

# Asymptotic error of cubic B-spline interpolation using prefiltering

Yvonne Percan · Stefan Zellmann · Ulrich Lang

Received: 17 September 2014 / Accepted: 23 November 2014 / Published online: 30 November 2014  
© The Author(s) 2014. This article is published with open access at Springerlink.com

**Abstract** A popular class of reconstruction filters that are used in signal and image processing is based on cubic B-splines. One reason for their popularity is the fact that they can be efficiently implemented. This is specifically true with modern GPUs where cubic B-spline filtering can be implemented by means of linearly interpolating texture fetches so that the actual number of memory accesses can be significantly reduced. The curve obtained from filtering with the cubic B-spline does in general not interpolate the original data set. The latter can however be achieved by applying a prefiltering step that transforms the original data set. We study the asymptotic behavior of the reconstruction error of the cubic B-spline *interpolation* filter using a state of the art method that is based on a Taylor series expansion and that was carefully adjusted to accommodate the infinite support of this reconstruction filter.

**Keywords** Splines · Reconstruction · Asymptotic error

## 1 Introduction

Practical implementations of reconstruction algorithms from signal or image processing rely on filters with a finite support. GPUs are a popular target platform for 2-d or 3-d image processing algorithms because of their high peak performance. GPUs provide hardware implementations for zeroth and first-order reconstruction in one, two,

---

Y. Percan · S. Zellmann (✉) · U. Lang  
Chair of Computer Science, University of Cologne, Weyertal 121, 50931 Cologne, Germany  
e-mail: zellmann@uni-koeln.de

Y. Percan  
e-mail: ypercan@gmx.de

U. Lang  
e-mail: lang@uni-koeln.de

and three dimensions. While the *theoretical* time it takes the first-order reconstruction algorithm to execute is  $2^D$  times as high ( $D$  here denotes dimension) compared to the zeroth-order reconstruction algorithm, *in practice* both implementations incur the same latency on contemporary GPUs. While zeroth and first-order reconstruction is unambiguous - application of a zeroth-order filter in general results in a step function while a first-order filter results in a set of connected line segments - there exist different ways to reconstruct a data set when using higher-order functions. Two popular reconstruction filters based on piecewise cubic functions are the cubic B-spline [8]

$$\beta_1^3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^3}{2} & : 0 \leq |x| < 1 \\ \frac{(2-|x|)^3}{6} & : 1 \leq |x| < 2 \\ 0 & : 2 \leq |x| \end{cases}, \tag{1}$$

as well as the Catmull-Rom spline [1]

$$c(x) = \begin{cases} \frac{2}{3}|x|^3 - \frac{5}{2}|x|^2 + 1 & : 0 \leq |x| < 1 \\ -\frac{1}{2}|x|^3 + \frac{5}{2}|x|^2 - 4|x| + 2 & : 1 \leq |x| < 2 \\ 0 & : 2 \leq |x| \end{cases}. \tag{2}$$

Reconstruction using one of the two filters in  $I$ -d can be expressed as the convolution sum

$$f_x = w_0(\tau)f_{i-1} + w_1(\tau)f_i + w_2(\tau)f_{i+1} + w_3(\tau)f_{i+2}, \tag{3}$$

where  $x \in \mathbb{R}$ ,  $i = \lfloor x \rfloor$ ,  $\tau = x - i$ ,  $f_{i-1}$  through  $f_{i+2}$  are the samples that are available from the original series, and  $f_x$  is the reconstructed value.  $w_0(\tau)$  through  $w_3(\tau)$  represent weight functions that are obtained by either letting  $w_0(\tau) = \beta_1^3(\tau + 1)$ ,  $w_1(\tau) = \beta_1^3(\tau)$ ,  $w_2(\tau) = \beta_1^3(\tau - 1)$ , and  $w_3(\tau) = \beta_1^3(\tau - 2)$  for the cubic B-spline, or by letting  $w_0(\tau) = c(\tau + 1)$ ,  $w_1(\tau) = c(\tau)$ ,  $w_2(\tau) = c(\tau - 1)$ , and  $w_3(\tau) = c(\tau - 2)$  for the Catmull-Rom spline. Unlike for the Catmull-Rom spline, the convolution sum from (3) can be reformulated as follows for the cubic B-spline [14]:

$$f_x = g_0(\tau) f_{\tau-h_0(\tau)} + g_1(\tau) f_{\tau+h_1(\tau)}, \tag{4}$$

with

$$\begin{aligned} g_0(\tau) &= w_0(\tau) + w_1(\tau), \\ g_1(\tau) &= w_2(\tau) + w_3(\tau), \\ h_0(\tau) &= \tau - 0.5 + \frac{w_1(\tau)}{w_0(\tau) + w_1(\tau)}, \\ h_1(\tau) &= \tau + 1.5 + \frac{w_3(\tau)}{w_2(\tau) + w_3(\tau)}. \end{aligned} \tag{5}$$

This reformulation is only possible for functions which fulfill the convex combination property

$$af_i + bf_{i+1} \Leftrightarrow (a + b)f_{\frac{i+b}{a+b}}, \tag{6}$$

where  $a, b \in \mathbb{R}$ , so that  $0 \leq \frac{b}{a+b} \leq 1$ . Note that in (4),  $\tau - h_0(\tau) \in \mathbb{R}$  and  $\tau + h_1(\tau) \in \mathbb{R}$ , while  $i - 1$  through  $i + 2$  from (3) are natural numbers. GPUs provide memory

accesses like the one on the right-hand side from (6) in hardware. With the above reformulation, the number of memory accesses necessary for reconstruction using the cubic B-spline filter can thus be reduced from  $4^D$  zeroth-order texture fetches to  $2^D$  first-order texture fetches. This is advantageous because memory access operations incur a much higher latency than e.g. arithmetic or comparison operations on von Neumann architectures like GPUs. Unfortunately, the convex combination property from (6) does not hold for the Catmull-Rom spline. However, the cubic B-spline does in general not interpolate the original dataset, but only approximates it. For scientific applications, this behavior is potentially undesirable, so that those applications resort to interpolation using Catmull-Rom splines which incur  $4^D$  memory accesses.

An alternative approach is to prefilter the data set during a preprocessing step [17] so that cubic B-spline filtering using the altered data set results in a curve that actually reconstructs the original data set (cf. Fig. 1). Prefiltering is achieved by calculating coefficients  $c_k \forall k \in \mathbb{Z}$  so that

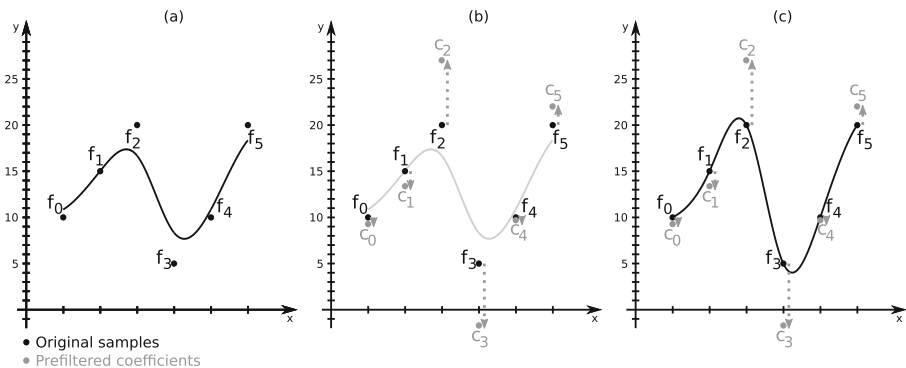
$$f_k = \sum_{l \in \mathbb{Z}} c_l b_1^3(x-l) \Big|_{x=k}, \tag{7}$$

where  $b_1^3(x)$  is the discrete cubic B-spline, i.e.

$$b_1^3(k) = \beta_1^3(x) \Big|_{x=k}. \tag{8}$$

Solving for  $c_k$  thus yields the following convolution of the inverse discrete cubic B-spline with the original series  $f_k$  to obtain the new coefficients and that way an appropriately prefiltered data set so that the cubic B-spline filter reconstructs the original data set:

$$c_k = (b_1^3)^{-1}(k) * f_k. \tag{9}$$



**Fig. 1** Cubic B-spline interpolation with prefiltering. **a** The cubic B-spline only approximates the original data set. **b** New coefficients  $c_k$  are computed. **c** The new curve interpolates the original data set

Because this filter reconstructs the original data set, the literature refers to it as cubic B-spline *interpolation* [15–17]. This publication is concerned with the asymptotic error of the cubic B-spline interpolation reconstruction filter. The remainder of the publication is structured as follows. In Section 2, we review related publications that are concerned with numerical errors in scientific visualization applications. In Section 3, we formally derive the error bound of the cubic B-spline interpolation filter. Section 4 concludes this publication. In the [Appendix](#), we present some short proofs that accompany the derivation of the asymptotic error bound from Section 3.

## 2 Related work

The recent popularity of cubic B-spline interpolation is based on the fact that the filtering procedure can be implemented efficiently on GPUs [12, 14]. Ruijters and Thévenaz [13] also presented an efficient implementation of the prefiltering step. Nehab et al. [11] showed how the aforementioned implementation can be improved in terms of memory access patterns and reported significant speedups. Their approach employs various optimizations to the original algorithm, such as overlapping of the causal and anticausal phases that the original implementation incorporates, or data sharing between compute kernels without having to store temporary data in DDR memory. Champagnat and Le Sant [2] proposed to truncate and normalize the infinite prefilter to accelerate its application on the original data set. They evaluated their approach using an error metric on a series of test images and found that the deviation was negligible.

Error estimation is an important topic in the field of scientific visualization because applications like those from medical imaging or engineering rely on algorithms with a minimal error. Because with numerical methods one can usually not avoid making errors, it is crucial that error bounds can be sufficiently estimated. Error estimation and classification for 3-d imaging algorithms that rely heavily on reconstruction were e.g. provided by Marschner and Lobb [7] or Etienne et al. [3]. Machiraju and Yagel [6] evaluated the error bound of reconstruction filters based on filter size. Keys [4] derived the asymptotic error of the Catmull-Rom spline (which his publication refers to as *cubic convolution interpolation*) as being of order  $O(T^3)$ , with  $T$  denoting the sampling distance.

Lu and Vaswani [5] suggested a method to reconstruct sparse signals, i.e. here the reconstruction filter was not considered to *introduce* additional errors, but rather than that a carefully crafted filter can help to *reduce* the error that is incurred by an incomplete signal.

While we focus on the asymptotic behavior of the reconstruction error, image reconstruction may also suffer from precision issues. Ruijters et al. [12] noted precision problems with cubic B-spline filtering, which especially emerged with the original implementation proposed by Sigg and Hadwiger [14]. The latter precalculated the terms from (5) for a set of finite coordinates rather than calculating them on-the-fly.

Our contribution adapts the approach of Mller et al. [9, 10] to estimate the asymptotic error of cubic B-spline interpolation based on the sampling distance, so that their method is reviewed in more detail. We consider the general convolution of a reconstruction filter  $h$  with a set of samples  $f_k$  to obtain the reconstructed signal

$$f_x = \sum_{k \in \mathbb{Z}} f_k h\left(\frac{x}{T} - k\right), \tag{10}$$

where  $T$  again denotes the sampling distance. By letting  $T \rightarrow 0$ , the signal is reconstructed using more samples and thus, the asymptotic error is reduced. Since  $T$  is determined during the data acquisition phase rather than being a variable during reconstruction, the method by Mller et al. accounts for the reliability of a filter  $h$  given a fixed resolution of the image or grid that accommodates the original data series. Assuming that the first  $N + 1$  derivatives of  $f_k$  exist, the Taylor series expansion in  $f$  about  $x$

$$f_k = \sum_{n=0}^N \frac{f_x^{(n)}}{n!} (kT - x)^n + \frac{f_\xi^{(N+1)}}{(N + 1)!} (kT - x)^{(N+1)}, \tag{11}$$

where  $x \leq \xi \leq kT$  and  $f^{(n)}$  denotes the  $n$ th order derivate of  $f_x$ , is substituted into (10) as follows:

$$f_x = \sum_{n=0}^N a_n f_x^{(n)} + r_N \tag{12}$$

with coefficients

$$a_n = \frac{1}{n!} \sum_{k \in \mathbb{Z}} (kT - x)^n h\left(\frac{x}{T} - k\right) \tag{13}$$

and the residual term

$$r_N = \frac{1}{(N + 1)!} \sum_{k \in \mathbb{Z}} f_\xi^{(N+1)} (kT - x)^{(N+1)} h\left(\frac{x}{T} - k\right) \tag{14}$$

of which the latter in the following no longer needs to be considered. The coefficients  $a_n$  are further simplified to:

$$a_n = \frac{T^n}{n!} \sum_{k=-M}^M (k - \tau)^n h(\tau - k) \tag{15}$$

by letting  $\tau = \frac{x}{T} - \lfloor \frac{x}{T} \rfloor$  be a normalized offset and considering only filters with finite extent  $M$ . By expanding (12) as follows:

$$f_x = a_0 f_x^{(0)} + a_1 f_x^{(1)} + a_2 f_x^{(2)} + \dots + a_N f_x^{(N)} + r_N \tag{16}$$

one can see that in the case of *ideal* reconstruction, the residual term  $r_N$  as well as all coefficients  $a_n$  should be zero, except for  $a_0$ , which should be one. The authors then classified the reconstruction filter  $h$  by choosing the largest  $N$  so that all  $a_n$ ,  $1 \leq n \leq N$  evaluate to zero. The reconstruction filter  $h$  is said to be a  $k$ th degree error filter ( $k$ -EF) if the asymptotic behavior of the error is proportional to  $O(T^k)$ . With

$T \rightarrow 0$ , higher values for  $k$  are in general preferred. The authors evaluated several cubic filters and reported that the cubic B-spline from (1) is a 2-EF, and that the Catmull-Rom spline from (2) is a 3-EF.

### 3 Asymptotic error of the cubic B-spline interpolation filter

In the ensuing section, we show how the approach by Mller et al. can be adapted to derive the asymptotic error behavior of the cubic B-spline interpolation filter. Because this filter has *quasi* infinite support, the approach by the authors is however not applicable directly, so that the formal proof of the filter’s error bound is preceded by a general discussion on how to adapt the approach by Mller et al. to accommodate this specific filter.

#### 3.1 Preliminary considerations

Equation 13 can in general only be solved if the sum over  $k \in \mathbb{Z}$  has finite support like in (15), or if the sum tends towards a constant expression. However, neither is the case with the cubic B-spline interpolation filter. This becomes obvious when considering the following observations.

In order to apply the method by Mller et al. to the cubic B-spline interpolation filter with its prefiltering step, one has to substitute the prefiltered coefficients  $c_k$  into (10) and to substitute the cubic B-spline for the filter  $h$ :

$$f_x = \sum_{k \in \mathbb{Z}} c_k \beta_1^3 \left( \frac{x}{T} - k \right). \tag{17}$$

Due to the fact that the  $c_k$  are calculated by convolving each sample from the original series with the inverse discrete cubic B-spline (cf. (9)), (17) becomes

$$f_x = \sum_{k \in \mathbb{Z}} ((b_1^3)^{-1}(k) * f_k) \beta_1^3 \left( \frac{x}{T} - k \right). \tag{18}$$

In order to apply Mller et al.’s approach, the *original* data series must be isolated to allow for Taylor series expansion in  $f$  about  $x$ . This is achieved by exploiting the associativity of the convolution operation with scalar multiplication as follows:

$$f_x = \sum_{k \in \mathbb{Z}} f_k \sum_{l \in \mathbb{Z}} (b_1^3)^{-1}(l) \beta_1^3 \left( \frac{x}{T} - l - k \right). \tag{19}$$

Unser [16] showed that the *inverse* discrete cubic B-spline is given by

$$(b_1^3)^{-1}(l) = \frac{-6\alpha}{(1 - \alpha^2)} \alpha^{|l|} \tag{20}$$

with  $l \in \mathbb{Z}$  and  $\alpha = \sqrt{3} - 2$ , so that (19) becomes

$$f_x = \sum_{k \in \mathbb{Z}} f_k \sum_{l \in \mathbb{Z}} \frac{-6\alpha}{(1 - \alpha^2)} \alpha^{|l|} \beta_1^3 \left( \frac{x}{T} - l - k \right). \tag{21}$$

Now, simplifying so that coefficients like those from (15) are obtained is no longer possible because the sum over  $l \in \mathbb{Z}$  has infinite support and does not tend towards a constant expression. Thus, the expression

$$h(x) = \frac{-6\alpha}{(1 - \alpha^2)} \sum_{l \in \mathbb{Z}} \alpha^{|l|} \beta_1^3\left(\frac{x}{T} - l - k\right) \tag{22}$$

with  $k \in \mathbb{Z}$  represents a filter with an infinite support. Note that this notion of infinity is only theoretical because only parts of the filter are actually applied at reconstruction time, while the “infinite” part of the filter is applied during a preprocessing step. Furthermore, the prefiltering procedure is typically applied to a finite set of samples like the pixels of an image or the data items at the cells of a general grid. Although the sum from (22) cannot simply be adapted to represent a filter with a finite extent from  $-M$  to  $M$ , the following considerations allow to estimate the summands and are used during the formal derivation in the following subsection.

While the sum over  $l \in \mathbb{Z}$  from (22) has infinite support, the extent of the cubic B-spline  $\beta_1^3\left(\frac{x}{T} - l - k\right)$  is finite. Specifically, only four summands of the infinite sum will not equal zero following from the definition of the cubic B-spline. Let  $\tau \in [0, 1)$  be an offset which is defined as above, and let further, without loss of generality,  $T = 1$ . We notice that the cubic B-spline may in general only evaluate to a value other than zero for the cases  $\beta(\tau - 2)$ ,  $\beta(\tau - 1)$ ,  $\beta(\tau)$ , and  $\beta(\tau + 1)$ . We further observe the following circumstances from (22):

$$\beta_1^3(\tau - 2) \neq 0 \Rightarrow -l - k = -2, \tag{23}$$

$$\beta_1^3(\tau - 1) \neq 0 \Rightarrow -l - k = -1, \tag{24}$$

$$\beta_1^3(\tau) \neq 0 \Rightarrow -l - k = 0, \tag{25}$$

$$\beta_1^3(\tau + 1) \neq 0 \Rightarrow -l - k = 1. \tag{26}$$

We solve the right-hand sides of (23) through (26) for  $k$  respectively so that we can conveniently refer to them during the course of the ensuing calculations:

$$-l - k = -2 \Leftrightarrow k = 2 - l, \tag{27}$$

$$-l - k = -1 \Leftrightarrow k = 1 - l, \tag{28}$$

$$-l - k = 0 \Leftrightarrow k = -l, \tag{29}$$

$$-l - k = 1 \Leftrightarrow k = -1 - l. \tag{30}$$

For the same reasons, we also tabulate:

$$\beta_1^3(\tau - 2) = \frac{1}{6}\tau^3, \tag{31}$$

$$\beta_1^3(\tau - 1) = -\frac{1}{2}\tau^3 + \frac{1}{2}\tau^2 + \frac{1}{2}\tau + \frac{1}{6}, \tag{32}$$

$$\beta_1^3(\tau) = \frac{1}{2}\tau^3 - \tau^2 + \frac{2}{3}, \tag{33}$$

$$\beta_1^3(\tau + 1) = -\frac{1}{6}\tau^3 + \frac{1}{2}\tau^2 - \frac{1}{2}\tau + \frac{1}{6}. \tag{34}$$

From the definition of the cubic B-spline, we also know that

$$\sum_{k \in \mathbb{Z}} \beta_1^3(k + \tau) \stackrel{!}{=} 1. \tag{35}$$

This knowledge is useful to fully comprehend the simplifications necessary to carry out the following derivation.

### 3.2 Derivation of the asymptotic error

By again letting  $\tau = \frac{x}{T} - \lfloor \frac{x}{T} \rfloor$ , we obtain the general coefficients  $a_n \forall n \in \mathbb{N}$  from (15) for the cubic B-spline interpolation filter as follows:

$$\begin{aligned} a_n &= \frac{T^n}{n!} \sum_{k \in \mathbb{Z}} (k - \tau)^n \sum_{l \in \mathbb{Z}} \frac{-6\alpha}{(1 - \alpha^2)} \alpha^{|l|} \beta_1^3(\tau - l - k) \\ &= \frac{T^n}{n!} \frac{-6\alpha}{(1 - \alpha^2)} \sum_{l \in \mathbb{Z}} \alpha^{|l|} \sum_{k \in \mathbb{Z}} (k - \tau)^n \beta_1^3(\tau - l - k) \end{aligned} \tag{36}$$

Since having identified the four cases for which the cubic B-spline may take on values other than zero, the sum over  $k \in \mathbb{Z}$  from (36) can be simplified as follows:

$$\begin{aligned} a_n &= \frac{T^n}{n!} \frac{-6\alpha}{(1 - \alpha^2)} \sum_{l \in \mathbb{Z}} \alpha^{|l|} \left[ \underbrace{(-1 - l - \tau)^n}_{Eq. 30} \underbrace{\left( -\frac{1}{6}\tau^3 + \frac{1}{2}\tau^2 - \frac{1}{2}\tau + \frac{1}{6} \right)}_{Eq. 34} \right. \\ &\quad + \underbrace{(-l - \tau)^n}_{Eq. 29} \underbrace{\left( \frac{1}{2}\tau^3 - \tau^2 + \frac{2}{3} \right)}_{Eq. 33} \\ &\quad + \underbrace{(1 - l - \tau)^n}_{Eq. 28} \underbrace{\left( -\frac{1}{2}\tau^3 + \frac{1}{2}\tau^2 + \frac{1}{2}\tau + \frac{1}{6} \right)}_{Eq. 32} \\ &\quad \left. + \underbrace{(2 - l - \tau)^n}_{Eq. 27} \underbrace{\left( \frac{1}{6}\tau^3 \right)}_{Eq. 31} \right] \end{aligned} \tag{37}$$

We now calculate the coefficients to find the first  $n$  so that  $a_n \neq 0$ . For  $n = 0$ , the whole sum over  $k \in \mathbb{Z}$  evaluates to one and we find that:

$$a_0 = \frac{-6\alpha}{(1 - \alpha^2)} \left[ \sum_{l \in \mathbb{Z}} \alpha^{|l|} \right] = \frac{-6\alpha}{(1 - \alpha^2)} \left[ \frac{2}{1 - \alpha} - 1 \right] = 1. \tag{38}$$



In the [Appendix](#), we proof that the terms in the square brackets are equivalent.

After simplifying the sum over  $k \in \mathbb{Z}$ , the Taylor series coefficient for  $n = 1$  is determined as follows:

$$a_1 = T \frac{-6\alpha}{(1 - \alpha^2)} \left[ - \sum_{l \in \mathbb{Z}} \alpha^{|l|} l \right] = T \frac{-6\alpha}{(1 - \alpha^2)} 0 = 0. \tag{39}$$

We show that the term in the square brackets evaluates to zero in the [Appendix](#).

For  $n = 2$ , calculating the coefficient yields

$$\begin{aligned} a_2 &= \frac{T^2}{2} \frac{-6\alpha}{(1 - \alpha^2)} \left[ \sum_{l \in \mathbb{Z}} \alpha^{|l|} \left( l^2 + \frac{1}{3} \right) \right] \\ &= \frac{T^2}{2} \frac{-6\alpha}{(1 - \alpha^2)} \left[ \sum_{l \in \mathbb{Z}} \alpha^{|l|} l^2 + \frac{1}{3} \sum_{l \in \mathbb{Z}} \alpha^{|l|} \right] \\ &= \frac{T^2}{2} \frac{-6\alpha}{(1 - \alpha^2)} \left[ - \frac{1}{3\sqrt{3}} + \frac{1}{3} \left[ \frac{2}{1 - \alpha} - 1 \right] \right] \\ &= 0. \end{aligned} \tag{40}$$

We once again delay the proof that  $\sum_{l \in \mathbb{Z}} \alpha^{|l|} l^2 = -\frac{1}{3\sqrt{3}}$  for  $\alpha = \sqrt{3} - 2$  to the [Appendix](#). The equality  $\sum_{l \in \mathbb{Z}} \alpha^{|l|} = \frac{2}{1-\alpha} - 1$  was already used for the derivation of  $a_0$  and can also be found in the [Appendix](#).

Since  $a_2$  also evaluates to zero for arbitrary  $\tau$ , we proceed to compute the coefficient for  $n = 3$ :

$$\begin{aligned} a_3 &= \frac{T^3}{6} \frac{-6\alpha}{(1 - \alpha^2)} \left[ \sum_{l \in \mathbb{Z}} \alpha^{|l|} (-l^3 - l) \right] \\ &= \frac{T^3}{6} \frac{-6\alpha}{(1 - \alpha^2)} \left[ - \sum_{l \in \mathbb{Z}} \alpha^{|l|} l^3 - \sum_{l \in \mathbb{Z}} \alpha^{|l|} l \right] \\ &= 0. \end{aligned} \tag{41}$$

Again the whole term in the square brackets evaluates to zero, so that the whole expression is zero. The equality  $\sum_{l \in \mathbb{Z}} \alpha^{|l|} l = 0$  was already used above, and  $\sum_{l \in \mathbb{Z}} \alpha^{|l|} l^3 = 0$  is shown in the [Appendix](#).

We find that the asymptotic error of the cubic B-spline interpolation filter is also not bounded by  $O(T^3)$  and proceed to the case  $n = 4$ :

$$\begin{aligned}
 a_4 &= \frac{T^4}{24} \frac{-6\alpha}{(1-\alpha^2)} \left[ \sum_{l \in \mathbb{Z}} \alpha^{|l|} \left( l^4 + 2l^2 - \tau^4 + 2\tau^3 - \tau^2 + \frac{1}{3} \right) \right] \\
 &= \frac{T^4}{24} \frac{-6\alpha}{(1-\alpha^2)} \left[ \sum_{l \in \mathbb{Z}} l^4 \alpha^{|l|} + 2 \sum_{l \in \mathbb{Z}} l^2 \alpha^{|l|} + \left( -\tau^4 + 2\tau^3 - \tau^2 + \frac{1}{3} \right) \sum_{l \in \mathbb{Z}} \alpha^{|l|} \right] \\
 &= \frac{T^4}{24} \frac{-6\alpha}{(1-\alpha^2)} \left[ \frac{1}{3\sqrt{3}} - 2 \frac{1}{3\sqrt{3}} + \left( -\tau^4 + 2\tau^3 - \tau^2 + \frac{1}{3} \right) \left[ \frac{2}{1-\alpha} - 1 \right] \right] \\
 &= \frac{T^4}{24} \frac{-6\alpha}{(1-\alpha^2)} \left[ -\frac{1}{3\sqrt{3}} \right] + \frac{T^4}{24} \frac{-6\alpha}{(1-\alpha^2)} \left[ \frac{2}{1-\alpha} - 1 \right] \\
 &\quad \left( -\tau^4 + 2\tau^3 - \tau^2 + \frac{1}{3} \right) \\
 &= \frac{T^4}{24} \left( -\frac{1}{3} \right) + \frac{T^4}{24} \left( -\tau^4 + 2\tau^3 - \tau^2 + \frac{1}{3} \right) \\
 &= \frac{T^4}{24} \left( -\tau^4 + 2\tau^3 - \tau^2 \right). \tag{42}
 \end{aligned}$$

We prove the fact that  $\sum_{l \in \mathbb{Z}} l^4 \alpha^{|l|} = \frac{1}{3\sqrt{3}}$  in the [Appendix](#). The various similar equations are already known from above and are, as already stated, also derived in the [Appendix](#). One can easily show that the fourth-order polynomial from (42) has only one real root in the interval  $[0, 1)$  for the special case where  $\tau = 0$ . We have thus shown that cubic B-spline interpolation in general is a 4-EF.

## 4 Conclusion

Cubic B-spline interpolation using prefiltering is a popular means to provide high-quality reconstruction, which is desirable e.g. when transforming 2-d or 3-d images. We studied the asymptotic error behavior of cubic B-spline interpolation using the method proposed by Möller et al., which is based on the Taylor series expansion of the convolution of the original data set with the reconstruction filter. The original method is however not suitable for filters with an infinite support. We showed how to adjust the method by Möller et al. to derive the asymptotic error of cubic B-spline interpolation using prefiltering and conclude that this reconstruction filter is a 4-EF.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

### Appendix

In order to derive the Taylor series coefficients for the cubic B-spline interpolation from (21), we show the following identities that we used to estimate the sum over  $l \in \mathbb{Z}$ .

For the following proofs we make use of the closed-form solution of the infinite geometric series for  $0 \leq a < 1$

$$\sum_{l=0}^{\infty} a^l = \frac{1}{1-a}. \tag{43}$$

We now want to show the identity from (38). Let in the following  $\alpha = \sqrt{3} - 2$ , i.e.  $0 \leq \alpha < 1$ . From the definition of the geometric series follows that

$$\sum_{l \in \mathbb{Z}} \alpha^{|l|} = 2 \left[ \frac{1}{1-\alpha} \right] - 1. \tag{44}$$

In order to comprehend (39), we further need to show that  $\sum_{l \in \mathbb{Z}} \alpha^{|l|} l = 0$ . Therefore we observe that

$$\sum_{l=0}^{\infty} \alpha^{|l|} l = \sum_{l=0}^{\infty} \alpha \frac{d}{d\alpha} \alpha^{|l|} = \alpha \frac{d}{d\alpha} \sum_{l=0}^{\infty} \alpha^{|l|} = \alpha \frac{d}{d\alpha} \frac{1}{1-\alpha} = \frac{\alpha}{(1-\alpha)^2} \tag{45}$$

and

$$\sum_{l=-\infty}^0 \alpha^{|l|} l = - \sum_{l=0}^{\infty} \alpha \frac{d}{d\alpha} \alpha^{|l|} = -\alpha \frac{d}{d\alpha} \sum_{l=0}^{\infty} \alpha^{|l|} = -\alpha \frac{d}{d\alpha} \frac{1}{1-\alpha} = -\frac{\alpha}{(1-\alpha)^2} \tag{46}$$

so that

$$\sum_{l \in \mathbb{Z}} \alpha^{|l|} l = \left[ \sum_{l=0}^{\infty} \alpha^{|l|} l \right] + \left[ \sum_{l=-\infty}^0 \alpha^{|l|} l \right] - \alpha^0 0 = 0. \tag{47}$$

In order to show (40), we need to proof that  $\sum_{l \in \mathbb{Z}} \alpha^{|l|} l^2 = -\frac{1}{3\sqrt{3}}$ . This can be seen by observing that

$$\sum_{l=0}^{\infty} \alpha^{|l|} l^2 = \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \sum_{l=0}^{\infty} \alpha^{|l|} = \alpha \frac{d}{d\alpha} \frac{\alpha}{(1-\alpha)^2} = \frac{\alpha(1+\alpha)}{(1-\alpha)^3}. \tag{48}$$

We also notice that from symmetry

$$\sum_{l=0}^{\infty} \alpha^{|l|} l^2 = \sum_{l=-\infty}^0 \alpha^{|l|} l^2 \tag{49}$$

and conclude that

$$\sum_{l \in \mathbb{Z}} \alpha^{|l|} l^2 = 2 \left[ \frac{\alpha(1+\alpha)}{(1-\alpha)^3} \right] - \alpha^0 0^2 = -\frac{1}{3\sqrt{3}}. \tag{50}$$

Equation 41 relies on the fact that  $\sum_{l \in \mathbb{Z}} \alpha^{|l|} l^3$  evaluates to zero. Therefore we show

$$\begin{aligned} \sum_{l=0}^{\infty} \alpha^{|l|} l^3 &= \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \alpha \sum_{l=0}^{\infty} \alpha^{|l|} = \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \frac{\alpha}{(1-\alpha)^2} \\ &= \alpha \frac{d}{d\alpha} \frac{\alpha(1+\alpha)}{(1-\alpha)^3} = \frac{\alpha(\alpha^2 + 4\alpha + 1)}{(1-\alpha)^4}. \end{aligned} \quad (51)$$

It can easily be seen from

$$\sum_{l=0}^0 \alpha^{|l|} l^3 = - \sum_{l=0}^{\infty} \alpha^{|l|} l^3 = - \frac{\alpha(\alpha^2 + 4\alpha + 1)}{(1-\alpha)^4} \quad (52)$$

that

$$\sum_{l \in \mathbb{Z}} \alpha^{|l|} l^3 = \left[ \frac{\alpha(\alpha^2 + 4\alpha + 1)}{(1-\alpha)^4} \right] + \left[ - \frac{\alpha(\alpha^2 + 4\alpha + 1)}{(1-\alpha)^4} \right] - \alpha^0 0^3 = 0. \quad (53)$$

The last identity that we need to show in order to proof (42) is  $\sum_{l \in \mathbb{Z}} \alpha^{|l|} l^4 = \frac{1}{3\sqrt{3}}$ . Similar to the previous cases we notice that

$$\begin{aligned} \sum_{l=0}^{\infty} \alpha^{|l|} l^4 &= \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \alpha \sum_{l=0}^{\infty} \alpha^{|l|} = \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \frac{\alpha}{(1-\alpha)^2} \\ &= \alpha \frac{d}{d\alpha} \alpha \frac{d}{d\alpha} \frac{\alpha(1+\alpha)}{(1-\alpha)^3} = \alpha \frac{d}{d\alpha} \frac{\alpha(\alpha^2 + 4\alpha + 1)}{(1-\alpha)^4} \\ &= \frac{\alpha(\alpha^3 + 11\alpha^2 + 11\alpha + 1)}{(1-\alpha)^5} \end{aligned} \quad (54)$$

From symmetry follows that

$$\sum_{l=0}^{\infty} \alpha^{|l|} l^4 = \sum_{l=-\infty}^0 \alpha^{|l|} l^4 \quad (55)$$

so that

$$\sum_{l \in \mathbb{Z}} \alpha^{|l|} l^4 = 2 \left[ \frac{\alpha(\alpha^3 + 11\alpha^2 + 11\alpha + 1)}{(1-\alpha)^5} \right] - \alpha^0 0^4 = \frac{1}{3\sqrt{3}}. \quad (56)$$

## References

1. Catmull, E., Rom, R.: A class of local interpolating splines. *Computer Aided Geometric Design*. Academic Press, pp 317–326 (1974)
2. Champagnat, F., Le Sant, Y.: Efficient cubic b-spline image interpolation on a GPU. *J. Graph. Tools* **16**(4), 218–232 (2012)
3. Etiene, T., Jonsson, D., Ropinski, T., Scheidegger, C., Comba, J., Nonato, L., Kirby, R., Ynnerman, A., Silva, C.: Verifying volume rendering using discretization error analysis. *IEEE Trans. Vis. Comput. Graph.* **20**(1), 140–154 (2013)
4. Keys, R.G.: Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics Speech Signal Process.* **29**(6), 1153–1160 (1981)

5. Lu, W., Vaswani, N.: Exact reconstruction conditions and error bounds for regularized Modified Basis Pursuit (Reg-modified-BP). In: 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), pp. 763–767. IEEE (2010)
6. Machiraju, R., Yagel, R.: Reconstruction error characterization and control: A sampling theory approach. *IEEE Trans. Vis. Comput. Graph* **2**(4), 364–378 (1996)
7. Marschner, S.R., Lobb, R.: An evaluation of reconstruction filters for volume rendering. In: Bergeron, R.D., Kaufman, A.E. (eds.) *IEEE Visualization*, pp. 100–107. IEEE Computer Society (1994)
8. Mitchell, D.P., Netravali, A.N.: Reconstruction filters in computer-graphics. *SIGGRAPH Comput. Graph* **22**(4), 221–228 (1988)
9. Möller, T., Machiraju, R., Mueller, K., Yagel, R.: Evaluation and design of filters using a Taylor series expansion. *IEEE Trans. Vis. Comput. Graph.* **3**(2), 184–199 (1997)
10. Möller, T., Mueller, K., Kurzion, Y., Machiraju, R., Yagel, R.: Design of accurate and smooth filters for function and derivative reconstruction. In: *Proceedings of the 1998 IEEE Symposium on Volume Visualization, VVS '98*, pp. 143–151. ACM, New York (1998)
11. Nehab, D., Maximo, A., Lima, R.S., Hoppe, H.: GPU-efficient recursive filtering and summed-area tables. *ACM Trans. Graph. (Proc. ACM SIGGRAPH Asia 2011)* **30**(6), 176 (2011)
12. Ruijters, D., ter Haar Romeny, B.M., Suetens, P.: Efficient gpu-based texture interpolation using uniform b-splines. *J. Graph. Tools* **13**(4), 61–69 (2008)
13. Ruijters, D., Thévenaz, P.: Gpu prefilter for accurate cubic b-spline interpolation. *Comput. J.* **55**(1), 15–20 (2012)
14. Sigg, C., Hadwiger, M.: Fast third-order texture filtering. In: Pharr, M. (ed.) *GPU Gems 2*, pp. 313–329. Addison-Wesley (2005)
15. Unser, M.: Splines: A perfect fit for signal and image processing. *IEEE Signal Process. Mag.* **16**(6), 22–38 (1999). IEEE Signal Processing Society's 2000 magazine award
16. Unser, M., Aldroubi, A., Eden, M.: Fast b-spline transforms for continuous image representation and interpolation. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(3), 277–285 (1991)
17. Unser, M., Aldroubi, A., Eden, M., Fellow, L.: B-spline signal processing: Part I-theory. *IEEE Trans. Signal Process.* **41**, 821–833 (1993)