Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A FAST SEARCHING ALGORITHM BASED ON ADAPTIVE DIRECTION HYBRID TEMPLATE OF X264

[1]LUO YUAN, [2]YOU CHIHONG, [3]ZHANG YI, [4]TAN SHAOFENG

[1] Chongqing University of Posts and Telecommunications,
Engineering Research& Development Center of Information Accessibility,
Optical fiber communication technology key laboratory of Chongqing university,
Chongqing 400065, China
E-mail: [2]378945613@qq.com

ABSTRACT

By analysis the motion estimation algorithm of the open source H.264/AVC video codec, it is showed that the number of unnecessary search point will effect the speed of the algorithms. In this paper, a novel fast hybrid temp searching algorithm based on adaptive search direction of x264 is presented. An adaptive threshold for the matching macro block is set and a series hybrid search patterns were utilized adaptively to reduce the number of unnecessary search points. Moreover, improved MRACO PIXEL_SAD_C is used to reduce the computing time of SAD. Experiment results show that the novel algorithm is effectively.

Keywords: *motion estimation, X264, adaptive search, hybrid template*

1. INTRODUCTION

H.264 is the mainstream standard of video coding[1], and motion estimation is a key technology, which directs impact on the speed, quality, and bit rate of the encoded video encoding. The motion estimation time is about 40%-80% of the entire coding time, so reducing the coding time depends on optimizing the search strategy, the matching criteria and the initial search point selection and so on. This paper focuses on the matching criterion. Block matching method (BMA) is widely used matching algorithm while the full search (FS) is the most accurate estimates algorithm in BMA. FS need matches all pixels in the search area to achieves the best match, so it is not suitable for the real-time requirements. In order to reduce the computational complexity, lots of scholars put forward some fast algorithms such as three-step search (TSS), two-dimensional logarithmic law (2 LOGS), four-step search (4SS), diamond search method (DS)[2-3], hexagon search (HEXBS)[4] and (Unsymmetrical-cross Multi-Hexagon-gird)UMH[5] algorithms. However, these algorithms have problem of certain redundant or fall into the limitations optimal. In this paper, a novel fast hybrid temp searching algorithm based on adaptive search direction of x264 is presented. An adaptive threshold for the matching macro block is set and a series hybrid search patterns were utilized adaptively to reduce the number of unnecessary search points. Moreover, improved MRACO PIXEL_SAD_C is used to reduce the computing time of SAD. Experiment results show that the novel algorithm is effectively.

The following section discusses the video characteristics and presents the X264_AHT algorithm, in this section, some images are presented to show the different characteristics of video. Then we present the SAD algorithm optimization. Section 4 shows some experimental results. Section 5 summarizes our work.

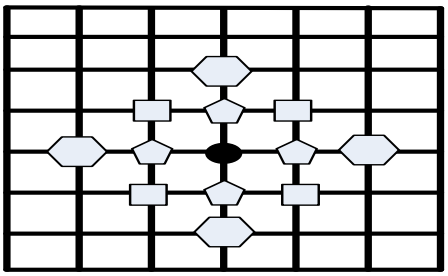
2. FAST SEARCH ALGORITHM DESIGN BASED ON ADAPTIVE DIRECTION HYBRID TEMPLATE

2.1 Video Characteristic

Most video has two characteristics. Such as center-biased characteristics[6-9] and different videos have different motion intensity. The center of a 5×5 square area, which includes 81.81% of motion vectors, even if Stefan_qcif motions violently, it also has 72 %[10], where the 45.4 percentage is zero vector, it shows that most of video is stationary or quasi-static. Fig 1 showed 5×5 square area of MV distribution. Thus, the diamond-shaped section and the cross-shaped part are the priority area to start search.

Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(● A ⬠ B □ C ⬭ D)

Fig 1 5×5 *square area of MV distribution*

According to the different motion intensity, the video can be classified as the stationary or quasi-stationary state, smaller movement, more intense video motion and strenuous motion.

Based on the video characteristics, this paper presents a hybrid adaptive direction template search algorithm (X264_AHT), using different search strategies and search template in view of the different movement types, and set a threshold value to judgment.

2.2 X264_AHT algorithm

X264_AHT algorithm set a threshold value SAD_THRESH (500) to reduce the search time. Motion macro block type is decided prediction of motion vector MV= (V_{xi}, V_{yi}),

$$MV\_JUDGE_i = |V_{xi}| + |V_{yi}|;$$

$$MAX\_MV\_JUDGE = \max(MV\_JUDGE_i);$$

When MAX_MV_JUDGE is equal to zero stands for static or quasi static, determine whether exit directly through the threshold.

When MAX_MV_JUDGE is equal to 1 stands for small movement, then motion vector is distribution around the center, a small diamond search can be as shown in figure 2.

When MAX_MV_JUDGE is between 1 and 4 stands for more intense video motion, different macros block have different characteristics[11], according to the different blocks can adopt different search modes.

When MAX_MV_JUDGE is more than 4 stands for violent movement. Asymmetric diamond search is used to judge the best position of point, just as shown in figure 6 or figure 7. When the best point fall into a horizontal axis, the cross search Pattern is used, when the best point fall into vertical axis, asymmetric diamond search is used as shown in figure 8 or figure 9.

Search steps are as follows:

Step1: turn step2 if MV_JUDGE is equal to zero, the Judgment motion type is static or quasi static block, turn step3 if MV_JUDGE equal to 1 judgment macro block is a small movement; turn step4 if MV_JUDGE greater than 1 less than 4 judgment macro block is a violent sports; turned step5 if MV_JUDGE greater than 4 macro block is violent;

Step2: turn step6 if the value less than the threshold value, if not, turn step3;

Step3: repeated small diamond search (SDS) until MBD point in the center, turn step6;

Step4: use middle cross search if macro block is 16×16 or 8×8 in Figure 3, turn step3 if MBD point in the center into the small diamond search, otherwise continue to the cross search; turn step3 if a macro block is 16×8 or 8×4 use Horizontal-asymmetric cross search in Figure 4, turn step3 if MBD point in the center into the small diamond search, otherwise continue to use level asymmetric cross search; use Vertical-asymmetric cross search if a macro block is 8×16 or 4×8 in Figure 5, turn step3 if MBD point fall into the small diamond search in the center, otherwise continue to use vertical asymmetric cross search; direct access to hexagon search if a macro block is 4×4.

Step5: use right asymmetric diamond search when forecasting point x coordinate greater than 0, use left asymmetric diamond search when the x coordinate less than 0, turn step3 when the MBD point in the center, into the small diamond search;

Use the cross search when the MBD point fall into a horizontal axis, turn step3 when MBD point in the center, fall into the small diamond search; otherwise repeat use the cross search. Turn step3 when the best point fall in the longitudinal axis, when the y coordinate greater than 0 the asymmetric diamond search, when the y coordinate less than 0 adopt asymmetric diamond search, when the MBD point in the center, into the small diamond search, otherwise repeat use up asymmetric diamond or down asymmetric diamond search.

Step6: end search.

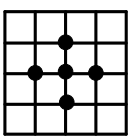


Fig 2 small DA pattern

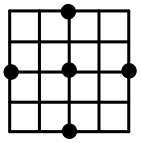


Fig 3 Middle cross pattern

Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

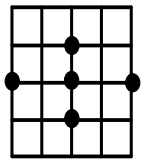


Fig4 H-Asymmetrical-cross

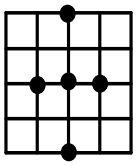


Fig5 V-Asymmetrical-cross

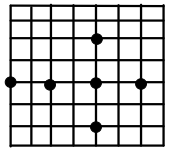


Fig 6 L-Asymmetrical-DA

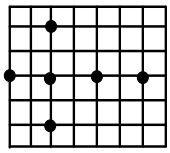


Fig 7 R-Asymmetrical-DA

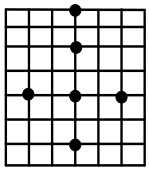


Fig 8 U-Asymmetrical-DA

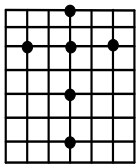


Fig 9 D-Asymmetrical-DA

3. SAD ALGORITHM AND CODE OPTIMIZATION

3.1 Macro Improvement of PIXEL_SAD_C

PIXEL_SAD_C macro is used to calculate the matching errors of i_sum which is stands for SAD value, $lx \times ly$ stands for the type of macro block. $lx \times ly$ time cycle is need to calculate traditionally. It would cause unnecessary waste. bcost is stands for the best SAD, Local variables bcost in the function of x264_me_search_ref become global variables referenced in # DEFINE PIXEL_SAD_C (name, lx, ly), and together with the judgment, when i_sum> = bcost, return i_sum value and exit, otherwise the loop continues.

3.2 Optimization of X264

When determining the best starting point, there is great probability that the best point is the (0, 0). COST_MV(0， 0) can be calculated from the function of x264_me_search_ref. When (0, 0) is the best point, it means that it needs no calculate, thus, plus a judgment statement if((bmx!=0)&&(bmy!=0)) COST_MV(0, 0) in the code. In it, bmx, bmy represents the x and y coordinates of the best point. Executed COST_MV(0，0) when the best starting point is not (0，0), otherwise skip.

The total flow chart shown in figure 10.

4. RESULTS AND ANALYSIS

The reference model of H.264- X264 is used to verification. The proposed algorithm is mark as AHT. AHT is compared with HEX, UMH and ESA in X264, use the three kinds of sequence akiyo_qcif, foreman_qcif and Stefan_qcif which are standard sequences to test, the video sequence format is QCIF, frame rate is 25, coding format is IPPPPPP. The tables 1-3 give results of the parameters' comparison between the AHT and standard algorithms in different sequence. The tables 4-6 give results of parameters' comparison before and after optimization of macro PIXEL_SAD_C in different sequence..

Table 1 the result of testing various motion estimation based on sequence akiyo_qcif

Performance index	HEX	UMH	ESA	AHT
Frames per second (fps)	36.21	31.40	26.34	43.05
Bitrates (kb/s)	25.24	25.16	25.29	25.34
PSNR	39.974	40.001	39.996	39.964

Table 2 the result of testing various motion estimation based on sequence foreman_qcif

Performance index	HEX	UMH	ESA	AHT
Frames per second (fps)	15.05	10.82	7.34	16.82
Bitrates (kb/s)	121.22	121.03	120.85	121.80
PSNR	37.069	37.076	37.085	36.067

Table 3 the result of testing various motion estimation based on sequence stefan_qcif

Performance index	HEX	UMH	ESA	AHT
Frames per second (fps)	3.82	2.15	1.22	4.00
Bitrates (kb/s)	1008.4	1003	1016	1013.38
PSNR	36.36	36.37	36.322	36.359

In table 1, table 2 and table 3, the dates show that the new algorithm has exceeded the common else. Firstly, handling the akiyo_qcif with lower motion complexity, simple background and short reduce of PSNR, the new approach cuts down the process time distinctly in order to be 13% faster than the hexagon algorithm. Besides, keeping the same PSNR, the new approach processes the foreman_qcif 11% faster than hexagon algorithm still. Secondly, compared with UMH and ESA algorithm, the new approach sacrifices 0.01db and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

0.02db PSNR to make an approximate 40% and 60% superiority in process time as well. Thirdly, keeping the same PSNR, the new algorithm processes the stefan_qcif, in large range of motion, 10% faster than the hexagon algorithm. Moreover, compared with the UMH and ESA, after abandoning about 0.02db PSNR, the new approach handles the stefan_qcif, in large range of motion, 45% and 70% faster.

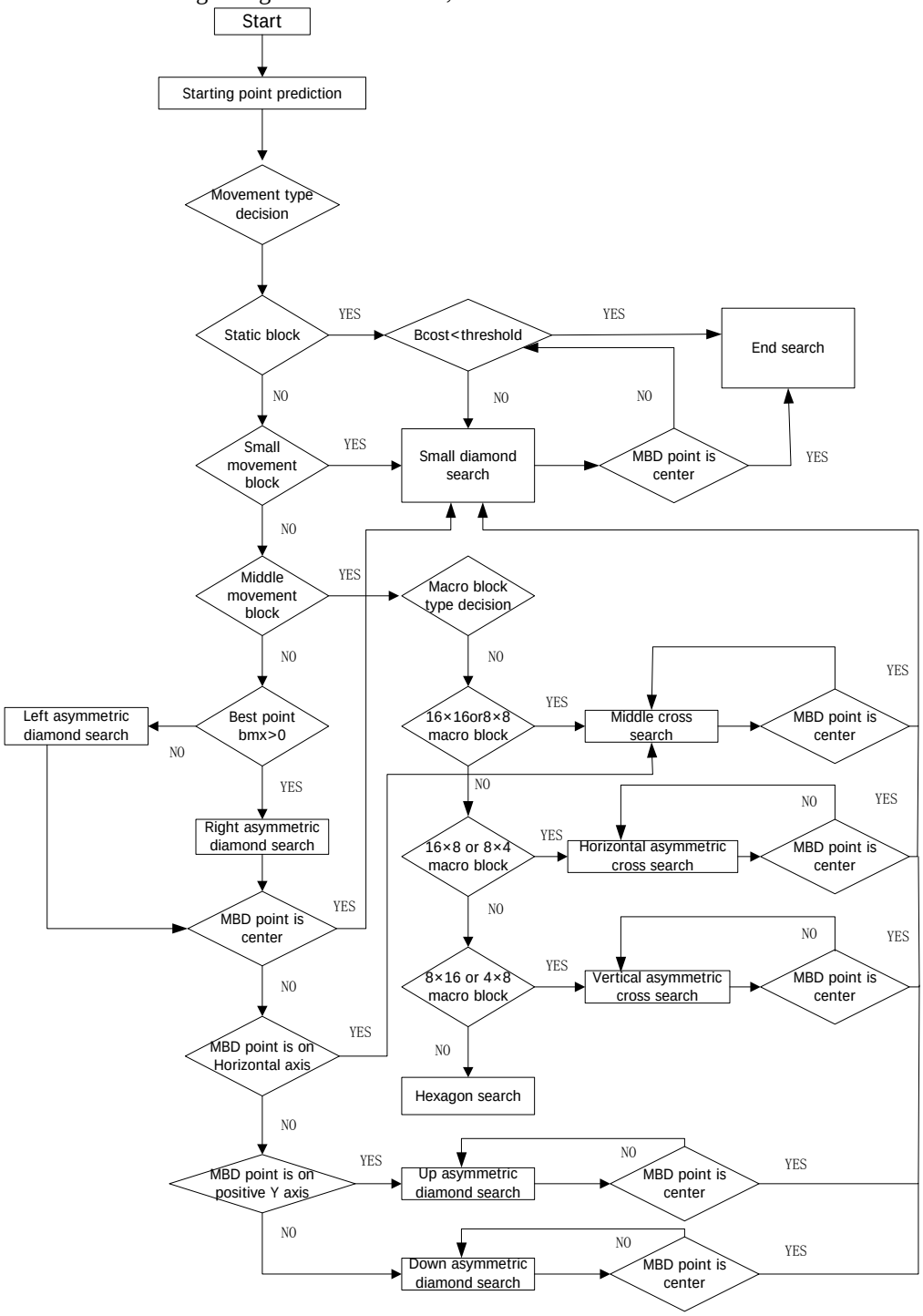


Fig10 X264_AHT Algorithm Flow

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4 the result of testing various motion estimation before and after optimization macro PIXEL_SAD_C based on sequence akiyo_qcif

Before	HEX	UMH	ESA	AHT
Frames per second (fps)	36.21	31.40	26.34	43.05
Bitrates (kb/s)	25.24	25.16	25.29	25.34
PSNR	39.974	40.001	39.996	39.964
After	HEX	UMH	ESA	AHT
Frames per second (fps)	41.52	38.48	37.83	45.77
Bitrates (kb/s)	25.6	25.31	25.2	25.20
PSNR	39.986	39.998	39.965	39.976

Table 5 the result of testing various motion estimation before and after optimization macro PIXEL_SAD_C based on sequence foreman_qcif

Before	HEX	UMH	ESA	AHT
Frames per second (fps)	15.05	10.82	7.34	16.82
Bitrates (kb/s)	121.22	121.03	120.85	121.80
PSNR	37.069	37.076	37.085	36.067
After	HEX	UMH	ESA	AHT
Frames per second (fps)	16.36	12.81	15.71	17.06
Bitrates (kb/s)	121.62	120.85	123.05	122.24
PSNR	37.050	37.055	37.031	37.043

Table 6 the result of testing various motion estimation before and after optimization macro PIXEL_SAD_C based on sequence stefan_qcif

Before	HEX	UMH	ESA	AHT
Frames per second (fps)	3.82	2.15	1.22	4.00
Bitrates (kb/s)	1008.42	1003.10	1016.18	1013.38
PSNR	36.36	36.37	36.322	36.359
After	HEX	UMH	ESA	AHT
Frames per second (fps)	4.08	3.01	3.62	4.22
Bitrates (kb/s)	1008.80	1003.23	1016.42	1011.73
PSNR	36.329	36.371	36.322	36.351

From the above three table, the test results can be concluded that the various X264 motion estimation algorithm speed has been greatly improved after SAD algorithm optimized, especially the ESA, but the PSNR are almost consistent.

5. CONCLUSION

Considering the characteristics of the video, the X264_AHT algorithm use different search strategies and search template for different movement types. Different characteristics of the different macro block type are considered in the medium movement types. Thus, adopt different search template to the（16×16，8×8）、（16×8，8×4）、（8×16，4×8）and（4×4）. Vector probability distribution have been fully considered in the violent movement type of motion, and set a threshold value judgment in advance, so the search speed is improved greatly. The algorithm can obviously improve the search efficiency and rarely have any impact on the quality of the premise.

The disadvantage of this paper is that it ignores the Human Vision System, therefore can't grab the Region of Interest (ROI). So the future is to adopt different coding for different region based on the Human Vision System.

Acknowledgement: this paper is supported by Scientific Research Project of Chongqing Educational Committee (KJ120519), National Natural Science Foundation of China (510 75420).

REFRENCES:

[1] Wiegand T.Sullivan G.J.Luthra A.Overview of the H.264/AVC video coding standard[J].IEEE Transactions on circuits and System for video Technology,2003,13(7):560-576.

[2] Zhu S,Ma K K.A new diamond search algorithm for fast block matching motion estimation [J].IEEE Transactions on Image Processing 2000,9(2):287-290

[3] Tham J.Y.Ranganath S.Kassim A.A.A novel unrestricted center biased diamond search algorithm for block motion estimation[J].IEEE Transaction on circuits and Systems for video Technology,1998,8(4):369-377.

[4] Zhu C.Lin X.Chau L.-P.Hexagon-based search pattern for fast block motion estimation[J].IEEE Transaction on circuits and Systems for video Technology,2002,12(5):345-355.

[5] Chen Z.Xu J.He Y.Zheng J.Fast integer pel and fractional-pel motion estimation for H.264 / AVC [J]. Journal of Visual Communication and image representation,Journal of Visual Communication and image representation,2006,17(2)：264-290.

[6] Liang yan,Liu Wenyao Adaptive Cross quasi diamond search algorithm based on initial search point algorithm[J]Optics and precision Engineering,2005,13(2):163－166

[7] Liu Haihua ,Lei Yi,Xie Changsheng Fast Block-Matching Motion Estimation Based on Dual Cross Search A Algorithm[J], Journal of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Computer Research and development 2006,43(9)：1666-1673

[8] Lin Zhaohua,Xie cunxi Unit-Rood Fast Motion Estimation Algorithm Based on Starting-Point Prediction[J],Journal of South China University of Technology (Natural Science Edition) 2007,35(8):53-58

[9] C H cheung,L M Po A novel small cross diamond hexagonal search algorithm for fast video coding and videoconferencing applications[C],IEEE Int'l conf on Image Processing, New York,2002

[10] Xu J, Chen Z, He Y. Efficient fast ME predictions and early termination strategy based on H.264 statistical characters [J].In the 2003 IEEE Pacific-Rim Conference On Multimedia Dec.2003,1(6):218-222

[11] Li Yufeng, Xiao Jufei, Shen Lianfeng Motion Estimation Algorithm Research Based on H.264 Encoding [J] .TelecommunicationsScience2012 (8):64-68