Design Of Lossy And Lossless Algorithms For Roi-Based Video Compression

Mahalakshmi Ramadoss, Dr.S.K Mahendran

Abstract: Video compression is the art of reducing the size of video without losing important visual details and can be performed either in a lossless or lossy fashion. In order to increase the compression performance, another method called ROI (Region of Interest) based compression algorithms also exist. In this method, the ROI region is compressed using a lossless algorithm, while the non-ROI region is compressed using a lossy algorithm. In this paper, two compression algorithms that are tuned to improve the performance of the lossy and lossless compression is proposed. A lossy algorithm that combines Discrete Wavelet Packet Transformation (DWPT), Singular Value Decomposition (SVD) and enhanced Run Length Encoding (RLE) is proposed to compress the background and non-ROI regions. An enhanced Binary Wavelet Transformation (BWT) is proposed to perform the lossless compression on the ROI region. The enhancement operations include method to make BWT work with color video frames, reduce the searching time, reduce the time complexity and improve the compression ratio. Experimental results prove that the methods proposed are effective in compressing a video while maintaining maximum visual quality

Index Terms: Binary Wavelet Transformation, Lossless Compression, Lossy Compression, ROI-Based Video Compression, Video Compression, Wavelet Packets.

1. INTRODUCTION

Demands for HD (High Definition) quality videos has increased storage requirements and network resources. Moreover, the phenomenal growth envisaged in transmission technology has provided new manners of propagating and sharing of video data. As developments in recording devices continue to change, the market requires new compression algorithms and/or enhancement of existing algorithms, to meet the current communication requirements. Video compression is a term that is used to quantify the amount of reduction in video data representation size achieved by a compression algorithm. Several algorithms have been proposed to improve the working of video compression algorithms (Hussain and Ahmed, 2019; Wang et al., 2019), but still the field is ripe. This paper in continuation with these works, proposes a ROI-based compression algorithm to improve the process of video compression. A ROI-based video compression allows the assignment of different priorities to different regions (like background, ROI, other non-ROI regions) of the video in terms of compression rate based on their importance in to application of interest(Ahmed et al., 2018). That is, these algorithms provide more quality to important regions (ROI) at the expense of reduced quality of other regions. These compression algorithms work on the principal that a viewer often gives more importance to certain regions over others and manipulation of this principal can produce better compression rate. A ROI-based video compression algorithm consists of two main components, namely, region separator and compressor (Xue et al., 2016). The region separation component is used to separate a video frame into interested (ROI) and non-interested regions.

- *Mahalakshmi Ramadoss is currently pursuing Ph.d Research in Computer Application in Bharathiyar University, Coimbatore, Tamilnadu, India. E-mail: maha2001new@rediffmail.com*
- *Dr.S.K Mahendran is currently working as a Assistant Professor, Department of Computer Science, Government Arts College, Udhagamandalam. Tamilnadu.India. E-mail: sk.mahendran@yahoo.co.in*

The compression component can be either lossy or lossless. The lossy compression is used on non-interested or non-ROI regions, which has minimum visibility to the human eyes and therefore, the reconstructed video is allowed to have some loss.The lossless compression is used on interested or ROI regions, where the reconstructed video is the exact replica of the original one and no loss is allowed. This paper focuses on the design of enhanced lossy and lossless algorithms to compress the ROI and non-ROI regions of a video. The ROI considered is the human face, which is the most frequently occurring object in applications like video conferencing, video calling and video sharing. It is assumed that the region separation component is already applied and the result has three regions, namely, background (BG), foreground non-ROI (FG-non-ROI) and foreground ROI (FG-ROI) regions.The region separation algorithm used frame differencing algorithm and a face detection algorithm to obtain the three regions. The face detection algorithm is based on Hybrid Color Model Created using Dynamic Color Component Selection Algorithm (HCD2CS) with random forest classifier. A tracking algorithm based on Kalman Filter was also proposed to track the detected regions across the frames in a video. Detailed description of these algorithms are provided in our previous work (Mahalakshmi and Mahendran, 2019). This paper proposes algorithms to perform lossy and lossless compression on these regions detected. The rest of the paper is organized as follows. Section 2 presents the methodology behind the design of lossy and lossless compression algorithms. Section 3 presents the results of performance evaluation conducted, while Section 4 concludes the work with future research directions.

2. METHODOLOGY

The proposed work is a methodology of compressing the frames in a video and the general framework of the proposed video compression algorithm is shown in Figure 1.

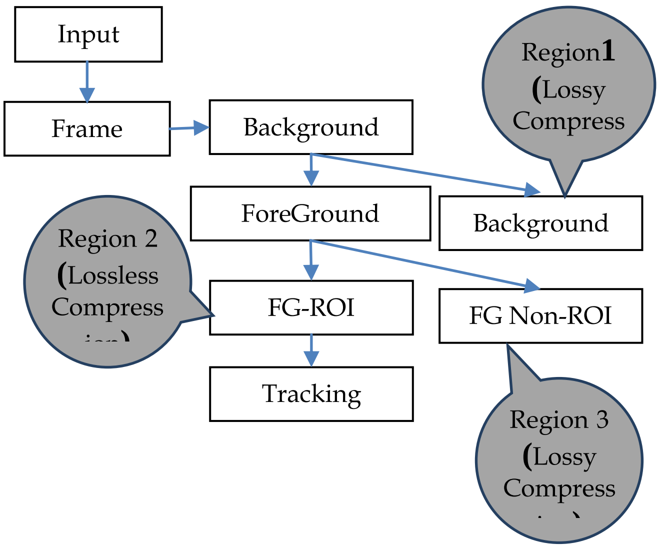


Figure 1 :Design of Proposed Video Compression Framework

The main goal here is to increase the compression ratio while maintaining the required visual quality of the video. An effective compression algorithm can make efficient use of the memory resource and can help to reduce the time required during transmission. This is made possible by removing unwanted or irrelevant data along with redundant information in the video. The proposed framework begins by separating the input video into various frames. The set of frames, thus obtained, is called a frame sequence. In the next step, the region separation algorithm is used on each frame, to obtain, the BG, FG-ROI and FG-non-ROI regions.In the proposed approach, the region of interest, that is, human face, is described by a bounding rectangle, whose location is described and coded using its corner points. The ROI-location information is defined as in Equation (1).

$$Ri = \{index, xi, yi, hi, wi\} \quad (1)$$

where Ri refers to the ith face in the video frame, index refer to the ROI region in the video sequence, xi and yiare the horizontal and vertical coordinates of the top-left corner pixel of Ri, hi and wi are the height and width of Ri. An example describing these parameters is shown in Figure 2.After obtaining the three regions a lossy compression algorithms are applied to the BG and FG-non-ROI and lossless algorithm is applied to the FG-ROI region.

Figure 2 : ROI Region

2.1. Lossy Algorithm for Compressing Background and non-ROI Regions

Usage of hybrid methods, which can combine the advantages of more than one algorithm, is always more beneficial than using a single method (Ac and Gülcan, 2019). In continuation with this fact, in this paper, three well-known and frequently used algorithms, namely, Discrete Wavelet Packet Transformation (DWPT), Singular Value Decomposition (SVD) and enhanced Run Length Encoding (RLE) are combined to compress the BG region of a video frame. This algorithm is referred to as L-BFnROI (Lossless Algorithm for Compressing BackGround and FG-non-ROI Regions) in this paper. The flow of the algorithm is shown in Figure 3.

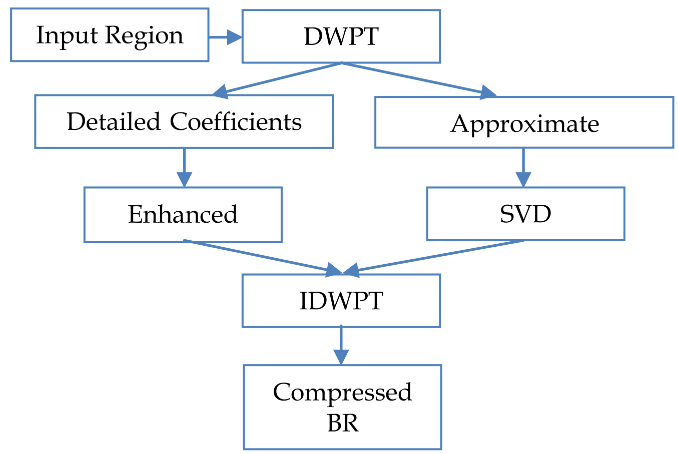


Figure 3 : Flow of L-BFnROI

The L-BFnROI begins by performing a 2-level Haar DWPT decomposition to obtain four subbands, LL (more relevant and important features that can be recognized by human eyes), HL, LH and HH (horizontal, vertical and diagonal coefficients have less important features). In order to reduce the computational complexity involved with DWPT, in the next step, the SVD decomposition is applied on the HL, LH and HH subbands using different K values. The value K denotes the number of eigenvalues used to reconstruct the compression region. Applying SVD decomposition on each subband produces three matrices Ui, Si and Vi respectively for each subband. Here, U and V are orthogonal matrices and S is the diagonal matrix having the singular values (SVs). The SVs are the square root of the eigenvalues of the product of the matrix (A) and its transpose. After the application of SVD, compression can be achieved by discarding the insignificant SVs. In general, a high SV indicate significant information (Ponnapalliet al., 2011). Thus, compression is achieved by first arranging the SVs in descending order and then retaining the higher values while discarding the lower SVs that has very low contribution towards visual quality. The amount of SVs discarded is controlled through the value of K. It was noted during experimentations, that with low K value, compression ratio is high and PSNR is low. This indicates that the information loss is high with lower K value. Thus, a suitable value that can achieve a reasonable compression ratio while preserving acceptable BG/FG-nROI region quality is needed. After several runs, the K value was set to 251 during experimentations. As the omitted singular values are lost permanently, it can never be recovered and hence produce a lossy compression.In the next step, the LL subband is compressed using an enhanced RLE encoder described in Section 2.3. Application of Inverse DWPT

(IDWPT) produced the compressed version of the regions. Reverse application of these steps produces the decompressed regions.

2.2. Lossless Algorithm for Compressing ROI Region

The application of Discrete Wavelets for lossless compression have been very successful (JPEG 2000 verification model version 8.6, 2000; Said and Pearlman, 1996) due to the various advantages like high coding efficiency, multi-resolution image representation and moderate computational complexity. In this work, the Binary Wavelet Transformation (BWT) is used during the design of a lossless compression algorithm for compressing the FG-ROI region of a frame. BWT was chosen because of its multiple advantages like (i) being extremely fast as it uses simple Boolean operations to perform arithmetic operations, (ii) introduces no quantization effects and (iii) being fully invertible. The proposed lossless compression algorithm for FG-ROI is termed as LL-FG-ROI, which is an enhanced variant of BWT. The proposed algorithm uses a Progressive Partitioning Binary Wavelet-tree Coder(Pan et al., 2007)whose search, complexity and compression performance is enhanced through the use of

- Joint Bit Scanning Method (to improve the search process)
- Lifting scheme (to reduce computational complexity)
- Burrows-Wheeler transformation with enhanced RLE (to increase compression ratio)

The methodology used by LL-FG-ROIare presented in Figure 4. The enhanced RLE used is described in Section2.3. BWT has the disadvantages of able to work only with binary images and having high search process. In this work, these problems are solved by modifying BWT to use a bit-plane decomposition to make it work with color video frames. The search time is reduced through the use of a wavelet tree. Further, joint bit scanning method and Burrows Wheeler Transform improved through the use of an enhanced run length coding (Section 2.3) is also used to improve the speed and compression capacity. In this algorithm, the BWT starts with a transformation which groups similar input bits together followed by a move-to-front algorithm which groups all frequently occurring bit sequences to the front. These bits are then compressed using the enhanced Run Length Encoder (RLE). The RLE is enhanced to remove the duplication problem of the traditional RLE.

1. Construct a series of bi-level images using the bitplane decomposition method considering the R, G, B components of the frame.
2. Apply Binary Wavelet Transformation to each individual bi-level bitplane image.
3. Perform Enhanced Joint-Bit Scanning using scanning sequence. The procedure used is given below.
    - For each pass, scan bits horizontally. When a significant bit is identified (non-zero bit), output remaining bits of this significant bit and then switch to vertical scanning. Vertical scanning is performed similar to horizontal scanning. The coefficients tagged significant are not considered in subsequent passes. This method sorts the coefficients more accurately according to their importance when compared to traditional vertical or horizontal zig-zag scanning.
4. Perform Progressive partitioning binary wavelet-tree coder.
5. Perform BWT with MTF and enhanced run length encoding to optimize coding process.

Figure 4 : Lossless Algorithm for Compressing FG-ROI Region

2.3. Enhanced RLE

The conventional RLE performs compression by encoding a sequence of pixels having the same intensity value by first recording the value, followed by the number of occurrence (referred to as run). For example, the sequence of pixels (88888844445) is encoded by RLE as (8 6 4 5), where 6 and 5 are the runs for the intensity values 8 and 4, respectively. The RLE algorithm performs well when the frame has long runs of pixels with the same intensity value but its performance degrades when the frame has high spatial activity (Luse, 1993). The inefficiency is due to the high variation in pixel intensity values in which case there are more intensity-run pairs to encode and hence increasing the file size. This is best illustrated with an example sequence having dissimilar pixels (1 2 3 4 5), which will be encoded as (1 1 2 1 3 1 4 1 5 1) by RLE. Obviously, adding the runs increases the cardinality (size) of the set (double in this case) and this issue is termed as duplication problem (Al-Wahaib and Wong, 2010). Avoiding this issue can enhance the performance of RLE and is considered in this paper. The pseudo code of the Enhanced Run Length Encoding (ERLE) that avoids duplication problem isgiven in Figure 5. The algorithm assumes that a general expression always starts with d, followed by x and ends with y. These are referred to as starting and ending rule. The encoding stage starts by assigning codewords to the intensity levels. The assignment is done according to the Probability of Occurrence (PoO) of each level such that the intensity levels of higher PoO are assigned to shorter codewords, and vice versa. The algorithm uses two codewords namely, Ending Bit Independent (EBI) and Ending Bit Dependent (EBD). The pixels (P) are then scanned in raster order and encoded by using EBI codewords. Since the ending group y is not defined at the pre-encoding stage, y in each codeword is utilized as a flag to determine the status of the next (neighbor) codeword. In particular, the flag can be set to indicate three status, namely, S1, S2 and S3, where S1 < S2 < S3.

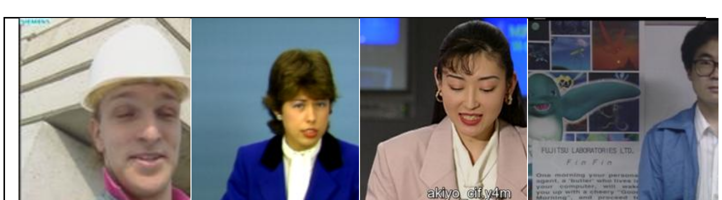
```
Initialize Step : i= 1; count = 0;
Repeat
        Read P
        if (P(i) = P (i-1))
                count=count+1;
        else
                If (count = 3) then encode with EBI codeword along with its count and set y to S3
                If (count = 2) then encode with EBI codeword and set y to S2
                If (count = 1) then encode with EBI codeword and set y to S1
        end if
Until all BG detailed coefficients are scanned
```

Figure 5 : Enhanced Run Length Encoding

In case the encoder reads a run of three or more pixels of the same intensity value, these pixels are encoded as a codeword that represents their intensity level. The ‘y’ in this codeword is set to S3 to indicate that the next codeword encodes the count of pixels in the run. Here, the count of run is encoded by using the EBD codewords, which can encode an infinite number of pixels when considering more than one general expression. When the encoder reads a run of two pixels, these pixels are encoded into a single codeword in which its y is set to S2, indicating that the encoded pixel intensity value is repeated twice and the next codeword encodes the intensity level of the next pixel(s). The ERLC is an approach that gains a reduction (in terms of file size) from a run of only two pixels by encoding them to a single codeword. Other RLE approaches encode the pair of pixels to two codewords, i.e., one codeword for the intensity value and another codeword for the count (Salomon, 2004; Foley et al., 1990). Finally, in case the encoder reads a single pixel (i.e., not repeated or run is unity), this pixel is encoded by a single codeword, and its y is set to S1 to indicate that the next codeword is the intensity level of the next pixel(s).

3. EXPERIMENTAL RESULTS

Several experiments were conducted to evaluate the proposed lossless and lossy compression algorithms using videos downloaded from https://media.xiph.org/video/derf. Example frames from four videos, selected for discussion, are shown in Figure 6. Two performance metrics, namely, Peak Signal to Noise Ratio (PSNR) (dB) and Compression Ratio (CR) (%) were used to evaluate the algorithmsproposed. Figures 7 and 8 show the PSNR and CR of the of the proposed algorithms for the four selected videos indicated as Video1 to Video4 respectively. A similar trend in results were envisaged with other videos also.



Foreman	Claire	News	Bowing

Figure 6 : Sample Frames from Video Dataset

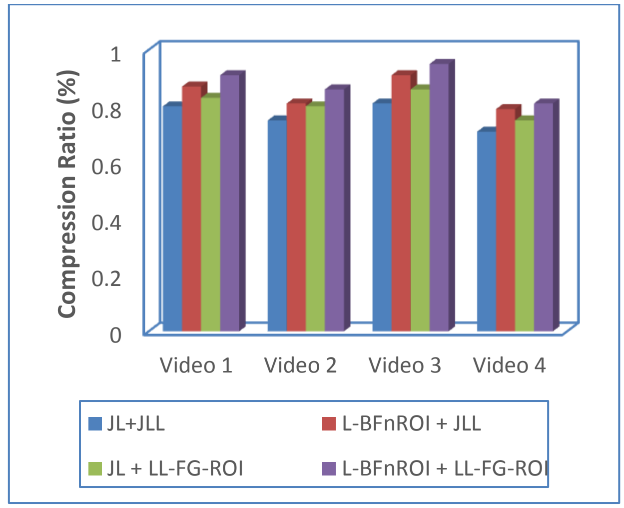


Figure 7 : Compression Ratio (%)

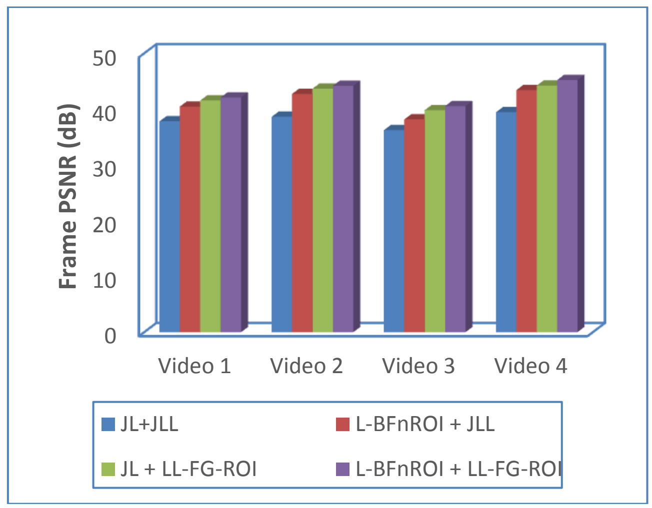


Figure 8 :PSNR (dB)

From the figures, it is clear that the proposed compression algorithm that uses the lossy BFnROI algorithm to compress background and foreground-ROI regions and lossless FG-ROI algorithm to compress foreground ROI region is more effective. On average, this algorithm could achieve an average efficiency gain of 13.03% when compared to the usage of conventional JL + JLL algorithm, with respect to compression ratio. Analysis of PSNR showed an average increase of 11.57% in quality while using the L-BFnROI + LL-FG-ROI over JL+JLL. Thus, from the results, it is clear that the proposed ROI-based compression algorithm is efficient in reducing the space required while maintaining important quality related information.

4. CONCLUSION

In this work, a ROI-based algorithm for compression videos was proposed. The proposed algorithm is composed of several algorithms, which when combined can improve the compression performance. Initially, a region separation algorithm, that separates a video frame into three regions, namely, background, foreground ROI and foreground non-

ROI regions was used. The ROI considered is the human face. A lossy algorithm that combined DWPT with SVD and RLE was then used to compress the background and non-ROI regions. Another algorithm that enhances BWT was used to compress the foreground ROI region in a lossless fashion. The BWT was first modified to work with color video frames and its working was further enhanced to enhance its searching process, reduce its computational complexity and increase compression performance. Both the algorithms use an RLE algorithm that is enhanced to avoid the duplication problem. Experimental results proved that the proposed algorithm using the proposed lossy and lossless algorithms improved both compression ratio and PSNR, indicating that it is well suited to compress videos. In future, a separate lossy algorithm may be developed for foreground non-ROI region and a method to provide secure compression is also to be probed.

REFERENCES

[1]. Ac, C.I. and Gülcan, H. (2019) A Modified Dragonfly Optimization Algorithm for Single- and Multiobjective Problems Using Brownian Motion, Computational Intelligence and Neuroscience, Vol. 2019, Article ID 6871298, 1-17.

[2]. Ahmed, S., Osama, A. and El-Sayed, E. (2018) Comparative study of video compression techniques. Menoufia J. of Electronic Engineering Research, Vol. 27, No. 1, Pp. 1-33.

[3]. Al-Wahaib, M.S. and Wong, K. (2010) A Lossless Image Compression Algorithm Using Duplication Free Run-Length Coding, Second International Conference on Network Applications, Protocols and Services, IEEE Computer Society, Pp. 245-250.

[4]. Foley, J.D., Dam, A., Feiner, S.K. and Hughes, J.F. (1990) Computer graphics:Principles and Practice (second edition), USA, Addison-Wesley, Longman Publishing Co., Inc. https://media.xiph.org/video/derf, Last Accessed During July, 2019.

[5]. Hussain, A.J. and Ahmed, Z. (2019) A survey on video compression fast block matching algorithms, Neurocomputing, Vol. 335, Pp. 215-237.

[6]. JPEG 2000 Verification Model Version 8.6 (VM8.6) (ISO/IEC,2000).

[7]. Luse, M. (1993) Bitmapped graphics programming in C++, Boston, Addison-Wesley Longman Publishing Co., Inc., USA.

[8]. Mahalakshmi, R. and Mahendran, S.K. (2019) Design of enhanced region separation algorithm for effective video compression, International Journal of Recent Technology and Engineering, Vol. 1, Issue 8, Pp. 2374-2379.

[9]. Pan, H., Siu, W.C. and Law, N.F. (2007) Lossless image compression usingbinary wavelet transform, IET Image Process., Vol.1, No.4, Pp. 353–362.

[10]. Ponnapalli, S.P., Saunders. M.A., Loan, V.C.F. and Alter, O (2011) A Higher-Order Generalized Singular Value Decomposition for Comparison of Global mRNA Expression from Multiple Organisms, PLoS ONE, Vol. 6, Issue 12, e28072, Pp. 1-12.

[11]. Said, A. and Pearlman, W.A. (1996) An image multiresolution representation for lossless and lossy compression, IEEE Trans. Image Process., Vol. 5, Pp. 1303–1310.

[12]. Salomon, D. (2004) Data compression : The complete reference, 3rd Edition, Library of Congress Cataloging-in-Publication Data, Published by Springer-Verlag London.

[13]. Wang, Z., Hou, Z., Hu, R. and Xiao, J. (2019) A Lossless Recompression Approach for Video Streaming Transmission, IEEE Access, Vol. 7, Pp. 35162-35172.

[14]. Xue, H., Zhang, Y. and Wei, Y. (2016) Fast ROI-based HEVC coding for surveillance videos, 19th International Symposium on Wireless Personal Multimedia Communications, Pp. 299-304.