Far East Journal of Electronics and Communications
Volume 10, Number 2, 2013, Pages 131-139
Published Online: July 2013
Available online at http://pphmj.com/journals/fjec.htm
Published by Pushpa Publishing House, Allahabad, INDIA

DESIGN OF TWO-POINT INTERPOLATION FILTERS BY MODIFYING LINEAR INTERPOLATION KERNEL

Hoon Yoo[+], Seok-Pil Lee and Beakcheol Jang[*]

Department of Digital Media
Sangmyung University
7 Hongji-dong, Jongno-gu
Seoul, 110-741, Korea

Abstract

This paper presents a design of two-point interpolation filters. The proposed design method provides novel interpolation kernels including trigonometric polynomials and it is considered to be a generalized version of the standard linear interpolation. Also, the proposed design method gives us an efficient hardware implementation by adding our modifying functions to the linear interpolation. Experimental results indicate that the proposed kernels outperform the existing kernels.

1. Introduction

Interpolation has been a very active research area since digital signals such as video, image, audio and modem signals require a technique for resampling or resizing. Because different applications need different interpolation methods, there have been many techniques such as finite fixed kernels [1-3, 9], spline kernels [4-5], adaptive kernels [6-8] and so on. In general, interpolation performance improves as the support of the kernels

Keywords and phrases: linear interpolation, two-point interpolation filter, low-complexity.
This research was supported by a 2013 Research Grant from Sangmyung University.

[+]First author: Hoon Yoo. [*]Corresponding author: bjang@smu.ac.kr

Received March 26, 2013

increases, whereas interpolation complexity also increases. Thus, a short kernel is sometimes required if the cost of interpolation is concerned [2, 10-12]. Among short kernel-based methods, the linear interpolation method is considered as a standard method, especially for hardware devices [12]. Thus, much less attention has been paid to developing a new one such as two-point filters. However, there is an ever-lasting need to improve interpolation performance. Moreover, a two-point filter is recently required for odd components of signals [10].

In this paper, a design of two-point interpolation kernels is proposed to keep interpolation complexity low and to enhance interpolation performance. We introduce a general form for two-point interpolation kernels utilizing the hat function that is used in the linear interpolation. Also, we introduce a parametric trigonometric polynomial to enhance the performance of the hat function. We then apply several useful conditions to the parametric kernel to determine the parameter. In addition, it turns out that our method provides an efficient implementation by modifying the linear interpolation method; thus the proposed interpolation method can be easily implemented by adding our modifying function to the linear interpolation. This means that our method is a generalized version of the standard linear interpolation.

2. Proposed Design for Interpolation Filters

2.1. Formulation

For a given set of discrete samples $f(x_k)$, an interpolation process produces a continuous signal $f(x)$ which is defined by

$$f(x) = \sum_k f(x_k)\beta(x - x_k), \tag{1}$$

where $\beta(x)$ is the interpolation kernel. This interpolation process is called by the convolution-based interpolation due to the nature of formula (1). To be an interpolator, the kernel $\beta(x)$ should satisfy the zero-crossing condition, which means that $\beta(0) = 1$ and $\beta(x) = 0$ for any nonzero integer x. Note that the computational cost increases as the support of an interpolation kernel increases, especially in hardware implementations. Thus, we consider a design of an interpolation kernel that is as short as possible. Consequently, the length of the kernel is set to be two. Then, we can define a kernel of length 2 as

$$\beta(x) = \begin{cases} \beta_R(x), & 0 \le x \le 1, \\ \beta_L(x), & -1 \le x \le 0. \end{cases} \tag{2}$$

From the zero-crossing condition, the kernel should hold that $\beta_R(1) = \beta_L(-1) = 0$ and $\beta_R(0) = \beta_L(0) = 1.$ Also, it is preferable to apply the *y*-axis $(x = 0)$ symmetric condition to the kernel, which means that $\beta_L(x) = \beta_R(-x).$ In addition, it is preferable to apply the partition of unity condition (DC-constancy)

$$\beta_R(x) + \beta_L(x-1) = 1. \tag{3}$$

Note that this condition guarantees that the energy of the resampled signal remains unchanged [3]. If the *y*-axis symmetric condition and the partition of unity condition are applied to (3), then this yields

$$\beta_R(x) + \beta_R(1-x) = 1. \tag{4}$$

This states that the kernel $\beta_R(x)$ is point-symmetric with respect to the point $(1/2,\ 1/2).$ And then this is helpful to choose a function for the kernel $\beta_R(x).$

The choice of the kernel $\beta_R(x)$ is considered to be the design of the interpolation process. From the point-symmetric condition, we can easily define functions passing through the point $(1/2,\ 1/2).$ The simplest function is the line function coming from $(0,\ 1),$ passing through $(1/2,\ 1/2),$ and going to $(1,\ 0).$ The kernel $\beta_R(x)$ is simply defined by $\beta_R(x) = 1 - x$ and it is utilized in the linear interpolation method. Note that many interpolation applications employ the linear interpolation kernel for its simplicity and moderate quality, especially in hardware systems. It is natural that the design of interpolation kernels is based on the linear interpolation kernel since the linear interpolation is very common and easy to employ it in the literature. For this reason, we define the kernel $\beta_R(x)$ as

$$\beta_R(x) = 1 - x + wM(x), \tag{5}$$

where the function $M(x)$ is a modifying function and the value w is a weighting factor. Consequently, the design of the interpolation kernel $\beta_R(x)$ is converted to the choice of a modifying function $M(x)$ and a weighting factor w.

If we apply the zero-crossing and point-symmetric conditions to (5), then the modifying function $M(x)$ should pass through three points $(0,\ 0)$, $(1/2,\ 0)$ and $(1,\ 0)$ and it should be point-symmetric with respect to the point $(1/2,\ 0)$. Based on these properties, we introduce four modifying functions that are $\sin(2\pi x)$, $x - 1/2(1 - \cos(\pi x))$, $x(1-2x)(1-x)$ and $quad(x)$, where $quad(x) = x(1-2x)$ for x in $[0,\ 1/2]$ or $quad(x) = (1-2x)(1-x)$ for x in $[1/2,\ 1]$. Although there may be lots of modifying functions, discussions with the four introduced modifying functions are enough to understand the proposed design.

2.2. Interpolation formula

In the previous section, we discuss that how to choose a modification function for an interpolation kernel. This section describes that how to produce a continuous function using the functions that are proposed in the previous section. When the condition that the length of an interpolation kernel is two is applied to (1), we have

$$f(x) = f(x_k)\beta_R(x - x_k) + f(x_{k+1})\beta_L(x - x_{k+1}), \text{ for } x_k \le x \le x_{k+1}. \tag{6}$$

Here, let us assume that samples are uniformly spaced. Then, the distance between x_{k+1} and x_k can be set to be one without loss of generality. Also, we apply a distance parameter $s = x - x_k$ to (6). This gives us

$$f(x) = f(x_k)\beta_R(s) + f(x_{k+1})\beta_L(s-1), \quad 0 \le s \le 1. \tag{7}$$

Substituting (5) to (7), we have

$$f(x) = f(x_k)(1 - s + wM(s)) + f(x_{k+1})(s + wM(1 - s)). \tag{8}$$

Note that the function $M(s)$ is point-symmetric with respect to the point $(1/2,\ 0)$ and this yields $M(1 - s) + M(s) = 0$ or $M(1 - s) = -M(s)$. From this, (8) is represented in the form

$$f(x) = f(x_k)(1 - s_M) + f(x_{k+1})(s_M), \tag{9}$$

where $s_M = s - wM(s)$. It is easily seen that (9) is exactly the same as the linear interpolation formula except modifying the distance parameter *s*.

2.3. Determination of weighting factor *w*

After a modifying function is chosen, the weighting factor *w* should be determined. The factor *w* is considered to be a parameter to control the strength of modification. To determine the parameter *w*, we impose C^1-continuity on the kernel in this paper. The continuity constraint enables the kernel smooth and it may give us a substantial gain in image quality. Now, we apply the continuity constraint to the kernel. Then we have an equation with respect to the parameter *w* in the form

$$\frac{\partial}{\partial x}\beta_R(x,\ w)\bigg|_{x=0} = \frac{\partial}{\partial x}\beta_L(x,\ w)\bigg|_{x=0} = \frac{\partial}{\partial x}\beta_R(-x,\ w)\bigg|_{x=0}. \tag{10}$$

It states that the kernel $\beta(x)$ is C^1-continuous at the point $x = 0$. Substituting (5) into (10) and rearranging with respect to *w*, we have

$$w = \frac{1}{M'(0)}. \tag{11}$$

It can be easily seen that the parameter *w* is related to the modifying function that we choose and it is simply calculated by (11). Table 1 shows a list of the weighting factor *w* for the modifying functions discussed in this paper.

3. Discussion and Experimental Results

The proposed design is intended for a two-point interpolation filter. However, there are existing designs of interpolating filters of length N and they can produce interpolating filters of length two as their subset. Thus, we discuss two different designs with the proposed design. One is the design of the CCI method [1] based on the cubic polynomial; the other is the design of the trigonometric polynomial interpolation (TPI) method [9] based on the Fourier analysis.

CCI is intended for an interpolating filter of length four though it can provide an interpolating filter of length six. Also, CCI has a parameter a as a design factor and the parameter a is set to be $-1/2$ if C^3-continuity is applied to the kernel. In case that the parameter a is zero, the support of the CCI kernel is limited and thus the CCI is considered to be an interpolating filter of length two. This CCI kernel is defined as $\beta_{CCI}(x) = 2|x|^3 - 3|x| + 1,\ |x| \leq 1.$ Interestingly, this kernel can be derived from the proposed design if $M(x) = x(1-2x)(1-x)$ and $w = 1.$

Table 1. Modifying functions and factors for some kernels

	CCI	TPI	quad	Sin
$M(x)$	$x(1-2x)(1-x)$	$x - 1/2\,(1-\cos(\pi x))$	$quad(x)$	$\sin(2\pi x)$
w	1	1	1	$1/2\pi$
Note	[1]	[9]	Our Method 1	Our Method 2

Also, TPI is for N-point interpolating filters. Basically, this design is based on the linear combination of trigonometric polynomials such as cosine functions. Thus, we can have a two-point interpolation filter if we choose just a cosine function in the design of TPI. The kernel of two-point TPI is defined as $\beta_{CCI}(x) = (1+\cos(|\pi x|))/2,\ |x| \leq 1.$ Definitely, the proposed method can produce the two-point TPI filter if we set $M(x) = x - 1/2\,(1-\cos(\pi x))$ and $w = 1.$

The two designs, CCI and TPI are quite different but they can be unified by our design. In other words, the existing designs for two-point interpolating filters are simply converted by choosing modifying functions and factors in our design. Also, our design has much higher flexibility to create two-point interpolating filters because we can choose various modifying functions and their modifying factor. For examples, we introduced the new modifying functions such as $\sin(\ )$, $quad(\ )$ that are listed in Table 1.

Another important property of the proposed design is that the kernels from our design are accomplished by the linear interpolation with simple modification, referring (9), this is possible and Figure 1 illustrates a diagram that includes the existing linear interpolation block and the added block. Based on (9), the added block is to modify the parameter *s*. This property is very helpful to redesign the linear interpolating filter in hardware as well as software.

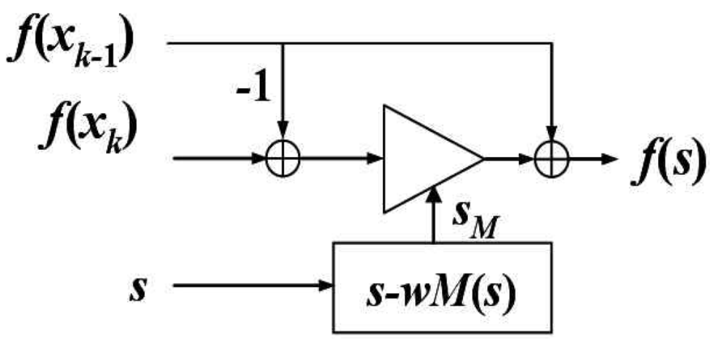

Figure 1. Efficient hardware structure of proposed design.

To evaluate our interpolation methods, we conduct an image scaling experiment that includes up-scaling and followed by down-scaling [7]. That is, we up-scale test images by a factor of root(2) and then down-scale the up-scaled images by a factor of 1/root(2). This experiment enables us to compare the original images and the resulting images by measuring PSNR. The PSNR results are shown in Table 2. We can see that the proposed kernels outperform the existing kernels such as the kernels of the linear interpolation, CCI and TPI.

Table 2. PSNR results from zooming experiment (dB)

Images	Linear	CCI	TPI	Quad	Sin
Lena	40.92	43.96	44.32	45.35	46.84
Baboon	30.25	33.23	33.60	34.63	36.14
Peppers	39.00	41.82	42.18	43.15	44.66
Airplane	38.46	41.22	41.57	42.60	43.97
Finger	36.14	39.23	39.61	40.66	42.15
Average	36.95	39.90	40.26	41.28	42.75

4. Conclusion

We have presented a design for two-point interpolation filters. The proposed design is considered to be a generic version of the standard linear interpolation. Also, our design provides very efficient structure in hardware implementation. Experimental results have shown that the proposed method outperforms the existing methods such as the linear, CCI and TPI methods.

References

[1] R. G. Keys, Cubic convolution interpolation for digital image processing, IEEE Trans. Acoust., Speech, Signal Process. 29 (1981), 1153-1160.

[2] E. Maeland, On the comparison of interpolation methods, IEEE Trans. Medical Imaging 7 (1988), 213-217.

[3] T. M. Lehmann, C. Gönner and K. Spitzer, Survey: interpolation methods in medical image processing, IEEE Trans. Medical Imaging 18 (1999), 1049-1075.

[4] M. Unser, Splines: a perfect fit for signal and image processing, IEEE Signal Process. Mag. 16 (1999), 22-38.

[5] T. Blu, P. Thevenaz and M. Unser, Linear interpolation revitalized, IEEE Trans. Image Process. 13(5) (2004), 710-719.

[6] X. Li and M. Orchard, New edge-directed interpolation, IEEE Trans. Image Process. 10(10) (2001), 1521-1527.

[7] J. K. Han and S. U. Baek, Parametric cubic convolution scalar for enlargement and reduction of image, IEEE Trans. Consumer Electron. 46(2) (2000), 247-256.

[8] H. Yoo, Closed-form least-squares technique for adaptive linear image interpolation, IET Electron. Lett. 43 (2007), 210-212.

[9] D. Fu and A. N. Wilson Jr., Trigonometric polynomial interpolation for timing recovery, IEEE Trans. Circuits and Systems-I: Regular Papers 52 (2005), 338-349.

[10] B.-D. Choi and H. Yoo, Design of piecewise weighted linear interpolation based on even-odd decomposition and its application to image resizing, IEEE Trans. Consumer Electron. 55(4) (2009), 2280-2286.

[11] C. Hughes, P. Denny, M. Glavin and E. Jones, Equidistant fish-eye calibration and rectification by vanishing point extraction, IEEE Trans. Patt. Anal. Mach. Intel. 32(12) (2010), 2289-2296.

[12] M. Tomasi, M. Vanegas, F. Barranco, J. Diaz and E. Ros, Real-time architecture for a robust multi-scale stereo engine on FPGA, IEEE Trans. VLSI Syst. 20(12) (2012), 2208-2219.