10

Enhanced Efficient Diamond Search Algorithm for Fast Block Motion Estimation

Yasser Ismail and Magdy A. Bayoumi
The Center for Advanced Computer Studies, University of Louisiana at Lafayette
USA

1. Introduction

Although the conventional *FS* algorithm achieves the best quality amongst various Motion Estimation (*ME*) algorithms and it is straightforward and has been successfully implemented on *VLSI* chips [1], its computational complexity is very high. In contrast, real time and portable multimedia devices require ultra computationally efficient video codec designs that will allow for a robust and reliable video quality. Many sub-optimal but faster *ME* techniques have been proposed to tackle the previous computational complexity problem. One technique is based on simplifying or easing the matching criteria (*SAD*) based on spatial and/or temporal Macro Blocks' (*MB*) features. Partial Distortion Elimination algorithm (*PDE*) and Successive Elimination Algorithm (*SEA*) [2] are examples of such techniques. Another technique is based on reducing the number of search points in the search area [3]. Although this technique reduces the computational complexity, there will be degradation in *PSNR*. New Three Step Search (*N3SS*) [4], Four Step Search (*FSS*) [5], Predictive Motion Vector Field Adaptive Search Technique (*PMVFAST*) [6], Hexagon Based Search (*HEXBS*) [7], Diamond Search (*DS*) [8], and Cross Diamond Search (*CDS*) [9] are examples for such techniques.

In this chapter, a new simple and efficient fast *MDS* is developed for higher complexity reduction than *DS* without further *PSNR* and bit-rate degradation compared to *FS*. *DS* is the most accurate suboptimal *ME* algorithm among others. This is why it was chosen to be implemented in the reference software of the standard H.264. The proposed *MDS* algorithm uses a mixed flavor approach of the previous two techniques. A computationally efficient set of stop search algorithms that skip the unnecessary operations both internally within the *MB* and externally between *MB*s utilizing accurate adaptive threshold models for reducing computations with no significant degradation in *PSNR* compared to the *FS* algorithm. Internal *SAD* operations are minimized using *DISS*. *DESS* algorithm is used to eliminate the irrelevant candidate pixels in the search area. Moreover, we are proposing to enhance the mechanism of the *DS* algorithm using zero and center bias properties. A set of adaptive and accurate thresholds that early terminate the search or select between Small Diamond Search Pattern (*SDSP*) or Large Diamond Search Pattern (*LDSP*) results in an additional computational saving. It is worth mentioning that some related work of this chapter was published in [11].

Coming up in section 2, details on the proposed modified search algorithm are explained. Following that, in section 3 comparisons are made between the proposed *MDS* and other fast motion estimation algorithms. Finally, we draw conclusion in section 4.

2. The proposed modified diamond search

It is well known that the motion field of consecutive frames in a video sequence is smooth and gentle. This means that the optimum Global Motion Vectors (*GMVs*) are located at or very close to the search center most of the time. Fig.1 illustrates the distribution of the optimum *GMVs* using *FS* with a spiral search pattern for six video sequences that contain different motion activities (low, medium, and high motion activities). It is noted that, for low motion activity video sequences, Akiyo, Coast-Guard, and News, the optimum *GMV*s are located at the search center most of the time (zero bias property). The situation is changed for the medium and high motion activity video sequences, Mobile, Foreman, and Football, at which the distance of the optimum *GMV*s from the search center will slightly increase (center bias property).

In this chapter, we use the zero bias property to reduce the computational complexity by deciding if the Initial Search Center (*ISC*) can be considered as an *GMV* or not. This will be achieved by using a dynamic *ZMP* (Zero Motion Prejudgment) threshold (T_{DESS}) which is dynamically adapted according to the change of the current block's features. We also benefit from the center bias property to reduce the computational complexity of the *DS* by dynamically selecting the appropriate diamond search pattern (either *SDSP* or *LDSP*) using a dynamic threshold T_P.

Using *SDSP* in Fig.2.a will reduce the computational complexity but will also increase the possibility of falling into local minima if it is used at the beginning of the search in a high motion or irregular video sequence. In contrast, using the *LDSP* in Fig.2.b may tackle this problem but with an increase in the computational complexity since larger number of checking points will be used. To remedy the previous problems, we adaptively select between starting with either *SDSP* or *LDSP* according to the expected motion activity of the current block. If the motion activity is low, an *SDSP* will be used. Otherwise, an *LDSP* will be used. Using the fact that there is a high correlation between neighboring blocks, the motion activity of the current block can be easily expected from the optimum *GMV*s of the previously encoded neighboring blocks so far. The threshold value in Eq.1 will be used to decide the expected motion activity of the current block.

To explain the proposed *MDS* algorithm, let the search window size be (-Δ, Δ) and the displacement with respect to the current block located at (*u*, v) be (*x*, *y*). The *SAD* between the current block in frame *n* and the candidate block in frame *n*-Ψ is described in Eq. 2:

The new modified diamond search can be summarized as follows:

Initial step (*ZMP* decision): The $SAD_{curr}(i)$ between the current block and the candidate block *i* at the *ISC* will be estimated. Also, the thresholds T_{DESS} (the derivation of this threshold will be given in section 2.1.2) and T_P are calculated. If $SAD_{curr}(i) \le T_{DESS}$, then the search will stop immediately and outputs the candidate block *i* located at the *ISC* position as a best matching candidate block as seen in Fig.3.a. Otherwise, go to step (i).

Step(i) (*SDSP/LDSP* decision): Using the threshold T_P, the initial used diamond pattern will be decided according to the following condition. If $T_P \le 1$, go to step(ii). Otherwise, go to step(iii).

Step(ii) (Small Diamond Search Pattern (*SDSP*)): Four search points as in Fig.3.b will be checked one by one against the minimum *SAD* so far using the stop search procedure with both *DISS* and *DESS* techniques that will speed up the *ME* operation by eliminating unnecessary computations as will be discussed in section 2.1.1 and section 2.1.2. If all the

points in the *SDSP* didn't have any chance to hit the best matching block, then check the search point with the minimum *SAD*. If it is located at the center of *SDSP*, then safely stop the search. Otherwise, use this point as an *ISC* and repeat step ii (see Fig.3.c).

Step(iii) (Large Diamond Search Pattern (*LDSP*)): Eight points as in Fig.2.b will be checked one by one using the procedure of both the *DISS* and *DESS* techniques. If the entire candidate points in the *LDSP* are checked without hitting the best match, the search point with the minimum *SAD* will be checked. If this point is not located at the *ISC*, then start step(iii) over again and consider this point as an *ISC* for the new *LDSP*. Otherwise, if it is located at the *ISC*, an *SDSP* will be used as a final stage. If the best match is not caught by the procedure, then the point with the minimum *SAD* will be the best match as seen in Fig.3.d.

It is worth mentioning that the local minima problem, which appears in the conventional *DS* algorithm, disappears from the proposed *MDS* due to the high accuracy of estimating the motion vectors using the proposed thresholds.

3. The proposed stop search algorithms

The total computational complexity TC for any block based motion estimation technique can be expressed in Eq. 3. In this section, we reduce the computational complexity in Eq.3 by reducing the Sub, Abs, and Add operations per candidate block using smart DISS. Also we reduce the number of the search point S using smart DESS. The proposed algorithms are described in detail in the following sub sections.

3.1 Dynamic Internal Stop Search algorithm (DISS)

The main purpose of the proposed *DISS* is to reduce the computational complexity of *ME* process by reducing the *Sub*, *Abs*, and *Add* operations for each candidate block. This will consequently reduce the term $\left[l_1 \times l_2 \,.\left(Sub + Abs + Add\right)\right]$ in Eq.3. The *DISS* algorithm can be summarized in the following four steps.

Step (i): Both the current and the candidate blocks in the current and the reference frames respectively are divided into groups. Each group contains a number of row lines (2l), where $l = 0, 1, 2, \ldots, l_1 / 4$

Step (ii): Partially accumulate the *SAD* value starting at the first group in both the current and the candidate blocks.

Step (iii): compare the partially accumulated *SAD* value so far (*PSAD*) with a pre-determined dynamic threshold T_{DISS}.

Step (iv): If the accumulated $PSAD > T_{DISS}$, then stop accumulating *PSAD* for further groups and proceed to the next candidate block in the search window. Otherwise, continue accumulating *PSAD* by adding the next group of pixels to the previous *PSAD* and update the threshold accordingly.

The threshold T_{DISS} depends mainly on the normalized minimum *SAD* of the scanned blocks so far in the search window ($SADmin\text{-}curr/[l_1 \times l_2]$). T_{DISS} is expressed in Eq. 4.

However, sometimes some blocks are falsely skipped in the early stages of the algorithm. It was noticed experimentally that if the *PSAD* of the first group(s) for a candidate block is greater than the calculated threshold $T_{DISS}(j)$, the candidate block is skipped. Although, if we further continue accumulating the *PSAD* for the next group(s) in that block, the final accumulated *PSAD* might not exceed the threshold $T_{DISS}(j)$. This problem is avoided by

adding an accelerator parameter Ω to the threshold value in Eq.4 to control the rate of the *PSAD* skipping operation in the candidate block. The value of Ω parameter is illustrated in Eq.5:

Our exhaustive experiments reveal that the previous false skipping of *PSAD* operations is not the dominant case for all the candidate blocks. Also, these false skipping operations depend mainly on the choice of the initial group to be partially accumulated. So, the effect of the acceleration parameter Ω should be lessened with the further accumulating of *PSAD* from one group to the next one. This will be achieved by subtracting a relaxation parameter θ from the total threshold T_{DISS} in order to relax the effect of adding the accelerator parameter Ω. The value of θ is illustrated in Eq.6 where N is the total number of the groups in a block. At the last group of a block, Ω will be completely eliminated by θ and T_{DISS} will settle to the value of $SAD_{min-curr}$. The final form of the proposed *DISS* threshold is given in Eq.7.

It is worth mentioning that all the divisions and multiplications in Eq.7 can be simply replaced by shift operations to speed up the performance of the proposed algorithm.

3.2 Dynamic External Stop Search algorithm (DESS)

The main purpose of the *DESS* algorithm is to reduce the computational complexity given in Eq.3 by reducing the redundant candidate blocks in the search window that can not be considered as an optimum solution for the current block. If the previous *DISS* algorithm fails, a second level of reduction will be used. If the *PSAD* calculations of a reference block are not skipped, then the value of the $SAD_{min-curr}$ for the current block will be updated according to the value of the obtained final *PSAD* for that candidate block. If the final *PSAD* is less than the $SAD_{min-curr}$, then the value of the $SAD_{min-curr}$ is replaced by the current value of the final *PSAD*. Thereafter, we check this updated $SAD_{min-curr}$ against a pre-determined Dynamic External Stop Search threshold $T_{DESS}(i)$; where i is the index of the candidate block in the search window. If the updated $SAD_{min-curr} \le T_{DESS}(i)$, then skip all the remaining candidates in the search window and select the i block as our best candidate block.

Experimental analysis reveals that the best match block in the search window has a minimum *SAD* that is highly correlated with the minimum *SAD*s of the neighbors of the current block, i.e., blocks 1, 2, 3, and 4 in Fig.4. Therefore, the optimum minimum *SAD*s of the neighboring blocks to the current block can be used to build the function £ in the linear model of Eq.8 to form the threshold $T_{DESS}(i)$. A small matrix is required to store the minimum *SAD* values of the coded neighbor blocks so far. The function £ can be simply set to the average of the minimum *SAD*s of the neighbors surrounding the current block. Nevertheless, an improvement in the accuracy of the threshold $T_{DESS}(i)$ can be obtained by replacing the average value by the median function. Implicitly, using the median function is considered as the average of the best two neighboring blocks since it will exclude the irregular minimum *SAD* values of the neighbors to the current block from the calculations.

The proposed model in Eq.8 requires only three shift operations and one addition operation (required for the constant μ_1). If we use £ as the average, three addition operations and two shift operations will be also required. However, using the median would increase the complexity as it also requires comparison operations. Fig.5 illustrates the whole flow diagram of the proposed *DISS* and *DESS* algorithms combined together.

4. Simulation results

Our experiments have been conducted, based on the reference software JM12.4 [10], to show the effectiveness of the proposed MDS algorithm. Three main parameters are used for measuring the performance. First, the PSNR difference (ΔPSNR) of the reconstructed video which is the PSNR difference between FS and the fast ME techniques. Second, the increase in the bit-rate percentage (ΔBR%). Finally, the ME time saving percentage (METS%). ΔPSNR, ΔBR%, and METS% are measured with respect to FS. Due to lack of space, only four different video sequences, with different motion activities, are illustrated here. Motion Vector (MV) search is based on the luminance component of the block of size 16×16. The search window size is 32×32. The proposed MDS algorithm is compared against FS, N3SS, 4SS, PMVFAST, HEXBS, DS, and finally CDS algorithms. It is clear from table 1 that the speed up of the proposed MDS is significant compared with other fast ME techniques. Practically, it achieves approximately 99% and 20% saving in ME time, for high motion video sequence (Football), compared with FS and DS respectively. This is with a negligible degradation in both PSNR and bit-rate.

Fig.6 represents the average number of Absolute Difference (AB) operations per MV for Foreman video sequence. It is clear from the figure that the MDS algorithm has the lowest AB operations compared with other fast ME search techniques. This reflects the superior performance of our proposed MDS algorithm to speed up the ME process. The performance of the proposed algorithm is also measured using the Rate Distortion (RD) curves in Fig.7 (calculated at 30 frames per sec). It is clear from Fig.7 that the proposed MDS performs very close to the DS algorithm and better than CDS algorithm.

5. Tables and figures

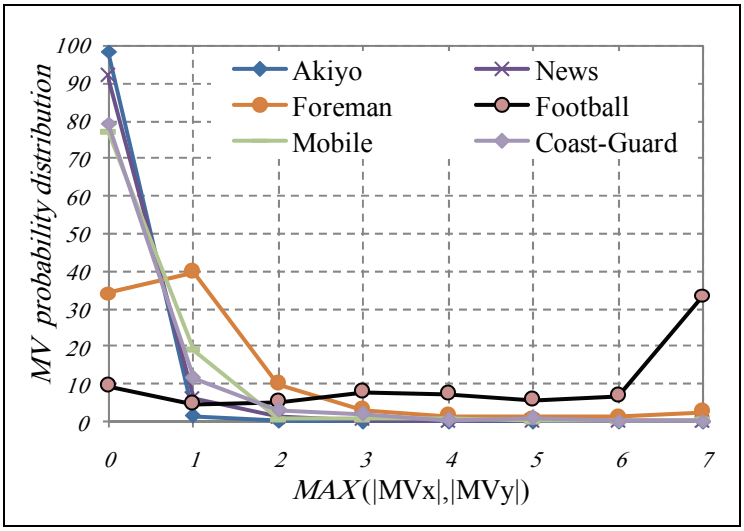


Fig. 1. Motion vector probability distribution using FS (16×16 search window size).

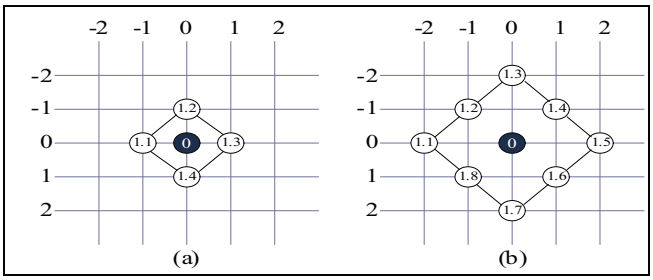


Fig. 2. Diamond Search Patterns (a) SDSP. (b) LDSP.

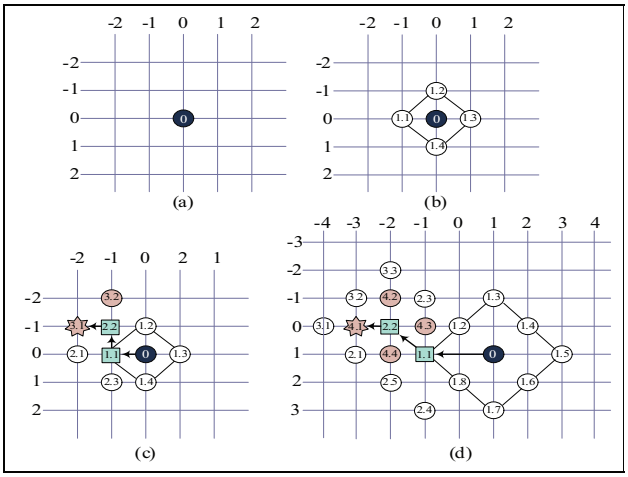


Fig. 3. Modified Diamond Search Algorithm (MDS).

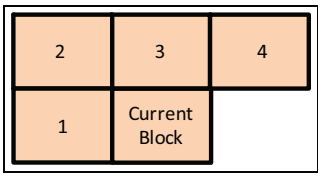


Fig. 4. Current Macro-block and its 4 neighbors used for calculating $T_{DESS}(i)$.

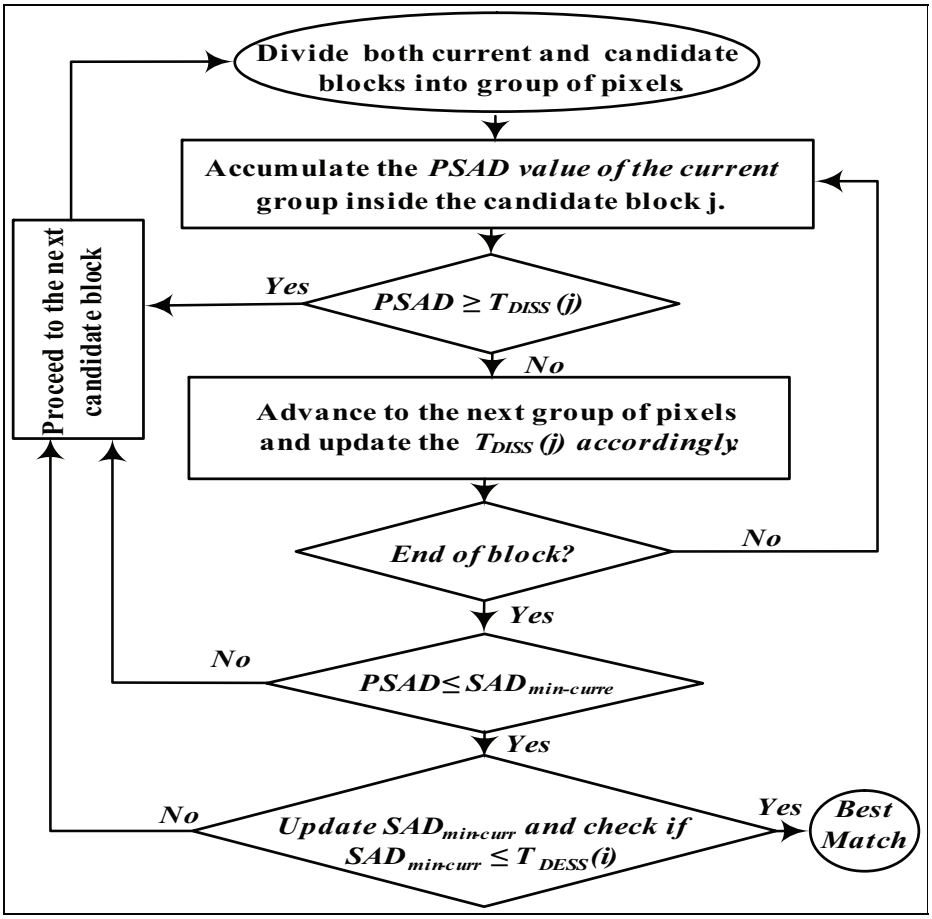


Fig. 5. The flow chart of the proposed DISS and DESS algorithms.

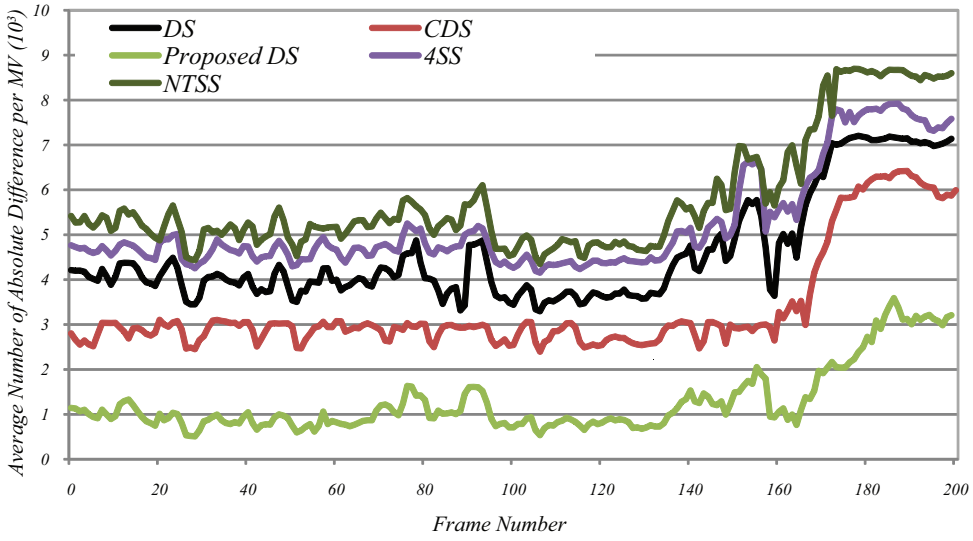


Fig. 6. The total computational complexity per MV for Foreman video sequence.

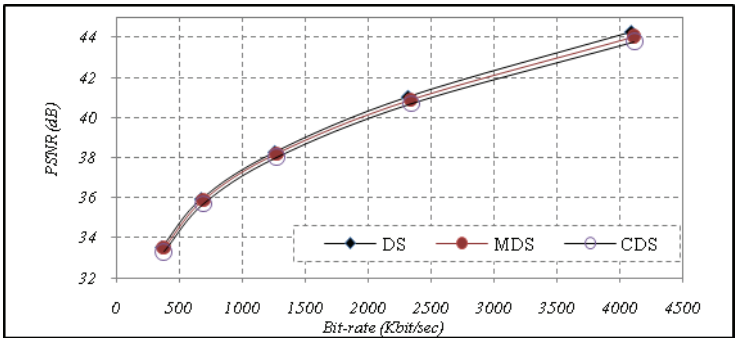


Fig. 7. Bit-rate Distortion Curves for Foreman video sequence using the proposed MDS and other fast ME techniques.

Table 1. The performance of the proposed MDS compared with FS algorithm and the state of the art fast motion estimation techniques using QP=28

ME technique	Akiyo (QCIF)			Forman (CIF)			Coast-Guard (QCIF)		
	PSNR(dB)	BR	METS%	PSNR(dB)	BR%	METS	PSNR(dB)	BR	MET
FS	38.55	28650	194.796	37.71	474840	1143.653	34.79	243900	671.3
	ΔPSNR(dB)	BR%	METS%	ΔPSNR(dB)	BR%	METS%	ΔPSNR(dB)	BR%	METS
N3SS [4]	-0.01	0.09	74.62	-0.4	0.81	70.04	-0.04	0.08	72.1
4SS [5]	0.00	0.05	79.93	-0.2	0.43	74.64	-0.01	0.06	77.2
PMVFAST[6]	-0.06	-0.21	85.71	-0.05	-2.06	95.13	-0.01	0.05	93.8
HEXBS [7]	0.00	0.03	95.01	-0.02	0.98	94.56	0.00	0.04	95.1
DS [8]	-0.06	-0.31	85.73	-0.03	-1.88	82.12	0.00	0.19	87.7
CDS [9]	-0.05	0.21	89.11	-0.05	1.98	88.63	-0.07	2.64	92.1
MDS	-0.01	0.17	99.37	-0.03	2.32	99.15	0.00	0.29	99.4

6. Equations

$$T_P = \text{Median}\{GMV_{min}(m)\}, \quad m = 1, 2, \ldots, K \tag{1}$$

Where GMV_{min} is the minimum *GMV*s of the neighboring blocks *m* and *K* is the number of the GMV_{min} that are used to calculate T_P. The small value of T_P is an indication that the best match is very close to the search center. A large value of T_P means that the best match block is far away from the *ISC*.

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|I_n(u,v) - I_{n-\Psi}(u+x, v+y)\right| \tag{2}$$

Where $-\Delta \le x, y \le \Delta$, *N* is the block size, and *I* is the pixel intensity.

$$TC = \left[l_1 \times l_2 . (Sub + Abs + Add)\right].S \tag{3}$$

Where l_1 and l_2 are the number of rows and columns of both the current block and the reference block respectively. S is the total search points in a search window of size $(-\Delta, \Delta)$. Sub, Abs, and Add are the number of subtraction, absolute value, and addition operations required for calculating the matching criteria defined by the SAD operation.

$$T_{DISS}(j) = j \times P \times \frac{SAD_{min-curr}}{l_1 \times l_2} \tag{4}$$

Where *j* and *P* are the group index and the number of pixels per group respectively. $SAD_{min-curr}$ is the minimum SAD of the scanned blocks so far in the search window.

$$\Omega = \varepsilon \times \frac{SAD_{min-curr}}{l_1 \times l_2}, \quad \varepsilon = 1, 2, \ldots, P/2 \tag{5}$$

The higher the value of Ω is, the harder it is to skip the *PSAD* calculations for the candidate block which consequently increases the ability to find the optimal *GMV*. But, this defeats our target here as it might cause an increase in the non-skipped blocks and hence increasing the computational complexity.

$$\theta = \frac{(j-1)\times\Omega}{(N-1)} \tag{6}$$

$$T_{DISS}(j) = j \times P \times \frac{SAD_{min-curr}}{l_1 \times l_2} + \Omega - \frac{(j-1)\times\Omega}{(N-1)} \tag{7}$$

$$T_{DESS}(i) = \mu_1 \times £ + \mu_2 \tag{8}$$

Where the statistical function £ is defined as:

$$£ = \frac{1}{N}\sum_{m=1}^{N} SAD_{min}(m) \quad \text{or,} \quad \text{Median}\{SAD_{min}(m)\}, \quad m = 1, 2, \ldots, K \tag{9}$$

Where K is the number of the closest neighbors coded so far, μ_1 is an accelerator parameter, and μ_2 is a constant factor set experimentally to zero. Note that this threshold is computed only once per a current block. The accelerator parameter μ_1 is set experimentally to 0.75.

7. Referring

(Yasser Ismail & Magdy A. Bayoumi, 2011).

8. Conclusion

A new *MDS* algorithm that utilizes an accurate and efficient dynamic stop search algorithm is proposed. Reduction in the complexity of *MDS* algorithm is achieved by employing an adaptive threshold to skip the unnecessary redundant internal *SAD* operations and also to skip the irrelevant blocks' operations in the search area. There is approximately 99% and 20% saving in *ME* time with negligible degradation in *PSNR* and bit-rate compared with the conventional *FS* and *DS*. This qualifies the proposed *MDS* algorithm to be applied to real time video applications such as video conferencing over wireless networks due to its superior speedup and performance.

9. References

[1] S. Goel, Y. Ismail, P. Devulapalli, J. McNeely, and M. Bayoumi, "An Efficient Data Reuse Motion Estimation Engine," IEEE Workshop on Signal Processing Systems (SIPS 06), Banff, Canada, Sept. 2006.

[2] S. Yang, L. Zhenyu, T. Ikenaga, and S. Goto, "Enhanced Strict Multilevel Successive Elimination Algorithm for Fast Motion Estimation," in IEEE International Symposium on Circuits and Systems, ISCAS'07., pp. 3659-3662, 2007.

[3] G.-L. Li and M.-J. Chen, "Fast Motion Estimation Algorithm by Finite-State Side Match for H.264 Video Coding Standard," in IEEE Asia Pacific Conference on Circuits and Systems, APCCAS'06. , pp. 414-417, 2006.

[4] R. Li, B. Zeng, and M. L. Liou, "A New Three-Step Search Algorithm For Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438-442, Aug. 1994.

[5] 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.

[6] Alexis M. Tourapis, Oscar C. Au, and Ming L. Liou, "Highly Efficient Predictive Zonal Algorithms for Fast Block-Matching Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 12, NO. 10, October 2002

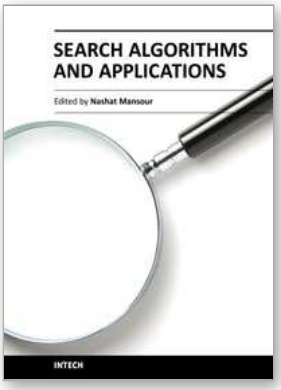
[7] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-Based Search Pattern For Fast Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol.12, pp. 349-355, May 2002.

[8] S. Zhu and K.-K. Ma, "A New Diamond Search Algorithm For Fast Block Matching Motion Estimation," IEEE Trans. Image Processing, vol. 9, pp. 287-290, Feb. 2000.

[9] C. H. Cheung and L. M. Po, "A Novel Cross-Diamond Search Algorithm For Fast Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 12, pp. 1168-1177, Dec. 2002.

[10] Joint Video Team, "Reference Software JM12.4,"
http://iphome.hhi.de/suehring/tml/download/".

[11] Yasser Ismail, Jason McNeely, Mohsen Shaaban, and Magdy A. Bayoumi, "A Generalized Fast Motion Estimation Algorithm Using External and Internal Stop Search Techniques for H.264 Video Coding Standard," IEEE International Symposium on Circuits and Systems (ISCAS), Seattle, Washington, USA, 18-21 May 2008.



Search Algorithms and Applications

Edited by Prof. Nashat Mansour

ISBN 978-953-307-156-5
Hard cover, 494 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Search algorithms aim to find solutions or objects with specified properties and constraints in a large solution search space or among a collection of objects. A solution can be a set of value assignments to variables that will satisfy the constraints or a sub-structure of a given discrete structure. In addition, there are search algorithms, mostly probabilistic, that are designed for the prospective quantum computer. This book demonstrates the wide applicability of search algorithms for the purpose of developing useful and practical solutions to problems that arise in a variety of problem domains. Although it is targeted to a wide group of readers: researchers, graduate students, and practitioners, it does not offer an exhaustive coverage of search algorithms and applications. The chapters are organized into three parts: Population-based and quantum search algorithms, Search algorithms for image and video processing, and Search algorithms for engineering applications.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yasser Ismail and Magdy A. Bayoumi (2011). Enhanced Efficient Diamond Search Algorithm for Fast Block Motion Estimation, Search Algorithms and Applications, Prof. Nashat Mansour (Ed.), ISBN: 978-953-307-156-5, InTech, Available from: http://www.intechopen.com/books/search-algorithms-and-applications/enhanced-efficient-diamond-search-algorithm-for-fast-block-motion-estimation

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821