincd}(N+1; N; (k-u)) \xleftrightarrow{DFT} \varphi_r^{(2)} = \begin{cases} \exp(i2\pi ur / N), & r = 0, 1, \dots, N/2 - 1 \\ 2 \cos(\pi u), & r = N/2 \\ \varphi_{N-r}^*, & r = N/2 + 1, \dots, N-1 \end{cases}; \quad (21)$$

$$\text{sincd}(\pm 1; N; (k - u)) \xleftrightarrow{DFT} \varphi_r^{(0)} = \begin{cases} \exp(i2\pi r / N), & r = 0, 1, \dots, N/2 - 1 \\ \cos(\pi r), & r = N/2 \\ \varphi_{N-r}^*, & r = N/2 + 1, \dots, N - 1 \end{cases} \quad (22)$$

which imply a sinc-interpolation algorithm shown in Fig. 6.

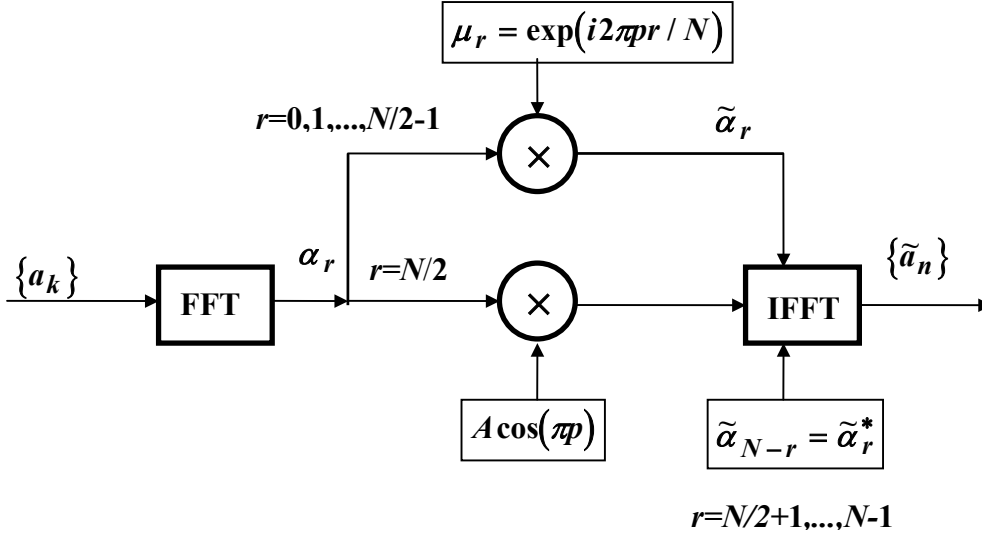


Fig. 6 A simplified algorithm for fast discrete signal sinc-interpolation

### Comparison of zero-padding and SDFT based sinc-interpolation algorithms.

Table compares the two methods for signal sinc-interpolation in terms of their computational complexity and flexibility.

	Zero padding method	SDFT based method
Computational complexity (general operations) of $L$ -fold zooming signal of $N$ samples with the use of FFT	$O(NL \log NL)$	$O(NL \log N)$
Computational complexity (general operations) of $L$ -fold zooming signal of $N$ samples in the vicinity of an individual sample (as, for instance, in locating position of signal maximum with subpixel accuracy)	$O(NL \log NL)$ , unless FFT pruned algorithms are used	$O(NL)$
Computational complexity (general operations) for signal shift by a fraction of the discretization interval	$O(NL \log NL)$ , unless FFT pruned algorithms are used; shift only by (power of 2)-th fraction of the discretization interval are possible when the most wide spread FFT algorithms are used.	$O(N \log N)$ ; arbitrary shifts are possible
Zoom factor	Power of 2 for the most widely used FFT algorithms	Arbitrary
Memory usage	Requires an intermediate buffer for $NL$ samples	Does not need an intermediate buffer

### Image rotation with sinc-interpolation: a three-pass rotation algorithm

Image rotation by an angle  $\theta$  as a geometrical transformation of signal coordinates can be described as a multiplication of signal coordinates  $(x, y)$  vector by a rotation matrix:

$$ROT(\theta) \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (23)$$

In order to simplify computation, one can factorize rotation matrix  $ROT(\theta)$  into a product of three matrices each of which modifies only one co-ordinate:

$$ROT(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} 1 & -\tan(\theta/2) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin \theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(\theta/2) \\ 0 & 1 \end{bmatrix} \quad (24)$$

( X-shift      Y-shift      X-shift).  
 $\quad \quad \quad \blacktriangledown \quad \quad \quad \blacktriangledown \quad \quad \quad \blacktriangledown$

This implementation of image rotation is known as a three pass rotation algorithm. It assumes, on each pass, only signal shift along one of the co-ordinates: along  $x$  on the first pass, along  $y$  on the second pass and again along  $x$  on the third pass as it is shown in Eq. (24). Each shift can be very efficiently implemented with the complexity of  $O(\log N)$  operations per pixel ( $N$  being image size). The work of the algorithm is illustrated in Fig. 7.

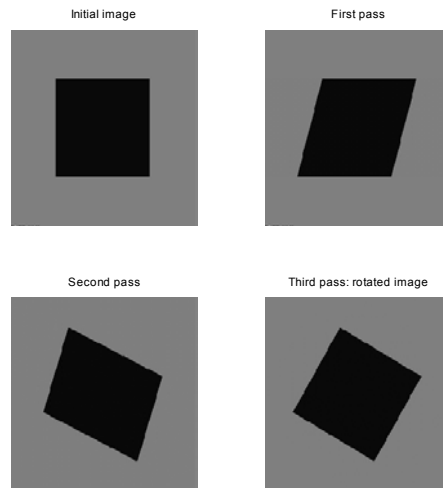


Fig. 7. Three passes of the image rotation algorithm

Because size of the image array is limited by the size of the initial image and for the sinc-interpolation Discrete Fourier Transform is used, rotation may cause image aliasing due to the cyclicity property of DFT. This phenomenon is illustrated in Fig. 8. These effects can also be seen on an illustrative example of image rotation in Fig. 9.

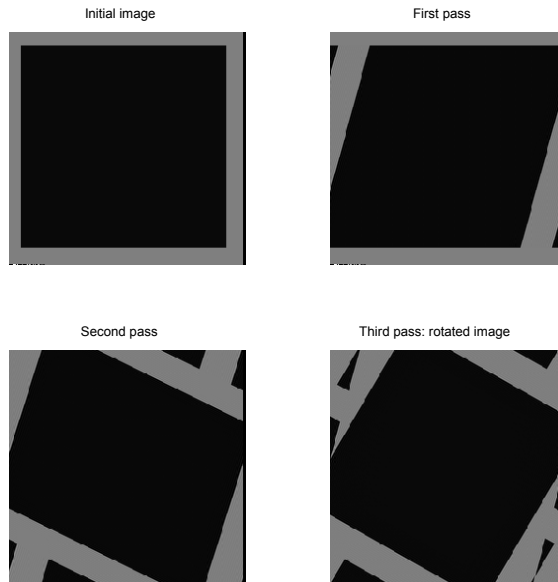


Fig. 8. Rotation aliasing effects.

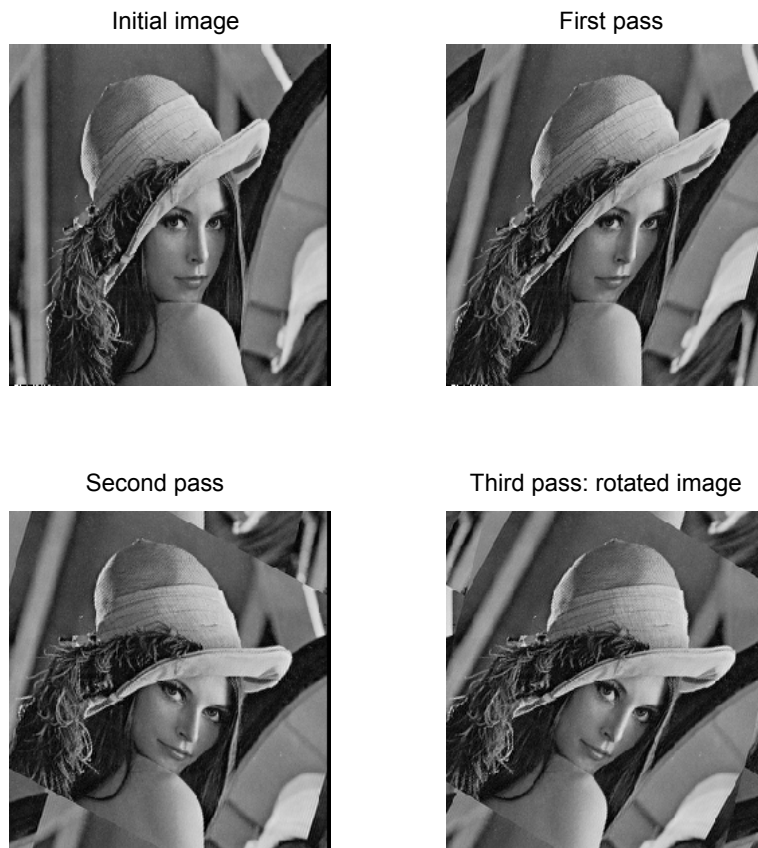


Fig. 9 An example of image rotation

### Image geometrical transformations by means of sinc-interpolated zooming

In many applications one needs to perform many different geometrical transformations of the same image. In such cases an efficient computational solution is to zoom image with sinc-interpolation and use the zoomed image as an approximation to a “continuous image” from which one can make arbitrary geometrical transformations by mapping on it raster points in the transformed coordinate system. If zoom factor is large enough, nearest neighbor interpolation on the zoomed image can be sufficient to keep required interpolation quality. Such a technology is schematically illustrated on Fig. 10.

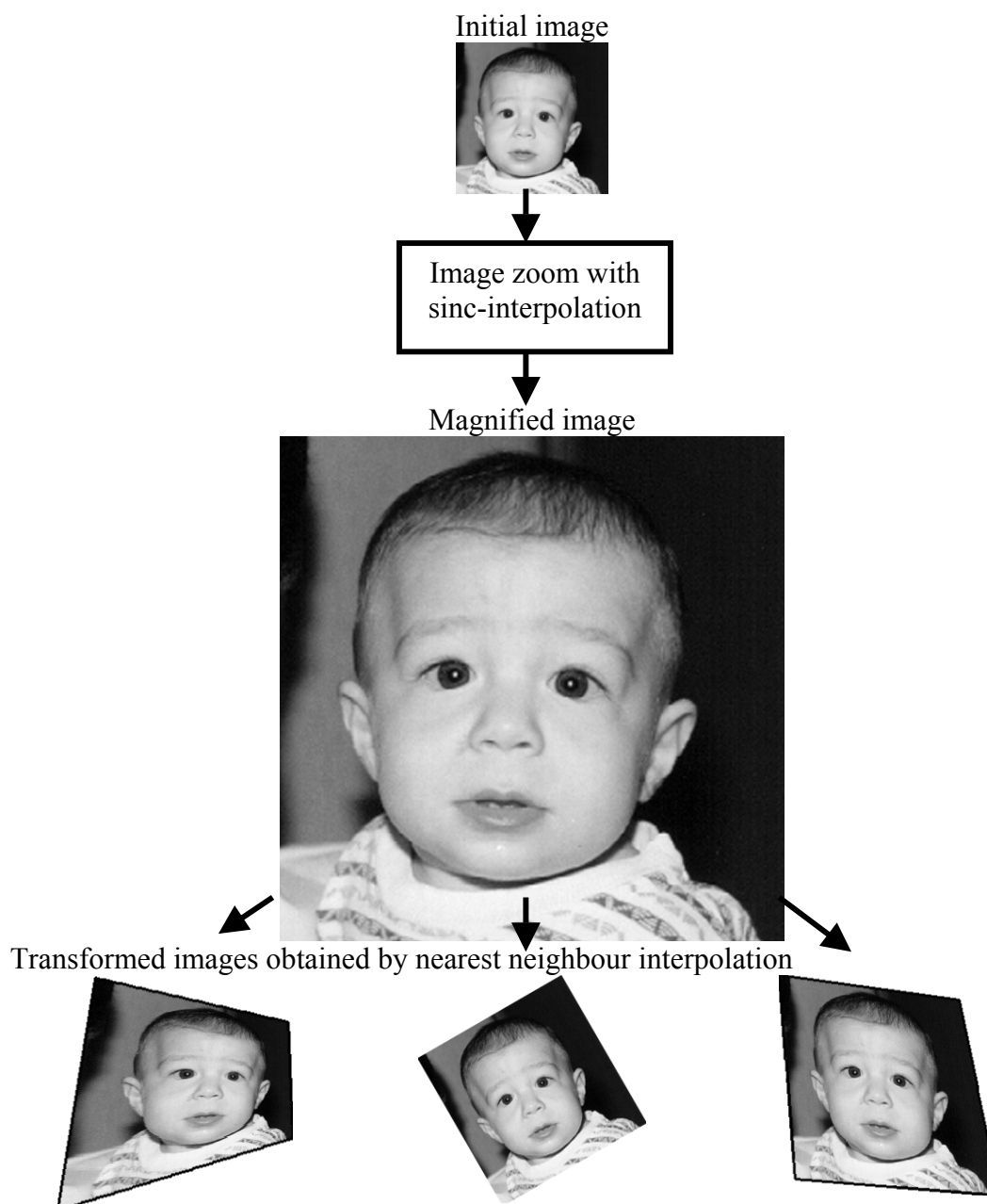
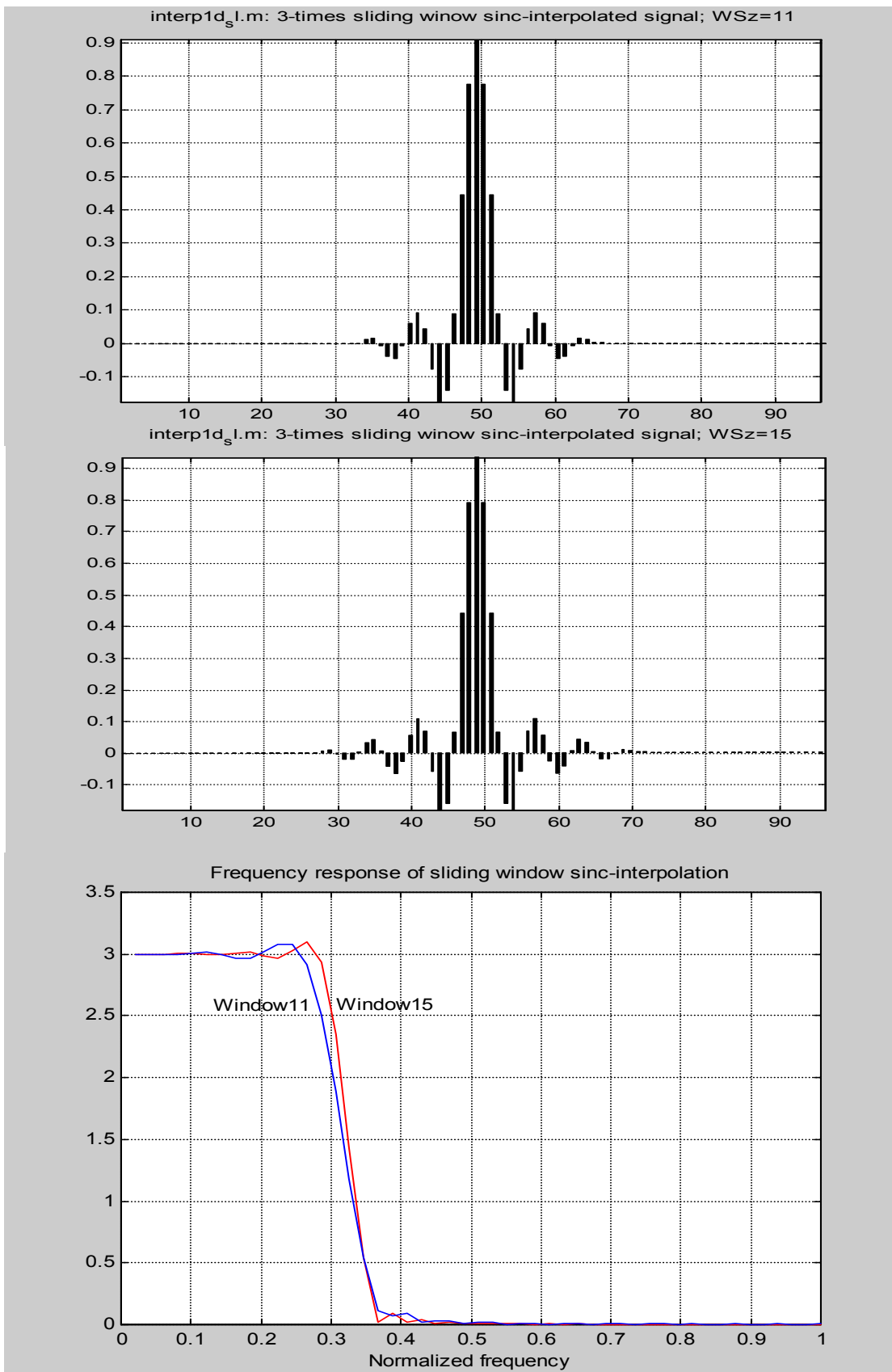


Fig. 10 Principle of image geometrical transformations via image zooming

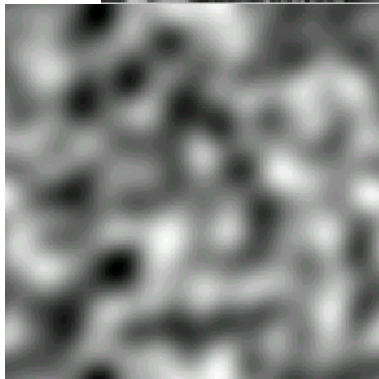
### Sinc-interpolation in sliding window:



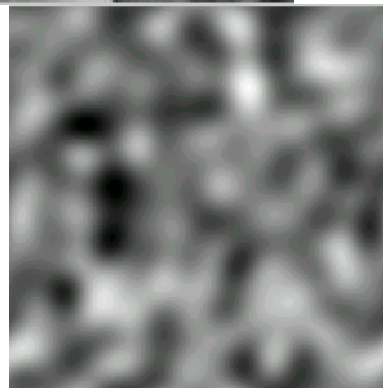
**Image zoom: global versus sliding window sinc-interpolation**



**Image co-ordinate local mapping**



Map-X



Map-Y

### **Additional references**

1. L. Yaroslavsky, M. Eden, Fundamentals of Digital Optics, Birkhauser, Boston, 1996
2. L. Yaroslavsky, Signal sinc-interpolation: a fast computer algorithm, Bioimaging, 4, p. 225-231, 1996
3. L.P. Yaroslavsky, Efficient algorithm for discrete sinc-interpolation, Applied Optics, Vol. 36, No.2, 10 January, 1997, p. 460-463
4. 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

## Test questions

1. What discrete signal interpolation (sub-sampling) methods you know?
2. Prove sampling theorem for continuous signals.
3. Why sinc-interpolation is a “gold standard” for convolution based interpolation of sampled data?
4. What is “zero-padding” sinc-interpolation method?
5. Explain and justify SDFT based fast sinc-interpolation algorithm
6. Compare “zero-padding” and SDFT-based fast sinc-interpolation techniques.
7. Why SDFT-based is so well suited for image rotation?
8. Describe fast sinc-interpolation in moving window and compare it with other known methods.