IMPLEMENTATION OF LOW MEMORY REQUIREMENT AND LOW COMPLEXITY IMAGE SCALING PROCESSOR

Vijaya Dharshana.S[1]

[1]*PG Scholar,M.E VLSI design,KPRInstituteofEngineeringandTechnology,TamilNadu, India.*

***Abstract*-Implementation of low memory requirement and low complexity image scaling processor is done through the image scaling algorithm which is a widely used algorithmcalled bilinear interpolation algorithm. The proposed image scaling algorithm consists of a combined filter which holds both the sharpening and clamp filter which acts as prefilters to minimize the memory buffer consumption and the computing resources a T-model and inversed T-model convolution kernels are created to realize the sharpening and clamp filters. Moreover, a reconfigurable calculation unit is invented for decreasing the hardware cost of the combined filter. The VLSI architecture in this work can achieve maximum Frequency of 97.847MHzwith 1.13-K gate counts,and a throughput of782.776 M(pixels/s) and its core area. Compared with previous low-complexity techniques, this work reduces gate counts by more than 34.4% and requires only a one-line-buffer memory.**

Index Terms—Bilinear, clamp filter, image zooming, reconfigurable calculation unit (RCU), sharpening spatial filter, VLSI.

1. INTRODUCTION

Image scaling is a significant process that involves the tradeoff between efficiency, smoothness and sharpness. It is the process of resizing a digital image, resizing may be either up scaling or downscaling according to the requirement of the user. An image can be scaled up to make it detailed or it can be scaled down to fit in small area or frame. Image scaling has been widely exploited in the fields of digital imaging devices such as , digital video recorders,digital cameras, high definition television, mobile phones, digital photo frames,Tablet pc, etc. There are many image scaling algorithms such as nearest neighborhood algorithm, bilinear interpolation, etc.

In recent years, many high-quality non-polynomial-based methods have been proposed. These algorithms improve the image quality, such as curvature interpolation bilateral filter, and autoregressive model. The methods mentioned earlier efficiently enhance the image quality as well as reduce the artifacts such as jagedness, blurring, blocking and alaising. However, these high-quality image scaling algorithms have the characteristics of high complexity and high memory requirement, which is not easy to be realized by VLSI technique. Thus, for real-time applications, low-complexity image processing algorithms are necessary for VLSI implementation

2. PROPOSED SCALING ALGORITHM

Fig. 1 shows the block diagram of the proposed scaling algorithm. the proposed scaling algorithm consists of a sharpening spatial filter, a clamp filter, and a bilinear interpolation. The spatial and clamp filters of the whole algorithm performs the prefiltering operation and also reduces the alaising artifacts produced by the bilinear interpolation . First he input pixels of the original image may consists of noise and they are filtered to enhance the edges and remove the noise associated. Second, the clamp filters are used to smoothen the unwanted edges and boundaries. Third, the filtered input is processed by the bilinear interpolation for the scaling process either up or down scaling. For the conservation of the computation resources and memory buffers, the both clamp and spatial filters are combined in to a combined filter. The details of each part will be described in the following sections.

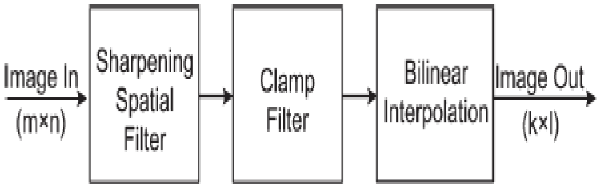


Fig.1 Block diagram of existing scaling algorithm for image zooming

2.1 LOW-COMPLEXITY SHARPENING SPATIAL AND CLAMP FILTERS

The sharpening spatial filter used here is a kind of high-pass filter, is used to reduce the artifacts

ISSN: 2278 – 7798

such as blurring which is defined by a kernel this increases the intensity of a center pixel relative to its neighboring pixels. The clamp filter is a type of low-pass filter and a 2-D Gaussian spatial domain filter and composed of a convolution kernel array. Kernel array contains a single positive value at the center and is completely surrounded by ones.

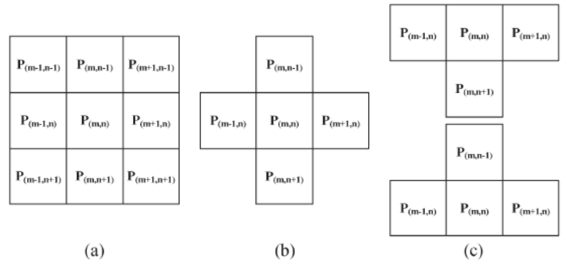


Fig.2 Weights of convolution kernels. (a) 3x3 convolution kernel. (b) cross- model convolution kernel. (c) T- model and inversed T- model convolution kernels

To reduce the complexity of the 3 × 3 convolution kernel, a cross-model formed is used here to replace the 3 × 3 convolution kernel, as shown in Figure 3.2(b) . It successfully cuts down on four of nine parameters in the 3 × 3 convolution kernel. Further to decrease more complexity and memory requirement of the cross- model convolution kernel a T-model and inversed T-model convolution kernels are proposed for realizing the sharpening spatial and clamp filters. As shown in Figure 2. , the T-model convolution kernel is composed of the lower four parameters of the cross-model, and the inversed T-model convolution kernel is composed of the upper four parameters. 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.

2.2 COMBINED FILTER

In proposed scaling algorithm, the input image is filtered by a sharpening spatial filter and then filtered by a clamp spatial filter again. Even though the sharpening spatial and clamp filters are simplified by T-models and inversed T-models, now it still needs two line buffers to store input data or intermediate values for each T-model or inversed T-model filter. So, to be able to reduce more computing resource and factors such as memory buffer requirement, sharpening spatial and clamp filters, which are formed by the alreadymentioned T-model or inversed T-model, should be combined together into a combined filter as

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

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

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

Where S and C are the sharp and clamp parameters and P_(m,n) is the filtered result of the target pixel P(m,n) by the combined filter

2.3 SIMPLIFIED BILINEAR INTERPOLATION

In the many proposed scaling algorithm, the bilinear interpolation method is chosen because of its characteristics with low complexity in operation and high quality in performance. The bilinear interpolation is an high quality operation that performs a linear interpolation in both direction, that is first in one direction and, then again immediately, in the other direction. Then 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 neighbor pixels. The target pixel P(k,l) can be calculated by

$$P_{(\mathrm{k,l})} = (1-dx)\times(1-dy)\times P_{(\mathrm{m,n})} + dx\times(1-dy)\times P_{(\mathrm{m+1,n})} + (1-dx)\times dy\times P_{(\mathrm{m,n+1})} + dx\times dy\times P_{(\mathrm{m+1,n+1})} \qquad (3)$$

where P(m,n), P(m+1,n), P(m,n+1), and P(m+1,n+1) are the four nearest neighbor pixels of the original image and the dx and dy are scale parameters in the horizontal and vertical directions.

3. VLSI ARCHITECTURE

The proposed scaling algorithm consists of two combined prefilters and one simplified bilinearinterpolator

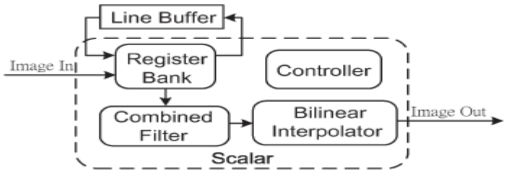


Fig.3 Block diagram of VLSI architecture for proposed real-time image scaling processor

For VLSI implementation, the bilinear interpolator can directly obtain two input pixels P’(m,n) andP’(m,n+1) from two combined prefilterswithout any additional line-buffer memory.

ISSN: 2278 – 7798

3.1 REGISTER BANK

In this brief, 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 uniquely designed with a single-line memory buffer, which is then used to provide the ten values for the immediate usage of the combined filter.

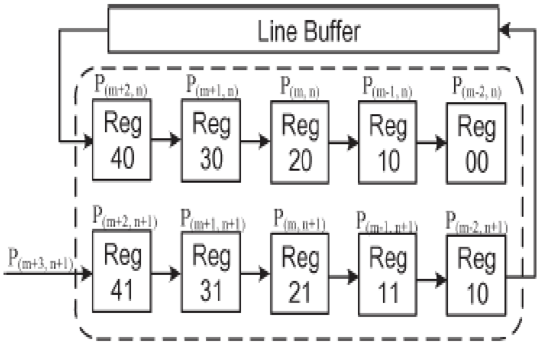
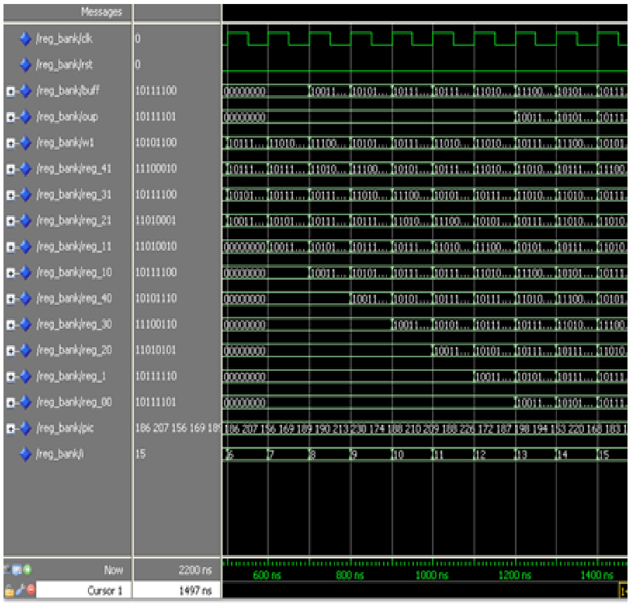


Fig.4 Architecture of register bank

Figure 3.4 shows the architecture of the register bank with a structure of ten shift registers. While the shifting process ,control signal is produced from the controller, a new value of P(m+3,n) will be read into the Reg41, and then 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) from the line-buffer memory, and every time each value in other registers belonging to row n will be shifted right into
the next register.

Fig 4 shows the output of the register bank. Here in this register bank the first input is stored in the register Reg 00by the shifting operation. The register bank is worked out with the use of Modelsim software. This is asusal done as shown in the figure, that is with the help of ten registers.

The following is the output of the register bank with ten registers as this is visible from the waveform itself.



```
Messages
/reg_bank/clk        0
/reg_bank/rst        0
/reg_bank/buff       10111100
/reg_bank/oup        10111101
/reg_bank/w1         10101100
/reg_bank/reg_41     11100010
/reg_bank/reg_31     10111100
/reg_bank/reg_21     11010001
/reg_bank/reg_11     11010010
/reg_bank/reg_10     10111100
/reg_bank/reg_40     10101110
/reg_bank/reg_30     11100110
/reg_bank/reg_20     11010101
/reg_bank/reg_1      10111110
/reg_bank/reg_00     10111101
/reg_bank/pic        186 207 156 169 18
/reg_bank/i          15
Now                  2200 ns
Cursor 1             1497 ns
```

Fig.4Output of register bank

3.2 COMBINED FILTER IN VLSI ARCHITECTURE

The combined T-model or inversed T-model convolution function of the sharpening spatial and clamp filters had been discussed in Section II, and the equation is represented in (3.3)

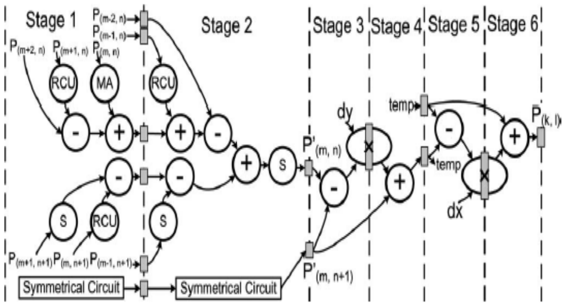


Fig. 5 Computational scheduling of the proposed combined filter and simplified bilinear interpolator

Figure 5. shows the six-stage pipelined architecture of the combined filter and bilinear interpolator, which reduces the delay path to improve the performance by pipeline technology. The stages 1 and 2 in Figure 5, show the computational scheduling of a T-model combined and an inversed T-model filter. Here the T-model or inversed T-model filter consists of three reconfigurable calculation units (RCUs), one multiplier–adder (MA), three adders (+),

ISSN: 2278 – 7798

three subtracters (-), and three shifters (S)

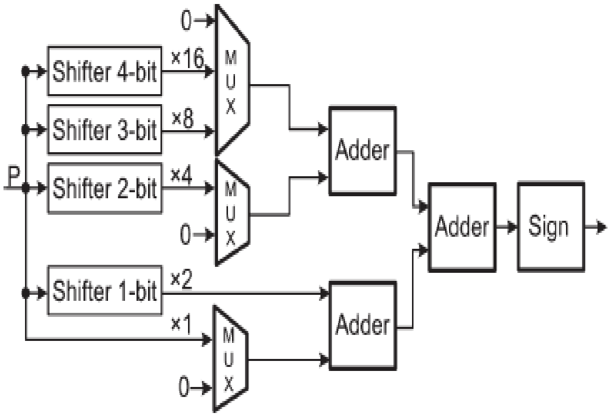
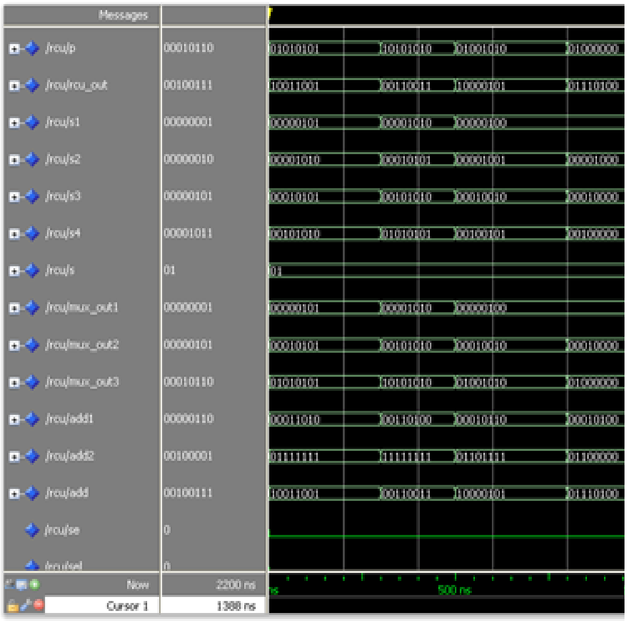


Fig. 6. Architecture of the RCU.

Figure 6, shows the architecture of the RCU. It consists of four shifters, three multiplexers (MUX), three adders, and one sign circuit. By this RCU design, the hardware cost of the combined filters can be efficiently reduced. Figure shows the output of the RCU unit.



```
Messages
/rcu/p           00010110   01010101  10101010  01001010  01000000
/rcu/rcu_out     00100111   10011001  00110011  10000101  01110100
/rcu/s1          00000001   00000101  00001010  00000100
/rcu/s2          00000010   00001010  00010101  00001001  00001000
/rcu/s3          00000101   00010101  00101010  00010010  00010000
/rcu/s4          00001011   00101010  01010101  00100101  00100000
/rcu/s           01         01
/rcu/mux_out1    00000001   00000101  00001010  00000100
/rcu/mux_out2    00000101   00010101  00101010  00010010  00010000
/rcu/mux_out3    00010110   01010101  10101010  01001010  01000000
/rcu/add1        00000110   00011010  00110100  00010110  00010100
/rcu/add2        00100001   01111111  11111111  01101111  01100000
/rcu/add         00100111   10011001  00110011  10000101  01110100
/rcu/se          0
/rcu/sel         0
Now              2200 ns    ns        500 ns
Cursor 1         1388 ns
```

Fig. 7Output of the RCU

3.3 BILINEAR INTERPOLATOR AND CONTROLLER

In the previous discussion, the bilinear interpolation is simplified as shown in (6). The stages 3, 4, 5, and 6 in Figure.3.5, show the four-stage pipelined architecture, and the two-stage pipelined multipliers are used to shorten 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 symmetrical circuit. By the hardware sharing technique, as shown in (6), 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 only one register. The controller of this is implemented by a finite-state machine circuit. It generates control signals to control the timing and pipeline stages of the register bank, combined filter, and bilinear interpolator.

4.CONCLUSION

A low-cost, a low-memory-requirement, a high quality, and ahigh-performance VLSI architecture of the image scaling processor has been proposed. The filter combining technique, hardware sharing technique, and reconfigurable techniques had been used to reduce hardware cost. Relative to previous low-complexity VLSI scalar designs, this work achieves at least 34.5% reduction in gate counts and requires only one-line memory buffer.

REFERENCES

[1].Cardells-Tormo. F and Arnabat-Benedicto. J, “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.

[2].Chen. P. Y, Huang C. C, Shiau. Y. H, and Chen. Y. T, “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.

[3].Chen. S. L, Huang. H. Y, and Luo. C. H, “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.

[4].Fons .M, Fons. F, and Canto.E, “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. [5].Han. J.

[5]W, Kim. K, Cheon S. H, Kim J.O, and Ko .S. J, “A novel image interpolation method using the bilateral filter,” IEEE Trans. Consum. Electron., vol. 56, no. 1, pp. 175–181, Feb. 2010.