FAST 3D-HEVC DEPTH MAPS INTRA-FRAME PREDICTION USING DATA MINING

1Mário Saldanha, 2,3 Gustavo Sanchez, 2César Marcon, 1Luciano Agostini
1Federal University of Pelotas – Pelotas, Brazil
2Pontifical Catholic University of Rio Grande do Sul – Porto Alegre, Brazil
3IF Farroupilha – Alegrete, Brazil
{mrdfsaldanha, agostini}@inf.utfpr.edu.br, gustavo.sanchez@acad.pucrs.br, cesar.marcon@pucrs.br

1. Introduction
3D-HEVC is an extension of the High Efficiency Video Coding (HEVC). - Adaption of the Multiview Video plus Depth (MVD).
Depth maps provide geometrical information of the scene.
- Each texture frame is associated with a depth map
- Essential to generate virtual views with high quality
3D-HEVC provides a flexible quadtree-based structure for depth maps coding, which is evaluated through Rate-Distortion Optimization (RDO) in 3D-HEVC Test Model (3D-TM).
- High encoding efficiency at the cost of a significant increase in the encoder computational complexity.
- Evaluate many combinations of encoding structures
This paper proposes a data mining approach to build static decision trees, defining if each CU should be or not split into smaller CUs at depth maps intra-frame prediction, without using the full RDO evaluation.

2. Initial Analysis and Motivation
Fig. 1(a) shows the complexity distribution (concerning processing time) between texture and depth maps using Quantization Parameter-pairs (QP-pair) values (QP-texture/QP-depth).
- Depth maps coding is 5.8 times more complex than the texture coding.
- Texture coding only applies the HEVC intra-frame prediction.
- Depth maps coding uses HEVC intra-frame prediction, DMMs, DIS and SDC evaluations.

Fig. 1. (a) Complexity distribution for texture and depth coding and (b) CU size distribution for depth coding.

Fig. 1(b) shows the CU size distribution for depth maps coding per QP-depth value, highlighting a variation in the QP-depth causes a different CU size distribution.
- For lower QP-depth values (QP-depth=34) about 67% of CUs were encoded with the size of 8×8
- QP-depth=45 about 50% of CUs were encoded with the size of 64×64

3. Encoder Attributes Evaluation
Fig. 2 presents the density probability of the 64×64 CUs do not split into smaller CUs for some collected attributes.
- MaxDiff and VAR_64 have lower values for those CUs that are not split into smaller CUs
- VAR_16 provides essential information for sub-blocks inside a 64×64 CU and can indicate presence of edges
The attributes evaluation allows concluding that only QP-depth, R-D cost, VAR, VAR_size, Average, and MaxDiff are relevant to build the static CU decision trees.

4. Proposed CU Trees
Three static decision trees to define when CUs of sizes 16×16, 32×32 and 64×64 should be or not split into smaller CUs.

Data mining process:
- Kendo video sequence (randomly selected) encoded in all-intra configuration
- CTU size has been limited to 16×16, 32×32 and 64×64 pixels for each evaluation
- For each video sequence was stored the attributes and the information indicating if the CU has been split

Training process:
- J48 algorithm available on Waikato Environment for Knowledge Analysis (WEKA)
- Balanced input data set: two sets of data with equal sizes containing inputs that result in (i) splitting and (ii) not splitting of CUs.

Fig. 3 illustrates the static decision tree generated for 64×64 CUs, where the leaves “N” and “S” correspond to the not split and split decisions, respectively.

Fig. 3. Decision tree for splitting decision in 64×64 CUs.

5. Experimental Results
The results using the 3D-HTM 16.0 and Common Test Conditions (CTC) are presented in Table 1:

- Not splitting of 35% for 16×16 CUs, 50% for 32×32 CUs and 60% for 64×64 CUs
- Complexity Reduction of 52.4% in the whole encoder.
- BD-rate increase of only 0.18% in synthesized views.

Table 1. Proposed solution results for CTC evaluation in all-intra configuration.

5. Conclusions
Three static decision trees were trained using WEKA software to define if a current encoding CU should or not be split into smaller CUs.
An evaluation of the most relevant encoder attributes was done, where some of these attributes were selected to be used in an offline training.
Experimental results:
- Complexity reduction of 52.4%
- BD-rate increase of 0.18%
Best results in both axes when compared to related works.

Only the Kendo video sequence was used in the offline training process and the decision trees creation.
- The remaining test sequences were evaluated to demonstrate that the proposed decision trees were not overfitted for the experimental analysis.
- The solution surpasses the related works in both BD-rate and complexity reduction.

All evaluated related works obtained smaller complexity reduction results and higher BD-rate.

Support: