An effective motion estimation scheme for H.264/AVC

Pengyu Liu, Kebin Jia
School of Electronic Information & Control Engineering, Beijing University of Technology, Beijing China, 100124
liupengyu@bjut.edu.cn

Abstract

H.264/AVC is the outstanding and significant video compression standard developed by ITU-T/ISO/IEC Joint Video Team (JVT). Motion estimation (ME) plays a key role in H.264/AVC, it concerns greatly on computational complexity especially when using the full search (FS) algorithm. Although many fast ME algorithms have been proposed to reduce the huge calculation complexity instead of FS, the ME still can not satisfy the critical real-time application's needs. In this paper, a fast integer pixel variable block motion estimation algorithm based on JVT which accepted UMHexagonS algorithm is presented for H.264/AVC encoder. With special considerations on the motion activity of the current macro-block, several techniques, i.e., adaptive search strategies have been utilized to significantly improve the video coding performance. The simulation results analysis shows that the proposed algorithm maintains an unnoticeable quality loss in terms of PSNR on average compared with FS and reduces nearly 20% motion estimation time compared with UMHexagonS while maintaining coding efficiency.

1. Introduction

H.264/AVC [1] is the most recent and promising international video coding standard developed by the ITU-T Video Coding Experts Group in conjunction with the ISO/IEC Moving Picture Experts Group. The new standard H.264/AVC aims at high-quality coding of video content at very low bit-rates and is designed for application in the areas such as broadcast, video-on-demand or multimedia streaming, multimedia messaging etc. over ISDN, DSL, Ethernet, LAN, wireless and mobile networks. Some new features and capabilities of the H.264 standard such as variable block size (7 types), quarter-sample-accuracy and multiple reference frames enable enhanced coding efficiency, compared with H.263, MPEG-2/4 advanced standards, up to 50% of bit-rate reduction can be achieved [2], but at the same time increase the complexity and computation load of motion estimation greatly in H.264/AVC encoder.

Motion estimation (ME) is the basic bandwidth compression method adopted in the video coding standards, while it is the most time consuming module. The block matching algorithm (BMA) is the most implemented one in real time for ME. BMA for ME is the mainstream algorithm for video compression, which has been adopted by many standards such as MPEG-1/2/4, H.263, H.264, etc. The key problem for BMA is to find the best matched motion vector in their reference frames for every macro-block.

The full search (FS) algorithm is well-known and widely used because it is simple and accurate. It exhaustively tests all the candidate points within a pre-defined searching area in a reference frame to get the best matched motion vector, the calculation is huge. Especially for H.264, it adopts some effective features mentioned above to improve the compression quality. So FS have to calculate all the search points in the search window for all the variable 7 types block sizes and 5 reference frames.

To reduce the calculation burden of full search block motion estimation, in the past years, many fast motion estimation algorithms were presented. They can be classified in two categories. One type is to reduce the search points, such as three step search algorithm (3SS) [3], four step search algorithm (4SS) [4], diamond search algorithm (DS) [5], hexagon-base algorithm (HEXBS) [6] etc. These fixed pattern search algorithms can reduce the search points largely and get a good image quality. But when the actual motion does not match the pattern well, the image quality will decrease. The other type of algorithm is to reduce the calculation for every search candidate. It uses sub-sampling method and partial calculation. Sub-sampling method is efficient but not accurate, while the partial calculation is accurate but not enough efficient.

The research work is supported by National Natural Science Foundation of China under Grant No. 60672050, the Natural Science Foundation of Beijing under Grant No.4062005 and Funding Project for Academic Human Resources Development No.00627.

978-0-7695-3278-3/08 $25.00 © 2008 IEEE
DOI 10.1109/IIH-MSP.2008.141

Recently several fast motion estimation algorithms were proposed for H.264. They combine many methods together and achieve both fast speed and good image quality. Especially, UMHexagonS[7] (Unsymmetrical-cross Multi-Hexagon-gird Search) algorithm has been adapted by H.264 reference software, claiming that nearly 90% computations can be saved on average compared with the fast full search algorithm in JVT reference software with a fairly good PSNR performance. However, based on many experiments results analysis, we find that the UMHexagonS algorithm is so complicated, speed-up of it is not very outstanding, and it can be speed-up further.

In this paper, we proposed a novel fast integer pixel motion estimation algorithm with adaptive search strategies for H.264 encoder according to the motion activity of the current macro-block. The proposed algorithm is much faster than UMHexagonS algorithm while maintaining an unnoticeable quality loss in terms of PSNR and Bit-rate compared with FS.

The rest of the paper is organized as follows: Section 2 discussed some background about UMHexagonS algorithm. Section 3 presents the proposed algorithm in detail. Section 4 shows the simulation results and some discussions. Finally, section 5 concludes the whole paper.

2. Overview of UMHexagonS algorithm

UMHexagonS algorithm has been accepted for integer-pixel block motion estimation by H.264 reference software. It can actually reduce the computational load for ME by reducing the number of candidate blocks within a search window. To achieve the high coding efficiency and avoid the local-minimum problem, UMHexagonS algorithm is widely conducted into two parts: the first is initial search center prediction and the second is use the hybrid of integer pixel search.

2.1 Initial search center prediction

Generally speaking, spatial and temporal predictions are the main mechanisms for motion estimation to find the motion vectors of the current block. These mechanisms generate four types of prediction means [8]: Median Prediction, Up-Layer Prediction, Corresponding block Prediction, and Neighboring Ref-frame Prediction. UMHexagonS algorithm uses these prediction means to predict the initial search center with high veracity.

2.2 Search strategies of UMHexagonS

The UMHexagonS algorithm can predict the motion vector accurately. There are three main steps in this search algorithm.

Step1. The search begins with the unsymmetrical cross search. Taking a search range of 16 the defined search window is shown with the search points in Fig.1 (step1).

Step2. The best match of step1 gives the center point for the step2 search which is the 5×5 full search with the search points are shown in Fig.1 (step2-1) and the uneven multi-hexagon-grid search. The search points for this search are shown in Fig.1 (step2-2).

Step3. The last search process uses extended hexagon-based search, composed of symmetrical-hexagon-grid search shown in Fig.1 (step3-1) and a small diamond search shown in Fig.1 (step3-2) until the center of the search pattern is the best candidate point.

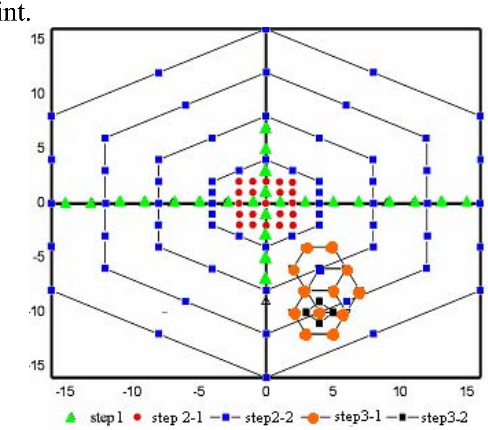


Fig1. Flow chart and the search pattern of UMHexagonS

To find the optimum motion vector in these steps, the UMHexagonS algorithm uses the hierarchical and hybrid motion search strategies. Obviously, the hybrid strategies exploit the irregularity of search patterns to find the best motion vector. The irregularity pattern search still causes a heavy computation overhead and limits the performance of the ME speed.

3. The proposed fast ME algorithm

In order to overcome the time-consuming motion estimation, reduce the search points and the computational complexity of ME in H.264/AVC, a fast motion estimation algorithm for variable block sizes by classifying motion activity of macro-block based on the UMHexagonS is proposed in this paper. The proposed method is composed of two parts.

3.1 Prediction of motion vector

Motion prediction is an important part in the ME. If we can get a good motion vector (MV) predictor, it means the search center we start is much nearer the best motion vector. It will need to calculate and compare less search points and have higher possibility to get the optimal MV. So we utilize four kinds of motion vector predictors, which are median motion vector predictor, up-layer motion vector predictor, corresponding block motion vector predictor and neighboring ref-frame motion vector predictor.

Median motion vector predictor is exploited the spatial relationship of neighbor macro-blocks. It is easy to find that neighbor macro-blocks have similar motion vector. So we can use the median value of the adjacent blocks on the left block (block A), top block (block B) and top-right (block C) of the current block (block E) shown in Fig.2 to predict the motion vector of the current block. The equation of the median predictor is described in equation (1).

$$Median\_preditor = median[MV_A, MV_B, MV_C] \quad (1)$$

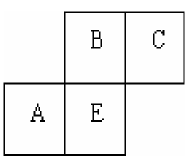


Fig2. Median motion vector predictor

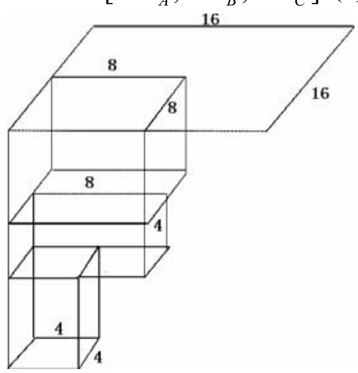


Fig3. Up-layer motion vector predictor

Up-layer motion vector predictor shown in Fig.3 is to utilize in the variable block sized motion estimation. In H.264, it will test all the 7 types of the current macro-block, choose the partition with lowest cost. We test big partition first, and then turn to smaller ones gradually. The motion vector search of big partition is a guide for the small partition. It shows the trend of the movement of the macro-block. So the 16×16 macro-block's motion vector can be referenced by 16×8 or 8×16 macro-block, etc. The equation of up-layer motion vector predictor is described in equation (2).

$$Up\_layer\_predictor = up\_layer[MV] \quad (2)$$

The moving track of a moving object is continuous in the major portion of the video sequence. Corresponding block motion vector predictor utilizes this characteristic to calculate the motion vector of the corresponding block in the last frame which is used as one motion vector candidate. The equation of corresponding block motion vector predictor is described in equation (3).

$$CP\_predictor = lastframe\_MV \quad (3)$$

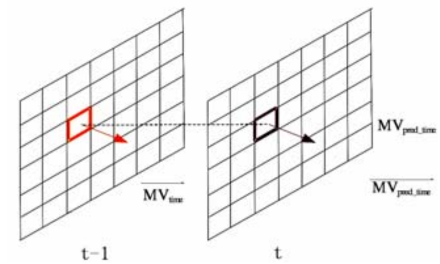


Fig4. Corresponding block motion vector predictor

Reference frame motion vector prediction is to utilize the temporal relation ship of the same macro-blocks in neighbor frames. The temporal neighbor of reference frame has the similar MV. So we can use this similarity to do motion vector prediction. The current block's motion vector in reference frame $t^{'}$ can be predicted by scaling the current block's motion vector in the reference frame $t^{'}+1$, equation (4) and Fig.5 shows the approach.

$$NR\_predictor = MV_{NR} \times \frac{t - t^{'}}{t - t^{'} - 1} \quad (4)$$

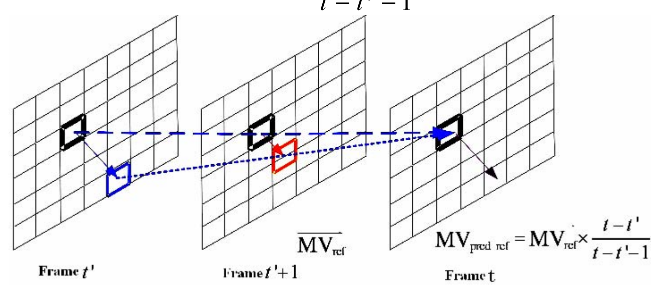


Fig5. Reference frame motion vector predictor

3.2 Flexible search criterion

After the initial search center having been predicted high accurately, adaptive search pattern will be selected according to the motion activity of the macro-block based on the original search pattern of UMHexagonS. This part includes three main techniques.

3.2.1 Modify the search pattern. For original search pattern of UMHexagonS algorithm in the step2-2, we can find that the uneven multi-hexagon-grid search adopts hexagon-grid search pattern with 16 points all the time. Assume that 16 points of the outmost layer can satisfy the search demand, it will be redundant for the benmost layer still adopts 16 points in the search criterion. So in the proposed algorithm, we modified the original uneven multi layers-hexagon-grid search pattern shown in Fig.6. In step2-2, the search points of each layer will increase with the extent of the search radius. From inside to outside layer, the search points are 8, 8, 10 and 16 respectively. The modified search pattern can maintain the search precision and reduce the unnecessary search points to enhance the efficiency of the motion estimation.

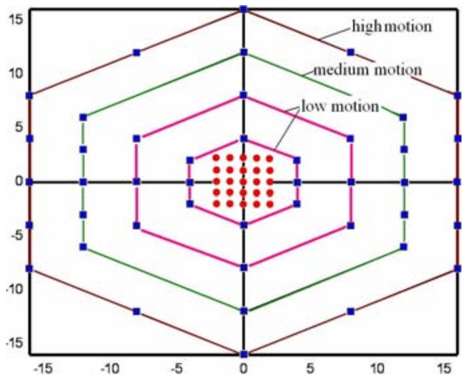


Fig6. Modified search pattern for step2-2

3.2.2 Search layer changed according to motion activity. According to the RD (Rate-Distortion) cost [9] of motion vectors in adjacent blocks, the motion search pattern is classified to three categories: low, medium, and high motion activity. The categories determine the search pattern respectively. Fig.6 shows the approach. The detail search strategy is as following.

In case of low motion activity, the uneven 2 layers-hexagon-grid search pattern is used because it is expected that the optimal current motion vector would be near the origin. In case of medium motion activity, the uneven 3 layers-hexagon-grid search pattern is performed. In the case of high motion activity, it is easily expected that the optimal motion vector would be far from the initial search center, so the uneven 4 layers-hexagon-grid search pattern is selected for search pattern.

The motion activity of the current macro-block is defined as follows.

$$\begin{aligned}
&\text{motion activity} = \text{low}; && \text{minRDcost} < (1+\varepsilon)\times\text{predminRDcost} \\
&\text{motion activity} = \text{medium}; && (1+\varepsilon)\times\text{predminRDcost} < \text{minRDcost} \\
& && < (1+\delta)\times\text{predminRDcost} \\
&\text{motion activity} = \text{high}; && \text{minRDcost} > (1+\delta)\times\text{predminRDcost}
\end{aligned} \tag{5}$$

In the above formula, minRDcost is the minimum RD cost of the current motion vector, predminRDcost expresses the minimum RD cost of the prediction motion vector. ε and δ are the adjustable coefficient.

$$\varepsilon = \frac{Bsize[blocktype]}{pred\ \min\ RD\cos t^2} - \alpha_{Radii\,1}[blocktype],$$

$$\delta = \frac{Bsize[blocktype]}{pred\ \min\ RD\cos t^2} - \alpha_{Radii\,2}[blocktype]$$

Here, array of $\alpha_{Radii1}[blocktype]$ and $\alpha_{Radii2}[blocktype]$ are defined as follows.

$$\alpha_{Radii1}[blocktype] = [-0.23,-023,-0.23,-0.25,-0.27,-0.27,-0.28]$$

$$\alpha_{Radii2}[blocktype] = [-2.39,-2.40,-2.40,-2.41,-2.45,-2.45,-2.48]$$

3.2.3 Selected 5×5 full search technique. In the original UMHexagonS algorithm, after doing initial search center prediction, it is expected that the optimal motion vector would be close to the origin search center, so it will do 5×5 full search primarily in step2-1. But according to the analysis above, if the motion activity is higher, the optimal motion vector is not near the origin search center, the 25 search points are unnecessary and time-consuming. So in the proposed algorithm, only when the motion activity belongs to the low motion, the 5×5 full search (shown in Fig.6) will selected.

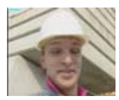
4. Experimental results and comparison

The proposed algorithm is integrated with JM 12.2 of the H.264 software for verification. We encoded six video sequences consist of different degrees and types of motion content in QCIF format, those are fast motion sequence Coastguard, middle-speed motion sequence Forman, slow motion sequence Miss America and Akiyo, with a lot of detail and scene horizontal motion sequence Mobile. The simulation experiment parameter setting is as follows. Each test sequence contains 100 frames, IPPP structure. The quantization parameter QP of the encoder is fixed at 28, B frame option is turn off, 5 reference frames, search range is 16 pixels, use Hadamard transform and CAVLC entropy coding.

We compared the performance of the proposed algorithm with that of FS and UMHexagonS algorithms. The simulation results show that the proposed algorithm consistently produces good performance of motion estimation time. Compared with UMHexagonS, it has saved nearly 20% motion estimation time, the PSNR is even higher than that of UMHexagonS and is compatible to that of FS, while maintaining the same coding efficiency level. Table1 shows the compared simulation results of motion estimation time, Bit-rate and PSNR for the three motion estimation algorithms.

$$\Delta\text{Time} = \frac{\text{ME time}\{\text{reference algorithm}\} - \text{ME time}\{\text{proposed algorithm}\}}{\text{ME time}\{\text{reference algorithm}\}} \times 100\%$$

Table1.The compared simulation results of the proposed algorithm with FS and UMHexagonS

Compared Algorithm / Simulation Results	Full Search algorithm			UMHexagonS algorithm			Proposed algorithm		
	ME-time (sec)	Bite-Rate (kbps)	PSNR-Y (dB)	ME-time (sec)	Bite-Rate (kbps)	PSNR-Y (dB)	ME-time (sec)	Bite-Rate (kbps)	PSNR-Y (dB)
1.Coastguard	131.989	244.85	34.01	332.678	245.66	34.01	26.648	245.16	34.02
2.Forman	128.340	132.62	36.44	24.251	131.50	36.43	20.198	131.87	36.43
3.Akiyo	128.581	29.47	38.25	10.397	29.53	38.28	9.001	29.45	38.28
4.Miss America	132.322	32.39	40.15	11.099	32.16	40.13	9.097	32.48	40.13
5.Mother & Daughter	131.103	46.99	37.44	14.472	46.88	37.40	12.084	47.11	37.43
6.Mobile	128.202	422.85	33.34	27.861	420.01	33.34	23.369	421.83	33.34
Performance comparison with FS, UMHexagonS algorithms on average	△ME-time %	Bite-Rate △%	PSNR-Y △(dB)	△ME-time %	Bite-Rate △%	PSNR-Y △(dB)	**Annotations:** ME-time: Motion Estimation time —: almost the same level "-" means reduce "+" means increase		
	-87%	—	—	-17%	—	+0.007			
Sequences	 1  2  3  4  5  6								

5. Conclusion

In this paper, we proposed a fast motion estimation algorithm based on UMHexagonS for H.264 encoder, which matches different motion contents of video sequence for macro-block. It can not only find the initial search center accurately, but also modified the search pattern of UMHexagonS and reduce the search points further to enhance the ME efficiency. Simulation experimental results indicate that, the proposed algorithm achieves the significant calculation burden reduction with almost the same level in PSNR performance compared with that of FS and UMHexagonS algorithms. The proposed method is a very efficient and robust ME algorithm for real-time video coding applications. The fast speed-up performance and unnoticeable quality losses make the proposed search criterion outperform the famous ME algorithm UMHexagonS.

6. Reference

[1] Joint Video Specification (ITU-T Rec. H.264/ ISO/IEC 14496-10 AVC) Joint Committee Draft. Joint Video Ream (JVT) of ISO/IEC MPEG and ITU-T VCEG, 7th Meeting: Pattaya, Thailand, 7-14 March, 2003.

[2] Gary J. Sullivan, Thomas Wiegand, Thomas Stochammer. Using the Draft H.26L Video Coding Standard for Mobile Applications. Proc. IEEE International Conference on Image processing, Thessaloniki, Greece, Sep. 2001, invited paper.

[3] J.Jain, A.Jain. Displacement measurement and its application in inter-frame image coding. IEEE Transactions on Communications, VOL. COM-29, 1799-1806, Dec. 1981.

[4] L. M. Po and W. C. Ma. A novel four-step search algorithm for fast block motion estimation. IEEE Transactions. Circuits Syst. Video Technol, Vol. 6, 313-317, Dec. 1996.

[5] S. Zhu and K.-K.M. A new diamond search algorithm for fast block matching motion estimation. Proc. Int. Conf. Inform., Commun., Signal Process, 292-296, Sept. 1997.

[6] C.Zhu, X.Lin, L. Chau. Hexagon-based search pattern for fast block motion estimation. IEEE Trans. On CSVT, Vol.12 No.5, 349-355, May. 2002

[7] Zhibo Chen, Peng Zhou, Yun He. Fast Integer Pel and Fractional Pel Motion Estimation for JVT. JVT-F017rl.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 6th meeting: Awaji, Island, JP,5-13 December, 2002.

[8] Zhibo Chen, Peng Zhou, Yun He. Fast motion estimation for JVT. JVTG016.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 7th Meeting: PattayaⅡ, Thailand, Mar. 7-14, 2003.

[9] G.J.Sullivan, T.Wiegand. Rate-Distortion Optimization for video compression. IEEE Signal Processing Magazine, vol. 15, No.6, 74-90, Nov.1998.