IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 22 September 2022, accepted 2 November 2022, date of publication 4 November 2022,
date of current version 15 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219861

RESEARCH ARTICLE

Deep Learning and Bidirectional Optical Flow Based Viewport Predictions for 360° Video Coding

JAYASINGAM ADHURAN[1], (Member, IEEE), GOSALA KULUPANA[1], AND ANIL FERNANDO[2], (Senior Member, IEEE)

[1]Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, U.K.
[2]Department of Computer and Information Science, University of Strathclyde, G1 1XQ Glasgow, U.K.

Corresponding author: Jayasingam Adhuran (j.adhuran@surrey.ac.uk)

ABSTRACT The rapid development of virtual reality applications continues to urge better compression of 360° videos owing to the large volume of content. These videos are typically converted to 2-D formats using various projection techniques in order to benefit from ad-hoc coding tools designed to support conventional 2-D video compression. Although recently emerged video coding standard, Versatile Video Coding (VVC) introduces 360° video specific coding tools, it fails to prioritize the user observed regions in 360° videos, represented by the rectilinear images called the viewports. This leads to the encoding of redundant regions in the video frames, escalating the bit rate cost of the videos. In response to this issue, this paper proposes a novel 360° video coding framework for VVC which exploits user observed viewport information to alleviate pixel redundancy in 360° videos. In this regard, bidirectional optical flow, Gaussian filter and Spherical Convolutional Neural Networks (Spherical CNN) are deployed to extract perceptual features and predict user observed viewports. By appropriately fusing the predicted viewports on the 2-D projected 360° video frames, a novel Regions of Interest (ROI) aware weightmap is developed which can be used to mask the source video and introduce adaptive changes to the Lagrange and quantization parameters in VVC. Comprehensive experiments conducted in the context of VVC Test Model (VTM) 7.0 show that the proposed framework can improve bitrate reduction, achieving an average bitrate saving of 5.85% and up to 17.15% at the same perceptual quality which is measured using Viewport Peak Signal-To-Noise Ratio (VPSNR).

INDEX TERMS 360° video, perceptual coding, Regions of Interest, viewport prediction, Versatile Video Coding.

I. INTRODUCTION

In recent years, virtual reality (VR) technology has rapidly grown in public markets, providing solutions to immersive in-home experiences and elevating the standards of media consumption [1], [2], [3]. In addition to the entertainment sector, VR technology also supports other business endeavours such as travel, education, and real estate. Consequently, the proliferation of high-resolution 360° videos required to boost VR-based multimedia applications demands higher bandwidth requirements. Therefore, there is a great need for efficient compression of such video content.

A 360° video can be regarded as a sequence of surface information that encloses a point source. The viewpoints on the virtual 360° surface are subjected to change as a user varies the head position. In the event that these viewpoints are uniformly distributed and placed at a constant distance from the point source, then the 360° surface becomes isotropic, hence can be defined as a spherical surface. A 360° surface can be represented by the spherical coordinate system and its parameters latitude θ $[-\pi/2, \pi/2]$, longitude ϕ $[-\pi, \pi]$ and unit radius r. Moreover, 360° videos are converted to 2-D representations, mostly the EquiRectangular Projection

The associate editor coordinating the review of this manuscript and approving it for publication was Khoa Luu.

(ERP) format, in order to use 2-D video encoders to compress the video contents [4]. However, in 360° videos, the Field of View (FOV) of the user encloses only a portion of the spherical information, also known as the viewport which is a rectilinear image generated from the user's head position. Because the remaining surface information is redundant to the user, it is expected that video compression technologies will effectively remove such information and save bitrates.

Video-compression technologies continue to evolve and facilitate new video-communication trends. The Versatile Video Coding (VVC) standard [5] is the latest initiative of the Joint Video Experts Team (JVET) which introduces numerous coding tools to improve the video compression efficiency [5], [6], [7], [8], [9], [10], [11]. VVC not only target high coding gains, but also targets versatility, and as such it supports encoding of various types of videos such as natural videos, screen content, Standard Dynamic Range (SDR), High Dynamic Range (HDR) and 360° videos and video formats such as 4:2:0, 4:4:4, lossless video formats, etc. In order to support 360° videos, VVC introduces specific coding tools including several re-projection, packaging and padding tools [12]. Particularly for ERP videos, VVC introduces Motion Vector (MV) wrap-around and vertical edge padding features to provide continuity in the 2-D plain as similar to that in a spherical surface [12]. The VVC Test Model (VTM) 7.0 is capable of improving the compression gains of 360° videos by 28.91% compared to its predecessor, High Efficiency Video Coding (HEVC). However, VVC does not account for the user perception when encoding an ERP video which could otherwise further improve the perceptual compression gains.

Perceptual video coding has been a vastly researched area in the video coding domain which focuses on enhancing the user perceived visual quality by improving the fidelity of human interested regions, also called the Region of Interests (ROI) [13], [14], [15]. In the context of 2-D videos, perceptual characteristics can be directly exploited using video compression techniques. However, ROI based coding can be a challenging problem for 360° videos because the FOV of the user is limited to the viewports as opposed to the conventional ERP representation of 360° videos as illustrated in FIGURE 1. Therefore, failure to utilize viewport information during the encoding of an ERP video can result in an abundance of non-observed video information being coded, incurring an additional transmission cost. Also, individual encoding of several viewports of a given ERP frame cannot address this problem because different users can opt to view different viewports. In response, this paper proposes a novel viewport dependent ERP coding framework that exploits viewport information in VVC encoding processes. To this end, in light of the success of image processing tools and data-driven technologies, this research employs a hybrid technique incorporating Gaussian filtering technique and bidirectional optical flow estimation [16], and a deep learning network constructed from the components of Spherical Convolutional Neural Network (Spherical CNN) [17] and Salient Resnet [18] in order to predict the user observed viewports. Furthermore, this paper also illustrates the generation a VVC compliant, ROI aware weightmap by non-linearly fusing the predicted viewports. Subsequently, the developed weightmap is used in the removal of spatial redundancy in the ERP videos and the optimization of the video coding parameters.

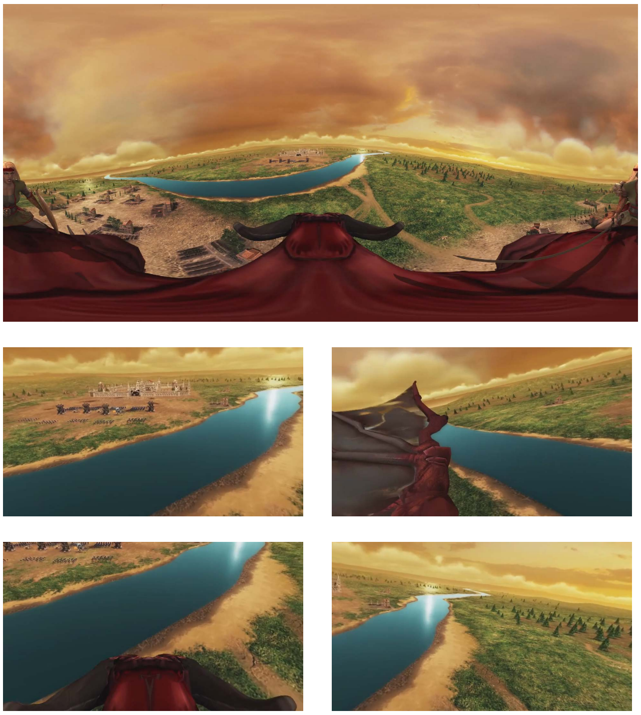


FIGURE 1. 360° video: EquiRectangular Projection (ERP) format representation (top) and four random user observed viewports (bottom).

The novel contributions provided in the proposed ERP coding framework are summarized as follows.

- Hybrid viewport prediction technology for video coding that fuses bidirectional optical flow estimation and Gaussian filtering techniques.
- Deep learning based viewport prediction technique for video coding that incorporates Spherical CNN and Salient Resnet components.
- VVC compliant weightmap derivation from non-linear fusion of predicted viewports to mask the source video.
- Application of the weightmap in Lagrange optimization and adaptive Quantization Parameter (QP) derivation for VVC.

The remainder of this paper is organized as follows. Section II describes the related work in the area, Section III describes the proposed encoding framework, and Section IV illustrates the experiments and results, followed by concluding remarks in Section V.

II. RELATED WORK

360° video coding is a growing area of research and numerous solutions have been proposed in the literature to boost the compression efficiency of the 360° videos. In general, these research works on 360° video coding can be classified into three categories namely, pre and post coding, context adaptive coding and perceptual coding. Pre and post coding primarily discusses the re-projection techniques of existing 2-D projected 360° video contents as well as the packaging mechanisms of the video frames. In contrast, in context adaptive coding, the spherical properties of the 360 videos are exploited and used by video compression tools such as quantization and motion compensation in order to improve the coding efficiency. Finally, perceptual coding addresses the deployment of user perceptual models, specifically viewport dependent encoding approaches during the video coding procedures.

The pre and post coding tools for 360° videos play a major role in the 360° video coding domain. The main functionalities of the pre-coding tools are to convert the 360° videos into 2-D space, and rearrange them in rectilinear formats which are suitable for subsequent encoding. Their respective inverse operations at the decoder are performed by the post-coding tools. In this regard, the projection and packaging techniques such as rhombus dodecahedron projection [19], CubeMap (CMP) [20], octahedron projection [21], Truncated Square Pyramid (TSP) [22], icosahedron projection [23], and rotated sphere projection [24] have been studied. In contrast to the pre-coding techniques used in the literature, the proposed framework applies a viewport based weightmap on the input ERP video frame, to provide user perception to the codec.

Context based 360° video coding incorporates spherical characteristics during the video coding processes. Moreover, it is also vital to understand that the locations and magnitudes of the reference pixels pointed in the spatial and temporal domains by the 2-D codecs may not be accurate in a spherical projected 360° video as opposed to a conventional 2-D video. In general, the spherical projected 360° videos introduce artifacts such as discontinuity in the boundaries, shape distortion and redundancy in pixel samples. These can result in the encoding of invalid pixels, spatial prediction issues, and inefficient motion estimation and motion compensation, thereby not being able to achieve the potential maximum compression efficiency. In this regard, the application of spherical objective quality metrics and related algorithms in adaptive quantization techniques, Lagrange optimization, quantization parameter optimization, residual weighting, adaptive resolution techniques and Rate-Distortion Optimization (RDO) have been studied in numerous research works [4], [25], [26], [27], [28], [29], [30]. Furthermore, few studies report that the use of spherical characteristics in motion vector candidate selection, motion compensation and pixel padding can boost coding performance in the temporal domain [12], [31], [32]. As opposed to the aforementioned studies which incorporate spherical properties in video compression processes, the proposed research incorporates perceptual characteristics to the video coding tools.

Perceptual coding in 360° videos includes ROI detection which can be very challenging because the viewports are instantly constructed based on the user's head movements. Therefore, in 360° videos, ROI can be approximated by predicting viewport information. In the context of leveraging user observed viewports, many state-of-the-art studies primarily focus on video streaming applications and place less emphasis on the encoding of video content. The literature categorizes, the viewport centric 360° video streaming/coding techniques as tiled and non-tiled approaches. Benefiting from parallel processing features, tiled approaches are mainly applicable in streaming of the 360° video contents. As such, tiled approaches either follow a scalable coding solution [33], [34], [35], [36] or assign high bitrates to the tiles that represent primary viewports [37]. Moreover, there are research works that combines both solutions by encoding the viewport dependent tiles at a higher bitrate and proving scalable support to further enhance the Quality of Experience (QoE) of users [38], [39]. Furthermore, multi-layer streaming system with base and enhancement layers have also been studied [5], [35], [40]. Although tiled based coding systems are useful for streaming purposes, reducing transmission delays, providing higher flexibility and improving the QoE, the associated coding losses remain an issue. Furthermore, viewport driven RDO strategies for 360° video streaming have also been studied [35], [36]

Coding losses can be improved using non-tiled approaches. However, viewport dependent non-scalable non-tiled driven coding approaches have not been a popular research topic owing to problems related to generalization of viewport prediction, viewport mapping with ERP video frames and associated coding delays. Among the JVET approved projection schemes for VVC, TSP [22] is the only viewport dependent projection technique that specifically prioritizes the front viewport during the packaging of 360° video in 2-D platform. Furthermore, Sreedhar et al. [41] proposed a multiple viewport resolution centric, rectilinear packaging technique for ERP in which the front viewport has been biased with higher resolution as opposed to the other viewports. These two studies make an assumption that users tend to view the front viewport more often than the other viewports. In contrast, Facebook developed Barrel layout based on AI-driven saliency maps to identify user interested regions prior to encoding [42]. Furthermore, Hu et al. [43] reports learned weights driven viewport dependent Lagrange optimization and adaptive quantization techniques at Coding Tree Unit (CTU) level which improves the perceptual compression gains of HEVC by 26%.

Building on the non-tiled approach for 360° video coding, this paper proposes several novel approaches that differ from the state-of-the-art techniques [22], [41], [42], [43], in number of ways. Firstly, the proposed research applies two different techniques for viewport prediction; a hybrid approach that combines bi-directional optical flow estimation and a Gaussian filtering technique to extract saliency features in both spatial and temporal domains and an Spherical CNN incorporated deep learning approach to obtain spherical features only in the spatial domain. Secondly, the proposed

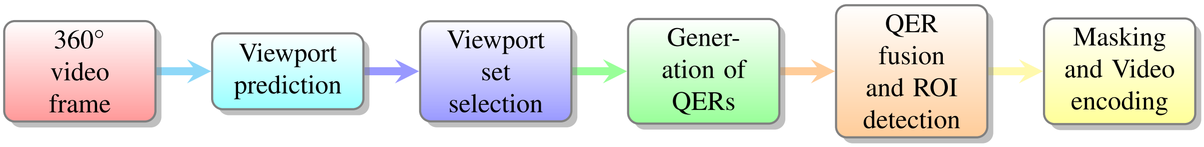


FIGURE 2. Sequential flow of the tasks in the proposed viewport dependent ERP coding framework.

research employs a Gaussian based viewport non-linear fusion technique in order to detect the ROI on the ERP video frames and generate a weightmap. Thirdly, the proposed research also adopts the generated ROI based weightmap in, 1) masking the input ERP frames, 2) adaptive quantization and 3) Lagrange optimization.

III. METHODOLOGY

A. OVERVIEW OF THE PROPOSED 360° VIDEO CODING FRAMEWORK

This paper proposes a novel ERP coding framework that exploits viewport information in 360° videos to perceptually improve the compression efficiency. The sequential flow of the tasks in the proposed framework is shown in FIGURE 2. The proposed framework first predicts a set of user observed viewports. Once a set of viewports is obtained, they are mapped onto an ERP frame to generate Quality Emphasis Regions (QERs) on the ERP frame. Subsequently, QERs are fused using Gaussian operation in order to identify the ROI on the ERP frame. Thereafter, an ROI aware weightmap is generated for each frame and it is used to mask the corresponding video frame prior to encoding using VVC. Furthermore, the proposed framework also employs an adaptive Lagrange and quantization parameter optimization techniques based on the generated weightmap in order to further improve the compression efficiency of the encoded bitstream.

B. OVERVIEW OF VIEWPORT PREDICTION TECHNIQUES

The proposed framework experiments two different approaches for predicting the viewports which would be used to detect the ROI on an ERP frame. The first approach, Viewport Prediction Hybrid (VPredHyb) applies a hybrid implementation between bidirectional optical flow estimation and Gaussian filtering technique in obtaining a set of viewports from a predefined viewport set. The magnitudes of the optical flow vectors and the filtered pixels obtained respectively are weighted and the resulting magnitude is used in the selection of the required number of viewports. The second approach called the Viewport Prediction Spherical CNN (VPredSCNN) deploys a deep learning technique incorporating Spherical CNN and Salient Resnet components to directly predict a set of viewports as opposed to the viewport prediction procedure used in VPredHyb where viewports are selected from a predefined viewport set. In VPredSCNN, a k-means clustering technique is used after the viewport prediction process in order to obtain the required number of viewports.

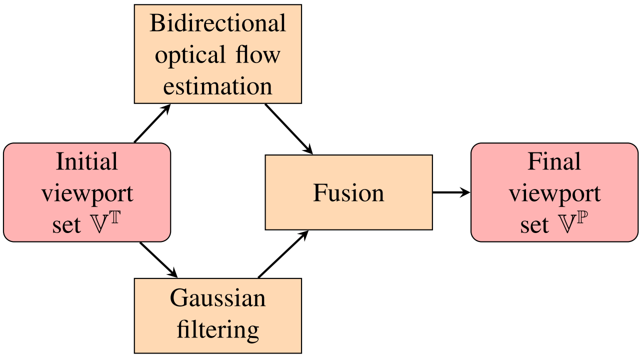

Bidirectional optical flow estimation
Initial viewport set $\mathbb{V}^{\mathbb{T}}$
Fusion
Final viewport set $\mathbb{V}^{\mathbb{P}}$
Gaussian filtering

FIGURE 3. Overview diagram of Viewport Prediction Hybrid (VPredHYB).

Viewport Generation: A user observed viewport in a 360° video frame is defined by the viewport centre (ϕ_o, θ_o), its width W_{vp}, its height H_{vp}, horizontal FOV angle F_h and vertical FOV angle F_v [44]. Here, the viewport centre (ϕ_o, θ_o) is derived from a given pair of spherical coordinates (ϕ, θ) of the 360° video frame. During viewport generation, the pixel values at location (x, y) in the ERP frame that corresponds to the sampling location (m, n) on the viewport are determined using interpolation techniques. In this context, the mapping relationship provided by [44] is used in this research for the generation of viewports.

C. VIEWPORT PREDICTION HYBRID

VPredHYB is one of the proposed viewport prediction techniques that fuses viewports predicted using bidirectional optical flow vectors and Gaussian filtered pixels to extract both spatial and temporal features. Optical flow provides a better estimate of the motion trajectory leading to a better approximation of human interest regions. As opposed to many available optical flow estimation methods, bidirectional optical flow utilizes the information from past frames as well as future frames during flow vector estimation, hence used in the proposed VPredHYB. Furthermore, a Gaussian filter is included to capture the low frequencies in the signal which are more sensitive to the human visual system.

FIGURE 3. illustrates the overview of the proposed VPredHYB technique. Although an arbitrary number of viewports can be constructed, doing so is impractical due to their high computational complexities and can also result in the generation of redundant viewports. Therefore, in VPredHYB,

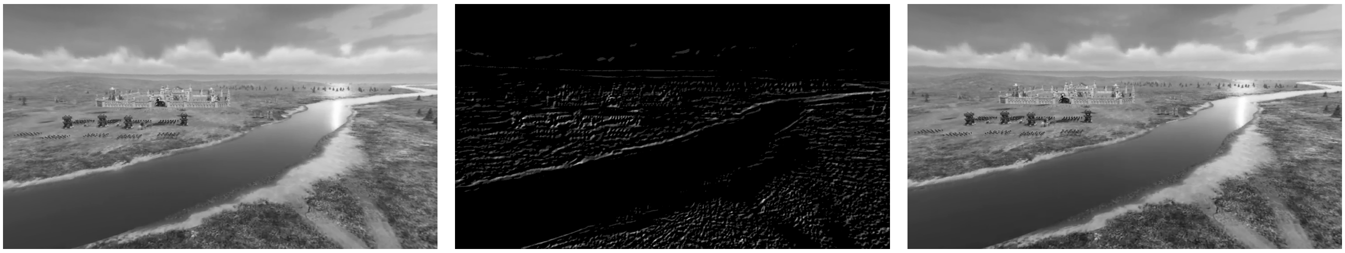

(a) Viewport at time $t0$ (b) Optical flow vectors (c) Viewport at time $t1$

FIGURE 4. Estimation of optical flow vectors using bidirectional viewports with *DragonTale* sequence at $(\phi_0, \theta_0) = (0.0°, 0.0°)$.

TABLE 1. Percentage of area of an ERP enclosed when mapping viewports onto EPR.

N^T $(\#\theta_o \times \#\phi_o)$	10 (2×5)	15 (5×3)	15 (3×5)	20 (5×4)	25 (5×5)
Area of ERP enclosed	68%	82%	85%	92%	98%

a set of viewports $\mathbb{V}^{\mathbb{T}}$ with length N^T are constructed at first. Here, the value of N^T is chosen such that the majority of the regions on the ERP can be enclosed while reducing the impact of the encoding complexity. Table 1. illustrates the percentage of the area that the number of viewports can enclose when projected onto an ERP at standardized coding resolution of 4432×2216. In the proposed VPredHYB N^T is set to 20, as it can provide a good trade off between the area enclosed and the complexity of search. Furthermore, the initial viewport centres are defined such that θ_o and ϕ_o take values in the ranges of $[-\pi/2, \pi/2]$ and $[-\pi, \pi)$ at $\pi/4$ and $\pi/2$ intervals respectively.

Each of the prediction techniques in VPredHYB is sequentially applied on all of the viewports in $\mathbb{V}^{\mathbb{T}}$. In the case of the Gaussian filtering technique, only a particular viewport from a given video frame is generated. In contrast, the viewports corresponding to the previous and subsequent frames are constructed during the estimation of optical flow vectors, hence this technique cannot be applied to the first and the last frames. Moreover, a weighting factor ξ is applied to the magnitudes of the optical flow vectors and the filtered pixels during the fusion of the two techniques. The resulting fused magnitude E_T^v for a given viewport v $(\in \mathbb{V}^{\mathbb{T}})$ can be obtained as shown in Eq. (1).

$$E_T^v = \begin{cases} E_f^v, & \text{if } f = 1, N^f \\ \xi E_f^v + (1-\xi)E_o^v, & \text{otherwise} \end{cases} \tag{1}$$

where E_f^v, E_o^v, f and N^f are the total magnitudes of the Gaussian filtered pixels, optical flow vectors, the frame number and the number of frames respectively. Furthermore, a viewport set $\mathbb{V}^{\mathbb{P}}(\subset \mathbb{V}^{\mathbb{T}})$ with length N^p for the frame f is constructed using N^p viewports that exhibit greater E_T^v values in order to be used in the generation of ROI aware weightmap for the ERP frame.

D. VIEWPORT PREDICTION BIDIRECTIONAL OPTICAL FLOW ESTIMATION

Viewport Prediction Bidirectional Optical Flow Estimation (VPredBOFE) is one of the techniques used in VPredHYB that applies flow vector estimation to predict the user observed viewports. An example of bidirectional optical flow estimation for a given viewport constructed at $(\phi_o, \theta_0) = (0.0°, 0.0°)$ is shown in FIGURE 4.

Let pixel intensity of a viewport at time $t, t0$ and $t1$ $(t0 < t < t1)$ be I^t, I^{t0} and I^{t1} respectively. Then, using the bidirectional optical flow concept [16], I^t can be written in terms of I^{t0} and I^{t1},

$$I^t = I^{t0} - Gx^{t0}Vx^{t0}(t-t0) - Gy^{t0}Vy^{t0}(t-t0) \tag{2}$$
$$I^t = I^{t1} - Gx^{t1}Vx^{t1}(t-t1) - Gy^{t1}Vy^{t1}(t-t1) \tag{3}$$

where Gx^{tm}, Gy^{tm}, Vx^{tm}, and Vy^{tm} are the horizontal image gradient, vertical image gradient, and horizontal and vertical optical flow vector components of a given viewport at time tm $(m \in \{0, 1\})$ respectively.

Furthermore, because only the previous and successive viewports are used in estimation in the proposed approach, the temporal difference becomes one, thus $t-t0 = t1-t = 1$. Furthermore, assuming that the motion is along the trajectory and there would not be greater variation of flow information between two successive viewports, the optical flow vector components can be written as $Vx^{t1} = Vx^{t0} = Vx$ and $Vy^{t1} = Vy^{t0} = Vy$. Then, from Eq. (2). and Eq. (3)., the error $\Delta_{i,j}$ at the pixel location (i, j) in the viewport can be derived as,

$$\Delta_{i,j} = Gt_{i,j} + Gx_{i,j}Vx_{i,j} + Gy_{i,j}Vy_{i,j} = 0 \tag{4}$$

where $Gx_{i,j} = Gx_{i,j}^{t1} + Gx_{i,j}^{t0}$, $Gy_{i,j} = Gy_{i,j}^{t1} + Gy_{i,j}^{t0}$ and $Gt_{i,j} = I_{i,j}^{t1} - I_{i,j}^{t0}$.

In order to approximate $Vx_{i,j}$ and $Vy_{i,j}$ at pixel location (i, j), the spatial gradients $Gx_{i,j}, Gy_{i,j}$ and the temporal gradient $Gt_{i,j}$ need to be obtained. The temporal gradient is estimated by the pixel intensity difference between the viewport at $t0$ and $t1$. Moreover, by convolving horizontal and vertical Sobel filters of size 3×3 over the entire viewport in the respective direction using a local window Ω of size 9×9, the individual spatial gradients of the viewports at $t0$ and $t1$ can be obtained. These can be used to extract the combined spatial gradients $Gx_{i,j}$ and $Gy_{i,j}$.

Furthermore, in order to obtain a closer approximation of the optical flow vectors, the error $\Delta_{i,j}$ can be minimized inside the window using least square estimation such that

$$min\left(\sum_{\Omega}\Delta^{2}_{(i,j)}\right)$$

However, given that there are many viewports to be processed, the least square estimation process can be complex. Therefore, the optical flow vectors can be estimated by setting the partial derivatives of $\Delta^{2}_{(i,j)}$ with respect to $Vx_{i,j}$ and $Vy_{i,j}$ to zero. Then the optical flow vectors at pixel location (i, j) (denoted as a in Eq. (5)), $Vx_{i,j}$ and $Vy_{i,j}$ can be estimated from the spatial and temporal gradients as given in Eq. (5).

$$\begin{aligned} Vx_a &= \frac{\sum\limits_{\Omega} Gt_aGy_a \sum\limits_{\Omega} Gx_aGy_a - \sum\limits_{\Omega} Gt_aGx_a \sum\limits_{\Omega} Gy_a^2}{\delta + \sum\limits_{\Omega} Gx_a^2 \sum\limits_{\Omega} Gy_a^2 - \left(\sum\limits_{\Omega} Gx_aGy_a\right)^2} \\ Vy_a &= \frac{\sum\limits_{\Omega} Gt_aGx_a \sum\limits_{\Omega} Gx_aGy_a - \sum\limits_{\Omega} Gt_aGy_a \sum\limits_{\Omega} Gx_a^2}{\delta + \sum\limits_{\Omega} Gx_a^2 \sum\limits_{\Omega} Gy_a^2 - \left(\sum\limits_{\Omega} Gx_aGy_a\right)^2} \end{aligned} \tag{5}$$

where δ is the denominator correction factor with a value of 0.0001. The total magnitude of the optical flow vectors E_o^v for each viewport v is then determined using Eq. (6).

$$E_o^v = \sum_{j=0}^{H_{vp}} \sum_{i=0}^{W_{vp}} \sqrt{Vx_{i,j}^2 + Vy_{i,j}^2} \tag{6}$$

E. VIEWPORT PREDICTION GAUSSIAN FILTERING

Viewport Prediction Gaussian Filtering (VPredGF) is the second technique used in VPredHYB, that uses the Gaussian filtered pixels to predict the viewports. A visual example of the obtained Gaussian filtered pixels using this technique is shown in FIGURE 5.

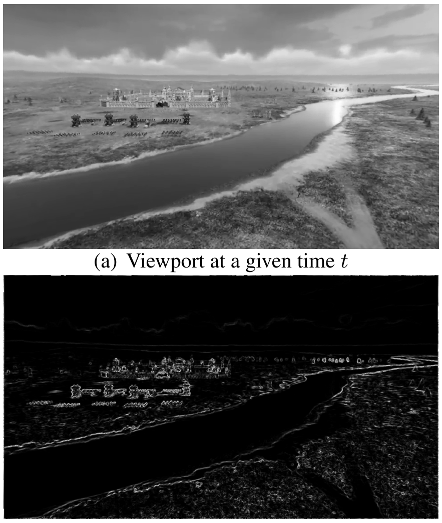
The Gaussian filter in spatial domain $h_{x,y}$ can be defined as,

$$h_{x,y} = \frac{1}{2\pi\sigma^2}\exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right) \tag{7}$$

where (x, y) are the spatial coordinates and σ is the standard deviation. In predicting the viewports, Gaussian filter $h_{x,y}$ of size $N \times N$, $(N = 7)$ and $\sigma = 1$ is used to convolve over the entire viewport image. Furthermore, a local window Ω with central pixel coordinates (i, j) and an equal size to the Gaussian filter, spanning N^2 pixels in the viewport image is used in the identification of the mean intensity. The mean intensity $\mu_{i,j}$ of a particular window Ω is computed as shown in Eq. (8).

$$\mu_{i,j} = \frac{1}{N^2}\sum_{y=\frac{(1-N)}{2}}^{\frac{(N-1)}{2}} \sum_{x=\frac{(1-N)}{2}}^{\frac{(N-1)}{2}} Ig_{i+x,j+y} \tag{8}$$

where $Ig_{i+x,j+y}$ is the Gaussian filtered pixel intensity at location (x, y) in the window Ω with central pixel coordinates (i, j). However, the mean intensity must be removed prior to the computation of the magnitude of the filtered pixels as it can be a bias to the low frequency signal components within the particular window [45]. Subsequently, the magnitude of the filtered pixels E_f^v of viewport v can be determined using Eq. (9).



(a) Viewport at a given time t

(b) Gaussian filtered pixels

FIGURE 5. Application of Gaussian filtering technique for the viewport of *DragonTale* sequence at $(\phi_0, \theta_0) = (0.0°, 0.0°)$.

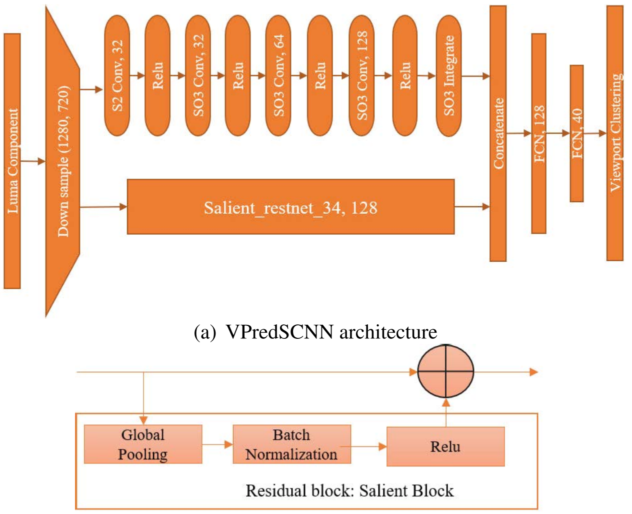
$$E_f^v = \sum_{j=0}^{H_{vp}} \sum_{i=0}^{W_{vp}} \sqrt{\sum_{y=\frac{(1-N)}{2}}^{\frac{(N-1)}{2}} \sum_{x=\frac{(1-N)}{2}}^{\frac{(N-1)}{2}} g_{x,y}\left(Ig_{i+x,j+y} - \mu_{i,j}\right)^2} \tag{9}$$

where

$$g_{x,y} = \frac{h_{x,y}}{\sum_{y=0}^{N-1}\sum_{x=0}^{N-1} h_{x,y}}$$

F. VIEWPORT PREDICTION SPHERICAL CNN

Viewport Prediction Spherical CNN (VPredSCNN) is an alternative approach to the VPredHYB, that can precisely predict a set of viewports. Although an arbitrary number of viewports can be generated from a single 360° video frame, user interest can be limited to only fewer viewports with their centres located close to one another. In such cases, the use of VPredHYB becomes a disadvantage as viewports selected would at least have a predefined distance between them which may hinder in enclosing the user interested regions. Moreover, the dual process of VPredBOFE and VPredGF in VPredHYB consumes a large amount of computational time in evaluating each viewport separately. In order to address these issues, as opposed to selecting a viewport subset $\mathbb{V}^{\mathbb{P}}$ from the viewport set $\mathbb{V}^{\mathbb{T}}$, in VPredSCNN the viewport set $\mathbb{V}^{\mathbb{P}}$ is directly predicted using trained deep learning models. In this regard, a deep learning architecture composed of Spherical CNN [17] and Salient-Resnet [18] components is employed.



(a) VPredSCNN architecture

(b) Salient block

FIGURE 6. Spherical CNN based proposed VPredSCNN network diagram.

1) SPHERICAL CNN

Spherical CNN [17] is a special type of convolutional neural network that defines rotation-equivariant spherical cross-correlation for spherical signals. As opposed to the planer CNN which applies cross-correlation on a 2D image, Spherical CNN is developed to support the signal on the sphere S^2. Here, S^2 can be defined as a set of points in 3D space $\mathbb{R}^3$ and can be parameterized by spherical coordinates $\alpha \in [-\pi, \pi]$ and $\beta \in [-\pi/2, \pi/2]$. Moreover, a set of rotations called the $SO(3)$ is parameterized using $\alpha \in [-\pi, \pi]$, $\beta \in [-\pi/2, \pi/2]$ and $\gamma \in [0, 2\pi]$ where γ performs rotation around the axis for a particular (α, β). In order to represent S^2 and $SO(3)$, Spherical CNN defines two types of cross-correlation called the $S2Conv$ and $SO3Conv$ respectively. Both $S2Conv$ and $SO3Conv$ output feature maps on $SO(3)$ space. However, $S2Conv$ receives its input as spherical signals with k channels as opposed to $SO3Conv$ which takes in feature maps with k rotations from $SO(3)$ space itself.

2) SALIENT-RESNET

Salient-Resnet [18] used in the proposed VPredSCNN is a CNN based architecture with Resnet [46] as its backbone which uses skip connection to address vanishing gradient problems. In the context of Salient-Resnet, a Salient block featuring Global pooling, batch normalization and relu unit has been used as the residual block of the Resnet.

3) PROPOSED VPredSCNN

The key idea of VPredSCNN is to extract spherical and salient features from the luma component of 360° video. In doing so, the proposed VPredSCNN is made up of two branch deep learning pipelines consisting of Spherical CNN based cross-correlation blocks, Salient-Resnet and Fully Connected Layers (FCN) as shown in FIGURE 6(a). Firstly, in the first branch, a single layer of $S2Conv$ and three layers of $SO3Conv$ have been used with equatorial grids as visual information is densely concentrated along the equator. In this regard, for $S2Conv$ α is set from $-\pi$ to π at 1024 intervals while $\beta = 0$. In the case of $SO3Conv$ α is set from $-\pi$ to π at 32 intervals, while $\beta = 0$ and γ takes values $-\pi/8$ and $\pi/8$. Moreover, the input to the network is a single channel luma component of an ERP frame. In each layer, features are increased to 32, 32, 64 and 128 while the bandwidths are reduced from 512 to 64, 32, 16 and 10. The bandwidth is initially set at 512 to maintain a higher resolution of the spatial grid as ERP frames are high resolution images. Furthermore, after the final layer, the signal is integrated over $SO3$ to obtain a tensor of 1×128 which is concatenated with the outputs from the Salient-Resnet. Here, Salient-Resnet is defined as a 34 layer network in order to reduce the computational time. Furthermore, this Salient Resnet block takes in the luma component of an ERP frame, learns the salient features and outputs 128 features. The features resulting from the Spherical CNN and Salient-Resnet are concatenated and passed through two layers of FCN before delivering the $\mathbb{V}^{\mathbb{T}}$ with 20 viewports.

The prediction of the 20 viewports can be treated as a regression problem. Both the networks are trained using a loss function to learn from the subjective data that offers Head Movement (HM) information in terms of ϕ_o and θ_o. The loss function L is defined as a combination of greater circle distance d_g and Euclidean distance d_e as sown in Eq. (10).

$$L = \lambda_g \sum_{s=1}^{n^s} \min_{\forall v} d_g \left(V_s, V_{pred,v}\right) + \lambda_e \sum_{s=1}^{n^s} \min_{\forall v} d_e \left(V_s, V_{pred,v}\right) \tag{10}$$

where s, n^s, V_s and $V_{pred,v}$ are the subject, number of subjects, ground truth viewport coordinate obtained from subject s and predicted viewport coordinate of viewport $v(\in \mathbb{V}^{\mathbb{T}})$ respectively. Furthermore, during the training of the network, a validation set is used to tune the hyper-parameters of the model. In this regard, training followed gradient descent algorithm with Adam optimizer with following hyperparameters: initial learning rate 1×10^{-3}, weight decay 1×10^{-5}, coefficient of greater circle distance $\lambda_g = 100$, coefficient of Euclidean distance $\lambda_e = 1$, batch size 1 and epochs 30. Batch size and epochs were limited by the computational complexity of the training process and availability of the resources.

Furthermore, the predicted viewports are clustered using k-means clustering to obtain a viewport set $\mathbb{V}^{\mathbb{P}}$. Here k is the same as the required number viewports N^P. Furthermore, centre coordinates of each cluster are obtained as pairs of (ϕ_o, θ_o) which are then used for QER generation on the ERP.

G. VIEWPORT FUSION

The viewports predicted using VPredHYB and VPredSCNN need to be mapped onto an ERP frame for encoding. The mapped viewports can then be fused to generate a pooled region of QERs that represent generalized ROIs of users. The generalized ROI represent the user interest on the ERP and are used to emphasize user interested regions during the encoding.

QER is a rectilinear region on the ERP with the centre known as the Quality Emphasis Centre (QEC) that represents an area corresponding to a viewport. In order to identify a QER on the ERP frame, the centre coordinates of the viewport $(W_{vp}/2, H_{vp}/2)$ and the four vertices $(0, 0)$, $(W_{vp}, 0)$, $(0, H_{vp})$ and (W_{vp}, H_{vp}) of the viewport are projected onto the ERP frame using the mapping relationship provided in [44]. Furthermore, the identified QERs cannot be directly used to construct the mask as it would cause undesirable edges in the ERP frame, disrupting the prediction schemes in VVC. Therefore, a Gaussian operation is performed to fuse all of the QERs as it does not cause a sudden decrease in pixel intensities along the spatial direction. By fusing all of the QERs, an ROI aware weightmap $w_{x,y}$ can be generated using Eq. (11).

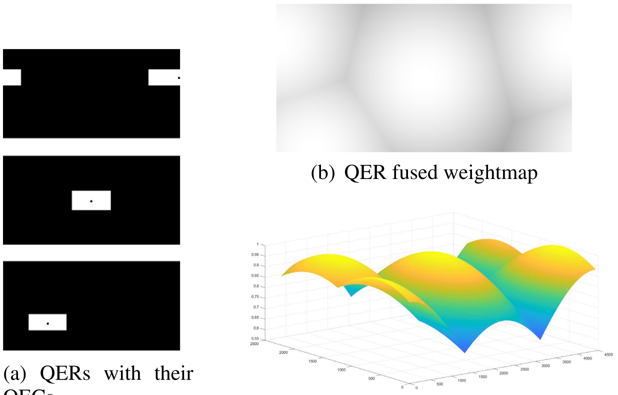
$$w_{x,y} = \max_{\forall q}\left(\exp\left\{-\frac{dx_{q,x,y}^2 + dy_{q,x,y}^2}{2\sigma_q^2}\right\}\right) \tag{11}$$

where $dx_{q,x,y}$, $dy_{q,x,y}$ and σ_q^2 are the shortest distance between the centre of QER q, QEC$_q$ and the pixel coordinates (x, y) in ERP, and variance respectively. Unlike the Gaussian filter operation where a filter is convolved over the entire viewport image, here entire ERP image is considered as a single unit where performing the Gaussian operation such that σ_q^2 is computed as shown in Eq. (12).

$$\sigma_q^2 = \frac{1}{WH}\sum_{y=0}^{H-1}\sum_{x=0}^{W-1}\left(dx_{q,x,y}^2 + dy_{q,x,y}^2\right) \tag{12}$$

Furthermore, when computing $dx_{q,x,y}$, $dy_{q,x,y}$ and σ_q^2, the continuity of a 360° image along the vertical edges of ERP frame is also considered. For example, consider a QEC$_q$ positioned near the left vertical edge of an ERP frame; then the shortest distance between that and a pixel located near the right vertical edge, would be measured through the vertical edges considering the fact that an ERP represents a spherical image and it is continuous along the normal to the longitude. Furthermore, in fusing the QERs, a *max* operation is performed in order to give priority to the nearest QER in enclosing all possible pixels in the ROI. An example of fusion of three QERs, corresponding ROI aware weightmap and a mesh diagram of the weightmap are shown in FIGURE 7(a)., FIGURE 7(b). and FIGURE 7(c). respectively.

Since VVC encoding includes spatial prediction, the error can be propagated from the quality degraded pixels to the neighbouring pixels including those enclosed in ROI. Therefore, a quality factor ρ is introduced to the weightmap in order to compensate for the quality degradation that may occur in the neighbouring pixels to the ROI during the encoding processes. The modified weightmap $\hat{w}_{x,y}$, hence can be defined as Eq. (13).



(a) QERs with their QECs

(b) QER fused weightmap

(c) Mesh plot of the weightmap

FIGURE 7. Fusion of QERs and generation of ROI aware weightmap.

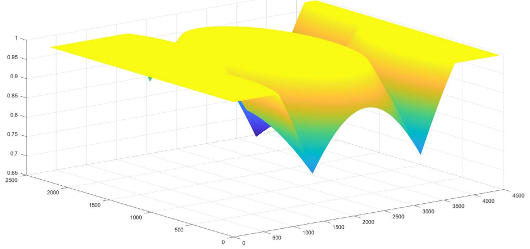


FIGURE 8. Mesh plot of the weightmap with quality factor ρ.

$$\hat{w}_{x,y} = \max_{\forall q}\left(\min\left(\rho\exp\left\{-\frac{dx_{q,x,y}^2 + dy_{q,x,y}^2}{2\sigma_q^2}\right\}, 1\right)\right) \tag{13}$$

The *min* operation used here duplicates a clipping mechanism that restricts the weights to one. Furthermore, the visualization of the weightmap after introducing the quality factor is shown in FIGURE 8.

H. LAGRANGE OPTIMIZATION AND ADAPTIVE QUANTIZATION

The proposed ERP coding framework further incorporates a Lagrange Optimization and Adaptive Quantization (LOAQ) techniques to enhance the RDO process in VVC. Denote D and R are the distortion and the bitrate required for a given coding process, then RDO in VVC is performed as,

$$minimize\ (J = D + \lambda R) \tag{14}$$

where J is the cost of the coding process and λ is the Lagrange multiplier. The optimization problem can be solved for different Coding Units (CU) selection in VVC. In this context, the proposed weightmap is introduced to the cost function in order to account for the perceptual characteristics. In this regard, the cost function for a given CU selection can be modified as,

$$minimize\ (J_{cu} = w_{cu}D_{cu} + \lambda_{base}R_{cu}) \tag{15}$$

where, $D_{cu}, R_{cu}, \lambda_{base}$ and w_{cu} are the distortion, bitrate, original Lagrange parameter and the proposed weights at CU level respectively. This can also be represented by Eq (16) [29].

$$minimize\ (J_{cu} = D_{cu} + \lambda_{cu} R_{cu}) \tag{16}$$

where λ_{cu} is the new Lagrange parameter of the CU which is given by $\lambda_{cu} = \frac{\lambda_{base}}{w_{cu}}$

Furthermore, the quantizer design in VTM is based on scalar quantization [10]. Here, quantization step Q is defined as,

$$Q^2 = 2^{(QP-12)/3} \tag{17}$$

Furthermore, since Q^2 is proportional to λ_{base} [47], λ_{base} and λ_{cu} can also be written in terms of original QP QP_{base} and the adaptive QP QP_{cu} as given in Eq. (18) and Eq. (19).

$$\lambda_{base} = c \cdot 2^{(QP_{base}-12)/3} \tag{18}$$

$$\lambda_{cu} = c \cdot 2^{(QP_{cu}-12)/3} \tag{19}$$

where c is a constant. Using equations Eq. (16), Eq. (18) and Eq. (19), the QP_{cu} can be derived as,

$$QP_{cu} = QP_{base} - 3log_2 w_{cu} \tag{20}$$

Furthermore, λ_{cu} and QP_{cu} can be deployed at CU level for the optimization of coding process. Although the weight w_{cu} adopts the weightmap derived for masking, it cannot be used in its normative form as greater pixel intensity variation may be seen inside a CU. w_{cu} is therefore defined as,

$$w_{cu} = \frac{1}{W_{cu} H_{cu}} \sum_{cu} \tilde{w}_{x,y} \tag{21}$$

where,

$$\tilde{w}_{x,y} = \begin{cases} 1, & \text{if } \forall_q \ \ (x, y) \in QER_q \\ \psi \hat{w}_{x,y}, & \text{otherwise} \end{cases}$$

and W_{cu}, H_{cu}, ψ are the width, height of CU and a constant CU weighting factor respectively. Furthermore, an offset value of 10 pixels to QERs have been used in this process in order to compensate for any projection error that occurred during the generation of the QERs.

*Signalling:*Signalling is important for the reconstruction of the encoded video sequence at the decoder. Because the proposed framework primarily involves preprocessing, there is no requirement to send any information to the decoder. However, the weights contributing to the adaptive quantization must be known at the decoder to predict the correct QP value. Since information including N^P, W_{vp}, H_{vp}, F_h and F_v can be present at the decoder, only the viewport centre (ϕ_o, θ_o) need to be signalled to the decoder for each frame encoded. In the case of VPredHYB, the index of the viewport set $\mathbb{V}^{\mathbb{T}}$ is signalled to the decoder. However, viewport coordinates predicted using VPredSCNN cannot be signalled it their normative forms as the decimal point values can increase the cost of transmission. Therefore, (ϕ_o, θ_o) are rounded to the nearest integer and QERs are constructed from the resulting values to be used in the derivations of w_{cu}.

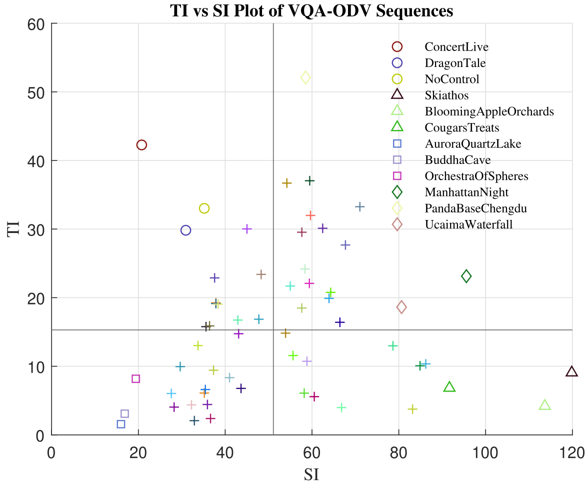


FIGURE 9. SI vs TI plot of 360° video sequences in VQA-ODV dataset. Test sequences are marked with distinctive objects and listed in the legend.

IV. EXPERIMENTS AND RESULTS

360° video sequences from the VQA-ODV dataset [48] are used during the experiments. It has 60 reference sequences with HM and Eye Movement (EM) data extracted from more than 200 subjects. Since VPredSCNN requires prior training, the dataset is split into train and test sets. Initially, the test set is determined using Spatial Information (SI) and Temporal Information (TI) such that each video sequence falls in each quadrant of the SI vs TI graph. As shown in the SI vs TI graph in FIGURE 9. three sequences from each quadrant, (high SI, low TI), (low SI, high TI), (low SI, low TI) and (high SI, high TI) are selected as the test sequences in order to represent all four quadrants in the experiments. Subsequently, the remaining 48 sequences are used for the training of VPredSCNN. Moreover, the selected test sequences vary in resolution between 4K and 8K resolutions.

A. TESTING

The proposed framework is developed using the 360 library version 10.0 [49] and incorporated in VTM 7.0. Both VPredSCNN and VPredHYB have been developed for VVC independent of each other and tested accordingly, using the same test sequences. The test sequences are coded in 4432 × 2216 resolution as per the Common Testing Conditions (CTC) recommended by JVET. Furthermore, the viewports are also constructed as per the following instruction from CTC: $W_{vp} = 1920$, $H_{vp} = 1080$, $F_h = 78.1°$ and $F_v = 49.1°$. The experiments are conducted under the All-Intra (AI) configuration since the ROI is determined only from the spatial distance between the QEC and the pixel coordinates. Moreover, based upon the several proposed viewport prediction techniques and Lagrange optimization techniques, eight different variants from the proposed framework are derived and experimented.

TABLE 2. Variants of the proposed framework.

VpredSCNN	VPredSCNN with $N^p = 8$ and $\rho = 3.0$
VPredHYB	VpredHYB with $N^p = 6$, $\rho = 3.5$ and $\xi = 0.4$
VPredBOFE	VpredHYB with $N^p = 6$, $\rho = 3.5$ and $\xi = 0.0$
VPredGF	VpredHYB with $N^p = 6$, $\rho = 3.5$ and $\xi = 1.0$
LOAQSCNN	LOAQ with $\mathbb{V}^{\mathbb{P}}$ generated using VPredSCNN, $N^p = 8$, $\rho = 3.5$ and $\psi = 0.7$ (No frame masking)
LOAQHYB	LOAQ with $\mathbb{V}^{\mathbb{P}}$ generated using VPredHYB, $N^p = 6$, $\rho = 3.5$ and $\psi = 0.7$ (No frame masking)
VpredSCNN + LOAQSCNN	VPredSCNN with $N^p = 8$ and $\rho = 3.0$ and LOAQSCNN
VpredHYB + LOAQHYB	VpredHYB with $N^p = 6$ and $\rho = 3.5$ and LOAQHYB

The derived variants and corresponding coding parameters are listed in Table 2.

B. CODING PERFORMANCE

The coding performance is evaluated using Bjontegaard Bit Rates (BDR-Y) [50] for the luma component, Encoder Time (ET) and the Decoder Time (DT) with respect to the VTM 7.0 reference software. Here, in BDR-Y calculations, Viewport Peak Signal-To-Noise Ratio (VPSNR) [35], [36], [51] is used to measure the video quality as opposed to the conventional spherical objective quality metrics which do not have the ability to assess objective video quality of the 360° videos at viewport level. VPSNR constructs viewports from the HM data of the subjects and applies PSNR calculations between the reference viewport and the tested viewport, both constructed from the same HM coordinates. In this research, the HM data presented in the VQA-ODV dataset have used in the video quality assessment. Formally, let $Ori_{(i,j)}^{(s,f,v)}$ and $Imp_{(i,j)}^{(s,f,v)}$ denote the pixel values of original and impaired sequences at (i, j) coordinates of viewport v of frame f, and s represents a subject. Then the Viewport Mean Square Error ($VMSE_f$) and VPSNR ($VPSNR$) are given by,

$$
\begin{aligned}
VMSE_f &= \frac{\sum\limits_{\forall s} \sum\limits_{\forall v \in f} \sum\limits_{\forall i,j \in v} \left(Ori_{(i,j)}^{(s,f,v)} - Imp_{(i,j)}^{(s,f,v)} \right)^2}{n^s n^v W_{vp} H_{vp}} \\
VPSNR &= 10 \log \left\{ \frac{255^2 N^f}{\sum\limits_{\forall f} VMSE_f} \right\} \qquad (22)
\end{aligned}
$$

where n^v and N^f are the number of viewports sampled per frame and number of frames respectively. Furthermore, change in quality between the anchor and the proposed algorithms (ΔVPSNR) has also been used in the assessments of the coding performance. Here, positive value forΔ VPSNR indicates quality loss with respect to the anchor while a negative value for BDR-Y indicates an overall compression gain.

The coding performances of the proposed variants are shown in Table 3. and Table 4. Table 3. illustrates the performance of the proposed methods without the LOAQ component whereas Table 4. shows results with the inclusion of the proposed LOAQ. It is evident that all our proposed variants outperform the anchor implementation of the reference software. VPredSCNN has outperformed the other variants achieving an overall bitrate savings of 4.99% (and up to 15.96% for the *CougourTreats* sequence) with no adverse effect in the decoding times. This is further improved to 5.85% by the addition of LOAQ. However, this comes at a cost of increased computational complexity both at the encoder and the decoder. Especially, the decoding times has risen up by a factor of 2.65 to perform the necessary calculation in the reconstruction of the QPs. Moreover, in the case of VPredSCNN, the sequences with higher SI produce higher bitrate savings whereas those with lower SI such as *ConcertLive*, *AuroraQuaatzLake* and *BuddhaCave* produce low to no gain. This is due to the fact that the VPredSCNN model is only trained to extract features from the spatial domain.

Furthermore, VPredBOFE and VPredGF produce consistent gains across all the sequences and achieve up to 3.97% and 3.84% (for *PandaBaseChengdu* sequence) gains respectively. When they are combined to form VPredHYB, average compression gain increase to 2.10% and up to 9.43% (for *CougourTreats* sequence). Moreover, LOAQHYB could not improve the coding performance both in isolation and in combination with VPredHYB. This is mainly because the sparsely predicted coordinates can disrupt the intra prediction process owing to the biased pixel intensities of the non-neighbouring CUs. Furthermore, it is observed that VPredBOFE, VPredGF and VPredHYB suffer heavy encoding complexity which is in excess of 200% resulting from the use of an exhaustive search on all the viewports in $\mathbb{V}^{\mathbb{T}}$.

Table 5. illustrates the performance comparison between the state-of-the-art [22], [41], [43][1] works and the VPredSCNN + LQAOSCNN. It is evident that the proposed variant outperforms the state-of-the-art methods in terms of perceptual compression efficiency, however it remains computationally complex. Aforementioned state-of-the-art researches, perceptually suffer losses resulting from the inaccurate prediction of QERs in the ERP.[2] Furthermore, other than Hu et al. [43],[3] both Sreedhar et al. [41] and TSP [22] show inconsistent gains and heavy losses (up to 10.15 dB and 12.52 dB respectively) in viewport quality due to the application of re-sampling and re-packaging of

[1] Note that state-of-the-art works by Hu et al. [43] and Sreedhar et al. [41] are self implemented with VTM 7.0 by the authors due to the unavailability of the sources.

[2] Note that the state-of-the-art research works can produce significant gains with respect to codec PSNR (PSNR measure of the coded ERP) for VVC.

[3] The original results in Hu et al. [43] is generated with respect to HEVC Test Model 16.15. However the authors recreated this work for VTM 7.0 which is nearly 40% higher in compression efficiency.

TABLE 3. Coding performance of the proposed variants without Lagrange optimization and adaptive quantization.

	VPredBOFE				VPredGF				VPredHYB				VPredSCNN			
	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT
	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)
ConcertLive	-0.27	0.03	780	095	-2.10	0.12	643	104	-2.18	0.13	990	98	0.00	0.00	463	97
DragonTale	-0.23	0.00	254	100	-1.00	0.15	214	095	-0.23	0.00	318	98	-6.62	0.66	173	97
NoControl	-1.16	0.00	203	94	-1.30	0.02	179	97	-1.24	0.03	235	97	-5.81	-0.04	173	96
Skiathos	1.22	0.17	127	102	0.45	0.29	107	100	-1.19	0.02	155	99	-3.49	0.02	87	97
BloomingAppleOrchards	-1.58	0.26	127	94	-1.04	-0.23	107	98	-9.43	2.10	149	91	-7.08	1.02	134	96
CougarsTreats	-0.18	0.00	136	103	-0.67	0.00	121	108	-2.58	0.02	167	104	-15.96	0.33	102	106
AuroraQuartzLake	-0.43	0.01	195	97	-0.47	0.00	161	104	2.83	0.12	222	98	0.04	0.00	94	100
BuddhaCave	-1.58	0.11	593	99	-2.80	0.16	507	101	-5.99	0.26	703	97	0.00	0.00	465	97
OrchestraOfSpheres	-2.62	0.05	469	97	-1.51	0.01	356	96	-1.25	0.02	491	97	-1.68	0.75	187	102
ManhattanNight	-1.11	0.00	161	97	-1.20	0.00	129	96	-3.93	0.02	170	93	-4.17	0.00	110	98
PandaBaseChengdu	-3.97	0.00	229	92	-3.84	0.47	186	92	0.00	0.00	256	98	-8.57	0.96	189	91
UcaimaWaterfall	0.00	0.00	237	108	-0.76	0.01	201	104	0.00	0.00	281	104	-6.55	1.04	213	107
Average	-0.99	0.05	293	98	-1.35	0.08	243	100	-2.10	0.23	345	98	-4.99	0.40	199	99

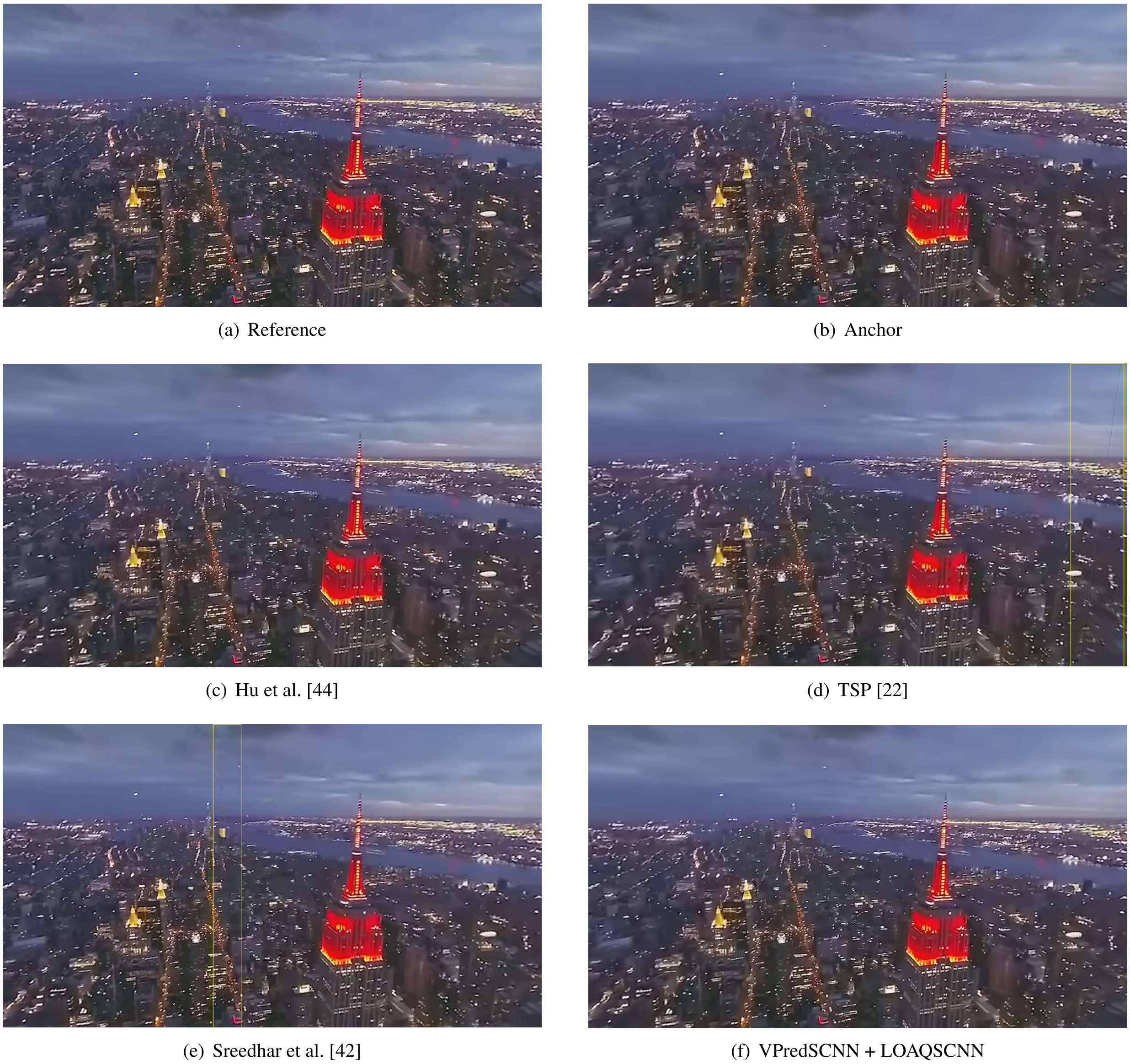
TABLE 4. Coding performance of the proposed variants with Lagrange optimization and adaptive quantization.

	LOAQHYB				VpredHYB + LOAQHYB				LOAQSCNN				VPredSCNN + LOAQSCNN			
	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT
	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)
ConcertLive	-0.47	-0.01	1031	290	-0.61	0.00	1018	325	0.54	0.08	582	318	0.54	0.08	588	307
DragonTale	0.26	0.10	274	299	0.09	0.10	353	275	-1.42	0.03	251	276	-7.87	0.69	218	270
NoControl	0.01	0.06	264	262	-2.18	0.05	261	289	-1.22	0.03	222	270	-6.89	0.00	214	261
Skiathos	0.28	0.30	150	281	0.28	0.30	144	300	-1.54	0.08	127	260	-4.93	0.10	123	249
BloomingAppleOrchards	-0.91	0.08	183	238	-10.30	2.16	166	203	-1.40	0.10	177	248	-8.34	1.11	163	218
CougarsTreats	-0.88	0.12	142	284	-0.88	0.12	153	296	-1.63	0.07	153	278	-17.15	0.40	127	266
AuroraQuartzLake	0.80	0.11	242	313	0.80	0.11	237	330	-0.08	0.06	123	283	-0.11	0.06	123	270
BuddhaCave	0.16	0.06	797	329	-2.64	0.28	741	300	-0.54	0.09	589	304	-0.54	0.09	591	295
OrchestraOfSpheres	-0.03	0.06	483	294	-1.55	0.06	402	290	-0.69	0.05	286	303	-2.21	0.79	238	276
ManhattanNight	0.48	0.20	168	273	0.48	0.20	152	271	-1.12	0.10	139	260	-5.23	0.10	136	253
PandaBaseChengdu	-1.57	0.03	306	251	-1.57	0.03	305	271	-1.42	0.07	268	256	-9.78	1.04	223	237
UcaimaWaterfall	1.21	0.28	352	315	1.21	0.28	367	352	-1.23	0.08	298	277	-7.64	1.11	250	281
Average	-0.06	0.11	366	286	-1.41	0.31	358	292	-0.98	0.07	268	278	-5.85	0.46	249	265

TABLE 5. Coding performance comparison between state-of-the-art and proposed VPredSCNN + LOAQSCNN.

	Hu et al. [43]				TSP [22]				Sreedhar et al. [41]				VPredSCNN + LOAQSCNN			
	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT	BDR-Y	ΔVPSNR	ET	DT
	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)	(%)	(dB)	(%)	(%)
ConcertLive	0.76	0.10	509	302	-3.75	4.08	347	102	30.17	0.74	194	111	0.54	0.08	588	307
DragonTale	1.21	0.18	229	262	-27.94	4.02	126	95	3.48	0.34	81	131	-7.87	0.69	218	270
NoControl	1.45	0.20	195	257	76.31	6.76	80	93	10.08	1.73	71	107	-6.89	0.00	214	261
Skiathos	0.70	0.22	110	247	N/A	12.52	33	87	N/A	10.15	32	99	-4.93	0.10	123	249
BloomingAppleOrchards	0.09	0.35	149	235	N/A	11.89	46	74	N/A	9.42	63	90	-8.34	1.11	163	218
CougarsTreats	-0.09	0.20	125	264	-32.34	5.47	27	92	-29.50	0.99	36	104	-17.15	0.40	127	266
AuroraQuartzLake	0.50	0.11	144	269	-17.53	3.13	83	102	-4.60	-0.44	043	113	-0.11	0.06	123	270
BuddhaCave	1.06	0.13	516	289	-28.54	3.84	154	98	8.13	1.95	119	103	-0.54	0.09	591	295
OrchestraOfSpheres	1.57	0.18	259	288	-18.14	3.82	175	99	15.50	0.42	86	109	-2.21	0.79	238	276
ManhattanNight	0.77	0.19	120	247	32.22	9.62	49	89	N/A	8.00	43	102	-5.23	0.10	136	253
PandaBaseChengdu	0.25	0.23	227	243	28.24	7.38	82	80	22.67	3.31	69	96	-9.78	1.04	223	237
UcaimaWaterfall	0.61	0.17	246	263	-7.92	7.34	44	95	35.15	5.09	75	107	-7.64	1.11	250	281
Average	0.74	0.19	236	264	0.06	6.66	104	92	7.59	3.48	76	106	-5.85	0.46	249	265

poorly predicted QERs in their methodologies. Moreover, the 35% compression performance achieved for CourgarsTreats sequences by TSP [22] comes at perceptual quality loss of 5.47dB. Furthermore, the HM data for the test sequences



(a) Reference

(b) Anchor

(c) Hu et al. [44]

(d) TSP [22]

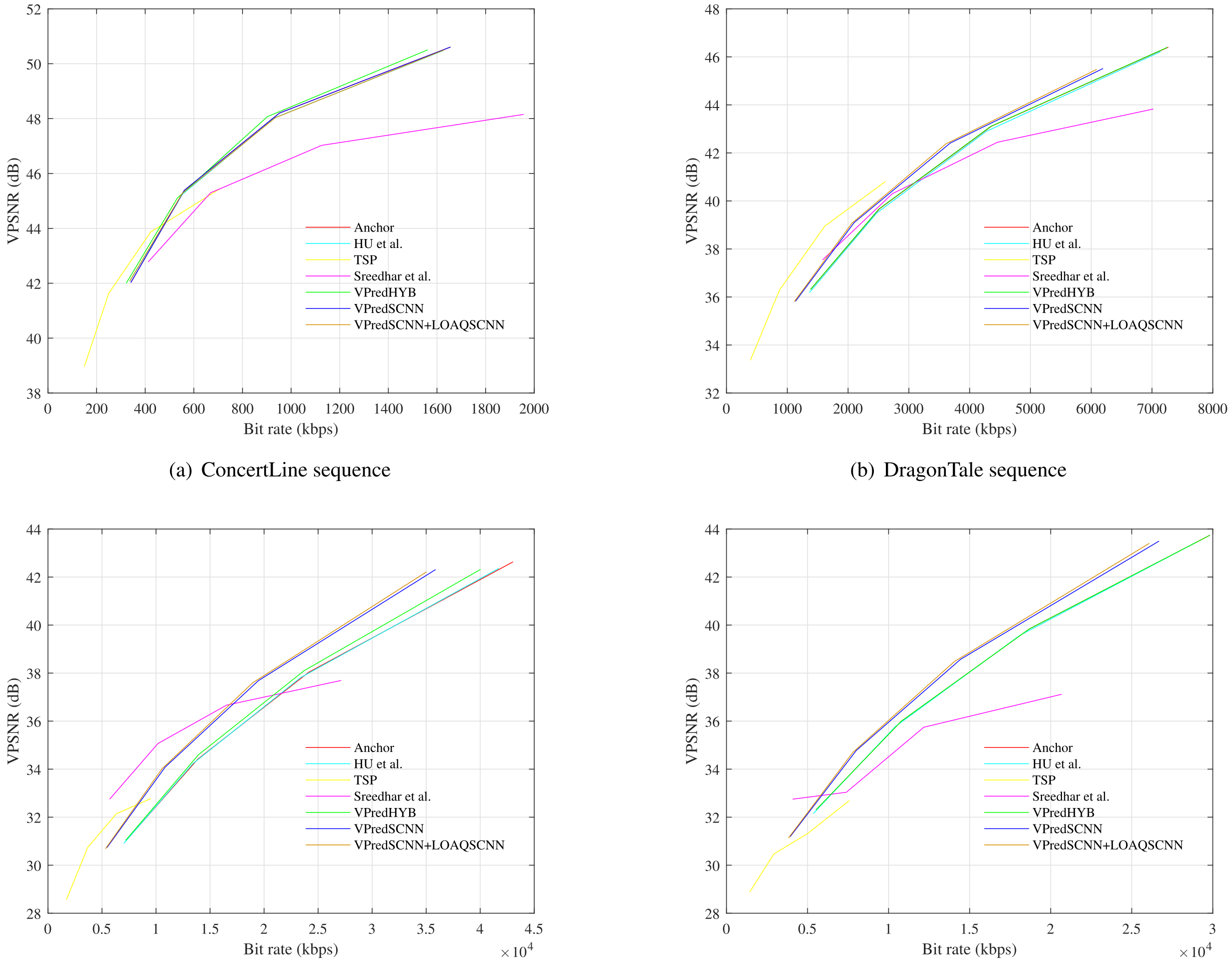
(e) Sreedhar et al. [42]

(f) VPredSCNN + LOAQSCNN

FIGURE 10. **Visual analysis of viewports generated with respect to *Manhattan* sequence: *Anchor, Hu et al. [43] and VPredSCNN + LOAQSCNN only exhibit the compression artefacts. Additionally, Sreedhar et al. [41] and TSP [22] also demonstrate seam artefacts as enclosed by yellow rectangles.***

comprise of both fine and coarse distributions of user observed viewport coordinates. The re-sampling strategies in Sreedhar et al. [41] and TSP [22] are cable of exploiting the fine distribution of viewport coordinates as opposed to the coarse distributions, which explains the variations in their respective perceptual compression performances and quality losses. Also, the idea of viewport dependent coding parameter adaptation at CTU level followed by Hu et al. [43] is not an efficient strategy for VVC due its finer QTBT+multi tree type partitioning structure. Moreover, for a better understanding, a visual comparison of a given viewport generated from the reference, anchor, state-of-the-art works and VPredSCNN + LOAQSCNN with respect to *ManHattan* sequence is shown in FIGURE 10. In the figure, it is noticeable that the anchor, Hu et al. [43] and the proposed VPredSCNN + LQAOSCNN display the common compression artefacts. As presented in Table 5., they do not exhibit greater variations in the viewport quality assessments. Moreover, the seam artefacts that appear for Sreedhar et al. [41] and TSP [22] are also illustrated in the figure.

FIGURE 11. illustrates Rate-Distortion curves for a selected sequence. It is observed that the proposed variants show improved performance compared with the anchor. An important observation from these plots are the behaviour of Sreedhar et al. [41] and TSP [22]. Both approaches tend to saturate in quality when bitrates are increased. They attain a crossover with the anchor at low bitrates, suggesting the



(a) ConcertLine sequence

(b) DragonTale sequence

(c) CougarsTreats sequence

(d) PandaBaseChengdu sequence

FIGURE 11. Comparison of Rate-Distortion Curves of viewport dependent 360° video coding techniques.

TABLE 6. Variation of bitrate saving for VPredSCNN.

$\rho \backslash N^p$	4	5	6	7	8	9	10	20
2.5	3.40	4.70	2.57	-2.95	-3.05	-2.3	-1.97	-0.89
3.0	-1.05	-0.39	-3.10	-4.85	-4.99	-4.70	-2.97	-0.33
3.5	-3.52	-2.63	-2.54	-4.03	-4.18	-2.96	-2.71	-0.10

TABLE 7. Variation of bitrate saving for VPredHYB.

(a) Variation along N^p

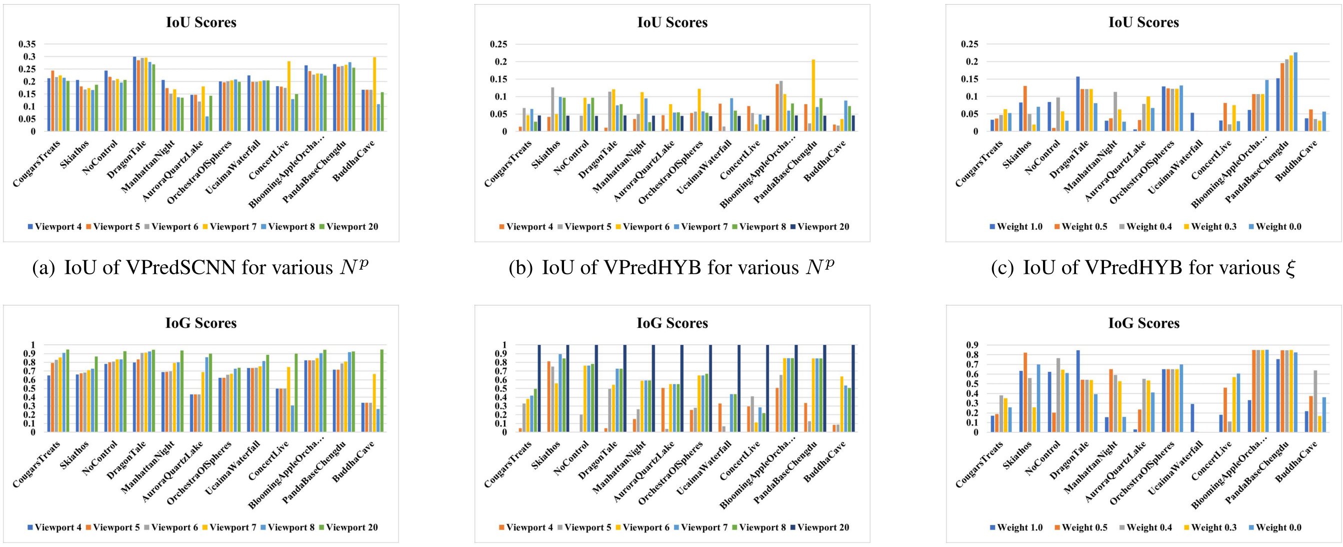
N^p	3	4	5	6	7	8
gain	0.06	-1.28	-1.11	-2.10	-0.70	-0.84

(b) Variation along ξ

ξ	1.00	0.75	0.50	0.40	0.30	0.00
gain	-1.35	-2.03	-2.06	-2.10	-1.81	-0.99

the re-sampling and the re-packaging strategies can bring benefits for low bitrate sequences. However, the proposed variants can produce gains at high bitrate ranges which can be a real benefit for 360° videos which need to be encoded at higher resolutions.

Furthermore, the variation of bitrate savings of VPredSCNN for various values of the quality factor ρ and N^p is presented in Table 6. When considering the BDR-Y values for the $\mathbb{V}^{\mathbb{P}}$ with $N^p \geq 6$, a parabolic pattern with the minimum value at ($N^p = 8, \rho = 3.0$) can be observed in both horizontal and vertical directions. This indicates that an increase in number of viewports and ρ can result in increased cost of bitrates. Conversely, a smaller number of viewports and smaller ρ values can also have a negative impact on the objective viewport quality, resulting in minor coding gains. Furthermore, Table 6. also report significant coding gains with the use of fewer viewports (i.e $6 < N^p$). However, these results are inconsistent across all the sequences as similar to the results obtained for Sreedhar et al. [41] and TSP [22]. Here, for certain sequences, the fusion of the QERs has accounted all the user observed viewports which resulted in massive gains. However, the smaller the number of viewports, the wider the predicted viewport centres would



(a) IoU of VPredSCNN for various N^p

(b) IoU of VPredHYB for various N^p

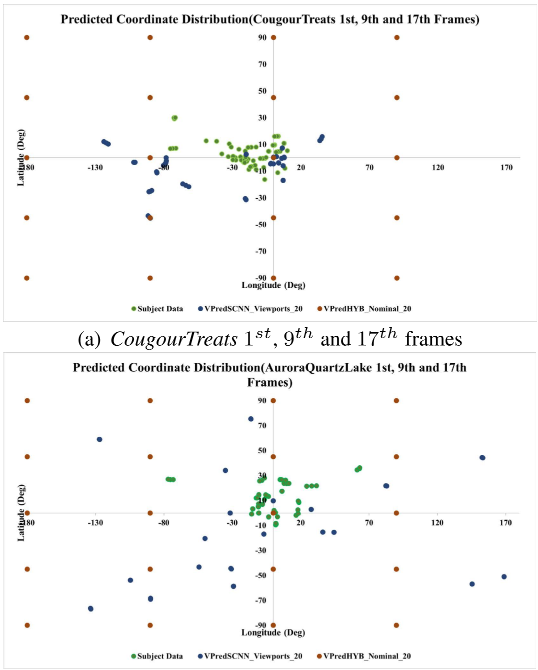
(c) IoU of VPredHYB for various ξ

(d) IoG of VPredSCNN for various N^p

(e) IoG of VPredHYB for various N^p

(f) IoG of VPredHYB for various ξ

FIGURE 12. IoU and IoG scores of VPredHYB and VPredSCNN.



(a) *CougourTreats* 1^{st}, 9^{th} and 17^{th} frames

(b) *AuroraQuartzLake* 1^{st}, 9^{th} and 17^{th} frames

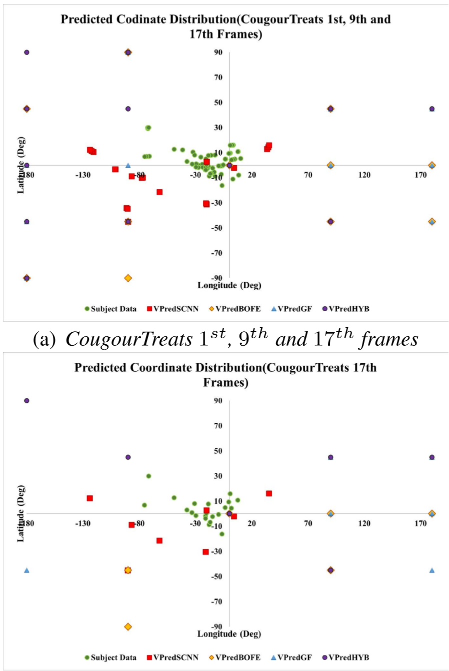
FIGURE 13. Comparison between 20 predicted coordinates from VPredHYB and VPredSCNN and the subject data for *AuroraQuartzLake* and *CougourTreats* sequences.

be. This would hinder the intra prediction as some regions in between two QERs can be adversely affected from the proposed masking process.

Table 7(a). and Table 7(b). present the variation of bitrate saving for VPredHYB for various values of the number of viewports N^p and the quality factor ξ at $\rho = 3.5$ respectively. Unlike in the case of VPredSCNN, the increase of viewports does not result in an increased gain. Since the initial coordinates are predefined at constant intervals, it is likely that additional redundant information is being coded when using a greater number of viewports. Conversely, visually degraded results are obtained when a lower number of viewports are deployed. Furthermore, the impact of ξ can be great when both VPredGF and VPredBOFE are combined, but the gain is likely to be saturated between $\xi = 0.4$ and $\xi = 0.75$.

C. VIEWPORT PREDICTION ACCURACY

The accuracy of viewport prediction techniques can be measured using metrics such as Intersection over Union (IoU) and Intersection over Ground truth (IoG). IoU measures the ratio between the intersection area and union area. Here, the intersection and the union areas are measured with respect to the QERs generated from the predicted viewport coordinates and the ground truth data. In this context, for a given set of predicted viewports $\mathbb{V}^{\mathbb{P}}$, IoU is measured by constructing N^p QERs from the predicted coordinates and n^s QERs from the ground truth data. Hence, the number of elements in $\mathbb{V}^{\mathbb{P}}$ is increased to obtain a better intersection. In doing so, IoU score of a given sequence is affected by the additional number of viewports used as the distribution of the viewport centres may vary from one sequence to another. Hence, for fixed number of viewports, IoU would not be able to accurately estimate the solitary measure of intersection across different sequences which accounts for the information loss in video coding approaches. As opposed to IoU, IoG measures the ratio between the intersection area and the QERs generated from the ground truth data, negating the effect of the union area. This provides a better estimate of whether user observed viewports are actually covered by the predicted



(a) *CougourTreats* 1^{st}, 9^{th} *and* 17^{th} *frames*

(b) *CougourTreats* 17^{th} frame

FIGURE 14. Distribution of predicted coordinates of proposed variants for *CougourTreats* sequence.

viewports when the number of viewport is fixed for all sequences.

The scores of the viewport prediction accuracy measures, IoU and IoG for VPredHYb and VPredSCNN are illustrated in FIGURE 12. In this figure, the variations of IoU and IoG with respect to the number of viewports, N^p are also shown in FIGURE 12(a)., FIGURE 12(b)., FIGURE 12(d). and FIGURE 12(e)., for both VPredSCNN and VPredHYB respectively. Furthermore, FIGURE 12(c). and FIGURE 12(f). show the variation of these measures for various values of ξ in VPredHYB for $N^p = 6$. Since prediction techniques produce fewer number of viewports than the actual subjects, it is apparent that IoU scores in all scenarios exhibit lower values. However, in the context of video coding, it is sufficient if the predicted QERs can account for the subject's viewports and avoid disruption to the intra prediction of the video codec. In support of this norm, the IoG scores exhibit higher values for both VPredSCNN and VPredHYB as the number of viewports increases. This is because an increased number of viewports can improve the overlapping of generated QERs with the subject's HM coordinates. Moreover, for VPredHYB, there is no greater difference in the IoU and IoG scores obtained for the several variations in ξ.

The distribution between the subject's HM coordinates (Subject data), the 20 viewports predicted using VPredSCNN and the predefined 20 viewports used in VPredHYB are illustrated in FIGURE 13 for 1^{st}, 9^{th} and 17^{th} frames of *CougourTreats* and *AuroraQuartzLake* sequences. The two sequences are chosen as one produces very high coding gains for the proposed variants, while the other does not. It is evident that distribution of the viewport coordinates are much closer to ground-truth HM data for *CougourTreats* sequence compared to *AuroraQuartzLake* sequence. Hence, it can be concluded that the subset of viewports selected from the 20 viewports in both VPredSCNN and VPredHYB have demonstrated better performance for *CougourTreats* sequence compared to *AuroraQuartzLake* sequence. Furthermore, the distribution of the predicted viewport coordinates from the proposed variants without LOAQ are shown in FIGURE 14. In this figure, it can be observed that the coordinates predicted using VPredSCNN ($N^p = 8$) are found to be closer to the subject data than the other proposed variants which substantiate the bitrate improvements with VPredSCNN in comparison with the other variants.

V. CONCLUSION

Existing perceptual video coding algorithms cannot be applied to 360° videos which are not represented in their visually observed format when encoding. In response, a novel 360° video coding framework has been developed to leverage the user observed viewport information in the VVC coding pipeline in order to reduce the bitrates at the same perceptual quality. To this end, the proposed framework first applies a deep learning architecture incorporating Spherical CNN components and a fusion between bi-directional optical flow estimation and a Gaussian filtering technique in order to develop two viewport prediction techniques namely, VPredSCNN and VPredHYB respectively. Furthermore, based on the predicted viewports, the proposed framework also generates QERs, to identify the ROI on the ERP. Subsequently, by fusing QERs, an ROI aware weightmap is developed and applied as a mask to the source video. Furthermore, the proposed framework also employs the weightmap to support the Lagrange optimization and adaptive quantization procedures in VVC.

In the context of 360 Lib 10.0 integrated VTM 7.0, the experiments conducted for different variants of the proposed framework outperform the state-of-the-art techniques and report significant coding gains. VpreddSCNN when combined with LOAQ yield the highest compression gains with average bitrate savings of 5.85% (and up to 17.15%) with an increase of 249% and 265% encoder and decoder complexities respectively. Additionally, it has been reported that the removal of LOAQ from the coding framework can nullify the decoder complexity and reduce the encoding time with a slight drop in coding gain. Moreover, temporal domain support to VPredSCNN, integration of the proposed VPredSCNN and VPredHyb techniques, several other viewport prediction techniques and inter coding compatibility for the proposed 360° video framework can be explored in the future.

REFERENCES

[1] P. Rosedale, "Virtual reality: The next disruptor: A new kind of worldwide communication," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 48–50, Jan. 2017.

[2] D. You, B.-S. Seo, E. Jeong, and D. H. Kim, ''Internet of Things (IoT) for seamless virtual reality space: Challenges and perspectives,'' *IEEE Access*, vol. 6, pp. 40439–40449, 2018.

[3] X. Yang, Z. Chen, K. Li, Y. Sun, N. Liu, W. Xie, and Y. Zhao, ''Communication-constrained mobile edge computing systems for wireless virtual reality: Scheduling and tradeoff,'' *IEEE Access*, vol. 6, pp. 16665–16677, 2018.

[4] Y. Zhou, L. Tian, C. Zhu, X. Jin, and Y. Sun, ''Video coding optimization for virtual reality 360° source,'' *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 1, pp. 118–129, Jan. 2020.

[5] B. Bross, J. Chen, J. R. Ohm, G. J. Sullivan, and Y.-K. Wang, ''Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC),'' *Proc. IEEE*, vol. 109, no. 9, pp. 1463–1493, Sep. 2021.

[6] Y.-W. Huang, J. An, H. Huang, X. Li, S.-T. Hsiang, K. Zhang, H. Gao, J. Ma, and O. Chubach, ''Block partitioning structure in the VVC standard,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3818–3833, Oct. 2021.

[7] J. Pfaff, A. Filippov, S. Liu, X. Zhao, J. Chen, S. De-Luxán-Hernández, T. Wiegand, V. Rufitskiy, A. Krishnan Ramasubramonian, and G. Van der Auwera, ''Intra prediction and mode coding in VVC,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3834–3847, Oct. 2021.

[8] H. Gao, S. Esenlik, E. Alshina, and E. Steinbach, ''Geometric partitioning mode in versatile video coding: Algorithm review and analysis,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3603–3617, Sep. 2021.

[9] X. Zhao, S.-H. Kim, Y. Zhao, H. E. Egilmez, M. Koo, S. Liu, J. Lainema, and M. Karczewicz, ''Transform coding in the VVC standard,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3878–3890, Oct. 2021.

[10] H. Schwarz, M. Coban, M. Karczewicz, T.-D. Chuang, F. Bossen, A. Alshin, J. Lainema, C. R. Helmrich, and T. Wiegand, ''Quantization and entropy coding in the versatile video coding (VVC) standard,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3891–3906, Oct. 2021.

[11] M. Karczewicz, N. Hu, J. Taquet, C.-Y. Chen, K. Misra, K. Andersson, P. Yin, T. Lu, E. Francois, and J. Chen, ''VVC in-loop filters,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3907–3925, Oct. 2021.

[12] Y. Ye, J. M. Boyce, and P. Hanhart, ''Omnidirectional 360° video coding technology in responses to the joint call for proposals on video compression with capability beyond HEVC,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1241–1252, May 2020.

[13] Z. Chen, W. Lin, and K. N. Ngan, ''Perceptual video coding: Challenges and approaches,'' in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2010, pp. 784–789.

[14] M. Wang, T. Zhang, C. Liu, and S. Goto, ''Region-of-interest based dynamical parameter allocation for H.264/AVC encoder,'' in *Proc. Picture Coding Symp.*, May 2009, pp. 1–4.

[15] H. Meuel, M. Munderloh, M. Reso, and J. Ostermann, ''Optical flow cluster filtering for ROI coding,'' in *Proc. Picture Coding Symp. (PCS)*, Dec. 2013, pp. 129–132.

[16] Y.-W. Chen, H.-C. Chuang, X. Li, L. Zhang, W.-J. Chien, J. Chen, and M. Karczewicz, ''Motion vector reconstructions for Bi-directional optical flow (bio),'' U.S. Patent 15 861 515, Jul. 5, 2018.

[17] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, ''Spherical CNNs,'' in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.

[18] D. Shen, S. Zhao, J. Hu, H. Feng, D. Cai, and X. He, ''ES-Net: Erasing salient parts to learn more in re-identification,'' *IEEE Trans. Image Process.*, vol. 30, pp. 1676–1686, 2021.

[19] C. W. Fu, L. Wan, T. T. Wong, and C. S. Leung, ''The rhombic dodecahedron map: An efficient scheme for encoding panoramic video,'' *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 634–644, Jun. 2009.

[20] M. Zhou, *Ahg8: A Study on Compression Efficiency of Cube Projection*, document JVET-D0022, Chengdu, China, 2016.

[21] H. Lin, C. Li, J. Lin, S. Chang, and C. Ju, *An Efficient Compact Layout for Octahedron Format, JVET Doc*, document D0142, 2016.

[22] G. Van der Auwera, M. Coban, M. K. Hendry, and M. Karczewicz, *Ahg8: Truncated Square Pyramid Projection (TSP) for 360° Video*, document JVET-D0071, Joint Video Exploration Team ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 4th Meeting, 2016.

[23] S. Akula, A. Singh, R. Kk, R. Gadde, V. Zakharchenko, E. Alshina, and K. Choi, *Ahg8: Efficient Frame Packing Method for Icosahedral Projection (ISP)*, document JVET-G0156, 2017.

[24] A. Abbas and D. Newman, *Ahg8: Rotated Sphere Projection for 360° Video*, document JVET-F0036, Hobart, RI, Australia, vol. 31, 2017.

[25] S. Jaballah, A. Bhavsar, and M.-C. Larabi, ''Perceptual versus latitude-based 360° video coding optimization,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3423–3427.

[26] F. Racapé, F. Galpin, G. Rath, and E. Francois, *Ahg8: Adaptive QP for 360° Video Coding*, document JVET-F0038, 2017.

[27] X. Xiu, Y. He, and Y. Ye, ''An adaptive quantization method for 360° video coding,'' in *Proc. SPIE*, vol. 10752, Sep. 2018, Art. no. 107520.

[28] J. Adhuran, G. Kulupana, C. Galkandage, and A. Fernando, ''Multiple quantization parameter optimization in versatile video coding for 360° videos,'' *IEEE Trans. Consum. Electron.*, vol. 66, no. 3, pp. 213–222, Aug. 2020.

[29] Y. Li, J. Xu, and Z. Chen, ''Spherical domain rate-distortion optimization for omnidirectional video coding,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1767–1780, Jun. 2019.

[30] J. Carreira, S. M. M. de Faria, L. M. N. Tavora, A. Navarro, and P. A. Assuncao, ''Versatile video coding of 360° video using adaptive resolution change,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3398–3402.

[31] M. Budagavi, J. Furton, G. Jin, A. Saxena, J. Wilkinson, and A. Dickerson, ''360° video coding using region adaptive smoothing,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 750–754.

[32] Y. Wang, L. Li, D. Liu, F. Wu, and W. Gao, ''A new motion model for panoramic video coding,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 1407–1411.

[33] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, ''Adaptive 360° video streaming using scalable video coding,'' in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 1689–1697.

[34] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl, ''Tile based HEVC video for head mounted displays,'' in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 399–400.

[35] J. Chakareski, ''Viewport-adaptive scalable multi-user virtual reality mobile-edge streaming,'' *IEEE Trans. Image Process.*, vol. 29, pp. 6330–6342, 2020.

[36] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, ''Viewport-driven rate-distortion optimized 360° video streaming,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.

[37] Y. S. D. La Fuente, G. S. Bhullar, R. Skupin, C. Hellge, and T. Schierl, ''Delay impact on MPEG OMAF's tile-based viewport-dependent 360° video streaming,'' *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 18–28, Mar. 2019.

[38] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, ''Adaptive 360° video streaming using layered video coding,'' in *Proc. IEEE Virtual Reality (VR)*, Mar. 2017, pp. 347–348.

[39] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, ''Viewport-adaptive navigable 360° video delivery,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[40] D. Naik, I. D. D. Curcio, and H. Toukomaa, ''Optimized viewport dependent streaming of stereoscopic omnidirectional video,'' in *Proc. 23rd Packet Video Workshop*, Jun. 2018, pp. 37–42.

[41] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, ''Viewport-adaptive encoding and streaming of 360° video for virtual reality applications,'' in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 583–586.

[42] E. V. Kuzyakov, R. Peng, and C.-N. Chen, ''Systems and methods for provisioning content using barrel projection representation,'' U.S. Patent 10 579 898, Mar. 3, 2020.

[43] Q. Hu, J. Zhou, X. Zhang, Z. Shi, and Z. Gao, ''Viewport-adaptive 360° video coding,'' *Multimedia Tools Appl.*, vol. 79, no. 17, pp. 12205–12226, 2020.

[44] Y. Ye, E. Alshina, and J. Boyce, *Algorithm Descriptions of Projection Format Conversion and Video Quality Metrics in 360Lib (Version 5)*, document JVET-H1004, Joint Video Exploration Team ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 2017.

[45] K. Seshadrinathan and A. C. Bovik, ''Motion tuned spatio-temporal quality assessment of natural videos,'' *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 335–350, Feb. 2009.

[46] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[47] F. Zhang and D. R. Bull, ''Rate-distortion optimization using adaptive Lagrange multipliers,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 10, pp. 3121–3131, Oct. 2019.

[48] C. Li, M. Xu, X. Du, and Z. Wang, "Bridge the gap between VQA and human behavior on omnidirectional video: A large-scale dataset and a deep learning model," 2018, *arXiv:1807.10990*.

[49] Y. He, K. Choi, J.-L. Lin, Y. Sun, M. Coban, Y. Lu, A. Abbas, M. Zhou, Z. Deng, and H.-M. Ohj, *JVET 360Lib Software Manual (Version 5)*, document IEC JTC1/SC29/WG11, 2020.

[50] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33, 2001.

[51] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, Sep. 2015, pp. 31–36.

JAYASINGAM ADHURAN (Member, IEEE) received the B.S.Eng. degree (Hons.) in electronics engineering and the M.Eng. degree in microelectronics and embedded systems from the Asian Institute of Technology (AIT), Thailand, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey. He was a Lecturer (Probationary) at the University of Jaffna, from July 2018 to September 2018. He is also an Researcher and a Development Engineer with BBC, U.K. His current research interest includes video coding.

GOSALA KULUPANA received the B.Sc.Eng. degree (Hons.) from the University of Moratuwa, Sri Lanka, in 2011, and the Ph.D. degree in HEVC video coding from CVSSP, University of Surrey, U.K., in 2017. From 2011 to 2014, he was an Engineer at Mobitel (Pvt.) Ltd., Sri Lanka. He was a Research Fellow at CVSSP, before joining the BBC Research and Development Team, U.K., as a Research Engineer, in 2018. His research interests include intra prediction, 360° video coding, VVC, HEVC, and resource optimization.

ANIL FERNANDO (Senior Member, IEEE) received the B.Sc. degree (Hons.) in electronics and telecommunications engineering from the University of Moratuwa, Sri Lanka, in 1995, the M.Sc. degree (Hons.) in telecommunications from the Asian Institute of Technology (AIT), Thailand, in 1997, and the Ph.D. degree in video coding from the Department of Electrical and Electronic Engineering, University of Bristol, U.K., in 2001. He was a Reader at the University of Surrey, U.K., a Senior Lecturer at Brunel University, U.K, and an Assistant Professor at AIT. He is currently a Professor of video coding and communications with the University of Strathclyde, U.K., and a Visiting Professor with the University of Surrey, U.K. He has authored over 350 international publications in video coding, machine learning, and communications and signal processing. His research interests include video coding and communications, machine learning, AI solutions for industrial applications, quality of experience modeling, autonomous systems, and 5G/6G application developments. He is a fellow of the Higher Education Academy, U.K., and a member of EPSRC College, U.K.