COMPLEXITY COMPARISON OF FAST BLOCK-MATCHING MOTION ESTIMATION ALGORITHMS

Yilong Liu and Soontorn Oraintara

Department of Electrical Engineering, University of Texas at Arlington,
416 Yates St., Arlington, TX, 76019-0016 USA
Email:yilong@msp.uta.edu, oraintar@uta.edu

ABSTRACT

Block-matching algorithm (BMA) for motion estimation (ME) has been widely adopted by real-time video coding applications due to its effectiveness and simplicity in implementation. Most fast BMAs are based on the assumption that ME matching error decreases monotonically as the search approaches the position of the global minimum error. This paper measures the contributions on computational costs reduced by different fast BMAs including four-step search, diamond search, hexagon-based search and recently proposed adaptive multi-mode search (AMMS). Comparison results show that the AMMS algorithm achieves a significant improvement based on the given mathematical models. Experimental coding results are also presented.

1. INTRODUCTION

Block-matching algorithm (BMA) for motion estimation (ME) has been widely adopted by the current video compression standards, such as H.261, H.263, MPEG-1, MPEG-2, MPEG-4 and H.264 [1] due to its effectiveness and simple implementation. The most straightforward BMA is the *full search* (FS), which exhaustively evaluates all the possible candidate blocks within the search window. However, this method is very computationally intensive, and can consume up to 80% of the computational power of the encoder. This limitation makes ME the main bottleneck in real-time video coding applications. Consequently, fast BMAs are indispensable to decrease the computational cost.

In the past decades, many fast BMAs were proposed for alleviating the heavy computation of the FS, such as three-step search (TSS) [2], new three-step search (NTSS) [3], four-step search (4SS) [4], block-based gradient descent search (BBGDS) [5], diamond search (DS) [6, 7], and hexagon-based search (HEXBS) [8] algorithms, etc. Because of the center-biased global minimum motion vector (MV) distribution characteristics, more than 80% of the blocks' MVs could be regarded as stationary or quasi-stationary and most of the MVs are very close to the central area of the search window [6]. Therefore, it is reasonable to assume that ME matching error decreases monotonically as the search moves toward the position of the globally optimal solution. Based on this assumption, square-shaped search patterns of different sizes are employed in the NTSS, 4SS and BBGDS [3]-[5]. The DS [6, 7] employs diamond-shaped patterns, which results in similar distortion performance as compared to those of the NTSS and 4SS, with even fewer search points. This approach has been shown to be very effective for the center-biased MV distribution model. The DS adopts two diamond-shaped search patterns [6]: 1) large diamond search pattern (LDSP) with nine search points and 2) small diamond search pattern (SDSP) with five search points. The LDSP is repeated until that it reaches the edge of the search window, or a new minimum matching distortion point occurs at the center of LDSP. The search pattern is then switched to SDSP, which is used to refine the search algorithm [7]. Similarly, HEXBS applies the same search strategy only by replacing the diamond-shaped search pattern with a hexagon-shaped search pattern which saves computational energy with slightly decreased performance [8].

In these fast BMAs, since the search patterns are fixed, it is not efficient when the shapes of the fixed search patterns do not match the actual motion. Recently, we propose a novel fast BMA, namely, adaptive multi-mode search (AMMS) algorithm [9]. Instead of using the same search pattern, by using MV prediction together with the SDSP as the initial search step, we categorize them into four search modes with different shapes based on the analysis of the initial step. By carefully choosing the best suitable mode, it is more effective to capture the true motion direction. In this paper, the computational complexities in these fast BMAs are compared based on the assumption of center-biased MV distribution. Two statistical models are used to characterize the MV distribution. Both theoretical and experimental comparisons are presented.

2. ADAPTIVE MULTI-MODE SEARCH ALGORITHM (AMMS)

Fig. 1 shows the four different modes used in the AMMS algorithm. From Fig. 1, let $P1$ and $P2$ denote the two points of the SDSP around the starting search point with smallest and second smallest sum of absolute differences (SADs) respectively. Geometrically, there are four types of possible combinations of $P1$ and $P2$: A) $P1$ is the center point, B) $P2$ is the center point, C) neither $P1$ nor $P2$ is the center point and they are on the same edge of the SDSP, D) neither $P1$ nor $P2$ is the center point and they are unconnected vertices of the SDSP. Suppose the SAD is chosen as a block matching distortion measure. The AMMS algorithm can be summarized as follows [9].

(1) Calculate the SADs for the MV of (0,0) and the MVs of the blocks immediate left and above to the current one, and choose the one with the smallest SAD as the predicted starting point.

(2) Apply SDSP search by evaluating and comparing the SADs of the five candidates to identify $P1$ and $P2$.

(3) Use the combination of $P1$ and $P2$ to determine the search modes as described above.

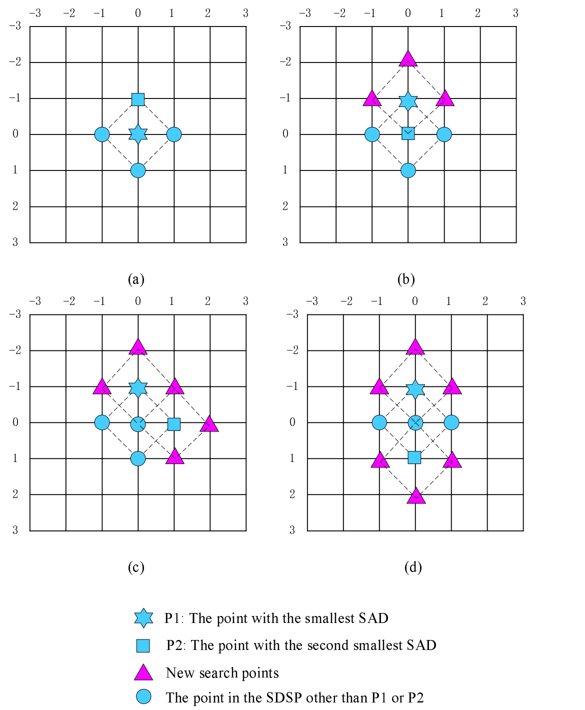


Fig. 1. Four different modes in the AMMS algorithm: (a) $P1$ is the center point, it is chosen as the target MV; (b) $P2$ is the center point, three more candidates around $P1$ are evaluated; (c) $P1$ and $P2$ are on the edge of the small diamond, five more points neighboring to them are evaluated; (d) $P1$ and $P2$ are vertices of the small diamond, six more surrounding candidates are evaluated.

(4) Mode A): select $P1$ as the final solution and stop the search; Mode B): evaluate three more candidates around $P1$; Mode C): evaluate five more points neighboring to $P1$ and $P2$; and Mode D): evaluate six more surrounding points of $P1$ and $P2$. If it reaches the border, stop the search; else return to (3).

3. COMPLEXITY ANALYSIS

As discussed previously, it is assumed that the global minimum is located near the starting search point with the benefit of small search patterns, i.e. the center-biased MV distribution. Based on this motivation, this section aims to quantify the speed improvement obtained by the DS, HEXBS and AMMS. Since the 4SS performs best compared to other conventional BMAs such as the 3SS and NTSS, and has been implemented in industrial hardware [7], it is chosen as a reference for our comparison.

In order to simplify the analysis, we assume that the starting search point is always at the $(0,0)$ position. Let $n_{s,\cdot}(i,j)$ be the minimum number of search points used to calculate SADs before reaching the global minimum when it occurs at (i,j) position, where $\cdot$ indicates the search method. Fig. 2(a), (b), (c) and (d), respectively, show $n_{s,4SS}(i,j)$, $n_{s,DS}(i,j)$, $n_{s,HEXBS}(i,j)$ and $n_{s,AMMS}(i,j)$ within the central $\pm M$ region with $M = 4$ for the 4SS, DS, HEXBS and AMMS. It can be noticed that $n_{s,\cdot}(i,j)$ in every search method grows as $|i|$ or $|j|$ increases, i.e. when it is further away from the starting point. It is evident from Fig. 2

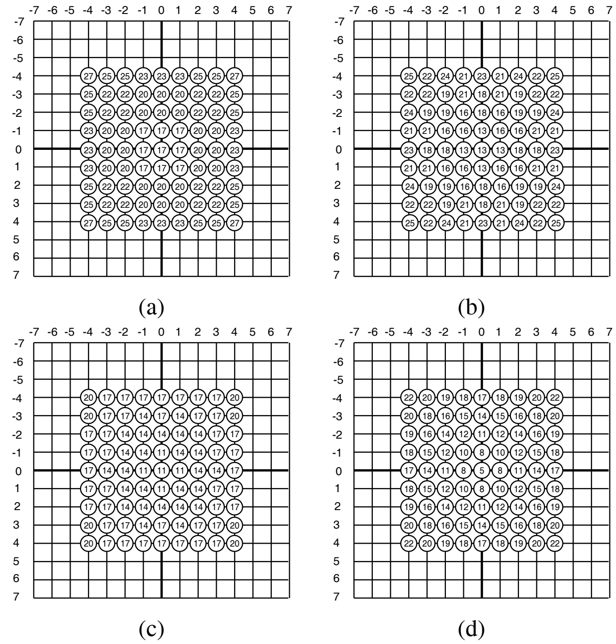


Fig. 2. Possible minimum number of search points for fast BMAs: (a) 4SS, (b) DS, (c) HEXBS and (d) AMMS.

that $n_{s,AMMS}(i,j)$ is uniformly lower than $n_{s,DS}(i,j)$ which implies that the AMMS has lower computational complexity than that of the DS in all situations. In the case of $n_{s,HEXBS}(i,j)$, they are higher than $n_{s,AMMS}(i,j)$ at most of the positions except for some points at the top and bottom borders where they are slightly lower by one or three search points. The average gain in term of number of search points per block of the DS, HEXBS or AMMS over 4SS can be quantified as follows [7]:

$$\overline{G_{\cdot}} = \sum_{i=-M}^{M} \sum_{j=-M}^{M} [n_{s,4SS}(i,j) - n_{s,\cdot}(i,j)] \times P_s(i,j), \quad (1)$$

where $(2M+1) \times (2M+1)$ is the area where we assume that the probability of having MV is nonzero, $\cdot$ is either DS, HEXBS or AMMS, and $P_s(i,j)$ is the probability that the optimal MV is located at the (i,j) position. Since the matching errors are assumed to be monotonically decreasing toward the global minimum point, we use the Gaussian and Laplacian distributions as appropriate models for our statistical analysis. We assume that the optimal MVs are within the $\pm M$ region ($M = 4$) since the probability of MVs distributed outside the ± 4 region is lower than 8% [10].

3.1. $\overline{G_{\cdot}}$ based on the Gaussian distribution model

Assume that x and y are independent Gaussian random variables for horizontal and vertical directions respectively with variance of σ^2 and zero mean. The 2-D probability density function (pdf) can be written as:

$$f_{XY}(x,y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (2)$$

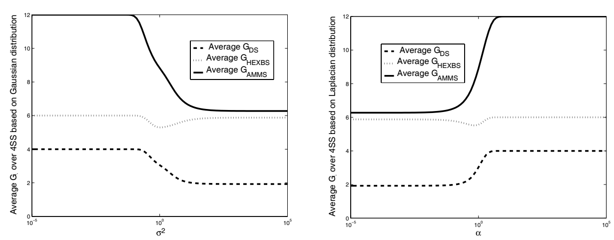
$$\text{Let} \quad P_s(i,j) = \int_{i-0.5}^{i+0.5}\int_{j-0.5}^{j+0.5} f_{X,Y}(x,y)\,dxdy$$
$$= \int_{i-0.5}^{i+0.5}\int_{j-0.5}^{j+0.5} \frac{1}{2\pi\sigma^2}\cdot e^{-\frac{x^2+y^2}{2\sigma^2}}\,dxdy, \quad (3)$$

for $-4 \leq i,j \leq +4$. Since it is assumed that no MV is located outside the ± 4 region, the corresponding probability $P_s(i,j)$ is forced to be zero, i.e. $P_s(i,j) = 0, |i| > 4, |j| > 4$. The normalized probability $\hat{P}_s(i,j)$ is defined as:

$$\hat{P}_s(i,j) = \frac{P_s(i,j)}{\sum\limits_{i=-\infty}^{\infty}\sum\limits_{j=-\infty}^{\infty} P_s(i,j)} = \frac{P_s(i,j)}{\sum\limits_{i=-4}^{4}\sum\limits_{j=-4}^{4} P_s(i,j)} \quad (4)$$

Therefore $\overline{G_\cdot}$ can be calculated as follows:

$$\overline{G_\cdot} = \sum_{i=-4}^{4}\sum_{j=-4}^{4} [n_{s,4SS}(i,j) - n_{s,\cdot}(i,j)] \times \hat{P}_s(i,j). \quad (5)$$



(a) (b)

Fig. 3. Theoretical analysis: $\overline{G_{DS}}$, $\overline{G_{HEXBS}}$ and $\overline{G_{AMMS}}$ based on (a) Gaussian distribution, and (b) Laplacian distribution.

Fig. 3(a) shows the plots of $\overline{G_{DS}}$ (dashed line), $\overline{G_{HEXBS}}$ (dotted line) and $\overline{G_{AMMS}}$ (solid line) for different values of σ. It is clear that $\overline{G_\cdot}$ is positive for the three cases at all values of σ. Specifically, as σ increases from 0 to ∞, $\overline{G_{DS}}$ decreases from 4.00 to 1.93, $\overline{G_{HEXBS}}$ decreases from 6.00 to 5.30 then increases to 5.88, and $\overline{G_{AMMS}}$ decreases from 12.00 to 6.27.

When it is highly concentrated at the center, i.e. $\sigma \ll 1$, $\overline{G_{DS}}$, $\overline{G_{HEXBS}}$ and $\overline{G_{AMMS}}$ have the highest gains of 4.00, 6.00 and 12.00, respectively. Among these three fast BMAs, it is obvious that AMMS has the highest computational gains for small motion video sequences. For the DS and AMMS, the worst case occurs when $\sigma \to \infty$ where $\overline{G_{DS}}$ converges to 1.93 and $\overline{G_{AMMS}}$ converges to 6.27, while the worst case occurs when $\sigma = 1$ for HEXBS, where $\overline{G_{HEXBS}}$ has the lowest gain of 5.30. This implies that even when the motion is highly unpredictable ($\sigma = \infty$), AMMS still yields substantial computational gains over the other two search methods.

3.2. $\overline{G_\cdot}$ based on the Laplacian distribution model

Similar to the above calculation for the case of Gaussian distribution, the Laplacian distribution is used to calculate $\overline{G_\cdot}$ in this subsection. Let

$$f_{XY}(x,y) = \frac{\alpha^2}{4}\cdot e^{-\alpha(|x|+|y|)} \quad (6)$$

be the Laplacian pdf where α is a constant parameter ranging from 0 to ∞, determining degree of concentration of the distribution. Using (6) in (3) and (5), $\overline{G_\cdot}$ can be re-calculated for the cases of the DS, HEXBS and AMMS as plotted in Fig. 3(b). It is evident that $\overline{G_\cdot}$ is uniformly positive for the three fast BMAs and achieves the same lower bounds and upper bounds as those obtained under the Gaussian distribution model.

Obviously, the computational complexity of the AMMS is consistently lower than that of the DS and HEXBS. When the search window size is getting larger, $\overline{G_{HEXBS}}|_{\alpha=0}$ may increase. For fast changing sequences, $n_{s,AMMS}(i,j)$ beyond the $|i|,|j| > 4$ region can be higher than $n_{s,HEXBS}(i,j)$, and thus the HEXBS seems to be more efficient. However, both prediction of the starting search point and the adaptive mode selection techniques employed in the AMMS can optimize and improve the speed in approaching the global minimum. In addition, such a situation is very unlikely with low probability [10].

4. EXPERIMENTAL RESULTS

The four fast BMAs, the 4SS, DS, HEXBS and the proposed AMMS algorithms, are simulated using the luminance of popular test video sequences listed in Table 1, which consist of different formats and motion contents. SAD is used to evaluate the distortions for block matching with the block size of 16×16 and search window of ± 7.

The average numbers of search points and average MSE values, which are obtained from these video sequences by using the 4SS, DS , HEXBS and AMMS algorithms, are presented in Table 2 and Table 3, respectively.

From Table 2, the actual $\overline{G_{DS}}$ is about 2.12 to 3.58, $\overline{G_{HEXBS}}$ varies from 5.83 to 7.73, and $\overline{G_{AMMS}}$ has the range of 10.08 to 13.08. Compared with the analysis results obtained from Section 3, the actual $\overline{G_{DS}}$ confirms the theoretical gain, $\overline{G_{HEXBS}}$ and $\overline{G_{AMMS}}$ keep values higher than the lower bounds, but exceed the upper bound in some situation. The reason is that many MVs of the real-world sequences are highly distributed within the center ± 1 region, which increases the actual upper bound. For the AMMS, another reason is because of MV prediction before the initial SDSP which further eliminates some unnecessary search steps. Thus additional gain can be added to the theoretical results which are based on the assumption that the starting search point is located at the $(0,0)$ position for all methods.

Notice that our AMMS algorithm reduces the number of search points remarkably over the 4SS, DS and HEXBS while improving the MSE slightly at the same time for all the tested sequences as shown in Table 3. It can be seen from Table 2 that the AMMS actually performs very competitively in terms of low block MSE distortion while deducting more than 50% search steps for fast changing sequences, such as “Football” and “Foreman”. The speed improvement is also quite substantial for sequences, such as “Tempete”, where large quantities of small moving directions are involved.

Table 1. Video Sequences Used for Simulation.

Format	Video Sequence	Number of frames
QCIF (176×144)	Carphone	96
	Football	111
SIF (352×240)	Garden	114
	Mobile	140
CIF (352×288)	Foreman	300
	Tempete	260

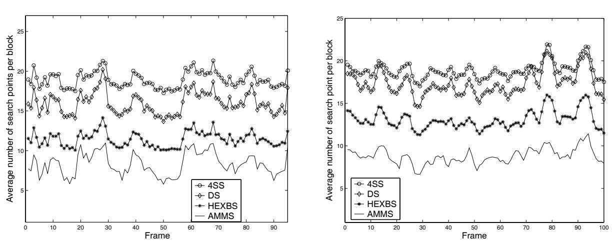
Table 2. Average Number of Search Points Per Block With Respect to Different Methods and Different Video Sequences (Search Window: ± 7).

Video	4SS	DS	HEXBS	AMMS
Carphone(QCIF)	18.94	15.98	11.43	8.29
Football(QCIF)	19.30	16.45	12.39	8.39
Garden(SIF)	19.87	17.45	13.10	6.78
Mobile(SIF)	17.33	13.75	10.62	5.16
Foreman(CIF)	19.44	17.32	12.95	8.12
Tempete(CIF)	16.93	14.46	11.10	6.85

Table 3. Average MSE per pixel for Different Methods and Different Video Sequences (Search Window: ± 7).

Video	4SS	DS	HEXBS	AMMS
Carphone(QCIF)	34.37	34.31	34.63	34.26
Football(QCIF)	74.64	74.46	74.59	74.20
Garden(SIF)	88.62	88.26	88.78	88.06
Mobile(SIF)	104.21	104.29	104.34	104.13
Foreman(CIF)	34.60	34.47	34.95	34.43
Tempete(CIF)	58.35	58.40	58.46	58.29

Fig. 4 plots the average search points per block for each frame using different fast BMAs for the "Carphone (QCIF)" and "Foreman (CIF)" sequences. Comparing frame by frame, it is evident that the number of search points for the case of AMMS is uniformly lower than that of other search methods. Fig. 5 plots the corresponding frame-wise MSE measurements, which are similar among all fast BMAs. These figures further verify that the AMMS algorithm consistently yields comparable distortion error in term of MSE while maintaining substantial improvement on the searching speed.

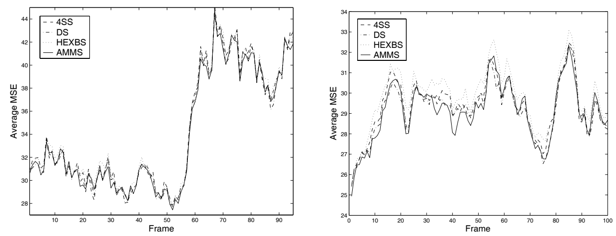


(a) (b)

Fig. 4. Frame-wise performance comparison of average number of search points per block between different BMAs with search window ± 7: (a) Carphone(QCIF), (b) Foreman(CIF).

5. CONCLUSION

In this paper, a theoretical complexity analysis for fast BMAs is discussed based on the assumption that ME matching error decreases monotonically as the search approaches the position of the global minimum error. The Gaussian and Laplacian distribution models are applied to our analysis, which shows that our AMMS outperforms the other popular fast BMAs, such as the 4SS, DS and HEXBS while significantly reducing the computational complexities. The experimental results show that the AMMS achieves the



(a) (b)

Fig. 5. Frame-wise performance comparison of mean square error per pixel between different BMAs with search window ± 7: (a) Carphone(QCIF), (b) Foreman(CIF).

greatest improvement.

6. REFERENCES

[1] Abdul H. Sadka. *Compressed Video Communications*. John Wiley and Sons Ltd, 2002.

[2] T. Koga, K. Iinuma, A. Hiranoa, Y. Iijima, and T. Ishiguro. "motion compensated interframe coding for video conferencing". pages G5.3.1–G5.3.5, Nov. 29-Dec. 3 1981.

[3] R. Li, B. Zeng, and M.L. Liou. "a new three-step search algorithm for block motion estimation". *IEEE Trans. Circuits Syst. Video Technol.*, 4:438–442, Aug. 1994.

[4] L. M. Po and W. C. Ma. "a novel four-step search algorithm for fast block motion estimation". *IEEE Trans. Circuits Syst. Video Technol.*, 6:313–317, June 1996.

[5] L. K. Liu and E. Feig. "a block-based gradient descent search algorithm for block motion estimation in video coding". *IEEE Trans. Circuits Syst.Video Technol.*, 6:419–423, Aug. 1996.

[6] Shan Zhu and Kai-Kuang Ma. "a new diamond search algorithm for fast block-matching motion estimation". *IEEE Trans. Image Processing*, 9:287–290, Feburary 2000.

[7] 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.*, 8:369–277, Aug. 1998.

[8] Ce Zhu, Xiao lin, and Lap-Pui Chau. "hexagon-based search pattern for fast block motion estimation". *IEEE Trans. Circuit and Systems for Video Technology*, 12:349–355, Feb. 2002.

[9] Y. Liu and S. Oraintara. "adaptive multiple-mode search algorithm for fast block-matching motion estimation". *submitted to IEEE Trans. Circuits and Systems for Video Technology*, Sept. 2003.

[10] C-H. Cheung and L-M. Po. "a novel cross-diamond search algorithm for fast block motion estimation". *IEEE Trans. Circuits and Systems for Video Technology*, 12:1168–1177, December 2002.