FPGA structures with concentrated vs distributed memory for images comparison

Sergio Geninatti
Univ. Nacional de Rosario
Rosario – Argentina
foco@fceia.unr.edu.ar

Gerardo Gennai
Univ. Nacional de Rosario
Rosario - Argentina
ggennai@eie.fceia.unr.edu.ar

Santiago Roatta
Univ. Nacional de Rosario
Rosario - Argentina
sroatta@eie.fceia.unr.edu.ar

Eduardo Boemo
Univ. Autónoma de Madrid
Madrid - España
eduardo.boemo@uam.es

***Abstract*—This paper proposes a strategy for calculating the windowed correlation between histograms using FPGA. The processing is done extracting image features directly from the compressed video stream. Thus, maximum speed is obtained; the circuit can manage up to 200.000 photograms per second. Two options for calculating the histogram are proposed: one based on distributed registers, and other based on blocks of RAM. The performance of the two alternatives is compared. Exhaustive registers-level segmentation for the algorithms and block memory-level segmentation for histograms is applied. The final circuit is evaluated taking into account its adaptation to different video resolutions, number of histogram bins, parallelism in the concurrent computation, and progress of FPGA technology.**

Keywords— FPGA - Image Analisis - Video Processing - Luminosity Frequency - Similarity Calculation - Arithmetic - algorithms - processing images -histogram.

I. INTRODUCTION

In this paper, the word RAMBlock is used to name the RAM blocks integrated in Xilinx FPGA Spartan chips. However, all results apply to any RAM integrated in a FPGA chip.

Video analysis is a field where FPGA-based solutions are necessary to fulfill real-time requirements. For example, these devices are present in applications of texture analysis [1], background segmentation in video surveillance applications [2], Real-time Visual Detection and Matching [3], and stereo correspondence to estimate objects position in systems like autonomous driving, UAVs and robots [4].

In this paper, we propose a FPGA implementation to calculate the similarity between two video frames by correlating their histograms. This technique is very useful to locate editions effect in a video (shots and gradual transitions), to compare video sequences, or to carry out temporal video segmentation.

The employed technique is based on previous studies presented in [5], where the authors compare consecutive video frames applying the windowed correlation to the DC coefficient of the video frames. The use of these coefficients has an important impact in the amount of data to be processed in each frame, allowing a fast determination of the similarity value.

The concurrent computation of histograms is an issue widely discussed by researchers. However, its implementation on FPGA is strongly linked to the way that the data are handled. This is confirmed in [6], where multiple partial histograms on the same image are computed through a network of RAMBlocks, or [7] that exploits certain repetitions in parts of overlapping histograms. In [8], specific accumulators with a structure of decoders for histogram equalization are combined.

In this paper, we propose two different circuits to calculate the similarity between video frames. In addition we make a comparison between different generations of FPGA in order to assess the fitting of the designs with the technology evolution.

The paper is organized as follows. Next section shows the formulation of the similarity computation. Section III presents the proposed design at high level, pointing out the connection between the main functional units: histogram and correlation units. Section IV describes two different strategies for histogram implementation. Section V gives details about the design of the correlation unit. Section VI performs a performance analysis of the proposed design. Finally, the main conclusions of this work are discussed in the last section.

II. SIMILARITY CALCULATION

The correlation is an algorithm frequently implemented in FPGA for similarity evaluation. Work [9] use it for automatic people identification and [10] use it for fingerprint matching.

The similarity value, S(i-1, i), between two frames f_{i-1} and f_i can be calculated using a windowed correlation of the frames histograms according to the following expression:

$$S(i\text{-}1,\, i) = \frac{\sum_b H_{i-1}[b] \cdot W_i[b]}{\sqrt{\sum_b H_{i-1}[b] \cdot W_{i-1}[b]} \cdot \sqrt{\sum_b H_i[b] \cdot W_i[b]}} \quad (1)$$

Where

$$W_i[b] = \sum_{v=-1}^{v=1} H_i[b+v]$$

978-1-4799-6848-0/14/$31.00 ©2014 IEEE

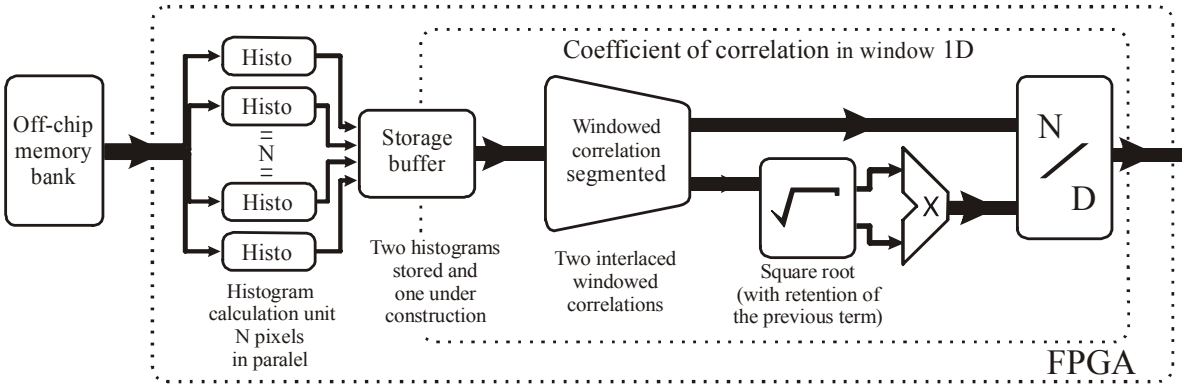

Fig 1. Block diagram of similarity calculation by brightness.

In the previous expression, $H_i[b]$ is the frame´s histogram b-th bin value. $W_i[b]$ is the windowed value of the b-th histogram bin. It is calculated by adding the bin contents within a window of size 3 centered on the b-th bin.

III. FPGA DESIGN

In previous works, Kokufuta and Maruyama [6] and Jamro et al. [7] proposed the use of RAMBlocks for histogram implementation. In a different way, Alsuwailem and Alshebeili employed distributed adders [8]. In our case, the two architectures for histograms calculation are based on centralized and distributed memory, respectively. Both design cases try to exploit the inherent concurrency lying in the correlation computation.

A proposal to solve efficiently sums of products with FPGAs can be seen in [11]. The improvement made in this work deals with the reuse of concurrent data and simplifications allowed by the regularity of operations. Furthermore, we solve irregularities in equation (1). For example, the calculation of the term W_i requires the two nearby values to the H_i. This is an irregular calculation, because the first and the last histogram values only use a neighbor value.

While implementing parallel and pipelined algorithms, irregularities can also be found. For instance, the number of points of a frame might not be divisible by the input number of histogram calculation unit.

The diagram of Fig. 1 shows the complete circuit that calculates the similarity between consecutive frames of a video sequence. We distinguish three zones:

- An off-chip memory bank.
- The histogram calculation block.
- The implementation of the similarity calculation (Coefficient of correlation in window 1D).

In [4], the performance achieved with high-resolution videos is attributed to the use of the high bandwidth of on-chip memory banks. In our work, we show that a comprehensive architectural design overcome the limitations of off-chip memory bank.

We have applied the *level segmentation by memory* technique in the circuit structure, overlapping the processing time between stages. Thus, while it is calculating the histogram of a frame, the correlation circuit processes the data from the two previous frames.

The *intermediate buffer* contains three histograms of consecutive frames and, using a dual port structure, it allows us to supply data to the circuit that process the similarity function while the calculation of the next histogram is completed.

IV. HISTOGRAM CALCULATION

In order to exploit concurrency, we proposed to process multiple image pixels simultaneously. Each processed pixel increments the value of the accumulator corresponding to its luminance level. The parallel processing of multiple pixels generates write conflicts when a specific position of the accumulator is updated by more than one pixel.

Collisions due to parallel histogram computation on GPU were studied in [12]. That work employs a kernel where each processing thread updates its own partial histogram (called "sub-histogram"). Thus, to estimate the final histogram, a second kernel is started to reduce all sub-histograms.

In our work, we reproduce the idea of [12] using a RAM Block-based architecture, but improving it with a dedicated-registers- solution. It leads to increase the speed by a factor of 1.5. Then, we develop two different implementations of the block that process histogram updating. The first defines a histogram calculation unit based on exclusive memory (sub-histogram). The second option employs a circuit to solve the collision in each histogram register (dedicated registers).

A. *RAM Block version (RBV)*

The use of RAM Blocks to concentrate accumulating registers is an alternative to histogram calculation (Fig 2). The main difference respect to the dedicated registers scheme is that no exclusive accumulator for luminance value is utilized. Each calculation unit generates partial histograms, which are then added in the output adder.

Each RAM has two separate ports: a 16-bit port is connected to the accumulator circuit and a 32-bit port allows simultaneous extraction of two values to calculate the output sum. The advantages of this configuration are:

- It occupies a small number of Xilinx CLBs.
- It facilitates the storage of previous values of histograms.

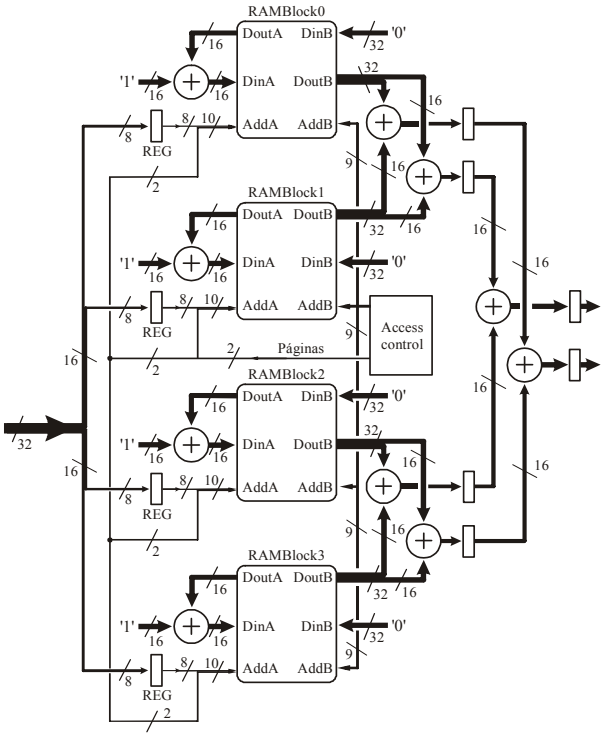

Fig 2. Histograms calculation by RAM Block.

B. *Dedicated registers version (DRV)*

This version is implemented with a combination of accumulators, multiplexers, and LUTs (Fig 3). The multiplexers generate a signal that addresses to the register that must be increased. The LUTs read signals from the multiplexers and decide the increment value for each register.

The advantages of this configuration are:

- Histogram values are obtained concurrently, which facilitates the parallel calculation in the next step.
- All registers can be erased simultaneously to start present in one frame to another the calculation of a new histogram.

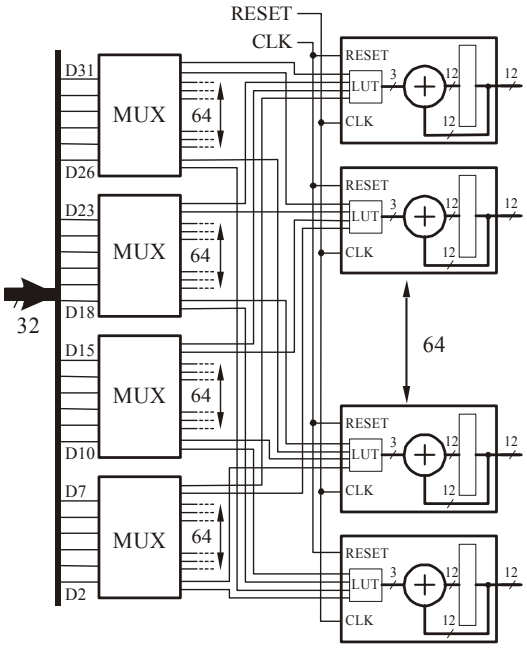

Fig 3. Histograms calculation by dedicated registers (implemented using CLBs).

V. WINDOWED CORRELATION

The design of the circuit that calculates the sums in equation (1) takes advantage of the repetition of certain terms in the computation of correlation between consecutive frames. Thus, the autocorrelation of the denominator of equation (1) for frame i, can be employed again in the next calculation, as the autocorrelation of frame i-1.

A similar strategy is adopted with correlation products. For example, the windowed term $W_{i-1}(b)$ multiplies $H_{i-1}(b)$ in the numerator and $H_i(b)$ in the denominator. Thus, both products are performed concurrently.

Consequently, only two sums for each frame must be done. The third one, present in the denominator of Equation (1), is reused from the previous frame. We employ a single pipelined adder for interlaced calculation of the two sums. The other adder that calculates the windowed terms also exploit the repetition of partial sums. Thus, a further optimization of the resulting performance is obtained.

The following example illustrates the above paragrapth. The expression of two consecutive windowed terms is:

$$Wi[j] = Hi[j-1] + Hi[j] + Hi[j+1]$$

$$Wi[j+1] = Hi[j] + Hi[j+1] + Hi[j+2]$$

Thus, if these sums are carried out concurrently, the expression Hi[j] + Hi[j +1] can be utilized twice.

We chose to provide inputs to the adder blocks in such a way that eight histogram bins are computed simultaneously. It gives six windowed terms each clock cycle.

Table 1 shows the input sequence of histogram bins in the correlation block. It is noticeable that data from the consecutive histograms, H_{i-1} and H_i are alternated to obtain both sums with the same circuit. Moreover, the values in the input IN7 and IN8 are repeated in the input IN1 and IN2 of the next cycle, to solve the edge effect of windowed calculation.

This repetition of terms of the input vectors in the correlation block does not affect the overall performance, because the correlation is overlapped with the calculation of the next histogram.

Experimental data have been obtained employing 64 histogram bins for each frame. Our solution needs 22 clock cycles to read input data vectors, and 5 cycles to discharge the segmented calculation. Therefore, after 27 clock cycles the two sums of windowed correlation are obtained.

TABLE 1 SEQUENCE OF INPUT VECTORS FOR THE CORRELATION CALCULATION.

Histo	cycle	IN1	IN2	IN3	IN4	IN5	IN6	IN7	IN8
Hi	1	---	---	0	0	1	2	3	4
Hi-1	2	---	---	0	0	1	2	3	4
Hi	3	3	4	5	6	7	8	9	10
Hi-1	4	3	4	5	6	7	8	9	10
Hi	5	9	10	11	12	13	14	15	16
Hi-1	6	9	10	11	12	13	14	15	16
Hi	7	15	16	17	18	19	20	21	22
Hi-1	8	15	16	17	18	19	20	21	22
Hi	9	21	22	23	24	25	26	27	28
Hi-1	10	21	22	23	24	25	26	27	28
Hi	11	27	28	29	30	31	32	33	34
Hi-1	12	27	28	29	30	31	32	33	34
Hi	13	33	34	35	36	37	38	39	40
Hi-1	14	33	34	35	36	37	38	39	40
Hi	15	39	40	41	42	43	44	45	46
Hi-1	16	39	40	41	42	43	44	45	46
Hi	17	45	46	47	48	49	50	51	52
Hi-1	18	45	46	47	48	49	50	51	52
Hi	19	51	52	53	54	55	56	57	58
Hi-1	20	51	52	53	54	55	56	57	58
Hi	21	57	58	59	60	61	62	63	63
Hi-1	22	57	58	59	60	61	62	63	63

VI. COMBINED PERFORMANCE

We have tested the performance of the whole processor by connecting each of the two implementation of the histogram calculation with the windowed correlation scheme of Section V.

The experimental values in this section are obtained using a Spartan model XC3S1000-4C, with two 32-bit external memory banks. The hardware for testing uses standardized clock frequencies of 80 MHz and 160 MHz, which are derived from the 40 MHz external clock.

The histograms are calculated using DC images of the compressed video at a resolution of 352 x 288 pixels. During the on-circuit test, we utilized 4 concurrent pixeles and 64 bins histograms, reaching a throughput of 200.000 photograms per second.

The simulations results of different parallelism options and clock frequencies are shown in sections *A*, *B,* and *D* . In section *E* the results are extended to a Xilinx Spartan 6 model XC6SLX16-CSG324-3

A. *Effect of data segmentation*

As it was explained in section 4, the circuit is pipelined at frame level using an external memory buffer (Fig. 1). Two clock frequencies are necessary: on for calculating the histogram (Clk in) and another for the windowed correlation (Clk out).

The RBV allows a clock frequency 1.27 times higher than the DRV. However, DRV outperform RBV in processed frames figures by a ratio of 1.58. This is because the RBV take two clock cycles to process each histogram level while the DRV takes only one clock cycles.

With respect to the frequency value of Clk out, it is 1.25 times faster for the DRV design than for the RBV, because the assembly of the data vectors for the windowed correlation is simpler using DRV.

The calculation of the similarity function overlaps with the following histogram calculation thanks to the "intermediate buffer" (Fig. 1).

Fig. 4 shows a timeline for both histogram calculation circuits with 16 input histogram calculation unit. We have represented the overlap between the calculation of the next histogram synchronized by Clock In and the calculation of the windowed correlation synchronized by Clock Out.

The increment in parallelism reduces the number of clock cycles used for the histogram calculation. However, when more than 16 concurrent histograms are used to calculate the correlation needs more time to calculate the histogram, offsetting the effects of increased parallelism.

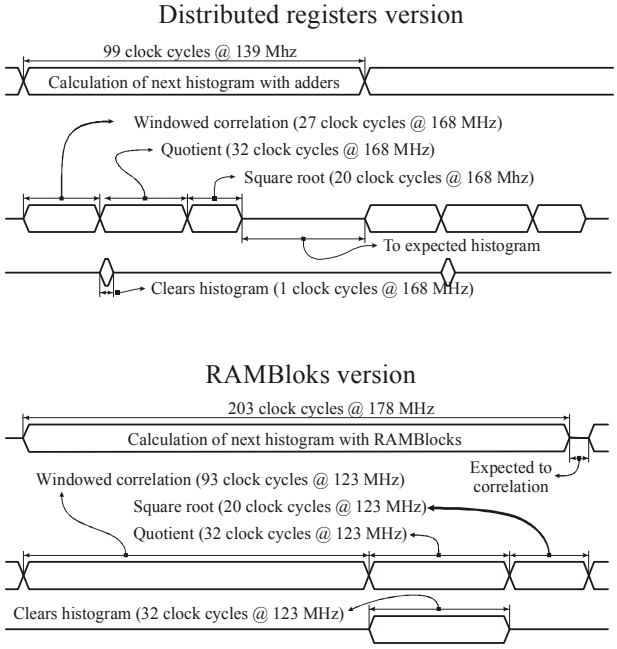

Fig 4. Details of use of overlapping times

B. *Effect of parallelism variation*

In order to test the scalability of our solution, we have evaluated the histogram computation performance by reading a variable number of frame pixels in parallel. In table 2 we show the obtained values when four pixels are read in parallel.

TABLE 2. TEST WITH 4 HISTOGRAM CALCULATION UNIT.

Type of circuit	FF	LUT	RAM Block	Clk[MHz]		Frame x1000
				In	out	
Custom adder	3720	7683	0	139	168	351
RAMBlock	167	377	4	177	134	222

We have also increased the parallelism by reading 8 pixels in parallel; the results are shown in Table 3. We note that the maximum clock frequency is maintained, so that the quantity of frames processed per second is duplicated and the relation of performance between the two versions is preserved.

TABLE 3. TEST WITH 8 HISTOGRAM CALCULATION UNIT.

Type of circuit	FF	LUT	RAM Block	Clk[MHz]		Frame x1000
				in	*out*	
Custom adder	3720	10349	0	139	168	702
RAMBlock	223	505	8	180	127	449

C. *Effect of video resolution*

Two aspects in the proposed circuits are affected by changes in the video resolution:

- The number of pixels to process grows quadratic with the video resolution.
- The resolution of the histogram bins must match the number of video pixels.

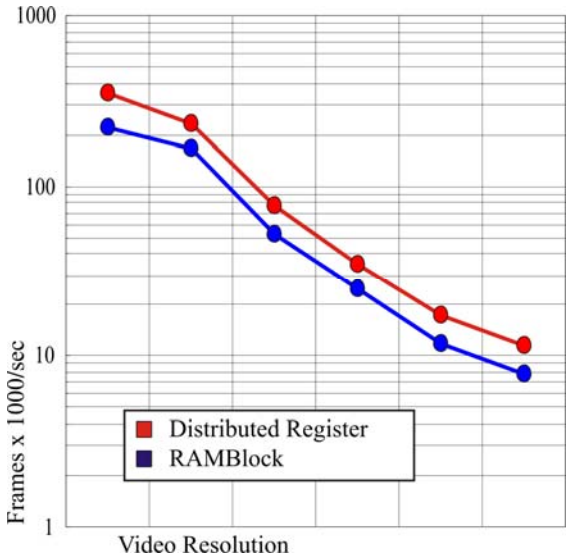

Fig 5. Performance in terms of the video resolution with Spartan3E

Fig. 5 shows the performance in thousands of frames per second of MPEG compressed video with increasing resolution from 352x288 to 2048x1536. The curve - in logarithmic scale - shows that the DRV based solution maintains a sustained improvement factor of 1.5 compared to the RBV based.

D. *Effect of histogram bins*

The number of luminance pixel determines the number of accumulators needed to construct the histogram. This greatly affects the structure of the DRV circuit and has a small influence in the RBV based design.

TABLE 4: VARIATIONS FROM THE BINS NUMBER

Bins FF		LUT	logic [nS]	route [nS]	MHz	Fotog. x1000
64	2048	2531	3,774	1,072	206	520
128	4096	5061	3,774	1,072	206	520
256	8192	10133	3,774	1,072	206	520

Table 4 shows that although the occupation of the circuit increases with the number of bins, the clock frequency is kept constant. Despite the architectural modification, the DRV circuit retains its advantages over RBV.

E. *Results for current process technology*

We repeated the experiments using a Spartan 6 model XC6SLX16-CSG324-3 (Fig. 6).

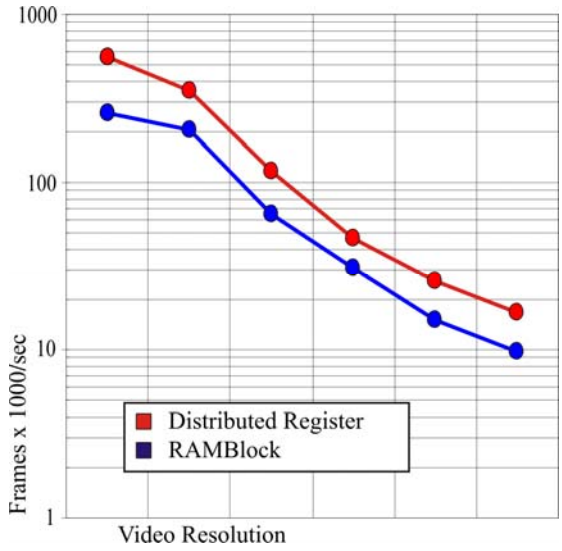

Fig 6. Performance in terms of the video resolution with Spartan6.

These tests show that process technology improvements reinforces the advantages of DRV circuit, instead of RBV in a proportion of 77%, surpassing even further the advantage of 50% achieved with the model SPARTAN3E. This advance is produced by the different architecture of CLBs that benefit the place and route of the distributed registers, as well as the incorporation of 6-LUTs (six input LUTs) instead of the 4-LUTs. The architecture of the SPARTAN6 CLBs, with higher number of FF, LUT, and specific carry logic, allows designing dedicated accumulators with less wiring delays.

VII. Conclusion

We have proposed and compared two FPGA architectures for the correlation of consecutive frame histograms. Main application is temporal video segmentation. The work solves the concurrency conflicts present in the parallel computation of histograms. The RBV based solution is also utilized in similar circuits on GPUs. But that DRV suit well only with FPGA. This option beats RBV in all the possible variations of parameters, such as: video resolution, number of histogram bins, and parallelism of the calculation.

We have shown that the DRV benefits from the architecture of the FPGA, and its advantage over RBV increases with the technological evolution of the FPGA. The ISE synthesis reports show that banks of 64 or 128 bits external memory and independent frequency clocks improve the performance.

Future work proposes a similar analysis using contour information to refine the temporal segmentation of video.

Another topic for future work is to study the coupling between the histogram calculation circuit, and the circuit for calculating convolution. The coupling between stages of concurrent computation is not a trivial problem.

VIII. Acknowledgements

This work is supported by *Fundación Vodafone España* under contracts FUAM 085212 and 085206. Additional funds have been obtained from the *Convenio Marco UAM-UNR* and *Euroform Polo Español* and *AVE program* of the National University of Rosario, Argentina.

IX. References

[1] Maroulis, D.; Iakovidis, D.; Bariamis D.; "FPGA-based System for Real-Time Video Texture Analysis" - Journal of Signal Processing Systems - 2008

[2] Bermak, M. S.; Chandrasekaran, A.; Amira, S.A. ; "An Efficient FPGA Implementation of Gaussian Mixture Models-Based Classifier Using Distributed Arithmetic" - Electronics, Circuits and Systems, IEEE 2006

[3] J. Wang, S. Zhong, L. Yan, and Z. Cao (2014, March) "An Embedded System-on-Chip Architecture for Real-time Visual Detection and Matching" Circuits and Systems for Video Technology, IEEE Transactions on (Volume:24 , Issue: 3).

[4] Schumacher, F., & Greiner, T. (2013, October). "Extension and FPGA architecture of the Generalized Hough Transform for real-time stereo correspondence." In Design and Architectures for Signal and Image Processing 2013 Conference on (pp. 223-229). IEEE.

[5] Yeo, B.L.; Liu, B.; "Rapid Scene Analysis on Compressed Video" – Transactions on circuits and systems for video technology - IEEE 1995

[6] K. Kokufuta, T. Maruyama "Real-time processing of contrast limited adaptive histogram equalization on FPGA" - FPL 2010 - IEEE

[7] Jamro, E.; Wielgosz, M.; Wiatr, K.; "FPGA Implementaton of Strongly Parallel Histogram Equalization"- DDECS 2007 – IEEE

[8] Alsuwailem, A.M.; Alshebeili, S.A.; "A new approach for real-time histogram equalization using FPGA" - ISPACS 2005 – IEEE

[9] Danese, G., Giachero, M., Leporati, F., Matrone, G., & Nazzicari, N. (2009, August). "An FPGA-based embedded system for fingerprint matching using phase-only correlation algorithm." In Digital System Design, Architectures, Methods and Tools, 2009. DSD'09. 12th Euromicro Conference on (pp. 672-679). IEEE.

[10] Lindoso, A., & Entrena, L. (2007). High performance FPGA-based image correlation. Journal of Real-Time Image Processing, 2(4), 223-233.

[11] Dinechin, F.; Pasca, B.; Cret, O.; Tudoran, R.; "An FPGA-specific Approach to Floating-Point Accumulation and Sum-of-Products" - FPT 2008 – IEEE

[12] Nugteren, C.; Van den Braak, G. J.; Corporaal, H; Mesman, B; "High Performance Predictable Histogramming on GPUs: Exploring and Evaluating Algorithm Trade-offs" - GPGPU-4 Mar 05-05 2011, Newport Beach, CA, USA.