Vol 8. No. 1 – March, 2015
African Journal of Computing & ICT

www.ajocict.net

Estimation of Missing Video Frame Block Using Bi-Directional Temporal Approach for Low Motion Sequences

A.D. Shelotkar
Department of Electronics Engineering,
Sant Gadge Baba Amravati University,
Amravati, Maharashtra, India
anishelotkar@gmail.com

P.V. Ingole
G. H. Raisoni College of
Engineering and Management,
Anjanagaon Bari Road, Badnera-
Amravati, Maharashtra, India
prashant.ingole@raisoni.net

ABSTRACT

With improvements in network technology, demand for video streaming application is on the rise. In video streaming applications, the quality of service is usually not guaranteed to reduce delay. The side effect of this approach is that there may be loss of a packet during video transmission which may corrupt block of data or even an entire video frame. There are many techniques proposed to recover loss of macroblocks in video frame, our bi-directional temporal; block estimation approach guarantees it at minimal complexity. For each pixel in the lost frame, the proposed algorithm estimates missing block using reconstructed frame and the next frame. We also extend the proposed algorithm for second order prediction to improve the performance. Upon macroblock recovery, an averaging filter or median filter is applied to reconstructed block as a deblocking filter. In case of scene change, we estimate missing block based on next frame and neglect previous reconstructed frame. Experiments are carried with different YUV streams and compressed domain AVI videos. The experimental results show improvements in PSNR as well as visual quality.

Keywords: Error concealment, Data Prediction, De blocking filter

African Journal of Computing & ICT Reference Format:
A.D. Shelotkar & P.V. Ingole (2015) Estimation of Missing Video Frame Block Using Bi-Directional Temporal Approach for Low Motion Sequences. Afr J. of Comp & ICTs. Vol 8, No. 1. Pp 213-218.

1. INTRODUCTION

With improvements in wired and wireless networks, more and more users are demanding video services, including video streaming and video conferencing over the Internet. However, the Internet does not provide guaranteed quality of service (QoS).Traffic congestion usually result in the loss of data packets. In wireless networks, packet loss happens frequently due to multipath fading, shadowing and noise disturbance of wireless channels. Video transmission uses compressed video streams for transmission so that video data can be transmitted even with poor network bandwidth situations. A loss of packet over transmission in compressed stream introduces severe distortion because the compression algorithms use temporal and spatial estimation methods to improve compression efficiency. Therefore a single distorted block within a frame may lead to errors not only in present frame but also propagate error over several frames.

Many error resilience and decoder error concealment techniques have been proposed to control amount of error in reconstructed frame [1]-[8].A simple error resilience strategy is to use feedback channels and request for retransmission whenever there is error. This is the most robust technique and the recovered data would always be correct. However, it involves halting decoding process till error block of data is received again. This is an inefficient approach in terms of delay involved in process. Another way to avoid errors is to embed error checks in encoded video bit streams and transmit over the channels. This method though avoids retransmission of video, it affects compression efficiency of the encoder and thus increased usage of network bandwidth. Hence, a set of post processing algorithms on the decoder side are proposed for error concealment. The advantage with decoder error concealment is that it does not require any change in encoding or decoding process.

It simply appends a post processing block which recovers erroneous data. Hence, there is no increase in bit rate or delay. Fig.1 shows block diagram for post processing of video sequence to recover loss of macroblocks. Therefore these techniques can be used in real time video applications like video-voice over internet and streaming applications [3].Temporal error concealment technique uses temporal neighbors to estimate erroneous block of data. It utilizes previous frame and/or next frame to conceal errors in current frame. Most of the temporal error concealment methods assume that only a few macroblocks (MB) or slices in a video frame are lost. Typically, temporal reconstruction process recovers lost data with the help of motion field interpolation (MFI) [3].

However, using just one of the frame may not be sufficient in case of scene change during video sequence. In case of scene change, Mean Absolute Difference (MAD) between two frames is very large and hence the temporal estimation methods fail. Therefore in this paper, we propose an approach based on temporal error concealment using averaging and Gaussian filters with the scene change into consideration. The algorithm selects macroblocks from reconstructed frames and next frame. The estimation of missing block of data is done based on selected macroblocks, and then smoothing filters are used to avoid blocking artifacts in reconstructed block of data. This method usually produces a relatively high Peak Signal-to-Noise Ratio (PSNR)value.

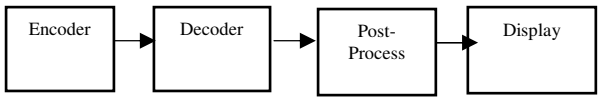

Fig. 1 Generic Block Diagram for Video Post Processing

The rest of the paper is organized as follows,
Section 2 reviews the theory behind the algorithm and proposes new algorithm to improve error concealment, section 3 shows experimental results and section 4 provides conclusion and future work in the same area.

2. BI-DIRECTIONAL TEMPORAL BLOCK ESTIMATION ALGORITHM

Temporal error concealment methods assume that motion in the video is not dramatic and do follow some linear or quadratic model. We first detect the video shot boundary to detect the scene change. If the block loss happened in the frame is not the key frame then we can use bidirectional predication and else we restrict the algorithm as a unidirectional prediction. Literature gives multiple algorithms for the video shot boundary detection [9], [10].We use Sum of Square Difference (SSD) as a measure for shot boundary detection,

$$SSD_n = \sum_i \sum_j (I_{n+1}(i,j) - I_n(i,j))^2 \quad \ldots\ldots\ldots\ldots\ldots(1)$$

Where, $I_n(i,j)$ is current frame and $I_{n+1}(i,j)$ is next frame. We apply adaptive threshold for detecting scene change based on SSD [10]. The threshold use is,

$$Th_n = \frac{\sum_N SSD_n}{N} \quad \ldots\ldots\ldots\ldots\ldots(2)$$

Where
N is the size of SSD vector window use for calculation.

If the block loss happen in the frame which is not a key frame (frame after scene change) but having second frame after key frame then we use the first order bidirectional prediction as,

$$IB_n(i,j) = \frac{IB_{n-1}(i,j) + IB_{n+1}(i,j)}{2} \quad \ldots\ldots\ldots\ldots\ldots(3)$$

Where
$IB(i,j)$ represent lost block and n is frame index. If the loss happens in the frame which has two frame before and after then we extent our algorithm as weighted quadratic predication,

$$IB_n(i,j) = \frac{k_1 * IB_{n-2}(i,j) + k_2 * IB_{n-1}(i,j) + k_2 * IB_{n+1}(i,j) + k_1 * IB_{n+2}(i,j)}{4} \quad \ldots\ldots\ldots\ldots\ldots(4)$$

Where
k_1 and k_2 are coefficient of quadratic model. We chose coefficients such that $k_2 > k_1$ and $k_1 + k_2 = 1$.

If block loss happen in the decoding of residual frame (we do not consider the loss of side information of previous and next frame, which contain the motion vector information) of compressed video. We replace the damaged block with the content of the previous frame at the motion compensated location.

$$IB_n(i,j) = \frac{IB_{n-1}(i+u, j+u) + IB_{n+1}(i-u, j-v)}{2} \quad \ldots\ldots(5)$$

Where
u and v are the horizontal and vertical motion vector at loss block location calculated from the motion vector of previous and next frame motion vectors,

$$u(i,j) = \frac{MVX_{n-1}(i,j) + MVX_{n+1}(i,j)}{2} \quad \text{.....(6)}$$

$$v(i,j) = \frac{MVY_{n-1}(i,j) + MVY_{n+1}(i,j)}{2} \quad \text{.....(7)}$$

Where
MVX and MVY are horizontal and vertical motion vector fieldsinside information of compressed video.

Another approach to recover the block of data is to use spatial information instead of temporal. In this method, it is assumed that the blocks surrounding the missing block of data are reconstructed perfectly. However, it may not be always necessary that the spatial reconstructed data be correct always.

Therefore, to we propose a technique which uses previous reconstructed frame and the next frame to estimate the data. Fig 2.explains proposed first order prediction algorithm.

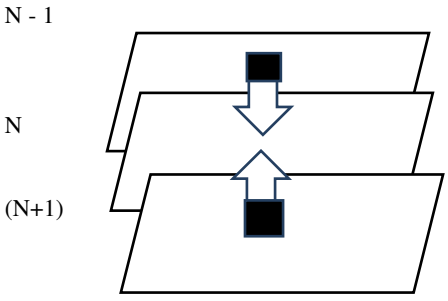

Fig.2 Proposed Block Recovery Algorithm.

Upon recovering the missing block, we use further improve estimation based on averaging and Gaussian filter to avoid blocking artifacts in reconstructed block.

We simply convolve reconstructed block with filter under consideration and obtain the final result.

$$Y = X * H \quad \text{.....(8)}$$

Where X is reconstructed block with averaging across two frames as depicted in Fig.2, H is filter to avoid blocking artifacts and Y is reconstructed block of data and '* ' indicates 2 dimensional convolution operation. Filter H can take either H_{avg} or H_{gauss} values as mentioned below.

$$H_{avg} = \begin{matrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{matrix} \quad \text{.....(9)}$$

$$H_{gauss} = {}^{1}/_{(2*\pi*\sigma)} \times e^{\frac{-(x^2+y^2)}{2*\sigma^2}} \quad \text{.....(10)}$$

$$H_{gauss} = \begin{matrix} 0.1096 & 0.1118 & 0.1096 \\ 0.1118 & 0.1141 & 0.1118 \\ 0.1096 & 0.1118 & 0.1096 \end{matrix} \quad \text{.....(11)}$$

$$\sigma^2 = 5 \quad \text{.....(12)}$$

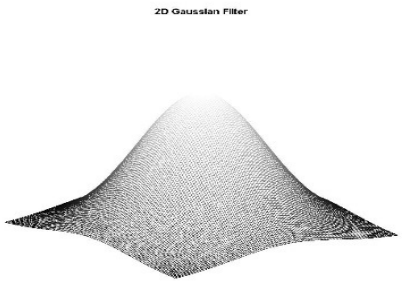

Fig. 3 Gaussian Kernel

The stepwise algorithm is as follows:

1. Decoded video sequence is used as an input to the algorithm.
2. Detect the video scene change using proposed algorithm.
3. Determine the block to be recovered in a given frame.
4. Estimate block using previous reconstructed frame and next frame either based of first order or quadratic prediction.
5. Apply smoothing filter to remove any blocking artifacts which may occur in the recovered frame.
6. Recovered data is smoothened by applying image smoothening filter.
7. Experiments are compared with averaging and Gaussian smoothening filter functions as mentioned.
8. Compute PSNR of smoothened frame.

3. EXPERIMENTAL RESULTS

This section describes experimental results for the proposed algorithm. The following assumptions are made while conducting experiments, video sequences used for RAW video error concealment are of QCIF resolution at 30fps, YUV format and AVI, MP4 videos for compressed video error concealment. Fig 4shows the original frame at top left, corrupted frame at top right, recovered frame at bottom left and the filtered frame at bottom right for 'akiyo' sequence. Similarly Fig 5 shows results for 'forman' sequence. Fig.6. and Fig. 7. shows frame number vs PSNR graph for foreman and akiyo video sequence.

Fig.4 Results for 'Akiyo' Sequence.

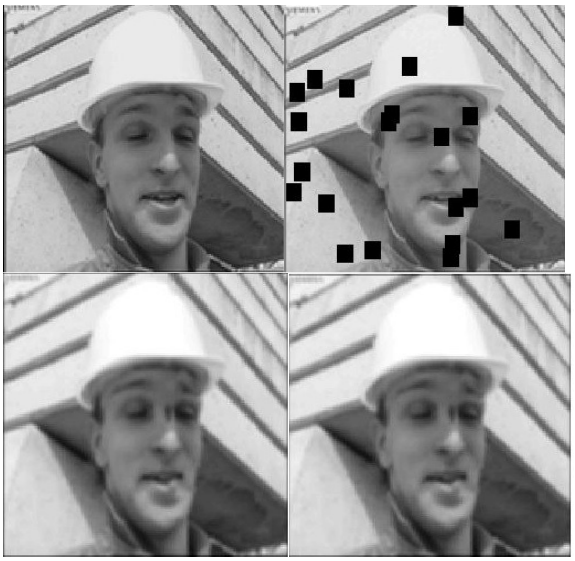

Fig.5 Results for 'Forman' Sequence.

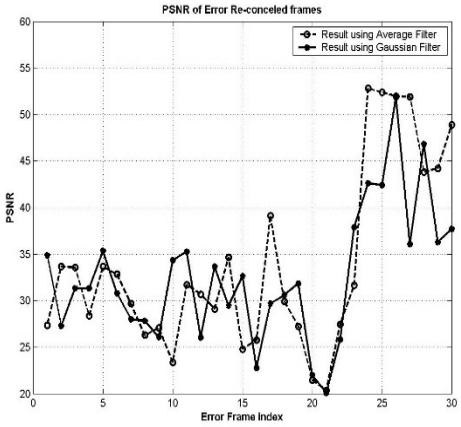

Fig. 6 Frame Vs PSNR for 'Foreman' Sequence

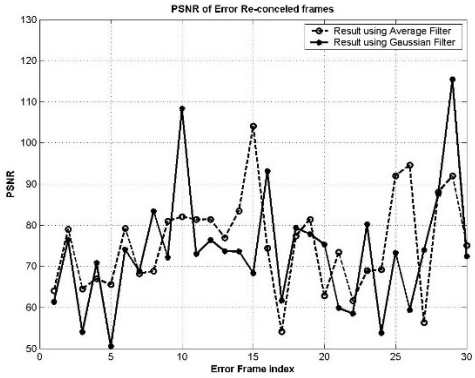

Fig. 7 Frame vs PSNR for 'Akiyo' Sequence.

To show the comparative study we have used the algorithm given in [11], as it is conceptually similar to the proposed algorithm. Fig. 8. Shows the concealment results for one frame using reference and proposed algorithm. Fig. 9. Shows the PSNR comparison respectively.

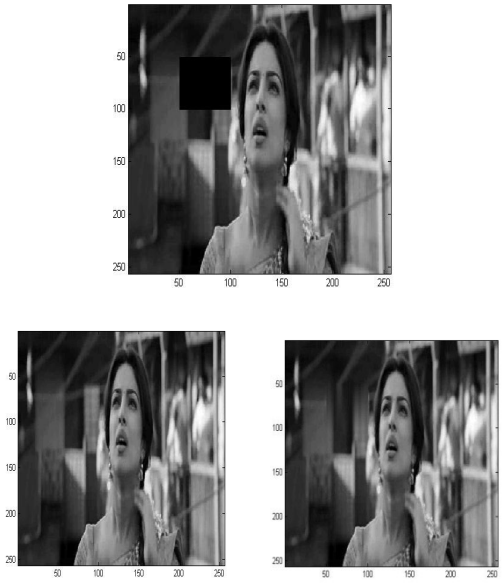

Fig .8 Results for Compressed Sequence (Bottom right is proposed and bottom left is as per algorithm in [11]).

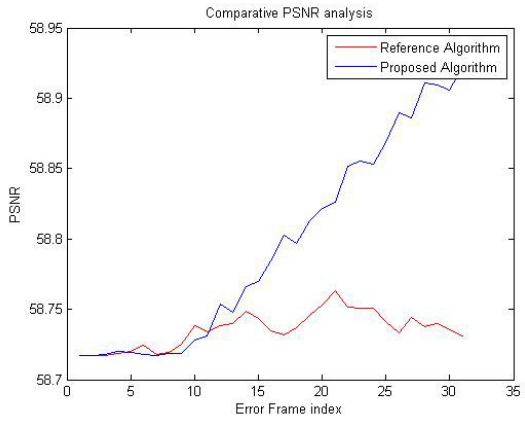

Fig.9 Comparison between algorithm given in [11] and proposed algorithm.

4. CONCLUSION

In this paper, we propose a bi-directional temporal error concealment algorithm. The proposed method exploits the information from previous frame and next frame to estimate pixel and avoids blocking artifacts by smoothing filter function. We have applied the method on RAW and compressed video sequence and it clearly indicates that the method give comparative results as compare to existing methods in terms of PSNR and visual quality. Moreover, the method is also efficient in stopping error propagation. Future work includes to use spatial and temporal information adaptively to reconstruct lost pixel information

REFERENCES

[1] Y. Wang, S. Wenger, J. Wen and A. Katsaggelos, "ErrorResilient Video Coding Techniques," IEEE Signal Processing Magazine, pp. 61-82, July 2000.

[2] M.E. Al-Mualla, N. Canagarajah and D. R. Bull, "Temporal error concealment using motion field interpolation," IEEE electronic Letters, pp. 215-217, Feb. 1999.

[3] M.E. Al-Mualla, N. Canagarajah and D.R. Bull, "Multiple reference temporal error concealment," Proc. IEEE ISCAS2001, vol. 5, pp. 149-152.

[4] S.H. Lee, D.H. Choi and C.S. Hwang, "Error concealment using affine transform for H.263 coded video transmissions," Electronics Letters, 37(4), pp. 218-220, Feb.2001.

[5] Yu Chen, Keman Yu, Jiang Li, Shipeng Li, "An Error Concealment algorithm for entire frame loss in Video transmission,"

[6] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, "An Error Concealment Algorithm for Streaming Video," Proc.IEEE ICIP 2003.

[7] Q. Peng, T.W. Yang and C.Q. Zhu, "Block-based temporal error concealment for video packet using motion vector extrapolation," IEEE Communications, Circuits andSystems and West Sino Expositions, 2002.

[8] Girod. B, "Efficiency analysis of multi-hypothesis motion compensated prediction for video coding," IEEE Transactions on Image Processing, vol. 9, no. 2, pp.173 –183, Feb. 2000.

[9] Warhade, Krishna K., S. N. Merchant, and Uday B. Desai. "Shot boundary detection in the presence of illumination and motion." Signal, Image and Video Processing 7.3 (2013): 581-592.

[10] Parul Arora Bhalotra, Bhushan D. Patil, "Video Shot Boundary Detection using Ridgelet Transform", Springer AISC series, Vol- 249, pp 163-171, 978-3-319- 03094-4, Sep 2013.

[11] Branislav H., Jan M. and Stanislav M., "Extended error concelmant algorithm for intra frame in H.264/AVC", Acta Electrotechnica et Informatica, Vol. 10, No. 4, 2010, pp. 59–63