JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 34, 51-64 (2018)
DOI: 10.6688/JISE.2018.34.1.4

Hardware Implementation of an Image Interpolation Method with Controllable Sharpness

PEI-YIN CHEN, SHIH-HSIANG LIN AND PO-CHUN CHEN
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, 701 Taiwan
E-mail: pychen@mail.ncku.edu.tw; shlin.hayashi@gmail.com; roy800117@hotmail.com

The technique of image scaling plays a critical role in the digital image processing applications for the coordination between different display devices. In this paper, an improved algorithm is proposed by using a controllable sharpness coefficient to obtain better resulting images under different scaling magnifications. In addition, a power saving module including a condition judgment mechanism is applied to reduce unnecessary power consumption as the scaling ratio increases. Experimental results show that the proposed architecture is still feasible for very large-scale integration implementation and effectively enhances the quality of image scaling. By using TSMC 0.13 μm cell library, the synthesis results show that the circuit can achieve 300 MHz with 12.1k gate counts and cell area is 293×293 μm^2. The total power consumption is 6.75mW.

Keywords: controllable sharpness, image scaling, interpolation, low-power design, VLSI

1. INTRODUCTION

Image scaling is an indispensable technique for amplifying the image size between different display systems with different resolutions to present a clear and accurate image for users [1]. For most practical applications in consumer electronic products, the module executing the image scaling is contained in the user's terminal device. Other than image quality, the algorithm should be capable of real-time processing, have low complexity, and be feasible for very large-scale integrated (VLSI) circuit implementation.

Many scaling algorithms have been proposed. The famous nearest neighbor (NN) [2], and bilinear (BL) [3] interpolation approaches have the advantage of low complexity but the resulting images have aliasing artefacts. The bicubic (BC) method [4, 5] produces superior results with more complex architectures. [6, 7] have proposed algorithms using area pixels instead of traditional point pixels. In [8], a modified BL method named EASE was presented to reduce the blur effect. An isophote-oriented, orientation-adaptive scaling method was presented in [9]. The algorithm proposed in [10] uses edge preserving interpolation (EPI), which is the most suitable image scaling method among [2-10] because it includes most of the aforementioned advantages and features. Recently, some high-quality image scaling methods and their hardware implementation have also been proposed [15, 16]; however, the area cost is extremely high when compared to other algorithms mentioned above.

At some scaling magnifications, however, it does not obtain superior results compared to other algorithms, particularly when the magnifications increase. The other draw-

Received May 4, 2016; revised July 9, 2016; accepted September 16, 2016.
Communicated by Tzung-Pei Hong.

back in the original architecture of EPI is that some identical calculations are performed repeatedly when the image is scaled up; as a result, some unnecessary power consumption is wasted. To enable the obtainment of higher image quality, this work proposes an improved EPI in which a controllable sharpness coefficient is added to the equation. To solve the problem of redundant operations, an effective solution would be proposed to make the data reusable. Because of the rise of environmental consciousness as well as the large power consumption of smart devices, a power-saving technique is designed instead of merely reusing the data. Therefore, not only does the proposed design keep the data for reuse, it also reduces the power consumption of the entire system.

With the equation modification and the additional designs to the hardware architecture, this algorithm retains its low complexity. A 12-stage pipeline architecture was derived to satisfy the real-time processing requirement. Using the Synopsys Design Compiler for synthesis, Synopsys IC Compiler for auto placement and routing, the experiment results show that the proposed circuit could achieve 300 MHz with gate counts of 12.1k and total power consumption of 6.75 mW when the Taiwan Semiconductor Manufacturing Company's (TSMC) 0.13 μm cell library is used. The remainder of the paper is organized as follows. Section 2 describes the EPI and its numerical analysis theory. Section 3 presents the modification of the controllable sharpness coefficient and the power-saving design. Section 4 introduces the VLSI implementation architecture. Section 5 compares the proposed method with other algorithms. Finally, the conclusions are presented in Section 6.

2. EDGE-PRESERVING INTERPOLATION

The EPI [10] algorithm is a low-cost interpolation method with an efficient VLSI architecture. Fig. 1 shows how the algorithm is performed. Assuming that the target pixel is the pixel to be interpolated, four intermediate pixels are calculated in advance by using four vertical interpolations. After the intermediate pixels are obtained, the value of the target pixel is then calculated using horizontal interpolation. The main concept of EPI is the interpolation error theorem [11, 13], which is an equation that defines the deviation of images. $u(x)$ is the multitime continuously differentiable function and $p(x)$ is the linear interpolation function. In the bounded interval $[a, b]$, there exists a discrete set $S = \{(x_i, u(x_i))|x_i \in [a, b], i = 0, 1, \ldots, N\}$. Assuming that $p_N(x)$ is the Nth-order polynomial interpolation function, then for every $x \in [a, b]$, there is a point $\xi x \in [a, b]$ such that

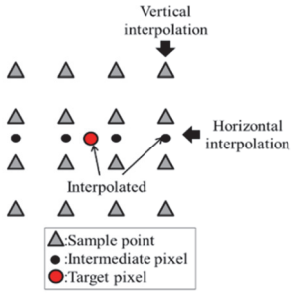


Fig. 1. The example of multiple one-dimensional interpolation.

$$E(x) = u(x) - p_N(x) = \frac{u^{(N+1)}(\xi_x)}{(N+1)!} \prod_{i=0}^{N} (x - x_i). \tag{1}$$

$E(x)$ is the error between interpolation function $p_N(x)$ and continuous function $u(x)$. Let $N = 1$ and h be the interval length of 1 to further derive the linear equation. For $x = x_i + s \times h$, $x \in [x_i, x_i + h]$ where $0 < s < 1$ and $s \in R$, note that

$$u(x_{i+s}) = [(1-s)u(x_i) + su(x_{i+1})] - \frac{u''(\xi x_{i+s})}{2} s(1-s). \tag{2}$$

The linear interpolation error value in Fig. 2 is indicated by the second-derivative term in Eq. (2). Because the error value has been determined, the correction to the linear interpolation makes the result much closer to the continuous function. Because $u(x)$ is a real multitimes continuously differentiable function, $u''(\xi x_{i+s})$ can be estimated by using the interpolation errors of nearest two nodes. As shown is Fig. 3, the linear interpolation error of x_i and x_{i+1} can be represented by E^L and E^R. By using Eq. (3), E^R and $u''(\xi x_{i+1})$ can be obtained by assuming $h = 2$ and using x_i, x_{i+1}, and x_{i+2}. Repeating this method, E^L and $u''(\xi x_i)$ can also be derived by Eq. (4).

$$u(x_{i+1}) - \frac{u(x_i) + u(x_{i+2})}{2} = -\frac{u''(\xi_{x_{i+1}})}{2} = E^R. \tag{3}$$

$$u(x_i) - \frac{u(x_{i-1}) + u(x_{i+1})}{2} = -\frac{u''(\xi_{x_i})}{2} = E^L. \tag{4}$$

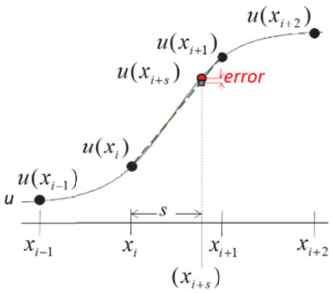


Fig. 2. Example of linear interpolation error.

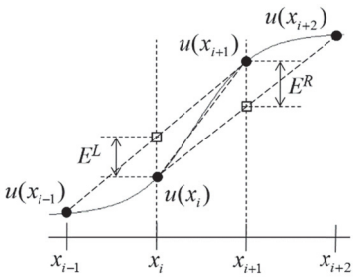


Fig. 3. Using the interpolation errors to estimate $u''(\xi x_{i+s})$.

After obtaining the value of $u''(\xi x_i)$ and $u''(\xi x_{i+1})$, $u''(\xi x_{i+s})$ can be estimated as follows:

$$u''(\xi_{x_i+s}) \approx (1-s)u''(\xi_{x_i}) + su''(\xi_{x_{i+1}}). \tag{5}$$

The bilateral error amenders E^L and E^R can be used to replace the value of $u''(\xi x_i)$ and $u''(\xi x_{i+1})$ in Eq. (5). Using the edge-weighted concept in [12], the EPI equation could be written as

$$\hat{u}(x_{i+s}) = (1-s)u(x_i) + su(x_{i+1}) + \gamma[(1-s)E^L + sE^R]s(1-s). \tag{6}$$

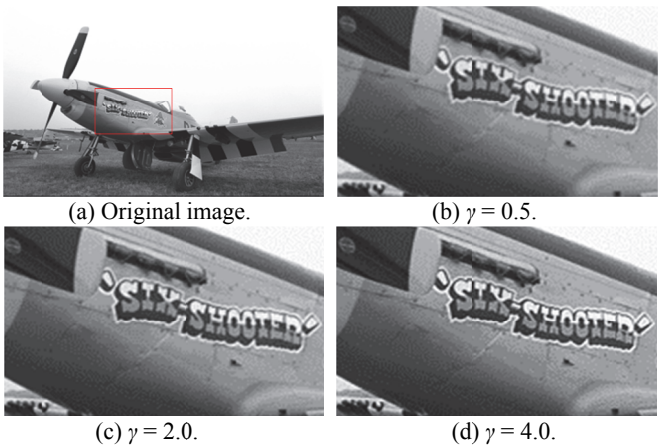
The γ is edge-weighted coefficient, which is set to two (fix value) to balance hardware implementation costs and resulting image quality.

3. THE PROPOSED METHOD

This chapter briefly introduces the modified EPI (MEPI) algorithm, which contains two main modifications: the controllable sharpness coefficient for higher image quality after scaling and the power-saving design for lower power consumption.

3.1 The Controllable Sharpness Coefficient

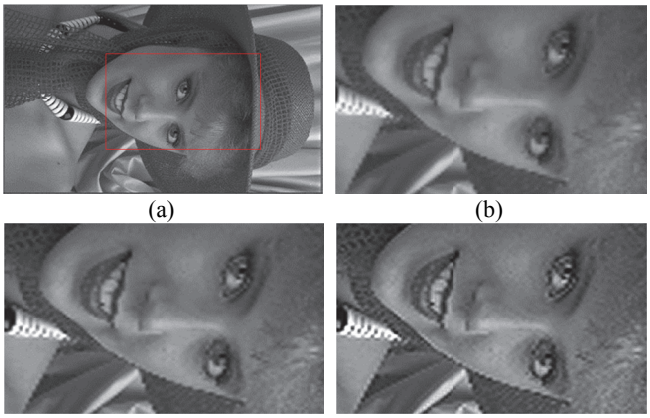
Eq. (6) can be separated into two partitions: linear interpolation and error estimation. Observations show that the value of γ could adjust the sharpness of edges after scaling. Although some results, such as those shown in Fig. 4, suggest that a higher γ value could enhance image quality, the resulting images in Figs. 5 and 6 suggest otherwise. The checkerboard effect (ringing) becomes severe and the quality is reduced when γ is too large. The same problem occurs with higher scaling magnifications. To obtain higher image quality under every magnification, a fixed γ value is insufficient; a controllable coefficient is required for different situations.



(a) Original image. (b) $\gamma = 0.5$.

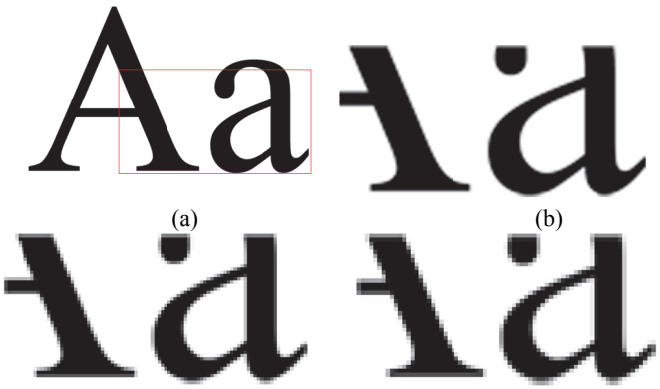
(c) $\gamma = 2.0$. (d) $\gamma = 4.0$.

Fig. 4. The resulting images with different γ as the scaling magnification are 0.5.



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

Fig. 5. The resulting images with different γ as the scaling magnification are 4.0; (a) Original image; (b) $\gamma = 0.5$; (c) $\gamma = 2.0$; (d) $\gamma = 4.0$.



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

Fig. 6. The resulting text images with different γ as the scaling magnification are 4.0; (a) Original image; (b) $\gamma = 0.5$; (c) $\gamma = 2.0$; (d) $\gamma = 4.0$.

A statistical analysis was performed to evaluate the relationship between the scaling magnification and γ value. When the scaling magnification was small, a higher γ yielded higher quality, and vice versa. The fixed γ value in Eq. (6) was replaced with a controllable variable γ', and the equation was rewritten as

$$u(x_{i+s}) = (1-s)u(x_i) + su(x_{i+1}) + \gamma'[(1-s)E^L + sE^R]s(1-s). \tag{7}$$

Because the variable γ' increases the computational complexity, Eq. (7) could be modified to a correlation form of four pixels and their weights by expanding E^R (3) and E^L (4). It could be observed that there exists identical terms in Eq. (8), thus they are rewritten and the computation sequence of each weight is listed in order in Eq. (9). The weights of four pixels $u(x_{i-1})$, $u(x_i)$, $u(x_{i+1})$, and $u(x_{i+2})$ are denoted by W_0, W_1, W_2, and W_3. Since W_3 is the simplest term, it should be calculated in the first priority and W_0 could be derived by using the result of W_3. Following the principle of reusing, W_1 and W_2 could

also be obtained by combining W_0 and W_3. The proposed computation order in Eq. (9) minimizes the resources needed in hardware architecture. Since different γ' values generate a series of different equations, it would be difficult to let γ' be a random value for hardware implementation. As a result, eight types of equations with different γ' values and a multiplexer are proposed, thus the most suitable computation could be selected according to user requirements.

$$\begin{aligned} u(x_{i+s}) = {} & u(x_{i-1}) \times [-\frac{\gamma'(1-s)^2 s}{2}] \\ & + u(x_i) \times [(1-s) + \gamma'(1-s)^2 s - \frac{\gamma'(1-s)s^2}{2}] \\ & + u(x_{i+1}) \times [s + \gamma'(1-s)s^2 - \frac{\gamma'(1-s)^2 s}{2}] \\ & + u(x_{i+2}) \times [-\frac{\gamma'(1-s)s^2}{2}]. \end{aligned} \tag{8}$$

$$\begin{cases} W_3(s) = -[(1-s)\times s]\times s\times\dfrac{\gamma'}{2} \\ W_0(s) = -[(1-s)\times s\times\dfrac{\gamma'}{2} + W_3(s)] \\ W_1(s) = (1-s) - 2W_0(s) + W_3(s) \\ W_2(s) = s - 2W_3(s) + W_0(s). \end{cases} \tag{9}$$

3.2 Power Saving Design

The reuse possibility was discovered under certain conditions, particularly those in which the image was scaled up. The condition exists in Fig. 7, in which the same 16 pixels, denoted by the triangle, are referenced by both the target pixel and previous interpolated pixel. The four intermediate points are clearly identical. In this case, not only can the value of the intermediate points be reused, but the modules performing the vertical interpolations can be turned off to reduce unnecessary power consumption.

The power-saving module is implemented by using a clock-gating technique and judgment controller to determine whether the module should be used. The clock-gating design replaces the original clock signal attached to the target modules with another controlled clock signal and additional logic elements. Eq. (10) states how the controlled clock is determined by the judgment controller.

$$controlled \quad clock = \begin{cases} 0, & \text{if } \lfloor x_{pre} \rfloor = \lfloor x_{cur} \rfloor \\ \text{original clock}, & \text{otherwise.} \end{cases} \tag{10}$$

The x_{pre} and x_{cur} represent the previous and current horizontal coordinates of the target pixel when mapped to the original resolution. Because the controlled clock signal is set to zero or is gated, the module performing the intermediate point computation is not executed. Because the state switching of flip-flops is the main segment that consumes

power, the switching consumption of the target modules decreases to zero when the clock is pruned.

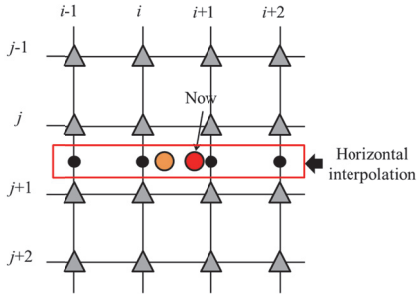


Fig. 7. Condition under which the power-saving module is used.

4. HARDWARE IMPLEMENTATION

The highly efficient VLSI architecture of the proposed MEPI is introduced in this chapter. The system is divided into nine modules, and a 12-stage pipeline architecture is derived to achieve the real-time processing requirement. A block diagram of the proposed architecture is shown in Fig. 8. Some exclusive designs are applied to the submodules for high efficiency hardware implementation. Because some parts of the system are similar to the architectures in [10], only the modifications are introduced. The inputs of the system are the source image, user-defined coefficient γ', and scaling factor which indicates the magnification of the scaling size. Scaling factor from 0.5x to 4.0x are evaluated in this work. The corresponding coordinates of the target pixel to be interpolated are calculated using the coordinate accumulator module. The sequential input of the source image is stored in the line buffer, and the register bank is a reordering module that produces an output of rearranged values for the following vertical interpolation.

The coefficient generator calculates the four weights mentioned in Eq. (9). It is clear that strong data dependency exists in the calculation; hence, minimization of the multiplication operation can be achieved with a hardware resource sharing technique. Because of the data dependency, the module has a four-stage pipeline architecture, as illustrated in Fig. 9. To reduce the computation complexity, the multiplication operations for the γ' coefficient are replaced with a series of adders and shifters. Notably, the power-saving module controls the clock attached to the registers when the results are reusable. When the entire module is turned off, the next module receives identical values. Thus, the power consumption caused by state switching is reduced. After the weights from the coefficient generator are obtained, vertical interpolation for intermediate points is performed by using the convolution interpolator. The four intermediate points are all required for the interpolation of the target pixel; however, with the reuse of the derivative, the horizontal interpolation can be implemented using a set of only three intermediate points. With the use of the target pixel $tf_{n,m}$ and four intermediate points $if_{nj\text{-}1}$, $if_{n,j}$, $if_{n,j+1}$, and $if_{n,j+2}$, the calculation of $if_{n,j+2}$ can be modified as

$$f_{n,j+2} = (w_{1,n} \times f_{i,j+2} + w_{2,n} \times f_{i+1,j+2}) - (w_{0,n} \times f_{i\text{-}1,j+2} + w_{3,n} \times f_{i+2,j+2}). \quad (11)$$

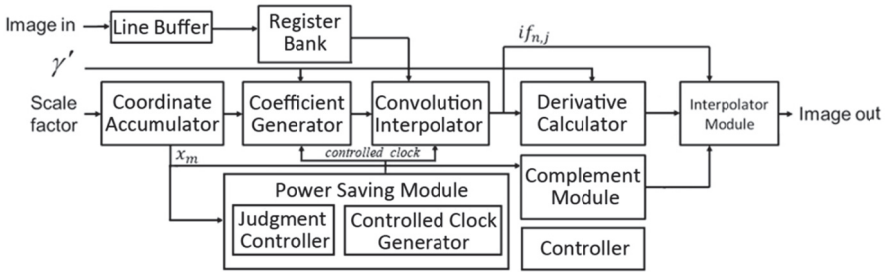


Fig. 8. Block diagram of the proposed architecture.

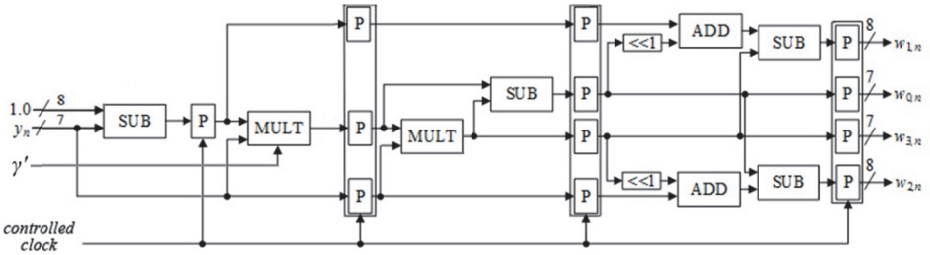


Fig. 9. Architecture of coefficient generator.

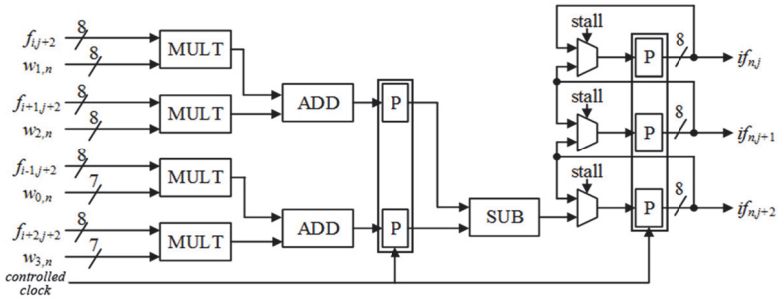


Fig. 10. Architecture of convolution interpolator.

The architecture of the convolution interpolator is illustrated in Fig. 10. There are only three outputs in the module. The calculation of the fourth intermediate point and the final step of the horizontal interpolation are executed by the derivative calculator and interpolator module, respectively. The P blocks in Figs. 9 and 10 indicate the pipeline registers, which are controlled by the power saving signal controlled clock. The gated clock will turn off the registers and the inputs of the remaining modules will be kept. As a result, the unnecessary power consumption is reduced.

5. EXPERIMENT RESULTS

The experiments comprised two parts: determining the most suitable coefficient value under different scaling magnifications, and comparing the proposed method with other algorithms. The test set consisted of 24 Kodak gray level sample images, the reso-

lution are 768×512. Each image was scaled up or down via bilinear interpolation and then restored to its original resolution by using various algorithms. For instance, when 0.5x scaling is evaluated, the images are scaled up to 1536×1024 and restored to original size 768×512 by different algorithms. When 1.5x scaling is evaluated, the images are scaled down to 512×341 using bilinear interpolation and restored to original size 768× 512 by different algorithms. The qualities of the resulting images by different methods were evaluated by comparing to the original image using peak signal-to-noise ratio (PSNR) for objective quantitative testing and structural similarity (SSIM) for subjective visual testing. The PSNR is used to obtain a quantitative measurement and is calculated as Eq. (12), Where MSE represents the mean square error, $M \times N$ is the size of the image, and Y and $\overline{Y}$ are the original and scaled image, respectively. Since the images used are 8-bit per pixel, the term MAX in Eq. (12) is 255. Therefore, the quality is expressed as Eq. (13).

$$PSNR = 10\log_{10}(\frac{MAX^2}{MSE}), MSE = \frac{\sum_i \sum_j (Y(i,j) - \overline{Y}(i,j))^2}{M \times N} \tag{12}$$

$$PSNR(dB) = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{13}$$

In addition to PSNR, the SSIM index is also applied to measure the quality of images scaled by the proposed method. SSIM is designed to improve the traditional objective methods such as PSNR or MSE, which are occasionally inconsistent with human perception. The method is feasible for measuring the similarity between two images and the calculation is given by Eq. (14), where μ denotes the average, σ is the variance, and $\sigma_{Y\overline{Y}}$ is the covariance of two images.

$$SSIM(Y, \overline{Y}) = \frac{(2\mu_Y \mu_{\overline{Y}} + c_1)(2\sigma_{Y\overline{Y}} + c_2)}{(\mu^2{}_Y + \mu^2{}_{\overline{Y}} + c_1)(\sigma^2{}_Y + \sigma^2{}_{\overline{Y}} + c_2)} \tag{14}$$

In the first experiment, the effect of different sharpness coefficients under different magnifications was examined. Table 1 lists the PSNR and SSIM values when different γ' values were used for particular magnifications. For example, for magnifications 0.5 and 4.0, the most suitable γ' values to achieve higher quality after scaling are 4 and 0.5, respectively. It is evident that when the scaling magnification is small, a higher γ' value produces a higher PSNR or SSIM, and vice versa. Fig. 11 is a transformation curve which describes the relationship between the scaling size and the coefficient that generates the best quality of scaled image. The proposed controllable architecture can be considered as a soft intellectual property (soft IP). As a result, users could follow this trend to select a proper coefficient according to their requirements even the scaling size is not one of the evaluated values.

To further evaluate the quality of the MEPI algorithm, the output images of the proposed method were generated using the synthesized gate-level VLSI architecture mentioned in Section 4. The scaled images of other algorithms were generated using C programs. The results show that the proposed design produced a superior quantitative quality compared with the other methods. The proposed MEPI was compared with the BC [4], Win (winscale in [6]), Lwin (Low-cost winscale in [7]), EASE [8], new orienta-

tion adaptive-interpolation (NOAI) [9], and EPI [10]. The values in Tables 2 and 3 are the average of 24 results. Figs. 12 and 13 show the resulting images generated using different algorithms under different scaling magnifications. The VLSI architecture proposed in this paper is implemented using the Verilog hardware description language. For comparison with other scaling methods regarding hardware implementation, the architecture was synthesized using the Synopsys Design Compiler with the TSMC 0.18 μm and TSMC 0.13 μm cell libraries. Auto placement and routing (APR) is performed by Synopsys IC Compiler, and verification including design rule check (DRC) and layout versus schematic (LVS) was evaluated using Mentor Graphic Calibre. The power consumption of proposed method was measured by Synopsys Power Compiler after post-layout simulation.

Table 1. Statistic of PSNR and SSIM for different magnitude and γ' value.

	$\gamma'=0.5$		$\gamma'=1.0$		$\gamma'=2.0$		$\gamma'=4.0$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.5x	39.80	0.998	41.13	0.998	44.51	*0.999	*49.09	*0.999
0.66x	38.01	0.997	39.42	0.997	43.12	*0.999	*47.08	*0.999
0.75x	37.20	0.996	38.53	0.997	41.90	0.998	*45.76	*0.999
1.5x	31.44	0.986	32.02	0.988	*32.80	*0.990	31.63	0.988
2.0x	28.98	0.976	29.22	0.978	*29.28	*0.979	27.62	0.971
2.5x	27.38	0.965	*27.41	*0.966	27.10	0.965	25.29	0.952
3.0x	*26.14	0.954	26.09	*0.955	25.68	0.953	24.01	0.936
4.0x	*24.57	*0.936	24.44	*0.936	23.93	0.931	22.35	0.907

* is the best value

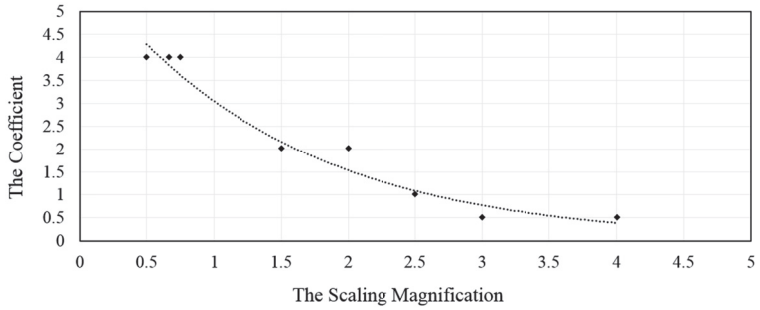


Fig. 11. The relationship between the scaling magnification and coefficient, and the curve indicates the trend.

Table 2. PSNR for different algorithms under different magnifications.

	NN [2]	BL [3]	BC [4]	WIN [6]	Lwin [7]	EASE [8]	NOAI [9]	EPI [10]	MEPI
0.5x	37.67	38.25	40.43	35.48	39.10	40.14	43.07	44.51	*49.09
0.66x	36.47	36.52	38.90	35.86	38.22	38.04	40.98	43.12	*48.10
0.75x	34.94	35.81	38.10	35.53	36.87	37.19	40.04	41.90	*46.03
1.5x	29.73	30.73	31.20	30.43	31.64	31.15	32.12	*32.80	*32.80
2.0x	26.99	28.59	28.24	26.99	28.55	28.81	29.12	*29.28	*29.28
2.5x	25.55	27.20	26.57	25.88	26.85	27.31	27.12	27.10	*27.41
3.0x	24.66	26.06	24.95	24.19	25.19	26.12	25.78	25.68	*26.14
4.0x	23.06	24.59	23.01	22.41	23.91	*24.61	24.11	23.93	24.57

* is the best value

The results show that the gate count of the proposed design was 12.5k, cell area is 398×398 μm^2, the clock period was 3.8 ns, and the operating frequency was 263 MHz when the TSMC 0.18μm cell library was applied. For the TSMC 0.13μm cell library, the gate count of the proposed design was 12.1k, cell area is 293×293μm^2, the clock period was 3.3ns, and the operating frequency achieved 300 MHz. The hardware implementation results of the different algorithms are listed in Table 4. The proposed method is compared to the original EPI [10], hardware-based BC (EVDB [14]), and the low-cost version of Winscale (Lwin [7]). Since EASE [8] and NOAI [9] are software-based algorithm, they are not listed for the hardware comparison. The proposed method is focused on features of low-power and low-cost that is practical for VLSI architecture. For the method proposed by Liu *et al.* [15], the synthesis results by TSMC 65-nm technology show that the maximal clock frequency is 240MHz, cell area is 1102875μm^2, and gate count is 765k. Chen *et al.* [16] proposed a VLSI architecture for 4K image upscaling and the method achieves the throughput as high as 497Mpixel/s. However, the area cost is extremely high. The synthesis results show that the gate count is 504k. Although both of the works produce high quality scaled images, the costs are extremely higher than our method (gate count is 13k for 0.13μm cell library). The resource in terms of gate count of our work is only 1.7% of the method proposed by Liu *et al.* and 2.6% of the method proposed by Chen *et al.* When the system worked at the maximum frequency and the power-saving module was activated, the consumption was 13.63 mW with a 1.8 V supply voltage when the TSMC 0.18μm cell library was applied. For the TSMC 0.13μm technology and the same conditions, the consumption was 6.75 mW with a 1.32 V supply voltage. The power consumption was reduced by 7%-16% compared with that of [10]. Thus, the system has a superior performance in every aspect, including area cost, operating frequency, and power consumption.

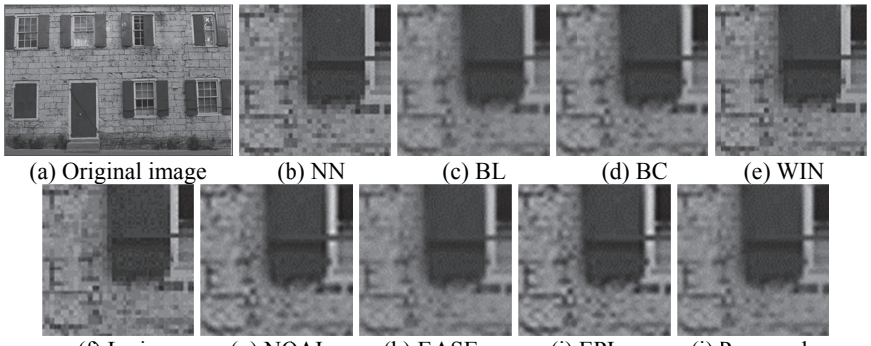
6. CONCLUSIONS

In this paper, an improved edge-preserving algorithm, called the MEPI, is proposed. With the provision of a controllable γ' value, higher image quality can be obtained under different scaling magnifications. To achieve the goal of a low-power design, a power-saving module that included a clock-gating technique and judgment controller was applied. Although a significant modification was implemented, the proposed VLSI architecture retained low computation complexity, and experiments showed that the system

Table 3. SSIM for different algorithms under different magnifications.

	NN[2]	BL[3]	BC[4]	WIN[6]	Lwin[7]	EASE[8]	NOAI[9]	EPI[10]	MEPI
0.5x	0.997	0.997	0.998	0.994	0.994	0.998	*0.999	*0.999	*0.999
0.66x	0.995	0.996	0.997	0.995	0.994	0.997	0.998	*0.999	*0.999
0.75x	0.994	0.995	0.997	0.994	0.993	0.996	0.998	0.998	*0.999
1.5x	0.979	0.983	0.986	0.984	0.986	0.985	0.988	*0.990	*0.990
2.0x	0.966	0.974	0.974	0.966	0.961	0.975	0.978	*0.979	*0.979
2.5x	0.952	0.963	0.962	0.956	0.952	0.964	0.964	0.965	*0.966
3.0x	0.941	0.953	0.947	0.940	0.949	0.952	0.953	0.953	*0.954
4.0x	0.919	0.936	0.920	0.909	0.931	*0.936	0.932	0.931	*0.936

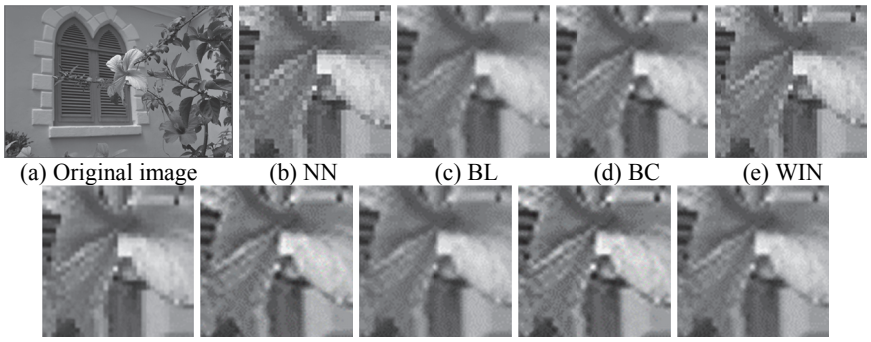
* is the best value



(a) Original image (b) NN (c) BL (d) BC (e) WIN

(f) Lwin (g) NOAI (h) EASE (i) EPI (j) Proposed

Fig. 12. Resulting images of different scaling techniques when scaling magnification is 0.5 for Kodak 1.



(a) Original image (b) NN (c) BL (d) BC (e) WIN

(f) Lwin (g) NOAI (h) EASE (i) EPI (j) Proposed

Fig. 13. Resulting images of different scaling techniques when scaling magnification is 3.0 for Kodak 7.

Table 4. Hardware implementation results of proposed method and other designs.

	TSMC Cell Library	Gate Counts (k)	Max Frequency (MHz)	Total Cell Area (μm^2)	Total Power (mW)
Lwin [7]	0.18μm	10.4	200	389×389	16.47
EVDB [14]	0.13μm	30.6	279	498×498	19.17
EPI [10]	0.18μm	12.9	200	405×405	16.80
	0.13μm	13.0	278	295×295	11.69
Proposed (*1)	0.18μm	12.3	200	392×392	14.78
	0.13μm	12.0	278	291×291	7.19
Proposed (*2)	0.18μm	12.3	200	392×392	12.45
	0.13μm	12.0	278	291×291	6.52
Proposed (*3)	0.18μm	12.5	263	398×398	16.02
	0.13μm	12.1	300	293×293	7.81
Proposed (*4)	0.18μm	12.5	263	398×398	13.63
	0.13μm	12.1	300	293×293	6.75

*1: The system worked at the same frequency as [10] and power saving module is not activated.
*2: The system worked at the same frequency as [10] and power saving module is activated.
*3: The system worked at the maximum frequency and power saving module is not activated.
*4: The system worked at the maximum frequency and power saving module is activated.

exhibited superior performance compared with existing designs. Most important, the scaled images obtained using the proposed method exhibited higher quality than those generated by other algorithms in both objective (PSNR) and subjective (SSIM) measurements. The proposed method still has the feature of low-cost and the frequency of the power-saving module increased with the scaling ratio. Thus, the proposed algorithm is suitable for high-magnification image scaling, and the 300 megapixels/s processing rate is sufficient for applications such as 4K ultra high definition video (3840×2160, 30 fps) real-time amplification.

REFERENCES

1. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed., Prentice-Hall, NJ, 2002.
2. S. Fifman, "Digital rectification of ERTS multispectral imagery," in *Proceedings of Significant Results Obtained from Earth Resources Technology Satellite*, Vol. 1, 1973, pp. 1131-1142.
3. J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of interpolation method for image resampling," *IEEE Transactions on Medical Imaging*, Vol. MI-2, 1983, pp. 31-39.
4. H. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics*, *Speech*, *and Signal Processing*, Vol. ASSP-26, 1978, pp. 508-517.
5. R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics*, *Speech*, *and Signal Processing*, Vol. ASSP-29, 1981, pp. 1153-1160.
6. C. H. Kim, S. M. Seong, J. A. Lee, and L. S. Kim, "Winscale: An image-scaling algorithm using an area pixel model," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, 2003, pp. 549-553.
7. P. Y. Chen, C. Y. Lien, and C. P. Lu, "VLSI Implementation of an edge-oriented image scaling processor," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 17, 2009, pp. 1275-1284.
8. Y. Cha and S. Kim, "The error-amended sharp edge (EASE) scheme for image zooming," *IEEE Transactions on Image Processing*, Vol. 16, 2007, pp. 1496-1505.
9. Q. Wang and R. K. Ward, "A new orientation-adaptive interpolation method," *IEEE Transactions on Image Processing*, Vol. 16, 2007, pp. 889-900.
10. C. C. Hang, P. Y. Chen, and C. H. Ma, "A novel interpolation chip for real time multimedia applications," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, 2012, pp. 1512-1525.
11. J. F. Epperson, *An Introduction to Numerical Methods and Analysis*, Wiley, NY, 2007.
12. H. Kim, S. Jin, S. Yang, and J. Jeong, "Enhanced edge-weighted image interpolation algorithm," in *Proceedings of the 8th International Conference on Hybrid Intelligent Systems*, 2008, pp. 957-958.
13. S. Thurnhofer and S. Mitra, "Edge-enhanced image zooming," *Optical Engineering*, Vol. 35, 1996, pp. 1862-1870.
14. C. C. Lin, M. H. Sheu, H. K. Chiang, C. L. Liaw, and Z. C. Wu, "The efficient VLSI design of BI-CUBIC convolution interpolation for digital image processing," in *Pro-*

ceedings of IEEE International Conference on Circuits Systems, 2008, pp. 480-483.
15. Y. N. Liu, Y. C. Lin, Y. L. Huang, and S. Y. Chien, "Algorithm and architecture design of high-quality video upscaling using database-free texture synthesis," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 24, 2014, pp. 1221-1234.
16. Q. Chen, H. Sun, X. Zhang, H. Tao, J. Yang, J. Zhao, and N. Zheng, "Algorithm and VLSI architecture of edge-directed image upscaling for 4k display system" *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 26, 2016, pp. 1758-1771.

Pei-Yin Chen (陳培殷) received the B.S. degree from National Cheng Kung University, Tainan, Taiwan, in 1986, the M.S. degree from Pennsylvania State University, University Park, in 1990, and the Ph.D. degree from National Cheng Kung University, in 1999, all in Electrical Engineering. He is currently a Professor with the Department of Computer Science and Information. His research interests include very large scale integration chip design, video compression, fuzzy logic control, and gray prediction.

Shih-Hsiang Lin (林世祥) received the B.S. degree in Computer Science and Information Engineering from National Cheng Kung University, Tainan, Taiwan, in 2014, where he is currently pursuing the Ph.D. degree in Computer Science and Information Engineering. His current research interests include image processing, vert large-scale integrated chip design, and embedded systems.

Po-Chun Chen (陳柏均) received the B.S. degree in Computer Science and Information Engineering from National Sun Yat-sen University, Kaohsiung, Taiwan in 2013, and the M.S. degree in Computer Science and Information Engineering from National Cheng Kung University, Tainan, Taiwan in 2015. He is now with NovaTeck Inc., Hsinchu, Taiwan. His research interests include very large scale integration chip design, and embedded systems.