Ohio State University Technical Report OSU-CISRC-1/95-TR03, January 1995.

Error-Bounded and Adaptive Image Reconstruction

Raghu Machiraju, Edward Swan, and Roni Yagel

Department of Computer and Information Science
The Advanced Computing Center for the Arts and Design
The Ohio State University

Abstract

Reconstruction is imperative whenever an image needs to be resampled as a result of transformation such as an affine or perspective transform, or texture mapping. We present a new method for the characterization and measurement of reconstruction error. Our method, based on spatial domain error analysis, uses approximation theory to develop error bounds. We provide, for the first time, an efficient way to guarantee an error bound at every point by varying filter size. We go further to support position-adaptive and data-adaptive reconstruction which adjust filter size to the location of reconstruction and the data in its vicinity. We demonstrate the effectiveness of our methods with 1D and 2D examples.

1. Introduction

Reconstruction is the process of recovering a continuous function from a set of samples. It is one of the fundamental operations in computer graphics and image processing. Many algorithms, such as texture mapping, image transformation (e.g., rotation, scaling), and volume rendering, transform a raster (2D or 3D) from a *source space* to a *target space*. All these algorithms must reconstruct the underlying function in source space before transforming it to target space, where resampling takes place. The work described here is aimed to give the user, for the first time, the ability to set a *point-wise* error bound. Unlike existing methods which use frequency domain analysis to guarantee some global error bound, we use spatial domain error analysis to guarantee that, for a given threshold ε, the difference between the reconstructed function and the real function is not more than ε, at *any point* in the source space.

Our spatial domain analysis culminates in a formal expression for the error bound at every point in the source space (Equation 14). Examining this expression, we observe a dependency between error magnitude and the location of reconstruction and data values. Unlike existing methods we can, therefore, adapt filter size to both reconstruction location and data complexity, using rigorous estimates.

In Section 2 we introduce the terminology and methods currently emphasized in reconstruction methods. In Section 3 we describe our approach and in Section 4 we present our results.

2. Background

An image or a volume is usually in the form of a regular rectilinear *grid* or a *mesh* of sampled function values termed pixels (image) or voxels (volume). When 2D images are subjected to affine transformations (e.g., translation, scaling, rotation) or when they are subjected to non-affine grid deformation (perspective, texture mapping, warping [1]), the function value in the form of pixel intensity has to be computed on the target grid, commonly called the *resampling grid*. Similarly, reconstruction is also needed when a 3D image (volume [9]) is subjected to affine transformations (e.g., translation, scaling, or orthographic ray casting [23]), or non-affine transformations (e.g., perspective ray casting [8], warping [4][6], and registration).

2.1 Ideal Reconstruction

Much has been written about the reconstruction of sampled datasets in signal processing (1D data) [15] or image processing applications (2D data) [7][22]. We briefly discuss some of the important assumptions and results from this body of literature. Another body of work on the same problem is available in the applied mathematics literature [19], where the process is usually referred to as *interpolation*. Although the following discussion is in terms of 1D signals, it is also applicable to 2D and 3D signals.

We denote by $f(x)$ a continuous function (*the signal*) which is sampled into the discrete function $f_s(kh)$, where h is an equidistant gap between samples and k is an integer. In computer graphics $f(x)$ is not available; we only have f_s, which is the discrete image we need to manipulate. Prior to resampling one must reconstruct from f_s, the continuous function, which we denote by $f_R(x)$. In principle, the error we want to measure and control is $|f_R(x)-f(x)|$.

The fundamental assumption made in this paper is that the original continuous function, $f(x)$, is *bandlimited*. A function is bandlimited if there exists a frequency ω_c, called the *cut-off frequency*, such that the strength of any frequency component greater than ω_c is zero. Inherent in this assumption is that the function is analytic, i.e., the function and its derivatives exist at all points. Thus, we are precluding step functions and other discontinuous functions. The bandlimited assumption leads to the following restrictions on the *Fourier spectrum* $F(\omega)$ of the function f:

$$\int_{Đ\omega_c}^{\omega_c} |F(\omega)|^2 < \infty \qquad \int_{Đ\omega_c}^{\omega_c} |F(\omega)| < \infty \tag{1}$$

where

$$F(\omega) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} F(\omega)e^{-j\omega t}d\omega \tag{2}$$

The first condition of Equation 1 indicates that *f* has finite energy, i.e., *f* belongs to the L^2 space. The second condition indicates the integrability of *F(ω)* or that *f* belongs to the L^1 space. Although it is not essential, we also assume that our function is spatially limited.

These assumptions are not too restrictive, because the notion of bandlimitedness is very general and can be applied to many forms of sampled data used in computer graphics and scientific visualization. During the process of acquiring digital images, acquisition devices (e.g., cameras, scanners) perform a filtering operation and bandlimit the function. Images generated by numerical simulations of physical phenomenon (common in disciplines such as computational fluid dynamics) are also bandlimited, since typically robust numerical solutions can only be obtained if the algorithm incorporates a smoothing step. And rendering and scan-conversion algorithms, in order to provide antialiased images, typically employ a filtering step which bandlimits the image. Even if we cannot assume that the function is bandlimited, we can subject the image to yet another filtering step (often called pre-filtering) to achieve bandlimitedness.

Another important assumption is that the continuous signal *f* is sampled at or above the *Nyquist frequency.* The Nyquist frequency of a signal is defined as twice the maximum frequency of the signal. Thus, in our context of bandlimited functions, the Nyquist frequency ω_n is given by $2\omega_c$ and the sampling frequency ω_s is always greater than or equal to ω_n. This assumption is essential if we are to reconstruct the function exactly. The Whitaker-Shannon-Koletnikov (WKS) theorem, known commonly as Shannon's Sampling theorem, states that any bandlimited continuous signal *f(x),* if sampled at or above its Nyquist frequency (yielding the discrete function f_s), can be reconstructed as shown in Equation 3 (yielding the continuous function f_R) [15][19]. A final assumption is that $f_s(x)$ is uniformly sampled from *f(x).*

Thus, we reconstruct with the formula

$$f_R(x) = \sum_{k=-\infty}^{\infty} S(x,k,h)f_s(kh) \tag{3}$$

where

$$S(x,k,h) = \begin{cases} 1 & x = kh \\ \dfrac{\sin\frac{\pi}{h}(x-kh)}{\frac{\pi}{h}(x-kh)} & x \neq kh \end{cases} \tag{4}$$

The function *S(x,k,h)* is called the *Sinc* function. The sampling frequency ω_s is determined by the inter-sample distance *h*; it is equal to *π/h*. Reconstruction is essentially a convolution operation between the *Sinc* function and the sampled dataset. A weighted sum of all samples is used to determine the function value at some point *x*. Contributions from samples significantly small and/or far away are minimal.

Another, albeit functional, perspective is that the reconstructed function is composed from an orthogonal basis of *Sinc* functions *S(x,k,h)*, $-\infty < k < \infty$ [19]. Equation 3 readily provides this perspective, since the contribution at point *x* is the weighted sum of *Sinc* functions indexed by *k*. Also, at the sampled data locations the right-hand side of Equation 3 evaluates to the original function value. Thus, the *control points* or coefficients of the basis representation are the sampled data points themselves. This property of the *Sinc* series expansion allows us to forgo both expensive transformations (such as *Wavelet* and *Fourier Transform*), and the computation of control points (such as *Spline* based methods). This perspective is later used to estimate the error of reconstruction.

We can extend this discussion to multiple dimensions. We can use *separable filters,* which sample the data successively along each axis. Thus, in 2D the reconstruction equation becomes:

$$f_R(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} S(x, i, h_x) S(y, j, h_y) f_s(ih_x, jh_y) \tag{5}$$

The quantities h_x and h_y are the sampling periods along each of the principal axes. For 2D images, the reconstruction requires the product of two *Sinc* functions, and similarly we can use three *Sinc* functions in 3D. The processing order of the axes is important and can determine the final quality of the image. Although separable filters are fast and easily implemented, separability introduces anisotropic effects since the 2D or 3D separable filter is aligned with the principal axes [13]. Anisotropic effects are always present unless a radially symmetric filter is employed. In this paper we limit ourselves to separable filters, although similar results can be obtained from radially symmetric filters.

2.2 Practical Reconstruction Methods and Errors

The ideal reconstruction process from Equation 3 cannot be realized in practice, because the *Sinc* function is infinite in extent. An obvious solution is to truncate the function to include only *L* integer valued points:

$$f_R(x, M) = \sum_{k=-M}^{M} S(x, k, h) f_s(kh) \tag{6}$$

where *M = L div 2*. Such a filter is called an *FIR* or *finite impulse-response* filter of order *L*. However, this causes aliasing in the frequency spectrum, thereby altering the filter's frequency characteristics. We achieve truncation by multiplying the infinite filter by a spatially limited rectangular window. In the frequency domain this is equivalent to convolving a *Sinc* (the Fourier transform of a rectangular window) with a box (the Fourier Transform of a *Sinc*). The resulting filter has frequencies beyond ω_c which have non-zero strength. Also, some frequencies below ω_c have an attenuated amplitude less than the amplitude of the ideal filter. The resulting error has been called *post-aliasing* by Mitchell and Netravalli [13]. Since we emphasize spatial methods in this paper, we shall call this error *truncation error* (Equation 7), denoted by e_t. Note that this error also includes the data the filter is applied against.

$$e_t(f_s, x, M, h) = \sum_{|k| > M}^{\infty} S(x, k, h) f_s(kh) \tag{7}$$

The truncation error manifests itself as *blurring* (due to attenuation) and *jaggies (*due to including frequencies beyond ω_c) in an image.

To minimize the impact of truncation, a *window function* besides a rectangle can be used. Among the most popular window functions are: *triangular (Bartlett)*, *Hanning*, *Hamming*, and *Kaiser* [15]. These window functions alter the characteristics of the truncated frequency spectrum. The choice of one window function over another is a tradeoff between blurring and aliasing. However, since these filters differ from the *Sinc* function, a further source of error is introduced. We call it the *non-Sinc error.* It is not imperative that we only use a windowed *Sinc*. We can use any of the commonly used resampling filters [21].

If *NS* is the filter obtained from windowing, then the reconstruction function is now given by:

$$\tilde{f}_R(x, M) = \sum_{k = -M}^{M} NS(x, k, h) f_s(kh) \tag{8}$$

The non-*Sinc* error, denoted by e_{ns}, is the difference between the continuous function reconstructed by the truncated *Sinc* filter, and the one reconstructed by the *NS* filter:

$$e_{ns}(f_s, x, M, h) = \left| f_R(x, M) - \tilde{f}_R(x, M) \right| \tag{9}$$

Thus, the total error is the sum of the non-*Sinc* error and the truncation error:

$$e(f_s, x, M, h) = e_t(f_s, x, M, h) + e_{ns}(f_s, x, M, h) \tag{10}$$

2.3 Previous Work

The study of reconstruction errors has not received much attention in the graphics and image processing literature. In [17], Parker et al. compare the effectiveness of some resampling filters. However, they propose no metrics which can be used to judge the goodness of a resampling filter. Mitchell and Netravalli [13] first introduced the reconstruction metrics *pre-aliasing* and *post-aliasing*. Marschner and Lobb [12] further characterize post-aliasing artifacts; the proposed metrics, *smoothing* and *post-aliasing*, can help design a suitable filter. However, none of these measures are well suited for determining the accuracy of an interpolation or filtering scheme on a sampled dataset.

There exist many filter design methods which lie in either the frequency domain or the spatial domain. For example, in the image processing and graphics literature Keys [10], Max [11], Park and Schowengerdt [16] and Mitchell and Netravalli [13] use spatial methods to design resampling filters which satisfy certain functional properties (the existence of derivatives, etc.). All of these methods make assumptions about the interpolated function, and they deliver filters which perform well on smooth functions. On the other hand, Carlbom [2] uses frequency methods to design filters which are solutions of a non-linear optimization process. This yields a filter of finite length whose frequency response closely approximates the ideal filter response.

Among these efforts only [2] considers reconstruction error. However, this error is defined in the frequency domain and only measures the deviation of the frequency spectrum from the ideal spectrum (a box in frequency domain). Also, this metric is global in nature and does not provide error control on a point-wise basis. Moreover, this and other global frequency domain methods are not conducive to our idea of adapting the filter size to the resampling location and to local data characteristics.

Our method estimates the filter size for a given resampling location so we can interpolate to a desired level of accuracy efficiently. For example, a less expensive interpolation scheme can be used at some locations (e.g., near grid locations in source space). Similarly, a more expensive scheme is warranted at other locations (e.g. far from grid locations). In addition, we also determine the filter size from the complexity of the data at the resampling point. This gives us an efficient yet accurate resampling method.

There is a body of work dedicated to filtering in texture spaces which attempts to address the issues of error control and adaptivity. Fournier and Fiume [3] use spatial methods and a least square error (L^2 norm) estimate (with data included) to guide efficient and accurate anti-aliasing of textures. They also allow adaptive filtering in a manner to similar to *MIP maps*. However, their method is complicated and does not use memory space efficiently. Norton et al. [14] use a frequency domain approach to perform adaptive filtering with a simple box filter. They use a coarse measure of goodness to clamp all frequencies beyond a certain range. However, the adaptivity is not driven by any user defined error threshold, but is guided by an ad-hoc measure of the robustness of the filtering operation in the frequency domain.

Adaptive filtering in the frequency domain is also reported by Totsuka and Levoy [20] for 3-D volumes. Again, the adaptivity is not driven by an error threshold. Also, the technique requires that the filtering be conducted in the frequency domain, which requires transforming a large 3D dataset.

In summary, many of the past efforts neither provide any means of controlling the reconstruction error nor provide adaptive reconstruction of the continuous signal. The methods which provide error specification and control in the spatial domain are rather complex, while the methods that provide adaptivity are either inconvenient or available in the frequency domain only. This motivates our attempt to develop spatial domain methods which allow the specification of error bounds and allow the use of different filter lengths adapted to the resampling operation and local data complexity. Our method can be successfully employed in resampling operations and texture mapping. In the next section we provide the necessary theory and develop adaptive, spatial-domain methods.

3. Reconstruction Error Estimates

In this section we provide estimates of *truncation* error and the *non-Sinc* error. Researchers in mathematics and electrical engineering have been long concerned with the accuracy of sampling schemes. Some work has been done in both fields which can be applied to estimate the truncation error. We present some of appropriate results from Complex Analysis for the sake of completeness. Later we obtain bounds for the truncation error. The estimates for *non-Sinc* error are much more easier to obtain and we also present our estimates in this section.

3.1 Truncation Error Estimates

Equation 3 is the starting point for our effort. The true error can be computed for spatially limited signals especially, images and volumes. However, in the presence of a large number of sampled data points, the computation can be prohibitively expensive. From Equation 3, we can compute the function value at the resampling point ***R***, as follows

$$f_R(x) = \sum_{k = \text{Ð}\infty}^{\infty} \frac{\sin\frac{\pi}{h}(x \text{ Ð } kh)}{\frac{\pi}{h}(x \text{ Ð } kh)} f_S(kh) \tag{11}$$

Let the resampling point ***R*** lie in cell *n*, i.e., between sampled data points S_n and S_{n+1}. Also, let τ be the distance of ***R*** from S_n, in other words, $x=nh+\tau$. The truncation error is obtained from Equation 11 by dropping $L=2M+1$ terms of the above infinite summation.

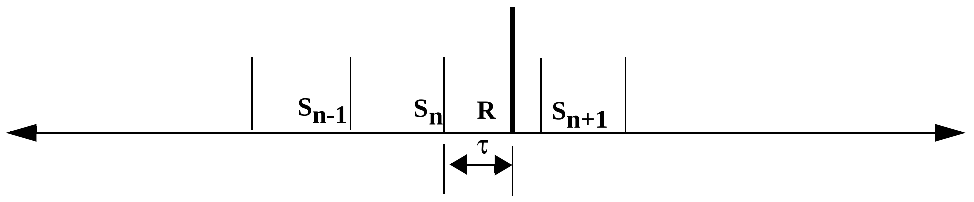


FIGURE 1. Interpolation Scheme for sampled dataset $\{S_i, i=1,...,L\}$. The resampling point R lies a distance τ away from the sampled data S_n. The point R is generated as a result of transforming an image or a volume, supersampling pixel values or during texture mapping.

Thus, the truncation error, $e_t(f_s,x,h,M)$ is given in Equation 12.

$$e_t(f_s, x, h, M) \;=\; f_R(x) - \sum_{k=-M}^{M} \frac{\sin\frac{\pi}{h}(nh + \tau - kh)}{\frac{\pi}{h}(nh + \tau - kh)} f_s(kh) \tag{12}$$

A change of variable (let $m = n - k$) allows one to rewrite the above equation in a more convenient form as shown in Equation 13.

$$e_t(f_s, n, \tau, h, M) \;=\; \sum_{|m| > M}^{\infty} \frac{\sin\frac{\pi}{h}(mh + \tau)}{\frac{\pi}{h}(mh + \tau)} f_s((n - m)h) \tag{13}$$

Expanding the sinusoidal term inside the summation and letting $\sin \pi m = 0$ and $\cos \pi m = (-1)^m$ for all values of *m,* we get Equation 14.

$$e_t(f_s, n, \tau, h, M) \;=\; \frac{\sin\frac{\tau}{h}}{\frac{\pi}{h}} \sum_{|m| > M}^{\infty} \frac{(-1)^m}{(mh + \tau)} f_s((n - m)h) \tag{14}$$

We now make two important observations from Equation 14.

Observation 1: The truncation error depends on the location of the resampling point ***R***. If ***R*** is located at the center of a grid cell in the source space, i.e. τ=0.5, it attains its maximum value and drops off to zero as one moves closer to the sampled data locations (τ=0 or τ=1). In [17] similar observations are made. However, these observations were made in the frequency domain. Also, the error was not quantified. An important implication from this observation is that one can use filters of different lengths depending on the location of the resampling point. In the next section we shall provide evidence to illustrate this fact using 1D and 2D examples.

Observation 2: For large values of *m*, the terms in the infinite sum cancel each other if the function is smooth and does not change drastically in small or reasonably sized neighborhoods. The implications of this observation is that we can effectively limit ourselves to reasonably sized neighborhoods. This observation allows in even more efficient implementations of the filtering operation since even smaller length filters can be used.

It is easy to determine a bound from Equation 14 and is listed in Equation 15.

$$e(f_s, n, \tau, h, M) \le \frac{h\left|\sin\frac{\tau}{h}\right|}{\pi} \sum_{m > |M|}^{\infty} \frac{|f_s((n - m)h)|}{|(mh + \tau)|} \tag{15}$$

This is still not a practical bound to use, since all sampled data points have to be considered to compute the error bound. Moreover, this bound overestimates the error since it does not take into account the oscillating nature of the *Sinc* function. We now consider some tractable error bounds that can be used in practice. The error bound for this infinite sum can be found by resorting to complex analysis [24]. However, before we state the relevant results we discuss an important idea of *frequency guards*.

Frequency guard bands allow the approximation of the infinite sum in Equation 15 by the integral of an analytic function which exists on the real line. A frequency guard of width r, $0<r<1$ implies that there exists no frequency in the signal beyond $r\omega_c$, where ω_c is the cut-off frequency. This is a stricter requirement than just bandlimitedness, but not too restrictive. The frequency guard band can be found by determining the ratio of the maximum significant frequency of the spectrum and the cutoff frequency. However for most graphics application it is sufficient to use very crude estimates. We shall address this issue further in Section 4 when we discuss results and implementation results. Thus, if $F(\omega)$ is the Fourier spectrum of the continuous signal f, the restrictions on the spectrum now are defined as follows

$$\int_{-r\omega_c}^{r\omega_c} |F(\omega)|^2 < \infty, \quad \int_{-r\omega_c}^{r\omega_c} |F(\omega)| < \infty \tag{16}$$

The methodology used in [5][24] can be applied to determining the error of any polynomial approximation scheme. In fact, such a methodology has been used to estimate the error of Legendre and Hermite Polynomial interpolation. The mainstays of this approach are theory of analytic functions and the application of *Cauchy's Integral Formula and Residue Theorem* [18].

3.1.1 Results from Complex Analysis

Inherent to this approach is the use of a contour ***C*** or a directional closed path in the complex plane. Such a contour is shown in Figure 2. Thus, the contour (dark counterclockwise path) in our example is a rectangle of size $d \times L+1$ centered at the resampling point ***R***. The contour is larger than the filter by a distance of unit h (or a distance of $h/2$ on both sides). This allows for all the $2M+1$ sampled data points required for interpolation to be inside the contour. The intuition behind using the contour is that it generalizes the process of filtering to the complex plane. The contour ***C*** is the equivalent to the filter used to reconstruct real valued signals and could be of any shape. However, it needs to be closed and directional [18]. The rectangle is normally chosen given its simplicity.

We now state **Theorem 1** which provides us with a way to compute the error bound for filtering interpolation schemes. This contour integration used in **Theorem 1** once again generalizes the infinite sum of the contributions from sampled data points to the reconstructed value. In this section discussion is to applicable to the space of continuous functions and we therefore use the symbol everywhere. Many of the concepts expounded herein can be found in any standard text on Complex Analysis including [18].

Theorem 1: Let ***C*** be the contour (shown in Figure 2) over the domain ***D*** (a subset of the complex plane) and let function f be analytic everywhere therein (i.e., it is defined every where and all derivatives exist). Let $G(z) = \sin(\pi z/h)$, where z is a point in the complex plane.

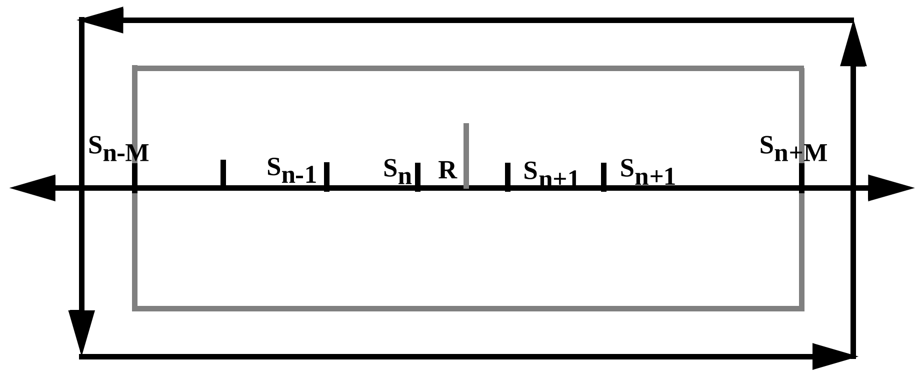


FIGURE 2. Contour Integration applied to the computation of error bounds. The rectangle with shaded edges is equivalent to the actual filter with half length M, while the rectangle with darker edges is the contour ***C***.

Then, $f_1(z) = f(z)/G(z)$ is analytic everywhere except at $z = kh$, $n\text{-}M\text{-}1 < k < n\text{+}M\text{+}1$, (sampled data points) where the function $G(z)$ evaluates to zero. If the function is resampled at $z=x$ on the real line, then the truncation error there is given by

$$e_t(f, x, k, h, M) = \frac{G(x)}{2\pi i}\oint_C \frac{f(z)}{G(z)(z \text{ Ð } x)}dz \tag{17}$$

The integral in Equation 17 is performed over the contour ***C***. By using other functions for $G(z)$ we can determine the error for various function approximations.

Proof: The starting point of this proof is the Cauchy Integral Formula [18] which allows us to compute the function $f_1(x)$ follows

$$\frac{f(x)}{\sin\pi\frac{x}{h}} = \frac{1}{2\pi i}\oint_C \frac{f(z)}{\sin\left(\pi\frac{z}{h}\right)(z \text{ Ð } x)}dz \tag{18}$$

As mentioned earlier, the integral is performed on a contour ***C*** (Figure 2). To actually evaluate the integral, the contour ***C***, is altered since the function G(z) evaluates to zero at all sampled data points, $z = kh$, $n\text{-}M\text{-}1 < k < n\text{+}M\text{+}1$. Thus at these points *poles* are introduced and the new contour ***CÕ*** now skirts around these points (Figure 2). The integration over the contour ***CÕ*** is divided as follows:

- the integrals evaluated on clockwise contours around sampled point S_k, $n\text{-}M\text{-}1 < k < n\text{+}M\text{+}1$, each of which is denoted by $Q(S_k)$,
- the integrals along both the straight lines leading to and from the poles; these cancel each other,
- the integrals along the horizontal contours $\boldsymbol{C_1}$ and $\boldsymbol{C_3}$ and
- the integrals along the vertical contours $\boldsymbol{C_2}$ and $\boldsymbol{C_4}$.

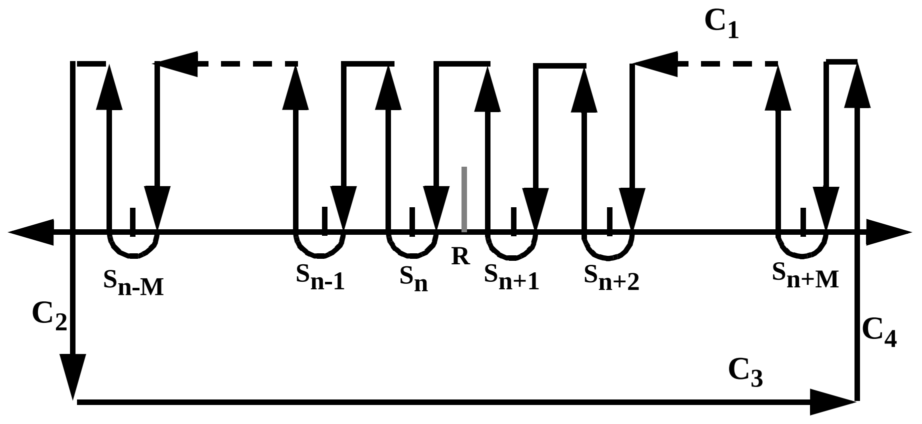


FIGURE 3. Contour Integration of Equation 20. The top portion of the contour in Figure 2 is modified to include the poles created by the sampling points. The new contour now includes, the segmented portion on the top (*C_1*), a complete horizontal path (*C_3*), full vertical paths (*C_2* and *C_4*) and *2M+1* clockwise paths leading to and from the sampled points.

The residues or the quantities *$Q(S_k)$* at each one of the poles is evaluated in the limiting case and is given by

$$Q(S_k) = \frac{f(kh)}{\left.\frac{d}{dz}\left(\sin\pi\frac{z}{h}\right)\right|_{z = kh}(kh\,\text{Ð}\,x)} = \frac{(\text{Ð}1)^k f(kh)}{\frac{\pi}{h}(kh\,\text{Ð}\,x)} \tag{19}$$

Thus the right-hand side in Equation 18 can be written as

$$\frac{f(x)}{\sin\pi\frac{x}{h}} = \frac{1}{2\pi i}\oint_{(C_1+C_2+C_3+C_4)}\left(\frac{f(z)}{\sin\left(\pi\frac{z}{h}\right)(z\,\text{Ð}\,x)}\right)dz + \sum_{k\in Ind} Q(S_k) \tag{20}$$

$$Ind(M) = \{k|n\text{Ð}M\,\text{Ð}\,1<k<n+M+1\}$$

The second term in Equation 20 corresponds to the first *M* terms of the infinite reconstruction sum (Equation 14) and hence it can be inferred that the first term in is the truncation error. Hence, it is true that the truncation error is given by

$$e_t(f, x, k, h, M) = \frac{G(x)}{2\pi i}\oint_{C'}\frac{f(z)}{\sin\left(\pi\frac{z}{h}\right)(z\,\text{Ð}\,x)}dz \tag{21}$$

$$C' = C_1 + C_2 + C_3 + C_4$$

3.1.2 Determination of Truncation Error Estimates from Complex Analysis

Having shown how the truncation error can be determined, we can now set about to obtain the bounds for a bandlimited function. The error bound can be obtained in terms of either the maximum of the function value Max_f, or the spectral energy of the function, E_f. Since, all signals under consideration have finite energy, are real and available as sampled datasets, E_f can be simply determined by Equation 22. This relationship is a direct consequence of the Cardinal Series Expansion [19]. All the sampled data points are included in the summation of Equation 22.

$$E_f = \sqrt{\frac{h\sum_k |f_s(k)|^2}{2\pi}} \tag{22}$$

We now state and provide a sketch of the proof of **Theorem 2** which expresses the truncation error bound in terms of the spectral energy of the function.

Theorem 2: The truncation error $e_t(f_s,x,k,h,M)$ in terms of the total energy of the signal is bounded from above by the quantity

$$e_t(f_s, x, k, h, M) \leq \frac{2\sqrt{\frac{rE_f}{h}}\left|\sin\frac{\pi x}{h}\right|}{\pi^2(1 \text{ Ð } r)N} \tag{23}$$

Proof: To obtain a proof of this theorem we consider each part of the ***C'*** which has a non-zero contour integral. Along the horizontal parts of the contour, $\boldsymbol{C_1}$ and $\boldsymbol{C_3}$, the contribution to the integral in Equation 21 is zero. We can show this by first considering the denominator *sin(πz/h)*, where *z=x'+jy'* is any point in the complex plane. It is true that

$$\left|\sin\frac{\pi(x' + jy')}{h}\right| \geq \cosh\frac{\pi}{h}|y'| \geq \sinh\frac{\pi}{h}|y'| \tag{24}$$

The numerator in Equation 21 for all contours is bounded by $E_f \cosh(\pi r|y'|/h)$ [5]. Since the *cosh* function grows faster than the *sinh* function [18] for the same argument, on contours $\boldsymbol{C_1}$ and $\boldsymbol{C_3}$ in the limiting case the numerator becomes zero. Now let us consider contours $\boldsymbol{C_2}$ and $\boldsymbol{C_4}$. The contour $\boldsymbol{C_2}$ lies along line *x=h(n- M -1/2)* and thus

$$\left|\frac{f(z)}{\sin\left(\pi\frac{z}{h}\right)(z \text{ Ð } x)}\right| \leq \frac{E_f\cosh\frac{\pi}{h}r|y'|}{\sqrt{\left(x' \text{ Ð } h\left(n \text{ Ð } M \text{ Ð } \frac{1}{2}\right)\right)^2 + y'^2}\cosh\frac{\pi}{h}|y'|} \tag{25}$$

After further simplification the integrand along contour $\boldsymbol{C_2}$ is now bounded by

$$\left|\frac{f(z)}{\sin\left(\pi\frac{z}{h}\right)(z - x)}\right| \le \frac{E_f \cosh\frac{\pi}{h}ry'}{\sqrt{(Mh)^2 + y^2}\cosh\frac{\pi}{h}y'} \qquad (26)$$

It can be shown that the integrand along $\boldsymbol{C_4}$ is also bounded by the same quantity. Recognizing that *cosh(z)* grows faster than e^z and then evaluating the integrals for both remaining contours we get the required error bound

$$e_t(f_s, x, k, h, M) \le \frac{4E_f\left|\sin\frac{\pi x}{h}\right|}{\pi^2 N(1 - r)} \qquad (27)$$

Thus we are able to express the truncation error bound in terms of the energy of a function and the frequency guard *r*. If once again *x=nh+t*, i.e, it lies in the cell *n* of the source grid, we can replace *x* with τ (refer Equation 14). We now state another theorem, **Theorem 3**, which expresses the error bound in terms of the maximum of a function. The proof for this theorem is very similar to **Theorem 2** and we are therefore not including it for sake of brevity.

Theorem 3: The truncation error $e_t(f_s,x,k,h,M)$ in terms of the maximum value of a function, Max_f is bounded from above by the quantity

$$e(f_s, x, k, h, M) \le \frac{Max_f\left|\sin\frac{\pi x}{h}\right|}{\pi M\cos\frac{r\pi}{2}} \qquad (28)$$

We now characterize the error that arises from the use of a function different from a *Sinc* function.

3.2 Non-Sinc Error

The use of the truncated sinc induces visual artifacts, namely smoothing (blurring) and aliasing (jaggies). Therefore, another function suitably modulated by a window can be used. We however need to estimate the error that arises from the use of windowed function. Once again we can either use the spectral energy or the maximum value of the function. Using Parseval's Theorem [15] and Equation 22 one can write

$$e_s(f_s, x, n, h, M) \le E_f(M)\sqrt{\frac{1}{2\pi}\int_{-M}^{M}|S(t, 0, h) - NS(t, 0, h)|^2 dt} \qquad (29)$$

The integral computes the difference between the two functions in the L^2 norm space. The quantity $E_f(M)$ is the energy of the signal in a *2M+1* sized neighborhood around the resampling point $\boldsymbol{R}$. Since the filters are space invariant one evaluate the filters when placed at *x=0* for sake of convenience. One can similarly define a bound including the maximum value of the function (Equation 30). The quantity $Max_f(M)$ is the maximum of the function values in a *2M+1* neighborhood. Once again we are determining the difference in the areas of the two filter functions.

$$e_s(f_s, x, n, h, M) \le Max_f(M) \int_{ĐM}^{M} |S(x, h)\ Đ\ NS(x, h)|\, dx \tag{30}$$

Any filter could be used instead of a windowed-Sinc filter. However, we shall limit ourselves to windowed Sinc filters in this filter. An example of a window which we also employ in our paper is the Hamming window (Equation 31). The significant aspect of this function is that it falls gradually to zero at the corners of the window and hence reduces the impact of aliasing caused through the use of the rectangular window.

$$w_H(x) = \begin{cases} 0.54 + 0.46\cos\left(\dfrac{\pi x}{M+1}\right) & |x| < M + 1 \\ 0 & otherwise \end{cases} \tag{31}$$

An appropriate filter can be obtained by multiplying this function with the *Sinc* to obtain a modified *Sinc* function. For multiple dimensions one can use a product of two 1-D window functions to get a 2-D function. Thus a 2D Hamming window would look like

$$W_H(x, y) = w_H(x) w_H(y) \tag{32}$$

In this section we described the errors that arise from filtering operations. In the Section 4 we these measures to predict reconstruction errors that arise from representative resampling operations and then show how they can be used to perform adaptive reconstruction.

4. Results

In this section we first test the validity of the bounds on 1-D signals. We also illustrate the usefulness and viability of adaptive schemes. In the latter part of this section we implement our schemes for a particular 2D resampling schemes, namely rotation.

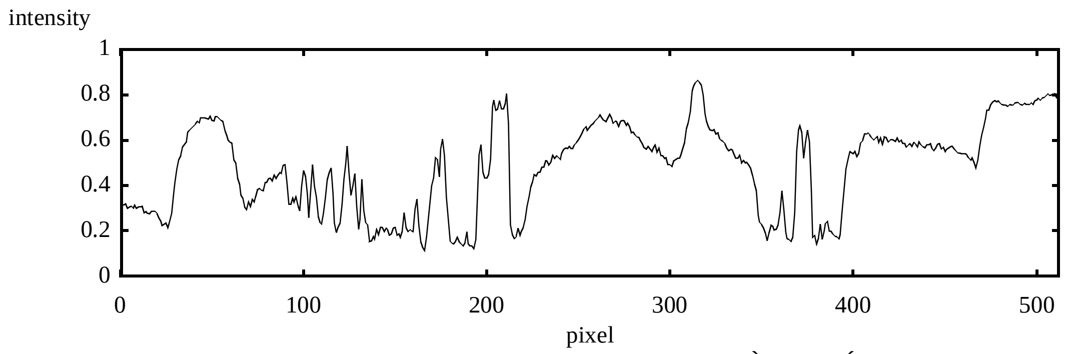


FIGURE 4. Intensity values for scanline 300 of the 512x512 ÒLennaÓ image. The intensity values are normalized to the maximum possible pixel value, namely 255.

4.1 Accurate and Adaptive Reconstruction of 1D Signals

We consider a representative resampling scheme that frequently arises in computer graphics. In this section we distinguish between resampling schemes and reconstruction or filtering operations. Resampling schemes provide the points where the functions are reconstructed. The resampling scheme of choice occurs when a 2D image or a 3D volume is subjected to affine transformation (including scaling), or during texture mapping, or during ray-casting. The signal is resampled onto another grid and the number of data points can change as a result of scaling inherent to the resampling operations.

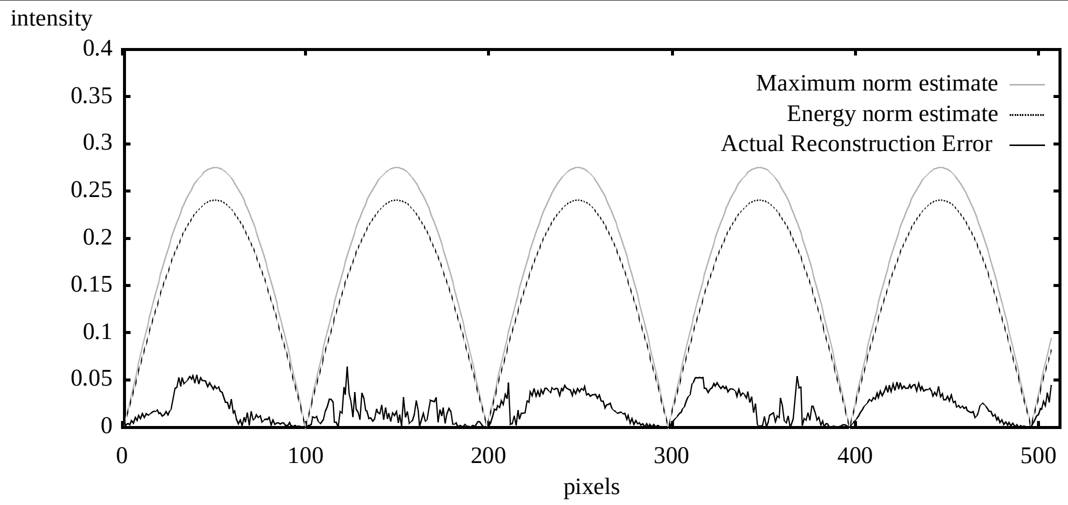


FIGURE 5. Comparison of Error Estimates. The actual error and the estimates from Equation 23 (energy) and Equation 28 (maximum) are plotted. Both estimates bound the actual error well, the energy estimate being closer.

In Figure 4 we consider a 1D signal obtained from row 300 of the Lenna image (Figure 9). It is worth noting that the signal under consideration has very small energy content. One can estimate the value of the frequency guard, *r*, by simply computing the first few Fourier coefficients above a user defined threshold. It was observed that the value of the guard was usually less than 0.1 for all images considered for this work. In other words, most of the energy of the function is characterized by the first one-tenth of the Fourier coefficients. The more accurately one measures the frequency guards, the better are the estimates. However, even coarse estimates can suffice for many signals and resampling situations in computer graphics.

The actual error from Equation 14, the error estimates from Equation 23 (using energy) and Equation 28 (using maximum values) are determined, when the signal is resampled onto a new grid (Figure 5). The function is reconstructed at $x_k=x_0+0.99*k$, where x_0 is the location of the first row pixel and k is an integer. The estimates from Equation 28 are looser and we found the energy estimates closer to the actual error for many signals and resampling schemes.

Here we can actually see evidence for **Observation** 1 made in Section 3.1. The error behaves in a sinusoidal manner for the representative resampling scheme. If a larger resampling frequency is chosen, the periodicity of the error is higher. One can readily conclude that the filters of the same length need not be used everywhere during the resampling operation. To use different filters at different resampling positions, one can use the error estimates of Equation 23 and Equation 28. For instance one can set the point wise error to ε for all x along the length of the signal. The

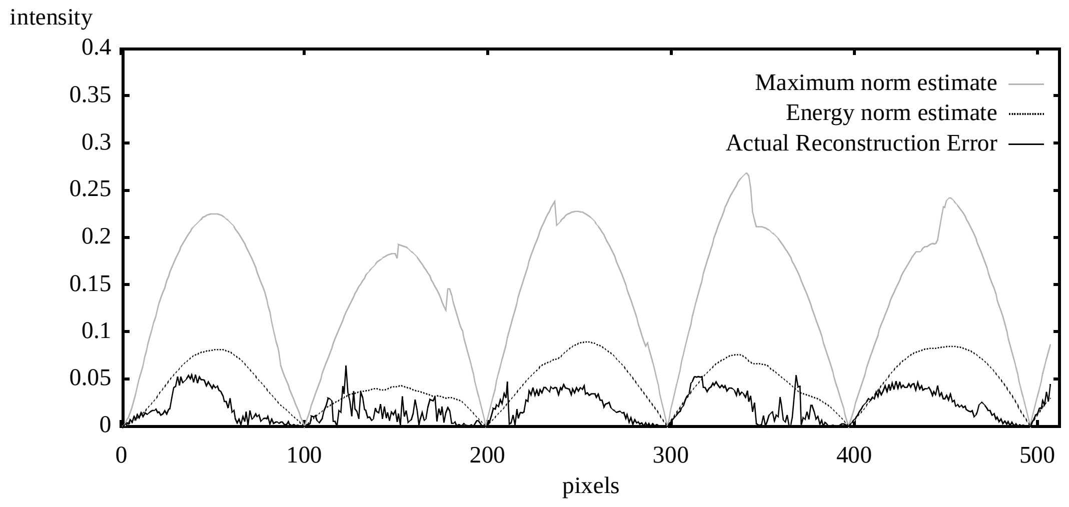


FIGURE 6. Comparison of Error Estimates. The actual error and the estimates from Equation 23 (energy) and Equation 28 (maximum) are plotted when a neighborhood of size 25 is used to compute the maximum or the energy. The estimate based on energy follows the actual error very closely.

required filter length at resampling point *x* can be determined from the computation listed in Equation 33.

$$M(x) = \frac{2\sqrt{\frac{rE_f}{h}}\left|\sin\frac{\pi x}{h}\right|}{\pi^2(1 - r)\varepsilon} \tag{33}$$

Figure 7 shows the minimum filter length at all points required for the resampling of the signal of Figure 4 to obtain an user defined accuracy of ε*=0.02*. The maximum filter length employed for reconstruction is *27* (*=13*2+1*). We call this filtering scheme *position-adaptive*, since the size of the filter is influenced only by the position of the resampling point. If frequency domain methods of filter design are used it is generally the case that filters of even greater length are obtained from the design process. Also, it is not certain that the desired level of accuracy is guaranteed from the application of such a filter.

In Figure 5 we used the total energy or the global maximum value of the function to compute the bounds. The estimated bounds are conservative. Taking into account the rapid decay of the *Sinc* function as one moves away from the resampling point, it might be useful to consider the energy or maximum of a function over a neighborhood of somewhat significant size as stated in **Observation 2** of Section 3.1. The problem is now reduced to determining a window of appropriate size which is suitable for a given signal. This can be determined easily from the estimates of the bounds itself. One can set the minimum error of resampling ε_{min} that can possibly arise during resampling. Then one can simply calculate the neighborhood size M_e by using either Equation 23 or Equation 28. We use energy estimates to determine the optimum neighborhood size.

$$M_e = \frac{2\sqrt{\frac{rE_f}{h}}\left|\sin\frac{\pi}{2h}\right|}{\pi^2(1 \text{ Ð } r)\varepsilon_{min}} \tag{34}$$

The value of τ is set 0.5 to cover all possible resampling positions. Now one can determine the maximum or the energy over a neighborhood of this size. Equation 33 then becomes

$$M(x) = \frac{2\sqrt{\frac{rE_f(M_e)}{h}}\left|\sin\frac{\pi\tau}{h}\right|}{\pi^2(1 \text{ Ð } r)\varepsilon} \tag{35}$$

A pre-processing step is now required which computes the energy or the maximum of the function value over neighborhoods. In Figure 6 we plot the true error and the estimates using energy and maximum values over a neighborhood of size M_e*=25*. The estimate based on energy is now very close to the actual error. Figure 7 also shows the sizes of the filters used when a neighborhood of size *25* is used for the signal of Figure 4. The maximum and average size of the filters are significantly lowered. The use of smaller neighborhoods yields smaller filters and hence savings in reconstruction time We call filtering scheme *data-adaptive*.

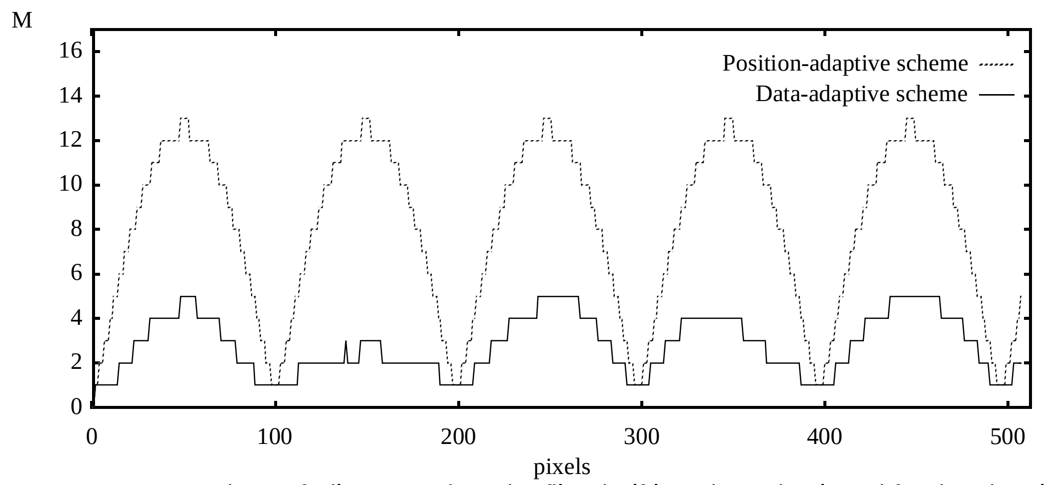


FIGURE 7. Comparison of Filter Lengths. The filter half length, M, is plotted for the signal of Figure 4. The solid curve provides the length of the filter at the resampling points when the position independent scheme is used. The dashed curve provides the filter lengths for the data adaptive scheme. The lengths were obtained using energy estimates.

The behavior of the bounds is different when the same signal is subjected to a plain translation. This form of resampling occurs during shearing transformations, image registration, optimized versions of volumetric ray-casting, etc. The error does not behave in the same periodic manner as before since the displacement along the source space grid is constant. The truncation error is dictated more by the data complexity rather than the position of the resampling point. We however do not include any plots to illustrate the behavior of error for the sake of brevity.

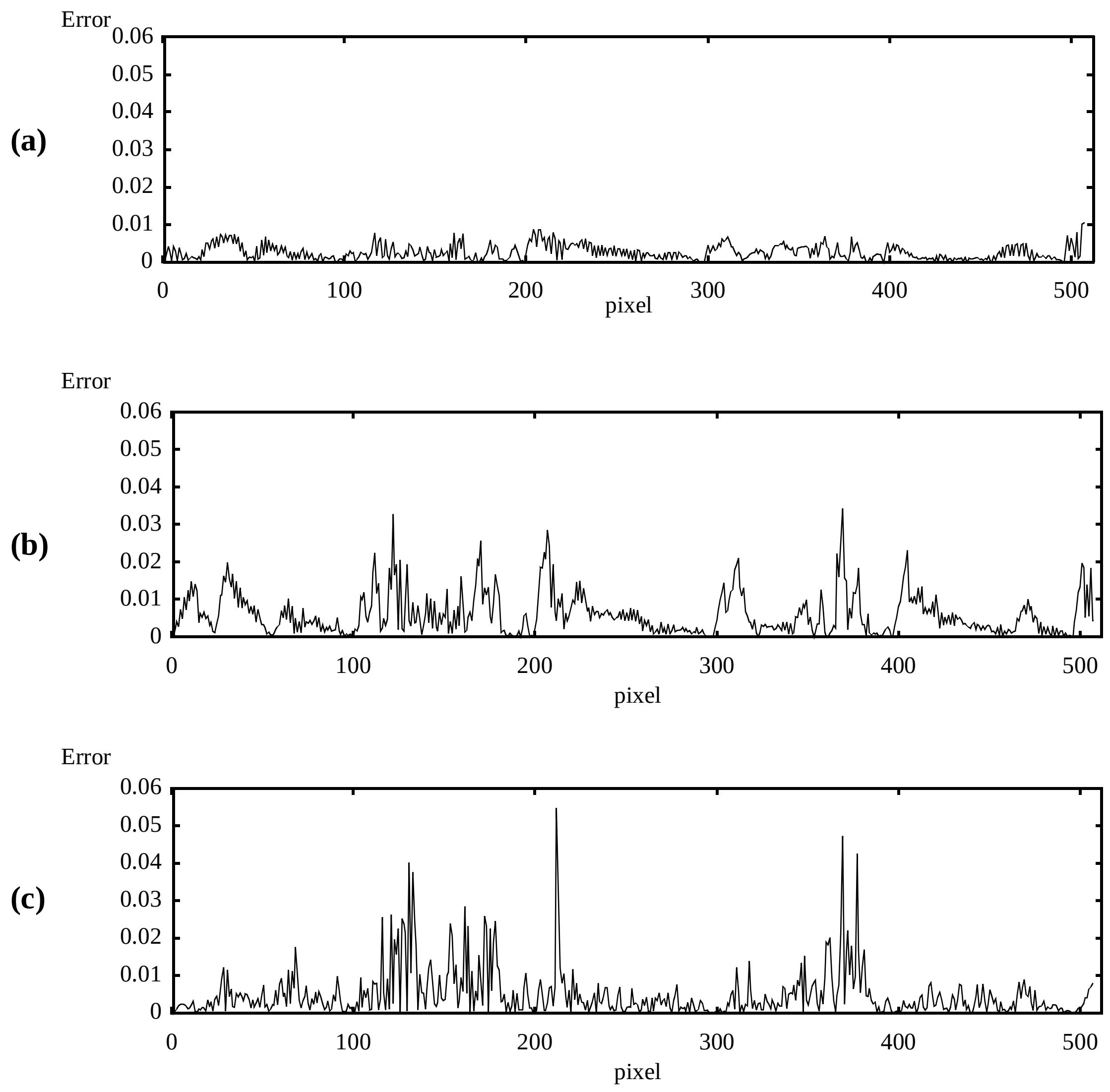


FIGURE 8. Effectiveness of Reconstruction Schemes. (a) Error obtained from position-adaptive scheme. (b) Error from data-adaptive scheme. (c) Error from a cubic convolution filter.

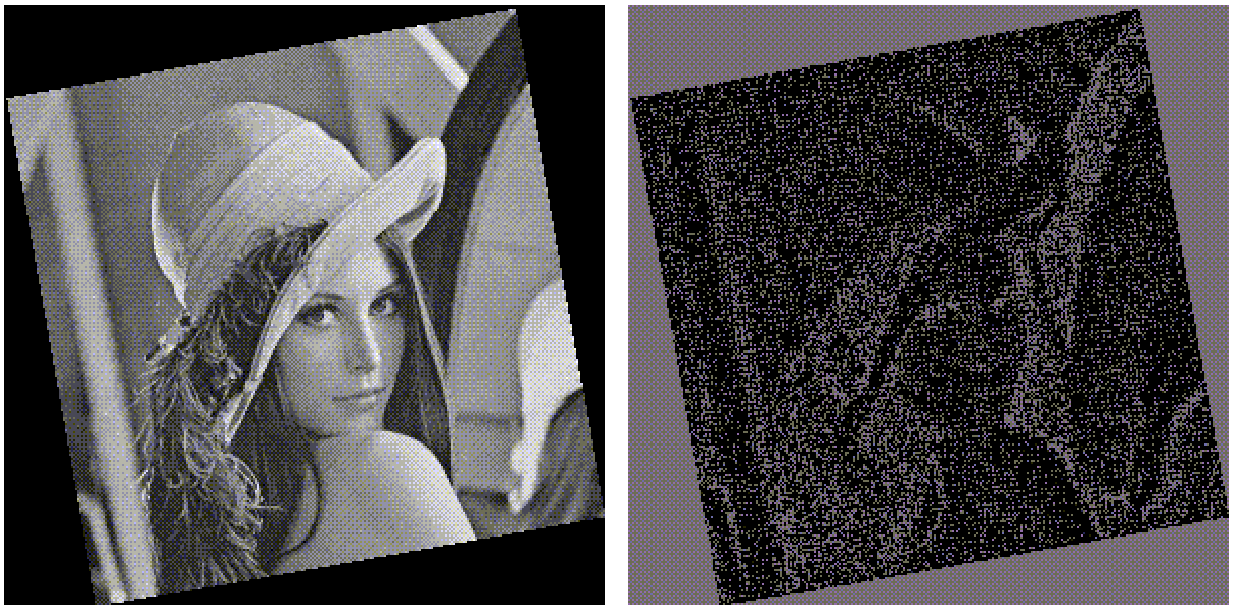
In all our experiments we used the rectangle window function and hence did not incur the non-Sinc error. If another window function was employed the total error would no longer be the same. Although the total error no longer reaches zero, the periodic nature of the resampling error will still remain unchanged. Even though a non-rectangular window is used, the adaptive scheme can still be based on the truncation error.

Finally, to provide a basis of comparison we reconstructed the signal of Figure 4 with an

- infinitely long *Sinc* filter (no truncation),
- truncated *Sinc* filter whose length is not dependent on the data complexity (position-adaptive),
- truncated *Sinc* filter whose length is influenced by the data complexity (data-adaptive)

- cubic spline filter described in [13].

We then determined the errors of reconstruction by computing the difference between the perfectly reconstructed function (using the infinitely long *Sinc* filter) and the functions reconstructed using the non-adaptive, adaptive and cubic spline filters. We also set a threshold of *0.02* for both filtering schemes as before. In Figure 8 we plot the reconstruction errors as measured against the perfectly reconstructed signal. The position-adaptive scheme always delivered reconstruction to the desired level of accuracy (*=0.02*). The data-adaptive scheme for most of the signal fared well. However, in regions of rapid changes in function value it underestimated the error. The cubic convolution scheme on the other hand was not sensitive to either the position of the resampling location or the data complexity. The error of reconstruction was also sometimes much larger than the desired level of *0.02*. If the desired level of accuracy is reduced to 0.002, both adaptive schemes fare well, while the performance of the cubic convolution filter remains the same. Having shown the effectiveness of our error measures we now provide 2D examples.



(a) (b)

FIGURE 9. (a) Lenna rotated by 10° and scaled down by half. Reconstruction was done with the position-adaptive method with ε=0.007. (b) A difference image between the position-adaptive and the data-adaptive methods. Differences were exaggerated to make the pattern visible.

4.2 Accurate and Adaptive Reconstruction of 2D Images

We considered a few two dimensional images to show the usefulness of the methods developed here. Equation 23 and Equation 28 now simply become

$$e(f, x, k, h, M) \le \frac{16E_f\left|\sin\frac{\pi\tau_x}{h}\right|\left|\sin\frac{\pi\tau_y}{h}\right|}{\pi^4 N^2 (1 \text{ Ð } r_x)(1 \text{ Ð } r_y)} \tag{36}$$

$$e(f, x, k, h, M) \leq \frac{Max_f \sin\pi\frac{x}{h}\sin\pi\frac{y}{h}}{\pi^2 M^2 \cos\frac{r\pi}{2}\cos\frac{r\pi}{2}} \tag{37}$$

The quantities r_x and r_y are frequency guards for each of the dimensions in the frequency domain and can be crudely estimated from the FFT. The energy and maximum values are now determined for all points in the image.

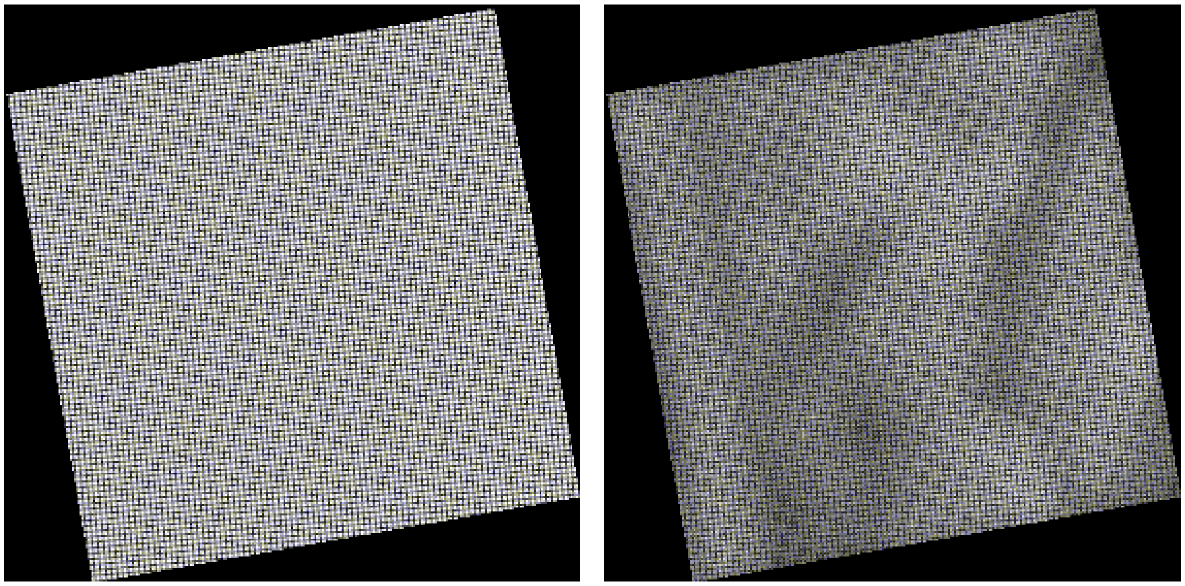


FIGURE 10. Filter size used for generating Figure 9 for position-adaptive filters (left) and for data-adaptive method (right). Bright values stand for larger filters sizes.

The measures and filtering schemes can be adapted to images and volumes very easily. The position-adaptive scheme does not require any pre-processing, while the data-adaptive scheme requires that the energy or the maximum of the underlying function be determined over a neighborhood. By specifying the minimum desired error one can use a derivative of Equation 37 (and similar to Equation 34) to determine the size of neighborhood required to achieve the desired error of ε_{min}. The local energy and maximum values are then stored for each pixel (2D image) or voxel (3D volume). At each resampling point, filter size is then determined by using the error estimates and applied in the 2D or 3D neighborhood. We also employ the 2D *Hamming* window to obtain images of higher quality.

Figure 9(a) shows the image of Lenna rotated by 10° and scaled by half along both dimensions while guarantying error threshold ε=0.007. We employed a very simple resampling scheme unlike the more complicated shearing schemes used in image manipulation packages. A bounding box is first found and all pixels within the bounding box are scanned and mapped back to the source space of the original image. Figure 9(b) provides a comparison between the error in the data-adaptive and the position-adaptive schemes. One can see that in darker areas the difference is larger (and the difference image is brighter).

TABLE 1. Half-size of filters used in the reconstruction of Figure 9.

Method	Minimum	Maximum	Average
Position-Adaptive	1	11	6.75
Data-Adaptive	1	5	2.33
Cubic Convolution	2.5	2.5	2.5

Figure 10 shows the different filter sizes used for the rotation in the form of a gray scale image. The difference in the filter lengths at various resampling points is determined and assigned suitable gray-scale values where white represent larger filter sizes. As evident from Figure 10(a), the filter size changes in a periodic sinusoidal fashion. Also, the filter size adapts to the data complexity as shown in Figure 10(b). For instance, the dark areas around the hat are reconstructed with smaller length filters. The position of the resampling point still modulates the filter size. Table I provides a comparison of the filter sizes for the position-adaptive, data-adaptive, and the traditional cubic convolution filter [13].

We provide another example of an image obtained from a fluid dynamics simulation. The image in Figure 11(a) is rotated by an angle of 30° and scaled by a factor of 0.75 along both dimensions. The result is shown in Figure 11(b) for a threshold of ε=0.007. Filters of small length are employed around the boundaries as shown in Figure 11(c). It is worthy to note that one can achieve good boundary reconstruction with resorting to sophisticated resampling schemes. If a larger threshold of ε=0.02 is used, the adaptive scheme performs even better. The filter lengths used are even smaller as shown in Figure 11(d). Table II provides a comparison between the different filtering schemes for the fluid dynamics simulation image.‘

TABLE 2. Half-size of filters used in the reconstruction of Figure 11.

Method	Minimum	Maximum	Average
Position-Adaptive	1	12	7.15
Data-Adaptive	1	5	1.96
Cubic Convolution	2.5	2.5	2.5

Since average filter size for the data-adaptive method is lower than the one used by common high quality reconstruction methods performance achieved is comparable. When the position-adaptive method is used, average filter size increases, however, the desired error bound is always guaranteed.

5. Conclusions

We developed a new approach to the characterization and measurement of reconstruction error. Our method, based on spatial domain error analysis, uses approximation theory to develop error bounds for reconstruction. We provide an efficient way to guarantee an error bound at every point by varying filter size. In addition, we support position-adaptive and data-adaptive reconstruction

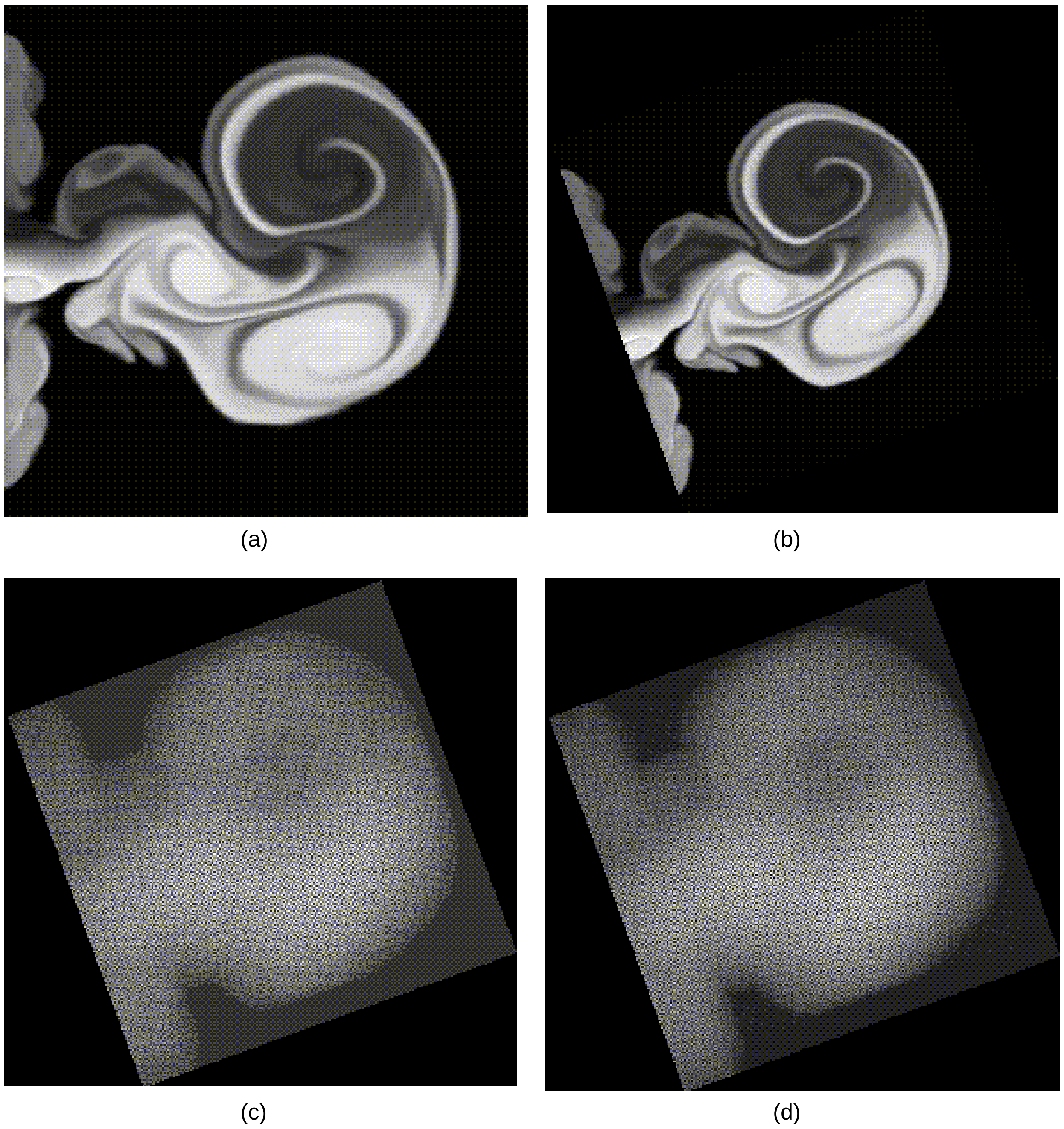


FIGURE 11. (a) An image from a fluid dynamics simulation. (b) The same image reconstructed by the data-adaptive method where $\varepsilon = 0.02$, rotate by 30°, and scaled by a factor of 0.75. (c) Filter size used for $\varepsilon = 0.007$. (d) Filter size used for $\varepsilon = 0.02$.

which adjust filter size to the location of reconstruction and the data complexity. Performing accurate reconstruction can potentially shift the burden from resampling to reconstruction thus allowing the use of simpler resampling techniques in many computer graphics applications such as image processing, volume rendering, and texture mapping. Our methods provide the user with a powerful tool for achieving any desired image quality while incurring space and computation cost that is comparable to existing methods.

Acknowledgments

This work was partially supported by National Science Foundation Grant CCR-9211288, and by the Advanced Research Projects Agency Contract DABT63-C-0056. We would like to thank the following individuals at The Ohio State University - Yair Kurzion for reading the manuscript and for offering useful suggestions, Prof. Ashok Krishnamurthy, Department of Electrical Engineering, for suggesting that we look at spatial methods, Prof. Bhavik Bhakshi, Department of Chemical Engineering, for discussions on functional representation of sampled datasets, errors and adaptive methods, Prof. Wayne Carlson, Advanced Computing Center for the Arts and Design and Prof. Don Stredney, The Ohio SuperComputer Center for the generous use of various resources.

6. References

[1] Bier T. and Neely S., “Feature Based Image Metamorphosis”, *Proceedings of SIGGRAPH ‘92, Computer Graphics*, 26, (2):35-42, July 1992.

[2] Carlbom I., “Optimal Filter Design for Volume Reconstruction and Visualization”, *Proceedings of Visualization ‘93*, IEEE CS Press, pp. 54-61, 1993.

[3] Fournier A. and Fiume E., “Constant-Time Filtering with Space-Variant Kernels”, Computer Graphics, 22(4):229-238, August 1988.

[4] He T., Wang S., and Kaufman A., “Wavelet-Based Volume Morphing”, *Proceedings of Visualization’94*, IEEE Computer Society Press, 1994, pp. 85-92.

[5] Helm H.D., Thomas, J.B., “Truncation Error of Sampling Theorems”, *Proceedings of the IRE*, Vol. 50, pp. 179-184, February 1962.

[6] Hughes J.F., “Scheduled Fourier Volume Morphing”, *Proceedings of SIGGRAPH ‘92, Computer Graphics*, 26, (2):43-46, July 1992.

[7] Jain A.K., “*Fundamentals of Digital Image Processing*”, Prentice Hall Inc., Englewoods Cliffs, NJ, 1989.

[8] Levoy M., “Display of Surfaces from Volume Data”, *IEEE Computer Graphics and Applications*, 8(5):29-37, May 1988.

[9] Kaufman A. (ed.), “*Volume Visualization*”, IEEE Computer Society Press, 1990.

[10] Keys R.G., “Cubic Convolution Interpolation for Digital Image Processing”, *IEEE Transaction on Acoustics, Speech, and Signal Processing,* ASSP-29(6):1153-1160, December 1981.

[11] Max N., “An Optimal Filter for Image Reconstruction”, *Graphics Gems II,* Academic Press, pp. 101-104, 1991.

[12] Marschner S.R. and Lobb R.J., “An Evaluation of Reconstruction Filters for Volume Rendering”, *Proceedings of Visualization ‘94,* IEEE CS Press, pp. 100-107, October 1994.

[13] Mitchell D.P. and Netravali A.N., “Reconstruction Filters in Computer Graphics”, *Computer Graphics*, 22(4):221-228, August 1988.

[14] Norton A., Rockwood A. P., Skolmoski P.T., “Clamping: A Method of Antialiasing Textured Surfaces by Bandwidth Limiting in Object Space”, Proceedings of SIGGRAPH ‘82, Computer Graphics, 16, (3):1-8, July 1992.

[15] Oppenheim A.V. and Schafer R.W., “*Digital Signal Processing*”, Prentice Hall Inc.,

Englewoods Cliffs, NJ, 1975.

[16] Park S.K. and Schowengerdt R.A., “Image Reconstruction by Parametric Cubic Convolution”, *Computer Vision, Graphics, and Image Processing*, 23:258-272, 1983.

[17] Parker J.A., Kenyon R.V. and Troxel D.E., “Comparison of Interpolation Methods for Image Resampling”, *IEEE Transactions on Medical Imaging*, MI-2(1):31-39, March 1983.

[18] Saff E.B. and Snider A.D., “Fundamentals of Complex Analysis for Mathematics, Science, and Engineering”, Prentice Hall Inc., Englewoods Cliffs, NJ, 1976.

[19] Stenger F., “*Numerical Methods Based on Sinc and Analytic Functions*”, Springer Verlag, 1993.

[20] Totsuka T. and Levoy M., “Frequency Domain Volume Rendering”, *Computer Graphics Proceedings SIGGRAPH ‘93*, ACM Press, pp. 271-278, August 1993.

[21] Turkowski K.,”Filters for Common Resampling Tasks”, *Graphics Gems I*, Academic Press, pp. 147g-165, 1991.

[22] Wolberg G., *“Digital Image Warping”*, IEEE Computer Society Press, 1990.

[23] Yagel R. and Kaufman A.E., “Template-Based Volume Viewing”, *Computer Graphics Forum*, 11(3): 153-157, September 1992.

[24] Yao K. and Thomas J.B., “On Truncation Error Bounds for Sampling Representations of Band-Limited Signals”, *IEEE Transactions on Aerospace and Electronic Systems*, AES-2(6):640-647, November 1966.