

On Windowing for Gradient Estimation in Volume Visualization

Thomas Theußl
mailto:theussl@cg.tuwien.ac.at

Institute of Computer Graphics
Vienna University of Technology
Vienna / Austria

Abstract

Reconstruction of gradients from sampled data is a crucial task in volume visualization. Gradients are used, for instance, as normals for surface shading or for classification in standard ray casting techniques. Using the ideal derivative reconstruction filter, which can be derived from the ideal function reconstruction filter, is impracticable because of its infinite extend. Simply truncating the filter leads to problems due to discontinuities at the edges. To overcome these problems several windows have been defined, which are discussed in this paper with respect to gradient estimation.

Keywords: volume visualization, reconstruction, gradient, filter, frequency response

There is a lot of perhaps unnecessary lore about choice of a window function, and practically every function that rises from zero to a peak and then falls again has been named after someone.

in **Numerical Recipes in C** [17]

1 Introduction

Usually the goal of volume visualization is to get meaningful, two-dimensional pictures from three-dimensional data sets (for instance from CT or MRI imaging). A useful concept for that purpose is the gradient, which can be interpreted as a normal to a surface of equal values (an iso-surface) passing through the point of interest. Consequently, the gradient can be used to shade surfaces, since most lighting models (for instance Gouraud shading [5] or Phong shading [16]) make use of it or it can be used for classification in ray casting techniques [7]. Möller et al. even concede gradient reconstruction having a greater impact on image quality than function reconstruction itself [12].

From signal processing theory we know that the ideal reconstruction filter is the sinc function [15] given by

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \quad (1)$$

Provided that the function was sampled properly (according to Shannons sampling theorem, which states that the sampling frequency must be at least twice the highest frequency present in the function [18]) it can be reconstructed exactly by convolving (denoted by $*$) the samples with the sinc function:

$$\begin{aligned}
 f_r(x) &= f_s(x) * \text{sinc}(x) & (2) \\
 &= \int_{-\infty}^{\infty} f_s(u) \cdot \text{sinc}(x - u) du \\
 &= \sum_{n=-\infty}^{\infty} f[n] \cdot \text{sinc}(x - n)
 \end{aligned}$$

where $f_r(x)$ denotes the reconstructed function, $f_s(x)$ the sampled function and $f[n]$ the samples.

Sampling and reconstruction are often investigated in frequency domain. A function can be transformed by means of the Fourier transformation from the spatial domain to the frequency domain, where additional information about its behavior can be obtained (reconstruction filters, for instance, can be compared to the ideal one, i.e., the sinc filter). An excellent introduction to this topic is given by Blinn [2, 3]. In order to reconstruct the derivative of the function it is possible to reconstruct the function and derive it discretely. However, we chose another approach which is to derive the ideal reconstruction filter and use it as a derivative reconstruction filter directly.

Deriving the sinc function (Eq. 1) yields the cosc function [1] (note that the definition is not analogous):

$$\text{cosc}(x) = \frac{\cos(\pi x)}{x} - \frac{\sin(\pi x)}{\pi x^2} = \frac{\cos(\pi x) - \text{sinc}(x)}{x} \quad (3)$$

The frequency response of the cosc function is depicted together with the frequency response of the sinc function in Fig. 1 on the right, whereas the functions itself are depicted on the left. Since both sinc and cosc filter are of infinite extend we have to use in practice other, finite, filters which can, obviously, never reach the optimal frequency response. However, we can compare them to the optimal one and, consequently, gain additional information about their quality.

In this paper we will compare various filters obtained by bounding the ideal derivative reconstruction filter given by Eq. 3 to some finite extend. The simplest way to do this is to truncate the cosc function, which is, anyway, not the best way because of discontinuities at the edges. So special functions (adversely called windows) were designed to get a smoother transition to zero. Their definitions and a discussion of their frequency responses is given in Section 3 after a brief overview of some work already done in this area in Section 2. In Section 4 the test scenario used to compare the windowed cosc filters is described and the result are given and discussed in Section 5.

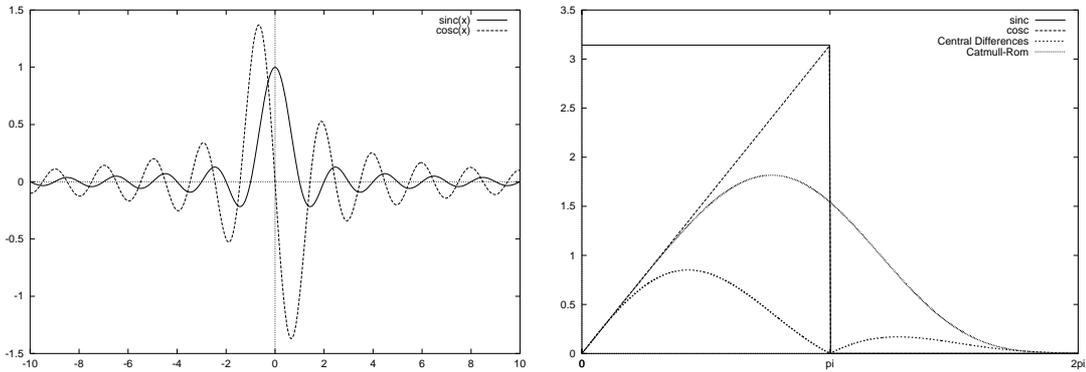


Figure 1: Sinc and cosc function on the left and their frequency responses on the right together with the frequency responses of the central difference operator with linear interpolation and the derivative of the Catmull-Rom spline.

2 Related Work

The most widely used method for gradient estimation is still the central difference operator which simply computes an averaged difference of values along each axes:

$$\begin{aligned}
 g_x &= \frac{f_{x+1,y,z} - f_{x-1,y,z}}{2} \\
 g_y &= \frac{f_{x,y+1,z} - f_{x,y-1,z}}{2} \\
 g_z &= \frac{f_{x,y,z+1} - f_{x,y,z-1}}{2}
 \end{aligned} \tag{4}$$

The same result is obtained by convolving the data set with the series $[\frac{1}{2}, 0, -\frac{1}{2}]$ in each dimension, the frequency response of this series can be computed analytically [4]. However, this only estimates gradients on sample positions, in between some kind of interpolation has to be performed. The frequency response of the central difference operator combined with linear interpolation (as done by Bentum et al [1]) is depicted in Fig. 1 on the right. It reveals the drawback of this method, which is that high frequencies in the data are attenuated and therefore a rather great amount of smoothing is introduced.

Bentum et al. [1] therefore proposed to use derivatives of cubic splines (which already were quite popular for function reconstruction [10]) as derivative reconstruction filters and Möller et al. [11, 13, 14] provide an analysis and analytic comparison of their performances. Machiraju and Yagel [8] discuss different gradient estimation schemes such as first reconstructing the function and then differentiating it or first differentiating the filter and convolving the data with that filter. An elaborated comparison is again given by Möller et al. [12].

Windowing itself is not that popular in computer graphics. Turkowski [19] used a Lanczos windowed sinc-function for decimation and interpolation of 2D image data. Goss [4] eventually proposed to use a Kaiser windowed cosc function for gradient estimation. This method will be more closely examined in Section 3.9.

3 Window Definitions

In the following, definitions of some windowing functions are given. All of the windows, except one, the Parzen Window (Section 3.4), can be of arbitrary extend which is specified by a parameter τ . Fig. 2 on the left depicts all windows with width two. Also all frequency responses (Fig. 2 on the right) were generated (using the discrete Fourier transform) with windows of extend two. Consequently, we can directly compare the various windowed cosc filters with cubic spline derivatives which have the same extend (the frequency response of the derivative of the Catmull-Rom spline is depicted in Fig. 1).

3.1 Rectangular Window

The simplest method to bound the ideal derivative reconstruction filter to some finite extend is to truncate it. This is tantamount to multiplying it with a rectangular function which is 1 inside some extend and 0 outside.

$$\text{Rect}(x, \tau) = \begin{cases} 1 & |x| < \tau \\ 0 & \text{else} \end{cases} \quad (5)$$

Although the frequency response approximates the ideal one quite good below π , it is significantly different from zero in form of bumps above, which is due to the truncation and particularly annoying in gradient reconstruction because it introduces artifacts even more perceivable than in function reconstruction. To avoid these bumps, other windows trade them against the rather good approximation below π .

3.2 Bartlett Window

A Bartlett Window is simply a tent function, i.e.

$$\text{Bartlett}(x, \tau) = \begin{cases} 1 - \frac{|x|}{\tau} & |x| < \tau \\ 0 & \text{else} \end{cases} \quad (6)$$

The frequency response approaches zero more smoothly but does not approximate the ideal frequency response below π as good as the truncated cosc function. This rather simple windowing function, although said to be sufficient in the Numerical Recipes in C (used in another context) [17], performs quite poor when used for gradient reconstruction.

3.3 Welch Window

Another quite simple window, which uses only polynomials, is the Welch window.

$$\text{Welch}(x, \tau) = \begin{cases} 1 - \left(\frac{x}{\tau}\right)^2 & |x| < \tau \\ 0 & \text{else} \end{cases} \quad (7)$$

Its frequency response is better than that of the Bartlett windowed cosc function below π , but it shows again a rather distinctive bump above.

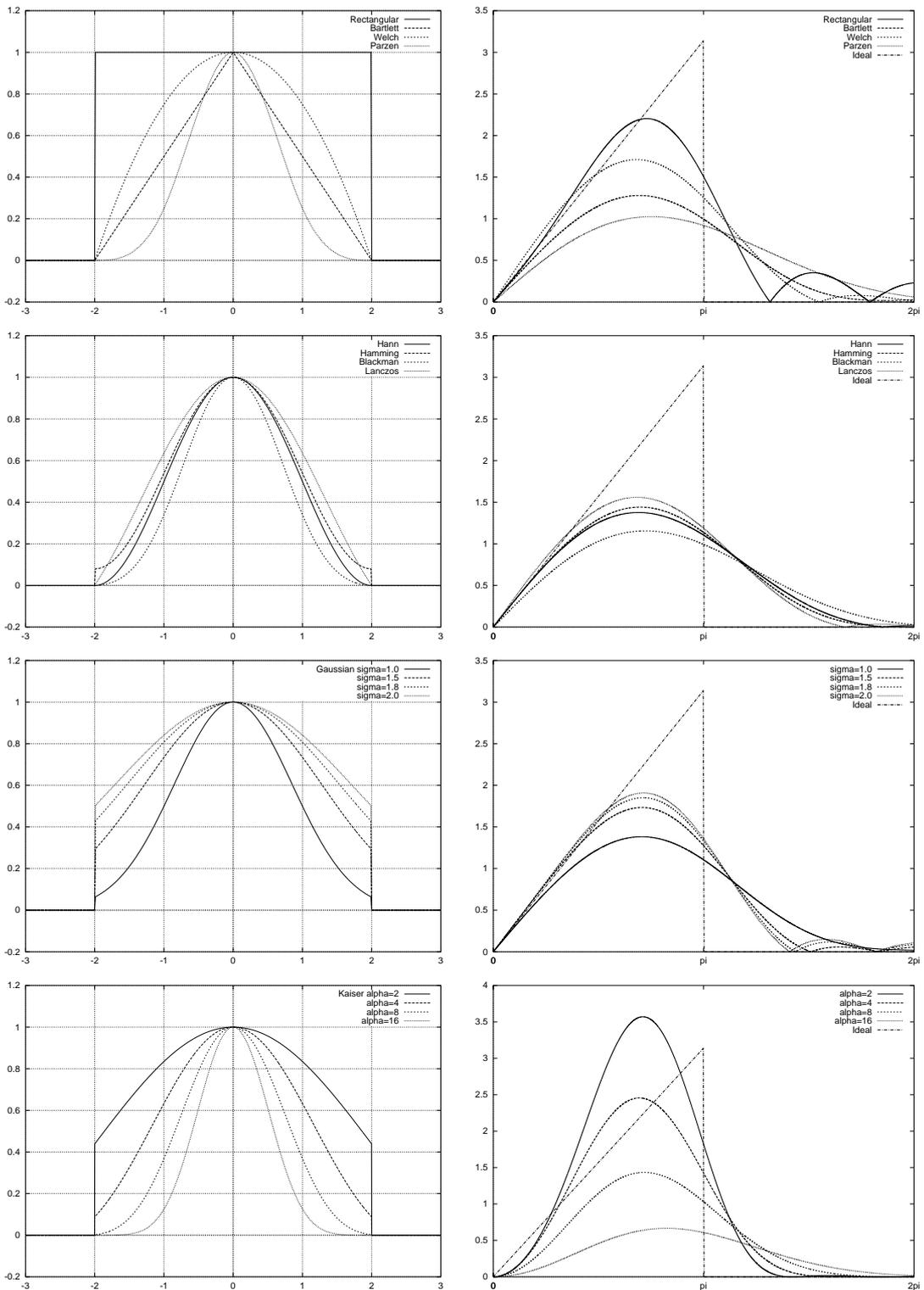


Figure 2: Frequency responses of windowed cosc functions on the right, the windows themselves on the left.

3.4 Parzen Window

The Parzen window is a piece-wise cubic approximation of the Gaussian Window of extend two. It is the only window, that is discussed in this paper, that is fixed to a certain width.

$$\text{Parzen}(x) = \frac{1}{4} \begin{cases} (2+x)^3 & -2 \leq x < -1 \\ 4 - 6x^2 - 3x^3 & -1 \leq x < 0 \\ 4 - 6x^2 + 3x^3 & 0 \leq x < 1 \\ (2-x)^3 & 1 \leq x < 2 \\ 0 & \text{else} \end{cases} \quad (8)$$

However, its frequency response is not really promising for it is worse than that of the Bartlett windowed cosc function (generally, when compared to all other frequency responses, it approximates the ideal one most poorly above π and is not really good below).

3.5 Hann and Hamming Window

The Hann (due to Julius van Hann, often wrongly referred to as Hanning window [20], sometimes just cosine bell window) and Hamming window are quite similar, they only differ in the choice of one parameter α :

$$\text{H}(x, \tau, \alpha) = \begin{cases} \alpha + (1 - \alpha) \cos(\pi \frac{x}{\tau}) & |x| < \tau \\ 0 & \text{else} \end{cases} \quad (9)$$

with $\alpha = \frac{1}{2}$ being the Hann window and $\alpha = 0.54$ the Hamming Window. The Hamming window also has the disadvantage of being discontinuous at the edges (and has therefore a, however not so severe, bump above π), which leads to clearly visible artifacts in the images.

3.6 Blackman Window

The Blackman window is quite similar, too, to Hann and Hamming window, but it has one additional cosine term to further reduce the ripple ratio.

$$\text{Blackman}(x, \tau) = \begin{cases} 0.42 + \frac{1}{2} \cos(\pi \frac{x}{\tau}) + 0.08 \cos(2\pi \frac{x}{\tau}) & |x| < \tau \\ 0 & \text{else} \end{cases} \quad (10)$$

Although its frequency response is slightly worse below π , it approaches zero more smoothly above and turns out to produce quite satisfying results.

3.7 Lanczos Window

The Lanczos window is the central lobe of a sinc function scaled to a certain extend.

$$\text{Lanczos}(x, \tau) = \begin{cases} \frac{\sin(\pi \frac{x}{\tau})}{\pi \frac{x}{\tau}} & |x| < \tau \\ 0 & \text{else} \end{cases} \quad (11)$$

Turkowski [19] reported it being superior for two-dimensional image resampling tasks, i.e. using it to bound the sinc function. However, using it to bound the cosc function is also possible, although the frequency response shows a little bump above π .

3.8 Gaussian Window

The Gaussian Window in its general form is defined by

$$\text{Gauss}(x, \tau, \sigma) = \begin{cases} 2^{-\left(\frac{x}{\sigma}\right)^2} & |x| < \tau \\ 0 & \text{else} \end{cases} \quad (12)$$

with σ being the standard deviation. The higher σ gets, the wider gets the Gaussian window and, on the other hand, the more severe gets the truncation. Several Gaussian windows with different standard deviations are depicted in Fig. 2 in the third row. The higher σ the better the frequency response approximates the ideal one below π but also the more distinctive are the bumps above π .

3.9 Kaiser Window

The Kaiser window [6] has an adjustable parameter α which controls how quickly it approaches zero at the edges. It is defined by

$$\text{Kaiser}(x, \tau, \alpha) = \begin{cases} \frac{I_0(\alpha\sqrt{1-(x/\tau)^2})}{I_0(\alpha)} & |x| \leq \tau \\ 0 & \text{else} \end{cases} \quad (13)$$

where $I_0(x)$ is the zeroth order modified Bessel function (for a definition, and a more detailed discussion, of the Bessel functions see, for instance, the Numerical Recipes in C [17]). The higher α the narrower gets the window and therefore, due to the not so severe truncation then, the less severe are the bumps above π . In Fig. 2 again, but in the fourth row, several Kaiser windows are depicted with different values for α . The frequency responses on the right shows that the parameter α directly controls its shape.

Goss [4] used this window to obtain an adjustable gradient filter, but he used it only on sample points so that, in between sample points, some kind of interpolation has to be performed, which he does not state explicitly. In this work, the Kaiser windowed cosc function will be used to reconstruct gradients at arbitrary positions which is, of course, more costly.

4 Test Scenario

The windowing functions described theoretically in the previous section were tested in practice with two different data sets which were reconstructed with always the same reconstruction scheme so that the only difference between the pictures is the windowing function used for gradient reconstruction.

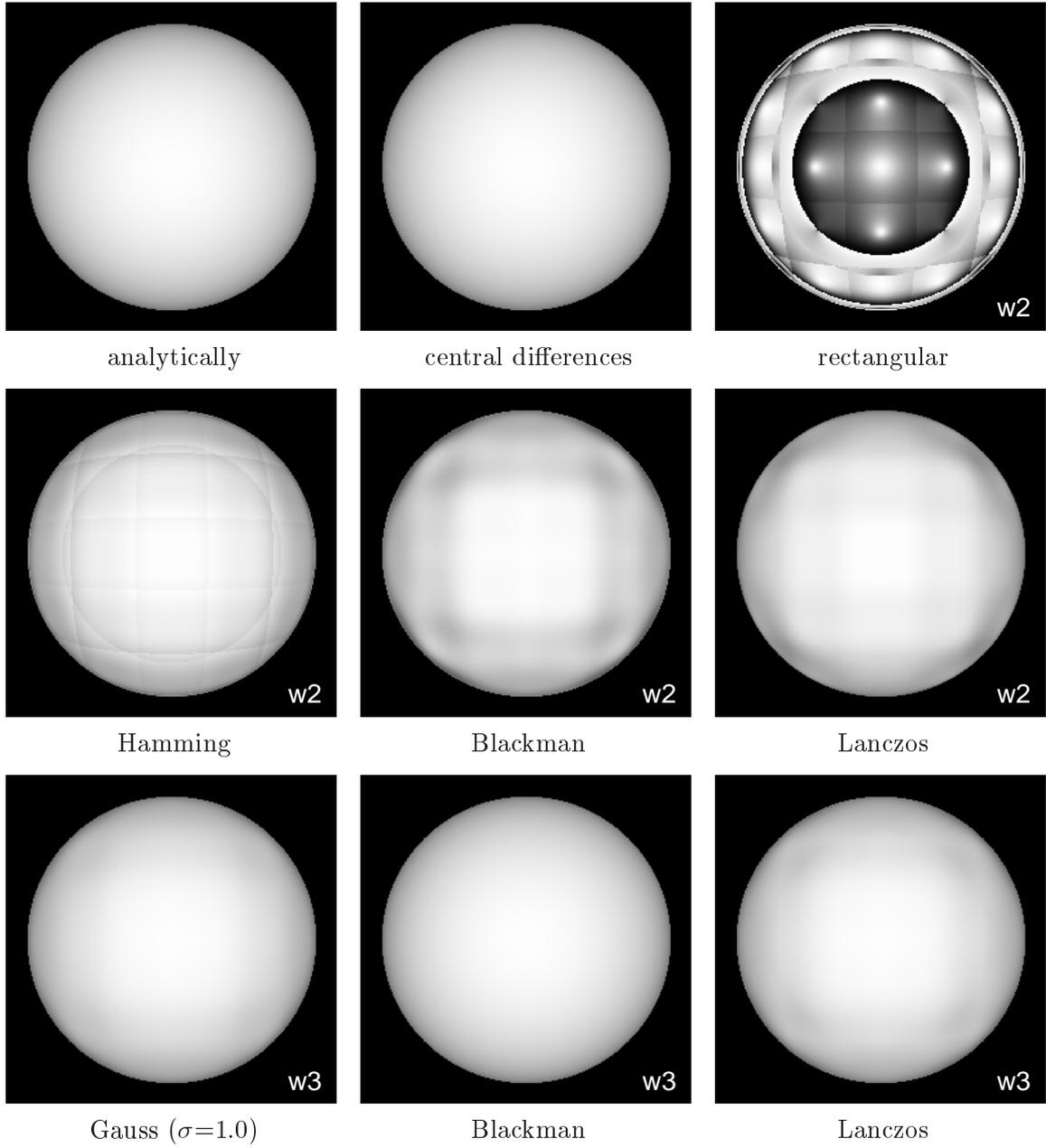


Figure 3: Sphere data set. Gradients reconstructed analytically, with central differences and various windowed cosc filters with window width as depicted in the lower right corners.

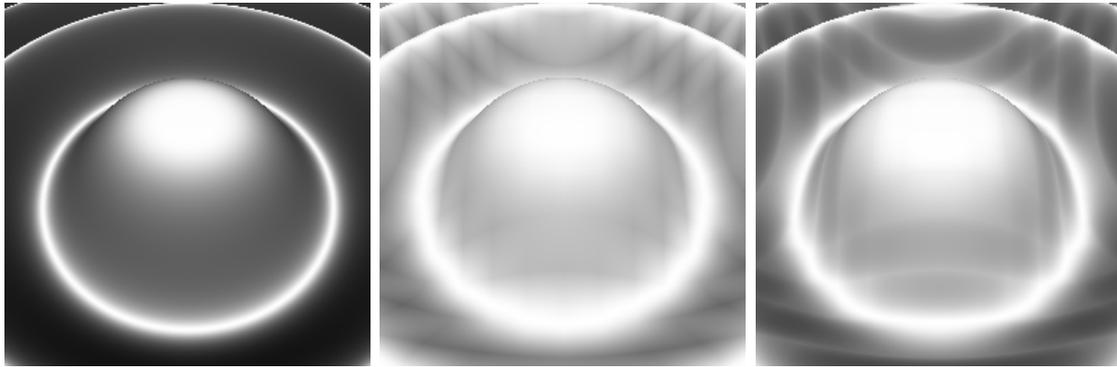


Figure 4: Marschner Lobb function with analytically calculated gradients on the left, central differences used for gradient reconstruction in the middle and Catmull-Rom derivative on the right.

The first data set was simply a distance function (i.e., its iso-surfaces are spheres), sampled on a 64 by 64 by 64 grid. It is depicted with analytically calculated gradients in Fig. 3 on the top left.

The second data set was a, by now quite standard, test signal proposed by Marschner and Lobb [9], it is depicted with analytically calculated gradients in Fig. 4 on the top left. This signal was sampled on a 40 by 40 by 40 grid in the range $-1 < x, y, z < 1$ (however, the actual part depicted is zoomed in to the middle portion), it is a little bit more complex than a sphere and should therefore provide more clues of the differences between and the applicability of the various windowed cosc functions.

Additionally to the windowed cosc functions, the central differences operator and the derivative of a piece-wise cubic spline, known as Catmull-Rom spline [1], were used for comparison purposes.

5 Results

In Fig. 3 the results of several gradient reconstruction schemes of the sphere data set are depicted. In the first row on the left the gradients were calculated analytically, in the middle the central difference operator was used, which gave, and that is quite interesting, for this data set the best looking result. Surprisingly, the truncated cosc function (rectangular windowed with width two, first row on the right) gives a really bad result. Other windows with width two are better but not really satisfying, as can be seen in Fig. 3 in the second row. Only the Gauss windowed (with $\sigma = 1.0$) and the Blackman windowed cosc function with window width three yield comparable results to the central difference operator for this data set (depicted in the third row left and middle image, for the right one a Lanczos windowed cosc function of the same width was used which, admittedly, shows some irregularities again).

Another test series was carried out with the Marschner Lobb test signal. It can be seen, with analytically calculated gradients, in Fig. 4 on the left. The images in the

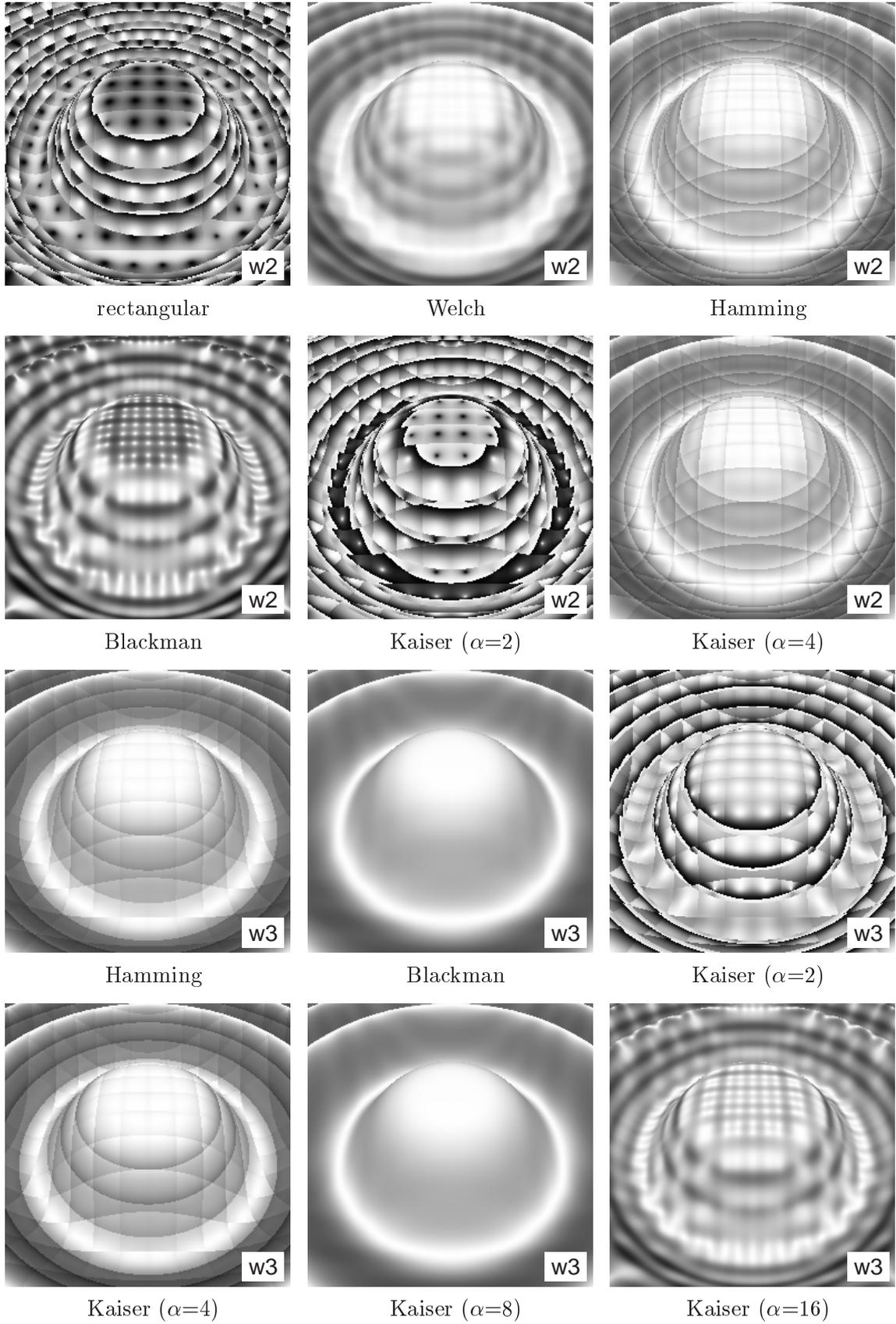


Figure 5: Marschner Lobb data set. Gradients reconstructed with various windowed cosc function with window width as denoted in the lower right corners.

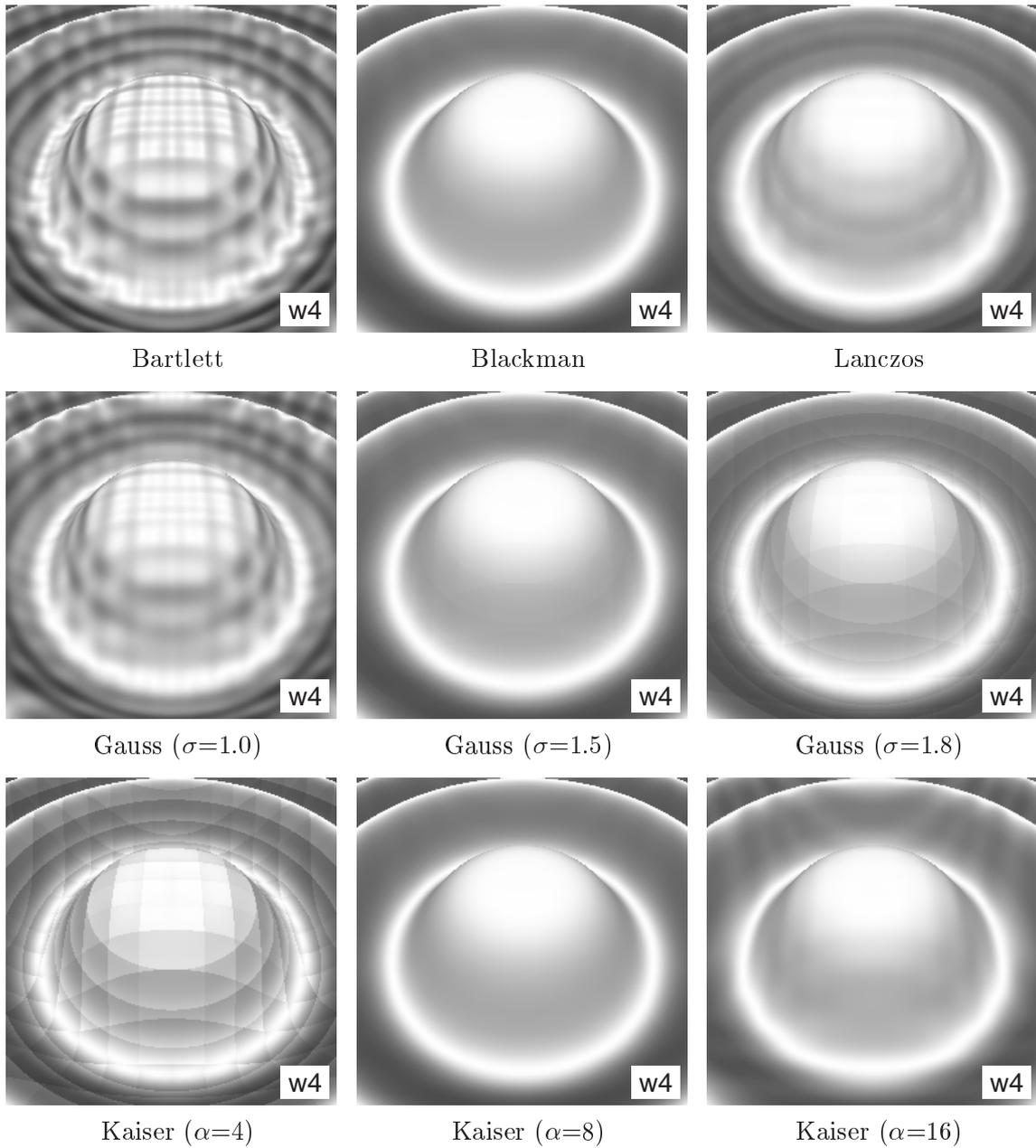


Figure 6: Marschner Lobb data set. Gradients reconstructed with various windowed cosc functions with window width four.

middle, reconstructed with central differences, and on the right, reconstructed with the derivative of the Catmull-Rom spline, show some obvious irregularities.

Bounding the cosc function with windows of width two does not yield much better results, as depicted in Fig. 5 first and second row. Again, a simple truncation yields really bad results (first row left image). Some windows show a slight improvement, but the visual appearance of the central difference and, at any rate, the Catmull-Rom spline derivative is still better. However, worth mentioning is the adjustability of the Kaiser window with its parameter α (as shown in the second row where the right picture with $\alpha = 4$ is much more appealing than the middle one with $\alpha = 2$).

Extending the window width to three, eventually, yields quite satisfying results (Fig. 5 third and fourth row). The two images in the middle column (Blackman window in the third and Kaiser window with $\alpha = 8$ in the fourth) are, at last, quite smooth and visually appealing. However, the left column shows images with conspicuous artifacts due to the discontinuities at the edges of the Hamming and Kaiser (with $\alpha = 4$) window. The right column shows that also windows of width three can yield quite bad results. Notable again is the adjustability of the Kaiser window, which ranges from really bad (with $\alpha = 2$, third row right image) over getting better (with $\alpha = 4$, fourth row left image) to really good (with $\alpha = 8$), and it gets worse again with an α too high (for instance, $\alpha = 16$ in the bottom right image).

Further extending the window width yields, not very surprisingly, even better results, however, just with certain windows. Fig. 6 (on the top left) shows that, for instance, the Bartlett windowed cosc function even with width four is quite a bad choice and the Lanczos window (top right), although much better, still shows some artifacts. Really good results, on the other hand, were obtained by use of the Blackman window (first row, middle image) which had quite a good result with width three already (Fig. 5 middle image on the top). Also, the Gaussian window gets now interesting, in the second row images obtained by varying σ are depicted and at least the one with $\sigma = 1.5$ is quite good. The third row, again, shows the usefulness of the Kaiser window by varying its parameter α .

For further information and additional pictures please refer also to URL <http://www.cg.tuwien.ac.at/studentwork/CESCG99/TTheuss1/>

6 Conclusions

Using windowed cosc functions for gradient reconstruction is superior to other approaches (like central differences or cubic spline derivatives) when the window has an adequate width. The tests performed in this work showed that this width should at least be three. This means that it is, on the other hand, more costly than using cubic spline derivatives (which have width two) and central differences anyway. For some functions, for instance, the sphere data set used in this work, these simpler methods yield even better results.

However, when the gradients of some more complex functions must be reconstructed the choice of the windowing function is crucial. The tests performed with the Marschner Lobb test signal showed that the Blackman window seems to be a

first good choice. If more control over the reconstruction function is necessary, the Gaussian window, on the one hand, can be adjusted by its parameter σ . However, this seems to be a good just if the window width is four or more. On the other hand, the Kaiser windowed cosc function yielded good results with a window width of only three and is also adjustable by its parameter α .

Most of the windows discussed in this paper seem not to be useful for gradient reconstruction. Especially the ones with discontinuities at the edges, which are also visible in the frequency spectrum as bumps above π , showed some clearly visible artifacts in the rendered images.

7 Future Work

All images in this paper were generated by evaluating the windowed cosc functions analytically. This means that the cosc function was evaluated analytically (one cosine and one sine term) and the windowing function as well (which is quite complex in some cases), which is too costly for practical purposes. To avoid this, the windowing functions could be sampled into a lookup table. This would, however, affect the image quality, dependent on the sampling rate and, consequently, the size of the lookup table. This aspect should be thoroughly investigated to obtain an adequate speed-memory trade-off.

References

- [1] Mark J. Bentum, Barthold B. A. Lichtenbelt, and Tom Malzbender. Frequency Analysis of Gradient Estimators in Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):242–254, September 1996.
- [2] James F. Blinn. Jim Blinn’s corner — what we need around here is more aliasing. *IEEE Computer Graphics and Applications*, 9(1):75–77, January 1989.
- [3] James F. Blinn. Return of the jaggy. *IEEE Computer Graphics and Applications*, 9(2):82–89, March 1989.
- [4] Michael E. Goss. An adjustable gradient filter for volume visualization image enhancement. In *Proceedings of Graphics Interface '94*, pages 67–74, Banff, Alberta, Canada, May 1994. Canadian Information Processing Society.
- [5] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, C-20(6):623–629, June 1971.
- [6] J. F. Kaiser and R. W. Schafer. On the Use of the I_0 -Sinh Window for Spectrum Analysis. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-28(1):105, 1980.
- [7] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, February 1987.

- [8] R. Machiraju and R. Yagel. Reconstruction error characterization and control: A sampling theory approach. *IEEE Transactions on Visualization and Computer Graphics*, 2(4):364–378, December 1996.
- [9] S. R. Marschner and R. J. Lobb. An evaluation of reconstruction filters for volume rendering. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of the Conference on Visualization*, pages 100–107, Los Alamitos, CA, USA, October 1994. IEEE Computer Society Press.
- [10] Don P. Mitchell and Aru N. Netravali. Reconstruction filters in computer graphics. *Computer Graphics*, 22(4):221–228, August 1988.
- [11] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. Classification and local error estimation of interpolation and derivative filters for volume rendering. In *Proceedings 1996 Symposium on Volume Visualization*, pages 71–78, September 1996.
- [12] Torsten Möller, Raghu Machiraju, Klaus Mueller, and Roni Yagel. A comparison of normal estimation schemes. In Roni Yagel and Hans Hagen, editors, *IEEE Visualization '97*, pages 19–26. IEEE, November 1997.
- [13] Torsten Möller, Raghu Machiraju, Klaus Mueller, and Roni Yagel. Evaluation and Design of Filters Using a Taylor Series Expansion. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):184–199, April 1997.
- [14] Torsten Möller, Klaus Mueller, Yair Kurzion, Raghu Machiraju, and Roni Yagel. Design of accurate and smooth filters for function and derivative reconstruction. In *IEEE Symposium on Volume Visualization*, pages 143–151. IEEE, ACM SIGGRAPH, 1998.
- [15] A.V. Oppenheim and R. W. Schaffer. *Digital Signal Processing*. Prentice Hall, Inc., 1975.
- [16] Bui-Tuong Phong. Illumination for computer generated pictures. *CACM June 1975*, 18(6):311–317, 1975.
- [17] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge UP, 1988.
- [18] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37:10–21, January 1949.
- [19] Ken Turkowski. Filters for common resampling tasks. In Andrew S. Glassner, editor, *Graphics Gems I*, pages 147–165. Academic Press, 1990.
- [20] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1993.