ISSN: 2321-7782 (Online)
Volume 2, Issue 2, February 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

An Efficient Low cost Image scaling Technique for less power Consumption

R. Sundar
PG Scholar
Department of ECE
Kumaraguru College of Technology
Coimbatore – India

Abstract: ***High performance systems along with low power consumption are mandatory in any systems to be efficient. In this paper, a high performance algorithm is proposed for an image scaling processor which is less complexity and needs reduced memory. The image scaling algorithm is implemented by using spatial filter, clamp filter and a bilinear interpolation. The aliasing artefacts resulted by the bilinear interpolation is reduced by using spatial and clamp filters. The design complexity is reduced by using a T-model and inversed T-model convolution kernels. Moreover, the combined filter is replaced by a dynamic estimation unit to reduce the hardware cost. This architecture is designed with a SPST adder which reduces the power consumption by considerable amount. It produces 320MHz with 6.08-K gate counts. It is implemented using Xilinx 8.1. Compared with the existing methodologies, this method shows better performance with respect to cost and complexity.***

Keywords: Image scaling; Bilinear interpolation; T-model, spatial filter; clamp filter; SPST.

I. INTRODUCTION

Image scaling has been broadly applied in the fields of digital imaging and mostly on Electronic based Imaging devices. Image scaling is defined as the process of scaling down the high quality frames and pictures so as to fit small size LCD panel of electronic displays. Scaling algorithm can be categorized into two types: polynomial-based and non-polynomial-based. Nearest neighbor algorithm is one of the uncomplicated and simplest polynomial algorithm till date. One of the major drawback of nearest neighbor is the resultant images are with full of aliasing artifacts. Bilinear interpolation algorithm [15] and Bi-cubic algorithm [14] are the other major polynomial based methods which are widely used to acquire the target pixels. In recent years many non-polynomial high performance methods have been proposed [1-3]. These methods use effective techniques like bilateral filter [2], interpolation [1], and autoregressive model [3]. The latter methods enhance the image quality by diminishing the artifacts, aliasing and blurring effects. But these Image scaling algorithms are very complex to implement in VLSI due to high complexity and large memory requirement. Thus, for fast, practical applications, low complex scaling algorithms are required [5-9]. Area pixel model winscale method is previously proposed for less complex implementation. The addition of sharpening spatial and clamp filter improves the image quality along with bilinear interpolation algorithm. SPST is used for more efficient working as it reduces the area in addition to power. Hence the cost of the hardware is reduced much with low memory requirement.

II. PROPOSED SCALING ALGORITHM

Fig.1 shows the block diagram of the proposed scaling algorithm for zooming the images. The sharpening spatial and clamp filters [5] act as the pre- filters [4]. They help in reducing the blurring and aliasing artifacts produced by the bilinear interpolation algorithm. Spatial filter filters the pixels of the original input image to remove the noise and also enhances the edges. Clamp filter removes the unwanted discontinuous edges and boundaries. To save the computational resource and

memory buffer, the spatial and clamp filters are simplified and combined into a single combined filter which provides the functionalities of the two filters.

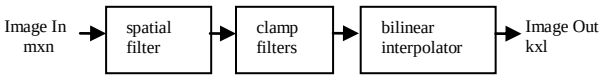

Fig.1 The block diagram of the proposed scaling algorithm

A. Less -Complexity Sharpening Spatial and Clamp Filters

The sharpening spatial filter is a kind of high-pass filter which is used to diminish the blurring artifacts. The filter is a kernel that is used to increase the intensity of a center pixel relative to its neighboring pixels. The clamp filter is a low pass filter. It is a 2-D Gaussian spatial domain filter and is composed of a convolution kernel array. It contains a single positive value at the center and it is fully surrounded by ones. The clamp filter reduces aliasing artifacts and smoothes the unwanted discontinuous edges of the boundary areas.

The sharpening spatial and clamp filters is denoted by convolution kernels. The size of the convolution kernel increases thereby the quality of images will be higher. However, a larger size of convolution filter will demand more memory and therefore will also increase the hardware cost. For example, a 6 × 6 convolution filter needs at least a five-line-buffer memory and 36 arithmetic units, which is much higher than the two-line-buffer memory and nine arithmetic units of a 3 × 3 convolution filter. In the previous works, each of the sharpening spatial and clamp filters was realized by a 2-D 3 × 3 convolution kernel as shown in Fig. 2(a). It needs at least a four-line-buffer memory for two 3 × 3 convolution filters. To reduce the complexity of the 3 × 3 convolution kernel, a cross-model form is used to replace the 3 × 3 convolution kernel as shown in Fig. 2(b). It cuts down on four of nine parameters in the 3 × 3 convolution kernel. To decrease the complexity and memory requirement of the cross model convolution kernel further, T-model and inversed T-model convolution kernels are proposed for implementing the sharpening spatial and clamp filters.

The T-model convolution kernel is composed of the lower four parameters of the cross-model, whereas the inversed T-model convolution kernel is composed of the upper four parameters. Both the T-model and inversed T-model filters are used to enhance the image quality in the proposed scaling algorithm, The T-model or inversed T-model filter is simplified from the 3 × 3 convolution filter which reduces the complexity of the convolution filter and decreases the memory requirement from two to one line buffer for each convolution filter. The T-model and the inversed T-model provide the low-complexity and low memory-requirement convolution kernels for the sharpening spatial and clamp filters to integrate the VLSI chip of the proposed low-cost image scaling processor.

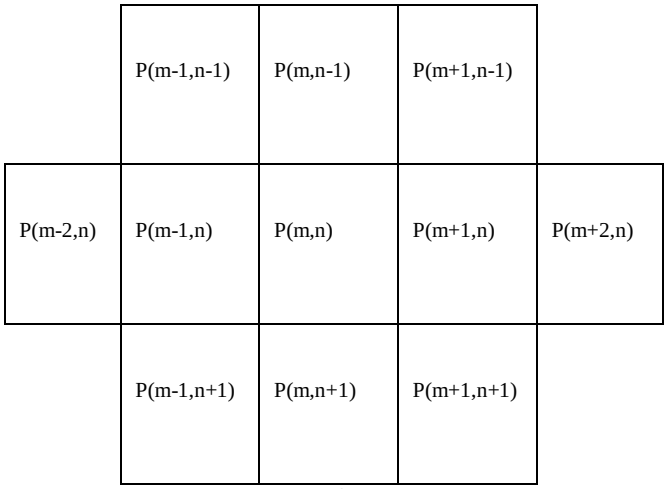

Fig. 2 (a) Cross Weights of 3 x 3 convolution kernel

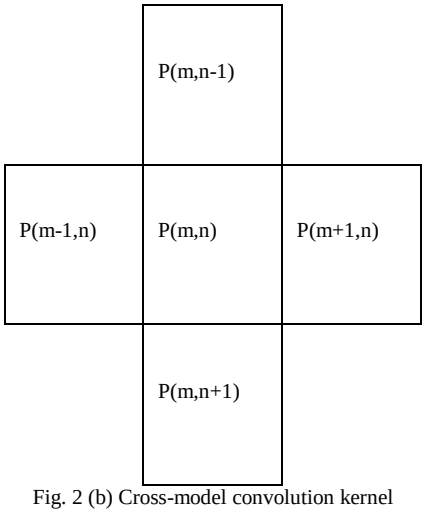

Fig. 2 (b) Cross-model convolution kernel

Combined Filter

In the proposed scaling algorithm, the input image is first filtered using a sharpening spatial filter followed by filtered with clamp spatial filter. Although the sharpening spatial and clamp filters are simplified by T-models and inversed T-models, it still requires two line buffers for storing either the input data or intermediate values for each T-model or inversed T-model filter. Thus, to be able to decrease the computing resource and memory requirement, the sharpening spatial and clamp filters, which are formed by the T-model or inversed T-model, are combined together into a combined filter as,

$$P'_{(m,n)} = \left[\frac{P^*_{(m,n)}\begin{bmatrix} -1 & S & -1 \\ & -1 & \end{bmatrix}}{(S-3)}\right] * \frac{\begin{bmatrix} 1 & C & 1 \\ & 1 & \end{bmatrix}}{(C+3)}$$

$$= P^*_{(m,n)}\begin{bmatrix} -1 & S-C & SC-2 & S-C-1 \\ & -2 & S-C & -2 \\ & & -1 & \end{bmatrix}[(S-3)X(C+3)] \qquad (1)$$

$$\cong \frac{P^*_{(m,n)}\begin{bmatrix} -1 & S-C & SC-2 & S-C-2 \\ & -2 & S-C & -2 \end{bmatrix}}{[(S-3)X(C+3)]} \qquad (2)$$

where S and C are the sharp and clamp filter parameters and P'(m,n) is the filtered result from target pixel P(m,n) by the combined filter. The separate T-model sharpening spatial filter and a T-model clamp filter have been replaced by the combined T-model filter as shown in (1). To reduce the one-line-buffer memory, the only parameter in the third line, parameter −1 of P(m,n−2), is removed, and the weight of parameter −1 is added into the parameter S-C of P(m,n−1) by S-C-1 as shown in equation (2). The combined inversed T-model filter can be obtained in the similar way. In the new architecture of the combined filter, the two T-model or inversed T-model filters are combined into one combined T-model or inversed T-model filter. By this filter-combination technique, memory requirement can be decreased from two to one line buffer, which diminishes memory access requirements for software systems and the hardware memory costs for VLSI implementation.

B. Bilinear interpolation

In the proposed scaling algorithm, the bilinear interpolation method is selected because of its characteristics of high quality and low complexity. The bilinear interpolation is an operation that first performs a linear interpolation in one direction and, again, in the other direction. The output pixel P(k,l) can be calculated by the operations of the linear interpolation in both x- and y-directions with the four nearest neighbour pixels. We can easily find that the computing resources of the bilinear interpolation cost eight multiply, four subtract, and three addition operations. It costs a considerable chip area to implement a bilinear

interpolator with eight multipliers and seven adders. Thus, an algebraic manipulation has been used to reduce the computing resources of the bilinear interpolation.

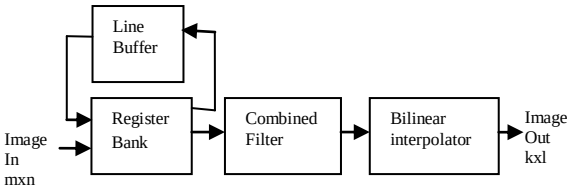

Fig. 3 Block diagram of the VLSI architecture for proposed real-time image scaling processor

The original equation of bilinear interpolation is presented and the simplifying procedures of bilinear interpolation can be described. As the function of dy × (P(m,n+1) − P(m,n)) + P(m,n) appears twice in, one of the two calculations for this algebraic function can be reduced by the characteristic of the executing direction in bilinear interpolation [15], the values of dy for all pixels that are selected on the vertical axis of n row equal to n + 1 row, and only the values of dx must be changed with the position of x. The result of the function "[P(m,n) + dy × (P(m,n+1) − P(m,n))]" can be replaced by the result of "[P(m+1,n) + dy × (P(m+1,n+1) − P(m+1i,n))]" as shown in (6). The simplifying procedures successfully reduce the computing resource from eight multiply, four subtract, and three add operations to two multiply, two subtract, and two add operations.

III. VLSI ARCHITECTURE

The scaling algorithm proposed in this paper consists of two combined pre-filters: sharpening spatial filter and clamp filter and one simplified bilinear interpolator. For VLSI implementation, the bilinear interpolator can directly obtain two input pixels from two combined pre filters without any line buffer memory.

A. Register Bank

The combined filter is filtering to produce the target pixels of P_(m,n) and P_(m,n+1) by using ten source pixels. The register bank is designed with a one-line memory buffer. It is used to provide the ten values for the immediate usage of the combined filter. Fig. 4 shows the architecture of the register bank with the structure of ten shift registers.

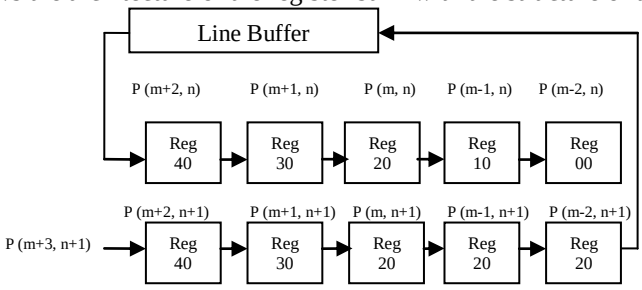

Fig.4. Architecture of the register bank

When the shifting control signal is produced from the controller, a new value of P(m+3,n) will be read into Reg41, and each value stored in other registers belonging to row n + 1 will be shifted right into the next register or line-buffer memory. The Reg40 reads a new value of P(m+2,n) and from the line-buffer memory, and each value in other registers belonging to row n will be shifted right to the next register.

B. Combined Filter

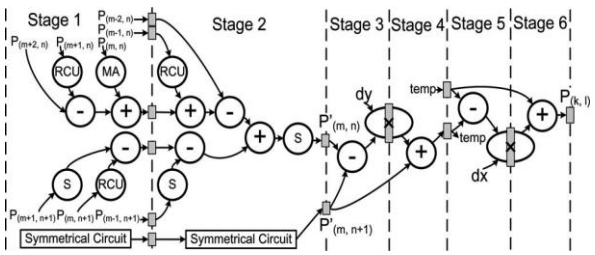

Fig.5. Computational scheduling of the proposed combined filter and simplified bilinear interpolator

Fig. 5 shows the six-stage pipelined architecture of the combined filter and bilinear interpolator. It minimizes path delay to improve the performance by using the pipeline technology. The stages 1 and 2 in Fig. 5 shows the computational scheduling of a T model combined and an inverse T-model filter. The T-model or inversed T-model filter has three reconfigurable calculation units (RCUs), one multiplier–adder (MA), three adders (+), three subtracters (−), and three shifters (S). The hardware architecture of the T-model combined filter can be directly mapped with convolution equation shown in (1). The values of the ten source pixels can be obtained from the register bank.

The symmetrical circuit, as shown in stages 1 and 2 of Fig.5, is the inversed T-model combined filter designed for producing the filtered result of p_ (m,n+1). The T-model and the inversed T-model are used to obtain the values of p_ (m,n) and p_ (m,n + 1) simultaneously. The architecture of the symmetrical circuit is a similar symmetrical structure of the T-model combined filter, as shown in stages 1 and 2 of Fig. 5. Both of the combined filter and symmetrical circuit consist of one MA and three RCUs. The MA can be implemented by using a multiplier and an Adder. The RCU is designed for producing the calculation functions of (S-C) and (S-C-1) times of the source pixel value, which is implemented with C and S parameters. The C and S parameters can be set by users according to image characteristics.

Table I lists the parameters and computing resource for the RCU. With the selected C and S values listed in Table I, the gain of the clamp or sharp convolution function is {8, 16, 32} or {4, 8, 16}, which can be eliminated by a shifter rather than a divider. Fig. 6 shows the architecture of the RCU. It consists of four shifters, three multiplexers (MUX), three SPST adders, and one sign circuit. This RCU design greatly reduces the hardware cost of the combined filters when compared with other techniques

Table 1
Parameters and computing resource for RCU

Parameters	**Values**	**Computing resource**
C	5,13,29	Add and Shift
S	7,11,19	Add and Shift
S-C	2,-6,-22,6,-2,-18,14,6,-10	Add ,Shift and sign
S-C-1	1,-7,-23,5,-3, -19,13,5,-11	Add ,Shift and sign

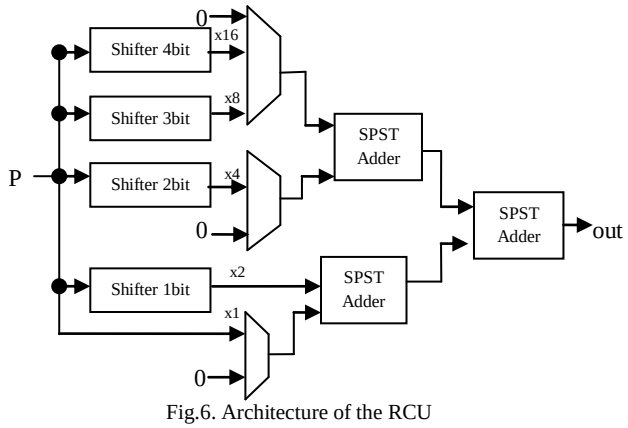

Fig.6. Architecture of the RCU

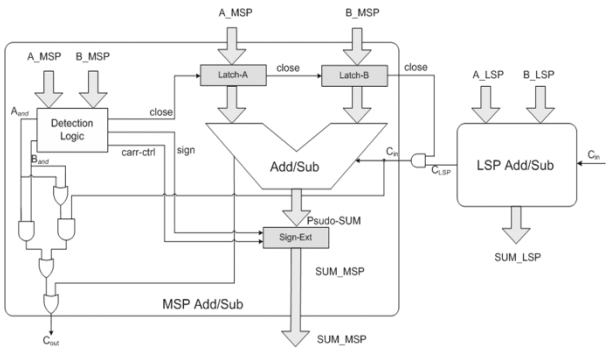

Fig.7. Low-power adder/subtracter design example adopting the proposed SPST.

C. Bilinear Interpolator and Controller

In the previous discussion, the bilinear interpolation is simplified as shown in (5). The stages 3, 4, 5, and 6 in Fig. 4 show the four-stage pipelined architecture, and two-stage pipelined multipliers are used to reduce the delay path of the bilinear interpolator. The input values of P_ (m,n) and P_ (m,n+1) are obtained from the combined filter and the symmetrical circuit. By using the the hardware sharing technique, as shown in (4), the temperature result of the function "P_ (m,n) + dy × (P_ (m,n+1) − P_ (m,n))" can be replaced by the previous result of "P_(m+1,n) + dy × (P_(m+1,n+1)− P_(m+1i,n))." It also means that one multiplier and two adders can be successfully reduced by adding one register. The controller is implemented with finite-state machine circuit. It generates control signals to control the timing and pipeline stages of the register bank, combined filter, and the bilinear interpolator thus reducing the power consumption.

IV. RESULT

Xilinx ISE8.1 is used for synthesis and implementation of a design. In order to evaluate performance of the proposed scheme first pixel value was calculated using MATLAB tool. From that a binary equivalent of a pixel is obtained, for FPGA implementation. FPGA is implemented in SPARTAN 3E.

V. CONCLUSION

In this work a low complexity, a low-cost, low-memory-requirement, high-performance and high quality VLSI architecture of the image scaling processor had been proposed. The spatial and clamp filter combining, sharing of hardware and reconfigurable techniques had been used to reduce the cost of hardware. Relative to the previous low-complexity VLSI scalar designs, this work achieves at least 34.5% reduction in gate counts and requires only one-line memory buffer. Usage of SPST adders reduces the power consumption to a great extent.

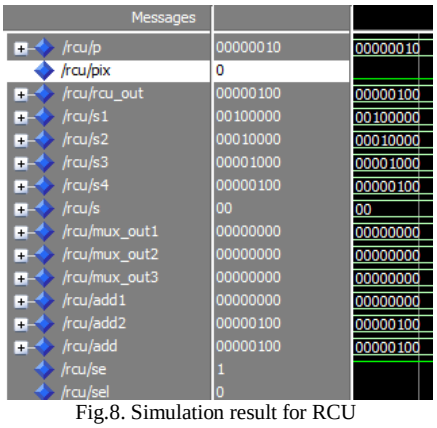

Fig.8. Simulation result for RCU

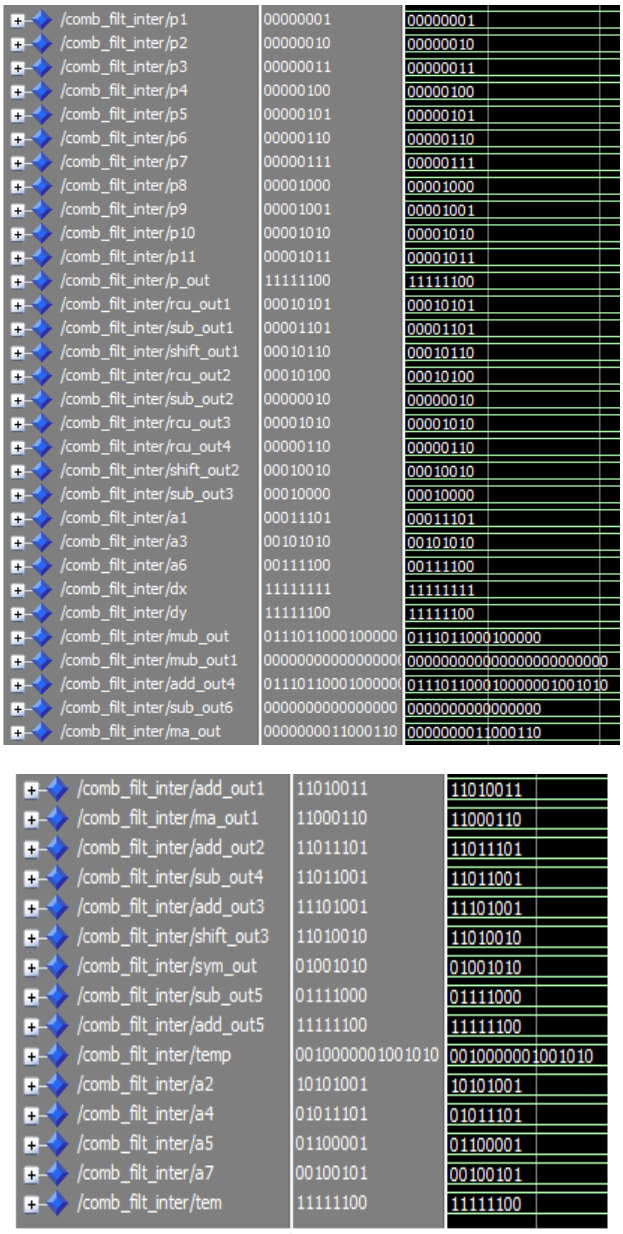

Fig.9. Simulation result for Computational scheduling

References

1. H. Kim, Y. Cha, and S. Kim, "Curvature interpolation method for image zooming," IEEE Trans. Image Process., vol. 20, no. 7, pp. 1895–1903, Jul. 2011.

2. J. W. Han, J. H. Kim, S. H. Cheon, J.O.Kim, and S. J. Ko, "Anovel image interpolation method using the bilateral filter," IEEE Trans. Consum.Electron., vol. 56, no. 1, pp. 175–181, Feb. 2010.

3. X. Zhang and X.Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," IEEE Trans. Image Process., vol. 17, no. 6, pp. 887–896, Jun. 2008.

4. F. Cardells-Tormo and J. Arnabat-Benedicto, "Flexible hardware-friendly digital architecture for 2-D separable convolution-based scaling," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 7, pp. 522–526, Jul. 2006.

5. S. Ridella, S. Rovetta, and R. Zunino, "IAVQ-interval-arithmetic vector quantization for image compression," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 47, no. 12, pp. 1378–1390, Dec. 2000.

6. S. Saponara, L. Fanucci, S. Marsi, G. Ramponi, D. Kammler, and E. M. Witte, "Application-specific instruction-set processor for Retinexlink image and video processing," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 7, pp. 596–600, Jul. 2007.

7. P. Y. Chen, C. C. Huang, Y. H. Shiau, and Y. T. Chen, "A VLSI implementation of barrel distortion correction for wide-angle camera images," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 1, pp. 51–55, Jan. 2009.

8. M. Fons, F. Fons, and E. Canto, "Fingerprint image processing acceleration through run-time reconfigurable hardware," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 12, pp. 991–995, Dec. 2010.

9. C. H. Kim, S. M. Seong, J. A. Lee, and L. S. Kim, "Winscale : An imagescaling algorithm using an area pixel model," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 6, pp. 549–553, Jun. 2003.

10. C. C. Lin, Z. C. Wu, W. K. Tsai, M. H. Sheu, and H. K. Chiang, "The VLSI design of winscale for digital image scaling," in Proc. IEEEInt. Conf. Intell. Inf. Hiding Multimedia Signal Process., Nov. 2007, pp. 511–514.

11. P. Y. Chen, C. Y. Lien, and C. P. Lu, "VLSI implementation of an edgeoriented image scaling processor," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 17, no. 9, pp. 1275–1284, Sep. 2009.

12. C. C. Lin, M. H. Sheu, H. K. Chiang, W. K. Tsai, and Z. C. Wu, "Real-time FPGA architecture of extended linear convolution for digital image scaling," in Proc. IEEE Int. Conf. Field-Program. Technol., 2008, pp. 381–384.

13.] C. C. Lin, M. H. Sheu, H. K. Chiang, C. Liaw, and Z. C. Wu, "The efficient VLSI design of BI-CUBIC convolution interpolation for digital image processing," in Proc. IEEE Int Conf. Circuits Syst., May 2008, pp. 480–483.

14. S. L. Chen, H. Y. Huang, and C. H. Luo, "A low-cost high-quality adaptive scalar for real-time multimedia applications," IEEE Trans. Circuits Syst.Video Technol., vol. 21, no. 11, pp. 1600–1611, Nov. 2011.

15. K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," IEEE Trans. Image Process., vol. 4, no. 3, pp. 285–295, Mar. 1995

