Frequency Domain Analysis of B-Spline Interpolation

Abstract

This paper describes B-spline interpolation and compares it with other reconstruction methods, especially in three-dimensional space. We first consider the B-spline bases in the terms of convolution in signal processing. Presented analysis requires careful usage of continuous and discrete representation of B-splines. Emphasis is given to important difference between B-spline interpolation and approximation. The difference is shown through frequency domain analysis, so we derive frequency responses of the B-spline interpolation and approximation. We conclude by demonstrating the use of several reconstruction filters and appropriate gradient estimators in the volume rendering. Exact reconstruction in the volume visualization is very important in many industrial applications, such as material cavity control.

Key words: B-spline, interpolation, volume reconstruction, cubic filters.

1 Introduction

The volume data is generally in the form of large, uniformly spaced three-dimensional scalar or vector field samples. One of the fundamental operations in visualization algorithms is the reconstruction of a continuous function from a set of such samples using interpolation. The simplest approaches are, for example, the nearest neighbor or trilinear interpolation. But the interpolation scheme can have a dramatic effect on the final result. For the better interpolation Mitchell and Netravali [1] introduced class of cubic splines (BC-splines) and classified the parameter space into different regions of dominant reconstruction artifacts.

Further investigations of the BC-splines are made by Marschner and Lobb [2]. In order to verify the reconstruction they proposed error metrics for each of the reconstruction artifacts. For the Catmull-Rom spline and its derivative, Moller, Machiraju, Mueller and Yagel [4] found the most accurate reconstruction among the class of the BC-spline filters. In order to analyze, classify and estimate error of the applied filters, they have used Taylor Series expansion of the convolution sum [3], [4].

The correct application of the B-spline interpolation is of the crucial importance. For image representation and interpolation Unser, Aldroubi and Eden [5] introduce direct and indirect spline transform to achieve efficient scaling mechanisms and resampling of the image. Unlike existing methods for the three-dimensional reconstruction required in volume rendering, we propose interpolation B-spline. This concept will be illustrated by deriving frequency response of the B-spline.

Reconstruction errors may produce incorrect information in the reconstructed volume, which as a consequence produces invalid valuations and classifications in industrial applications.

2 B-splines

B-splines of order n are piecewise polynomial functions of degree n. These functions are differentiable n-1 times, or have derivations up to order n-1. Any continuous n-th degree polynomial piecewise function which is also differentiable n-1 times can be represented using B-spline functions of the same order. In the case of uniform spacing between knot points, such function can be represented in the form [5]:

$$\Phi_n(x) = \sum_{k=-\infty}^{\infty} c_n(k)\beta_n(x-k)\,, \tag{1}$$

were $\beta_n(x)$ is n-th order B-spline function. Sequence $c_n(k)$ denotes B-spline coefficients. B-spline functions of order n are determined by the equation:

$$\beta_n(x) = \frac{1}{n!}\sum_{i=0}^{n+1}(-1)^i \left(\begin{array}{c} n+1 \\ i \end{array}\right)(x-i+\frac{n+1}{2})^n H(x-i+\frac{n+1}{2})\,,$$

where $H(x)$ is Heaviside step-function. This equation is not suitable for analysis in frequency domain, because it defines B-spline function of n-th order on n+1 consecutive segments. Moreover, equation is piecewise polynomial of degree n which leads to repeated partial integration when calculating corresponding frequency response. This is a very tedious process which can be avoided using convolution property of B-splines. If we define $\beta_0(x)$ with:

$$\beta_0(x) = \left\{\begin{array}{ll} 1, & -\frac{1}{2} \le x \le \frac{1}{2} \\ 0, & otherwise \end{array}\right.\,, \tag{2}$$

then B-spline functions of any order can be defined with recursive relation:

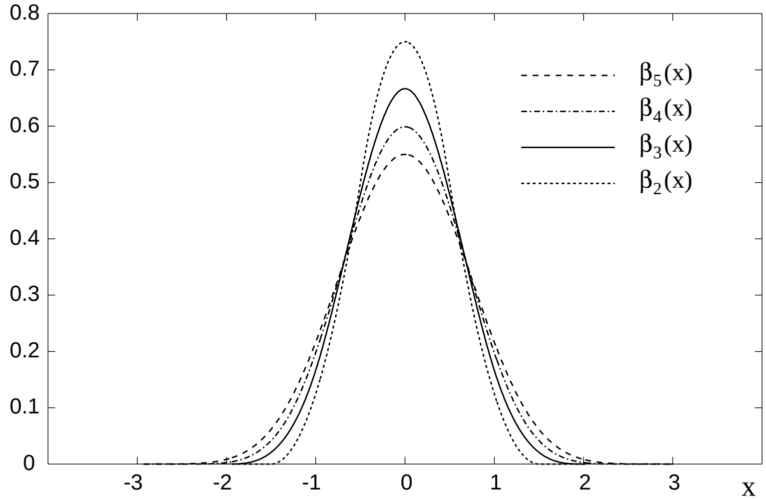

Figure 1: B-spline functions of order n from 2 to 5

$$\beta_n(x) = (\beta_{n-1} * \beta_0)(x)\,, \tag{3}$$

where operator * denotes convolution. In other words, B-spline function of order n can be obtained by convolving B-spline function of order 0 with itself n times. This is the well-known *convolution property* of B-splines which will be used to perform B-spline analysis and interpolation in frequency (Fourier) domain. Few B-splines are shown in Figure 1.

It is important to notice that $\Phi_n(x)$ approximates the values of $c(k)$ (except for n=0 and n=1). Coefficients $c(k)$ in the interpolation case must have specific values so their approximation leads to interpolation of $f(k)$.

In the process of interpolation, we start with discrete sequence $f(k)$. This sequence represents values of a given function (signal) at the knot points[1]. The main goal now is to reconstruct the values of that function at arbitrary position (point) on the real line. B-spline interpolation relies on the function $\Phi_n(x)$. Important condition is that the values of $\Phi_n(x)$ at knot points must match the values of interpolated function $f(x)$. In other words, we have relation $\Phi_n(k) = f(k)$, where k denotes knot points. In order to interpolate given discrete sequence $f(k)$, one must first calculate sequence $\{c_n(k)\}$ of corresponding B-spline coefficients. This sequence is uniquely determined by $f(k)$. The most convenient way is to use matrix representation of the problem. This approach reduces to inverting tridiagonal matrix using LU decomposition or Gauss elimination. An alternative method is to use signal processing approach. In this case, sequence $c_n(k)$ is derived from $f(k)$ by discrete filtering.

3 Continuous Representation of Discrete Sequences

Analysis of B-spline interpolation involves both discrete and continuous relations. This may be impractical in frequency domain analysis because of the requirement to use both *discrete* and *continuous* Fourier transform at the same time. So, before proceeding with analysis of B-splines, it is useful to introduce appropriate mechanism which will enable us to express discrete sequences in continuous form. In order to do that, we will introduce impulse-functions and their properties.

First we introduce the term "impulse function" which denotes any function in continuous domain (or more correctly: distribution) that has a form of the Dirac's δ-distributions train with varying amplitudes. Such functions are completely determined with discrete sequence of amplitudes of the δ-distributions, and the distance between two adjacent pulses. Here, only uniform case of spacing between pulses with distance equal to one will be considered. Impulse functions will be denoted with "$\hat{}$" over the name of the function.

We may define two such (impulse) functions $\hat{\varphi}(x)$ and $\hat{\psi}(x)$ with:

$$\begin{aligned}\hat{\varphi}(x) &= \textstyle\sum_{k=-\infty}^{\infty} \varphi(k)\delta(x-k)\\ \hat{\psi}(x) &= \textstyle\sum_{l=-\infty}^{\infty} \psi(l)\delta(x-l)\,.\end{aligned} \tag{4}$$

where $\varphi(k)$ and $\psi(l)$ are sequences of δ-distribution amplitudes centered at position k. Then their convolution $\hat{h}(x)$ is:

[1] Although spacing between knot points can be arbitrary, here it is assumed to be unity. This can be done without loss of generality, because any other constant value of knot-point spacing can be reduced to unity.

$$\hat{h}(x) = (\hat{\varphi} * \hat{\psi})(x) = \int_{-\infty}^{\infty} \hat{\varphi}(\tau)\hat{\psi}(x-\tau)d\tau = \\ = \sum_{u=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} \varphi(k)\psi(u-k)\right)\delta(x-u)\,. \tag{5}$$

This equation shows that *continuous convolution* of impulse functions $\hat{\varphi}(x)$ and $\hat{\psi}(x)$ is equal to impulse function $\hat{h}(x)$ whose amplitudes are formed as *discrete convolution* between sequences of δ-amplitudes from functions $\hat{\varphi}(x)$ and $\hat{\psi}(x)$. Moreover, if we have impulse function $\hat{\varphi}(x)$ and continuous function $r(x)$, their convolution $\varphi_r(x)$ is given with:

$$\varphi_r(x) = (\hat{\varphi} * r)(x) = \sum_{k=-\infty}^{\infty} \varphi(k)r(x-k)\,. \tag{6}$$

4 B-Spline Interpolation

As stated before, the problem of B-spline interpolation consists of two major steps: finding B-spline coefficients, and then calculating the values of function $\Phi_n(x)$ at the required position (using previously calculated coefficients).

At knot points, by the definition of interpolation, functions $\Phi_n(x)$ and $f(x)$ must evaluate to equal values:

$$\Phi_n(k) = f(k) = \sum_{l=-\infty}^{\infty} c_n(l)\beta_n(k-l)\,,$$

which is discrete convolution analogous to parenthesized expression in equation (5). Now we will apply mentioned property of impulse functions by substituting $\hat{f}(x)$, $\hat{c}_n(x)$ and $\hat{\beta}_n(x)$ in place of $\hat{h}(x)$, $\hat{\varphi}(x)$ and $\hat{\psi}(x)$ respectively into equation (5), which yields:

$$\hat{f}(x) = (\hat{c}_n * \hat{\beta}_n)(x)\,. \tag{7}$$

Functions $\hat{f}(x)$, $\hat{c}_n(x)$ and $\hat{\beta}_n(x)$ are impulse functions built from sequences $f(k)$, $c_n(k)$ and $\beta_n(k)$ respectively, where we have:

$$\hat{\beta}_n(x) = \sum_{k=-\infty}^{\infty} \beta_n(k)\delta(x-k)\,. \tag{8}$$

Equation (7) relates sequence $f(k)$ and B-spline coefficients $c_n(k)$ in the form of *continuous* convolution. This enables further analysis of B-splines using continuous Fourier transform. Transformation of identity (7) gives:

$$\tilde{F}(\omega) = \tilde{C}_n(\omega)\tilde{B}_n(\omega)\,, \tag{9}$$

where $\tilde{F}(\omega)$, $\tilde{C}_n(\omega)$ and $\tilde{B}_n(\omega)$ are Fourier domain representations of $\hat{f}(x)$, $\hat{c}_n(x)$ and $\hat{\beta}_n(x)$ respectively[2]. Fourier transform of $\hat{c}_n(x)$ can easily be derived from (9) in the form

$$\tilde{C}_n(\omega) = \frac{1}{\tilde{B}_n(\omega)}\tilde{F}(\omega)\,. \tag{10}$$

This equation is valid only if $\tilde{B}_n(\omega)$ has no zeroes on the frequency axis ω. This is satisfied for all B-splines. Now we may define new impulse function $\hat{p}_n(x)$ whose Fourier transform is

$$\tilde{P}_n(\omega) = \frac{1}{\tilde{B}_n(\omega)}\,,$$

and write $\tilde{C}_n(\omega) = \tilde{P}_n(\omega)\tilde{F}(\omega)$. In the spatial domain this equation has the form of convolution between $\hat{f}(x)$ and $\hat{p}_n(x)$, i.e.:

$$\hat{c}_n(x) = (\hat{p}_n * \hat{f})(x)\,. \tag{11}$$

Property (6) implies that function $\Phi_n(x)$ from (1) can be represented as continuous convolution:

$$\Phi_n(x) = (\hat{c}_n * \beta_n)(x)\,. \tag{12}$$

Function $\hat{p}_n(x)$ represents discrete B-spline *inverse filter*. It enables computation of B-spline coefficients in frequency domain. Substitution of $\hat{c}_n(x)$ from (11) into (12) and using commutation property of convolution gives:

$$\Phi_n(x) = (\hat{f} * \hat{p}_n * \beta_n)(x)\,. \tag{13}$$

In frequency domain, equation has the following form:

$$\mathcal{F}[\Phi_n(x)] = \tilde{F}(\omega)\frac{B_n(\omega)}{\tilde{B}_n(\omega)}\,. \tag{14}$$

This equation pair forms the basis for frequency domain analysis B-spline interpolation.

[2] It is well known property of Fourier transform which states that the frequency domain of impulse-function is always periodic. Tilde character over the functions symbolizes their periodicity.

5 Frequency Domain Analysis

The properties shown in the previous section can simplify analysis of B-splines in frequency domain. To find frequency response, we must first determine impulse response of the B-spline interpolation.

Its impulse response can be obtained by interpolation of single δ-impulse centered at zero. The first step is substitution of δ-distribution as input impulse train function $\hat{f}(x)$ and calculation of corresponding B-spline coefficients. This leads to:

$$\begin{gathered} \hat{f}(x) = \delta(x) \quad \Rightarrow \\ \eta_n(x) = (\delta * \hat{p}_n * \beta_n)(x) = (\hat{p}_n * \beta_n)(x) \,. \end{gathered} \tag{15}$$

Polynomial function that corresponds to B-spline interpolation of $\delta(x)$ as input function (impulse response) is denoted as $\eta_n(x)$. As $\hat{p}_n(x)$ is impulse function, it is clear that it may be expressed as:

$$\hat{p}_n(x) = \sum_{k=-\infty}^{\infty} p_n(k)\delta(x-k) \,.$$

Equation (15) can be simplified using property expressed in (6):

$$\eta_n(x) = (\hat{p}_n * \beta_n)(x) = \sum_{k=-\infty}^{\infty} p_n(k)\beta_n(x-k) \,. \tag{16}$$

Using impulse response, polynomial function $\Phi_n(x)$ from (13) can be expressed in the following form:

$$\Phi_n(x) = (\hat{f} * \eta_n)(x) \,.$$

Frequency response $H_n(\omega)$ is given with

$$H_n(\omega) = \tilde{P}_n(\omega)B_n(\omega) = \frac{B_n(\omega)}{\tilde{B}_n(\omega)} \,, \tag{17}$$

which follows from (14). It is evident that numerator $B_n(\omega)$ and denominator $\tilde{B}_n(\omega)$ must be known in order to calculate frequency response of B-spline interpolation. As stated before, denominator $\tilde{B}_n(\omega)$ has no zeroes on the frequency axis. There are certain problems with non-centered B-splines, where $\tilde{B}_n(\omega)$ can have zeroes for even values of n. In that case $\hat{\beta}_n(x)$ has non-stable inverse filter, but impulse response of such B-spline interpolation is still stable if Fourier transform of continuous B-spline $B_n(\omega)$ has zeroes at the same points as $\tilde{B}_n(\omega)$. At such points there is additional condition on $B_n(\omega)$; it must approach zero "faster" than $\tilde{B}_n(\omega)$.

Fourier transform of $\beta_0(x)$ can be easily found to be:

$$B_0(\omega) = \frac{\sin(\frac{\omega}{2})}{\frac{\omega}{2}} = \operatorname{sinc}(\frac{\omega}{2\pi}) \,, \tag{18}$$

where usual definition for *sinc* function is used:

$$\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \,.$$

Frequency response of B-spline function of any order n can be derived from its definition (3), using the convolution theorem for Fourier transform. Hence, $B_n(\omega)$ is given with

$$B_n(\omega) = B_0^{n+1}(\omega) = \operatorname{sinc}^{n+1}(\frac{\omega}{2\pi}) \,. \tag{19}$$

Numerator of (17) is frequency representation of sampled (impulse train modulated) B-spline function $\hat{\beta}_n(x)$. As shown in (8) this function is completely determined by its samples $\beta_n(k)$. Its Fourier transform is given with:

$$\begin{aligned} \tilde{B}_n(\omega) &= \int_{-\infty}^{\infty} \left(\textstyle\sum_{k=-\infty}^{\infty} \beta_n(k)\delta(x-k)\right) e^{-i\omega x} dx = \\ &= \textstyle\sum_{k=-\infty}^{\infty} \beta_n(k) e^{-ik\omega} \,. \end{aligned} \tag{20}$$

Equation follows immediately after interchanging the order of the integration and summation using the absorption property of δ-distributions:

$$\int_{-\infty}^{\infty} \delta(x-k) e^{-i\omega x} dx = e^{-ik\omega} \,.$$

In order to calculate the second row of (20), which is the discrete Fourier transform of sequence $\beta_n(k)$, we must find values (samples) $\beta_n(k)$ of B-spline functions. From recursive convolution relation (3) it is evident that all B-splines are even functions, so we may state: $\beta_n(-k) = \beta_n(k)$. Moreover, B-splines are spatially-limited functions so their sequence of samples $\beta_n(k)$ is of finite length. The values of $\beta_n(k)$ vanish for $|k| > \lfloor n/2 \rfloor$,

Table 1: Discrete B-splines from orders 1 to 5

k	0	1	2
$\beta_1(k)$	1	0	0
$\beta_2(k)$	$3/4$	$1/8$	0
$\beta_3(k)$	$2/3$	$1/6$	0
$\beta_4(k)$	$230/384$	$76/384$	$1/384$
$\beta_5(k)$	$66/120$	$26/120$	$1/120$

Table 2: Frequency responses of B-spline interpolation for n=1 to 5

$H_1(\omega)$	$\frac{\mathrm{sinc}^2(\frac{\omega}{2\pi})}{1}$
$H_2(\omega)$	$\frac{4\,\mathrm{sinc}^3(\frac{\omega}{2\pi})}{3+\cos(\omega)}$
$H_3(\omega)$	$\frac{3\,\mathrm{sinc}^4(\frac{\omega}{2\pi})}{2+\cos(\omega)}$
$H_4(\omega)$	$\frac{192\,\mathrm{sinc}^5(\frac{\omega}{2\pi})}{115+76\cos(\omega)+\cos(2\omega)}$
$H_5(\omega)$	$\frac{60\,\mathrm{sinc}^6(\frac{\omega}{2\pi})}{33+26\cos(\omega)+\cos(2\omega)}$

were operator $\lfloor\cdot\rfloor$ denotes the largest integer value not greater than its argument. Table 1 shows positive half of several discrete B-splines. Using those facts, expression from (20) may be rewritten:

$$\tilde{B}_n(\omega) = \beta_n(0) + 2\sum_{k=1}^{\lfloor\frac{n}{2}\rfloor} \beta_n(k)\cos(k\omega)\,.$$

Frequency domain of B-spline prefilter $\hat{p}_n(x)$ is reciprocal of this function. According to (17), frequency response of B-spline interpolation is a product between frequency domain representations of B-spline prefilter and corresponding B-spline function. Table 2 shows frequency responses of few B-splines.

The difference between B-spline interpolation and approximation is apparent in frequency domain. Frequency response of n-th order B-spline approximation is given with (19). On the other side, frequency response of B-spline interpolation of the same order has additional factor: Fourier transform of prefilter. Although the *surface* under every B-spline is equal to one, the increasing order of n causes its *energy* to decrease. When order n in (19) runs towards infinity, this function has value *one* at $\omega = 0$ and *zero* otherwise. Such function would smooth any input impulse function and turn periodic one into a (continuous) constant function, a non-periodic one into zero function. In other words, it preserves only DC component of interpolated function.

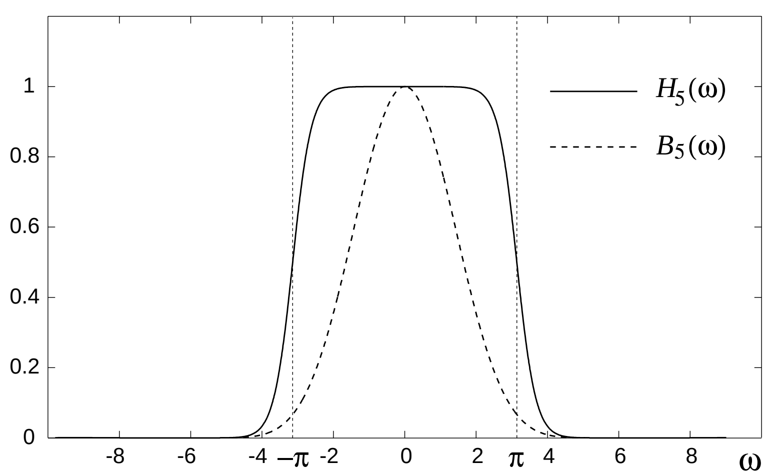

Figure 2: Frequency responses of B-spline interpolation and approximation of order 5

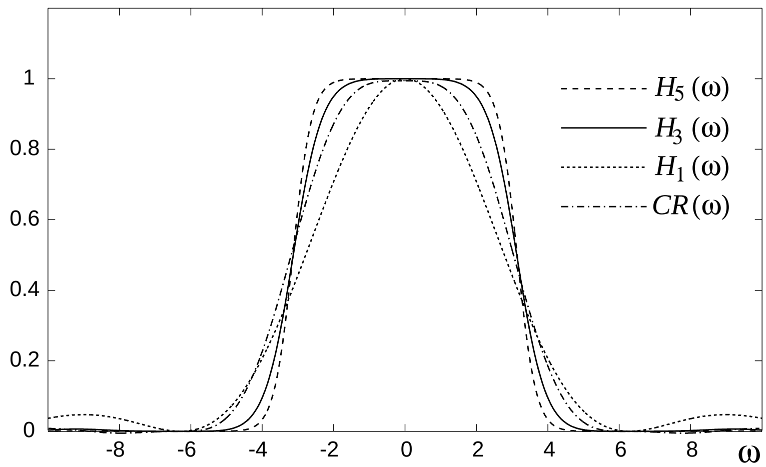

Figure 3: Frequency responses of B-spline interpolation (for order $n = 1$, 3, and 5) and 3-rd order Catmull-Rom spline

Frequency response of B-spline interpolation has previously mentioned additional factor from B-spline inverse filter. This works as a "correction" to the B-spline, so its frequency response is wider. Increasing order n leads to frequency response which is getting closer to the ideal lowpass filter. Figure 2 shows frequency responses of B-spline approximation of order 5 and corresponding B-spline interpolation. The difference in the interval $[-\pi, \pi]$ is obvious. Frequency responses from Table 2 are shown in Figure 3, where one can see that increasing order of B-spline interpolation leads to the better approximation of ideal lowpass filter. Also shown is a frequency response of Catmull-Rom spline, which was recommended by several researchers as a very good reconstruction filter [1], [6].

6 Results

Presented analysis shows the advantages of the B-spline interpolation over the widely used (and often misused) B-spline approximation. Various papers show that good numerical features of the reconstruction filter do not imply acceptable optical features. For complete evaluation of proposed method, we will present obtained results.

Four images shown in Figure 4 were generated by the same volume ray-casting algorithm applied to standard "MRbrain" dataset. Before taking the projection, volume dataset has been downsampled from originally 256^3 samples to 64^3 samples using gaussian filter, with apropriate cutoff frequency. Shown images differ in the reconstruction and derivation filters which were used by ray-caster. As a *derivative* filter, we have used *derivation of reconstruction* filter [6].

Figure 4a) shows image using trilinear interpolation. Strong visual artifacts are visible. Figure 4b) were generated by the well-known Catmull-Rom spline. Artifacts are suppressed, but still very visible. Using B-spline approximation of the 3-rd order yields Figure 4c). This image is virtually free of visible artifacts, but has another disadvantage; B-spline approximation has excessivelly blurred the image. Finally, Figure 4d) shows the resulting image obtained using B-spline interpolation of the 3-rd order. This image appears to be most satisfactory.

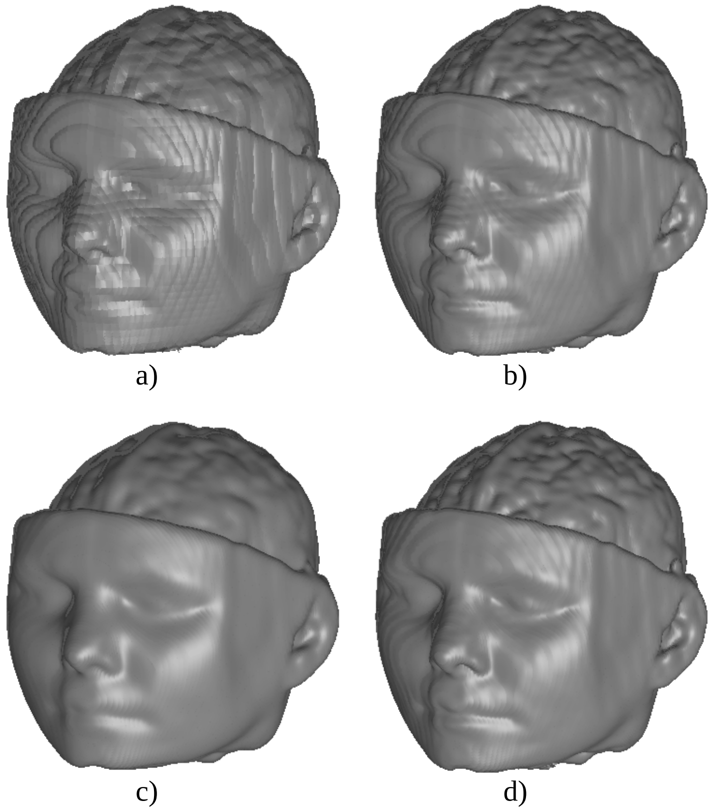

Figure 4: Four different reconstruction schemes; a) Trilinear interpolation, b) Catmull-Rom spline interpolation, c) 3-rd order B-spline approximation, and d) 3-rd order B-spline interpolation

References

[1] D. P. Mitchell and A. N. Netravali, "Reconstruction Filters in Computer Graphics", Computer Graphics, Vol 22, No. 4, pp 221-228, Aug. 1988.

[2] S. R. Marschner and R. J. Lobb, "An Evaluation of Reconstruction Filters for Volume Rendering", Proc. Visualization '94., pp 100-107, IEEE CS Press, Oct. 1994.

[3] R. Machiraju and R. Yagel, "Reconstruction Error Characterization and Control: A Sampling Theory Approach", IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 4, December 1996.

[4] T. Moller, R. Machiraju, K. Mueller and R. Yagel, "Evaluation and Design of Filters Using a Taylor Series Expansion", IEEE Transactions on Visualization and Computer Graphics, Vol. 3, No. 2, April-June 1997.

[5] M. Unser, A. Aldroubi and M. Eden, "Fast B-spline Transforms for Continuous Image Representation and Interpolation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 3, March 1991.

[6] M. J. Bentum, B. A. Lichtenbelt, T. Malzbender, "Frequency Analysis of Gradient Estimators in Volume Rendering", IEEE Transactions on Visualization, Vol. 2, No. 3, pp 242-253, Sept. 1996.