Proceedings of the 6th EuroGraphics Rendering Workshop, pages 33–44, Dublin, Irland, 1995.

Spatial Domain Characterization and Control of Reconstruction Errors

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 or a volume needs to be resampled as a result of an affine or perspective transformation, texture mapping, or volume rendering. 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 the 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 suitable 2D and 3D examples.

1. Introduction

A fundamental operation of many graphics algorithms is reconstruction. The image or the volume is in the usually in the form of a regular rectilinear *grid* or a *mesh* of sampled function values termed pixels (image) or voxels (volume). Many algorithms, such as texture mapping, image and volume transformation (e.g., rotation, scaling, translation), and volume exploration and rendering, imply a transformation of a raster (2D or 3D) from a *source space* to a *target space*. All these algorithms must reconstruct the underlying function either in source or target space. For instance, when 2D images or 3D volumes are subjected to affine transformations or when they are subjected to non-affine grid deformation (texture mapping, warping, registration), the function value in the form of pixel or voxel intensity has to be computed on the target grid, commonly referred to as the *resampling* grid. Similarly, reconstruction is needed when a 3D volume is sliced [13] or rendered [5] [6] [18] [20]. Either the voxel intensity, or the opacity, or color needs to be determined at intermediate points inside the volume. Interpolation is the reconstruction method of choice in all the afore-mentioned algorithms [5][6][20]. Reconstruction in the target space is much rarer and is conducted through the use of reconstruction kernels. Splatting [18], a volume rendering algorithm is the only method known to reconstruct functions in target space. However, both forms of reconstruction are equivalent and in this paper we shall examine reconstruction in terms of function interpolation.

Given the essential nature of the reconstruction operation it is surprising that although much work has been expended in the design of reconstruction filters, not much attention has been paid to characterize and control its numerical accuracy, The work described here is aimed to give the user, for the first time, the ability to set a *point-wise* numerical 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 12). 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 finally in Section 4 we present our results.

2. Background

Much has been written about the reconstruction of sampled datasets in signal processing (1D data) [11] or image processing applications (2D data) [4][19] and computer graphics [2]. Another body of work on the same problem is available in the applied mathematics literature [15][22]. Although the terminology in the following discussion is pertinent to 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 $f_R(x)$. We now state some assumptions about the original function $f(x)$ which we use in this paper:

- The sampled dataset $f_S(x)$ is uniformly sampled from $f(x)$ with frequency $\omega_S = \pi/h$.
- The 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.
- The function $f(x)$ is analytic, i.e., the function and its derivatives exist at all points. Thus, we are precluding step functions and other discontinuous functions.
- The function $f(x)$ is spatially limited. To overcome the mathematical difficulties thus created, we assume that our function is actually of infinite extent but is zero outside the extent of the dataset.
- 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.

The assumption of bandlimitedness is not too restrictive and holds for many forms of sampled data used in computer graphics and scientific visualization. During the process of acquiring digital images (e.g., cameras, scanners, numerical simulation) a filtering operation is included thus bandlimiting the function. Also, synthetic and procedural methods in order to provide antialiased images, typically employ a filtering step which bandlimits the image.

The last assumption regarding the sampling frequency is essential if we are to reconstruct the function exactly. If this assumption is violated the sampled signal is corrupted with pre-aliasing error and reconstruction is never correct. Thus, in our context of bandlimited functions, the Nyquist frequency is given by $2\omega_c$ and the sampling frequency ω_S is always greater than or equal to the Nyquist frequency. The Whitaker-Shannon-Koletnikov (WKS) theorem, known commonly as *Shannon's Sampling Theorem,* allows for the perfect reconstruction of a signal from its samples. It 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 1 (yielding the continuous function f_R) [11][15][22]. Thus, we reconstruct with the formula

$$f_R(x, h) = \sum_{k=-\infty}^{\infty} S(x, k, h) f_S(kh) \tag{1}$$

where

$$S(x, k, h) = \begin{cases} 1 & x = kh \\ \dfrac{\sin\frac{\pi}{h}(x - kh)}{\frac{\pi}{h}(x - kh)} & x \neq kh \end{cases} \tag{2}$$

The function $S(x,k,h)$ is called the *Sinc* or the *Cardinal* function. We can extend this discussion to multiple dimensions through the use of *separable filters,* which sample the data successively along each axis. For 2D images, the reconstruction requires the product of two *Sinc* functions, and similarly we can use three *Sinc* functions in 3D. Thus, the volume reconstruction equation becomes:

$$f_R(x, y, z, h_x, h_y, h_z) = \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} S(x, i, h_x) S(y, j, h_y) S(z, k, h_z) f_S(ih_x, jh_y, kh_z) \tag{3}$$

The quantities h_x, h_y and h_z are the sampling periods along each of the principal axes. 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 [9]. Anisotropic effects are always present unless a radially symmetric filter is employed. In this paper we limit ourselves to separable filters, although a similar analysis can be applied to radially symmetric filters.

2.1 Practical Reconstruction Methods and Errors

The ideal reconstruction process from Equation 1 although realizable is rarely used in practice, since it requires the contribution from all elements of the dataset. An obvious solution is to truncate the function to include only L integer valued points:

$$f_{Rt}(x, M, h) = \sum_{k=-M}^{M} S(x, k, h) f_s(kh) \tag{4}$$

where $M = L \text{ div } 2$. The ensuing interpolation filter is called an *finite impulse-response* (FIR) filter of order L and *zero phase*. Truncation induces an error in function reconstruction that is often referred to as post-aliasing [8][9]. Since we emphasize spatial methods in this paper, we shall call this error *truncation error* (Equation 5) denoted by e_t. Note that this error also includes the data the filter is applied against.

$$\begin{aligned} e_t(f_s, x, M, h) &= \left| f_R(x, h) - f_{Rt}(x, M, h) \right| \\ e_t(f_s, x, M, h) &= \left| \sum_{|k|>M}^{\infty} S(x, k, h) f_s(kh) \right| \end{aligned} \tag{5}$$

The truncation error manifests itself as *blurring* (due to attenuation) and *jaggies (*due to including frequencies beyond ω_c) in an image. In practice, the function is never reconstructed with a truncated *Sinc*. Truncation is tantamount to multiplying the infinite filter by a spatially limited *rectangular window* causing aliasing in the frequency spectrum, thereby altering the filter's frequency characteristics. A commonly used interpolation filter is *cubic convolution* [9], which possesses a reasonably good frequency response. If computational cost is a concern trilinear interpolation (a cone filter) is used. Yet another way to design filters is to use a *window function* besides a rectangle [4][8][11]. These functions alter the characteristics of the truncated frequency spectrum and the choice of one function over another is a trade-off between blurring and aliasing. If *NS* is a filter different from the *Sinc* function, then the reconstructed function is now given by:

$$\tilde{f}_{Rt}(x, M, h) = \sum_{k=-M}^{M} NS(x, k, h) f_s(kh) \tag{6}$$

Thus the total reconstruction error, denoted by e_R, is equal to:

$$e_R(f_s, x, M, h) = \left| f_R(x, h) - \tilde{f}_{Rt}(x, M, h) \right| \tag{7}$$

The total error can be divided into two components:

$$\begin{aligned} e_R(f_s, x, M, h) &= \left| f_R(x, h) - f_{Rt}(x, M, h) + f_{Rt}(x, M, h) - \tilde{f}_{Rt}(x, M, h) \right| \\ e_R(f_s, x, M, h) &\le \left| f_R(x, h) - f_{Rt}(x, M, h) \right| + \left| f_{Rt}(x, M, h) - \tilde{f}_{Rt}(x, M, h) \right| \end{aligned} \tag{8}$$

The first component is the truncation error e_t while the second error arises from the use of a filter besides a *Sinc* function. We call the latter *non-Sinc error* denoted by e_{ns}. Thus, the total reconstruction error:

$$e_R(f_s, x, M, h) \le e_t(f_s, x, M, h) + e_{ns}(f_s, x, M, h) \tag{9}$$

A naive application of this error characterization leads to a paradox, since one is lead to believe that reconstruction with functions different from a truncated *Sinc* lead to lower quality images. The use of truncated *Sinc* functions cause undue *ringing* in the reconstructed function, which is a manifestation of the *Gibbs phenomenon* [11]. Ringing occurs when one approximates a function with a sharp discontinuity. All filters suffer from this effect; however it is more

pronounced in the case of truncated *Sinc* functions. The resulting error from ringing cannot be described by the Cardinal series expansion of Equation 1 and must be considered separately. Marschner and Lobb characterize this undesirable effect as the *overshoot* error and provide a way to measure it [8]. The error is obtained by convolving a step function with the reconstruction filter and then measuring the maximum deviation of reconstructed function from the original function. We denote this error as e_o. Thus a complete characterization of the error is:

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

It is important to note that this characterization of the reconstruction error is purely numerical and is not based on perceptual considerations which address the issue of the suitability of an image to a human observer. Although there is a positive correlation between numerical error and image quality, not much is known about their relationship [14]. We do not even attempt to address these very complex questions here. Our exercise in this work is merely devoted to identify the sources of error and quantify them.

The study of reconstruction errors has received only modest attention in the graphics and image processing literature. In [12] 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 [9] first introduced the reconstruction metrics *pre-aliasing* and *post-aliasing*. Marschner and Lobb [8] further characterize post-aliasing artifacts; the proposed metrics, *smoothing*, *post-aliasing*, and *overshoot*, can help design a suitable filter. However, none of the proposed measures are well suited for determining the accuracy of an interpolation or filtering scheme on a sampled dataset. Carlbom on the other hand takes into consideration the error incurred by an interpolation filter while designing *optimal* filters [2]. However, this error is defined in the frequency domain and only measures the deviation of the frequency spectrum from the ideal spectrum. Also, this metric is global in nature and does not provide error control on a point-wise basis.

Moreover, none of the above characterizations is conducive to the idea of adapting the filter size to the resampling location and to local data characteristics. Adaptive filtering methods have been reported [10], [16]. However, these methods derive their adaptivity in the frequency domain and hence are cumbersome to use. Also, these methods are not driven by any user defined error threshold. Our method, on the other hand, operates in the spatial domain. It can estimate with minimum computational burden the filter size for a given resampling location so we can efficiently interpolate to a desired level of accuracy. 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. In the following 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. A more comprehensive treatment is available in [8]. We do not attempt to provide any new estimates for *overshoot* error here.

3.1 Truncation Error Estimates

Let the resampling point $\boldsymbol{R}$ at x lie in cell n, i.e., between data points $f_s(nh)$and $f_s((n+1)h)$. Also, let τ be the distance of $\boldsymbol{R}$ from data point $f_s(nh)$, in other words, $x=nh+\tau$. The truncation error was defined in Equation 5 in Section 2.1. After replacing x and the *Sinc* function $S(x,k,h)$ with appropriate expressions we get:

$$e_t(f_s, x, M, h) = \left| \sum_{|k| > M}^{\infty} \frac{\sin\frac{\pi}{h}(nh + \tau - kh)}{\frac{\pi}{h}(nh + \tau - kh)} f_s(kh) \right| \tag{11}$$

Expanding the sine term in Equation 11 as the sum of sines and cosines, replacing n-k with m, and after some simplification we get Equation 12.

$$e_t(f_s, n, \tau, M, h) = \left| \frac{\sin\frac{\tau}{h}}{\frac{\pi}{h}} \sum_{|m| > 0}^{\infty} \frac{(-1)^m}{(mh + \tau)} f_s((n-m)h) \right| \tag{12}$$

We now make two important observations from Equation 12.

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=0.5h, it attains its maximum value and drops off to zero as one moves closer to the sampled data locations (τ=0 or τ=h). An important implication is that one can use filters of different lengths depending on the location of the resampling point.

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 neighborhoods. The implications of this observation is that we can effectively limit ourselves to reasonably sized neighborhoods. This observation allows more efficient reconstruction since even smaller length filters can be used.

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. We now list 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 [21]. However, before we state the relevant results we discuss an important idea of *frequency guards*. 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 applications it is sufficient to use very crude estimates. The methodology used in [21] can be applied to determining the error of any polynomial approximation scheme.

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. The quantity E_f can be simply determined by Equation 13 [15]. All the sampled data points are included in the summation of Equation 13.

$$E_f = \sqrt{\frac{h\sum_k \left|f_s(kh)\right|^2}{2\pi}} \tag{13}$$

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

$$e_t(f_s, x, h, M) \le \frac{2\sqrt{\frac{rE_f}{h}}\left|\sin\frac{\pi x}{h}\right|}{\pi^2(1-r)M} \tag{14}$$

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 $x=nh+\tau$, i.e, it lies in the cell *n* of the source grid, we can replace *x* with τ (refer Equation 12). We now the error bound in terms of the maximum of a function. 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

$$e_t(f_s, x, h, M) \le \frac{Max_f\left|\sin\frac{\pi x}{h}\right|}{\pi M\cos\frac{r\pi}{2}} \tag{15}$$

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

3.2 Non-Sinc Error

Once again we can either use the spectral energy or the maximum value of the function. The bound which includes the spectral energy is listed in Equation 16. 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 ***R***. Since the filters are space invariant one evaluate the filters when placed at $k=0$ for sake of convenience.

$$e_{ns}(f_s, x, n, h, M) \le E_f(M)\sqrt{\frac{1}{2\pi}\int_{-M}^{M}\left|S(t,0,h)-NS(t,0,h)\right|^2 dt} \tag{16}$$

One can similarly define a bound including the maximum value of the function (Equation 17).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.

$$e_{ns}(f_s, x, n, h, M) \le Max_f(M) \int_{-Mh}^{Mh} |S(t, h) - NS(t, h)| dt \tag{17}$$

The function *NS* can be any realizable filter. We however use windowed *Sinc* filters given their ability to lend themselves to adaptive use. Also, in [8] it was shown that windowed Sinc filters compare favorably with other filters in terms of the smoothing and post-aliasing metrics defined therein. We employ the *Hamming window i*n our interpolation schemes (Equation 18) [11]. The significant aspect of these windows is that it falls gradually to zero at the corners of the window and hence reducing the impact of aliasing caused through the use of the rectangular window. For multiple dimensions one can use a product of two or three 1-D window functions.

$$w_H(x, M, h) = \begin{cases} 0.54 + 0.46 \cos\left(\frac{\pi x}{(M+1)h}\right) & |x| < (M+1)h \\ 0 & otherwise \end{cases} \tag{18}$$

In this section we described the errors that arise from filtering operations. Detailed explanations and derivations were not provided here can be found in [8]. In the Section 4 we use 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 then show how these bounds can be employed in adaptive filtering schemes. In the latter part of this section we implement our schemes in 2D and 3D examples.

4.1 Accurate and Adaptive Reconstruction of 1D Signals

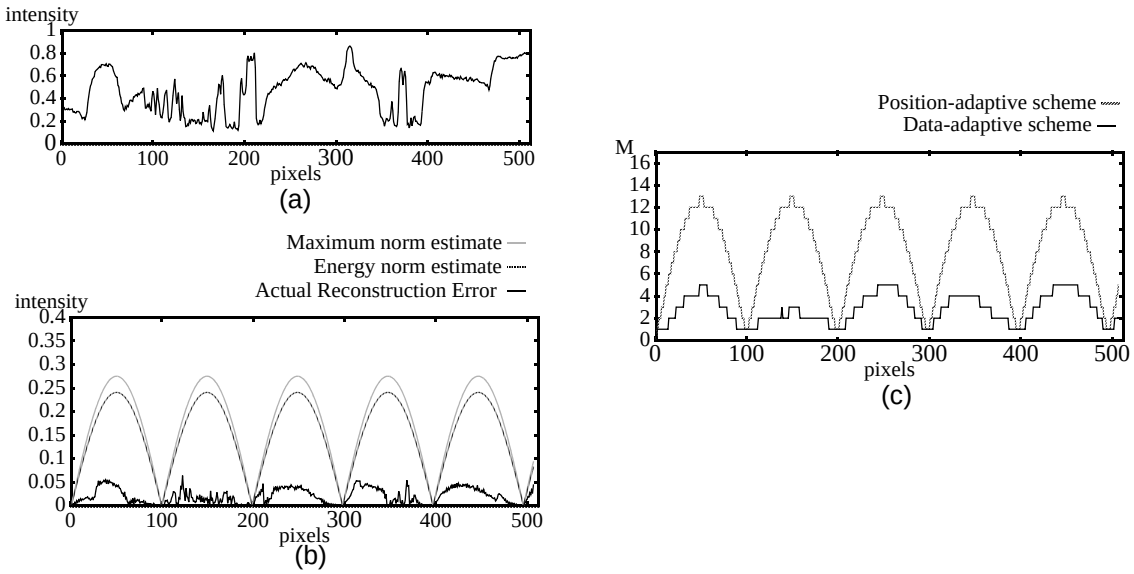


FIGURE 1. Comparison of Error Estimates. (a) 1D signal obtained from a row of an image. (b) The actual error and the estimates from Equation 14 (energy) and Equation 15 (maximum). Both estimates bound the actual error well, the energy estimate being closer. (c) Filter sizes for the position- and data-adaptive schemes.

We consider a representative resampling scheme that frequently arises in computer graphics. In this section we distinguish between resampling schemes and reconstruction operations. Resampling schemes provide the points where the functions are reconstructed. The resampling scheme under consideration occurs when a 2D image or a 3D volume is subjected to affine transformation (including scaling), or during texture mapping, or during ray-casting of 3D volumes [6][20]. 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.

In Figure 1a we consider a 1D signal obtained from a row of a scanned image. 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 scanlines considered for this work. The actual error from Equation 12, the error estimates from Equation 14 (using energy) and Equation 15 (using maximum values) are determined, when the signal is resampled onto a new grid (Figure 1b). The locations of resampling points are defined by the function $x_k=x_0+0.99*k$ (resampling scheme), where x_0 is the location of the first row pixel and k is an integer. The estimates from Equation 15 are looser and we found the energy estimates closer to the actual error for many 1-D 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. 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 14 and Equation 15. For instance one can set the point wise error to ε for all x along the length of the signal. The required filter length at resampling point x can be determined from the computation listed in Equation 19.

$$M(x) = \frac{2\sqrt{\frac{rEf}{h}}\left|\sin\frac{\pi x}{h}\right|}{\pi^2(1-r)\varepsilon} \tag{19}$$

Figure 1c shows the minimum filter length at all points required for the resampling of the signal of Figure 2 to obtain an user defined accuracy of $\varepsilon=0.02$. 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. The estimated bounds in Figure 1b 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 14 or Equation 15. We provide one way to compute M_e in Equation 20.

$$M_e = \frac{2\sqrt{\frac{rEf}{h}}\left|\sin\frac{\pi}{2h}\right|}{\pi^2(1-r)\varepsilon_{min}} \tag{20}$$

The value of τ is set 0.5 to cover all possible resampling positions. A pre-processing step is now required which computes the energy or the maximum of the function value over neighborhoods. Figure 1d shows the sizes of the filters used when a neighborhood of size *25* is used. The use of smaller neighborhoods yields smaller filters and hence savings in reconstruction time We call filtering scheme *data-adaptive*. In all our experiments we used the rectangle window function and hence did not incur the non-Sinc error or the overshoot 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.

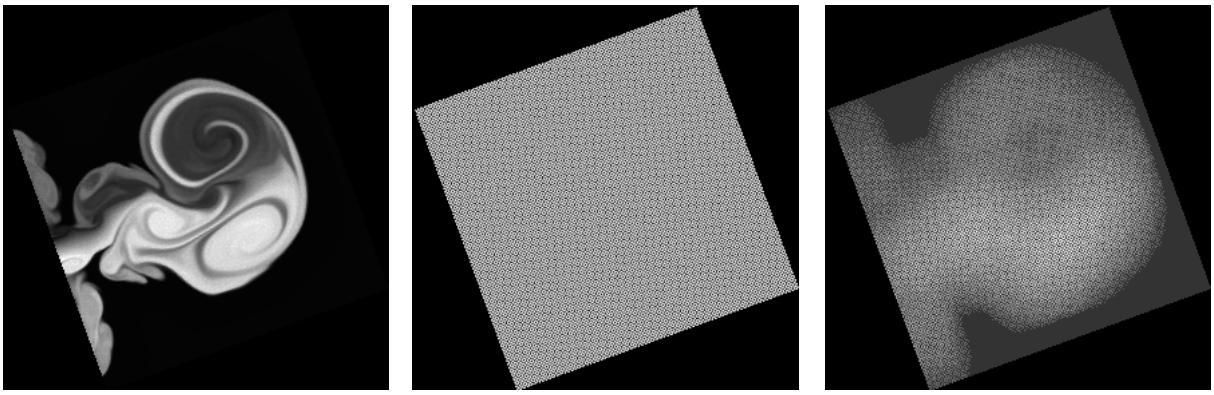
4.2 Multidimensional Examples

The error bounds and filtering schemes can also be adapted to 2D images and 3D volumes. Essentially, the error is now the product of the error incurred in each of the individual dimensions. The data-adaptive scheme requires that the energy or the maximum of the underlying function be determined over a neighborhood. Efficient algorithms to compute these quantities were developed that sweep the image or volume in a systematic manner and exploit the inherent coherency of the computations through the use of circular buffers. Once the local energy or maximum is found, it is stored in an image or a volume of the same size as the original.

We consider two-dimensional images first. The expression for the truncation error expressed in terms of spectral energy is given by Equation 21. We found that for 2D images the error bound involving the maximum was larger and therefore we did not employ that bound in our adaptive methods.

$$e(f, x, y, h_x, h_y, M) \le \frac{16 E_f \left| \sin \frac{\pi \tau_x}{h_x} \right| \left| \sin \frac{\pi \tau_y}{h_y} \right|}{\pi^4 M^2 (1 - r_x)(1 - r_y)} \tag{21}$$

By specifying the minimum desired error, one can use a derivative of Equation 21 (and similar to Equation 20) to determine the size of neighborhood required to achieve the desired error of ε_{min}. At each resampling point, filter size is then determined by using the error estimates and the filter is applied in the neighborhood. Figure 2a shows an image obtained from a fluid dynamics simulation rotated by an angle of 30° and scaled uniformly by a factor of 0.75. 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. The values of the frequency guards r_x and r_y were measured to be at 0.34 and 0.10. Figure 2b shows the position-adaptive filter sizes, while figure 2c shows the data-adaptive. As evidenced from Figure 2b the filter size changes in a periodic sinusoidal fashion in the position-adaptive scheme. On the other hand, the filter size adapts to the data complexity in the data-adaptive scheme. A minimum error of 0.02 was used to determine the neighborhood size for the data-adaptive scheme. The average half-filter size for the position-adaptive scheme was measured at 7.15, while the average filter size for the data-adaptive scheme was measured at 1.96. The data-adaptive scheme thus is better than a cubic convolution scheme, which has a filter size of 5 (or half-filter size of 2.5). Moreover, it guarantees reconstruction to a desired level of accuracy. A 2D Hamming window was used to improve the visual quality of image. For successive rotations, the local neighborhood values of energy need not be recomputed. A similar scheme can be used in texture mapping.



(a) (b) (c)

FIGURE 2. (a) An image from a fluid dynamics simulation reconstructed by the data-adaptive method where ε =0.02, rotated by 30 degrees×, and scaled by a factor of 0.75 (b) Filter sizes used for the position-adaptive method (c) Filter sizes used for the data-adaptive method. Bright values stand for larger filters sizes.

In the case of 3D volumes we found that the error bound using the maximum value was more useful for data-adaptive methods. Given the sparse spectral energy content of 3D datasets, this bound reported a larger variance of filter sizes than the other bound. In Equation 22 we list the bound for the 3D reconstruction error.

$$e(f, x, y, z, h_x, h_y, h_z, M) \le \frac{Max_f \sin \pi \frac{x}{h_x} \sin \pi \frac{y}{h_y} \sin \pi \frac{z}{h_z}}{\pi^6 M^3 \cos \frac{\pi r_x}{2} \cos \frac{\pi r_y}{2} \cos \frac{\pi r_z}{2}} \tag{22}$$

We also implemented our adaptive methods in a slicing algorithm for volumes. The reported implementation projects the desired slicing plane (described by a direction vector and an offset from the origin) onto the XY-plane thus defining an implicit affine transformation. As before in the 2D example, resampling points on the XY-plane are transformed to points inside the 3D volume. The advantage of this slicing implementation over others is in its ability to control the resampling scheme. A further optimization is obtained if the zero-phase property of Equation 4 is not used [17]. We use a neighborhood of size *2M* along each dimension instead in a fashion similar to what is done with cubic convolution filters. Figure 3a and Figure 3b show two different slices of a 256^3 volume MRI head dataset. The values of frequency guards r_x, r_y and r_z were found to be 0.086, 0.055, 0.031 respectively. We employ once again *Hamming* windows to obtain images of higher visual quality. Figure 3c and Figure 3d shows the filter sizes used in the position- and data-adaptive schemes respectively for the slicing plane of Figure 3b. The error threshold is 0.01, while the minimum error used to determine the optimal neighborhood size is 10^{-5}. This minimum error threshold translates to a

neighborhood size of 31 for the data-adaptive scheme. The average half filter size for position-adaptive filtering scheme were measured closer to 2 (Figure 3c), while the same quantity was measured at 1 for the data-adaptive scheme (Figure 3d). As evinced in Figure 3d, higher order filters are used only at some resampling points. A cubic convolution scheme, on the other hand, would require a neighborhood of at least 64 grid points at each resampling point. However, if one uses our adaptive scheme, such a large neighborhood is needed only in some portions of the volume. Thus the saving in computational time can be considerable. This savings is especially useful in volume ray-casting [6]. To obtain accuracy it is common to step along the ray using small increments and thus increase the number of resampling points. A data-adaptive scheme similar to the one presented here would be very useful to guide the reconstruction process.

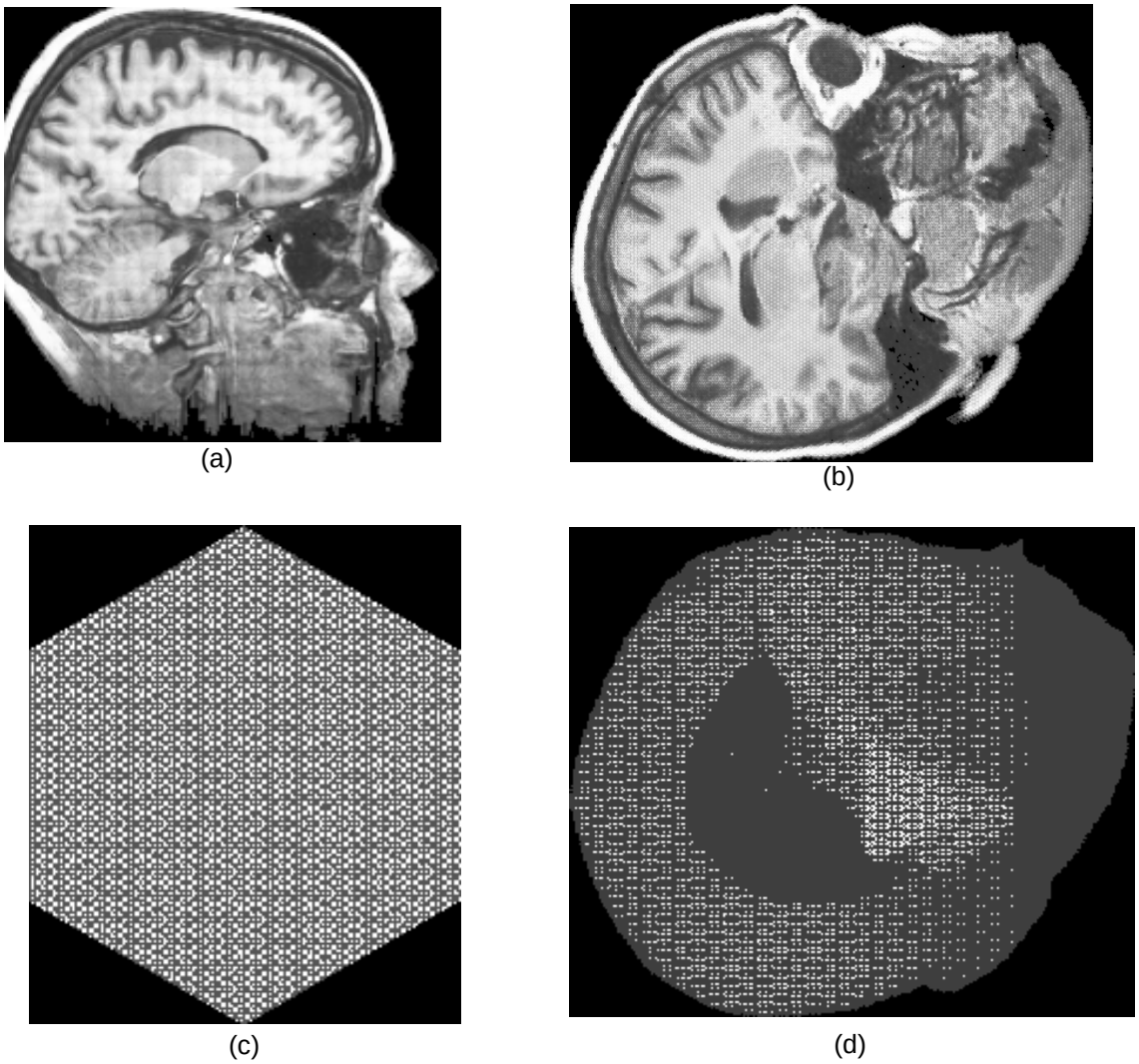


FIGURE 3. Oblique slicing of 3D MRI head dataset. (a) An almost vertical slice of the volume through the center. (b) A slice with directional vector (1,1,1) passing through the center. The error ε is 0.01. (c) The filter sizes used in the position-adaptive scheme for (b). (d) The filter sizes used in the data-adaptive scheme for (b).

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 which adjust filter size to the reconstruction location and the data complexity. Our methods provide the user with a powerful tool for achieving any desired image quality while incurring space and computation costs that are comparable to existing methods. Also, the methods developed herein allow us to forgo

both expensive transformations (such as *Wavelet* and *Fourier Transform*), and the computation of control points (such as *Spline* based methods) to achieve our goals.

6. 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 for supporting our efforts: Prof. Ashok Krishnamurthy, Prof. Bhavik Bhakshi, Prof. Wayne Carlson, and Don Stredney.

7. References

[1] Carlbom I., "Optimal Filter Design for Volume Reconstruction and Visualization", *Proceedings of Visualization '93*, IEEE CS Press, pp. 54-61, 1993.

[2] Glassner A.,"Principles of Digital Image Synthesis", Morgan Kaufmann, 1995.

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

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

[5] Hanrahan P., "Three-Pass Affine Transforms for Volume Rendering", *Computer Graphics*, 24(5):71-77, November 1990.

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

[7] Machiraju, R., Swan, E., Yagel, R., "Error Bounded and Adaptive Image Reconstruction," *Technical Research Report*, Department of Computer and Information Science, The Ohio State University, OSU-CISRC-1/95-TR03, January 1995.

[8] 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.

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

[10] 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.

[11] Oppenheim A.V. and Schafer R.W., "*Digital Signal Processing*", Prentice Hall Inc., Englewoods Cliffs, NJ, 1975.

[12] 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.

[13] Rhodes M., Glenn W., Azzawi Y., "Extracting Oblique Planes from Serial CT Sections", *Journal of Computer Assisted Tomography*, 4(5): 649-657, October 1980.

[14] Schreiber, W., Troxel, D., "Transformation Between Continuous and Discrete Representations of Images: A Perceptual Approach", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-7(2): 178-186, March 1985.

[15] Stenger F., "*Numerical Methods Based on Sinc and Analytic Functions*", Springer Verlag, 1993.

[16] Totsuka T. and Levoy M., "Frequency Domain Volume Rendering", *Computer Graphics Proceedings SIGGRAPH '93*, ACM Press, pp. 271-278, August 1993.

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

[18] Westover L.,"Footprint Evaluation for Volume Rendering", *Computer Graphics*, 24(4):367-376, August 1990.

[19] Wolberg G., "*Digital Image Warping*", IEEE Computer Society Press, 1990.

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

[21] 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

[22] Zayed A.I., "Advances in Shannon's Sampling Theory", CRC Press, Boca Raton, FL, 1993.