Mantissa-Exponent-Based Tone Mapping for Wide Dynamic Range Image Sensors

Jie Yang, Ulian Shahnovich, and Orly Yadid-Pecht, *Fellow, IEEE*

***Abstract*—The dynamic range of a scene is defined as the ratio between the maximum and minimum luminance in it. Wide dynamic range (WDR) means this ratio is so large that it exceeds the dynamic range of a traditional image sensor. Nowadays, WDR image sensors enable the capture of WDR scenes. However, the captured WDR image requires an additional tone mapping step to compress the high bit pixel of WDR image to low rate pixel so that it can be displayed on the screen. The tone mapping algorithm is mostly done in an image signal processor or with a specific software application. This brief proposes a tone mapping technique that is suitable for direct processing of the output of a WDR image sensor bitstream. The algorithm acquires statistics on the mantissa and exponent parts of the pixel value and then generates a refined histogram for tone mapping. Experiments that evaluate the image quality and hardware efficiency are carried out. The results indicate that the proposed mantissa exponent-based algorithm provides visually pleasing results and preserves details of the original WDR image better than other similar algorithms. The hardware resources' efficiency of the algorithm makes the system on chip implementation possible.**

***Index Terms*—Wide dynamic range, tone mapping, image sensor, mantissa exponent representation.**

I. Introduction

The DYNAMIC range is defined as the ratio of the intensity of the brightest point to the intensity of the darkest point in a scene or image. A typical image sensor has a dynamic range between 60-70 dB. However, the dynamic range of a real scene can go beyond 120 dB so that it exceeds the capture capability of an image sensor. To capture a wide dynamic range (WDR) scene, one can take multiple images with different exposures and fuse these images to form an image. However, this is an indirect approach which highly requires stability and also time-consuming. A direct way to acquire WDR image is to use a WDR image sensor with an extended dynamic range. Various solutions have been proposed in recent years to achieve this purpose. Logarithmic response image sensor is a common approach for dynamic range extension [1], [2]. Multimode sensors such as [3]–[5] can operate as conventional linear pixels at low illumination conditions; whereas, at high illuminations, they operate as companding pixels. Capacitance adjustment sensors use capacity well adjustment method to extend the dynamic range [6]. Dual capture image sensor can two different integration times to gain a higher dynamic range than conventional sensors [7]. Although these sensors can have a very high dynamic range, they suffer from a remarkable loss of sensitivity which affects the image quality [8]. Image sensors such as [9]–[12] which can autonomously control over the integration time and reset pixels, show great performance in terms of noise reduction and detail preservation [8].

After the WDR image is acquired, one needs to tone map the high bit pixel value to a low bit value so that it can be displayed on the screen, because the conventional screens can only show 8-bit depth images. The process of compressing a WDR image for display is called tone mapping. Tone mapping algorithms are often refereed as tone mapping operators (TMOs) and they can be classified into two categories as global TMO and local TMO. A global tone mapping process is to apply a single global function to all pixels in the image where identical pixels will be given an identical output value within the range of the display devices. Local tone mapping algorithms take local intensity statistics into account and are generally good at preserving details. However, local TMOs can produce 'halo' and other unpleasant artifacts which greatly affect the image quality [13]. In fact, some researches report that human observers prefer global TMO to local TMO [14], [15].

The acquisition and display of the WDR are often regarded as two separate problems and are solved in two different systems, namely an imaging system and a signal processing system. Most research focus on either the hardware implementation of WDR image sensor or the tone mapping algorithm development. However, a simple combination of the two systems can hardly be applied to real-time WDR video processing applications due to various issues such as data transmission bottle-neck, the long processing time delay caused by the algorithm and high cost bought by the CPU or GPU based processing system. Recently, there is some research that realizes tone mapping algorithms on a system on chip (SoC), which make real-time low-cost WDR video processing possible [16]–[20]. In this brief, we present a global tone mapping algorithm which is based on mantissa-exponent processing. Unlike the traditional tone mapping algorithms that manipulate the pixel values directly, the proposed algorithm operates on the mantissa and exponent of the pixels. It reduces the bit-width required for storage. The algorithm needs only two histograms based on the mantissa and exponent

AQ1 Manuscript received September 12, 2018; revised November 27, 2018 and January 14, 2019; accepted February 25, 2019. This brief was recommended by Associate Editor M. Chrzanowska-Jeske. *(Corresponding author: Jie Yang.)*

AQ2 AQ3 The authors are with the University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: jie.yang2@ucalgary.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2019.2903101

which further reduces the required logic resources. The two histograms give more statistical information about the pixel distribution to help tone mapping. Mathematical approximation is used in FPGA implementation to achieve real-time WDR processing while maintaining high resource efficiency.

The rest of this brief is organized as follows: Section II briefly introduces the mantissa exponent WDR image sensor. Section III describes the proposed approach for tone mapping. Section IV presents the hardware implementation. Section V analyzes the experimental results, followed by conclusion in Section VI.

II. Mantissa-Exponent Based WDR Image Sensor

Image sensors usually give integer outputs for the captured pixel intensities with certain bit-width ranging from 9 to 12 bits. The output luminance value is determined by the following equation:

$$m = f(\Delta t * I) \tag{1}$$

where Δt is the exposure time and I is the luminance intensity. $\Delta t * I$ is the integration procedure of the image sensor. f is the response function of the sensor which ideally has a linear characteristic. m is the output value. However, for a certain exposure time Δt, if the light intensity is large, the integration result of $\Delta t * I$ could exceed the range of the response function and cause saturation. Mantissa exponent WDR image sensor was first introduced in 1999 [21] and has been developed over years [9]–[12]. It can autonomously reset the integration process when $\Delta t * I$ reaches a certain threshold. For example, if any pixel detects $\Delta t * I$ is larger than the threshold, it would reset itself and redo the integration using an exposure time that equals to $\Delta t/2$. If the integration result $\Delta t/2 * I$ still reaches the threshold, the pixel will reset again and reduce the exposure time to $\Delta t/4$ to recalculate integration. This reset process repeats e times until $\Delta t/(2^e) * I$ is smaller than the threshold. For a pixel that resets itself e times, we can have

$$m = f(\Delta t/2^e * I) \tag{2}$$

Since response function f is considered linear, we multiple 2^e on both sides of the equation

$$m * 2^e = f(\Delta t * I) \tag{3}$$

The output of each pixel when under the same exposure time Δt will be represented in a mantissa exponent format where m is the output of the last integration and e is the number of resets.

This mantissa exponent representation has its unique advantage in extending the dynamic range and reducing the memory requirements. For example, $m \in [0, 1023]$ can be coded with 10 bits and $e \in [0, 7]$ can be coded with 3 bits, then Eq. (1) can represent a huge range from 0 to 130944 with a total number of only 13 bits.

III. The Proposed Algorithm

This mantissa exponent output naturally represents the pixel intensity in the logarithmic domain. It is a great advantage because the human visual system has a logarithmic response

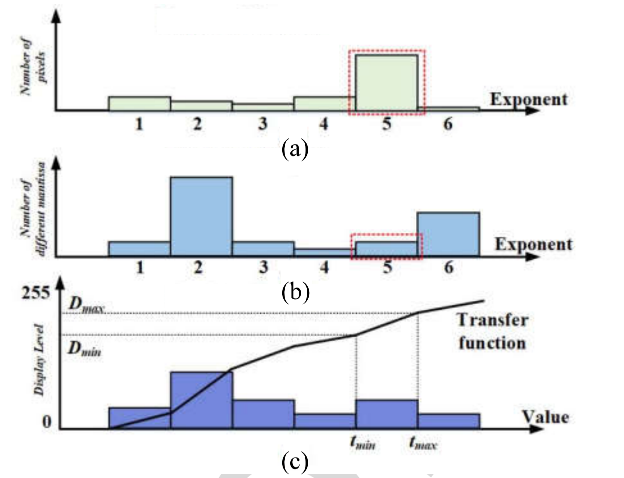

Fig. 1. Histograms generated based on the mantissa-exponent representation. (a) Histograms that calculates the number of pixels of that have the same exponent value. (b) Histogram that calculate the number of different mantissas. (c) Histogram fused from (a) and (b) and the corresponding tone mapping curve.

to the light intensity [22]. The exponent values act as labels that segment the captured images in the logarithmic domain. An illustration is shown in Fig. 1 (a). The histogram H_E counts the number of pixels that have the same exponent values. $H_E(i)$ represents the number of pixels with exponent value that equals to i. Intuitively, a higher $H_E(i)$ value in this histogram means there are more pixels that have this exponent value, hence they have a bigger chance to be more informative. To better preserve the information, we should give these pixels more discrete display levels during tone mapping. A histogram such as Fig. 1 (a) can give us a big picture of how the pixel intensities are distributed. For example, from Fig. 1 (a), we can tell that most pixels of the image are bright. However, there are chances that a WDR image contains a large part of background such as bright or dim sky; in such situations, a histogram solely based on the exponent distribution would misguide us about which segment is more important. To overcome this problem, we make another histogram H_M where each bin $H_M(i)$ counts the number of different mantissae when the pixel exponent value equals to i. Fig. 1 (b) shows an example. If a large number of pixels have values that close to each other, they would have the same exponent value and several different mantissa values. This will give a corresponding low $H_M(i)$ value despite the fact that the number of pixels is larger. In contrast, there are chances that pixel values are more scattered where only a few pixels have the same mantissa values. This will give a corresponding higher $H_M(i)$ value. Compared to the traditional histogram based tone mapping [23] where there is only one histogram is used, the proposed approach includes two histograms that not only gives density distribution information using H_E but also gives further dispersion information using H_M. For example, in Fig. 1 (a) and Fig. 1(b), the dashed red rectangles show the 5-th bins of the two histograms. $H_E(5)$ is much larger than $H_M(5)$, which means there are many pixels in this region but most of them have the same mantissa values. These pixels could belong to a

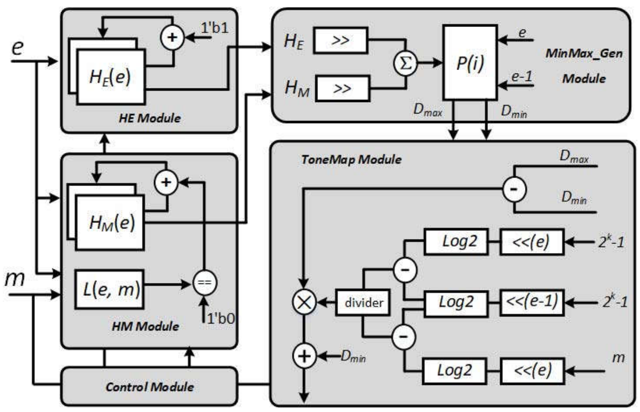

Fig. 2. Implemented hardware architecture.

Algorithm 1 HM Module Function

Input: mantissa m; exponent e;

Output: Histogram H_M

1: **if** $L(e, m) == 1'b0$ **then**

2: $\quad L(e, m) = 1'b1$

3: $\quad H_M(e) = H_M(e) + 1$

4: **end if**

Algorithm 2 ToneMap Module Function

Input: D_{max}, D_{min}, e, m

Output: Pixel value d

1: $t_{min} = 2^{e-1} \times (2^K - 1)$

2: $t_{max} = 2^e \times (2^K - 1)$

3: $d = \frac{log_2(m \times 2^e) - log_2(t_{min})}{log_2(t_{max}) - log_2(t_{min})} \times (D_{max} - D_{min}) + D_{min}$

uniform bright background. During tone mapping, we ideally want to give these pixels fewer display levels. To achieve this goal, we first normalize the two histograms $H_E(i)$ and $H_M(i)$ to $H'_E(i)$ and $H'_M(i)$ and then combine them to form a new histogram H_t using the following equation:

$$H_t(i) = \alpha * H'_E(i) + (1 - \alpha) * H'_M(i), \quad i = 0, 1, 2 \ldots N \quad (4)$$

where α is a weight factor between 0 and 1. It balances between the two histograms so that combined histogram is not biased.

An accumulative probability function can then be defined as

$$P(i) = \sum_{j<i} H_t(j) \quad (5)$$

Then the accumulative probability function is used to tone map WDR image pixels to an LDR display level. Fig. 1 (c) shows the balanced histogram of Fig. 1 (a) and Fig. 1 (b), the 5-th bin is reduced because of the balanced effect of Eq. (4). During tone mapping, the resulting piece-wise linear function will give fewer display level to the pixels in the 5-th bin.

IV. Hardware Design

The hardware design is shown in Fig. 2. It mainly consists of five modules: *HM Module*, *HE Module*, *Control Module*, *MinMax_gen Module* and the *ToneMap Module*. The WDR image sensor outputs mantissa value m, exponent value e and control signals to the *HM Module*, *HE Module* models and *Control Module* simultaneously. The logic operations of *HE Module* can be expressed as $H_E(e)$++ where a memory will store the exponent histogram H_E. Memory content will be automatically increased by one if address e is indexed. Considering an $M \times N$ resolution image sensor outputs X bits exponent per pixel, we will need $2^X * \lceil log2(M * N) \rceil$ bits memory. The functionality of *HM Module* is similar to *HE Module*, and it can be expressed in Algorithm 1.

L is a memory that records if pixel with exponent value e and mantissa m has been accounted or not. H_M is the mantissa histogram and it is recorded by another memory. $L(e, m)$ will be set to 1 if exponent value e and mantissa m are recorded for the first time and $H_M(e)$ value will be increased by one each time it is indexed. Considering a pixel has X bits exponent and Y bits mantissa, the total number of the required register bits for the L register are $2^X \times 2^Y$ and $2^X * \lceil log2(M * N) \rceil$ for the register H_M.

To implement the proposed algorithm, one needs to store the entire frame to extract the histograms. Under such circumstance, significant time and resource consuming are inevitable. To overcome the problem, we use the histogram obtained from the last frame to tone map the current frame. This is because without any exaggeration changes in a scene, there is tiny variation between the statistics of successive image frames. In the hardware design, we duplicate memories in *HM Module* and *HE Module*, and the two identical memory sets are used interchangeably to record the histogram of the last and current frame. The two sets of registers are switched by the output signal of the *Control Module*.

The H_E and H_M values are read by the *MinMax_gen module* to compute $H_t(i)$ of Eq. (4) and the accumulative probability function $P(i)$ of Eq. (5). The normalization of $H'_E(i)$ and $H'_M(i)$ can be easily performed by shift operations because the total number of pixels is usually a power of 2. The *MinMax_gen module* will output two values *Dmax* and *Dmin* which are the two end values of the piece-wise function on the y-axis (shown in Fig. 1 (c)).

The *ToneMap module* implements the corresponding piece-wise linear function. Its function is described in Algorithm 2. It takes the current exponent value e, mantissa value m and the output of *MinMax_gen module*, D_{max}, D_{min} as input to generate the tone mapped pixel value d. It first computes two end values of the piece-wise function on the x-axis t_{min} and t_{max} (shown in Fig. 1 (c)). In line 1 and 2 of Algorithm 2, K is a fixed value which represents the bit-width of the mantissa value. The tone mapped pixel value is calculated based on the equation in line 3. To reduce hardware resources, the log computation in the *ToneMap module* is approximated with Taylor expansion. Considering the convergence range for Taylor expansion, we divide the pixel value into a fractional part which is smaller than 1 and a multiplicative factor which is an order of 2

$$log_2(x) = log_2(\frac{x}{2^N} * 2^N) = log_2(\frac{x}{2^N}) + N, \ 2^N \geq x \quad (6)$$

If we change the base of the natural logarithm, we get the following:

$$log_2(\frac{x}{2^N}) = log(\frac{x}{2^N})/log(2) \tag{7}$$

Using Taylor expansion for natural logarithm, Eq. (7) is approximated with:

$$((\frac{x}{2^N} - 1) - \frac{1}{2} \times (\frac{x}{2^N} - 1)^2 + \epsilon) \times (1 + \frac{1}{2} - \frac{1}{16}) \tag{8}$$

ϵ is the higher order terms of Taylor expansion and it is omitted during computation. The maximum error is less than 6.8×10^{-2}. $1/log(2)$ is approximated by using $(1 + 1/2 - 1/16)$ with an approximation error of 5.2×10^{-3}. It can be implemented with simple shift operations instead of division.

V. Experimental Results

The proposed hardware design for the proposed mantissa-exponent based tone mapping algorithm was modeled in Verilog HDL and synthesized on an Altera Cyclone III FPGA (EP3C120F780) which is fabricated under 60 nm technology and contains about 120 K logic elements. We have compared this brief with other three works, namely Ambalathankandy *et al.* [17], Vylta *et al.* [18], and Hassan and Carletta [20]. Ambalathankandy *et al.* proposed a tone mapping algorithm and the hardware implementation. Vylta *et al.* implemented the gradient domain tone mapping algorithm [24], Hassan and Carletta implemented the Reinhard algorithm [25]. We first evaluated the tone mapped image quality using an objective metric, and then compared the hardware efficiency.

Tone mapping quality index (TMQI) [26] is an algorithm that is used to evaluate the performance of tone mapping algorithms. It calculates the structural similarity and naturalness of the tone mapped image and combines them to give an overall quality index. We apply the four different tone mapping algorithms to the same WDR images and compare the TMQI scores. However, most of the existing WDR images are in 'hdr' or 'exr' format where each pixel is stored as a floating point value. To obtain an integer format for testing and simulate the sensor output, we linearly mapped the minimum and maximum of the WDR images to 1 and 13499 with 5% margin. The pixels are then transferred to mantissa and exponent representation. The mantissa has 10 bits and the exponent has 3 bits. For [17], we used the code provided by the author. For the other two methods, we used the code from the HDR Toolkit by Artusi *et al.* [27] to realize the corresponding algorithms. All algorithms use the default parameter setting. Our algorithm has a free parameter α to balance the two calculated histograms. We find that α value between 0.4 and 0.6 usually presents good results. Hence, we choose $\alpha = 0.5$ in the following experiment. Fig. 3 shows two example images tone mapped with different algorithms. The proposed algorithm produces images with better brightness and contrast when compared to the other three works. The TMQI values are listed in Table I. The proposed algorithm achieves the highest TMQI value for the two images. We tested 200 WDR images from various sources including the HDReye dataset [28], the companion disk of [29] and some other sources. The obtained average TMQI value is listed in Table II. The proposed algorithm gains the highest average TMQI value among all four algorithms.

Our hardware target is processing WDR image sensors that have 1024×768 resolution and output 10 bits mantissa and 3 bits exponent. The synthesized working clock frequency of our hardware implementation is 100 MHz. We compare the hardware usage with other works and show the result in Table III. Although [18] and [20] are implemented in two different Altera FPGAs other than the Cyclone III which is used in both this brief and [17], the hardware resource consumption results are all measured by the number of the standard Altera logic elements and memory bits. Compared to the other three implementations that use complicated calculations, the core computation of our implementation is just recording two histograms. Moreover, the logarithmic

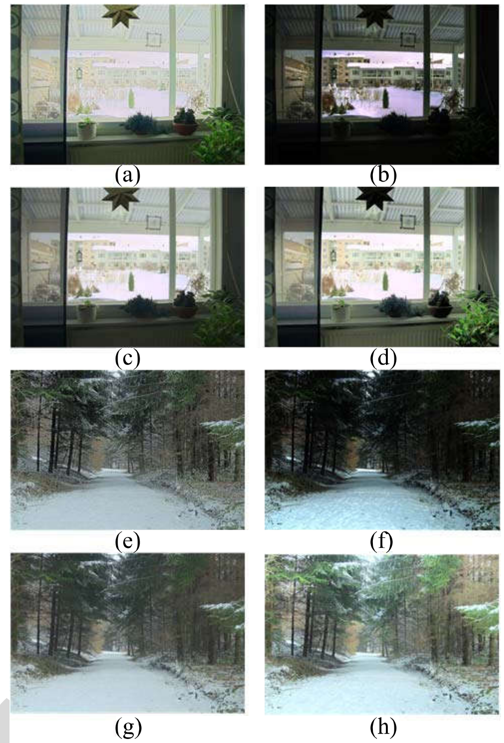

Fig. 3. Tone mapped images using different algorithms. (a, e) image tone mapped using Ambalathankandy *et al.* method [17]. (b, f) image tone mapped using Vylta *et al.*'s method [18]. (c, g) image tone mapped using Hassan and Carletta's method [20]. (d, h) image tone mapped with the proposed algorithm.

TABLE I
TMQI Scores of the Images of Fig. 3

Image	Ambalathan-kandy *et al.* [17]	vylta *et al.* [18]	Hassan *et al.* [20]	Proposed
Image 1	0.8858	0.7967	0.8798	**0.9256**
Image 2	0.8561	0.8663	0.8920	**0.8992**

TABLE II
Average TMQI Scores for 200 WDR Images

Algorithm	Ambalathan-kandy *et al.* [17]	vylta *et al.* [18]	Hassan *et al.* [20]	Proposed
Average TMQI	0.7985	0.7960	0.8826	**0.9111**

TABLE III
COMPARISON WITH OTHER HARDWARE IMPLEMENTATIONS

Works	Image size	FPS	Logic elements	Memory (bits)
Hassan et al. [20]	1024×768	60	34,806	3,153,048
Vytla et al. [18]	1 Megapixel	100	9019 + 88 DSP	307,200
Ambalathankandy [17]	1024×768	126	93,989	87,176
This work	1024×768	126	15,471	107,408

computation using Taylor approximation further reduces logic resources. Hence, our implementation consumes the least amount of logic elements. However, the implementation uses more memory bits than [17] because the HM module and HE need large memory to record the histograms. As we have stated in the previous section, if the number of exponent bits is 3, and the number of mantissa bits is 10, and the image resolution 1024×768. The required bits for HE and HM module are $2 \times \{2^3 * log2(1024 * 768) + 2^3 * 2^{10} + 2^3 * log2(1024 * 768)\}$, which add up to 17024 bits. All memories were compiled using the Altera memory IP core. Due to the memory consumption of the two histograms, our total memory usage is a bit higher than [17].

VI. CONCLUSION

In this brief, a tone mapping algorithm and hardware implementation are proposed. The algorithm takes advantage of the mantissa exponent representation to build two histograms. The two histograms are used to calculate a piece-wise linear transfer function that is used for mapping the mantissa-exponent values to display. A hardware design that implements the algorithm is also proposed. Our experiments evaluate and compare the image quality of different algorithms with objective metric. The results indicate that our algorithm can generate images with a better quality. Hardware implementation assessment shows that our algorithm acquires smaller hardware resources when compared to other similar works.

REFERENCES

[1] S. Kavadias *et al.*, "A logarithmic response CMOS image sensor with on-chip calibration," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1146–1152, Aug. 2000.
[2] Y. Ni, Y. Zhu, and B. Arion, "A 768×576 logarithmic image sensor with photodiode in solar cell mode," in *Proc. Int. Image Sensor Workshop*, 2011, pp. 1–4.
[3] G. Storm *et al.*, "Extended dynamic range from a combined linear-logarithmic CMOS image sensor," *IEEE J. Solid-State Circuits*, vol. 41, no. 9, pp. 2095–2106, Sep. 2006.
[4] S. Vargas-Sierra, G. Liñán-Cembrano, and Á. Rodríguez-Vázquez, "A 151 dB high dynamic range CMOS image sensor chip architecture with tone mapping compression embedded in-pixel," *IEEE Sensors J.*, vol. 15, no. 1, pp. 180–195, Jan. 2015.
[5] M. Bae *et al.*, "A linear-logarithmic CMOS image sensor with adjustable dynamic range," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5222–5226, Jul. 2016.
[6] S. Decker, D. McGrath, K. Brehmer, and C. G. Sodini, "A 256×256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 2081–2091, Dec. 1998.
[7] J. Lee, I. Baek, and K. Yang, "Memoryless wide-dynamic-range CMOS image sensor using nonfully depleted PPD-storage dual capture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 1, pp. 26–30, Jan. 2013.
[8] A. Spivak, A. Belenky, A. Fish, and O. Yadid-Pecht, "Wide-dynamic-range CMOS image sensors—Comparative performance analysis," *IEEE Trans. Electron Devices*, vol. 56, no. 11, pp. 2446–2461, Nov. 2009.
[9] A. Fish, A. Belenky, and O. Yadid-Pecht, "Wide dynamic range snapshot APS for ultra low-power applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 11, pp. 729–733, Nov. 2005.
[10] A. Belenky, A. Fish, A. Spivak, and O. Yadid-Pecht, "Global shutter CMOS image sensor with wide dynamic range," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 12, pp. 1032–1036, Dec. 2007.
[11] A. Spivak, A. Belenky, A. Fish, and O. Yadid-Pecht, "A wide-dynamic-range CMOS image sensor with gating for night vision systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 2, pp. 85–89, Feb. 2011.
[12] A. Spivak, A. Belenky, and O. Yadid-Pecht, "Very sensitive low-noise active-reset CMOS image sensor with in-pixel ADC," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 10, pp. 939–943, Oct. 2016.
[13] K. Kim, J. Bae, and J. Kim, "Natural HDR image tone mapping based on Retinex," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1807–1814, Nov. 2011.
[14] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi, "Evaluation of HDR tone mapping methods using essential perceptual attributes," *Comput. Graph.*, vol. 32, no. 3, pp. 330–349, 2008.
[15] X. Cerdá-Company, C. A. Parraga, and X. Otazu, "Which tone-mapping operator is the best? A comparative study of perceptual quality," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 35, no. 4, pp. 626–638, 2018.
[16] U. Shahnovich, A. Hore, and O. Yadid-Pecht, "Hardware implementation of a real-time tone mapping algorithm based on a mantissa-exponent representation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2016, pp. 2210–2213.
[17] P. Ambalathankandy, A. Horé, and O. Yadid-Pecht, "An FPGA implementation of a tone mapping algorithm with a halo-reducing filter," *J. Real Time Image Process.*, pp. 1–17, Sep. 2016. AQ4
[18] L. Vytla, F. Hassan, and J. E. Carletta, "A real-time implementation of gradient domain high dynamic range compression using a local Poisson solver," *J. Real Time Image Process.*, vol. 8, no. 2, pp. 153–167, 2013.
[19] V. Popovic, E. Pignat, and Y. Leblebici, "Performance optimization and FPGA implementation of real-time tone mapping," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 10, pp. 803–807, Oct. 2014.
[20] F. Hassan and J. E. Carletta, "An FPGA-based architecture for a local tone-mapping operator," *J. Real Time Image Process.*, vol. 2, no. 4, pp. 293–308, 2007.
[21] O. Yadid-Pecht and E. Fossum, "CMOS APS with autoscaling and customized wide dynamic range," in *Proc. IEEE Workshop Charge Coupled Devices Adv. Image Sensors*, vol. 3650, 1999, pp. 48–51. AQ5
[22] S. Hecht, "The visual discrimination of intensity and the Weber–Fechner law," *J. Gen. Physiol.*, vol. 7, no. 2, pp. 235–267, 1924.
[23] G. W. Larson, 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./Dec. 1997.
[24] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 249–256, 2002.
[25] 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, 2002.
[26] H. Yeganeh and Z. Wang, "Objective quality assessment of tone-mapped images," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 657–667, Feb. 2013.
[27] A. Artusi, F. Banterle, K. Debattista, and A. Chalmers, *Advanced High Dynamic Range Imaging: Theory and Practice*. Boca Raton, FL, USA: CRC Press, 2011.
[28] Y. Dong, E. Nasiopoulos, M. T. Pourazad, and P. Nasiopoulos, "High dynamic range video eye tracking dataset," in *Proc. 2nd Int. Conf. Electron. Signal Process. Commun.*, Athens, Greece, 2015, pp. 56–59.
[29] E. Reinhard *et al.*, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Amsterdam, The Netherlands: Morgan Kaufmann, 2010.