

## 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 [1]. We here propose a hardware-oriented image-enlargement algorithm for the SR algorithm based on frame-bufferless box filtering, and exhibit novel circuits of the proposed enlargement algorithm and the SR algorithm for FPGA, aiming at the development of single-image SR module for practical embedded systems.

### 1. Introduction

Super high-resolution displays, such as retina displays, 4K/8K ultra high definition televisions (UHDTV), and so on, have been spotlighted in digital home appliance [2]. Super resolution (SR) techniques, which increase resolution of images, are thus necessary for transcoding existing low-resolution media on high-resolution displays. A SR system has to be implemented on 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 [3], however, they require multiple frame buffers, and are thus unsuitable for compact hardware implementation.

Considering these backgrounds 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 [4]). Interpolation-based algorithms employ digital local filters such as bi-linear filters, bi-cubic filters, the Lanczos filters, etc., for 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 learnings. These three approaches may not fully satisfy both frame-rate and image-quality constraints of present digital home appliance.

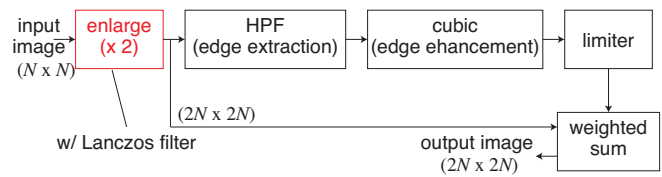


Figure 1: Gohshi's single-image super resolution model [1].

Recently, Gohshi *et al.* proposed a novel straightforward algorithm for single-image SR [1]. The algorithm seems to be suitable for hardware implementation because it requires no iterations (and thus no frame buffers), while exhibiting drastic performance as compared with performances of conventional interpolation-based algorithms, by reproducing the frequency spectrum exceeding the Nyquist frequency. The processing flow is illustrated in Fig. 1. The Lanczos filter will generally be employed for enlargement of input images, however, upon the hardware implementation, the filter requires many floating operations on wide filter kernels (Lanczos 2: 4x4, Lanczos 3: 6x6) [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 keeping almost the same enlargement quality as Lanczos 2. Furthermore, we exhibit novel circuits of the proposed enlargement algorithm and Gohshi's SR algorithm for FPGA, and show the simulation, synthesis, and experimental results.

### 2. Novel Enlargement Algorithm based on Box Filtering

Figure 2 shows 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 a normalization units. The box filter performs blurring to attenuate jaggies in the enlarged image. Edge refinement of the box-filtered image is performed based on the normalized data (local max and min data). Finally, the output image is obtained by down-sampling, and the resulting image size is  $2N \times 2N$ . These processing flow with a small input-image example ( $3 \times 3$ ) is

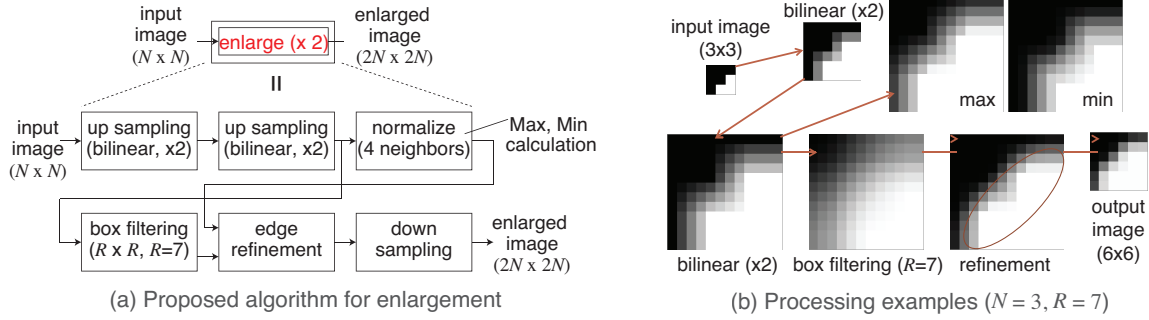


Figure 2: Processing flow of proposed enlargement algorithm.

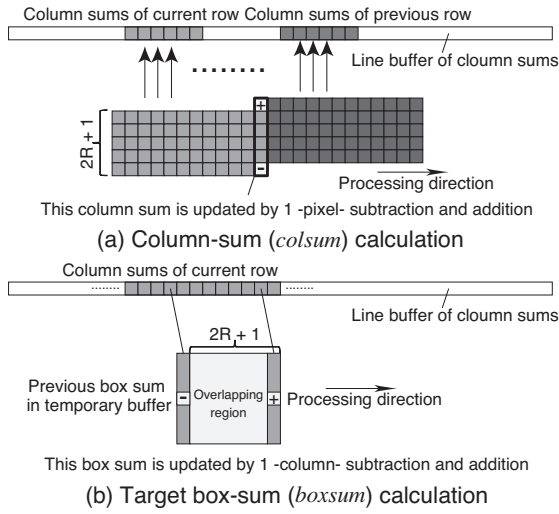


Figure 3: Efficient and fast box filtering.

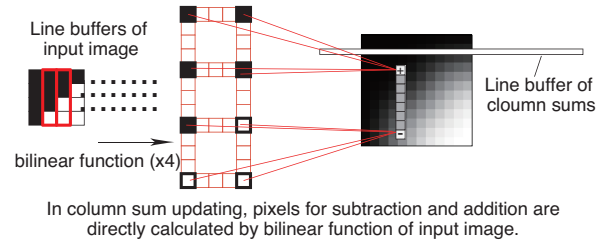


Figure 4: Box filtering of proposed enlargement (ex:  $R = 7$ ).

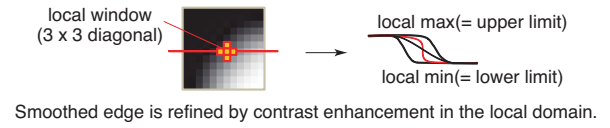


Figure 5: Edge refinement process.

shown in Fig. 2(b). It should be noticed that inputs always flow to outputs straightforwardly in this model.

Generally, a burring filter with a wide kernel is required for obtaining smooth edges, and the number of calculations for convolution 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 only [6]. Therefore we here employ box filters which basically calculate an average of surrounding pixels inside a box region.

As shown in Fig. 3, by introducing one line buffer for keeping summed values in 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 at the subsequent row are given by present *colsum* + (top pixel value of the target column) – (bottom pixel value of the column), as shown in Fig. 3(a). Likewise, a  $(2R + 1) \times (2R + 1)$  box filtering can be per-

formed 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 is given by 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 one line buffer requires i) accessing two pixels, ii) four additive/subtractive operations, and iii) normalization operation. Furthermore, since the top and bottom pixel values for updating *colsum* values described above can be obtained from the low-resolution 4x image (outputs of the second bi-linear process), pixel values of a box-filtered image can directly be obtained by calculating among four line buffers storing a part of the low-resolution image (Fig. 4).

Edges of box-filtered image are refined by conventional contrast enhancement based on normalization using maximum and minimum values in a local domain (Fig. 5). Finally, the edge-refined image is downsampled, and the resulting image is obtained as 2x enlarged image.

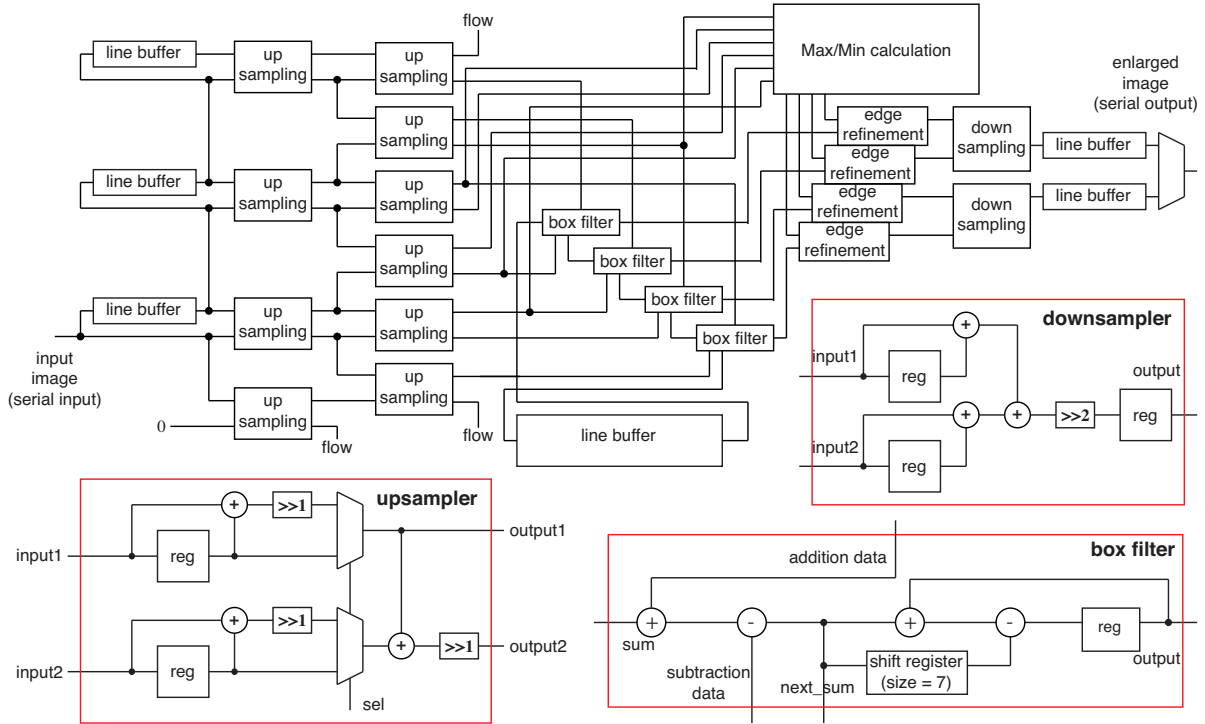


Figure 6: Overall view of proposed enlargement circuit with five line buffers.

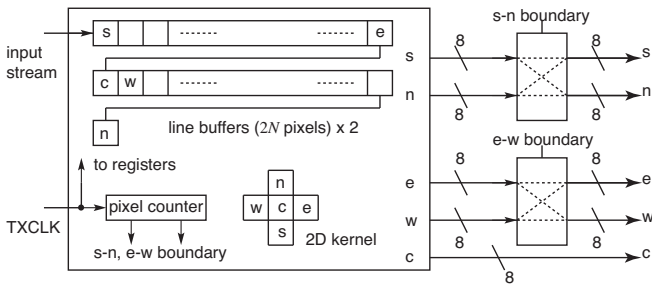


Figure 7: Kernel decoder of super resolution filter.

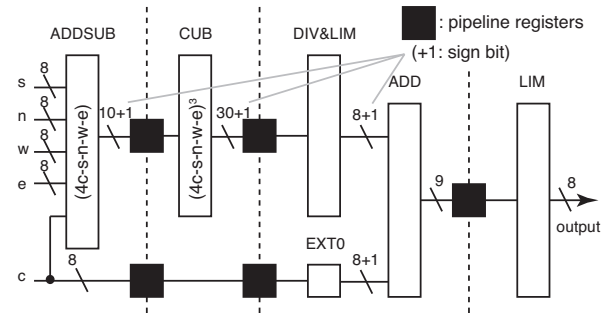


Figure 8: Super resolution filter based on Gohshi's model.

### 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) conventional contrast enhancer consisting of a max/min and four edge refinement modules, and iv) 2 conventional downsamplers. 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 downsamplers (to 2), and then re-serialized 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 re-serialized stream is given to a 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 being synchronous to 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 given to a pipelined SR filter circuit (Fig. 8), where ADDSUB module detects spatial edges, CUB module enhances the edges, DIV&LIM module compresses the enhanced edge and limits the compressed edges, ADD module sums the limited-and-compressed edges and sign-extended c values, and 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
400x400	400x400	8-bit gray	16,651 (SR only)	31,732 (SR only)	90 MHz	60 Hz



Figure 9: Experimental sets (enlarged input and SR output).

#### 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 (200x200) was given 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 (400x400) were displayed on a separate monitor through the output DVI port (Fig. 9). Then, the processed SR images were transmitted to PC via the inrevium TB-5V-LX330-DDR2-E board (Tokyo Electron Device, LTD.). The input and processed SR images are shown in Fig. 10 left and right, respectively. The image was flattened while the edges were clearly kept (Fig. 10 right). Table 1 summarizes specification and performance of the SR circuits on FPGA. All the line buffers were implemented by FFs of the FPGA. The the number of registers listed in Tab. 1 includes registers in both primary circuits and line buffers.

#### 5. Summary

We implemented an algorithm of single-image super resolution (SR) [1] on 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 interfaces between the enlargement and SR blocks. Line buffers in the kernel decoder may be shared by an output line buffer in the



Figure 10: Demonstration of proposed super resolution filter.

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] 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.
- [2] [http://www.itu.int/net/pressoffice/press\\_releases/2012/31.aspx#.UPNg2OS6eXg](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.
- [3] Q. Shan, Z. Li, J. Jia, and C.-K. Tang, "Fast image/video upsampling," ACM Trans. Graphics, 27(5), 2008.
- [4] Y.W. Tai, S. Liu, M.S. Brown, and S. Lin, "Super resolution using edge prior and single image detail synthesis," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2400-2407, 2010.
- [5] [http://en.wikipedia.org/wiki/Lanczos\\_resampling](http://en.wikipedia.org/wiki/Lanczos_resampling)
- [6] M.J. McDonnell, "Box-filtering techniques," Computer Graphics and Image Processing, 17(1), pp. 65-70, 1981.