

## Real-time HDTV to 4K and 8K-UHD conversions using anti-aliasing based super resolution algorithm on FPGA

Prasoon Ambalathankandy<sup>a,\*</sup>, Shinya Takamaeda<sup>a</sup>, Motomura Masato<sup>a</sup>, Tetsuya Asai<sup>a</sup>, Masayuki Ikebe<sup>a</sup>, Hotaka Kusano<sup>b</sup>

<sup>a</sup> Gradutae School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan

<sup>b</sup> Canon Inc., Tokyo 146-8501, Japan



### ARTICLE INFO

#### Keywords:

Super resolution  
Over up-sampling  
Anti aliasing  
Natural lookig  
Edge reconstruction/enhancement  
HDTV to UHD conversion  
Frame bufferless

### ABSTRACT

The demand for light-weight and high-speed super resolution (SR) techniques are growing because of super high-resolution displays, such as 4K/8K ultra high definition televisions (UHDTVs), which have become common. We propose a single pass over up-sampled anti-aliasing based SR method. Our method can attenuate jaggies and perform natural-looking contrast improvement focusing only on the shadow part in the edge of an enlarged image without the need to preserve the entire enlarged image. Therefore, this method is suitable for hardware implementation, and the proposed architecture requires five-line buffers only (in the memory section). We implemented the proposed method on a field programmable gate array (FPGA) and demonstrated HDTV-to-4K and -8K SR processing in real time (60 frames per second).

### 1. Introduction

Super high-resolution displays such as retina displays and 4K/8K ultra high definition televisions (UHDTVs) have been spotlighted among digital home appliances [1]. Thus, super resolution (SR) techniques, which increase the resolution of images, are necessary for transcoding existing low-resolution media on high-resolution displays. An SR system has to be implemented on hardware if the appliance requires real time processing, where the system produces outputs simultaneously with the inputs in finite latency. SR techniques that utilize videos have been proposed in the literature [2,4]; however, they require multiple frame buffers and are thus unsuitable for compact hardware implementation. Therefore, in this paper, we focus on single-image SR. Single-image SR can roughly be categorized into three types: i) interpolation-based, ii) reconstruction-based, and iii) statistical or learning-based single-image SR (e.g., see [5,6]). Interpolation-based algorithms utilize digital local filters such as bilinear filters, bi-cubic filters, and Lanczos filters for interpolation of missing pixels, which cause blurring and aliasing in the resulting image. Reconstruction-based algorithms solve the 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 the frame-rate and image-quality constraints of today's digital home appliances.

Recently, in the learning-based single-image SR [7,8], deep learning method, which is Super-Resolution Convolutional Neural Network (SRCNN), has been proposed and has attracted attention. SRCNN consists mainly of three processes - patch extraction, nonlinear mapping and reconstruction. When a low-resolution image is input to SRCNN, local features are extracted as the patch library in the first convolutional layer. Then, in the second layer, these features are nonlinearly mapped to the high-resolution patch representation. Finally, in the third layer, the high-resolution image is generated by its surrounding predictions. This method can generate high quality images with high PSNR values. However, this method requires enormous memory resources and calculation cost. For this reason, its implementation has been limited to the software.

We have studied anti-aliasing SRs for obtaining smooth edges [9,10]. In this paper, we present an FPGA-based SR implementation which generates not only jaggy-less edges using anti-aliasing, but also the output images have natural-looking edges. The remainder of this paper is organized as follows. Our super resolution algorithm is described in Section 2, and the FPGA implementation is presented in Section 3. In Section 4 we discuss our experimental results using a performance summary table, and conclude this work in Section 5.

\* Corresponding author.

E-mail address: [prasoon@lalsie.ist.hokudai.ac.jp](mailto:prasoon@lalsie.ist.hokudai.ac.jp) (P. Ambalathankandy).

## 2. Our super resolution algorithm

### 2.1. Related work

Surveying the current literature results in many related works. A hardware based SR system is presented in [11] which makes use of existing motion estimations from a decoding block and aims to minimize the memory cost. This proposed system makes use of very long instruction words (VLIW) and ARM processors. No deblurring or image sharpening is performed in this work. In [12] an FPGA-based iterative back projection (IBP) method is presented. This architecture requires memory access to all low-resolution (LR) frames used in the high-resolution (HR) estimate, and results are computed only after multiple passes through the hardware. This implementation reaches operating frequency of 80 MHz. According to authors [12] this is sufficient to allow real-time execution outputting 25 VGA  $2 \times$  super resolved frames, allowing implementation of a maximal of 10 iterations. The drawback of this implementation, according to authors of [13], are high memory requirements and the fact that, in order to output a super-resolved image, multiple passes through the hardware are required. The work in [13] reports FPGA implementation of a super-resolution image reconstruction based on IBP. In this approach, additional details are reconstructed based on exploitation of sub-pixel shifts caused by warping ([14]). The processing is done at pixel level by means of weighted mean optical flow, denominated weight based picture elements (pels) merging. Weight are estimated based on the inter-frames motion estimations for pel patch matching. The proposed architecture, implements 10 iterative stages, mapped on the FGPA device reaching a frequency of 58 MHz. In their configuration the system was could super-resolve 61 CIF LR images to  $1280 \times 720$  pels per second. However, satisfactory quality level requires 20 iterations, which appears to be beyond the capability of the supporting device. This limitation is due to the on-chip memory constraint. In [15] Szydik et al. proposed a non-iterative approach. However, their results report only QCIF to CIF super resolution for 25 FPS. According to the authors, their implementation is not scalable and limited to work with only two reference frames.

Gohshi et al. proposed a non-iterative algorithm for single-image SR

[16]. Their processing flow is illustrated in Fig. 1. After applying an interpolation filter, the algorithm performs edge detection and edge enhancement by high-pass filtering (HPF) and cubic function. This algorithm seems to be suitable for hardware implementation because it does not require iterations (and thus no frame buffers), while exhibiting better performance when compared with the performance of other conventional interpolation-based algorithms, by reproducing the frequency spectrum exceeding the Nyquist frequency. The Lanczos filter will generally be utilized for the interpolation of input images; however, upon implementing the hardware, the filter requires many floating-point operations on wide filter kernels (Lanczos 3:  $6 \times 6$ ) [17]. Moreover, line buffers are also needed between the interpolation stage and the edge-enhancement stage. In this paper, we propose a novel SR algorithm, which is based on the anti-aliasing method, it performs direct calculation of the anisotropic interpolation, local-statistics-based edge reconstruction and natural-looking edge enhancement, while requiring only five-line buffers. Our contributions are as follows:

- Anti-aliasing based SR method with redundant internal resolution and post down-sampling, whose output are less jagged.
- Low calculation cost edge reconstruction, utilizing local statistics.
- Natural-looking edge enhancement based on asymmetric weighted halo components.
- FPGA implementation for real-time HDTV to 4K- and 8K UHD conversion and its demonstration.

### 2.2. Proposed super resolution (SR) algorithm

Fig. 2 shows the concepts of our single-pass SR algorithm, that requires no frame buffers. With SR algorithms, it is difficult to construct good edges. Multi sampling with anti-aliasing makes jaggy free edges and this algorithm uses that concept. However, with jaggies it is easy to perform over enhancement. Additionally, we can apply simple edge enhancement techniques. In this method, an over-sampled image with jaggies is compressed into target resolution for suppressing them. Also, our algorithm generates an over-up-sampled image, enhances the edge until the jaggies occur, and compresses the image into the target

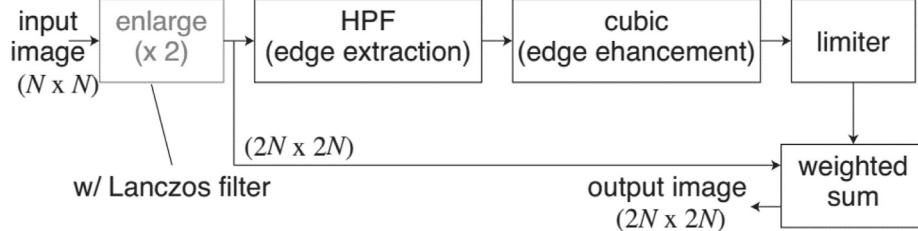


Fig. 1. Goshi's single image super resolution model [1].

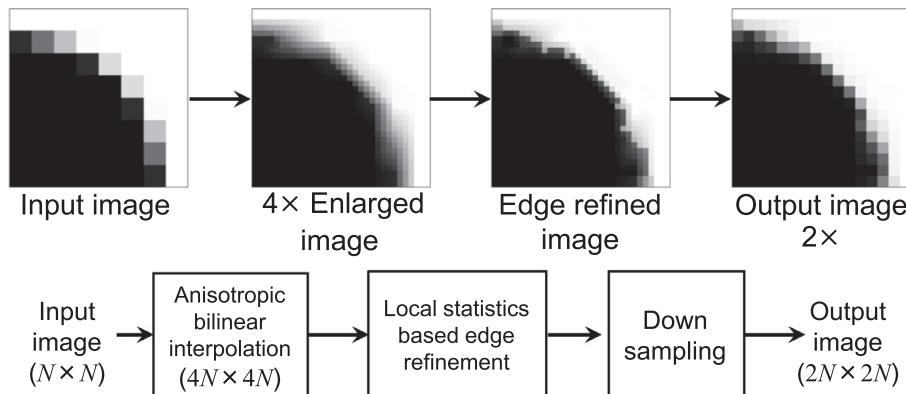


Fig. 2. Processing flow of the proposed anti-aliasing based SR algorithm.

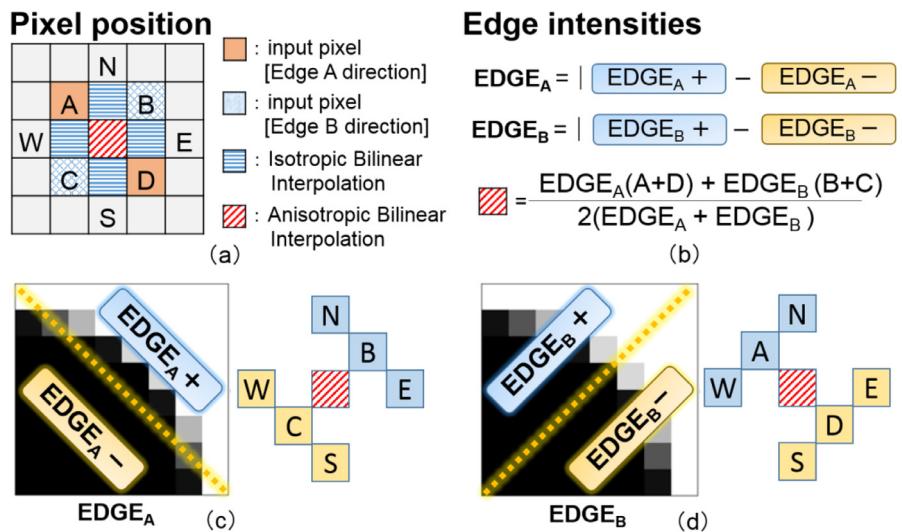


Fig. 3. Proposed anisotropic bilinear interpolation. (a) Pixel position. (b) Edge intensities. (c) Diagonal edge component  $E_A$ . (d) Diagonal edge component  $E_B$ .

### Stair-like artifacts

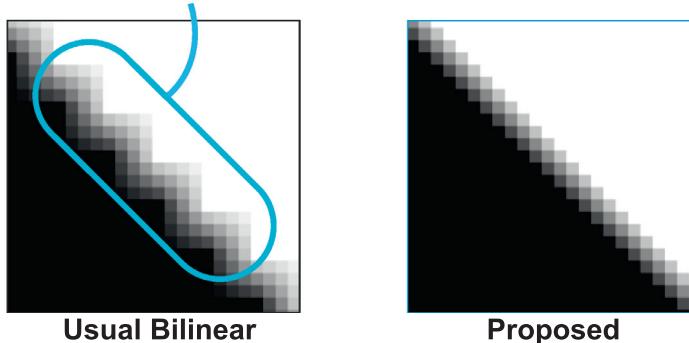


Fig. 4. Stair like artifacts in bilinear interpolation, while no visible artifacts in our proposed method.

resolution. An input image ( $N \times N$ ) is enlarged twice by up-sampling using anisotropic bilinear interpolation. Then, the enlarged image ( $4N \times 4N$ ) is given to local-statistics-based edge-refinement units. These units compute the mean, middle, max and min values, which are required for detecting the local edges and reconstructing those edges. They also handle the quasi local mean prediction, and compute asymmetric weighted halo components, which are required for the natural-looking edge enhancement. Finally, the output image is obtained by down-sampling, and the resulting image size is  $2N \times 2N$ . This method requires no external memory and only line buffers are required, as mere calculations are done.

### 2.3. Anisotropic bilinear interpolation

Reflecting diagonal components in images is important to enable high quality interpolation. Therefore, we utilize anisotropic bilinear filtering for refining the diagonal components with fewer pixels. Our interpolation method can be understood from the following equations:

$$\text{EDGE}_A = |P_N + P_B + P_W - (P_E + P_C + P_S)| \quad (1)$$

$$\text{EDGE}_B = |P_N + P_A + P_E - (P_W + P_D + P_S)| \quad (2)$$

$$P_{\text{CENTER}} = \frac{\text{EDGE}_A(P_A + P_D) + \text{EDGE}_B(P_B + P_C)}{\text{EDGE}_A + \text{EDGE}_B} \quad (3)$$

In the above equations,  $P_X$  is input pixel;  $\text{EDGE}_A$  and  $\text{EDGE}_B$  are components of Prewitt filtering for diagonal directions.

As shown in Fig. 3(a), the interpolated pixel, which is placed at the

centre, reflects the connection between the value of diagonal pixels by obtaining diagonal-edge components and a weighted sum of pixels at diagonal positions, as shown in Fig. 3(b). When the diagonal edge components ( $\text{EDGE}_A$  and  $\text{EDGE}_B$ ) are close to equal, blurring occurs in the interpolated edge image because this processing is also close to the usual isotropic bilinear filtering. Of course, in our horizontal/vertical interpolation, the process achieved is the same as isotropic one with low calculation cost. Although, this process increases the edge blurring, it can be refined using the next edge- refinement/enhancement process. Fig. 4 shows the result of our interpolation. Stair-like artifacts were observed in the usual bilinear interpolated patch. On the other hand, we confirmed that the smoothed diagonal edge with blurring was obtained using our interpolation.

### 2.4. Local statistics based edge reconstruction

In this section, we describe the proposed edge reconstruction. We use statistics which are max, min, mean, and middle in  $5 \times 5$  local patches. The local middle value stands for  $(\text{local max} + \text{local min})/2$ . For the edge reconstruction, it is important to process the edge and gradation area separately and to extract the edge shape. Since our method performs same filtering for all the pixels, it means that our reconstruction does not judge whether it is an edge or a gradation. Therefore, there is a possibility that artifacts such as Mach bands occur in gradation part.

Here we describe the edge-part detection using the mean and the middle for suppressing the edge reconstruction in the gradation part. Figs. 5 and 6 shows the mean and the middle values in edge/gradation

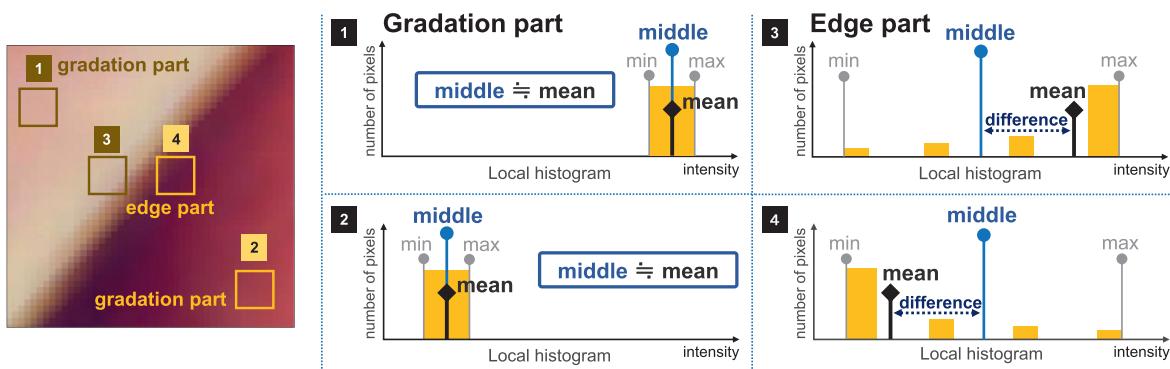


Fig. 5. Local histogram statistics for gradation and edge.

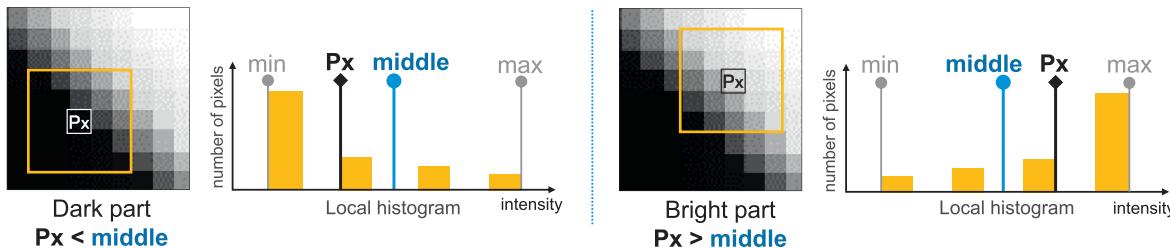


Fig. 6. Local histogram statistics for the edge condition.

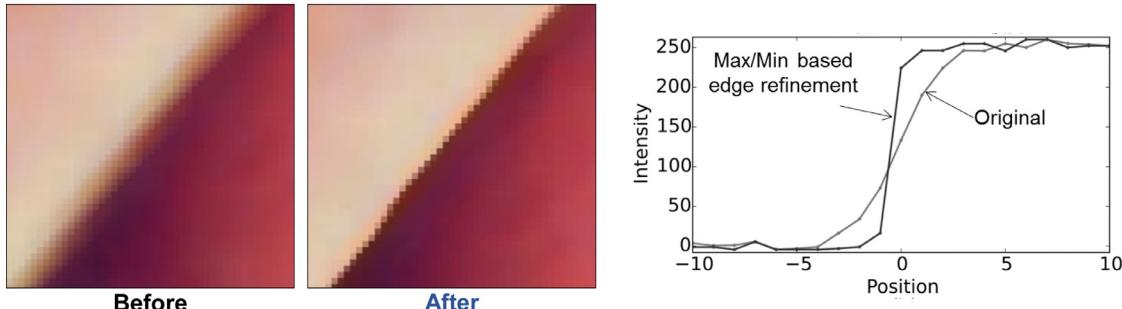


Fig. 7. Proposed edge reconstruction.

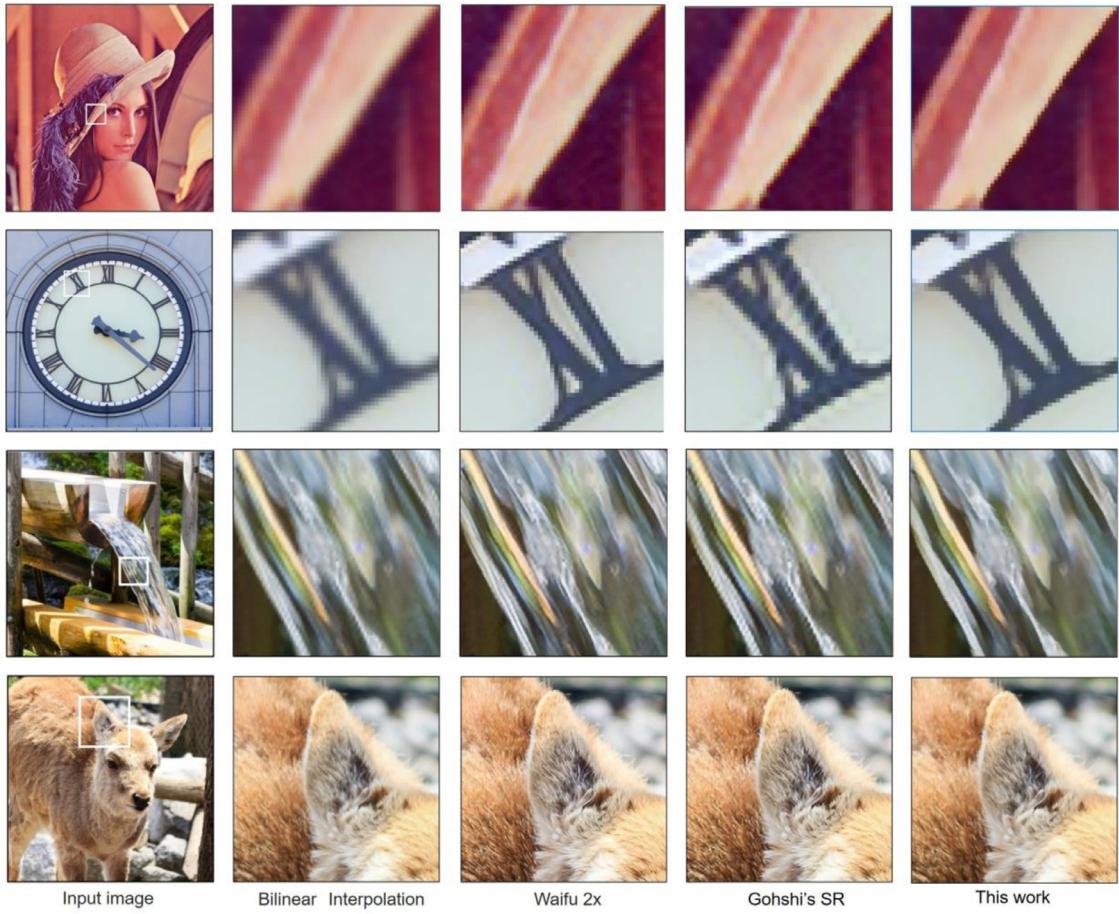
part local histogram. When the filter is in the gradation part, pixels have small variation; the middle becomes similar to the mean. Therefore, the difference between them becomes nearly zero. This characteristic does not change when the filter moves through the gradation part, and the difference remains substantially zero. On the other hand, the edge-part histogram has bias, so the difference has larger values. Thus, we can use the middle and the mean for the edge-part definition. We can also detect the edge part using high pass filtering. However this processing also detects noise components; the middle and the mean of which are robust for the noise.

Next, we consider the positions of the centre pixel and local middle value in the histogram. When the centre pixel of a local patch belongs to a dark section near the edge, the centre pixel is smaller than the local middle value. However, if the centre pixel is in the bright section; the centre pixel is larger than the local middle value (in Fig. 7). Therefore, we can use the centre pixel and the local middle value to find the edge condition. Although this edge detection is very simple, it evaluates only the difference between the centre pixel and local middle value. Because calculating the local max/min includes sorting operations, the middle value is also calculated. Using these components drastically refines only the blurred edge in the range from the min to the max (in Fig. 7). Although the edge-reconstructed image has jaggies, down sampling can suppress them.

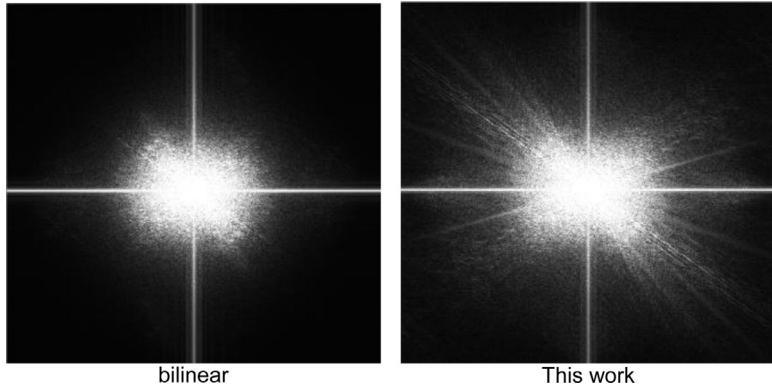
We found our SR algorithm generated a clearer  $2 \times$  image without artifacts near the edge (in Fig. 8). Also, the FFT spectrum of our SR image demonstrated that the image has high-frequency components over the Nyquist frequency (in Fig. 9). Also, we calculated the peak signal-to-noise ratio (PSNR) for 12 color images outputted by several methods (in Fig. 10). Waifu2x [18] is an SR method with a deep-learning based algorithm [7]. The Waifu2x showed high PSNR values. However, its hardware implementation is difficult because of calculation complexity. Our SR algorithm is suitable for hardware implementation, and it could attain close high PSNR values comparable to Waifu2x.

## 2.5. Natural-Looking edge enhancement by controlling light and dark halos

The details hidden in an image can be revealed by enhancing the contrast. Most of these contrast enhancement methods rely upon adding the original image with its own high pass filtered image. However, these methods can introduce undesirable artifacts around the edges known as halos. Edges, in an image are the most important information, which are used by many image processing/machine vision applications in their problem-solving operations. The presence of halos can make images appear unnatural and visually unpleasant. Furthermore, they could lead to loss of intensity scale and finer details. Also, it could lead



**Fig. 8.** Comparison with other SR methods.



**Fig. 9.** FFT spectrum of SR images (reference image is Lenna).

to wrong outcomes, in image processing applications that rely on these edge information. Suppression of halo artifacts have been studied extensively for multilayer images and several solutions have been proposed that make use of some form of an edge preserving filter. Kimura and Ikebe [19] proposed a halo reduction and control method that utilizes a weighted local histogram. In this [19], they differentiate and group these halo artifacts into light and dark halos and studied the effectiveness of these halos on visual perception. Their study concludes that presence of light halos creates a feel of unnaturalness, whereas, dark halos give an impression of clearer edges and thus was more acceptable to the participants and were perceived as closer to the natural image.

Drawing inspiration from their work [19], we make use of dark

halos as a means to enhance the dark edges thus improving the overall appearance of the resulting image. In Fig. 11, we can notice the impact of light and dark halos. It may be noted here that the lighter halos tend to give abnormal features around the edges, altering the overall impression of the image. Whereas, the dark halo (see Fig. 11) enhances the edges, making it easier for the viewers to take notice of the object boundaries, thus appearing visually more acceptable. We make use of two user defined gain parameters  $\alpha$  and  $\beta$  to control the light and dark halos individually, their upper ( $0.2 < \alpha < 1.0$ ) and lower ( $0.2 < \beta < 0.5$ ) values have been determined experimentally by subjective evaluation. The influence of these parameters on output images are presented in Fig. 11(b). This edge enhancement method is very suitable for our implementation, as we can see that our edge

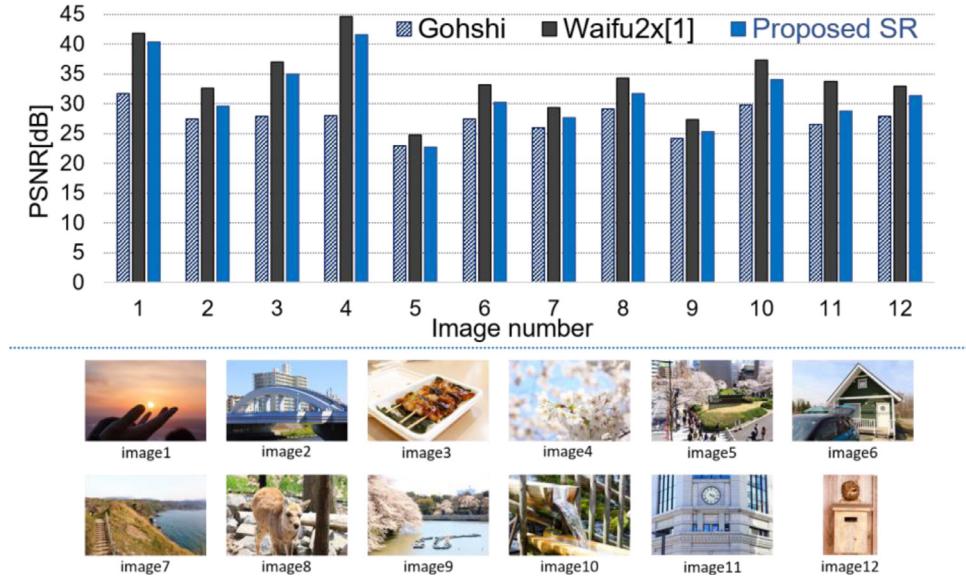
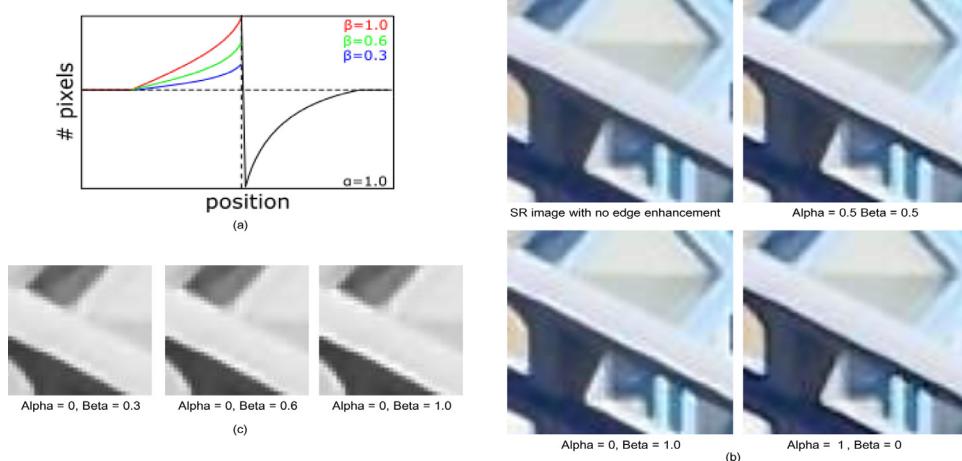


Fig. 10. Performance comparison with other SR methods. .

Fig. 11. (a) High frequency component. (b) SR image with varying gain parameters: (top-left) no edge enhancement, (top right) Equal gain  $\alpha = \beta = 0.5$ , (bottom left)  $\alpha = 0$  and  $\beta = 1$  and (bottom-right)  $\alpha = 1$  and  $\beta = 0$ . (c) Impact of gain parameter  $\beta$  on the output image, higher  $\beta$  makes light halos more prominent, thus giving an unnatural impression.

reconstruction module is able to clearly detect between light and dark edges (see also Sect. 2.4). Thus, our edge enhancement method can easily adapt and exploit the potential of the edge reconstruction algorithm to detect and group edges, while grouping halos.

### 3. FPGA implementation

#### 3.1. Management of color space

RGB signals from HDMI input are 10-bit resolution. When the SR operations are applied to all RGB components, the SR system requires three-times HW resources. Furthermore, individual management of each RGB components may generate color artifacts near edges. Therefore, in our FPGA implementation, we converted the RGB color space to HSV and applied the SR operation to only the V component. In the reverse conversion HSV to RGB, each RGB component is calculated by using the ratio of the outputted/inputted ( $V'/V$ ) component. The conversion equation is given as Eq. (4). S denotes each RGB component.

$$S' = (V'/V)S \quad (4)$$

#### 3.2. Overall view of proposed SR circuit

Fig. 12 illustrates our SR circuits implementing the proposed

algorithm. The circuit consists of four blocks:

- (i) 5-line buffers for an input image. ( $1920 \times 5 \times 10$  bits)
- (ii) 4 anisotropic-bilinear interpolators. (each unit has  $16 \times 10$ -bit input and output)
- (iii) 16 edge refinement units. (each unit has  $25 \times 10$ -bit input and  $16 \times 10$ -bit output)
- (iv) 4 conventional down-samplers. (each unit has  $4 \times 10$ -bit input and  $1 \times 10$ -bit output)

The process is as follows: The input image is serialized, and then given to the interpolator. Five lines of the input image is extracted using the line buffers. We extract the  $5 \times 5$  neighborhood for further processing by anisotropic bilinear interpolator. The accepted pixel streams are processed in parallel (4 way); the parallel outputs are also processed in parallel (16 way) at the edge-refinement stage; the outputted 16 pixels are bound by the down-samplers (to 4 pixels). Note that the input and output of the SR-system circuit are represented by serial pixel-data streams.

#### 3.3. Anisotropic-bilinear interpolation circuit

Fig. 13 shows the anisotropic-bilinear interpolator. It consists of two blocks:

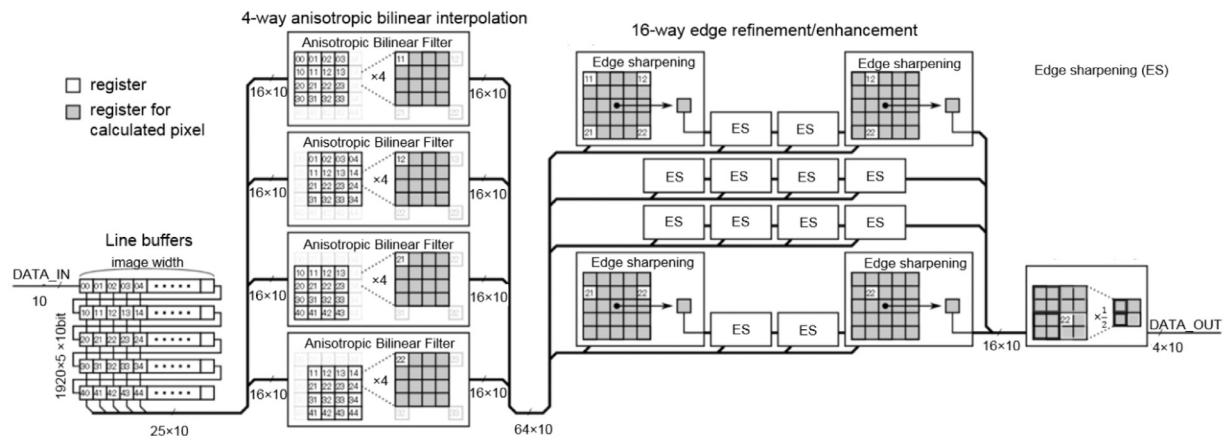


Fig. 12. Overall view of proposed SR circuit with five-line buffers.

- (i) Diagonal-edge detection module.
- (ii) 4 × enlargement module. This module includes two sub blocks:
  - (a) Anisotropic-bilinear filter module.
  - (b) 2 × enlargement module.

The diagonal-edge detection module calculates the absolute edge intensities (EDGE\_A and EDGE\_B) like Prewitt filtering. The anisotropic-bilinear interpolation module generates the center pixel (PIXEL\_OUT) from the ratio of the diagonal edges (EDGE\_A and EDGE\_B). We use a Xilinx IP for the divider that calculates the ratio. The 2 × enlargement module including the anisotropic-bilinear interpolation module generates four pixels for 2 × enlargement processing (three gray pixels of OUTPUT and pixel “00”). Pixels (a) and (b) are calculated from a usual bilinear function with adding and shift operations. Each pixel is

enlarged to  $4 \times 4$  pixels (15 Gy pixels of CORE\_OUT and pixel “11”) through cascaded  $2 \times$  enlargement modules at the end of this core.

### 3.4. Edge reconstruction circuit

Fig. 16 shows an edge refinement core. It consists of four blocks:

- (i) Local mean module.
- (ii) Local middle value module. (Bitonic sorter)
- (iii) Max/Min module.
- (iv) Edge enhancement module.

The  $5 \times 5$  interpolated pixels are given to the edge refinement core. We utilized a Bitonic sorter to calculate middle, max, and min values (in

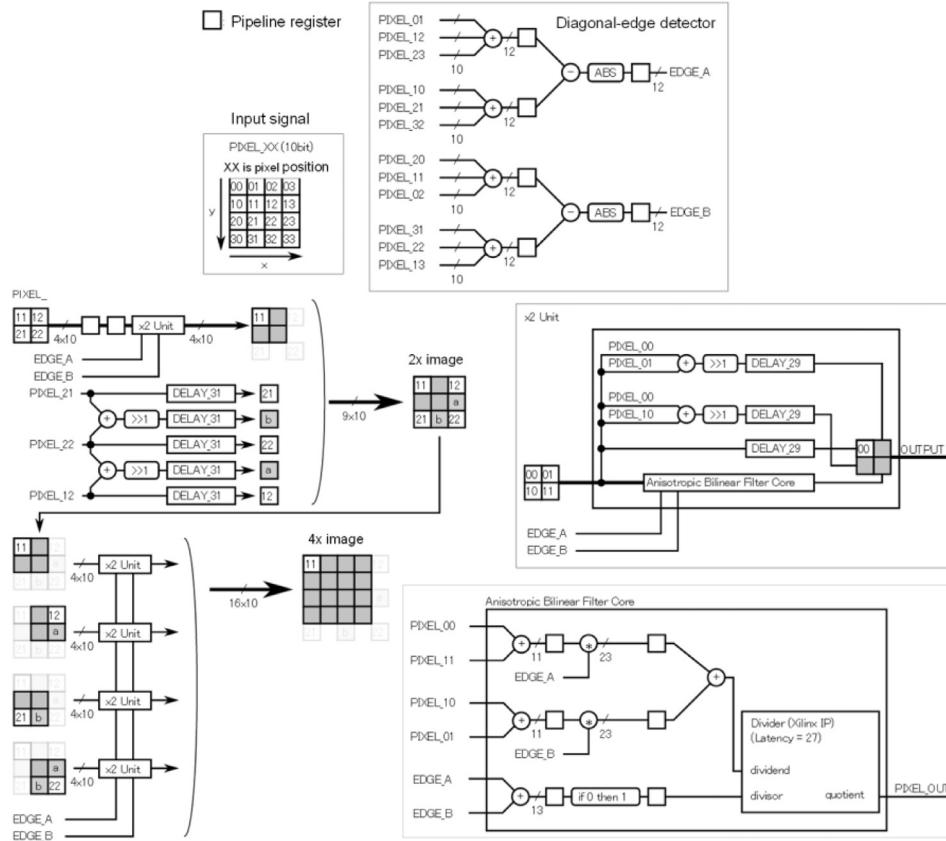


Fig. 13. Anisotropic-bilinear interpolator.

**Fig. 14).** The Bitonic sorter performs high-speed sorting; it is suitable for parallel implementation. However, it requires  $2n$  inputs, so we implemented the sorter with 32 inputs for 25 pixels. The seven inputs of the sorter are dummies. The calculated local middle value:  $(\text{Sorted } [0] + [\text{24}])/2$  is used for Max/Min processing. In this processing, the deference between the center pixel and the middle value selects the local max/min (max: Sorted [24], min: Sorted [0]). This means that contrast near the edge is refined in the range from the local min to the local max. Furthermore, the edge enhancement can also be performed by adding the edge component, which is the difference between the local mean and the refined edge at the end of this core (in Fig. 15).

### 3.5. Edge enhancement circuit

Figs. 16 and 17 shows our edge enhancement core, this module consists of the following blocks:

- Average L and H, computes local averages for light and dark patches.

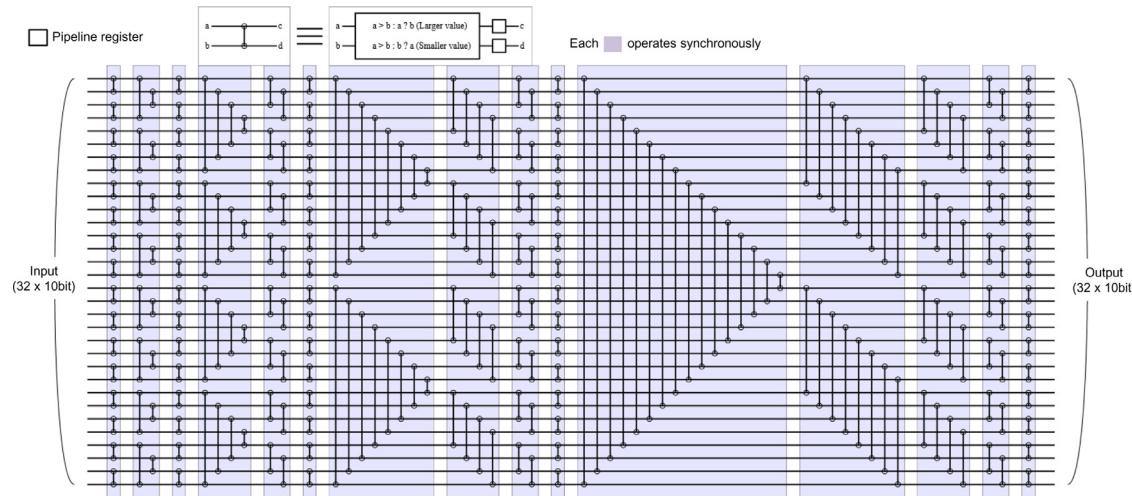


Fig. 14. Bitonic sorter ( $32 \times 10$  bit).

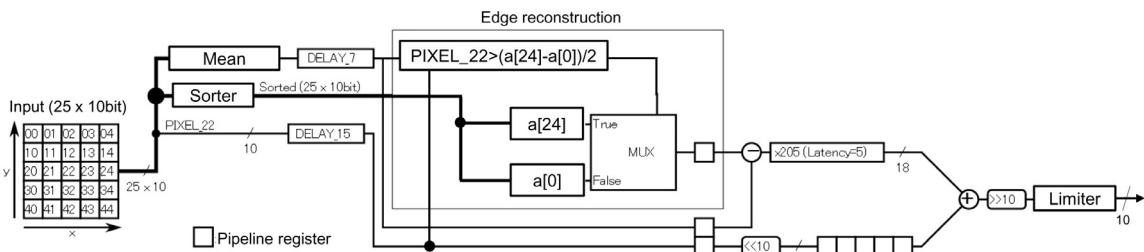


Fig. 15. Edge reconstruction circuit.

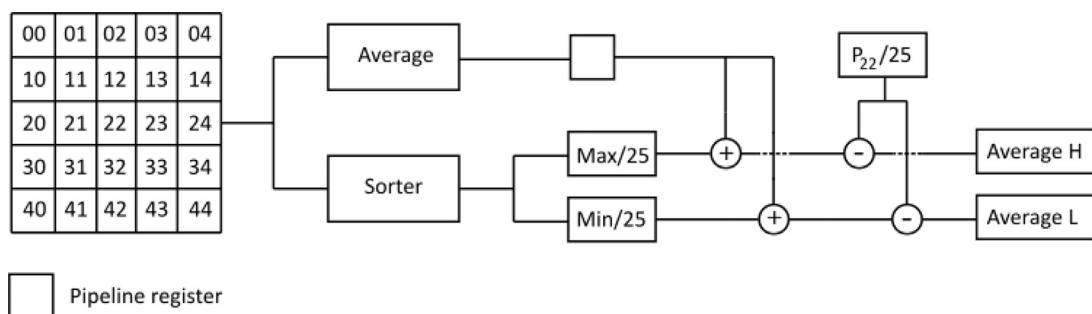


Fig. 16. Edge enhancement module (1).

- Normalized local minimum and maximum values.
- Multiplexers to make appropriate logical selections.

### 4. Experiment results

We implemented the proposed circuits on a commercial FPGA (Tokyo Electron Device, LTD., inrevium, TB-7K-325TIMG, Xilinx Kintex-7). The circuits shown in Figs. 12 to 17 were coded in Verilog HDL and were synthesized and placed and-routed by ISE. The input image ( $1920 \times 1080$ ) was given through the HDMI FMC board (Tokyo Electron Device, LTD., TB-FMCH-HDMI2). The processed SR images were separated to four  $1080 \times 1080$  p regions and outputted through the HDMI FMC boards. The four  $1080 \times 1080$  p images were processed by a 4K-UHD combiner (QMC-41HH-PRO); the 4K output was connected to a 4K display (ASUS 287Q). Figs. 18 and 19 show our experimental sets. For the 4-way  $1080 \times 1080$  p output, we implemented 2-way SR systems to the FPGA for the top/bottom  $2 \times 1080 \times 1080$  p outputs. This implementation can be shrunk if the FPGA board corresponds to the 1-way 4 K output. In the case of the 8K-UHD evaluation, one HDMI output of the first FPGA

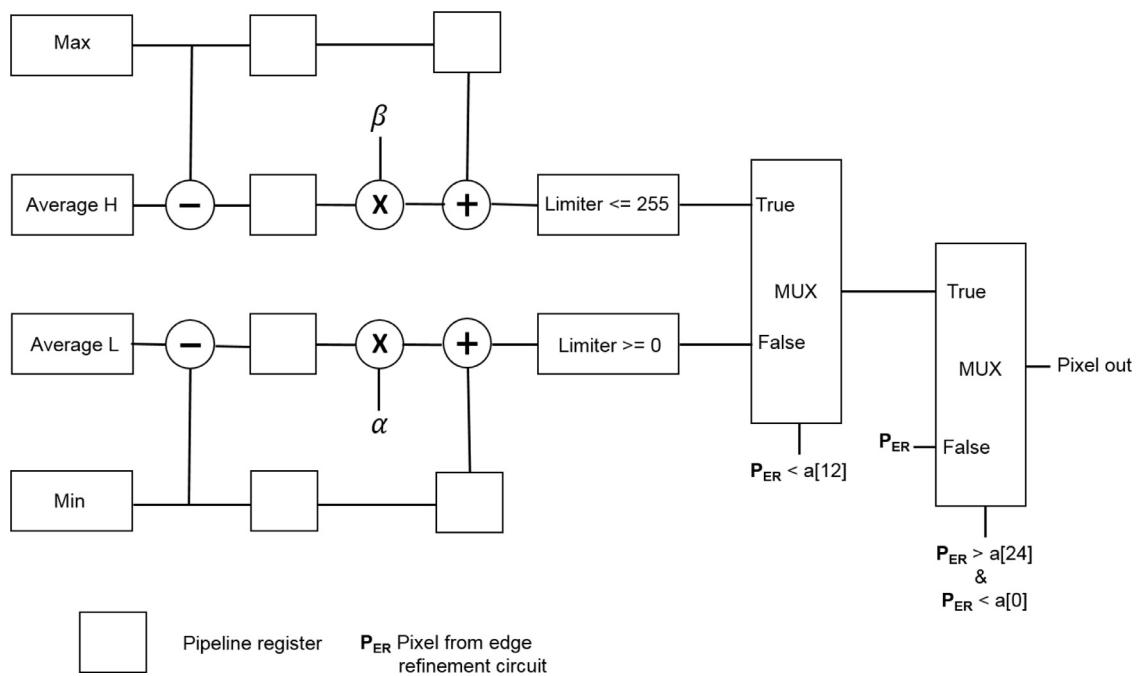


Fig. 17. Edge enhancement module (2).

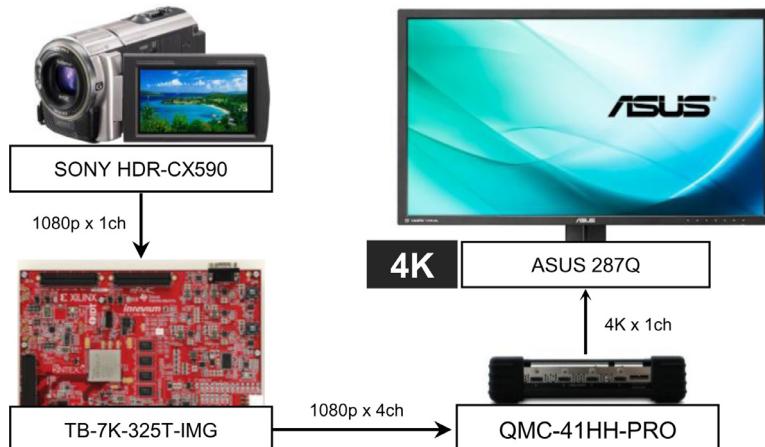


Fig. 18. Experimental setup for 4K resolution. .

board was cascaded to the second FPGA board with the same SR implementation. If an input movie is specified as 1080 p, the cascade connecting can be allowed in our SR system. For full 8K outputs, the system requires five FPGA boards in this implementation. We

determined one part of the 8K outputs using two FPGA boards. Table 1 summarizes the specifications and performance of the SR circuits on the FPGA. All the line buffers were implemented using the block RAMs of the FPGA. The number of registers listed in Table 1 includes registers in

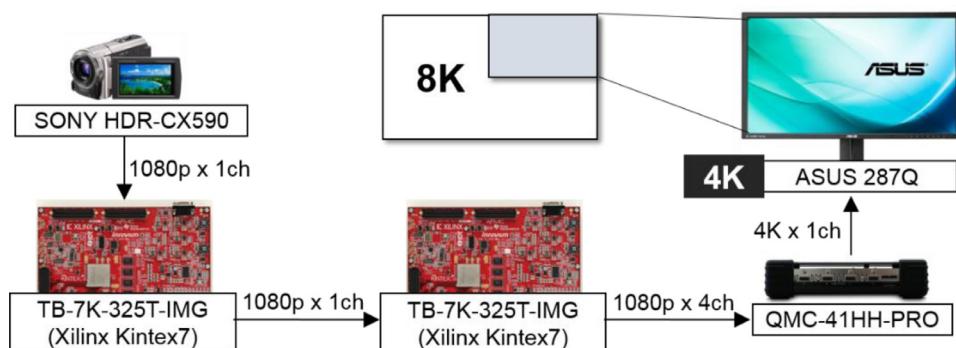


Fig. 19. Experimental setup for 8K resolution. .

**Table 1**  
Implementation summary.

| Module (FPGA resources) | Slices (50,950) | Slice Reg (407,600) | LUTs (203,800) | LUTRAM (203,800) | BRAM FIFO (445 + 890) | DSP48E1 (840) |
|-------------------------|-----------------|---------------------|----------------|------------------|-----------------------|---------------|
| Total <sup>a</sup>      | 19,541          | 89,556              | 68,051         | 1,692            | 75                    | 104           |
| Interpolator            | 5283            | 22,291              | 13,653         | 844              | 69                    | 40            |
| Edge refinement         | 14,191          | 67,024              | 57,656         | 848              | 0                     | 64            |
| Edge enhancement        | 320             | 320                 | 420            | 496              | 0                     | 0             |

<sup>a</sup> These results were obtained from 2-way SR implementation corresponding to 4-way 1080p outputs

**Table 2**  
Performance summary.

|             |                       |
|-------------|-----------------------|
| Input       | 1920 × 1080 (Full HD) |
| Output      | 3840 × 2160 (4 K UHD) |
| Color space | RGB:10-bitb           |
| Framerate   | 60 FPS                |
| FPGA clock  | 162 MHz               |

<sup>b</sup>Our SR system converts RGB color to HSV and calculates only V value for the SR operation.

both primary circuits and line buffers. Table 2 summarizes the performance of our SR system.

Our's is a single pass single image SR algorithm. Recall, from Figs. 8 and 10 our algorithm outputs are of good quality, as is evident from consistent high PSNR values. In table III we now compare our proposed hardware system with other real-time hardware implementations. Along with this work, [9] and [15] are based on non-iterative algorithms, whereas implementations in [3] and [4] are iterative algorithms. Hence they will have associated latencies. The non iterative method proposed in [15] is based on a Non-Uniform Grid Projection Algorithm (which they call SRiuma) [21] can resolve only QCIF to CIF resolution, and this implementation is not scalable. Also, this work is known to be limited to work with two frames only. The implementation in [9] is reported to resolve a 200 × 200 image to 400 × 400 output image. The implementation in [3] requires 20–50 iterative stages to super resolve CIF input image to 1280 × 720 output image. The bottleneck for this implementation is block RAM, as the amount of BRAM scales linearly with the image width. Thus, if processing larger images, frame rate has to be reduced. The work reported in [4] also requires 10 iterations to render twice super resolved frame while processing VGA size input images. This implementation requires memory access to all low resolution frames used for estimating the super resolved pixels, while making multiple passes through the hardware. All these earlier works are reported to super resolve smaller resolution images. Thus, they would require to operate at much higher frequencies with larger memories if they are to process higher resolution input images. However, with an operating frequency of 162 MHz and comparatively smaller memory usage our implementation is able to up-scale 1920 × 1080 input images to 4 K UHDs. The work reported in [20] implements a super-resolution algorithm on a CPU, GPU and many

FPGA platforms. They report a SR hardware implementation on Xilinx Virtex 5 device, and by utilizing pixel level and task level optimization they were able to achieve faster implementations. The footprint of one of their SR implementation on Xilinx Virtex 5 (xc5vlx110t) with a parallelization factor of 4 is as follows: LUT usage is 22% and 89% RAMB for up-sampling a 1920 × 1080 image to 3840 × 2160. This SR algorithm is based on [22] which assumes that resolution adaptation is only applicable at lower bitrates. This assumption could lead to content dependence on applications. There are several new SR algorithms being reported based on neural networks like the recent ones [7,23,24]. However, hardware implementations of such algorithms will require complex designs and have high computational cost.

## 5. Conclusion

We implemented an anti-aliasing based super resolution algorithm that performs HDTV to 4K and 8K Ultra HD conversions in real-time. Our single pass up-sampling algorithm went beyond the Nyquist frequency without generating any artifacts. Our hardware implementation requires no frame buffers thus making them attractive for practical real-life applications. The experimental results show, that the edge enhancement module presented here are able to produce images with more natural looking edges. The hardware system has been implemented on a Xilinx Kintex-7 device and the output images verified using a UHD display device for 1920 × 1080 input images. As our implementation is able to attenuate jaggies in the enlarged image edges, and it is not required to preserve the entire enlarged image making it relevant for applications that pose hard memory constraints. Modern day cellphones, portable GPS devices monitors are known to process red, green and blue channel at subpixel levels. However, these hand-held devices have strict area and power constraints. Thus, SR algorithm and an implementation technique like ours can be an attractive option. As for future work we can consider implementing a GPU-based SR system. Using training based methods can produce high resolution images better than the input images as they can establish relationship between the super-resolved images and the low-resolution input images. With this knowledge these algorithms can predict the missing high-resolution details for the low-resolution input images. Such learning methods can also be considered for future algorithm improvement.

**Table 3**  
Comparison with related works.

| Algorithm  | System CLK | Resolution                | FPS | System (Footprint)                             | Memory                  |
|--|------------|---------------------------|-----|--|-------------------------|
| [3]IBP (20–50 iterations)  | 58 MHz     | CIF → 1280 × 720          | 61  | Xilinx Virtex II (Slices – 8% + DSP – 35%)     | On-chip + 8 MB external |
| [4] IBP (> 10 iterations)  | 60 MHz     | VGA → 2 ×                 | 25  | Xilinx Virtex II (Slices – 42%)                | BRAM 40%                |
| [15] SRiuma* (Non-iterative)                                     | 109 MHz    | QCIF → 2 ×                | 25  | Xilinx Virtex 5 (Slices – 22% + DSP – 12%)     | BRAM 44%                |
| [9]Bilinear interpolation + Box filtering (Non iterative)        | 90 MHz     | 200 × 200 → 400 × 400     | 60  | Altera Stratix II (ALUT & ALM – 59% REG – 15%) | NA                      |
| This work – Anisotropic bilinear interpolation + edge refinement | 162 MHz    | 1920 × 1080 → 3840 × 2160 | 60  | Xilinx Kintex 7 (Slices – 38% + DSP – 12%)     | BRAM 16%                |

## References

- [1] [http://www.itu.int/net/pressoffice/press\\_releases/2011/39.aspx#.Wr4eEi6uxhE](http://www.itu.int/net/pressoffice/press_releases/2011/39.aspx#.Wr4eEi6uxhE)  
Ultra High Definition Television (UHDTV) nearing reality ITU experts create enhanced television viewing experience (2011).
- [2] Qi Shan, et al., Fast image/video upsampling, ACM Trans. Graphics (TOG) 27.5 (2008) 153.
- [3] Oliver Bowen, Christos-Savvas Bouganis, "Real-time image super resolution using an FPGA, Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on, IEEE, 2008.
- [4] MariaE. Angelopoulou, et al., Robust real-time super-resolution on FPGA and an application to video enhancement, ACM Trans. Reconfig. Technol. Syst. 2.4 (2009) 22.
- [5] Yu-Wing Tai, et al., Super resolution using edge prior and single image detail synthesis, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010.
- [6] J.D. Van Herk, Image super-resolution survey, Image Vision Comput. 24.10 (2006) 1039–1052.
- [7] Chao Dong, et al., "Learning a deep convolutional network for image super-resolution, European Conference on Computer Vision, Springer, Cham, 2014.
- [8] Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." arXiv preprint (2016).
- [9] Yuki Sanada, et al., "FPGA implementation of single-image super-resolution based on frame-Bufferless box filtering, J. Signal Process. 17.4 (2013) 111–114.
- [10] Hotaka Kusano, et al., "An FPGA-optimized architecture of anti-aliasing based super resolution for real-time HDTV to 4K-and 8K-UHD conversions, ReConfigurable Computing and FPGAs (ReConFig), 2016 International Conference on, IEEE, 2016.
- [11] GustavoM. Callicó, et al., Low-cost implementation of a super-resolution algorithm for real-time video applications, Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, IEEE, 2005.
- [12] MariaE. Angelopoulou, et al., FPGA-based real-time super-resolution on an adaptive image sensor, International Workshop on Applied Reconfigurable Computing, Springer, Berlin, Heidelberg, 2008.
- [13] Oliver Bowen, Christos-Savvas Bouganis, "Real-time image super resolution using an FPGA, Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on, IEEE, 2008.
- [14] D. Barreto, et al., Real-time super-resolution over raw video sequences, VLSI Circuits and Systems II 5837 International Society for Optics and Photonics, 2005.
- [15] Tomasz Szydzik, GustavoM. Callico, Antonio Nunez, Efficient FPGA implementation of a high-quality super-resolution algorithm with real-time performance, IEEE Trans. Consum. Electron. 57.2 (2011).
- [16] Seiichi Gohshi, "A new signal processing method for video: Reproduce the frequency spectrum exceeding the Nyquist frequency, Proceedings of the 3rd Multimedia Systems Conference, ACM, 2012.
- [17] [http://en.wikipedia.org/wiki/Lanczos\\_resampling](http://en.wikipedia.org/wiki/Lanczos_resampling).
- [18] <http://waifu2x.me/index.en.html>.
- [19] Yuta Kimura, Masayuki Ikebe, "Halo control for LHE based local adaptive tone mapping, Image Processing (ICIP), 2015 IEEE International Conference on, IEEE, 2015.
- [20] Georgios Georgis, George Lentaris, Dionysios Reisis, "Acceleration techniques and evaluation on multi-core CPU, GPU and FPGA for image processing and super-resolution, J. Real-Time Image Process. (2016) 1–28.
- [21] Sebastian Lopez, et al., "A novel real-time DSP-based video super-resolution system, IEEE Trans. Consum. Electron. 55.4 (2009).
- [22] Georgios Georgis, George Lentaris, Dionysios Reisis, "Reduced complexity super-resolution for low-bitrate video compression, IEEE Trans. Circuits Syst. Video Technol. 26.2 (2016) 332–345.
- [23] Mark Sabinia-msabini, -gili Gili Rusak, Example-Based Image Super-Resolution Techniques, (2016).
- [24] Neeraj Kumar, Amit Sethi, "Fast learning-based single image super-resolution, IEEE Trans. Multimedia 18.8 (2016) 1504–1515.



**Prasoon Ambalathankandy** is currently a Ph.D. candidate of the Graduate School of Information Science and Technology, Hokkaido University. He received his B.Tech. in Electronics and Communication Engineering from Cochin University of Science and Technology, India and M.Sc. degree in Electrical Engineering from the University of Calgary, Canada.



**Shinya Takamaeda** received his B.S. and M.S. degrees in physics and Ph.D. degree in electrical engineering in 2011, 2014, respectively. His current research interests are the image processing, Computer Architecture, FPGA System, High-level Synthesis, Machine Learning, Ising Computer.



**Masato Motomura** received his B.S. and M.S. degrees in physics and Ph.D. degree in electrical engineering in 1985, 1987, and in 1996, all from Kyoto University. He was with NEC and NEC Electronics from 1987 to 2011, where he was engaged in the research and business development of dynamically reconfigurable processor and on-chip multi-core processor. Now a professor at Hokkaido University, his current research interests include reconfigurable and parallel architectures and low power circuits. He has won the IEEE JSSC Annual Best Paper Award in 1992, IPSJ Annual Best Paper Award in 1999, and IEICE Achievement Award in 2011, respectively. He is a member of IEICE and IEEE.



**Testsuya Asai** received his B.S. and M.S. degrees in electronic engineering from Tokai University, Japan, in 1993 and 1996, respectively, and his Ph.D. degree from Toyohashi University of Technology, Japan, in 1999. He is now an Associate Professor in the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. His research interests are focused on developing nature-inspired integrated circuits and their computational applications. Current topics that he is involved with include intelligent image sensors that incorporate biological visual systems or cellular automata in the chip, neurochips that implement neural elements (neurons, synapses, etc.) and neuromorphic networks, and reaction-diffusion chips that imitate vital chemical systems.



**Masayuki Ikebe** received his B.S., M.S., and Ph.D. degrees in electrical engineering from Hokkaido University, Sapporo, Japan, in 1995, 1997, and 2000, respectively. During 2000–2004, he worked for the Electronic Device Laboratory, Dai Nippon Printing Corporation, Tokyo, Japan, where he was engaged in the research and development of wireless communication systems and image processing systems. Presently, he is an Associate Professor at Graduate School of Information Science and Technology, Hokkaido University. His current research includes CMOS image sensor and analog circuits. He is a member of IEICE and IEEE.



**Hodaka Kusano** received his B.E. and M.E. degrees in electrical engineering from Hokkaido University, Sapporo, Japan, in 2015 and 2017, respectively. Currently he is with Canon Inc Japan.