A Fixed-Point Tone Mapping Operation for HDR Images in the RGBE Format

Toshiyuki DOBASHI*, Tatsuya MUROFUSHI*, Masahiro IWAHASHI† and Hitoshi KIYA*,
*Tokyo Metropolitan University, Tokyo, 191-0065, Japan
†Nagaoka University of Technology, Niigata, 980-2188, Japan

***Abstract*—A fixed-point tone mapping operation (TMO) instead of a floating-point TMO is proposed in this paper. A TMO generates a low dynamic range (LDR) image from a high dynamic range (HDR) image. Since pixel values of an HDR image are generally expressed in a floating-point data format, e.g. RGBE and OpenEXR, a TMO is also implemented in floating-point arithmetic in conventional approaches. However, it requires a huge computational cost, even though a resulting LDR image is expressed in simple integers. The proposed TMO is implemented with only fixed-point arithmetic. As a result, the proposed method reduces a computational cost. Experimental results shows the PSNR of LDR images in the proposed method are comparable to those of the conventional methods.**

I. Introduction

High dynamic range (HDR) images are rapidly spreading from field of computer graphics and photographic to the other fields, such as car-mounted camera and medical imaging recently. In contrast, display devices which can handle wide dynamic range of pixel values in HDR images are not still popular. Thus, a tone mapping operation (TMO) is important to compress dynamic range of HDR images, in order to be treated with covnentional display devices.

So far, various researches on TMOs have been done. Most of these are focused on finding tone mapping function which suitable for human visual system [1]–[4]. Recently, some researches reduced the cost for transmitting HDR images by combining with data compression technologies [5]–[8]. Unlike those previous researches, this paper discusses on 'resources' for a TMO such as a computational cost for light implementation of tone mapping.

In general, it is important to reduce a computational cost in image processing, including tone mapping which discussed here. Heavy demand for computation is continuously increasing, e.g, large variety of color depth, a huge size of images, and resolution of display devices. In particular, resources on embedded systems, such as camera, are limited. Therefore, it is still necessary to study the way to implement signal processing under limited resources for economic reason, even though faster machines appear in the future. A TMO especially requires heavy resources, since it is generally composed of floating-point arithmetic for an HDR data format such as RGBE and OpenEXR [2].

A fast and flexible TMO has been proposed in [9]. Visibility and contrast were simply controlled with a single parameter. However it does not directly contribute to reducing resources. The global tone mapping in [1], [2] has been widely used due to its simplicity. However, a function for this TMO is limited to a specific one. Moreover, a function itself is just a part of whole TMO.

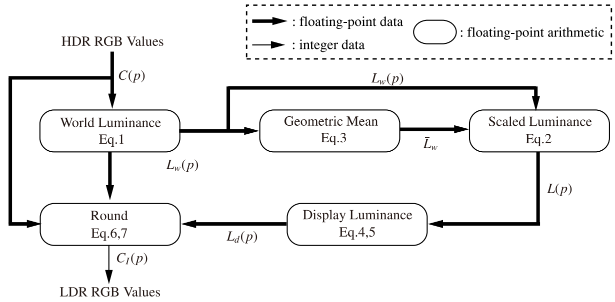

Fig. 1. The outline of the photographic tone reproduction [1].

Unlike those approaches, the method which is based on integer values of tone mapping has been proposed in [10]. Considering not only a function itself but also whole procedure of a TMO, this method try to resolve the essential problem on high demand of resources. In this method, any kind of functions can be utilized as a global tone mapping. Moreover, it is implemented with 'integer' input and 'integer' output. It offers almost the same result of tone mapping as a conventional method under reduced resources of computation. However, internal arithmetic of this method is still floating-point.

The proposed method is based on [10], but it consists of fixed-point arithmetic, that is, it can be implemented without floating-point arithmetic. By this feature, the method can be executed under limited resources, such as processors without floating-point number processing unit (FPU). The experiment confirms that the proposed method offers high quality of tone mapped images comparable to conventional methods.

II. Photographic Tone Reproduction

The procedure of a tone mapping is described in this section. It generates an integer low dynamic range (LDR) image from a floating-point HDR image. This section describes one of well-known tone mapping procedures [1], and Figure 1 shows the outline of it. First, it calculates world luminance $L_w(p)$ of a pixel p as

$$L_w(p) = 0.27R(p) + 0.67G(p) + 0.06B(p), \tag{1}$$

where $R(p)$, $G(p)$, and $B(p)$ are floating-point RGB values of the input HDR image. Next, the scaled luminance $L(p)$ is calculated as

$$L(p) = k \cdot \frac{L_w(p)}{\bar{L}_w}, \tag{2}$$

where $k \in [0, 1]$ is a parameter called "key value." $\bar{L}_w$ denotes geometric mean of world luminance $L_w(p)$, and it is defined as

$$\bar{L}_w = \exp\left(\frac{1}{N}\sum_p \log_e\left(L_w(p)\right)\right), \tag{3}$$

where N is the total number of pixels in the input HDR image. Note that Eq. (3) has singularity due to zero value of $L_w(p)$. It is avoided by introducing a small value in [1]. However, its arbitrariness is not negligible for pixel values in a floating point format, since its pixel value is also small. Therefore, in [10], only non-zero values are included in the geometric mean. Next, display luminance $L_d(p)$ is computed with a tone mapping function $y()$ as

$$L_d(p) = y(L(p)), \tag{4}$$

where Reinhard's global operator [1] is specified as

$$y_{\text{Reinhard}}(L(p)) = \frac{L(p)}{1 + L(p)}. \tag{5}$$

Finally, floating-point LDR pixel value $C_F(p)$ is derived as

$$C_F(p) = L_d(p) \cdot \frac{C(p)}{L_w(p)}, \tag{6}$$

where $C(p) \in \{R(p), G(p), B(p)\}$ is the floating-point RGB value of input HDR image, and $C_F(p) \in \{R_F(p), G_F(p), B_F(p)\}$. Moreover, 8-bit integer LDR value $C_I(p)$ is generated as

$$C_I(p) = \text{round}\left(C_F(p) \cdot 255\right), \tag{7}$$

where $\text{round}(x)$ rounds x to the nearest integer value and $C_I(p) \in \{R_I(p), G_I(p), B_I(p)\}$.

III. Proposed method

A. Floating-point pixel value in the RGBE format

This method deals with the RGBE format as an example of a floating-point data format of HDR images. Each pixel is 32 bits long in this format. It consist of 8 bit common exponent and 8 bit mantissa for each RGB channel [2]. Its exponent $F_E(p)$ and mantissa $F_M(p)$ for floating-point value $F(p)$ are calculated by

$$F_E(p) = \lceil \log_2 F(p) + 128 \rceil, \tag{8}$$

$$F_M(p) = \left\lfloor F(p) \cdot 2^{136 - F_E(p)} \right\rfloor, \tag{9}$$

where $0 \leq F_E(p) \leq 255$ and $0 \leq F_M(p) \leq 255$. In the equation above, $\lceil x \rceil$ rounds x to the nearest integer greater than or equal to x, and $\lfloor x \rfloor$ rounds x to the nearest integer less than or equal to x. In this format, the original floating-point value $F(p)$ is represented as

$$F(p) = \frac{F_M(p) + 0.5}{256} \cdot 2^{F_E(p) - 128}. \tag{10}$$

B. Integer TMO

The proposed method is based on the integer TMO [10]. In this section, the integer TMO [10] is described. Figure 2 shows the outline of this method. This method is designed for HDR images in the RGBE format. In the RGBE format,

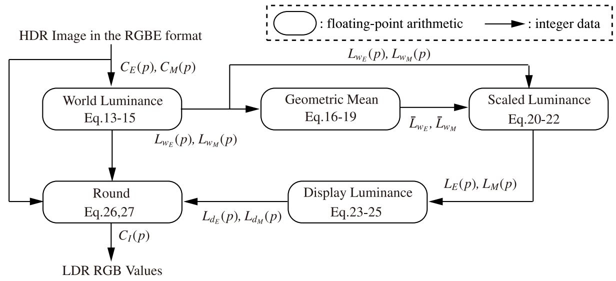

Fig. 2. The outline of the integer TMO [10].

the common exponent $C_E(p)$ and each RGB value mantissa $C_M(p)$ are calculated as

$$C_E(p) = \lceil \log_2\{\max(R(p), G(p), B(p)\} + 128 \rceil, \tag{11}$$

$$C_M(p) = \left\lfloor C(p) \cdot 2^{136 - C_E(p)} \right\rfloor, \tag{12}$$

where $0 \leq C_E(p) \leq 255$, $0 \leq C_M(p) \leq 255$, and $C_M(p) \in \{R_M(p), G_M(p), B_M(p)\}$. If $\max\left(R(p), G(p), B(p)\right) < 10^{-38}$, this method sets $C_E(p) = C_M(p) = 0$ [2]. And if $C_M(p) = 256$, this method sets $C_M(p) = 255$. This method treats $C_E(p)$ and $C_M(p)$ as integer numbers.

First, the integer values pair for world luminance $L_w(p)$ is calculated; exponent $L_{w_E}(p)$ and mantissa $L_{w_M}(p)$ of world luminance $L_w(p)$ are given as

$$L_{w_E}(p) = \lceil \log_2(ML(p) + 0.5) + C_E(p) - 8 \rceil, \tag{13}$$

$$L_{w_M}(p) = \left\lfloor (ML(p) + 0.5) \cdot 2^{C_E(p) - L_{w_E}(p)} \right\rfloor, \tag{14}$$

$$ML(p) = 0.27R_M(p) + 0.67G_M(p) + 0.06B_M(p), \tag{15}$$

where $0 \leq L_{w_E}(p) \leq 255$ and $0 \leq L_{w_M}(p) \leq 255$. If $C_M(p) = 256$, this method sets $C_M(p) = 255$. Next, integer valued geometric mean is calculated; exponent $\bar{L}_{w_E}(p)$ and mantissa $\bar{L}_{w_M}(p)$ of geometric mean $\bar{L}_w$ are derived as

$$\bar{L}_{w_E} = \lceil SL_{w_M} + SL_{w_E} + 128 \rceil, \tag{16}$$

$$\bar{L}_{w_M} = \left\lfloor 2^{SL_{w_M} + SL_{w_E} - \bar{L}_{w_E} + 136} \right\rfloor, \tag{17}$$

$$SL_{w_E} = \frac{1}{N}\sum_p \left(L_{w_E}(p) - 136\right), \tag{18}$$

$$SL_{w_M} = \frac{1}{N}\sum_p \log_2\left(L_{w_M}(p) + 0.5\right), \tag{19}$$

where $0 \leq \bar{L}_{w_E} \leq 255$ and $0 \leq \bar{L}_{w_M} \leq 255$. Here, $\bar{L}_{w_E}$ and $\bar{L}_{w_M}$ are computed by only non-zero $L_{w_E}(p)$'s. Then, exponent $L_E(p)$ and mantissa $L_M(p)$ of scaled luminance $L(p)$ are calculated as

$$L_E(p) = \left\lceil \log_2(AL_w(p)) + L_{w_E}(p) - \bar{L}_{w_E} + 128 \right\rceil, \tag{20}$$

$$L_M(p) = \left\lfloor AL_w(p) \cdot 2^{136 + L_{w_E}(p) - L_E(p) - \bar{L}_{w_E}} \right\rfloor, \tag{21}$$

$$AL_w(p) = k \cdot \frac{L_{w_M}(p) + 0.5}{\bar{L}_{w_M} + 0.5}. \tag{22}$$

The method sets $L_E(p) = L_M(p) = 0$ if $L_E(p) < 0$, and $L_E(p) = L_M(p) = 255$ if $L_E(p) > 255$. That is, $0 \leq L_E(p) \leq 255, 0 \leq L_M(p) \leq 255$. Then, the method calculates exponent $L_{d_E}(p)$ and mantissa $L_{d_M}(p)$ of display luminance $L_d(p)$. This calculation depends on tone mapping functions. Here, tone mapping function of Eq. (5) is used as an example

$$L_{d_E}(p) = \lceil \log_2(FL(p)) + 128 \rceil, \tag{23}$$

$$L_{d_M}(p) = \left\lfloor FL(p) \cdot 2^{136 - L_{d_E}(p)} \right\rfloor, \tag{24}$$

$$FL(p) = \frac{L_M(p) + 0.5}{L_M(p) + 0.5 + 2^{136 - L_E(p)}}. \tag{25}$$

The method sets $L_{d_E}(p) = L_{d_M}(p) = 0$ if $L_{d_E}(p) < 0$, and $L_{d_E}(p) = L_{d_M}(p) = 255$ if $L_{d_E}(p) > 255$. That is, $0 \leq L_{d_E}(p) \leq 255, 0 \leq L_{d_M}(p) \leq 255$.

The final RGB value $C_I(p)$ of the LDR image is obtained as

$$C_I(p) = \text{round}\left(RL(p) \cdot 2^{C_E(p) + L_{d_E}(p) - L_{w_E}(p) - 136} \cdot 255\right), \tag{26}$$

$$RL(p) = \frac{(L_{d_M}(p) + 0.5)(C_M(p) + 0.5)}{L_{w_M}(p) + 0.5}. \tag{27}$$

C. Fixed-point arithmetic

The integer TMO [10] can be implemented with integer input and integer output. However, its internal arithmetic uses floating-point arithmetic. In the proposed method, these are implemented with fixed-point arithmetic to reduce required resources.

Most of equations can be calculated with integer and fixed-point arithmetic because each variable are expressed as integer. However, Eq. (25) is difficult to be calculated without floating-point arithmetic because the range of value of the denominator is very wide. Because of this, the method deforms Eq. (25) as follows

$$FL(p) = \frac{1}{1 + \frac{2^{136 - L_E(p)}}{L_M(p) + 0.5}}. \tag{28}$$

Furthermore, the method branches Eq. (28) into three cases and approximates it based on the power of two in the denominator as follows.

Case 1:If $136 - L_E(p) > 22$ in Eq. (28), '1' in the denominator can be ignored because the right part of the denominator is very large, and so it is approximated as

$$FL(p) = \frac{L_M(p) + 0.5}{2^{136 - L_E(p)}}, \tag{29}$$

$$L_{d_E}(p) = \lceil \log_2(L_M(p) + 0.5) - (136 - L_E(p)) + 128 \rceil, \tag{30}$$

$$L_{d_M}(p) = \left\lfloor (L_M(p) + 0.5) \cdot 2^{L_E(p) - L_{d_E}(p)} \right\rfloor. \tag{31}$$

Case 2:If $136 - L_E(p) < -16$ in Eq. (28), the right part of the denominator can be ignored because it is very small, and so it is approximated as

$$FL(p) = 1, \tag{32}$$

$$L_{d_E}(p) = 128, \tag{33}$$

$$L_{d_M}(p) = 255. \tag{34}$$

Case 3:Otherwise, it can be calculated with fixed-point arithmetic.

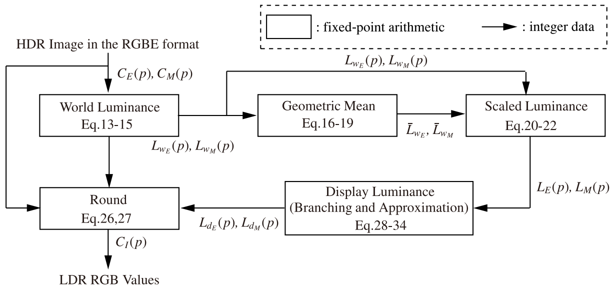

Fig. 3. The outline of the proposed method.

The method can calculate all equation of TMO with only fixed-point arithmetic by these branching and approximation. Note that the conventional method [1] consists of floating-point data and floating-point arithmetic. The integer TMO [10] consists of integer data and floating-point arithmetic. In contrast, the proposed method consists of integer data and fixed-point arithmetic. Figure 3 shows the outline of the proposed method.

IV. Experimental Results

To evaluate the proposed method, the method was implemented, and the experiment was carried out.

A. Conditions

This experiment compared the proposed method and the conventional method, in terms of the PSNR and the processing time.

LDR images produced by the proposed and the conventional methods [1], [10] were used in the PSNR comparison. This comparison experiment used 24 HDR images in the RGBE format.

The processing time of the proposed method and the conventional method [1] was compared in the processing time comparison. This comparison experiment performed tone mapping operation for the HDR image in the RGBE formt of 512 × 768 pixels 100 times, and calculated the average time. The experiment platform was Marvell PXA270 ARM processor 624MHz with 128MB RAM.

In both comparison experiment, the methods were implemented using C language. The proposed method worked with 32-bit fixed-point arithmetic. The conventional methods [1], [10] worked with 64-bit floating-point arithmetic.

B. PSNR comparison

Table I shows the peak signal-to-noise ratio (PSNR) between the proposed method and the conventional method [10] (integer TMO). Note that the internal arithmetic of the integer TMO is floating-point. From this table, every case indicates a high PSNR value, and the average PSNR was 66.2 dB. Therefore, the deterioration from the change from floating-point to fixed-point is small. Table II shows the PSNR between the proposed method and the conventional method with floating-point data and arithmetic [1]. From this table, the average PSNR was 56.1 dB. Note that all the arithmetic of the conventional method [1] is floating-point. Figure 4 shows LDR images obtained by the proposed method and the conventional

TABLE I
PSNR BETWEEN THE PROPOSED AND THE INTEGER TMO [10].

Image	PSNR[dB]	Image	PSNR[dB]
1	69.6	13	68.9
2	70.0	14	70.1
3	69.8	15	69.7
4	70.1	16	73.6
5	69.7	17	54.8
6	70.3	18	54.1
7	70.7	19	56.0
8	53.3	20	70.6
9	54.7	21	69.3
10	70.1	22	69.4
11	69.1	23	54.6
12	69.0	24	70.4

TABLE II
PSNR BETWEEN THE PROPOSED AND THE CONVENTIONAL METHOD WITH ALL FLOATING-POINT DATA AND ARITHMETIC [1].

Image	PSNR[dB]	Image	PSNR[dB]
1	56.7	13	56.0
2	56.5	14	58.1
3	56.3	15	56.7
4	56.6	16	55.0
5	57.5	17	55.0
6	55.2	18	55.0
7	58.2	19	55.4
8	54.9	20	55.5
9	55.9	21	56.2
10	56.3	22	55.3
11	57.3	23	54.7
12	56.9	24	55.7

method [1]. It indicates that it is impossible for human eyes to distinguish these two images. From the above results, it was confirmed that the proposed method has a high accuracy, even though it is with only fixed-point arithmetic.

C. Processing time comparison

Figure 5 shows the processing time of the proposed method and the conventional method. The proposed method was 23.1 times faster than the conventional method [1]. It was confirmed that the proposed method reduced a computational cost by using a fixed-point arithmetic.

V. CONCLUSION

A fixed-point tone mapping operation has been proposed in this paper. The proposed method can perform a TMO with only fixed-point arithmetic. The method calculates the equation which is difficult to calculate without floating-point arithmetic by branching and approximation. As a result, the method reduces a computational cost, and it can be executed under limited resources, such as processors without FPU. The experimental results confirmed that the proposed method has a high accuracy even though it is with only fixed-point arithmetic.

Conventional method [1]   Proposed method

Fig. 4. LDR image comparison (It is impossible to distinguish these two images.)

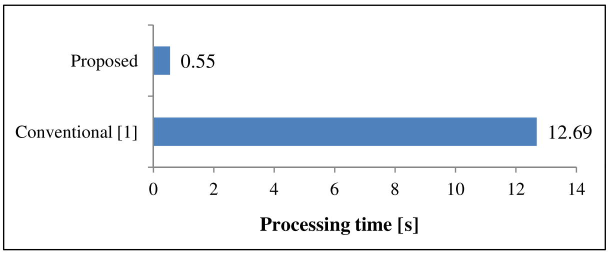

Fig. 5. The processing time of the proposed method and the conventional method [1].

REFERENCES

[1] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic Tone Reproduction for Digital Images" *ACM Tran.Graphics*, Vol.21, No.3, p.267-276, July.2002.
[2] E. Reinhard, G. Ward, S. Pattanaik, P. Debevec, W. Heidrich, and K. Myszkowski, "High Dynamic Range Imaging - Acquisition, Display and Image based Lighting" Morgan Kaufmann, 2010
[3] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes" *Computer Graphics Forum*, Vol.22, p.419-426, 2003.
[4] R. Fattal, D. Lischinski, and M. Werman, "Gradient Domain High Dynamic Range Compression" *ACM Tran.Graphics*, Vol.21, No.3, pp.249-256, July.2002.
[5] M. Iwahashi and H. Kiya, "Efficient Lossless Bit Depth Scalable Coding for HDR Images" *APSIPA*, ASC2012, no.OS.49-IVM.17-5, December 2012.
[6] M. Iwahashi and H. Kiya, "Two Layer Lossless Coding of HDR Images" *ICASSP2013*, pp.1340-1344, 2013.
[7] R. Xu, S. N. Pattanaik, and C. E. Hughes, "High-Dynamic-Range Still Image Encoding in JPEG2000" *IEEE Computer Graphics and Applications*, Vol.25, No.6, pp.57-64, Nov.2005
[8] Y. Zhang, E. Reinhard, and D. Bull, "Perception-based high dynamic range video compression with optimal bit-depth transformation" *IEEE ICIP 2011*, pp.1321-1324, 2011.
[9] Jiang Duan, Guoping Qiu, "Fast tone mapping for high dynamic range images" *Proc. International Conference, Pattern Recognition (ICPR)*, Vol.2, pp.847-850, Aug. 2004.
[10] T. Murofushi, M. Iwahashi, and H. Kiya, "An Integer Tone Mapping Operation for HDR Images Expressed in Floating Point Data," *ICASSP2013*, pp.2479-2483, 2013.