

# DCT-Based Quarter-pel Interpolation: A Frontier Method for ME in HEVC

Kalyan Kumar Barik<sup>1</sup>, Manas Ranjan Jena<sup>2,\*</sup>, Asutosh Das<sup>1</sup>

<sup>1</sup>Department of ETC, SIET, DHENKANAL, ODISHA

<sup>2</sup>Department of ETC, Department of ELTCE, VSSUT, BURLA, ODISHA

\*Corresponding author: [manas.synergy@gmail.com](mailto:manas.synergy@gmail.com)

Received June 03, 2014; Revised June 13, 2014; Accepted June 16, 2014

**Abstract** The new demands for high-definition and high resolution video applications are pushing the development of new techniques in the video coding area. Motion estimation (ME) is one of the key elements in video coding standard which eliminates the temporal redundancies between successive frames. In the recent international video coding standards, sub-pixel ME is proposed for its excellent coding performance. DCT (Discrete Cosine Transform) has been widely used for digital signal processing and multimedia applications, especially video codec. In regards to interpolation for motion estimation and compensation, the conventional HEVC employs 8-tap and 4-tap filters for luma and chroma samples respectively. Coefficients in such filters are determined by DCT. This paper presents and implements the DCT-based Quarter-pixel Interpolation filter for motion estimation and compression of HEVC standard using Matlab.

**Keywords:** HEVC, DCT, ME, quarter pixel interpolation, motion compensation

**Cite This Article:** Kalyan Kumar Barik, Manas Ranjan Jena, and Asutosh Das, "DCT-Based Quarter-pel Interpolation: A Frontier Method for ME in HEVC." *American Journal of Systems and Software*, vol. 2, no. 3 (2014): 60-64. doi: 10.12691/ajss-2-3-2.

## 1. Introduction

With the irresistible trend of high-definition video, the next generation video codecs will be expected to achieve at least 4kx2k Quad Full High Definition resolution for ultra-high definition [1]. High efficiency video coding (HEVC) is currently the latest video coding standard for ITU-T Video Coding Experts Group and ISO/IEC Moving Picture Experts Group. The HEVC project was launched to archive a major saving in bit rate relative to prior standards, e.g, reduction by amount half [2]. The main goal of HEVC standardization effort is to enable significantly improved compression performance relative to the existing standards- in the range of 50% bit-rate reduction for equal perceptual video quality [5].

Motion Estimation (ME) is a technique used by video coders to reduce the amount of information transmitted to a decoder by exploiting the temporal redundancy present in the video signal [11]. In ME the picture to be coded is first divided into block, an encoder searches reference pictures to find its best matching block. The best matching block is called the prediction of the corresponding block and the difference between the original and the prediction signal is coded by various means, such as transform coding, and transmitted to a decoder. The relative prediction with respect to original block is called a motion vector and it is transmitted to the decoder along with the residual signal. The true displacements of moving objects between pictures are continuous and don't follow the

sampling grid of the digitized video sequence. Hence, by utilizing fractional accuracy for motion vectors instead of integer accuracy, the residual error is decreased and coding efficiency of video coders is increased. If the motion vector has a fractional value, the reference block needs to be interpolated accordingly. The interpolation filter used in video coding standards are carefully designed taking into account many factors, such as coding efficiency, implementation complexity and visual quality [7].

## 2. Principles of the Quarter-pixel Interpolation in HEVC

HEVC introduces several new features including newly designed interpolation filters for both Luma and Chroma as well as high-accuracy motion compensation process which is mostly free from rounding errors for both uni- and bi-directional predictions. The key differences between H.264/AVC and HEVC interpolation can be summarized as:

**Re-designed Luma and Chroma interpolation filter:** to improve the filter responses in the high frequency range, both Luma and Chroma interpolation filters are re-designed. Luma interpolation process uses symmetric 8-tap filter for half-pel positions while asymmetric 7-tap filter is used for quarter-pel positions to minimize the additional complexity of the motion compensation process. For Chroma samples, a 4-tap filter is introduced [3].

**Non-cascaded process for quarter-pel positions:** rather than averaging two neighbouring samples, HEVC directly derives quarter-pel samples by applying two one-dimensional filters similar to the half-pel center position in H.264/AVC. Since it is consistent for all quarter-pel positions, the inconsistency issues for different quarter-pel positions in H.264/AVC no longer exist in HEVC.

**High-accuracy motion compensation operation:** in HEVC, intermediate values used in interpolation are kept at a higher accuracy. In addition, the rounding of two prediction blocks used in the bi-directional prediction is delayed and merged with the rounding in the bi-directional averaging process. It should be noted that HEVC interpolation process guarantees that no 16 bits overflow at any intermediate stage happens by controlling the accuracy according to the source bit-depth.

Quarter-pixel interpolation is one of the most computationally intensive parts of High Efficiency Video Coding (HEVC) video encoder and decoder. In H.264 standard, a 6-tap FIR filter is used for half-pixel interpolation and a bilinear filter is used for quarter-pixel interpolation [4]. In HEVC standard, 3 different 8-tap FIR filters are used for both half-pixel and quarter -pixel interpolations. In H.264, 4x4 and 16x16 block sizes are used. However, in HEVC, prediction unit (PU) sizes can be from 4x4 to 64x64. Therefore, HEVC sub-pixel interpolation is more complex than H.264 sub-pixel interpolation.

The samples of the PB for an intra-picture-predicted CB are obtained from those of a corresponding block region in the reference picture identified by a reference picture index, which is at a position displaced by the horizontal and vertical components of the motion vector. Except for the case when the motion vector has an integer value, fractional sample interpolation is used to generate the prediction samples for non-integer sampling positions. [11] As in H.264/MPEG-4 AVC, HEVC supports motion vectors with units of one quarter of the distance between luma samples. For chroma samples, the motion vector accuracy is determined according to the chroma sampling format, which for 4:2:0 sampling results in units of one eighth of the distance between chroma samples.

Finite impulse response (FIR) filters are used for both Luma and Chroma interpolation in HEVC. The coefficients of the FIR filter are designed using Fourier decomposition of discrete cosine transform. The DCT-based filter coefficient for luma and chroma fractional sample interpolation of HEVC are given in Table 1 and Table 2 respectively [6].

**Table 1. Filter coefficients for luma fractional sample interpolation in HEVC[5]**

Index[i]	filterl(i)	N
hfilter[i]:1/4	{-1, 4, -10, 58, 17, -5, 1}	8.7
qfilter[i]:1/2	{-1, 4, -11, 40, 40, -11, 4, -1}	9.5

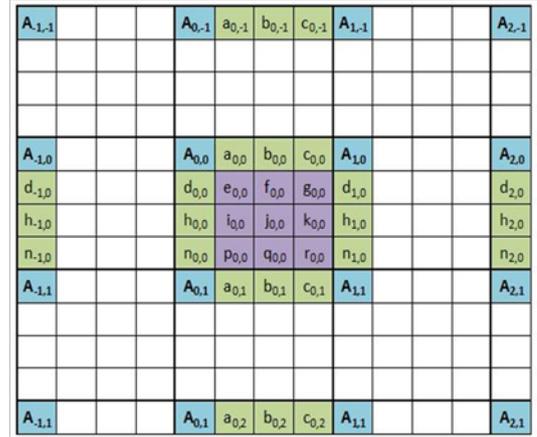
**Table 2. Filter coefficients for chroma sample interpolation in HEVC [5]**

index	filterl(i)	$\sigma$
filter1[i]:1/8	{-2, 58, 10, -2}	0.012
filter2[i]:1/4	{-4, 54, 16, -2}	0.016
filter3[i]:3/8	{-6, 46, 28, -4}	0.018
filter4[i]:1/2	{-4, 36, 36, -4}	0.020

Where N and  $\sigma$  indicates regularization parameters for Luma and chroma respectively.

DCT can be derived to be any tap filters. It is clear that the filters with more tap tend to provide more accurate interpolation results (tend to high efficiency) but the interpolation complexity will be increased. So for HEVC 8-tap and 7-tap DCTIF(DCT-based interpolations) are used for Luma samples instead of 12-tap and which shows a good trade-off between complexity and performance for both HE (high efficiency) and LC (low complexity) cases.

## 2.1. Interpolation Process of Luma Sample



**Figure 1. Integer and fractional sample positions for luma interpolation [2]**

In Figure-1 the positions labeled with upper-case letters  $A_{i,j}$ , represent the available luma samples at integer sample locations, whereas the other positions labeled with lower-case letters represent samples at non integer sample locations, which need to be generated by interpolation.

The samples labeled  $a_{0,0}$ ,  $b_{0,0}$ ,  $c_{0,0}$ ,  $d_{0,0}$ ,  $h_{0,0}$ , and  $n_{0,0}$  and are derived from the samples by applying the eight-tap filter for half-sample positions [5] and the seven-tap filter for the quarter-sample positions as follows:

$$e_{0,0} = \left( \sum_{v=-3.3} a_{0,v} \text{qfilter}[v] \right) \gg (B-8)$$

$$a_{0,j} = \left( \sum_{i=-3.3} A_{i,j} \text{qfilter}[i] \right) \gg (B-8)$$

$$b_{0,j} = \left( \sum_{i=-3.4} A_{i,j} \text{hfilter}[i] \right) \gg (B-8)$$

$$c_{0,j} = \left( \sum_{i=-2.4} A_{i,j} \text{qfilter}[1-i] \right) \gg (B-8)$$

$$d_{0,0} = \left( \sum_{i=-3.3} A_{0,j} \text{qfilter}[j] \right) \gg (B-8)$$

$$h_{0,0} = \left( \sum_{i=-3.4} A_{0,j} \text{hfilter}[j] \right) \gg (B-8)$$

$$n_{0,0} = \left( \sum_{i=-2.4} A_{0,j} \text{qfilter}[1-i] \right) \gg (B-8)$$

where the constant  $B \geq 8$  is the bit depth of the reference samples (and typically  $B = 8$  for most applications) and the filter coefficient values for luma is given in table-1. In these formulae  $\gg$  denotes an arithmetic right shift operation [5,8].

The samples labelled  $e_{0,0}$ ,  $i_{0,0}$ ,  $p_{0,0}$ ,  $f_{0,0}$ ,  $j_{0,0}$ ,  $q_{0,0}$ ,  $g_{0,0}$ ,  $k_{0,0}$  and  $r_{0,0}$  shall be derived by applying the 8-tap filter to the samples  $a_{0,i}$ ,  $b_{0,i}$  and  $c_{0,i}$  where  $i = -3$  to 4 in vertical direction.

$$e_{0,0} = \left( \begin{array}{c} -a_{0,-3} + 4*a_{0,-2} - 10*a_{0,-1} \\ + 58*a_{0,0} + 17*a_{0,1} - 5*a_{0,2} + a_{0,3} \end{array} \right) / 64$$

$$i_{0,0} = \left( \begin{array}{c} -a_{0,-3} + 4*a_{0,-2} - 11*a_{0,-1} + 40*a_{0,0} \\ + 40*a_{0,1} - 11*a_{0,2} + 4*a_{0,3} - a_{0,4} \end{array} \right) / 64$$

$$p_{0,0} = \left( \begin{array}{c} a_{0,-2} - 5*a_{0,-1} + 17*a_{0,0} + 58*a_{0,1} \\ - 10*a_{0,2} + 4*a_{0,3} - a_{0,4} \end{array} \right) / 64$$

$$f_{0,0} = \left( \begin{array}{c} -b_{0,-3} + 4*b_{0,-2} - 10*b_{0,-1} + 58*b_{0,0} \\ + 17*b_{0,1} - 5*b_{0,2} + b_{0,3} \end{array} \right) / 64$$

$$j_{0,0} = \left( \begin{array}{c} -b_{0,-3} + 4*b_{0,-2} - 11*b_{0,-1} + 40*b_{0,0} \\ + 40*b_{0,1} - 11*b_{0,2} + 4*b_{0,3} - b_{0,4} \end{array} \right) / 64$$

$$q_{0,0} = \left( \begin{array}{c} b_{0,-2} - 5*b_{0,-1} + 17*b_{0,0} + 58*b_{0,1} \\ - 10*b_{0,2} + 4*b_{0,3} - b_{0,4} \end{array} \right) / 64$$

$$g_{0,0} = \left( \begin{array}{c} -c_{0,-3} + 4*c_{0,-2} - 10*c_{0,-1} + 58*c_{0,0} \\ + 17*c_{0,1} - 5*c_{0,2} + c_{0,3} \end{array} \right) / 64$$

$$k_{0,0} = \left( \begin{array}{c} -c_{0,-3} + 4*c_{0,-2} - 11*c_{0,-1} + 40*c_{0,0} \\ + 40*c_{0,1} - 11*c_{0,2} + 4*c_{0,3} - c_{0,4} \end{array} \right) / 64$$

$$r_{0,0} = \left( \begin{array}{c} c_{0,-2} - 5*c_{0,-1} + 17*c_{0,0} + 58*c_{0,1} \\ - 10*c_{0,2} + 4*c_{0,3} - c_{0,4} \end{array} \right) / 64$$

## 2.2. Quarter Pixel Motion Estimation Algorithm

Different fast sub-pixel ME algorithms have been proposed, and a number of them have been adopted by the HM reference software [3]. The common idea is to simplify the search pattern by applying refined prediction algorithms, and improved adaptive threshold schemes to terminate unnecessary search positions. In an algorithm that uses the temporal information featuring the motion activities of the blocks in the previous frame has been proposed. The characteristics of the motion activities of the blocks in the current frame are predicted using this temporal information. The algorithm is shown in figure as a flowchart [9].

If the MV is not equal to the best predicted MV, the current block is classified into the special block group. Otherwise, it is an ordinary block. Then, sub-pixel refinement is carried out surrounding the best integer-pixel point. The sub-pixel refinement procedure is done in two parts: half-pixel refinement and quarter-pixel refinement [10]. For half and quarter-pixel refinements, the 4 nearest neighbours points are checked, then the 2 far neighbour points beside the best near neighbour point are checked. Before quarter-pixel refinement, decision on early termination is made. The conditions for early termination are:

1. The current block is a prejudged ordinary block.
2. The MV of its co-located block is a zero vector,
3. The MV of the current block is a zero vector after half-pixel refinement.

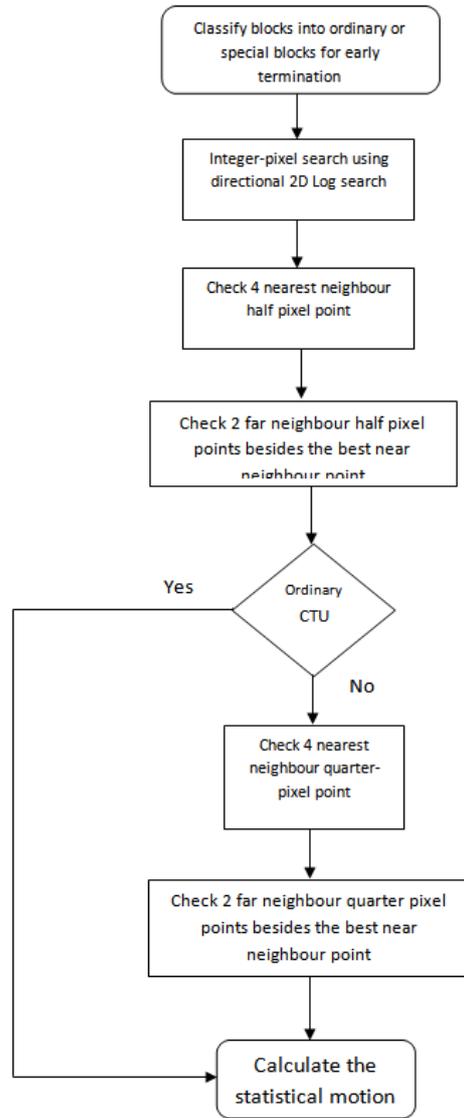


Figure 2. Flowchart of motion estimation

## 3. Simulation Results & Analysis

### 3.1. Results of interpolation and Quarter-pel ME



Figure 3. Results of interpolation

HEVC supports 7-tap and 8-tap filters for implementing the process of luma interpolation. Two interpolation results are obtained by taking a reference block size of 16x16 at location (50,50) and 32x32 at location (88x88) of Lena image as given Figure 3.

### 3.2. Results of Quarter-pixel motion estimation

In this section, to illustrate the implementation result of quarter-pixel motion estimation in HEVC, experiment have been carried out using MATLAB. Firstly, to demonstrate the results we are taking two qcif.yuv video frames (one is reference frame and second one is candidate or current frame). As discussed in previous section we implemented the algorithm to find the motion vector, predicted frame with PSNR and with motion compensation and the corresponding 3D mesh plot of residual.

The experiment has been carried out by taking size of CTU is 8X8. The reference video frame and candidate video frame is given in Figure 4.. For 8x8, CTU size the search block ze20x20.

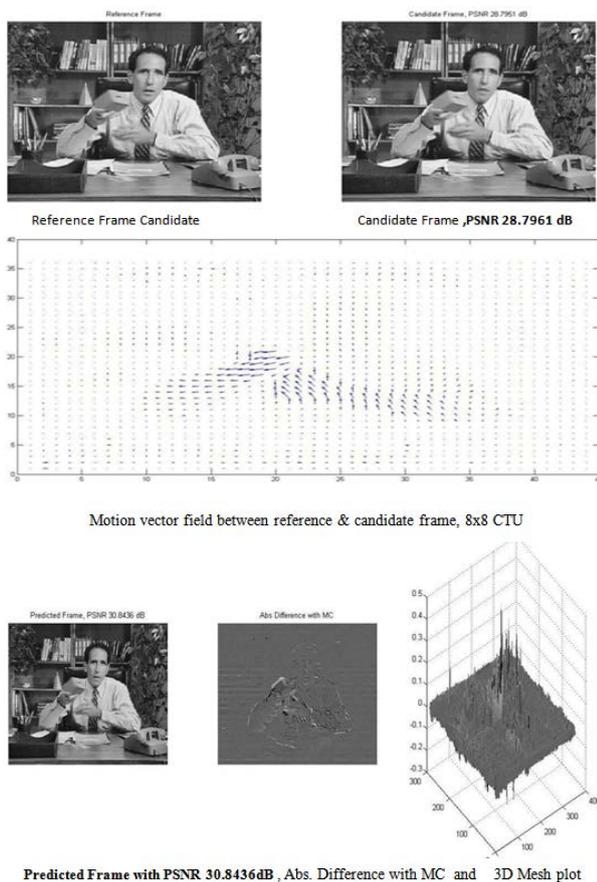


Figure 4. Results of ME

### 3.3. Comparison with DCT-based Quarter-pixel Interpolation filter of H.264/AVC

1. HEVC supports 8-tap interpolation filter for half-pixel positions and 7-tap filter for quarter-pixel positions where H.264 supports the 6-tap filter for half-pixel interpolation and the average filter for quarter-pel interpolation.

2. PSNR of predicted frame produced by the use of H.264 interpolation filter is much less in comparison to Quarter-pixel Interpolation filter of HEVC.

3. Frequency response of quarter-pel filters in HEVC are superior to those in H.264 since in the pass band the filter in HEVC are much flatter and have much smaller ripples than those in H.264/AVC.

4. The performance of the quarter-pel filters in H.264/AVC is relatively poor, especially the filters for the quarter-pel pixels e, g, p and r in the diagonal direction. In general performance gain (more than 10%) of interpolation filters in HEVC compared to H.264/AVC comes from the quarter-pel interpolations.

5. But the complexity of interpolation process in HEVC is much larger than that in H.264/AVC[12].

## 4. Conclusion

An optimized process for the quarter-pixel interpolation based motion estimation and compression in HEVC is implemented in MATLAB. According to the experimental results, this implementation of quarter-pixel interpolation based motion estimation working as like as HEVC reference software HM5.2 with the promotion of the next generation video coding standard HEVC. Quarter-pixel interpolation using 7-tap and 8-tap filter in HEVC gives more details in comparison to 6-tap filter base quarter-pixel interpolation in H.264/AVC. The computational cost of this process can be reduced utilizing parallel processing technique in the hardware implementation. To enable a parallel processing, the macro blocks (CTU) are processed on dedicated and special processor, so the all motion vectors will be outputted for an individual frame at the same time. This technique definitely reduces the time and computation complexity, but also increases the requirement of hardware resources.

## References

- [1] Jens-Rainer Ohm and G. J. Sullivan, "High Efficiency Video Coding : The Next Frontier in Video Compression," IEEE Signal Processing Magazine, pp. 153-158, January 2013
- [2] Wang Gang, C. Hexin, C. Maianshu, "A Study on Sub-pixel Interpolation Filtering Algorithm and Hardware Structural Design Aiming at HEVC," Telkomnia, Vol. 11, No. 12, pp. 7564-7570, Dec. 2013.
- [3] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, T. Wiegand, "High Efficiency Video Coding (HEVC) Text Specification Draft 7", JCTVC-I1003, May 2012.
- [4] M.T. Pourzad, C. Doutre, M. Azimi, P. Nasiopoulos, "HEVC: The New Gold Standard for Video Compression", IEEE Consumer Electronics Magazine, July 2012.
- [5] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.
- [6] Kim, et al, "Block portioning structure in the HEVC standard,"IEEE Trans. On circuits and system for video technology, vol. 22, pp. 1697-1706, Dec. 2012.
- [7] Chih-Ming Fu, Elena Alshina, A. Alshin, Y.W. Huang, C.Y. Chen, "Sample Adaptive Offset in the HEVC Standard," IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1755, Dec. 2012.
- [8] F. Bossen, Et. Al, HEVC complexity and implementation analysis," IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1685-1696, Dec 2012.

- [9] Gary J. Sullivan and Jens-Rainer Ohm. Recent developments in standardization of High Efficiency Video Coding (HEVC) volume 7798. SPIE, 2010.
- [10] T. Wiegand and G. J. Sullivan, "The H.264 video coding standard", IEEE Signal Processing Magazine, vol. 24, pp. 148-153, March 2007.
- [11] G. Sullivan, P. Topiwala and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions", SPIE conference on Applications of Digital Image Processing XXVII, vol. 5558, pp. 53-74, Aug. 2004.
- [12] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 7, July 2003.