A Fast and Low-complexity Motion Estimation for HEVC

김성오, 박찬식, 전형주, 김재문
삼성전자 DMC 연구소 Multimedia 연구팀
sungoh77.kim@samsung.com, chansik.park@samsung.com, hyungju.chun@samsung.com,
jaemoonc.kim@samsung.com

A Fast and Low-complexity Motion Estimation for HEVC

Sungoh Kim, Chansik Park, Hyungju Chun, Jaemoon Kim
Multimedia research team, DMC R&D center, Samsung Electronics

Abstract

In this paper, we propose a fast and low-complexity Motion Estimation (ME) algorithm for High Efficiency Video Coding (HEVC). Motion estimation occupies 77~81% of the amount of computation in HEVC. After all, the main key of codec implementation is to find a fast and low-complexity motion estimation algorithm and architecture. The proposed algorithm uses only 1% of the amount of operations compared to full search algorithm while maintaining compression performance with slight loss of 0.6% (BDBR).

1. Introduction

High Efficiency Video Coding (HEVC) is a new video compression standard providing 50% coding gain over H.264/AVC. HEVC is finalized by the Joint Collaborative Team on Video Coding (JCT-VC) established by ISO-IEC/MPEG and ITU-T/VCEG in Jan. 2013.[1] In order to achieve this coding gain it includes an enlarged coding unit (called as tree block) of 64x64 and an increased interpolation filter tap of 8 and 4 for luma and chroma data, respectively. The most noticeable difference from the previous codec is to increase coding unit size and type; 64x64, 64x32, 32x64, 32x32, 32x16, 16x32. As the result of this distinction, the complexity of motion estimation increases more than 3~4 times.

Fig. 1 shows the run-time portion of each tool in HEVC. Motion estimation named by inter prediction occupies 77~81% of the amount of computation. After all, the main key of codec implementation is to find a fast and low-power motion estimation algorithm and architecture, which spends the most time in codec.

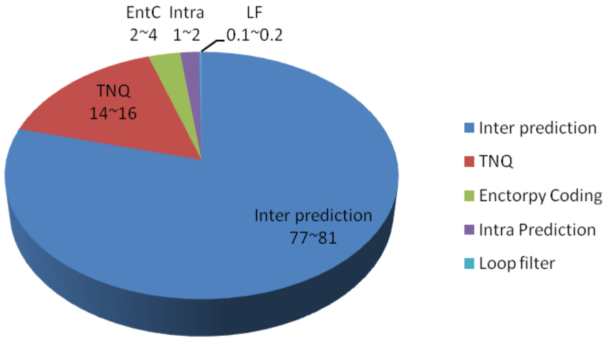


Fig. 1. The run-time portion of each tool in HEVC

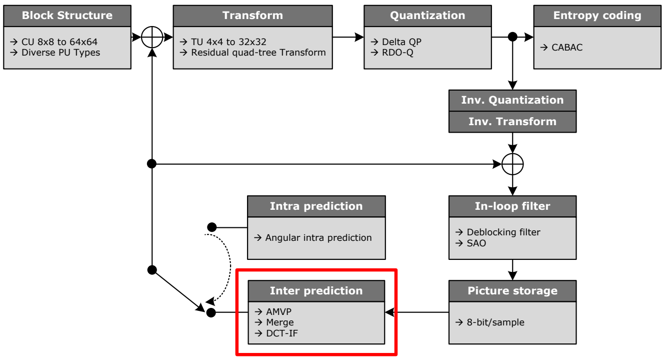


Fig. 2. Block diagram of HEVC

2. Previous Algorithm

Fig. 2 illustrates the block diagram of HEVC. Each frame is divided into square blocks called Coding Units (CUs) with maximum size of 64x64 and recursively subdivided into square blocks up to 8x8. The CUs are assigned to a quad-tree where each CU is sub-divided into quad-tree based prediction blocks called Prediction Units (PUs) of either intra or inter or skip type. Each PU is again partitioned into quad-tree based transform blocks called Transform Units (TUs) specifying transform size. Motion estimation is the process of finding the best current matched block in a search window of the reference frame. The motion estimation uses Sum of Absolute Difference (SAD), Lagrangian multiplier for the rate of motion vector difference. [2]

Fast motion estimation algorithms have been studied for a long time from MPEG-2 to H.264; Significant reductions in search positions, simplification of matching criterion, bit-width reduction, predictive search, hierarchical search and fast full search approach. Firstly, significant reduction methods in search positions include Three Step Search (TSS) [3], Two-dimensional

Logarithmic Search (TLS) [4], orthogonal search and Diamond Search (DS) [5-8]. Secondly, simplification techniques of matching criterion are pixel selection pattern in Chan's scheme, pixel difference classification (PDC) [9] and minimax criterion [10]. Thirdly, predictive search method is the way to search the surrounding Motion Vector Predictor (MVP). Finally, Hierarchical search uses sub-sampled domain search. In addition, there are the algorithm combining predictive search and hierarchical search method named by Multi-Resolution Motion Estimation algorithm with Multiple Candidates (MRMCS). [11]

3. Proposed Algorithm

We propose three kind of fast motion estimation techniques; 1) motion vector estimation method of large block types 2) Adaptive Multi-Resolution and Multi-Candidates (AMRMCS) 3) sad calculation using bit-width reduction.

Firstly, motion vector estimation method of large block type is applied. Fig. 3 describes this technique. The fundamental idea is based on the fact, statistically proven, that the motion vectors of the small block types are similar to those of the large block types. From this perspective, we calculate only up to 16x16 block type and estimate the motion vectors of large block. This method can reduce 60~70% of computation with slight loss.

Secondly, we propose the hierarchical search and predictive search based on Adaptive Multi-Resolution and Multi-Candidates (AMRMCS). Fig. 4 depicts AMRMCS method. It is divided into eight area of 4:1 sub-sampled domain and search and get the best candidates at each division first to avoid local minimum problem. After that, we can get the minimum SAD roughly in 4:1 global search range of [±128, ±64]. Then, we find final motion vector by executing refinement search in local search range (X-axis: [-8, 7], Y-axis: [-7, 7]), which uses the best motion vectors of sub-sampled domain and predictive motion vectors (MVA, MVB, MVP).

Finally, Bit-width reduction is a way of using truncated reference and current pixel data. Then, the SAD is calculated by only MSB-part (4-bit) as in Fig. 5. We can reduce the hardware by half and power consumption significantly.

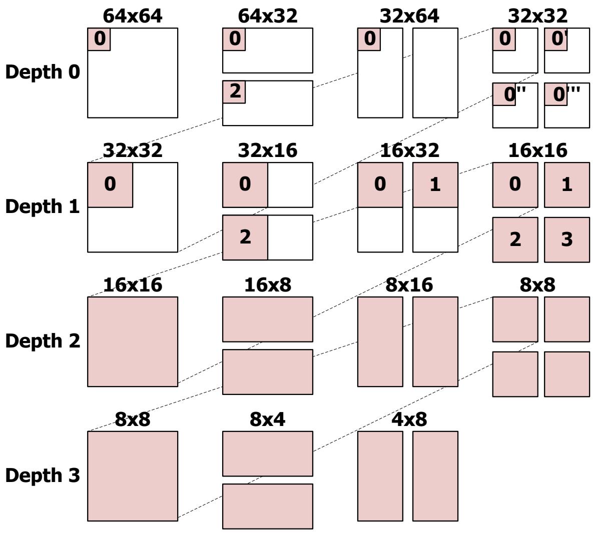
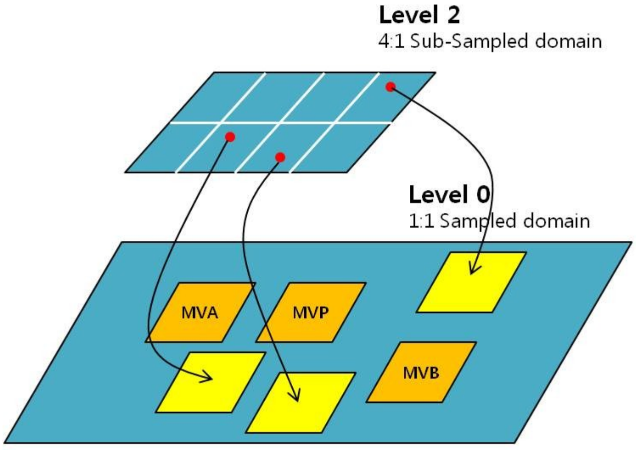


Fig. 3. Motion vector estimation method of large block types



MVA: Left 16x16 motion vector
MVB: Upper 16x16 motion vector
MVP: Motion vector predictor

Fig. 4. Concept of AMRMCS

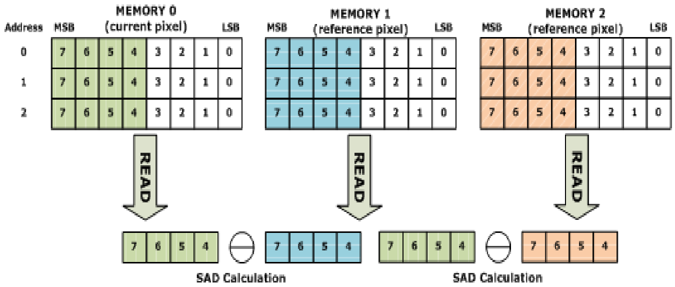


Fig. 5. SAD calculation using bit-width reduction

4. Experimental Result

We use Class A~ Class F sequences, Call For Paper (CFP) sequences, including UHD, FHD, VGA etc. and the latest HEVC reference software, HM10.0. Table 1. Shows Comparison with Full Search Algorithm in HM10.0. Proposed algorithm is occurred less loss of 0.6% (BDBR [12]). When compared to previous schemes, we achieved far superior performance while maintaining coding efficiency and visual quality.

Table 1. Comparison with Full Search Algorithm in HM10.0
Anchor: H.264 HW CODEC

	EPZS (in HM)	Our Algorithm
BDBR[%]	-41.8	-41.2

5. Conclusion

Our proposed motion estimation uses only 1% of the amount of operations compared to the general full search algorithm while maintaining compression performance. And we represented that our motion estimation is definitely superior to previous scheme. Finally, the proposed motion estimation algorithm will help you to have the smallest G/C and the lowest power consumption among existing motion estimation engines for HEVC.

References

[1] B. Bross, et al, “High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Last Call)”, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG Doc. JCTVC-L1003 (2013)

[2] Purnachand, et al, “Fast Motion Estimation Algorithm for HEVC”, (ICCE-Berlin), 34 – 37 (2012)

[3] T. Koga, et al, “Motion compensated inter-frame coding for video conferencing,” in Proc. Nat. Telecommun. Conf., C9.6.1–C9.6.5 (1981)

[4] J. Jain and A. Jain, “Displacement Measurement and its Application in Internal Image Coding,” IEEE Trans. Commun., vol.COM-29, no. 12, 1799–1808 (1981)

[5] J.Y. Tham, et al, “A Novel Unrestricted Center-biased Diamond Search Algorithm for Block Motion Estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 8, no. 4, 369–377 (1998)

[6] S. Zhu and K.K. Ma, “A New Diamond Search Algorithm for Fast Block-matching Motion Estimation,” IEEE Trans. Image Processing, vol. 9, no. 2, 2000, pp. 287–290.

[7] A.M. Tourapis, et al, “Optimizing the mpeg-4 Encoder - advanced Diamond Zonal search,” in Proc. of IEEE Int. Symp. Circuits Syst. (ISCAS’00), 674–677 (2000)

[8] A.M. Tourapis, O.C. Au, and M.L. Liu, “Highly Efficient Predictive Zonal Algorithms for Fast Block-matching Motion Estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 10, 934–947 (2002)

[9] H. Gharavi and M. Mills, “Block Matching Motion Estimation Algorithms - New Results,” IEEE Trans. Circuits Syst., vol. 37, no. 5, 649–651 (1990)

[10] M.J. Chen, et al, “A New Block-matching Criterion for Motion Estimation and its Implementation,” IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 3, 231–236 (1995)

[11] Lee, J. H. et al, “A fast multi-resolution block matching algorithm and its LSI architecture for low bit-rate video coding”Circuits and Systems for Video Technology, IEEE Transactions on Vol. 11, 1289-1301 (2001)

[12] G. Bjontegaard, "Calculation of Average PSNR Differences Between RD-curves," document VCEG-M33, ITU-T Video Coding Experts Group (VCEG) Meeting, Austin, TX (2001)