論文
4-12-20

An Efficient Center-Biased Hybrid Search Algorithm
(효율적인 Center-Biased Hybrid 탐색 알고리즘)

홍수봉(Su-Bong Hong)[1], 정수목(Soo-Mok Jung)[2]

ABSTRACT

In this paper, we propose an Efficient Center-Biased Hybrid Search (ECBHS) for motion estimation based on Center-Biased Hybrid Search(CBHS). This proposed algorithm employ hybrid of a compact plus shaped search, X shaped search, and diamond search to reduce the search point for motion vectors which distributed within 3pels radius of center of search window. ECBHS reduces the computations for motion estimation of CBHS with similar accuracy. The efficiency of the proposed algorithm was verified by experimental results.

요 약

본 논문에서는 움직임 추정을 위한 Efficient Center- Biased Hybrid Search (ECBHS) 알고리즘을 제안하였다. 제안된 알고리즘은 Center-Biased Hybrid Search (CBHS) 알고리즘에 기초하고 있다. 제안된 알고리즘은 +, X, 다이아몬드 탐색형태를 결합하여 탐색 윈도우의 중앙으로부터 3pel 이내에 분포하는 움직임 벡터를 찾는 과정에서 탐색점을 효과적으로 감소시킨다. ECBHS는 UCBDS 및 CBHS와 거의 동일한 움직임 추정 정확도를 유지하면서 움직임 추정을 위한 연산량을 효과적으로 감소시킨다. 제안된 알고리즘의 효율성을 실험을 통하여 확인하였다.

Keyword : Block Matching Algorithm, Center-biased Search, Motion Estimation, H.263

1) 정회원 : LG전자 연구소

2) 정회원 : 삼육대학교 컴퓨터학과 부교수

논문접수 : 2003. 12. 15.
심사완료 : 2003. 12. 24.

I. Introduction

The temporal redundancy in consecutive video frames makes it possible to achieve high coding efficiency in video coding by reducing the temporal redundancy with motion estimation scheme. Motion estimation schemes are classified into two classes of block matching algorithms(BMA) and pel-recursive algorithms(PRA). Block matching algorithm is more simple than PRA for implementation. So, block matching algorithms have been widely adopted by various video coding standards such as MPEG, CCITT H.261, and ITU-T H.263[1].

In BMA, the current image frame is partitioned into fixed-size rectangular blocks. Block matching algorithm is to estimate the amount of motion on a block by block basis to find optimum motion vector. To determine the best matching block, a block distortion measurement (BDM) is used in motion estimation such as SAD. Although full search algorithm(FS) has the highest accuracy, it has high computational complexity. To reduce the complexity of FS, many block matching algorithms were proposed such as three-step search(TSS) algorithm[2], four-step search(4SS) algorithm[3], new fast three-step search(NFTSS) algorithm[4], Unrestricted Center- biased Diamond Search(UCBDS) algorithm[5].

To reduce the complexity of UCBDS, our research team proposed Center-Biased Hybrid Search(CBHS)[6] and [7]. In this paper, we improved and refined the motion vector searching scheme used in [6]. By using the ECBHS in motion estimation, the computational complexity was reduced with the motion estimation accuracy of CBHS.

II. Center-Biased Hybrid Search Algorithm

Our research team proposed Center-Biased Hybrid Search(CBHS)[6] algorithm using hybrid of a plus shaped search and diamond search to reduce the computational complexity for searching motion vector which is distributed at the center of the search window. Compared with UCBDS[5], dramatically it can reduce computation complexity. In real world, because most of the motion vector is distributed at the center of search window, at first CBHS perform plus shaped search with expectation that initial point may be motion vector, and then perform diamond search to find motion vector located at the other area within search window.

Though CBHS algorithm can reduce the number of search points dramatically for searching motion vector at the center of the search window, the motion vectors distributed nearby the center except (0,0) is not considered to reduce the number of search point efficiently.

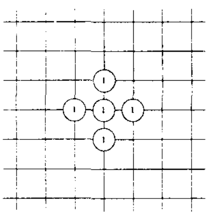
Figure 1 illustrated the motion vector distribution within the central (±3,±3) at the search window for typical test video sequences, "stefan.qcif" and "foreman.qcif".

	-3	-2	-1	0	+1	+2	+3
+3	.00/ .00	.01/ .01	.03/ .01	.03/ .05	.00/ .01	.00/ .00	.01/ .01
+2	.01/ .01	.01/ .02	.07/ .06	1.05/ .36	.03/ .05	.01/ .01	.00/ .01
+1	.01/ .02	.04/ .06	.71/ .37	1.20/ 2.59	.06/ .49	.03/ .08	.01/ .02
0	.05/ .11	.28/ .19	2.17 1.38	89.04/ 80.39	2.36/ 3.02	.07/ .16	.02/ .03
-1	.04/ .05	.11/ .12	.72/ .84	.61/ 2.78	.22/ 1.76	.03/ .15	.01/ .02
-2	.01/ .00	.01/ .02	.06/ .75	.21/ .90	.06/ .71	.02/ .34	.01/ .01
-3	.00/ .01	.01/ .01	.01/ .06	.15/ .09	.03/ .07	.02/ .01	.00/ .01

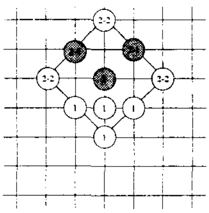
Fig. 1. Center-biased motion vector distribution characteristic of well known test video sequence(stefan/foreman)

Among the distributed motion vectors except the center of the search window (i.e. (0,0)), 80.39% motion vector of "stefan.qcif" and 89.04% motion vectors of "foreman.qcif" are distributed within shaded area.

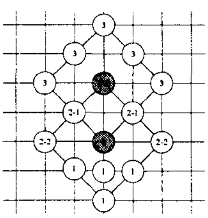
Figure 2. depicts search points configuration used in CBHS which employs two search patterns. The first pattern, called plus-shaped search pattern, comprises five checking points to find the motion vector at the center of search window efficiently (Figure 2.(a)), and the second pattern, called diamond search pattern, searches motion vector distributed in the remainder area of the search window. (Figure 2.(b)). In other to perform a diamond search in second step, five new search points BDM is required. Figure 2.(c)(d) show the positions of the diamond, with respect to the previous position, for the next search step along the diamond's vertex and face, respectively. Figure 2.(e) illustrates the final search where the diamond is shrunk to only four new candidates for internal-point checking



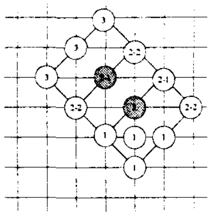
(a) Plus-shaped search-point configuration



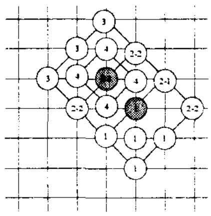
(b) Next step along a diamond's face



(c) Next step along a diamond's vertex



(d) Next step along a diamond's face



(e) Final step

Fig. 2. CBHS search pattern.

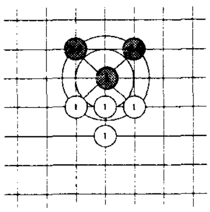
III.I Efficient Center-Biased Hybrid Search Algorithm

The purpose of ECBHS *algorithm* is speedup through reducing the search points to find motion vector in which is located shading area at Figure 1. As the aforementioned, among the motion vectors distributed nearby the center except (0,0), the portion of motion vector distributed over the shaded area at Figure 1. is significant. So if the number of search points to find such motion vectors can be reduced, then the performance of CBHS algorithm can be improved.

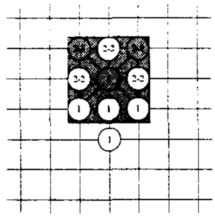
Figure 2, 3 depict a search point configuration used in ECBHS. ECBHS employs three search patterns. The first pattern, called plus-shaped search pattern, comprises five checking points to find the motion vector at the center of search window efficiently (Figure 1), and the second pattern, called X-shaped search pattern, determines whether the motion vector is in shaded area of Figure 1 or not(Figure 3.(a)), and the third pattern, called diamond search pattern, searches motion vector distributed in the remainder area of the search window.(Figure 2.(b)). Unlike CBHS, ECBHS employs X-shaped search to determine whether the motion vector is in shaded area of Figure 1. or not with two additional search points (Figure 3.(a)) between plus search process and diamond search process. If the minimum BDM occurs at the center of X, the search will go to the final X-search step with 3x3 search window.

Figure 3.(b) show the position of search point configuration of X-shaped final step. Three new candidate points is added(Figure 3.(b), marked 2-2) and the minimum BDM point is choosen as the estimated motion vector. If one of the two added point for X-shaped search has minimum BDM, performs next step with diamond search like as CBHS dose. In other to perform a diamond search, five new search points BDM is required, but two search points BDM was calculated in X-shaped searching. That is to say, compared to CBHS, ECBHS has no additional search points for X-shaped search.

In real world scene, generally most motion vector distribution is center-biased ,and the error surface of BDM is unimodal, X-shaped search pattern can predict the location of motion vector efficiently.



(a) Next step along a plus's vertex. (X-shaped search)



(b) X's Final step

Fig. 3. X-shape search pattern in ECBHS.

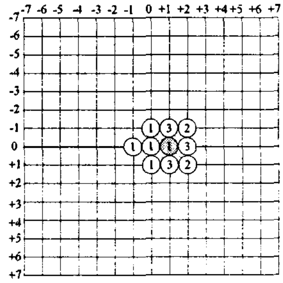


Fig. 4. Example of search path strategy using ECBHS to locate a motion vector at (1,0).

Figure 4. illustrates an example of the search path of ECBHS. Assume that the true motion vector of the block is (m_x,m_y) = (+1,0). We begin at (0,0) with an original plus pattern marked as 1. For each of the five candidate search points, the BDM is computed and compared. If the minimum BDM was found at (0,0) then finish the search process and determine the motion vector as (0,0). Suppose that the lowest BDM is found at (+1,0). We then proceed the next step (marked as 2) in which the X-shape is centered at (+1,0). If the lowest BDM in X-shape was found at (+1,0) then X-shaped final search is done. In this example, ECBHS requires only 10 search point to find motion vector. But CBHS algorithm requires 13 search points to find motion vector.

The above search path strategy using the ECBHS algorithm can be summarized as follows.

step 1 (Starting): The Plus pattern [Figure 2. (a)] is placed at (0,0), the center of the search window. The BDM is evaluated for each of the five candidate search points. If the minimum BDM point is found to be at the center (0,0) of the diamond, then the (0,0) point is chosen as the estimated motion vector; otherwise, proceed to Second Searching. If the minimum BDM point in step1 is located at one of the four outer points [i.e., (-1.0), (+1,0), (0,-1) or (0,+1)], then proceed to step 2.

step 2 (X-shaped Searching): X-shaped Searching pattern of Figure 3.(a) is used with the center of the new X coinciding with lowest BDM point[i.e. updating the center(c,c)]. Two new candidate search points are evaluated. If the lowest BDM is the center of the X, then proceed to step3 otherwise, proceed to step4.

step 3 (X's Final Searching): Three new candidate search points are evaluated. In this case, If the minimum BDM point is found to be at the center (c,c) of the X, then this point is chosen as the estimated motion vector (m_x,m_y); otherwise, the estimated motion vector(m_x,m_y) is chose the minimum BDM point which is found to be at one of the three points (marked as 2-2)

step 4 (Diamond Searching): Diamond searching is followed UCBDS algorithm[5]. If the minimum BDM point in the previous search step is located at one of the four vertices [i.e., either (c-2,c), (c+2,c),(c,c-2), or (c,c+2)], then proceed to step4.1. Else, if it is located at one of the four possible faces of the previous diamond [i.e., either (c-1,c+1), (c-1,c-1), (c+1,c-1), or (c+1,c+1)], then proceed to step 4.2.

step 4.0 (Vertex Search): The diamond pattern of Figure 2.(e) is used with center of the new diamond coinciding with the lowest BDM point. Five new candidate search points are evaluated.

step4.1 (Face Search): The diamond pattern of Figure 2.(d) is used with the center of the new diamond coinciding with the lowest BDM point. Three new candidate search points are evaluated. Note that any candidate point that extends beyond the search window is ignored. The minimum BDM is again identified. If the minimum BDM is found at (c,c), then proceed to step 5; otherwise, proceed to step 4 to continue the next search step.

step 5 (Ending): The shrunk diamond pattern of Figure 2.(e) is used with the same center (c,c). Now, the final four internal points of the previous diamond are evaluated. Similarly, any internal points of the previous diamond are evaluated. Similarly, any internal candidate point that extends beyond the search window is also ignored. The candidate point that gives the lowest BDM is choose as the estimated motion vector (m_x,m_y). The current blocks search process is completed. Proceed to step1 for the next block.

III.II Teoretical Analysis of ECBHS

This subsection aims to investigate theoretically how speed improvement can be obtained over CBHS algorithm. Our main argument in this analysis is based heavily on the observed center-biased motion vector distribution. We first analyze the minimum number of search points Ns within a region of 3 pixels about the motion vector (0,0). It is shown in Figure 5 and Figure 6 . It can be easily observed that, within this region, especially shaded points, ECBHS gives lower value of N_s as compared CBHS. To get a better picture of the gain in N_s , we subtract the corresponding candidate points of ECBHS from CBHS over this region, By doing so, we can obtain a saving of as block matches per block.

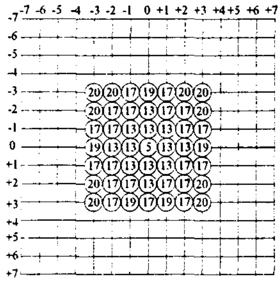


Fig. 5. Minimum possible number of search points for each motion vector locating using CBHS

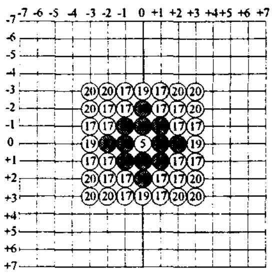


Fig. 6. Minimum possible number of search points for each motion vector using ECBHS

To further quantify this gain in N_s for block estimation, we define the following probabilities of occurrence:

P_0: probability of stationary block [i.e., the motion vector is (0,0)]

P_1: probability of quasi-stationary blocks within ±1, but excluding (0,0)

P_2: probability of quasi-stationary blocks within ±2, but excluding the ±1 region at the center

P_3: probability of quasi-stationary blocks within ±3, but excluding the ±2 region at the center

P': probability of blocks in the region where $4 \le |mx|, |my| \le W$

By taking the average of the difference in N_s between CBHS and ECBHS over each of the above regions, the statistical average gain of ECBHS over CBHS can be represented as Gain in

$$N_g = 0P_0 + \left(\frac{24}{8}\right)P_1 + \left(\frac{12}{16}\right)P_2 + 0P_3 + nP' \qquad (1)$$

where P' = 1 (P0+ P1+ P2+ P3), and n is some positive number. Suppose further that we assume a uniform probability distribution over the ±3 region at the center, and that no motion vector lie outside of this region. Then from (1), we will have a uniformly distributed average gain of uniform gain in N_s = 0.25x(0+4+0.75+0)=0.9 search points per block. Theoretically average gain of 0.9 search points per block can be expected.

IV. Experimental Results

In all of our experiments, the SAD block distortion measure, block size N = 16, and search window size W = 7 were used. For testing, a total of two sequences that are widely used H.263 test sequence. We use 100 frames of the "stefan.qcif", and "foreman.qcif". In our experiment, we choose the quantization parameter to 10.

We compared the ECBHS against UCBDS and CBHS using the following three test criteria

1) **Average number of search points N_s per block** : This provides an equivalent measure of the actual CPU run time
2) **Speedup** : This indicate how many times faster than UCBDS
3) **Probability of finding true motion vector per block** : This gives the likelihood of the suboptimal predicted block motion vectors to be the same as those found using the optimum FS; this also provides an indication of the susceptibility of each suboptimal search method being trapped in local optima.

Tables 1, 2 show the experimental performance of each search technique over the two test criteria using three representative sequences. The first column of the table represents the average number of search point.

It is shown that ECBHS is 6.8%~7.9% faster than CBHS algorithm. The order of the average search points per block is ECBHS<CBHS<UCBDS for all tested sequence.

The second column is a speedup factor with respect to UCBDS. We can see that ECBHS generally more efficient (i.e, it has a faster search) then the other schemes in the low bit-rate sequence, such as conference of the video using H.263.

The third column of the Table 1 shows the percentage of the probability of finding true motion vector per block. We can see that the accuracy of ECBHS is similar to CBHS. Therefore our X-shaped prediction has some good accuracy.

Table 1. Performance comparisons using "stefan.qcif"

Algorithm	Avg. search points	Speed up	Avg. probability
UCBDS	13.1		98.21
CBHS	6.2	52.7%	98.43
ECBHS	5.3	59.5%	98.42

Table 2. Performance comparisons using "foreman.qcif"

Algorithm	Avg. search points	Speed up	Avg. probability
UCBDS	13.9		98.40
CBHS	6.7	51.8%	98.93
ECBHS	5.6	59.7%	98.87

IV. Conclusion

In this paper, several suboptimal algorithms have been described. Then, we proposed a Efficient Center-Biased Hybrid Search(ECBHS) algorithm for fast suboptimal block- based motion estimation employing a hybrid of plus-shaped search and X-shaped search and diamond search. It is motivated by the center-biased motion vector distribution characteristic (especially distributed within 2pels radius of center of search window) of real-world video sequences. We then explain the algorithm development of ECBHS, and perform a theoretical analysis of its efficiency. Experimental results show that ECBHS exploit the center-biased characteristic more efficiently then other suboptimal block matching algorithm such as UCBDS, and CBHS. The ECBHS is 7~8% faster than CBHS algorithm with similar accuracy.

If a appropriate prediction algorithm is applied to our algorithm for choosing initial motion vector, then it can show better improved performance.

References

[1] ITU-T Video coding for low bitrate communication. ITU-T Recommendation H.263 version 2. January 1998
[2] T.Koga, K. Ilinuma, A. Hirano, Y.Iijima, and T.Ishiguro, "Motion compensated inter frame coding for video conferencing," in Proc. NTC 81, New Orleans, LA, Nov./Dec. 1981, pp. C9.6.1-C9.6.5.
[3] L.M. Po and W.C.Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., Vol. 6, pp. 313- 317, June 1996.
[4] Jong-Nam Kim and Tae-Sun Choi, "A fast motion estimation for software based real-time video coding," IEEE Transaction on Consumer Electronics, Vol. 45, No. 2, pp. 417-425, May 1999.
[5] Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath, and Ashraf Ali Kassim, "A novel unrestricted center- biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., Vol. 8, No. 4, pp. 369-377, August 1998.
[6] Sung-Chul Shin; Hyunki Baik; Myong-Soon Park; Dong Sam Ha, "A center-biased hybrid search method using plus search pattern for block motion estimation", IEEE ISCAS 2000, vol. 4, pp 309-312, 2000.

정수목

1984 경북대학교 전자공학과(학사)
1986 경북대 대학원 전자공학과(석사)
1986~1991 LG 정보 통신 연구소 연구원
2002 고려대 대학원 컴퓨터학과(박사)
1998~현재 삼육대학교 컴퓨터학과 부교수
관심분야: 멀티미디어, 컴퓨터시스템

홍수봉

1997.2. 고려대학교 수학과 졸업
2002.2. 고려대학교 대학원 컴퓨터학과 졸업
현재 LG전자 연구소
관심분야: 멀티미디어, 영상처리