A Fast and Low-complexity Motion Estimation for UHD HEVC

Sungoh Kim\(^a\), Chansik Park\(^a\), Hyungji Chun\(^a\), and Jaemoon Kim\(^a\)

Abstract

In this paper, we propose a novel fast and low-complexity Motion Estimation (ME) algorithm for Ultra High Definition (UHD) High Efficiency Video Coding (HEVC). Motion estimation occupies 77–81% of the amount of computation in HEVC. After all, the main key of video codec implementation is to find a fast and low-complexity motion estimation algorithm and architecture. We analyze the previous motion estimation algorithms and propose three optimal algorithm to reduce the computation proportion for HEVC. The proposed algorithm uses only 0.36% of the amount of operations compared to full search algorithm while maintaining compression performance with slight loss of 1.1%.

Keyword: HEVC, Motion Estimation, Low-power, Low-complexity, Fast Algorithm

I. Introduction

High Efficiency Video Coding (HEVC) is a new video compression standard providing about 50% coding gain over H.264/AVC [1]. HEVC was finalized by the Joint Collaborative Team on Video Coding (JCT-VC) established by ISO-IEC/MPEG and ITU-T/VCEG in Jan. 2013 [2]. In order to achieve this coding gain, it includes an enlarged coding unit (called as tree block) of 64x64 and an increased interpolation filter taps of 8 and 4 for luma and chroma data, respectively. The most noticeable differences
from the previous codec are coding unit size (e.g., 16x16 to 64x64) and a variety of hierarchical block partitioning. Consequently, the complexity of motion estimation (ME) of HEVC increases more than 3-4 times than that of H.264/AVC.

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 video codec implementation into a hardware is to find a fast and low-power motion estimation algorithm and architecture, which spends the most time in codec.

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-depth reduction, predictive search, hierarchical search and fast full search approach. First, 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]. Second, simplification techniques of matching criterion are pixel selection pattern in Chan's scheme, pixel difference classification (PDC) [9] and minimax criterion [10]. Third, 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]. Some of the important algorithms that are implemented in H.264/AVC reference software are UMHexS (Unsymmetrical-cross Multi-hexagon-grid Search), SUMHexS (Simple UMHexS) and EPZS (Enhanced Predictive Zonal Search). TZS (Test Zone Search) is also used in encoder of HEVC reference software[12]. However, those previous algorithms for a variety of block sizes and partition types to perform ME were not suitable. Thus, we propose three optimized scheme for HEVC.

This paper is organized as follows. In Section II, our proposed algorithm is presented. Experimental results are presented in Section III. and concluding remarks are given in Section IV.

II. Proposed Algorithm

Fig. 2 illustrates example of the partition units in 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) with intra or inter type. Each PU is again partitioned into quad-tree based transform blocks called Transform Units (TUs) specifying transform size [13]. Because of a variety of block sizes and partition types in HEVC, the complexity of ME can be extremely increased. Therefore, a new fast algorithm is required for HEVC.

To develop a new ME algorithm for HEVC, we analyzed encode bitstream. Fig. 3 shows the screen shot of the analyzer, which consists of block partition information, motion vector information, prediction mode distribution, the total bit amount of CU, distortion and cost. This tool should be useful to find the optimum algorithm easily.
We propose three 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-depth reduction.

Firstly, This method predicts the best motion vectors of large block types, 16x16 to 64x64, as discussed in Fig. 4. The top-left best motion vector (MV) of 16x16 block type is used by the best MVs of large block types. The fundamental idea is based on the fact, statistically proven, that the MVs of the small block types are similar to those of the large block types. Fig. 5 shows the best MV in 16x16 block type. Fig. 6 also depicts the best MV in 64x64 block type. As can be seen, two MVs may be alike. From this
Secondly, we propose the hierarchical search and predictive search based on AMRMCS as shown in Fig. 7. AMRMCS consists of two hierarchical levels, i.e., upper and lower levels. For a search range of \([-p, +p]\), in the upper level (level 2), images are 4:1 downsampled from the original in the horizontal and vertical direction and then a motion search is performed for a 4x4 sub-block with a search range of \([-p/4, +p/4]\). From the upper level search, three search points corresponding to the smallest SADs (Sum of absolute differences) are selected as initial search points for the lower level (level 0). And it is divided into eight areas of 4:1 sub-sampled domain and search the best candidates at each division first and get some of them whose numbers are flexible by image resolution, bandwidth and configuration to avoid local minimum problem in level 2. After that, we can get the minimum SAD candidates in 4:1 global search range of \([-\pm 128, \pm 64]\). We find final MV by executing refinement search again in local search range (X-axis: \([-8, 7]\), Y-axis: \([-7, 7]\)), which uses the best MVs of sub-sampled domain and MV of the left neighbor (MVA), MV of the top neighbor (MVB) and motion vector predictor (MVP). This predictor is calculated by using the median value of the MVs of the three adjacent, on the left, top, and top-right (or top-left if top-right is not available) perspective, we find the best motion vector only up to 16x16 block type and predict the best MVs of large blocks. This method can reduce 60-70% of computation with slight loss.
blocks to the current block.

Finally, bit-depth reduction is a way of using truncated reference and current pixel data. Fig. 8 shows comparison of average Bjøntegaard delta bit rate (BDBR) [14] by bit-depth reduction. The x-axis represents the number of bit-depth and the y-axis shows a set of measures such as BDBR.

As this experimental result, even though bit-depth is reduced from 8 to 4-bit, the performance degradation is only less than 0.5 % (BDBR). Therefore, our proposed algorithm uses the 4-bit pixel data for motion estimation computation. Then, the SAD is calculated by only MSB-part (4-bit) as in Fig. 9. Using this algorithm, we can reduce the hardware by half and power consumption significantly.

Fig. 10 shows the amount of computational complexity for each algorithm. It means the combined proportion between the number of search points and the quantity of SAD calculation. As a result of this figure, our proposed algorithm uses only 0.36% of the amount of operations compared to the general full search algorithm.
III. Experimental Result

We used video test sequences, including UHD crop (2560x1600), FHD (1920x1080) and so on. The test sequence depicted in Fig. 11, 1920x1080 size with 30 frame-per-second, has various moving objects along with global motion (e.g., camera zoom in/out and object tracking). Fig. 12 illustrates difference of MVs between full search algorithm and our proposed algorithm. The 64x64 coding unit position of white thick square is (11, 5) and the coding unit size (yellow thick square) is 8x8 type. Then, the best MV of the two algorithm is equal...
to (-7, 13). Therefore, it can be confirmed that our proposed algorithm has a similar performance compared to full search algorithm which cab be the most complex.

Table I and II show comparison with the full search algorithm in HM10.0. The proposed algorithm achieved far superior performance in speed, power and cost than any legacy codecs while maintaining visual quality with loss of a bit coding efficiency.

<table>
<thead>
<tr>
<th>Anchor: Full Search (in HM: TZS Off)</th>
<th>Unit: BDBR[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Name</td>
<td>(1) (2) (3) Total</td>
</tr>
<tr>
<td>Kimono1 (24Hz)</td>
<td>0.2 0.5 0.4 1.4</td>
</tr>
<tr>
<td>ParkScene (24Hz)</td>
<td>0.0 0.3 0.1 0.7</td>
</tr>
<tr>
<td>Cactus (50Hz)</td>
<td>0.1 0.3 0.9 1.1</td>
</tr>
<tr>
<td>BasketballDrive (50Hz)</td>
<td>0.2 0.6 0.7 1.8</td>
</tr>
<tr>
<td>BQTerrace (60Hz)</td>
<td>0.1 0.4 0.2 0.6</td>
</tr>
<tr>
<td>QuarterBackSneak (30Hz)</td>
<td>0.0 0.2 0.5 0.9</td>
</tr>
</tbody>
</table>

(1): motion vector estimation of large block types
(2): AMRMCs
(3): SAD Calculation using bit-depth reduction

Table 2. Comparison with Full Search Algorithm in HM10.0 (Total sequence included FHD, UHD, VGA etc)

<table>
<thead>
<tr>
<th>Anchor: Full Search (in HM: TZS Off)</th>
<th>Unit: BDBR[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Average</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Our proposed motion estimation uses only 0.36% 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 hardware size and the lowest power consumption among existing motion estimation engines for HEVC.

**References**


김 성 오

- 2008년 ~ 현재 : 삼성전자 DMC 연구소 Multimedia 연구팀 책임 연구원
- 주관심분야 : 비디오 코덱, 신호처리

박 찬 식

- 2002년 ~ 현재 : 삼성전자 DMC 연구소 Multimedia 연구팀 책임 연구원
- 주관심분야 : 비디오 코덱, 신호처리

진 형 주

- 2008년 ~ 현재 : 삼성전자 DMC 연구소 Multimedia 연구팀 신입 연구원
- 주관심분야 : 비디오 코덱, 신호처리

김 재 문

- 2010년 ~ 현재 : 삼성전자 DMC 연구소 Multimedia 연구팀 책임 연구원
- 주관심분야 : 비디오 코덱, 신호처리