A Unified Tone Mapping Operation for HDR Images Including Both Floating-Point and Integer Data

Toshiyuki Dobashi[1(✉)], Masahiro Iwahashi[2], and Hitoshi Kiya[1]

[1] Tokyo Metropolitan University, Tokyo, Japan
dobashi-toshiyuki1@ed.tmu.ac.jp
[2] Nagaoka University of Technology, Niigata, Japan

Abstract. This paper considers a unified tone mapping operation (TMO) for HDR images. This paper includes not only floating-point data but also long-integer (i.e. longer than 8-bit) data as HDR image expression. A TMO generates a low dynamic range (LDR) image from a high dynamic range (HDR) image by compressing its dynamic range. A unified TMO can perform tone mapping for various HDR image formats with a single common TMO. The integer TMO which can perform unified tone mapping by converting an input HDR image into an intermediate format was proposed. This method can be executed efficiently with low memory and low performance processor. However, only floating-point HDR image formats have been considered in the unified TMO. In other words, a long-integer which is one of the HDR image formats has not been considered in the unified TMO. This paper extends the unified TMO to a long-integer format. Thereby, the unified TMO for all possible HDR image formats can be realized. The proposed method ventures to convert a long-integer number into a floating-point number, and treats it as two 8-bit integer numbers which correspond to its exponent part and mantissa part. These two integer numbers are applied the tone mapping separately. The experimental results shows the proposed method is effective for an integer format in terms of the resources such as the computational cost and the memory cost.

Keywords: High dynamic range · Tone mapping · Unified · Integer · Floating-point

1 Introduction

High dynamic range (HDR) images are spreading in many fields: photography, computer graphics, on-vehicle cameras, medical imaging, and more. They have wider dynamic range of pixel values than standard low dynamic range (LDR) images. In contrast, display devices which can express the pixel values of HDR images are not popular yet. Therefore, the importance of a tone mapping operation (TMO) which generates an LDR image from an HDR image by compressing its dynamic range is growing.

Y.-S. Ho et al. (Eds.): PCM 2015, Part I, LNCS 9314, pp. 321–333, 2015.
DOI: 10.1007/978-3-319-24075-6_31

Fig. 1. The bit allocation of the RGBE format.

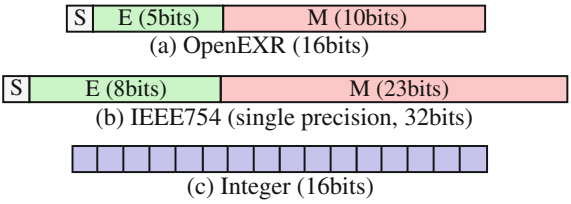


Fig. 2. The bit allocation of the OpenEXR, the IEEE754, and the integer format. Each of RGB channels has this. Let S, E, and M denote a sign, a exponent, and a mantissa, respectively.

Various research works on tone mapping have so far been done [1–11]. Reference [1–9] focus on compression techniques or quality of tone mapped images, and [10,11] focus on speeding-up of a tone mapping function. In [10,11], visibility and contrast are simply controlled with a single parameter. Nevertheless, tone mapping functions for these approach is limited to a specific one. Moreover, the tone mapping function is only one process out of many processes in a TMO.

Unlike these research works, an integer TMO approach which deals with resource reduction was proposed in [12–14]. Considering not only a function itself but also the whole process of a TMO, this method tries to resolve the essential problem on high demand of resources. In these methods, any kind of global tone mapping functions can be used. The method in [12] treats a floating-point number as two 8-bit integer numbers which correspond to a exponent part and a mantissa part, and applies tone mapping to these integer numbers separately. The method reduces the memory cost by using 8-bit integer data instead of 64-bit floating-point data. Moreover, using 8-bit integer data facilitates executing calculations with fixed-point arithmetic because it eases the limitation of the bit length. Fixed-point arithmetic is often utilized in image processing and embedded systems because of the advantages such as low-power consumption, the small circuit size and high-speed computing [15–17]. The method in [13] executes the integer TMO with fixed-point arithmetic, and therefore it reduces the computational cost as well. However, this integer TMO approach is designed for the RGBE format; its performance is not guaranteed for other formats. In [18,19], the intermediate format was introduced, and the integer TMO was extended for it. By using the intermediate format, it can be applied for other formats such as the OpenEXR and the IEEE754. This method can also be implemented with fixed-point arithmetic, and it applies tone mapping with low resources. Nevertheless, in [18,19], only floating-point HDR image formats are used as input images; a long-integer which is one of the HDR image formats is not considered.

This paper extends the method in [19] to an integer format, and confirms its efficacy. The proposed unified TMO can treat all possible HDR image formats

including both floating-point data and integer data. There are two patterns of tone mapping for an integer format. One is a method of processing an integer format directly without format conversion. Although this method is simple, it can not be used as a unified TMO because it is exclusive use of the integer format. The other is a method of processing after converting an integer format to an intermediate format. The latter, namely, the proposed method can be used as a unified TMO which can process various HDR image formats in addition to the integer format. In addition, the experimental results shows the proposed method is more effective for an integer format in terms of the resources such as the computational cost and the memory cost.

2 Preliminaries

2.1 Floating-Point HDR Image Formats

This section describes HDR image formats. There are several formats, this paper focuses on the integer format. However, the proposed method is not limited to the integer format. The method is a unified tone mapping operation, and it can be used for other formats as well. The HDR image formats supported by the proposed method are the RGBE [20], the OpenEXR [21], the IEEE754 [22], and the integer. Figure 1 shows the bit allocation of the RGBE format, and Fig. 2 shows that of the others. The RGBE, the OpenEXR, and the IEEE754 are floating-point format.

2.2 Global Tone Mapping Operation

A TMO generates an LDR image from an HDR image by compressing its dynamic range. There are two types of a TMO: global tone mapping and local tone mapping, this paper deals with global tone mapping. A procedure of "Photographic Tone Reproduction" [1], which is one of the well-known global TMOs, is described in this section.

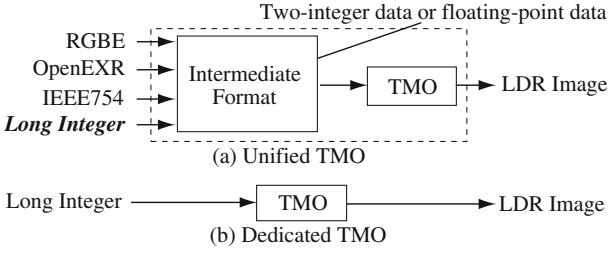


Fig. 3. The scheme of a unified tone mapping operation.

First, the world luminance $L_w(p)$ of the HDR image is calculated from RGB pixel values of the HDR image,

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

where $R(p), G(p)$, and $B(p)$ are RGB pixel values of the HDR image, respectively.

Next, the geometric mean $\bar{L}_w$ of the world luminance $L_w(p)$ is calculated as follows

$$\bar{L}_w = \exp\left(\frac{1}{N}\sum_p \log_e\left(L_w(p)\right)\right), \tag{2}$$

where N is the total number of pixels in the input HDR image. Note that Eq. (2) has the singularity due to zero value of $L_w(p)$. It is avoided by introducing a small value as shown in [1]. However, its affection is not negligible for pixel values in a resulting LDR image because a typical HDR image format such as the RGBE can express a small pixel value. Therefore, only non-zero values are used in this calculation.

Then, the scaled luminance $L(p)$ is calculated as

$$L(p) = k \cdot \frac{L_w(p)}{\bar{L}_w}, \tag{3}$$

where $k \in [0, 1]$ is the parameter called "key value".

Next, the display luminance $L_d(p)$ is calculated as follows

$$L_d(p) = \frac{L(p)}{1 + L(p)}, \tag{4}$$

Finally, the 24-bit color RGB values $C_I(p)$ of the LDR image is calculated as follows

$$C_I(p) = \text{round}\left(L_d(p) \cdot \frac{C(p)}{L_w(p)} \cdot 255\right), \tag{5}$$

where $C(p) \in \{R(p), G(p), B(p)\}$, round$(x)$ rounds x to its nearest integer value, and $C_I(p) \in \{R_I(p), G_I(p), B_I(p)\}$.

Despite the resulting LDR image is integer data, the data and arithmetic in the above procedure are both floating-point. Large computational and memory cost is required because of this.

3 Proposed Method

This section describes a scheme of a unified TMO, an intermediate format, an integer TMO for the intermediate format, and the way to execute the integer TMO with fixed-point arithmetic.

3.1 Unified TMO

Figure 3 shows the scheme of a unified tone mapping. In (a), various input HDR image formats are converted to an intermediate format, and then a TMO for the intermediate format is applied. Thus, (a) is a unified method which can process various HDR image formats using a single common TMO. On the other hand, in (b), each input HDR image format is processed by a TMO dedicated to each format. This paper focuses on the TMO (a) including the long-integer as the input HDR image format.

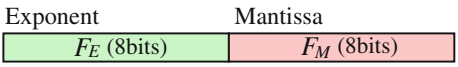


Fig. 4. The bit allocation of the proposed intermediate format.

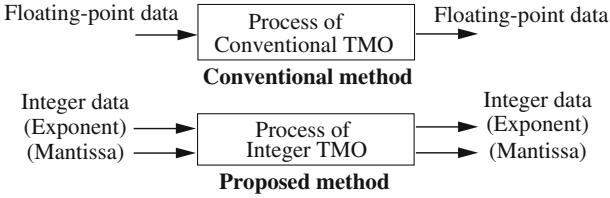


Fig. 5. The difference between the conventional method [1] and the integer TMO.

3.2 Intermediate Format

An input HDR image is converted to an intermediate format at the first step. This paper considers two types of an intermediate format; the IEEE754 format and the proposed intermediate format. The IEEE754 format is the standard floating-point format as described earlier. In this section, the proposed intermediate format is described.

Figure 4 shows the bit allocation of the proposed intermediate format. Unlike the RGBE format, the exponent part of each RGB channel in this format is independent, and it reduces the error of the format conversion. The format with 8-bit exponent part and 8-bit mantissa part is selected to reduce the memory cost. Because the memory cost is proportional to the format, it is estimated to be 1/2 and 1/4, compared to the IEEE754 single and double precision format. Details of the memory cost are described later. The encode functions which yield the exponent part F_E and the mantissa part F_M of each RGB channel F are defined as

$$F_E = \lceil \log_2 F + 128 \rceil, \quad F_M = \left\lfloor F \cdot 2^{136 - F_E} \right\rfloor, \tag{6}$$

where $\lceil x \rceil$ 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. On the other hand, the decode function which yields the original RGB value from the intermediate format is defined as

$$F = (F_M + 0.5) \cdot 2^{F_E - 136}. \tag{7}$$

3.3 Integer TMO for the Intermediate Format

The integer TMO converts input and output data of each process to two 8-bit integer data. Using 8-bit integer data facilitates executing calculations with fixed-point arithmetic because it eases the limitation of the bit length. Figure 5 shows the difference between the integer TMO and the conventional method [1]

described in Sect. 2.2. Note that this technique of the integer TMO works well by using the proposed intermediate format. The technique does not work well for the IEEE754 format because it has denormalized numbers as well as the OpenEXR [18]. The integer TMO defines new processes and replaces each tone mapping process by them. These new processes are composite functions. Each process of the proposed method is described as follows.

The integer TMO converts RGB values $C(p)$ into the intermediate format described in Sect. 3.2 at the first step. The exponent parts $C_E(p) \in \{R_E(p), G_E(p), B_E(p)\}$ and the mantissa parts $C_M(p) \in \{R_M(p), G_M(p), B_M(p)\}$ are calculated as

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

$$C_M(p) = \left\lfloor C(p) \cdot 2^{136 - C_E(p)} \right\rfloor . \tag{9}$$

Then, the exponent part $L_{w_E}(p)$ and the mantissa part $L_{w_M}(p)$ of the world luminance $L_w(p)$ of the HDR image are calculated as

$$L_{w_E}(p) = \lceil \log_2 ML(p) - 8 \rceil , \tag{10}$$

$$L_{w_M}(p) = \left\lfloor ML(p) \cdot 2^{-L_{w_E}(p)} \right\rfloor , \tag{11}$$

$$\begin{aligned} ML(p) = {} & 0.27(R_M(p) + 0.5) \cdot 2^{R_E(p)} + \\ & 0.67(G_M(p) + 0.5) \cdot 2^{G_E(p)} + \\ & 0.06(B_M(p) + 0.5) \cdot 2^{B_E(p)}, \end{aligned} \tag{12}$$

where $0 \le L_{w_E}(p) \le 255$ and $0 \le L_{w_M}(p) \le 255$. The method sets $L_{w_E}(p) = L_{w_M}(p) = 0$ if $C_E(p) = 0$, and the method sets $L_{w_M}(p) = 255$ if $L_{w_M}(p) = 256$.

Next, the exponent part $\bar{L}_{w_E}$ and the mantissa part $\bar{L}_{w_M}$ of the geometric mean $\bar{L}_w$ of the HDR image are calculated as

$$\bar{L}_{w_E} = \lceil SL_{w_M} + SL_{w_E} + 128 \rceil , \tag{13}$$

$$\bar{L}_{w_M} = \left\lfloor 2^{SL_{w_M} + SL_{w_E} - \bar{L}_{w_E} + 136} \right\rfloor , \tag{14}$$

$$SL_{w_E} = \frac{1}{N} \sum_p \left(L_{w_E}(p) - 136 \right) , \tag{15}$$

$$SL_{w_M} = \frac{1}{N} \sum_p \log_2 \left(L_{w_M}(p) + 0.5 \right) , \tag{16}$$

where $0 \le \bar{L}_{w_E} \le 255$ and $0 \le \bar{L}_{w_M} \le 255$. Here, $\bar{L}_{w_E}$ and $\bar{L}_{w_M}$ are computed using only non-zero $L_{w_E}(p)$'s.

Then, the exponent part $L_E(p)$ and the mantissa part $L_M(p)$ of the scaled luminance $L(p)$ of the HDR image are calculated as

$$L_E(p) = \left\lceil \log_2(AL_w(p)) + L_{w_E}(p) - \bar{L}_{w_E} + 128 \right\rceil , \tag{17}$$

$$L_M(p) = \left\lfloor AL_w(p) \cdot 2^{136 + L_{w_E}(p) - L_E(p) - \bar{L}_{w_E}} \right\rfloor , \tag{18}$$

$$AL_w(p) = k \cdot \frac{L_{w_M}(p) + 0.5}{\bar{L}_{w_M} + 0.5} . \tag{19}$$

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, $0 \leq L_E(p) \leq 255, 0 \leq L_M(p) \leq 255$.

Next, the method calculates the exponent part $L_{d_E}(p)$ and the mantissa part $L_{d_M}(p)$ of the display luminance $L_d(p)$. This calculation depends on tone mapping functions. Here, the tone mapping function of Eq. (4) is used as an example,

$$L_{d_E}(p) = \lceil \log_2(FL(p)) + 128 \rceil, \tag{20}$$

$$L_{d_M}(p) = \left\lfloor FL(p) \cdot 2^{136 - L_{d_E}(p)} \right\rfloor, \tag{21}$$

$$FL(p) = \frac{L_M(p) + 0.5}{L_M(p) + 0.5 + 2^{136 - L_E(p)}}. \tag{22}$$

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, $0 \leq L_{d_E}(p) \leq 255, 0 \leq L_{d_M}(p) \leq 255$.

Finally, the 24-bit RGB pixel values $C_I(p)$ of the LDR image is obtained as

$$C_I(p) = \text{round}\left(RL(p) \cdot 2^{C_E(p) + L_{d_E}(p) - L_{w_E}(p) - 136}\right), \tag{23}$$

$$RL(p) = \frac{(L_{d_M}(p) + 0.5)(C_M(p) + 0.5)}{L_{w_M}(p) + 0.5} \cdot 255. \tag{24}$$

In the above processes, the input and output data of each calculation are all 8-bit integer data. The next section describes fixed-point arithmetic in the proposed method.

3.4 Fixed-Point Arithmetic

In the integer TMO, only the data are converted to integer, and the memory cost is reduced. However, the internal arithmetic of the integer TMO is still with floating-point. The proposed method introduces fixed-point arithmetic to reduce the computational cost as well. This section describes the way to execute the internal arithmetic with fixed-point arithmetic. Most of equations can be calculated with fixed-point arithmetic because each variable is expressed in 8-bit integer [13]. Nevertheless, Eq. (22) 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. (22) as follows

$$FL(p) = \frac{1}{1 + \frac{2^{136 - L_E(p)}}{L_M(p) + 0.5}}. \tag{25}$$

Furthermore, the method branches Eq. (25) into three cases and approximates it based on the power of two in the denominator as follows.

Case 1: If $136 - L_E(p) > 15$ in Eq. (25), '1' in the denominator can be ignored because the right part of the denominator is very large, and so it is approximated as

$$FL(p) = \frac{L_M(p) + 0.5}{2^{136 - L_E(p)}}, \tag{26}$$

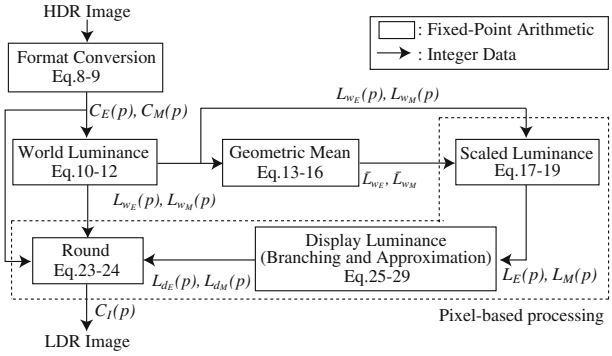


Fig. 6. The outline of the proposed method.

$$L_{d_E}(p) = \lceil \log_2(L_M(p)+0.5) - (136 - L_E(p)) + 128 \rceil, \tag{27}$$

$$L_{d_M}(p) = \left\lfloor (L_M(p) + 0.5) \cdot 2^{L_E(p) - L_{d_E}(p)} \right\rfloor. \tag{28}$$

Case 2: If $136 - L_E(p) < -8$ in Eq. (25), the right part of the denominator can be ignored because it is very small, and so it is approximated as

$$FL(p) = 1, \;\; L_{d_E}(p) = 128, \;\; L_{d_M}(p) = 255. \tag{29}$$

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

In addition, the method uses pre-calculated tables for calculations of 2^x (in Eq. (14)) and $\log_2$ (in Eq. (16)). Each table consists of 16×256 bits. In Eqs. (8)–(29), division operations are simply done by division, not right shift. Moreover, 2^x and $\log_2$ are conducted by using simple bit shift operation except Eqs. (14) and (16). The method can calculate all equations of the TMO with only fixed-point arithmetic by these branching, approximation, and tables.

4 Experimental and Evaluation Results

This paper proposed the unified TMO with the intermediate format. The proposed method can be executed with fixed-point arithmetic to reduce the computational cost. However, errors can occur by these format conversion and fixed-point arithmetic. To confirm the efficacy of the proposed method and the errors involved with it, the experiments and evaluation were carried out. These experiments and evaluation consist of measurements of peak signal-to-noise ratio (PSNR), the structural similarity index (SSIM) [23] of the resulting LDR images and processing time of the TMO, and evaluation of memory usage. Figure 7 shows the block diagram of these. These experiments and evaluation compared the proposed method (with floating-point and fixed-point arithmetic), the IEEE754 methods (double and single precision), and the integer dedicated method. The common conditions in the experiments and the evaluation are listed in Table 1. All of the methods were implemented in C-language.

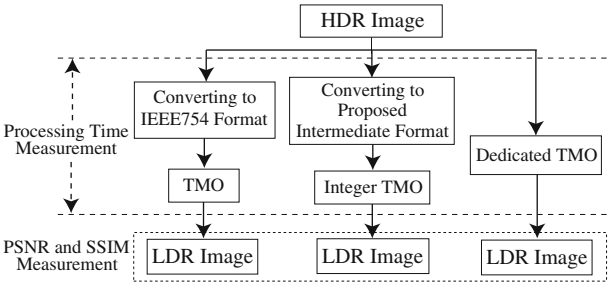


Fig. 7. The block diagram of the experiments.

Table 1. The common conditions in the experiments and the evaluation.

	Arithmetic	Data
IEEE754 (double precision)	64-bit Floating-point	64-bit Floating-point
IEEE754 (single precision)	64-bit Floating-point	32-bit Floating-point
Integer dedicated	64-bit Fixed-point	16-bit Integer
Proposed (floating-point)	64-bit Floating-point	8-bit Integer
Proposed (fixed-point)	32-bit Fixed-point	8-bit Integer

4.1 Comparison of Tone-Mapped LDR Images

This experiment applied tone mapping for 73 HDR images in 16-bit integer format using each method, and measured the PSNR and the SSIM. The tone mapped images $I_{\mathrm{LDR}}(p)$ and $I'_{\mathrm{LDR}}(p)$ are given by

$$I_{\mathrm{LDR}}(p) = \mathrm{T_c}[I_{\mathrm{HDR}}(p)], \;\; I'_{\mathrm{LDR}}(p) = \mathrm{T_p}[I_{\mathrm{HDR}}(p)], \tag{30}$$

where $I_{\mathrm{HDR}}(p)$ is an input HDR image, and $\mathrm{T_c}[\cdot]$ and $\mathrm{T_p}[\cdot]$ are TMO of the conventional method and the proposed method, respectively. The PSNR between $A \times B$ sized LDR images is given by

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, \tag{31}$$

$$MSE = \frac{1}{AB} \sum_{p=1}^{AB} [I_{\mathrm{LDR}}(p) - I'_{\mathrm{LDR}}(p)]^2. \tag{32}$$

The SSIM offers more subjective evaluation than the PSNR. If $I_{\mathrm{LDR}}(p) = I'_{\mathrm{LDR}}(p)$, the PSNR will be ∞ and the SSIM will be 1.0. This experiment used the resulting LDR image of the IEEE754 double precision method as a true value because it was executed with most plenty resources. That is, $\mathrm{T_a}[\cdot]$ was the IEEE754 double precision method and $\mathrm{T_b}[\cdot]$ was the proposed method or the integer dedicated method.

Table 2 shows the maximum, minimum, and average PSNR and the average SSIM. The integer dedicated method gave high PSNR and SSIM values because

(a) The IEEE754 double precision method.

(b) The proposed method with fixed-point arithmetic. (SSIM = 0.999)

Fig. 8. LDR images comparison. It is impossible for human eyes to distinguish these images

Table 2. The maximum, minimum, and average PSNR and the average SSIM of the methods.

	PSNR [dB]			SSIM
	Maximum	Minimum	Average	
Integer dedicated	91.76	49.91	60.10	0.9993
Proposed (floating-point)	77.28	50.39	56.24	0.9985
Proposed (fixed-point)	66.56	50.39	55.84	0.9985

it is designed for integer format. On the other hand, the proposed method also gave a high SSIM value. Although the PSNR of the proposed method were slightly lowered, they were still sufficiently high values. Moreover, the proposed method with fixed-point arithmetic also maintains the sufficiently high values. Figure 8 shows example LDR images obtained by this experiment. It indicates that it is impossible for human eyes to distinguish these images. From the above results, this experiment confirmed that the proposed method can execute the TMO with high accuracy.

4.2 Comparison of the Memory Usage

Table 3 shows the memory usage of each calculation when the size of the input HDR image is $A \times B$ pixels. The rest of calculations which is not included

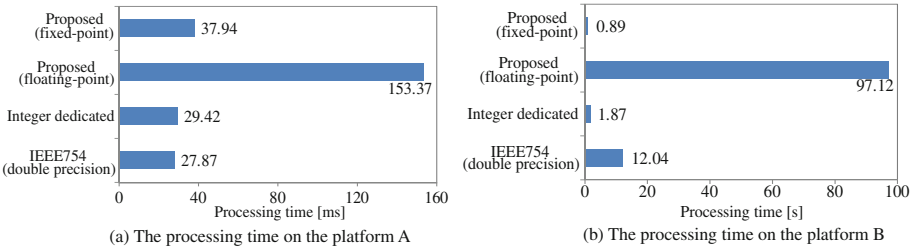
Table 3. The memory usage of the methods.

	Memory usage [bytes]				
	IEEE754 (double precision)	IEEE754 (single precision)	Integer dedicated	Proposed (floating-point)	Proposed (fixed-point)
An HDR image	$A \times B \times 24$	$A \times B \times 12$	$A \times B \times 6$	$A \times B \times 6$	$A \times B \times 6$
World luminance	$A \times B \times 8$	$A \times B \times 4$	$A \times B \times 4$	$A \times B \times 2$	$A \times B \times 2$
Geometric mean	8	4	4	2	2
Table	–	–	393216	-	1024

this table can be conducted per pixel, and it is indicated in Fig. 6. The proposed method and the integer dedicated method used the pre-calculated tables in order to calculate with fixed-point arithmetic. To reduce the table of data, sometimes interpolation is used. However, these implementation did not use interpolation in order to avoid its influence such as errors and load. Therefore, the table corresponding to bit-length of the data used in the method is required. Because the data is 8-bit each in the exponent and the mantissa, the table of the proposed method is smaller than the integer dedicated method. The memory usage of the proposed method which depends on the image size is 75 %, 50 %, and 20 % less than the IEEE754 double precision method, the IEEE754 single precision method, and the integer dedicated method, respectively.

Table 4. The platforms used in the experiment.

	Processor	FPU	RAM
Platform A	Intel Core i7 3930 K 3.2 GHz	Yes	16 GB
Platform B	Marvell PXA270 624 MHz	No	128 MB



(a) The processing time on the platform A

(b) The processing time on the platform B

Fig. 9. The processing time of the methods.

4.3 Comparison of the Processing Time

This experiment applied tone mapping for HDR images with 512 × 768 pixels in the integer format using each method, and measured the processing time of the methods. This experiment was carried out on two platforms listed in Table 4.

Figure 9 (a) compares the processing time of the methods on the platform A. This platform has an FPU which can process the IEEE754 floating-point data. Thus, IEEE754 method was processed at the highest speed. The processing time of single precision and double precision were almost same because single precision is treated as double precision in the processor.

On the other hand, Fig. 9 (b) compares the processing time of the methods on the platform B. The methods with fixed-point arithmetic were processed fast

on this platform because it does not have an FPU. The proposed method with fixed-point arithmetic was 13.53 and 2.10 times faster than the IEEE754 double precision method and the integer dedicated method, respectively. Therefore, this experiment confirmed that the proposed method with fixed-point arithmetic can perform high speed processing on the processor without an FPU.

5 Conclusion

This paper proposed the unified TMO including both floating-point data and long-integer data. The proposed method ventures to convert a long-integer number into the floating-point number, and treats it as two 8-bit integer numbers which correspond to its exponent part and mantissa part. The memory usage of the method is reduced by using these 8-bit integer numbers. Moreover, the method with fixed-point arithmetic can be executed fast on a processor without an FPU. The method is the unified TMO which can process various HDR image formats including a long-integer, namely a dedicated TMO for each format is not required. The proposed unified TMO can treat all possible HDR image formats including both floating-point data and integer data. The experimental and evaluation results confirmed that the method can be executed with fewer resources than the other methods, while it offers high accuracy of tone mapping.

References

1. Reinhard, E., Stark, M., Shirley, P., Ferwerda, J.: Photographic tone reproduction for digital images. ACM Trans. Graph. **21**(3), 267–276 (2002)
2. Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., Myszkowski, K.: High Dynamic Range Imaging - Acquisition, Display and Image based Lighting. Morgan Kaufmann, Burlington (2010)
3. Drago, F., Myszkowski, K., Annen, T., Chiba, N.: Adaptive logarithmic mapping for displaying high contrast scenes. Comput. Graph. Forum **22**(3), 419–426 (2003)
4. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. ACM Trans. Graph. **21**(3), 249–256 (2002)
5. Iwahashi, M., Kiya, H.: Efficient lossless bit depth scalable coding for HDR images. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), no.OS.37-IVM.16-4 (2013)
6. Iwahashi, M., Kiya, H.: Two layer lossless coding of HDR images. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1340–1344 (2013)
7. Xu, R., Pattanaik, S.N., Hughes, C.E.: High-dynamic-range still image encoding in JPEG2000. IEEE Trans. Comput. Graph. Appl. **25**(6), 57–64 (2005)
8. Zhang, Y., Reinhard, E., Bull, D.: Perception-based high dynamic range video compression with optimal bit-depth transformation. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 1321–1324 (2011)
9. Iwahashi, M., Yoshida, T., Mokhtar, N.B., Kiya, H.: Bit-depth scalable lossless coding for high dynamic range images. EURASIP J. Adv. Sig. Process. **2015**, 22 (2015)

10. Thakur, S.K., Sivasubramanian, M., Nallaperumal, K., Marappan, K., Vishwanath, N.: Fast tone mapping for high dynamic range images. In: Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–4 (2013)
11. Duan, J., Qiu, G: Fast tone mapping for high dynamic range images. In: Proceedings of the International Conference on Pattern Recognition (ICPR), pp. 847–850 (2004)
12. Murofushi, T., Iwahashi, M., Kiya, H.: An integer tone mapping operation for HDR images expressed in floating point data. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2479–2483 (2013)
13. Dobashi, T., Murofushi, T., Iwahashi, M., Kiya, H.: A fixed-point tone mapping operation for HDR images in the RGBE format. In: Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), no.OS.37-IVM.16-4 (2013)
14. Dobashi, T., Murofushi, T., Iwahashi, M., Kiya, H.: A fixed-point global tone mapping operation for HDR images in the RGBE format. IEICE Trans. Fundam. **E97–A**(11), 2147–2153 (2014)
15. Lampert, C.H., Wirjadi, O.: Anisotropic gaussian filtering using fixed point arithmetic. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 1565–1568 (2006)
16. Chang, W.-H., Nguyen, T.Q.: On the fixed-point accuracy analysis of FFT algorithm. IEEE Trans. Sig. Process. **56**(10), 4673–4682 (2008)
17. Rocher, R., Menard, D., Scalart, P., Sentieys, O.: Analytical approach for numerical accuracy estimation of fixed-point systems based on smooth operations. IEEE Trans. Circ. Syst. Part-I **59**(10), 2326–2339 (2012)
18. Murofushi, T., Dobashi, T., Iwahashi, M., Kiya, H.: An integer tone mapping operation for HDR images in OpenEXR with denormalized numbers. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), no.TEC-P10.6 (2014)
19. Dobashi, T., Tashiro, A., Iwahashi, M., Kiya, H.: A fixed-point implementation of tone mapping operation for HDR images expressed in floating-point format. APSIPA Trans. Sig. Inf. Process. **3**(11), 1–11 (2004)
20. Ward, G.: Real pixels. In: Arvo, J. (ed.) Graphic Gems 2, pp. 80–83. Academic Press, San Diego (1992)
21. Kainz, F., Bogart, R., Hess, D.: The OpenEXR image file format. In: ACM SIGGRAPH Technical Sketches & Applications (2003)
22. Information technology - Microprocessor Systems - Floating-Point arithmetic. ISO/IEC/IEEE 60559 (2011)
23. Wang, Z., Bovik, A.C., Seikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)