Motion-Vector-Driven Lightweight ROI Tracking for Real-Time Saliency-Guided Video Encoding

Tero Partanen, Miika Kotajärvi, Alexandre Mercat, and Jarno Vanne
Ultra Video Group, Tampere University, Tampere, Finland
{tero.partanen, miika.kotajarvi, alexandre.mercat, jarno.vanne}@tuni.fi

***Abstract*—The huge computation burden of state-of-the-art video coding technologies can be mitigated with Region-of-Interest (ROI) techniques that limit the highest coding effort to salient regions. However, the complexity overhead of saliency detection can easily cancel out the speed gain of ROI coding. This work introduces a lightweight ROI tracking technique that can be used in place of compute-intensive ROI detection to guide a video encoder in inter coding. Low computational overhead is achieved by feeding motion vectors (MVs) of a video encoder back to our neural network that is trained for accurate estimation of ROI movement and size changes. The network training is carried out with our new dataset that is also released in this work to foster the development of head tracking techniques in applications like video conferencing. Our experimental results demonstrate substantial speedups with minimal accuracy trade-offs over traditional salient object detection (SOD) methods. In scenarios, where a single ROI is tracked with a 64-frame detection interval, our solution obtains up to 50-fold speedup with accuracy of 87% and an average ROI center error of 16 pixels. These results confirm that our ROI tracking approach is a potential technique for low-cost and low-power streaming media applications.**

Keywords—Saliency-guided encoding, Region-of-Interest (ROI), ROI tracking, deep learning, motion vector (MV)

I. INTRODUCTION

The skyrocketing growth of visual data consumption by humans and machines has led to an unprecedented surge in global video traffic. This trend, coupled with the advent of high-quality immersive media applications, calls for more sophisticated video compression technologies that are able to overcome the constraints imposed by existing network and storage capacities. The latest video coding standards, like *High Efficiency Video Coding* (*HEVC/H.265*) [1] and *Versatile Video Coding* (*VVC/H.266*) [2], are in place to mitigate video bandwidth demands, but their computational requirements are cumbersome to reach without optimizations, particularly in real-time streaming media domain.

Traditional video coding tools, like those of HEVC and VVC, primarily focus on eliminating statistical, spatial, and temporal redundancies inherent in video. However, they tend to be agnostic to the perceptual redundancy of human visual attention. This gives rise to a paradigm shift towards content-aware video coding that is built upon saliency-based or *region of interest* (*ROI*) techniques [3]. These techniques were initially developed for image compression [4] and subsequently extended for video domain [5]. In recent years, they have gained momentum, particularly as machines have become active content consumers.

Saliency-guided techniques compress less efficiently the pixels or regions deemed visually important, whereas higher compression is applied to remaining areas. They effectively seek to strike a balance between bitrate reduction and the preservation of visually relevant details. Fig. 1 illustrates the basic operation principle of saliency-guided compression. Typically, a saliency map is generated for raw video frames before encoding to distinguish between areas of high and low importance. This approach not only reduces bitrate but it can also sustain or even enhance the *quality of experience* (*QoE*) of human viewers and comprehension in machine vision.

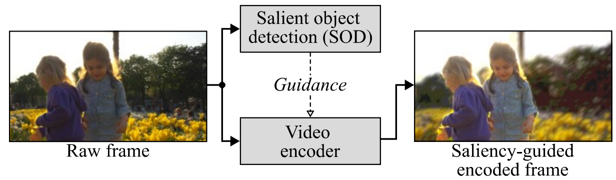

Fig. 1. Basic operating principle of saliency-guided compression.

The rise of *deep learning* (*DL*)-based *salient object detection* (*SOD*) methods [6] has been pivotal in advancing saliency-guided coding in recent years. Early models like those by Itti *et al.* [7] and Achanta *et al.* [8] utilized basic signal processing techniques that often lacked the accuracy of modern DL approaches [9]–[11]. On the other hand, the advanced DL models tend to carry significant computational overhead that increases computation time beyond the limits of real-time applications. For instance, state-of-the-art models by Hou *et al.* [9], Qin *et al.* [10], and Wang *et al.* [11], were limited to 2 *frames per second* (*fps*), 25 fps, and 24 fps, respectively, even with high-end *graphics processing units* (*GPUs*) and downscaled video inputs.

In this work, we introduce a lightweight ROI tracking scheme for real-time saliency-guided video coding. Our main objective is to avoid executing compute-intensive ROI detection on every frame by replacing it with the proposed ROI tracking on intermediate inter frames and thereby increasing the interval between successive detections. Our proposal is based on interdependent communication between ROI tracking and video encoding process. The video encoder receives an updated saliency-based guidance on every frame, whereas motion information of the encoder, i.e., *motion vectors* (*MVs*) [12] are fed back to ROI tracking when an encoder operates in inter mode.

To the best of our knowledge, this is the first work to utilize MV-driven ROI tracking to mitigate computational cost of SOD in saliency-guided video coding. Our solution comes with a novel DL-based model designed to estimate the movement and size changes of ROI objects with minimal computational overhead. The model is trained with our new video dataset that is also released to offer an online resource for the development of MV-based tracking technologies.

The rest of the paper is organized as follows. Section 2 reviews the literature on ROI tracking. Section 3 presents our saliency-guided video coding framework and Section 4 introduces the proposed lightweight ROI tracking method and a dataset used to develop it. Experimental setup and results are reported in Section 5. Finally, Section 6 concludes the paper.

This work was supported in part by the AI-based Situational Awareness (AISA) project led by Nokia and funded by Business Finland, and the Research Council of Finland (decision no. 349216).

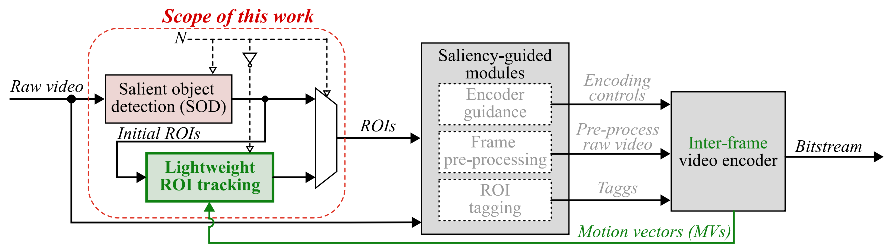

Fig. 2. The system architecture of the proposed saliency-guided video coding framework, where the main contribution of this work is in green.

TABLE I. RELATED WORKS FOR MV-BASED OBJECT TRACKING

Target	Ref	Description
Object tracking	[14]	Demonstrated object tracking using MVs from MPEG-2 encoders, pioneering the use of encoder-generated MVs for tracking specific regions.
	[15]	Introduced an improved tracking algorithm with separate layers for more precise MV-based tracking and size estimation.
	[16] [17]	Fused traditional tracking-by-detection with MV-based tracking, combining methodologies for improved object tracking efficiency.
Noise reduction in tracking	[18]	Proposed spatiotemporal filtering for MV-based tracking, addressing noise in MV data for more accurate tracking.
	[19]	Improved MV-based tracking accuracy by presenting MV filtering techniques to clean noisy and outlier MVs.
	[20]	Employed MVs to assist a particle filter-based object tracking, showcasing the utility of MVs for better tracking accuracy.
Computational complexity reduction	[21]	Utilized MVs to reduce the computational complexity of semantic segmentation tasks, highlighting the efficiency benefits of MVs in computer vision.
	[22]	Proposed a detection skipping method for MV-based object tracking, significantly reducing detector utilization and demonstrating the computational efficiency of using MVs in tracking.
Encoder guidance	[23]	Explored MVs for global motion compensation, laying groundwork for saliency-guided coding.
	[24]	Utilized MVs for temporal-based saliency, enhancing video coding efficiency through a temporal lens.
	[25]	Enhanced video coding efficiency by using MVs for saliency enhancement, improving compression strategies.

II. RELATED WORKS IN ROI TRACKING

Object tracking is a fundamental task in computer vision [13]. It involves detecting and locating objects over consecutive video frames, maintaining their identities, and tracking their trajectories. Conventionally, this task is executed through tracking-by-detection techniques, utilizing pixel-domain data from video frames. While effective, these methods often face challenges such as high computational requirements and sensitivity to occlusions and motion blur.

The transition to employing MVs from video encoders streamlines the object tracking process by bypassing the intensive computation of motion information from pixel data. Table I categorizes the existing works that utilize encoder-derived MVs for object tracking [14]–[25]. Notably, the literature reveals limited research on leveraging MVs for encoder guidance since prior studies mainly focus on the decoder side. Our solution goes even one step further by using a feedback loop from the encoder to enhance both coding efficiency and accuracy.

III. SALIENCY-GUIDED VIDEO CODING FRAMEWORK

Fig. 2 depicts our saliency-guided video coding framework that is accelerated with the proposed ROI tracking method. The SOD module processes the raw input video to extract salient regions that are used to specify ROIs. The ROI data may be used, e.g., by:

- An encoder guidance module that may use ROIs to adjust the *quantization parameter* (*QP*) values for each coding block so that higher QPs are assigned to less important areas like the background to save bits [24], [26].
- A frame pre-processing module that can process the raw input video before encoding with filters that might blur or obscure non-essential regions, leading to bit savings. This approach is particularly useful in computer vision and *Video Coding for Machines* (*VCM*) [27].
- A ROI tagging module that can enrich the bitstream with detailed metadata, like bounding box coordinates and object types, derived from SOD-identified salient objects.

Our approach diverges from traditional saliency-guided coding scheme by integrating a *lightweight ROI tracking* module, which is designed to monitor the spatial movement and size changes of ROIs across successive frames. This approach reduces computational burden by avoiding the need for continuous SOD execution. It applies initial ROIs with MVs of the frame being encoded to accurately update the position and size of ROIs for the next frame. MVs computed by the encoder speedup tracking as they represent the displacement of pixel blocks between frames.

For the first frame of a tracking sequence, initial ROIs are derived from SOD; for subsequent frames, they are carried over from the encoding output of the previous frame. The tracking module is applied over N consecutive frames, where N can be preset or dynamically adjusted as per the encoding process. Traditional saliency-guided coding schemes tend to be compatible with intra and inter encoders, whereas our proposal particularly requires an inter encoder, like that of HEVC or VVC, that is capable of generating MVs.

The remainder of this paper focuses on the design and evaluation of the proposed *lightweight ROI tracking* method, as highlighted in Fig 2.

IV. PROPOSED LIGHTWEIGHT ROI TRACKING

To streamline the development of our *lightweight ROI tracking* method, we focus on head tracking, a domain where human attention naturally converges [28]. The pivotal role of head detection underscores its relevance [29]–[32] in real-time applications, such as in video conferencing, surveillance, and driver monitoring. Head tracking also represents a common scenario, in which the ROI is a distinct, moving object against a static background. Hence, it broadens the adaptability and potential of our method for a variety of real-world applications.

A. Proposed Head Tracking Dataset

Creating a robust DL model for the applied video encoders requires a comprehensive dataset that is composed of raw, uncompressed video sequences. Because none of the existing dataset met our specific requirements, we generated a new test video set specialized in head tracking. It is available online at

https://ultravideo.fi/datasets.html

The proposed dataset encompasses 85 video sequences (labeled from *a1.yuv* to *a85.yuv*), each ranging from 5 to 30 seconds, totaling to 29 824 frames. They were captured with *Logitech Brio 4K Ultra HD* webcam and *Logitech Meetup 4K* conferencing camera in 1920×1080 resolution at 30 fps in 8-bit YUYV 4:2:2 format and converted to 8-bit YUV 4:2:0 videos using *ffmpeg* [33]. The footage reflects common scenarios in video communication settings, where moving individuals are against mostly static background. The proposed dataset includes a range of movements, with a focus on lateral motion and varying distances from the camera, to represent realistic interactions in an office environment.

Annotations in the dataset include bounding boxes around people's heads. In addition, the dataset includes MVs that represent a transition of 16×16 blocks in each frame. The MVs were generated with our *Kvazaar* HEVC encoder [34] that was set to *ultrafast* preset that attains real-time coding speed.

B. Proposed Lightweight ROI Tracking Scheme

Fig. 3 presents the proposed DL-based *lightweight ROI tracking* module that uses full-frame MVs and initial ROI coordinates from the preceding frame as inputs. It outputs estimated ROI positions and dimensions for the current frame. By using DL to improve MV-based tracking, we aim to eliminate the need for various other refinement steps such as MV filters [18], [19] or hand-crafted ROI size estimation [15].

The encoder-generated MVs are formatted into a three-channel input: horizontal movement vector, vertical movement vector, and a flag indicating either intra or inter prediction [1], [2]. Intra blocks lack motion data, so their vectors are set to zero. The proposed method initially selects the MVs that either partially or fully overlap with the ROI on the frame, as illustrated in Fig. 3. The selected MVs are subsequently processed by a multi-layer neural network, designed to infer both the displacement (Δx, Δy) and size changes (Δx_{scale}, Δy_{scale}) of the ROI.

In the *ROI update* phase, the initial ROI coordinates from the SOD are used as a starting point, and the network outputs are applied to incrementally adjust the ROI coordinates from frame to frame. The updated ROI coordinates are also employed to select the MVs of the next frame, ensuring continuous tracking until a new SOD detection cycle is initiated. For scenarios involving multiple ROIs, each ROI is processed independently.

C. Implementation and Training of the Proposed Network

The proposed multi-layer neural network was implemented using the *PyTorch* framework (version 2.2.1) [35]. The network architecture is made up of a resizing layer, two convolutional layers to extract features by filtering the input, and a fully connected layer to generate the outputs (Δx, Δx_{scale}, Δy, and Δy_{scale}). *Rectified linear units* (*ReLU*) serve as activation functions on the outputs of the convolutional layers.

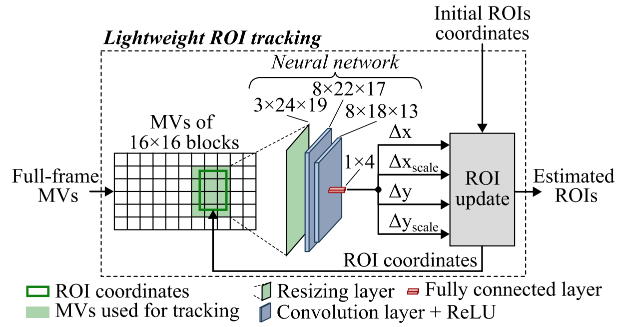

Fig. 3. Overview of the proposed lightweight ROI tracking module.

The resizing layer was configured to standardize the input dimensions to 3×24×19 for the convolutional layers, which then produced output sizes of 8×22×17 and 8×18×13, respectively. Finally, the data were processed by a fully connected layer consisting of 4 neurons.

The network was trained with our dataset using 47 video sequences (*a1.yuv* through *a47.yuv*), which contain a total of 18 018 frames. Ground truths for training were derived from the changes in ROI bounding box dimensions and positions across successive frames in a sequence. After hyperparameter optimization, the network training was characterized by an input size of 24×19, spanning 15 epochs, with a batch size of 2, learning rate set at 10^{-5}, employing *stochastic gradient descent* (*SGD*) as the optimizer, and *mean squared error* (*MSE*) for the loss function. These parameters were meticulously chosen to balance model accuracy and complexity.

V. Experimental Setup and Results

A. Experimental Setup

The aim of the proposed *lightweight ROI tracking* method is to replace the SOD model by integrating movement-compensated ROI estimations between detections. Therefore, we used frame-by-frame SOD detection results as an anchor to assess the accuracy of our proposal. For the sake of a real-time application context, YOLOv8n was employed for SOD model, as it is the most lightweight model in the state-of-the-art YOLOv8 object detection framework [36]. The pre-trained SOD model was retrained on *HollywoodHeads* [37] dataset for head detection.

The proposed test videos labeled a48−a85 were used for evaluation because they were excluded in the training phase. For the generalizability of the results, supplementary evaluations were conducted on five well-known test videos: *Johnny*, *KristenAndSara*, *vidyo1*, *vidyo3*, and *vidyo4* [38]. Their results were reported separately.

Our method was evaluated against two practical approaches:

- ***No-tracking*** that bypasses SOD detection for *N* frames and maintains static ROIs throughout the detection interval.
- ***Mean*** that averages MV inputs for ROI tracking by giving specific emphasis to MVs that partially overlap with ROI edges, as described by Sutter *et al.* [15].

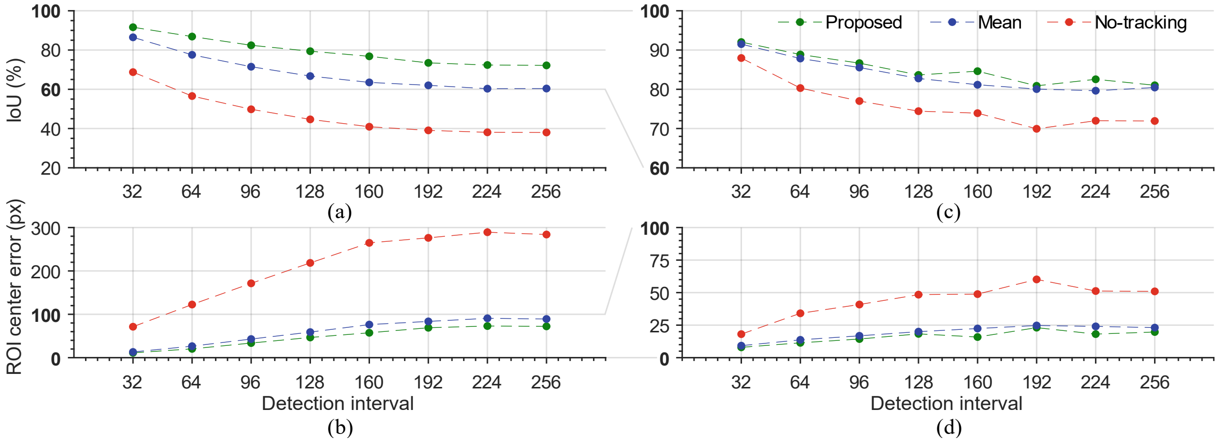

Fig. 4. Accuracy comparison. (a) Average IoU and (b) average ROI center error across the proposed test videos. (c) Average IoU and (d) Average ROI center error across the well-known test videos [38]. Note that the vertical scales differ from (a) to (c), and from (b) to (d).

TABLE II. FRAME-WISE COMPARISON OF COMPUTATIONAL COST

		YOLOv8n		Proposed
Complexity in FLOPs		8.7G		0.47M
Time in ms (frame-wise speedup)	1 ROI	**Single-core** **97** (×1)	**Multi-core** **53** (×2)	**0.43** (×226)
	5 ROIs			**1.68** (×58)
	10 ROIs			**3.16** (×31)

Average accuracy of the approaches was measured across a range of SOD detection intervals, from 32 to 256 frames. The applied evaluation metrics were *intersection over union* (*IoU*) and ROI center error, which were both computed using frame-by-frame YOLOv8n detection results as anchor. Computational costs were measured in terms of *floating-point operations* (*FLOPs*) and execution times on a high-end laptop equipped with *Intel i7-12700H* CPU. For a fair comparison, all experiments were conducted on a single CPU core. Additionally, multi-core results for SOD were executed for reference.

B. *Experimental Results*

Fig. 4 presents the accuracy comparison between the three approaches. Fig. 4(a) and 4(c) show the IoU scores averaged across the proposed and the well-known test videos, respectively. Similarly, Fig. 4(b) and Fig. 4(d) depict the ROI center error metrics for both sets. A better IoU score indicates higher tracking accuracy, whereas a lower ROI center error denotes a greater precision.

Fig. 4(a) shows that the proposed method outperforms the others and the *no-tracking* approach underperforms significantly. An anticipated reduction in IoU scores is observed as the detection interval increases. In Fig. 4(b), the deviation of the *no-tracking* approach is even more pronounced, though the accuracies of *mean* and the proposed one are relatively closer. This phenomenon can be attributed to the proposed ROI size estimation that enhances IoU scores without directly influencing the center error. However, the ROI size estimation may also slightly reduce the center error as it allows for more accurate selection of MVs for predicting subsequent ROIs. The results with additional test videos in Fig. 4(c) and 4(d) confirm the superior performance of the proposed method across diverse content.

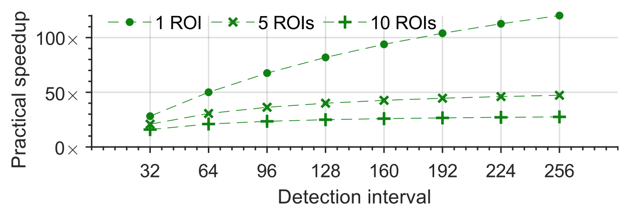

Fig. 5. Practical speedup of the proposed scheme (SOD and *lightweight ROI tracking*) over SOD only.

Table II shows the frame-wise computational complexity comparison of the SOD and our proposal. Frame-wise speedup calculations were benchmarked against single-core execution time of YOLOv8n. Notably, the computational cost of YOLOv8n exceeds that of our tracking method by approximately 18 500 times in terms of FLOPs. Furthermore, our method demonstrates substantial frame-wise speedups over the SOD model, around ×226, ×58, and ×31 for tracking 1, 5, and 10 ROIs, respectively. The speedup diminishes as the number of tracked ROIs increases, because the proposed technique takes care of each ROI individually and is here implemented through a sequential processing.

Fig. 5 illustrates the practical speedup achieved by integrating the proposed lightweight ROI tracking with standard SOD, relative to using SOD alone. The results show that the speedup scales with the detection interval and inversely with the number of ROIs, aligning with the results of Table II. For instance, a 50-fold increases in speed is noted for single ROI tracking over an interval of 64, resulting in IOU accuracy of about 87% as shown in Fig. 4(a).

VI. CONCLUSION

This work introduced a novel lightweight ROI tracking method that is designed to reduce the computational overhead of saliency detection in video coding. By utilizing MVs for ROI tracking, we achieved a frame-wise speedup of up to ×226 over the traditional SOD techniques. The results confirm the feasibility of our proposal for low-cost and energy-efficient streaming applications.

Our future work will explore more sophisticated tracking techniques, like occlusion handling, adaptive interval control, and incorporating temporal memory to improve performance in dynamic video contexts. All these new techniques are seen to pave the way for more advanced saliency-guided coding.

References

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

[2] 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.

[3] Y. Zhang, L. Zhu, G. Jiang, S. Kwong, and C. C. Jay Kuo, "A survey on perceptually optimized video coding," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–37, Dec. 2023.

[4] J. Mannos and D. Sakrison, "The effects of a visual fidelity criterion of the encoding of images," *IEEE Trans. Inf. Theory*, vol. 20, no. 4, pp. 525–536, Jul. 1974.

[5] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1304–1318, Oct. 2004.

[6] H. Zhou, Y. Lin, L. Yang, J. Lai, and X. Xie, "Benchmarking deep models on salient object detection," *Pattern Recognit.*, vol. 145, 109951, Jan. 2024.

[7] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[8] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Miami, Florida, USA, Jun. 2009, pp. 1597–1604.

[9] Q. Hou, M. M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, Apr. 2019.

[10] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, California, USA, Jun. 2019, pp. 7471–7481.

[11] H. Wang, C. Chenglizhao, L. Linfeng, and P. Chong, "Video saliency object detection with motion quality compensation," *Electron.*, vol. 12, no. 7, Mar. 2023.

[12] K. Ugur et al., "Motion compensated prediction and interpolation filter design in H.265/HEVC," *IEEE J. Sel. Topics Signal Process*., vol. 7, no. 6, pp. 946–956, Jul. 2013.

[13] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: a literature review," *Artif. Intell.*, vol. 293, Apr. 2021.

[14] L. Favalli, A. Mecocci, and F. Moschetti, "Object tracking for retrieval applications in MPEG-2," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 427–432, Apr. 2000.

[15] R. D. Sutter, K. D. Wolf, S. Lerouge, and R. V. de Walle, "Lightweight object tracking in compressed video streams demonstrated in region-of-interest coding," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 97845, Jan. 2007.

[16] L. Bommes, X. Lin, and J. Zhou, "MVmed: fast multi-object tracking in the compressed domain," in *Proc. IEEE Conf. Ind. Electron. Appl.*, Kristiansand, Norway, Nov. 2020, pp. 1419–1424.

[17] Q. Liu, B. Liu, Y. Wu, W. Li, and N. Yu, "Real-time online multi-object tracking in compressed domain," *IEEE Access*, vol. 7, pp. 76489–76499, Jun. 2019.

[18] R. C. Moura and E. M. Hemerly, "A spatiotemporal motion-vector filter for object tracking on compressed video," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveillance*, Boston, Massachusetts, USA, Oct. 2010, pp. 427–434.

[19] W. Li and D. Powers, "Multiple object tracking using motion vectors from compressed video," in *Proc. Int. Conf. Digit. Image Comput.: Techn. Appl.*, Sydney, New South Wales, Australia, Nov. 2017, pp. 1–5.

[20] H. Wang, J. Shen, Z. Chen, and J. Shen, "A fast object tracking approach based on the motion vector in a compressed domain," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 1, Jan. 2013.

[21] S. Jain and J. E. Gonzalez, "Fast semantic segmentation on video using block motion-based feature interpolation," in *Proc. Eur. Conf. Comput. Vis. Workshops*, Sep. 2018, pp. 3–6.

[22] T. Ujiie, M. Hiromoto, and T. Sato, "Interpolation-based object detection using motion vectors for embedded real-time tracking systems," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, Utah, USA, Jun. 2018, pp. 616–624.

[23] S. Zhu and Z. Xu, "Spatiotemporal visual saliency guided perceptual high efficiency video coding with neural network," *Neurocomputing*, vol. 275, pp. 511–522, Jan. 2018.

[24] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19–33, Jan. 2014.

[25] S. Zhu, C. Liu, and Z. Xu, "High-definition video compression system based on perception guidance of salient information of a convolutional neural network and HEVC compression domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 1946–1959, Jul. 2020.

[26] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational HEVC coding with hierarchical perception model of face," *IEEE J. Sel. Topics Signal Process*., vol. 8, no. 3, pp. 475–489, Jun. 2014.

[27] 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.

[28] M. Cerf, P. Frady, and C. Koch, "Faces and text attract gaze independent ofthe task: experimental data and computer model," *J. Vis.*, vol. 9, no. 12, pp. 1–15, Nov. 2009.

[29] X. Deng, M. Xu, and Z. Wang, "A ROI-based bit allocation scheme for HEVC towards perceptual conversational video coding," in *Proc. Int. Conf. Adv. Comput. Intell.*, Hangzhou, China, Oct. 2013, pp. 206–211.

[30] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational HEVC coding with hierarchical perception model of face," *IEEE J. Sel. Topics Signal Process*., vol. 8, no. 3, pp. 475–489, Jun. 2014.

[31] Z. Liu, X. Pan, Y. Li, and Z. Chen, "A game theory based CTU-level bit allocation scheme for HEVC region of interest coding," *IEEE Trans. Image Process.*, vol. 30, pp. 794–805, Nov. 2021.

[32] J. Bi, L. Wang, Y. Han, and C. Zhou, "Real-time face perception based encoding strategy optimization method for UHD videos," *IET Image Process*., vol. 17, no. 9, pp. 2764–2779, May 2023.

[33] FFmpeg. [Online]. Available: https://www.ffmpeg.org/. (accessed Feb. 29, 2024).

[34] A. Lemmetti, M. Viitanen, A. Mercat, and J. Vanne, "Kvazaar 2.0: fast and efficient open-source HEVC inter encoder," in *Proc. ACM Multimedia Syst. Conf.*, Istanbul, Turkey, Jun. 2020, pp. 237–242.

[35] A. Paszke *et al.*, "Pytorch: an imperative style high-performance deep learning library," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Vancouver, British Columbia, Canada, Dec. 2019, pp. 8026–8037.

[36] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO." [Online]. Available: https://github.com/ultralytics/ultralytics/. (accessed Jan. 31, 2024).

[37] T. -H. Vu, A. Osokin, and I. Laptev, "Context-aware CNNs for person head detection," in *Proc. IEEE Int. Conf. Comput. Vision*, Santiago, Chile, Dec. 2015, pp. 2893–2901.

[38] Xiph.org. Video Test Media. [Online]. Available: https://media.xiph.org/video/derf/ (accessed Jan. 31, 2024).