IEIE Transactions on Smart Processing and Computing

Real-Time Continuous-Scale Image Interpolation with Directional Smoothing

Yoonjong Yoo[1], Jeongho Shin[2], and Joonki Paik[1]

[1] Image Processing and Intelligent Systems Laboratory, Department of Advanced Imaging, Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang University / Seoul, Korea {whitener, paikj}@cau.ac.kr
[2] Department of Web information Engineering Hankyong University / Gyeonggi, Korea shinj@hknu.ac.kr

***** Corresponding Author: Joonki Paik

Received November 22, 2013; Revised January 5, 2014, Accepted March 20, 2014; Published June 30, 2014

** Regular Paper*

***Abstract*: A real-time, continuous-scale image interpolation method is proposed based on a bi-linear interpolation with directionally adaptive low-pass filtering. The proposed algorithm was optimized for hardware implementation. The ordinary bi-linear interpolation method has blocking artifacts. The proposed algorithm solves this problem using directionally adaptive low-pass filtering. The algorithm can also solve the severe blurring problem by selectively choosing low-pass filter coefficients. Therefore, the proposed interpolation algorithm can realize a high-quality image scaler for a range of imaging systems, such as digital cameras, CCTV and digital flat panel displays.

***Keywords*: Interpolation, Smoothing, Scaler

1. Introduction

Image interpolation plays a fundamental role in image processing systems. For this reason, image interpolation has a range of application areas, such as surveillance, image acquisition and medical imaging.

Interpolation is commonly implemented by convolving an image with a small kernel using a proper weighting function. Popular convolution-based methods of interpolation include the nearest neighbor interpolation, bi-linear interpolation, cubic B-spline, and regularized interpolation. Although nearest neighbor and bi-linear interpolation methods are simple and fast, they have blocking artifacts. On the other hand, the cubic B-spline algorithm results in blurring artifacts, at the cost of reduced blocking artifacts. Regularized interpolation is unsuitable for real-time processing because of its iterative procedure.

This paper proposes a simple, bi-linear interpolation-based method using adaptive low-pass filters in different directions for low-computation. The interpolated image of the proposed algorithm is more natural by smoothing the over- amplified edge and blocking artifacts.

In the proposed interpolation method, the spatial gradients are used to extract the edge information of each local area in an image. Five different edge types were identified by utilizing the spatial gradients correctly. Each edge type determines which interpolation techniques should be applied to the corresponding local region. Given the edge directions, the interpolation procedure begins with the proposed adaptive low-pass filter. For practical implementation, an efficient hardware structure of the proposed system is also described based on each functional module.

This paper is organized as follows. Section 2 introduces some existing interpolation methods. The proposed algorithm is presented in section 3. Section 4 summarizes experimental results, and section 5 concludes the paper.

2. Mathematical Background for Image Interpolation

Image interpolation generally consists of two steps: (i) the creation of a new pixel location, and (ii) the assignment of a pixel value to those new locations. This section summarizes a number of interpolation algorithms.

2.1 Convolution-based Interpolation

The nearest neighbor interpolation is also called the zero-order hold method. The interpolation is performed by convolving an $M \times N$ input image with a proper kernel, which is denoted by H. For interpolation by a factor of two, for example, the corresponding kernel can be expressed as follows:

$$H = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{1}$$

A bi-linear interpolation is also called the first-order method, which calculates the intensity by convolving the input image with the kernel given as follows [1]:

$$H = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}. \tag{2}$$

A comparison with the nearest neighbor interpolation showed that the bi-linear interpolation is equivalent to convolution with the kernel in (1) twice, one for the horizontal and the other one for the vertical direction.

Both the nearest neighbor and bi-linear interpolations are used widely in low-cost, compact imaging systems because of their simple structures. As a result of the information theory introduced by Shannon in the late 1940's, the sinc function is the ideal kernel of the interpolation. On the other hand, it has an infinite impulse response (IIR) and is unsuitable for the finite impulse response (FIR) implementation.

As a compromise between the simple convolution-based methods and the ideal interpolator, the cubic B-spline is an efficient alternative. In a cubic B-spline interpolation, the zero-order filter interpolation kernel is convolved three times, which makes the resulting interpolation closer to the ideal interpolation [2-4]. The B-splines are symmetrical, bell-shaped functions constructed from the (n+1)-fold convolution of a zero-order filter pulse β^0 as follows [5-8]:

$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & otherwise \end{cases},$$

and

$$\beta^n(x) = \beta^0 * \beta^0 * \cdots \beta^0(x). \tag{3}$$

The nearest neighbor (n = 0), bi-linear (n = 1), and cubic spline (n = 3) interpolation have a common basis function called the B-Spline. The resulting image by a cubic B-spline interpolation becomes smoother than those of the above mentioned methods. On the other hand, the cubic interpolation tends to produce over-amplified edges [5].

3. Proposed Algorithm

The proposed algorithm uses directionally adaptive low-pass filtering to solve this problem. The basic computational process uses a bi-linear interpolation, and a user can control the coefficients of a low-pass filter to smooth the blocking artifacts. In particular, the proposed algorithm can adaptively select either a sharpness or smoothness filter. Fig. 1 shows the procedure of the proposed algorithm.

In Fig. 1, the filled large and dotted line circles represent pixels of low resolution and high resolution images, respectively. The edge detection procedure obtains the direction in 3×3 LR image pixels. In addition, the creation of new pixel locations procedure produces pixel locations of the HR image. Finally, a directionally adaptive low-pass filtering procedure processes the filtering of interpolated pixels.

Fig. 2 presents a block diagram of the proposed interpolation algorithm. Although most parts of the image are interpolated using the bi-linear method, the neighbors of the directional edges are interpolated by directional filtering to reduce the number of blocking artifacts in a high-resolution image. The detailed step-by-step process will be explained in the following subsections.

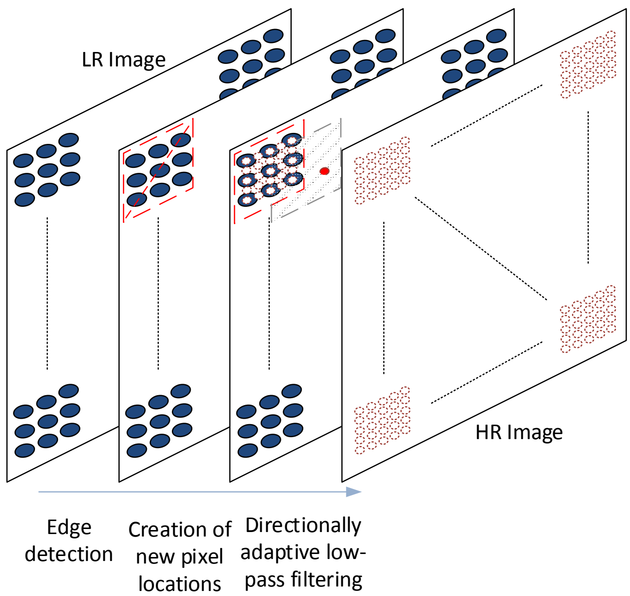


Fig. 1. Illustration of the proposed interpolation system.

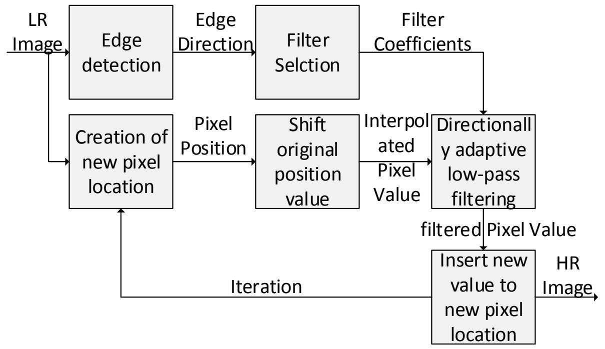


Fig. 2. Block diagram of the proposed algorithm.

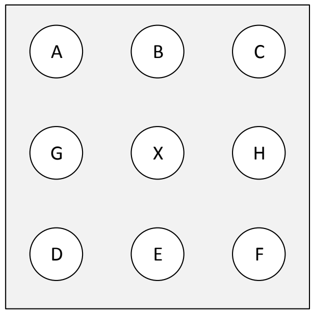


Fig. 3. 3×3 **kernel to detect the edge direction.**

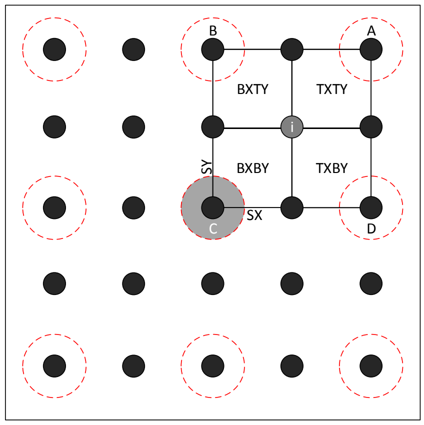


Fig. 4. Indication chart of the HR new pixel and LR pixel position.

3.1 Edge Detection

The algorithm takes a LR Image as the input, and decides the representative direction of the current pixel. Two methods can be used to select the representative direction by either the spatial gradient or directional frequency analysis. The frequency-domain analysis method is computationally efficient at the expense of accuracy [9]. On the other hand, the spatial-domain method can detect a variety of directions with a high computational load. In addition, other directional low-pass filters, such as a bi-lateral and epsilon filter, has a heavy computational load and memory requirement. Therefore, the proposed algorithm uses five simple Gaussian filters to reduce the computational load.

The proposed algorithm reduces the computational load using the following process. The spatial gradients were calculated using a 3×3 kernel to detect four different directions. A maximum of four different gradients were then selected for the representative direction. In addition, if the four different gradients are lower than a pre-specified threshold value, it was decided that there would be no directional component in the neighborhood of the pixel.

In Fig. 3, X represents the location of the current pixel on which the direction of the edge is to be detected. The gradient value of each direction is calculated as follows:

$$\begin{aligned} DI_{horizontal} &= \left|(A+B+C)/3+(D+E+F)/3\right| \\ DI_{vertical} &= \left|(A+D+G)/3+(C+F+H)/3\right| \\ DI_{45} &= \left|(B+C+H)/3+(D+E+G)/3\right|, \\ DI_{135} &= \left|(A+B+G)/3+(E+F+H)/3\right| \\ DI_{flat} &= Threshold \end{aligned} \tag{4}$$

where DI represents the gradient values, and the horizontal, vertical 45, and 135 means each direction. The threshold is the arbitrary number by user selection. Therefore, the flat direction means a non-edge area. The threshold is a pre-specified constant for a flat region. The representative direction of the current pixel is selected as follows:

$$DI = \max(DI_{horizontal}, DI_{vertical}, DI_{45}, DI_{135}, DI_{flat})\,. \tag{5}$$

3.2 Interpolation and Low-pass Filtering

Given the ratio of the interpolation, both interpolation and directionally adaptive filtering were performed about the representative direction in the edge detection process. If the two processes are individually separated, the interpolated image needs to be memorized before filtering. This is the problem of implementation in hardware because of the additional memory.

Fig. 4 shows the case of the interpolation procedure of ratio 5/3. The dotted line circle represents pixels of the input low-resolution image, and the filled circles show the regions that require filtering. SX and SY are the horizontal and vertical scale factors of interpolation, respectively, which are defined as

$$\begin{aligned} \mathrm{SX} &= \frac{(\text{ori\_width-1})}{(\text{inter\_width-1})} \\ \mathrm{SY} &= \frac{(\text{ori\_height-1})}{(\text{inter\_height-1})} \end{aligned}, \tag{6}$$

where ori_width, inter_width, ori_height and inter_height respectively represent the widths and heights of the low-resolution and high-resolution images. Both SX and SY are equal to 1/2 in Fig. 4. The pixel to be interpolated is 1/2 pixel apart from (0, 0). TXTY, TXBY, BXTY, and BXBY are the distances between the interpolated and original pixels of the low-resolution image, which are defined as

$$\begin{aligned} TXTY &= (1-SX)\times(1-SY) \\ TXBY &= (1-SX)\times SY \\ BXTY &= SX\times(1-SY) \\ BXBY &= SX\times SY. \end{aligned} \tag{7}$$

The proposed algorithm uses five 5×5 low-pass filters, which have different directional components. The proposed algorithm uses the modified interpolation filter based on a bi-linear filter. The weighting values are defined by the area as follows:

$$i=\left(A\times BXBY\right)+\left(B\times TXBY\right)+\left(C\times TXTY\right)+\left(D\times BXTY\right), \tag{8}$$

where A, B, C, and D represent the original pixels of the low-resolution image in Fig. 4. The interpolated pixel i was influenced by the closest pixel value. For example, if the position (SX, SY) is (0.2, 0.2), the calculated BXBY, TXBY, TXTY, and BXTY are 0.04, 0.16, 0.16, and 0.64, respectively. In this case, the closest pixel is C, and it was multiplied by the largest value, 0.64.

The interpolated pixel was determined by the directional low-pass filtering along the representative direction. The pixels on the same directional edge were smoothed by the corresponding low-pass filter, whereas the pixels in the normal direction were less smoothed by a factor of 2/3. Using this method, the edge sharpness can be preserved during the interpolation process. Fig. 5 shows five directional low-pass filter masks.

The transfer function h(x) in the restoration filter uses the general Gaussian equation in the spatial domain as follows [9]:

$$h(x)=e^{-X^2/2\sigma^2}, \tag{9}$$

where X and σ^2 represent distance from the center of filter and variance, respectively. The value of σ was changed by the edge direction and the interpolation ratio. The empirically chosen σ is given as

$$\sigma = 0.013182\times(\text{interpolation ratio})\times10+(\text{constant}), \tag{10}$$

where interpolation ratio represents the ratio between the size of the HR and LR images. In addition, the constant represents the parameter controlling the smoothness. 0.013182 is the value that is considered a characteristic of the cut-off frequency response of the Gaussian filter. The

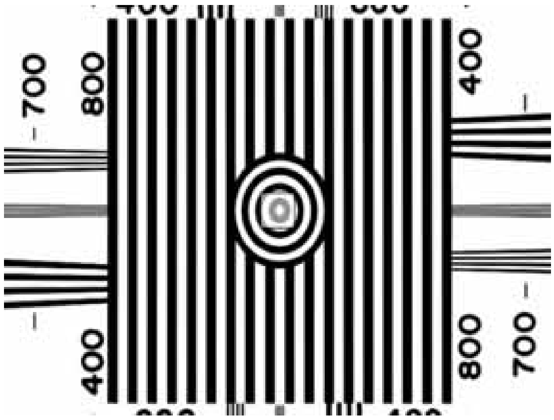
(a) (b) (c) (d) (e)

Fig. 5. Low-pass filter (a) 45°, (b) 135°, (c) 180°, (d) 90°, and (e) flat filter.

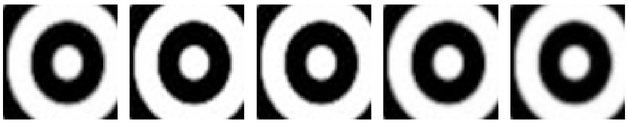
(a)

(b) (c) (d) (e) (f)

Fig. 6. (a) High resolution Lena image, (b) bi-linear interpolation image, (c) cubic sharp interpolation image, (d) cubic smooth interpolation image, (e) proposed sharp filter image, (f) proposed smooth filter image.



(a)



(b) (c) (d) (e) (f)

Fig. 7. (a) High resolution 1956 EIA resolution chart image, (b) bi-linear interpolation image, (c) cubic sharp interpolation image, (d) cubic smooth interpolation image, (e) proposed sharp filter image, and (f) proposed smooth filter image.

interpolated image can be obtained by filtering the 5×5 regions with this filter.

4. Experimental Results

This section evaluates the performance of the proposed interpolation method and compares it with several existing interpolation methods. In this experimental, the proposed algorithm uses both sharp and smooth filters. The low-scale image was made by only sub-sampling without low-pass filtering.

Fig. 6(a) shows the interpolated 512×512 image from the 320×240 low-scale version.

Figs. 6(b)-(f) show magnified versions of the square region in Fig. 6(a). Fig. 6(b) shows the result using the bi-linear interpolation. Figs. 6(c) and (d) show the results using a bicubic interpolation with and without a sharpening filter, respectively, in Adobe photoshop. Figs. 6(e) and (f) show the results using the proposed algorithm with sharpening and smoothing filters, respectively. As shown in Fig. 6, the proposed algorithm reduces the blocking artifacts compared to other methods.

Fig. 7(a) shows the interpolated 1024×768 image from the 720×480 low-scale version of the 1956 EIA resolution Chart.

Figs. 7(b)-(f) show magnified versions of the square region of Fig. 7(a). Fig. 7(b) shows the result using a bi-linear interpolation. Figs. 7(c) and (d) show the results from using the bicubic interpolation with and without sharpening filter, respectively. Figs. 7(e) and (f) show the results using the proposed method using sharp and smooth filters, respectively. In this experiment, the horizontal and vertical interpolation ratios were not the same, which results in severe blocking artifacts when existing methods are used. On the other hand, the proposed algorithm reduces the blocking artifacts significantly.

Table 1. PSNR values of five interpolation methods with interpolation ratio 2.0.

Interpolation Method	Interpolation Ratio	Red channel [dB]	Green channel [dB]	Blue channel [dB]
Bi-linear	2.0	30.867	28.259	28.804
Cubic-sharp	2.0	30.985	28.463	28.698
Cubic-smooth	2.0	31.182	28.624	28.854
Proposed-Sharp	2.0	30.877	28.280	28.844
Proposed-Smooth	2.0	30.700	28.213	28.887

Table 2. PSNR values of the five interpolation methods with an interpolation ratio of 4.0.

Interpolation Method	Interpolation Ratio	Red channel [dB]	Green channel [dB]	Blue channel [dB]
Bi-linear	4.0	27.543	25.134	26.436
Cubic-sharp	4.0	28.360	25.694	26.449
Cubic-smooth	4.0	28.689	26.039	26.902
Proposed-Sharp	4.0	27.551	25.146	26.453
Proposed-Smooth	4.0	27.562	25.188	26.530

Table 1 lists the PSNR values using the five different interpolation methods.

For this experiment, the 512×512 input image was down scaled by a factor of two, and interpolated back to the input image size. Table 2 lists the PSNR values with an interpolation ratio of 4.0.

As shown in both tables, the PSNR variation was small.

Therefore, the subjective criterion is more important than the PSNR comparison.

5. Conclusion

The proposed method enables a real-time, continuous-scale interpolation with less blocking artifacts of existing interpolation methods for a standard definition image scaler in the digital set-top box. For computational efficiency, the proposed algorithm is based on the bi-linear structure with adaptive filtering along the local edge direction. The proposed algorithm was developed by a simple filtering and process pipeline without a frame memory, heavy computational load and iterative processing. The quality of the interpolated image is improved further with a user selecting filtering process. As shown in the experimental results, the proposed algorithm efficiently reduces the blocking artifacts and provides user-friendly control of the sharpness using multiple filter sets.

Acknowledgement

This study was supported in part by the Basic Science Research Program through National Research Foundation (NRF) of Korea funded by the Ministry of Education, Science and Technology (2013R1A1A2061847), and Seoul R&BD Program (PA130170). .

References

[1] T. Blu, P. Thevenaz, and M. Unser, "Linear interpolation revitalized," *IEEE Trans. Image Processing*, vol. 13, no. 5, pp. 710-719, May 2004. Article (CrossRef Link)

[2] T. Lehmann, C. Gonner, and K. Spitzer, "Survey: interpolation methods in medical image processing," *IEEE Trans. Medical Imaging*, vol. 18, no. 11, pp. 1049-1075, November 1999. Article (CrossRef Link)

[3] R. Schafer, and L. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, no. 6, pp. 692-702, June 1973. Article (CrossRef Link)

[4] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proc. IEEE*, vol. 90, no. 3, pp. 319-342, March 2002. Article (CrossRef Link)

[5] M. Unser, "Splines: a perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22-38, November 1999. Article (CrossRef Link)

[6] S. Jiazheng and S. Reichenbach, "Image interpolation by two-dimensional parametric cubic convolution," *IEEE Trans. Image Processing*, vol. 15, no. 7, pp. 1857-1870, July 2006. Article (CrossRef Link)

[7] H. Hsieh and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Signal Processing*, vol. 26, no. 6, pp. 508-571, December 1978. Article (CrossRef Link)

[8] E. Maeland, "On the comparison of interpolation methods," *IEEE Trans. Medical Imaging*, vol. 7, no. 3, pp. 213-217, September 1988. Article (CrossRef Link)

[9] K. Hong, J. Paik, H. Kim, and C. Lee, "An edge preserving image interpolation system for a digital camcorder," *IEEE Trans. Consumer Electronics*, vol. 41, no. 3, pp. 279-284, August 1996. Article (CrossRef Link)

Yoonjong Yoo received his B.S. degree in electronic engineering from Chung-Ang University, Seoul, Korea, in 2005. He received his M.S. degree in image engineering from Chung-Ang University, Seoul, Korea, in 2007. From 2009 to 2013, he joined Nextchip, where he designed the auto exposure, auto white balance, and wide dynamic range for a surveillance camera. Currently, he is pursuing a Ph.D. degree in image processing at Chung-Ang University. His research interests include image enhancement and restoration for display processing, video compression standards and surveillance video applications.

Jeongho Shin received his B.S. and the M.S. degrees in electronic engineering from Chung-Ang University, Seoul, Korea, in 1994 and 1998, respectively, and his Ph.D. degree in image engineering from Chung-Ang University, Seoul, Korea, in 2001. From 2003 to 2006, he was a research professor at the department of image engineering, Chung-Ang University, Seoul, Korea. Currently, he is an assistant professor at the department of web information engineering, Hankyong National University, Gyeonggi, Korea. His current research interests include an enhancement and restoration of image and video, object tracking, and data fusion.

Joonki Paik received his B.S. degree in control and instrumentation engineering from Seoul National University in 1984. He received his M.S. and the Ph.D. degrees in electrical engineering and computer science from Northwestern University in 1987 and 1990, respectively. From 1990 to 1993, he joined Samsung Electronics, where he designed the image stabilization chip sets for consumer's camcorders. In 1993, he joined the faculty at Chung-Ang University, Seoul, Korea, where he is currently a Professor in the Graduate school of Advanced Imaging Science, Multimedia and Film. From 1999 to 2002, he was a visiting Professor at the Department of Electrical and Computer Engineering at the University of Tennessee, Knoxville. Dr. Paik was a recipient of the Chester-Sall Award from IEEE Consumer Electronics Society, Academic Award from the Institute of Electronic Engineers of Korea, and Best Research Professor Award from Chung-Ang University. He served the Consumer Electronics Society of IEEE as a member of the Editorial Board. Since 2005, he has been head of the National Research Laboratory in the field of image processing and intelligent systems. In 2008, he worked as a full-time technical consultant for the System LSI Division in Samsung Electronics, where he developed various computational photographic techniques including an extended depth of field (EDoF) system. From 2005 to 2007, he served as Dean of the Graduate School of Advanced Imaging Science, Multimedia, and Film. From 2005 to 2007, he was Director of Seoul Future Contents Convergence (SFCC) Cluster established by Seoul Research and Business Development (R&BD) Program. Dr. Paik is currently serving as a member of the Presidential Advisory Board for Scientific/Technical policy of the Korean Government and is a technical consultant of the Korean Supreme Prosecutor's Office for computational forensics.