

# An Adaptive Cross Search Algorithm for Block Matching Motion Estimation

Jiancong Luo<sup>1</sup>, Ishfaq Ahmad<sup>1</sup> and Xizhang Luo<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019

<sup>2</sup>Department of Electronics, Zhong Shan University, Guangzhou 510275, China,

Email: Email: jluo@cse.uta.edu, iahmad@cse.uta.edu, isslxz@zsu.edu.cn

**Abstract**— Motion estimation plays an important role in the motion compensated video coding framework. Due to the high computational complexity of the exhaustive search, many sub-optimal fast search algorithms, aiming to achieve the best trade off between distortion and search speed, are proposed. We observe that distortion gradient of a search point on the block distortion surface (BDS) monotonously decreases with increasing distance from that point to the global minimum point. Based on this property, we propose a novel adaptive cross search (ACS) algorithm that can distribute the computation powers over the search space efficiently. The simulation shows that ACS achieves competitive reconstruction visual quality as well reduced computational complexity.

**Keywords** (3-8 words): motion estimation, block matching, motion compensation, video compression

## I. INTRODUCTION

In a motion compensated video coding system, advantages of temporal correlation between successive frames are taken to achieve high compression ratio. The current frame is predicted by a reference frame. Only the prediction errors and motion vectors are encoded and transmitted. Motion estimation (ME), who aims at finding the motion between consecutive frames, plays an important role in motion compensated video coding framework. Among many ME algorithms, block matching ME (BMME), due to its simplicity, is widely adopted in many video compression standards, e.g. MPEG-4 [1] and H.263 [2]. In BMME, a frame is divided into blocks, ME algorithm searches the best matching block in the

reference frame and returns a motion vector that points to the matching block.

Full search (FS) block matching algorithm exhaustively tests all candidates in the predefined search window, thus always guarantees an optimal solution. However, FS suffers from its extremely high computational complexity. Accordingly, fast and accurate BMMEs are highly desired to reduce the encoding delay while maintaining the visual quality. Many fast block matching algorithms are proposed. The three-step search (TSS) algorithm [3] is a popular fast ME due to its simplicity and regularity. It sub-samples the candidates within the search space uniformly. Based on the zero-biased distribution of motion vectors, the new three-step search (NTSS) algorithm [6] adds 8 neighborhood search points adjacent to the zero motion vector. Other fast BMMEs includes 2-D logarithmic search (LOGS) [4], cross search (CS) [5], block-based gradient descent search (BBGDS) [7], diamond search (DS) [8], etc. These fast ME algorithms combine different search patterns and search strategies. They efficiently reduce the computational complexity and keep the search accuracy close to FS.

In this paper, we proposed an adaptive cross search (ACS) algorithm that can efficiently distribute the computation power in the search space.

## II. OBSERVATION

We define a block distortion surface (BDS) as a 2D scalar field which consists of the block distortion values of all the candidate motion vectors in the search space. The sum of absolute difference (SAD) is used to measure the block distortion, which is given by

$$SAD(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f_{i-v}(i+u, j+v) - f_t(i, j)| \quad (1)$$

where  $(u, v)$  is the candidate motion vector,  $f_t(i, j)$  is the luminance value of pixel  $(i, j)$  at frame  $t$ .

A typical BDS is depicted in figure 1.

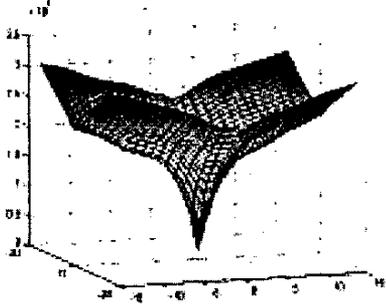


Figure 1. Average BDS of "news" sequence

By evaluating of a large number of BDSs of different test sequences, we have the following observations:

1. The distortion of a search point monotonously increases with increasing distance from that search point to the global minimum point (GMP);
2. When a search point is further from the GMP, the surface at that point is flatter, i.e., the gradient of distortion is smaller; when a search point is closer to the GMP, the surface at that point is steeper, i.e. the gradient of distortion is larger.

The first observation is well known as the *unimodal* property of BDS. It has been widely adopted as a fundamental assumption of many fast BMMEs.

We conclude the second observation as the property of distortion gradient as follows:

*On a BDS, The distortion gradient of a search point monotonously decreases with increasing distance from that search point to the GMP.*

The property is testified in table 1, which illustrates the average gradients versus the distance to the GMP by applying the FS algorithm to five popular video sequences. The column "distance" is the chess-board distance to the GMP.

The above property is the fundamental assumption of the proposed ACS algorithm. With this property, one can

estimate the distance from the current search point to the GMP. The regions with the higher gradient value have higher probabilities of finding the global minimum. Therefore, more computation powers should be allocated to these regions to refine the motion vector. Otherwise, a coarse search is employed to speed up the search process. In addition, based on the zero-bias property of the motion vector distribution [6], the region around the zero motion vector has higher probability of finding the global minimum as well. Based on the above properties, we proposed a ME algorithm that employs different search patterns on different regions in order to distribute the computation over the search space efficiently.

Table 1. Average gradient of distortion versus distance to the GMP

Distance	News	Akiyo	Forman	Stefan	Silent
1	1317.7	741.7	644.3	1294.1	1195.0
2	863.5	466.8	518.5	1108.5	623.1
3	680.4	373.6	451.6	819.0	483.0
4	536.3	315.2	381.1	508.0	396.9
5	437.8	285.9	347.6	353.4	348.3
6	355.6	258.6	317.1	291.9	310.9
7	289.6	234.9	295.9	284.0	284.0
8	246.9	217.3	280.2	205.9	261.5

### III. THE PROPOSED ALGORITHM

The proposed ACS algorithm employs adaptive size cross search pattern. That a cross (+) shape is chosen is because of the fact that most of the motions in real-world video is along horizontal or vertical direction due to the camera panning and tilting. A cross pattern consists of 4 points,  $(S, 0)$ ,  $(0, -S)$ ,  $(-S, 0)$ ,  $(0, S)$ ,  $S \geq 1$ . Factor  $S$  is used to adjust the size of the cross pattern. Figure 2 depicts cross patterns with scaling factor  $S=1$  and  $S=2$ .

Due to the zero-biased motion vector distribution, the algorithm starts the search process with the smallest size cross pattern, i.e.  $S=1$ , and initially sets the center of the cross pattern at  $(0, 0)$ . During the searching process, factor  $S$  is adjusted according to the estimated gradient of distortion on the minimum distortion point (MDP) in previous step. The gradient of distortion is estimated by

$$grad = \frac{MBD_i - MBD_{i-1}}{S_i} \quad (2)$$

where  $i$  is the current step and  $i-1$  is the previous step.  $MBD_i$  is the minimum block distortion of step  $i$ .

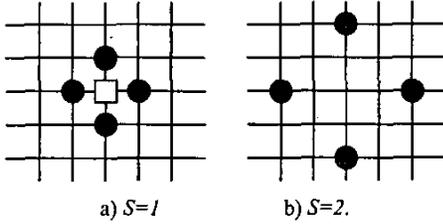


Figure 2. Examples of adaptive size cross pattern.

A small gradient means the current MDP probably is relatively far from the GMP. Thus,  $S$  is adjusted to a larger value in the next step. A large gradient means current MDP point is relatively close to the GMP, a smaller scaling factor should be chosen in the next step. A threshold  $T$  is used in ACS, such that if the gradient of current MDP is larger than  $T$ , increase  $S$ , and vice versa. In our simulation, we assume that if the estimate distance to the GMP is less than 4, we reduce the size of the cross pattern, otherwise, we enlarge the search pattern. Thereby,  $T$  is set to be the mean of gradients with distance of 3 and 4 in table 1, i.e.  $T = 500$ .  $S$  can be adjusted as follows,

$$S_i = \begin{cases} S_{i-1} + 1 & \text{grad} < T, \text{ and BDP is not the} \\ \max(1, S_{i-1} / 2) & \text{center of the cross pattern;} \\ & \text{otherwise.} \end{cases} \quad (3)$$

As expressed in equation (3), if the current MDP occurs at the center of the cross pattern, which means that the global minimum is within the area covered by the current search pattern, a smaller size pattern is chosen and the search window is set to  $S_{i-1}$  to avoid further evaluating the candidates which is estimated to be far from the global optimum. The searching process will be terminated when the current minimum occurs at the center of a cross pattern and  $S$  is 1.

We summarize the ACS algorithm as follows:

- Step 1) Initially  $S=1$ . Point (0,0) and the cross pattern is tested. If the current MDP is located at the center of the cross, terminate the search. Or else, go to step 2
- Step 2) Compute the gradient of the current MDP by (2). Adjust  $S$  by (3)

- Step 3) Move the search center to the MDP. Check the cross pattern with size  $S$ . If the current MDP is located at the center, go to step 4, or else go to step 2

- Step 4) If  $S=1$ , stop. Otherwise, set the search window size to  $S$ , then adjust  $S$  by (3) and go to step 2

Compared with other fast BMME algorithm, the proposed algorithm has the following distinct features:

First, unlike the algorithms which restrict the number of search steps so that only candidates within a fixed size search window are tested, ACS does not restrict the number of search steps. Thus, ACS is able to employ different sizes of search window. To this point of view, ACS is able to adapt to the maximum motion vector length of different video encoders.

Second, most real-world sequences have a zero-biased motion vector distribution. For example, for the “salesman” sequence, nearly 80% blocks can be regarded as stationary and quasi-stationary [6]. ACS starts the search process with the smallest size cross pattern, thus for most low motion block, the search process can be terminated in early steps.

Third, unlike the conventional fixed pattern ME whose search points are evenly distributed, by exploiting the property of the BDS, the proposed algorithm will reduce the step size when the search points are getting close to the global minimum. The density of search points is larger in the area close to the global minimum and the zero point, than in the other areas. Therefore, the computations are distributed efficiently.

Finally, in the implementation of ACS, the positions of tested search points are recorded to avoid the overlap computations.

#### IV. EXPERIMENT RESULTS

In our simulation, four test sequences are used. Among all the test sequences, “Akiyo” and “Mother\_daughter” are slow motion talking head sequences, “news” contains a talking head foreground and a fast motion dancing background, “foreman” contains fast motion and camera panning and zooming scenes. The set of test sequences covers the most common types of real-world videos. The test sequences and their sizes and lengths are listed in table

2. To be consistent with other BMMEs, the search window size is  $W=7$ , although ACS is not restricted to a fixed window size. The block size is  $16 \times 16$ .

Table 2. Sequences used for simulation experiment

Sequence	Frame size	Length	Sequence	Frame size	Length
Akiyo	352 x288	300	Foreman	352 x288	300
News	352 x288	300	Mother_daughter	352 x288	300

The comparing BMME algorithms are TSS, NTSS, 4SS, BBGDS, DS and the proposed ACS algorithm. Computational complexity and visual distortion are compared. The computational complexity is measured by the average number of search points (NSP) per block required by the algorithm. The visual distortion is measured by the peak signal to noise ratio (PSNR).

Table 3-4 compare ACS and the other fast BMMEs in terms of PSNR and average NSP per block. Figure 3-4 show the per frame comparison of the average PSNR and the average NSP per block of "Akiyo" and "foreman".

The experiment results show that in terms of reconstruction image quality, the proposed ACS algorithm is marginally worse compared to the other search algorithms. However, ACS reduces the computational complexity substantially. Another comparison of trade-off between average distortion and computational complexity is performed, as shown in table 5. ACS overwhelms other search techniques in the computation cost comparison. Compared with other fast BMMEs, ACS computational complexity is 42% ~ 59% less, while keeping the visual quality drop less than 0.13dB.

## V. Conclusion

In this paper, we propose an *adaptive cross search* (ACS) algorithm for motion estimation. By exploiting the distortion gradient property of the block distortion surface, the ACS adaptively adjusts the search pattern. The ACS algorithm concentrates the search points in the more important areas and pays less attention to less important area. By effectively distributing the computation powers over the search space, the ACS algorithm successfully

reduces the computational complexity while maintaining to a marginal distortion level compared to the other popular sub-optimal fast motion estimation algorithms. ACS is a competitive algorithm among the existing fast BMMEs.

## REFERENCES

- [1] ISO/IEC CD 14496/2 (MPEG-4 Video), "Information technology---Coding of natural/visual objects, Part 2: Visual" 1999
- [2] ITU-T, "Video Codec for Low Bit-Rate Communication," *Recommendation H.263*, Dec. 1995
- [3] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. NTC81*, New Orleans, LA, Nov./Dec. 1981, pp. C9.6.1-C9.6.5
- [4] J.R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp.1799-1808, Dec. 1981.
- [5] M. Ghanbari, "The Cross-Search Algorithm for Motion Estimation," *IEEE Trans. Commun.*, vol. 38, NO. 7, pp. 950-953, July 1990
- [6] R. Li, B. Zeng, and M. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 4, NO. 4, pp. 438-442, Aug. 1994
- [7] L.K. Liu and E. Feig, "A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 6, NO. 4, pp. 419-422, Aug. 1996
- [8] S. Zhu and K.K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Trans. Image Processing*, vol. 9, NO. 2, pp. 287-290, Feb. 2000

Table 3. PSNR (dB) comparison

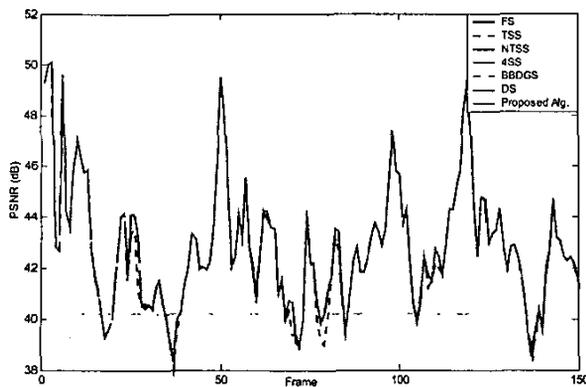
	Akiyo	Foreman	News	Mother_daughter
FS	42.94	31.22	36.72	40.40
TSS	42.82	30.73	36.51	40.24
NTSS	42.93	30.99	36.54	40.29
4SS	42.86	30.77	36.48	40.28
BBGDS	42.94	30.84	36.48	40.30
DS	42.93	30.85	36.51	40.29
ACS	42.91	30.70	36.42	40.21

Table 4. Average computational complexity comparison  
(NSP per block)

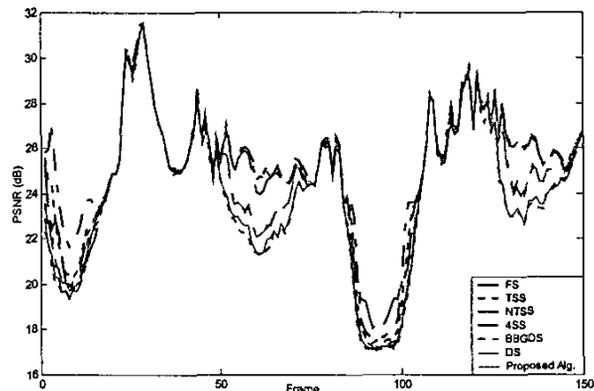
	Akiyo	Foreman	News	Mother_daughter
FS	203.28	203.28	203.28	203.28
TSS	23.21	23.33	23.21	23.29
NTSS	15.92	21.51	16.56	18.00
4SS	15.85	18.91	16.18	16.85
BBGDS	8.53	15.77	9.29	10.21
DS	12.27	17.30	12.81	13.58
ACS	4.98	12.53	5.94	7.49

Table 5. Average performance of ACS versus other search algorithm

	PSNR decrease (dB)	Complexity decrease
FS	0.26	98%
TSS	0.02	79%
NTSS	0.13	69%
4SS	0.04	69%
BBGDS	0.08	42%
DS	0.08	59%

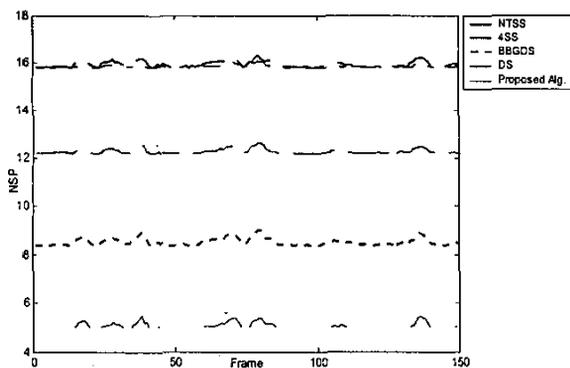


a) "Akiyo"

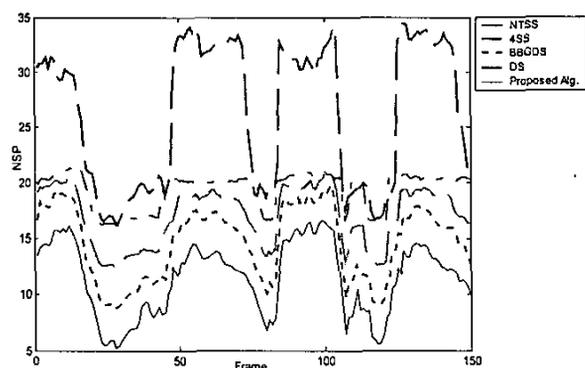


b) "Foreman"

Figure 3. Per-frame PSNR comparison



a) "Akiyo"



b) "Foreman"

Figure 4. Per-frame Average NSP per block comparison