Fast Inter Prediction Mode Decision Algorithm Based on Data Mining

Tengrui Shi[1,2], Xiaobo Guo[2], Daihui Mo[3,4], and Jian Wang[1](✉)

[1] Nanjing University, NJU,
Nanjing 210023, Jiangsu, People's Republic of China
wangjnju@nju.edu.cn
[2] Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory, The 54th Institute of CETC, Shijiazhuang 050081, Hebei, People's Republic of China
[3] Department of Electronic Engineering, Tsinghua University, Beijing 100084, People's Republic of China
[4] Academy of Military Sciences PLA China, Beijing 100091, People's Republic of China

Abstract. The HEVC greatly improves coding efficiency. However, this is accompanied by an increase in the complexity of the coding calculation, which is higher than H.264. We find that there are several features that are highly correlated with the CU's best split decision in inter prediction. As a result, we choose decision trees to solve the splitting decision problem. We implement the decision trees on official software HM16.2 and test the algorithm on the testing set. Experiments indicate that the fast decision algorithm improve the coding performance more efficiently than some existing algorithms.

Keywords: HEVC · Inter prediction · Data mining · Decision trees

1 Introduction

The performance of HEVC in many aspects is better than previous standards with more flexible data structures and other new technologies [1]. The improved intra prediction and inter prediction technology has greatly improved accuracy in sample prediction and so on. Nevertheless, these improvements result in a significant increase in coding computational complexity [2].

For each frame of the input encoder, it is divided into some block CTUs (Coding Tree Units). The coding trees are used to divide CTUs into multiple CUs (coding units). CU is the leaf node of a coding tree, and a CU can contain one or more PUs(prediction unit). There are nine division modes in the inter mode, including three square shapes (MSM, 2Nx2N, NN), two rectangular shapes (2NN, N2N), and four asymmetrical shapes (2NnU, 2N nD, nL2N, nR2N) which are presented in Fig. 1.

L. Meng and Y. Zhang (Eds.): MLICOM 2018, LNICST 251, pp. 95–102, 2018.
https://doi.org/10.1007/978-3-030-00557-3_10

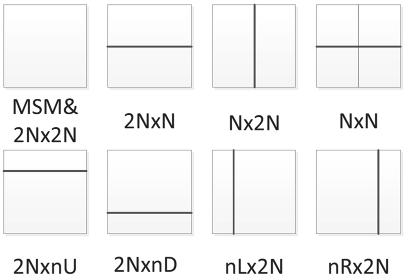

Fig. 1. PU partition modes

In this paper, we make use of a few new features and Correa's original features to establish four decision trees which are aimed to determine whether CU is segmented into smaller PUs in inter prediction. The reminder is arranged as follows: Sect. 2 introduces our motivation and overview of related work. Section 3 presents the fast decision algorithm with decision trees. Section 4 shows the experimental results. Finally, we do a summary of the full text in Sect. 5.

2 Motivation and Related Work

Numerous papers have studied the algorithms on reduction of computation complexity of HEVC encoders. By using top omitting and bottom pruning, Guo [3] present an algorithm based on subtree distribution. Xiong [4] proposed an algorithm based on SAD estimation. Zhong [5] proposed an algorithm about CU segmentation between adjacent frames. However, all these works bring about losses related to R-D efficiency, and the losses can not be ignored.

All prediction modes are performed in the encoder, and eventually the division between the modes is not equal. Figure 2 is a graph of the probability distribution of each mode of the CU. The figure indicates that most of 8×8 and 16×16 CUs are encoded as MSM. All modes are tested, which wastes a lot of time. Nevertheless, the increase of R-D costs is great, when we directly delete other modes from Table 3 in Sect. 4. If we can precisely predict the split situation of CU, once the prediction results that the current CU has to be encoded as only one PU, the remaining partition modes can be directly ignored without being tested. This will effectively reduce the coding time for inter prediction without significantly affecting coding efficiency.

Related research was proposed by Correa [8,9] and Li [10], which used machine learning to reduce the computational the coding complexity. They conducted some data analysis and selected some of the features associated with CU splitting. Finally, they used machine learning as a tool to leverage these features to build decision trees which could predict whether each CU would be split into smaller PUs. This algorithm selects some more comprehensive features, reducing the computational complexity while maintaining a low loss of R-D efficiency.

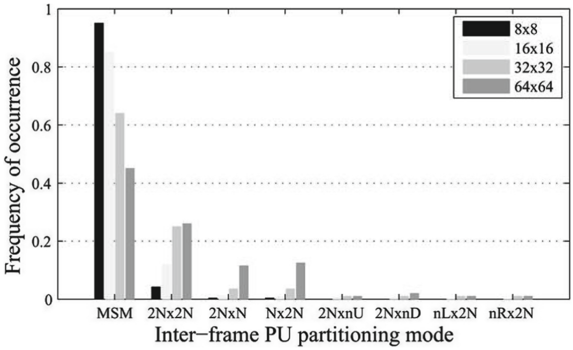

Fig. 2. Partition modes in inter prediction

3 Fast Algorithm Based on Data Mining

3.1 Data Analysis and Optimization

Data Mining is a procedure of analyzing multiple data, and gathering it into valuable information and modes. Supervised learning is one class of data mining [11], which is in connection with the involving algorithm in this paper. Decision trees [12] are models built through predictive Supervised Learning.Decision trees are built by using C4.5 algorithm [13] to obtain encoding optimization.

The encoding features are revealed in this subsection by presenting a series of statistics, and we optimized these features to obtain four accurate decision trees.

In order to obtain features that contribute to CU splitting decisions, many attributes are listed below. In [8,9], Correa thinks that some features (2Nx2N and MSM mode RD cost, 2Nx2N and MSM mode residue cost, the RD cost ratio between 2Nx2N and MSM mode, and splitting decision in CU of previous tree depth) are highly correlated with the split situation. Besides those mentioned by Correa, we found that the lower of MSM mode RD-cost and the lower of 2Nx2N mode RD-cost, the MV of 2Nx2N PU mode and the MV of MSM PU mode are highly correlated with the best PU partition decision.

We use the relevant features in 1616 CU of the FourPeople sequence as an example for analysis. As shown in Fig. 3, whether it is the MSM mode or the 2Nx2N mode, the RD cost of CU that is not divided into multiple PUs is much smaller than which is divided into multiple PUs. It is undeniable that the range of the rate distortion value is closely related to the features of test sequences such as resolution, texture information, motion information and so on. Therefore, we need to normalize these features. We can find that when the ratio is smaller, the likelihood of no-splitting is higher in Fig. 4 which is similar to other size CUs.

We found MV plays an important role in the PU mode selection. To simplify the calculation of the absolute MV, we only use the absolute values of the x and y MV directions, as shown in (1). MV_x means the horizontal value of MV, and MV_y

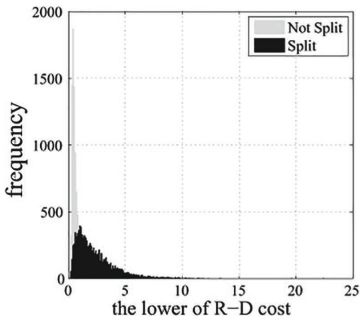

Fig. 3. The lower of RD cost

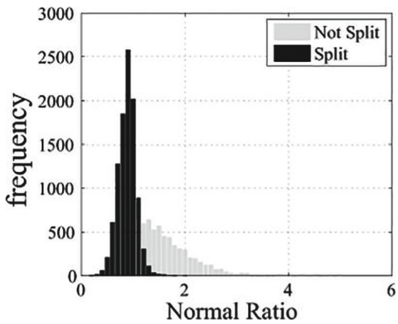

Fig. 4. Normal ratio

means the vertical value of MV, and we need to normalize these features. From Figs. 5 and 6, it is noticed to the normalized MV_{abs} present a high correlation with CU-splitting. At last, we decide to combine the features of MV and the lower of normalized MSM mode and normalized 2Nx2N mode RD-cost in our algorithm.

$$MV_{abs} = |MV_x| + |MV_y| \tag{1}$$

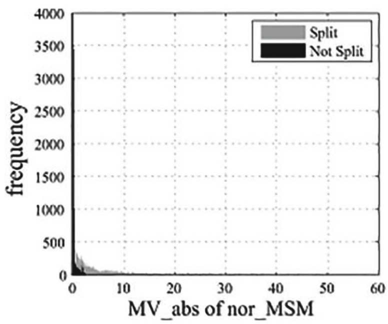

Fig. 5. CU not splitting and MV of normal MSM

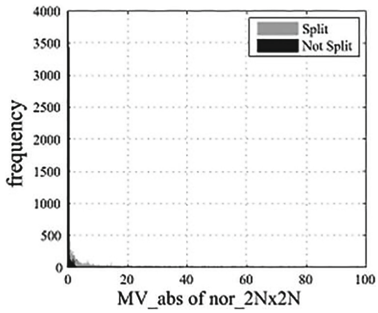

Fig. 6. CU not splitting and MV of normal 2Nx2N

3.2 Implementation of Decision Trees

The features chosen by previous analysis are as follows abs_2Nx2N_var (absolute 2Nx2N residue variance), nor_2Nx2N_var (normalized 2Nx2N residue variance), abs_mv_MSM (absolute MSM MV_{abs}), nor_mv_MSM (normalized MSM MV_{abs}), abs_mv_2Nx2N (absolute 2Nx2N MV_{abs}), nor_mv_2Nx2N (normalized 2Nx2N MV_{abs}) and Nei_Depth (the CU depth).

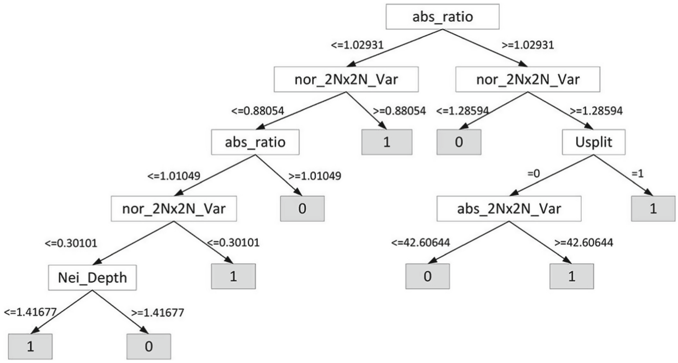

Fig. 7. 32×32 decision tree

Four different decision trees are built from four kinds of different sizes CUs in inter prediction. For fairness [14], we randomly select a data set, half of which consists of splitting CU and the other consists of no-splitting CU from the training sequence data set in Table 2. We use WEKA [15] to build decision trees. Figure 7 displays the obtained 32×32 decision tree. We have a test on the accuracy of the decision trees to evaluate these trees. From Table 1, we find they have high accuracy, but the decision tree of 64×64 is relatively less accurate than other trees because 64×64 decision tree has no Usplit attribute. The decision trees are within six layers with low complexity, so computational complexity can be hardly increased when we apply the decision trees into HM encoder.

Table 1. The accuracy of my decision trees

CU size	Accuracy	Node num	Leave num	Layer
64×64	76.96%	15	8	5
32×32	81.33%	17	9	6
16×16	85.47%	19	10	6
8×8	89.35%	11	6	5

4 Experimental Results and Analysis

4.1 Experimental Environment

On the latest version of the official reference software HM16.2, we applied this decision algorithm. In the following, we list computer parameters: CPU of i7-7700, frequency of 3.6 GHz and system of win10.

The video coding standard gives us a series of standard video sequences. These video test sequences are divided into two parts: train set (BlowingBubbles, RaceHorses, BQMall, SlideShow, Johnny, BasketballDrive, ParkScene, Traffic) and test set (BQSquare, BasketballPass, BasketballDrill, fourpeople, SlideEditing, Catus, BQTerrace, PeopleOnStreet). In experiments, we use low delay configuration. Each test sequence selected four QPs (22, 27, 32 and 37) to encode them respectively, so that we can get average data among the four conditions for every test sequence.

In order to evaluate this algorithm proposed, we compare three encoder versions: the original HM16.2, the simple HM16.2 and the modified HM16.2 with only MSM and 2Nx2N modes enabled. Finally we used CCR and BD-rate to make a comparison among these algorithms.

4.2 Experimental Results

The results of 8 sequences encoded with the simple encoders are shown in Table 2. The results of 8 sequences encoded with our proposed encoders are presented in Table 3.

From Table 2, the simple encoder increases average BD-rate by 4.08%, and reduces encoding time by 55.94% in contrast to the original HM 16.2 software model. From Table 3, our proposed algorithm increases average BD-rate by 0.25% and reduces encoding time by 30.18% compared to the original HM 16.2 software model. Average BD-rate of our proposed is 16.32 times smaller than simple algorithm. The BD-rate/CCR of the proposed encoder is 7.3 times smaller than the simple. The BD-PSNR/CCR is 10 times smaller than the simple, which means that our proposed algorithm reduces complexity efficiently with negligible loss.

Table 2. The simple method

Sequence	CCR(%)	BD-rate(%)	BD-PSNR(%)	BD-rate/CCR(%)	BD-PSNR/CCR(%)
BasketballPass	56.35	7.64	−0.33	13.56	−0.59
BQSqure	55.615	5.99	−0.22	10.77	−0.40
BasketballDrill	52.3	2.16	−0.081	1.132	−0.15
FourPeople	56.85	2.699	−0.081	4.132	−0.15
SlideEditing	57.065	2.18	−0.31	3.82	−0.68
Cactus	56.6	2.96	−0.06	5.23	−0.11
BQTerrace	57.02	3.33	−0.056	5.84	−0.1
PeopleOnstreet	57.34	3.23	−0.14	5.63	−0.24
Average	55.94	4.08	−0.18	7.3	−0.33

Table 3. Our proposed method

Sequence	CCR(%)	BD-rate(%)	BD-PSNR(%)	BD-rate/CCR(%)	BD-PSNR/CCR(%)
BasketballPass	31.50	0.40	−0.016	1.3	−0.051
BQSqure	20.45	0.21	−0.0090	1.0	−0.044
BasketballDrill	28.15	0.22	−0.0081	0.76	−0.029
FourPeople	38.13	0.22	−0.0063	0.57	−0.017
SlideEditing	45.02	−0.053	0.0060	−0.12	0.013
Cactus	28.05	0.29	−0.0057	1.0	−0.020
BQTerrace	26.55	0.23	−0.0043	0.85	−0.016
PeopleOnstreet	23.61	0.53	−0.024	2.2	−0.10
Average	30.18	0.25	−0.0084	1.0	−0.033

5 Conclusion

In this paper, we introduce Data Mining briefly and regard CU-splitting problem as classification problem. Then we decide to use decision trees to solve the classification problem. Finally, the algorithm is proposed in HM16.2 and we perform experiments related to the test sequences. Based on the above experimental results, we find that this fast decision algorithm can effectively shorten the encoding time and has little effect on the encoding performance.

References

1. Ohm, J., Sullivan, G.J., Schwarz, H., Tan, T.K., Wiegand, T.: Comparison of the coding efficiency of video coding standards including high efficiency video coding (HEVC). IEEE Trans. Circ. Syst. Video Technol. **22**, 1669–1684 (2012)
2. Bossen, F., Bross, B., Suhring, K., Flynn, D.: HEVC complexity and implementation analysis. IEEE Trans. Circ. Syst. Video Technol. **22**, 1685–1696 (2012)
3. Guo, L., Zhou, L., Tian, X., Chen, Y.: Adaptive coding-unit size selection based on hierarchical quad-tree correlations for high-efficiency video coding. J. Electron. Imaging **24**, 023036–023036 (2015)
4. Xiong, J., Li, H., Meng, F., Wu, Q., Ngan, K.N.: Fast HEVC inter CU decision based on latent SAD estimation. IEEE Trans. Multimed. **17**, 2147–2159 (2015)
5. Zhong, G.Y., He, X.H., Qing, L.B., Li, Y.: Fast inter-mode decision algorithm for high-efficiency video coding based on similarity of coding unit segmentation and partition mode between two temporally adjacent frames. J. Electron. Imaging **22**, 381–388 (2013)
6. Fernández, G., Cuenca, P., Barbosa, L.O., Kalva, H.: Very low complexity MPEG-2 to H.264 transcoding using machine learning. In: Proceedings of the 14th Annual ACM International Conference on Multimedia, pp. 931–940 (2006)
7. Van, L.P., et al.: Fast transrating for high efficiency video coding based on machine learning. In: 2013 20th IEEE International Conference on Image Processing (ICIP), pp. 1573–1577 (2013)

8. Correa, G., Assuncao, P., Agostini, L., da Silva Cruz, L.A.: A method for early-splitting of HEVC inter blocks based on decision trees. 2014 Proceedings of the 22nd European Signal Processing Conference (EUSIPCO), pp. 276–280 (2014)
9. Correa, G., Assuncao, P.A., Agostini, L.V., da Silva Cruz, L.A.: Fast HEVC encoding decisions using data mining. IEEE Trans. Circuits Syst. Video Technol. **25**, 660–673 (2015)
10. Li, K., Wang, J.: Fast CU-splitting decisions based on data mining. In: IEEE International Conference on Consumer Electronics-China, pp. 1–5 (2017)
11. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Mag. **17**, 37 (1996)
12. Shan, S.: Decision tree learning. In: Shan, S. (ed.) Machine Learning Models and Algorithms for Big Data Classification, vol. 36, pp. 237–269. Springer, Boston (2016). https://doi.org/10.1007/978-1-4899-7641-3_10
13. Quinlan, J.R.: C4. 5: Programs for Machine Learning. Morgan Kaufmann, Los Altos (1993)
14. Orriols-Puig, A., Bernadó-Mansilla, E.: The Class imbalance problem in UCS classifier system: a preliminary study. In: Kovacs, T., Llorà, X., Takadama, K., Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) IWLCS 2003–2005. LNCS (LNAI), vol. 4399, pp. 161–180. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71231-2_12
15. Russell, I., Markov, Z.: An introduction to the WEKA data mining system. In: ACM SIGCSE Technical Symposium on Computer Science Education, pp. 742–742 (2017)