EXTENDED VERSION OF ICIP (INTERNATIONAL CONFERENCE ON IMAGE PROCESSING) PAPER

Md. Zahirul Islam[1], *Tanvir Ahmed Redoy*[1], *Ashek Ahmmed*[2], *Manoranjan Paul*[3], *and Manzur Murshed*[4]

[1] Department of CSE, Rajshahi University of Engineering and Technology, Bangladesh.
[2] School of Computer Science, University of Sydney, Australia.
[3] School of Computing and Mathematics, Charles Sturt University, Australia.
[4] School of Info Technology, Deakin University, Australia.

1. EXPERIMENTAL RESULTS (BD TABLE)

1.1. Using the Information of Cuboids

In Table 1, using the **cuboids map**, all the encoding results are reported and these results are submitted to **ICIP** conference except the last 2 sequences namely *Tango2* and *FoodMarket4*.

Table 1. Encoding complexity vs Bjontegaard delta data from the testset. (ICIP, Only 64 by 64, **Cuboids**)

Sequence	BD-BR	BD-PSNR	ΔT
Vehicles	1.27%	−0.08 dB	9.49%
CatRobot	2.67%	−0.09 dB	10.33%
DaylightRoad2	2.36%	−0.05 dB	8.61%
Campfire	6.87%	−0.15 dB	14.29%
ParkRunning3	1.40%	−0.07 dB	10.48%
Tango2	5.5%	−0.11 dB	13.16%
FoodMarket4	8.6%	−0.18 dB	18.68%
Average	**4.10%**	**−0.11** dB	**12.15%**
Average Except Last 2	**2.91%**	**−0.09** dB	**10.64%**

1.2. Using the Information of Edge

In Table 2, using the **edge image**, all the encoding results are summarized. In this case, **Sobel** edge detection approach is used to generate the edge image.

1.3. Using the Information of Cuboids+Edge

In Table 3, the encoding outputs are presented using **both cuboids map and edge information**. When the split decision of a certain CU (64×64) is **equal** then only this decision is taken by **merging** these two separate approaches. On the other hand, if these two distinct methods contradict to obtain a decision of a specific CU, the decision is come from **VVC** (Simply, this CU is handovered to VVC to get the split decision).

Table 2. Encoding complexity vs Bjontegaard delta data from the testset. (Only 64 by 64, **Edge**)

Sequence	BD-BR	BD-PSNR	ΔT
Vehicles	0.97%	−0.06 dB	4.74%
CatRobot	2.02%	−0.07 dB	8.45%
DaylightRoad2	0.84%	−0.02 dB	9.82%
Campfire	1.32%	−0.03 dB	11.17%
ParkRunning3	0.85%	−0.05 dB	13.39%
Tango2	2.05%	−0.04 dB	16.17%
FoodMarket4	4.68%	−0.11 dB	32.84%
Average	**1.82%**	**−0.05** dB	**13.80%**
Average Except Last 2	**1.20%**	**−0.05** dB	**9.51%**

1.4. Using the Information of Cuboids+Edge+Threshold

In Table 4, the encoding outputs are shown by using the information of cuboids map, edge and setting a threshold. This method is almost analogous to subsection 1.3. However, there is a subtle difference between the subsection 1.3 and 1.4. A short description is given below of this method.

Assuming that the split decision of certain CU (64×64) is d_1 which obtained from cuboids map. On the other hand, the decision is d_2 that is coming from edge model. Now, the final decision is calculated in the following equ., where, d is the final split decision, W_1=W_2= **0.5** and the threshold = **0.5**.

$$d = \frac{(d_1 * W_1) + (d_2 * W_2)}{(W_1 + W_2)} \tag{1}$$

2. EXPERIMENTAL RESULTS (BD CURVE)

In this section, all the BD curves for various test sequences are shown. Fig. 1, Fig. 2, Fig. 3, Fig. 4, Fig. 5, Fig. 6 and Fig. 7 show the RD performance of *Vehicles, CatRobot, DaylightRoad2, Campfire, ParkRunning3, Tango2, FoodMarket4* respectively.

Table 3. Encoding complexity vs Bjontegaard delta data from the testset. (Only 64 by 64, **Edge + Cuboids**)

Sequence	BD-BR	BD-PSNR	ΔT
Vehicles	0.20%	−0.01 dB	0.68%
CatRobot	0.94%	−0.03 dB	2.23%
DaylightRoad2	0.29%	−0.01 dB	3.44%
Campfire	0.36%	−0.01 dB	3.31%
ParkRunning3	0.23%	−0.01 dB	6.29%
Tango2	1.51%	−0.03 dB	7.01%
FoodMarket4	1.47%	−0.03 dB	7.43%
Average	**0.71%**	**−0.02 dB**	**4.34%**
Average Except Last 2	**0.40%**	**−0.01 dB**	**3.19%**

Table 4. Encoding complexity vs Bjontegaard delta data from the testset. (Only 64 by 64, **Edge + Cuboids + Threshold**)

Sequence	BD-BR	BD-PSNR	ΔT
Vehicles	0.37%	−0.02 dB	1.01%
CatRobot	2.13%	−0.07 dB	3.79%
DaylightRoad2	1.04%	−0.03 dB	5.62%
Campfire	1.06%	−0.03 dB	5.21%
ParkRunning3	0.50%	−0.03 dB	5.60%
Tango2	4.51%	−0.09 dB	11.11%
FoodMarket4	8.02%	−0.17 dB	16.87%
Average	**2.52%**	**−0.06 dB**	**7.03%**
Average Except Last 2	**1.02%**	**−0.04 dB**	**4.23%**

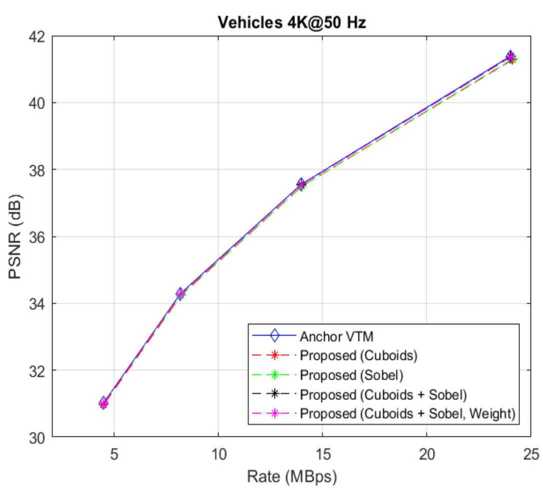

Fig. 1. RD curve for the *Vehicles* sequence.

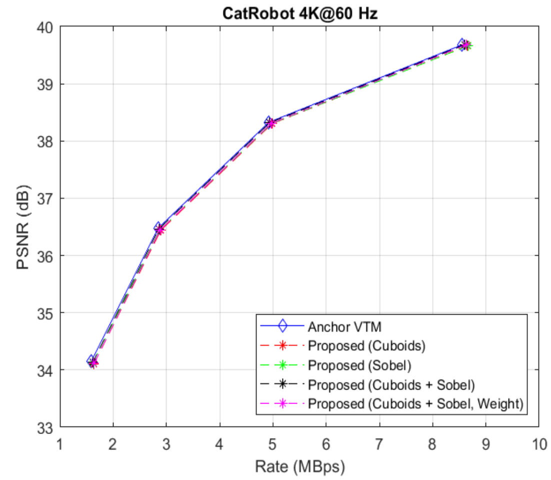

Fig. 2. RD curve for the *CatRobot* sequence.

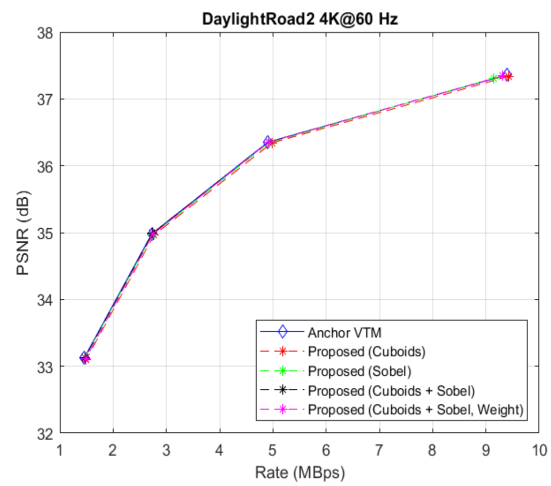

Fig. 3. RD curve for the *DaylightRoad2* sequence.

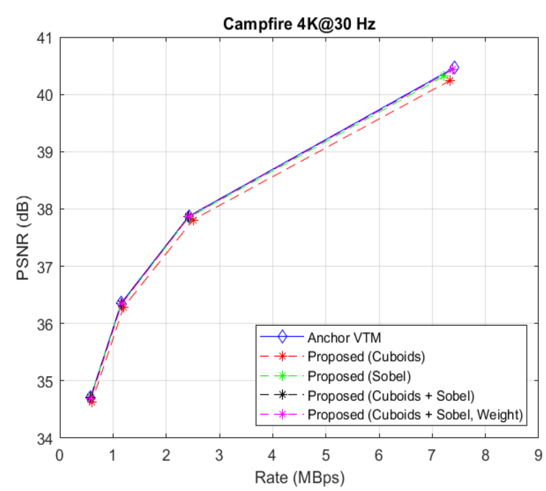

Fig. 4. RD curve for the *Campfire* sequence.

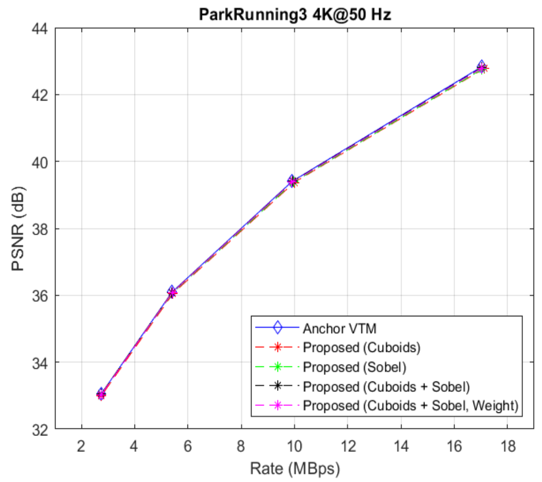

Fig. 5. RD curve for the *ParkRunning3* sequence.

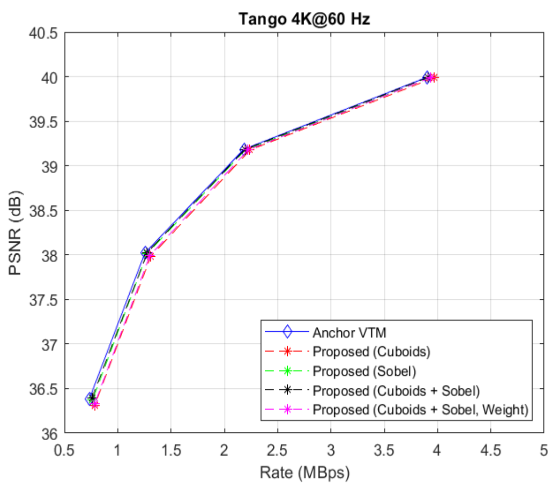

Fig. 6. RD curve for the *Tango2* sequence.

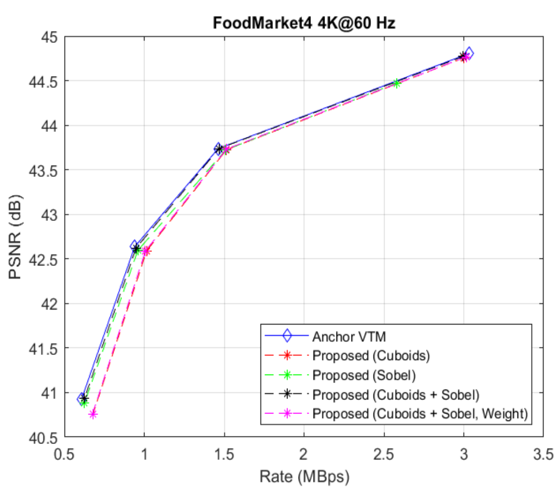

Fig. 7. RD curve for the *FoodMarket4* sequence.

3. CONCLUSION

- It is evident that, **edge model** performs better in case of BD-BR compared to cuboids method and it is shown in Table 2.

- **Merge** options of **edge and cuboids** have the **outstanding performance** in terms of BD-BR in comparison with the single methods. This results are shown in Table 3 and 4.

- ***Tango2 and Foodmarkte4*** act as **outliers** in terms of BD-BR for all the methods that are used here. We can simply discard these two in this regard or these sequences can be reported but needs to be shown as outliers.

- I prefer the method of using the information of **cuboids+edge** right now that is described in **subsection 1.3** for getting the decision of 64×64 CU. The reason of choosing this one is that this technique has the **lowest bit rate loss** in terms of BD-BR among the others.

- On top of that, if our method is able to anticipate the decisions **more perfectly** at 64×64 level, definitely, at the successive bottom levels, for instance, 32×32 or 16×16, our proposed method will **perform well**. In contrast, if the decision is made as **wrong** at 64×64 level, the bit rate as well as RD performance for the successive lower levels may be deteriorated greatly like as exponentially.

- Though the time savings of cuboids+edge (subsection: 1.3) method at 64×64 level is comparatively small among the others, we will get the **better results** at 32×32 level as these step has **lowest bit rate loss**. Also, we have the experimental results that is up to 70% time savings starting from 50% can be achieved if our method is able to provide more accurate split decision.