EXTENDED INTELLIGENT EDGE-BASED LINE AVERAGE WITH ITS IMPLEMENTATION AND TEST METHOD

Yu-Lin Chang, Shyh-Feng Lin, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering, Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan

ABSTRACT

In this paper, an extended intellgient edge-based line average - EIELA is proposed. The EIELA could be used in intra-field de-interlacing to remove the jagged edges and the blurring effect of the bilinear interpolation or the traditional algorithms. The EIELA could change its performance dynamically according to the user's requirement. The proposed method can achieve the highest image quality of the traditional ELA if all the taps in the module are enabled, and it also possesses the ability to reduce the taps usage and instruction counts through the turn-off of some calculating sub-modules. The results and VLSI implementation show the proposed method achieve high image quality and low hardware complexity.

1. INTRODUCTION

De-interlacing becomes an important technique today because of the popularity of the progressive TV devices. While traditional SDTV video sequences are all in interlaced scanned format, a conversion must be done if we want to display the SDTV video on the progressive scanned devices(such as PDPs, LCD-TVs). If the de-interlacing is not done perfectly, defects like line crawling, edge-flicker and jagged effect would appear and make the viewer uncomfortable.

Two low complexity de-interlacing methods, BOB and Weave [1] , are commonly adopted in the software approach. BOB is an intra-field interpolation method, which uses a single field to reconstruct one progressive frame. However, the vertical resolution is halved and the image is blurred. Weave is a simple inter-field de-interlacing method directly combining two interlaced fields into one progressive frame. However, the line-crawling effect will occur in the motion areas.

Motion adaptive methods are commonly used in consumer electronics. The motion adaptive de-interlacing combines the advantages of both intra-field de-interlacing and inter-field deinterlacing. Lin [2] proposed a motion adaptive de-interlacing method with morphological motion detection to detect more accurate motion in fast motion video sequences. And he also proposed a four field horizontal motion adaptive de-interlacing method [3] to reduce the hardware complexity and achieve much more accurate motion detection.

Some motion compensated techniques have been presented to improve image quality. Sugiyama and Nakamura [4] proposed a method of motion-compensated adaptive interpolation. They used motion compensation and motion adaptive interpolation to reconstruct the missing field with the information obtained from the backward and the forward fields. Hilman [5] and Haan [6] proposed motion-compensated frame-rate conversion algorithms with interpolation to reduce the 3:2 pull-down artifacts. Chang [7] used global motion estimation to convert the interlaced video with camera panning, zooming, rotating to progressive one.

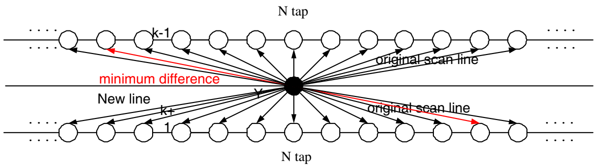


Fig. 1. N+N tap ELA Algorithm

However, the motion adaptive de-interlacing and the motion compensated de-interlacing method need a good intra-field interpolation or the motion areas and the missing areas cannot be interpolated very well. Kuo [8][9] proposed a 3+3 taps Edge-based Line Average algorithm to enhance the edge in the picture. Lee et al. [10][11] enhanced the 3+3 taps ELA algorithm to 5+5 taps and this modified ELA can be easily implemented on hardware. Yoo [12] proposed a much bigger ELA algorithm and extend the directional difference into direction vector difference to increase the directions that ELA can detect. Haan [13] proposed an Edge-Dependent De-Interlacing method which supports up to 7+7 taps ELA.

In this paper, a new edge-based line average(ELA) algorithm and its architecture, which can save the taps usage and achieve lower instructions requirement. The proposed N+N taps extended intelligent edge-based line average (EIELA) will be described in section 2. Its architecture and hardware implementation issues will be discussed in section 3. The simulation results in section 4 show the performance of the proposed method. At last, the conclusion remarks the proposed EIELA.

2. THE N+N TAPS EXTENEDED INTELLIGENT ELA ALGORITHM

2.1. The N+N taps Edge-based Line Average

The intra-field interpolation accounts a lot for the final image quality of de-interlacing. The most common intra-field interpolation is the bilinear interpolation. The bilinear interpolation of intra-field interpolation in de-interlacing is a two taps filter, which reference two pixels, each from the previous scanline and the next scanline. The result of the bilinear interpolation is to average these two pixels. The best of theoretical results(such as PSNR, MSE) of the intra-field interpolation is the bilinear interpolation, since bilinear interpolation minimizes the error and the MSE between the inter-

0-7803-8251-X/04/$17.00 ©2004 IEEE

polated field and the original field will be the smaller one. Nevertheless, the subjective view of the interpolated field by the bilinear interpolation is blurred. Blurred pictures may let people suffer uncomfortable visual effect. That's why the Edge-based Line Average (ELA) algorithm be proposed. As the computing ability for computers and ICs grow everyday, the taps number for the intra-field interpolation grows, too. The traditional ELA is a 3+3 taps filter. The "3+3" means it references three pixels from the previous scanline and three pixels from the next scanline. If there is a taps number limit for this algorithm, we should utilize the taps number to raise the performance to its extent.

The N+N taps ELA is shown as Fig.1. The "N+N" means it reference N pixels from the previous scanline and N pixels from the next scanline. Here we denote the pixel value of the interpolated pixel as $x(i, j)$, the (i, j) means the horizontal and the vertical position of the pixel in the de-interlaced frame. As for the bilinear interpolation, the interpolated pixel $x(i, j)$ should be the value $\frac{|x(i,j-1)+x(i,j+1)|}{2}$. If there is a N+N taps ELA filter shown as Fig.1, the first thing for the filter as the input signal comes in is to calculate the difference between the opposite pixels in each direction indicated by an arrow. We can divide the differences into three groups. The first group is the middle group, which contains only one direction, in which direction the opposite two pixels are $x(i, j-1)$ and $x(i, j+1)$. The second group is the left group, which contains $\frac{N-1}{2}$ directions and the difference of each direction can be denoted as following equation:

$$d_r(n) = \frac{|x(i+\frac{n-1}{2}, j-1) + x(i-\frac{n-1}{2}, j+1)|}{2}, n \in \{1, 3, 5, ...\} \tag{1}$$

And the third group is the right group, which also contains $\frac{N-1}{2}$ directions and the difference of each direction can be denoted as following equation:

$$d_l(n) = \frac{|x(i-\frac{n-1}{2}, j-1) + x(i+\frac{n-1}{2}, j+1)|}{2}, n \in \{1, 3, 5, ...\} \tag{2}$$

Then we should find a direction with the smallest difference, that is :

$$(n', p) = Arg(min(d_l(n), d_r(n))), n \in \{1, 3, 5, ...\} \tag{3}$$

where n' denote the found direction with a smallest difference, and p means whether the left or the right part the difference is in.

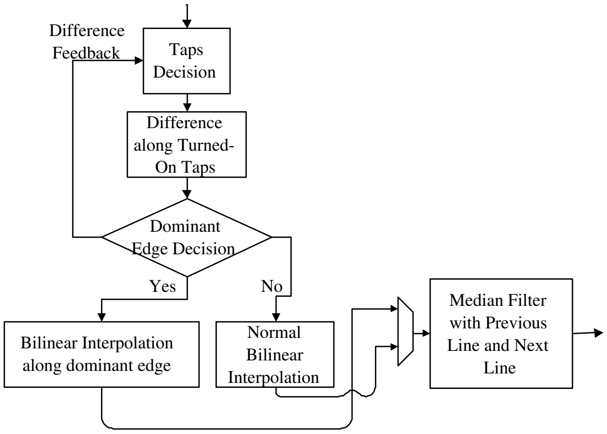
The next thing we have to do is to check if the chosen direction is the dominant edge. If the chosen direction is on the right side, the difference of the chosen direction is $d_r(n')$, and it should conform to the following equation:

$$|d_r(n') - d_l(n)| < \theta, n \in \{1, 3, 5, ...\} \tag{4}$$

where θ is a given threshold which help to check the dominant edge, usually in the range of 10∼30. Finally, if the chosen direction is the dominant edge, then the final output should be the bilinear interpolation along the dominant edge, else it should be the bilinear interpolation of $x(i, j-1)$ and $x(i, j+1)$.

2.2. Extended Intelligent Edge-based Line Average - EIELA

The proposed Extended Intelligent Edge-based Line Average, which is called EIELA, aims at the reduction of taps usage and the instruction counts, which can save more power and run fast if the algorithm is designed properly. And the performance of the proposed method should not drop too much while comparing to the best N+N taps ELA algorithm.



Difference Feedback
Taps Decision
Difference along Turned-On Taps
Dominant Edge Decision
Yes
No
Bilinear Interpolation along dominant edge
Normal Bilinear Interpolation
Median Filter with Previous Line and Next Line

Fig. 2. Extended Intelligent ELA Flow Chart

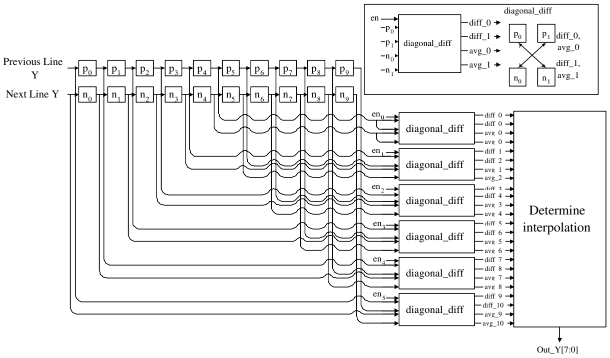


Fig. 3. VLSI Architecture for EIELA

The proposed EIELA is shown in Fig.2. The differences of the EIELA and the N+N taps ELA algorithm are the taps decision part and the difference feedback. Before the N+N taps ELA, the EIELA would decide which "N" the N+N taps ELA will need. The difference of the dominant edge direction of the previously processed pixel is fed back into the taps decision part. The EIELA algorithm starts with $N = 1$. If the difference is greater than a certain threshold, named "difference threshold", it means that the previous N+N taps ELA cannot find a real dominant edge since the texture is complex or there are many edges in the field. In this condition, the "N" should be increased to handle the more complex images in the future. Else if the difference is smaller than the difference threshold, it means the texture in the previously processed pixel is smooth, then the "N" should be decreased to reduce the taps usage and the instruction load will be lower, too.

3. VLSI ARCHITECTURE

As described in the section of simulation results, the best choice of the maximal taps number and the difference threshold are 11+11 taps and 2. So the VLSI architecture of the proposed EIELA should be like Fig.3. $P_0 - P_9$ and $N_0 - N_9$ denote the 10 registers of the previous scanline and those of the next scanline. Each register passes its data to the next register and the "diagonal_diff" module. Every "diagonal_diff" module reads four pixels and one enable signal. The four pixels are the four taps shown in Fig.3, then the "diagonal_diff" module calculates the difference and the average of the two directions constituted by the four input pixels. The difference and the average are passed into the "Determine Interpolation" module to decide which difference is the smallest difference and which direction is the dominant edge.

The "Determine Interpolation" module consists of two comparator tree. One is for the left side differences, another is for the right side differences. After the comparison of the differences of the left side and the right side, there is another comparator tree to choose the minimal difference between the minimal differences of the left side and the right side and the middle difference. After finding the minimal difference direction, the dominant edge decision is implemented by two comparator tree, too. After comparing with the opposite side comparator tree, the final dominant edge direction is outputted with its difference and average. The difference will be fed back into a "taps_decision" module, and it will change the taps number according to the previous dominant edge difference value and output $en_0 - en_5$ to the "diagonal_diff" modules, and the corresponding register behind the current pixel register can be turned off by gated clock. While there is only 1+1 taps ELA needed, $en_1 - en_5$, $P_6 - P_9$ and $N_6 - N_9$ can be turned off. While the texture is complex, the taps should be all on, then the "taps_decision" will output $en_0 - en_5$ as on to turn on all the "diagonal_diff" modules, and all the register would be turned on.

4. SIMULATION RESULTS

4.1. Hardware Simulation

Before the hardware implementation, we should check what number should be the maximal taps number and what the difference threshold for the taps_decision module should be. Table 1 shows the usages of the total taps and the instruction counts for several different conditions. The PSNR is tested with a rotating rectangle sequence shown as Fig.6. The difference threshold is tested with several values. However, the image quality degrades with larger difference threshold. After all, the EIELA with maximal 11+11 taps and difference threshold 0 can achieve high PSNR, low taps usage and low instruction counts. Comparing to the traditional 11+11 taps ELA in the second row of Table 1, the taps usage of EIELA is ten times smaller than that and the instruction has been reduced to 64% without much performance degradation. The VLSI implementation has been done with Artisan TSMC 0.25μm cell library. The total gate count for the EIELA is 5644, and the specification for this EIELA module is 1920×1080i, 60fps.

4.2. PSNR Comparison and New Test Sequence

The PSNR comparison of common video test sequences is shown in Fig.4. However, as mentioned above, the PSNR of common video test sequences doesn't represent the real visual quality of a image. The contrast is much more important for people, so we made a test sequence which is a rotating blue rectangle in a black space with D1 progressive size. The rectangle would rotate to every angle and its edge will tilt to every angle. The test sequence is decimated into interlaced format and will be deinterlaced by the intra-field interpolation. If the intra-field interpolation is done perfectly, the result should be the same as the original D1 progressive video sequence and the PSNR should be as high as possible. The resulting PSNR comparison of the rectangle video test sequence is shown in Fig.5. It shows that the traditional 11+11 taps ELA achieves highest PSNR value. As for our hardware implementation, if the taps_decision module always turns on $en_0 - en_5$, the performance would be the same as the traditional 11+11 taps ELA. So the user can decide how high the image quality he needs. If the power issue is concerned, the taps_decision module may help to reduce the power consumption. If the image quality issue is concerned, the taps_decision module should enable all the diagonal_diff modules to achieve highest image quality.

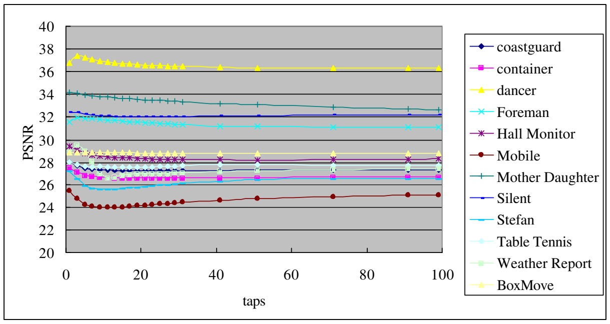


Fig. 4. PSNR Comparison of common video test sequences

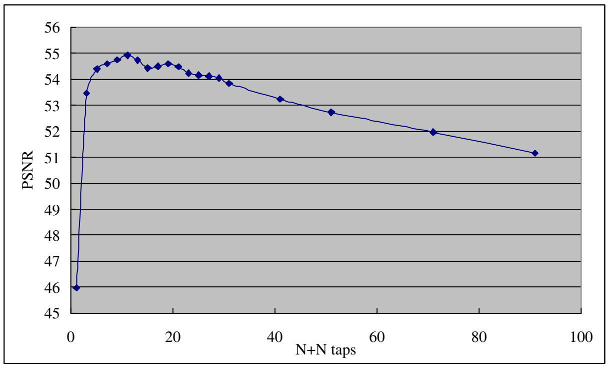


Fig. 5. PSNR Comparison of the Rectangle video test sequence

4.3. Subjective View

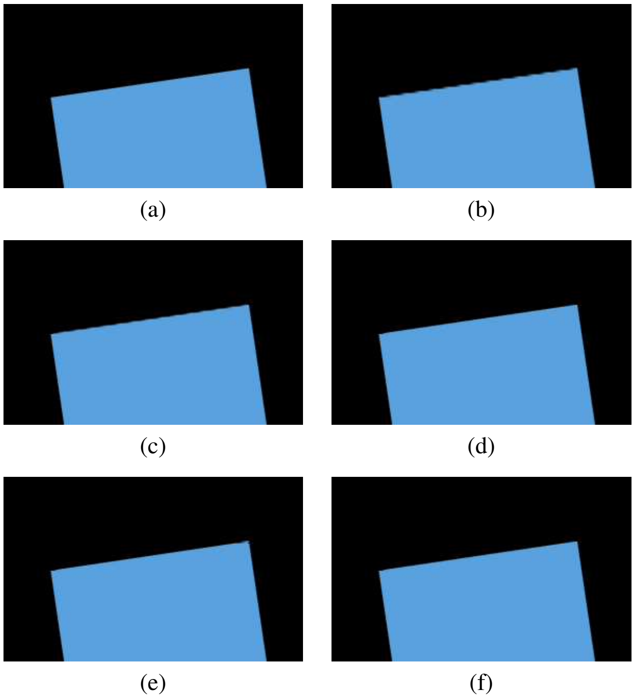
The subjective view results are shown in Fig.6. The original rotating rectangle is shown as Fig.6(a), and Fig.6(b)(c)(d)(e)(f) are the results of the bilinear interpolation, the 5+5 taps ELA, the 11+11 taps ELA, the 31+31 taps ELA, and the proposed EIELA. Jagged edges appear at the boundary of the rectangle if de-interlaced by the bilinear interpolation and the 5+5 taps ELA. However, the subjective views look alike with the 11+11 taps ELA, the 31+31 taps ELA, and the EIELA. So the proposed EIELA could achieve great subjective view quality.

5. CONCLUSION

An Extended Intelligent Edge-based Line Average (EIELA) algorithm and its VLSI module implementation for intra-field deinterlacing are proposed in this paper. The EIELA algorithm can

Table 1. PSNR, Taps and Instructions comparison with different max taps and difference threshold

Max Taps/Line	Frame Count	Time(sec)	PSNR(dB)	Diff. thres.	Total Taps Used	Instructions (MIPS) D1-60fps
All 1	100	18	45.9683		17316000	3550.9866
All 11	100	20	55.0328		189033000	5698.3458
11	100	19	51.0476	2	17714599	3667.7148
11	100	18	51.2430	0	17734267	3668.0922
21	100	19	51.1898	2	17838269	3677.3058
21	100	18	51.3858	0	17865103	3677.6832
31	100	19	51.1661	2	17948262	3687.0516
31	100	19	51.3626	0	17977627	3687.4290



(a) (b) (c) (d) (e) (f)

Fig. 6. Subjective View Results (a) Original Rectangle (b) Rectangle with bilinear interpolation (c) Rectangle with 5+5 taps ELA (d) Rectangle with 11+11 taps ELA (e) Rectangle with 31+31 taps ELA (f) Rectangle with EIELA

achieve best image quality to the extent of the traditional ELA if all the 11+11 taps are enabled. If there is power issue concerning, the taps_decision module would reduce the taps usage and the instruction counts of the EIELA module. The specification for its VLSI implementation is 1920×1080i, 60fps, and the gate count is small. The PSNRs and subjective views show that the proposed method can achieve high image quality for intra-field deinterlacing application.

6. REFERENCES

[1] "http://nickyguides.digital-digest.com/index.htm," .

[2] Shyh-Feng Lin, Yu-Lin Chang, and Liang-Gee Chen, "Motion adaptive interpolation with morphological operation and 3:2 pull-downed recovery for de-interlacing," *IEEE International Conference on Multimedia and Expo*, Aug. 2002.

[3] Shyh-Feng Lin, Yu-Lin Chang, and Liang-Gee Chen, "Motion adaptive de-interlacing by horizontal motion detection and enhanced ela processing," *IEEE International Symposium on Circuits and Systems*, May 2003.

[4] Kenju Sugiyama and Hiroya Nakamura, "A method of de-interlacing with motion compensated interpolation," *IEEE Transactions on Consumer Elec.*, vol. 45, no. 3, pp. 611–616, Aug. 1999.

[5] Kevin Hilman, Hyun Wook Park, and Yongmin Kim, "Using motion-compensated frame-rate conversion for the correction of 3:2 pulldown artifacts in video sequences," *IEEE Transactions on Circuits Syst. Video Technol.*, vol. 10, no. 6, Sept. 2000.

[6] Gerard de Haan and E.B. Bellers, "De-interlacing of video data," *IEEE Transactions on Consumer Electronics*, vol. 43, pp. 819–825, Aug. 1997.

[7] Yu-Lin Chang, Ching-Yeh Chen, Shyh-Feng Lin, and Liang-Gee Chen, "Motion compensated de-interlacing with adaptive global motion estimation and compensation," *IEEE Conference on Image Processing*, Sept. 2003.

[8] Chung J. Kuo, Ching Liao, and Ching C. Lin, "Adaptive edge-based interpolation for scanning rate conversion," *IEEE International Conference on Accoustics, Speech, and Signal Processing*, vol. 4, pp. 2120–2123, May 1996.

[9] Chung J. Kuo, Ching Liao, and Ching C. Lin, "Adaptive interpolation technique for scanning rate conversion," *IEEE Transactions on Circuits and Systems for video technology*, vol. 6, no. 3, June 1996.

[10] Ho Young Lee, Tae Min Bae, Jin Woo Park, Kun Woen Song, and Yeong Ho Ha, "Efficient scan conversion with edge enhancement and its vlsi architecture," *IEEE TENCON*, 1999.

[11] Ho Young Lee, Jin Woo Park, Tae Min Bae, Sang Um Choi, and Yeong Ho Ha, "Adaptive scan rate up-conversion system based on human visual characteristics," *IEEE Transactions on Consumer Electronics*, vol. 46, Nov. 2000.

[12] Hoo Yoo and Jechang Jeong, "Direction-oriented interpolation and its application to de-interlacing," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 4, Nov. 2002.

[13] Gerard de Haan and Rogier Lodder, "De-interlacing of video data using motion vectors and edge information," *IEEE International Conference on Consumer Electronics*, June 2002.