MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Fast Adaptive Fuzzy Post-Filtering for Coding Artifacts Removal in Interlaced Video

Yao Nie, Hao-Song Kong, Anthony Vetro and Kenneth Barner

TR2005-018   December 2005

Abstract

This paper presents a new method using fuzzy filtering to remove the coding artifacts in compressed video. The method takes interlaced video format into consideration and processes each field format into consideration and processes each field separately. For deblocking, a 1-D fuzzy filter with different window size is used to remove the horizontal and vertical blocking artifacts respectively. For deringing, each 8 X 8 block in a field is first classified into one of the four categories, i.e., strong edge, weak edge, texture and smooth blocks. According to each blocḱs type and the neighboring blocḱs type, the spread parameter of a 2-D fuzzy filter is adaptively decided and the filter is applied. To speed up the proces, the fuzzy filter weights are generated using a piecewise linear membership function instead of the conventional Gaussian function. The experimental results show that the proposed method has better detail preservation and lower computational costs than our previous method. It achieves comparable deblocking and superior deringing performance to the MPEG-4 standard method at similar computation costs.

IEEE, ICASSP 05

FAST ADAPTIVE FUZZY POST-FILTERING FOR CODING ARTIFACTS REMOVAL IN INTERLACED VIDEO

Yao Nie[2], Hao-Song Kong[1], Anthony Vetro[1], Huifang Sun[1] and Kenneth E. Barner[2]

[1]Mitsubishi Electric Research Labs
201 Broadway, Cambridge, MA 02139
hkong@merl.com, avetro@merl.com, hsun@merl.com

[2]Department of Electrical and Computer Engineering
University of Delaware, Newark, DE 19716
ynie@ee.udel.edu, barner@ee.udel.edu

ABSTRACT

This paper presents a new method using fuzzy filtering to remove the coding artifacts in compressed video. The method takes interlaced video format into consideration and processes each field separately. For deblocking, a 1-D fuzzy filter with different window size is used to remove the horizontal and vertical blocking artifacts respectively. For deringing, each 8×8 block in a field is first classified into one of the four categories, i.e., strong edge, weak edge, texture and smooth blocks. According to each block's type and the neighboring block's type, the spread parameter of a 2-D fuzzy filter is adaptively decided and the filter is applied. To speed up the process, the fuzzy filter weights are generated using a piecewise linear membership function instead of the conventional Gaussian function. The experimental results show that the proposed method has better detail preservation and lower computational costs than our previous method. It achieves comparable deblocking and superior deringing performance to the MPEG-4 standard method at similar computation costs.

1. INTRODUCTION

Coding artifacts removal is an important issue in many digital video applications. The most prominent coding artifacts in highly compressed video are the blocking and ringing artifacts, for which~~,~~ many post-filtering methods have been proposed [1]~[5]. These methods, however, either have high computational complexity, or cannot remove both types of artifacts successfully. To address the problem, in our previous work [6], a post-filtering method based on the block classification and fuzzy filtering was proposed to remove the blocking and ringing artifacts. In this method, an adaptive 1-D linear filter is applied along the boundaries of all 8×8 blocks for deblocking. Then each block is classified into *edge* or *non-edge* block, and a 2-D fuzzy filter is applied to edge blocks only for deringing. The method can effectively remove the blocking and ringing artifacts while preserving the strong edges well. However, computing the fuzzy filter weights needs considerable computation. This makes the method relatively slow when compared with the MPEG-4 standard algorithm [5], which is known as the fastest in the literature. Moreover, fixed spread parameter of the fuzzy filter leads to blurring of some image fine details. Hence, we are motivated to develop a fast and adaptive fuzzy filtering method to improve the artifacts removal performance. In addition, noting that in interlaced video, a single frame may be coded using frame-based and field-based coding jointly, making the appearance of the artifacts more complicated, we develop a new filtering scheme which can handle this complexity.

In the proposed method, to reduce the overall computational complexity, a piecewise linear function is designed to replace the Gaussian function for generating the fuzzy filter weights. To achieve better artifacts removal and better detail preservation, the two fields of each video frame are processed separately. A 1-D fuzzy filter with different window size is applied to remove the horizontal and vertical blocking artifacts respectively. The 8x8 blocks in a field are finely classified into *strong edge, weak edge, texture and smooth* categories, and a 2-D fuzzy filter with adaptive spread parameter is applied to remove the ringing artifacts. The neighboring block information is also utilized to preserve image details. An overview diagram of the proposed method is shown in Fig. 1.

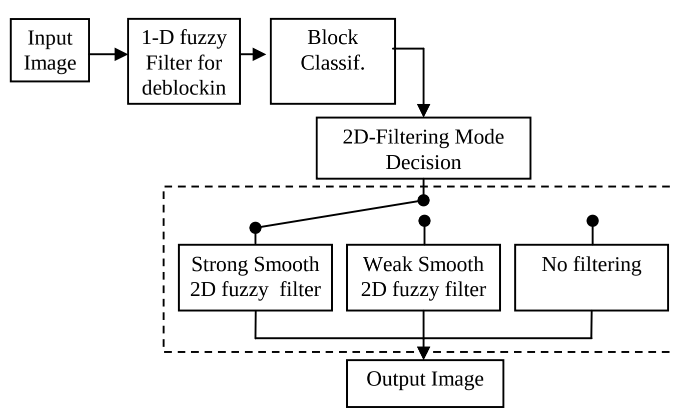


Fig.1 Overview diagram of the proposed method.

2. FAST FUZZY FILTERING

The fuzzy filter and the new method for obtaining the filter weights are introduced in this section. The fuzzy filter utilized in our method is developed by the authors in [7] based on the *fuzzy transformation theory* and is defined as

$$y = \frac{\sum_{j=1}^{N} x_j w_j}{\sum_{j=1}^{N} w_j} = \frac{\sum_{j=1}^{N} x_j \cdot \exp[-(x_c - x_j)^2 \big/ 2\xi^2]}{\sum_{j=1}^{N} \exp[-(x_c - x_j)^2 \big/ 2\xi^2]}, \quad (1)$$

where N is the window size, x_j are the inputs, and w_j are the filter weights, $x_c \in \{x_1, x_2, ...x_N\}$ is the center pixel in the window, ξ is referred to as the spread parameter. Note that larger ξ leads to stronger smoothing effects. Since the fuzzy filter weights are adaptive in nature to the local feature, it can preserve the image strong edges very well while removing the artifacts. However, obtaining the fuzzy filter weights requires evaluation of the Gaussian function, which is computationally expensive. Therefore, we design a piecewise linear function to approximate the Gaussian function

$$\mu_L(x) = \begin{cases} 1, & 0 \le x \le (2-e^{0.5}) \cdot \xi \\ -e^{-0.5}\xi^{-1}x + 2e^{-0.5}, & (2-e^{0.5}) \cdot \xi < x < 2\xi, \\ 0, & x \ge 2\xi \end{cases} \quad (2)$$

so that $w_j = \mu_L(|x_c - x_j|)$. The function is decided by a single parameter ξ which corresponds to the spread parameter and controls the fuzzy filter smoothing capability. Note that the non-constant part of the function represents the tangent line of the Gaussian curve at the point where the second derivative of the Gaussian curve is zero. This replacement significantly reduces the computation without degrading the filtering performance.

3. DEBLOCKING

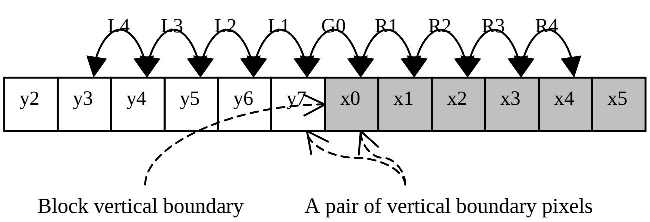


Fig.2 Detect the vertical boundary gap in a row across the block vertical boundary.

Considering that a frame in an interlaced video can be composed of two fields having drastically different contents, it is desirable to filter each field separately to remove the blocking artifacts. However, due to the joint frame-based and field-based coding, the horizontal blocking artifacts may not only appear along the horizontal boundary of each 8×8 block, but also along the center horizontal line across the block. Thus separate filtering schemes are needed for the removal of the vertical and horizontal blocking artifacts.

The deblocking is performed first in the vertical direction and followed by the horizontal direction. Since the fuzzy filter has great strong edge preserving property, simple blocking artifacts detection is employed along the vertical boundary of each 8×8 block in the input field, as shown in Fig. 2. In each row, the difference between two boundary pixels, *x0* and *y7*, is calculated and denoted by *G0*, i.e., *G0*=|*x0*-*y7*|. Then the difference between each pair of adjacent pixels on the left and right-hand side of the block boundary are also calculated and denoted by *Li* and *Ri* (*i=1,2,..4*), respectively. If *MAX(L1,L2,L3,L4)<G0* (3), or *MAX(R1,R2,R3,R4)<G0* (4), we claim that a *boundary gap* is detected and the current row is marked. After checking the 8 pairs of boundary pixels along the vertical boundary of the block, if the number of the boundary gaps exceeds a predetermined threshold TH=1, a blocking artifacts is claimed. The 1-D fuzzy filter with $\xi = 30$ and window size 5 is applied to {*x0* ,*y7*, *y6*} or {*y7*, *x0*, *x1*} along the marked rows according to whether Eqn.(3) or Eqn.(4) holds.

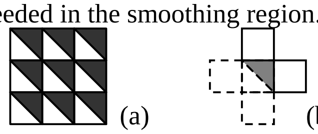
In the horizontal direction, blocking artifacts is detected not only along the horizontal boundary of each 8×8 block but also the horizontal centerline across the block, using a similar method to the one used in the vertical direction. Considering that the field vertical resolution is only half of the original image, finer detection is needed to avoid excessive smoothing. Therefore, besides the condition that the number of gaps along the checked horizontal line exceeds TH=5, the four, i.e., the immediate upper-left, upper-right, bottom-left, and bottom-right, neighboring vertical boundaries are checked. If at least two of them have more than 5 boundary gaps, or at least one has 8 boundary gaps, a horizontal blocking artifact is claimed. Then a 1-D fuzzy filter with a smaller window size 3 is applied.

4. DERINGING

In our previous method, the ringing artifacts are detected by finding the edge blocks since they usually occur around the image edges. All edge blocks are filtered by a 2-D fuzzy filter with fixed spread parameter. However, weak edges yield minor ringing artifacts and are more vulnerable to blurring effects when compared with strong edges, therefore, different edges should be treated differently. Moreover, granular artifacts may occur when the quantization step is small, which can also be observed in texture area. Hence, in the proposed method, each 8x8 blocks in a field is classified into one of the four categories, i.e., strong edge, weak edge, texture and smooth blocks. This is done by computing the standard deviation (STD) in a 3×3 window around each pixel and comparing the maximum STD in each 8x8 block with a set of predetermined thresholds as follows:

$$MaxSTD \begin{cases} \in [40, +\infty) & \Rightarrow \quad StrongEdgeBlock \\ \in [20, 40) & \Rightarrow \quad WeakEdgeBlock \\ \in [10, 20) & \Rightarrow \quad TextureBlock \\ \in [0, 10) & \Rightarrow \quad SmoothBlock \end{cases}.$$

Then a 2-D fuzzy filter with large spread parameter $\xi = 20$ is applied to the strong edge blocks, and a small spread parameter $\xi = 10$ is applied to the weak edge blocks. Note that filtering in a strong edge block can be saved when it is surrounded by strong edge blocks. This is because the ringing artifacts do not appear prominent in a large region of strong edges due to masking effects. Secondly, since minor artifacts only appear prominent in a smooth region, weak edge blocks need to be filtered only when smooth blocks are in the neighborhood. Fig. 3 shows the above-mentioned two cases. Filtering in the texture blocks is optional and can be turned on/off by the user according to the video compression rate. Finally, no filtering is needed in the smoothing region.



(a) (b)

Fig. 3 (a) A strong edge block surrounded by 8 strong edge blocks. (b) A weak edge block with at least 2 neighboring smooth blocks.

5. EXPERIMENTAL RESULTS

Four interlaced sequences are tested to show the performance of the proposed method. The resolution of Mobile is 720×576, and the resolution of the other three sequences is 720×480. The number of frames in each sequence ranges from 150~250. A 3×3 filter window is applied. The total processing time for each sequence using different methods is shown in Table I, where the proposed method consumes only 50% processing time of our pervious method and is comparable to that of the MPEG-4 standard method. Due to space constrain, only parts of the Mobile sequence are shown in Fig. 4 to demonstrate the visual results of the test method. All the samples, especially those in Column A, show that the proposed method achieves the same deringing performance as the previous method and evidently outperforms the MPEG-4 standard method. Column B shows that the proposed method performs similarly to the MPEG-4 method in deblocking. Although less smooth in certain areas, it preserves the red ball surface texture better. Finally, Column C shows that the proposed method has better detail preservation than the previous method, as can been seen in the little pigs' faces.

Table I. Total Processing Time Comparison (seconds)

Methods	Mobile	Soccer	Singers	Harbor
MPEG4	58.50	14.25	13.64	12.41
Previous	123.78	32.06	31.05	34.44
Proposed	68.35	16.70	15.81	15.75

6. CONCLUSION

In this paper, a new method for coding artifacts removal in compressed video is proposed. There are three major contributions. First, a fast fuzzy filtering method employing piecewise linear function to obtain the filter weights is developed. It greatly reduces the computational cost without influencing the filtering performance. Secondly, an adaptive fuzzy filtering scheme based on block classification and neighboring block information is developed which improves the detail preservation and further reduces the computation. Finally, a new deblocking scheme is developed, taking interlaced video format into consideration. The overall performance of the method is comparable to the MPEG-4 standard algorithm in terms of both image quality and processing time, with evident superiority in deringing. It can serve as an excellent alternative to the standard method, especially in moderately compressed video applications such as HDTV.

REFERENCES

[1] C. B. Wu, B. D. Liu, and J. F. Yang, "Adaptive postprocessors with DCT-based block classifications," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 5, pp. 365-375, May. 2003.

[2] ISO/IEC 14496-2:2001, "Information Technology – Generic Coding of Audio-Visual Objects – Part 2: Visual," *2nd Ed., Appendix F3 – Post – processing for coding noise reduction*, 2001.

[3] Z. Xiong, M. T. Orchard, and Y. Q. Zhang, "A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 7, No. 2, pp. 433-437, Aug. 1997.

[4] T. Chen, H. R. Wu and B. Qiu, "Adaptive post-filtering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 11, No. 5, pp. 594-602, May 2001.

[5] S. D. Kim, J. Yi, H. M. Kim and J. B. Ra, "A deblocking filter with two separate modes in block-based video coding," *IEEE Tran. On Circuits and System for Video Technol*, vol. 9, No. 1, Feb., 1999.

[6] H.-S. Kong, Y. Nie, A. Vetro, H. Sun and K. E. Barner, "Coding artifacts reduction using edge map guided adaptive and fuzzy filtering," *IEEE ICME04*, Taibei, Taiwan, June 2004.

[7] Y. Nie, and K. E. Barner, "Fuzzy transformation and its applications," *IEEE ICIP, Barcelona, Spain, Sept., 2003*

Fig. 4. The visual results of parts of the Mobile sequence generated by the tested methods. (1) The input images. (2) The output of the MPEG-4 standard method. (3) The output of our previous method. (4) The output of the proposed method. Note that (3) and (4) use the same deblocking algorithm proposed in the paper since the previous deblocking method only applies to progressive videos.