Jaakko Laitinen; Tero Partanen; Alexandre Mercat; Jarno Vanne; Miska Hannuksela; Honglei Zhang, Alireza Aminlou, Francesco Cricri, "Feasibility Study of Multi-Layer VVC Coding Scheme for Hybrid Machine-Human Consumption," in Proc of IEEE International Conference on Multimedia and Expo (ICME), Niagara Falls Marriott, Niagara Falls, Canada, July, 2024.

DOI: 10.1109/ICME57554.2024.10687938

https://ieeexplore.ieee.org/document/10687938

FEASIBILITY STUDY OF MULTI-LAYER VVC CODING SCHEME FOR HYBRID MACHINE-HUMAN CONSUMPTION

Jaakko Laitinen[1], Tero Partanen[1], Alexandre Mercat[1], Jarno Vanne[1], Miska Hannuksela[2], Honglei Zhang[2], Alireza Aminlou[2], and Francesco Cricri[2]

[1]Ultra Video Group, Tampere University, Finland
[2]Nokia Technologies, Finland
{jaakko.laitinen, tero.partanen, alexandre.mercat, jarno.vanne}@tuni.fi,
{miska.hannuksela, honglei.1.zhang, alireza.aminlou, francesco.cricri}@nokia.com

ABSTRACT

The proliferation of machine vision applications necessitates developing more efficient visual data compression schemes for machine consumption. However, numerous automated use cases still require keeping humans in the loop, leading to the need for a machine-optimized video streaming with the option for human supervision. This paper investigates the feasibility of using the multi-layer coding approach of the emerging Versatile Video Coding (VVC) standard to create favorable conditions for hybrid machine-human consumption. We introduce a multi-layer coding scheme, where the base layer (BL) is optimized for machines and the enhancement layer (EL) complements the stream for human vision. Our results demonstrate that the bitrate of the proposed multi-layer stream (BL + EL) is, on average, 11% higher than that of a single-layer VVC. However, the more compact BL yields overall bandwidth savings as long as the EL is required less than 80% of the time.

Index Terms—Region-of-interest (ROI), Versatile Video Coding (VVC), Multi-layer video coding, Video Coding for Machines (VCM), hybrid machine-human video consumption

1. INTRODUCTION

Recent years have witnessed a significant increase in automated visual data analysis in applications such as autonomous driving, intelligent transportation, smart manufacturing, and surveillance. This trend has catalyzed a new MPEG/JVET standardization activity called *Video Coding for Machines* (*VCM*) to develop video coding techniques for machine consumption. VCM seeks to utilize the synergy between compression and analytics to strike a balance between the needs of machinery and human beings [1].

Versatile video coding (*VVC*) [2] is the latest MPEG video coding standard, initially developed to compress video from the perspective of the human visual system. Recently, JVET has explored various machine-oriented VVC optimization techniques as well as pre- and post-processing methods to improve VVC coding efficiency without compromising the accuracy of machine analysis [3]. Although some of these approaches result in suboptimal visual quality for human viewing, they cater to the needs of fully automated, machine-only use cases. However, there exist also numerous human-in-the-loop applications, where preserving the original, human-viewable video is a key enabling factor.

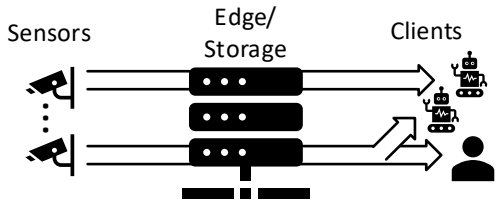


Fig. 1. General use case for human-machine video consumption.

Fig. 1 illustrates the general use case for hybrid human-machine video consumption considered in this study. A set of camera sensors capture video streams that are either processed and stored at the edge layer or transmitted to the cloud layer for further processing and storage. Typically, these streams serve two purposes: 1) continuous monitoring and analysis tasks that are automated by various machine vision algorithms; and 2) on-demand supervision by human operators in some specific situations or adverse operating conditions, either in real time or offline. Exemplary use cases encompass intelligent traffic, assembly line, and security monitoring systems.

In this paper, we investigate the viability of VVC multi-layer, or scalable coding, scheme [4] to facilitate hybrid machine-human consumption in usage scenarios generalized in Fig. 1. The VVC multi-layer scheme is comprised of a *base layer* (*BL*) and an *enhancement layer* (*EL*), where coding gains over simulcast coding are obtained by copying data from the BL to the EL through inter-layer references. In the proposed approach, the BL contains a pre-processed, i.e., machine-optimized video stream for continuous machine monitoring, whereas the EL carries an unaltered human-consumable video for on-demand monitoring. Our study also evaluates two different *region-of-interest* (*ROI*) based pre-processing methods for the BL video: background blurring and replacing the background with a constant grey color, also referred to as greying. Furthermore, additional coding gains are sought by scaling the BL video with different spatial scaling ratios.

The remainder of the paper is organized as follows. Section 2 provides a literature overview of hybrid human-machine coding and ROI-based processing methods. Section 3 presents our proposal in more detail, along with other potential coding schemes. Our experimental setup is described in Section 4 and the obtained results are reported in Section 5. Finally, Section 6 concludes the paper.

2. RELATED WORK

This section provides an overview of relevant prior work in two key areas: multi-layer video coding schemes for facilitating hybrid human-machine consumption and ROI-based pre-processing methods for video coding.

2.1. Multi-layer coding for machine and human consumption

Wang et al. [5] and Yang et al. [6] proposed extracting facial features from images for machine analysis. The additional data provided in the EL was utilized to reconstruct the original image for human viewing. Yan et al. [7] extended this concept to a broader range of images. Lin et al. [8] and Choi et al. [9] proposed an end-to-end learned *neural network* (*NN*) based codec with multiple layers for video. In their approach, the BL was comprised of extracted features from the input to facilitate simple machine tasks, whereas the EL provided motion and texture information to reconstruct full video for human consumption.

Harell et al. [10] proposed a NN-based codec that is solely trained for the BL. A separate NN model is used to reconstruct an image from this BL, integrating it into the *decoded picture buffer* (*DPB*) of a conventional VVC encoder. The integration enabled efficient compression of the EL using inter coding tools of VVC.

Seppälä et al. [11] adopted an opposite approach, wherein the standard VVC BL was used for human viewing and the EL to improve machine task performance. This was achieved by extracting features from both the original and decoded VVC images and subsequently coding the feature residuals using an end-to-end learned NN based codec.

None of these existing approaches provided standardized coding tools or generalized approaches, which can pose challenges in practical deployment.

2.2. Region-of-interest (ROI) coding

ROI-based video compression is based on the grounds that pixel saliency is non-uniform across a video frame. Essentially, certain pixels or regions within a frame are less important or even irrelevant to a video consumer. In lossy compression, this disparity allows for higher coding efficiency in the non-ROI areas. Typically, ROI detection employs an object-based approach that yields a binary ROI mask, which is found useful by human viewers and particularly the machine tasks including object detection and tracking.

There are two primary methods to leverage ROI information in video compression: 1) frame pre-processing; and 2) embedded encoder guidance. In frame pre-processing, spatially variable blur or other image processing techniques are applied to non-ROI areas to reduce the bits required for compression. The blurring focuses on non-ROI (background) regions of the input frames before compression, serving as a low-pass filter that reduces high-frequency components and the bitrate for the coded non-ROI area. Frame pre-processing is straightforward and compatible with different video encoders, whereas encoder guidance has the potential to offer more precise control, but it requires encoder-specific implementations. With encoder guidance, ROI information is used to guide an encoder to allocate more bits to the important regions and fewer bits to the background.

For the time being, several notable approaches have been presented for ROI-based video compression. For example,

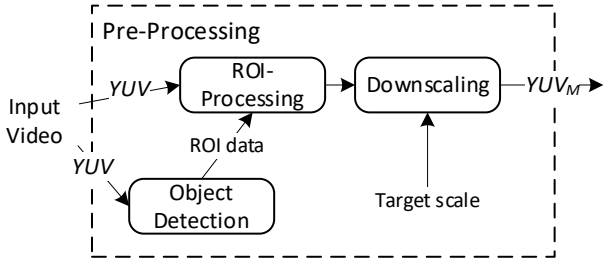


Fig. 2. Proposed pre-processing step.

Ogasawara et al. [12] proposed an object-based video coding scheme for human consumption, wherein the background regions were blurred using Gaussian filters. Similarly, Grois and Hadar [13] introduced an adaptive and complexity-aware pre-processing scheme for ROI-based *scalable video coding* (*SVC*). They applied the ROI-based pre-processing adaptively by adjusting the pre-filter parameters for each SVC layer. Additionally, Bagdanow et al. [14] proposed a ROI-based background smoothing scheme for hybrid human-machine consumption in video surveillance applications. Furthermore, JVET has investigated various ROI-based pre-processing approaches [3]. They proposed either masking the non-ROI regions using a uniform color, thus, completely removing the background, or applying a Gaussian blurring filter to the background.

3. VIDEO CODING SCHEMES FOR HYBRID MACHINE-HUMAN CONSUMPTION

For this contribution, we have explored the feasibility of different video coding schemes to implement the general use case of interest (Fig. 1). Some of these schemes also incorporate a separate pre-processing stage for machine-optimized video. For consistency, the same pre-processing method is employed across all cases.

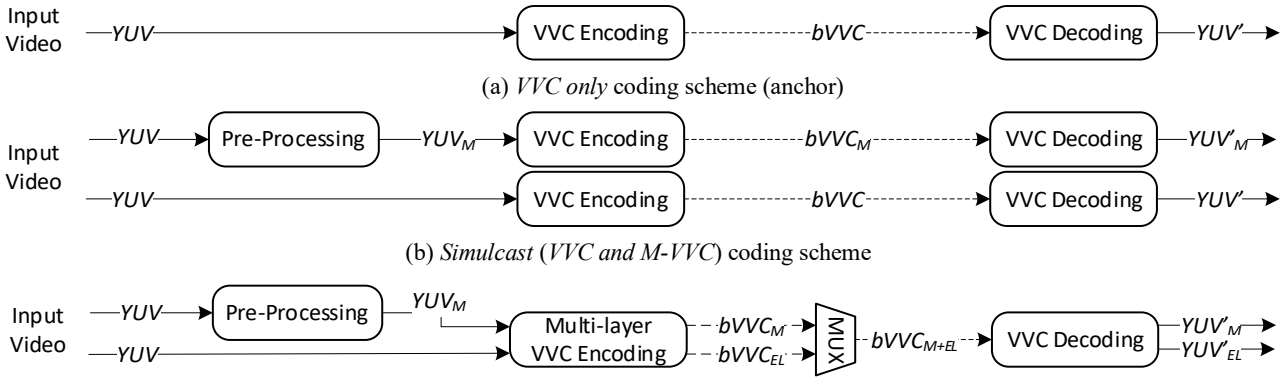
3.1. Machine-optimized pre-processing methods

In this work, we employ a ROI-based pre-processing method to generate a machine-optimized video. As outlined in Fig. 2, our proposal consists of three steps: 1) ROI detection; 2) ROI-guided processing; and 3) downscaling.

An object detector is utilized to generate a ROI map within the input video. The regions, identified by the detector, then guide the ROI-processing step, where the input video is modified by selectively processing the background areas.

The ROI-processing step aims to enhance compressibility at the expense of quality in non-ROI regions. We investigate two background modification methods: 1) background blurring that heavily blurs the background, rendering it more easily compressible while maintaining some general details; and 2) background greying, which completely removes the background and replaces it with grey. The background greying approach provides better coding efficiency but results in the total loss of background details. On the other hand, blurring might integrate more seamlessly with the multi-layer coding tools. Nevertheless, since machine tasks generally prioritize salient regions over the background, altering the background usually does not significantly degrade the performance of these tasks.

Both methods can also be executed with different downscaling ratios for additional bitrate savings and to facilitate spatial scalability with multi-layer VVC.



(a) *VVC only* coding scheme (anchor)

(b) *Simulcast* (*VVC and M-VVC*) coding scheme

(c) Proposed *multi-layer VVC* coding scheme

Fig. 3. Studied coding flows.

Table 1. Characterization of coding schemes

Coding scheme	Machine compatible	Human compatible	Partial rewind	Full rewind
VVC only	Yes	Yes	Yes	Yes
M-VVC only	Yes	No	No	No
VVC or M-VVC	Yes	Yes	Yes	No
VVC and M-VVC	Yes	Yes	Yes	Yes
Multi-layer VVC	Yes	Yes	Yes	Yes

3.2. Exploration of video coding schemes

Our exploration includes the following five video coding schemes:

- **VVC only:** A standard VVC bitstream is continuously sent for machine and human operators.
- **M-VVC only:** A machine-optimized VVC bitstream is continuously sent for machine operators without any additional streams for human operators.
- **VVC or M-VVC:** A machine-optimized VVC bitstream is sent for machine operators by default but it can be replaced by a standard VVC bitstream upon request by a human operator, i.e., no simulcasting.
- **VVC and M-VVC:** A standard VVC and a machine-optimized VVC bitstream are continuously simulcasted.
- **Proposed multi-layer VVC:** VVC and a machine-optimized VVC bitstreams are continuously sent using multi-layer coding tools of VVC.

In Table 1, these five schemes are characterized in terms of their compatibility for human and machine consumption and support for rewind. A human operator may need to rewind to see events of interest afterwards. Partial rewinding limits the accessible events to the times when a VVC bitstream was transmitted to the operators. Full rewinding means that a human-viewable bitstream is always available.

Only three of these schemes, namely the VVC only, VVC and M-VVC, and multi-layer VVC coding schemes, fulfill all the necessary requirements of the addressed use case (Fig. 1). Henceforth, we will concentrate solely on these schemes. Fig. 3 illustrates their flow diagrams.

3.3. *VVC only* coding scheme

Fig. 3(a) depicts the *VVC only* coding scheme. At the sending end, a standard VVC encoder is utilized to encode the input video into the base VVC bitstream, denoted as *bVVC*. The bitstream is decoded back to a human-viewable output format (*YUV'*) with a standard VVC decoder at the receiving end.

This approach is compatible with most machine tasks, as they are often trained on regular, unprocessed images. However, it might not always offer the lowest bitrate because it includes information that is necessary for human consumption but redundant or unnecessary for machine processing.

3.4. *Simulcast* (*VVC and M-VVC*) coding scheme

Fig. 3(b) illustrates the *simulcast* coding scheme, where *VVC and M-VVC* bitstreams are sent in parallel. In this scheme, the input video undergoes the following two independent encoding processes:

1) The conventional encoding path that is identical to the *VVC only* coding scheme, where the input video is encoded into *bVVC* and decoded back into *YUV'* with a standard VVC encoder and decoder, respectively.
2) The machine-optimized encoding path that includes a pre-processing step (Section 3.1) to create a machine-optimized version of the input video (YUV_M). YUV_M is then encoded into a machine-optimized bitstream ($bVVC_M$), which is decoded to a machine-optimized output video called YUV'_M.

It is worth noting that while $bVVC_M$ could theoretically be created with any VCM-like technology, they are not necessarily VVC compliant solutions.

3.5. Proposed *multi-layer VVC* coding scheme

Utilizing multi-layer coding tools in VVC allows encoding multiple video streams simultaneously and merging them into a single bitstream. The BL is encoded, and can also be decoded, independently, whereas the EL uses the BL as an inter-layer reference. Therefore, the BL is also required to decode the EL.

Fig. 3(c) shows the proposed *multi-layer VVC* coding scheme, where the original input video and the machine-optimized version of it, *YUV* and YUV_M (Section 3.1), are encoded with a multi-layer VVC encoder to generate two distinct bitstreams: $bVVC_M$ for the BL and $bVVC_{EL}$ for the EL. They are then combined into a single multi-layer bitstream, denoted as $bVVC_{M+EL}$. The decoding of this bitstream yields two reconstructed outputs: YUV'_M for the BL, tailored for machine tasks, and YUV'_{EL} for the EL, suitable for human viewing. When the machine-optimized video is required, only the BL bitstream $bVVC_M$ is transmitted, but if the full video is

Table 2. Test sequences and relevant coding parameters used in the experiments

Class	Sequence	Seq. ID	Resolution	Fps	# skipped frames	# encoded frames	Intra period	QPs
A	Traffic	A1	2560x1600	30	117	33	32	39, 45, 48, 51, 54, 58
B	ParkScene	B1	1920x1080	24	207	33	32	32, 36, 40, 44, 48, 52
	Cactus	B2	1920x1080	50	403	97	64	43, 46, 49, 52, 55, 58
	BasketballDrive	B3	1920x1080	50	403	97	64	40, 43, 46, 49, 52, 55
	BQTerrace	B4	1920x1080	60	471	129	64	40, 43, 46, 49, 52, 55
C	BasketballDrill	C1	832x480	50	403	97	64	27, 31, 35, 39, 43, 47
	BQMall	C2	832x480	60	471	129	64	27, 32, 37, 42, 47, 52
	PartyScene	C3	832x480	50	403	97	64	31, 35, 39, 43, 47, 51
	RaceHorses	C4	832x480	30	235	65	64	27, 31, 35, 39, 43, 47

Table 3. Coding gain in BD-rate(mAP) of the proposed *multi-layer VVC* over *VVC only* coding scheme for machine only consumption

Seq. ID	Blurred Background				Greyed Background			
	BL Spatial Scaling Ratio				BL Spatial Scaling Ratio			
	1×	0.75×	0.5×	0.25×	1×	0.75×	0.5×	0.25×
A1	-18.3%	-25.9%	-16.4%	-43.0%	-12.2%	-24.8%	-32.3%	-40.0%
B1	-22.9%	-30.8%	-40.8%	-44.1%	-64.0%	-68.2%	-68.4%	-59.0%
B2	-13.1%	-33.1%	-51.4%	-69.1%	-20.5%	-37.3%	-59.0%	-75.3%
B3	-3.3%	-17.6%	-21.6%	-29.7%	-7.9%	-19.3%	-16.3%	-35.3%
B4	-13.9%	-24.8%	-33.6%	-56.6%	-18.8%	-28.1%	-39.3%	-49.1%
Avg.	**-14.3%**	**-26.4%**	**-32.8%**	**-48.5%**	**-24.7%**	**-35.5%**	**-43.1%**	**-51.7%**
C1	4.8%	-13.6%	-12.9%	-9.6%	1.9%	-13.2%	-14.4%	-20.2%
C2	5.3%	-3.9%	-7.6%	-12.8%	3.7%	-9.1%	-1.9%	-8.6%
C3	-16.5%	3.3%	-7.9%	-26.3%	-10.8%	-26.9%	-18.0%	-30.9%
C4	-0.8%	-19.9%	-27.3%	-9.1%	4.9%	-21.1%	-25.3%	-9.9%
Avg.	**-1.8%**	**-8.5%**	**-13.9%**	**-14.5%**	**-0.1%**	**-17.6%**	**-14.9%**	**-17.4%**

Table 4. Coding gain in BD-rate(PSNR) of the proposed *multi-layer VVC* over *simulcast* coding scheme for human consumption

Seq. ID	Blurred Background				Greyed Background			
	BL Spatial Scaling Ratio				BL Spatial Scaling Ratio			
	1×	0.75×	0.5×	0.25×	1×	0.75×	0.5×	0.25×
A1	-21.8%	-11.9%	-6.1%	-1.3%	-20.1%	-11.7%	-5.7%	-1.5%
B1	-14.5%	-8.5%	-4.0%	-0.7%	-14.7%	-9.3%	-3.9%	-0.5%
B2	-33.0%	-19.4%	-7.9%	-1.7%	-32.7%	-19.3%	-7.6%	-1.4%
B3	-39.9%	-21.4%	-8.0%	-1.1%	-39.7%	-21.5%	-8.0%	-1.0%
B4	-23.6%	-13.1%	-4.1%	-0.6%	-23.6%	-13.2%	-3.7%	-1.1%
Avg.	**-26.5%**	**-14.9%**	**-6.0%**	**-1.1%**	**-26.2%**	**-15.0%**	**-5.8%**	**-1.1%**
C1	-46.1%	-18.0%	-5.3%	0.4%	-45.2%	-18.2%	-5.1%	0.3%
C2	-43.2%	-15.9%	-4.4%	0.6%	-44.4%	-15.9%	-4.3%	0.7%
C3	-37.1%	-15.4%	-4.0%	0.5%	-36.0%	-15.4%	-3.8%	0.6%
C4	-47.7%	-17.9%	-4.8%	0.0%	-47.6%	-17.8%	-4.9%	0.2%
Avg.	**-43.5%**	**-16.8%**	**-4.6%**	**0.4%**	**-43.3%**	**-16.8%**	**-4.5%**	**0.5%**

needed for human consumption, the complete bitstream $bVVC_{M+EL}$ is transmitted.

4. EXPERIMENTAL SETUP

Our experimental setup follows the *common test conditions* (*CTC*) [15] defined by JVET.

4.1. Evaluation metrics

The *peak signal-to-noise ratio* (*PSNR*) is utilized to evaluate the encoding distortion in human consumption, whereas the machine performance is assessed with the *mean average precision* (*mAP*). To calculate mAP, the decoded output (*YUV'*, YUV'_M, or YUV'_{EL}) is passed to the machine task used for evaluation (i.e., detectron2 [16]) as defined in the CTC [15].

Bjontegaard Delta Bitrate (*BD-rate*) [17] is applied to quantify the relative bitrate difference between two coding schemes for equal mAP or PSNR. The rate-distortion curves for the BD-rate calculation are interpolated through the rate-distortion points specified by the *quantization parameter* (*QP*). Since mAP is prone to producing non-monotonic rate-distortion curves, cubic curve fitting is applied to the mAP values so that BD-rate values can be calculated in all tests.

4.2. Test sequences and parameters

Table 2 lists the used CTC [15] test sequences that are divided into three different classes with varying resolutions and frame rates. Smaller resolutions are excluded because pre-processing and spatial scalability cannot effectively be used with them. Ground truth data for mAP calculation is provided by SFU [18].

Our study delves into two distinct pre-processing methods, i.e., background blurring and greying. The pre-processing step for the $bVVC_M$ bitstream utilizes YOLOv8x [19] for object detection in ROI map generation. The ROI maps, represented by bounding boxes generated by the YOLOv8x, divide the video frames into background and ROI regions. The background regions are pre-processed, while the ROI regions are left unmodified. The ROI maps are aggregated based on GOP length, here equal to the intra period, to obtain a more stable saliency map that is shared by the pictures in the given GOP.

The modified video is also downscaled with ffmpeg [20] with the spatial scaling ratios of 0.75×, 0.5×, and 0.25×, where each spatial dimension is reduced by the given ratio. Blurring uses a Gaussian filter with a kernel size of 31 and a sigma of 10.

Encoding runs are carried out with the *random access* (*RA*) configuration of the VVC reference software VTM (ver. 20.1) [21]. The QPs used for BD-rate calculations and other coding parameters are provided in Table 2 for each sequence as per the CTC [15]. For multi-layer coding, the default multi-layer configuration as well as the same QPs are used for each layer.

5. EXPERIMENTAL RESULTS

The investigated coding schemes are benchmarked in terms of 1) the machine performance of *multi-layer VVC* over the *VVC only* coding scheme; 2) the human performance of *multi-layer VVC* over the *simulcast* coding scheme; and 3) the coding overhead of *multi-layer VVC* over the *VVC only* coding scheme. Each of these comparisons contain results for both pre-processing methods as well as for four scaling ratios of the BL.

Table 5. Overhead in BD-rate(PSNR) of the proposed *multi-layer VVC* over *VVC only* coding scheme for human consumption

Seq. ID	Blurred Background				Greyed Background			
	BL Spatial Scaling Ratio				BL Spatial Scaling Ratio			
	1×	0.75×	0.5×	0.25×	1×	0.75×	0.5×	0.25×
A1	20.5%	24.4%	18.4%	10.8%	12.6%	17.7%	14.1%	8.0%
B1	15.4%	18.7%	14.3%	8.7%	8.4%	12.4%	10.4%	6.8%
B2	24.4%	37.7%	30.9%	17.5%	19.3%	32.0%	27.3%	16.0%
B3	15.1%	33.6%	28.5%	14.5%	15.0%	33.2%	28.4%	14.5%
B4	25.4%	31.8%	25.1%	13.2%	23.2%	28.4%	23.4%	11.8%
Avg.	**20.1%**	**29.2%**	**23.5%**	**12.9%**	**15.7%**	**24.7%**	**20.7%**	**11.4%**
C1	7.4%	42.2%	35.7%	18.5%	9.5%	41.9%	35.9%	18.3%
C2	11.5%	45.0%	37.0%	19.7%	10.3%	44.8%	36.9%	19.6%
C3	13.3%	31.8%	26.1%	13.4%	15.0%	31.6%	26.2%	13.7%
C4	4.7%	44.6%	38.0%	18.7%	5.0%	44.9%	38.0%	18.9%
Avg.	**9.2%**	**40.9%**	**34.2%**	**17.6%**	**9.9%**	**40.8%**	**34.2%**	**17.6%**

Table 6. Break-even point of the proposed *multi-layer VVC* over *VVC only* coding scheme

Seq. ID	Blurred Background				Greyed Background			
	BL Spatial Scaling Ratio				BL Spatial Scaling Ratio			
	1×	0.75×	0.5×	0.25×	1×	0.75×	0.5×	0.25×
A1	47.2%	51.5%	47.0%	80.0%	49.1%	58.4%	69.6%	83.3%
B1	59.8%	62.2%	74.1%	83.6%	88.4%	84.6%	86.8%	89.7%
B2	35.0%	46.8%	62.5%	79.8%	51.4%	53.9%	68.4%	82.5%
B3	18.1%	34.4%	43.1%	67.2%	34.5%	36.8%	36.5%	71.0%
B4	35.3%	43.8%	57.2%	81.1%	44.7%	49.8%	62.7%	80.7%
Avg.	**39.1%**	**47.7%**	**56.8%**	**78.3%**	**53.6%**	**56.7%**	**64.8%**	**81.4%**
C1	0.0%	24.4%	26.5%	34.3%	0.0%	23.9%	28.6%	52.5%
C2	0.0%	7.9%	17.1%	39.5%	0.0%	16.8%	4.8%	30.5%
C3	55.3%	0.0%	23.2%	66.2%	41.7%	46.0%	40.7%	69.3%
C4	13.8%	30.9%	41.8%	32.7%	0.0%	32.0%	39.9%	34.4%
Avg.	**17.3%**	**15.8%**	**27.2%**	**43.2%**	**10.4%**	**29.7%**	**28.5%**	**46.6%**

5.1. Coding efficiency

Table 3 shows the BD-rate(mAP) performance of the proposed *multi-layer VVC* over *VVC only* coding scheme for machine only consumption ($bVVC_M$ vs. $bVVC$). For Class AB, the BD-rate savings of the *multi-layer VVC* scheme range from −14.3% (at 1× scale) to −48.5% (at 0.25× scale) when the background is blurred. With background greying, the respective savings range from −24.7% to −51.7%. For class C, the savings are more modest with the highest values being −14.5% at 0.25× scale for blurred background and −17.6% at 0.75× scale for greyed-out background. However, with some sequences of class C the *multi-layer VVC* scheme has a higher bitrate than the anchor due to the ROI mask covering most of the frame.

Table 4 reports the BD-rate(PSNR) performance of *multi-layer VVC* over the *simulcast* scheme for human consumption ($bVVC_{M+EL}$ vs. $bVVC_M + bVVC$), where YUV'_{EL} and YUV' are used for the PSNR calculation, respectively. In contrast to the machine performance results, the gain of multi-layer coding decreases as the BL scale decreases. At 0.25× scale, the EL is not able to effectively take advantage of inter-layer references due to the small BL resolution, resulting in effectively the same bitrate as the anchor.

Table 5 lists the coding overhead of the *multi-layer VVC* coding scheme compared to *VVC only*. For class AB, the aggressive downscaling at 0.25× scale overall causes the smallest overhead with a ~11%-13% increase in bitrate over a standard VVC bitstream. However, the 1× scale case has the second lowest overhead with Class AB and is, on average, the most efficient for class C at around 10% BD-rate increase.

5.2. Break-even point

Finally, break-even points [9] are specified between the *multi-layer VVC* and *VVC only* coding schemes. To define the conditions when multi-layer coding is more efficient, we compare the total bitrate of the proposed *multi-layer VVC* scheme to that of the *VVC only* scheme, i.e.,

$$t_h \cdot R_{M+EL} + (1 - t_h) R_M \leq R, \tag{1}$$

where t_h is the fraction of time spent streaming the entire multi-layer bitstream, given a bitrate of R_{M+EL}. The rest of the time is for streaming the BL with a bitrate of R_M. Here, R represents the bitrate of the *VVC only scheme*. Interpreting BD-rate as the relative difference in bitrate between two coding schemes, we can formulate R_M and R_{M+EL} in terms of R using BD-rate:

$$R_M = (1 + BDR_M)R, \tag{2}$$
$$R_{M+EL} = (1 + BDR_{M+EL})R, \tag{3}$$

where, BDR_M is the BD-rate of the BL over *VVC only*, given in Table 3, and BDR_{M+EL} is the BD-rate of *multi-layer VVC* over *VVC only*, given in Table 5. Additionally, in order to convert the BD-rates to fractions, we need to include the addition with one. Substituting (2) and (3) into (1) and solving for t_h we get

$$t_h \leq \frac{-BDR_M}{BDR_{M+EL} - BDR_M}. \tag{4}$$

Using (4) we can calculate the break-even points shown in Table 6.

The overhead of multi-layer coding can be mitigated if the EL is only needed occasionally. When the share of the time for human consumption is less that the break-even point presented in Table 6, the *multi-layer VVC* scheme outperforms the *VVC only* scheme.

Overall, a smaller scale allows the EL to be used more frequently while keeping the total bandwidth lower than only using VVC. At 0.25× scale and class AB, EL could be used roughly 80% of the time and still be equal to *VVC only*. Without BL downscaling, this value drops down to 40%-50%. For some class C sequences however, multi-layer coding is never better than *VVC only*, because of the positive BD-rate of the machine optimized bitstream in those cases. Regardless, with smaller scales, class C provides break-even points ranging from 27% to 46%.

5.3. Discussion

The greyed-out background pre-processing method is shown to be up to 10% better than blurring in terms of machine performance, but the pre-processing method does not have a significant effect on the multi-layer coding efficiency. This makes both pre-processing methods viable options for multi-layer coding despite them making it harder to use inter-layer references.

In terms of BD-rate, the BL scale has a larger effect than the pre-processing method. The best machine performance is achieved with

the BL scale of 0.25×, at least in the examined QP range. On the other hand, scale has the opposite effect on the EL efficiency, resulting in nearly no improvement over simulcast in the 0.25× case. As such, choosing the BL scale is a tradeoff between BL and EL efficiency. Examining the rate-distortion curves of 0.25× scale, at higher bitrates, shows that the absolute mAP falls short of *VVC only*. As such, smaller scales work best if a minimal bitrate is desired, but higher scales should be chosen for maximal machine performance.

In general, the proposed *multi-layer VVC* scheme may not work well with smaller resolutions that have large objects, but it can provide notable benefits in hybrid machine-human use cases with larger resolutions or if the human bitstream is required infrequently. The pre-processed BL comes with a notable overhead over the *VVC only* scheme, but if the EL is not needed at all times, the multi-layer approach can still be more efficient, as seen in Table 6. In the best case, the full bitstream could be used 80% of the time, and the benefit keeps increasing even further with lower EL usage. Moreover, even with the BL scale of 1×, the EL usage frequency of 50% is still reasonable with higher resolutions, but if the resolution is small, the break-even point can be considerably smaller. In addition, with certain types of content, multi-layer coding is never better than *VVC only* coding because the machine optimized bitstream does not provide any benefit over VVC.

6. CONCLUSION

In this paper, we proposed using VVC multi-layer coding tools to fulfill the needs of hybrid machine-human coding in the context of VCM. This is achieved by the proposed coding scheme that supports a machine optimized BL and a human-viewable EL. Furthermore, we investigate two ROI-based pre-processing methods, called blurring and greying, for generating the machine optimized input. Our results show that both pre-processing methods are compatible with multi-layer coding, which provides significant coding gains over simulcast. Even though the multi-layer approach comes with a notable overhead compared to using a single VVC bitstream, our results show that when the human-viewable bitstream is not always required, our proposal is still more efficient than a standard VVC bitstream.

7. REFERENCES

[1] L. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: a paradigm of collaborative compression and intelligent analytics," *IEEE Trans. Image Process.*, vol. 29, pp. 8680–8695, Aug. 2020.

[2] ITU-T and ISO/IEC JTC 1, "Versatile Video Coding," Recommendation ITU-T Rec. H.266 and ISO/IEC 23090-3 (VVC), Sep. 2023.

[3] J. Chen, C. Hollmann, and S. Liu, "Optimization of encoders and receiving systems for machine analysis of coded video content (draft 3)," *document JVET-AE2030-v1*, Geneva, Switzerland, Jul. 2023

[4] B. Bross et al., "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736-3764, Oct. 2021.

[5] S. Wang, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Scalable facial image compression with deep feature reconstruction," in *Proc. Int. Conf. Image Process.*, Taipei, Taiwan, Aug 2019.

[6] S. Yang, Y. Hu, W. Yang, L. -Y. Duan, and J. Liu, "Towards coding for human and machine vision: scalable face image coding," *IEEE Trans. Multimedia*, vol. 23, pp. 2957-2971, Mar. 2021.

[7] N. Yan, C. Gao, D. Liu, H. Li, L. Li, and F. Wu, "SSSIC: semantics-to-signal scalable image coding with learned structural representations," *IEEE Trans. Image Process.*, vol. 30, pp. 8939-8954, Oct., 2021.

[8] H. Lin, B. Chen, Z. Zhang, J. Lin, X. Wang, and T. Zhao, "DeepSVC: deep scalable video coding for both machine and human vision," in *Proc. ACM Int. Conf. Multimedia*, New York, New York, USA, Oct. 2023.

[9] H. Choi and I. V. Bajić, "Scalable video coding for humans and machines," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Shanghai, China, Sep. 2022.

[10] A. Harell, Y. Foroutan, and I. V. Bajić, "VVC+M: plug and play scalable image coding for humans and machines," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Brisbane, Australia, Aug. 2023.

[11] J. Seppälä et al., "Enhancing image coding for machines with compressed feature residuals," in *Proc. IEEE Int. Symp. Multimedia*, Naple, Italy, Jan. 2021.

[12] K. Ogasawara, T. Miyazaki, Y. Sugaya, and S. Omachi, "Object-based video coding by visual saliency and temporal correlation," *IEEE Trans. Emerging Topics Comput.*, vol. 8, no. 1, pp. 168-178, Jan.-Mar. 2020.

[13] D. Grois and O. Hadar, "Complexity-aware adaptive preprocessing scheme for region-of-interest spatial scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 1025-1039, Jun. 2014.

[14] A. D. Bagdanov, M. Bertini, A. Del Bimbo, and L. Seidenari, "Adaptive video compression for video surveillance applications," in *Proc. IEEE Int. Symp. Multimedia*, Dana Point, California, USA, Dec. 2011.

[15] S. Liu and C. Hollman, "Common test conditions for optimization of encoders and receiving systems for machine analysis of coded video content," *document JVET-AF2031-v1*, Hannover, Germany, Oct. 2023.

[16] Y. Wu, A. Kirillov, F. Masa, W.-Y. Lo, and R. Girschick, "Detectron2," 2019. Accessed: Dec. 2023. [Online]. Available: https://github.com/facebookresearch/detectron2

[17] G. Bjøntegaard, "Improvements of the BD-PSNR model," *document VCEG-AI11*, Berlin, Germany, Jul. 2008.

[18] H. Choi, E. Hosseini, S. R. Alvar, R. Cohen, I. V. Bajić, "A dataset of labelled objects on raw video sequences," *data in brief*, Feb. 2021.

[19] G. Y. Jocher, A. Chaurasia, and J. Qiu, "YOLOv8," Ultralytics, 2023. [Online]. https://github.com/ultralytics/ultralytics

[20] "FFmpeg," Accessed: Nov. 2023. [Online]. Available: https://ffmpeg.org

[21] "VVC Reference Software Version 20.1," Accessed: Nov. 2023. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-20.11