

# MULTI-MODE CONTENT-AWARE MOTION ESTIMATION ALGORITHM FOR POWER-AWARE VIDEO CODING SYSTEMS

*Siou-Shen Lin, Po-Chih Tseng, Chia-Ping Lin, and Liang-Gee Chen*

DSP/IC Design Lab, Graduate Institute of Electronics Engineering,  
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan  
E-Mail: {sslin, pctseng, cplin, lgchen}@video.ee.ntu.edu.tw

## ABSTRACT

In this paper, a multi-mode content-aware motion estimation algorithm is presented for power-aware video coding systems. By exploiting the characteristics of video signal, two content-aware decision criteria are proposed to identify the complexity of motion vectors. Based on these two decision criteria as well as different combinations of various motion estimation algorithms, four different modes are proposed to allow dynamically varying the computation resources between different power constraints. Besides, the proposed decision criteria also enable the maximization of quality under each power constraint by quality-driven diversity-based search approach. According to our simulation results, the proposed algorithm can effectively reduce the computation resources to 40%, 21%, and 3.73% with only 0.0036dB, 0.01dB, and 0.16dB average quality degradation, respectively. As a result, the proposed algorithm is well-suited for video coding systems that desire power-awareness feature.

## 1. INTRODUCTION

### 1.1. Power-Aware Computing

Power-aware computing [1, 2, 3, 4, 5], which is the computing paradigm enabling to vary the power consumption in response to changing operating conditions, is the emerging concept of embedded system design. For example, when using a portable video device, the user may demand an extremely high quality at the cost of more power resources and thus shorten the lifetime of battery. The opposite could also be true, i.e., the user may endure a worse perceptual quality for extending the lifetime of battery. Such tradeoffs can only be optimally realized when power-awareness issue is taken into consideration.

In general, a well-designed power-aware system meets two main goals. One is the ability to dynamically vary the power consumption between different power constraints, and

the other is the maximization of quality under each power constraint. In order to achieve efficient response to changing operating conditions, the content-aware algorithms have been proposed to exploit signal variations for power-aware computing [1, 3, 5].

### 1.2. Motion Estimation

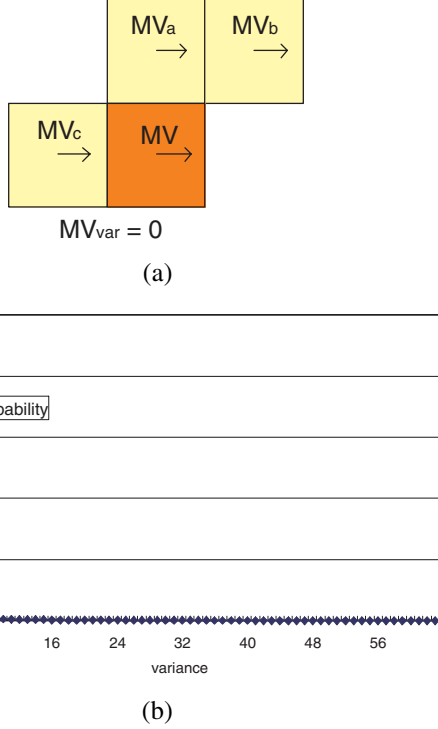
Motion estimation is the fundamental technique of video coding, which effectively reduces the temporal redundancy among video sequences. In order to achieve better video quality, full search block-matching algorithm (FSBMA) can be adopted for motion estimation. The FSBMA determines the motion vector by identifying the macroblock with minimum distortion from a set of all possible candidate blocks in the search window, and therefore enables to achieve the optimal search result. However, it takes huge amount of computation to perform full search of all possible candidate blocks.

In order to reduce the computational complexity and lower power consumption, many fast search algorithms have been proposed in the literature. These fast search algorithms reduce the computation by decreasing the number of matching candidates in the search window, such as the three step search algorithm (3SS)[6], the four step search algorithm (4SS) [7], the diamond search algorithm (DS)[8], and the hexagon based search algorithm (HEXBS)[9]. Although a large amount of computation is eliminated, these fast search algorithms might usually achieve sub-optimal search result because of the reduced search space. However, due to the non-stationary characteristics of video signal, the motion vectors could belong to either simple or complex. For simple motion, fast search algorithms are able to achieve near the same result as full search. But for complex motion, full search could always derive much better result than fast search algorithms. This observation motivates us to work for a content-aware motion estimation algorithm that exploits the characteristics of video signal for power-aware computing.

Based on two content-aware decision criteria, the vari-

---

This work was supported in part by National Science Council, Republic of China, under the grant number 92-2215-E-002-015, and in part by MediaTek Inc.



**Fig. 1.** (a) The variance of motion vectors (b) The variance distribution of motion vectors in the sequence stefan, for CIF format, block size of  $16 \times 16$ , search range from -16 to +15.

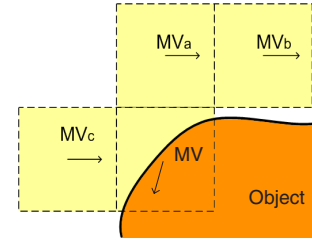
ance of motion vectors and the accuracy of predictive motion vectors, a multi-mode content-aware motion estimation algorithm is proposed. This algorithm provides four different modes to allow dynamically varying the computation resources between different power constraints. Besides, under each mode, the quality are maximized by quality-driven diversity-based search approach. As a result, the proposed algorithm is well-suited for video coding systems that desire power-awareness feature. In the following of this paper, the content-aware decision criteria are described in Sec. 2. Based on the content-aware decision criteria, the multi-mode content-aware motion estimation algorithm is discussed in Sec. 3. Sec. 4 illustrates the performance evaluation results of proposed algorithm. Finally, a brief summary is given to conclude this paper.

## 2. CONTENT-AWARE DECISION CRITERIA

The proposed content-aware motion estimation algorithm is based on two criteria: the variance of motion vectors and the accuracy of predictive motion vectors. They will be described in detail in the following subsections.

**Table 1.** The probability that the variance of motion vectors is smaller than 3, for CIF format, block size of  $16 \times 16$ , search range from -16 to +15.

sequence	Probability of small variance				total
	0	1	2	3	
coastguard	0.5709	0.0665	0.1069	0.1041	0.8483
foreman	0.3807	0.0721	0.0788	0.1030	0.6346
mobile	0.6224	0.0688	0.0896	0.1189	0.8996
stefan	0.7161	0.0608	0.0375	0.0359	0.8502
silent	0.3668	0.0576	0.0518	0.0643	0.5405
weather	0.8617	0.0409	0.0263	0.0246	0.9536
<b>avg</b>	<b>0.5864</b>	<b>0.0611</b>	<b>0.0651</b>	<b>0.0751</b>	<b>0.7878</b>



**Fig. 2.** Because the current block is in another object, the motion vector in current block is not around the predictive motion vectors.

### 2.1. The Variance of Motion Vectors

The motion vectors of the current block and those of the neighbor blocks are highly correlative, because these blocks may reside in the same foreground or background. When the motion vectors of neighbor blocks are the same, the probability is quite high that the motion vectors of the current block are around the motion vectors of neighbor blocks, as shown in Fig. 1(a). By analyzing the variance distribution of motion vectors, the variance is approach to zero most of the time, as shown in Fig. 1(b). The variance is defined as below.

$$MV_{mean} = (MV + MV_a + MV_b + MV_c)/4$$

$$MV_{var} = |MV - MV_{mean}| + |MV_a - MV_{mean}| + |MV_b - MV_{mean}| + |MV_c - MV_{mean}| \quad (1)$$

From Table 1, it can be shown that the probability is 79% in average when the variance of the current block and neighbor blocks is smaller than 3. It can be inferred that when the variance of the motion vectors in the neighbor blocks is small, using the fast search algorithms from the predictive motion vectors instead of full search can get the correct motion vector with much less computation.

## 2.2. The Accuracy of Predictive Motion Vectors

### 2.2.1. Analysis of Boundary Blocks

In general cases, the variance of motion vectors in the neighbor blocks is a good measure to select the appropriate algorithm to get the correct motion vectors with much less computation. In some special cases, however, it is not good enough. When the neighbor blocks are in one object and the current block is in another, the variance of motion vectors in the neighbor blocks is still small, but the motion vectors of the current block may not be around the predictive motion vectors, as shown in Fig. 2.

To detect this condition, the accuracy of predictive motion vectors must be taken into consideration. The accuracy can be estimated in another method: the matching difference of the predictive candidate block. When the variance of motion vectors in neighbor blocks is small, if the current block and the neighbor blocks belong to different objects with different motions, then the matching difference of the predictive candidate block will be large. In other words, when the matching difference is larger than a threshold, the algorithms which are suitable for complex motion are more appropriate than the algorithms which are suitable for simple motion. The SAD value is taken as the criterion of matching difference for the consideration of implementation.

### 2.2.2. Advanced SAD Threshold

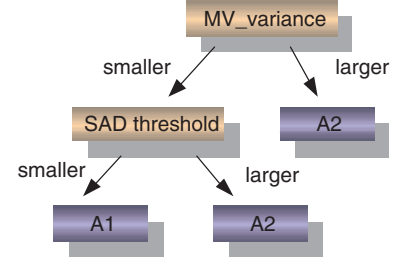
First, the constant SAD threshold is experimented for the sake of fewer computation resources. When the variance of neighbor blocks is smaller, the SAD of predictive candidate block will be compared with the constant SAD threshold to determine the accuracy of the predictive motion vectors.

The constant SAD threshold is not suitable for every sequence. Sometimes, the threshold is too large for one sequence and too small for another. Therefore, the adaptive SAD threshold should be used. The adaptive SAD threshold is determined by neighbor blocks, as shown in Eq. 2.

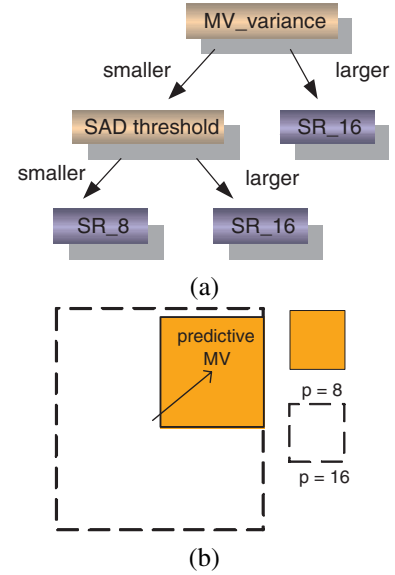
$$SAD_{mean} = (SAD_a + SAD_b + SAD_c)/3$$

$$SAD_{threshold} = R \times SAD_{mean} \quad (2)$$

There are still some problems for the adaptive SAD threshold. For example, there is no limitation of the SAD threshold when the calculated SAD values in neighbor blocks are too large. Especially in the sequence with complex texture, such as the sequence mobile. Hence, the SAD threshold should be limited in a reasonable range. Combining the advantages of the constant threshold and the adaptive threshold, the advanced SAD threshold is proposed in the following equation.



**Fig. 3.** The flow of the proposed content-aware motion estimation algorithm.



**Fig. 4.** (a) The flow of adaptive search range mode. (b) Illustration of SR-8.

$$SAD_{mean} = (SAD_a + SAD_b + SAD_c)/3$$

If  $R \times SAD_{mean} > Constant$ ,

$$SAD_{threshold} = Constant;$$

Else,

$$SAD_{threshold} = R \times SAD_{mean} \quad (3)$$

With the advanced SAD threshold, the complete content-aware motion estimation algorithm with multi-mode is proposed in next section.

## 3. MULTI-MODE CONTENT-AWARE MOTION ESTIMATION ALGORITHM

The flow of the proposed multi-mode content-aware motion estimation algorithm is shown in Fig. 3. The A1 stands

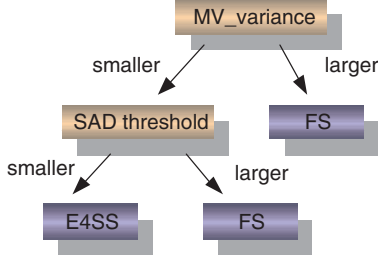


Fig. 5. The flow of adaptive E4SS/FS mode.

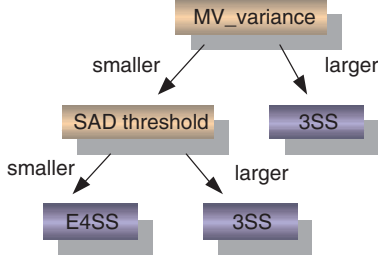


Fig. 6. The flow of adaptive E4SS/3SS mode.

for the algorithm which is suitable for simple motion, while the A2 stands for the algorithm which is suitable for complex motion. When the variance of motion vectors of the neighbor blocks is larger than a threshold, the A2 is appropriate to perform the motion estimation. When the variance of neighbor blocks is small and the accuracy of predictive motion vectors is high, the A1 is more appropriate than the A2 to get the correct motion vectors with fewer computation resources.

Different combinations of the A1 and the A2 constitute different modes of content-aware motion estimation algorithm. These modes are full search (FS) mode, adaptive search range mode, adaptive E4SS/FS mode, and adaptive E4SS/3SS mode.

### 3.1. FS Mode

The FS mode is suitable for the requirement of high quality motion estimation without any power constraint.

### 3.2. Adaptive Search Range Mode

The FS algorithm is adopted in this mode, and the flow is shown in Fig. 4(a). The SR\_8 stands for the FS algorithm in search range from -8 to +7 around the predictive motion vectors, as shown in Fig. 4(b). The SR\_16 stands for the FS algorithm in search range from -16 to +15 around the origin. When the variance of neighbor blocks is small and the accuracy of predictive motion vectors is high, the SR\_8

Table 2. The PSNR drop (a) and the cost (b) of the adaptive search range mode. For CIF format, block size of  $16 \times 16$ , search range from -16 to +15,  $MV_{var}$  threshold = 6, constant = 3072, and R = 3.

	PSNR drop			
	FS	SR_16/SR_8	moving_8	fix_8
coastguard	0.0000	<b>0.0018</b>	0.0071	0.0135
foreman	0.0000	<b>0.0065</b>	0.4012	0.8751
mobile	0.0000	<b>0.0027</b>	0.0330	0.0334
silent	0.0000	<b>0.0053</b>	0.3236	0.4510
stefan	0.0000	<b>0.0050</b>	0.7112	1.2821
weather	0.0000	<b>0.0001</b>	0.0046	0.0052
avg.	0.0000	<b>0.0036</b>	0.2468	0.4434

(a)

	cost (%)			
	FS	SR_16/SR_8	moving_8	fix_8
coastguard	100.00	29.40	25.00	25.00
foreman	100.00	41.01	25.00	25.00
mobile	100.00	42.86	25.00	25.00
silent	100.00	34.08	25.00	25.00
stefan	100.00	62.03	25.00	25.00
weather	100.00	31.50	25.00	25.00
avg.	100.00	<b>40.15</b>	25.00	25.00

(b)

is performed for less computation. When the variance of neighbor blocks is large, the SR\_16 is performed to obtain the correct motion vector.

### 3.3. Adaptive E4SS/FS Mode

The FS algorithm and the Enhanced 4SS (E4SS) are adopted in this mode, and the flow of this mode is shown in Fig. 5. When the variance of neighbor blocks is small and the accuracy of predictive motion vectors is high, the E4SS is performed for less computation. When the variance of neighbor blocks is large, the FS is performed to obtain the correct motion vector.

### 3.4. Adaptive E4SS/3SS Mode

The 3SS and the E4SS are adopted in this mode, and the flow of this mode is shown in Fig. 6. When the variance of neighbor blocks is small and the accuracy of predictive motion vectors is high, the E4SS is performed for less computation. When the variance of neighbor blocks is large, the 3SS is performed to obtain the correct motion vector.

**Table 3.** The PSNR drop (a) and the cost (b) of the adaptive E4SS/FS mode. For CIF format, block size of  $16 \times 16$ , search range from -16 to +15,  $MV_{var}$  threshold = 4, constant = 3548, and  $R = 2$ .

	PSNR drop			
	DS	4SS	3SS	E4SS/FS
coastguard	0.0094	0.0126	0.6622	<b>0.0031</b>
foreman	0.3387	0.4607	1.1393	<b>0.0237</b>
mobile	0.0762	0.0745	0.4892	<b>0.0099</b>
silent	0.5050	0.4652	0.4102	<b>0.0099</b>
stefan	0.6451	0.6828	1.9339	<b>0.0052</b>
weather	0.0886	0.1444	0.4853	<b>0.0087</b>
avg.	<b>0.2772</b>	<b>0.3067</b>	<b>0.8533</b>	<b>0.0101</b>

(a)

	cost(%)			
	DS	4SS	3SS	E4SS/FS
coastguard	3.28	3.84	4.69	<b>8.84</b>
foreman	3.08	3.48	4.69	<b>24.86</b>
mobile	2.61	3.10	4.69	<b>22.43</b>
silent	2.78	2.99	4.69	<b>16.52</b>
stefan	3.31	3.73	4.69	<b>45.83</b>
weather	2.59	2.80	4.69	<b>11.80</b>
avg.	<b>2.94</b>	<b>3.32</b>	<b>4.69</b>	<b>21.71</b>

(b)

**Table 4.** The PSNR drop (a) and the cost (b) of the E4SS/3SS mode. For CIF format, block size of  $16 \times 16$ , search range from -16 to +15,  $MV_{var}$  threshold = 55, constant = 5120, and  $R = 3$ .

	PSNR drop				
	E4SS/3SS	3SS	4SS	E4SS	DS
coastguard	<b>0.0094</b>	0.6622	0.0126	0.0096	0.0094
foreman	<b>0.1927</b>	1.1393	0.4607	0.2893	0.3387
mobile	<b>0.0538</b>	0.4892	0.0745	0.0699	0.0762
silent	<b>0.2503</b>	0.4102	0.4652	0.3874	0.5050
stefan	<b>0.3551</b>	1.9339	0.6828	0.4790	0.6451
weather	<b>0.1034</b>	0.4853	0.1444	0.0882	0.0886
avg.	<b>0.1608</b>	<b>0.8533</b>	<b>0.3067</b>	<b>0.2205</b>	<b>0.2772</b>

(a)

	cost (%)				
	E4SS/3SS	3SS	4SS	E4SS	DS
coastguard	<b>4.50</b>	4.69	3.84	4.49	3.28
foreman	<b>3.84</b>	4.69	3.48	3.82	3.08
mobile	<b>3.82</b>	4.69	3.10	3.73	2.61
silent	<b>3.09</b>	4.69	2.99	3.05	2.78
stefan	<b>4.13</b>	4.69	3.73	3.93	3.31
weather	<b>2.98</b>	4.69	2.80	2.89	2.59
avg.	<b>3.73</b>	<b>4.69</b>	<b>3.32</b>	<b>3.65</b>	<b>2.94</b>

(b)

## 4. PERFORMANCE EVALUATION

### 4.1. FS Mode

The FS mode can obtain the correct motion vectors without any quality degradation. Define the cost as the candidate blocks being compared during motion estimation, and then the cost percentage of the FS mode is 100%.

### 4.2. Adaptive Search Range Mode

According to the simulation result, the cost is about 40% of the FS mode, and the PSNR is dropped only 0.0036dB in average, as shown in Table 2. Compared with the FS with search range from -8 to +7 around the origin (fix\_8) and around the predictive motion vectors (moving\_8), the quality is much better and the cost is not increased too much. The adaptive search range mode is suitable for the requirement of high quality motion estimation with some power constraint.

### 4.3. Adaptive E4SS/FS Mode

According to the simulation result, the cost is about 21% of the FS mode, and the PSNR is dropped 0.01dB in average, as shown in Table 3. Compared with other fast search algorithms, such as the DS, 4SS, etc., the quality is much better. Compared with the FS mode, the cost is much smaller. This mode is a good trade-off between the FS and other fast search algorithms. The adaptive E4SS/FS mode is suitable for the requirement of good quality motion estimation with more power constraint.

### 4.4. Adaptive E4SS/3SS Mode

According to the simulation result, the cost is about 3.73% of the FS mode, and the PSNR is dropped 0.16dB in average, as shown in Table 4. Compared with other fast search algorithms, such as the DS, 4SS, etc., the quality is much better and the cost is at the same order. This mode exploits the different characteristics of 3SS and 4SS effectively. The adaptive E4SS/3SS mode is suitable for the requirement of good quality motion estimation with strict power constraint.

## 5. CONCLUSION

This paper presents a multi-mode content-aware motion estimation algorithm for power-aware video coding systems. By exploiting the characteristics of video signal, two content-aware decision criteria are proposed to identify the complexity of motion vectors. Based on these two decision criteria as well as different combinations of various motion estimation algorithms, four different modes are proposed to allow dynamically varying the computation resources between different power constraints. Besides, the proposed

decision criteria also enable the maximization of quality under each power constraint by quality-driven diversity-based search approach. According to our simulation results, the proposed algorithm can effectively reduce the computation resources to 40%, 21%, and 3.73% with only 0.0036dB, 0.01dB, and 0.16dB average quality degradation, respectively. As a result, the proposed algorithm is well-suited for video coding systems that desire power-awareness feature.

## 6. REFERENCES

- [1] M. Bhardwaj, R. Min, and A. P. Chandrakasan, "Quantifying and enhancing power awareness of vlsi systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 6, pp. 757–772, Dec. 2001.
- [2] Y. H. Lu, L. Benini, and G. De Micheli, "Power-aware operating systems for interactive systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 2, pp. 119–134, Apr. 2002.
- [3] A. Sinha, A. Wang, and A. P. Chandrakasan, "Energy scalable system design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 2, pp. 135–145, Apr. 2002.
- [4] O. S. Unsal and I. Koren, "System-level power-aware design techniques in real-time systems," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1055–1069, July 2003.
- [5] P. Jain, A. Laffely, W. Bursleson, R. Tessier, and D. Goeckel, "Dynamically parameterized algorithms and architectures to exploit signal variations," *Journal of VLSI Signal Processing*, vol. 36, pp. 27–40, Jan. 2004.
- [6] T. Koga, K. Linuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motioncompensated interframe coding for video conferencing," in *Proc. NTC*, pp. C9.6.1–9.6.5, Nov. 1981.
- [7] L. M. Po and W. C. Ma, "A new center-biased search algorithm for block motion estimation," *IEEE Transaction on Image Processing*, pp. 23–26, Oct. 1995.
- [8] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *Information, Communications and Signal Processing*, pp. 9–12, Sept. 1997.
- [9] C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Transaction on Circuit and System for Video Technology*, pp. 349–355, May 2002.