ISSN:2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 6, June 2015

Implementation of Image Scalar based on Bilinear Interpolation Using FPGA

Vidyashree.N
Dept. of ECE
Dr. AIT, Bangalore

Usharani.S
Associate Professor
Dept. of ECE
Dr. AIT, Bangalore

Abstract—In this work, A bilinear interpolation method is been used to improve the quality of the Scaled image. A high quality algorithm is proposed which includes sharpening spatial filter and clamp filter used as a combined filter to preprocess the image before interpolating. This increases the quality of the image and as well reduces the drawbacks such as aliasing artifacts produced from bilinear interpolator. some of the other major improvements of this algorithm say utilization of T & inverse T model kernel for combined filter consists of RCU (reconfigurable calculation unit) to control filter operations reduces the memory buffer and Hardware, Reduction in the bilinear interpolation computation results in lesser area for the design in VLSI intern increases the speed and reduces power, cost of overall implementation. This work achieves 168.18MHz with 1.507K gate counts. It is implemented using Xilinx 13.4. Compared to previous low complexity techniques, this method produces better image quality.

Index Terms—sharpening spatial filter, clamp filter, bilinear interpolator, CLA, T model, Inverse T model.

I. INTRODUCTION

In the growing modern electronic devices the display quality plays a very important role. With this growth the need of the Image scaling also has increased. The process of fitting a larger resolution image into a small display device or the other way round is known as image resizing, In Digital terminology it is called as Image scaling. This has been widely spread over the different domains for various purposes such as military, consumer electronics, multimedia devices and medical equipment's. If a Smaller resolution image is increased with size to fit a large display the quality of the image will be reduced and hence it requires additional processing on the resized image to increase the quality. With Increasing demand of the modern electronics in various domains increases the demand of higher quality scaling methods.

Scaling methodscan be categorized into polynomial and non-polynomial methods. One of the simplest polynomial methods is the nearest neighbor interpolation,which replaces every pixel with number of pixels of the same color. It has an advantage of low complexity and ease of implementation. Another polynomial method is the bilinear interpolation, which uses linear interpolation to calculate unknown pixels in both horizontal and vertical directions. It has a good computational efficiency and good image quality. The bicubic interpolation is an extension of cubic interpolation for interpolating image pixels on a 2D regular grid. It provides a good quality of image but the computational effort is more.

There are many high quality non-polynomial based methods like curvature interpolation [1], bilateral filter [2], autoregressive model [3], which greatly enhances the image quality and also reduces the blocking, aliasing and blurring artifacts. But the complexity and memory requirement of these non-polynomial methods are morewhich is a main drawback for real time applications. Hence, a low complexity and high quality algorithm is preferred for VLSI implementation of a real time application.

The goal of this work is to improve the quality of the restructured image by using bilinear interpolation technique and reduce the complexity and memory requirement of VLSI architecture.

II. PROPOSED SCALING ALGORITHM

Fig.1 shows the proposed image scaling algorithm. It consists of sharpening spatial filter, clamp filter and bilinear interpolator.

Fig.1 Proposed image scaling algorithm

A. *Sharpening spatial filter*

Sharpening spatial filter is a kind of high pass filter which is used to enhance the edges as well as the details of objects and is effective in removing associated noise. It is defined by a kernel to increase brightness of the center pixel relative to its neighbor pixels. It contains a single positive value at center and completely surrounded by negative valued pixels. Sharpening spatial filter is used to improve the appearance of images by increasing the delineation between bright and dark regions.

$$\text{Kernel}_S = \begin{bmatrix} -1 & -1 & -1 \\ -1 & S & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

B. *Clamp filter*

It is a kind of low pass filter, which is used to smoothen the unwanted discontinuous edges of the boundary regions and reduce aliasing artifacts. The kernel of the clamp filter is combined with matrix coefficients that show the dependence of a filtered pixel on its neighbors. This kernel contains a single positive value at its center and is completely surrounded by ones.

$$\text{Kernel}_{c=}\begin{bmatrix}1 & 1 & 1\\ 1 & C & 1\\ 1 & 1 & 1\end{bmatrix}$$

C. *Convolution kernels*

The sharpening spatial filter and clamp filter are realized by convolution kernels. There are different types of convolution kernels. The simplest one is the 3x3 convolution kernel but it requires four line memory buffers. To reduce the memory requirement of a 3x3 convolution kernel, a cross model convolution kernel is used which successfully cuts down on 4 of 9 parameters in 3x3 convolution kernel. To reduce complexity and memory requirement of cross model convolution kernel, T-model and Inverse T-model convolution kernel are used for sharpening spatial filter and clamp filter. The T-model and inverse T- model convolution kernel improves the quality of images, reduces the complexity of the convolution kernel and also decreases the memory requirement from 2 to 1 line buffer. Using these kernels we can achieve low complexity and low memory requirement for sharpening spatial filter and the clamp filter.

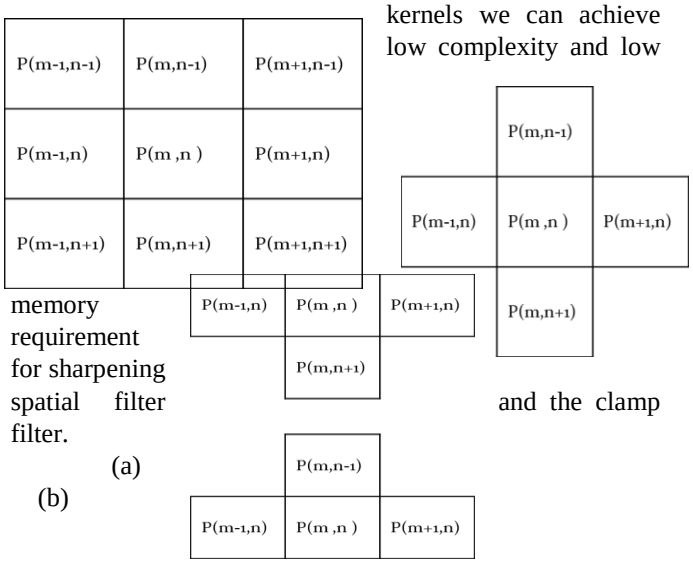

Fig. 3 Weights of the convolution kernels. (a) 3x3 convolution kernel. (b) Cross model convolution kernel (c) T-model and inverse T model convolution kernel

D. *Combined filter*

Sharpening spatial filter and the clamp filter are the two pre-filters which are combined into combinedfilter to reduce the memory requirement because it requires two separate T-model and inversed T-model filters to store input data or intermediate values. Thus, to reduce the more computing resource and memory requirement, T-model or inverse –T model can be combined together into a combined filter as

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

$$= P_{(m,n)}\begin{bmatrix}-1 & S-C & SC-2 & S-C & -1\\ & -2 & S-C & -2 & \\ & & -1 & & \end{bmatrix} / [(S-3)\times(C+3)] \quad (1)$$

$$\approx P_{(m,n)}\begin{bmatrix}-1 & S-C & SC-2 & S-C & -1\\ & -2 & S-C & -2 & \end{bmatrix} / [(S-3)\times(C+3)] \quad (2)$$

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. By this technique of combined filter, the demand of memory can be efficiently reduced from 2 to 1 line buffer, which greatly reduces the memory access requirement of software systems or hardware memory costs for hardware design.

E. *Simplified bi-linear Interpolator*

The Bi-linear interpolation has the characteristics of low complexity and high quality which performs linear interpolation in both horizontal and vertical direction. The output pixel $P_{(k,l)}$ can be calculated in both X and Y directions with the 4 nearest neighbor pixels. The target pixel $P_{(k,l)}$ can be calculated by

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

$P_{(m,n)}$, $P_{(m+1,n)}$, $P_{(m,n+1)}$ and $P_{(m+1,n+1)}$ are the 4 nearest neighbor pixels of the original image. And dx and dy are scale parameters in the horizontal and vertical directions.

$$P_{(k,l)} = \{[P_{(m+1,n)} + dy\times(P_{(m+1,n+1)} - P_{(m+1,n)})] - Pm,n+dy\times Pm,n+1-Pm,ndx+ [P_{(m,n)} + dy\times(P_{(m,n+1)} - P_{(m,n)})] \quad (4)$$

The simplified equation 4 successfully reduces the computing resource from 8 multiply, 4 subtract and 3 add operations to 2 multiply, 2 subtract and 2 add operations.

III. VLSI ARCHITECTURE

Fig.4 shows the block diagram of the proposedhigh qualityimage scaling processor. It consists ofa register bank, combined filter, bi linear interpolator and controller.

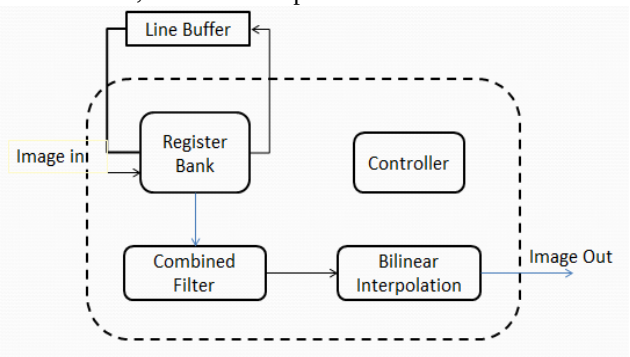

Fig.4 Block diagram of the proposed real time image scaling processor

A. *Register Bank*

The combined filter produces the values P'(m,n) and P'(m,n+1) by making use of ten pixels at a time and in order

to provide these values ten valued memory buffer is used. There are two one line memory of 5 bit values used to storethe valueeach for one combined filter of T model and the other for the symmetrical circuit of inverse T model to produce the P'(m,n) value of the first T model combined filter and the P'(m,n+1) value of the inverse T model combined filter. When the controller providesnew clock cycle, the m,m+1,m+2,m-1 and m-2 values will get loaded to the in line buffer with the n as constant. And this cycle repeats for every pixel calculation for each clock.

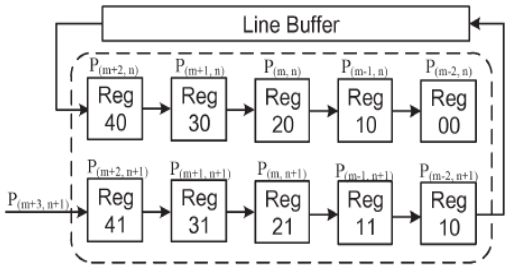

Fig. 5Architecture of register bank

B. Combined filter

Fig 6.shows the pictorial representation of the final equation.It consists of 6 stages where in the first 2 stages comprises of the combined filters and the later 4 stages realizes the bilinear

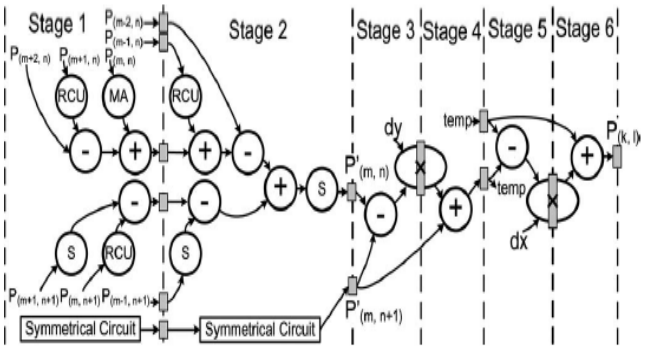

Fig6.Computational scheduling of proposed combined filter and simplified bilinear interpolator.

interpolator. The bilinear interpolation as shown in the equation (3) consists of two multiplications, two subtractions and two addition operations along with the filtered output which is free from sharpening and aliasing values the dx and dy values of the image has been provided to the bilinear interpolator to perform the operation.

The initial stage 1 and 2 which comprises of the inverse T model filter makes use of one MA (multiplier adder), three RCU reconfigurable unit. The Multiplier first multiplies the nibble data of the two neighboring pixels and then produces the added output of the 4pixels. This data is then passed on to the adder along with the RCU data to get the filtered value.

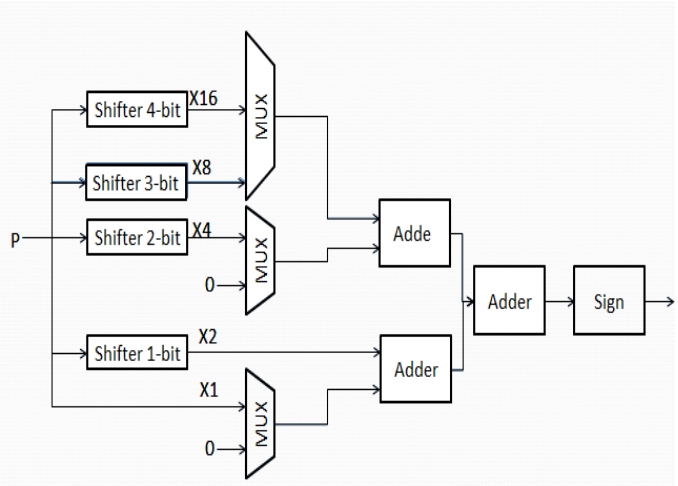

Fig. 7Architecture of the RCU

C.Bilinear Interpolator

Stage 3 to stage 6 in the above figure comprises of the bilinear interpolator, the bilinear interpolator makes use of the temp variables to store the intermediate values obtained from the combined filter and later use them for the calculation of the interpolated pixel value.

The dx and dy are the deviation values of the pixel which have been used in the equation.

The stages have been used to reduce the delay of the data flow between the input and output, the data from input to output will have a delay of 4 clocks.

C. Controller.

Controller is used to handle the data flow of the pixel values. It is designed to provide the clocks which is used in various stages of the computation,s uch as memory storage of the register bank, value shifts in the RCU and the MA. Instead of the division to be carried out in the combined filter the shifting of the bits has been used in which the shifting is handled by the controller. The passing of values between the different stages of the computation is also handled by the controller including the bi linear interpolator.

IV. SIMULATION RESULTS

The VLSI architecture of this work is using the hardware description language Verilog. Xilinx 13.4 is used for synthesis and implementation of the design. The input image is resized into 100x100 image using Matlab and the pixel value of the resized image is obtained through a block memory generator and it is fed to combined filter and bilinear interpolator through register bank to improve the quality of the resized image. The processed image through all stages is observed in Matlab. Virtex-5 is used for FPGA implementation.

ISSN:2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 6, June 2015

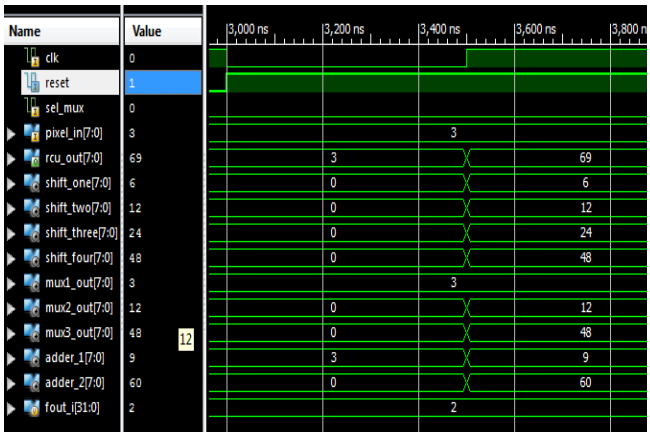

Fig. 8Simulation results of RCU

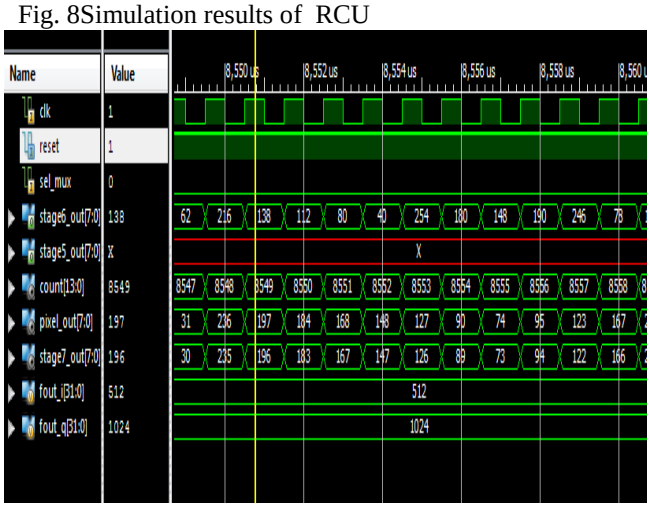

Fig. 9Simulation results of combined filter and bilinear interpolator

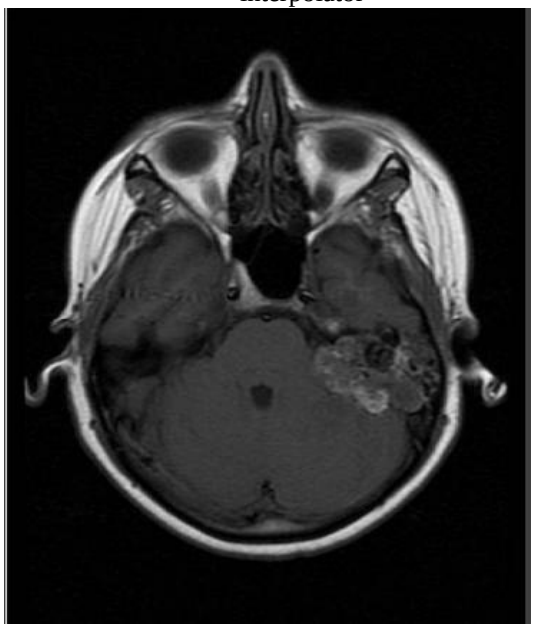

Fig.10Original image

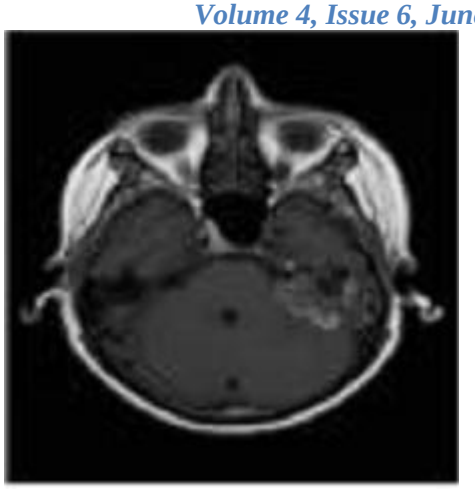

Fig.11Resized image(100x100)

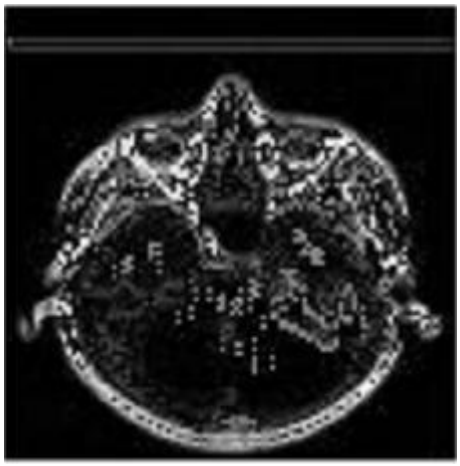

Fig.12 combined filter output

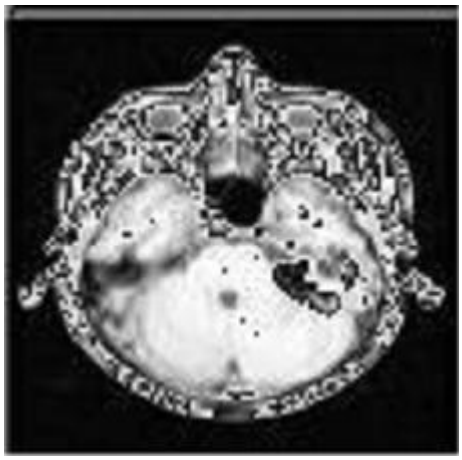

Fig.12 Final processed image of combined filter and bilinear interpolator

V. OUTCOME OF THE WORK

	Win	Bi Cubic convolution	Adaptive Scalar	This Work
Process	FPGA	0.13µm	0.13µm	FPGA
Gate counts	29K	30.6K	9.28K	1.507K
Simulation Power	N/A	19.17mW	9.6mW	4.57mW
Frequency	65MHz	279MHz	280MHz	168.180MHz
Throughput (pixel/sec)	N/A	N/A	280M	N/A
Line buffer Memory	4 lines	6 lines	4 lines	1 lines

VI. CONCLUSION

The VLSI architecture of this work reduces the gate count by 5.23% and requires only one line memory buffer and achieve 168.18MHz frequency and its chip area is $5um^2$ synthesized by Xilinx ISE Simulator.In this VLSI architecture the quality of the image is greatly enhanced by using a bilinear interpolation algorithm.

VII. 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, "A novel 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]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.

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

[6] 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.

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

[8] 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.

[9] S. L. Chen, "VLSI implementation of a low-cost high-quality image scaling processor," IEEE Trans. Circuits Syst., vol. 60, no.1, pp. 31-35, Jan.2013.