

# Performance Optimization and FPGA Implementation of Real-Time Tone Mapping

Vladan Popovic, *Student Member, IEEE*, Elieva Pignat, and Yusuf Leblebici, *Fellow, IEEE*

**Abstract**—This brief analyzes the performance of the hardware-based tone mapping operators for compression of high dynamic range images. The bottlenecks of a tone mapping system are determined and a high-performance field programmable gate array (FPGA) implementation of an operator is introduced. The operator utilizes polynomial mapping technique, adaptive to the pixel values; hence preserving high contrast areas. The technique is further optimized for the presented resource-efficient FPGA implementation. We show that the timing optimization does not reduce the image quality, by obtaining high peak signal-to-noise-ratio of the resulting images. The timing comparison to the similar implementations shows 2.5 times increase in the achieved throughput, irrespective of the hardware platform.

## I. INTRODUCTION

Modern digital cameras are still not able to capture the full dynamic range of natural scenes, *i.e.* the ratio between intensity of the brightest and the darkest pixel. This results in underexposed or overexposed regions of the taken image and the lack of local contrast. Fig. 1 shows three shots taken under different exposure settings of the camera. The underexposed and overexposed images show fine details in very bright and very dark areas, respectively. These details cannot be observed in the moderately exposed image.

The increase of dynamic range and contrast enhancement are only a few of the methods used to create a realistic representation of a scene. High dynamic range (HDR) imaging technique was introduced to increase the dynamic range of a captured scene by encoding images with higher precision than standard 24-bit RGB. The most common method of obtaining HDR images is by taking several low dynamic range (LDR) photographs, all under different exposures. Such example is given in Fig. 1. Debevec and Malik [1] developed an algorithm for creating wide range radiance maps from multiple LDR images. The algorithm included obtaining camera response curve, creation of HDR radiance map and storage in RGBE format [2]. Alternatively, scenes can be represented using exposure fusion [3]. Exposure fusion is a pipelined process on LDR images and it is not necessary to create a large HDR image, which significantly reduces memory requirement. A similar principle is used for contrast enhancement using a single LDR image [4].

V. Popovic, E. Pignat and Y. Leblebici are with the Microelectronic Systems Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland. e-mail: vladan.popovic@epfl.ch.

This work has been partially funded by the Science and Technology Division of the Swiss Federal Competence Center Armasuisse.

Copyright © 2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org

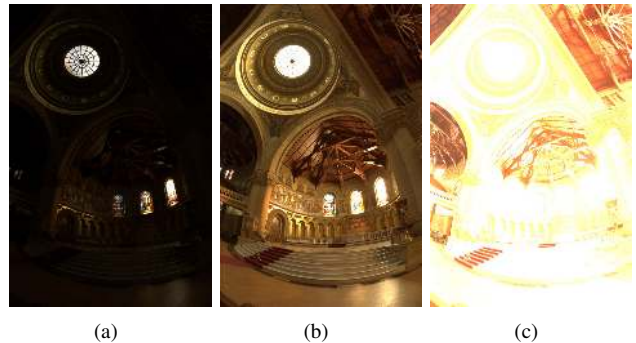


Fig. 1. Stanford Memorial church photographed with: (a) short, (b) medium and (c) long exposure time. Courtesy of Paul Debevec.

Apart from capturing natural scenes, another problem arises when displaying them. Current displays are limited to a very small dynamic range, and suffer from problems of displaying even standard LDR images. Thus, tone mapping operation [5] is introduced to map the real pixel values to the ones adapted to the displaying device. The purpose of tone mapping is to reduce the contrast in the HDR image, while preserving natural features of the scene. Due to characteristics of the human visual system (HVS), it is enough to tone map only the luminance component of the pixels.

Tone mapping operators can be divided into two main groups named global and local operators. Global operators are spatially invariant because they apply the same transformation to each pixel in the image. These algorithms usually have low complexity and high computational speed. One of the first complex global techniques was introduced by Ward et al. [5], which included histogram equalization and sensitivity of HVS to contrast in the image. Later, Pattanaik et al. [6] created a time-dependent operator based on HVS model and subjective experiments. The operator also takes into account color appearance and changes of color perception over time. One of the latest global techniques is based on adaptive logarithmic mapping and it was introduced by Drago et al. [7]. Even though it is considered as a global technique, it applies different mapping curves on pixels based on their luminosity. The curves vary from  $\log_2$  for the darkest pixels to  $\log_{10}$  for the brightest.

However, global operators do not preserve local contrast in the images where the luminance is uniformly occupying the full dynamic range. Oppositely, local operators are more flexible and adaptable to the image content, which may drastically improve local contrast in regions of interest [8]–[10]. Since they differently operate on different regions of the

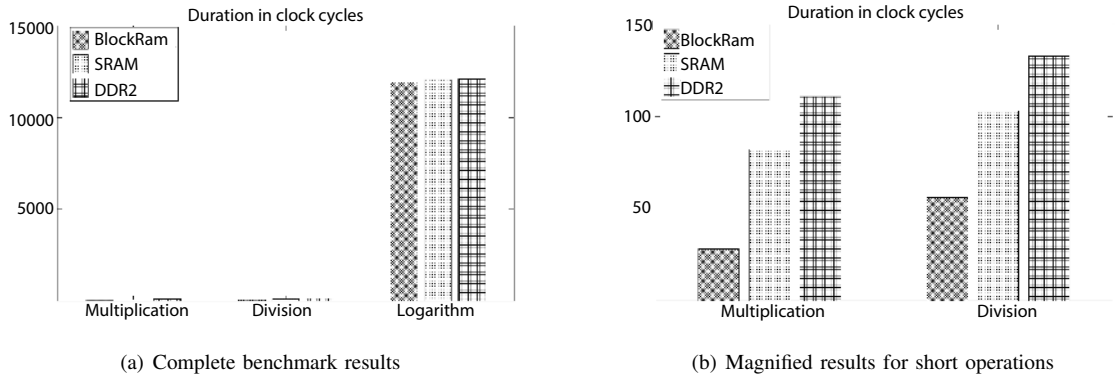


Fig. 2. Performance measurement results of the benchmark system. Duration is measured in clock cycles for three mathematical operations and three data storage mediums. The results show that (a) logarithm is the most process-intensive operation, and that (b) the relative influence of the storage medium is higher in faster operations.

image, they are significantly more expensive and resource-demanding. Hassan and Carletta [11] proposed an FPGA architecture for Reinhard operator [12], which introduces a local adaptation inspired by photographic film development in order to avoid halo artifacts. However, the proposed local adaptation requires a Gaussian pyramid decomposition, which requires a vast amount of resources, especially in terms of the required memory. Additionally, Hassan and Carletta [11] propose to estimate the logarithm function by the integer part of the operand and then refine it using the look-up table (LUT) with a fixed number of bits. A large number of bits should be considered in order to preserve the dynamic range, which enlarges the LUT size even further. A method with a lower number of bits reduces the precision and the dynamic range of the image.

Lapray et al. [13] presented a full imaging system with an FPGA as a processing core. Even though the used camera sensor streams 60 fps signal, the system provides only 30 fps output video for 1 Mpixel frame. The loss of frame rate is due to a calibration step needed before tone mapping and a slow non-pipelined implementation of computations, such as division and logarithm. The computational results are precise, but the cost is significantly reduced frame rate. Apart from FPGA systems, a system-on-chip (SOC) solution was presented by Chiu et al. [14]. ARM SOC platform was used for both global and local tone mapping processor. The achieved frame rate was 60 fps for a  $1024 \times 768$  pixels resolution frame.

A detailed comparison of the major tone mapping operators is published by Yoshida et al. [15]. The comparison was realized by human subjects grading several aspects of the constructed image, such as contrast, brightness, naturalness and detail reproduction. One of the best graded techniques in this review was the global operator by Drago et al. [7]. Therefore, this operator is taken as a base for development of the FPGA-suitable operator.

In this work, we analyze performance of the tone mapping procedure on 1080p60 ( $1920 \times 1080$ ) input and output images and determine the logarithm calculation as the main bottleneck of the system. We constrain (1) the image quality to be at least 8 bits per color channel; (2) the system frame rate to be higher

than 60 fps, and we aim to find a function that satisfies both constraints. We propose a polynomial approximation of the Drago operator, which overcomes the system's performance bottleneck, and show that this tone mapping function fits our constraints. An efficient FPGA implementation of the operator is presented together with the supporting prototype system. Our implementation exploits pipeline processing of the polynomial approximations, which increases performance compared to the previously used LUT-based methods. Furthermore, performance and image quality measurements show a significant increase in the achieved frame rate, without any visually perceivable loss of quality in the image.

## II. PROBLEM ANALYSIS

Similarly to the majority of the global operators, Drago operator uses logarithmic mapping function expressed in (1), where displayed luminance  $L_d$  is derived from the ratio of world luminance  $L_w$  and its maximum  $L_{max}$ . The algorithm adapts the mapping function by changing the logarithm base  $t$  as a function of the bias parameter  $b$  [7], as shown in (2). The base value is a function of the pixel luminance, and it is bounded on the interval  $[2, 10]$ .

$$L_d = \frac{\log_t(L_w + 1)}{\log_{10}(L_{max} + 1)} \quad (1)$$

$$t(b) = 2 + 8 \cdot \left( \frac{L_w}{L_{max}} \right)^{\frac{\ln b}{\ln 0.5}} \quad (2)$$

Even though this mapping is created for interactive applications, its speed is very low for video applications. The reported frame rate is below 10 frames per second (fps), even for  $720 \times 480$  pixels image, without any approximations that decrease the image quality [7].

In order to analyze complexity and timing performance of the tone mapping algorithm, we implemented a benchmark system on Xilinx ML501 Development Kit, which includes XC5VLX50 Virtex-5 FPGA, and both Zero-bus turnaround (ZBT) SRAM and DDR2 memory chips. The benchmark system included a MicroBlaze microprocessor, memory controllers for code and data storage, DVI (Digital Visual Interface) controller for display of the tone mapped image, and

timer for measuring performance. The tone mapping code was written in C and it was run on MicroBlaze.

The purpose of this benchmark is to determine the bottleneck of the system in terms of timing performance. We wanted to determine what is the most time-consuming operation, and what is the influence of the external memory on performance of this FPGA system. We benchmarked multiplication, division and logarithm, since they appear in Equations (1) and (2). Furthermore, operand data are stored in three different locations: internal BlockRAMs (BRAM), external SRAM, and external DDR2 memory. Hence, nine different cases were observed and measured. The instantiated timer measured the number of clock cycles needed for each operation, in each case. Thus, a performance analysis is independent of the clock frequency. The numbers are obtained by averaging 10000 measurements per each case.

Fig. 2(a) shows that the most time-consuming operation is logarithm, which is the bottleneck of the system. Furthermore, it shows that the relative improvement when the fast memories are used is negligible. Fig. 2(b) illustrates in more detail the duration of multiplication and division, which is two orders of magnitude lower than the duration of logarithm calculation. The dominant factor in these two cases is the memory access, as the duration can even triple when external DDR2 memory is used.

The analysis showed two main issues in tone mapping operation: (1) Fast calculations are significantly affected by external memory access, and (2) Long duration of the logarithm operation. In order to resolve these issues, we implemented the whole tone mapping as a hardware-only system. The system resembles an accelerating unit, with reduced access to the external memory and faster logarithm calculation.

### III. TONE MAPPING OPTIMIZATION

The dynamic range of the natural scenes reaches values as high as 400'000, according to [1] and [7]. Although an exact representation requires 19 bits, modern displays using DVI standard do not support more than 8 bits per color. We decided to use 16-bit calculations in order to achieve computational precision higher than 8 bits, with large enough error margin.

To avoid long calculation time and large LUTs, we have developed an optimized operator based on (1) and (2). Using the logarithm properties, we can change its base and calculate only natural and base-10 logarithms. Expressions in the form of  $\log(1+x)$  can be efficiently approximated by the Chebyshev polynomials of the first kind  $T_i(x)$  [16]. The approximation needs only six integer coefficients to achieve the desired 16-bit precision. The intermediate approximation step is given in (3), where  $L_{wa}$  is the world adaption luminance calculated as the log-average of all pixels' luminance.

$$L_d = \frac{\sum_{i=0}^5 c_e(i) T_i\left(\frac{L_w}{L_{wa}}\right)}{\log_{10}\left(\frac{L_{max}}{L_{wa}} + 1\right) \cdot \ln\left(2 + 8\left(\frac{L_w}{L_{max}}\right)^{\frac{\ln b}{\ln 0.5}}\right)} \quad (3)$$

The Chebyshev approximation can be applied to both natural and base-10 logarithm by only changing the coefficients. The

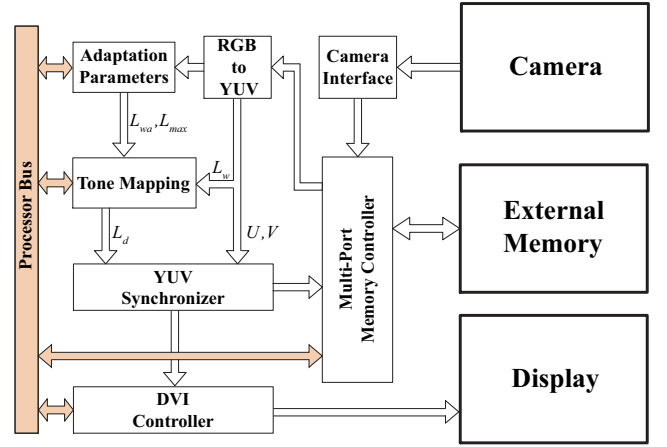


Fig. 3. Top-level architecture of the implemented system. Camera, memory and display are external to the FPGA. Internal architecture of the tone mapping and adaptation parameters blocks is shown in Fig. 4. MicroBlaze is connected as a master to the Processor Bus, but it is not shown in the diagram.

coefficients for the natural logarithm are denoted as  $c_e(i)$ , while  $c_{10}(i)$  are for base-10 in (4). According to [7], the best visually perceived results are obtained for the bias parameter  $b \approx 0.85$ . We have fixed this parameter to  $b = 0.84$ , to simplify the hardware implementation, as generic power functions are difficult to implement. The exponent in the denominator becomes 0.25 and the argument can be evaluated by two consecutive calculations of the square root. The square root is also approximated using the Chebyshev polynomials.

$$L_d = \frac{\sum_{i=0}^5 c_e(i) T_i\left(\frac{L_w}{L_{wa}}\right)}{\sum_{i=0}^5 c_{10}(i) T_i\left(\frac{L_{max}}{L_{wa}}\right) \cdot \ln\left(2 + 8\left(\frac{L_w}{L_{max}}\right)^{\frac{1}{4}}\right)} \quad (4)$$

The natural logarithm term in the denominator cannot be precisely approximated by Chebyshev polynomial, due the arguments much higher than 1. A suitable approximation of the expression  $\ln x$  is a fast convergence form of the Taylor series, which is expressed in (5). This expression needs only three non-zero coefficients to achieve a sufficient 16-bit precision, but the argument should be preconditioned as shown in (5). The world adaptation luminance  $L_{wa}$  is calculated as the log-luminance average of all  $N$  pixels. Thus, it can be calculated using Taylor approximation:

$$\ln x = 2 \sum_{k=1}^3 \frac{1}{2k-1} \left(\frac{x-1}{x+1}\right)^{2k-1} \quad (5)$$

$$L_{wa} = \frac{1}{N} \sum_N \ln(L_w) \quad (6)$$

The equations (4)-(5) describe the optimized tone mapping operator suitable for hardware implementation. The set of required mathematical operations is reduced to only addition, multiplication and division.

### IV. FPGA IMPLEMENTATION

The internal FPGA architecture of the system is shown in Fig. 3. Camera block represents an acquisition device, which

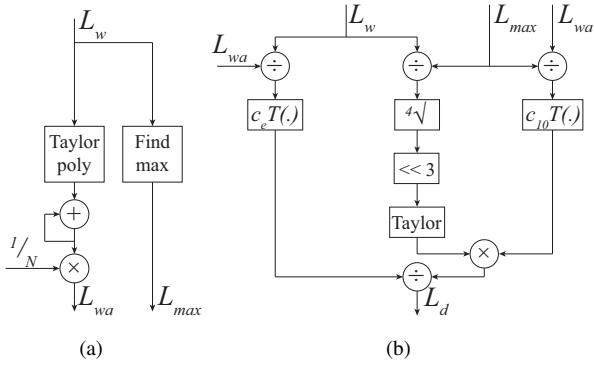


Fig. 4. Internal architecture of (a) adaptation parameters calculator, (b) tone mapping operator. Taylor and Chebyshev polynomials are evaluated using pipelined Horner scheme.

can be a single HDR sensor streaming video signal [13] or a single LDR camera taking multiple shots under different exposure settings. In both cases, the Camera Interface outputs HDR images that are written to the memory. Since the main goal of this work is to implement a tone mapping operator, the camera block is emulated and it streams a pre-calculated HDR radiance map [1] and stores it in DDR2. The pixel values are read from DDR2 and transformed into YUV color system.

The tone mapping implementation consists of two parts: finding the adaptation parameters ( $L_{max}$  and  $L_{wa}$ ) and tone mapping curve implementation.  $L_{max}$  and  $L_{wa}$  must be calculated before starting the core tone mapping operation. Since the presented system processes the HDR video stream, the parameters are determined based on the previous frame, under the assumption that the scene illumination does not vary faster than response time of the HVS. The parameters are updated at the end of each frame.

The block diagram of a subsystem for finding the adaptation parameters is shown in Fig. 4(a). The problem of finding  $L_{max}$  consists of finding the maximum value in a sequence of read luminances. The world adaptation luminance is calculated by accumulating Taylor approximation evaluations and averaging them by the total number of pixels in the frame.

Fig 4(b) presents the block diagram of the tone mapping function. The Chebyshev polynomials approximating logarithm and square root are evaluated using the pipelined Horner scheme, and the division block is implemented using the fast Anderson algorithm [16]. The Taylor approximation blocks in Fig. 4(a) and Fig. 4(b) follow the implementation given in Algorithm 1. Taylor approximation of the logarithm given in (5) is accurate only in the range  $[0, 1]$ . However, logarithm argument in the denominator of expression (4) is in the range  $[2, 10]$ . Thus, the argument is scaled down to the  $[0, 1]$  range. The scaling factor is determined by the location of the first “1” in the 16-bit fixed-point representation. Additionally, the argument is preconditioned according to (5) in order to achieve fast convergence. The polynomial is evaluated using the pipelined implementation of the Horner scheme. The scheme requires both zero and non-zero coefficients  $q(k)$  to be provided; hence, the processing pipeline of (5) comprises six stages instead of three. The polynomial is scaled back to

#### Algorithm 1 Taylor polynomial of $\ln x$

```

1:  $y := position\_of\_the\_first\_1(x)$ 
2:  $x_{scaled} := x / 2^y$  { % Scale to  $[0,1]$  range }
3:  $t := \frac{x_{scaled}-1}{x_{scaled}+1}$  { % Precondition for fast convergence }
4: { % Horner scheme }
5:  $temp(6) := 0$ 
6: for  $k = 5$  downto  $0$  do
7:    $poly(k) := temp(k+1) + q(k)$ 
8:    $temp(k) := poly(k) \cdot t$ 
9: end for
10:  $result := poly(0) + y \cdot \ln 2$  { % Bring to original range }

```

the original range after the evaluation.

The output  $L_d$  of the tone mapping block is synchronized with the corresponding chrominance values in the YUV Synchronizer. The tone mapped image can be directly shown on a screen, or written back to the memory.

## V. EXPERIMENTAL RESULTS

The tone mapping system was implemented on ML501 Development Kit with Xilinx XC5VLX50FFG676 Virtex-5 FPGA. The maximum operational frequency reported by the synthesis tool is 214.27 MHz and the utilization summary is given in Table I. The summary presents the utilization of a single tone mapping block, and the full system which includes DDR2 and DVI controllers. As a comparison, the utilization summary from [13] is also given, since the same FPGA family is used. Our implementation instantiates more DSP blocks, which are used for multiplication, whereas utilization of all the other resources is significantly reduced.

The comparison of the proposed implementation in terms of achievable pixel throughput is given in Table II. Comparison data are calculated from the reported values in original publications. The systems in the comparison were developed on different platforms. In order to have a fair comparison, we additionally synthesized our design for Stratix-II and for UMC 180 nm technology node, and obtained throughput results from post-place-and-route simulations. The comparison results show approximately 2.5 times improvement over the currently best implementation. The advantage of our implementation is

TABLE I  
FPGA DEVICE UTILIZATION SUMMARY

Resources	Tone Mapping	Full System	[13]
Slice LUTs	918	4536	14168
Slice Registers	540	5036	8132
BlockRAM/FIFO	0	8	40
DSP48Es	30	30	4

TABLE II  
PIXEL THROUGHPUT COMPARISON. VALUES ARE IN MPIXELS/S

Platform / Node	GPU	Virtex-5		Stratix-II		UMC 180	TSMC 130
System	[7]	Our	[13]	Our	[11]	Our	[14]
Throughput	13	133	39	111	47	124	47





Fig. 5. Examples of the tone mapped HDR images using (a) our fast implementation, and (b) Drago operator [7]. The reduction in quality is not noticeable in the images when the fast implementation is used. Radiance maps courtesy of Paul Debevec and Raanan Fattal.

TABLE III  
HDR IMAGE QUALITY MEASUREMENTS

	<i>House</i>	<i>Synagogue</i>	<i>Cathedral</i>	<i>Memorial</i>
PSNR [dB]	59.7	73.46	60.79	74.15
SSIM	0.9990	0.9999	0.9996	0.9995

the fully pipelined operation, which reduces the critical path of the system. Furthermore, our implementation allows larger image resolutions if a lower frame rate is acceptable.

Visual quality testing was applied on a set of HDR images provided by Debevec [1] and Fattal [9]. These images were stored into DDR2 and loaded twice by the tone mapping block. Maximum and log-average luminance are calculated in the first “pass”, and the tone mapping is realized after. Screenshots of four tone mapped examples are shown in Fig. 5(a). The results show that the proposed approximations and hardware implementation do not introduce visual differences compared to the results of [7] which are shown in Fig. 5(b).

Apart from visual appearance comparison, an objective image quality comparison was performed. Peak signal-to-noise-ratio (PSNR) and Structural Similarity Index (SSIM) [17] values are calculated and shown in Table III. The tone mapped images in Fig. 5(b) are taken as the reference images. SSIM values almost equal to 1 confirm that the structure of the scene’s objects is not affected. High PSNR values show that the objective image quality is high, despite lower computational time when the proposed fast implementation is used. Moreover, the PSNR is high enough that the difference in images cannot be visually noticed on the standard LDR screens.

## VI. CONCLUSION

In this paper, we proposed an optimized global tone mapping operator based on Drago operator. The operator is suitable for high frame rate operation and the presented resource-efficient FPGA implementation. The operator compresses luminance component of the HDR image using Taylor and Chebyshev polynomials, since the number of needed coefficients for high precision calculation is low. Horner scheme is used for evaluation of the polynomials and it is implemented as a fully pipelined operation, resulting in high performance.

High operating frequency and increased frame rate make this algorithm and implementation an excellent choice for real-time and video HDR applications.

## REFERENCES

- [1] P. E. Debevec and J. Malik, “Recovering High Dynamic Range Radiance Maps from Photographs,” in *ACM SIGGRAPH 97*, New York, NY, USA, 1997, pp. 369–378.
- [2] G. Ward, *Graphics Gems II*. San Diego, CA, USA: Academic Press, 1991, ch. Real Pixels, pp. 80–83.
- [3] T. Mertens, J. Kautz, and F. Van Reeth, “Exposure Fusion,” in *Pacific Conf. on Computer Graphics and Applications*, 2007, pp. 382–390.
- [4] A. Saleem, A. Beghdadi, and B. Boashash, “Image fusion-based contrast enhancement,” *EURASIP Journal on Image and Video Processing*, vol. 2012, no. 10, 2012.
- [5] G. Ward, H. Rushmeier, and C. Piatko, “A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes,” *IEEE Trans. Vis. Comput. Graphics*, vol. 3, no. 4, pp. 291–306, Oct. 1997.
- [6] S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg, “Time-dependent visual adaptation for fast realistic image display,” in *ACM SIGGRAPH 00*, New York, NY, USA, 2000, pp. 47–54.
- [7] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, “Adaptive Logarithmic Mapping For Displaying High Contrast Scenes,” *Computer Graphics Forum*, vol. 22, no. 3, pp. 419–426, 2003.
- [8] F. Durand and J. Dorsey, “Fast Bilateral Filtering for the Display of High-Dynamic-Range Images,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, Jul. 2002.
- [9] R. Fattal, D. Lischinski, and M. Werman, “Gradient Domain High Dynamic Range Compression,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 249–256, Jul. 2002.
- [10] R. Mantiuk, S. Daly, and L. Kerofsky, “Display adaptive tone mapping,” *ACM Trans. Graph.*, vol. 27, no. 3, Aug. 2008.
- [11] F. Hassan and J. E. Carletta, “An FPGA-based architecture for a local tone-mapping operator,” *Journal of Real-Time Image Processing*, vol. 2, no. 4, pp. 293–308, 2007.
- [12] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, “Photographic Tone Reproduction for Digital Images,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 267–276, Jul. 2002.
- [13] P.-J. Lapray, B. Heyrman, M. Rosse, and D. Ginhac, “HDR-ARTiSt: High Dynamic Range Advanced Real-time Imaging System,” in *IEEE Int. Symp. on Circuits and Systems*, 2012, pp. 1428–1431.
- [14] C.-T. Chiu, T.-H. Wang, W.-M. Ke, C.-Y. Chuang, J.-R. Chen, R. Yang, and R.-S. Tsay, “Design optimization of a global/local tone mapping processor on arm SOC platform for real-time high dynamic range video,” in *IEEE Int. Conf. on Image Processing*, 2008, pp. 1400–1403.
- [15] A. Yoshida, V. Blanz, K. Myszkowski, and H.-P. Seidel, “Perceptual Evaluation of Tone Mapping Operators with Real-World Scenes,” in *SPIE Human Vision & Electronic Imaging X*, 2005, pp. 192–203.
- [16] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 3rd ed. Berlin, Germany: Springer-Verlag, 2007.
- [17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–312, Apr. 2004.