www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 6 June, 2013 Page No. 1834-1844

PERFORMANCE AND ANALYSIS OF BLOCKING ARTIFACTS REDUCTION USING PARALLEL DEBLOCKING FILTER FOR H.264/AVC ON LOW BITRATE CODING

[1]M.Anto bennet [2]Dr.I.Jacob Raglend [3]Dr.C.Nagarajan[4]P.Prakash

[1]Department of Electronics and Communication Engineering, Nandha Engineering College, Erode- 638052,India.

[2]Department of Electrical and Electronics Engineering, Noorul Islam University, TamilNadu-629025,India.

[3]Department of Electrical and Electronics Engineering, Muthayammal Engineering College, Rasipuram,India.

[4]Final UG Student, Department of Electronics and Communication Engineering Vel Tech Engineering College, Chennai, India.

ABSTRACT

The visual effects of blocking artifacts can be reduced by using deblocking filter. Appearance of the image is not clear while an artifact occurs. While parallelization, deblocking filter has complicated data dependencies to provide insufficient parallelization. The performance can be easily affected by the synchronization, when the number of cores increases. The load imbalance problem will be occurs. The proposed algorithm has three separate filtering strong filter, normal filter & no filtering. The vertical (flat) directions used as strong filter because of human visual system is sensitive. The horizontal directions used as smoothing filter. The deblocking filter process is divided into two parts BSC & EDF. BSC is Boundary Strength Computation; EDF is Edge Discrimination & Filtering. BS value determines the strength of the filtering, when BS value is zero it is no filtering, BS value of 4 is performed strong filtering, other values uses normal filtering. BSC uses METPMHT (Markov Empirical Transition Probability Matrix and Huffman Tree). EDF uses IPCAP (Independent Pixel Connected Area Parallelization). These two methods are used to increase the parallelization and reduce the synchronization.

Keywords: Deblocking filter, blocking Artifacts, METPMHT (Markov Empirical Transition Probability Matrix and Huffman Tree). IPCAP (Independent Pixel Connected Area Parallelization), Discrete cosine Tranform.

1.INTRODUCTION

Deblocking filter can improves the visual quality in some aspects but also reduce image quality in other aspects. It is a video filter, applied to blocks in decoded video. The visual quality & prediction performance can be improved by smoothing the sharp edges which can form between macro blocks. The filter aims to improve the appearance of decoded pictures. The filter operates on the edges of each 4*4 or 8*8 transform block in the luma & chroma. The components are Y, Cb& Cr. Component Y is luma it represents brightness. Cb& Cr are chroma for representing the color from gray towards blue and red. The human visual system is more sensitive to luma than chroma. The main advantage of the filter is to raises the integral quality of the image. Compared with original image the PSNR values also increased by using deblocking filter.

The images compressed by block transforms have visually annoying artifacts along the block boundaries is known as blocking artifacts. For reducing the blocking artifacts at the decoder end, many post processing algorithms are used. Blocking artifacts are the most serious defects that occur in images and video streams compressed to low bitrates using block DCT-based compression standards. At high or moderate bit rates, the DCT coded image has excellent reproduction without any artifacts. At low bit rates, the reconstructed images generally suffer from visually annoying artifacts as a result of quantization, that artifact is the blocking artifacts which appears as artificial block boundaries between adjacent blocks. Blocking artifacts reduces the visual quality of the reconstructed images. Blocking artifacts due to the coarse quantization of low frequency DCT coefficients yielding decompressed image it look like a mosaic at smooth regions.

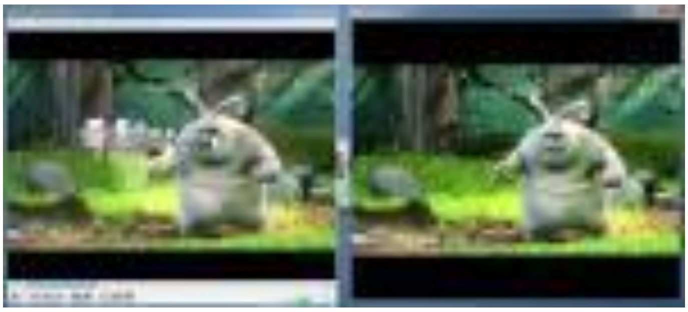


Fig1.Blocking artifacts

The deblocking filter is adopted to remove the discontinuity between adjacent blocks called blocking artifacts shown in fig [1].The deblocking filter is separately applied to left vertical & top horizontal boundaries of 4*4 blocks for both luminance & chrominance in the MB. The deblocking filter is applied, after a frame is encoded, bit stream still contains blocking artifacts those are removed by improving the coding efficiency. Blocking artifacts are removed by modification of DCT coefficients. If the edges is not detected at the block boundaries to preserve sharpness of edge deblocking filter is used. Macro block is techniques based on discrete cosine transform used on still images and video frames. It is composed of block of pixels. In JPEG standard macroblocks are called MCU blocks. Size of the block depends on the codec is a multiple of 4.In MPEG 2 and other early codec a block had a fixed size of 8*8 pixels. A macro block comprised four Y blocks, one cb blocks and one cr block, 16*16 pixel square of Ycbcr 4:2:0.

2. RELATED WORK

Ahmet Gürhanlı, Charlie Chung-Ping Chen, proposed a GOP (Group of Pictures) level approach due to its high scalability, mentioning solution approaches for the well-known memory issues [1]. Our design revokes the need to a scanner for GOP start-codes which was used in the earlier methods. This approach lets all the cores work on the decoding task. It does not need a GOP start code scanner.

Bart Pieters, Charles-Frederik proposed the in-loop deblocking filter in the MPEG-4 AVC/H.264 video coding standard is designed to reduce blocking artifacts caused by quantization[2]. The proposed scheme enables concurrent macro block deblocking of luma samples with limited synchronization effort, independently of slice configuration, and is compliant with the MPEG-4 H.264/AVC standard. A novel parallel processing algorithm for the deblocking filter in MPEG-4 AVC/H.264 is presented, facilitating concurrent filtering of the luma component of macro blocks on the massively-parallel architecture of the GPU, while staying compliant with the standard.

Bo-RueiChiou proposed Block mode selection is one new way to improve the performance of distributed video coding (DVC) [3]. Since there are many factors influencing the correctness of the block mode selection, the decision of block mode is not an easy work. A low complexity block mode selection model is proposed to select the block modes more correctly in the WZ (Wyner-Ziv) frame. The proposed block mode selection modules can increase the RD performance up to 2.8 dB as compared with the traditional DVC codec's; moreover, the subjective quality can also be improved by using the proposed deblocking filter when the objective quality is kept the same. Chi Ching Chi, Ben Juurlink, CorMeenderinck proposed. The two implemented parallel algorithms, the Task Pool (TP) and the novel Ring-Line (RL) approach,[4] both exploit macro block-level parallelism. The TP implementation follows the master-slave paradigm and is very dynamic so that in theory perfect load balancing can be achieved. The RL approach is distributed and more predictable in the sense that the mapping of macro blocks to processing elements is fixed. Chenggang Yan, Feng Dai, Yongdong Zhang, proposed H.264/AVC is the newest coding standard which brings a high efficient video compression method to the multimedia industry[5]. Due to its compression efficiency, it has been applied to several important applications including video telephony, video storage, broadcast, video streaming, HDDVD. The deblocking filter process which is composed of three parts, BS computation, ED and filtering. Gang proposed a high-performance intra prediction architecture that can support H.264/AVC high profile[6]. The proposed MB/block co-reordering can avoid data dependency and improve pipeline utilization. Therefore, the timing constraint of real-time 4k×2k encoding can be achieved with negligible quality loss. 16×16 prediction engine and 8×8 prediction engine work parallel for prediction and coefficients generating. A reordering interlaced reconstruction is also designed for fully pipelined architecture. It takes only 160 cycles to process one macro block (MB). Hardware utilization of prediction and reconstruction modules is almost 100%. Furthermore, PE-reusable 8×8 intra predictor and hybrid SAD & SATD mode decision are proposed to save hardware cost. The design is implemented by 90nm CMOS technology with 113.2k gates and can encode 4k×2k video sequences at 60 fps with operation frequency of 310MHz.

HongtaoXie, KeGao proposed a novel GPU-based scale invariant interest point detector, coined Harris-Hessian[7]. This method significantly reduces the pixel-level computation complexity and has better parallelism. a new scale invariant interest point detection algorithm according to the characteristics of GPU and CUDA. To avoid multistage simulation, the new algorithm detects interest points with Harris detector in low scale and refines the points in higher scale-space relying on the determinant of Hessian matrix. It has good parallelism and can satisfy the demand of real-time detection under large scale data with high detection accuracy.

Javier Taibo proposed Motion estimation [8] is known to be one of the most expensive tasks in video coding as it is usually performed through blind search-based methods. However, in the particular case of computer-generated video, the rendering stage provides useful information to speed up the process. A fast motion estimation algorithm, designed to run completely inside the GPU, to compute the optical flow required to estimate motion vectors at the same time as the graphical rendering process by using high-level information about the objects, viewpoints and effects that define each frame. The proposed method takes advantage of GPU parallelism and avoids bottlenecks in the CPU-GPU communication as the entire rendering and encoding process is performed completely inside the GPU. Avoiding search, motion estimation has very little overhead, negligible when compared with rendering and (the rest of the) video encoding costs while maintaining reasonably good quality.

JuRen, Mei Wen, Chunyuan Zhang, Huayou Su, proposed a parallel motion estimation algorithm based on stream processing [9]. Many approaches are explored to enable high data reuse efficiency and computation parallelism for GPUs or other programmable processors .A profile of ME to analyze its inherent parallelism and restrictions. Then design a Parallel Streaming Motion Estimation algorithm(PSME) with the purpose of exploiting the ME parallelism. The most complex component in video compression standard, ME has been setting high demand on the processors.

Keol Cho, Ki-Seok Chung and Won-Jin Kim proposed a novel parallel H.264/AVC decoding scheme on a homogeneous multi-core platform [10]. Parallelization of H.264/AVC decoding is challenging not only because parallelization may incur significant synchronization overhead but also because software may have complicated dependencies. To overcome such issues, we propose a novel approach called Stage-based Frame-Partitioned Parallelization (SFPP). Parallel software implementation may be roughly classified into two categories: task parallelism or data parallelism. Task parallelism is to divide a task into multiple sub-tasks and to distribute each sub-task to multiple cores simultaneously. Data parallelism is to divide a block of data into multiple sub-blocks, and let them be processed by multiple cores in parallel. It is very challenging to achieve high performance by a software implementation because its complexity is pretty high.

Mauricio Alvarez-Mesa, Chi Ching Chi, proposed the project development is implemented into the HEVC test Model (HM)[11], which is the reference software following the standard developments. The parallelization strategies and improvements for an HEVC software realization to support real-time HD and near real-time 4k on a standard PC platform. HEVC also includes two new filters that are applied after the deblocking filter: Sample Adaptive Offset (SAO) and Adaptive Loop Filter (ALF).

R.Rodriguez proposed Performance evaluation is done with a CUDA implementation for MPEG-2 video [12], though results are valid for other formats, and it has been tested as part of the rendering and encoding engine of a real-world system that provides server-side visually-rich interactive applications to lightweight clients equipped with standard MPEG video decoders. While in terms of computational complexity the BMAs certainly have an edge over Mesh based ME, since mesh based models involve interpolation of motion vectors which requires more complex architecture.

Song Hyun Jo, Seongmin Jo, and Yong Ho Song proposed the H.264/AVC decoder suggests a method for controlling the parallel execution of threads [13]. The H.264/AVC decoder proposed parallelization technique achieves a 25% increase in speed. This technique is used to improve the performance.

Tiago Dias, Nuno Roma, proposed a highly modular framework for developing parallel H.264/AVC video encoders in multi-core systems is presented [14]. Such framework implements an efficient hardware/software code sign methodology, which enables replacing the software implementation of any operation in the video encoder application by a corresponding system call to a hardware accelerator. For this an implementation of a multicore H.264/AVC video encoder using an ASIP IP core as a ME hardware accelerator is presented.

Won-Jin Kim, Keol Cho proposed parallel entropy coding algorithms have been proposed. Syntax element partitioning is an algorithm for parallelization of Context Adaptive Binary Arithmetic Coding (CABAC)[15]. It proposed Multi-Threaded Syntax Element Partitioning (MT-SEP) for parallel entropy decoding. The parallel video decoding over hardware implementations will be that versatile video codec's can be implemented flexibly.

Wenying Wang, Dongming Zhang, proposed Here implement the method on modern GPU in parallel[16]. Computations are distributed evenly to threads with load balancing, and the memory accesses are optimized and bitmap based parallel scan is exploited.

Yu-Shan Pai proposed Distributed video coding (DVC) is a new coding paradigm targeting applications with the need for low-complexity encoding at the cost of a higher decoding complexity[17]. In the DVC architecture based on a feedback channel, the high decoding complexity is mainly due to the request-decode operation with repetitively fixed step size (induced by Slepian-Wolf decoding). A parallel message-passing decoding algorithm for low density parity check (LDPC) syndrome is applied through Compute Unified Device Architecture (CUDA) based on General-Purpose Graphics Processing Unit (GPGPU). Furthermore, we propose an approach to reduce the number of requests dubbed as Ladder Request Step Size (LRSS) which leads to more speedup gain. Experimental results show that, through our work, significant speedup in decoding time is achieved with negligible loss in rate-distortion (RD) performance.

3. PROPOSED SYSTEM

H.264/MPEG-4 or AVC (Advanced Video Coding) is a standard for video compression, and is currently one of the most commonly used formats for the recording, compression, and distribution of high definition video. H.264 codec's consist of an H.264 encoder and decoder pair. An encoder takes a video as input and it removes the artifact by using specific video encoding algorithm. H.264 encoders take the digital frames of a video and divides into two parts then process it by using Deblocking filter. H.264 processes video frames by first dividing them into units called macroblocks. A macroblock is a 16*16 array of pixels. By using BSC and EDF the noises are removed in input video.

The entire deblocking filter process is divides into two parts: BSC and EDF. The BSC first parallelized before the EDF, which increases the parallelism. The BSC is time consuming and has the load imbalance problem. BSC has biased statistical distribution in temporal domain, and model it as a Markov chain, which can be characterized as an ETPM. A Huffman tree is derived from the ETPM. Using the Huffman tree, the expected BSC can be minimized and alleviate the load imbalance problem of BSC.

The EDF has limited parallelism and lots of synchronization. The data dependencies in the pixel-level introduced in the research. There are two chroma and one luma components in the video color space used by H.264/AVC. EDC present different IPCAPs for each component, which greatly increases parallelism and reduces the synchronization overhead.

3.1. INPUT VIDEO:

The Video which is going to be encoded using H 264/AVC deblocking filter.

3.2. BSC & EDF:

In contrast with older MPEG-1/2/4 standards, the H.264 deblocking filter is not an optional additional feature in the decoder. It is a feature on both the decoding path and on the encoding path, so that the in-loop effects of the filter are taken into account in reference macro blocks used for prediction. When a stream is encoded, the filter strength can be selected, or the filter can be switched off entirely. Otherwise, the filter strength is determined by coding modes of adjacent blocks, quantization step size, and the steepness of the luminance gradient between blocks.

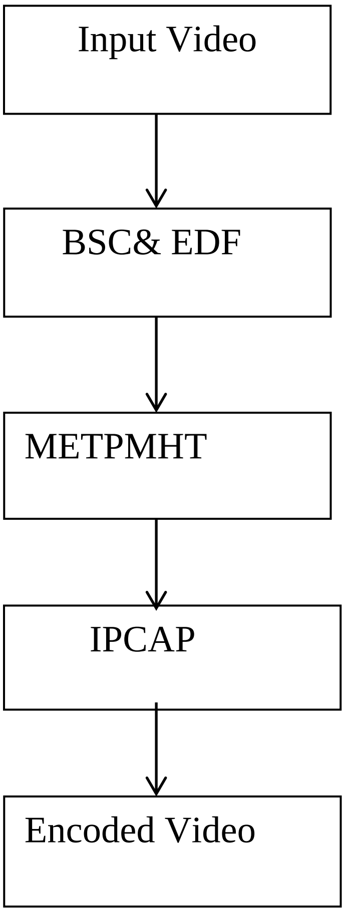


Fig 2. Block diagram of proposed system

The filter operates on the edges of each 4×4 or 8×8 transform block in the luma and chroma planes of each picture. Each small block's edge is assigned a boundary strength based on whether it is also a macroblock boundary, the coding (intra/inter) of the blocks, whether references (in motion prediction and reference frame choice) differ, and whether it is a luma or chroma edge. Stronger levels of filtering are assigned by this scheme where there is likely to be more distortion. The filter can modify as many as three samples on either side of a given block edge (in the case where an edge is a luma edge that lies between different macroblocks and at least one of them is intra coded). In most cases it can modify one or two samples on either side of the edge (depending on the quantization step size, the tuning of the filter strength by the encoder, the result of an edge detection test, and other factors).

3.3. METPMHT:

A discrete-time random process involves a system which is in a certain state at each step, with the state changing randomly between steps. The steps are often thought of as moments in time, but they can equally well refer to physical distance or any other discrete measurement; formally, the steps are the integers or natural numbers, and the random process is a mapping of these to states. The Markov property states that the conditional probability distribution for the system at the next step (and in fact at all future steps) depends only on the current state of the system, and not additionally on the state of the system at previous steps.

Since the system changes randomly, it is generally impossible to predict with certainty the state of a Markov chain at a given point in the future. However, the statistical properties of the system's future can be predicted. In many applications, it is these statistical properties that are important.

3.4. IPCAP:

After we first parallelize the BSC, the EDF shows the same data dependencies as the entire deblocking filter process. The order of the deblocking filter should not change arbitrarily, which may influence the results of H.264/AVC encoder and decoder. MBs can be processed out of scan order, provided these dependencies are satisfied. The 2D-wavefront method, MBRLPM, and ECPM study the data dependencies in MB-level.The data dependencies in the pixel-level. In the video color space used by H.264/AVC, there are two chroma and one luma components, which can be processed in parallel. The IPCAP in the luma component is different from that in the chroma component are shown in fig[2].

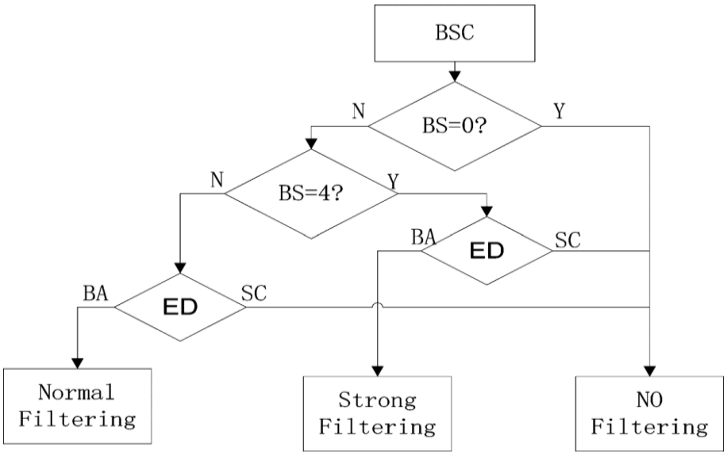


Fig 3. Algorithm for reducing blocking Artifacts using deblocking filter

The BS value determines the strength of filtering. No filtering is needed with a BS value of 0, whereas the strongest filtering is performed with a BS value of 4, and normal filtering with other values. ED distinguishes the blocking artifacts (BA) from the sharpness of the content (SC), which determines adopt the filtering. Filtering is then applied according to the value of BS shown in fig[3].

Since boundary strength computations is a major factor for computational complexity of H.264/AVC deblocking filter and these computations are primarily used to select between strong and normal filter.

Therefore, it is worth mentioning to analyze the frequency of usage of strong and normal filter in H.264/AVC deblocking filter for various video sequences. H.264/AVC based loop deblocking filter employs two kinds of filters i.e., (1) normal filter and (2) strong filter based on boundary strength. In proposed research, the source code of H.264/AVC deblocking filter has been modified to insert flags at pixel level, macroblock level and frame level to study the effects of the employment of these two filters in H.264/AVC deblocking filter. All sequences used in experimentation are configured by taking intra period as 0, i.e., only first frame is taken as intra frame.

3.5. Parallelize BSC Filtering

It describes a novel approach adopted in deblocking filter for H.264/AVC video. Performance analysis of H.264/AVC loop factor shows that it reduces the blocking artifacts significantly at low bit rates. However, it is highly computationally complex, as it takes one-third of computational resources of the decoder according to an analysis of run-time profiles of decoder sub- functions.

The main reasons for high computational complexity of filter are: (1) heavy conditional processing for edge strength computations on block edge, (2) pixel level required for filtering decision, and (3) to select one type of filter out of two filters: strong/normal. Very little work has been reported in the area of algorithmic optimization in comparison to efficient hardware implementation of H.264/AVC deblocking filter. A novel approach of incorporating motion activity of video sequences in deblocking filter is proposed, which results in optimized deblocking algorithm with respect to computing complexity. The proposed algorithm not only reduces blocking artifacts without significant loss of subjective quality but also has significant reduction in computing complexity in comparison with original H.264/AVC deblocking filter.

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in 1D signal is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

At first, many MBs have to wait for completion of others. For example, MB4 in core2 can be processed when MB1 and MB2 have been processed in core1. We find that most of the deblocking computation resources are spent on BSC, which is independent from the EDF. Therefore, we can parallelize BSC first and apply 2D-wavefront method to the EDF. In order to reduce synchronization of BSC, calculate the number of BSCs and assign them to cores equally each core has almost the same number of BSC, and all the cores are at full capacity at the beginning. Although the number of independent MBs at the start of the EDF is the same as the 2D-wavefront method, the stall becomes smaller.

4. EXPERIMENTAL RESULTS

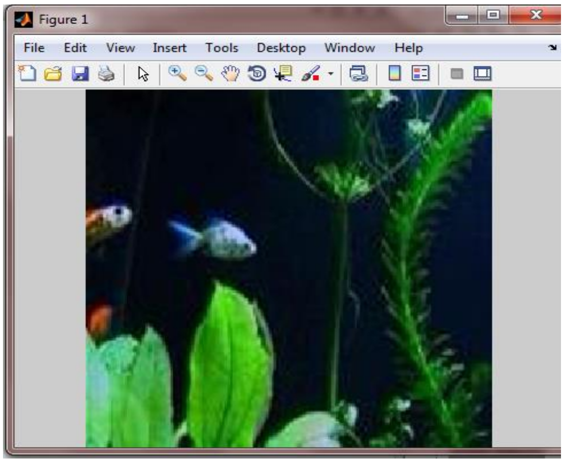


Fig 4.Input video

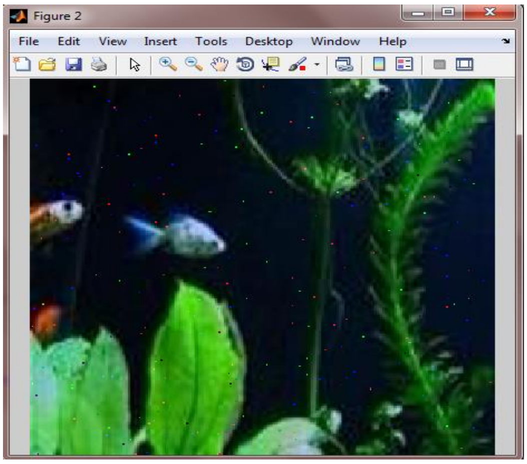


Fig 5.Noisy Video

.

The video is encoded using H.264/AVC standard as shown in Fig 4. By using Deblocking filter the input is divided into two parts. One is BSC and another one is EDF. Using these two algorithms the input is encoded.

While encoding the artifacts will occurs. Artifact is a kind of noise occurring during encoding. The Input video is corrupted with noise. It can be removed by using Deblocking filter. The salt and pepper noise is occurred in input as shown in Fig 5. This will be eliminated in proposed system.

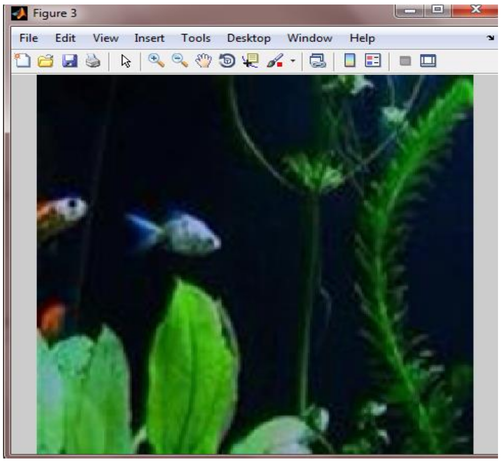
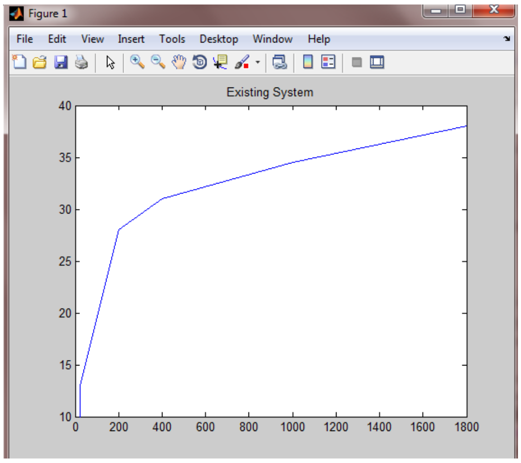


Fig 6.output video

The noise is removed and the efficiency is increased in proposed system. Finally the output video is decoded as shown in Fig 6. By the improvement of blocking artifacts the PSNR value is calculated. The PSNR value is compared with the existing system shown in table values [1] and graph [7,8 &9].



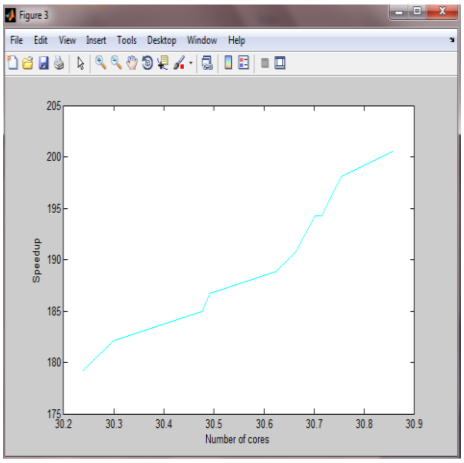


Fig 7. Existing PSNR Values

Fig 8. Improvement of PSNR (Proposed)

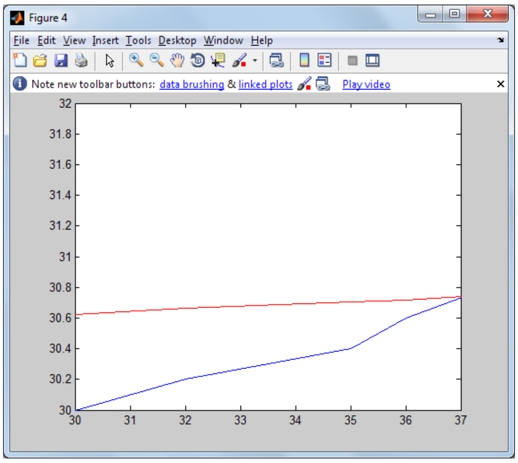


Fig 9. Improvement of PSNR Compared with the Existing System

S NO	Existing PSNR values	Proposed PSNR values
1	10	28.85
2	13	28.75
3	28	30.71
4	31	31.70
5	32.5	32.66
6	33	33.49

Table.1 Performance of PSNR Values

5. CONCLUSION

The main objective of this is design and implementation of deblocking filter for low bit rate video coding with reduced computing complexity. It presented two case studies with respect to low bit rate video coding: Performance analysis of H.264/AVC video coding standard with existing standards and evaluation of H.264/AVC deblocking filter. The results of performance analysis of H.264/AVC proved its superiority over the other video coding standards at low bit rates. While, the evaluation of H.264/AVC deblocking filter demonstrated the effectiveness of filter for suppressing blocking artifacts. On the other hand, computing complexity of H.264/AVC deblocking filter is recognized. By taking computing complexity reduction as a target, deblocking algorithm based on motion activity of video sequences is proposed. Due to availability of motion vectors from motion estimation, absolute sum of motion compensation vectors are used for categorizing different video sequences according to motion activity. Furthermore, impact of H.264/AVC's strong and normal filter is analyzed on various different video sequences. It is found through experimentation that for low to moderate video sequences, strong filter can be replaced by normal filter. As a result, the boundary strength that takes the major chunk of computations is not used for low to moderate motion activity video sequences in proposed approach. Computational complexity comparison of proposed approach with original deblocking algorithm revealed significant amount of reduction in various computing operations. Furthermore, objective analysis of proposed approach shows that PSNR of low to moderate video sequences does not change much in comparison with original algorithm. The extensive subjective comparison demonstrate that perceptual quality of video using proposed method is comparable with original H.264/AVC deblocking algorithm. The proposed deblocking algorithm can be used in applications, where reduced computing complexity with low bandwidth is desired.

REFERENCES

[1]AhmetGürhanlı, Charlie Chung-Ping Chen, and Shih-Hao Hung" Coarse Grain Parallelization of H.264 Video Decoder and Memory Bottleneck in Multi-Core Architectures"*Vol. 3, No. 3, June 2011*

[2]. Bart Pieters, Charles-Frederik J. Hollemeersch, Jan De Cock" Parallel Deblocking Filtering in MPEG-4 AVC/H.264 on Massively-Parallel Architectures" IEEE trans july 2010.

[3] B.-R. Chiou, Y.-C.Shen,H.-P. Cheng, and J.-L.Wu, "Performance improvement of distributed video coding by using block mode selection," in *Proc. ACM Conf. Multimedia*, 2010, pp. 1207–1210.

[4] C. C. Chi, B. Juurlink, and C. Meenderinck, "Evaluation of parallel H.264 decoding strategies for the cell broadband engine," in *Proc. Int. Conf. Supercomput.*, 2010, pp. 105–114.

[5] C. Yan, F. Dai, Y. Zhang, Y. Ma, L. Chen, L. Fan, and Y. Zheng, "Parallel deblocking filter for H.264/AVC implemented on Tile64 platform," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2011, accepted for publication.

[6] G. He, D. Zhou, J. Zhou, and S. Goto, "Intra prediction architecture for H.264/AVC QFHD encoder," in *Proc. Picture Coding Symp.*, 2010, pp. 450–453.

[7] HongtaoXie, KeGao, Yongdong Zhang, Jintao Li, Yizhi Liu "GPU-BASED FAST SCALE INVARIANT INTEREST POINT DETECTOR" IEEE ICASSP 2010.

[8] J. Taibo, v. M. Gulías, P.Montero, and S. Rivas, "GPU-based fastmotion estimation for on-the-fly encoding of computer-generated video streams," in *Proc. NOSSDAV*, 2011, pp. 75–80.

[9] J. Ren, M. Wen, C. Zhang, H. Su, Y. He, and N. Wu, "A parallel streaming motion estimation for real-time HD H.264 encoding on programmable processors," in *Proc. Frontier Comput. Sci. Technol.*, 2010, pp. 154–160.

[10] Keol Cho, Ki-Seok Chung "Stage-based Frame-Partitioned Parallelization
of H.264/AVC Decoding" IEEE Transactions on Consumer Electronics, Vol. 56, No. 2, May 2010.

[11] M. Alvarez-Mesa, C. C. Chi, V. George, T. Schierl, and B. Juurlink, "Parallel video decoding in the emerging HEVC standard," in *Proc.37th Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2012, accepted for publication.

[12] R. Rodriguez, G. Fernandez-Escribano, J. M. Claver, and J. L. Sanchez, "Accelerating H.264 inter prediction in a GPU by using CUDA," in *Proc. IEEE Int. Conf. Consum. Electron.*, 2010, pp. 463–464.

[13] S. H. Jo, S. Jo, and Y. H. Song, "Efficient coordination of parallel threads of H.264/AVC decoder for performance improvement," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1963–1971, Aug. 2010.

[14] Tiago Dias, NunoRoma ,Leonel Sousa "H.264/AVC framework for multi-core embedded video encoders" 978-1-4244-8277-1/10/$26.00c 2010 IEEE.

[15] Won-Jin Kim, Keol Cho and Ki-Seok Chung, Member, IEEE "Multi-Threaded Syntax Element Partitioning for Parallel Entropy Decoding" IEEE Transactions on Consumer Electronics, Vol. 57, No. 2, May 2011.

[16] Wenying Wang, Dongming Zhang, Yongdong Zhang, Jintao Li "PARALLEL SPATIAL MATCHING FOR OBJECT RETRIEVAL IMPLEMENTED ON GPU" 978-1-4244-7493-6/10/$26.00 ©2010 IEEE.

[17] Y.-S. Pai, H.-P.Cheng, Y.-C.Shen, and J.-L. Wu, "Fast decoding for LDPC based distributed video coding," in *Proc. ACM Conf. Multimedia*, 2010, pp. 1211–1214.