

## 4. Analytical Reconstruction-Error Comparison

In this chapter, an expression is derived for the peak relative-error bound of reconstructed signals as a function of feature scale. The error bounds of pyramid, Gaussian, and cubic reconstruction filters are compared. These are all separable filters normally applied in two or three dimensions. The approach here is to study the application of these filters to one-dimensional features of known scale. The results are then extended to higher dimensions. The signal feature that we use is the normalized Gaussian whose scale is  $\sigma$  by our definition. The results are intended to compliment the findings in chapter three and thereby strengthen the claim the cubic filters produce lower reconstruction error than pyramid or Gaussian filters.

### 4.1. Pyramid Filter

The reconstruction-error bound for a pyramid filter is computed by considering a signal which has only one feature with an explicit representation  $G(x)$  and comparing that with the reconstructed signal  $R(x)$ . The maximum difference between  $G(x)$  and  $R(x)$  is the upper bound of the error produced at any resampling point. Let

$$G(x) = (2\pi\sigma^2)^{-1/2} \exp(-x^2/2\sigma^2) \quad (4.1)$$

be the explicit representation where  $\sigma$  is the feature scale centered at zero (Fig. 4.1).  $R(x)$  is defined over  $x \mid (x_0 \geq x \geq x_1)$  by the linear-interpolation function:

$$R(x) = (G(x_0) (x_1 - x) + G(x_1) (x - x_0)) / (x_1 - x_0) \quad (4.2)$$

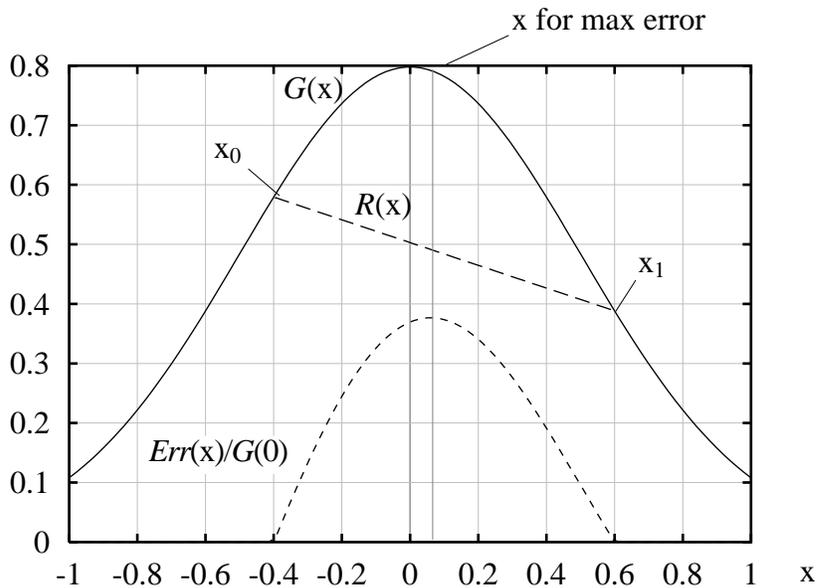


Fig. 4.1 - Linear reconstruction of 1D gaussian

Error is defined as

$$Err(x) = |G(x) - R(x)| \quad (4.3)$$

We wish to find  $x_0$  and  $x$  that maximizes  $Err(x)$  for a fixed  $\sigma$  and  $h = x_1 - x_0$ . Let

$$Ers(x) = (G(x) - R(x))^2 \quad (4.4)$$

Let  $d_0 = (x - x_0) / h$  and  $d_1 = (x_1 - x) / h$ . Then equation 4.4 can be expanded into

$$\begin{aligned} Ers(x) = & G(x)^2 - 2 d_1 G(x) G(x_0) - 2 d_0 G(x) G(x_1) + \\ & + 2 d_1 d_0 G(x_0) G(x_1) + d_1^2 G(x_0)^2 + d_0^2 G(x_1)^2 \end{aligned} \quad (4.5)$$

The peak error is found by differentiating equation 4.5 and finding values of  $x_0$  and  $x$  that make the derivative zero for all  $\sigma$  and  $h$ .

$$\begin{aligned} \partial Ers(x) / \partial x = & \partial G(x)^2 / \partial x & (a) \\ & - 2 G(x_0) \partial d_1 G(x) / \partial x & (b) \\ & - 2 G(x_1) \partial d_0 G(x) / \partial x & (c) \\ & + 2 G(x_0) G(x_1) \partial (d_1 d_0) / \partial x & (d) \\ & + G(x_0)^2 \partial d_1^2 / \partial x & (e) \\ & + G(x_1)^2 \partial d_0^2 / \partial x & (f) \end{aligned} \quad (4.6)$$

All six terms of equation 4.6 become zero or cancel each other when  $x_1 = -x_0 = h/2$  and  $x = 0$ . Terms (a) and (d) become zero. Term (b) cancels (c), and (e) cancels (f). This is where  $R(x)$  is constant and equal to  $G(h/2)$  while approximating  $G(x)$  over its region of maximum curvature or value of  $G''(x)$ . Figure 4.2 illustrates error for a family of  $x_0$  positions given a fixed  $\sigma$  and  $h$ . The upper curves are  $G(x)$ . The straight line segments adjacent to the upper curves are a family of  $R(x)$  linear approximations with fixed  $h$

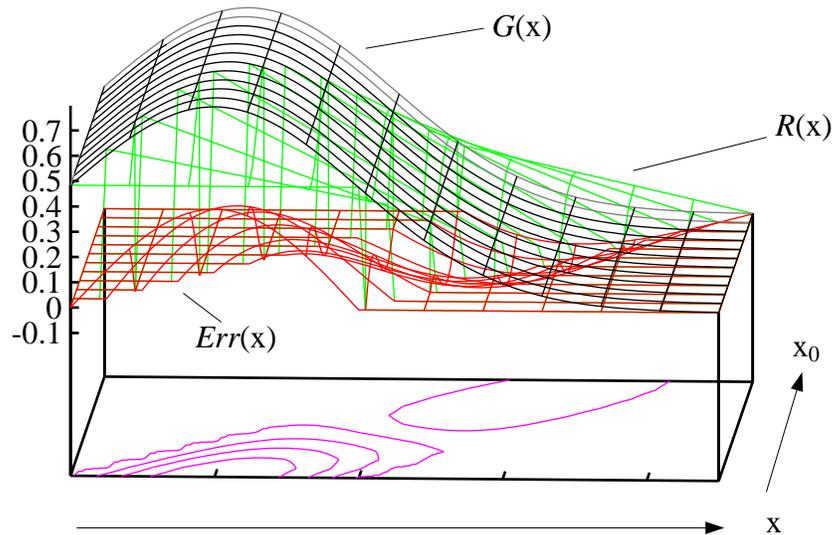


Fig. 4.2 -  $G(x)$ ,  $R(x)$ , and  $Err(x)$  as a function of  $x_0$

where  $x_0$  is different for each member of the family. The error of the the straight-line approximation is the series of bumps in the lowest family of curves. The iso-contours of  $Err(x)$  at the bottom illustrate the error peak at  $x = 0$ ,  $x_1 = -x_0 = h/2$ . The case illustrated is for  $\sigma = 0.5$  and  $h = 1.0$ .

The maximum normalized error for the one-dimensional case is the error bound for any sample and defined as

$$\begin{aligned}
 Nerr1D(\sigma) &= \text{Max}(Err(x)) / G(0) \\
 &= (G(0) - G(h/2)) / G(0) \\
 &= 1 - \exp(-(h/2)^2 / 2\sigma^2) \\
 &= 1 - \exp(-h^2 / 8\sigma^2)
 \end{aligned} \tag{4.7}$$

Since  $G$  and  $R$  are separable, peak error in two dimensions occurs where  $x_1 = y_1 = -x_0 = -y_0 = h/2$ . Because  $G$  is radially symmetric, this can be reduced to a one-dimensional problem with  $x_0' = \langle x_0, y_0 \rangle$ ,  $x_1' = \langle x_1, y_1 \rangle$ , and  $h' = 2^{1/2}h$ .

$$\begin{aligned}
 Nerr2D(\sigma) &= (G(0) - G(2^{1/2}h/2)) / G(0) \\
 &= 1 - \exp(-2(h/2)^2 / 2\sigma^2) \\
 &= 1 - \exp(-2h^2 / 8\sigma^2)
 \end{aligned} \tag{4.8}$$

Similarly, in three dimensions

$$Nerr3D(\sigma) = 1 - \exp(-3h^2 / 8\sigma^2) \tag{4.9}$$

Figure 4.3 plots equations 4.7, 4.8, and 4.9 as a function of  $\sigma$  for  $h = 1$ .

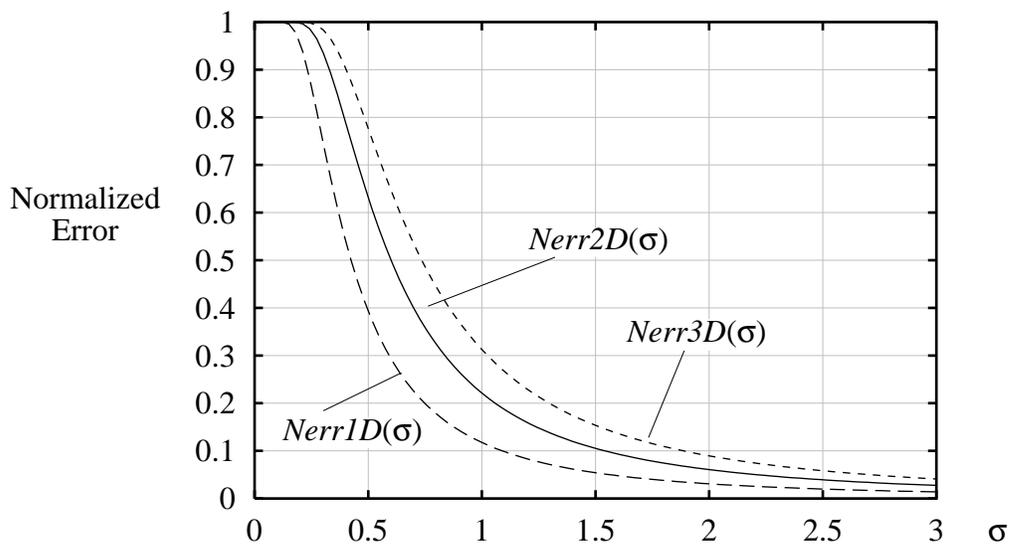


Fig. 4.3 - Pyramid filter error as a function of feature size

## 4.2. Gaussian Filter

The explicit signal representation given by equation 4.1 can be modified to allow an offset  $\varepsilon$  ( $0 \leq \varepsilon < 1$ ) to the feature location.

$$G(x) = (2\pi\sigma^2)^{-1/2} \exp(-(x - \varepsilon)^2 / 2\sigma^2) \quad (4.10)$$

The Gaussian filter kernel is defined as

$$K(\chi) = (2\pi\nu^2)^{-1/2} \exp(-\chi^2 / 2\nu^2) \quad (4.11)$$

Hold the sample rate constant at  $h = 1$  and constrain the sample data-points to fall on integer  $x$ -coordinates. The reconstructed signal is a summation of all sample points weighted by the filter kernel. In practice the summation is limited to a small number ( $d$ ) of samples on each side of the reconstructed point.

$$R(x) = \sum_{i=s}^{i=t} G(i) K(x - i) \quad | \quad s = \lfloor x \rfloor - d + 1, t = \lfloor x \rfloor + d \quad (4.12)$$

Substitute equations 4.10 and 4.11 into 4.3. Square and differentiate the result to find  $x$  and  $\varepsilon$  that maximize error for any  $\sigma$  and  $\nu$ . To keep the equations manageable, we set  $d = 1$ , but larger values of  $d$  also yield the result that error is maximized when  $x = \varepsilon = 0.5$ . Constraining  $x \mid (0 < x < 1)$ , makes  $s = 0$  and  $t = 1$ .

$$Err(x) = G(x) - G(0) K(x) - G(1) K(x - 1) \quad (4.13)$$

$$\begin{aligned} Ers(x) &= Err(x)^2 \\ &= G(x)^2 - 2 G(x) K(x) G(0) - 2 G(x) K(x - 1) G(1) \\ &\quad + 2 K(x) K(x - 1) G(0) G(1) + K(x)^2 G(0)^2 + K(x - 1)^2 G(1)^2 \end{aligned} \quad (4.14)$$

$$\begin{aligned} \partial Ers(x) / \partial x &= \partial G(x)^2 / \partial x && (a) \\ &\quad - 2 G(0) \partial G(x) K(x) / \partial x && (b) \\ &\quad - 2 G(1) \partial G(x) K(x - 1) / \partial x && (c) \\ &\quad + G(0) G(1) \partial K(x) K(x - 1) / \partial x && (d) \\ &\quad + G(0)^2 \partial K(x)^2 / \partial x && (e) \\ &\quad + G(1)^2 \partial K(x - 1)^2 / \partial x && (f) \end{aligned} \quad (4.15)$$

All six terms of equation 4.15 become zero or cancel each other when  $x = \varepsilon = 0.5$ . Terms (a) and (d) become zero. Term (b) cancels (c), and (e) cancels (f). Figure 4.4 illustrates  $G(x)$ ,  $R(x)$ , and  $Err(x)$  for a set of feature positions where  $0 \leq \varepsilon \leq 0.8$ ,  $d = 3$ , and  $h = 1$ . The upper family of curves are  $G(x)$  and the middle curves are  $R(x)$ , where  $\varepsilon$  is offset by 0.1 for each member of a family. The reconstruction error is shown as the lowest curve family. The peak error occurs at  $x = \varepsilon = 0.5$ .

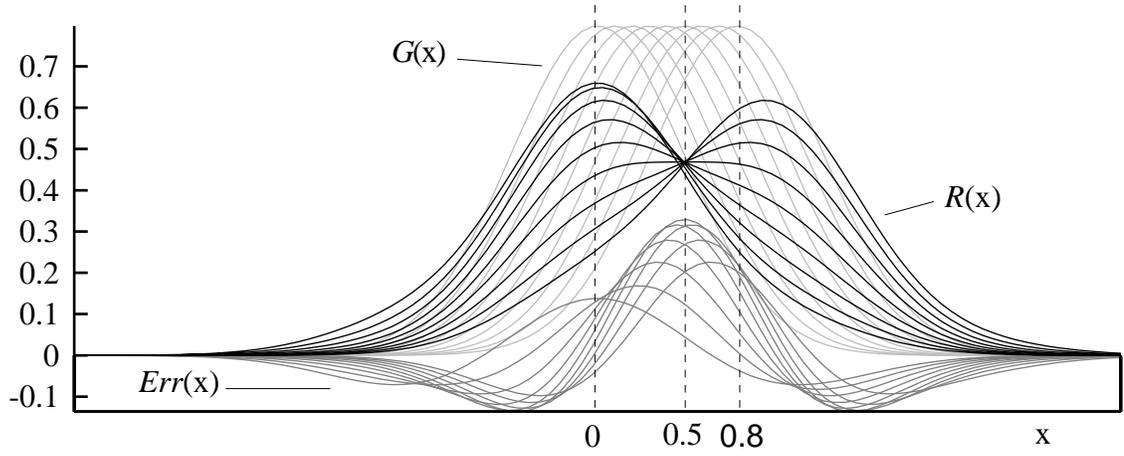


Fig. 4.4 -  $G(x)$ ,  $R(x)$ , and  $Err(x)$  with  $\epsilon$  as a parameter

The one-dimensional normalized error is

$$\begin{aligned}
 Nerr1D(\sigma) &= \text{Max}(Err(x)) / G(\epsilon) \mid \epsilon = 0.5 \\
 &= (G(0.5) - R(0.5)) / G(0.5) \\
 &= 1 - \sum_{i=s}^{i=t} G(i) K(0.5 - i) / G(0.5) \mid s = -d + 1, t = d, d \geq 1 \\
 &= 1 - ((G(-d + 1) K(0.5 - d + 1) + \dots + G(0) K(0.5) \\
 &\quad + G(1) K(-0.5) + \dots + G(d) K(0.5 - d)) / G(0.5) \tag{4.16}
 \end{aligned}$$

Due to the symmetry of  $G$  about  $\epsilon = 0.5$ , and  $K$  about 0, equation 4.16 can be simplified.

$$\begin{aligned}
 Nerr1D(\sigma) &= 1 - (2 / G(0.5)) (G(1) K(0.5) + \dots + G(d) K(d - 0.5)) \\
 &= 1 - (2 (2\pi v^2)^{-1/2}) (\exp((0.5 - 1)^2 / 2\sigma^2) \exp((0.5 - 1)^2 / 2v^2) + \dots \\
 &\quad + \exp((0.5 - d)^2 / 2\sigma^2) \exp((0.5 - d)^2 / 2v^2)) \\
 &= 1 - (2 (2\pi v^2)^{-1/2}) (\exp((0.5 - 1)^2 (v^2 + \sigma^2) / 2\sigma^2 v^2) + \dots \\
 &\quad + \exp((0.5 - d)^2 (v^2 + \sigma^2) / 2\sigma^2 v^2)) \tag{4.17}
 \end{aligned}$$

Figure 4.5 illustrates the normalized error as a function of feature size for several kernel shapes. The kernel is truncated at  $d = 5$  to produce this plot. The kernel shape for  $v = 0.55$  gives the lowest overall error of the shapes tested. This is consistent with the finding in chapter three that the best image comparisons are obtained with kernel  $v$ -values between 0.5 and 0.6.

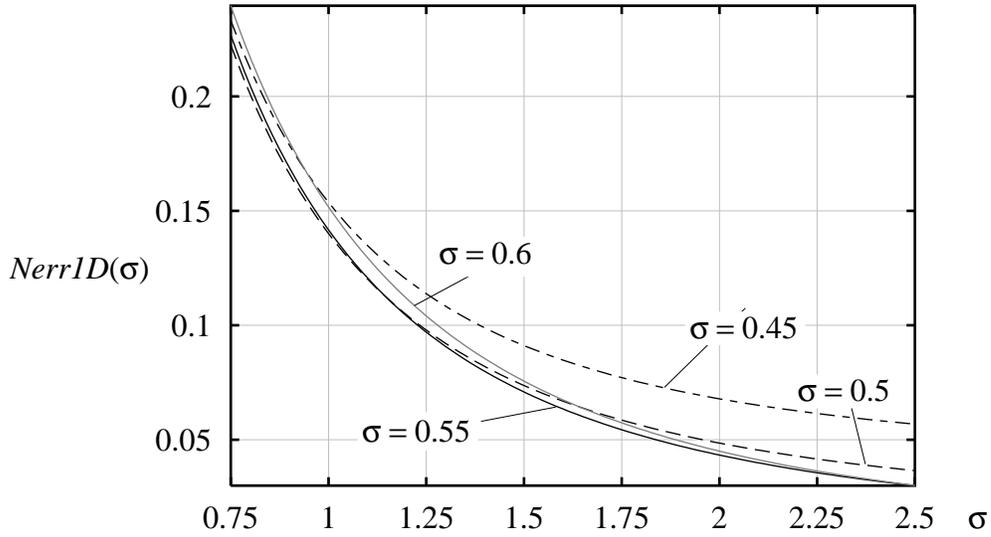


Fig. 4.5 -  $Nerr1D$  for Gaussian kernel  $\nu = 0.45, 0.5, 0.55,$  and  $0.6$

Since  $G$  and  $K$  are both separable, the error for higher-dimensional cases is readily obtained by raising the one-dimensional reconstruction terms to the appropriate power.

$$Nerr2D(\sigma) = 1 - (2 / G(0.5))^2 (G(1) K(0.5) + \dots + G(d) K(d - 0.5))^2 \quad (4.18)$$

$$Nerr3D(\sigma) = 1 - (2 / G(0.5))^3 (G(1) K(0.5) + \dots + G(d) K(d - 0.5))^3 \quad (4.19)$$

Figure 4.6 shows the normalized-error bounds for one, two, and three dimensions while holding the kernel shape constant at  $\nu = 0.55$ .

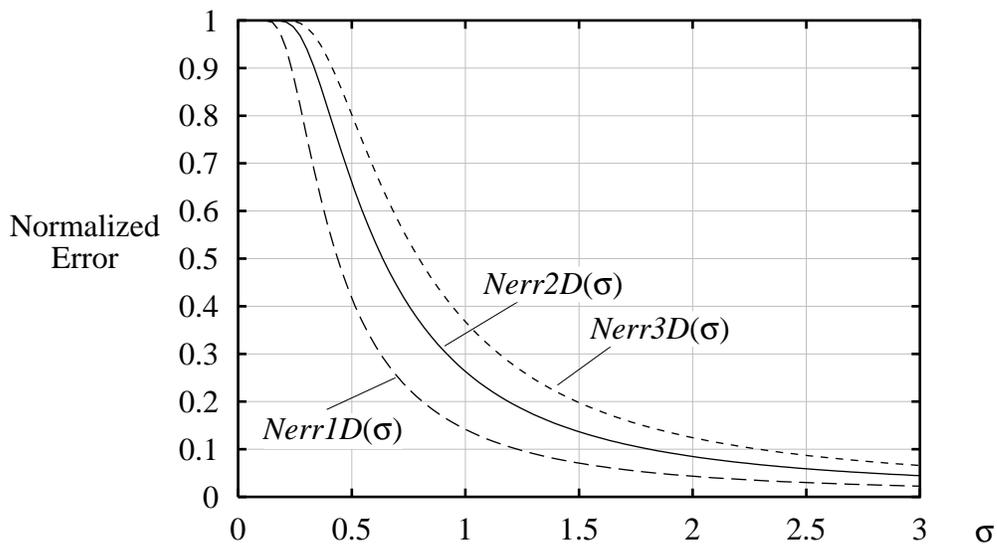


Fig. 4.6 - Gaussian filter error as a function of feature size

The Gaussian filter does not have a constant response to a constant-input signal. The variations in output are referred to as *ripple* and contribute to error. The ripple amplitude is a function of  $\nu$ , and it produces a regular pattern that may produce a more visible artifact than the error induced by reconstruction. The ripple component of the error should be kept low relative to the reconstruction error which is only appreciable for small features. In one dimension, the ripple amplitude for a constant field  $f(x) = 1.0$  is

$$\begin{aligned} \text{Ripple}(\nu) &= R(0) - R(0.5) \\ &= \sum_{i=s}^{i=t} K(i) - K(0.5 - i) \mid s = -d + 1, t = d, d \geq 1 \\ &= (2\pi\nu^2)^{-1/2} \sum_{i=s}^{i=t} \exp(-i^2/2\nu^2) - \exp(-(0.5 - i)^2/2\nu^2) \end{aligned} \quad (4.20)$$

The peak-to-peak ripple error, given by equation 4.20 as a function of  $\nu$ , is plotted in figure 4.7. It shows that the ripple for a one-dimensional kernel shape with  $\nu = 0.55$  is about one percent. In the test images produced in chapter three, ripple artifacts are only visually-apparent for  $\nu = 0.4$ . For a volume renderer using the splatting approach with a two-dimensional filter, the ripple increases to about two percent. Ripple is the asymptotic limit of the reconstruction error as  $\sigma$  increases. For  $\nu = 0.55$ , figure 4.6 shows that moderately small-scale features ( $\sigma \leq 3$ ) produce reconstruction error that is appreciably higher than the ripple component.

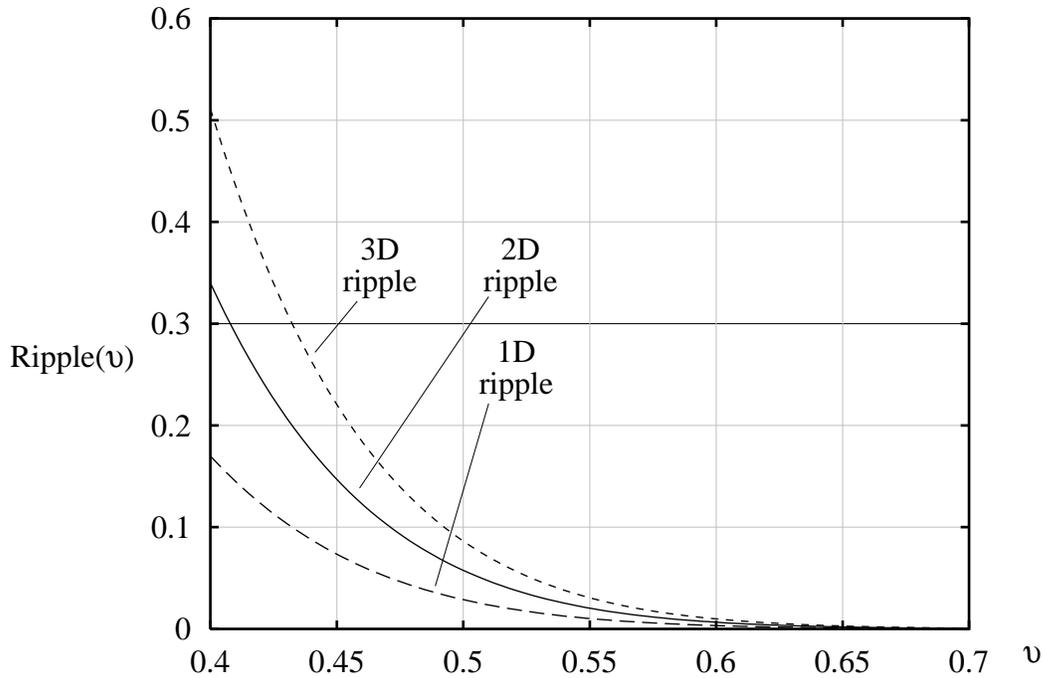


Fig. 4.7 - Normalized peak-to-peak ripple amplitude

### 4.3. Separable Cubic Filter

A general two-dimensional filter is described as  $K(x,y)$ ; a separable filter may also be written as  $K(x)K(y)$ . Separable filters are desirable since for a given filter extent  $\tau$  and dimension  $\delta$ , general filters require  $O(\tau^\delta)$  operations while separable filters require  $O(\tau \delta)$  operations. The filters considered here are the family of piecewise-cubic polynomials with continuous first-derivatives. They are constrained to have a constant flat-field response (i.e.: they have no ripple) and to be exact at the sample points. The filter kernel  $K(x)$  is parameterized by two variables  $B$  and  $C$  whose domain is  $[0, 1]$ . The kernel is zero outside of the domain  $(-2, 2)$ , inside of which it is given by

$$K(\chi) = \begin{cases} 1/6 [ (12 - 9B - 6C) |\chi|^3 + (-18 + 24B + 6C) |\chi|^2 + (6 - 2B) ] & | 0 \leq |\chi| < 1 \\ 1/6 [ (-B - 6C) |\chi|^3 + (6B + 30C) |\chi|^2 + (-12B - 48C) |\chi| + (8B + 24C) ] & | 1 \leq |\chi| < 2 \end{cases} \quad (4.20)$$

Using equation 4.10 for  $G(x)$ , the reconstructed signal for  $x \mid (0 < x < 1)$  is

$$R(x) = K(x - 2) G(-1) + K(x - 1) G(0) + K(x) G(1) + K(x + 1) G(2) \quad (4.21)$$

Reconstruction error is

$$\begin{aligned} Err(x) &= G(x) - R(x) \\ &= G(x) - [ K(x - 2) G(-1) + K(x - 1) G(0) + \\ &\quad K(x) G(1) + K(x + 1) G(2) ] \end{aligned} \quad (4.22)$$

Differentiate equation 4.22 to find the point of maximum error.

$$\begin{aligned} 0 &= G(-1) \partial K(x - 2) / \partial x + & (a) \\ &G(0) \partial K(x - 1) / \partial x + & (b) \\ &G(1) \partial K(x) / \partial x + & (c) \\ &G(2) \partial K(x + 1) / \partial x & (d) \end{aligned} \quad (4.23)$$

Equation 4.23 is satisfied for all  $B$  and  $C$  parameters when  $x = \epsilon = 0.5$ . Under those conditions  $G(-1) = G(2)$  and  $G(0) = G(1)$ . Due to the symmetry of the kernel, the terms (a) and (d) cancel each other, as do terms (b) and (c).

Other solutions to equation 4.23 exist, but they are not maximum critical-points for parameters of interest. If the parameters satisfy  $2C + B = 1$ , the filter has quadratic convergence and becomes exact at the sample points (i.e.:  $R(x) = G(x)$ ) [Mitchell<sup>+</sup>88]. Among these, it has been shown that the Catmull-Rom spline, obtained with parameters  $B = 0$  and  $C = 0.5$ , is the filter providing the lowest error with cubic convergence [Mitchell<sup>+</sup>88] [Keys81] [Park<sup>+</sup>83]. Based on this evidence and the fact that in chapter

three it provided the best image quality, it is used in the remainder of this section.

Figure 4.8 illustrates the error of a Catmull-Rom filter for  $\epsilon = \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ . The upper family of curves are  $G(x)$  for  $\sigma = 0.5$ . The middle curves are  $R(x)$  and the lower curves are  $Err(x)$ .

The one-dimensional normalized error bound is

$$\begin{aligned}
 Nerr1D(\sigma) &= \text{Max}(Err(x)) / G(\epsilon) \quad | \quad \epsilon = 0.5 \\
 &= 1 - 1 / G(0.5) [ K(-1.5) G(-1) + K(-0.5) G(0) + \\
 &\quad K(0.5) G(1) + K(1.5) G(2) ] \\
 &= 1 - 2 / G(0.5) [ K(0.5) G(1) + K(1.5) G(2) ] \\
 &= 1 - 2 [ K(0.5) \exp(-0.5^2 / 2\sigma^2) + K(1.5) \exp(-1.5^2 / 2\sigma^2) ] \\
 &= 1 - 2 [ 1.8125 \exp(-0.5^2 / 2\sigma^2) + 0.5625 \exp(-1.5^2 / 2\sigma^2) ] \quad (4.24)
 \end{aligned}$$

Since  $G$  and  $K$  are separable, the two and three-dimensional error bounds are easily obtained.

$$Nerr2D(\sigma) = 1 - 4 [ 1.8125 \exp(-0.5^2 / 2\sigma^2) + 0.5625 \exp(-1.5^2 / 2\sigma^2) ]^2 \quad (4.25)$$

$$Nerr3D(\sigma) = 1 - 8 [ 1.8125 \exp(-0.5^2 / 2\sigma^2) + 0.5625 \exp(-1.5^2 / 2\sigma^2) ]^3 \quad (4.26)$$

Figure 4.9 plots the normalized error bounds as a function of the feature size.

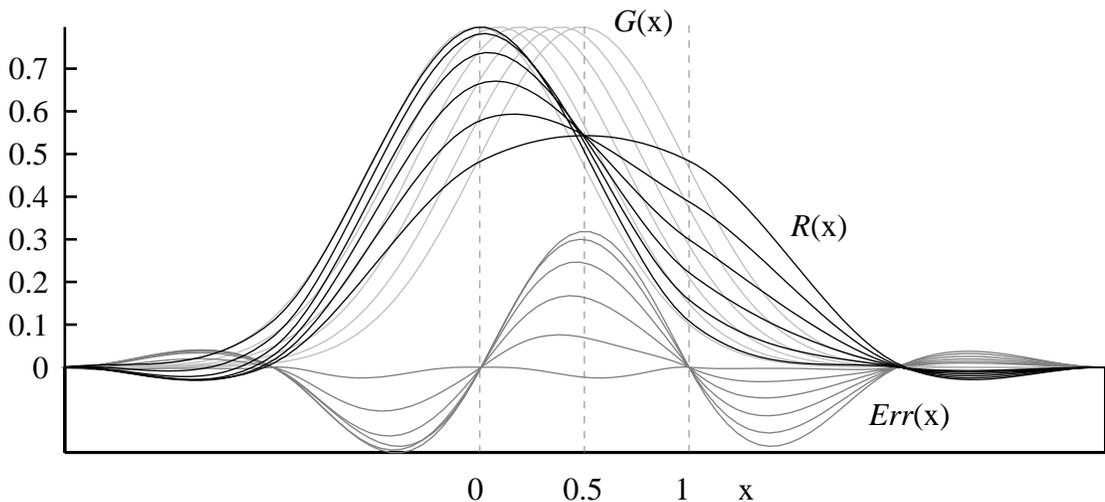


Fig. 4.8 -  $G(x)$ ,  $R(x)$ , and  $Err(x)$  as a function of  $\epsilon$

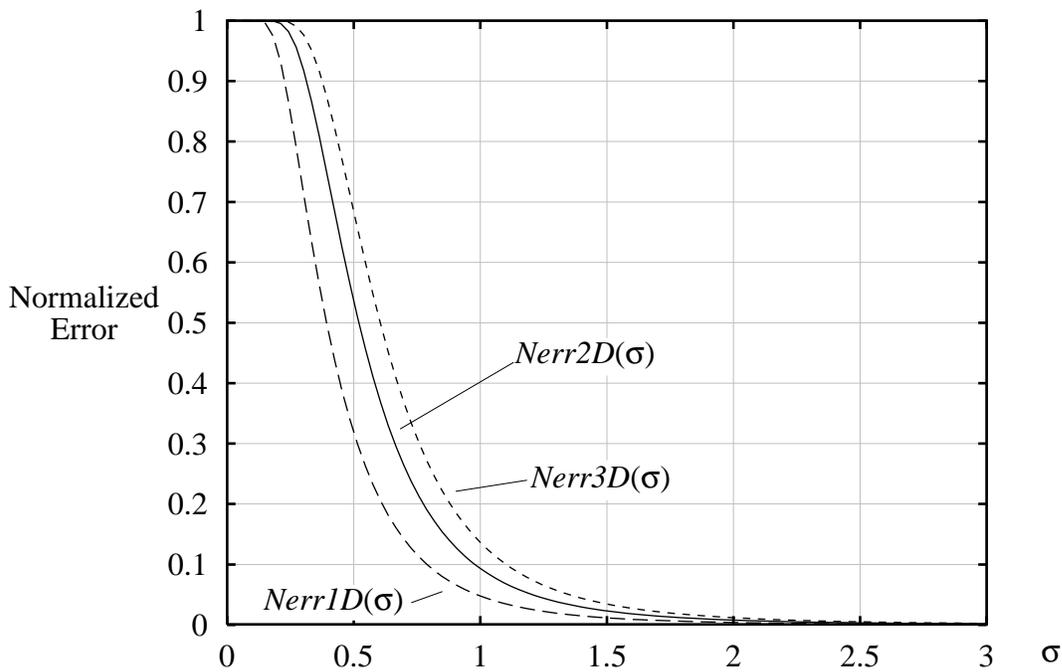


Fig. 4.9 - Cubic filter error as a function of feature size

#### 4.4. Filter Comparison

Figure 4.10 compares the three reconstruction filters. The lowest (and heaviest) line for each filter is the one-dimensional error, the middle-weight line is the two-dimensional case, and the highest (and thinnest) line is the three-dimensional case. The most important difference is found in the 1% to 10% error range. That level of error can produce visible artifacts. The cubic filter is clearly superior while the Gaussian and pyramid filters are fairly similar. These results qualitatively agree with the results of the image tests performed in chapter three. As a result of their agreement, the thesis statement is considered to be verified, *"A separable cubic filter provides more accurate volume reconstruction than a pyramid filter or a Gaussian filter."*

Recall that figure 3.5 showed the proportion of a feature's spectrum that was above the Nyquist limit, and this proportion was called the prealiasing error. If the reconstruction errors plotted in this chapter are compared to the prealiasing error, the reconstruction errors are always greater showing that the filters are contributing additional error.

One ramification of the different filter performances is that lower-resolution volumes can be used with better filters to obtain the same reconstruction fidelity obtained with larger volumes rendered with lower-quality filters. To the extent that rendering speed is a function of the volume size, lower-resolution data will be rendered more quickly. The next chapter compares the speed of rendering methods that use these filters.

It remains to see what differences these filters make when rendering real data in an

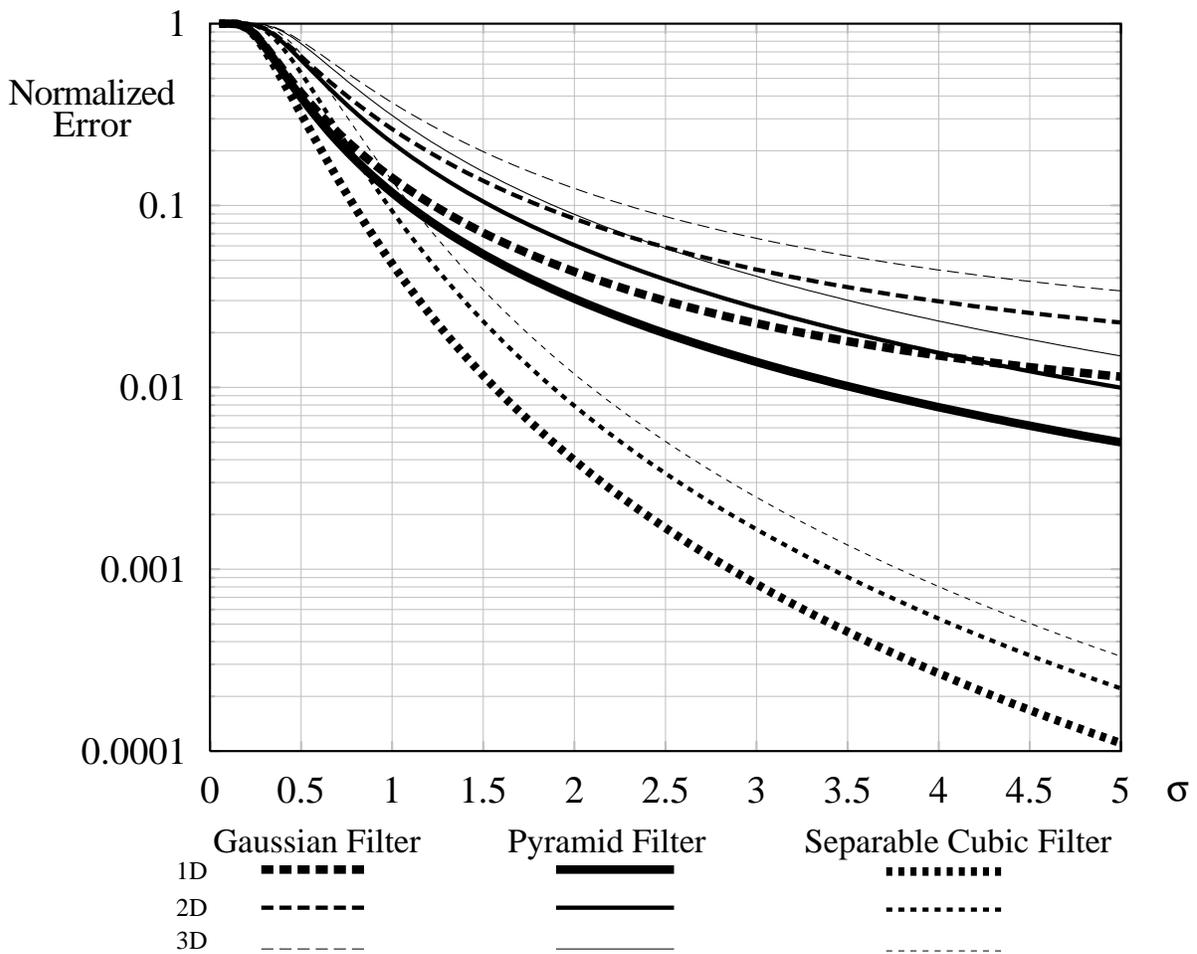


Fig. 4.10 -  $Nerr(\sigma)$  for one, two, and three dimension filters

application with classification and shading functions. The images in figure 4.11 and 4.12 are produced by using the splatting renderer with selectable filters used to obtain figure 3.21. The renderer is augmented with an isosurface classifier and a Phong shader. Resampling is performed before classification and shading. The data gradients are precomputed at the object-lattice points. Resampling with one of the three filters produces data and gradient values at the image-lattice points. Isosurface intersections are found by testing the image-lattice values produced from adjacent data-slices. A pixel whose nearest-slice value is below the surface threshold and whose farthest-slice value is above the threshold has intersected the isosurface. The isosurface location along the image z-axis is approximated by linear interpolation between the two adjacent-slice image-lattice data values. Once the isosurface position is found, the gradient at that point is approximated by linear interpolation of the resampled image-lattice data gradients. After normalization, the interpolated gradient is used in a Phong lighting-calculation to produce the pixel color. There are two lights used for all these examples, one light is above the viewer and the other is to the lower left of the viewer.

Figure 4.11 shows the  $64^3$  *mixed* data rendered by splatting with a) a 2D pyramid filter,

b) a 2D Gaussian filter ( $\sigma = 0.6$ ), and c) a 2D Catmull-Rom cubic filter. The images are all rendered at  $256^2$  pixels. The main visible difference between the images appears at the silhouette edges of the surface. The jagged aliasing-artifacts are more-regular with the cubic and Gaussian filters; they are rougher-looking with the pyramid filter. The rougher edges may suggest features that are not there. Aliasing of the silhouette edges arises because the pixels are point-sampled, and edges contain much high frequency information - this is a form of postaliasing. The aliasing can be eliminated by either modifying the classification function to produce "fuzzy" surfaces or filtering its output - neither approach is used here. The efficient antialiasing of volume-rendered images is a difficult problem and outside the scope of this work. The small sphere, just to the lower-right of center, shows differences between the filters that are more-related to reconstruction accuracy. The sphere is a small feature and best reconstructed in the images made with the cubic and pyramid filters. The image produced with a Gaussian filter shows a smaller and more-deformed sphere.

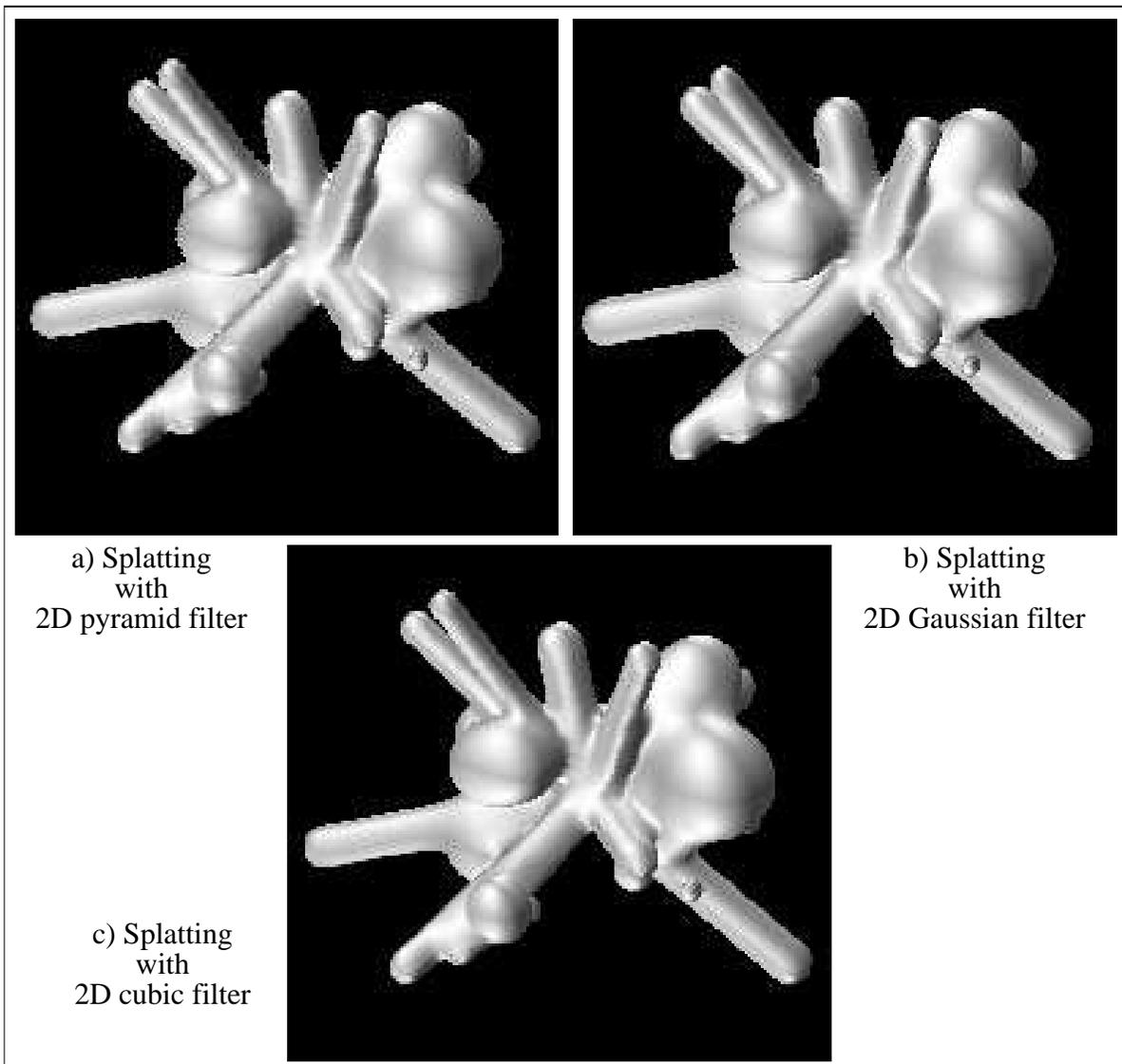


Fig. 4.11 - Isosurface renderings of *mixed* data

Figure 4.12 shows 256×256 images of a 128<sup>3</sup> volume of CT data. These images were also made with the splatting renderer described above in the context of figure 4.11. The images all show artifacts which are clearly not evident on an actual-child's head. The artifacts are due to noisy data and they overwhelm the differences between the filters. It is difficult to determine which image is most correct since there is no standard against which to compare. Real data is often noisy and in that case the choice of reconstruction kernels has only a secondary impact on the rendered images.

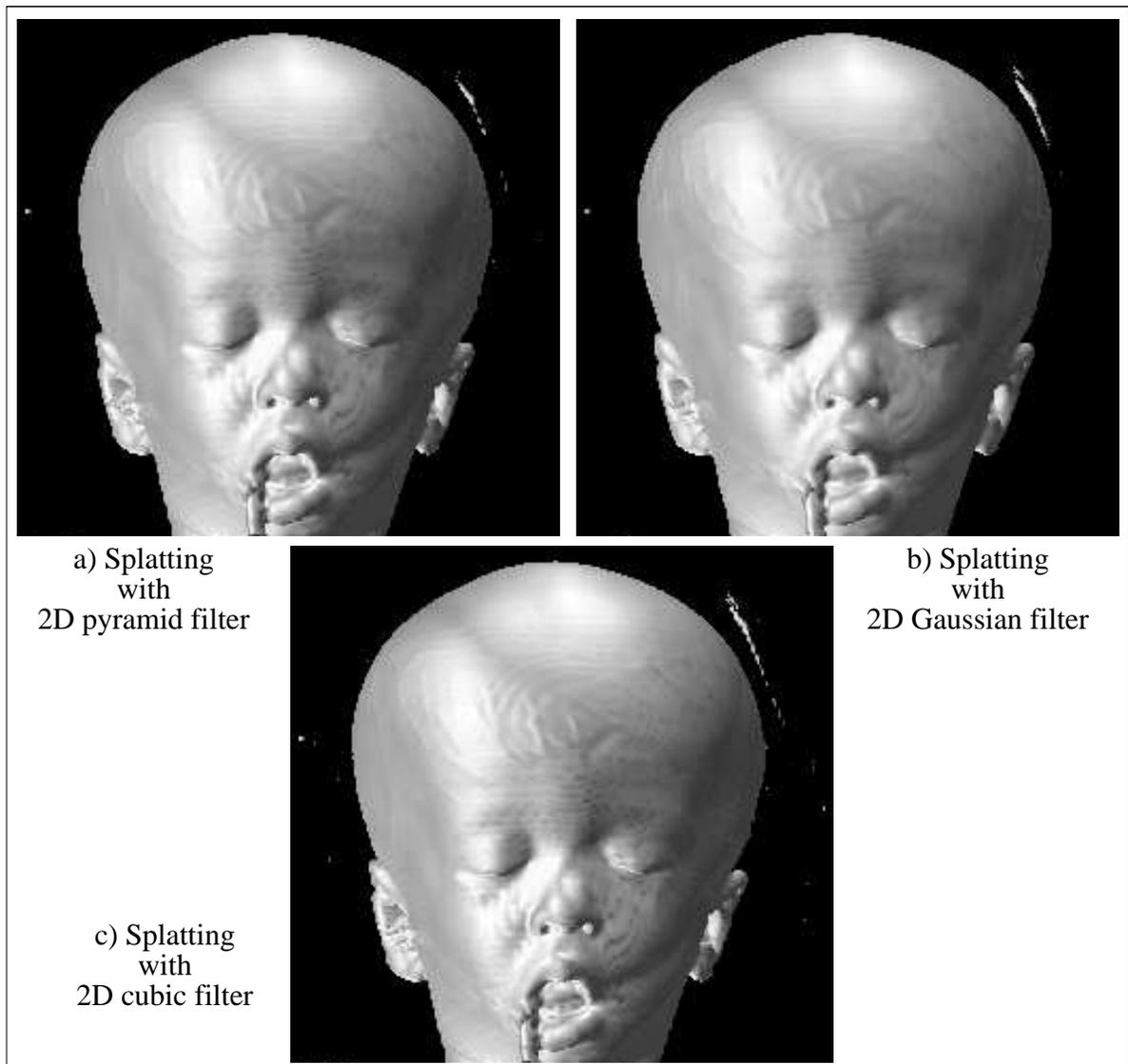


Fig. 4.12 - Isosurface renderings CT data

Data is courtesy of Dr. Frans Zonneveld, N.V. Phillips