Reinforcement Learning for HEVC Screen Content Intra Coding on Heterogeneous Mobile Devices

Yuanyuan Xu(✉) and Quanping Zeng

Hohai University, Nanjing 211100, China
yuanyuan_xu@hhu.edu.cn

Abstract. Intra coding of HEVC screen content coding has to evaluate HEVC intra coding modes and additional modes for screen contents, which poses a challenge for coding such a content on mobile devices. Furthermore, the heterogeneous mobile devices have varying complexity requirements. In this paper, a flexible screen content intra coding scheme is proposed, which can trade between encoding complexity and rate-distortion performance degradation via reinforcement learning (RL). Through the design of states, actions, and more importantly, the reward function for RL, the proposed scheme can learn a flexible coding policy offline. Experimental results show the effectiveness of the proposed scheme.

Keywords: Screen content coding · Coding mode decision · Reinforcement learning

1 Introduction

New applications, such as virtual desktop, wireless displays, cloud gaming, and massive online courses, generate an increasing demand in screen sharing between mobile devices. Compared with traditional camera captured videos, screen content videos have a substantial amount of computer generated graphics and text. Several distinguished properties, such as repeated patterns, irregular motions, limited colors, are presented in screen content videos. These properties motivate the screen content coding extension (SCC) of High Efficiency Video Coding (HEVC) standard [10,13]. New coding tools such as intra-block copy (IBC) and palette (PLT) mode are developed in HEVC-SCC, which make screen content intra coding more complex than that in the computationally intensive HEVC. Coding of such contents poses a great challenge for mobile devices with varying limited computation capabilities.

Supported by National Natural Science Foundation of China under Grant No. 61801167 and Natural Science Foundation of Jiangsu Province of China under Grant No. BK20160874.

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 759–768, 2019.
https://doi.org/10.1007/978-3-030-32388-2_62

To address the complexity issue of screen content intra coding, fast intra prediction methods have been proposed in the literature [3,4,7–9,14]. In [8], coding units (CUs) are classified into natural content ones and screen content ones based on the statistical information, where early termination of splitting operations are performed accordingly. In [9], neighboring luminance gradient information and coding bits are exploited to perform early skipping of depth decision and mode prediction. Besides exploiting observed statistical information, machine learning techniques can be utilized to design fast screen content coding schemes. In [3], texture information of current CU and sub-CUs is utilized by neural network to guide coding unit (CU) partition, while decision trees are used to determine whether a CU is a natural image block or a screen content block, needs partitioning or not, and selects directional or non-directional modes in [4]. In [14], two classifiers are designed to determined whether the current CU is split into sub-CUs and whether SCC modes or traditional intra modes are performed for the unsplit CU, where texture information of current CU, coding information of current and neighboring CUs are used as features. In [7], dynamic and static information of current CU is utilized by decision trees to check either IBC or PLT mode for screen content blocks.

In these existing works, the amount of complexity reduction is fixed for a given screen content video, which cannot accommodate varying requirements of heterogeneous mobile devices with different computing capabilities, e.g., mobile phones, wireless head mounted displays (HMDs). In this paper, we propose a flexible screen content intra coding scheme which can adjust between encoding complexity reduction and rate-distortion performance degradation via reinforcement learning (RL). RL tries to learn a policy which maximizes the total rewards depending on inter-correlated decisions. It has been used in video coding for video encoder control [5], rate control [6] and unit split decision [2]. As far as we know, none of the existing works on screen content coding uses RL. The flexible screen content intra coding which selectively searches through different modes according to the capability of device is modeled as a RL problem. Motivated by the work in [5], the trade-off between complexity and rate-distortion performance of screen content intra coding is represented by a reward function in RL.

The rest of the paper is organized as follows. In Sect. 2, we provide preliminary information of screen content intra coding. The proposed flexible reinforcement learning based screen content intra coding scheme is presented in Sect. 3. Section 4 shows the experimental results, while Sect. 5 concludes the paper.

2 Preliminary

Prior to the HEVC, the H.264/MPEG-4 AVC standard [12] supports nine Intra_4 $\times$ 4, four Intra_16 $\times$ 16 and I_PCM prediction modes for traditional 16×16 luma samples. The coding structure of the HEVC is more complicated than that in H.264/MPEG-4 AVC. In the HEVC [11], variable-size coding tree units (CTUs) are supported, where the size of luma coding tree blocks (CTBs) may be equal to 16×16, 32×32, and 64×64. Each CTB can be used as a coding block

(CB) or further split into multiple CBs recursively using the quadtree syntax until a minimum allowed luma CB size is reached. The prediction block (PB) for intrapicture prediction is the same as the CB, except CBs with the smallest size which can be further split into four PBs. For the transform blocks (TBs), a luma CB can be further partitioned into multiple square TBs recursively, where the maximum depth of the residual quadtree is constrained and indicated in sequence parameter set (SPS). Figure 1 shows an example of CTB partition using the quadtree syntax. For each CB, 33 different directional modes, a planar prediction and a DC prediction mode are defined for intrapicture prediction, where the neighboring TBs are used to form the reconstruction signal.

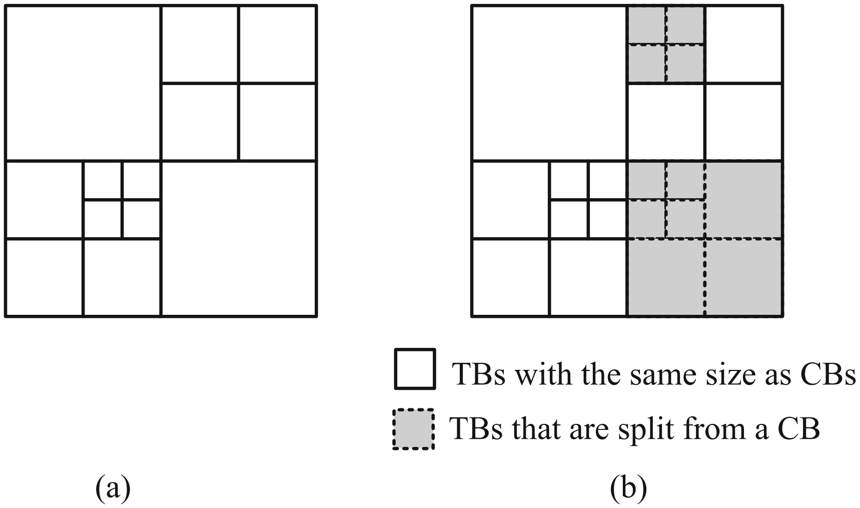

Fig. 1. An example of coding tree block partition using the quadtree syntax (a) Code blocks. (b) Transform blocks.

Besides the above-mentioned coding framework of HEVC, the SCC extension of HEVC introduces new coding tools, including IBC and PLT modes. IBC is a new coding mode for CUs with repeated patterns, which uses similar reconstructed blocks in the same picture as a prediction signal. PLT mode is designed for blocks with limited colors, which lists all the color values and sends an index of color for each sample instead of coding each sample. For each CU, mode decision of intra coding is determined by exhaustive search, and it is implemented in the HEVC-SCC reference software as follow [7]. The IBC predictor is performed first which uses a few options of block vector (BV) from most recently coded CUs and neighboring CUs in the IBC mode. Then the intra coding modes of HEVC are evaluated, followed by examining IBC merge and skip mode. The IBC merge and skip mode is similar as those for interpicture prediction. Only a skip flag and the merge index are sent in the skip mode, while the merge mode allows residual coding. If the IBC skip mode is not the best mode so far, the IBC search is conducted. At last, the PLT mode is evaluated. Among all those modes, the coding mode with the smallest cost, $D + \lambda R$, is selected, where R, D, and λ are the coding bits, the distortion, and a Lagrange multiplier, respectively. To further complicate the intra coding procedure, the final partition of CTU into CUs is determined by evaluating all the possible partitions and choosing the one with the smallest cost. For each partition, intra coding modes for all of its CUs have to be decided as mentioned above. Machine learning methods can be used to develop fast intra coding scheme.

3 Reinforcement Learning Based Screen Content Intra Coding

In this section, a flexible screen content intra coding scheme is proposed using a RL approach. In the following, the framework of the proposed scheme is presented, followed by the design of feature selection and reward function for RL.

3.1 Framework

Since we want to take into account the cost of coding mode selection errors when applying the coding strategy, RL is utilized which considers the classification error in the reward function. The framework of the proposed RL based intra coding scheme is presented in Fig. 2. In this framework, a mobile device passes a trade-off coefficient between rate-distortion performance and complexity, μ, to the RL module. For a given μ, a coding policy is learned offline via RL by using the training set of screen content videos. The mobile device can use the learned coding policy as a static part of the coding to speed its intra coding mode decision procedure. Depending on the requirements of the mobile devices, different adjustment factors μ can be used in the RL module to make a flexible trade-off between coding efficiency and complexity. A mobile device with less computational resources passes a larger value of μ, while a smaller μ is associated with abundant computational resources. A learnt coding policy can be used for all the mobile devices with the same type.

3.2 Coding Policy Learning via RL

The proposed fast scheme tries to learn a coding policy that reduces the number of evaluated coding modes according to observed information of a CB. This coding policy is learned through RL module. In the RL module, the learning agent interacts with the learning environment (coding using screen content videos training set) repeatedly. The intra coding process can be seen as a series of coding decision episodes that repeatedly evaluating selected intra coding modes. At a time point t, the learning agent selects an action from the set of available actions (evaluating selected coding modes) to act on the learning environment

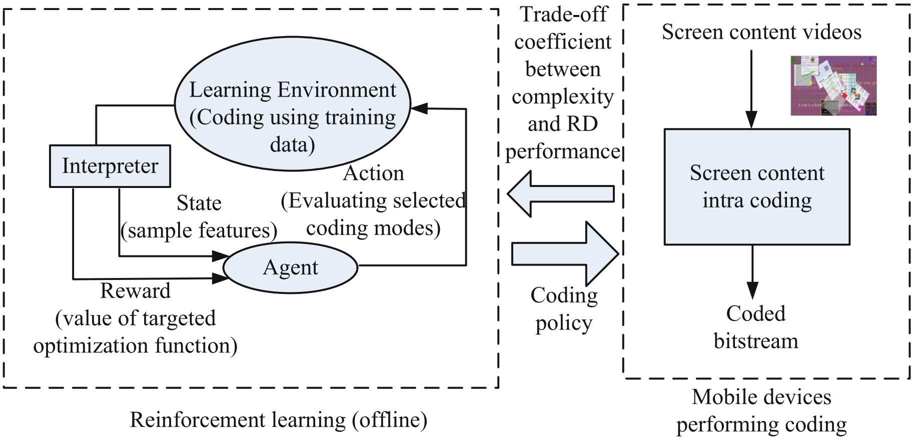
Learning Environment (Coding using training data)
Interpreter
State (sample features)
Action (Evaluating selected coding modes)
Agent
Reward (value of targeted optimization function)
Reinforcement learning (offline)
Trade-off coefficient between complexity and RD performance
Coding policy
Screen content videos
Screen content intra coding
Coded bitstream
Mobile devices performing coding

Fig. 2. Framework of the proposed reinforcement learning based intra coding scheme.

based on the environmental state information (sample features) s_t. After the action is executed, the interpreter feeds back information about the new state $s_{(t+1)}$ of the environment and reward $r_{(t+1)}$ (value of targeted optimization function) associated with the performed action. In the following, we will present the design of features, actions, and reward function in the RL module.

About the actions in the RL, all the coding modes associated with HEVC-SCC are roughly divided into three categories, which are HEVC coding modes, IBC mode, and PLT mode. Correspondingly, three actions are allowed for coding mode evaluations, which correspond to evaluating the HEVC intra mode, IBC mode, and PLT mode, respectively. Note that the IBC mode includes its predictor mode, merge and skip mode, and IBC search mode. Although only three actions are defined in this paper, the proposed work can be extended to the case with more actions.

About the feature design, we uses statistical information of a CB according to the allowed actions. For screen content coding, CBs with limited number of colors and the coding unit with sharp boundaries are usually coded with the PLT mode. The area of screen contents where the hue is discontinuous is usually encoded using IBC or palette mode. A uniform region usually uses an intra coding mode. Therefore, the following features of a CB are used: variance, the number of colors, the largest number of pixels with the same value, the maximum run length of pixel values horizontally, and the maximum run length of pixel values vertically.

The trade-off between rate-distortion performance and coding complexity of intra coding is achieved by designing a reward function for RL. The goal of RL is to maximize the expected reward in the future real coding process. The learning algorithm of this scheme estimates the reward through experiments on a set of N training samples $\sum_i r_i$, where r_i is the reward for CB i. The reward of performing one of the three actions for CB i can be defined as follows

$$r_i = -(c_i - c_{i,min})/c_{i,min} + \mu(t_{i,sum} - t_i)/t_{i,sum}, \tag{1}$$

where r_i, c_i, t_i are the reward of performing action a_i for the CB i, the minimum coding cost $(D + \lambda R)$ of coding modes associated with a_i, and the total time expense of evaluating coding modes associated with a_i, respectively. $t_{i,sum}$ is the consumed time in evaluating all of traditional intra modes, IBC mode, and PLT mode for the CB i, while $c_{i,min}$ is the cost of the best mode in terms of rate-distortion performance for the CB i. The reward of a coding strategy on the i-th training sample consists of two parts: the rate-distortion cost reduction and the coding complexity reduction. μ ($\mu > 0$) is the weight of the encoding complexity. The larger the weight is, the more the encoder limits the computational complexity. By adjusting this weight, trade-off between the coding efficiency and complexity can be flexibly adjusted to suit the needs of different applications. For, example, smartphone-based HMD devices should use a larger weight than computer-based HMD devices.

3.3 Coding Policy Learning Algorithm

With the above design of features, actions, and reward function, we use Q-learning to learning coding policy. Due to the aim of RL is to speed intra coding of screen content, a simple ternary classifier is used to represent the relationship between the value of features and selected action. The input layer consist of 6 nodes, while the output layer consists of 3 nodes. The three outputs correspond to three allowed actions. The classifier is configured by θ. The coding policy learning via RL can be summarized in Algorithm 1.

Algorithm 1. coding policy learning via RL

```
Initialize the classifier parameter $\theta$
Initialize the learning parameter $\gamma = 0.9$
for samples $i = 0 \rightarrow N - 1$ do
    Calculate the values of features, $s_i$
    Choose the action with the maximum value $a_i = argmax_{a_i} Q(s_i, a_i; \theta)$
    Execute action $a_i$, observe the reward $r_i$
    Update $\theta$ with the new $Q'(s_i, a_i; \theta) = Q(s_i, a_i; \theta) + \gamma(r_i - Q(s_i, a_i; \theta))$
    Decrease $\gamma$
end for
```

After the coding policy, i.e., fixed parameter θ for classifier, is learned, it is sent to the mobile device and implemented as static part for coding on such a device. During the intra coding procedure on a mobile device, only the coding modes associated with the following action for each CU are evaluated.

$$a_i = argmax_{a_i} Q(s_i, a_i; \theta) \tag{2}$$

4 Experimental Results

The experimental results are obtained implementing the proposed method in the HEVC-SCC reference software HM-16.18 SCM 8.7. The all intra (AI) configuration is used. The coding performance is compared with the anchor that exhaustively searches through all the coding options in SCM 8.7. The video sequences used for coding policy learning are listed in Table 1, where TGM, M, and CC represent text and graphics with motion, mixed content, and camera-captured content, respectively.

Table 1. Training video sequences

Resolution	Sequence name	Category
1920 × 1080	sc_FlyingGraphics_1920 × 1080_60_8bit	TGM
1280 × 720	sc_Programming_1280 × 720_60_8bit	TGM
1280 × 720	sc_SlideEditing_1280 × 720_30_8bit_420	TGM
832 × 480	BasketballDrillText_832 × 480_50	M
1280 × 720	KristenAndSara_1280 × 720_60	CC
416 × 240	BlowingBubbles_416 × 240_50	CC

In the RL module, the training data for neural network are obtained in coding training video sequences using HM-16.18 with SCM 8.7. Specifically, for each CB, the rate-distortion costs and the consumed time measured in microseconds are collected for the cases of performing IBC predictor, HEVC intra, IBC merge and skip, IBC search, and PLT modes. Note that the time complexity of IBC mode is the sum of those performing IBC predictor, IBC merge and skip, and IBC search, while the rate-distortion cost of IBC mode is the minimum cost associated with the above options. A subset using the a cropped window on the first frame of these sequences are used as training data. 8000 training samples are generated and randomized. The coefficients are learned using varying training steps with $\mu = 0.5$.

Table 2. Test video sequences

Resolution	Sequence name	Category
1920 × 1080	sc_desktop_1920 × 1080_60_8bit	TGM
1920 × 1080	MissionControlCllip3_1920 × 1080_60p_8b444	M
1280 × 720	sc_web_browsing_1280 × 720_30_8bit_420_r1	TGM
1280 × 720	sc_SlideShow_1280 × 720_20_8bit	TGM
1920 × 1080	Kimono1_1_1920 × 1080_24_10bits	CC

After the coding policies are learned, they are implemented in the intra coding of testing video sequences as listed in Table 2. Performance of the proposed scheme is compared with the benchmark of HM-16.18 SCM 8.7 for testing video sequences. Bjøntegaard delta rate (BD-rate) [1] is used to measure the rate-distortion performance degradation, in terms of the percentage of bitrate saving (negative values) or increasing (positive values). The coding complexity is measured by the percentage of encoding time saving. The comparison results using different video sequences are listed in Table 3. Among the screen content video sequences, the performance of the proposed scheme is better for the "WebBrowsing" than the other sequences, because most of training sequence are 4:2:0 sequences whose color format is the same as the one for "WebBrowsing" sequence. In our experiment, we also found that mode selection among traditional Intra modes, IBC mode, and PLT mode hardly affects the rate-distortion performance of camera captured sequences. Therefore, the camera captured "Kimino" sequence achieves almost 31% reduction in encoding time with only a slight BD-rate increase of 0.1%. The coding time comparison with varying QP values is shown in Table 4. It can be seen from the table that the proposed scheme can achieve up to 31.5% savings in coding complexity reduction for a fixed QP value. Performance of the proposed scheme gets better as the value of QP gets smaller.

Table 3. Coding performance compared with HM-16.18 SCM 8.7

Sequence	BD-rate	Encoding time
Desktop	12.1%	−17.7%
MissionControlClip3	10.1%	−19.0%
WebBrowsing	2.8%	−18.8%
SlideShow	14.1%	−7.0%
Kimino	0.1%	−31.0%
Average	7.8%	−18.7%

Table 4. Coding time comparison with HM-16.18 SCM 8.7 using varying QP

QP	Encoding time
37	−12.9%
32	−16.1%
27	−21.6%
22	−31.5%
Average	−20.5%

5 Conclusion

In this paper, a flexible screen content intra coding scheme is proposed to address the varying complexity requirements of heterogeneous mobile devices. In this scheme, a coding policy can be learned for a targeted type of devices through RL offline. The trade-off between encoding complexity and rate-distortion performance degradation is controlled by designing a reward function for RL. The learned coding policy is then utilized as a static part of coding at mobile devices to speed the intra coding of screen contents. The effectiveness of the proposed scheme is verified by the experimental results.

References

1. Bjøntegaard, G.: Calculation of average PSNR differences between RD-curves. In: Proceedings of the ITU-T Video Coding Experts Group (VCEG) Thirteenth Meeting, January 2001
2. Chung, C.H., Peng, W.H., Hu, J.H.: HEVC/H.265 coding unit split decision using deep reinforcement learning. In: Proceedings of 2017 IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp. 570–575 (2017)
3. Duanmu, F., Ma, Z., Wang, Y.: Fast CU partition decision using machine learning for screen content compression. In: Proceedings of 2015 IEEE International Conference on Image Processing (ICIP), pp. 4972–4976, September 2015
4. Duanmu, F., Ma, Z., Wang, Y.: Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension. IEEE J. Emerg. Sel. Topics Circuits Syst. **6**(4), 517–531 (2016)
5. Helle, P., Schwarz, H., Wiegand, T., Müller, K.R.: Reinforcement learning for video encoder control in HEVC. In: Proceedings of 2017 IEEE International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 1–5 (2017)
6. Hu, J.H., Peng, W.H., Chung, C.H.: Reinforcement learning for HEVC/H.265 intra-frame rate control. In: Proceedings of 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2018)
7. Kuang, W., Chan, Y., Tsang, S., Siu, W.: Machine learning based fast intra mode decision for HEVC screen content coding via decision trees. IEEE Trans. Circuits Syst. Video Technol. 1 (2019, early access)
8. Lei, J., Li, D., Pan, Z., Sun, Z., Kwong, S., Hou, C.: Fast intra prediction based on content property analysis for low complexity HEVC-based screen content coding. IEEE Trans. Broadcast. **63**(1), 48–58 (2017)
9. Lu, Y., Liu, H., Lin, Y., Shen, L., Yin, H.: Efficient coding mode and partition decision for screen content intra coding. Sig. Process. Image Commun. **68**, 249–257 (2018)
10. Peng, W.H., et al.: Overview of screen content video coding: technologies, standards, and beyond. IEEE J. Emerg. Sel. Topics Circuits Syst. **6**(4), 393–408 (2016)
11. Sullivan, G.J., Ohm, J., Han, W., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
12. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 560–576 (2003)

13. Xu, J., Joshi, R., Cohen, R.A.: Overview of the emerging HEVC screen content coding extension. IEEE Trans. Circuits Syst. Video Technol. **26**(1), 50–62 (2016)
14. Yang, H., Shen, L., An, P.: Efficient screen content intra coding based on statistical learning. Sig. Process. Image Commun. **62**, 74–81 (2018)