RESEARCH ARTICLE

Editorial Process: Submission:05/05/2018 Acceptance:07/24/2018

The Architectural Design of Deblocking Filter for Image Enhancement in the Diagnosis of Cancer

Neelu Kumari K S[1], Vijayabaskar S

Abstract

Objective: This method is to reduce the risk of imprecise diagnosis associated with poor-quality CT images, this paper presents a new technique designed to enhance the quality of medical CT images. The main objective is to improve the appearance of CT images in order to obtain better visual interpretation and analysis, which is expected to ease the diagnosis process. The proposed technique involves applying a deblocking filter is to enhance the visual nature of a picture by diminishing the blocking artifacts. The appearance of a picture isn't clear while an antique happens. The proposed deblocking filter calculation gives a strategy to expel the ancient rarities by smoothing the sharp edges of a picture. **Methods:** With a specific goal to lessen the quantity of information access, multifaceted nature and consequently to upgrade the proficiency, a six-staged pipelined structure for picture pixels are proposed. Besides, to enhance the subjective and target nature of a picture the deblocking filter performs identification of the antique at the coded square limits and weakens them by applying a chose filter. **Result:** The proposed algorithm is implemented in HDL using Xilinx FPGA. The input image is converted into decimal pixel values using Matlab and this value is used as the input in HDL. The proposed algorithm is compared with other blocking algorithms. **Conclusion:** To design an effective deblocking filter with low cost, low complexity and high intensity, pipeline based systems are used. In addition to that the number of memory accesses and timing efficiency also be reduced using this method. The deblocking filtering operations can also easily perform in parallel on multiple processors by using six-stage of pipelined, two-line deblocking filter. The parameter mean, variance, standard deviation, resolution, contrast and PSNR values are compared with the previous method. Hence it shows the implementation of deblocking filter using pipelining is more efficient than others.

Keywords: Deblocking filter- blocking artifacts- data access- efficiency- pipelining

Asian Pac J Cancer Prev, **19 (9)**, 2631-2635

Introduction

A deblocking filter is connected to decoded packed picture to enhance the visual quality by diminishing the blocking antiquities. Blocking artifacts is visual discontinuities introduce in a picture. This blocking curio is a standout amongst the most irritating issues happens at the every limit of a picture, which corrupts the nature of a picture. Deblocking filter expands the picture quality by applying a specific filter to smoothen the sharp edges on the limits. The imperative point in outlining of deblocking filter is to choose whether to filter has to be applied and the separation depends on the limit quality of a picture. This filter works on the edges of each 4x4 or 8x8, which changes obstruct in the luma and chroma planes of a picture. Every little edge of the block is relegated a limit quality macro- block limit, bury and intra coding of the squares. The deblocking filter is an exceptionally effective filter that alters its quality relying on pressure method of a full scale block (Intra or Inter), the quantization parameter and the pixel esteems.

A Deblocking filter based on HEVC entropy coding algorithm was proposed by (Pourazad et al., 2012). To establish the Joint Collaborative Team on Video Coding (JCT-VC) the objective is to develop a new high-performance video coding standard. Since then, JCT-VC has put a considerable effort toward the development of a new compression standard known as the high-efficiency video coding (HEVC) standard, with the aim to significantly improve the compression efficiency compared with the existing H.264/AVC high profile.

An efficient parallel architecture for the adaptive deblocking filter in H.264/AVC video coding standard was proposed by (Chen and Chen, 2005). They use six forwarding shift register arrays(of which each contains 4x4 8-bit shift registers) with two transposing operations and two sets of filter operation(each set contains four edge filter operations) to support simultaneous processing of the horizontal and vertical filtering. The proposed architecture is called "parallel Filtering Architecture(PFA)". Moreover, the number of total memory references is reduced by 63% and 25% respectively compared to the basic and advanced

*Department of Electronics and Communication Engineering, P.A. College of Engineering and Technology, Pollachi, India. *For Correspondence: neeluvijay@gmail.com*

architectures of the previous proposals.

A three-step framework considering task-level segmentation and data-level parallelization to efficiently paralleize the deblocking filter was proposed in (Zhang et al., 2012). First, they review the entire deblocking filter process 4x4 block edge-level and divide it into two parts:1) boundary strength computation(BSC) and 2) edge discrimination and filtering (EDF), which increases the parallelism. Then, they apply the Markov empirical transition probability matrix and Huffman tree (METPMHT) to the BSC, which alleviate the load imbalance problem.

The compression capability of several generations of video coding standards is compared by means of peak signal-to-noise ratio (PSNR) and subjective testing results. A unified approach is applied to the analysis of designs, including H.262/MPEG-2 video, H.263, MPEG-4 Visual, H.264/MPEG-4 Advanced video coding (AVC), and High Efficiency Video Coding (HEVC) was proposed by (Ohm et al., 2012. The HEVC design is shown to be especially effective for low bit rates, high resolution video content, and low-delay communication applications. The measured subjective improvement somewhat exceeds the improvement measured by the PSNR metric.

Norkin et al., (2012) describes the in-loop filter used in the upcoming High efficiency video coding(HEVC) standard to reduce visible artifacts at block boundaries. The deblocking filter performs detection of the artifacts at the coded block boundaries and attenuates them by applying a selected filter which is discussed in (Ohm et al., 2012). Compared to the H.264/AVC deblocking filter, the HEVC deblocking filter has lower computational complexity and better parallel processing capabilities while still achieving significant reduction of the visual artifacts.

Materials and Methods

The proposed deblocking filter calculation gives a strategy to expel the artifacts by smoothing the sharp edges of the picture. To lessen the quantity of information access, complexity and to improve the effectiveness, a six-staged pipelined structure for picture pixels is proposed. Moreover, to enhance the subjective and target nature of a picture the Deblocking filter plays out the identification of the antiques at the coded piece limits and weakens them by applying a chose filter. The high-level architectural overview of the proposed Deblocking Filter along with the re-organization flow of the RFU data structure is comprised of two major processing elements, Restructure Element (RE) and Filter Element (FE), with a central controller as shown in Figure1.

The block diagram of the proposed deblocking filter is shown in Figure 2. It consists of input frame unit, controller, current buffer, edge buffer, filtering buffer, six-staged pipeline filter and output frame unit.

Input Frame

The image pixels are taken from the MATLAB and given as an input to the deblocking filter. The image pixel has the size of 128x128. Those pixels are stored in an input frame.

Restructure Element and Memory Access Flow

The Restructure Element contains one current buffer, edge buffers and control signals as shown in the Figure 2. Besides, the edge buffers are comprised a few sub-blocks of left buffers, top buffers and corner buffers. Besides, the fundamental separating unit, FU, will be developed by joining pixels from current, left, top and corner buffers. Each FU will be sent to the FE for filtering and conveyed to the yield output frame memory. In this manner, with a specific end goal to develop the RFU, the controller will send "Buffer Selection" signs for choosing which blocks need to acknowledge the input data, as per the places of the approaching pixels. Correspondingly, the controller commands the selected buffers to convey the pixels with FU size to the FE, as indicated by the RFU and FU structures. One of the developments of the proposed configuration is that, rather than just utilizing the top buffer, top buffer with a different corner buffer is utilized. This is expecting to rearrange the deliver age plot and to decrease the weight of memory gets to.

On the other hand, in order to reduce the complexity of the address generation circuit and to mitigate the burden of memory accesses, in the proposed buffering system the pixels are consistently stored in two independent buffers, top and corner buffers respectively. In this way, a much improved and more predictable memory unraveling and get to design can be planned regardless of which LCUs are separated. Moreover, no extra memory gets to activity is required in the proposed memory structure.

Filter Element

The Filter Element is in responsible of performing the filtering process on the pixels. There are two major blocks in this element, a six-stage pipelined, two-line filter engine and a Filtering buffer. In this way, the Filtering buffer can be utilized to store one FU that is as of now handled. The pipelined design is to accomplish a high intensity. The operations for the six-stage pipelined architecture are:

(1) Parameter Calculation (PC)

This stage fetches the BS, QP (Quantization Parameter) and several offset values to calculate the Tc and β. There is no pixels access at this stage.

(2) Buffer Read (BR)

This stage fetches the pixels from the filtering buffer into the filter for future decision and filtering.

(3) Filtering Decision (FD)

This stage determines the filtering strength for the current edge and decides whether this edge needs to be filtered.

(4) Pre-Filtering (PFIL)

Pre-filtering and the following stage will filter the edge using weak or strong filter depending on the results of the FD stage. The reason to divide into PFIL and the following FIL is to balance the pipeline stages. For the weak filtering, this stage will calculate the values for the next FIL stage.

For the strong filtering, this stage will operate the partial strong filter's operation.

(5) Filtering (FIL)

This stage performs the rest operation of filtering process where the filtered pixels are generated and clip operations are conducted.

(6) Buffer Write (BW)

This is the last stage where the modified pixels are written back to the memory. The pixels from the filtering buffer is given to the pipelined stages for filtering, smoothening and to reduce the visual artifacts. Furthermore, the pixels are filtered from pipelining. After completion of six-pipelined stages the pixels are given back to the filtering buffer. The output frame read the pixels from the filtering buffer and writes back to the memory.

DESIGN FLOW

The deblocking filter reduces the blocking artifacts (visible discontinuities in an image) caused by block-based encoding with strong quantization. It is applied by modifying samples along horizontal and vertical boundaries of PUs and TUs. Filtering is applied separately in P and Q blocks, as shown in figure 3. The image pixels are taken by using MATLAB software. The boundary strength (BS) information defines an edge between two blocks of pixels. The boundary strength parameter is previously calculated based on the prediction mode, unit edge judgment, residual coefficient and motion vector etc. The BS parameter contains three possible levels (0 to 2) representing severity of artifacts at the boundary, where 0 is the weakest and 2 is the strongest.

As shown in design flow, first the boundary strength of an image pixels are calculated. If boundary strength is greater than zero then the filtering on/off condition1 will be checked, else no filtering. If condition1 is true then it will checks the further conditions2 to 7. If the condition 2 to 7 is true then strong filtering conditions are applied. If the conditions2 to 7 are false then the condition (10) will be checked. If the condition (10) is true then weak filtering conditions are applied, otherwise no filtering. Filtering On/Off Decision

The filtering on/off conditions is described as follows.

$$(p2, 0 - 2p1, 0 + p0, 0) + (p2,3 - 2p1,3 + p0,3) + (q2,0 - 2q1,0 + q0,0) + (q2,3 - 2q1,3 + q0,3) < \beta \quad (1)$$

In deblocking filter decisions, the quantization parameter (QP) value is taken into account. The parameter β control the edge which has to be filtered and control the selection between normal and strong filter. It also controls how many pixels from the block boundary are modified in the normal filtering operations. Equation1 evaluates only the first and fourth lines of a block to reduce the complexity.

$$(p2, 0 - 2p1, 0 + p0, 0) + (q2, 0 - 2q1, 0 + q0, 0) < \beta/8 \quad (2)$$

$$(p2, 3 - 2p1, 3 + p0, 3) + (q2, 3 - 2q1, 3 + q0, 3) < \beta/8 \quad (3)$$

$$(p3, 0 - p0, 0) + (q0, 0 - q3, 0) < \beta/8 \quad (4)$$

If the equation 2, 3 and 4 hold for both lines 0 and 3, the strong filtering is applied to the block boundary. Otherwise, normal filtering is applied. The condition4 checks the difference in intensities of samples on two sides of the block boundary does not exceed the threshold, which is a multiple of the clipping value Tc dependent on QP

$$(p3, 3 - p0, 3) + (q0, 3 - q3, 3) < \beta/8 \quad (5)$$

$$(p0, 0 - q0, 0) < 2.5Tc \quad (6)$$

If equation 5 is satisfied, the two nearest pixels to the block boundary can be modified in block P. Otherwise, only the nearest pixel in block P can be modified. Similarly, if condition6 holds the two nearest pixels to the block boundary can be modified in block Q. Otherwise, only the nearest pixel can be modified.

$$(p0, 3 - q0, 3) < 2.5Tc \quad (7)$$

$$(p2, 0 - 2p1, 0 + p0, 0) + (p2, 3 - 2p1, 3 + p0, 3) < 3/16 \beta \quad (8)$$

$$(q2, 0 - 2q1, 0 + q0, 0) + (q2, 3 - 2q1, 3 + q0, 3) < 3/16\beta, \delta0 = (9(q0 - p0) - 3(q1 - p1) + 8) >> 4 \quad (9)$$

Furthermore, the deblocking filtering is applied to the row or column of samples across the block boundary, if and only if the following expression holds:

$$(\delta0) < 10 * Tc \quad (10)$$

If this above condition does not hold, it is likely that the change of the signal on both sides of the block boundary is caused by a natural edge and not by a blocking artifact.

Results

Input Image

The input images are collected from local radiologist and a database of 50 images are created of different level of tumor. The figure 4 shows an input image of a deblocking filter. It has the resolution of 96DPI and it has the size of 128x128. This image will be taken as image pixels for the input of deblocking filter by using ModelSim.

Output Image

The figure 5 shows an output image of a deblocking filter. It has the same resolution of an input image (96DPI) with increasing the value of variance, standard deviation and intensity. It has size of 128x128.

Parameter Analysis

The above table 1 shows the parameter analysis of an input and output image. Here, the variance, standard deviation and brightness of an output image are increased

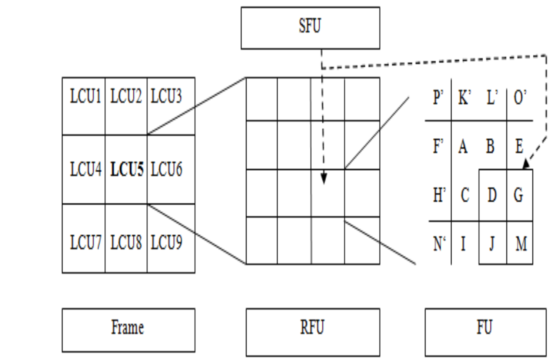

Figure 1. Illustrations of RFU, FU and SFU

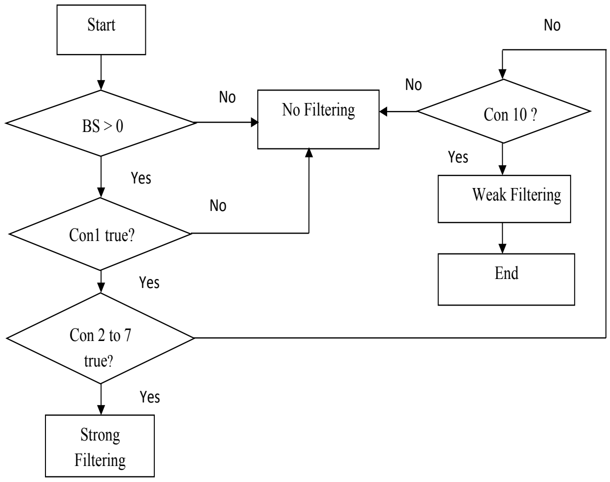

Figure 3. Design Flow for the Proposed Structure

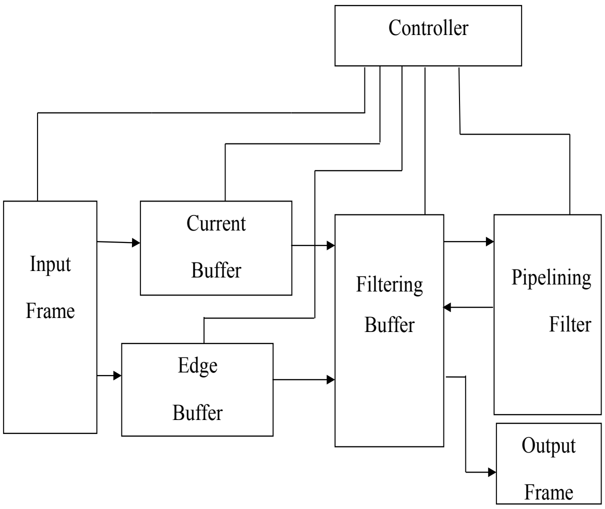

Figure 2. Block Diagram for the Proposed Deblocking Filter

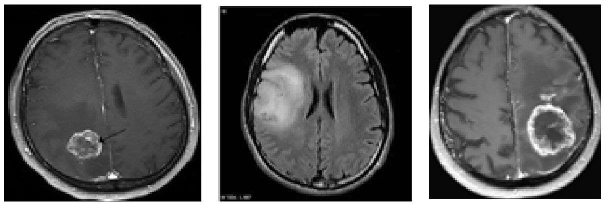

Figure 4. Input Image

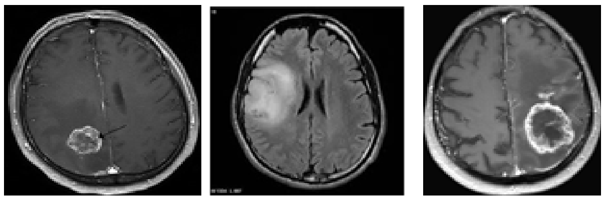

Figure 5. Output Image

Table 1. Parameter Analysis of Images

Parameter	Input Image 1	Output Image 1	Input Image 2	Output Image 2	Input Image 3	Output Image 3
Mean	221.2886	221.1487	211.2886	211.1886	221.7886	221.1487
Variance	5.88E+03	5.89E+03	5.68E+03	5.79E+03	5.81E+03	5.81E+03
Standard deviation	76.6661	76.7422	75.6661	75.7661	76.3211	76.3321
Brightness	102.3636	103.5217	100.3636	100.7636	102.5465	103.6542
Contrast	255	255	255	255	255	255
Resolution	96 DPI	96 DPI	96 DPI	96 DPI	96 DPI	96 DPI

and the mean value of an output image is decreased compared to an input image. The resolution and contrast of the output image is same as input image. The parameters are estimated for 50 samples.

True positive, True Negative, False Positive and False Negative are the confusion matrix features that are used for measuring the specificity, sensitivity and accuracy of the algorithm. The parameters are calculated based on the equations i, ii, iii and iv.

$$\text{True Positive} = \frac{\text{No. of resulted images having brain tumor}}{\text{Total number of images}} \quad \text{(i)}$$

$$\text{True Negative} = \frac{\text{No. of images without tumor}}{\text{Total number of images}} \quad \text{(ii)}$$

Table 2. Results of Output Image

Cases	Proposed method	
	True positive ratio (sensitivity)	True negative ratio (sensitivity)
Patient 1	0.8	0.44
Patient2	1	1
Patient3	0.889	0.8

$$\text{False Positive} = \frac{\text{No. of images falsely detected as tumor}}{\text{Total number of images}} \quad \text{(iii)}$$

$$\text{False Negative} = \frac{\text{No. of images having tumor and not detected}}{\text{Total number of images}} \quad \text{(iv)}$$

Table 3. Comparison Results Output Image PSNR Value

	Pipeline based	FDBS based [17]	Zhai method [20]	Visibility map method [2]
Technology	90 nm	0.8 um	90nm	90nm
Memory architecture	Two port	Two port	Two port	Two port
PSNR value	41.816 dB	35.67 dB	31.40 dB	29.37 dB

The Performance measure is based on classification accuracy, i.e. number of samples that has been properly detected into normal or abnormal as shown in Table 2.

Table 3 Comparison results output image is decreased compared to an input image. The resolution and contrast of the output image is same as input image. In the similar manner the parameters are estimated for hundred images. The comparison results between pipelines based algorithm, FDBS algorithm and deblocking algorithm. Here, the PSNR value of pipeline based algorithm is greater than other algorithm. Hence, the greater PSNR value leads to high quality of an image.

Discussion

The concept is to implement a deblocking filter and to design an effective deblocking filter with low cost, low complexity and high intensity, pipeline based systems are used. In addition to that the number of memory accesses and timing efficiency also be reduced using this method. The deblocking filtering operations can also easily perform in parallel on multiple processors by using six-stage of pipelined, two-line deblocking filter. The parameter mean, variance, standard deviation, resolution, contrast and PSNR values are compared with the previous method. Hence it shows the implementation of deblocking filter using pipelining is more efficient than others.

In future, the area complexity and throughput can be further reduced by using pipelining (High efficiency video coding) systems in the deblocking filter. Furthermore, the diabetic retinopathy can be detected by using this filter with the segmentation and feature extraction process. It can be applicable for other medical purpose.

Statement conflict of Interest

Architectural design of deblocking filter for image enhancement in the diagnosis of cancer.

References

Bossen F, Bross B, Suhring K, Flynn D (2012). HEVC complexity and implementation analysis. *IEEE Trans Circuits Syst Video Technol*, **2**, 1685-96.

Chen CM, Che CH (2005). Parallel processing for deblocking filter in H.264/AVC. Proceeding of the fourth IASTED International conference on CIIT, pp 188-91.

Chetouani A, Mostafaoui G, Beghdadi A (2009). Deblocking method using a perceptual recursive filter. IEEE International Conference on Image Processing, **2009**, **pages???**

Hautala I, Boutellier J, Hannuksela J, Silven O (2014). Programmable Low-Power Multicore Coprocessor Architecture for HEVC/H.265 In- Loop Filtering. *IEEE Trans Circuits Syst Video Technol*, **vol???**, 1217- 30.

Henot JP, Ropert M, Tanou JL, Kypreos J, Guionnet T (2013). High efficiency video coding (HEVC): Replacing or complementing existing compression standards?. *IEEE (BMSB)*, pp 1-6.

Liu TM, Lee WP, Lin TA , Lee CY (2005). A memory-efficient deblocking filter for H.264/AVC video coding. Advances in Multimedia Info Processing, pp 2140-3.

Li M, Zhou J, Zhou D, Peng X, Goto S (2012). De-blocking filter design for HEVC and H. 264/AVC. Advances in Multimedia Info Processing, pp 273-84.

Nightingale J, Wang Q, Grecos C (2012). HEVStream: a framework for streaming and evaluation of high efficiency video coding (HEVC) content in loss-prone networks. *IEEE Trans Consum Electron*, **58**, 404-12.

Norkin A, Bjontegaard A, Fuldseth M, et al (2012). HEVC Deblocking Filter. *IEEE Trans Circuits Syst Video Technol*, **22**, 1746-54.

Ohm JR, Gary JS, Schwarz H, et al (2012). Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC). *IEEE Trans Circuits Syst Video Technol*, **22**, 1669 – 84.

Ostermann J, Bormans J, List P, et al (2004). Video coding with H.264/AVC: tools, performance, and complexity. *IEEE Circuits Syst Mag*, **4**, 7-28.

Ozcan E, Adibelli Y, Hamzaoglu I (2013). A high performance deblocking filter hardware for high efficiency video coding. *IEEE Trans Consum Electron*, **59**, 714-20.

Pourazad MT, Doutre C, Azimi M, Nasiopoulos P (2012). HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC?. *IEEE Consumer Electro Magaz*, **1**, 36-46.

Shen S, Shen W, Fan Y, Zeng X (2013). A pipelined VLSI architecture for Sample Adaptive Offset (SAO) filter and deblocking filter of HEVC. *IEICE Electron Express*, **10**, 1-11.

Shen W, Shang Q, Shen S, Fan Y, Zeng X (2013). A high-throughput VLSI architecture for deblocking filter in HEVC. *IEEE J ISCAS*, **vol???,** 673-6.

Shih SY, Chang CR, Lin YL (2005). An AMBA-compliant deblocking filter IP for H.264/AVC. *IEEE J ISCAS*, **vol???**, 4529-32.

Viitanen M, Vanne J, Hamalainen TD, et al (2012). Complexity analysis of next-generation HEVC decoder. *IEEE J ISCAS*, **vol???,** 882-5.

Yan C, Zhang Y, Dai F, et al (2014) . Parallel deblocking filter for HEVC on many-core processor. *Electron Lett*, **50**, 367-8.

Zhai G, Zhang W, Yang X, Lin W (2008). Efficient image deblocking based on post filtering in shifted windows. *IEEE Trans Circuits Syst Video Technol*, **18**, 122-6.

Zhang Y, Yan C, Dai F, Yike MA (2012). Efficient parallel framework for H. 264/AVC deblocking filter on many-core platform. *IEEE Trans on Multimedia*, **14**, 510-24.