Hardware-Friendly Motion Estimation Algorithms and its Architectures for High Definition Videos

Mário Saldanha, Gabriel Balota, Gustavo Sanchez, Luciano Agostini, Marcelo Porto

Group of Architectures and Integrated Circuits - GACI, Federal University of Pelotas - UFPEL
Pelotas, Brazil

{mrdfsaldanha,gmbalota,gfsanchez,agostini,porto}@inf.ufpel.edu.br

***Abstract*— This paper presents details about three hardware-friendly motion estimation algorithms focused on high quality to high definition videos. The Dynamic Multi-Point Diamond Search (DMPDS), Spread and Iterative Search (S&IS) and Low Density and Iterative Search (LD&IS) are fast motion estimation algorithms focused on hardware implementation. These algorithms were evaluated for ten high definition sequences, and the results shows that DMPDS algorithm reaches the best trade-off between video quality and computational complexity. Three hardware architectures were developed and the synthesis results are also presented in this paper. These architectures performance are focused on real time processing for high definition videos with low hardware resources utilization. These architectures were developed in VHDL e synthesized to Altera Stratix 4 FPGA.**

***Keywords*— Motion Estimation, Hardware Architecture, High Definition Videos**

I. Introduction

With the increase of devices that support high definition video (HD), the research in developing more efficient algorithms for higher compression ratios, without reducing a high quality video has increased. For transmission or storing these videos, keeping a good quality video and a low bandwidth, the development the new algorithms is very important. The objective of encoding digital videos is to explore redundancies presents in a video (temporal, spatial and entropic). The temporal redundancy, that is the focus of this paper, is explored by the motion estimation (ME).

The ME is the current encoder tool that requires the highest computational effort, representing more than 80% the total encoder complexity. However, the ME is the main responsible for the high compression ratios achieved by current coding standards [1].

The ME has the objective to find the best block matching for each block in the current frame according to a reference frame. When the best block match is found, the ME defines a motion vector pointing to it.

The H.264/AVC video coding standard [2], and also the emerging High Efficiency Video Coding (HEVC) standard [3], does not restricts how is the ME is done. Based on this fact, there is a vast space to explore new algorithmic solutions able to achieve a high trade-off among quality video, compression rate and computational cost. Besides, the idea is that these algorithms ME can be implemented in hardware, because with current technology, software solutions could not achieve the performance and energy requirements, when working on real time application for high definition videos, and especially for mobile devices. So, hardware friendly algorithms are really desired.

This paper presents details about three fast hardware friendly ME algorithms. These algorithms were focused on high digital video quality for high definition videos. These algorithms are called: Dynamic Multi-Point Diamond Search (DMPDS) [4], Spread and Iterative Search (S&IS) [5] and Low Density and Iterative Search (LD&IS) [6].

This paper is organized as follows: Section 2 presents the ME algorithms evaluated. In section 3 it is presented software results of these algorithms. In Section 4 three architectures for ME algorithms are presented and their synthesis results. Finally the section 5 presents the conclusions.

II. Hardware Friendly Algorithms for ME

This section presents the three evaluated fast hardware-friendly motion estimation algorithms. The well known Diamond Search algorithm is also discussed since it is used as search core for the DMPDS, S&IS and LD&IS.

A. *Diamond Search Algorithm*

The DS algorithm can significantly reduce the computational complexity of ME, with low quality losses for low resolution videos, when compared to Full Search [7]. The DS uses a diamond pattern to perform its search. In this algorithm it is used two standards in diamond pattern to perform the search, the Large Diamond Search Pattern (LDSP) and Small Diamond Search Pattern (SDSP). First is applied the LDSP in the center of the search area where nine blocks candidates are compared, if the block with similarity is found in the central position, then it is applied the SDSP refinement, that evaluate four candidate blocks around in the central position central, and also evaluates the central block. When the refinement ends, a motion vector is generated.

If the LDSP does not find the best result in the center, the LDSP is applied again, now with central position where the block with more similarity was found. The central position can belongs an edge or a vertex of diamond, when a search is conducted for edge three new blocks candidates around of the central position are compared, when a search is conducted for vertex five new blocks candidates around of the central position are compared. It is not possible to determine how many iterations are performed in DS, the number the iterations vary according with the video characteristics.

With video resolution increasing, the DS algorithm loses efficiency when compared to FS, as shown in [4]. It happens because traditional fast algorithm often chose local minima.

B. Dynamic Multi-Point Diamond Search

The Dynamic Multi-Point Diamond Search (DMPDS) algorithm is strongly based in algorithm Multi-Point Diamond Search (MPDS) [8]. Fig. 1 presents the search mechanism of MPDS. The MPDS algorithm applies the DS algorithm in the central position of the search area, as the original DS. However, the search area is divided in four sectors, and another DS is applied for each one of these sectors. The MPDS defines a *d* parameter that determines the distance, in pixels, where each DS should be applied from the center of the search area. The final result of MPDS will the best result found among the comparison about the five DS. For low motion activity videos, the algorithm should achieve best results with small values for the *d* parameter, since candidate blocks with more similarity should be found near to the center of the search area. For videos with high motion activity, the DMPDS algorithm should achieve its best results with larger values of *d*. When combining these two factors the DMPDS algorithm can achieve high quality for any kind of video.

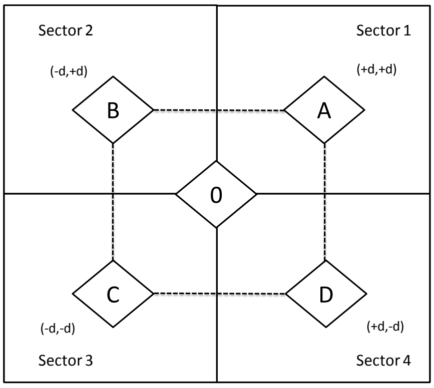

Fig. 1. Cores algorithm MPDS

In the DMPDS algorithm the value of the parameter *d* is controlled dynamically, according with the video characteristic. So, the d parameter can be reduced, for scenes with little motion activity, or increased, for scenes with higher motion.

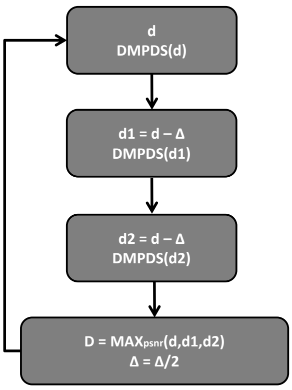

Fig. 2. Behavior of the *d* parameter in DMPDS algorithm

The DMPDS algorithm works with the *d* parameter and Δ parameter. Initially, a value for *d* and Δ are defined. The first frame is processed with the initial value of the *d*, the second frame with $d1 = d - \Delta$ and the third with $d2 = d + \Delta$. The *d* parameter value used for the frame with the best result, becomes the new *d* and the value of the Δ will be updated for $\Delta = \Delta / 2$. The process is repeated until the value of Δ is equal one. When the value of Δ becomes one the algorithm restarts, as shown in Fig. 2.

C. Spread and Iterative Search

The Spread and Iterative Search algorithm (S&IS) divides the search area into four sectors. A pseudo-random generator is used to generate *N* spread positions inside the search area. Each one of the chosen spread position is evaluated, then the position with the best results is selected. An evaluation using the DS in the central co-located block is also done to obtain better results in low motion activity videos, when blocks with higher similarity are found near to the center of the search area. After finding the best candidate block in the random step, it is compared with the best candidate block found in the central evaluation step, and the best blocks selected, and he final motion vector is generated for this block. Fig. 3 shows a flowchart of the S&IS algorithm.

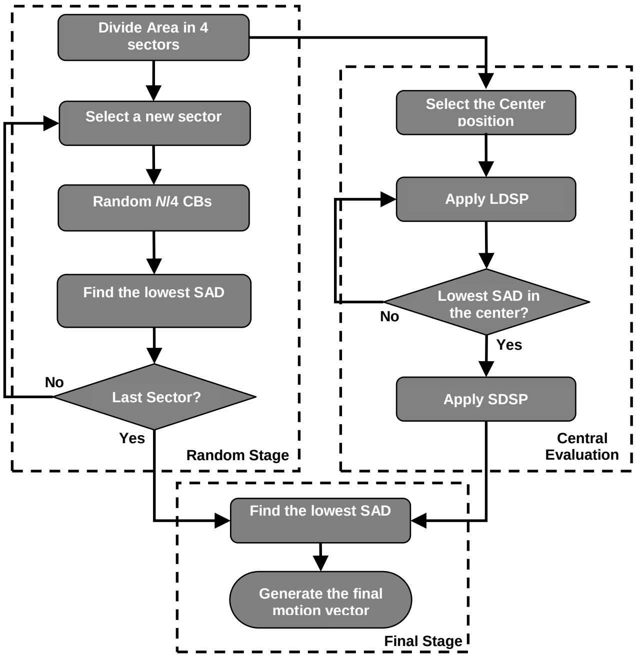

Fig. 3. S&IS algorithm

The insertion of the random spread blocks evaluation does not increase significantly the ME complexity, when compared with DS algorithm. The S&IS complexity is directly dependent on the *N* parameter. With the random evaluation and DS into S&IS algorithm, is possible obtain a high trade-off between video quality and computational complexity. In S&IS the use of randomness allows the algorithm to transpose some local minima, and a higher quality results is obtained, especially for high definition videos.

D. Low Density and Iterative Search

The Low Density and Iterative Search algorithm (LD&IS) aims also to obtain a high quality in high definition videos, with a lower computational complexity. The LD&IS is very similar to the S&IS, however, the random step is replaces by a low density evaluation. The LD&IS algorithm fixes 100 points, 25 in each sector of the search area. These 100 candidate blocks are evaluated and compared, and the best candidate block is selected. The distance of these points from the center of the search area are 5, 10, 20, 30 and 40 pixels for each side in both X and Y axes. These step aims to reduce the local minima.

The center of the search area, the LD&IS algorithm applies the same evaluation as presented for the S&IS algorithm. The best candidate from these two steps is selected and the motion vector is generated for them.

III. Software Evaluation

Evaluations were made for the following algorithms: DS, FS, DMPDS, S&IS and LD&IS. Ten high definition videos sequences were used: blue_sky, man_in_car, pedestrian_area, Riverbed, rolling_tomatoes, rush_hour, station2, SunFlower, Traffic and Tractor [9]. The results are presented in Table 1 and it presents the number of Evaluated Candidate Blocks (ECB) and quality results in PSNR. The block size was defined in 16x16 samples with a 4:1 sub-sampling rate for all evaluated algorithms. For high definition video the use of smaller sizes blocks does not contribute significantly to the efficiency of the coding process [8]. Then, 16x16 or bigger blocks are a good option if only one block size is supported.

The sub-sampling causes an expressive complexity reduction with an acceptable quality loss. As the hardware architectures work with these characteristics, the algorithms were evaluated for the same parameters. Besides, the DS algorithm are restricted to five iterations, for DMPDS, S&IS and LD&IS, to achieve a high processing rate, in order to establish a good trade-off between video quality and computational complexity.

As expected, the FS algorithm achieved the best results with an average PSNR of 35.71 dB, but with the highest amount of candidate blocks compared. This occurs because the FS algorithm compares all the candidate blocks in a given search area to find the best candidate block. The FS is used as a quality parameter for the comparison with other algorithms.

The DS algorithm achieved an average PSNR loss of 2.97 dB when compared to FS. However, the FS algorithm needs to compare 293 times more candidate blocks than DS. This occurs because the DS does not always generate an optimum motion vector with the lowest residues block, as previously explained, the DS does not explores the whole search area. In Table 1 is possible to note that DS is the algorithm that realize the smaller number of comparisons of candidate blocks, since this traditional algorithm easily choose local minima.

The S&IS and LD&IS algorithms results achieved higher PSNR, 1.41 dB and 1.40 dB, respectively, in comparison with DS, with an increase in the number of compared blocks of four times. This is justified because the S&IS and LD&IS performs the DS step in the central region of the search area, and also the evaluation (random for the S&IS and fixed on the LD&IS) on the sectors, to avoid local minima.

The algorithm that obtained the best tradeoff between video quality and computational cost was the DMPDS algorithm, achieving a PSNR of 34.87 dB, only 0.84 dB lower than presented by FS. The number of compared blocks is 45 times lower than presented by FS. As previously mentioned, the DMPDS applies five DS and works with a parameter dynamically controlled that adapts itself according to the characteristic of the video, thus the algorithm obtains a high quality for any type of video, for this reason the DMPDS achieved good results in video quality with a low computational complexity.

Table 1 – Software evaluation

Video	DS		FS		DMPDS		S&IS		LD&IS	
	PSNR (dB)	**#ECB ($x10^9$)**	**PSNR (dB)**	**#ECB ($x10^9$)**	**PSNR (dB)**	**#ECB ($x10^9$)**	**PSNR (dB)**	**#ECB ($x10^9$)**	**PSNR (dB)**	**#ECB ($x10^9$)**
blue_sky	30.01	0.04	34.43	14.66	33.73	0.24	31.12	0.20	31.74	0.20
man_in_car	37.80	0.03	39.99	14.66	39.60	0.24	39.31	0.18	39.26	0.19
pedestrian_area	32.22	0.05	35.97	14.66	35.25	0.34	34.83	0.19	34.79	0.20
Riverbed	24.42	0.06	27.72	14.66	26.86	0.36	26.47	0.21	26.50	0.21
rolling_tomatoes	37.38	0.03	38.18	14.66	38.32	0.28	37.87	0.18	37.87	0.19
rush_hour	36.48	0.03	37.40	14.66	37.28	0.36	36.99	0.18	36.95	0.19
station2	37.76	0.04	38.64	14.66	38.50	0.22	37.98	0.19	38.03	0.19
Sunflower	37.11	0.05	39.00	14.66	38.53	0.43	37.90	0.19	37.88	0.21
Traffic	24.90	0.07	32.45	14.66	28.81	0.39	28.27	0.21	28.03	0.22
Tractor	29.26	0.06	32.25	14.66	31.85	0.33	30.71	0.21	30.32	0.22
Average	**32.74**	**0.05**	**35.71**	**14.66**	**34.87**	**0.32**	**34.15**	**0.20**	**34.14**	**0.20**

Table 2 – Synthesis results

Architecture	Technology	Frequency (MHz)	Area	Memory (KBits)	Cycles per block	Full HD fps	QFHD fps
DMPDS [4]	Stratix 4	187.58	34.5 KALUTs	46.2	170	136	34
LD&IS [6]	Stratix 4	256.84	18.5 KALUTs	46	174	180	45
S&IS [5]	Stratix 4	210.5	18.5 KALUTs	37	174	149	37.3
S&IS [5]	90nm	169	84.32 KGates	55.9	174	119.9	30

IV. ME HARDWARE ARCHITECTURES

As all the developed algorithms have the characteristic of being hardware-friendly, allowing an easily architecture developed for them. The main objective is reach high performance using low hardware resources. This section presents the developed architectures for DMPDS [4], S&IS [5] and LD&IS [6] algorithms.

The developed architectures works with block size 16x16 and uses 4:1 pixel sub-sampling. The DS core used in all those architectures is restricted to 5 iterations. The restriction of the number of iterations of DS was used to ensure a high processing rate and to reach a good trade-off between quality and complexity. Based on this fact, the restriction guarantees a high processing rate.

The DMPDS architecture performance is focused on real time Quad Full HD (QFHD) videos. According to the synthesis results reported in Table 2, the DMPDS architecture can work with an operating frequency of 187.58 MHz, using only 19% of ALUTs, corresponding to 24% of the total resources of the FPGA, and also a memory bits usage of less than 1% of available FPGA memory. It processes up to 34 QFHD (4 times larger than HD 1080p) frames per second [4].

The S&IS architecture was described in VHDL and synthesized for an Altera Stratix 4 EP4S40G2F40I2 FPGA and for 90nm ST standard cells technology. According to Table 2, the Altera Stratix 4 FPGA synthesis can achieve a processing rate of 37.3 QFHD frames per second, working with a maximum frequency of 210.5 MHz and using 18.5K ALUTs. The ASIC 90nm implementation, can reach a processing 30 frames per second QFHD, works with a frequency of 169 MHz and uses 84.32K ALUTs [5].

The LD&IS architecture was described in VHDL and synthesized for an Altera Stratix 4 FPGA EP4S40G2F40I2. According to the results shown in Table 2, only 16% of ALUTs, 10 % of the total available registers and 1% of the available memory bits was used. To encode a block of 16x16 samples, the architecture requires 174 clock cycles, and can reach 254.84 MHz, processing 45 QFHD frames per second [6].

According to Table 2 the DMPDS architecture uses a larger area and more memory bits than S&IS and LD&IS architectures. However, the DMPDS architecture can process a block in a fewer number of clock cycles. Synthesis results shows that all architectures are capable to process more them 30 QFHD frames per second, surpassing the requirements to process QFHD videos in real time. Comparing to other architectures the LD&IS architecture obtained the higher processing rate with low hardware resources utilization.

V. CONCLUSIONS

This paper presented three fast and hardware-friendly motion estimation algorithms: The Dynamic Multi-Point Diamond Search (DMPDS), Spread and Iterative Search (S&IS) and Low Density and Iterative Search (LD&IS). These algorithms were evaluated for ten high definition videos and software results are presented in the paper. The results evaluation shows that the DMPDS algorithm was the one that achieved the best trade-off between video quality and computational complexity, achieving good results in PSNR with a low number of candidate blocks compared. This occurs because the DMPDS applies five DS, one in the center and the other four in each sector of the search area. The distance of these four DS from the center of the search area is dynamically controlled by a parameter which adapts itself according to the video content characteristics, achieving high quality for any kind of video.

Furthermore, the developed architectures for these three ME algorithms were presented. These architectures were described in VHDL and synthesized for Altera Stratix 4 FPGA and 90nm ASIC. The results showed that the designed architectures achieve a good processing rate with low hardware resources utilization. The LD&IS architecture obtained the highest processing rate and also the lowest hardware resources utilization and was capable to process 45 QFHD frames per second.

REFERENCES

[1] Cheng,Y. , et al. 2009. An H.264 Spatio-Temporal Hierarchical Fast Motion Estimation Algorithm for High-Definition Video. IEEE ISCAS 2009. pp. 880-883.

[2] JVT Editors (T. Wiegand, G. Sullivan, A. Luthra), Draft ITU-T Recommendation and final draft international standard of joint video specification (ITU- T Rec.H.264|ISO/IEC 14496-10 AVC), 2003.

[3] JCT. Working Draft 3 of High-Efficiency Video Coding. JCTVC-E603, 2011.

[4] Sanchez, G., et al. 2012. Real Time QFHD Motion Estimation Architecture for DMPDS Algorithm. IEEE SPL 2012.

[5] Sanchez, G., et al. Spread and Iterative Search: A High Quality Motion Estimation Algorithm for High Definition Videos and its VLSI Design. IEEE ICME 2012.

[6] Sanchez, G., A Fast Hardware-Friendly Motion Estimation Algorithm and its VLSI Design for Real Time Ultra High Definition Applications. IEEE LASCAS 2013.

[7] I. Richardson, Video codec design: developing image and video compression systems. Wiley, 2002.

[8] Porto, M., et al., "An Efficient ME Architecture for High Definition Videos Using the New MPDS Algorithm", ACM SBCCI 2011.

[9] Xiph.org: Test media, available at <http://media.xiph.org/video/derf/>, Oct, 2011.