COMPUTATIONALLY CONTROLLABLE INTEGER, HALF, AND QUARTER-PEL MOTION ESTIMATOR FOR MPEG-4 ADVANCED SIMPLE PROFILE

Wei-Min Chao, Tung-Chien Chen, Yung-Chi Chang, Chih-Wei Hsu, and Liang-Gee Chen

DSP/IC Design Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University
1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan
{hydra,djchen,watchman,jeromn,lgchen}@video.ee.ntu.edu.tw

ABSTRACT

A cost-effective hardware architecture of integer, half, and quarter-pel motion estimation for MPEG-4 Advanced Simple Profile is proposed in this paper. Three-step hierarchy scheme is employed to cope with different pixel accuracy. For integer-pel estimation, the proposed computation-controllable algorithm makes it easy to be integrated into the coding system according to the power, quality, and timing conditions. For half and quarter-pel motion estimation, hardware-oriented algorithm and related architecture are proposed for cost reduction and provide 1.63 to 4.81 dB improvement in PSNR quality. The implementation takes 15K gates at 54MHz for sequences of CIF format at 30 fps.

1. INTRODUCTION

To achieve a cost-effective implementation of a video codec for specific applications, MEPG-4 has defined Profiles and Levels as conformance points. Advanced Simple Profile (ASP) can achieve lower bits and maintain low delay targeted for Internet and streaming video applications. Three key tools adopted in ASP to improve the coding gain are B-VOP, quarter-pel estimation /compensation (QME/QMC), and global motion estimation /compensation (QME/QMC). Among them, QME/QMC can achieve about 1 dB gain as compared with Simple Profile [1]. Therefore, QME/QMC is attractive to be incorporated into the codec system.

To keep the acceptable computational power, the hierarchy scheme [2] for integer, half, and quarter pixel accuracy of motion estimation (IME, HME, and QME) is usually employed as shown in Fig. 1. First of all, the integer-pel ME (IME) is performed to find the best matching macroblock (MB) labeled as task 1. Then the half-pel motion vector is searched around the previous integer-pel motion vector from +1 to -1 half-pel unit, which is labeled as task 2. QME, labeled as task 3, is performed around the half-pel motion vector about +1 to -1 quarter-pel unit at the last stage. Different algorithms can be employed to handle these tasks separately. In the past literatures [3], there are many optimized architecture designs proposed for integer-pel ME. However, without considerations of HME and QME, it will still result in a large portion of cost in the coding system.

For IME, many fast algorithms are proposed to reduce the huge computational power of full search algorithm which can find the global minimum matching error in the reference search window. They can be classified into two categories [3]. The first one uses the subset of the search candidates. The other uses the subset of pixels for the matching criteria. Both these ME algorithms may be trapped into the local minimum and it causes the sub-optimal results of video coding. In addition, the processing power is usually proportional to the matching quality among various kinds of fast algorithms. While it requires huge computational power for full search algorithm, three step search [4] for search -16 to +15 only requires about 3.22% computational power but sacrifices much compression quality. Another disadvantage in some of these algorithms such as diamond search [5] is that the computational power can not be determined in advance. It always depends on features of sequences and takes various processing cycles. The system is compelled to decide scheduling dynamically and takes care of the worst cases to satisfy the real-time applications. To sum up, in these algorithms, the coding system cannot adjust the algorithms fine granularity according to the power, quality, and timing conditions.

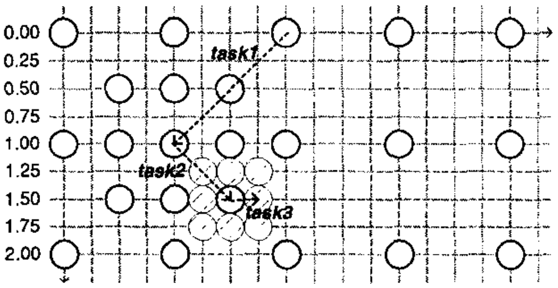

Figure 1: Hierarchy scheme for IME, HME, and QME

For HME and QME, it needs to interpolate fractional samples before computing matching criteria (sum of absolute difference, SAD) for eight candidates separately in the hierarchy scheme. In MPEG-4 ASP, the 8-tap interpolation filter for half-pel positions is followed by bilinear interpolation to derive quarter-pel samples and it costs a lot for hardware implementation in the tight timing specification.

To solve these problems addressed above, we propose the hardware architecture of IME, HME, and QME with hardware-oriented algorithms to achieve cost-effective implementation for MPEG-4 ASP. In this architecture, the two stage and computationally controllable ME algorithm and it VLSI implementation are proposed

0-7803-7761-3/03/$17.00 ©2003 IEEE

to solve IME according to the system conditions. Then the simplified but efficient HME and QME algorithms are combined into this architecture. The organization in this paper is as follows. In Sec. 2, we introduce the computationally controllable IME algorithm. In Sec. 3, the hardware-oriented HME and QME algorithm and analysis are given. In Sec. 4, the hardware design is presented. In Sec. 5, the experiment results and discussions are described. Finally, Sec. 6 concludes this paper.

2. COMPUTATIONALLY CONTROLLABLE IME

The proposed algorithm belongs to the category of using the subset of the search candidates. To reduce the complexity and maintain the prediction quality is the key consideration to define the sub-sampled search pattern. Many analyses of the distribution of motion vectors show that they are closely located in the center of the search window or around the predictors derived from those of the spatially neighboring MBs. Following these concepts, we define a tunable pattern with two parameters to adjust the computational power from low to high and improve the quality in the direct proportion. Fig. 2 depicts the pattern which has two exclusive regions. One, named fine region (FR), is the interior part of the search window with the higher probability in the appearance of best-matching MBs. The other is the exterior part, named coarse region (CR). The predictor is chosen as the first candidate and others are from the origin of the search window and then spiral out in order. The direction is denoted as the dot-line. Two parameters, the fine-to-coarse threshold (FC) and the jumping parameter (JP), are defined to achieve the fine granularity in the computational power. FC controls the region ratio of CR and FR. While FC turns to be larger, the area of FR becomes larger and the prediction quality and computational power grows up at the same time. The resolution of FR is fixed as one pixel unit but that of CR is defined by JP. If JP is larger, then skipping candidates in CR becomes more and hence more computational power is reduced. By adjusting JP and then FC dynamically we distribute the computational power to each MB.

After finding the winner candidate through the first stage, we apply the diamond search to find the best motion vector in the local area. The iteration count of the diamond refinement is the half of JP. Besides, the small diamond pattern without iteration is used if JP is two. The feedback control scheme in the system part is shown in Fig.2. Every ten frames are united to be a group. For 30 fps real-time specification, each group has 1/3 seconds to carry out the coding process. By adjusting JP and FC, the working cycles for each MB can be adjusted to achieve the best compression under the system conditions such as computing, power, and timing constraints.

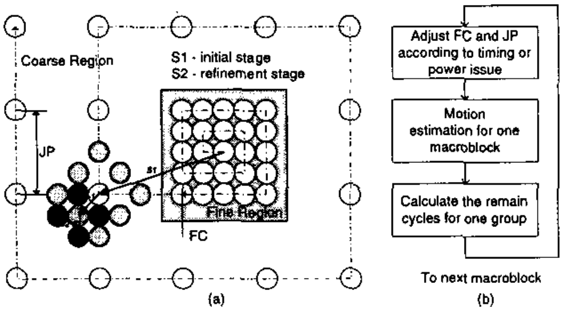
Coarse Region
S1 - initial stage
S2 - refinement stage
JP
S1
Fine Region
FC
(a)
Adjust FC and JP according to timing or power issue
Motion estimation for one macroblock
Calculate the remain cycles for one group
To next macroblock
(b)

Figure 2: (a)Tunable search pattern (b)System control scheme

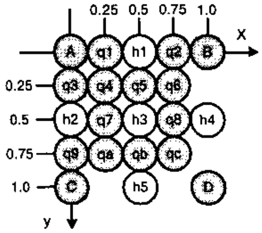

Figure 3: Half and quarter sample positions for ME.

3. HARDWARE CONSIDERATION OF HME AND QME

In this section, HME and QME in MEPG-4 ASP are analyzed and a efficient algorithm is proposed. Before calculating the matching criteria for the half or quarter sample positions, the interpolation process is applied to the original integer samples in the reference frame. Fig. 3 shows the integer, half, and quarter prediction samples. The prediction values labeled as 'h' at half position are generated from a 8-tap filter with symmetry tap values {160,-48,24,-8} in X and/or Y dimension. Then the prediction values label as 'q' at quarter positions are generated by bilinear operations among neighboring samples, that is, averaging the samples at integer and half sample positions in the X and/or Y direction. In QMC, the interpolation operation should coincide with that defined in the standard to avoid error draft in decoding sides. However, the operation is not forced to be the same in QME, which required 16 times of computational load compared with QMC for eight candidates in half sample positions and eight for quarter. If QME uses the same filter as that in QMC, the matching criteria can be correct to find the best matching MB. To reduce the cost, we simplify the filter operations to emulate the results that the 8-tap filter does. That is, if the matching error of the prediction MB from the simplified filter and the current MB is smaller, it will have the higher trend that the current MB is more similar to that generated from the 8-tap filter. First, the bilinear operation is applied to replace the vertical 8-tap filtering operations. Besides, the moving resolution in the horizontal direction is finer than that in the vertical direction in most sequences as shown in Table 1. It is reasonable to use the correct interpolation method for quarter samples in horizontal direction (h1 and h5) and the vertical ones are emulated by the bilinear filter. Four kinds of interpolation methods in Table 2 are evaluated in Sec.5.

4. HARDWARE ARCHITECTURE

Fig. 4 depicts the overall architecture for IME, HME, and QME. It contains two local buffers for storing current MB (MBMEM) and the search window data (SWMEM) separately to decrease the system bus traffic. Before ME starts, the system loads data from external memory to these two local buffers. Two stages are designed in the proposed architecture as like to the previous work [6]. The pattern generation (PG) stage generates candidates according to

Table 1: Motion vector distribution in various pixel grids(I:integer, H:half, Q:quarter position)

	Horizontal(%)			Vertical(%)		
Seq.	I	H	Q	I	H	Q
Foreman	45.2	11.4	43.2	48.0	10.6	41.1
Mobile	23.5	2.6	73.6	52.3	7.5	39.8
Coast.	28.5	12.8	58.5	57.2	3.2	39.2
Stefan	38.7	9.1	52.0	43.1	5.8	50.8
Weather	88.7	1.1	10.1	88.5	1.3	9.9

Table 2: Various interpolation methods for fractional samples

Method	Horizontal	Vertical
FIR	8-tap filter	8-tap filter
VBI	8-tap fitler	2-bilinear
HBI	bilinear	8-tap filter
VHBI	bilinear	bilinear

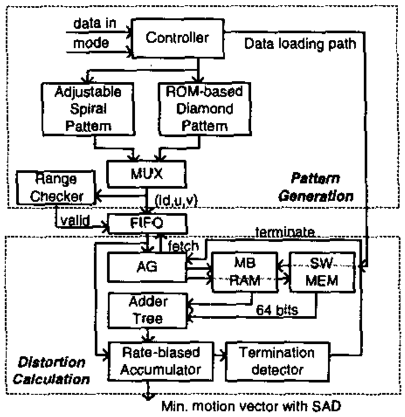

Figure 4: Architecture of proposed motion estimator

the spiral pattern or the diamond pattern after the algorithm mode is decided. The distortion calculation (DC) stage takes responsibility for calculating SAD of each candidate. A FIFO inserted between PG and DC stages can improve the hardware utilization owing to the mismatch of the throughput in PG and DC stage.

The eight-way interleaved memory organization is used to dynamically fetch eight pixels in one cycle without reading collision in the same memory bank. Fig. 5 depicts this organization with the search range -16 to +15 pixels. The co-located 48 by 48 pixel size of search windows in the reference frame are required to be loaded to SWMEM with three strips before ME begins for each MB labeled as A, B, C, and D. Under the leftmost MB of the frame, all data located in these three strips must be loaded. However, the left MBs in the same row can reuse two-third of search window of the immediately previous MB. For example, it should load the entire search window to Strip 0, 1, and 2 in SWMEM to perform ME for the MB A. At the next MB B, the data in the Strip 1 and 2 can be reused and only the right one third of search window for the MB B have to be loaded to Strip 0. With rotation and modulation operations for addressing, this column-by-column data reuse scheme is applied to this ME architecture. The bus traffic for loading search window is then reduced from 26.10 to 9.49 Mbytes per second for CIF format with the search range of -16 to +15 pixel. In each strip, eight horizontal neighboring pixels (a half row) are stored into eight separate RAMs with the linear addressing. While reading a half-row of pixels randomly, two consecutive addresses are calculated first from the two-dimension coordinates. Then the proper circular rotation operations are applied to the data read out from the memory banks.

Fig. 6 shows the architecture to generate the integer, half, and quarter-pel samples from the reference frame according the specified coordinates of some candidates. In the case of integer-pel positions, the pixel data are read out and used as the reference pixels without interpolation. But in the case of half or quarter-pel positions, the integer pixels are immediately interpolated into the fractional pixels by fractional pixel generator (FPG) without any additional temporary memory. Two different architectures for FPG are both designed to compare the performance and the cost. One in Fig.6 (b) applies VBI. Two 64 bit registers hold sixteen horizontal pixels noted as p1 to p16 every two cycle and they are used as the inputs to eight of 8-tap filters in order at once. For example, the p1 to p8 are the inputs of the first 8-tap filter and the p2 to p9 are the inputs of the second 8-tap filter. Neighboring tap values can be shared with each other and the boundary mirroring are avoided to use extended four reference pixels directly at both sides. Then three 64-bit registers are cascaded as a delay line for the inputs of vertical bilinear operations. The last and first registers in the delay line are the up and down eight horizontally sampled pixels of two rows in the spatial domain. Eight processing elements perform bilinear operation to these eight pairs of pixels in the vertical direction. The alternative with lower cost is shown in Fig.6 (c) applies VHBI. Five of the eight pixels are used as the inputs of four processing elements. Each process element performs bypass or bilinear operations on its input pixels. The top four PEs take responsibility for the horizontal samples and bottom ones are for vertical samples which are from the top PEs and the delay line registers. It can also generate a half row every two cycles.

DC stage gets eight pixels at one cycle for IME and therefore can compute partial SAD for a half of row in one MB per cycle. Besides, the partial distortion elimination applied to DC stage and it can terminates the accumulation of SAD if the current accumulating SAD is larger than the current minimum one. Applying this halfway termination skill, the DC stage can process one MB less than 32 cycles. Due to good initial guess that the smaller SAD is located around the center or predictors, the predictor first and then spiral order can reduce 30% processing. It results in 22 cycles for one candidate in integer sample position in the average.

While HME and QME are applied, the one more cycle are required to interpolate the fractional samples. Hence, it takes two cycles for computing partial SAD in a half of row in a MB. The halfway termination skill can also be applied to reduce the processing cycles. Averagely, it takes 45 cycles for one candidate in the half or quarter sample position.

5. EXPERIMENT RESULTS

Table 3 depicts the performance for IME. Two active sequences which are not handled well in most fast algorithms are used as inputs. The working frequency is increasing to meet the real-time

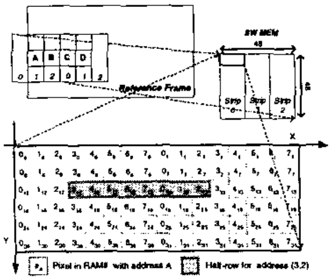

Figure 5: Interleaved memory organization for reference pixels

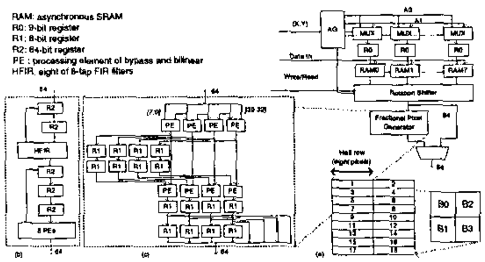

Figure 6: Fractional pixel generator in the SWMEM unit (a) reading sequence (b) VBI (c) VHBI

specifications while the matching error is decreasing. In the extreme case that CF equals to 1023, this algorithm turns into full search and it guarantees the global minimum will be found. In the opposite that CF equals to 1 and the JP is unrestricted, it is degraded to be center-biased diamond search. In these two extreme cases for Stefan sequence, the working frequency can be scaled from 157MHz to 7MHz and hence the power consumption can be controlled from 100% to 4.46%.

Table 4 depicts the performance of HME and QME around the found integer-pel motion vector. The improvement from HME is about 1.63 to 2.29 dB and that from QME is about 1.63 to 4.81 dB. They can provide the substantial improvement for these testing sequences and especially for those with finer texture. Hence, it is worth to integrate the QME into the coding system. Besides, VBI performs well compared to VHBI while VBI filter takes 8851 gates and VHBI filter takes 2300 gates.

The design is written in Verilog and synthesized by 0.35 μm 1P4M CMOS cell library. The local memory size is 39,080 bits. The timing constraints are targeted to 54MHz and the total logic size with VHBI is 15K gates and that with VBI is 22K gates.

6. CONCLUSION

In this paper, a cost-effective hardware architecture for IME, HME, and QME is proposed. Hierarchy scheme is employed to cope with various pixel resolutions. First, the proposed computation-controllable algorithm can be adjusted by the system according to the power, quality, and timing conditions. Second, the hardware-oriented half and quarter-pel refinement algorithm can improve 1.63 to 4.81 dB in PSNR quality. The hardware architecture with VHBI filter takes 15K gates and 39,080 bits memory for search range from -16.0 to +15.0 at both axes and it meets the real-time requirements for MPEG-4 ASP at Level 3 (352x288 pixels at 30 fps).

Table 3: Performance of IME under different working frequency

Stefan		Table		Parameters		
MHz	Min.SAD	MHz	Min.SAD	CF	JP	L
157	2763	200	1483	1023	1	1
66	2842	72	1527	224	2	1
46	2856	55	1532	168	3	1
34	2937	39	1541	120	4	1
26	2858	30	1520	80	5	2
20	2929	22	1518	48	6	3
14	2929	14	1531	24	7	4
7	3074	5	1726	1	20	15

Table 4: Quality of various pixel accuracy for testing sequences under the similar bit-rate (+-5%)

Seq.	PSNR	Gain				
	I	H	Q FIR	Q VHBI	Q VBI	Q HBI
Foreman	29.32	2.09	2.26	2.11	2.20	2.17
Mobile	23.76	2.55	4.84	4.79	4.82	4.81
Coast.	27.20	2.26	2.52	2.42	2.51	2.42
Stefan	25.93	2.97	3.45	3.37	3.40	3.40
Weather	27.69	1.63	2.07	1.63	1.64	1.63

7. REFERENCES

[1] MPEG-4 Video Group, AMENDMENT 4, "Streaming Video Profile," vol. ISO/IEC JTC 1/SC 29/WG11 N3904, 2001.

[2] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 19–31, Feb 1997.

[3] P. Kuhn, *Algorithms, Complexity Analysis, and VLSI Architectures for MPEG-4 Motion Estimation*, Kluwer Academic Publications, 1999.

[4] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc. Nat. Telecommun. Conf.*, vol. 9, no. 2, pp. G5.3.1V5.3.5, Nov 1981.

[5] S. Zhu and K.K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. on Image Processing*, vol. 9, no. 2, pp. 287–290, Feb 2000.

[6] W.M. Chao, C.W. Hsu, Y.C. Chang, and L.G. Chen, "A novel hybrid motion estimator supporting diamond search and fast full search," *IEEE Int. Symp. on Circuits and Systems*, vol. 2, pp. 492–495, June 2002.