International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:268
Vol.2, No.4, April 2013 DOI:10.15693/ijaist/2013.v2i4.76-79

Vlsi Implementation Of Adaptive Pre-Filtering Techniques For Image Resampling Noise Compensation

* Mr.D.Praveen Rajadurai ** Mrs.Karthika Devi M.E *** Dr.R.Ganesan M.E., Ph.D.
* PG Student, M.E (VLSI Design), Sethu institute of Technology.
**Asst. professor (Senior Grade), Department of ECE, Sethu institute of Technology.
*** Head of the Department, Department of VLSI Design, Sethu institute of Technology.

Abstract— **A scaling algorithm is proposed for the implementation of image scaling. The method consists of a bilinear interpolation, a clamp filter, and a sharpening spatial filter. The bilinear interpolation is selected due to its low complexity and high quality. An adaptive technology is used to enhance the effects of clamp and sharpening spatial filters. The clamp and sharpening spatial filters are added as pre-filters to solve the blurring and aliasing effects produced by bilinear interpolation. To reduce memory buffers and computing resources for the very large scale integration (VLSI) implementation, the clamp filter and sharpening spatial filters both convoluted by a 3×3 matrix coefficient kernel are combined into a 5×5 combined convolution filter, Compared with previous techniques, this paper not only reduces gate counts by more than 46.6% and power consumptions by 24.2%, but also improves average quality by over 0.42 dB. . The bilinear interpolation is simplified by the co-operation and hardware sharing technique to reduce computing resource and hardware costs. The PSNR values for the scaled image are used to signify the overall quality of the scaling Algorithm.**

Keywords— **Image scaling, adaptive, bilinear interpolation, clamp filter, image zooming, sharpening spatial filter.**

I. INTRODUCTION

Digital image processing is a subset of the electronic domain where in the image is converted to an array of small integers, called pixels, representing a physical quantity such as scene radiance, stored in a digital memory, and processed by computer or other digital hardware. Digital image processing, either as enhancement for human observers or performing autonomous analysis, offers advantages in cost, speed, and flexibility, and with the rapidly falling price and rising performance of personal computers it has become the dominant method in use. In many applications, from consumer electronics [1], [2] to medical imaging [3], it is desirable to improve the restructured image quality and processing performance of hardware implementation. For example, a video source with a 640×480 video graphics array (VGA) resolution may need to fit the 1920×1080 resolution of a high-definition multimedia interface (HDMI). On the other hand, a high-resolution image may need to scale down to a small size in order to fit the lower resolution of small LCD panels. In recent years, a number of efficient scaling methods have been developed [4]–[14] for image zooming. In the mathematical field of numerical analysis, interpolation is a method of constructing new data points within the range of a discrete set of known data points. In engineering and science, one often has a number of data points, obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable. It is often required to interpolate (i.e. estimate) the value of that function for an intermediate value of the independent variable. This may be achieved by curve fitting or analysis.

One of the simplest methods is linear interpolation. Generally, linear interpolation takes two data points, say (x_a,y_a) and (x_b,y_b). At the point (x,y) Linear interpolation is quick and easy, but it is not very precise. Another disadvantage is that the interpolant is not differentiable at the point x_k.

In words, the error is proportional to the square of the distance between the data points. The error in some other methods, including polynomial interpolation and spline interpolation (described below), is proportional to higher powers of the distance between the data points. These methods also produce smoother interpolates.Among the many polynomial-based methods [4]–[8] proposed for image scaling, the nearest-neighbor algorithm [4], [5] is considered the simplest method. It affords both the benefits of low complexity and ease of implementation. However, the images produced by this algorithm are full of blocking and aliasing artifacts. The bilinear algorithm, which uses linear interpolation models to calculate unknown pixels, is the most widely-used method to reduce the blocking and aliasing effects [6], [7]. Moreover, in many image-processing applications, bilinear interpolation algorithm is popular due to its computational efficiency and image quality. Nevertheless, the produced image exhibits blurring edges and aliasing effect after scaling. However, it causes the edges of the scaled images to become blurred and aliased after interpolation.

This paper is organized as follows. In Section II, the bilinear interpolation, clamp filter, and sharpening special filter are briefly introduced. The novel adaptive scaling algorithm is presented in Section III. Section IV describes the VLSI architecture of this paper. Section V shows the experimental results and chip implementation. Finally, in Section VI, the conclusions are presented.

II. BILINEAR INTERPOLATION AND FILTERS

This section begins with a brief review of bilinear interpolation algorithm. We then introduce a generalized weighted smoothing filter, a clamp filter, for smoothing away the uncontrollable condition known as an aliasing effect. Finally, a sharpening spatial filter, which is a type of high-pass filter, is presented to show how it reduces blurring effects.

A. BILINEAR INTERPOLATION

Bilinear interpolation is an image-restoring algorithm, which linearly interpolates four nearest-neighbor pixels of an unrestored image to obtain the pixel of a restored image as a forward function. The principle behind the bilinear interpolation algorithm is executing a linear interpolation in one direction, and then repeating the same function in the other direction. As shown in Fig. 1, P(i, j), P(i+1,j), P(i,j+1), and P(i+1,j+1) are the four nearest neighbor pixels of the original image with i = [0, 1, 2, . . . M] and j = [0, 1, 2, . . . N]. Here, M is the number of pixels having the width of the original image and N is the number of pixels corresponding to the length of the original image.

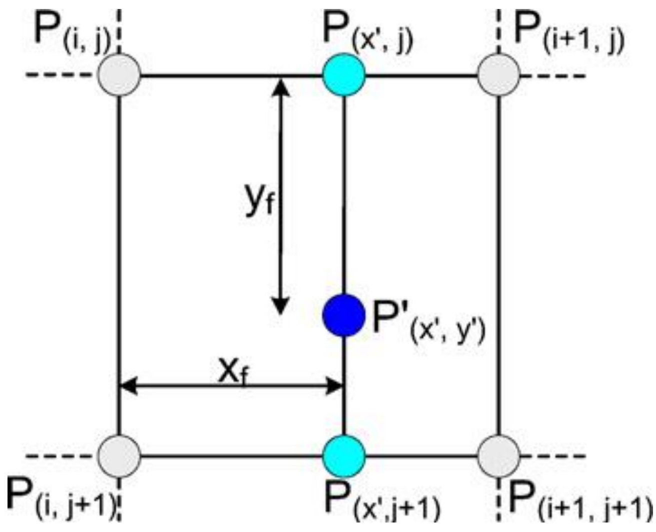

Fig.2. Bilinear Interpolation.

The temporary pixels $P(x',j)$ and $P(x',j+1)$are created by linear interpolation in horizontal direction and can be calculated as

$$P(x\_, j)= (1 - xf) \times P(i, j)+xf\times P(i+1, j) \quad (1)$$

$$P(x\_, j+1) = (1- xf) \times P(i, j+1) + xf\times P(i+1, j+1) \quad (2)$$

Where xf is the scale parameter in horizontal direction. After interpolating in horizontal direction, the values of temporary pixels P(x',j) and P(x',j+1) are generated. The resulting output Pixel P(x',y') can be obtained by one more linear interpolation in the other direction. Alternatively, the output can be produced by implementing linear interpolation in the vertical direction and can be calculated as.

$$P(x\_, y\_) = [(1 - xf) \times P(i, j) + xf\times P(i+1, j)] \times (1 - yf)+[(1 - xf) \times P(i, j+1) + xf\times P(i+1, j+1)] \times yf \quad (3)$$

Where y_f is the scale parameter in vertical direction. Bilinear interpolation is popular in the implementation of VLSI chips due to its low complexity and simple architecture.

However, its high-frequency response behavior is poor as a result of linear changes to the output pixel value according to sampling position. Results show that the edges become blurry and the aliasing effect is visible after being processed using bilinear interpolation.

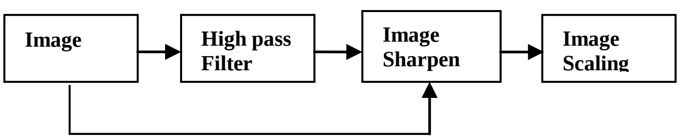

Fig.3. Existing Block Diagram

B. SHARPENING SPATIAL FILTER

The sharpening spatial filter is a kind of high pass filter, is used to enhance the edges as well as the details of objects, and is extremely effective at removing associated noise. The kernel of the sharpening spatial filter is designed to increase the brightness of the center pixel relative to neighboring pixels. It usually contains a single positive value at its center and completely surrounded by negative values. The following array is an example of a 3 × 3 kernel for a sharpening spatial filter

$$Kernel_s = \begin{bmatrix} -1 & -1 & -1 \\ -1 & S & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Where S is a sharp parameter that can be set according to the characteristics of the images. The degree of sharpening can be adjusted by changing the sharp parameter. In this paper, the sharpening spatial filter is used to improve the appearance of images by increasing the delineation between bright and dark regions, which reduces blurring effects caused by bilinear interpolation. As with the clamp filter, results show that the filtered image is clearer than the original one.

Fig.3 Block Diagram of Our Adaptive Scaling Algorithm for Image Zooming.

III. ADAPTIVE SCALING ALGORITHM

Fig.3 shows the block diagram of the proposed 2-D adaptive scaling algorithm. This novel scaling algorithm consists of a clamp filter and a sharpening spatial filter, which serve as pre-filters in solving the blurring and the aliasing problems caused by the bilinear interpolation. To conserve memory, these two 3 × 3 filters are combined into one 5 × 5 filter.

Moreover, an adaptive technology is used to enhance the effects of clamp and sharpening filters. As shown in Fig.5. 4, the results of the sharpening spatial filter are adaptively written or stored back to combine with the image input pixels and shall be used as input data for subsequent filtering.

The feedback activity makes the combined filter an infinite impulse response (IIR) filter. To prevent the IIR filter from exhibiting an unstable. Filter response, the parameters of the clamp and sharpening spatial filters are selected to ensure stability. In the proposed scaling algorithm, because the clamp and sharpening spatial filters are implemented by a convolution operation, the filtered results are amplified by filter gains. To keep the average intensity of scaled image near to the original value, the results from the filter should be divided by the filter gain. The CF gain (clamp filter gain) can be calculated as

$$\text{CF gain} = C + 8$$

Where C represents the clamp parameter. Next, the sharpening spatial filter is used to enhance the edges of the smoothed image. After which its gain (SSF gain) is divided accordingly. The SSF gain can be computed as

SSF gain = S − 8

Where S stands for the sharp parameter. Finally, the filtered image is scaled from the image sizes of i × j to k × l by bilinear interpolation.

A. *Combined Filter*

In our scaling algorithm, the original image is filtered by 3×3 clamp filter and again by a 3×3 sharpening spatial filter. In real multimedia applications, the intermediate values which are filtered by clamp filter must be stored in three line-buffers as input data to the sharpening spatial filter. For example, if the image width is 1920 pixels, the 3 × 1920 × 8 bits of intermediate values must be buffered in the memory. It requires considerable memory access for a software system or a large silicon area to integrate three line-buffers into VLSI scalar chip. Moreover, the CF gain and SSF gain can be combined, after which the results only need to be divided by the combined filter gain once. The combined filter gain can be calculated as

Combined filter gain = (C + 8) × (S − 8)

Where C and S are the clamp and sharp parameters, respectively. The memory buffer requirement decreases from five to four line-buffers. This combined-filter technology greatly enhances performance, such as by reducing memory access requirements for software systems or reducing hardware memory costs for hardware design.

B. *Adaptive Convolution Algorithm*

The combined 5 × 5 pre-filter not only provides an efficient method to reduce a one line-buffer memory requirement, but also lessens the blurring and aliasing effects from bilinear interpolation. However, the improvement in image quality is inconspicuous. Thus, adaptive technology is employed to further improve the quality of scaled images. The resulting pixels of the combined filter are adaptively written or stored into the third and fourth line buffers. Hence, two out of five input pixels of the combined filter are read from the line-buffers, which are full of filtered pixels. Since 25 values are stored in the registers of the combined filter as input for convolution computation, the results of the combined filter are produced by these stored values.

C. *High-pass filter*

A high-pass filter (HPF) is an electronic filter that passes high-frequency signals but attenuates (reduces the amplitude of) signals with frequencies lower than the cut off frequency. The actual amount of attenuation for each frequency varies from filter to filter. A high-pass filter is usually modelled as a linear time-invariant system. It is sometimes called a low-cut filter or bass-cut filter. High-pass filters have many uses, such as blocking DC from circuitry sensitive to non-zero average voltages or RF devices. They can also be used in conjunction with a low-pass filter to make a bandpass filter.

IV. Experimental Results and Chip Implementation

To be able to analyze the qualities of the proposed adaptive scaling algorithm, eight sample images of 512 × 512 3-byte color pixels were selected as a test set. Fig. 12 lists the eight test images, some of which have also been used as test images in previous papers dealing with image interpolation or image processing. For each test image, a fixed low-pass filter (averaging filter) is used to filter the original image and scale up/down the filtered image by various algorithms, and then refine the scaled images back to the original size by the same interpolation algorithm. Four well-known scaling algorithms, Winscale (Win), modified Winscale (M−Win), bicubic (BC) [8], and bilinear (BL) [6], [7], and each step of the proposed adaptive scaling algorithm are used to analyze the qualities of each algorithm by using the eight testing images.

To attain a quantitative measurement of the quality of the refined image, the mean square error (MSE) is used to quantify a noisy approximation of the $m \times n$ refined image and the original image. The MSE is defined as

$$MSE = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left[P(i,j) - P'(i,j)\right]^2$$

(4)

where MN is the size of the original image since the intensity of each pixel varies between zero and 255. Moreover, a peak signal-to-noise ratio (PSNR) is used for this purpose and is measured as

$$PSNR = 10\log_{10}\frac{MAX^2}{MSE}. \quad (5)$$

When the pixels are represented using eight bits per sample, the maximum value of each pixel (MAX) is 255. Therefore, the quality tends to be expressed in dB with a peak signal-to noise ratio PSNR given as

$$PSNR = 10\log_{10}\frac{255^2}{MSE}. \quad (6)$$

To obtain the qualities of four well-known algorithms with this paper, MATLAB (Mathworks, Natick, MA) is used to evaluate the PSNR values between the refined and original images.

Fig 4.1 Input and Upscaled Image with Edge sharpen filter.

Fig 4.2 Input and Downscaled Image with Edge sharpen filter.

Test Images	512 x 512 to 1024 x 1024				
	Bi cubic	Win scale	PSNR Values Using Filters		
			LPF	Laplace	Adaptive
Lena	31.23	30.51	51.55	48.17	57.78
Airplane	30.01	29.51	48.56	45.60	55.80
Pepper	29.89	29.37	52.52	49.73	59.08
Mandrill	22.86	22.70	44.37	41.98	52.16

Table I. Up scaled Image

Test Images	512 x 512 to 256 x 256				
	Bi cubic	Win scale	PSNR Values Using Filters		
			LPF	Laplace	Adaptive
Lena	30.76	29.74	50.5	51.24	59.75
Airplane	28.90	27.79	45.28	46.42	54.85
Pepper	29.22	27.67	49.84	51.69	59.65
Mandrill	22.09	22.29	50.45	40.41	49.73

Table II. Downscaled Image

A low computational complexity, low-memory-buffer, low power consumption, high-performance, and high-quality scaling algorithm is needed for multimedia applications.

The hardware architecture of the 5 × 5 filter can be easily mapped with the convolution parameters shown in Fig. 3 and the convolution equation as shown in (8). The integrated architecture of the register bank and the combined filter not only promotes the performance to achieve cycle-by-cycle, but also reduces data access time with frame memory or line buffer memory. Distinct from the shifter, shifter-adder, or multiplier-adder of the combined filter, the calculating unit must be implemented with the clamp and sharp parameters.

V. Conclusions

In this paper, a novel adaptive scaling algorithm is proposed for developing a low-cost, low-power, and high-quality VLSI scaling circuit for image zooming applications. Bilinear interpolation is selected as an interpolation method due to its low complexity and high quality. A clamp filter and a sharpening spatial filter are added as pre-filters to solve the shortcomings of blurring and aliasing effects caused by the bilinear interpolation. With added adaptive skill, the quality of the resulting scaled images is notably improved. Techniques of filter combination, co-operation, and hardware sharing greatly reduce memory buffer requirements and hardware costs. Relative to other VLSI scaling circuit architectures, the work described here achieves a 46.6% reduction in gate counts, a 24.2%, improvement in power consumption and a 0.42 dB increase in average image quality. The proposed work is to eliminate the brightness and contrast noise in the scaled images.

References

[1] A. Amanatiadis and I. Andreadis, "An integrated architecture for adaptive
Image stabilization in zooming operation," *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, pp. 600–608, May 2008.
[2] C. Weerasnghe, M. Nilsson, S. Lichman, and I. Kharitonenko, "Digital zoom camera with image sharpening and suppression," *IEEE Trans.Consumer Electron.*, vol. 50, no. 3, pp. 777–786, Aug. 2004.
[3] T. M. Lehmann, C. Gonner, and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE Trans. Med. Imaging*, vol.18, no. 11, pp. 1049–1075, Nov. 1999.
[4] V. Caselles, J. M. Morel, and C. Sbert, "An axiomatic approach to image interpolation," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 376–386, Mar. 1998.
[5] E. Meijering, K. J. Zuiderveld, and M. A. Viergever, "Image reconstruction by convolution with symmetrical piecewise nth-order polynomial kernels," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 192–201, Feb. 1999.
[6] 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.
[7] W. Y. V. Leung, P. J. Bones, and R. G. Lane, "Statistical interpolation of sampled images," *Opt. Eng.*, vol. 40, no. 4, pp. 547–553, 2001.
[8] H. Hou and H. Andrew, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust. Speech Signal Process.* vol. 26, no. 6, pp. 508–517, Dec. 1978.
[9] L. Liang, "Image interpolation by blending kernels," *IEEE Signal Process. Lett.*, vol. 15, no. 1, pp. 805–808, Dec. 2008.
[10] 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.
[11] Q. Wang and R. K. Ward, "A new orientation-adaptive interpolation method," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 889–900, Apr.2007.
[12] H. Jiang and C. Moloney, "A new direction adaptive scheme for image interpolation," in *Proc. Int. Conf. Image Process.*, Jun. 2002, pp. 369– 372.
[13] A. M. Darwish, M. S. Bedair, and S. I. Shaheen, "Adaptive resampling algorithm for image zooming," in *Proc. Inst. Elect. Eng. Visual Image Signal Process.* 1997, pp. 207–212.
[14] M. Unser, "Enlargement or reduction of digital images with minimum loss of information," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 247–258, Mar. 1995.
[15] A. Raghupathy, P. Hsu, K. J. Ray Liu, and N. Chandrachoodan, "VLSI
Architecture and design for high performance adaptive video scaling," in *Proc. IEEE Int. Conf. Circuits Syst.*, Jul. 1999, pp. 406–409.