A 3D Predict Hexagon Search Algorithm for Fast Block Motion Estimation on H.264 Video Coding

Tsung-Han Tsai and Yu-Nan Pan

Department of Electronic Engineering, National Central University, Chung-Li, Taiwan

{han , kobbe}@dsp.ee.ncu.edu.tw

Abstract

In the upcoming video coding standard, MPEG-4 AVC/JVT/H.264, motion estimation is allowed to use multiple references and multiple block sizes to improve the rate-distortion performance. However, full exhaustive search of all block sizes is computational intensive with complexity increasing linearly to the number of allowed reference frame and block size. In this paper, a novel search algorithm, Three Dimensional Predict Hexagon Search (3DPHS) is proposed. The 3DPHS patterns depend on the characteristics of motion vector distribution. It can predict the hexagon search pattern in horizontal or vertical direction. The proposed algorithm also considered the characteristics of multiple references and multiple block sizes in H.264. It can save all the same level search points for higher definition video sequences. The analysis shows that the speed improvement of 3DPHS over the Diamond Search (DS) and the Hexagon Based Search (HEXBS) is about 58% and 53% respectively.

1. Introduction

H.264 will be a new international video coding standard of ITU-T and is jointly being made by ITU-T Video Coding Experts Group (VCEG) and ISO/IEC MPEG video group, named as Joint Video Team (JVT). The main goals of this standardization are to develop a simple and straightforward video coding design with enhanced compression performance. Compared to MPEG-4 advanced simple profile, up to 50% of bit-rate reduction can be achieved. Such improvement is come from the prediction part [1]. Especially, motion estimation at quarter-pixel with variable block size and multiple reference frames greatly reduces prediction errors.

Fig. 1 illustrates the H.264 encoder which can use seven different block types and multiple references for motion estimation and compensation. Simulation result shows that using seven different block sizes can save more than 15% of bit rate compared with only using 16x16 block size. It can also save about 5~10% of bit rate when using multiple reference frames. However, the processing time increases linearly with the number of block type and multiple reference frame used.

By exhaustively testing on all the candidate blocks (examine all seven block modes and reference frames), Full Search (FS) algorithm gives the global Minimum Block Distortion (MBD) position which corresponds to the best matching block at the expense of highly computation. To overcome this defect, many fast BMAs are developed such as Diamond Search (DS) [2], Hexagon Based Search (HEXBS) [3] and Predict Hexagon Search (PHS) [7].

In this paper we propose a new Three Dimensional (3D) predict hexagon search algorithm. Compared with the original PHS [7], the 3D consideration indicates the three critical predictions on the object movement in vertical and horizontal directions, the search center with variable block sizes and the search center with multiple reference frames. Two different search patterns are used to reduce the search points. In addition, the analysis of motion vector distribution is used to make a local search range. The extend method can cause our search center more closely to the object we want to search. The results show our proposed 3DPHS algorithm reduces the search points greatly compared to DS and HEXS, while still maintain the same PSNR with image quality. This paper is organized as follows. In Section 2 the characteristics of motion vector distribution, multiple references and multiple block sizes are analyzed. The PHS algorithm is described in Section 3. In Section 4 the simulation results are reported and the conclusions are given in Section 5.

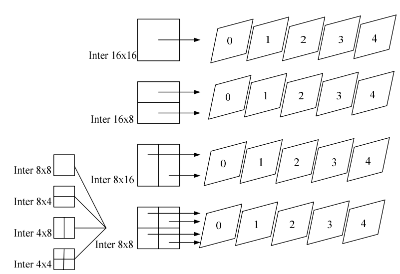


Fig. 1 Multiple references and multiple block sizes in H.264.

2. Characteristics of Motion Estimation on H.264

2-1 Characteristics of motion vector distribution

Table 1 shows the analysis on the motion vector probabilities based on 8 well-known sequences. These sequences are classified to QCIF and CIF size. Table 2 shows the average MV distribution using 8 QCIF/CIF test image sequence with search range of ±16. The row topic means the absolute of MVx, the column topic means the absolute of MVy. From this analysis we can see that the probability of the zero motion vectors is about 65%, and the probability to obtain the best motion

0-7803-9332-5/05/$20.00 ©2005 IEEE

vector within the 5x5 area is more than 85%. On the other hand we can see that the probability for the rood side is higher than the probability for the corner side. By using this characteristic, we can search the rood side in ±2 first to save the search points.

Table 1 Image sequence used in analysis.

Format	Sequence
QCIF (176x144, 200 frames)	Tennis Foreman Salesman Mobile
CIF (352x288, 200 frames)	Coastguard Stefan Silent Weather

Table 2 Average distribution at absolute distance |r| from the center of the search grid by using 8 QCIF/CIF image sequence for search window ±16.

r	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	64.934	6.722	1.974	1.224	0.479	0.362	0.332	0.258	0.110	0.129	0.067	0.001	0.092	0.092	0.049	0.098	0.104
1	6.304	1.81	0.516	0.301	0.153	0.079	0.073	0.079	0.030	0.043	0.018	0.141	0.012	0.024	0.030	0.018	0.012
2	2.312	0.750	0.375	0.215	0.092	0.073	0.036	0.006	0.018	0.024	0.024	0.018	0.012	0.024	0.001	0.024	0.006
3	1.550	0.596	0.227	0.141	0.104	0.036	0.012	0.030	0.024	0.024	0.006	0.000	0.018	0.006	0.006	0.006	0.024
4	0.719	0.381	0.172	0.104	0.024	0.018	0.006	0.012	0.006	0.006	0.018	0.018	0.018	0.006	0.001	0.001	0.006
5	0.455	0.227	0.129	0.067	0.018	0.049	0.018	0.012	0.018	0.006	0.001	0.006	0.001	0.018	0.006	0.001	0.012
6	0.369	0.159	0.073	0.116	0.030	0.030	0.001	0.012	0.012	0.012	0.012	0.012	0.006	0.006	0.001	0.001	0.006
7	0.319	0.123	0.055	0.086	0.018	0.024	0.001	0.012	0.001	0.001	0.006	0.001	0.006	0.001	0.006	0.001	0.001
8	0.141	0.049	0.116	0.049	0.018	0.030	0.006	0.001	0.012	0.001	0.001	0.001	0.006	0.001	0.006	0.001	0.006
9	0.110	0.024	0.061	0.018	0.012	0.006	0.018	0.012	0.006	0.001	0.001	0.006	0.001	0.001	0.001	0.001	0.001
10	0.123	0.055	0.018	0.024	0.018	0.012	0.006	0.012	0.006	0.006	0.001	0.006	0.006	0.012	0.001	0.001	0.001
11	0.116	0.030	0.055	0.049	0.006	0.012	0.018	0.006	0.001	0.001	0.006	0.001	0.001	0.006	0.006	0.001	0.006
12	0.104	0.024	0.012	0.012	0.006	0.001	0.012	0.001	0.001	0.001	0.001	0.006	0.001	0.006	0.001	0.001	0.001
13	0.043	0.043	0.043	0.043	0.006	0.001	0.006	0.006	0.006	0.001	0.006	0.006	0.001	0.001	0.006	0.001	0.006
14	0.098	0.024	0.024	0.043	0.012	0.006	0.012	0.012	0.006	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
15	0.067	0.024	0.006	0.001	0.006	0.001	0.006	0.018	0.012	0.012	0.001	0.001	0.001	0.006	0.006	0.001	0.001
16	0.061	0.018	0.001	0.049	0.012	0.018	0.018	0.001	0.006	0.001	0.001	0.006	0.0061	0.006	0.006	0.006	0.006

2-2. Characteristics of Multiple References and Multiple Block sizes

In H.264, we can use more than one reference frame. Fig. 2 shows the relationship of multiple references. As shown in the figure, the object moved from the up-right at previous frame (N-2) to the bottom-left at current frame (N). In the encoding process, we should refer the frame (N-1) first, and then refer the frame (N-2). If the distance of the object from current frame (N) to the previous frame (N-1) is X, and the distance of the object from current frame (N) to the previous frame (N-2) is Y, then we can get that the distance of the object from previous frame (N-1) to the previous frame (N-2) is Y-X. Thus, if we set the search center at the coordinate which we find at previous frame, we can save more search points.

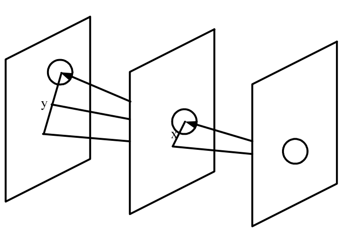


Fig. 2 The relationship of multiple references

There are seven types of block size can be used in H.264 encoding process. So it must have some relationships between the different kinds of block size. Fig. 3 shows the relationship of multiple block size. As we known, the 16x16 block size can be divided to 16x8, 8x16 and 8x8. The shape of upper-left 8x8 is the same as the left side of upper 16x8, up side of left 8x16 and the up-left side of 16x16. Thus, if the best coordinate of the upper-left 8x8 is found, the best coordinate of the upper 16x8, left 8x16 and 16x16 may be close to the best coordinate of the upper-left 8x8. If we utilize this characteristic we can save more search points.

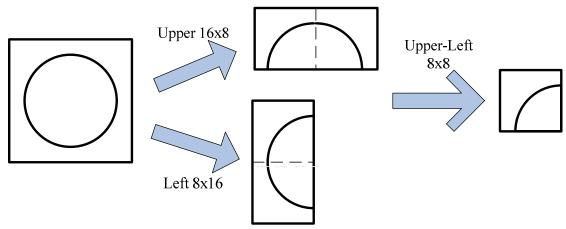


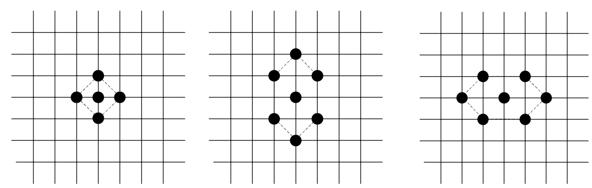
Fig. 3 The relationship of multiple block size

3. 3D Predict Hexagon Search algorithm

In combination of the above three results, we proposed a new search pattern which can search the rood side in ±2 first and also predict the shape of the hexagon in the same time.

3-1. Predict Hexagon Search Pattern

The predict hexagon search pattern is shown in Fig. 4. Fig.3 (a) shows a small predict hexagon search pattern (SPHSP) contains 5 checking points (left, right, up, down dots with distance 1 around the center dot). This is applied in the first, second and final step search pattern.



(a) (b) (c)

Fig. 4 PHS pattern: (a) Small PHS pattern (SPHSP) (b) Vertical Large PHS pattern (Vertical LPHSP) (c) Horizontal Large PHS pattern (Horizontal LPHSP).

Fig.3 (b) and (c) illustrate the large predict hexagon search pattern which contains 7 checking points. The vertical large predict hexagon search pattern (LPHSP) is used when the object is moving in the vertical direction. It excludes the right and left edge points. The horizontal LPHSP is used when the object is moved in the horizontal direction. It excludes the up and down edge points.

3-2. Algorithm Development

With the design of the predict hexagon search point configuration, we develop the algorithm flow as follows:
Step 1. The SPHSP with 5 search points is used. If the MBD point is found at the center, the final point will be

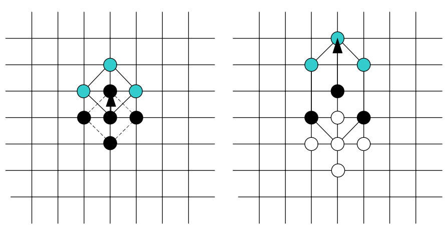
the center point; otherwise the point which was the MBD point will be changed to the center point and proceed to step 2. This case is shown in Fig. 4(a). If the MBD point is up or down dot, vertical LPHSP will be used in step 3. On the other hand, if the MBD point is left or right dots, Horizontal LPHSP will be used in step 3.

Step 2. The SPHSP is still used. Three new candidate points are checked and the MBD point is identified again. If the MBD point is the center point, the final point will be the center point; otherwise proceed to step 3.

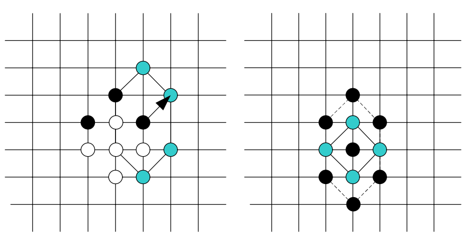
Step 3. Switch the search pattern from SPHSP to suitable LPHSP. For Fig.4 (b) three new candidate points are checked and the MBD point is identified again. For Fig. 4 (c) the new candidate will be four. If the MBD point is the center point, than go to step 5; otherwise proceed to step 4.

Step 4. With the MBD point in previous search step as the center, a new large hexagon is generated. Three new candidate points are checked and the MBD point is identified again. If the MBD point is the center point, than go to step 5; otherwise repeat this step continuously.

Step 5. Switch the search pattern form LPHSP to SPHSP. In Fig. 5 (d), four new candidate points are evaluated to compare with the current MBD point. The new MBD point is the final point of the motion vector.



(a) Case1: SPHSP->SPHSP (b) Case2: SPHSP->LPHSP



(c) Case3: SPHSP->LPHSP (d) Case4: LPHSP->SPHSP

Fig. 5 Four special cases of checking points overlapping when the MBD point found in the previous search step.

3-3.Extend method

Although we can save a lot of search points by using the proposed algorithm, but it is not enough for higher definition video sequences. Because we focus on the rood side in ±2 range first, it may cause we must pay more search points for higher definition video sequences. So we proposed an extend method to predict the search center and let our search center more closely to the macroblock we are interest. To consider the complexity, we just use the best-coordinate of 8x8 to have a simply calculation and the characteristics of multiple reference frames to predict the search center.

- For the first reference frame, we calculate the MV of the four 8x8 blocks and get the best coordinate first. Then, we calculate the search point for other 6 types of block sizes.
- For the block size 16x16, we use 4 coordinates of the block size 8x8 to calculate as the Fig. 6. The equation as follow:

Search Center$_{16x16}$= (best-coordinate$_{upper-left8x8}$+ best-coordinate$_{upper-right8x8}$ + best-coordinate$_{lower-left8x8}$+ best-coordinate$_{lower-right8x8}$)/4………………………………. (1)

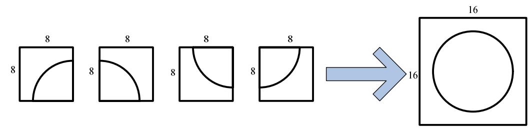


Fig. 6 To calculate the search center of block size 16x16.

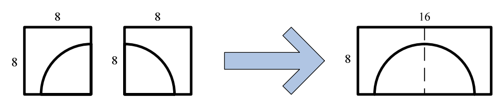
- For the block size of 16x8, we use 2 coordinates of the block size 8x8 to calculate as shown in Fig. 7 (a). For the block size of 8x16, we use 2 coordinates of the block size 8x8 to calculate as Fig. 7 (b). The equation as follow:

Search Center$_{upper16x8}$= (best-coordinate$_{upper-left8x8}$+ best-coordinate$_{upper-right8x8}$)/2……………………………… (2)

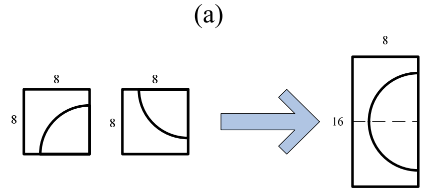
Search Center$_{lower16x8}$= (best-coordinate$_{lower-left8x8}$+ best-coordinate$_{lower-right8x8}$)/2……………………………… (3)

Search Center$_{left8x16}$= (best-coordinate$_{upper-left8x8}$+ best-coordinate$_{lower-left8x8}$)/2 ……………………………….(4)

Search Center$_{right8x16}$= (best-coordinate$_{upper-right8x8}$+ best-coordinate$_{lower-right8x8}$)/2……………………………... (5)



(a)



(b)

Fig. 7 To calculate the search center of (a)block size 16x8 (b)block size 8x16.

- For the block size of 8x4, 4x8 and 4x4, we just set the search center as the best-coordinate of block sizes of 8x8.
- After the first reference frame, we set the search center as the best-coordinate of the same block size at previous reference frame for the other reference frames.

4. Simulation Result

The proposed 3D PHS algorithm is simulated using 3 types (SD, cif, qcif) total 12 popular video sequences. They consist of different degrees and types of motion content. The search window is ±16, the reference frame is 5 and the block type is 7. The 3D PHS is compared with FS, DS, HEXS and original PHS.

Table 3 shows the average number of search points per macroblock. For block-matching motion estimation, computational complexity is mainly dependent on the average number of search points required for each macroblock. Based on the simulation results, we can see that the PHS can save a lot of search points. But if we use the 3DPHS, it can save more search points.

The average PSNR per frame shown in Table 4. The average PSNR of 3DPHS is almost the same compared with PHS, DS, HEXBS even FS.

Table 3 Average Number of search points per MB with different methods and different image sequences (search range is ±16, reference frame is 5, block type is 7).

	Night (720x480)	Sailormen (720x480)	Crew (720x480)	Harbour (720x480)	Coastguard (352x288)	Stefan (352x288)	Silent (352x288)	Weather (352x288)	Tennis (176x144)	Foreman (176x144)	Salesman (176x144)	Mobile (176x144)
FS	241997.58	241997.58	241997.58	241997.58	241997.58	241997.58	241997.58	241997.58	241997.58	241997.58	241997.58	241997.58
DS	4282.722	4594.312	4627.141	3945.963	3918.376	3804.037	3212.921	2895.779	3487.197	3742.133	2931.016	3432.631
HEXBS	3258.385	3745.198	3521.157	3274.705	3225.094	3136.55	2732.830	2603.785	2992.038	3113.066	2648.269	2996.991
PHS	2352.327	2796.472	2711.830	2263.762	2196.878	2167.096	1503.139	1236.635	1794.210	2148.029	1229.164	1845.757
3DPHS	1935.19	2192.531	2242.925	2018.737	2080.515	2049.626	1404.818	1207.692	1608.955	1860.523	1179.590	1709.535

Table 4 Average PSNR per frame (search range is ±16, reference frame is 5, block type is 7).

	Night	Sailormen	Crew	Harbour	Coastguard	Stefan	Silent	Weather	Tennis	Foreman	Salesman	Mobile
FS	35.971	35.061	36.587	34.88	34.42	34.96	35.87	36.70	35.62	35.53	35.48	32.72
DS	35.961	35.038	36.570	34.814	34.13	34.84	35.83	36.67	35.42	35.29	35.41	32.66
HEXBS	35.96	35.029	36.560	34.810	34.13	34.89	35.82	36.68	35.41	35.45	35.42	32.64
PHS	35.963	35.015	36.552	34.796	34.04	34.79	35.83	36.64	35.37	35.4	35.4	32.59
3DPHS	35.927	35.026	36.545	34.780	33.878	34.63	35.57	36.44	35.13	35.29	35.29	32.51

Table 5 Average speedup (search range is ±16, reference frame is 5, block type is 7).

	Night	Sailormen	Crew	Harbour	Coastguard	Stefan	Silent	Weather	Tennis	Foreman	Salesman	Mobile
Avg. speedup over DS by PHS (%)	45.07	39.13	41.39	42.63	43.93	43.03	53.21	57.26	48.54	42.59	58.06	46.22
Avg. speedup over HEXBS by PHS (%)	27.80	25.33	22.98	30.87	31.88	30.90	44.99	52.50	40.03	30.99	53.58	38.41
Avg. speedup over DS by 3DPHS (%)	54.81	52.27	51.54	48.84	46.90	46.11	56.27	58.29	53.86	50.28	58.06	50.19
Avg. speedup over HEXBS by 3DPHS(%)	40.60	41.45	36.30	38.35	35.48	34.65	48.59	53.61	46.22	40.23	55.45	42.95
Avg. speedup over PHS by 3DPHS (%)	17.73	21.59	17.29	10.82	5.29	5.42	6.54	2.34	10.32	13.38	4.03	7.38

The speed improvement in percentage of the proposed 3D PHS over PHS, DS and HEXBS is listed in Table 5. We can see that, the speedup percentage for higher definition video sequences is about 23% ~30% which is less than the speedup percentage for lower definition video sequences buy using PHS. But if we use 3DPHS the speedup percentage for higher definition video sequences is almost same level compare with lower definition video sequences.

Fig. 8 plots a frame-by-frame comparison of search points per macroblock for different algorithm. The results show the benefits between the 3DPHS and other algorithm.

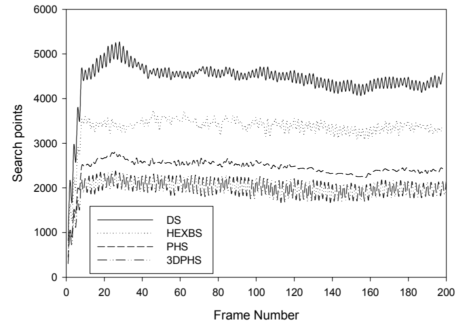


Fig. 8 Average search points per MB for each frame.

5. Conclusions

In this paper, a fast motion estimation algorithm of 3D predict hexagon based is presented. The proposed algorithm uses a rood-shaped search pattern at the fist two steps with a higher probability on search points located in the 5x5 area. It applies some simple equations to calculate the search center. The proposed algorithm predicts the object movement and the search center by using the characteristics of motion vector distribution, variable block sizes and multiple reference frames. Even for the higher resolution sequences, lots of search points are also saved. Simulation results show the speedup percentage for higher definition video sequences is almost the same as level compared with lower definition video sequences. The speed improvement of 3DPHS over DS and HEXBS is about 58% and 53% respectively. It can also save up to 20% search points compared with original PHS algorithm.

References

[1] ITU-T Rec.H.264/ISO/IEC 11496-10, “Advanced Video Coding “, Final Committee Draft, Document JVT-E022, September 2002.

[2] Shan Zhu and Kai-Kuang Ma, “A new diamond search algorithm for fast block-matching motion estimation”**,** *Image Processing, IEEE Transactions on*, Volume: 9, Issue: 2 , Feb. 2000 Pages: 287 - 290

[3] Ce Zhu, Xiao Lin and Lap-Pui Chau; **“**Hexagon-based search pattern for fast block motion estimation”, *Circuits and Systems for Video Technology*, IEEE Transactions Volume: 12, Issue: 5, May 2002

[4] Tsung-Han Tsai and Yu-Nan Pan, “A Novel Predict Hexagon Search for H.264”, *Asia-Pacific conference on circuits and system*. IEEE Transactions on Dec 2004

[5] Chun-Ho Cheung and Lai-Man Po; **“**A novel cross-diamond search algorithm for fast block motion estimation”, *Circuits and Systems for Video Technology*, IEEE Transactions on, Volume: 12, Issue: 12, Dec. 2002 Pages: 1168 – 1177

[6] Hongjun Jia and Li Zhang; **”**Directional diamond search pattern for fast block motion estimation” *Electronics Letters*, Volume: 39, Issue: 22, Oct. 2003 Pages: 1581 – 1583

[7] Tsung-Han Tsai and Yu-Nan Pan, “A Novel Predict Hexagon Search for H.264”, *Asia-Pacific conference on circuits and system*. Dec 2004