

Algorithm and VLSI Design for a High Quality Motion Estimation Focused on High Definition Videos

Gustavo Sanchez, Marcelo Porto, Luciano Agostini

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

Universidade Federal de Pelotas

Abstract

This paper presents the Spread and Iterative Search (S&IS) motion estimation algorithm, which uses a random spread evaluation together with a central iterative evaluation to avoid local minima falls and to increase the image quality for high definition videos. Considering Full HD videos, S&IS reached an average PSNR gain of 1.41dB when compared to Diamond Search, with an increase of about four times in the number of evaluated blocks. An efficient architecture for the S&IS algorithm is also presented in this paper. The architecture was designed targeting real time processing (30 fps) for QFHD videos (3840×2160 pixels). The architecture was described in VHDL and synthesized for TSMC 90nm standard cells technology. Synthesis results show that the architecture is able to process QFHD frames in real time and also is able to reach a good trade-off among area, memory and power consumption, processing QFHD videos with 62.2 mW.

1. Introduction

Motion Estimation (ME) is the video coding step that demands the highest computational effort (80% of the encoder complexity [1]) and is responsible for the highest compressing rates among all encoding tools. The current video coding standards, like MPEG2 [2], H.264/AVC [3] and the emerging HEVC (High Efficiency Video Coding) do not restrict how the ME is done [4]. Based on this fact, there is a vast space to explore new algorithm solutions, which are evaluated according to the trade-off between complexity and objective quality of the digital video. The new ME algorithmic solutions, besides to provide better quality and/or reduced complexity, also should consider the hardware implementation features. This is important because software solutions hardly can achieve real time (30 frames per second) when processing high definition (HD) videos. The Full Search (FS) is the optimum ME search algorithm in terms of quality. It explores all possibilities in the search area and the best possible match is always found. However, the FS computational complexity is extremely prohibitive, mainly when encoding HD videos [5].

There are a lot of fast algorithms based on different heuristics that explore video characteristics related to temporal locality, which are able to achieve good results in the trade-off between computational complexity and objective digital video quality. However, most of these algorithms have as weakness the fragility when dealing with local minima, especially when encoding HD videos. Simulation results show that the encoding process for low resolution videos is not substantially affected by these local minima falls and this fragility becomes imperceptible [6]. However, as higher is the resolution, as higher is the tendency of these fast algorithms fall in local minima, causing a quality degradation.

Most of fast ME published algorithms present their quality results only for low resolution videos. In a previous work of our group an investigation about ME showed that the quality can drastically change when the resolution is increased and fast algorithms are used [6] and concludes that with the resolution increase, traditional fast algorithms significantly lose quality when compared to FS.

The main objective of this work is to propose a new ME algorithm to increase the ME quality results for HD videos. This new algorithm was called Spread and Iterative Search (S&IS). Besides, S&IS hardware architecture was also presented in this paper, which is capable to process QFHD frames (3840×2160 pixels) in real time at 30 frames per second (fps). Software evaluations for HD videos show that S&IS considerably surpass the Diamond Search (DS) quality, adding a small computational complexity in the ME process. It is achieved inserting randomness in the ME, which is an efficient way to avoid local minima falls.

2. Related Works

There are a lot of ME algorithms and architectures published in the literature, however, only a few of them are focused on HD videos and most of them are based on algorithms which are susceptible to local minima falls. Based on this fact, there are a few works that can be fairly compared to S&IS because in low resolutions, many fast algorithms are able to achieve quality results comparable to that reached by FS. But these good results are not maintained for high definition videos.

A ME architecture with the algorithm Dynamic Variable Step Search (DVSS) is proposed in [7]. For low resolution videos, this algorithm achieves almost the same quality than FS, however, quality results for Full HD videos area not presented. A ME architecture for the Fast Top-Winners (FTW) algorithm is proposed in [8]. That work uses the level C data re-use scheme, which allows a reduced use of memory bits. The work [9] presents an algorithm for Multi-Resolution ME (MMEA). That work considers two reference frames in the ME

process but it does not present quality results for HD videos. Adaptive True Motion Estimation (ATME) algorithm is presented in [10]. It presents evaluations for Full HD videos. In that work are necessary only 104 cycles in average to process a 16x16 block.

It is difficult to find papers proposing ME algorithms that present quality results considering HD. Only the work [10] evaluates the proposed algorithms using Full HD videos. Then the comparisons are not fair, since for most of published fast algorithms as higher is the video definition as higher is the losses in image quality when compared to FS.

3. S&IS Algorithm

The main goal of the fast ME S&IS algorithm is to achieve high quality when encoding HD videos. This goal is reached using techniques to reduce local minima falls. A pseudo-random generator is used to generate N spread positions inside the search area, and then the Sum of Absolute Differences (SAD) is computed for these N candidate blocks. These SADs are compared and the lowest one is selected. This is the spread block evaluation part of S&IS. Together with the random step, a traditional approach is used to evaluate the center of the search area. The DS is used to explore the co-located block and its neighbors in an iterative process. This central evaluation is important especially when low motion videos are being encoded. The DS is used in this central evaluation of S&IS algorithm.

The S&IS works as follows: (1) the search area is divided in four sectors; (2) N spread positions are pseudo-randomly generated for each sector; (3) the SAD is computed for each one of these positions and the best one is selected; (4) an iterative process using DS algorithm is started at the center of the search area; (5) the best result between the spread positions and the center DS is selected as the best matching.

It is possible to notice that the random spread block evaluation and the iterative DS evaluation can be calculated in a parallel or in a sequential fashion. The DS is an iterative algorithm with data dependencies among these iterations. When the parallelism is explored in the hardware design of S&IS algorithm, the iterative DS step will define the final throughput, since all SAD calculations of the spread random step can be done in parallel.

4. S&IS VLSI Design

Fig. 1 presents the designed architecture for S&IS algorithm. This architecture was described in VHDL with the following specifications: $N = 96$ for each search area, a block size of 16x16 with 4:1 pixel sub-sampling and the maximum iterations of Diamond Search was restricted to five.

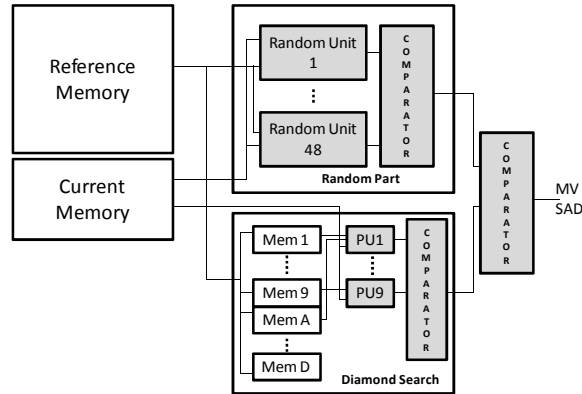


Fig. 1 – S&IS VLSI Design

In advance to start the ME process, it is necessary to fill the Reference Memory from an off-chip memory. By this time, the Linear Feedback Shift Register (LFSR) randomizes 48 positions (inside the Random Unit). To achieve a better processing rate, the reference memory starts to be written with the necessary positions for the first Large Diamond Search Pattern (LDSP) calculation. During this period, the DS local memories start to ask the Reference Memory for data of the nine positions of the LDSP and data for the refinement. When the local memories are filled, the data are sent to the Processing Units, where the SAD is computed. The comparator is responsible to find the lowest SAD.

Each PU receives eight samples per cycle. The PU processes these samples in four pipeline stages. The comparator receives the SADs calculated by the nine PUs and it is able to define the lowest SAD in five cycles.

Each Random Unit in Fig. 1 is composed by a Local Memory (8x8 bytes), a LFSR, a controller to select the position that will be written in the Local Memory and a PU. A half of the 96 randomly selected positions are evaluated in parallel, then, the hardware design used 48 Random Unities.

When the DS is processing and comparing the SADs for a LDSP iteration, the Reference Memory starts to be read by the Random Step. The Controller of each Random Unit is responsible to select if its Local Memory should store the information read from the Reference Memory.

When the DS finishes the first LDSP calculations, the Reference Memory stops to be read by the Random Step and the DS step starts to read the Reference Memory again. When the DS finishes the Reference Memory reading, the Random step continues to read the remaining lines, finishing the selection of the first half of the random positions. In the next DS LDSP iteration, the SAD of this first half of random positions is computed in parallel for these 48 positions. Then, the SADs are sent to the comparator of the random step and the best SAD is stored. The Random step comparator receives 48 SAD values and it is able to find the best SAD in seven clock cycles. This interlaced way to read the Reference Memory is repeated for the fourth and five LDSP iterations when the second group of random positions is read from the Reference Memory. When the Small Diamond Search Pattern (SDSP) process is evaluated, the last 48 positions of the Random Step are evaluated. The SADs calculated in this stage are compared with the best SAD of the previous random step and with the best DS SAD and the motion vector is generated indicating the block with the best SAD among these three results. The proposed architecture spends only 174 clock cycles to encode a 16x16 block.

5. Results and Comparisons

The S&IS algorithm was evaluated in a framework where the motion estimation quality was evaluated using ten standard video sequences were used [11]. The quality results (PSNR) and the number of computed blocks (NCB) are presented in Tab. 1. A comparison against DS, FS and the random step running alone (without DS) are also presented in Tab. 1. All evaluations considered: (1) a block size with 16x16 samples and using a 4:1 sub-sampling rate; (2) a search area with 96x96 samples and (3) a restriction of five iterations. Analyzing Tab. 1, the S&IS achieves an average quality gain of 1.41 dB in comparison to DS with a complexity increase of four times. In comparison to FS, the S&IS achieved an average PSNR loss of 1.56 dB, using 73 times less compared blocks than FS. The Random Step running alone obtains a PSNR loss of 3.92 dB when compared to DS. On other hand, the random step is responsible to an average of 48.2% of the best SADs obtained by the S&IS, which means that the good results of S&IS is function of an efficient combination of both: the random step and the DS step. The S&IS architecture has been described in VHDL and synthesized for 90nm ST standard cells technology. Synthesis results are presented in Tab. 2. The architecture used 84.32 KGates and presented a power consumption of 62.2 mW when running at 169 MHz. it is possible to process QFHD videos in real time. Tab. 2 also presents comparisons with related works. This work was compared with the works [7-10]. Tab. 2 presents the following parameters: technology, maximum operation frequency, consumed area, memory bits, cycles per block, frames per second for Full HD and QFHD videos and power consumption. Comparing with [7], our architecture uses a lower number of clock cycles to process one block and it achieves a better operational frequency, allowing S&IS architecture to reach a better processing rate. The work [8] proposes a ME architecture that uses a lower area than S&IS architecture and uses a lower number of memory bits. On other hand, our work achieves almost the double of the operation frequency and uses 9.5 times less cycles than [8]. Again, our architecture has a better processing rate. The work [9] is able to process Full HD videos in real time using less memory bits than S&IS architecture. However, the S&IS architecture can process QFHD videos in real time and it also uses lower area resources. Comparing the area results with the work [10], the S&IS architecture uses less hardware resources. The work [10] uses less clock cycles than S&IS to process one block. However, the S&IS architecture operational frequency is much higher, enabling our architecture to achieve a higher processing rate than [10]. Our solution uses high operation frequency to achieve real time processing of QFHD videos, however, if our work was focused on Full HD videos, the operation frequency should be a quart of the current one, which would reduce the power consumption by a factor of four. This means that S&IS architecture would consume about 16 mW, which is a competitive result of power to encode Full HD videos.

Tab.1 – S&IS quality and complexity evaluation and comparisons

V�deo	Random		DS		S&IS		FS	
	PSNR (dB)	#NCB (x10 ⁹)	PSNR (dB)	#NCB (x10 ⁹)	PSNR (dB)	#NCB (x10 ⁹)	PSNR (dB)	#NCB (x10 ⁹)
blue_sky	22.22	0.15	30.01	0.04	31.12	0.20	34.43	14.66
man_in_car	33.68	0.15	37.80	0.03	39.31	0.18	39.99	14.66
pedestrian_area	29.96	0.15	32.22	0.05	34.83	0.19	35.97	14.66
riverbed	25.74	0.15	24.42	0.06	26.47	0.21	27.72	14.66
rolling_tomatoes	33.59	0.15	37.38	0.03	37.87	0.18	38.18	14.66
rush_hour	31.87	0.15	36.48	0.03	36.99	0.18	37.40	14.66
station2	30.39	0.15	37.76	0.04	37.98	0.19	38.64	14.66
sunflower	29.51	0.15	37.11	0.05	37.90	0.19	39.00	14.66
traffic	25.64	0.15	24.90	0.07	28.27	0.21	32.45	14.66
tractor	26.63	0.15	29.26	0.06	30.71	0.21	32.25	14.66
Average	28.92	0.15	32.74	0.05	34.15	0.20	35.71	14.66

Tab.2 – Synthesis Results and Comparisons

Architecture	Technology	Frequency (MHz)	Area	Memory (Kbits)	Cycles per Block	Full HD fps	QFHD fps	Power (mW)
Tasdizen [7]	Virtex 5	130	2,282 KCLBs	0.51	467	34	8.5	n.a
Lai [8]	180nm	83.3	26 KGates	28.7	1282	8	2	60.84
Yin [9]	180nm	200	260 KGates	11.3	872	28	7	n.a
Cetin [10]	FPGA 90nm	63	33 KLUTS	8 dual-port block RAM	104 (average)	74.7	18.7	n.a
S&IS	90nm	169	84.32 KGates	55.9	174	119.9	30	62.2

6. Conclusions

This work presented the Spread and Iterative Search (S&IS) ME algorithm and the respective architecture focused on QFHD videos. The objective of this work was to propose a new fast algorithm capable to reach better image quality than other fast algorithms for high definition videos. The algorithm uses spread random positions as a strategy to reduce local minima falls. Evaluation results shows that the S&IS algorithm achieves an average gain of 1.41 dB in quality when compared to DS.

The proposed architecture was synthesized for ST 90nm standard cells technology. Synthesis results shows that the architecture is able to process QFHD videos consuming only 62.2mW. Comparing to others architectures in the literature, this work obtained a better frequency and a better processing rate with a considerable low number of used clock cycles per block.

7. References

- [1] Cheng, Y., et al., "An H.264 Spatio-Temporal Hierarchical Fast Motion Estimation Algorithm for High-Definition Video", IEEE ISCAS, pp. 880-883, 2009.
- [2] ITU-T e ISO/IEC JTC1, "Generic coding of moving pictures and associated audio information – Part 2: Video", ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2), 1994 November.
- [3] 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.
- [4] JCT. Working Draft 3 of High-Efficiency Video Coding. JCTVC-E603, 2011.
- [5] I. Richardson, Video codec design: developing image and video compression systems. Wiley, 2002.
- [6] Porto, M., et al., "An Efficient ME Architecture for High Definition Videos Using the New MPDS Algorithm", ACM SBCCI, pp 119 – 124, 2011.
- [7] Tasdizen, O., et al., "Dynamically variable step search motion estimation algorithm and a dynamically reconfigurable hardware for its implementation", IEEE TCE, vol. 55, n. 3, pp. 1645-1653, Aug. 2009.
- [8] Lai, Y., et al., "Hybrid Parallel Motion Estimation Architecture Based on Fast Top-Winners Search Algorithm", IEEE TCE, vol. 56, n. 3, pp. 1837-1842, Aug. 2010.
- [9] Yin, H., et al., "A Hardware-Efficient Multi-Resolution Block Matching Algorithm and Its VLSI Architecture for High Definition MPEG-Like Video Encoders", IEEE TCSVT, vol. 20, n. 9, pp. 1242-1254, Sept. 2010.
- [10] Cetin, M., et al., "An Adaptive True Motion Estimation Algorithm for Frame Rate Conversion of High Definition Video and Its Hardware Implementations", IEEE TCE, vol. 57, n. 2, May 2011.
- [11] Xiph.org: Test media, available at <<http://media.xiph.org/video/derf/>>, Oct, 2011.