

# Digital Filtering Tutorial, Part II

*Alvy Ray Smith*  
*Computer Graphics Project*  
*Lucasfilm Ltd*

*Technical Memo No. 44*  
*10 May 1982 (Revised 11 May 1983)*

Presented as tutorial notes at the 1983 SIGGRAPH, July 1983, and the 1984 SIGGRAPH, July 1984. This document was reentered in Microsoft Word on 8 Feb 2000. Spelling and punctuation are generally preserved, but trivially minor spelling errors are corrected. Otherwise additions or changes made to the original are noted inside square brackets or in footnotes.

## INTRODUCTION

This note is to be considered an extension of *Digital Filtering Tutorial for Computer Graphics* [1]. As mentioned in the concluding section, the theory outlined in that memo does not treat the case of variable rate sampling, as occurs, for example, in perspective mappings. The Sampling Theorem assumes samples are taken at a uniform rate. We continue to assume a given input is sampled uniformly—at the pixels—and that the output is sampled uniformly—also at the pixels. So we can still use the Sampling Theorem, for example, to reconstruct the original input from its samples. In case of magnification and minification (scaling the abscissa  $x$  by a constant factor), the spacing of the input samples is changed by the scaling but the spacing remains uniform. In this memo, however, a less restricted class of mappings is considered which, in general, does not preserve input sample spacing uniformity. A good illustration of such a mapping is

$$f(x) = \frac{ax + b}{cx + d}$$

which occurs in perspective [2].

We shall show in this memo that the simplifications obtained for magnification and minification do not go through for general mappings, including, unfortunately, perspective.

Then we shall discuss some practical filters which are useful for the case where the Sampling Theorem does hold: for scaling, translation, and rotation. The filters include the Catmull-Rom and B-spline cubic spline basis functions and the sinc function made finite with a variety of window functions: Bartlett, Hamming, Hanning, Blackman, Lanczos, Kaiser, etc.

## THE GENERAL CASE

We can use the Sampling Theorem for an arbitrary mapping  $f(x)$  as follows:

- (1) Reconstruct  $g(x)$  from the given  $\hat{g}(x)$  samples by convolution.

$$g(x) = [\hat{g}(x) * \text{sinc}(x)](x).$$

(2) Map  $g^{\uparrow|x|}$  to obtain  $\hat{g}^{\uparrow|x|}$ .

$$g'(x) = g(f(x)) = [\hat{g}(x) * \text{sinc}(x)](f(x)).$$

(3) Since high frequencies may have been introduced into the function by  $f^{\uparrow|x|}$ , we must low-pass filter it before resampling.

$$h(x) = [g'(x) * \text{sinc}(x)](x) = [[\hat{g}(x) * \text{sinc}(x)](f(x)) * \text{sinc}(x)](x).$$

(4) Output  $\hat{h}^{\uparrow|x|}$  is simply  $h^{\uparrow|x|}$  at the pixels.

## THE SCALING SIMPLIFICATION

In [1], we studied the case  $f^{\uparrow|x|} = ax$  where

$$g'(x) = \hat{g}(ax) * \text{sinc}(ax) * \text{sinc}(x)$$

or

$$g'(x) = \hat{g}(ax) * \text{sinc}(ax), \quad a < 1$$

$$g'(x) = \hat{g}(ax) * \text{sinc}(x), \quad a \geq 1$$

(to within a normalization factor). These simplifications are easy to prove by looking in the frequency  $s$  domain where

$$\text{box}(as)\text{box}(bs) = \text{box}(\min(a, b)s).$$

## ARBITRARY ABCISSA MAPPINGS

Notice that we used the fact above that

$$[g(x) * h(x)](ax) = [g(ax) * h(ax)](x).$$

This does not hold for arbitrary abscissa mappings  $f^{\uparrow|x|}$ :

$$[g(x) * h(x)](f(x)) \neq [g(f(x)) * h(f(x))](x),$$

in general.

Thus, for perspective in particular, sampling theory does not hold. Full convolution must be used in theory. That is, the sampled function must be reconstructed from its samples (for which sampling theory works). The reconstructed function mapped by the abscissa mapping (eg, perspective) is then convolved with a low-pass filter to remove high frequencies introduced by the mapping. This step cannot be simplified as was accomplished in scaling case. Finally, the filtered function is sampled at the desired rate for output. Sampling theory again applies.

Although the result above is negative, it is only theoretically so. Approximations can be made—although not easily—to solve the filtering problem for perspective without resorting to full convolution. A case in point is ADO (Amplex Digital Optics), a machine for flipping video frames into perspective as often seen on current television. This machine, designed by Steve Gabriel and Larry Evans, approximates the perspective solution by switching among filters as the sample spacing decreases or increases. The details of this technique have not been published.

## PRACTICAL FILTERS FOR COMPUTER GRAPHICS

In the figures accompanying these notes are two filters of support 4—ie, spanning 4 samples—and several of support 8. The support-4 filters are nothing more than the basis functions for the Catmull-Rom (cardinal) cubic spline and the B-spline (beta) cubic spline which are familiar to computer graphicists from other contexts. The support-8 filters are all obtained from the sinc function from a *window (function)*. Windowing is accomplished by multiplying a window function times the sinc function. In all cases the filters are FIR (Finite Impulse Response) filters [3, 4].

The most simple window is just a box function of unit height and support 8. This is called the *rectangular* or *Fourier* window, and windowing in this case is simple truncation. The next higher order window is the *triangular* or *Bartlett* window. In the figures are shown several more sophisticated windows named typically for their discoverers.

Since the ideal filter is the sinc, it must be the case that windowing causes a non-ideal filter. A feeling for the error introduced by windowing may be got from considering a step function—ie, a hard edge in a picture. Convolving the step with a truncated sinc introduces the so-called Gibbs phenomenon, under-shoot before the edge, overshoot after it, and ripples in both places. Furthermore the edge is no longer straight up and down but is skewed slightly. So the hard end is softened slightly and appears to “ring” from the ripples on either side. A successful window function minimizes these annoying artifacts. As can be seen from Figure WINDOW, the more sophisticated the window, the smaller are the ripples (the lower is the “peak amplitude of the side lobe”) but the softer is the edge (“transition width of main lobe”).

Figure WINDOW gives the formulas for Rectangular, Bartlett, Hamming, Hanning, and Blackman windows. The Kaiser window is a family of windows where the parameter  $w_a$  trades ringing for edge softening. A slightly different formulation [4] of the Kaiser family is

$$w(n) = \frac{I_0\left(\alpha\sqrt{i(2-i)}\right)}{I_0(\alpha)},$$

where

$$i = \frac{2i}{N-1},$$

and  $a$  is the tradeoff parameter.  $a = 0$  corresponds to a Rectangular window and as it is increased by integers to about 10 it approximates the hierarchy of windows listed above and then exceeds it in sophistication. The Hamming window, for example, corresponds to a Kaiser window with  $a$  of 4 or 5. Several of the windows are shown in the figures as well as the filter resulting from the application of them to the sinc function.

The modified zeroth-order Bessel function of the first kind is given by the following series:

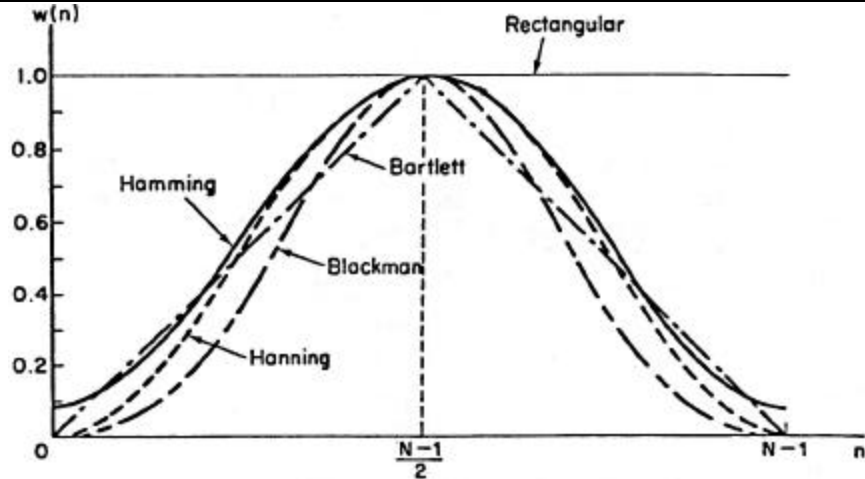
$$I_0(x) = 1 + \left(\frac{x}{2}\right)^2 + \frac{\left(\frac{x}{2}\right)^4}{1^2 \cdot 2^2} + \frac{\left(\frac{x}{2}\right)^6}{1^2 \cdot 2^2 \cdot 3^2} + \dots$$

A window not listed in Figure WINDOW is the Lanczos window which is the central lobe of the sinc function spread over support 8. Other filters which could be used include a truncated Gaussian and truncated cosine biased by its amplitude above zero.

For many applications, the cubic spline basis functions suffice. It should be noted that the windowing used above for obtaining support-8 filters could just as well have been used for arbitrary wide filters, say support-16 filters.

## REFERENCES

- [1] Smith, Alvy Ray. *Digital Filtering Tutorial for Computer Graphics*. Tech Memo No 27, Lucasfilm Ltd, Nov 20, 1981.
- [2] Catmull, Edwin, and Smith, Alvy Ray. *3-D Transformations of Images in Scanline Order*. **Computer Graphics**, Vol 14, No 3, 279-285, Jul, 1980. (SIG-GRAPH 80 Conference Proceedings).
- [3] Alan V Oppenheim and Ronald W Schafer, **Digital Signal Processing**, Prentice-Hall, Inc, Englewood Cliffs, NJ, 1975.
- [4] Lawrence R Rabiner and Bernard Gold, **Theory and Application of Digital Signal Processing**, Prentice-Hall, Inc, Englewood Cliffs, NJ, 1975.



Commonly used windows for FIR filter design.

Rectangular:

$$w(n) = 1, \quad 0 \leq n \leq N-1$$

Bartlett:

$$w(n) = \begin{cases} \frac{2n}{N-1}, & 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1}, & \frac{N-1}{2} \leq n \leq N-1 \end{cases}$$

Hanning:

$$w(n) = \frac{1}{2} \left[ 1 - \cos\left(\frac{2\pi n}{N-1}\right) \right], \quad 0 \leq n \leq N-1$$

Hamming:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

Blackman:

$$w(n) = 0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

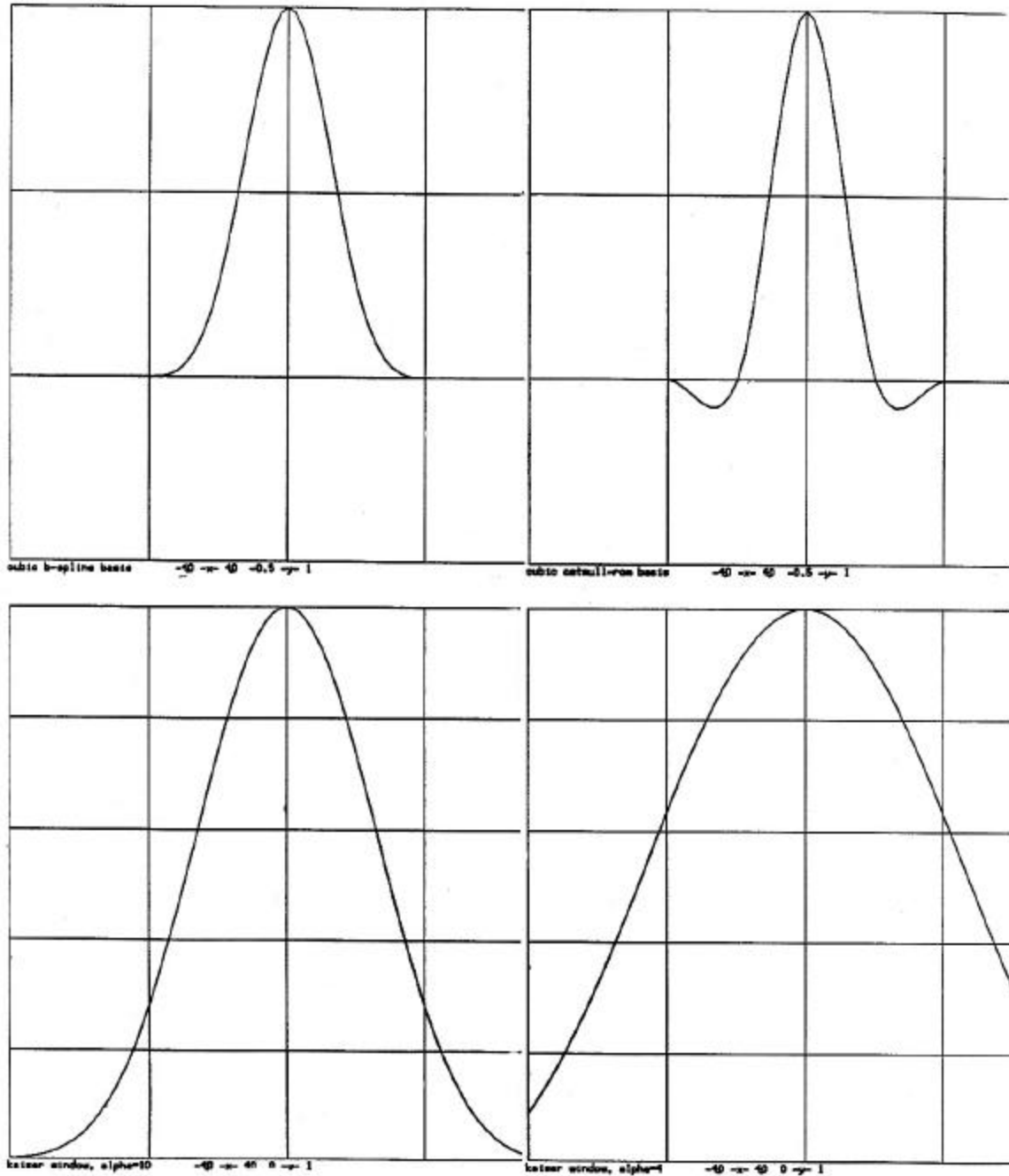
Kaiser:

$$w(n) = \frac{I_0 \left[ \omega_a \sqrt{\left(\frac{N-1}{2}\right)^2 - \left[n - \left(\frac{N-1}{2}\right)\right]^2} \right]}{I_0 \left[ \omega_a \left(\frac{N-1}{2}\right) \right]}$$

where  $I_0$  is the modified zeroth order Bessel function of the first kind.

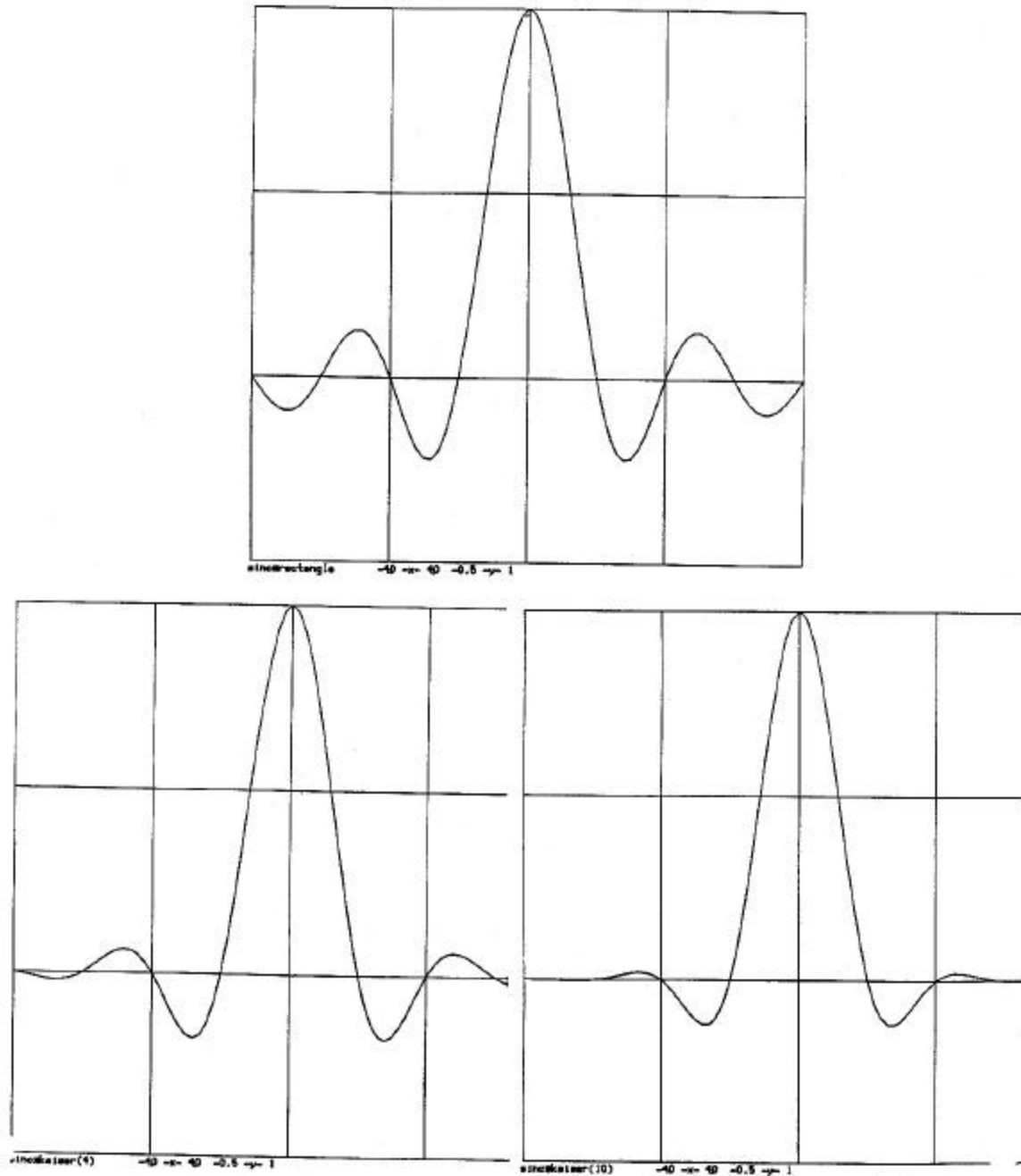
Window	Peak Amplitude of Side Lobe (dB)	Transition Width of Main Lobe	Minimum Stopband Attenuation (dB)
Rectangular	-13	$4\pi / N$	-21
Bartlett	-25	$8\pi / N$	-25
Hanning	-31	$8\pi / N$	-44
Hamming	-41	$8\pi / N$	-53
Blackman	-57	$12\pi / N$	-74

Figure WINDOW [adapted from [3]]



**Figure Captions:**

- upper left: cubic b-spline basis,  $-4.0 \leq x \leq 4.0$ ,  $0 \leq y \leq 1$
- upper right: cubic catmull-rom basis,  $-4.0 \leq x \leq 4.0$ ,  $0 \leq y \leq 1$
- lower left: kaiser window,  $\alpha = 10$ ,  $-4.0 \leq x \leq 4.0$ ,  $0 \leq y \leq 1$
- lower right: kaiser window,  $\alpha = 4$ ,  $-4.0 \leq x \leq 4.0$ ,  $0 \leq y \leq 1$



**Figure Captions:**

**upper: sinc \* rectangle, -4.0 £ x £ 4.0, -0.5 £ y £ 1**

**lower left: sinc \* kaiser(4), -4.0 £ x £ 4.0, -0.5 £ y £ 1**

**lower right: sinc \* kaiser(10), -4.0 £ x £ 4.0, -0.5 £ y £ 1**