Optimal DSP-Based Motion Estimation Tools Implementation For H.264/AVC Baseline Encoder

Imen Werda[2], Haithem Chaouch[2], Amine Samet[1], Mohamed Ali Ben Ayed[1], Nouri Masmoudi[2]

[1] University of Sfax, High Institute of Electronics and Communication, BP 868, 3018 Sfax, TUNISIA
[2] University of Sfax, National School of Engineering, BP W, 3038 Sfax, TUNISIA

Summary
A merging procedure joining search origin, search pattern and new variable block size motion estimation for H.264/AVC is proposed in this paper. Those approaches yield good tradeoffs between motion estimation distortion and number of computations since they invest and exploit the center-biased characteristics of the real world video sequences: A reliable predictor determines the search origin, localizing the search process. An efficient search pattern exploits structural constraints within the motion field. A new fast block size selection DSP-based algorithm allows simultaneous fidelity of the video quality and the reduction of the computational cost. Experimental results demonstrate the viability of the proposed algorithms in low bit rate video coding applications: Video conference. The proposed motion estimation algorithms provide substantially higher encoding speed as well as graceful computational degradation capabilities.

Key words:
H.264/AVC, Search center, Block matching algorithm, Pattern search, Variable block size, complexity, Video quality, PSNR SSIM.

1. Introduction

Several video coding standards [1] exploit the high correlation between successive frames when incorporating sophisticate motion estimation (ME) and motion compensation technologies to improve coding efficiency. However, the enormous computations of these techniques [2] prevent the H.264/AVC standard [3] from any real-time application possibility. Therefore, it would be greatly beneficial to optimize as much as possible motion estimation tools, which takes the main share of the computational time. Several contributions in the literature aiming the reduction of the computational motion estimation cost were adopted. The best solution from a video quality point of view is the full search algorithm that considers every possible detail. However, this approach is very computationally intensive.

Manuscript received May 5, 2007
Manuscript revised May 20, 2007

The encoder is not specified by the standard and thus there is quite a lot of flexibility in the encoder. The purpose of this survey is to develop new methods that are pertinent to the processor-based implementation of the motion estimation tools in the H.264/AVC encoder. Limited by the computing power and memory size of digital signal processor (DSP), new motion estimation proposed tools have to reduce its computational complexity when matching the DSP computing architecture. Proposed approaches, joining faithful predicted motion vector determination, flexible search pattern strategy and optimal variable block size (VBS) motion estimation, have a good potential in constructing a real time DSP-based encoder without sacrificing the compression efficiency.

The rest of the paper is organized as follows. Section 2 presents the new predicted motion vector approach. Section 3 gives a brief overview of several block matching algorithm (BMA) strategies and our proposed algorithm Line Diamond Pattern Search (LDPS). We describe a brief overview of several fast VBS motion estimation algorithms along with our proposed Fast Block Size selection Algorithm (FBSA) in section 4. Our comparative simulation results and analysis are presented in section 5. Finally, conclusion is proposed in section 6.

2. Potential predicted motion vector (PMV) determination

Motion in most videoconference image sequences involves a few blocks and lasts for a few frames. Therefore, spatially or temporally adjacent blocks often have similar motion vectors. Taking the advantage of the correlation among neighboring motion vectors, an interesting approach were proposed in [4] to determinate the search center before applying the search strategy: consider the median value between three motion vectors of spatially adjacent blocks MB0, MB1, and MB2 (see Figure 1). This algorithm can successfully detect motion

changes and effectively reduces the residual energy at the expense of increased algorithmic complexity.

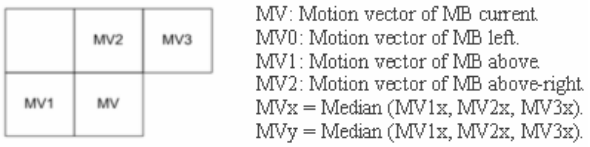

Fig 1. PMV determination approach.

Inspired from the previous approach, we propose a new software-based methodology which is more adapted to the DSP core architecture. In fact, conditional instructions used to extract the median in the previous approach, are not adapted to an optimal software implementation. When the CPU executes a conditional instruction, the pipeline must be "flushed". It must then start over with a new fetch and discard any instructions that have already been fetched. Minimizing the pipeline "flush" overhead is critical to any pipelined architecture since conditional instructions causes' pipeline discontinuity which increase motion estimation module cycles consumption. In the proposed approach, both spatially and temporally previously coded motion vectors representing neighbouring macroblocks are used to predict the motion vector. As illustrated is Figure 2, an additional motion vector of temporally adjacent blocks (collocated) MBc is considered as a forth candidate of the selection. Four SADs are computed to choose the best candidate for the search center. The candidate motion vector that yields to the minimum SAD is chosen to be the PMV. The SAD is calculated as sum of absolute differences between every alternate pixel in the corresponding blocks of reference frame and current frame:

$$SAD(x,y) = \sum_{i=0}^{15}\sum_{j=0}^{15} \left| Ycurr(i,j) - Yprev(x+i,y+j) \right| \qquad (1)$$

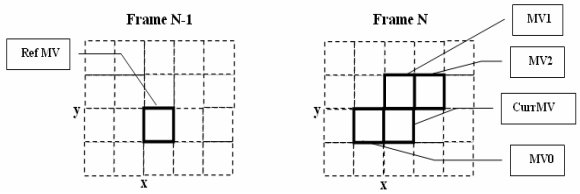

Fig 2. New PMV determination approach.

After the search center selection step, the search window can be constrained to a small clique surrounding this candidate position predicated based on the known neighboring motion vectors which reduce considerably the computational time. Analysis provided in [5] shows that the search window dimension can be restricted to (Horizontal: [-9, 9] and Vertical: [-7, 7]) assuming that the objects move in a translational manner in a video conference sequence.

The search window restriction, from (Horizontal: [-15, 15] and Vertical: [-15, 15]) to (Horizontal: [-9, 9] and Vertical: [-7, 7]), permits the highly localized search discussed next to originate from a more reliable location, as it distinguishes between motion and non-motion changes when deriving the predicted motion vector.

3. Block Matching Algorithms

Block matching algorithm (BMA) for motion estimation was been widely adopted by many video coding standard to reduce the temporal redundancy between different frames. Full search (FS) [6] evaluates exhaustively all the possible candidate motion vector within the search window to find the globally best matched block in the reference frame. However, its very computationally intensive nature prevents it from practical implementation on a processor for real-time applications since it is considered to be the bottleneck of video coding systems. Therefore, many gradient-based motion estimation algorithms, with carefully designed search patterns, have been developed to alleviate the computational load when limiting search points to a small subset of all possible candidates. This category includes the well-known three-step search (TSS) [7], the diamond search (DS) [8], the hexagon-based search (HEXBS) [9], the nearest neighbors search (NNS) [4] and the horizontal diamond search (HDS) [10]. Although this category of algorithms may be trapped into a local minimum point and hence the efficiency of the motion compensation may drop, they can considerably reduce the number of block matching computations.

3.1. Three Step Search

Three Step Search (TSS) [7] is one of the first non-full search algorithm. This search proceeds by moving the search area center to the best matching point in the previous step: The search starts with a step size equal to or slightly larger than half of the maximum search range. In each step, nine search points are compared. They consist of the central point of the square search and eight search points located on the search area boundaries as shown in Figure 3.

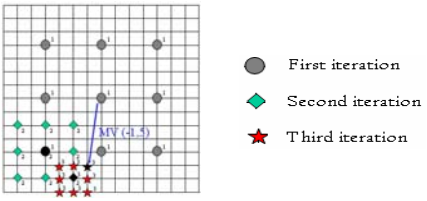

Fig 3. Three Step Search Strategy.

3.2. Diamond Search

The DS adopts two diamond-shaped search patterns [8], illustrated in Figure 4: large diamond search pattern (LDSP) with nine search points and small diamond search pattern (SDSP) with five search points.

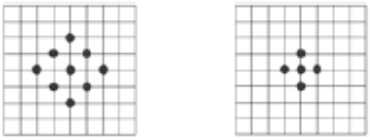

Fig 4. Diamond search patterns.

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 [8].

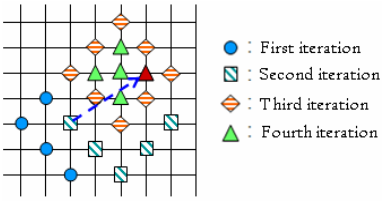

Fig 5. Diamond Search Strategy.

3.3. Hexagon-based search

HEXBS [9] uses two search patterns: large hexagonal search pattern (LHSP) and small hexagonal search pattern (SHSP) illustrated in Figure 6.

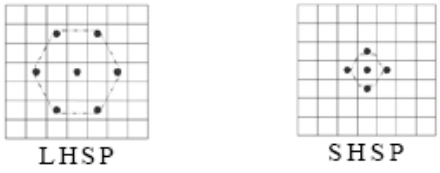

Fig 6. Hexagon-Based search patterns.

HEXBS pattern approximate the circle form better than the square or the diamond one which allow an optimal search space recovery saving computational energy with slightly decreased performance [9]. Figure 7 illustrates an example of HEXBS algorithm which adopts the same strategy as the DS algorithm.

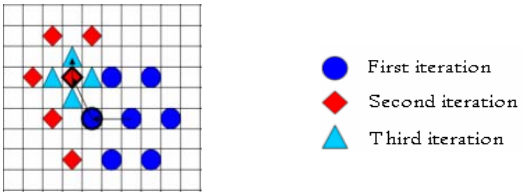

Fig 7. Hexagon-Based search Strategy

3.4. Nearest-Neighbors Search

The Nearest-Neighbors Search (NNS) algorithm [4] employs a novel motion vector prediction technique, a highly localized search pattern, and a computational constraint explicitly incorporated into the cost measure. As illustrated in Figure 8, the algorithm employs layers with different centers and containing at most four untested candidate motion vectors.

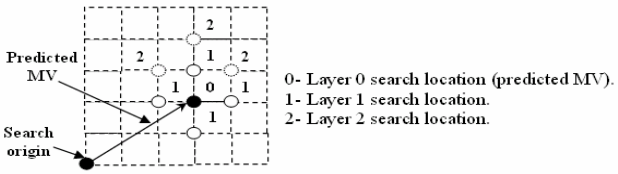

Fig 8. Nearest-Neighbors search Strategy.

3.5. Horizontal Diamond Search

The HDS [10] is a fast motion search algorithm with search pattern illustrated in Figure 9. The main advantage of this algorithm is to improve search on the horizontal motion component assuming that the objects move in a translational manner for at least few frames.

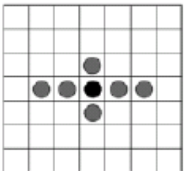

Fig 9. Horizontal diamond search pattern.

3.6. The Proposed Search Algorithm: Line Diamond Parallel Search (LDPS)

In order to design the optimal pattern and strategy search, two main proprieties were respected in the proposed approach:

- The distribution of the block motion fields is center biased so the smaller displacement is more probable and the motion vector (0, 0) has the highest probability of occurrence [11]. This is illustrated in Figure 10, which shows sample SAD surfaces for a given target macroblock and each candidate macroblock within a (Horizontal: [-15, 15] and Vertical: [-15, 15]) search window. Thus, a small pattern is used in a fist step to verify this propriety an avoid loosing a computational time in stationary blocks.
- The SAD value will decrease to the minima in some direction [12] so the BMA should use a second pattern with higher dynamic than the first one in order to quickly determine the convenient area.

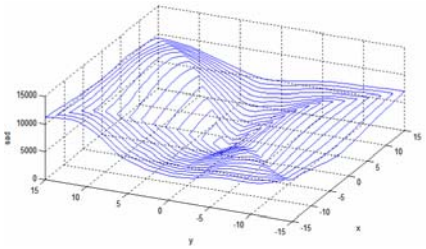

Fig 10. Sample Macroblock SAD distribution within a (Horizontal: [-15, 15] and Vertical: [-15, 15]) search window.

The proposed LDPS is a fast motion search algorithm with search pattern illustrated in Figure 11. The LDPS exploits the center-biased characteristics of the real world video sequences by using the SDSP pattern in the initial step. The second dynamic pattern improves search on the horizontal and vertical motion component as illustrated in Figure12.

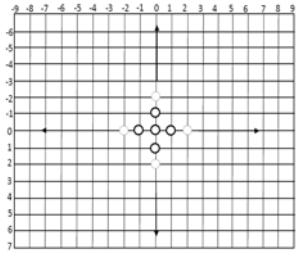

Fig 11. LDPS Search patterns.

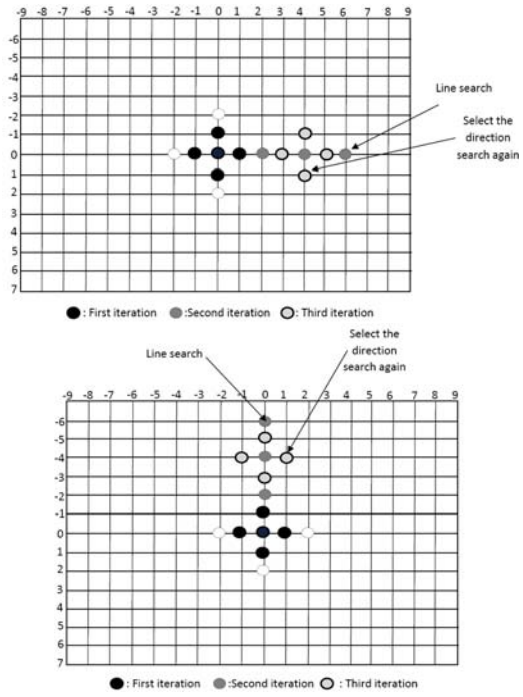

Fig 12. LDPS Search Strategy.

4. Mode Decision Algorithms

The computation burden is increased drastically for the H.264/AVC encoder because there are a number of combinations for partitioning a macroblock into sub-blocks ranging from 4x4 to 16x16. Potentially, each sub-block can have its own motion vector. This feature significantly increases the computational complexity in motion estimation module. With an optimal selection of the motion estimation tools the motion estimation algorithm could be designed to suit real time applications. Motion estimation up to 4x4 blocks is computationally complex task with insignificant coding efficiency improvement at low bit rate.

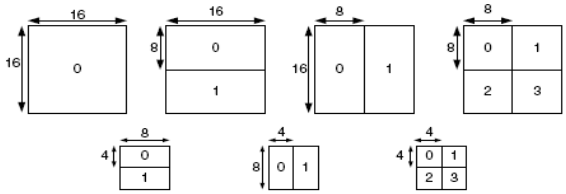

Fig 13. INTER modes with seven different block sizes ranging from 4x4 to 16x16.

Serious experiments on the test video sequences used in JVT Test Model Ad Hoc Group [13] show that there is an average of 35% homogeneous area in a typical video frame, and these areas are suitable for larger size inter mode coding. Therefore, several cost calculation of small size modes can be saved. Based on this consideration, several mode decision algorithms were proposed to reduce the number of candidate modes [14-16]. In [10], many Fast Variable Block-Size Motion Estimation algorithms were tested before extracting the Zoom Motion Estimation (ZME) algorithm [10] arranged as show in Figure 14:

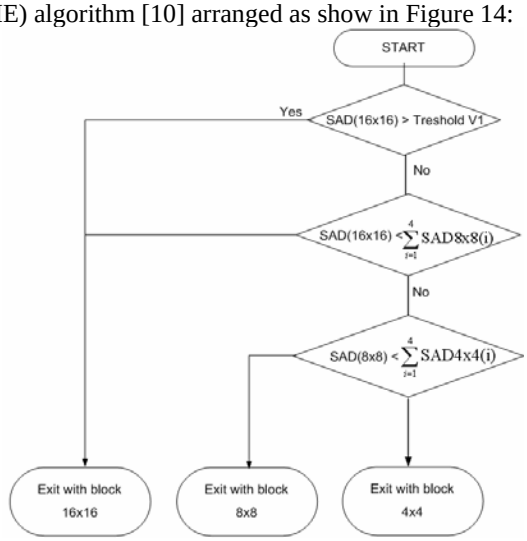

Fig 14. ZME algorithm.

For each position in the search window, we are computing and storing 16 4x4 SADs at the implementation phase. This allows reusing the SAD values for finding best match in sub-macroblock motion estimation at 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4 block levels. The best 16x16 block size corresponds to the minimum sum of 16 4x4 SADs. In a second step, when extracting the best 8x8 block size, there is no need to recompute SAD's for all iteration since they were previously computed and stored. We have just to sum respective 4 SAD's to extract the best 8x8 block size. This implementation method, reduce considerably computational complexity cycles at the expense of increased memory consumption: Mc= 16*32*W*H= 512 Kbytes of memory where W=32 and L=32 are respectively width and high of the search window.

The proposal of the new typical DSP-based VBS motion estimation algorithm is to extract, using the SAD criteria, the block-size introducing the minimum computational complexity with minimum memory requirement. In this algorithm, only 16x16, 16x8, 8x16, and 8x8 block-size for motion estimation are considered. This approach will reduce extensively the computational complexity since we are eliminating other 3 modes, without affecting the overall visual video quality. For better understanding, the following organigram explains in detail our proposed algorithm steps:

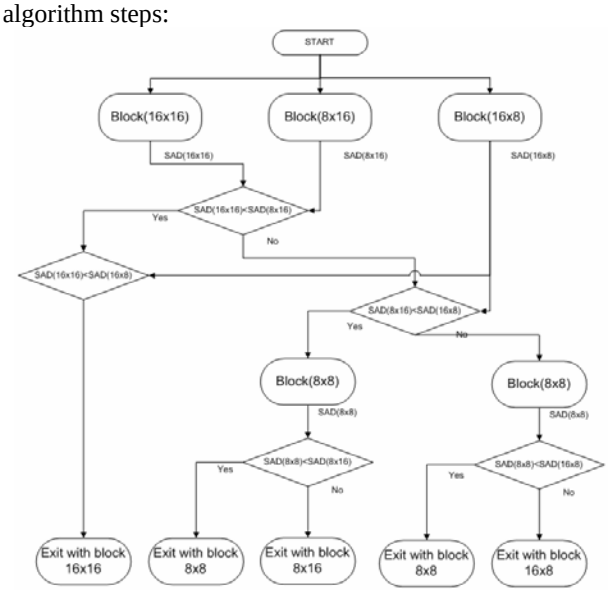

Fig 15. Fast block size selection algorithm FBSA organigram.

When implementing the following approach, for each iteration, only four 8x8 SAD are computed and stored in the internal memory: only 128 Kbytes of internal memory, against 512 Kbytes, is used to store SADs. In addition, only 4 SADs are computed in each iteration, against 16, which reduces computational complexity in motion estimation modules.

The best SAD of 16x16 block is computed as sum of SADs of each of its 4 8x8 block, the best 16x8 one is the minimum sum of 2 respective 8x8 SADs and the best 8x16 block size is the minimum sum of 2 respective 8x8 SADs. To extract the minimum 8x8 block size, we need just to find out the minimum 8x8 SADs. In summary, the fast mode decisions algorithms can be combined with inter prediction algorithms to achieve further speedup. The recent trend to further reduce the motion estimation calculations is to combine the techniques mentioned before which are: min SAD (MV0, MV1, MV2, Ref MV), LDPS search approach and the FBSA algorithm.

5. Simulation results and analysis

New motion estimation tools implementation, discussed in the present paper has been developed on the TMS320C6416 DSP of Texas Instruments [17] based on LETI encoder [18].

To evaluate the new motion estimation tools impact on the encoding process, several analysis on three CIF (352x288) sequences test with different characteristics (Figure 16), with different Qp values are proposed in this section.

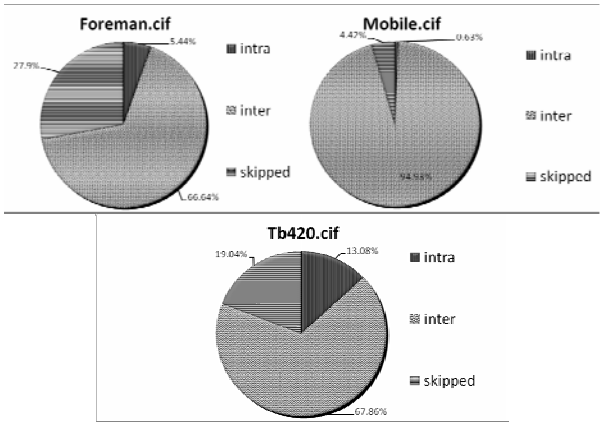

Fig 16. Percentages of ME (inter) intra and skip used in Foreman, Mobile and Tb420 CIF sequences.

We first present the experimental results of the new approach used to determinate the predicted motion vector compared to the referenced one using the spiral search (SS) algorithm as a search strategy. We then evaluate the fast block size selection algorithm (FBSA) improvement when compared to the ZME approach. Finally, we evaluate the performance of the new LDPS search strategy compared to DS, HEXBS, NNS, and HDS one, when

combined to “min SAD (MV0, MV1, MV2, Ref MV)” and FBSA algorithms.

5.1. Experimental results for the proposed PMV approaches

In this sub section, we will compare the MEDIAN (MV0, MV1, MV2) search center performance to the proposed min SAD (MV0, MV1, MV2, Ref MV), with QP =38 and search window (Horizontal: [-15, 15] and Vertical: [-15, 15]). Table 1 and Table 2 shows that the new predicted motion vector provides a complexity reduction up to 1.7%, with no video quality degradation.

5.2. Experimental results for block size prediction

In this sub section we will compare the ZME block mode selection algorithm performance to the proposed FBSA with QP =38 and search window (Horizontal: [-15, 15] and Vertical: [-15, 15]). In addition to the 25% memory saving, Tables 3 and 4 shows that the elimination of 4x8,8x4, and 4x4 block size from the mode selection dose not affects the video quality (PSNR and SSIM [19]). The FBSA algorithm provides better subjective and objective quality compared to the ZME approach with complexity reduction up to 44 % (see Table 5).

5.3. Experimental results for search pattern

Table 6 lists the video quality performance of NNS, DS, HDS, HEXBS, TSS, and LDPS algorithms in terms of PSNR in two search window dimensions when using the min SAD (MV0, MV1, MV2, MVc) approach to predict the search center and block size selection ranging from 16x16 to 4x4. It can be noted that the search window reduction does not affect the video quality. In addition the LDPS algorithm provides the best objective quality compared to others implemented approach. This quality performance is evaluated, as shown in table Table 7, in terms of speed performance. In fact, the speed performance of the LDPS algorithm is very interesting compared to the other motion estimation algorithm at the exception of the NNS where a 4 % loss of in speed performance is obtained.

5.4. Experimental results when combining the three approaches

When combining our proposed algorithms, we can explain the lost in terms of complexity for the LDPS compared to the NNS algorithm shown in Table 7. In fact, when combined to the 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4 mode selection, the code size of the merged ME module exceeds the cache memory size causing a cache miss overhead that affects the speed performance. When merged with the new approach, the code size of the ME module is reduced which alleviate the memory loads and show the right speed performance of the LDPS algorithm. As shown in Table 8, Table 9 and Table 10, the merge of new DSP-based motion estimation tools approaches have been able to eliminate memory loads and reduce the total number of motion estimation module cycles by 10 % without any loss in video quality.

6. Conclusions

In this paper, we proposed DSP based specific methods to decrease the motion estimation module complexity in the H.264/AVC Baseline encoder, enabling interactive real-time video applications for CIF resolution on a single TMS 320C6416 DSP. The implementation of combined new PMV method, LDPS search pattern, and the FSBA decomposition strategy, reduced the motion estimation module cycles consumption by 10 % and 90 % when compared respectively to NNS and SS algorithms, without any loss in video quality. The results illustrated in this paper lead us to conclude that if we use an accurate PMV with an appropriate LDPS search pattern combined with an optimal VBS algorithm, a better image quality with less important computing time will be obtained. As perspective, CPU-specific optimizations of the SAD module exploiting the architectural features of the TMS320C6416 will be conducted to emphasize the superiority of our proposed algorithm for different video sequences.

References

[1] Joch, F. Kossentini, H. Schwarz, T.Wiegand, and G. J. Sullivan, "Performance comparison of video coding standards using lagrangian coder control," *in Proc. IEEE Int. Conf. Image Processing, pp. 501–504, 2002.*

[2] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC Baseline Profile Decoder Complexity Analysis", *IEEE Trans. Circuits Syst. Video Technol, pp. 704-716, 2003.*

[3] Thomas W., "Study of Final Committee Draft of Joint Video Specification", *ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, Draft 1, 2002.*

[4] M. Gallant, G. Côté, F. Kossentini, "An Efficient Computation-Constrained Block-Based Motion Estimation Algorithm for Low Bit Rate Video Coding", *IEEE Trans. Image Processing,*vol*. 8, no. 12, 1999.*

[5] H.Chaouch, I.Werda, A.Samet and N. Massmoudi , "Search window impact on H.264/AVC motion search implementation on TMS320C6416 DSP" *IEEE Conference on Communication & Signal Processing SSD Vol 3: March pp19-22, 2007*

[6] I Richardson, "Full Search Motion Estimation," *in Video CODEC Design, pp. 99-101, 2002.*

[7] 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. 438–442,1994.*

[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, 2000.*
[9] C. Zhu, X. Lin, and L.P. Chau, "Hexagon-based search pattern for fast block motion estimation", *IEEE Trans. Circuit and Systems for Video Technology, vol. 12, pp. 349–355,2002.*
[10] M. A. Ben Ayed, A. Samet, N. Masmoudi, "Toward an Optimal Block Motion Estimation Algorithm for H.264/AVC", *International Journal of Image and Graphics (IJIG), 2006.*
[11] J.Y. Tham, S. Ranganath, M. Ranganath, and A.A.Kassim" A novel unrestricted center-biased diamond search algorithm for block motion estimation" *IEEE Trans. Circuit and Systems for Video Technology, vol. 8, pp. 369–377, 1998.*
[12] M. E. Al-Mualla, C. N. Canagarajah, and D.R. Bull, "Simplex minimization for single and multiple-reference motion estimation," *IEEE Trans. Circuit and Systems for Video Technology, vol. 11, pp. 1220–2001, 2001.*
[13] Y. Jiang, S. Li, S. Goto, "A Low Complexity Variable Block Size Motion Estimation Algorithm for Video Telephony Communication", *47th IEEE International Midwest Symposium on Circuits and Systems, pp. 465 -468, 2004.*
[14] Y. K. Tu, J. F. Yang, and M. T. Sun, "Fast Variable-size Block Motion Estimation Using Merging Procedure with an Adaptive Threshold", *IEEE International Conference on Multimedia and Expo, Baltimore, pp.789-792, 2003.*
[15] Z. Zhou, M. T. Sun, and Y. F. Hsu, "Fast variable block-size motion estimation algorithms based on merge and split procedures for H.264/MPEG-4 AVC", *IEEE International Symposium on Circuits and Systems, ISCAS, pp. 23-26, 2004.*
[16] Y. Jiang, S. Li, S. Goto, "A Low Complexity Variable Block Size Motion Estimation Algorithm for Video Telephony Communication", *IEEE International Midwest Symposium on Circuits and Systems, July 2004, pp.465-468.*
[17] Texas Instruments, "TMS32064xx Video/Imaging Fixed-Point DigitalSignal Processor", *Literature Number: SPRS200E July 2002 − Revised March 2004.*
[18] Circuit and System Group,LETI Laboratory, http://www.csgroup.tunet.tn.
[19] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity", *IEEE Transactions on Image Processing, vol. 13, no.4,2004.*

Imen WERDA was born in Sfax, Tunisia, in 1978. She received his degree in Electrical Engineering and his DEA in Electronic Engineering from Sfax National School of Engineering (ENIS), Tunisia, in 2003 and 2004 respectively.
In 2004, she joined the Ubvideo Tunisia Inc. as an R&D engineer.
In 2006, she joined the High Institute of technology of Sousse, Tunisia, as an Assistant Technologue.
She is a member of the Sfax Laboratory of Electronics and Information Technology. Her current research interests include DSP and hardware implementation of H.264 video coding standard for video conference applications.

Haithem CHAOUCH was born in Riyadh, Arabia Saudi, in 1982. He received his degree in Electrical Engineering, from Sfax National School of Engineering (ENIS), Tunisia, in 2006. Currently, he is working toward the Master, in Electronic Engineering at the ENIS. He is a member of the Sfax Laboratory of Electronics and Information Technology. His research interests include digital signal processing, image and video coding with emphasis on H.263 and H.264 standards and motion estimation algorithms.

Amine SAMET was born in Tunis, Tunisia, in 1977. He received his degree of Diplôme d'Ingénieur in Electrical Engineering, his DEA in Electronic Engineering, and his Ph. D. in Electronic Engineering at Sfax National School of Engineering (ENIS), Tunisia, in 2001, 2002, and 2006 respectively.
In 2002, he joined the High Institute of Electronics and Communication of Sfax, Tunisia, as an Assistant.
He is a member of the Sfax Laboratory of Electronics and Information Technology. His research interests include digital signal processing, image and video coding with emphasis on H.263 and H.264 standards and motion estimation algorithms.

Mohamed Ali BEN AYED was born in Sfax, Tunisia, in 1966. He received his B.S. degree in computer engineering from Oregon State University in 1988, his M.S. degree in electrical engineering from Georgia Institute of Technology in 1990, his DEA degree and Ph. D. in electronic engineering from Sfax National School of Engineering, Tunisia, in 1998 and 2004 respectively.
He was a co-fonder ob Ubvideo Tunisia, a wholly owned subsidiary of Ubvideo Inc. in 2003.
He is currently an Assistant Professor at High Institute of Electronics and Communication of Sfax.
His current research interests include DSP and hardware implementation of digital algorithms for multimedia services.

Nouri MASMOUDI was born in Sfax, Tunisia, in 1955. He received the electrical engineering degree from the Faculté des Sciences et Techniques de Sfax, Tunisia, in 1982, the DEA degree from the Institut National des Sciences Appliquées de Lyon and Université Claude Bernard de Lyon, France in 1982. From 1986 to 1990, he prepared his thesis at the laboratory of Power Electronics (LEP) at the Ecole Nationale d'Ingénieurs de Sfax (ENIS). He then received the 'Thèse de 3 ème Cycle' at the Ecole Nationale d'Ingénieurs de Tunis (ENIT), Tunisia in 1990.
From 1990 to 2000, he was an assistant professor at the electrical engineering department at the ENIS. Since 2000, he

has been an Associate Professor and head of the group 'Circuits and Systems' of the Sfax Laboratory of Electronics and Information Technology. Currently, he is responsible for the Electronic Master Program at ENIS.

His research activities have been devoted to several topics: power system, embedded systems, DSP and FPGA implementation

Table1: Objective quality performance (PSNR) of PMV approach's.

Sequences	Search center approach	
	MIN SAD (MV0, MV1, MV2, Ref MV)	MEDIAN (MV0, MV1, MV2)
Foreman	33.10	**33.14**
Mobile	**30.60**	**30.60**
Tb420	**31.83**	31.82

Table 2: Speed performance (Millions cycles) of PMV approach's.

Sequences	Search center approach		
	MIN SAD (MV0, MV1, MV2, Ref MV)	MEDIAN (MV0, MV1, MV2)	Improvement %
Foreman	**533.86**	543.23	1.75
Mobile	**512.83**	514.53	0.3
Tb420	**517.46**	522.35	1

Table 3: Objective quality performance (PSNR) of block mode selection algorithms.

Sequences	Block mode selection algorithms		
	All modes	ZME	FBSA
Foreman	30.73	30.67	30.72
Mobile	26.81	26.72	26.79
Tb420	29.50	29.47	29.49

Table 4: Subjective quality performance (SSIM) of block mode selection algorithms.

Sequences	Block mode selection algorithms		
	All modes	ZME	FBSA
Foreman	0.9883	0.9882	0.9883
Mobile	0.9824	0.9821	0.9824
Tb420	0.9853	0.9852	0.9853

Table 5: Speed performance (Millions cycles) of block mode selection algorithms.

Sequence	Block mode selection algorithms (Million cycles)				
	All modes	ZME	FBSA	Improvement %	
				FBSA / All modes	FBSA / ZME
Foreman	533.86	464.11	258.13	**51.64**	**44.30**
Mobile	512.83	465.92	251.09	**51.03**	**44.40**
Tb420	517.46	462.88	259.02	**49.94**	**45.70**

Table 6: Objective quality performance (PSNR) of block matching algorithms in two search window dimensions.

Algorithms	Foreman		Mobile		Tb420	
	x: [-15, 15] y: [-15, 15]	x: [-9, 9] y: [-7, 7]	x: [-15, 15] y: [-15, 15]	x: [-9, 9] y: [-7, 7]	x: [-15, 15] y: [-15, 15]	x: [-9, 9] y: [-7, 7]
TSS	33.37	33.37	30.80	30.80	32.37	32.37
DS	33.37	33.37	30.79	30.79	32.37	32.37
HS	33.41	33.41	30.79	30.79	32.37	32.37
HDS	33.44	33.44	30.78	30.78	32.37	32.37
LDPS	33.55	33.55	30.81	30.81	32.41	32.41
NNS	33.46	33.49	30.80	30.80	32.40	32.40

Table 7: Speed performance (Millions cycles) of block matching algorithms in two search window dimensions.

Algorithms	Foreman		Mobile		Tb420	
	x: [-15, 15] y: [-15, 15]	x: [-9, 9] y: [-7, 7]	x: [-15, 15] y: [-15, 15]	x: [-9, 9] y: [-7, 7]	x: [-15, 15] y: [-15, 15]	x: [-9, 9] y: [-7, 7]
TSS	39.88	37.29	37.81	36.35	39.08	36.80
DS	35.34	31.83	33.39	30.16	36.02	32.60
HS	33.54	33.23	31.22	30.83	35.84	33.71
HDS	32.54	32.77	30.07	29.88	33.88	33.66
LDPS	31.38	30.72	29.82	29.31	31.64	30.86
NNS	31.51	29.39	30.06	28.49	31.68	30.05

Table 8: Objective (PSNR) quality performance of block matching algorithms.

Sequences	LDPS	NNS	TSS	DS	HS	HDS
Foreman	33.95	33.95	33.95	33.95	33.95	33.95
Mobile	30.97	30.97	30.97	30.97	30.97	30.97
Paris	32.04	32.04	32.04	32.04	32.04	32.04

Table 9: Subjective (SSIM) quality performance of block matching algorithms.

Sequences	LDPS	NNS	TSS	DS	HS	HDS
Foreman	0.9953	0.9953	0.9953	0.9953	0.9953	0.9953
Mobile	0.9932	0.9932	0.9932	0.9932	0.9932	0.9932
Paris	0.9929	0.9929	0.9929	0.9929	0.9929	0.9929

Table 10: Speed performance (Millions cycles) of block matching algorithms.

Sequences	Block matching algorithms (Million cycles)							Improvement %	
	LDPS	NNS	TSS	DS	HS	HDS	SS	LDPS / NNS	LDPS / SS
Foreman	22.55	25.05	26.01	25.83	25.69	23.34	222.25	**11.17**	**89.85**
Mobile	22.97	25.22	25.71	26.52	25.50	29.88	235.19	**8.92**	**90.23**
Tb420	22.27	24.53	26.04	25.14	25.16	24.13	236.55	**9.09**	**90.58**