Virtual Frames as Long-Term Reference Frames for HEVC Inter-Prediction

Buddhiprabha Erabadda, Thanuja Mallikarachchi, Gosala Kulupana and Anil Fernando
Centre for Vision Speech and Signal Processing, University of Surrey, United Kingdom
Email: {e.harshani, d.mallikarachchi, g.kulupana, w.fernando}@surrey.ac.uk

***Abstract*—High Efficiency Video Coding(HEVC) employs both past or future frames when encoding the current frame in a video sequence. This paper proposes a framework for using virtual reference frames, to achieve increased coding gains in the long-term for repetitive scenes in static camera scenarios.**

I. Introduction

High Efficiency Video Coding (HEVC) reports $\sim$50% coding efficiency improvement compared to its predecessor, H.264/AVC [1]. This is achieved with the use of novel coding modes and the quadtree-based partitioning structure introduced in HEVC. However, advancements in multimedia technologies, and interactive media applications (e.g., virtual reality, 6-DOF video, autonomous driving etc.), demand further coding efficiency improvements to enable better utilisation of bandwidth and storage.

HEVC uses two types of predictions: intra-prediction and inter-prediction that exploit the spatial and temporal redundancies, respectively. For inter-prediction, there are two variations depending on the relative Picture Order Counts (POCs) of the reference pictures. If the POC is within the current Group of Pictures (GOP), the reference picture is regarded as a short-term reference picture. On the other hand, if its outside the range of current GOP, it is regarded as a long-term reference picture. HEVC allows 32 reference pictures for long-term reference, however, the standard does not specify how these pictures are selected.

In this context, this paper analyses the existing work that utilises the long-term reference pictures to improve coding gains. Then, a framework for calculating long-term reference frames using statistical and machine-learning approaches is proposed for static-camera applications with repeated shots.

The rest of the paper is organised as follows. Sec. II provides an overview of existing work on long-term reference pictures and Sec. III discusses the proposed framework. Finally, Sec. IV concludes and discusses future work.

II. Related Work

Recent literature reports numerous work that propose long-term reference frame selection methods for improved coding efficiency in HEVC inter-prediction. Zuo and Lu [2] propose a clustering-based approach for sequences with repeated shots.

This work was supported by the CONTENT4ALL project, which is funded under European Commission's H2020 Framework Program (Grant number: 762021).

Fig. 1. Repeated scenes in *synthetic.yuv*

In their work, the similar shots are clustered and a representative frame is taken as a long-term reference picture for the frames in the cluster. Their method shows a 14.14% coding gain on average, when compared with the HEVC reference software with adaptive GOP, where a scene change frame is encoded as an intra frame. This method uses an existing frame from the sequence, which might not be the optimal representation of the cluster.

Constructing a background frame to be used as a reference frame is widely used in surveillance applications, where camera is mostly static and the background doesn't significantly change over time. In this context, Zhang et. al [3] propose a method where two types of frames: *background frames* and *difference frames* are used for improved coding gains. Similar work have been proposed in [4], [5] which exploit the long-term correlation in the frames in surveillance and video conferencing applications by modelling the background as a reference frame. These methods focus only on surveillance applications with near-zero camera movements, that typically have only one background that does not change over time.

However, with the improvements of machine learning algorithms(including clustering and object detection algorithms), it is feasible to further exploit the long-term redundancy in video sequences with repeated shots, such as television dramas.

III. Proposed Framework

As explained in the previous section, the coding efficiency gains in the proposed framework are expected to be achieved with repeated shots in a sequence that employs static cameras. An example of such a sequence would be a television drama that has a limited number of shooting locations(referred to as *scenes* in the following text), where the locations are repeated in the episode multiple times.

We synthetically created such a sequence(*synthetic.yuv*) using three HD sequences by alternating the frames as depicted in Fig. 1. Here, *Xi*, *Yi*, and *Zi* correspond to blocks of 50 frames, each starting from the $(i-1)th$ location in the HD sequences *band*, *cafe*, and *beergarden*, respectively. The concepts in the proposed method are explained with reference to this synthetic video.

Typically, there are multiple shots of a scene in a given television episode. However, the number of scenes and the

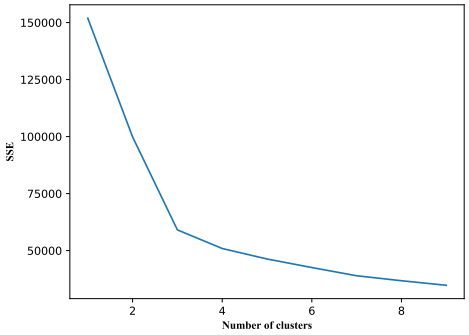

Fig. 2. Determining number of clusters with elbow method for *synthetic.yuv*

frames that belong to a particular scene are not known prior to encoding. It is possible to manually label the frames, however, it is a tedious task that requires a lot of work. Instead, simple statistical calculations and machine learning clustering algorithms can be employed to automatically identify the information related to the scenes in the sequence.

To prove the feasibility of the stated method, we used K-means clustering [6] with elbow method [7] to automatically identify the optimal number of clusters for the synthetic video.

Following the elbow method, Fig.2 shows the Sum of Square Error (SSE) vs. the number of clusters (*N*, from 1 to 10) for the synthetic video. The elbow becomes visible when $N = 3$, indicating the optimal number of clusters in the synthetic sequence, which corresponds to the number of sequences employed to create the synthetic video. Once this is identified, clustering is performed using K-means clustering.

As the first step of creating the virtual frame for a given cluster, the background for a scene should be extracted. Assuming the camera is static, the background is common across all frames in a given scene. The running average is calculated to identify the common background for each cluster.

In video coding, *I* frames are independently coded requiring the highest number of bits, whereas both *P* and *B* frames have higher compression requiring lesser number of bits. A scene change leads to intra-coded blocks in the scene change frame, which is unfavourable for low-bit rate encoding. However, with the introduction of Virtual Reference Frames (VRF), the scene change frame gets the background of the scene as a reference picture, as opposed to having a frame from a different scene. This enables improved coding efficiency.

Fig. 3 shows a sequence with *I*(*black*) and *P*(*cyan*) frames. For illustration purposes, it is assumed that a scene change occurs at each of the *I* frames. When this is encoded without long-term reference frames, it requires 6 *I* frames.

The updated coding structure with VRFs is depicted in Fig. 4. Here, frames marked as *V* denote the VRFs and it can be observed that some *I* frames are encoded as *P* frames as they can now be predicted with the corresponding VRF. Now only three *I* frames are required, reducing the number of *I* frames by 50% in this example.

To calculate the impact of this phenomenon, we encoded the *synthetic.yuv* enabling rate control(at 4000 kbps) for *low-delay* configuration. The same sequence encoded with VRFs

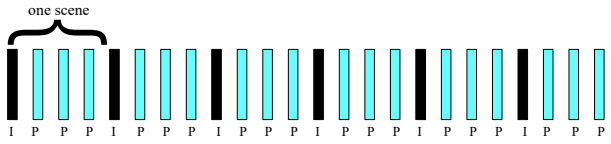

Fig. 3. I frames at scene change locations

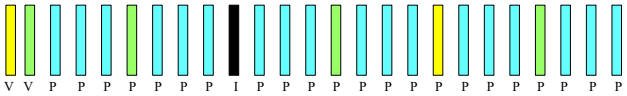

Fig. 4. VRFs added and I frames substituted with P frames

require three VRFs(corresponding to each HD sequence). The VRFs were calculated using the method explained above and they were encoded as *I* frames to calculate the bitrate required. Once VRFs are appended to the sequence, the frames at scene change locations require a lesser number of bits since the corresponding VRF is available as a reference frame. As a result the bitrate should reduce to 3733.8 kbps as opposed to 4000 kbps without VRFs. This represents more than 6% bitrate reduction.

IV. Discussion and Future Work

Utilising a background frame as a reference frame in surveillance videos, where the camera is mostly static is becoming a popular approach to improve the coding efficiency. As explained in the previous sections, it is possible to extend this phenomenon to other applications that have repeated shots with a static camera, such as television series episodes, where the shots are limited to a few scenes. This allows saving bit rate using *P* frames in the places where scene changes occur, which are typically encoded as *I* frames.

The future work will focus on identifying objects in a given cluster using object segmentation techniques to create rich VRFs to facilitate further coding gains.

References

[1] G.J.Sullivan, J. Ohm, W-J Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] Xu-guang Zuo and Lu Yu, "Long-term prediction for hierarchical-B-picture-based coding of video with repeated shots," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 3, pp. 459–470, 2018.

[3] Xianguo Zhang, Luhong Liang, Qian Huang, Yazhou Liu, Tiejun Huang, and Wen Gao, "An efficient coding scheme for surveillance videos captured by stationary cameras," in *Visual Communications and Image Processing 2010*. International Society for Optics and Photonics, 2010, vol. 7744, p. 77442A.

[4] Xianguo Zhang, Tiejun Huang, Yonghong Tian, and Wen Gao, "Background-modeling-based adaptive prediction for surveillance video coding," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 769–784, 2013.

[5] Xianguo Zhang, Yonghong Tian, Tiejun Huang, Siwei Dong, and Wen Gao, "Optimizing the hierarchical prediction and coding in HEVC for surveillance and conference videos with background modeling," *IEEE transactions on image processing*, vol. 23, no. 10, pp. 4511–4526, 2014.

[6] John A Hartigan and Manchek A Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[7] Trupti M Kodinariya and Prashant R Makwana, "Review on determining number of Cluster in K-Means Clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.