Journal of Signal Processing, Vol.17, No.4, pp.111-114, July 2013

SELECTED PAPER

FPGA Implementation of Single-Image Super-Resolution Based on Frame-Bufferless Box Filtering

Yuki Sanada, Takanori Ohira, Satoshi Chikuda, Masaki Igarashi, Masayuki Ikebe, Tetsuya Asai and Masato Motomura

Graduate School of Information Science and Technology, Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan
Phone: +81-11-706-6080, FAX: +81-11-706-7890, E-mail: asai@ist.hokudai.ac.jp

Abstract

Recently, a novel algorithm of filter-based single-image super-resolution (SR) has been proposed. We here propose a hardware-oriented image-enlargement algorithm for the SR algorithm based on frame-bufferless box filtering, and present novel circuits of the proposed enlargement algorithm and the SR algorithm for an field-programmable gate array (FPGA), aiming at the development of single-image SR module for practical embedded systems.

1. Introduction

Super high-resolution displays, such as retina displays and 4K/8K ultrahigh-definition televisions (UHDTV), have been spotlighted in digital home appliances [1]. Super-resolution (SR) techniques, which increase the resolution of images, are thus necessary for transcoding existing low-resolution media on high-resolution displays. An SR system has to be implemented in hardware if the appliance requires real-time processing, where the system produces outputs simultaneously with the inputs with finite latency. SR techniques that employ videos have been proposed in the literature [2]; however, they require multiple frame buffers, and are thus unsuitable for compact hardware implementation.

Considering the background above, in this paper, we focus on single-image SR. Single-image SR can roughly be categorized into the following three types: i) interpolation-based, ii) reconstruction-based, and iii) statistical- or learning-based single-image SR (e.g., see [3]). Interpolation-based algorithms employ digital local filters, such as bilinear filters, bicubic filters, and Lanczos filters, etc., for the interpolation of missing pixels, which causes blurring and aliasing in the resulting image. Reconstruction-based algorithms solve an optimization problem to reconstruct edges on images through many iterations of incremental conversions between high-resolution and low-resolution images. Statistical- or learning-based algorithms construct high-resolution image libraries through iterative learning. These three approaches may not fully satisfy both frame-rate and image-quality constraints of current digital home appliances.

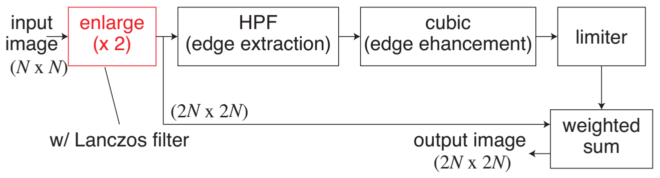
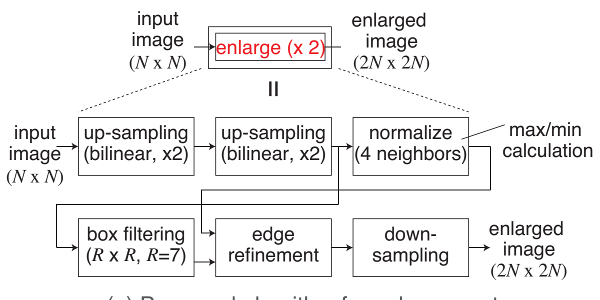


Figure 1: Gohshi's single-image super-resolution model [4]

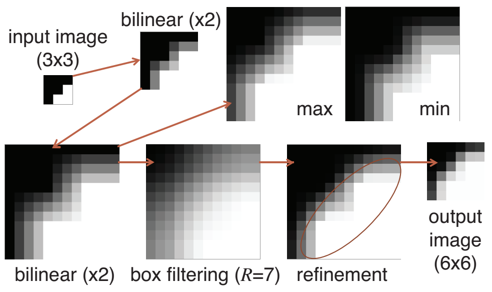
Recently, Gohshi proposed a novel straightforward algorithm for single-image SR [4]. The algorithm seems to be suitable for hardware implementation because it requires no iterations (and thus no frame buffers), while exhibiting drastically improved performance compared with the performances of conventional interpolation-based algorithms, by reproducing the frequency spectrum exceeding the Nyquist frequency. The process flow is illustrated in Fig. 1. A Lanczos filter will generally be employed for the enlargement of input images; however, upon hardware implementation, the filter requires many floating operations on wide filter kernels (Lanczos 2: 4$\times$4, Lanczos 3: 6$\times$6) [5]. Therefore, in this paper, we propose a novel enlargement algorithm based on box filtering that requires integer operations only between a small number of line buffers, while maintaining almost the same enlargement quality as Lanczos 2. Furthermore, we present novel circuits of the proposed enlargement algorithm and Gohshi's SR algorithm for an field-programmable gate array (FPGA), and show the simulation, synthesis, and experimental results.

2. Novel Enlargement Algorithm Based on Box Filtering

Figure 2 shows the concepts of our enlargement algorithm. As shown in Fig. 2(a), an input image ($N \times N$) is enlarged twice by upsampling with bilinear interpolation. Then, the enlarged image ($4N \times 4N$) is given to both a box filter and normalization units. The box filter performs blurring to attenuate jaggies in the enlarged image. Edge refinement of the box-filtered image is performed on the basis of the normalized data (local max and min data). Finally, the output image

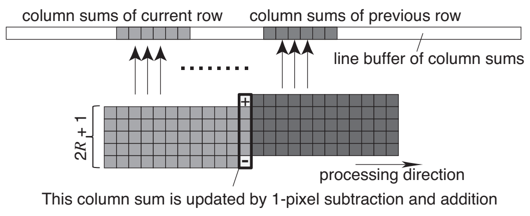


(a) Proposed algorithm for enlargement

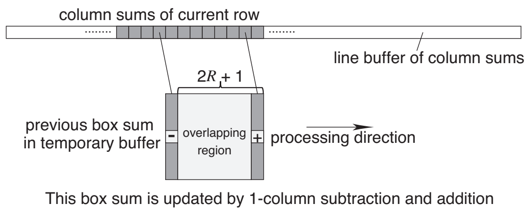


(b) Processing examples ($N = 3, R = 7$)

Figure 2: Process flow of proposed enlargement algorithm


column sums of current row
column sums of previous row
line buffer of column sums
2R + 1
processing direction
This column sum is updated by 1-pixel subtraction and addition

(a) Column-sum (*colsum*) calculation


column sums of current row
line buffer of column sums
2R + 1
previous box sum in temporary buffer
overlapping region
processing direction
This box sum is updated by 1-column subtraction and addition

(b) Target box-sum (*boxsum*) calculation

Figure 3: Efficient and fast box filtering

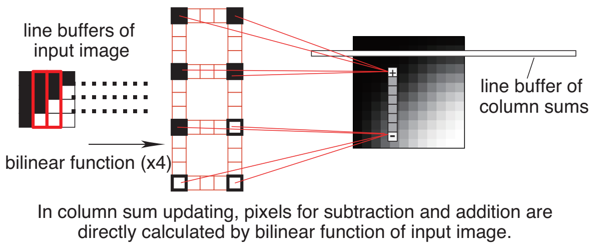

line buffers of input image
bilinear function (x4)
line buffer of column sums
In column sum updating, pixels for subtraction and addition are directly calculated by bilinear function of input image.

Figure 4: Box filtering of proposed enlargement (ex: $R = 7$)

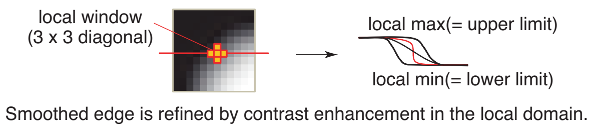

local window (3 x 3 diagonal)
local max(= upper limit)
local min(= lower limit)
Smoothed edge is refined by contrast enhancement in the local domain.

Figure 5: Edge refinement process

is obtained by downsampling, and the resulting image size is $2N \times 2N$. The process flow with a small input-image example (3×3) is shown in Fig. 2(b). It should be noted that inputs always flow to outputs straightforwardly in this model.

Generally, a blurring filter with a wide kernel is required to obtain smooth edges, and the number of calculations for convolution, *i.e.*, additions and multiplications, is given by $(2R + 1)^2$, where R represents the kernel radius in pixel counts. However, the number of calculations becomes independent of R if the kernel shape is limited to a box shape [6]. Therefore we here employ box filters that basically calculate the average of surrounding pixels inside a box region.

As shown in Fig. 3, by introducing a line buffer to keep the summed values in the column direction, the number of calculations in box filtering becomes independent of R. First, a summed value among $2R + 1$ pixels along a column centered by a selected row, which we call *colsum*, is calculated. Each *colsum* is stored in the line buffer at a corresponding column address. Then, *colsum* values of the subsequent row are given by the present *colsum* + (top pixel value of the target column) − (bottom pixel value of the column), as shown in Fig. 3(a). Likewise, $(2R + 1) \times (2R + 1)$ box filtering can be performed by summing $(2R + 1)$ *colsum*'s along a row centered by a selected column. We denote the summed value as *boxsum*. For the updates, similarly to updates of *colsum* values, the subsequent *boxsum* values are given by the present *boxsum* + (rightmost-column values of the target box) − (leftmost-column values of the box), as shown in Fig. 3(b). Consequently, box filtering with the line buffer requires i) accessing two pixels, ii) four addition/subtraction operations, and iii) a normalization operation. Furthermore, since the top and bottom pixel values used for updating *colsum* values described above can be obtained from the low-resolution $4\times$ image (outputs of the second bilinear process), pixel values of a box-filtered image can directly be obtained by calculation among four line-buffers that store a part of the low-resolution image (Fig. 4).

Edges of the box-filtered image are refined by conventional contrast enhancement based on normalization using maximum and minimum values in a local domain (Fig. 5). Finally,

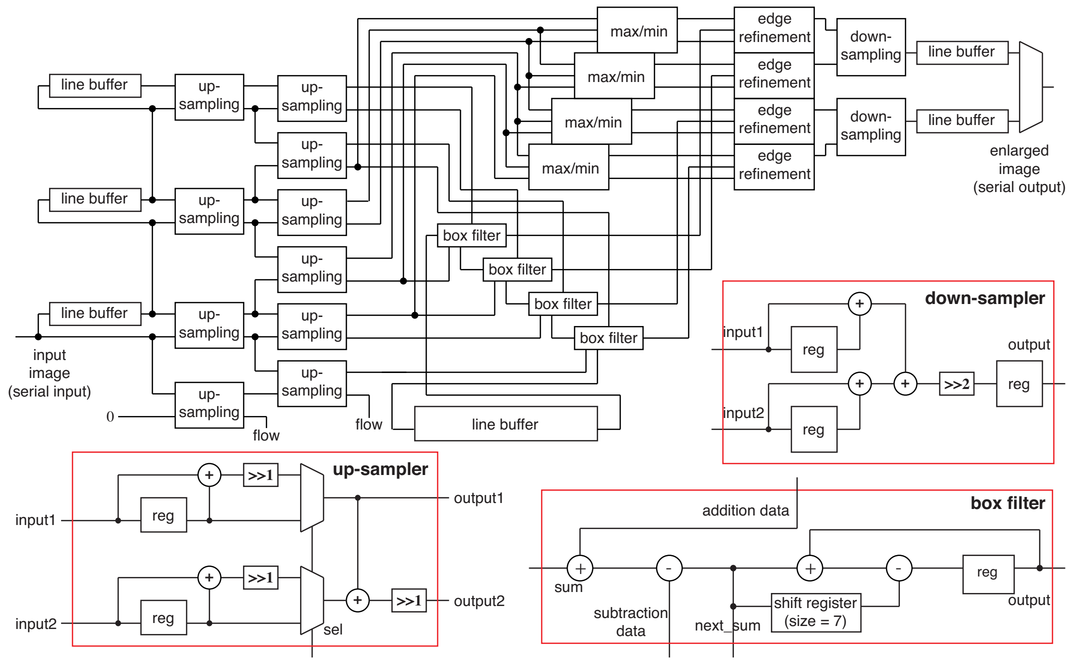


Figure 6: Overall view of proposed enlargement circuit with five line buffers

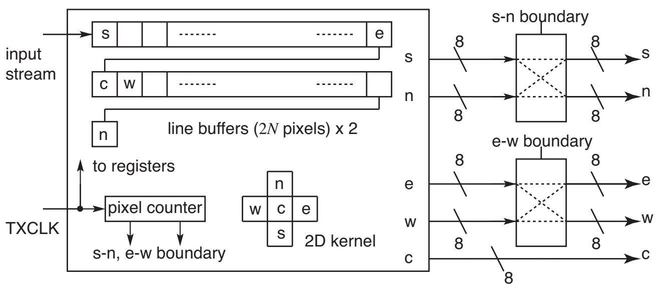


Figure 7: Kernel decoder of super-resolution filter

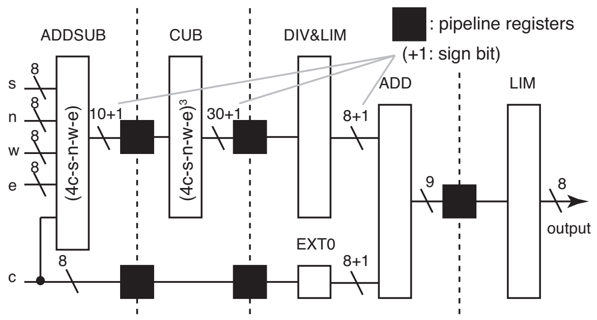


Figure 8: Super-resolution filter based on Gohshi's model

the edge-refined image is down-sampled, and the resulting image is obtained as a 2× enlarged image.

3. Hardware Implementation of Single-Image Super-Resolution with Proposed Enlargement Models

Figure 6 illustrates our enlargement circuits implementing the proposed algorithm. The circuit consists of five blocks: i) 4 (enlargement) + 2 (output control) line buffers, ii) 10 conventional upsamplers, iii) 4 box filters, iv) a conventional contrast enhancer consisting of four max/min and edge refinement modules, and iv) 2 conventional down-samplers. The input image is serialized, and then given to the enlargement circuit. The accepted pixel streams are processed in parallel (4 way), and the parallel outputs are bound by the down-samplers (to 2) and then reserialized by additional two line buffers and a selector. Note that the input and output of the enlargement circuit are represented by serial pixel-data streams.

The enlarged and reserialized stream is given to an SR kernel decoder (Fig. 7). The circuit extracts north (n), south (s), east (e), west (w), and center (c) pixel values from the input stream that is synchronous with a pixel-data transfer clock (TXCLK). The circuit also implements pixel counters to detect the vertical (s-n) and horizontal (e-w) boundaries (obeying the Neumann boundary). The extracted pixel values (s, n, w, e, c) are applied to a pipelined SR filter circuit (Fig. 8), where the ADDSUB module detects spatial edges, the CUB module enhances the edges, the DIV&LIM module compresses the enhanced edges and limits the compressed edges, the ADD module sums the limited-and-compressed edges and sign-extended c values, and the LIM module limits the summed value within the output bit width (8).

Table 1: Implementation and performance summary (of modules in Figs. 7 and 8 only)

Input Res.	Output Res.	Depth	ALUT&ALM counts	Register counts	FPGA CLK	VSYNC
400×400	400×400	8-bit gray	16,651 (SR only)	31,732 (SR only)	90 MHz	60 Hz

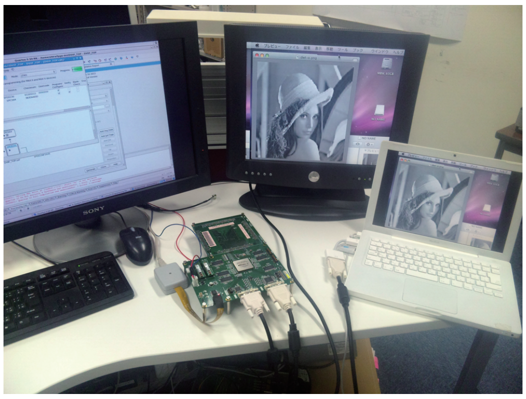


Figure 9: Experimental setups (enlarged input on right laptop monitor and SR output on center monitor)

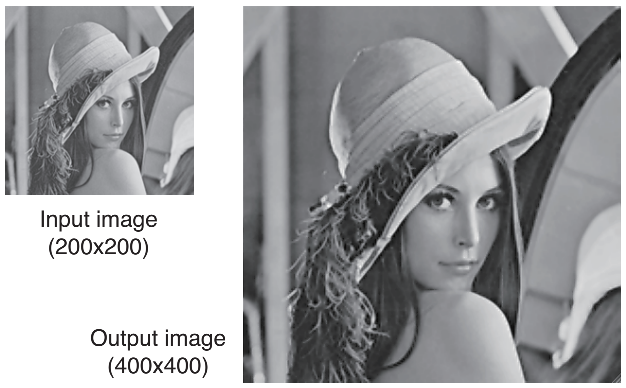


Figure 10: Demonstration of proposed super-resolution filter

4. Experimental Results

We implemented the proposed circuits on a commercial FPGA (MMS Co., Ltd., PowerMedusa, MU300-DVI, Altera Stratix II). The circuits shown in Figs. 7 and 8 were coded by VHDL, and were synthesized and place-and-routed by Quartus II. The input image (200×200) was applied to an RTL model of our enlargement block shown in Fig. 6 (coded by Verilog HDL), and the enlarged image was mirrored to the input DVI port of the FPGA board. The processed SR images (400×400) were displayed on a separate monitor connected via the output DVI port (Fig. 9). Then, the processed SR images were transmitted to a PC via an Inrevium TB-5V-LX330-DDR2-E board (Tokyo Electron Device, Ltd.). The input and processed SR images are shown on the left and right in Fig. 10, respectively. The image was flattened while the edges were clearly retained (Fig. 10 right). Table 1 summarizes the specifications and performance of the SR circuits on the FPGA. All the line buffers were implemented by FFs of the FPGA. The number of registers listed in Table1 includes registers in both primary circuits and line buffers.

5. Summary

We implemented an algorithm of single-image super-resolution (SR) [4] on an FPGA, where a novel hardware-oriented enlargement algorithm was employed. Although the proposed architecture has not been optimized well, one may further reduce the number of line buffers by considering the interfaces between the enlargement and SR blocks. Line buffers in the kernel decoder may be shared by an output line buffer in the last stage of the enlargement circuit.

Acknowledgment

This study was supported by a Grant-in-Aid for Scientific Research on Innovative Areas [20111004] from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan.

References

[1] http://www.itu.int/net/pressoffice/press_releases/2012/31.aspx#.UPNg2OS6eXg, Ultra high definition television: Threshold of a new age, ITU. 2012-05-24. Retrieved 2012-07-31.

[2] Q. Shan, Z. Li, J. Jia and C.-K. Tang: Fast image/video upsampling, ACM Trans. Graphics, Vol. 27, No. 5, pp. 1-7, 2008.

[3] Y. W. Tai, S. Liu, M. S. Brown and S. Lin: Super-resolution using edge prior and single image detail synthesis, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2400-2407, 2010.

[4] S. Gohshi: A new signal processing method for video —Reproduce the frequency spectrum exceeding the Nyquist frequency—, Proc. 3rd Multimedia Systems Conf., pp. 47-52, 2012.

[5] http://en.wikipedia.org/wiki/Lanczos_resampling

[6] M. J. McDonnell: Box-filtering techniques, Computer Graphics and Image Processing, Vol. 17, No. 1, pp. 65-70, 1981.