Rate-Distortion Optimized Temporal Segmentation Using Reinforcement Learning for Video Coding

Jung-Kyung Lee*, Nayoung Kim*, and Je-Won Kang*,†
* Department of Electronic and Electrical Engineering, Ewha W. University, Seoul, Korea
† Graduate Program in Smart Factory, Ewha W. University, Seoul, Korea
E-mail: jungkyong1204@gmail.com;12skdud21@naver.com;jewonk@ewha.ac.kr

***Abstract*—In this paper, we present a reinforcement learning (RL)-based coding method to recursively divide video frames into several groups displaying similar temporal characteristics and improve rate-distortion (R-D) performance. Although the previous works have attempted to challenge the problem with analytical models, it was difficult to address complicated dependencies of video frames. In the proposed method, we cast the recursive problem as a sequence of a state-action for an agent to conduct an RL, by partitioning the current group to the half. The optimal solution is obtained by maximizing a reward function of the RL policy. Experimental results demonstrate that the proposed method can adapt to a video sequence whereas a fixed coding scheme cannot efficiently achieve optimal coding performance in dynamic video sequences.**

I. Introduction

Video contents are everywhere in our daily life. The video coding technology with compression capabilities has been developed persistently, and the latest versatile video coding (VVC) yields coding efficiency about two times better than high efficiency video coding (HEVC) [1]. However, the growth rate of video consumptions becomes higher than ever before. Accordingly, to make a breakthrough for higher coding efficiency, many researchers are paying attention to the study of applying machine learning to video coding [2]–[7], [21].

Video coding efficiency can be significantly enhanced by exploiting the enormous temporal correlation which is the major part of the video compression task. There have been several studies to reduce temporal redundancy in the direction of improving more accurate representation of a motion vector and a reference block using deep learning [2]–[4]. In [2], [3], Lee et al. proposed a coding scheme for generating a virtual reference frame (VRF) from a previous video frame using convolutional neural network (CNN) in order to address irregular and dynamically changing motions and using the VRF to predict the current sample. In [4], super-resolution deep learning technology is applied as an upsampling method of a reference frame for a sub-pixel motion prediction. In [7], a prediction block generated by linearly combining two block signals in bidirectional prediction is further refined to increase the accuracy. CNN is trained to approximate the reference block to the original block.

Although many recent studies developed sophisticated inter-prediction methods, they have overlooked a temporal dependency between the current frame and the reference frame. In fact, since a coding of a preceding frame can significantly affect coding performance of a subsequent frames, the coding dependency needs to be carefully addressed. Previous studies have realized the intriguing problem and attempted to solve the temporal dependency to improve coding efficiency [9]–[11]. In [9], temporally dependent rate-distortion optimizations was conducted by calculating the distortion propagated from the reference frames. In [11], [12], graph theoretical approaches were employed. They could exploit more efficient predictions to use more bits for key frames, when a video included dynamic scene changes.

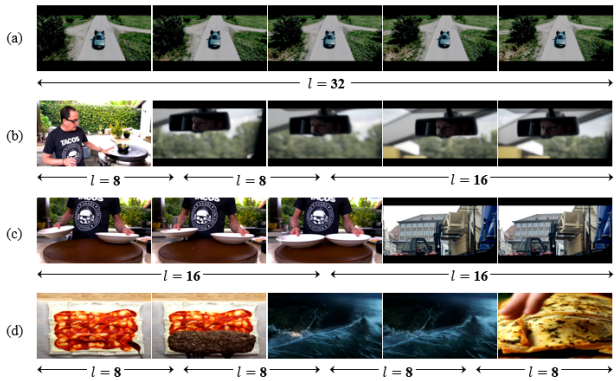

Fig. 1. Sample videos for which the proposed method conducts to divide video frames into several divisions.

In this paper, we introduce a reinforcement learning (RL) to recursively divide video intervals into several groups to include temporally consistent information and improve rate-distortion (R-D) performance. The previous works have attempted to challenge the problem with analytical models [9]–[11] or apply temporal action proposals to perform temporal localization and segmentation [8]. However, it was difficult to address complicated dependencies of videos displaying dynamic spatial and temporal characteristics. In the proposed method, we make a partition of a video interval as a sequence of a state-action for an agent to conduct an RL. Fig. 1 exhibits our motivation. When encoding a homogeneous video sequence as in Fig. 1, a larger group-of-picture (GOP) would be efficient as the reference frames share much information. In contrast, a smaller one can be exploited to handle the different properties after a scene change, by partitioning the current group to the

half. The optimal solution is obtained by maximizing a reward function of the RL policy. In our implementation, we use an adaptive GOP to encode the interval which is a power of two as shown in Fig. 1. In this manner, the proposed method can adapt to a video sequence whereas a fixed coding scheme in the previous works cannot efficiently achieve optimal coding performance in dynamic video sequences.

The rest of the paper is organized as follows. In Sec. II, we review the related works and the RL. We describe the proposed method in Sec. III. Experimental results are presented in Sec. IV. The conclusion is remarked in Sec. V.

II. Background

A. Reinforcement Learning

An RL system is defined as a tuple of (s, a, r, t), in which s is a state, a is an action, r is a reward, and t is a transition to the next state s'. The agent a trains a policy to take an action to maximize r. At every moment t, the policy $\pi(s_t)$ determines an action a_t to move to the next state from s_t to receive a immediate reward of r_{t+1} to maximize the total reward. The agent keeps producing a sequence of a state-action and a reward to interact to an environment. The process is accomplished by maximizing Q-function as the sum of rewards. In a policy π, we define $Q_\pi(s, a)$ [13] as

$$Q_\pi(s, a) = E_\pi \left[\sum_{i=0}^{I-1} r_{t+1+i} | s_t = s, a_t = a \right], \quad (1)$$

where I is the total iterations.

Then, we obtain the optimal policy π^*, by obtaining $Q_{\pi^*}(s, a) = \max_\pi Q_\pi(s, a)$ among the plausible states and actions. The best action a_t^* in the current time is obtained through the action-replay process [14] as follows:

$$a_t^* = \arg\max_a Q_{\pi^*}(s_t, a). \quad (2)$$

B. Temporal Prediction Structure

A temporal prediction structure (TPS) is used for maximizing coding performance while yielding useful features such as temporal scalability [15]. In TPS, a video frame is involved to a group-of-picture (GOP) to determine a reference frame for a current frame. In the group, intra- or inter-coded frames can be the key frames, and non-key frames in the middle of the two key frames are coded with the key frames as references. The coding order needs to be chosen in a way that reference frames are coded before used for inter-prediction.

The key frames are more critical to coding efficiency because the reconstruction quality will affect the coding of the other frames. Thus, the key frames are coded with the highest fidelity using the lowest value of a quantization parameter (QP) to assign more bits whereas the non-key frames are coded with a larger QP. The interval between two consecutive key frames (or the size of a GOP) is not necessarily uniform for coding efficiency. For instance, in one hand, a large group size is more beneficial to encode static video scenes because there are slight motion changes between two key frames. In the other hand, a small group size is chosen for dynamic motions. The GOP sizes can be changed based upon the temporal characteristics of a video sequence.

III. Proposed Method

A. Policy representation

We choose important video intervals to include rich motion information to affect a coding of a subsequent frame, and organize the intervals in an order to improve coding efficiency.

We present an adaptive temporal segmentation method of video intervals using an RL algorithm. The goal is to maximize coding efficiency by choosing a series of the intervals and to construct an optimal HPS for coding a video. Suppose that there are L frames, and the objective is to determine the dyadic divisions of the video frames $\mathcal{V} = \{v_0, \ldots, v_{L-1}\}$ into several different interval sizes of GOPs in an HPS.

Let a Q-function $Q_\pi(v_i, v_j; a)$ denote the expected costs to encode the frames from v_{i+1} to v_{i+j-1}, and $J(v)$ is the Lagrangian cost to encode a frame v, defined as

$$J(v) = D + \lambda R, \quad (3)$$

where D is the distortion, R is the bits, and λ is the Lagrangian multiplier. In Eq.2, the best action was given by maximizing the accumulated rewards in each episode. However, as J is the cost, the optimal solution of a Q-function to encode $\mathcal{V}$ is given as,

$$\min_a \{Q_\pi(i, \frac{j}{2}; a) + Q_\pi(i + \frac{j}{2}, j; a) + J_a(i + \frac{j}{2})\}, \quad (4)$$

where we replace $\max$ operation due to the cost function. This procedure can be periodically performed to encode a whole video sequence.

B. Deep Q-Learning

The optimal action a^* is obtained by conducting an optimization as follows:

$$a^* = \arg\min_a Q_{\pi^*}(i, j; a), \quad (5)$$

where we can recursively divide the intervals using Eq.4, and the near-optimal policy can be obtained by combining the local policies in each interval. However, the computation is too complicated.

Therefore, we adopt a deep Q-learning [16] to learn the policy and obtain a sub-optimal solution. DQN employees a two-step learning procedure with two different deep neural networks. The main network predicts the current Q-value, and the target network computes the subsequent value for the next state and action. In Q-learning, once an agent starts with an initial Q-value, the value is iteratively enhanced until an episode is over as follows:

$$q = Q_{\pi_\theta} - Q_{\pi_{\theta'}}, \quad (6)$$

where Q_{π_θ} and $Q_{\pi_{\theta'}}$ are the Q-values of the main and target networks, respectively. Although the two networks have the

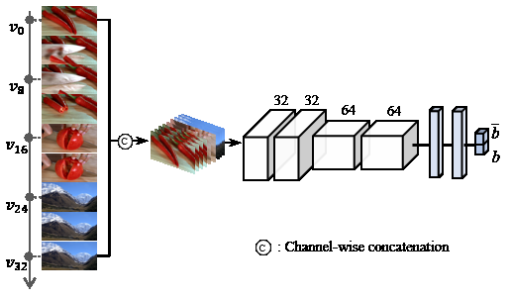

Fig. 2. The proposed network architecture using concatenated frames in a group.

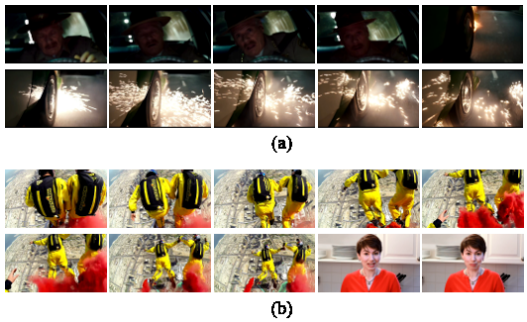

Fig. 3. Examples of YouTube training videos with static and dynamic video groups.

same architectures, the parameter of the target network is one-step ahead of the main network. The Q-network is updated with θ by minimizing q. In our implementation, we have imposed the Q-function to consider only the present cost to make a greedy decision. Furthermore, we use a replay memory [17] for complexity reduction.

C. Implementation

Fig. 2 displays a DQN motivated by [16]. We input all the frames between two key frames v_i and v_j in a group to determine if the group needs to be divided or not. The DQN extracts video features about an action and a state. Specifically, the input video frames are all concatenated and inputted to the network. The features are extracted from the convolution layers and two fully connected layers to calculate an action. For example, if $b = 1$, the current group keeps the same. Otherwise, the group is septated into two smaller ones. In the implementation, the size of the interval is a power of two as a GOP is set to 8, 16, or 32.

We implement the proposed network using Tensorflow and train the network with the Adam optimizer [18] to update the network parameters for 1000 episodes. We set ϵ = 0.01. for an ϵ-greedy policy.

IV. Experimental Results

A. Experimental Setting

We obtain high definition videos from YouTube and divide the video samples into training and testing sets. We resize the training videos to 416 $\times$ 240 and try to include more diversities regarding temporal dynamics. For this, we combine several videos to display scene changes as shown in Fig. 3. The number of training videos is around 9,000 videos. Furthermore, we also categorize the test video samples into two sets that are static videos with no scene change and dynamic videos. This categorization is used to evaluate how the proposed method perform differently with the characteristics. We also used JCT-VC test sequences for testing. The experiments are performed with a 3.60 GHz Intel CPU, 8.0 GB RAM, and NVIDIA TITAN X GPU.

B. Experimental Results

We evaluate the coding performance of the proposed method in comparison to an HEVC reference software, HM version 16.9 [19]. The reference software uses a fixed size of GOP 16. Furthermore, we compare the efficiency of the proposed method with the coding performance of fixed GOPs at sizes of 8 and 32, referred to as "fixed GOP 8" and "fixed GOP 32", respectively. The Bjontegaard-Delta rate (BD-rate) reductions are used for calculating the coding performance. We calculate the bits and PSNR for luma components.

TABLE I
BD-RATE (IN THE UNIT OF %) REDUCTION IN Y COMPONENT OF THE PROPOSED METHOD IN YOUTUBE TEST VIDEOS USING THE FIXED GOP 16 AS THE ANCHOR. FOR COMPARISONS, A FIXED GOP OF 8 AND FIXED GOP OF 32 ARE TESTED BY CHANGING THE SIZE OF THE GOP IN THE HM SOFTWARE.

Category	Sequence	Fixed GOP		Proposed method
		8	32	
Static	1	−2.1%	0.4%	0.4%
	2	0.4%	−7.6%	−7.6%
	3	0.3%	−1.1%	−1.1%
	4	−2.8%	0.2%	−2.9%
	5	14.6%	−0.6%	−0.6%
	6	1.6%	−1.9%	−1.1%
	7	0.5%	−2.6%	−2.7%
	8	−3.5%	0.3%	−3.5%
	9	−5.7%	6.3%	−5.7%
	10	6.6%	0.6%	0.0%
	11	1.9%	−6.4%	−6.4%
	12	0.1%	−3.2%	−3.2%
Average BD-rate		1.0%	−1.3%	−2.9%
Dynamic	13	−10.8%	−0.1%	−3.7%
	14	1.5%	−1.5%	−0.5%
	15	−1.0%	−1.9%	−2.2%
	16	1.6%	11.4%	−3.0%
	17	0.7%	4.6%	5.0%
	18	−8.1%	5.0%	−8.1%
	19	3.5%	−4.6%	−4.6%
	20	2.0%	18.0%	−4.1%
	21	2.6%	−0.9%	2.6%
	22	0.4%	0.3%	0.0%
	23	0.6%	15.8%	−0.3%
	24	−0.1%	5.2%	5.2%
Average BD-rate		−0.6%	4.3%	−1.1%
Total average BD-rate		0.2%	1.5%	−2.0%

Table I shows the enhanced R-D performance of the proposed method in YouTube videos. The proposed method yields better coding gains of more than 2.0% in BD-rate reductions on average. In comparisons, the fixed GOP 8 yielded slightly degraded coding performance of approximately 0.2%. The fixed GOP 32 even shows a substantial coding loss approximately 1.5 %. The proposed method provides different coding performance with various test video sequences. It is observed that the fixed GOP 32 provides coding performance approximately -1.3% in static videos whereas the coding gains are significantly degraded in dynamic videos. The fixed GOP 32 displays the opposite results. However, the proposed method provides improved coding performance both in the static and dynamic videos approximately -2.9% and -1.1%, respectively. It implies that the proposed method adapts to test video sequences, successfully and achieves better coding performance. We also provide the results in JCT-VC test sequences as shown in Table II. The proposed method provides improved coding performance of more than 1.9% in BD-rate reductions on average.

TABLE II
BD-RATE (IN THE UNIT OF %) REDUCTION IN Y COMPONENT OF THE PROPOSED TECHNIQUE IN JCT-VC SEQUENCES USING THE FIXED GOP 16 AS THE ANCHOR. FOR COMPARISONS, A FIXED GOP OF 8 AND FIXED GOP OF 32 ARE TESTED BY CHANGING THE SIZE OF THE GOP IN THE HM SOFTWARE.

Class	Fixed GOP		Proposed method
	8	32	
A1	0.8%	−1.1%	−1.1%
A2	1.3%	−1.1%	−1.1%
B	1.6%	−0.7%	−1.0%
C	2.8%	−2.6%	−2.6%
D	2.7%	−1.5%	−1.5%
E	1.9%	−4.4%	−4.4%
Average	1.9%	−1.8%	−1.9%

V. CONCLUSION

In this paper, a reinforcement learning (RL)-based coding method was proposed to divide video frames into several groups displaying similar temporal characteristics for coding efficiency. In the proposed method, the recursive problem has been solved with a sequence of a state-action for an agent. The decision make a partitioning of the current group to smaller groups. It was demonstrated with experimental results that the proposed method achieve improved coding performance.

ACKNOWLEDGMENT

This work has been supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No.NRF-2019R1C1C1010249). This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-2020-0-01460) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2020-0-00920, Development of Ultra High Resolution Unstructured Plenoptic Video Storage/Compression/Streaming Technology for Medium to Large Space)

REFERENCES

[1] B. Bross, J. Chen, J. Ohm, G. Sullivan, and Y. Wang , "Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC), " *Proceedings of the IEEE,* 2021 (Early access)

[2] J. K. Lee, N. Kim, S. Cho, and J. W. Kang, "Convolution Neural Network based Video Coding Technique using Reference Video Synthesis, " *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA),* 2018

[3] J. K. Lee, N. Kim, S. Cho, and J. W. Kang, "Deep Video Prediction Network-Based Inter-Frame Coding in HEVC," *IEEE ACCESS,* vol. 8, pp. 95906 - 95917, May 2020

[4] H. Lin, X. He, L. Qing, and S. Yang, "Improved Low-Bitrate HEVC Video Coding Using Deep Learning Based Super-Resolution and Adaptive Block Patching ," *IEEE Transactions on Multimedia,* vol. 21, pp. 3010 - 3023, May. 2019

[5] S. Ryu and J. W. Kang, "Machine Learning-Based Fast Angular Prediction Mode Decision Technique in Video Coding," *IEEE Trans. on Image Processing,* vol. 27, pp. 5525 - 5538, July 2018

[6] S. Park and J. W. Kang, "Fast Multi-type Tree Partitioning for Versatile Video Coding Using a Lightweight Neural Network," *IEEE Transactions on Multimedia,* Early Access, 2021

[7] H. Choi and I. V. Bajic, "Deep Frame Prediction for Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 30, pp. 1843 - 1855, June 2019

[8] Z. Shiping and X. Ziyao, "Spatiotemporal visual saliency guided perceptual high efficiency video coding with neural network," *Neurocomputing,* vol. 275, pp. 511 - 522, Jan. 2018

[9] Y. Gao and C. Zhu and S. Li and T. Yang, "Source Distortion Temporal Propagation Analysis for Random-Access Hierarchical Video Coding Optimization" *IEEE Transactions on Circuits and Systems for Video Technology*, 2017, 29.2: 546-559.

[10] J. Kang and Y. Lee and C. Kim and S. Lee, "Coding Order Decision of B Frames for Rate-Distortion Performance Improvement in Single-View Video and Multiview Video Coding," *IEEE transactions on Image Processing,* 2010, 19.8: 2029-2041.

[11] J. Kang and S. Cho and N. Hur and C. Kim and S. Lee, "Graph Theoretical Optimization of Prediction Structure in Multiview Video Coding," *IEEE International Conference on Image Processing,* Vol. 6. IEEE, 2007.

[12] Li, Houqiang and Li, Bin and Xu, Jizheng, "Rate-distortion optimized reference picture management for high efficiency video coding," *IEEE transactions on Circuits and Systems for Video Technology ,* 2012, 22.12: 1844-1857.

[13] Tsitsiklis, John N and Van Roy, B, "An analysis of temporal-difference learning with function approximation," *IEEE transactions on automatic control,* 1997, 42.5: 674-690.

[14] Watkins, Christopher JCH, and Peter Dayan., "Q-Learning," *Machine Learning,* 1992, 8.3-4: 279-292.

[15] Schwarz, Heiko, Detlev Marpe, and Thomas Wiegand, "Analysis of hierarchical B pictures and MCTF," *2006 IEEE International Conference on Multimedia and Expo,* IEEE, 2006. p. 1929-1932.

[16] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602,* 2013.

[17] Bellemare, M. G., Naddaf, Y., Veness, J., Bowling, M, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research* 2013, 47: 253-279.

[18] Kingma, Diederik P and Ba, Jimmy, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[19] HEVC Test Model (HM) reference software 16.9, https://hevc.hhi.fraunhofer.de/svn/svn/HEVCSoftware/tags/HM-16.9/.

[20] Chen, C., Ding, F., Zhang, D, "Perceptual hash algorithm-based adaptive GOP selection algorithm for distributed compressive video sensing," *IET Image Processing,* 2018, 12.2: 210-217.

[21] J. W. Kang, S. Ryu, N. Kim, M. Kang "Efficient Residual DPCM using an L-1 Robust Linear Prediction in Screen Content Video Coding," *IEEE Trans. on Multimedia,* vol. 18, pp. 2054 - 2065, Oct 2016