Low complexity heuristic for the inter-prediction PU decision step on HEVC encoders

Mateus Grellert[1], Felipe Sampaio[2], Luciano Agostini[1], Julio C. B. Mattos[1]
{mgdsilva, fmsampaio, Agostini, julius}@inf.ufpel.edu.br

[1]GACI - Federal University of Pelotas

[2]PPGC - Federal Universityof Rio Grande do Sul

Abstract

This paper presents the Motion Vectors Merging (MVM) heuristic, which is a method to reduce the HEVC inter-prediction complexity targeting the PU partition size decision. The goal of this work is to avoid several ME searches during the PU inter-prediction step in order to reduce the execution time of the entire encoding process. The MVM algorithm is based on merging NxN PU partitions in order to compose larger ones by using a fast heuristic instead of the original RDO approach. The proposed method was implemented into the HM test model and it provided an execution time reduction of up to 34% with insignificant rate-distortion losses (0.08 dB drop and 1.9% bitrate increase in the worst case). Besides, there are no related works in the literature that exploit PU-level decision optimizations. When compared with aggressive CU-level fast decision methods, the MVM shows itself an efficient solution, achieving results as good as those works.

1. Introduction

The recent advances in technology enabled many improvements regarding multimedia systems. Along with these advances, the consumer market constantly demands better quality media, such as higher resolution digital videos. When H.264/AVC [1] (the state of the art video-coding standard) was created, only a few specific devices supported 1080p videos. However, the current scenario is different, and 1080p videos are now supported by a plethora of electronic devices and higher resolutions begin to catch the market's attention. With that in mind, a group composed of video coding experts from ITU-T and ISO/IEC was formed under the name of Joint Collaborative Team on Video Coding(JCT-VC) [1]. The purpose of the JCT-VC was to develop a new video-coding standard with improved compressing tools focusing on higher resolution videos. From this collaboration, the High Efficiency Video Coding (HEVC) standard is emerging, aiming to double the compression rates when compared to H.264/AVC for the same video quality [2]. The final draft of this standard is expected for release in January 2013.

The HEVC standard is still under development, but its current innovations already bring negative expectations regarding its complexity. The reference software for this standard, called HEVC Model (HM),already contains a large set of complex tools in its current version, causing great concern when real time applications are considered. In addition, this complexity is also a critical drawback for mobile devices, since a great amount of energy will be required to perform video coding/decoding tasks.The inherent data structures defined in the HEVC standard can be highlighted as one of the main causes of its high complexity. Frames are now divided into Treeblocks, which can be subdivided into Coding Units (CU). Furthermore, CUs can be recursively partitioned, forming a quad-tree structure, which is illustrated in fig. 1.

The CU quad-tree decision is originally performed using the Rate-Distortion Optimization (RDO) technique, which evaluates the bitrate and the objective quality (generally expressed by the PSNR) produced by every possible configuration. In other words, the prediction, residual coding and entropy coding stages are performed for every possible CU partition. During the prediction stage, each CU is once again divided into Prediction Units (PU), introducing PU decision trees for each node of the CU decision tree. Each PU tree also considers the RDO technique as decision strategy.

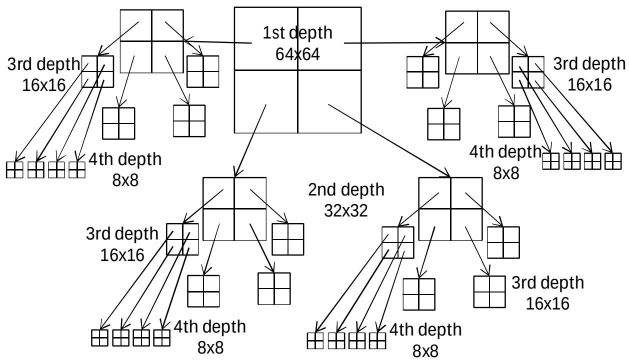


Figure 1. Example of quad-tree CU partitioning

There are several works in the literature that propose heuristic decision approaches in order to reduce the CU encoding time. In [3], the encoder execution time is controlled defining a specific target complexity. Then, using historical information of past decisions, the decision core is able to avoid some CU size checking assuming that they would not be chosen in a RDO configuration. The work of [4] performs a similar decision: it cuts some CU nodes in the quad-tree based on decisions that where taken inside the current frame. Due to this fact, the author classified its decision as a frame-level decision.

This work presents the Motion Vectors Merging (MVM) heuristic, a fast decision method for the inter-prediction PU trees, which aims to reduce the complexity involved in the coding process. The MVM decides the PU size using heuristic approaches instead of applying the Motion Estimation (ME) process for every possible configuration. In doing so, the PU decision complexity can be significantly reduced, therefore reducing the complexity associated to each one of the CU decision tree nodes. The MVM proposed heuristic checks all internal borders among the *NxN* partitions. Each border is analyzed, and a heuristic defines if the partitions can be merged or not. If any merging occurs, the rate-distortion cost is evaluated for the decided PU partition, generating enough information for the CU-level decision.

The MVM fast decision method for inter-prediction PUs was implemented inside the HM 3.4 test model [5], and rate-distortion results, as well as speed-up results, were generated. The obtained results show that MVM is capable of achieving a complexity reduction of 34% with insignificant loss in terms of quality (0.08 dB on average) and minor increase in bitrate (1.9% in the worst case). In addition, MVM can be easily coupled to other solutions that reduce encoding complexity on different levels, such as CU level.

2. Related works

To the best of our knowledge, there are no published works that present fast decision heuristic-based methods targeting the HEVC PU decision level. All related works focus on avoiding the processing in every depth of the CU quad-tree decision possibilities. These works perform cuts in the quad-tree by keeping the history of decision during the current frame encoding or using the past frames decisions history.

The work presented in [3] proposes a complexity control targeting power-constrained applications, the complexity control checks the battery status and decides by running the encoding process on a specific complexity: higher complexity, higher energy consumption and best rate-distortion results, or lower complexity, lower energy consumption and worst rate-distortion results. This work avoids some CU evaluations based on past frames analyses: (1) one frame is full RDO processed and, afterwards, (2) based on the best rate-distortion results the decision builds a history map of decisions that (3) will direct the future frame decisions using the already built map of past choices.

The work presented in [4] proposes a fast CU decision algorithm for frame-level or CU-level encoding process acceleration. The acceleration is achieved by skipping some CU processing based on past frames coding information and neighbor CU coding results. The PU-level decision is not considered in this work.

3. MVM fast PU decision method

The HM 3.4 implements and optimization that evaluates a particular set of PU partitions depending on whether the current CU is in the maximum depth (leaf node in the quadtree) or not. When non-leaf CUs are being processed, ME is performed for the*2Nx2N*, *2NxN* and *Nx2N* partitions. In case of leaf CUs, every possible partition is evaluated. Naturally this process requires a hugeamount of calculations to be carried out, and this is extremely unwanted considering real-time and energy constraints. Therefore, solutions that aim to reduce the number of ME operations are very important to make real-time HEVC encoding a feasible task. The Motion Vectors Merging algorithm described in this work was designed with such goal in mind. In order to accomplish this, a heuristic is applied to decide all PU sizes above *NxN*. When certain conditions are met, *NxN* partitions are merged into *2NxN*, *Nx2N* or *2Nx2N* partitions without performing ME operations for each one of the larger candidate PU sizes. If the conditions are not met, then *NxN* partitions are selected. It is important to emphasize that *2Nx2N* PUs must take priority in the event of a tie, since these partitions will produce a lower bitrate. This is explained by the fact that greater PUs require less motion information per pixel, such as motion vectors and reference frame index.

Fig. 2 (a) shows the proposed fast PU decision and (b) the heuristic function used in this solution. For each CU, the ME is initially performed for all four *NxN* partitions. With the vectors produced from ME, the heuristic function is applied to decide which partitions can be merged. The conditions that drive the merge decision are based on the vector similarity among the neighboring PUs. The PU decision tree is evaluated in a top-down manner. If all four PUs produced the same motion vectors, then *2Nx2N* is considered the best partition and ME is performed for that size only. When *2Nx2N* does not meet the restrictions, the rectangular partitions are considered following the same steps, i.e., if two neighboring PUs produced the same motion vectors, then the corresponding PU size will be used and ME is applied to produce the best motion vector for that PU size. Lastly, when all vectors are different, *NxN*partitions are selected and no further ME operations must be performed. It is important to notice that ME must be applied for the PU partitions that are selected via heuristic decision in order to achieve better RD results. In doing so, motion vectors that are different from the ones that were used to decide whether PUs are merged can be produced.

(a)

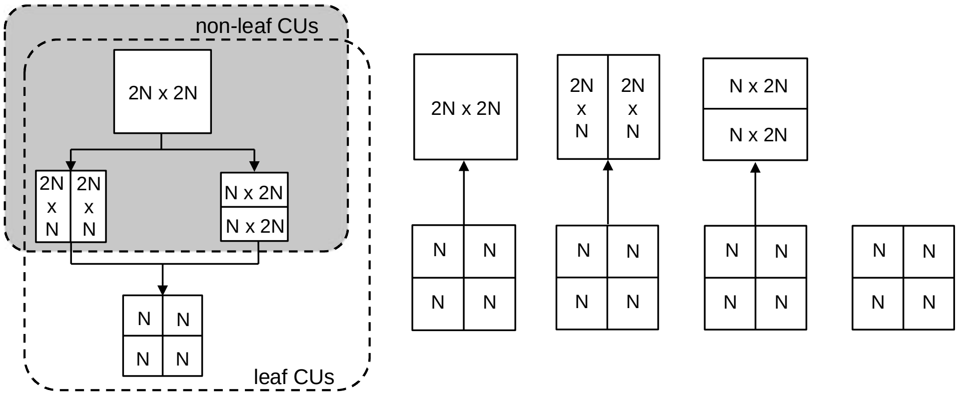
```
Proposed Fast Inter PU Decision Algorithm
A ← motion vector block 0
B ← border between blocks 0 and 2 of a CU
C ← border between blocks 1 and 3 of a CU
D ← border between blocks 2 and 3 of a CU
function checkInterBestPU( cu ):
    //four motion vectors for each block in NxN PU partition
    bestMV ← motionEstimation(cu, NxN)
    A ← (bestMV[0] == bestMV[1])
    B ← (bestMV[0] == bestMV[2])
    C ← (bestMV[1] == bestMV[3])
    D ← (bestMV[2] == bestMV[3])
    bestPartSize ← mergeBorders(A, B, C, D)
    resultsME ← motionEstimation(cu, bestPartSize)
    return calcRDCost (resultsME)
```

(b)

```
Borders Merging Heuristic
function mergeBorders( A, B, C, D ):
    if (A.B) + (B.D) + (A.C) + (C.D) then
        partSize ← 2Nx2N
    else if (A+D) then
        partSize ← 2NxN
    else if (B+C) then
        partSize ← Nx2N
    else
        partSize ← NONE
    return partSize
```

Figure 2. (a)Fast Inter PU decision algorithm and (b) border merging algorithm

This process saves ME operations, because when a particular *NxN* PU is merged into a greater partition, all other candidates are automatically discarded. This is clearly illustrated when fig.3(a) and (b) sections are compared: the original approach evaluates a greater number of partitions even when non-leaf nodes are processed, while the MVM evaluates, at most, two kinds of partitions.



(a) (b)

Figure 3. Partitions evaluated using the (a) RD approach and (b) the MVM algorithm with its 4 possible cases

Other heuristics with different conditions can be applied in order to achieve better results in terms of complexity or video quality. Heuristics that define more flexible conditions will accept a greater number of merges, which will definitely reduce complexity, but this will most likely bring negative impacts to video quality. The reverse applies for heuristics that establish more strict conditions, which will probably produce better quality results, but lack in terms of complexity decreasing.

4. Results and comparisons with related work

Tab.1 presents the results of rate-distortion evaluations for two video sequences: BQTerrace and Cactus [6]. The QP parameters were assigned to 22, 27, 32 and 37, as specified in the common test conditions document [7]. For all cases, the PNSR drop and the bitrate increase are negligible. The average results when using the Full Search algorithm show a PSNR drop of only 0.08 dB with a bitrate increase of 1.37% on average. When the TZ Search is adopted, similar behavior is detected, increasing only in 1.91% the bitrate and with a 0.06 dB of PSNR drop. The rate-distortion results show that the proposed MVM fast decision method is highly efficient when compared with the RDO implementation. For all cases, the RD results of the MVM is almost equal to the best corner case implementation, as presented in tab. 1.

Table 1. Related Works Comparison

QP	PSNR (dB)		Bitrate (kbps)	
	RDO Full	Proposed	RDO Full	Proposed
22	38.3865	38.33	38631	39000
27	36.26	36.21	13772	14041
32	34.0579	33.99	6091	6179
37	31.8876	31.8135	2883	2916

The MVM was integrated into the HM 3.4 test model. All rate-distortion evaluations were performed under the following settings: (a) low complexity configuration, (b) IPPP prediction structure, (c) search window dimension [±64] and (d) two search algorithms (Full Search and TZ Search).The results show that the proposed MVM fast inter PU decision method accelerates the encoding time in 35%. This optimization can be even higher when a more sophisticated heuristic is inserted in the inter-prediction module. These results were expected, since most part of the encoding effort is spent on the inter-prediction, more specifically in the ME step, and these are the modules that the proposed algorithm targets to optimize. Besides, this encoding time reduction is achieved with minimum rate-distortion loss.

Tab. 2 presents the comparison with related works. There are no published works that perform fast decision in PU level. All works assume that the full RDO is applied for all PU partition sizes. This way, the comparison will assume the CU-level heuristic algorithms that aggressively avoid some CU depths processing in order to save computation. The work of [3] performs several complexity target analyses. The comparison in table 1 considers only the 60% target, which is the nearest possible scenario that can provide a fair comparison.

Table 2. Related Works Comparison

	MVM (FS)	**Corrêa [3] (target 60%)**	**Leng [4]**
Specs.	PU-level	CU-level	CU-level
∆PSNR (average)	-0.08 dB	-0.01 dB	-0.04 dB
∆Bitrate (average)	1.37%	1.26%	-0.01%
∆Execution Time	-35%	-38%	-44%

5. Conclusion and Future Works

This paper proposed the Motion Vectors Merge algorithm, a novel fast heuristic decision for the inter-prediction PU trees of the emerging HEVC standard. MVM relies on saving the number of ME operations performed in each PU tree by merging *NxN* PUs into larger ones without applying the RDO approach to make this decision. When PUs are merged through the heuristic decision, ME is applied for the merged PUs in order to achieve better RD costs.

Experimental results point an execution time reduction of 35% on average by just acting in the inter-prediction in PU level. This speed-up is achieved with insignificant losses in the rate-distortion encoding results. When compared to other related work found in the literature, MVM proves to have achieved a significant complexity reduction with negligible video quality loss.

As future works, different heuristics will be elaborated intending to find better complexity reduction at the same video quality. In addition, hardware implementations of the MVM will be designed to estimate the resources overhead introduced by this solution in HEVC hardware encoders.

6. References

[1] JVT (T. Wiegand, G. Sullivan, A. Luthra), Draft ITU-T Rec. and final draft int. stand. of joint video spec. (ITU-T Rec.H.264|ISO/IEC 14496-10 AVC), 2003.

[2] JCT-VC Group,available at<http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx>,December 2011.

[3] Corrêa, Guilherme; Assuncao, Pedro; Agostini, Luciano; Cruz, Luis A. S.. “Complexity Control of High Efficiency VideoEncoders for Power-Constrained Devices”, IEEE Transactions on Consumer Electronics, v. 57, pp. 1866-1874, November 2011.

[4] Leng, Jie; Sun, Lei; Ikenaga, T.; Sakaida, S.."Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC", CMSP, v. 1, no., pp.56-59, 14-15 May 2011.

[5] ISO/IEC-JTC1/SC29/WG11, "HEVC Reference Software Manual," ed. Geneva, Switzerland, 2011.

[6] InstitutfürInformationsverarbeitung: Test media, available at <ftp://ftp.tnt.uni-hannover.de/>, January 2012.

[7] Common testconditionsand software referenceconfigurations, availableat<http://phenix.int-evry.fr/jct/>, February 2012.