International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 4, Issue 3, May-June 2015 ISSN 2278-6856

Block Based Motion Estimation Approach Using Triple Successive Selection Stages Pedestal on Descriptors

Bushra A. Sultan[1] , Nidaa F. Hassan[2]

[1]Baghdad University, College of Science, Dept. of Computer Science, Baghdad, Iraq

[2]Technology University , Dept. of Computer Science, Baghdad, Iraq

Abstract

triple successive selection stages of block matching approach based on block descriptors is introduced in this paper. Block descriptors serve the search and play the role of indexing. The searching scheme consist of three phases. Phase one include the calculations of block descriptor values. Phase two include triple successive selection stages, each stage use divergent block descriptor value to pick sub set of nominee seek points. Each selection stage has two filters. First filter based on the comparison between the descriptor value of the spots in the search region and the current block descriptor value. only the points that satisfy the selection condition passed that filter but the others obscured. Second filter regulate the number of points passed to the next selection stage. In phase three alteration quantity calculation performed to the most recent nominee seek spots to get the finest match and produce the movement information. For judgment purposes the well known Exhaustive Search (ES) and Three Step Search (TSS) algorithms are employed. The test results indicate that the approach matching block introduced quickly and efficiently, where its faster and more accurate than TSS as well as near the ES accuracy.

Keywords: Motion Estimation, Block Matching, Descriptor, Mean, Exhaustive Search, Three Step Search.

1. INTRODUCTION

Motion Estimation (ME**)** has been a hot research topic for years. It is the most important part in video compression and coding, it exploits as much temporal redundancies as possible to reduce the size of data required in digital video storage and transmission[1]. The most computationally expensive and resource hungry operation in the entire compression process is ME**.** Hence, researchers focused on it are still actively seeking on optimum tradeoff between speed and quality ,however, they are often conflicting goals.

Motion estimation and compensation examine the movement of objects in an image sequence try to obtain the vectors representing the estimated motion[2]. Its creates a model of the current frame based on available data in one or more previously encoded frames (reference frames). These reference frames may be 'past' frames (i.e. earlier than the current frame in temporal order) or 'future' frames (i.e., later in temporal order). The design goals for a motion estimation algorithm are to model the current frame as accurately as possible (since this gives better compression performance) whilst maintaining acceptable computational complexity[3].

Block based motion estimation methods are the most popular methods for motion estimation because of their simplicity and ease of implementation. The matching is seemingly used by the video compression standards because it can achieve a good balance between the complexity and coding efficiency[4].

The principle of Block-Based Motion estimation (BBME) in most of the video standards is that the video image frame is partitioned into blocks, and each block is an elementary unit. Motion estimation is performed by matching each block in the current frame against a region in a reference frame to find the best match[1]. Where the intermediate frame predicted from its nearest previous intra or predicted frame and motion compensation is guided that prediction leading to higher compression of these frames.

There are various type of BBME and also the proposed BBME in the last decades in video coding . Exhaustive Search (ES) is one of the well known BBME because it is the most accurate and straightforward approach of BBME method[5]**.** It exhaustively evaluate all possible blocks over the determined search window to find the best match, but this method has too high computational cost which makes slow process during encoding. The Three Step Search (TSS)[6] is a widely adopted simple and effective fast algorithm. However TSS uses uniformly allocated search pattern in the test step, which is not very efficient to cutch small motions appearing in stationary or qusi-station. To overcome this problem several adopted technique has been suggested to make the search more acceptable to motion scale and uncertainty. A New Three Step Search (NTSS) algorithm proposed in [7] in which the search pattern in each step is fixed and no thresholding operation are involved in this algorithm. A feature of NTSS are that it employs a center- biased checking point pattern in the first step, which is derived by making the search adaptive to motion vector distribution, and halfway-step technique to reduce the computation cost. The Four Step Search [8] and the Diamond Search [9], Hexagon based Search [10], and unsymmetrical cross multi hexagon grid search[11] algorithm has officially assumed to decrease calculation

density by using fairly few points with no corrupting image quality.

In this paper a new approach of block based motion estimation and compensation is adopted based on blocks' descriptors values entitled Descriptor Based Search **(**DBS**)**. This approach removes even less impossible candidate blocks with full search window, so, decreases heavy Sum of Square Differences (SSD) computations of the full search while providing an optimal motion vectors. The main concerns of the proposed search approach are:

- Each overlapped blocks in the frames is assigned three descriptors values, which are functions of block content.
- Reduce the number of block matching operation in the search window by replacing it with single value matching (matching the blocks' descriptors).
- The descriptors says if the blocks similar they must have approximately the same value, but blocks with the same descriptors values may not the same.

So, the proposed DBS approach specify the candidate blocks that similar in their descriptor values using single value matching to be examined later using block based matching operation.

2. THE PROPOSED SERACH SYSTEM

The structure of the proposed DBS system summarized in figure 1, where the system consists of following three phases :

- **Phase one "Descriptor's Calculation":** For all overlapped blocks in Y band of the two inputs frame compute the descriptor values (i.e. current frame Fn+1 and the referenced frame Fn)

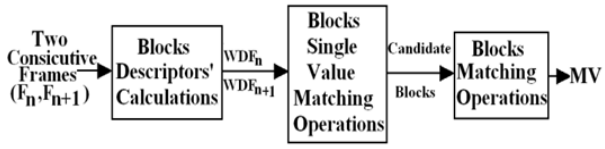


Figure 1 The structure of the proposed DBS

- **Phase two "Single Value Matching Operation":** This is the selection phase that promotes sub set of search points to be passed to the next phase (Block Matching Operation). The process is started by opening full window then triple successive selection stages had been applied where each stage use divergent blocks' descriptors value to pick sub set of nominee search points. So, each selection stage has two filters: First filter based on the comparison between the descriptor value of the points in the search area and the current block descriptor value. only the points that satisfy the selection condition passed that filter but the others obscured. Second filter regulate the number of points passed to the next selection stage and this number specified by passing condition .

- **Phase three "Blocks Matching Operation" :** For all candidate search points reached that stage compute block distortion measurement (i.e. SSD) to find the best match and produce the motion information.

  As a preprocessing stage the search area size (SAS) must be determine to construct the priority of the search point of that area (i.e. the center points assigned priority zero and the point around and adjacent to it is assigned priority one and so on the priorities assigned incrementally relative to the farness from center of the search area. Figure 2 give an example of the priority values assigned to each points in the search area when the SAS equal to 2 (i.e.[-2..2]).

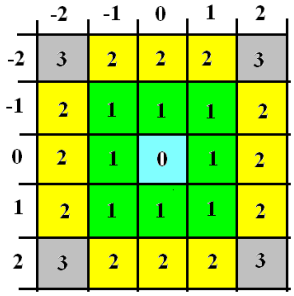


Figure 2 The priorities assigned to each point within the search area when (SAS=2)

Each phase of the proposed system is explained in more details in the following subsections:

2.1 Block Descriptors' Calculations:

Step1: Weights Calculation where weights defined as a magnitude of pixels within the block. One set of weight formed in lower-order weight to measure weighted means for each pixel within the block. Weights can be computed as:

$$W(i) = \begin{cases} \sqrt[3]{\left(\frac{L}{2} - i - \frac{1}{2}\right)} & for\ i = \left[0, \frac{L}{2}\right) \\ -W(L-1-i) & for\ i = \left[\frac{L}{2} - 1, L\right) \end{cases} \quad (1)$$

$$W1(i) = \lceil W(i) * 100 \rceil \quad for\ i = [0, L) \quad (2)$$

Where

L is the block length
W is the real part of the weights.
W1 is the integer part of the weights.

Table (1) lists the values of W and W1 when (L=8). The values of the weights shown in that table is symmetric and high weight values given to the pixels near the edge of the block while the pixels near the center of the block assigned the lowest weights value. This step is calculated once time after determining the block length.

Table 1: Weight values when (block length = 8)

L	0	1	2	3	4	5	6	7
W	1.16	1.12	1.05	0.91	-0.91	-1.05	1.12	1.16
W1	117	112	105	92	-92	-105	-112	-117

Step2: Descriptors' calculations, so in this step convert the two input frames from RGB color model to YUV color model then compute the block's descriptor values for each overlapped block of Y band using the following equations:

$$Mean(x,y) = \frac{1}{L^2}\sum_{r=x}^{x+L-1}\sum_{c=y}^{y+L-1} YB(r,c) \quad (3)$$

$$Mom\_X(x,y) = \sum_{r=x}^{x+L-1}\sum_{c=y}^{y+L-1} YB(r,c) * W1(r-x) \quad (4)$$

$$Mom\_Y(x,y) = \sum_{c=y}^{y+L-1}\sum_{r=x}^{x+L-1} YB(r,c) * W1(c-y) \quad (5)$$

Where :

L	is length of the block.
x & y	is top left hand corner coordinates of the block (i.e. the starting spot of the block).
YB	represents the Y band of the frame.
W1	is the integer part of the weights.
Mean	represents the mean (i.e. average) of the block.
Mom_X	& *Mom_Y* represent low order moments around x coordinate and around y coordinate respectively.

2.2 Blocks Single Value Matching Operations

In this phase the matching process performed between the block's descriptor values in the current frame and all blocks' descriptor values within the search area of the referenced frame to find the best match. The matching process included successive three stages of single value matching operation. Each stage used distinct descriptor value for matching where mean used by stage one, moment around x coordinate and moment around y coordinate used by stages two and three respectively. In each stage different steps of operation is performed in addition to the matching step like conditional selection and sorting. In other word the role of this phase is selection since it promotes sub set of candidate blocks to be passed to the next phase. The proposed steps are explained below where steps (1,2) belong to stage one, step3 belong to stage two finally step4 belong to stage three :

Step0: for all blocks in the current frame do the following steps:

Step1: compute the difference between mean value of the block in the current frame and all overlapped blocks within the search area in the reference frame starting from the least priority points using the following equation:

$$Dif_{(x,y)} = \lceil |\text{mean}_r(x+sx, y+sy) - \text{mean}_c(x,y)| \rceil \quad (6)$$

Where

x & y	is the center of the search area in the reference frame as well as top left corner coordinates of the block in the current frame
sx & sy	is the displacement from center of the search area
$Mean_r$	& $Mean_c$ represent the mean of the reference and the current frames respectively

all computed differences stored Buf_Dfr0, where it is a sequential array and each entrance of it holds four values (the difference ,Priority, X and Y are the coordinates of the upper left corner of the block). During differences calculation process the minimum and maximum difference determined the range of the acceptable difference and to flag the points with acceptable difference .

Step2: all differences stored in Buf_Dfr0 that are flagged as acceptable difference and had priority value less than or equal to $\beta\_1 \in \{6, \ldots, 12\}$ selected to be stored in another sequential array named Buf_Dfr1 Where each entrance that holds three values (the difference, X and Y are the coordinates of the upper left corner of the block). Sort later Buf_Dfr1 in climbing order according to the difference item and if the number of entries from Buf_Dfr1 rank greater than Z1 (i.e. passing condition1) then partially of them are passed to the successive selection stage. Otherwise, all passed.

Where $Z1 = Buf_{Len1} - \frac{Buf_{Len1}}{\sigma}$, Buf_{Len1} is the maximum number of entry in Buf_Dfr1 and $\sigma \in \{3,4,5\}$

Step3: for all passed blocks indexes (x1,y1) stored in Buf_Dfr1 compute their moment around x difference from the current block moment around x using:

$$Dif_{(x,y)} = \left\lceil \left| \frac{\text{Mom}_{X_r}(x1,y1) - \text{Mom}_{X_c}(x,y)}{|\text{Mom}_{X_r}(x1,y1)| + |\text{Mom}_{Xd_c}(x,y)| + \alpha} * N_{bin} \right| \right\rceil \quad 7)$$

Where

x & y	Is the center of the search area in the reference frame as well as top left corner coordinates of the block in the current frame
x1& y1	is the x and y coordinates of the search point stored in List_Dif1
Mom_X_r	& Mom_X_c represent the moment around x coordinate of the reference and the current frames respectively.
α	is the value of bias to avoid division by zero or very small value .
N_bin	is the number of holders to determine the value of the difference to be true in the range [0..N_bin].

all computed differences less than (N_bin- β_2) where $\beta\_2 \in \{5,6,7\}$ (i.e. satisfy the selection condition2) stored in new array called Buf_Dfr2 with the same entry definition of Buf_Dfr1. Sort later Buf_Dfr2 in climbing order according to the difference item and if the number of entries from Buf_Dfr2 rank greater than Z2 (i.e. passing condition2) then partially of them are passed to the successive selection stage. Otherwise, all passed.

Where $Z2 = Buf_{Len2} - \frac{Buf_{Len2}}{\sigma+1}$ and Buf_{Len2} is the maximum number of entry in Buf_Dfr2.

Step4: for all passed blocks indexes (x1,y1) stored in Buf_Dfr2 compute their moment around y coordinate difference from the current block moment around y coordinate using equation (7) but replace Mom_X term

with Mom_Y term. All computed differences less than (N_bin- β_2) (i.e. satisfy the selection condition2) stored in new array called Buf_Dfr3 with the same entry definition of Buf_Dfr1. Buf_Dfr3 sorted in ascending order according to the difference entry. The first M'th entries selected from Buf_Dfr3, where each pair of (x,y) represent the top left corner coordinate the candidate blocks that will be passed to the next phase.

2.3 Blocks Matching Operations:

Compute SSD for all candidates blocks M and the current block and then choose as little as one, but if the SSD calculated fewer than ¥ ∈ {0,1,2,3} leave out the remaining blocks and computes motion vector (MV).

3. TESTS RESULTS

The concert of the proposed searching scheme DBS were conducted to access using many sets of tests . Family and Flowers used as video test samples, (with frame size qualifications: size=320x240 pixels and size=352x240 pixels for Family and Flowers respectively, color depth=24 bit) , had been used. All methods tests done with (block length=8) and tests for DBS done when (α=10) and (N_bin=30).

The performance of the suggested DBS studied by taken into consideration Several parameters. Table (2) list all tests done for DBS with the first P-frame of two videos (Family and Flowers) when (β_1=8) and SAS [-7..7].

Table 2: List all tests done with the first P-frame of two videos (Family and Flowers) when (β_1=8) and SAS [-7..7]

			Family		Flowers	
σ	β_2	M	MAE	ST (Sec.)	MAE	ST (Sec.)
3	7	6	3.702	0.0203	9.3315	0.0218
		7	3.642	0.0199	9.1613	0.0223
		8	3.593	0.0208	9.0027	0.0234
		9	3.574	0.0203	8.8685	0.0237
		10	3.559	0.0214	8.7759	0.0239
		11	3.539	0.0221	8.7376	0.0242
		12	3.518	0.0220	8.682	0.0237
		13	3.496	0.0213	8.654	0.0247
		14	3.483	0.0225	8.584	0.0244
	6	6	3.711	0.0198	9.3686	0.0217
		7	3.646	0.0196	9.1467	0.0222
		8	3.594	0.0199	9.0035	0.0261
		9	3.574	0.0202	8.86	0.0239
		10	3.556	0.0207	8.7693	0.0231
		11	3.541	0.0239	8.7253	0.0235
		12	3.518	0.0211	8.6694	0.0241
		13	3.494	0.0214	8.639	0.0508
		14	3.482	0.0216	8.5692	0.0526
	5	6	3.714	0.0193	9.4126	0.0457
		7	3.643	0.0197	9.1474	0.0223
		8	3.596	0.0199	8.9964	0.0225
		9	3.573	0.0201	8.8684	0.0439
		10	3.554	0.0205	8.7719	0.0295
		11	3.539	0.0208	8.719	0.0235
		12	3.524	0.0210	8.6659	0.0238
		13	3.495	0.0213	8.64	0.0242
		14	3.484	0.0217	8.5749	0.0244
4	7	6	3.722	0.0196	9.4121	0.022
		7	3.661	0.0200	9.2337	0.0223
		8	3.607	0.0202	9.0787	0.0227
		9	3.590	0.0207	8.946	0.0231
		10	3.573	0.0203	8.8339	0.0233
		11	3.551	0.0203	8.79	0.0245
		12	3.531	0.0219	8.7055	0.0245
		13	3.511	0.0218	8.681	0.0248
		14	3.497	0.0211	8.5954	0.0245
	6	6	3.729	0.0196	9.4542	0.0211
		7	3.664	0.0188	9.2241	0.021
		8	3.610	0.0200	9.0845	0.022
		9	3.591	0.0212	8.9458	0.0229
		10	3.570	0.0210	8.8324	0.0228
		11	3.554	0.0445	8.7828	0.023
		12	3.530	0.0241	8.698	0.0225
		13	3.508	0.0218	8.6723	0.0238
		14	3.496	0.0221	8.587	0.0236
	5	6	3.731	0.0200	9.4863	0.0215
		7	3.661	0.0215	9.2177	0.0218
		8	3.611	0.0222	9.0702	0.0224
		9	3.590	0.0208	8.9378	0.0226
		10	3.568	0.0211	8.8219	0.0226
		11	3.552	0.0214	8.7632	0.0238
		12	3.535	0.0216	8.6781	0.0235
		13	3.508	0.0220	8.6529	0.0238
		14	3.497	0.0220	8.5709	0.0246
5	7	6	3.727	0.0202	9.5755	0.0223
		7	3.670	0.0195	9.3869	0.0227
		8	3.619	0.0203	9.1733	0.0234
		9	3.600	0.0206	9.0086	0.0235
		10	3.581	0.0206	8.9125	0.0233
		11	3.558	0.0214	8.8561	0.024
		12	3.533	0.0215	8.7599	0.024
		13	3.513	0.0213	8.7087	0.0241
		14	3.499	0.0217	8.6052	0.0243
	6	6	3.733	0.0191	9.5913	0.0221
		7	3.674	0.0203	9.3771	0.0229
		8	3.622	0.0210	9.1851	0.0224

		9	3.602	0.0210	9.0083	0.0233
		10	3.578	0.0212	8.9108	0.0231
		11	3.561	0.0356	8.8496	0.0236
		12	3.533	0.0334	8.753	0.0237
		13	3.511	0.0220	8.7277	0.0236
		14	3.498	0.0215	8.5979	0.0242
	5	6	3.735	0.0196	9.6304	0.0215
		7	3.670	0.0198	9.371	0.022
		8	3.623	0.0196	9.1738	0.0226
		9	3.600	0.0200	9.0028	0.0226
		10	3.575	0.0206	8.9034	0.0236
		11	3.558	0.0210	8.8344	0.0233
		12	3.537	0.0213	8.7351	0.0238
		13	3.509	0.0215	8.7086	0.0245
		14	3.4982	0.0218	8.5954	0.0246

Table (2) shows that some have an important impact parameter and the other did not. The belongings of the subsequent parameters are explored: (1) β_1, (2) β_2, (3) σ, (4) M and (5) SAS. Table (3) presents the accepted default values of these parameters, comprehensive tests were making before select the finest rate parameters.

Table 3: The values of the control parameters

Parameter	Ambit	Default
β_1	[6..12]	8
β_2	[5..7]	5
σ	[3..5]	4
M	[6..15]	12
SAS	[7..8]	

Table 4 shows a comparison between the mean absolute error (MAE) and search time (ST) in seconds. To test the application on the number of P-frames (1,2,3,4,5) video test using different search techniques (ES with SAS [-8,8] and [-7,7], TSS and DBS with SAS [-8,8] and [-7.7]). Results DBS is the average of all runs do with all parameters set pre-defined in Table (2) of the DBS (an average of best and worst cases) . Figure (3) summarize the results of tests comparing the performance of search techniques in the range of error, but the figure (4) to make comparisons in the period of time of the search, when the P-frame number is equal to one.

Table 4: A comparison between the error and the time to search through test applied to test video (P-frame No. ∈{1,2,3,4,5}) using (ES-7, ES-8, TSS, DBS-8, DBS-7).

		Family		Flowers	
Frame No.	Methods	MAE	ST (Sec.)	MAE	ST (Sec.)
1	TSS	3.823	0.0268	10.696	0.0344
	DBS-7	3.580	0.0215	8.926	0.0246
	DBS-8	3.581	0.0238	9.190	0.0269
	ES-7	3.283	0.1170	7.490	0.1327
	ES-8	3.170	0.1549	7.479	0.1858
2	TSS	4.093	0.0243	10.190	0.0330
	DBS-7	3.696	0.0198	8.179	0.0231
	DBS-8	3.661	0.0227	8.384	0.0263
	ES-7	3.460	0.1174	6.906	0.1302
	ES-8	3.285	0.1507	6.892	0.1649
3	TSS	4.603	0.0258	10.658	0.0321
	DBS-7	4.260	0.0203	8.680	0.0234
	DBS-8	4.198	0.0229	8.908	0.0264
	ES-7	3.890	0.1114	7.187	0.1313
	ES-8	3.678	0.1491	7.178	0.1643
4	TSS	2.017	0.0278	10.831	0.0325
	DBS-7	2.255	0.0209	8.707	0.0231
	DBS-8	2.296	0.0240	8.854	0.0274
	ES-7	2.013	0.1126	7.351	0.1317
	ES-8	2.011	0.1522	7.344	0.1932
5	TSS	4.011	0.0253	11.011	0.0323
	DBS-7	4.009	0.0208	9.650	0.0230
	DBS-8	3.921	0.0238	9.843	0.0264
	ES-7	3.717	0.1270	7.872	0.1297
	ES-8	3.566	0.1475	7.867	0.1634

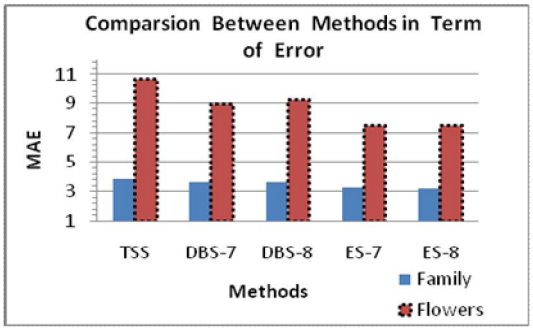


Figure 3 To compare the results of the tests Performance of search techniques in the error term

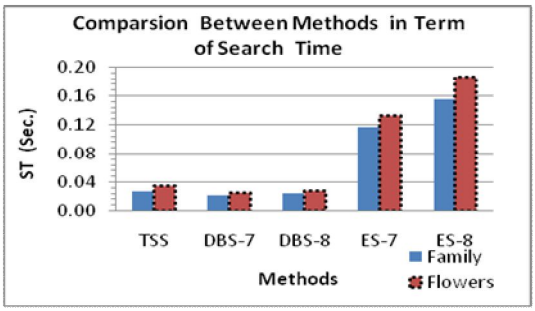


Figure 4 To compare the results of the tests Performance of search techniques in the error search time term

Figure (5) selects the shaded part from table (4) (i.e. P-frame No.1 of Family) to construct the judgment between performance of means in term of MAE vs. ST (Sec.)

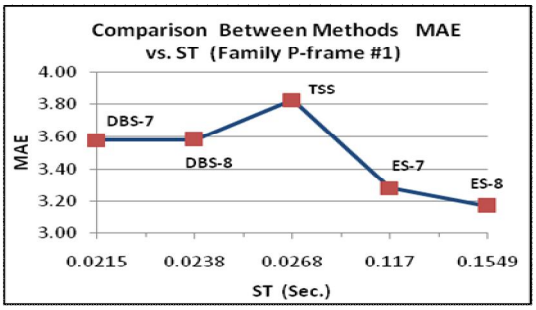
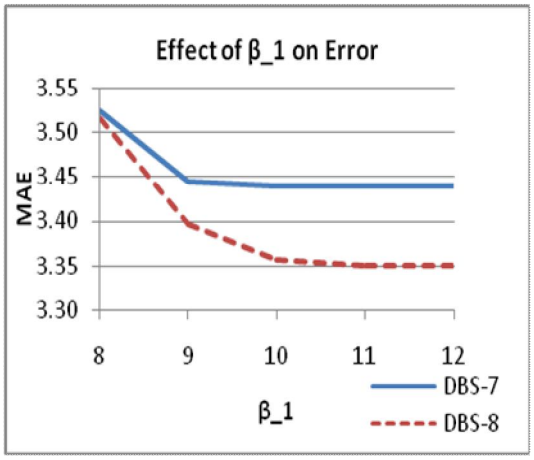
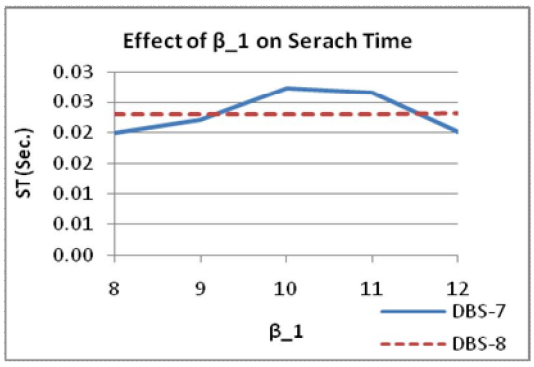


Figure 5 Testing results that evaluating the performance of diverse search ways in expression of MAE vs. ST for (P-frame No.1 of Family).

Figure (6.a) and (6.b) the effects of β_1 parameter on the error and the search appear in the time respectively. Whereas, the Figure (7.a) and (7.b) shows the effects of σ parameter on the error and the search time in a row. Figure (8.a) and (8.b) parameter effects β_2 appear on the error and search time in a row, Finally figure (9.a) and (9.b) the effects of M parameter appear in the error and the search time in a row. All of these standards apply to the family P- frame NO.1 video effect. It signed different effects when these criteria increase or decrease.

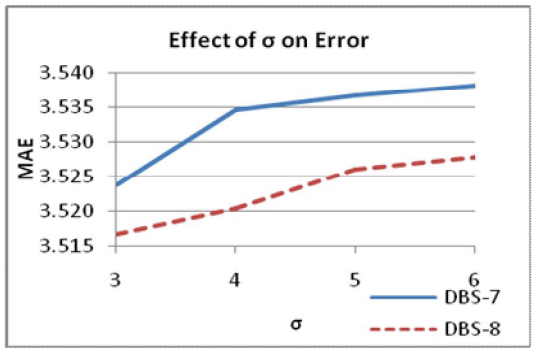


a.

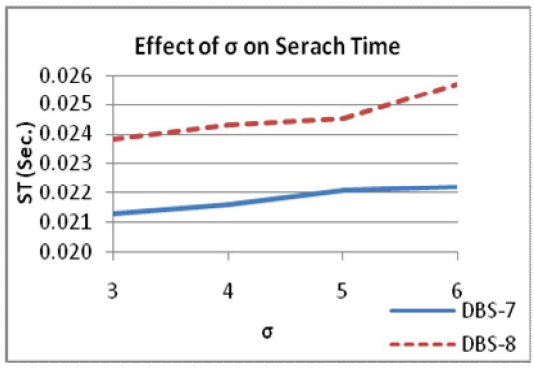


b.

Figure 6 The effect of β_1 (a. On error b. On search time**)**

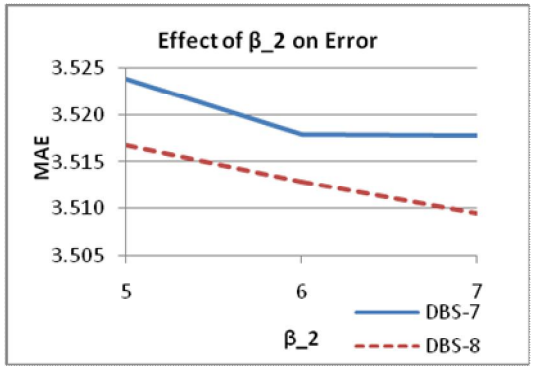


a.

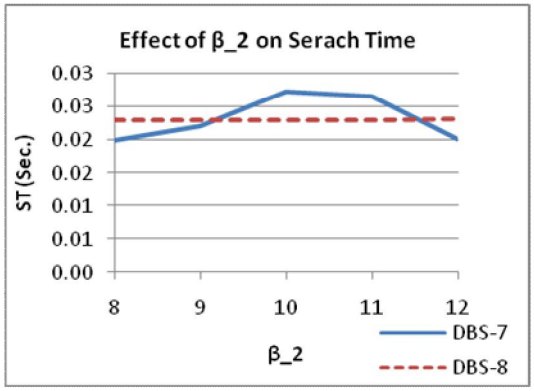


b.

Figure 7 The effect of σ (a. On error b. On search time**)**

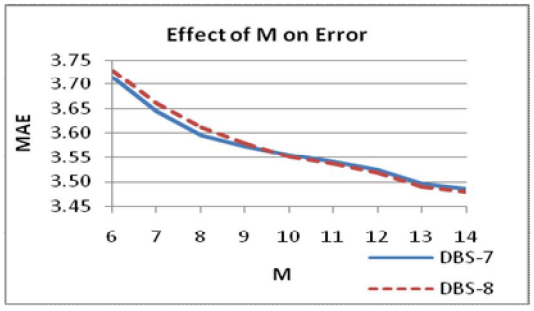


a.

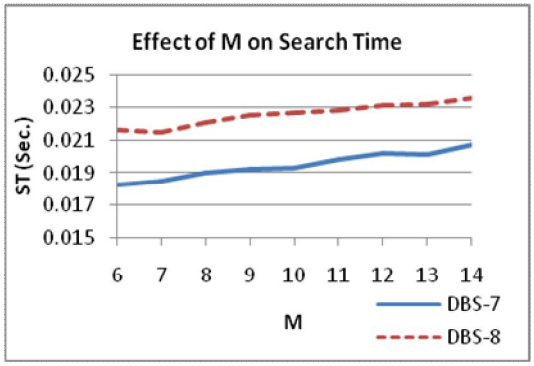


b.

Figure 8 The effect of β_2
(a. On error b. On search time**)**



a.



b.

Figure 9 The effect of M
(a. On error b. On search time**)**

4. CONCUSIONS

The proposed system of test results, are stimulated following observations:

- The DBS scheme is appropriate and efficient as well as it adds new concepts to the BBME since it facilitate the search using the descriptors. Block descriptors serve the search and play the role of indexing .
- The proposed DBS produce faster and more accurate results than TSS and near ES accuracy.
- For outlook, Different order of weight can use.
- The SAS could be minimize by adding predictive search strategy

References

[1] Chung, HY; Yung, NHC; Cheung, PYS,"fast motion estimation with search-center prediction", Optical Engineering, 2001, v. 40 n. 6, p. 952-963.

[2] Array Microsystem Inc.," An Introduction to Video Compression", Suit 6, 897 University Ave., Los Gatos, California 95030, UNITED STATES, White Paper, 1997.

[3] Richardson, I. E.; "The H.264 Advanced Video CompressionStandard", John Wiley& Sons, 2nd Edition, 2010.

[4] S. Sonongsathitanon, S. S. Dlay, " A New Orthogonal Logarithmic Search Algorithm for Fixed Block Based Motion Estimation for Video Coding", University of Newcalstil, report in 2001.

[5] Deepak J. Jayaswal, Mukesh A. Zaveri, " Probability Based Search Motion Estimation Algorithm", 2009 First International Conference on Computational Intelligience, Communication Systems and Networks, PP.357-362, 2009

[6] T. Koga, K. Linuma, A. Hirano, Y. Iijima, nd T Ishiguro, " Motion Compensated for Video Conference", in Proc. Nat. Telecommun. Conf., New Orleans LA, Nov. 29_Dec. # 1981, PP. G5.3.1-G5. 3.5.

[7] R. Li, B. Zeng and M. L. Liou, " A New Three-Step Search Algorithm for Block Motion Estimation",IEEE Transactions on Circuits and System for Video Technology. Vol.4, No. 4 , Augest 1994,PP.438-442

[8] L. M. Po and W. C. Ma, " A Novel Four- Step Search Algorithm for Fast Block Motion Estimation", IEEE Transactions on Circuits and System for Video Technology. Vol. 6 ,PP.438-442, June.1996.

[9] S. Zhu and K. K. Ma, " A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits and System for Video Technology. Vol. 9 ,PP.287-290, Feb, 2000.

[10] Zhu, X. Lin and L. Chau," Hexagon-Based Search Pattern for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits and System for Video Technology. ,PP.349-355, May, 2002.

[11] Z. Chen, Y. He and J. Xu, " Hybrid Unsymmetrical Cross Multihexagon-grid Search Strategy for Integer Pel Motion Estimation in H.264", in Proc. PCS,PP. 17-22, April, 2003.