

A Fast TU Mode Decision Algorithm Based on Residual Difference for HEVC

N. R. Li¹, K. W. Liang¹, Z. Y. Chen¹, H. Y. Jiang¹, J. T. Fang², and P. C. Chang¹

¹National Central University, Taoyuan, Taiwan

²Ming Chuan University, Taoyuan, Taiwan

Abstract

High Efficiency Video Coding (HEVC) is the latest video coding standard. The residual quadtree (RQT) coding structure which provides variable block size for transform coding, is adopted in HEVC to achieve high coding efficiency. However, compared with previous standards, encoding complexity is increased significantly in HEVC due to the advanced encoding structure. A fast transform unit (TU) mode decision algorithm using residual difference is proposed in this paper to reduce the computational complexity. The proposed algorithm utilized the residual difference to determine the criterion of early TU termination and early TU skip. The threshold was trained from the beginning samples of each sequence. Experimental results showed that the proposed algorithm saves up to 75.64% and on average 64% TU encoding time compared with HM 15.0 in low delay P configuration and the loss of average BD-BR is less than 0.5%.

Keywords HEVC; residual; inter; TU; Fast algorithm; Transform

1. Introduction

Over past few years, mobile phones have become very popular and people are used to sharing video on mobile devices. With the increasing demand for high quality and high resolution video content, the newest video coding standard High Efficiency Video Coding (HEVC) was standardized by the Joint Collaborative Team on Video Coding (JCT-VC) in January 2013. HEVC can save half the bit rate of H.264/AVC at the same subjective image quality [1]. However, the improvement of coding efficiency of HEVC accompanies with high computational complexity. Therefore, it is crucial to reduce the encoding time for video applications on mobile devices.

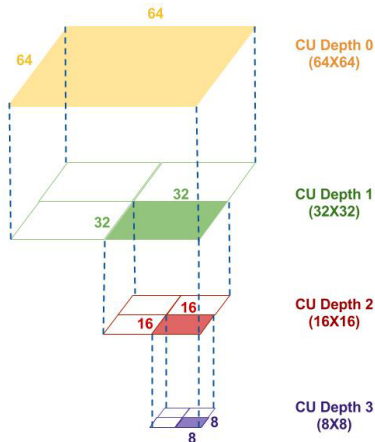


Fig. 1. Quadtree CU structure.

Similar to the role of macroblock (MB) in H.264, HEVC introduced Coding Unit (CU) based on a quadtree structure as shown in Fig. 1. Different from fixed size of MB, CU size varies from 64×64 to 8×8 which are denoted as depth 0 to 3 respectively, providing flexibility for encoding areas with smooth or complex texture. Moreover, a CU contains prediction units (PU) and transform units that also have variable block sizes to get better coding performance, as shown in Fig. 2. With more prediction modes, HEVC achieves more accurate predictions for video details.

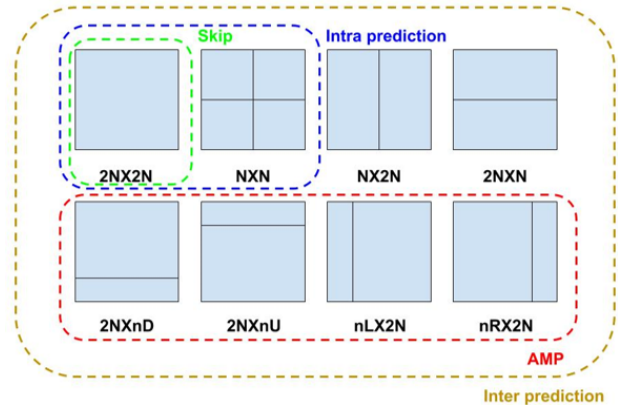


Fig. 2. PU mode of HEVC.

Transform Unit (TU) is the basic unit for transform, quantization, and entropy coding. The range of TU size covers from 4×4 to 32×32, which depends on the CU depth. Table 1 shows the default TU modes in HEVC. The RQT structure is employed to TU mode selection process, which provides the efficiently nested quadtree-based transform coding for various characteristics of prediction blocks.

Table 1. Default TU modes

CU Depth	TU size		
	Depth 0	Depth 1	Depth 2
0	32x32	32x32	16x16
1	32x32	16x16	8x8
2	16x16	8x8	4x4
3	8x8	4x4	

Figure 3 shows the RQT structure. The search order is labeled using the numeral order. After coding every four sub TUs, the Rate-Distortion Cost (RD Cost) is calculated and compared with the full TU to determine which it is belonged.

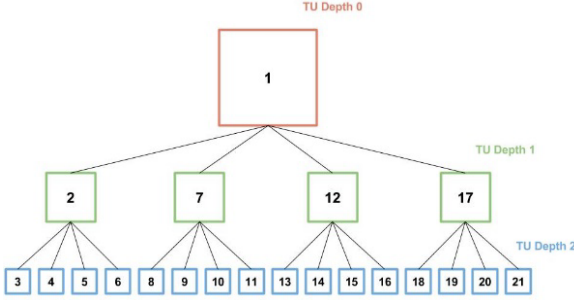


Fig. 3. Residual quadtree coding structure.

2. Related Works

The previously proposed algorithms [2]-[4] simplified the RQT processes by using the number of zero coefficients or the characteristics of transform coefficients. Both the two kinds of information are acquired after discrete cosine transform (DCT). However, to further accelerate the encoding process, [5] extended the concept of zero block detection in H.264 [6] and combined the property of zero block inheritance (ZBI) [7] to early skip or terminate the RQT processes.

3. Proposed Fast TU Decision Algorithm

In this paper, a residual difference scheme was proposed for the inter-prediction of TU mode (inter-TU). The residual of sub-blocks were calculated as residual difference SAD_{DIF} to train threshold, and the early termination or early skip of TU procedure was performed. The residual difference method was further combined with the zero-block detection [5] to early determine the TU depth. We also proposed a method to remove the redundant TU search.

3.1 Residual Difference

In [5], the prediction residual was analyzed and used as the representation for block similarity. The block similarity tends to be related to the TU mode decision. A transform block with four similar sub TUs is more likely to be encoded with current depth. By contrast, a TU with four sub TUs which are very different from each other would have a great chance to be split. We thus define residual difference to be the criterion to early terminate the TU mode decision process. We take sum of absolute differences, SAD , which is the sum of absolute values of the prediction residuals in one transform block as the fundamental information in the proposed method. SAD is calculated as (1):

$$SAD = \sum_{u=1}^{Height} \sum_{v=1}^{Width} |X[u, v]| \quad (1)$$

where $X[u, v]$ is the residual value of pixel located at $[u, v]$. After calculating the SAD of each sub TU, we further calculate the difference between them, denoted as SAD_{DIF} that is used to represent the similarity of the sub TUs in our work. The SAD_{DIF} is calculated as (2):

$$SAD_{DIF} = \frac{\sum_{i=0}^3 \sum_{j=0}^3 |SAD_i - SAD_j|}{2} \quad (2)$$

where the SAD_0 , SAD_1 , SAD_2 and SAD_3 indicate the SAD of four sub TUs. The result of SAD_{DIF} is referred as residual difference in our work.

To obtain the relationship between the TU mode and SAD_{DIF} , we analyzed the distribution of the SAD_{DIF} . As shown in Fig. 4, blue line indicates the SAD_{DIF} distribution of non-split TU, where the x-axis represents the value of SAD_{DIF} and y-axis represents the probability of TUs with the corresponding SAD_{DIF} value. The orange line indicates the TUs that are split into four sub blocks for transformation. It shows that the TUs with larger SAD_{DIF} have the tendency to be further split while the non-split TUs have smaller SAD_{DIF} values. We also list the mean values of SAD_{DIF} in different sequences with difference QPs to show a convincing argument as mentioned above.

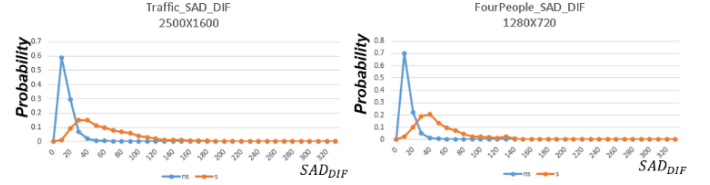


Fig. 4. The distribution of SAD_{DIF}

Table 2. Mean of SAD_{DIF} for non-split and split TUs

QP 22	Traffic	Basketball Drill	Basketball Pass	Four People
Non-split	7.42	7.52	6.29	5.84
Split	14.58	15.61	22.45	28.40
QP 27				
Non-split	10.24	8.95	8.83	9.57
Split	23.61	24.59	30.09	33.20
QP 32				
Non-split	16.16	14.79	19.85	18.30
Split	36.9	36.14	40.94	44.66
QP 37				
Non-split	29.51	24.74	33.79	29.11
Split	60.10	54.46	56.44	60.91

In addition to compare the mean values of SAD_{DIF} among differences, as listed in Table 2, the distributions for splitting and non-splitting based on SAD_{DIF} are shown in Fig. 5. It can be seen that even when two sequences have the close mean value of SAD_{DIF} , the curves could be different in shapes due to the characteristics of video contents. Video with high motion or complicated texture may result in more TU split blocks. By contrast, the smooth video tends to have more numbers of non-split TU blocks.

From the above observation and analysis, we propose a fast TU mode decision algorithm based on the characteristics of SAD_{DIF} . In this work, a threshold is required to determine early skip (ES) or early termination (ET) process.

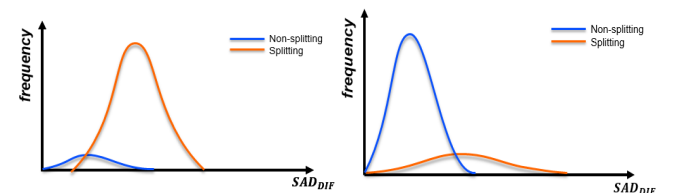


Fig. 5. Different amounts of SAD_{DIF} distribution. Left) Amount of non-split less than that of split. Right) Amount of non-split more than that of split.

When the current TU has SAD_{DIF} smaller than the threshold determined in the next paragraph, the algorithm enforces the TU to be early terminated in the current depth. Instead, an early skip would be performed when the current TU has larger SAD_{DIF} than the threshold. Fig. 6 shows the concept of the fast mode decision.

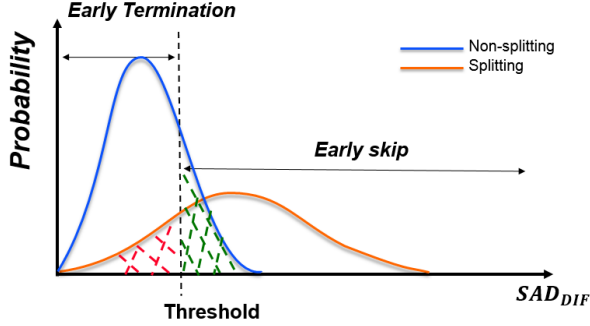


Fig. 6 Fast mode decision.

The threshold in our method is adapted to different sequences. For the sake of real-time application, the beginning video frames were used for training the threshold. After calculating the mean values of SAD_{DIF} of the split and non-split samples, the algorithm calculates the threshold by (3).

$$TH_{DIF} = \frac{X_{mean} + Y_{mean}}{2} + \left(Y_{mean} \times \left(\frac{X_{count}}{Y_{count}} - 1 \right) \right) \text{ for } X_{count} > Y_{count}$$

$$TH_{DIF} = \frac{X_{mean} + Y_{mean}}{2} + \left(X_{mean} \times \left(\frac{X_{count}}{Y_{count}} - 1 \right) \right) \text{ for } Y_{count} > X_{count} \quad (3)$$

where X_{mean} denotes the mean of the SAD_{DIF} of non-split samples while Y_{mean} represents the split ones. X_{count} and Y_{count} denote the number of non-split and split blocks in the training samples, respectively.

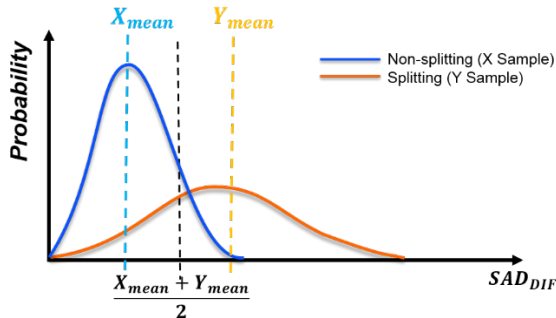


Fig. 7. Illustration of formula derivation.

The average of mean values $\frac{X_{mean} + Y_{mean}}{2}$ is used to be the main prediction of Th_{DIF} . However, this value $\frac{X_{mean} + Y_{mean}}{2}$ changes under different QPs. Moreover, as mentioned in the previous paragraph, the total number of split and non-split blocks

also influence the distribution of SAD_{DIF} . We thus add the adjustment quantity, $Y_{mean} (or X_{mean}) \times \left(\frac{X_{count}}{Y_{count}} - 1 \right)$ to the main prediction. The number of samples adopted for training is calculated by (4) and the result is listed in Table 3.

Table 3. Number of training samples for different video

	QP22	QP27	QP32	QP37
32×32	Sequence size 32	Sequence size 32×2	Sequence size 32×3	Sequence size 32×4
16×16	Sequence size 16	Sequence size 16×2	Sequence size 16×3	Sequence size 16×4

$$\text{Number of training samples} = \frac{\text{Sequence size}}{\text{Block size} \times N} \quad (4)$$

where $N=1, 2, 3, 4$ for QP=22, 27, 32, 37.

3.2 Redundant Search Removing

In most cases, residuals from different PU modes would be different. However, different prediction modes in the same CU depth could result in the same residual. Since the TU mode decision depends on the corresponding prediction residuals, we can infer that the same residual may probably result in the same TU mode. Thus, we utilize the first prediction residual of the current CU as the basis and compare the other residuals with it. When two residuals of two PU modes are the same, the TU mode decision process is early terminated. Fig. 8 illustrates this method.

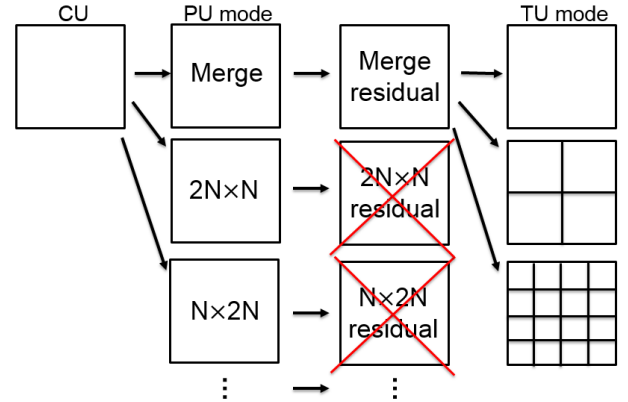


Fig. 8. Illustration of Redundant Search Removing.

3.3 Zero Block Detection

To achieve more accurate and faster coding efficiency, the zero block detection algorithm [5] was combined in the proposed scheme. The main concept is that the DC value is considered the largest transform coefficient in the whole block usually. Therefore, if the DC value in a block is less than 1, this block is close to a zero block. Then we can use the upper bound of SAD in pixel domain which is derived back from transform domain to make a judgment, the formula is as follows:

$$SAD < \frac{((hor \cdot ver \cdot C) \ll Qbits) - f(QP\%6) \cdot (64 \cdot N \cdot hor + 2 \cdot hor \cdot ver)}{64 \cdot 64 \cdot f(QP\%6)} \quad (5)$$

Where $C = \frac{427}{128}$, therefore we set the upper bound of the SAD value as follows:

$$TH_{ZB} = \frac{((hor-ver \cdot C) \ll Qbits) - f(QP\%6) \cdot (64 \cdot N \cdot hor + 2 \cdot hor \cdot ver)}{64 \cdot 64 \cdot f(QP\%6)} \quad (6)$$

When $SAD < TH_{ZB}$, this TU block is determined as a zero block. Further combine the concept of zero block inheritance [7], that is, when the full TU is a zero block, its sub TUs also have a great chance to be a zero block, and bitrate used in full TU is less than bitrate used in its sub TUs. Therefore, this TU block will be early terminated and the best mode of this TU block is determined to choose current depth of TU. By contrast, when $SAD \geq TH_{ZB}$, this TU block will early skip the current depth and split to the next depth.

3.4 Overall Algorithm

In this subsection, we describe the overall algorithm of fast TU mode decision. The parameters used in this paper are listed in Table 4.

Table 4. Parametric representation

Parameter	Notation
Depth of CU	CU_D
Temp Depth of CU	CU_{TD}
Depth of TU	TU_D
Residual value	Res
Temp of Residual value	Res_{Temp}
Sum of Absolute Residual	SAD
Residual difference	SAD_{DIF}
Threshold of Zero Block	Th_{ZB}
Threshold of Residual difference	Th_{DIF}
Ratio of non-split and split TUs	C_{rate}

The flowchart of the proposed overall algorithm is plotted in Fig. 9. In the beginning of TU encoding process, the residuals of this TU is checked to decide whether this TU will be skipped or terminated. If this TU is determined to be skipped, the algorithm will skip this depth and do normal procedure on next depth. If this TU is chosen to be terminated, it will only do normal procedure of this TU depth. Threshold is updated under different sequences and QPs as mentioned in the previous section.

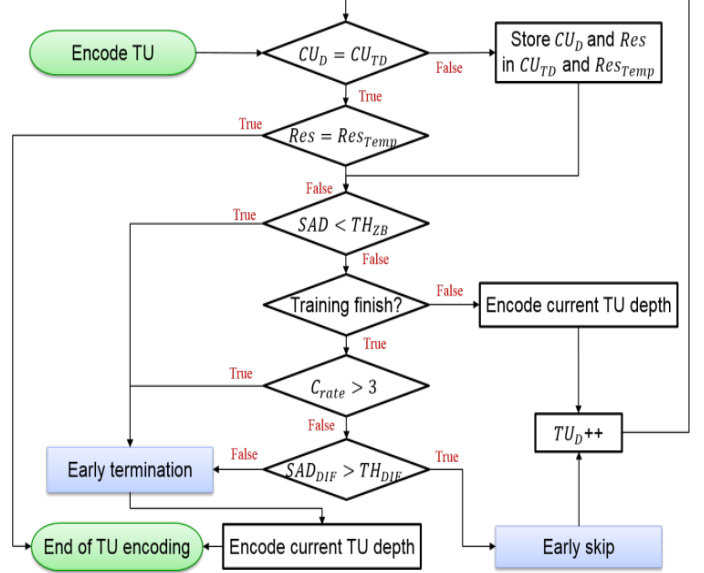


Fig. 9. The flowchart of the proposed overall algorithm

4. Experimental Results

The proposed fast TU decision algorithm was implemented using HEVC reference software (HM 15.0) with low-delay P configuration. The test platform was a Desktop PC with an Intel(R) Core(TM) i7-3770 @ 3.40-GHz CPU, 8-G bytes RAM, and Windows 7 system.

Table 5 shows the experimental results of the first 100 frames of Class A~E test sequences under QPs 22, 27, 32, and 37, respectively. Coding efficiency was evaluated in terms of BD-rate (BDBR) (%) [8], and time saving(ΔT) (%) represented the time saving in TU encoding procedure compared with HM 15.0.

$$\text{Time saving (\%)} = \frac{1}{4} \sum_{i=1}^{QP_i} \frac{TU \text{ Time}_{HM 15.0}^{QP_i} - TU \text{ Time}_{prop.}^{QP_i}}{TU \text{ Time}_{HM 15.0}^{QP_i}} \times 100 \quad (7)$$

$$QP = \{22, 27, 32, 37\}$$

The experimental results show that the proposed fast TU decision can provide 64.07% time saving on average with only 0.469% BD-rate loss. The proposed algorithm achieves much time saving for sequences, FourPeople and KristenAndSara, which are with low motion and less background transitions.., The number of zero blocks in these sequences are more than that of other sequences. The more zero blocks in the sequence, the more blocks will be skipped with ignorable loss. Hence the decision of the proposed algorithm can be more accurate and both BDBR and time saving performance can be better.

Compared the performance with [5] the BDBR of our proposed method increases smaller. The reason is that SAD_{DIF} values between split and non-split blocks are distinct, and we use adaptive threshold to early terminate the RQT procedure. Hence the BDBR is lower than the others.

Table 5. Simulation results.

Class	Sequence	[5]		proposed	
		BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
A	PeopleOnStreet	0.876	-62.10	0.870	-57.61
	Traffic	0.298	-65.45	0.350	-67.66
B	Kimono	0.765	-63.54	0.657	-64.92
	ParkScene	0.714	-64.03	0.815	-63.36
	BasketballDrive	0.483	-62.36	0.455	-64.67
	BQTerrace	0.250	-63.12	0.248	-63.06
	Cactus	0.068	-62.75	0.277	-66.19
C	BasketballDrill	0.509	-63.03	0.508	-64.89
	BQMall	0.444	-62.27	0.473	-63.63
	PartyScene	0.502	-59.72	0.450	-59.23
	RaceHorsesC	0.535	-59.66	0.482	-57.06
D	BasketballPass	0.325	-61.91	0.125	-61.29
	BlowingBubbles	0.522	-61.00	0.471	-60.61
	BQSquare	0.743	-61.43	0.775	-60.11
	RaceHorses	0.573	-60.02	0.629	-55.18
E	FourPeople	0.194	-68.15	0.288	-75.64
	Johnny	0.578	-68.42	0.433	-74.43
	KristenAndSara	0.089	-68.37	0.130	-73.76
Average		0.471	-63.16	0.469	-64.07

5. Conclusion

A fast TU decision algorithm method is proposed based on the information of residual. In the proposed algorithm, we use the residual difference to be the criterion of choosing the path of TU mode decision. The distribution of residual difference is also utilized to obtain an adaptive threshold for individual sequence. Combined with zero block detection and redundant search removing, the experimental results showed that the proposed fast TU decision algorithm can achieve significant time saving: at most 75.64% TU encoding time and on average 64% TU encoding time compared with HM 15.0 in low delay P configuration and the loss of average BD-BR is no more than 0.5%.

References

- [1] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [2] Y. Shi, Z. Gao, and X. Zhang, "Early TU Split Termination in HEVC Based on Quasi-zero-block," in *Proc. 3rd International Conference on Electric and Electronics (EEIC)*, Hong Kong, China, Nov. 2013, pp. 450–454.
- [3] K. Choi and E. S. Jang, "Early TU Decision Method for Fast Video Encoding in High Efficiency Video Coding," *Electronics Letters*, vol. 48, no. 12, pp. 689–691, June 2012.
- [4] J. Kang, H. Choi, and J. G. Kim, "Fast Transform Unit Decision for HEVC," *Proc. IEEE International Congress on Image and Signal Processing (CISP)*, Hangzhou, China, Dec. 2013, pp. 26–30.
- [5] Z. Y. Chen, J. T. Fang, C. S. Chiang, and P. C. Chang, "Computation reduction in high-efficiency video coding based on the similarity of

transform unit blocks," *Journal of Electronic Imaging*, vol. 23, no. 6, 061105-1–061105-10, Dec 2014.

- [6] W. He, and M. Xiao "Research on Zero-block Detection Threshold in H.264/AVC", in *Proc. the 2nd International Conference on Pervasive Computing and Applications (ICPCA)*, Jul. 2007, pp. 557–561.
- [7] S. W. Teng, H. M. Hang, and Y. F. Chen, "Fast Mode Decision Algorithm for Residual Quadtree Coding in HEVC," in *Proc. Visual Communications and Image Processing (VCIP)*, Nov. 2011, pp. 1–4.
- [8] G. Bjontegaard, "Calculation of Average PSNR Difference Between RD-curves," *ITU-T Q.6/SG16 VCEG 13th Meeting*, Document VCEG-M33, 2001.

Author Biography

Nian-Rong Li received the BS degree in communication engineering from National Central University, Taiwan, in 2014. Her research interest is video coding. She is currently pursuing the MS degree at the VAPLab in the Department of Communication Engineering at NCU, Taiwan.

Kai-Wen Liang received the MS degree in communication engineering from National Central University (NCU). She is currently pursuing the PhD degree at the VAPLab in the Department of Communication Engineering at NCU, Taiwan. Her research interest is video coding.

Zong-Yi Chen received the B.S. degree in electrical engineering from National Central University, Taiwan, in 2005, and the M.S. and the Ph.D. degree in communication engineering from National Central University, Taiwan, in 2007 and 2016, respectively. His research interests include multimedia processing and video compression (H.264/AVC, H.264/SVC, and HEVC).

Hui-Yu Jiang received the BS degree in electrical engineering from Tamkang University, Taiwan, in 2015. Her research interest is video coding. She is currently pursuing the MS degree at the VAPLab in the Department of Communication Engineering at NCU, Taiwan.

Jiunn-Tsair Fang received the PhD degree in electrical engineering from National Chung-Cheng University, Taiwan in 2004. Currently, he is an assistant professor in the Department of Electronic Engineering at Ming Chuan University, Taiwan. His researching interest includes video/image coding, audio processing, and channel coding.

Pao-Chi Chang received his PhD degree in electrical engineering from Stanford University, California, in 1986. From 1986 to 1993, he was a research staff member at IBM T. J. Watson Research Center, New York. In 1993, he joined the faculty of NCU, Taiwan, where he is presently a professor in the Department of Communication Engineering. His main research interests include speech/audio coding, video/image compression, image and music retrievals, and deep learning techniques.