TELKOMNIKA, Vol. 11, No. 12, December 2013, pp. 7564~7570
e-ISSN: 2087-278X

A Study on Sub-pixel Interpolation Filtering Algorithm and Hardware Structural Design Aiming at HEVC

Wang Gang[1,2], Chen Hexin[1], Chen Mianshu*[1], Liu Yuanyuan[1,3]
[1]School of Communication Engineering, Jilin University, Changchun,China
[2]School of Mechanical Engineering, Baicheng Normal College, Baicheng, China
[3]School of Information Technology, Jinlin Agricultural University, Changchun, China
*Corresponding author, e-mail: haohehe530@163.com

Abstract

Aiming at the new-generation video compression standard being formulated-HEVC, a kind of sub-pixel interpolation filtering algorithm is proposed (luminance: 1/4 precision, chrominance: 1/8 precision). Based on the algorithm, a hardware design with pipeline structure and high degree of parallelism is put forward. The hardware overhead is reduced by multiplex Wiener filter and the reduction of the size of register array. And the interpolation order of vertical priority is adopted to reduce the reading bandwidth of the storage. It is indicated from the performance analysis that this interpolation structure possesses better performance and smaller hardware overhead. This design also takes full consideration of the balance between speed and area, meeting the requirements of processing standard definition and high definition video image.

Keywords: *sub-pixel interpolation, hardware framework, HEVC, H.264*

1. Introduction

HEVC [1, 2] (High Efficiency Video Coding) is a younger-generation video coding standard being formulated by joint video coding group JCT-VT (Joint Collaborative Team on Video Coding) consisting of the experts of ISO/IEC MPEG and ITU-T/VCEG. It mainly geared toward high-definition TV (HDTV) and video coding system. The goal of HEVC is to reduce the code rate of the video streaming to 50% of H.264 standard under the condition of same peak signal to noise ratio (PSNR). With the increase of compression efficiency, the computation complexity of HEVC may be three to four times of H.264 standard, thus the complexity of coder and decoder of the hardware is also increased. Aiming at the heavy computation in interpolation process, an implementation schema of interpolation hardware structure based on pipeline is put forward in this paper.

2. Principles of the Sub-pixel Interpolation of Algorithm HEVC

2.1. Interpolation Process of Luminance Sample

Figure 1 shows the positions of the integer pixel sample, 1/2 pixel sample and 1/4 pixel sample of the luminance components of the reference image. It is supposed that luminance sample point $A_{i,j}$ is located at the position of integer sample point ($xA_{i,j}$, $yA_{i,j}$), then the predicated value from luminance point '$a_{0,0}$' to '$r_{0,0}$' at non-integer sample point positions can be obtained by the filter with the coefficient of (-1,4,-10,58,17,-5,1 , -1,4,-11,40,40,-11,4,-1), (1,-5,17,58,-10,4,-1).

(1) Horizontal and Vertical Interpolation Filtering of Integer Pixel

The values of 1/2 pixel points $b_{0,0}$, $h_{0,0}$ and 1/4 pixel point $a_{0,0}$, $c_{0,0}$;$d_{0,0}$,$n_{0,0}$ can be obtained by using the filter interpolation mentioned in the paper on the nearest integer pixel in the horizontal and vertical directions. The whole computing process is as following:

$$a_{0,0}=(-A_{-3,0}+4*A_{-2,0}-10*A_{-1,0}+58*A_{0,0}+17*A_{1,0}-5*A_{2,0}+A_{3,0})/64 \quad (1)$$

$$b_{0,0}=(-A_{-3,0}+4*A_{-2,0}-11*A_{-1,0}+40*A_{0,0}+40*A_{1,0}-11*A_{2,0}+4*A_{3,0}-A_{4,0})/64 \quad (2)$$

Received June 28, 2013; Revised August 15, 2013; Accepted September 3, 2013

$$c_{0,0}=(A_{-2,0}-5*A_{-1,0}+17*A_{0,0}+58*A_{1,0}-10*A_{2,0}+4*A_{3,0}-A_{4,0})/64 \quad (3)$$

$$d_{0,0}=(-A_{0,-3}+4*A_{0,-2}-10*A_{0,-1}+58*A_{0,0}+17*A_{0,1}-5*A_{0,2}+A_{0,3})/64 \quad (4)$$

$$h_{0,0}=(-A_{0,-3}+4*A_{0,-2}-11*A_{0,-1}+40*A_{0,0}+40*A_{0,1}-11*A_{0,2}+4*A_{0,3}-A_{0,4})/64 \quad (5)$$

$$n_{0,0}=(A_{0,-2}-5*A_{0,-1}+17*A_{0,0}+58*A_{0,1}-10*A_{0,2}+4*A_{0,3}-A_{0,4})/64 \quad (6)$$

(2) Vertical Interpolation Filtering of 1/2 Pixel and 1/4 Pixel

The values of 1/2 pixel point $j_{0,0}$ and 1/4 pixel point $e_{0,0}$,$i_{0,0}$, $p_{0,0}$, $f_{0,0}$, $q_{0,0}$, $g_{0,0}$, $k_{0,0}$and $r_{0,0}$ can be obtained by using the filter interpolation mentioned in the paper on the nearest 1/2 pixel or 1/4 pixel in vertical direction. The whole computing process is as following:

$$e_{0,0}=(-a_{0,-3}+4*a_{0,-2}-10*_{a0,-1}+58*a_{0,0}+17*a_{0,1}-5*a_{0,2}+a_{0,3})/64 \quad (7)$$

$$i_{0,0}=(-a_{0,-3}+4*a_{0,-2}-11*a_{0,-1}+40*a_{0,0}+40*a_{0,1}-11*a_{0,2}+4*a_{0,3}-a_{0,4})/64 \quad (8)$$

$$p_{0,0}=(a_{0,-2}-5*a_{0,-1}+17*a_{0,0}+58*a_{0,1}-10*a_{0,2}+4*a_{0,3}-a_{0,4})/64 \quad (9)$$

$$f_{0,0}=(-b_{0,-3}+4*b_{0,-2}-10*b_{0,-1}+58*b_{0,0}+17*b_{0,1}-5*b_{0,2}+b_{0,3})/64 \quad (10)$$

$$j_{0,0}=(-b_{0,-3}+4*b_{0,-2}-11*b_{0,-1}+40*b_{0,0}+40*b_{0,1}-11*b_{0,2}+4*b_{0,3}-b_{0,4})/64 \quad (11)$$

$$q_{0,0}=(b_{0,-2}-5*b_{0,-1}+17*b_{0,0}+58*b_{0,1}-10*b_{0,2}+4*b_{0,3}-b_{0,4})/64 \quad (12)$$

$$g_{0,0}=(-c_{0,-3}+4*c_{0,-2}-10*c_{0},-1+58*c_{0,0}+17*c_{0,1}-5*c_{0,2}+c_{0,3})/64 \quad (13)$$

$$k_{0,0}=(-c_{0,-3}+4*c_{0,-2}-11*c_{0,-1}+40*c_{0,0}+40*c_{0,1}-11*c_{0,2}+4*c_{0,3}-c_{0,4})/64 \quad (14)$$

$$r_{0,0}=(c_{0,-2}-5*c_{0,-1}+17*c_{0,0}+58*c_{0,1}-10*c_{0,2}+4*c_{0,3}-c_{0,4})/64 \quad (15)$$

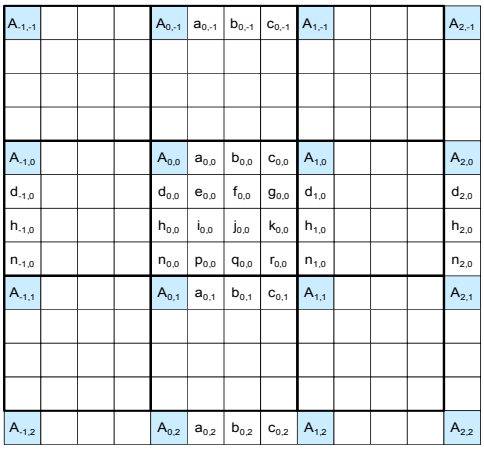

Figure 1. Position of Integer Sample Point and Non-integer Sample Point in the Interpolation of Luminance

2.2. Interpolation Process of Chrominance Sample

Figure 2 shows the positions of the integer pixel sample, 1/2 pixel sample, 1/4 pixel sample, 1/8 pixel sample of the chrominance components of the reference image. It is supposed that chrominance sample point $B_{i,j}$ is located at the integer sample point ($xB_{i,j}$, $yB_{i,j}$), then the predicted value from chrominance point '$ab_{0,0}$' to '$hh_{0,0}$' at non-integer sample positions can be obtained by the 4-beat filter with the coefficient of (-2,58,10,-2), (-4,54,16,-2), (-6,46,28,-4), (-4,36,36,-4), (-4,28,46,-6), (-2,16,54,-4), (-2,10,58,-2).

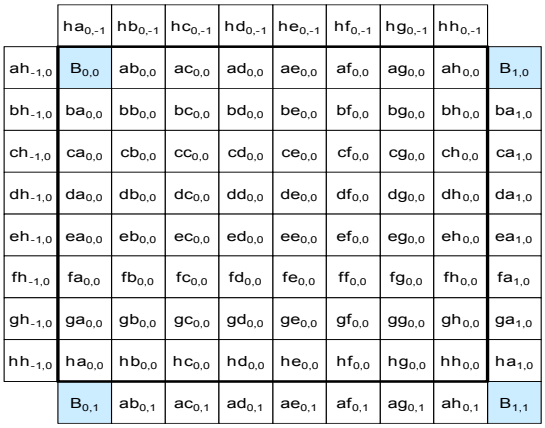

Figure 2. Positions of Integer Sample Point and Non-integer Sample Point in the Interpolation of Chrominance

(1) Horizontal and Vertical Interpolation Filtering of Integer Pixel

The values of 1/2 pixel points $ae_{0,0}$, $ea_{0,0}$;1/4 pixel point $ac_{0,0}$, $ag_{0,0}$, $ca_{0,0}$, $ga_{0,0}$; and 1/8 pixel point $ab_{0,0}$, $ad_{0,0}$, $af_{0,0}$, $ah_{0,0}$, $ba_{0,0}$, $da_{0,0}$, $fa_{0,0}$, $ha_{0,0}$ can be obtained by using filter interpolation mentioned in the paper on the nearest integer pixel in the horizontal and vertical directions. The whole computing process is as following:

$$ab_{0,0}=(-2*B_{-1,0}+58*B_{0,0}+10*B_{1,0}-2*B_{2,0})/64 \quad (16)$$

$$ac_{0,0}=(-4*B_{-1,0}+54*B_{0,0}+16*B_{1,0}-2*B_{2,0})/64 \quad (17)$$

$$ad_{0,0}=(-6*B_{-1,0}+46*B_{0,0}+28*B_{1,0}-4*B_{2,0})/64 \quad (18)$$

$$ae_{0,0}=(-4*B_{-1,0}+36*B_{0,0}+36*B_{1,0}-4*B_{2,0})/64 \quad (19)$$

$$af_{0,0}=(-4*B_{-1,0}+28*B_{0,0}+46*B_{1,0}-6*B_{2,0})/64 \quad (20)$$

$$ag_{0,0}=(-2*B_{-1,0}+16*B_{0,0}+54*B_{1,0}-4*B_{2,0})/64 \quad (21)$$

$$ah_{0,0}=(-2*B_{-1,0}+10*B_{0,0}+58*B_{1,0}-2*B_{2,0})/64 \quad (22)$$

$$ba_{0,0}=(-2*B_{0,-1}+58*B_{0,0}+10*B_{0,1}-2*B_{0,2})/64 \quad (23)$$

$$ca_{0,0}=(-4*B_{0,-1}+54*B_{0,0}+16*B_{0,1}-2*B_{0,2})/64 \quad (24)$$

$$da_{0,0}=(-6*B_{0,-1}+46*B_{0,0}+28*B_{0,1}-4*B_{0,2})/64 \quad (25)$$

$$ea_{0,0}=(-4*B_{0,-1}+36*B_{0,0}+36*B_{0,1}-4*B_{0,2})/64 \quad (26)$$

$$fa_{0,0}=(-4*B_{0,-1}+28*B_{0,0}+46*B_{0,1}-6*B_{0,2})/64 \quad (27)$$

$$ga_{0,0}=(-2*B_{0,-1}+16*B_{0,0}+54*B_{0,1}-4*B_{0,2})/64 \quad (28)$$

$$ha_{0,0}=(-2*B_{0,-1}+10*B_{0,0}+58*B_{0,1}-2*B_{0,2})/64 \quad (29)$$

(2) Vertical Interpolation Filtering of 1/2 Pixel and 1/4 Pixel

The value of sub-pixel sample point $bX_{0,0}$, $cX_{0,0}$, $dX_{0,0}$, $eX_{0,0}$, $fX_{0,0}$, $gX_{0,0}$ and $hX_{0,0}$ (among which, X presents any one in b, c, d, e, f, g and h) can be obtained by the 4-beat filter interpolation in the vertical direction. The whole computing process is as following:

$$bX_{0,0}=(-2*aX_{0,-1}+58*aX_{0,0}+10*aX_{0,1}-2*aX_{0,2})/64 \quad (30)$$

$$cX_{0,0}=(-4*aX_{0,-1}+54*aX_{0,0}+16*aX_{0,1}-2*aX_{0,2})/64 \quad (31)$$

$$dX_{0,0}=(-6*aX_{0,-1}+46*aX_{0,0}+28*aX_{0,1}-4*aX_{0,2})/64 \quad (32)$$

$$eX_{0,0}=(-4*aX_{0,-1}+36*aX_{0,0}+36*aX_{0,1}-4*aX_{0,2})/64 \quad (33)$$

$$fX_{0,0}=(-4*aX_{0,-1}+28*aX_{0,0}+46*aX_{0,1}-6*aX_{0,2})/64 \quad (34)$$

$$gX_{0,0}=(-2*aX_{0,-1}+16*aX_{0,0}+54*aX_{0,1}-4*aX_{0,2})/64 \quad (35)$$

$$hX_{0,0}=(-2*aX_{0,-1}+10*aX_{0,0}+58*aX_{0,1}-2*aX_{0,2})/64 \quad (36)$$

It is indicated from the above algorithm analysis that the interpolation of certain sub-pixel points can be processed only after some other sub-pixel points are obtained. For example, the interpolation of luminance at 1/2 pixel point $j_{0,0}$ and 1/4 pixel point $e_{0,0}$,$i_{0,0}$,$p_{0,0}$,$f_{0,0}$$q_{0,0}$,$g_{0,0}$,$k_{0,0}$,$r_{0,0}$ depends on 1/2 pixel point $b_{0,0}$ and 1/4pixel points $a_{0,0}$and $c_{0,0}$. In summary, the computation sequence is shown in Table 1.

Table 1. Computation Sequence of Interpolation at Sub-Pixel Point

Sequence	Sub-pixel Point
Step1	$b_{0,0}$, $a_{0,0}$, $c_{0,0}$
Step2	$d_{0,0}$, $e_{0,0}$, $f_{0,0}$, $g_{0,0}$
Step3	$h_{0,0}$, $i_{0,0}$, $_{0,0}$, $k_{0,0}$
Step4	$n_{0,0}$, $p_{0,0}$, $q_{0,0}$, $r_{0,0}$

3. Design and Realization of Hardware Framework

3.1. Structure of Luminance Interpolation

Sub-pixel interpolation filtering demands the expansion of reference macro-block. The order of filter decides the read amount of the reference data [3-5]. When the size of the macro-block is M×N, the reference data amount required to be read is (M+7) × (N+7). For example, the interpolation of an 8×8 reference block adopts 8-order filter, which requires expansion of 3 integer pixel points in the left and at the top of the 8×8 block, and expansion of 4 integer pixels points in the right and at the bottom of the block, thus 15×15 reference data amount are required to be read, as it can be seen in Figure 3. The 1/2 pixel value and 1/4 pixel value in the algorithm can be obtained by horizontal interpolation filtering, and then vertical interpolation filtering. Overall structure of the sub-pixel interpolation process of luminance is shown in Figure 4.

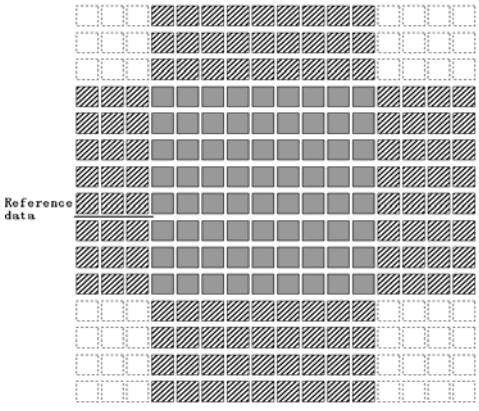

Figure 3. Reference Pixels Required at Sub-pixel Interpolation Position

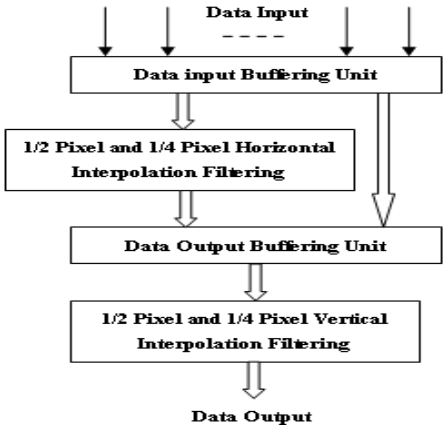

Figure 4. Overall Structure of Sub-Pixel Interpolation Process of Luminance

Three operators are adopted in this paper to conduct the 1/2 pixel and 1/4 pixel interpolation filtering in horizontal and vertical directions. The first operator, second operator and third operator are 8-tapping finite impulse response filter (FIR) of weight coefficient (-1,4,-10,58,17,-5,1)、(-1,4,-11,40,40,-11,4,-1)、(1,-5,17,58,-10,4,-1). Figure 5 shows the structure of 1/2 pix and 1/4 pixel interpolation filtering. The first operator and third operator conduct 1/4 pixel interpolation filtering and the second operator conducts 1/2 pixel interpolation filtering.

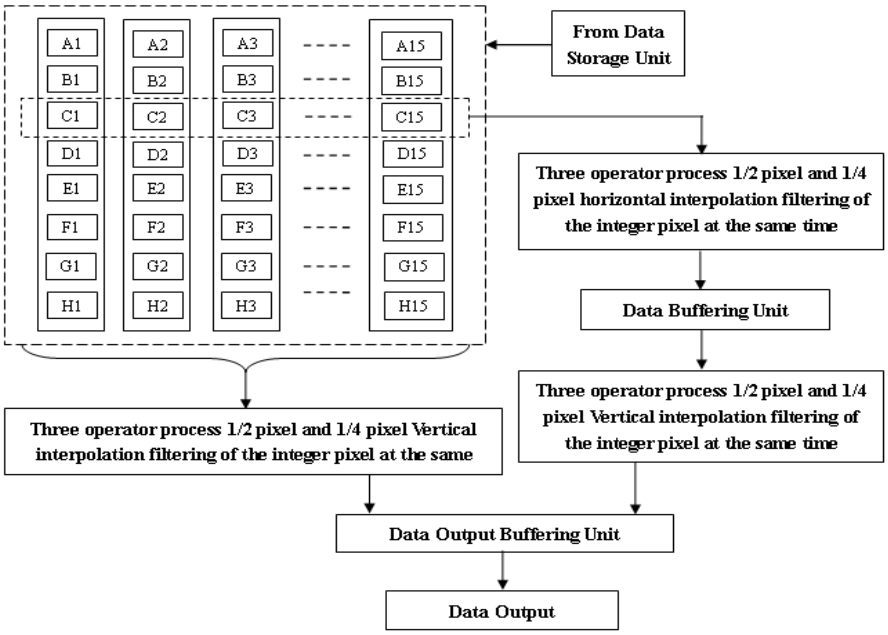

Figure 5. 1/2 Pixel and 1/4 Pixel Interpolation Filtering Structure

When the size of data storage unit is (M+7)×(N+7), input buffering unit includes (M+7) line registers which are used to store the data line consisting of (N+7) pixels, input by the data storage unit according to pixel storage. And multiple data items belonging to different lines are stored in respective line registers.

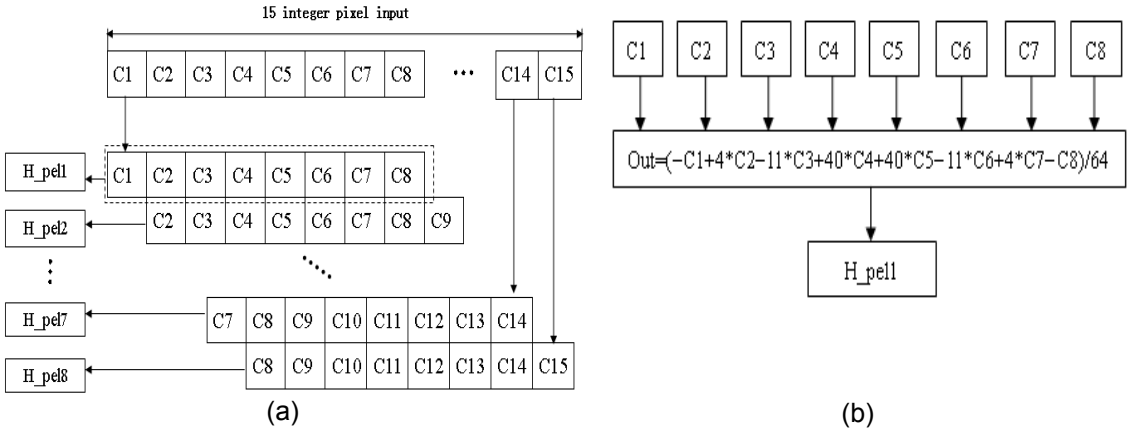

Figure 6. 1/2 Pixel Horizontal Interpolation Operation of the Second Operator

Among which C1, C2, C3, C4, C5, C6, C7, C8 is the input data item.

By analyzing Figure 6, 7 and 8, after inputting the 7 line data items into the input buffer, the first operator uses 7 integer pixel values (such as C1,C2,C3,C4,C5,C6 and C7) with the arrangement of a shifted pixel to conduct interpolation operation on the 1/4 pixel "$a_{0,0}$". Meanwhile the first operator conducts 9 times of 1/4 pixel interpolation operations in horizontal direction. Then, the results of interpolation H_pel1′…H_pe19′ are stored in the data buffering unit. When the first operator conducts 1/4 pixel interpolation filtering, the second and third operator conduct interpolation operation on 1/2 pixel "$b_{0,0}$" and 1/4 pixel "$c_{0,0}$" and 8 times of horizontal interpolation filtering, and the obtained interpolation results (H_pel1…H_pe18 and H_pel1″…H_pe18″) are stored in the data buffering unit.

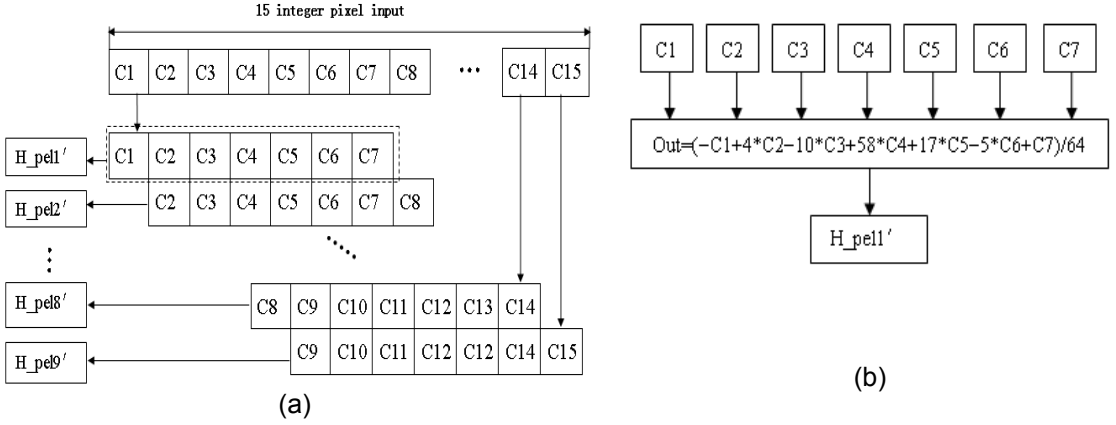

Figure 7. 1/2 Pixel Horizontal Interpolation Operation of the first Operator

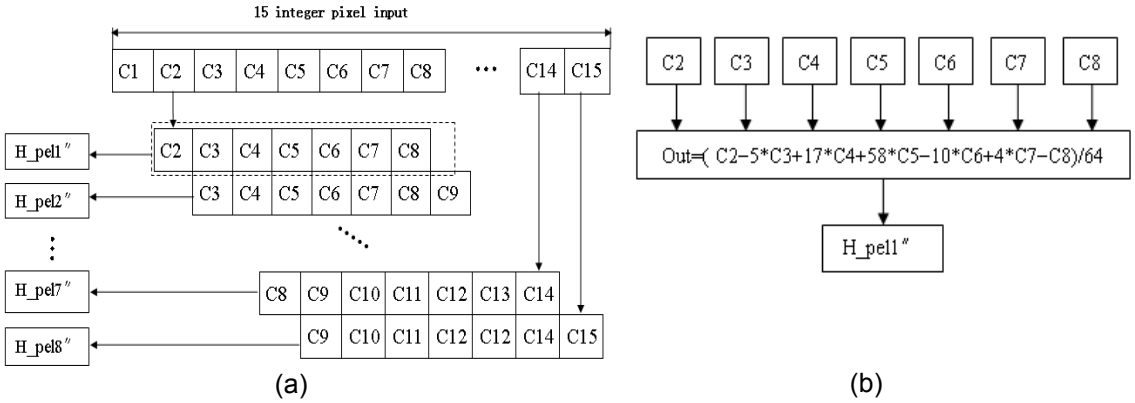

Figure 8. 1/2 Pixel Horizontal Interpolation Operation of the Third Operator

From the above analysis, the values of 1/2 pixel point and 1/4 pixel point between integer pixel lines in the horizontal direction are obtained by interpolation filtering on the nearest integer pixels using 3 operators. Similarly, the values of the 1/2 pixel point and 1/4 pixel point between integer pixel lines in the vertical direction are obtained by interpolation filtering on the nearest integer pixels using 3 operators. After obtaining the values of 1/2 pixel point and 1/4 pixel point between the integer pixel lines in the horizontal direction, the value of sub-pixel between the non-integer pixel lines and lists (such as: 1/2 pixel $j_{0,0}$ and 1/4 pixel $e_{0,0}$, $i_{0,0}$, $p_{0,0}$, $f_{0,0}$, $q_{0,0}$, $g_{0,0}$, $k_{0,0}$and $r_{0,0}$) can be obtained by interpolation filtering on the nearest 1/2 pixel and 1/4 pixel in vertical direction using 3 operators. The principles of interpolation process in horizontal direction and vertical horizontal are the same, thus the hardware structure chart of the sub-pixel interpolation process in vertical direction is not given in the paper.

3.2. Structure of Chrominance Interpolation

7 operators are adopted to conduct 1/2 pixel, 1/4 pixel and 1/8 pixel interpolation filtering of horizontal and vertical chrominance. Seven operators are 4-beat finite impulse response (FIR) filter with weight coefficient (-2,58,10,-2), (-4,54,16,-2), (-6,46,28,-4), (-4,36,36,-4), (-4,28,46,-6), (-2,16,54,-4), (-2,10,58,-2). The principles are the same with those of sub-pixel interpolation process of luminance, thus the hardware structure chart of sub-pixel interpolation process of chrominance is not presented in details.

4. Performance Analysis

The designed interpolation structure in this paper is capable of conducting 1/2 pixel and 1/4 pixel interpolation filtering; besides, the stored data in registers can be shifted in the unit of line to conduct sub-pixel interpolation. As the register is shifted, different operations related to interpolation can be conducted. Except for the initial 6 hours delay, for the adjacent 8×8 block in vertical direction, constant input of data is available, which does not require the filling of the pipeline. This can not only accelerate the computation and reduce delay, but also increase the reuse of interpolation data, thus 26% of the clock period can be saved. Register supports the sub macro-block with changing sizes, and the pixel is shifted up in the same direction, thus the 1/2 pixel and 1/4 pixel interpolation can be applied to blocks of all sizes, and the structure of the interpolation circuit can be simplified.

5. Conclusion

Similar to H.264/AVC, HEVC also adopts sub-pixel interpolation algorithm (luminance: 1/4 precision, chrominance: 1/8 precision). The luminance and chrominance adopt filters of different orders to realize the interpolation of the pixel at desired position. Different filtering algorithms with various precisions in the interpolation process of HEVC are discussed in details in this paper. Aiming at the large computation amount of interpolation operation process, a hardware framework design of interpolation based on pipeline structure is put forward. The design takes consideration of the balance between processing speed and area. It is indicated from the performance analysis that the structure is capable of decreasing the bandwidth and increasing the handling capacity, meeting the requirement of dealing with high-definition image.

Acknowledgements

This work was supported by the national natural science foundation of China (Grant No. 61171078), and Important Project of Baicheng Normal University.

References

[1] Ugur, Kemal, Andersson, Kenneth, Fuldseth, Arild, et al. Low complexity video coding and the emerging HEVC standard. *Picture Coding Symposium (PCS)*. 2010: 474~477.
[2] Wiegand T, Ohm JR, Sullivan GJ. et al. Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization. *IEEE Transactions on Circuits and Systems for Video Technology,* 2010;12; (20):1661~1666.
[3] Bi Hou-jie. A new generation of video coding standard (H.264/AVC). Beijing: Post &Telecom Press. 2005.
[4] ZHAO Zi-liang, ZHENG Shi-bao. Design and Implementation of Quarter-pixel Interpolation in H. 264 /AVC. *Journal of Image and Graphics*. 2007: 1740~1744.
[5] LAI Xiao-ling, SANG Hong-shi, et al. Sub- Pixel Motion Estimation Interpolation Method and its High-efficient VLSI Implementation for H.264*. Computer Engineering*. 2007: 187~189.
[6] Fang Yan-long, Zhou Jun, et al. Fast algorithm for H.264 interpolation.*Computer Engineering.* 2006: 218~223.
[7] Hu Li Wang Feng, Zhen Shi-bao, et al. A hardware architecture for quarter-pixel interpolation in H. 264. *Digital Television and Video*. 2005: 14~17.
[8] Yu Yuan-xi. *Installation and method of image interpolation by Applying 1/4 Pixel Interpolation in the Result of 1/2 Pixel Interpolation*. CN1703094A (Patent). 2005.
[9] Xiong Jie. Digital Medical Image Enhanced by WaveletIllumination-Reflection Model. *TELKOMNIKA.* 2013; 11(1): 19-27.
[10] Yuan LUO, Chao JI, Yi Zhang, Zhang fang. A Sub-pixel Detection Algorithm of the MEMS Dynamic Fuzzy Image. *TELKOMNIKA.* 2012; 10(8): 2075~2080.