High Quality Raster Image Resizing Using Linear Weighted Manipulation

Gian-Hung Liu, Hsiao-Yu Han Yung-Kuo Ho, Chieh-Yi Huang, Hung-Pin Shih & Kevin Cheng, Industrial Technology Research Institute, Hsinchu, Taiwan, R. O. C. E-mail address ericliu05@itri.org.tw

Abstract

Due to the versatile customer applications and especially for large size industrial printing needs, the conflict comes between the image enlargement and high resolution quality. The image resizing process introduces specific pixels to the original image, or leaves out specific pixels from the original image. In order to maintain image quality and non-distortion, the value of such pixels and their positions were carefully deliberated. Prior researches such as DCT-based [2], wavelets, pixel replication linear interpolation [1] [3] those were limited by bandwidth and storage spaces, therefore a high efficiency resizing method is proposed in this paper. We introduce a new simple and high quality image-resizing algorithm that using linear weighted method for enlarging and drawing out particular pixels for reducing image size. The enlarging is first procedure and following with reducing procedure, where the image boundaries between those procedures needed to be carefully identification by linear weighted to avoid resolution loss. The simulation suggested the image can be enlarged up to four times and reduced down to quarter of original image at no resolution deterioration, and save the storage space need of fifty percent. Besides, the separation procedure is independent on enlarging procedure that can be operated if the image only needs to be reduced, same performance as mentioned before. This algorithm had been implemented by hardware and it will easily operate in software function in near future.

Keywords: Linear Weighted Method, Enlarging Process, Reducing Process, Image Resizing

Introduction

There are many applications of data manipulation un-compressed images and videos such as image resizing, bit rate changing, and so on. The image resizing is so called uses image process to magnify/reduce image. Most resizing methods such as interpolation, weight function, filter (includes low pass filter and high pass filter), morphological and frequency domain process. An interpolation function is applied extensively on whole image. It makes use of inset pixels to enlarge original image. The frequency domain process generally filters low frequency and remains high frequency to reduce the image size likes DCT (Discrete Cosine Transform) and DFT (Discrete Fourier Transform). However, those are the complicated and inconvenient algorithm that be implemented hardware solution. In this paper, we use the linear weighted function to enlarge the original image which is become four times one. Then we choose some specific dots and operate some result that to magnify or reduce procedure by rows or columns in the enlarged image. It can be exhibited the simplicity and easy to understand algorithm. The goal of algorithm blends the resizing into the existing halftone method, and achieves resizing and halftone with no additional calculation which can reserve important information even though after reducing procedure. Now it had implemented a hardware circuit module.

An example is to clarify this concept. If one wants to enlarge the image size by three hundred percent and we can enlarge the image to four hundred percent then reduce it a quarter. Obviously, we can get the particular image size by using this algorithm. Moreover, we separate the procedure from two modules in our hardware solution: one is front-step enlarge module and the second is back-step reduce module. In the next section, we explain our approach in details.

Algorithm Description

Error diffusion of halftone process spreads error adjacent to the pixel between the halftone and original image. We employ this concept to extend error near the pixel which would be drawing out. Fig 1 shows the combination of halftone and resizing block.

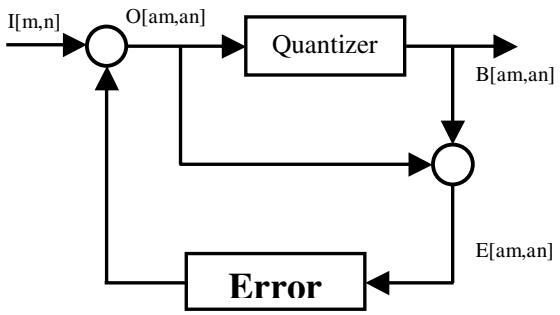

Fig 1. The combination of halftone and resizing

Fig 1 is similar to the halftone process, which is base on error-extend method. Nevertheless the output pixels have been decayed by the reduced rate. According Fig 1, the image process system cans be exhibit equation that shows below:

$$O[am,an] = I[m,n] + \sum_{k,r} E[am-k,an-r] \times c[k,r] \quad (1)$$

$$E[am,an] = O[am,an] - B[am,an] \quad (2)$$

I [m, n] is the input signal, and the O [am, an] is the input signal which been adjusted. The ”a” symbol is a reduced ratio, and its value is between twenty-five to one-hundred percent. The step of “a” is one percent. C [k, r] is the error-extend filter, and the value in the filter is weighted of error signal. [k, r] is the direction of error signal transmission. B[am, an] is final output.

$$B[am,an]=\begin{cases}1, & O[am,an]\geq\frac{2(D-1)-1}{2(D-1)}\\ \frac{D-2}{D-1},\ \frac{2(D-1)-3}{2(D-1)} & \leq O[am,an]<\frac{2(D-1)-1}{2(D-1)}\\ \vdots & \\ \frac{2}{D-1},\ \frac{3}{2(D-1)} & \leq O[am,an]<\frac{5}{2(D-1)}\\ \frac{1}{D-1},\ \frac{1}{2(D-1)} & \leq O[am,an]<\frac{3}{2(D-1)}\\ 0, & O[am,an]<\frac{1}{2(D-1)}\end{cases} \quad (3)$$

Now we give an example that according to our algorithm for 75% off image reducing. The original image is shown in Fig2 which includes the pixel P11~P59. The reducing process needs to delete some specific rows and columns which are depended on the resizing algorithm. The process includes reducing the 2nd, 4th, 6th, 8th row and column, the result of reduced 25 is shown in Fig3.

P11	P12	P13	P14	P15	P16	P17	P18	P19
P21	P22	P23	P24	P25	P26	P27	P28	P29
P31	P32	P33	P34	P35	P36	P37	P38	P39
P41	P42	P43	P44	P45	P46	P47	P48	P49
P51	P52	P53	P54	P55	P56	P57	P58	P59

Fig 2 The original image

P11	P13	P15	P17	P19
P31	P33	P35	P37	P39
P51	P53	P55	P57	P59

Fig 3 The twenty-five percent of original image

Generally we would resize image firstly before halftone process. According to P55, using Jarvis, Judice and Ninke error-extend filter, it can be calculated as follow:

P55 (n) = 1/48*E11 + 3/48*E13 + 5/48*E15 + 3/48*E17 +
1/48*E19 + 3/48*E31 + 5/48*E33+ 7/48*E35 + (4)
5/48*E37 + 3/48*E39 + 5/48*E51 + 7/48*E53

P55 (h) = Q (P55 (n)) (5)

We use our halftone process algorithm; it can also achieve resizing function. P55 becomes:

P55 (n) = 1/48*E33 + 3/48*E34 + 5/48*E35 + 3/48*E36 +
1/48*E37 + 3/48*E43 + 5/48*E44+ 7/48*E45 + 5
/48*E46 + 3/48*E47 + 5/48*E53 + 7/48*E54 (6)
= 1/48*E33 + 3/48*P34 + 5/48*E35 + 3/48*P36 + 1/48*E37 +
3/48*P43 + 5/48*P44+ 7/48*P45 + 5/48*P46 + 3/48*P47 +
5/48*E53 + 7/48*P54

Comparing the equation (4) (6). Although they have the same complexity of operation which has the equal of multiplier and addend, equation (6) possessed of resizing operation is provided with function and speed more than equation (4)..

Image Resizing Experiment Algorithm Contrast

In this section, we use grid-line image to experiment and compare algorithms such as Fig 4. The result of tradition sizing algorithm for resizing to twenty-five percent image size is shown Fig 5 which is provide with loss of characteristic lines in image. Even though the halftone process is still not repaired it which is shown in Fig 6.

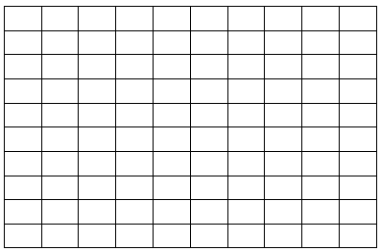

Fig 4 The original image

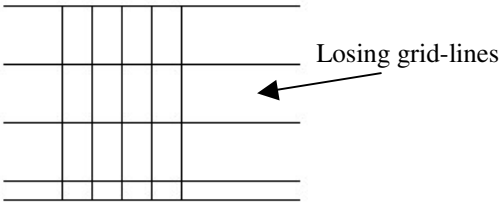

Fig 5 The resizing 25% of image with loss some lines

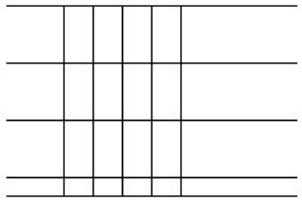

Fig 6 The halftone process image after resizing 25%

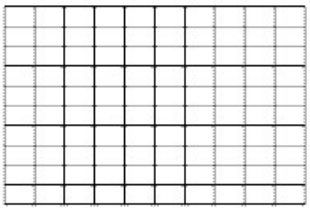

Fig 7 The 25% image size using our algorithm

We can know from Fig 5 that some grid-line disappear because reducing. Fig 6 just passes through halftone. However, we can observe Fig 7 to obtain two phenomena: Because of retaining some edge that holding some grid-line on. For the result of our algorithm has another image quality issue that the color is dimmisher than some neighbor. That is because the draw out pixel has been taken as error extending to near pixels and causing the halftone value over.

Dimmish improvement

In order to improve the dim image color and , we only extend error of draw out pixel to four neighbor pixels when using error-extend method. For example the P55 in Fig3, the equation becomes:

P55 (n) = 2/48*E33 + 10/48*E35 + 2/48*E37 + 5/48*E44+
7/48*E45 + 5/48*E46 + 10/48*E53 + 7/48*E54 (7)
= 2/48*E33 + 10/48*E35 + 2/48*E37 + 5/48*P44+
7/48*P45 + 5/48*P46 + 10/48*E53 + 7/48*P54

Also we use grid-line image to experiment. Fig 9 shows the image which has been pass through dimmish improvement.

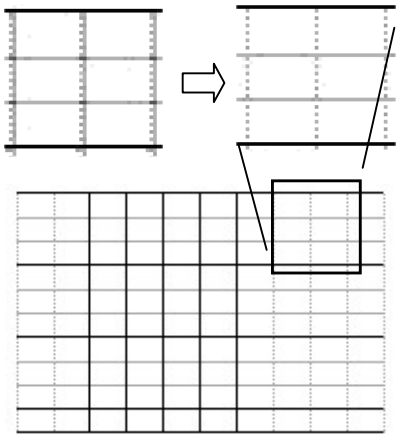

Fig 8 The dimmish improvement image (grid line)

Also we use Lena picture to show that the Fig 9(b) which has been calculated by equation (7) is brighter than Fig 9(a).

Fig 9(a) *Fig 9(b)*

Fig 9 The dimmish improvement image (Lena)

Hardware Implementation

For achieving digital image resizing algorithm, we implement it by hardware. Hardware design is in order to accomplish the image resizing algorithm to chip integration. The resizing hardware is composing of the enlarging module and the decreasing one. The both modules are performed in the role of upstream and downstream of data flow.

The resizing process must enlarge size to 400% from original image anyway and then reduce to any ratio what do you want in 400~25%. The data flow chart is shown below Fig10:

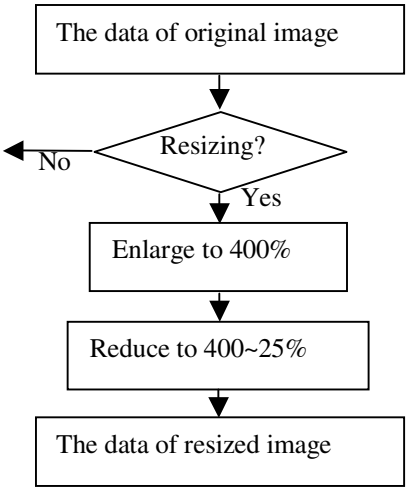

Fig 10 The data flow chart of resizing hardware module

Image Enlarging

A pixel must be expanded to the other three pixels on adjacent position for resizing ratio 400%. The expanded pixel data value is calculated by the original some neighbor pixels For example in Fig11, the expanded pixel of P33 are P34, P43 and P44 which can be calculating by P31, P33, P35, P37, P13, P53, P73 and P55.

P11	P13	P15	P17	P19
P31	P33 P34 P43 P44	P35	P37	P39
P51	P53	P55	P57	P59
P71	P73	P75	P77	P79

Fig 11 The expanded pixel image sketch

The equation is below for expanded pixel value:

$$P34\,(n) = (-P31+9*P33+9*P35-P37)/16$$
$$P43\,(n) = (-P13+9*P33+9*P53-P73)/16 \quad (8)$$
$$P44\,(n) = (P33+P35+P53+P55)/4$$

In the enlarging module, it would be enlarge input image size to 400%. Otherwise, it would be hold image quality and color smooth. So the enlarging algorithm should reference pixels which near the original pixel. We replace the value of pixel P31, P33, P35, P37, P13, P53, P73, P55 by A, B, C, D, E, F, G and H .The pixel P34, P43 and P44 is redefined symbol to Pright (Pr), Punder (Pu) and Pdiagonal (Pd). In other words, the equation (8) can be rewrite to (9)

$$Pr = (-A+9B+9C-D) \div 16$$
$$Pu = (-E+9B+9F-G) \div 16 \quad (9)$$
$$Pd = (B+C+F+H) \div 4$$

Fig 12 is the image enlarging chart. The signals output order is Br => Pr => Pu => Pd.

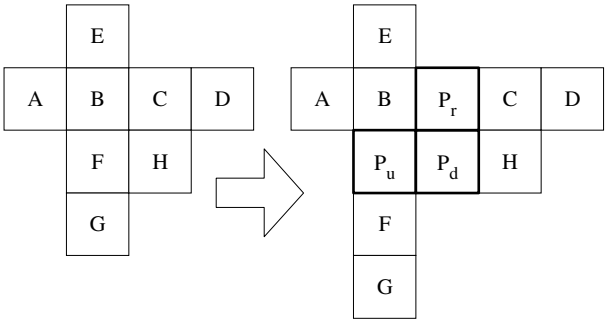

Fig 12 Pixel Enlarging diagram

Following describes the enlarging process. It divides three procedures: Pre-enlarging, Enlarging and enlarging output module. Fig13~Fig15 show those architectures.

The Pre-enlarging module receives the data of original image by way of DMA and continuously provides to Enlarging module. The data format of pre-enlarging module is a block type which process data independent others. The block size divides into two kinds which 28*28 pixels and 56*56 pixels. The enlarging module focuses on enlarging 400% the original image.

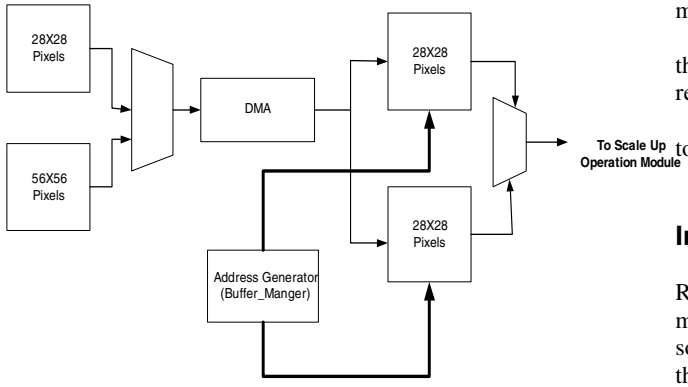

Fig 13 The pre-enlarging Module

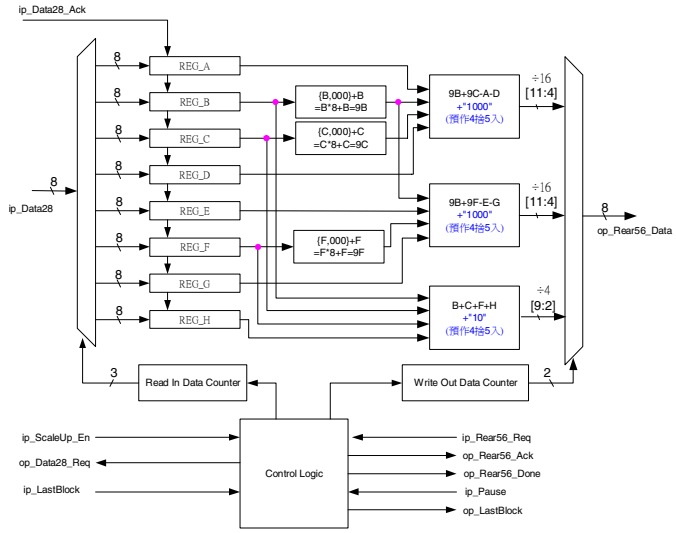

Fig 14 The enlarging Module

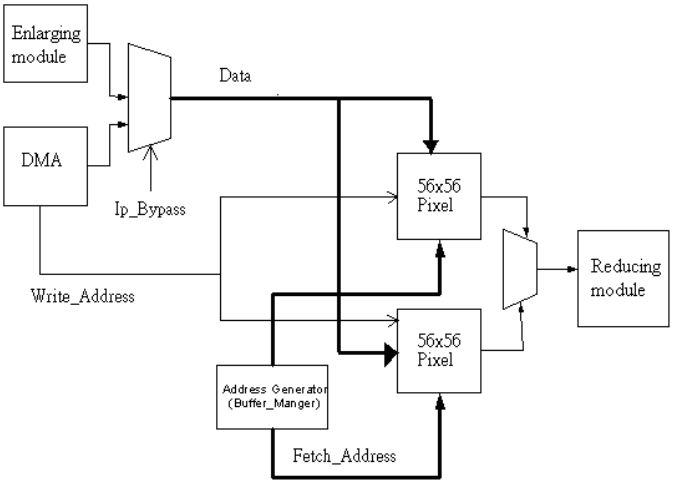

Fig 15 The enlarging output Module

The overall processes are:

F/W decides the image to enlarge or not. If we choice to enlarge image, pre-enlarging module sends data to enlarging module to produce enlarged data pixels.

The enlarged of data writes into the FPGA local memory from the enlarging module. According the access address from the reducing module to get the data from the local memory.

The local memory which is a double buffer structure switches to the other when it is used up.

Image Reducing

The image reducing module includes two processes: Smaller Resizing and DMA out buffer manager. The smaller resizing module reduces the image data which method will be abandoned some rows or columns according to a resizing ratio. Be diffused to the preserved pixel with some weighting by the abandoned ones, and this way can keep the characteristic of the original image. Of course some constrain is needed paid attention which the abandoned pixel must be discontinuous, in other words, at least one pixel must be preserved between two abandoned pixel.

In the details of checking every preserved pixel and executing reduced process in block base on four adjoining points which is shown in Fig 16. There are four kinds of situations need handling. The first situation is shown in Fig 16(a) which pass the pixel because it preserve all pixel in four adjoining ones. The second situation that has one abandoned pixel in four adjoining ones. The operation is shown in equation (10) which weighting value is 1/2. The third situation are two in four adjoining ones. The operation is shown in equation (11) which weighting value is 1/3. The fourth situation that has three abandoned pixels in 4 adjoining ones. The operation is shown in equation (12) which weighting value is 1/4.

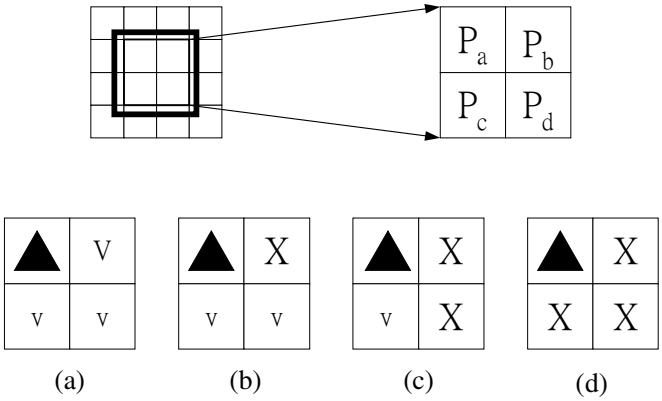

Fig 16 The four operational case in reduced process

In Fig 16, the three symbols are:
"▲"expresses the pixel is dealing with reducing process.
"v "expresses the pixel will be preserved in reducing process.
"X "expresses the pixel will be abandoned in reducing process.

$$Pa=(Pa + Pb)/2 \quad (10)$$
$$Pa=(Pa + Pb +Pd)/3 \quad (11)$$
$$Pa=(Pa + Pb + Pc +Pd)/4 \quad (12)$$

The smaller resizing architecture is shown in Fig 17.

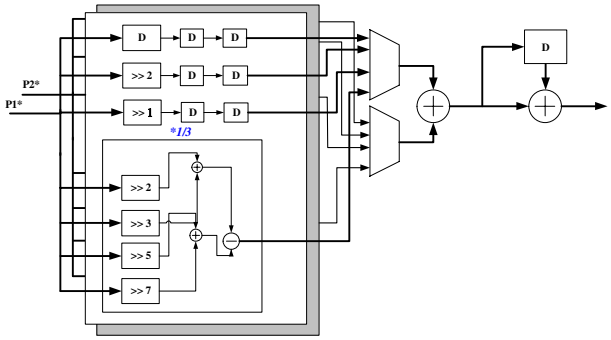

Fig 17 The smaller resizing hardware architecture

The reducing module DMA out buffer manager receives data from smaller resizing module and storage into internal buffer of FPGA. This result data is outputted by DMA channel.

Conclusion

An image resizing algorithm such as interpolation, filter, DCT and DFT are all complicated. In contrast with those, the linear weighted manipulation algorithm is simple and can be holding on high quality image by enlarging or reducing. The algorithm not only can eliminate dimmish image but also removing artifact which is produced by halftone. In order to implement image resizing algorithm into a chip, we design a simple architecture of enlarging and reducing operation hardware solution on the FGPA system platform.

Reference

[1] Saif alZahir, A Perceptually Perfect Image Resizing Scheme. Computer Science Department, UNBC 3333 University Way, PG, BC, V2N 429, CANADA, pp 83-84, Jan 2005.
[2] YoungSeo Park and HyunWook Park, Design and Analysis of an Image Resizing Filter in the Block-DCT Domain,DigitalObjectIdentifier10.1109/TCSVT.2003.819183, pp274-279, Feb 2004.
[3] Christian Hentschel, Stefan Schiemenz, High Quality, Low Complexity Image Scaler Suitable for Rational Factors, Brandenburg University of Technology, Cottbus, Germany, pp179-180, Jan 2006.
[4] M. Unser, A. Aldroubi, M. Eden, Enlargement or Reduction of Digital Images With Minimum Loss of Information, IEEE Trans. on Image Processing, pp. 247-258, March 1995.

Author Biography

Gian-Hung Liu received the M.S. degree in electrical engineering from National Taiwan University of Science and Technology, Taipei, Taiwan in 2004. He is currently working in Display Technology Center of Industrial Technology Research Institute, Hsinchu, Taiwan, as an associate Engineer. His current research areas include image processing, circuit design, IP design and printing system design.