Fast CU Partition Decision Using Machine Learning for Screen Content Compression

Fanyi Duanmu*, Zhan Ma+ and Yao Wang*
* New York University, Brooklyn, NY 11201, USA
+ Nanjing University, 22 Hankou Road, Nanjing 210093, P.R. China
(fanyi.duanmu@nyu.edu, mazhan@nju.edu.cn, yw523@nyu.edu)

ABSTRACT

Screen Content Coding (SCC) extension is currently being developed by Joint Collaborative Team on Video Coding (JCT-VC), as the final extension for the latest High-Efficiency Video Coding (HEVC) standard. It employs some new coding tools and algorithms (including palette coding mode, intra block copy mode, adaptive color transform, adaptive motion compensation precision, etc.), and outperforms HEVC by over 40% bitrate reduction on typical screen contents. However, enormous computational complexity is introduced on encoder primarily due to heavy optimization processing, especially rate distortion optimization (RDO) for Coding Unit (CU) partition decision and mode selection. This paper proposes a novel machine learning based approach for fast CU partition decision using features that describe CU statistics and sub-CU homogeneity. The proposed scheme is implemented as a "pre-processing" module on top of the Screen Content Coding reference software (SCM-3.0). Compared with SCM-3.0, experimental results show that our scheme can achieve 36.8% complexity reduction on average with only 3.0% BD-rate increase over 11 JCT-VC testing sequences when encoded using "All Intra" (AI) configuration.

Index Terms — Screen Content Coding, HEVC, Machine Learning, Neural Network, Partition Decision

I. INTRODUCTION

Cloud based applications, such as virtual desktop interface, online gaming, shared screen collaboration, distance education, etc., have drawn more and more attention. Correspondingly, new opportunities and challenges are introduced on how to efficiently compress high-definition computer-generated screen contents for limited bandwidth and computing resources. To better understand the natural statistics of screen contents and develop potential compression technologies, JCT-VC launched the development of screen content extension on top of HEVC in early 2014. New promising coding techniques (e.g., palette coding mode, intra block copy mode, adaptive color transform, etc.) were identified in the previous JCT-VC meetings and have been adopted in the official test model SCM-3.0 [1].

However, the intrinsic flexibility of HEVC partitioning scheme and the additional SCC tools impose significant computational burden on encoder, primarily during the seeking of optimal combinations of CU partitions and CU modes, as summarized into the following two fundamental problems:

Mode Decision: given current CU, which mode (among 35 intra modes, palette mode and intra block copy mode) should be chosen to minimize Rate-Distortion (RD) cost?

Partition Decision: given current CU, should it be further partitioned into smaller sub-CUs? Or equivalently, whether current CU optimal RD cost will exceed the combined RD cost of sub-CUs (each using optimal mode and partition)?

In SCM-3.0, an exhaustive search method is employed to solve these two problems by examing all possible modes for current CU and comparing RD cost of current CU using the best mode and the sum of RD costs from its sub-CUs, each using best mode and partition. However, for applications in real-time, such exhaustive mode/partition search strategy is never practical. Thus, it is important to develop fast algorithms for CU partition and mode decisions while maintaining the coding efficiency. There are many prior works on HEVC encoder acceleration. These papers can be roughly categorized into the following classes:

Class 1: Mode Reduction. A fast intra mode decision algorithm was proposed [2], which exploits the directional information of neighboring blocks to reduce the number of intra directional predictions to be considered in rate distortion optimization (RDO). Another gradient-based approach [3] uses CU direction histogram analysis to reduce the pool size of intra mode candidates before RDO calculation.

Class 2: Cost Replacement. An entropy-based fast Large Coding Unit (LCU) partition algorithm [4] was proposed, which replaces heavy RDO calculation by Shannon entropy calculation to boost encoder speed.

Class 3: Partition Fast Termination. In [5], A fast CU splitting decision scheme was proposed, using weighted SVM for early CU partition termination. Another fast termination algorithm [6] was recently proposed, using texture complexity of neighboring blocks to eliminate unnecessary partitions of current CU.

These prior contributions were proposed for natural video coding without considering the unique signal properties of screen contents and the newly-introduced coding modes for SCC. Statistically, computer-generated screen contents (such as

text, icon, graphics, etc.) tend to contain less distinct colors, sharper edges, irregular motions, repetitive patterns and less complicated local textures and orientations. On the other hand, the newly-introduced screen content coding tools (e.g., Intra Block Copy and Palette Coding Mode) are dependent on the previously encoded contents. Traditional homogeneity-based criterion on fast CU partition decision is not sufficient.

In this paper, we explore the feature-driven classification approaches for SCC fast CU splitting decision. By evaluating the feature relevance to the partition decision and comparing the prediction accuracies of several major classifier models, we propose a neural network based classification scheme with properly chosen features to achieve optimal "Rate-Distortion-Complexity" balance.

The sequel of this paper is structured as follows. Section 2 briefly reviews SCM-3.0 coding scheme, new coding tools and discusses the major challenges for fast CU partition beyond conventional HEVC. Section 3 explains our learning model selection, feature selection and model parameter training methodologies. In Section 4, the encoder implementation is illustrated from CU encoding workflow perspective. In Section 5, experimental results are presented and this paper concludes in Section 6 with future work summarized.

II. SCM-3.0 Quick Review

A. SCM-3.0 CU Size Optimization

SCM-3.0 shares exactly the same flexible quadtree block partitioning scheme as HEVC, which enables the use of CUs, Prediction Units (PUs) and Transform Units (TUs) to adapt to diverse picture contents. CU is the basic unit for mode decision and is always in square shape.

At encoder, pictures are divided into a series of non-overlapping LCUs and each LCU can be further partitioned into four equal-sized smaller CUs. These CUs can be recursively partitioned until the maximum hierachical depth is reached, as shown in Figure 1. At each CU-level, to determine the optimal encoding parameters (including partition decision and mode decision), an exhaustive search method is currently employed by examing and comparing RD cost of current CU (without splitting) using the best mode against sum of RD costs of 4 sub-CUs (each using best mode and partition).

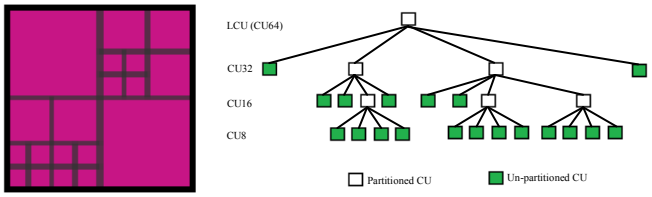

Figure 1. SCM-3.0 CU Hierachitical Quadtree Partitioning Structure

B. SCM-3.0 New Coding Modes

Beyond conventional HEVC, SCM-3.0 adopts two additional intra-frame coding tools, known as "Intra Block Copy" (IBC) [7][8] and "Palette Mode" [9] to compress screen contents more efficiently.

Intra Block Copy is an intra-frame version of the block matching scheme. To compress the current CU, the encoder will look back into previously coded area (either restricted or global) and find the best matching block. If chosen, an explicit "Block Vector" (BV) will be signaled to indicate the relative spatial offset between the best matching block and current CU.

Palette Mode encodes current CU as a combination of a color table and an index map. Color table stores representative color "triplets" of RGB or YUV. Then the original pixel block is translated into an index map indicating which color entry in the color table is used for each pixel.

C. SCC Fast Partition Chanllenges

Palette mode and intra block copy mode are both highly dependent on the graphic patterns and image colors that appeared previously. This "historical context dependency" makes fast CU partition problem much more complicated and challenging. For intra block copy (IBC) mode, depending on whether similar pattern appeared previously, IBC encoding costs of same CU pattern but at different locations may vary significantly. Similarly, for palette mode, two color tables are created. One is used for current CU and the other (also known as "palette predictor") is served as a dynamic lookup table storing the historical colors previously used. For the same CU pattern but at different locations, depending on whether similar colors appeared previously and how frequent these colors are, the palette mode encoding costs may vary significantly.

In conventional HEVC, CU homogeneity is an excellent indicator to decide whether to further partition. However, in SCM-3.0, homogeneity alone is not sufficient. New coding tools make "inhomogeneous" CUs possible to be encoded as a whole block. As shown in Figure 2, three 16x16 textual CUs in the top row are coded directly using palette mode (marked in green) without further partitioning into smaller 8x8 intra CUs (marked in purple) in the bottom row.

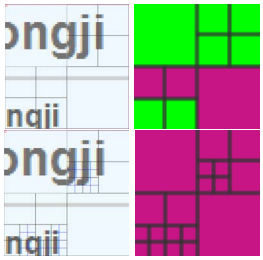

Figure 2. LCU Partition Decision Comparison between SCM-3.0 and HEVC (Top: Text LCU coded by SCM-3.0; Bottom: Text LCU coded by HEVC)

III. Mathine Learning For CU partitioning Decision

A. Feature Selection

Statistically, we have some prior knowledge for screen content coding and relationships between image pattern and mode selection. For instance, "Flat" or "Homogenous" regions are more likely to be encoded in larger CUs using Intra Mode. Palette-coded blocks are more likely to contain limited color number and sharper edges. Discontinuous-tone screen content areas are more likely to be encoded using IBC or Palette mode than Intra mode. Larger CUs are more likely to be further partitioned than smaller CUs especially under low QP setting.

With such prior knowledge, we may derive some dominant features, such as CU variance, CU distinct color number, CU gradient distribution and color concentration, etc. On the initial stage, 52 "direct" features are chosen, as follows:

- Group A: DC Values of CU and sub-CUs (5 terms);
- Group B: Variances of CU and sub-CUs (5 terms);
- Group C: Distinct Color Numbers in CU and sub-CUs (5 terms);
- Group D: Color Kurtosis of CU and 4 sub-CUs (5 terms);
- Group E: Gradient Kurtosis of CU and 4 sub-CUs (5 terms);
- Group F: Gradient Histograms of CU and 4 sub-CUs (Each histogram is directionally quantized into 4 bins, totally 20 terms);
- Group G: Edge Pixel Percentages of CU and 4 sub-CUs (5 terms that describe pixel proportion whose gradient magnitudes are above our predefined threshold 30);
- Group H: Quantization Parameter (QP);
- Group I: CU size.

These initial features were pre-evaluated using "Minimum Redundancy Maximum Relevance" (mRMR) [10]. This mutual information based algorithm gives us a rough estimation on feature importance. Then we further tested the feature rankings by excluding one feature term at a time and then evaluating the classification accuracy.

Preliminary tests show that CU size, QP value, CU variance, CU distinct color number, CU gradient kurtosis (which reflects the "peakiness" of CU gradient orientation histogram), CU color kurtosis (which reflects the "peakiness" of CU color histogram) and CU edge pixel percentage carry more weight in partition decision. Besides, the consistency among 4 sub-CUs are more correlated with partition decision than direct sub-CU features. Thus, we introduced a CU-level composite feature term named "sub-CU major directional inconsistency" (MDI) as defined in (1), where "min" and "abs" are minimum value and absolute value operator and "f_{ul}","f_{ur}", "f_{ll}" and "f_{lr}" are any possible direct features for upper-left, upper-right, lower-left and lower-right sub-CUs.

$$f_{MDI} = \min[abs(f_{ul} - f_{ur}) + abs(f_{ll} - f_{lr}), abs(f_{ul} - f_{ll}) + abs(f_{ur} - f_{lr})] \quad (1)$$

Instead of using QP and CU size as input features, we decide to train separate classifiers for different QP values and CU sizes to enhance the prediction accuracy.

The final feature set contains five CU direct features and five CU composite MDIs, as listed below.

- Feature 1: CU variance;
- Feature 2: CU distinct color number;
- Feature 3: CU color histogram kurtosis;
- Feature 4: CU gradient histogram kurtosis;
- Feature 5: CU edge pixel percentage;
- Feature 6: MDI for Sub-CU DC value;
- Feature 7: MDI for Sub-CU variance;
- Feature 8: MDI for Sub-CU distinct color number;
- Feature 9: MDI for Sub-CU color histogram kurtosis;
- Feature 10: MDI for Sub-CU gradient histogram kurtosis.

These 10 features provide high prediction accuracy on most CUs. However, for sequences dominated by IBC or palette blocks, such as "Console" and "Desktop" in [12], the prediction accuracy is reduced due to "context dependency". Even after we incorporate neighboring CU mode and partition information, we still cannot significantly improve the partition prediction accuracy. To resolve this insufficient accuracy, we apply a workflow-level compensation scheme by allowing a full-RDO test when the predicted partition decision is close to classifier decision boundary, as described with details in Section IV.

B. Classifier Model Selection

Several popular supervised learning methods, including "Logistic Regression" (LR) , "Linear Perceptron Classifier" (LPC), "Support Vector Machine" (SVM) [11] and "Neural Networks" (NN) were tested on 11 JCT-VC testing sequences [12]. Using the same features and dataset division, our preliminary experiments show that LR and LPC converge much faster but the prediction accuracy is lower than SVM and NN, especially on smaller CUs. For SVM, because our data samples are not linearly separable, we need to use Gaussian Radial Basis Function Kernel to implicitly map input features onto high-dimensional space. As a result, all supporting vectors (SVs) derived during the training stage have to be stored within the encoder for future prediction and the number of SVs tend to grow larger when training samples are increased. For instance, the number of SVs is over 2400 on average when we used 10000 random training samples. This will not only consume a huge amount of encoder memory but also make resultant classifier less adaptive to new data. Taken into account the prediction accuracy, encoder memory consumption and model simplicity, we finally adopt NN as our learning model.

C. Neural Network Parameter Training

Our training dataset contains CUs from first 30 frames of 11 SCC test sequences, totally 143,400 CU64 samples, 554,400 CU32 samples and 2,368,800 CU16 samples. For each CU size, the samples are divided into training-validation set (TVS) and testing set (TSS). TVS contains CUs from 8 sequences and TSS contains CUs from the other 3 sequences. To ensure all block patterns are examined, TVS samples are randomly permuted and divided into 4 equal-sized folds, with 3 folds treated as training set (TRS) and the remaining fold used as validation set (VLS). Due to the huge size of our TVS, we only used 75,000 random samples from TRS to train the neural network and 25,000 random samples from VLS to validate the generalization accuracy and tune NN parameters. Experiments show that the prediction accuracy on VLS does not vary much as we increase the TRS sample number.

The ground-truth CU partition decision labels are obtained by encoding 11 SCC sequences using SCM-3.0 encoder with default settings according to SCC common testing conditions [12]. Classifier models are trained on three CU levels (namely CU64, CU32 and CU16) for each QP setting (namely QP22, QP27, QP32 and QP37), respectively.

For simplicity, 2-layer NN structure is used with "sigmoid" transfer function between input and hidden layer and another "sigmoid" transfer function between hidden layer and output. NN training is implemented on Matlab using Neural Network Toolbox (Version 8.1).

To determine the optimal hidden layer node number for each CU size and QP level, cross-validations are used by randomly shuffling the folds between TRS and VLS. The node numbers that give the highest averaged validation accuracy among all shuffling are chosen. In our case, the optimal hidden node numbers are 6, 5 and 5 for CU64, CU32 and CU16, respectively.

IV. SCC FAST ENCODER IMPLEMENTATION

A pre-processing module with our network parameters (including weighting factors and bias terms for each CU size and QP level) is implemented and embedded in SCM-3.0 encoder software before intra-frame mode selection.

The NN classifier outputs a decision variable between 0 and 1, which can be interpreted as the probability that the CU belongs to "Partitioned CU" class. Rather than using a hard decision boundary (e.g., 0.5) to classify "Partitioned CU" and

"Non-Partitioned CU", we introduce a tunable parameter called "soft-decision testing margin" (SDTM) and apply SCM-3.0 full RD test to the CUs with NN outputs in the range of (0.5-SDTM, 0.5+SDTM). SDTM provides a controllable tradeoff between RD performance and complexity saving. A larger SDTM will preserve RD performance better but simultaneously reduce complexity saving. In our proposed encoder, a SDTM of value 0.2 is used. Namely, when NN decision is above 0.7, CU will be split directly and current CU-level mode selection will be bypassed; when NN decision is below 0.3, CU will not be split and all following sub-CU RDO will be terminated; when NN decision is between 0.3 and 0.7, SCM-3.0 default RDO will be used, as shown below in Figure 3.

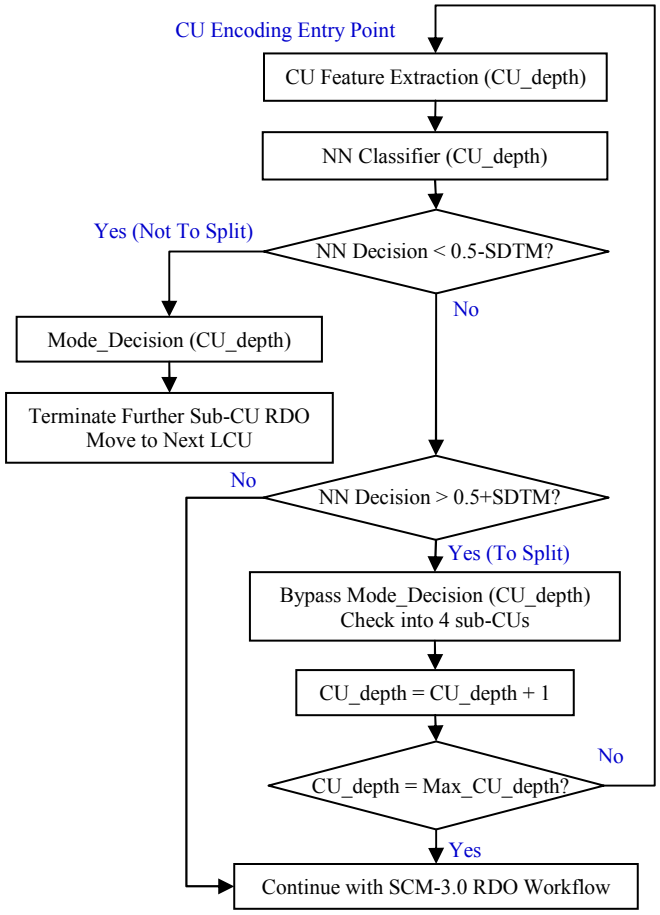

Figure 3. Proposed CU Encoding Workflow

V. EXPERIMENTAL RESULTS

Experiments are conducted on full-sequence data from the 11 screen content testing sequences selected by the experts in JCT-VC group [8]. These 8-bit sequences cover most typical screen content applications, such as "Desktop", "Console", "Map", "WebBrowsing", "SlideShow", etc., as shown in Figure 4. The encoding parameters (e.g. frame number, frame rate, QP, etc.) we used for each sequence are consistent with official common testing conditions described in [12].

Coding performances are evaluated using Windows 7 (64-bit) desktops with Intel-i5 CPU (2.67 GHz dual cores) and 4GB RAM. The complexity is measured directly using encoding time. Compared with Anchor software SCM-3.0, the encoder using our proposed scheme achieves a complexity reduction of 38% for "text/graphics" category, 35% for "mixed-content" category and 34% for "Animation" category with an averaged BD-rate increment within 3.0%. The detailed simulation results are provided below in Table I.

Figure 4. Sample Frames from JCT-VC SCC Sequences

TABLE I. PROPOSED ENCODER PERFORMANCE EVALUATION

Sequence	QP	*SCM-3.0*			*Proposed*			*Performance*	
		rate	*psnr*	*time*	*rate*	*psnr*	*time*	*BD-Rate*	*Complexity*
Map	22	54240	50.15	27165	55049	50.01	18385	+3.02%	-36.24%
	27	40268	45.89	24626	40803	45.71	16753		
	32	27657	41.20	21518	28182	41.12	13872		
	37	17333	36.85	18099	18325	36.80	9927		
Programming	22	51677	48.87	22921	52276	48.86	14643	+2.92%	-34.84%
	27	37056	45.08	20835	37575	45.00	13492		
	32	24962	41.04	19069	25414	40.89	12675		
	37	16144	36.83	17134	16682	36.65	11225		
SlideShow	22	7206	52.77	12556	7328	52.73	6521	+3.54%	-52.97%
	27	4595	49.43	11370	4683	49.39	5560		
	32	2806	46.01	10253	2889	45.92	4675		
	37	1756	42.42	9317	1851	42.36	3883		
WebBrowsing	22	8637	55.70	8050	8735	55.64	4947	+3.80%	-36.26%
	27	6826	51.48	7559	7015	51.39	4689		
	32	5252	46.97	6756	5347	46.62	4291		
	37	3846	40.77	6077	4012	40.22	4130		
Robot	22	64067	43.78	22786	64434	43.76	18240	+1.36%	-32.76%
	27	36727	39.81	19989	36997	39.80	15576		
	32	16981	35.96	16123	17164	35.94	10625		
	37	7443	33.07	11481	7706	33.07	5175		
BasketballScreen	22	188496	48.22	96716	190777	48.20	66459	+2.23%	-32.43%
	27	123053	44.51	86910	124091	44.46	60243		
	32	80237	40.90	74786	81148	40.83	51029		
	37	53154	37.20	65044	55740	37.04	41625		
MissionControl2	22	138049	48.44	88174	143401	48.41	55173	+2.84%	-38.30%
	27	94299	45.17	77712	96239	45.12	48105		
	32	61983	41.48	66520	62987	41.41	41380		
	37	39216	37.72	58254	40068	37.62	35021		
Console	22	35305	57.62	32945	36224	57.33	19732	+5.78%	-39.28%
	27	30632	53.00	31524	31651	52.89	19141		
	32	26072	48.17	29586	27668	47.93	18012		
	37	21515	42.46	27283	23253	42.09	16748		
Desktop	22	38051	55.76	39307	39242	55.51	25199	+3.96%	-33.74%
	27	33919	51.46	37576	34375	51.34	24413		
	32	29414	46.46	34957	30492	46.15	23253		
	37	24447	40.74	32228	25987	40.24	22382		
FlyingGraphics	22	78861	51.09	24694	79600	51.03	15334	+1.79%	-33.00%
	27	62954	46.48	23044	63255	46.37	14961		
	32	47223	41.69	21083	47933	41.55	14587		
	37	33893	36.84	18893	34553	36.76	13566		
MissionControl3	22	97624	48.50	54206	98916	48.51	35255	+2.06%	-34.75%
	27	68258	44.92	48674	69077	44.90	31556		
	32	46137	41.06	42244	46943	40.98	27561		
	37	30230	37.04	36364	30828	36.95	23962		

(rate: encoding bitrate in kbps; psnr: G/Y component PSNR in dB; time: encoding duration in second)

VI. CONCLUSIONS

In this paper, a novel machine learning based fast CU partition decision scheme is proposed to reduce the SCC encoder complexity. On top of SCM-3.0, our proposed neural network based classifier is implemented. Our simulation results show that the proposed scheme may introduce an average of 36.8% complexity reduction with only 3.0% BD-rate increase on average. Future investigations of this project may include (1) SCC advanced feature selection using deep learning techniques and (2) machine learning based SCC fast mode decision among Intra, Palette and IBC modes.

REFERENCES

[1] R. Joshi, J. Xu, R. Cohen, S. Liu, Z. Ma, and Y. Ye, Screen Content Coding Test Model 3 Encoder Description (SCM 3), Doc. JCTVC-S1014, Strasbourg, October 2014.

[2] Z. Liang, Z. Li, M. Siwei, and Z. Debin, "Fast mode decision algorithm for intra prediction in HEVC," IEEE Visual Communications and Image Processing (VCIP), Tainan, 2011, pp. 1–4.

[3] W. Jiang, H. Ma, and Y. Chen, "Gradient based fast mode decision algorithm for intra prediction in HEVC," International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012.

[4] M. Zhang, J. Qu, and H. Bai, "Entropy-Based fast Largest Coding Unit Partition Algorithm in High-Efficiency Video Coding," Entropy 2013, 15, 2277-2287.

[5] X. Shen, and Y. Lu, "CU splitting early termination based on weighted SVM," EURASIP Journal on Image and Video Processing (2013): 1-11.

[6] J. Hou, D. Li, Z. Li, X. Jiang, "Fast CU size decision based on texture complexity for HEVC intra coding", Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on, pp. 1096-1099, 2013.

[7] C. Pang, J. Sole, L. Guo, M. Karczewicz, and R. Joshi, Non-RCE3: Intra Motion Compensation with 2-D MVs, Doc. JCTVC-N0256, July 2013.

[8] J. Chen, Y. Chen, T. Hsieh, R. Joshi, M. Karczewicz, W.-S. Kim, X. Li, C. Pang, W. Pu, K. Rapaka, J. Sole, L. Zhang, and F. Zou, Description of screen content coding technology proposal by Qualcomm, Doc. JCTVC-Q0031, Valencia, April 2014.

[9] L. Guo, M. Karczewicz, and J. Sole, RCE3: Results of Test 3.1 on Palette Mode for Screen Content Coding, Doc. JCTVC-N0247, July 2013.

[10] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp.1226-1238, 2005.

[11] C. Hsu, C. Chang, and C.Lin, "A practical guide to support vector classification," Tech. rep (Computer Science Dept., National Taiwan University, 2003) http://www.csie.ntu.edu.tw/cjlin/guide/guide.pdf.

[12] H. Yu, R. Cohen, K. Rapaka, and J. Xu, Common conditions for screen content coding tests, Doc. JCTVC-S1015, Strasbourg, October 2014.