Synthesis Analysis Methods for Underwater Video Compression with Tensor Based Minimized Side Information

A. Robert Singh, Suganya A.

Abstract: ***Synthesis analysis is a common approach used to compress videos with more amounts of dynamic textures. Underwater videos contain more moving species captured by moving camera. These kinds of videos have two types of motion registered by both the species and the camera. In this paper, tensor, an N-way representation of data is used to store the side information obtained from the synthesis analysis approach. The Low multilinear rank approximation (LMLRA) with error correction using residual tensor is applied on the side information to reduce the memory space for side information. The host encoder in synthesis analysis approach plays an important role in providing high compression rate with minimal loss and hence H.265 is used as the host encoder. The results show that the proposed method achieves highest compression ratio with minimal loss due to distortion and saved bit rate which is highly consumed by dynamic textures.***

Keywords: LMLRA, residual tensor, side information, Video compression.

I. INTRODUCTION

Video compression [1] is an essential process for video storage, video conferencing, surveillance and other compressed video content processing like recognition of application specific components. In this paper, the underwater video database named Reefvid [2] is used to test the proposed video compression method. Synthesis analysis based video compression [3] is the approach for video compression by treating the video as some Group of Pictures (GoP) in a particular format. Then, the video frames in a GoP are treated as I-frames which are intra coded using a host encoder, P-frames that are coded regarding the I-frames and B-frames that are coded concerning the I-frames and P-frames in both directions. The general architecture of the synthesis analysis based video compression is given in Fig.1.

P-frames are applied with appropriate image segmentation techniques and the obtained segments are classified into static and dynamic segments depending on the level of motion registered by that segment. The segments are either coded by the host encoder or mapped by some motion model. The reconstruction of these segments is using the side information to improve the reconstructed contents. The advantage of using the host encoder is maintaining high quality and high compression ratio for I-frames as well as for the segments.

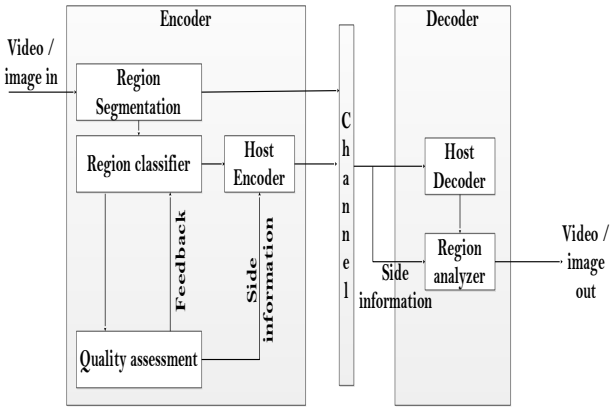

Fig.1 Video codec using synthesis analysis

Shruti Bansal et al. [3] proposed a dynamic texture synthesis for video compression. The dynamic texture segmentation has two steps called correspondance analysis to find the relationship between the find the optical flow of the motion between the frames and optic flow redsidue that is used to improve the performance of the seg,ment reconstruction. H.264/AVC is used as the host encoder in this paper. This method gives better compression ratio than H.264. Jha et al. [4] proposed a dynamic emprical mode decomposition method (DEMD) to encode P-frames and B-frames of a GoP. H.264 is the host encoder and wavelet decomposition is used on residual image.The method outperforms the H.264.

In this paper, a synthesis analysis based video compression method is proposed using tensor as a representation to store the intermediate data such as segments of different types. The tensor representation reduces the amount of side information and useful in video compression [14],[15].

II. PROPOSED METHOD

The pattern of GoP for video compression is IPPBPP and the entire video sequence is divided into number of GoPs of this pattern. The I-frame is directly coded by the host encoder H.265 using the intra-mode coding [5]. The P-frames are coded with reference to the I-frame using a motion model. The B-frame is coded with reference to the P-frames in both the directions. The overall architecture of the proposed method is given in Fig.2.

Revised Manuscript Received on December 15, 2019.
***** Correspondence Author
Dr. A. Robert Singh*, School of Computing, Kalasalingam Academy of Research and Education, Anand Nagar, India. Email: robertsinghbe@gmail.com
Suganya A., School of Computing, Sastra Deemed to be University, Thanjavur, India. Email: suganyarobert@gmail.com.

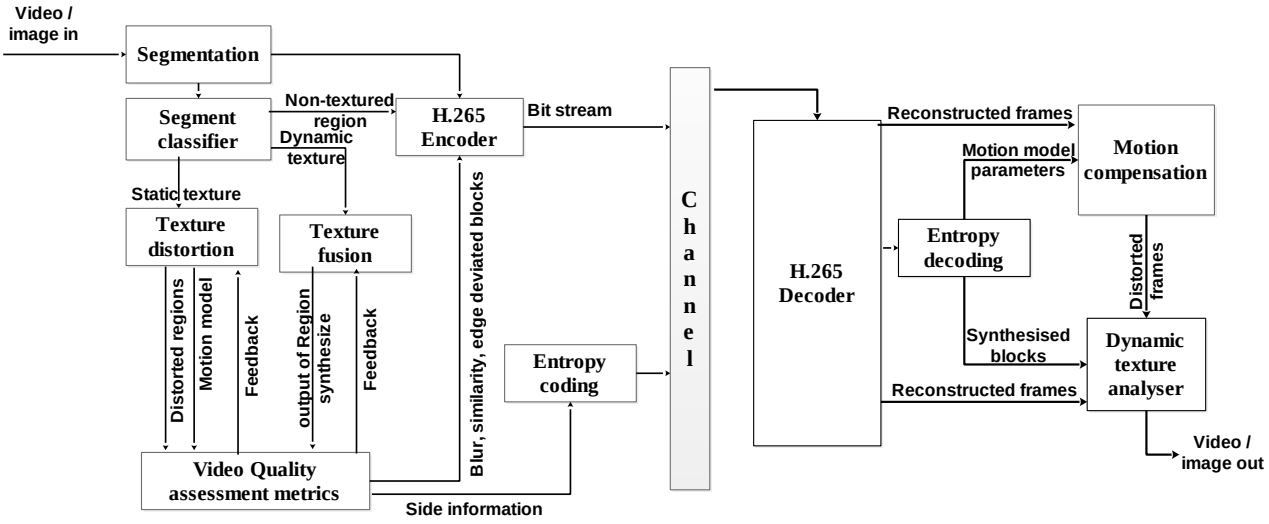

Fig.2 Architecture of proposed method

The P-frame is segmented using a Fuzzy C-means clustering method [9] with local complement membership and local data distances [6] that makes decision on the biased segmentation components. After finding the segments, they must be classified as of whether they are static, dynamic or a non-texture segment [10]. This classification is done using sub-bands of wavelets. If the magnitude of the sub-band of the maximum number of voxels is less than a given threshold value, then it is known as the non-texture segment and directly coded by H.265 encoder. Otherwise, it is considered as a textured segment. Further, the textured segments are divided into static and dynamic textures. Let R be the set of regions, for every $\mathrm{R_n} \in \mathrm{R}$, the type of segment to be determined. Each voxel in a position of (x,y) in $\mathrm{R_n}$ is checked for the thresholding condition $\mathrm{X(x,y)}$ of 8-subband wavelet as given in Eq.(1).

$$X(x,y) = \begin{cases} 1, & \sum_{i=0}^{8} A_i\left(R_n(x,y)\right) > th \\ 0, & otherwise \end{cases} \quad (1)$$

where $A_i\left(R_n(x,y)\right)$ is the amplitude of $i^{th}i^{th}$ subband of $R_n(x,y)$. Based on this matrix, the segment is classified whether it is a non-textured segment or not using Eq.(2).

$$\frac{\sum_{R_n}(X(x,y)|X(x,y)=1)\times 100}{\|R_n\|} > \propto \quad (2)$$

where $X(x,y)$ is the minimum percentage of voxels in the textured segment that is empirically derived. $\|R_n\|$ is the number of voxels in the segment. Based on the type, the sparse tensor that is identified by the previous step is divided into two sparse tensors that consist of textured and non-textured tensors respectively.

2.1 Classification of textured segments

Once textured segment is identified, it must be classified as whether it is a static or dynamic segment. This classification is done by evaluating the motion model of the segments, because dynamic texture will make uneven and difficult motion models. This is found by motion between the current textures segments and other near-by frames. Here, the textured segments are divided into 4×4 blocks and motion estimation is used with reference to neighbor frames. The motion of static segments is regular and slow. This is identified by analyzing motion vectors and differences based on motion-compensation. In other words, if the difference between the original segment and segment from motion-compensation is small, then the texture is static texture. Let S_O be the segment to be classified, S_{MC} be the output of motion compensation with neighbor frame, B_i be the 4×4 block, and V_i be the motion vector between the reference frame's block and the block to be coded, then the conditions for static and dynamic textures are given in Eq.(3).

$$T[S_0] = \begin{cases} static, & if \ \frac{|S_O(x,y)-S_{MC}(x,y)|}{\|S_O\|} < th_{s1} \\ & and \\ dynamic & \sum_{B_i \in S_O} abs\left(\frac{d^2(V_i)}{dx^2}\right)+abs\left(\frac{d^2(V_i)}{dy^2}\right) > th_{s2} \\ & otherwise \end{cases} \quad (3)$$

The output of texture classification is given in Fig.3. Green colored segments correspond to the dynamic texture, yellow colored segments correspond to the static textures, and red color represents non-textured segments that are directly coded by H.265 [13]. At this stage, the tensor that is used to represent textured segments is further divided into two tensors that contain static and dynamic texture segments respectively.

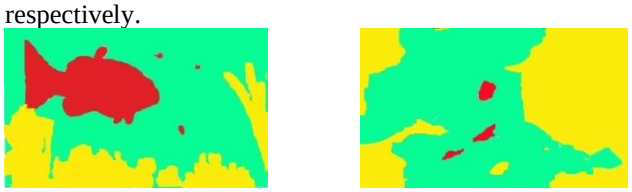

Fig.3 Output of segment classification

2.2. Texture fusion for motion model

The standard video compression algorithms are using motion estimation method to correlate the contents temporally to reduce the redundancy between frames. Transformation based on 2D-projection is generally used to perform motion estimation. In this paper,

a bi-directional texture fusion is used to process static textures in each B-frame. The motion parameters are computed using bilinear motion model with the least-square method is used to map translation of static textures. The bilinear motion model with eight parameters is given in Eq.(4).

$$\begin{aligned} x_1' &= b_1 + b_2x_1 + b_3y_1 + b_4x_1y_1 \\ y_1' &= b_5 + b_6x_1 + b_7y_1 + b_8x_1y_1 \end{aligned} \quad (4)$$

in which, (x_1', y_1') and (x_1, y_1) are the voxels of reference and current frames respectively. With more than four couple of voxels, the eight parameters will be calculated. The texture fusion process consists of three steps: 1) determine a motion vector, 2) estimate eight motion parameters and 3) estimation of distortion static textures.

a. Find motion vector

The motion of the current frame is to be estimated regarding the two nearest reference frames in both directions. Motion vector between the reference frames and the current frame within the static texture is calculated by MSE metric with a block size of 8×8. Here a weight matrix is used to represent the center of each block.

b. Find eight motion parameters

It is assumed that the motion vector records the movement of each block that is identified from the weight matrix. So Eq. (5) can be extended as Eq.(6).

$$\begin{cases} x_1' = x_1 + u \\ y_1' = y_1 + v \end{cases} \quad (5)$$

$$\text{where } \begin{bmatrix} u \\ v \end{bmatrix} = P.b \quad (6)$$

The displacement vector $\begin{bmatrix} u \\ v \end{bmatrix}$ is further represented as d for simplification. P is the matrix of sub-blocks and b is the vector of motion parameters as shown in Eq. (7) and (8) respectively.

$$P = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_1 & y_1 & x_1y_1 \end{bmatrix} \quad (7)$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{bmatrix} \quad (8)$$

The vector of eight parameters, $[b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8]^T$ can be calculated by the motion model with reference to the mean of least-squares method as given in Eq.(9).

$$b = (P^T WP)^{-1} P^T W.d \quad (9)$$

Here, W is the weight matrix calculated in the previous step. Using the motion parameters, the motion estimated vector d' can be computed. the comparison of d and d' concerning a maximum couple of voxels is calculated as given in Eq.10

$$(\bar{x}, \bar{y}) = \left| \max_{(x',y')} \left((x' - x)^2 + (y' - x)^2 \right) \right| \quad (10)$$

(x, y) and (x', y') are the entries in d and d' respectively. This is an iterative process until the maximum value is less than a threshold value.

c. Distortion of static textures

The static textures are reconstructed by distortion of the reference frame. The worthiness of all static textures (distorted) is evaluated by a set of video quality appraisal methods. In this paper, two among every seven frames are used as reference frames. The reference frame with maximum distorted blocks is selected for video quality appraisal. Based on the result of this appraisal, a new set of motion parameters are calculated for static textures which allows multiple simultaneous motions. This process is continued until there is no more static texture to be coded/ distorted regarding a reference frame.

2.3. Dynamic texture fusion

In conventional video compression methods, the coding of dynamic textures is a challenging process as they consume a significant part of bit rate. In this paper, a LMLRA analysis using tensor representation is used to model the dynamic textures. This is also known as Tucker decomposition. The 3D YCbCr representation is preferred in this paper than 4D RGB [8] to reduce memory space, and it ensures relatively minimal prediction error than other representations. This is a compact representation of frames with a single plane representation of the luminance-chrominance channel. Let us consider the set of frames as a tensor $\mathbb{T}$ be $(T_1 \times T_2 \times \ldots . T_n)$ of order n, let τ be the dimension that represents the temporal index. If there are t frames in a color video with the frame size of of $M \times N$, there are four dimensions (n=4) for the tensor that are represented as given in Eq. (11) to Eq.(14).

$$T_1 = M \quad (11)$$

$$T_2 = N \quad (12)$$

$$T_3 = t \quad (13)$$

$$T_4 = 3 \quad (14)$$

The fourth dimension represents the three color channels. The dynamic texture synthesis is done by two steps that give compact representation using tensor. The first step decomposes the tensor into a smaller core tensor (S) and n factor matrices $(U^i, i = 1,2, \ldots n)$ that are also known as orthogonal matrices. The decomposition is shown in Eq.(15).

$$\mathbb{T} = S \times_1 U^1 \times_2 U^2 \times_3 U^3 \ldots \times_\tau U^\tau \ldots . \times_n U^n \quad (15)$$

U^τ is the factor matrix that corresponds to the decomposed signal of the input segments in temporal axis.

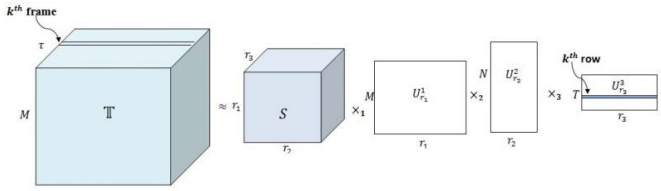

Fig.4 Representation of product between Core tensor and factor matrices to approximate original tensor.

In the proposed method, the value of t is five that represents the B-frame count in the video coding scheme. The k^{th} frame of $\mathbb{T}$ can be reconstructed by the multiplication of $S, U^1_{r_1}, U^2_{r_2}$ and k^{th} and the k^{th} row of $U^3_{r_3}$ as shown in Figure 4. Before the decomposition, a temporal average of pixels is identified in the temporal direction as shown in Eq.(16). which is used for maintaining the zero-mean element in the temporal axis.

$$TA_{i_1,i_2,i_3\ldots i_{\tau-1},i_{\tau+1}\ldots i_n} = \frac{1}{I_\tau}\sum_{k=1}^{I_\tau}(\mathbb{T})_{i_1,i_2,i_3\ldots i_{\tau-1},k,i_{\tau+1}\ldots i_n} \quad (16)$$

The autoregressive model is the famous moving average model that can construct a set of future temporal components from existing inputs of experience. In this proposed model, an autoregressive model of order one is used to define the moving average model.

III. PERFORMANCE ANALYSIS

The proposed method is compared for PSNR, mean structural similarity (MSSIM), bit rate reduction and compression ratio against H.264/AVC [12] and H.265/HEVC [11], [13]. The proposed system is tested with common test videos and some underwater videos from Reef-vid database (reefvid database). Experiments are conducted on a PC with Intel Core i3-2328M CPU (2.20 GHz), 6 GB RAM, and a 64-bit Windows 7 operating system. Fig.5 compares the bit rate saving between H.264, H.265 and the MPEG codec libvpx. The result shows that FVMDEC achieved a high bit rate saving than the other methods.

The PSNR values for different Reefvid videos are compared with H.264 and H.265 as shown in Table 1. The obtained results show that the proposed method achieves quality as equal as H.265. This is because of the FVMs used to improve the quality of reconstructed videos.

Table .1 PSNR comparisons between H.264, H.265 and FVMDEC for Reefvid database

File name	PSNR		
	H.264	**H.265**	**FVMDEC**
Clip 81	40.8053	43.64195	**43.33683**
Clip 114	31.76553	**31.21367**	30.38968
Clip 134	36.61402	**39.62066**	39.33939
Clip 2	33.43607	41.1682	**42.94047**
Clip 27	33.992	**35.55527**	35.47199
Clip 125	31.63212	32.46141	**33.56767**
Clip 294	41.50682	44.83933	**45.86249**
Clip 295	30.01561	38.97665	**39.67847**
Clip 297	36.75129	35.01432	**35.69376**
Clip 300	37.43889	40.78341	**41.09596**
Clip 303	32.79444	**41.52687**	41.52594
Clip 304	30.20816	42.90794	**44.00454**
Clip 305	31.88442	34.37441	**35.28965**
Clip 307	32.72902	**37.68307**	37.46131
Clip 308	38.7973	**41.13771**	40.73962
Clip 428	32.81715	**35.78717**	35.355
Clip 429	36.77007	43.03734	**44.29404**
Clip 472	32.8252	36.62318	**37.72798**
Clip 474	38.1077	40.59747	**41.04984**
Clip 476	31.71507	**32.1154**	31.94546
Clip 477	37.13672	44.95393	**45.47378**
Clip 428	32.72426	**35.85787**	34.899
Clip 429	42.35082	42.97327	**43.26176**
Clip 472	37.68415	**44.45834**	43.92125
Clip474	30.59986	35.83046	**37.15281**

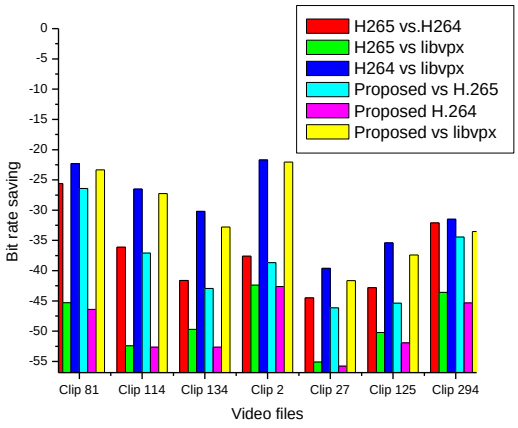

Fig.5 Comparison of bit rate saving between H.264, H.265, libvpx and FVMDEC

IV. CONCLUSION

This paper proposes a new synthesis analysis based video compression method for underwater video coding. The GoP pattern has sufficient number of frames to be processed. The host encoder H.265 ensures a high compression ratio of I-frame and other essential information. The synthesis analysis based video compression ensures a high bit rate saving than the other methods for Reefvid videos as well as common videos for video compression. The quality of the decompressed video is evaluated by PSNR and SSIM and it is evident that the proposed method ensures high video quality than others.

REFERENCES

1. H. Mukhtar, A. Al-Dweik and M. Al-Mualla, "Content-aware and occupancy-based hybrid ARQ for video transmission," *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Abu Dhabi, 2016, pp. 1-4.
2. Reefvid underwater video database, `http://www.reefvid.org/`
3. S. Bansal, S. Chaudhury and B. Lall, "Dynamic texture synthesis for video compression", *National Conference on Communications (NCC)*, 2013, pp. 1-5.
4. M. K. Jha, B. Lall and S. D. Roy, "Video Compression Scheme Using DEMD Based Texture Synthesis", *Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, Hubli, Karnataka*, 2013, pp. 90-93.
5. Abhilash Antony and Sreelekha G. 2017, "HEVC-based lossless intra coding for efficient still image compression", *Multimedia Tools Appl.*, 76(2), pp. 1639-1658, (2017).
6. R. R. Gharieb, G. Gendy and A. Abdelfattah, "Image Segmentation Using Fuzzy C-Means Algorithm Incorporating Weighted Local Complement Membership and Local Data Distances", *World Symposium on Computer Applications & Research*, pp. 6-11, (2016).
7. Jiayu Chen, Jinguo Liu, Jianzhi Li, Dezhu Kong and Da Yu , "A Video Synthesis Method For Flow Patterns", *Proc. of 2nd IEEE International Conference on Network Infrastructure and Digital Content*, pp 303 – 307, (2010).
8. R. Costantini, L. Sbaiz and S. Susstrunk, "Higher Order SVD Analysis for Dynamic Texture Synthesis," in *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 42-52, Jan. 2008. doi: 10.1109/TIP.2007.910956.
9. Yang Jiao, Jianshe Wu, Licheng Jiao, An image segmentation method based on network clustering model, *Physica A: Statistical Mechanics and its Applications*, Vol 490, 2018, Pages 1532-1542, ISSN 0378-4371, https://doi.org/10.1016/j.physa.2017.08.118.
10. A.T Smith, W Curran, Continuity-based and discontinuity-based segmentation in transparent and spatially segregated global motion, *Vision Research*, Volume 40, Issue 9, 2000, Pages 1115-1123, ISSN 0042-6989, https://doi.org/10.1016/S0042-6989(00)00013-4.
11. Seeling P, Reisslein M. Video Traffic Characteristics of Modern Encoding Standards: H.264/AVC with SVC and MVC Extensions and H.265/HEVC. *The Scientific World Journal*. 2014;2014:189481. doi:10.1155/2014/189481.
12. H.264/AVC(2017).http://www.videolan.org/developer/x.264.html
13. H.265/HEVC(2017).http://www.videolan.org/developers/x265.html
14. Suganya A, Dejey Dharma, "Compact video content representation for video coding using low multi-linear tensor rank approximation with dynamic core tensor order", *Computational and Applied Mathematics*, July 2018, Volume 37, Issue 3, pp 3708–3725.
15. Suganya Athisayamani, Dejey Dharma, "A Novel Video Coding Framework with Tensor Representation for Efficient Video Streaming", *Wireless Personal Communication*, 2019, DoI: https://doi.org/10.1007/s11277-019-06704-4.

AUTHORS PROFILE

Dr. A. Robert Singh is working as an Assistant Professor in Kalasalingam Academy of Research and Education. He has completed his Ph.D in the field of routing algorithm for AMI in smart grid. He has published papers in reputed journals. His areas of interest are soft computing and Internet of Things. He is a life time member of IEI and IEEE.

Suganya A. is working as an Assistant Professor in Sastra Deemed to be University. Her research areas are digital image and video processing, and Internet of Things.