Maciej WIELGOSZ, Marcin PIETROŃ, Michał KARWATOWSKI, Kazimierz WIATR
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY, 30 Mickiewicza Av., 30-059 Cracow, Poland
ACC CYFRONET AGH, 11 Nawojki St., 30-950 Cracow, Poland

Real time 8K video quality assessment using FPGA

Abstract

This paper presents a hardware architecture of the video quality assessment module. Two different metrics were implemented on FPGA using modern High Level Language for digital system design – Impulse C. FPGA resources consumption of the presented module is low, which enables module-level parallelization. Tests conducted for four modules working concurrently show that 1.96 GB/s throughput can be achieved. The module is capable of processing 8K video stream in a real-time manner i.e. 30 frames/second. Such high performance of the presented solution was achieved due to the series of architectural optimization introduced to the module, such as reduction of data precision and reuse of various module components.

Keywords: video quality, video metrics, image processing, FPGA.

1. Introduction

Nowadays, in addition to traditional Quality of Service(QoS), Quality of Experience (QoE) poses a real challenge for Internet audiovisual service providers, broad-casters and new Over-The-Top (OTT) services. The churn effect is linked to QoE impact; the end-user satisfaction is real added value in this competition. However, QoE tools should be proactive and innovative solutions well adapted for new audiovisual technologies. Therefore, objective audiovisual metrics are frequently dedicated to monitoring, troubleshooting, investigating, and setting benchmarks of content applications working in real-time or off-line.

The current audiovisual quality metrics are, so called, Full-Reference (FR), Reduced-Reference (RR) and No-Reference (NR) metrics for models standardized as ITU-T Recommendations. Most of the models have some limitations as they were usually validated using one of the following hypotheses: frame freezes lasting up to 2 seconds; no degradation at the beginning or at the end of the video sequence; no skipped frames; clean video reference (no spatial or temporal distortions); minimum delay supported between video reference and video (sometimes with constant delay); and up or down-scaling operations not always taken into account [1].

In the past, metrics based on three historical video artifacts (Blockiness, Jerkiness, Blur) were sufficient to provide an efficient predictive result. Consequently, most models are based on measuring these artifacts for producing a predictive Mean Opinion Score (MOS). Related research in [2] addresses measuring multimedia quality in mobile networks with an objective parametric model.

The concept proposed here, partly based on the framework for the integrated video quality assessment published in [3], which is able to isolate and focus investigation, set up algorithms, increase the monitoring period and guarantee better prediction. Depending on the technologies used in audiovisual services, the impact of QoE can change completely. The classical metric cannot provide pertinent predictive scores with certain new audiovisual artifacts such as exposure distortions. Moreover, it is important to detect the artifacts as well as the experience described and detected by the consumers. In realistic situations, when the video quality decreases in audiovisual services, customers can call a helpline to describe the annoyance and visibility problems for describing the outage; they are not required to provide a MOS. As such, the proposed concept is completely in phase with user experience. There are many possible reasons for video disturbance, and they can arise at any point along the video chain transmission (filming stage to end-user stage).

2. Video quality assessment

This paper addresses a challenging task of building a module capable of accelerating the metrics computations. Consequently, the designed module produces video quality assessment in a real-time for each video frame. The selected two metrics were implemented in hardware: blockiness and interlace.

The choice of the metrics was driven by their performance and hardware implementation feasibility. The authors designed and implemented a single module for both metrics. Such an approach enables hardware units sharing among the metrics architectures and it boosts the overall throughput of the video assessment quality module.

Blocking is the most visible image and video degradation of all artifacts. The effect is caused by all block-based coding techniques. It is a well-known fact that all compression techniques divide the image into small blocks and then compress them separately. Due to the coarse quantization, the correlation among blocks is lost, and horizontal and vertical borders appear as presented in Figure 1. The metric of the blocking artifact is calculated for pixels at boundaries of 8 × 8 blocks. Its value depends on two factors: magnitude of color difference at the blocks boundary, and picture contrast near boundaries [4].

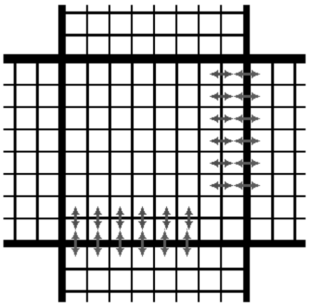


Fig. 1. Model of the video coding block

Interlace distortion is the result of special video compression where each frame is a connection between two frames in the original video. The method relies on using odd rows of pixels from odd frames and even rows from even frames. This compression reduces the amount of data transmitted since the rows of pixels are used twice during the time they were used in the uncompressed video. Interlace is noticeable as the difference between consecutive lines, and it becomes more distracting when the video includes motion (Fig. 2).

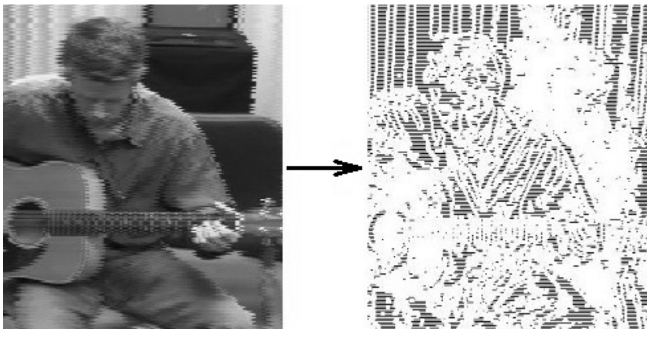


Fig. 2. Detection of interlace artifact

The authors came up with their own interlace metric. Interlace assessment is carried out in a block-by-block fashion of a single frame. A single block is 4×4 pixels denoted in the remaining part of the paper as micro-block. Every column in each micro-block is examined to determine a relationships between the neighboring pixels.

The sign of a difference of luminance is taken into account. A given micro-block is classified as an interlaced micro-block if the luminance of every pixel in the first row is bigger (smaller) than that of the neighboring pixel in the second row. The same is true for pixels in the third row and the pixels of the second and fourth row.

3. Impulse C implementation of the module

The module was implemented on Pico M503 platform [5]. Communication between a CPU and the FPGA is realized with 8 line PCIe, maximal data width is 128 bits.

Impulse C is a high level language based on Stream-C compiler. There are several features of the Impulse C language, which in the authors' view are superior to other currently used HLLs. It allows designers to reuse their HDL code by providing mechanisms, which facilitate the incorporation of existing modules. Furthermore, three different architectures are supported: combinational, pipelined and asynchronous, which cover a complete range of existing design scenarios. C compatibility makes it easy for software engineers to switch from GPP (General Purpose Processor) programming to FPGA design, as well as provide a platform for software-hardware integration within one design environment. Finally, Impulse C comes with a range of PSP (Platform Support Package), which provides a communication interface between the FPGA and GPP computational nodes. PSPs usage provides portability of an application across different platforms.

Video quality assessment module consists of 3 subblocks:

- Producer - reads video data from a file and sends it to the vqFPGA block using the InputStream.
- vqFPGA - reads data from the InputStream, executes video quality metrics assessment and sends results to the Consumer process using the OutputStream.
- Consumer - reads data from the OutputStream, analyses it and sends them to standard output stream.

The width of the Input and the Output stream i 128 bits - it is maximum width of Pico M503 platform stream. The scheme described above is parallelized four times in the real module - there are four producer, vqFPGA and consumer processes. Every Impulse C project is composed of a software and hardware part, so is the video quality assessment module.

The software part is composed of three functions: producer, consumer and the main program function which is used to launch the FPGA-based accelerator and all the application-related threads. It programs FPGA with a bit file and sends data to hardware, fetches the results.

The producer function opens the hardware stream (InputStream) and the video file. InputStream is 128-bit wide thus it is recommended to organize the data in such chunks so as the best possible throughput is achieved. Data is divided into microblocks which contain 16 8-bit data each. The process of sequential transfer of microblocks for a sample frame composed of 24 blocks is presented in Figure 3.

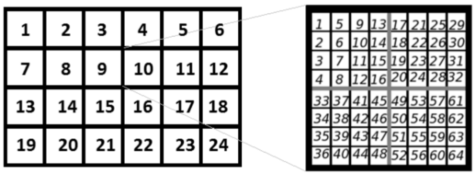


Fig. 3. Structure of a sample block and corresponding transfer sequence of a sample frame (left) and the order of pixels within each block (right)

The hardware part is composed of quality metric assessment modules and the additional hardware which handles data fetching and sending results to the software part. The hardware part, similarly to the software part, is equipped with two streams which are open before the data transfer is conducted and closed once it is finished. The resolution of the examined video is sent as the first frame and the module starts to process data in the infinite loop. Every 128-bit input word is read and divided into 16 chunks of 8-bit lenght which are the values of data in the microblock. As the next step, the counter is incremented and the results are sent to OutputStream. The module registers are reset after all the microblocks of a given frame are processed and a new frame comes in.

Hardware implementation of the blockiness algorithm involves calculating Intersums and Intrasums for the neighboring blocks. Such an approach guarantees an access to all the data required for the computations. It is worth noting that the most computationally important sections are located between blocks (block borders). This approach maximizes data usage and results in the best performance. The pixel shift operation (Figure 4) was introduced in order to optimize the memory access by reducing the number of data stored in BRAMs (Block RAM) in FPGA.

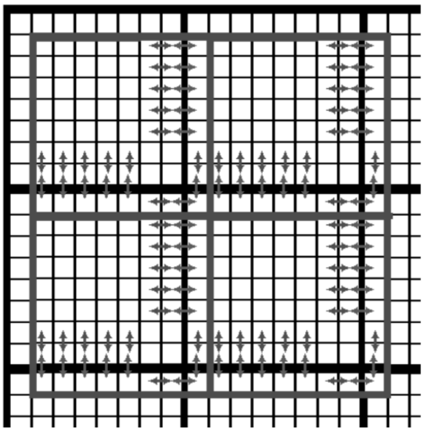
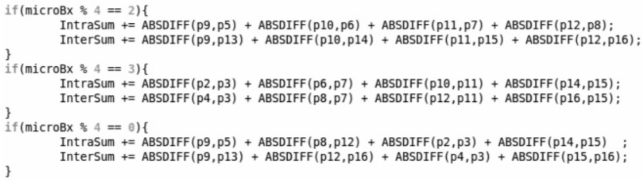


Fig. 4. Pixel shift operation

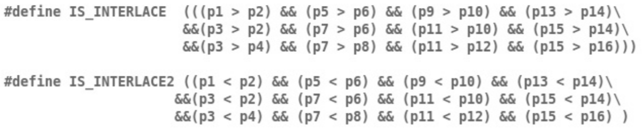
The source code presented in Figure 5 shows hardware implementation of the blockiness metric. Due to the efficient data serialization, the module is implemented with only nine lines of code which also results in low hardware resources consumption.



```
if(microBx % 4 == 2){
        IntraSum += ABSDIFF(p9,p5) + ABSDIFF(p10,p6) + ABSDIFF(p11,p7) + ABSDIFF(p12,p8);
        InterSum += ABSDIFF(p9,p13) + ABSDIFF(p10,p14) + ABSDIFF(p11,p15) + ABSDIFF(p12,p16);
}
if(microBx % 4 == 3){
        IntraSum += ABSDIFF(p2,p3) + ABSDIFF(p6,p7) + ABSDIFF(p10,p11) + ABSDIFF(p14,p15);
        InterSum += ABSDIFF(p4,p3) + ABSDIFF(p8,p7) + ABSDIFF(p12,p11) + ABSDIFF(p16,p15);
}
if(microBx % 4 == 0){
        IntraSum += ABSDIFF(p9,p5) + ABSDIFF(p8,p12) + ABSDIFF(p2,p3) + ABSDIFF(p14,p15)  ;
        InterSum += ABSDIFF(p9,p13) + ABSDIFF(p12,p16) + ABSDIFF(p4,p3) + ABSDIFF(p15,p16);
}
```

Fig. 5. Blockiness metric implementation

The way data is structured and transferred between the hardware and software part is adopted to this particular metric and improves the performance of the module. A single microblock is sent and the interlace detection is conducted just by examining IS_INTERLACE and IS_INTERLACE2 conditions (Figure 6).



```
#define IS_INTERLACE  (((p1 > p2) && (p5 > p6) && (p9 > p10) && (p13 > p14)\
                      &&(p3 > p2) && (p7 > p6) && (p11 > p10) && (p15 > p14)\
                      &&(p3 > p4) && (p7 > p8) && (p11 > p12) && (p15 > p16)))

#define IS_INTERLACE2 ((p1 < p2) && (p5 < p6) && (p9 < p10) && (p13 < p14)\
                     &&(p3 < p2) && (p7 < p6) && (p11 < p10) && (p15 < p14)\
                     &&(p3 < p4) && (p7 < p8) && (p11 < p12) && (p15 < p16) )
```

Fig. 6. Implementation of interlace metric

If one of those conditions is met, the result of the metric is incremented by 1. The sum of all the microblocks of a frame is a maximum possible value of the result which affected the choice of a variable used to store it i.e. co_uint32. After all the microblocks of the frame are received the variable is reset. The module is composed of 12 interconnected comparators which form a single huge XNOR gate. In addition, the module comprises an adder and 32-bit shift register for metricInterlace variable.

4. Experiment results

Experiments were conducted to determine the real performance of the module. The tests were carried out for several video resolutions:

- QVGA (320x240)
- VGA (640x480)
- FullHD (1920x1080)
- 4K (4096x2048)
- 8K (8192x4096)

The results were compared with the software solution for reference. As a real time framerate limit we chose 30 fps. For lower resolutions both software and hardware achieved required goals easily. However for 4K resolution, the software solution did not meet the real time requirements while the FPGA module gave satisfactory framerates for 8K resolution.

Tab. 1. Framerate reached by the software and hardware solutions

Resolution	Software, fps	Hardware, fps
FullHD	110	740
4K	23	195
8K	5	45

The hardware resources consumption of Pico platform as a function of the number of concurrent modules implemented is presented in Table 2, which also contains the achieved throughput.

Tab. 2. Hardware resources consumption (Xilinx Virtex-6LX240T FPGA)

# modules	# registers	# LUTs	Throughput, GB/s
1	23982 (7%)	11836 (6%)	0.66
2	29179 (10%)	15915 (7%)	1.27
3	34182 (12%)	19330 (9%)	1.60
4	39379 (15%)	23214 (11%)	1.96

5. Conclusions

This paper presents the implementation of video quality assessment metrics as a hardware module on the Pico M503 FPGA-based platform. The module was implemented in High Level Language: Impulse C. This significantly reduced the design time, facilitated the system integration process and enabled architectural optimization which boosted the overall performance of the solution. Consequently, the module is capable of processing video stream in a real-time manner. Due to the low resources consumption, it is possible to increase the performance of the system by migrating from the Pico M503 platform to the more advanced one in the future. It is worth noting that such a migration is straightforward and requires only changing Impulse C PSP (Platform Support Package). Examining compressed videos is an interesting research task and we may address it in our works to come. We would like to thank the reviewer for this remark. Since most of the video streams are compressed, the point made by the reviewer is even more important.

6. References

[1] ITU recommendations: http://www.itu.int/ITU-T/recommendations /index.aspx?ser=J [Accessed: 08.04.2016]

[2] Gustafsson J., Heikkila G., Pettersson M.: Measuring multimedia quality in mobile networks with an objective parametric model. Image Processing, 15th IEEE International Conference on Image Processing, 2008.

[3] Mu M., Romaniak P., Mauthe A., Leszczuk M., Janowski L., Cerqueira E.: Framework for the integrated video quality assessment, Multimedia Tools and Applications, 2012.

[4] Romaniak P., Janowski L., Leszczuk M., Papir Z.: Perceptual quality assessment for H.264/AVC compression. 2012 IEEE Consumer Communications and Networking Conference, 2012.

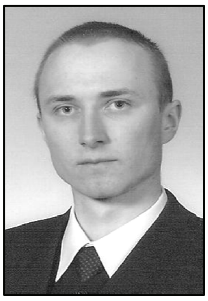
[5] Pico computing M503 manual: http://picocomputing.com/products/ hpc-modules/m-503/ [Accessed: 08.04.2016]

Received: 11.03.2016 Paper reviewed Accepted: 02.05.2016

Maciej WIELGOSZ, MSc, PhD, eng.

He received M.Sc. and Ph.D. degrees in electronic engineering in 2005 and 2010 respectively from the AGH University of Science and Technology, Kraków, Poland. He currently works in Academic Computing Centre CYFRONET AGH and University of Science and Technology. His research interests include hardware acceleration, text mining and hardware architectures for artificial intelligence.

e-mail: wielgosz@agh.edu.pl



Marcin PIETROŃ, MSc, PhD, eng.

He received MSc degree in electronic engineering and in computer science in 2003 and PhD in 2013 from the AGH University of Science and Technology, Cracow, Poland. He currently works in Academic Computing Centre CYFRONET AGH and University of Science and Technology. His research interests include parallel computing, automatic parallelization and data mining.

e-mail: pietron@agh.edu.pl

Michał KARWATOWSKI, MSc, eng.

He received the BSc eng. and MSc degrees in electronic engineering in 2013 and 2014 respectively from the AGH University of Science and Technology, Cracow, Poland. Currently student on PhD studies. His research interests include usage of hardware accelerators in complex computations, control and energy efficient systems of small and big scale, mainly using Field-Programmable Gate Arrays.

e-mail: mkarwat@agh.edu.pl

Prof. Kazimierz WIATR, PhD

Received the MSc and PhD degrees in electrical engineering from the AGH University of Science and Technology, Cracow, Poland, in 1980 and 1987, respectively, and the DSc degree in electronics from the University of Technology of Łódź in 1999. Received the professor degree in 2002. His research interests include design and performance of dedicated hardware structures and reconfigurable processors employing FPGAs for acceleration computing. He currently is a director of Academic Computing Centre CYFORNET AGH.

e-mail: wiatr@agh.edu.pl