ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

A CENTER-BISED HYBRID SEARCH METHOD USING PLUS SEARCH PATTERN FOR BLOCK MOTION ESTIMATION

Sung-Chul Shin[1], Hyunki Baik[1], Myong-Soon Park[1] and Dong Sam Ha[2]

[1]Computer Systems Laboratory
Department of Computer Science and Engineering Korea University
Sungbuk-ku, Seoul 136-701, Korea
[2]Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061, USA

ABSTRACT

This paper proposes a new method for block motion estimation based on Center-Biased Hybrid Search(CBHS) using plus search that employ a hybrid of a compact plus shaped search and a diamond search. It has only 5 search points for the best case and, on average, 6.6 search points. CBHS is about 34.3 times faster than the full-search algorithm. Hence, CBHS is consistently faster than the other suboptimal block-matching techniques. This paper compares the popular suboptimal block-matching technique with CBHS in which both the processing speed and the accuracy of motion compensation are tested over widely used H.263 test video sequences.

1. INTRODUCTION

Motion estimation is the most time consuming process for a video encoder. so a fast motion estimation algorithm is crucial for real-time applications. Block-based motion estimations are widely used for video compression such as MPEG and H.263 standard[4] owing to its effectiveness and simple implementation. However, the high computational complexity of the full-search algorithm has motivated a host of suboptimal but faster search strategies[1][2][3]. A good example is the three-step search(TSS) algorithm[5]. However, the uniformly spaced search pattern in TSS does not match well to real-world video sequences in which the motion vector distribution is mostly nonuniformly biased toward the zero vector. Such an observation inspired the four-step search(4SS) algorithm[6] which has a center-biased search pattern. 4SS algorithm exploit the center-biased motion vector distribution characteristic by utilizing a nine-point search pattern on a 5×5 grid. As a result of starting with a smaller search grid pattern, 4SS is, on the average, faster than TSS and yields a better motion estimation. Recently, Tham et al. proposed a novel method called unrestricted center-biased diamond search(UCBDS) algorithm[8]. The average number of search points is reduced to 13.7 for UCBDS when compared with 17.4 points for 4SS, while the performance of UCBDS is better than 4SS in motion compensation error.

In this paper, we propose a center-biased hybrid search (CBHS) algorithm which is a hybrid of a compact plus shaped search and a diamond search. CBHS is much faster than the previous techniques such as 4SS and UCBDS. The performance of the proposed CBHS algorithm is similar to that of UCBDS in motion estimation error. Our method is based on the observation that about 90%, of motion vectors are concentrated in the region of (0,0), (0,-1), (0,+1), (+1,0), and (-1,0) as shown Fige. 1. We get the initial candidate[(0,0)] from the median value of the left block's motion vector, upper block's motion vector and right-upper block's motion vector of the current block. The CBHS aims to reduce the number of search points in the region to speed up.

	-3	-2	-1	0	+1	+2	+3
-3	0.01	0.01	0.04	0.03	0	0.01	0
-2	0.01	0	0.05	0.07	0.02	0	0.01
-1	0	0.06	0.63	0.39	0.15	0.04	0.02
0	0.06	0.18	5.04	88.71	2.00	0.05	0.01
+1	0.03	0.09	0.74	0.58	0.21	0.01	0
+2	0.01	0	0.04	0.05	0.04	0.01	0
+3	0.01	0	0	0.02	0	0.01	0

Figure 1. Center-biased motion vector distribution characteristics of Miss America sequence

2. PROPOSED CENTER-BIASED HYBRID SEARCH ALGORITHM

2.1 Algorithm of CBHS

A frame is divided into blocks of size N $\times$ N pixels, and a block-based motion estimation algorithm is applied to individual blocks. For example, the block size of 16×16 is used for MPEG-1, MPEG-2, and H.263[4]. The search for a match in a block is usually performed over a 15×15 search area for low and very low bit-rate video applications. Hence, it requires to 225 search per block for the full search. This is computationally too expensive. Hence, the objective of our algorithm like any suboptimal center-biased search algorithm is to choose an appropriate subset of the 225 points. As a metric for the estimation, a block distortion measure (BDM) is used as the objective function for each block's evaluation. For simplicity, we employ the sum of absolute difference(SAD) in our algorithm CBHS.

0-7803-5482-6/99/$10.00 ©2000 IEEE

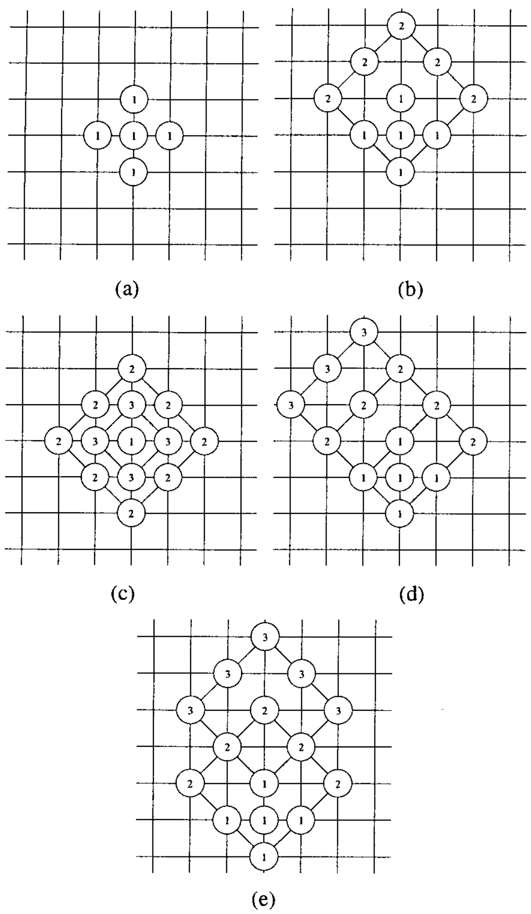


Figure 2. CBHS search pattern. (a) Plus-shaped search-point configuration. (b) Next step along a plus's vertex. (c) Final step. (d) Next step along a diamond's face, (e) Next step along a diamond's vertex

Fig. 2. depicts a search point configuration used in CBHS. This pattern is inspired by its compact structure which is very suitable for the center-biased characteristic (most of motion vector is concentrated on (0,0)) of motion vector distribution.

Fig. 3. illustrate an example of the unrestricted search path strategy using CBHS. Assume that the true motion vector of the block is $(m_x, m_y) = (+3, -1)$. 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 diamond is centered at (+1,0). In this example, we require five search steps. Where the shaded candidate points are the best points in each step. Notice that the minimum BDM point in step 4 coincides with that of step 3. This signal to shrink the diamond pattern for internal-point checking. Altogether, we have performed 22 block evaluations for this example.

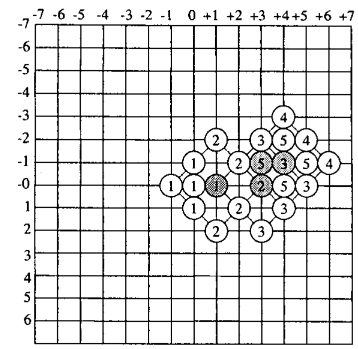


Figure 3. Example of search path strategy using CBHS to locate a motion vector at (+3,-1)

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

- **Starting** : The Plus pattern [Fig. 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**.
- **Second Searching** : If the minimum BDM point in the Starting step is located at one of the four outer points [i.e., (-1.0), (+1,0), (0,-1) or (0,+1)], then proceed to Second Searching. Second Searching pattern of Fig. 2.(b) is used with the center of the new diamond coinciding with lowest BDM point [i.e., updating the center(c,c)]. Five new candidate search points are evaluated, then proceed to **Diamond Searching**.
- **Diamond Searching** : Diamond searching is followed UCBDS algorithm[8]. 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 **Vertex Search**. 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 **Face Search**.

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

\- **Face Search** : The diamond pattern of Fig. 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 **Ending**; otherwise, proceed to **Diamond Searching** to continue the next search step.

- **Ending** : The shrunk diamond pattern of Fig. 2.(c) 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 chosen as the estimated motion vector (m_x, m_y). The current block's search process is completed. Proceed to **Starting** for the next block, if any.

2.2 Theoretical Analysis of CBHS

We have explained earlier the motivations for having center-biased search algorithm such as the 4SS, UCBDS, and CBHS. This subsection aims to investigate theoretically why CBHS is truly center biased, and how speed improvement can be obtained over search algorithm. In particular, we are comparing CBHS with UCBDS. Our main argument in this analysis is based heavily on the observed center-biased motion vector distribution.

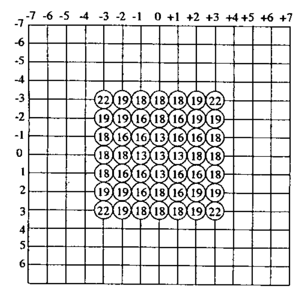


Figure 4. Minimum possible number of search points for each motion vector locating using UCBDS

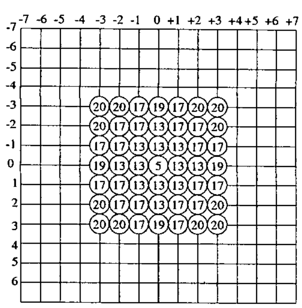


Figure 5. Minimum possible number of search points for each motion vector using CBHS

To begin, we first analyze the minimum number of search points N_s within a region of ±3 pixels about the motion vector (0,0). This is depicted in Figure 4 and Figure 5. It can be easily be observed that, within this region, CBHS gives lower value of N_s, we subtract the corresponding candidate points of CBHS from UCBDS over this region, By doing so, we can obtain a saving of as high as block matches per block.

To further quantify this gain in N_s for block estimation we define the following probabilities.

- P_0: probability of stationary block [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 $4 \le |m_x|, |m_y| \le W$

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

$$\text{Gain in } N_s = 8P_0 + (\frac{12}{8})P_1 + (\frac{20}{16})P_2 + (\frac{4}{24})P_3 + nP' \quad (1)$$

where $P' = 1 - (P_0 + P_1 + P_2 + P_3)$, 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

uniforrn gain in $N_s = 0.25 \times (8 + 1.5 + 1.25 + 0.17) = 2.73$ search points per block.

However, observations from most real-world sequences show very peaked probabilities around P_0 and P_1, as depicted Fig. 1. This means that an average gain of more than 2.73 search points per block can be expected. More experiment results later will justify this statement.

3. EXPERIMENTAL RESULTS AND COMPARISONS

This section aims to investigate the actual experimental performance of CBHS. 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 eight sequences that are widely used H.263 test sequence.

Search Algorithm	Avg. Search points	Speedup	Avg. probability
Using " Miss America" Sequence			
FS	225.0	1.0	100.00
TSS	25.0	9.0	96.87
4SS	17.0	13.2	97.07
NFTSS	13.0	17.3	87.95
UCBDS	13.1	17.2	98.10
CBHS	5.9	38.1	98.41
Using "Trevor" Sequence			
FS	225.0	1.0	100.00
TSS	25.0	9.0	90.98
4SS	17.5	12.9	96.86
NFTSS	13.0	17.3	81.69
UCBDS	13.8	16.3	98.42
CBHS	6.4	35.2	97.51
Using "Suzie" Sequence			
FS	225.0	1.0	100.00
TSS	25.0	9.0	89.60
4SS	17.7	12.7	94.01
NFTSS	13.2	17.0	68.68
UCBDS	14.1	16.0	96.70
CBHS	7.6	29.6	94.36

Table 1. Performance Comparisons

However due to space limitations, we will only present the result of three representative sequences. First, we use 100 frames of the "Miss America" sequence, which is a typical videoconferencing scene with limited object motion and a stationary background. Second, we choose 100 frames of the "Trevor". Third, we selected

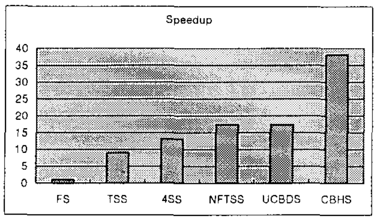
100 frames of the "Suzie" sequence. In our experiment, we choose the quantization parameter to 10.

We compared the CBHS against five other block-based motion estimation methods – FS, TSS, NFTSS[7], 4SS, and UCBDS – using the following three test criteria

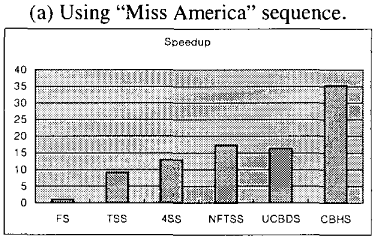
1) Average number of search points N_s per block : This provides an equivalent measure of the actual CPU run time.

2) Speedup : This indicate how many times faster than FS.

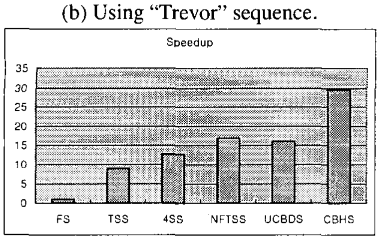
3) Probability of finding true motion vector per block : This gives the likelihood of the suboptimal predicted block motion vectors to be the same as those found using the optimum FS; this also provides an indication of the susceptibility of each suboptimal search method being trapped in local optima.



(a) Using "Miss America" sequence.



(b) Using "Trevor" sequence.



(c) Using "Suzie" sequence

Figure 6. Performance comparisons of FS, TSS, FSS, NFTSS, UCBDS and CBHS over Speedup criteria Using Different Video Sequence.

Table 1. summarizes the experimental performance of each search technique over the four test criteria using three representative sequences. The first column tabulates the average number of search point, The second column tabulate speedup factor with respect to FS is reported. This speed factor shows that how many times faster than FS. The average N_s per block CBHS < NFTSS <UCBDS <4SS < TSS <FS and the speedup; such observations were true for all of the test sequences we used. This shows that CBHS is generally more efficient (i.e., it has a faster search) than the other schemes in the low bit-rate sequences, such as video conference used for H.263.

4. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we proposed a Center-Biased Hybrid Search (CBHS) Using Plus Search algorithm for fast suboptimal block-based motion estimation. Motivated by the center-biased motion vector distribution characteristics (especially concentrate on (0,0), (+1,0), (-1,0), (0,+1), (0,-1) motion vector) of real-world video sequences, CBHS was consist of two phase, first phase was designed with very compact plus-shaped (+) search point configuration, and second phase was designed with the compact diamond search point configuration. We then explained the algorithm development of CBHS, and performed a theoretical analysis of its efficiency. Experimental results were presented to show that CBHS is more efficient and robust as compared to some other popular block-matching algorithm such as FS, TSS, 4SS, NFTSS and UCBDS. In short, CBHS has the following advantages over the other algorithm.

- Efficiency: CBHS is highly center biased, and it has a very compact plus-shaped (+) search point and diamond search points configuration. This allows a minimum of only 5 candidate search points per block, and 2.08 times faster than UCBDS and 34.3 times faster than FS.
- Robustness: As CBHS is unrestricted and does not have a predetermined number of search steps, it is flexible enough to work well for any search range/window size.

5. REFERENCES

[1] J.Lu and M.L.Liou, "A simple and efficient search algorithm for block-matching motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol.7, pp.429-433, Apr.1997.

[2] J.Kim and T.S.Choi, "Reduction of checking points using unimodal error surface assumption for fast motion estimation," SPIE vol.3460, San Diego, pp.124-134, Jul.1998.

[3] R.Li, B.Aeng and M.L.Liou,"A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol.4, pp.313-317, Jun.1996.

[4] ITU-T, Video coding for low bitrate communication, ITU-T Recommendation H.263 version 2, January 1998.

[5] T.Koga, K.llinuma, 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

[6] 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.

[7] J. Kim and T.S.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.

[8] J. Y. Tham, S. Ranganath, M. Ranganath, and A. 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.