ORIGINAL ARTICLE

A fixed-point implementation of tone mapping operation for HDR images expressed in floating-point format

TOSHIYUKI DOBASHI[1], ATSUSHI TASHIRO[1], MASAHIRO IWAHASHI[2] AND HITOSHI KIYA[1]

A tone mapping operation (TMO) for HDR images with fixed-point arithmetic is proposed. A TMO generates a low dynamic range (LDR) image from a high dynamic range (HDR) image by compressing its dynamic range. Since HDR images are generally expressed in a floating-point data format, a TMO also deals with floating-point data even though resulting LDR images have integer data. As a result, conventional TMOs require many resources such as computational and memory cost. To reduce the resources, an integer TMO which treats a floating-point number as two 8-bit integer numbers was proposed. However, this method has the limitation of available input HDR image formats. The proposed method introduces an intermediate format to relieve the limitation of input formats, and expands the integer TMO for the intermediate format. The proposed integer TMO can be applied for multiple formats such as the RGBE and the OpenEXR. Moreover, the method can conduct all calculations in the TMO with fixed-point arithmetic. Using both integer data and fixed-point arithmetic, the method reduces not only the memory cost, but also the computational cost. The experimental and evaluation results show that the proposed method reduces the computational and memory cost, and gives almost same quality of LDR images, compared with the conventional method with floating-point arithmetic.

I. INTRODUCTION

High dynamic range (HDR) images are diffusing 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.

Various research works on tone mapping have so far been done [1–9]. Many of these are focused on compression techniques or quality of tone mapped images. Unlike these research works, our method proposes to reduce resources such as computational and memory cost during a TMO.

[1]Department of Information and Communication Systems, Faculty of System Design, Tokyo Metropolitan University, Hino-shi 191-0065, Japan
[2]Department of Electrical, Electronics and Information Engineering, Nagaoka University of Technology, Nagaoka-shi 940-2188, Japan

Corresponding author:
Hitoshi KIYA
Email: kiya@tmu.ac.jp

In general, reducing computational and memory cost is an important issue in image processing. HDR images are generally expressed in floating-point data formats such as the OpenEXR [10] and the RGBE [11]. Because of this, a TMO is executed with floating-point arithmetic, and it requires large computational and memory cost. Specifically, embedded systems often have only limited resources: low-memory or low-performance processor without a floating-point unit (FPU). Furthermore, real-time processing, such as an HDR video, requires speeding-up or parallelization of computing. For these reasons, reducing computational and memory cost regarding a TMO is an important issue.

To reduce the computational cost, fixed-point arithmetic is effective. 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 [12–16]. However, executing a TMO with fixed-point arithmetic is difficult due to the wide range value of HDR images. On the other hand, fast tone mapping functions were proposed in [17–20]. The authors in [17, 18] focus on speeding up of the trilateral filter-based HDR tone mapping technique. However, it still takes a long time for processing because the trilateral filter-based technique itself is heavy. In [19, 20], visibility and contrast are

simply controlled with a single parameter. Nevertheless, it does not directly contribute to reducing resources, and tone mapping functions for this approach is limited to a specific one. Moreover, the tone mapping function is only one process out of many processes in a TMO. The proposed method considers the whole process of a TMO, and focuses on global tone mapping [1, 2].

Unlike the lightweight tone mapping approach, an integer TMO, which deals with resource reduction was proposed in [21, 22]. 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 [21] 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 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. The method in [22] executes the integer TMO with fixed-point arithmetic, and therefore it reduces the computational cost as well.

The proposed method is based on [21, 22]. The integer TMO [21, 22] is designed for the RGBE format; if using for other formats, it is necessary to design a new algorithm dedicated to each format. The method introduces an intermediate format to relieve the limitation of the input HDR image format, and expands the integer TMO for the intermediate format. The proposed integer TMO can be applied for multiple HDR image formats by converting the input image to the intermediate format. The method can conduct all the calculations of the TMO with only fixed-point arithmetic. By these features, the method can be executed under limited resources, such as processors without a FPU or low-memory. The experiments and evaluation confirmed that the proposed method reduces the computational cost and the memory cost, and keeps the quality of tone mapped images, compared to the conventional method with floating-point arithmetic [1].

II. PRELIMINARIES

This section describes widely used HDR image formats and the conventional TMO.

A) HDR image formats

1) THE RGBE FORMAT

Figure 1 shows the bit allocation of the RGBE format [11]. Each pixel is 32 bits long in this format. It consists of 8-bit common exponent and 8-bit mantissa for each RGB channel. The relation among a real number F_1, the exponent part F_{1E} and the mantissa part F_{1M} is given as

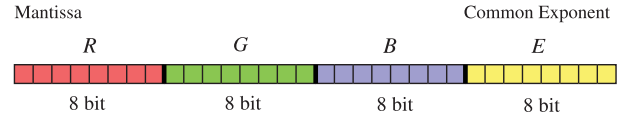$$F_1 = (F_{1M} + 0.5) \cdot 2^{F_{1E} - 136}. \tag{1}$$
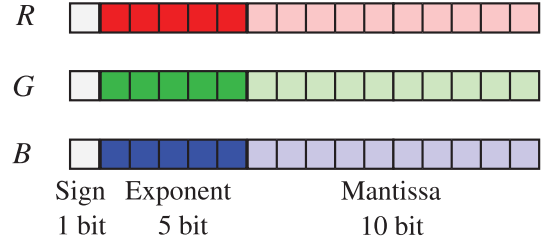

Fig. 1. Bit allocation of the RGBE format.

Fig. 2. Bit allocation of the OpenEXR format.

Table 1. The range of the normalized number and the denormalized number in the OpenEXR format.

	Range
Denormalized number	$0, 2^{-24} - 2^{-14}$
Normalized number	$2^{-14} - 65504$

2) THE OPENEXR FORMAT

Figure 2 shows the bit allocation of the OpenEXR format [10]. Each pixel is 48 bits long in this format. This format has two expressions: the normalized numbers and the denormalized numbers. The range of the normalized numbers is restricted as shown in Table 1. The denormalized numbers are used to express the small absolute values which cannot be expressed in the normalized numbers. Therefore, the OpenEXR format has two different encoding processes and decoding processes, respectively. The relation among a real number F_2, the sign $s_2 \in \{0, 1\}$, the exponent part F_{2E} and the mantissa part F_{2M} in the decoding process for a normalized number is given as

$$F_2 = (-1)^{s_2} \cdot (1 + F_{2M} \cdot 2^{-10}) \cdot 2^{F_{2E} - 15}. \tag{2}$$

On the other hand, the relation for a denormalized number is described as

$$F_2 = (-1)^{s_2} \cdot (F_{2M} \cdot 2^{-10}) \cdot 2^{-14}. \tag{3}$$

3) THE IEEE754 FORMAT

The IEEE754 format is a standard specification which is used for calculation of floating-point number [23]. The processor with a FPU can perform high-speed processing for this format. Figure 3 shows the bit allocation of the IEEE754 single precision format. The relation among a real number F_3, the sign $s_3 \in \{0, 1\}$, the exponent part F_{3E} and the mantissa part F_{3M} is given as

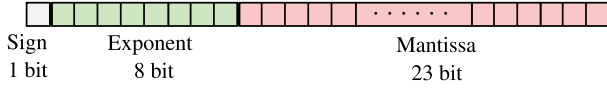$$F_3 = (-1)^{s_3} \cdot (1 + F_{3M} \cdot 2^{-23}) \cdot 2^{F_{3E} - 127}. \tag{4}$$

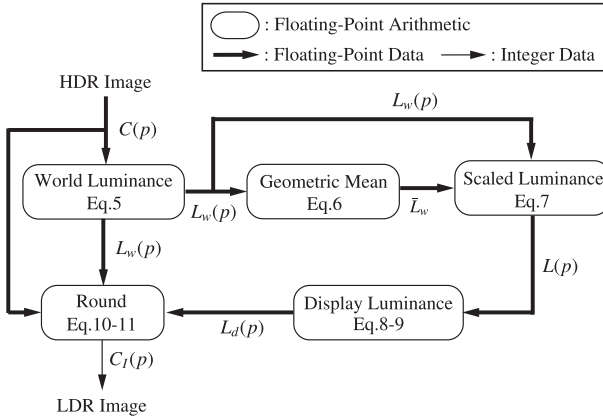**Fig. 3.** Bit allocation of the IEEE754 single precision format.

Fig. 4. Outline of the photographic tone reproduction [1].

B) Global TMO

A procedure of the conventional TMO [1] is described in this section. A TMO generates an LDR image expressed in integer from an HDR image expressed in floating-point. There are two types of a TMO: global tone mapping and local tone mapping; this paper deals with global tone mapping. Figure 4 shows "Photographic Tone Reproduction" which is one of the well-known global TMOs [1]. Each step in this figure is described as follows.

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

$$L_w(p) = 0.27R(p) + 0.67G(p) + 0.06B(p), \quad (5)$$

where $R(p)$, $G(p)$, and $B(p)$ are floating-point 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:

$$\bar{L}_w = \exp\left(\frac{1}{N}\sum_p \log_e\left(L_w(p)\right)\right), \quad (6)$$

where N is the total number of pixels in the input HDR image. Note that equation (6) 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

$$L(p) = k \cdot \frac{L_w(p)}{\bar{L}_w}, \quad (7)$$

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

Next, the display luminance $L_d(p)$ is calculated using a tone mapping function $y()$ as follows:

$$L_d(p) = y(L(p)). \quad (8)$$

The Reinhard's global operator [1] which is one of the well-known tone mapping functions is defined as

$$y_{\text{Reinhard}}(L(p)) = \frac{L(p)}{1 + L(p)}. \quad (9)$$

Finally, the floating-point pixel values $C_F(p)$ of the LDR image is calculated as follows:

$$C_F(p) = L_d(p) \cdot \frac{C(p)}{L_w(p)}, \quad (10)$$

where $C(p) \in \{R(p), G(p), B(p)\}$ is the floating-point RGB value of the input HDR image, and $C_F(p) \in \{R_F(p), G_F(p), B_F(p)\}$.

The 24-bit color RGB values $C_I(p)$ of the LDR image is derived from

$$C_I(p) = \text{round}\left(C_F(p) \cdot 255\right), \quad (11)$$

where 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 from this.

The integer TMO [21, 22] implements a global TMO with integer data and fixed-point arithmetic. Using integer data and fixed-point arithmetic reduces the memory cost and the computational cost. However, this method works on only the RGBE format; its performance is not guaranteed for other formats such as the OpenEXR.

The proposed method is an extended version of [21, 22]. The method introduces the intermediate format to relieve the limitation of the input HDR image format, and expands the integer TMO for the intermediate format.

III. PROPOSED METHOD

This section describes an intermediate format, an integer TMO for the intermediate format, and the way to execute the integer TMO with fixed-point arithmetic. Sections IIIA and IIIB contain extensions of [21, 22], and Section IIIC belongs to [22].

A) Intermediate format

An input HDR image is converted to the intermediate format (Fig. 5) at the first step of the proposed method. The proposed integer TMO can be applied for multiple HDR image formats by converting the input image to the intermediate format. This format can be applied for the RGBE, the OpenEXR, and the IEEE754. Figure 5 shows the bit allocation of the intermediate format. Unlike the RGBE format,
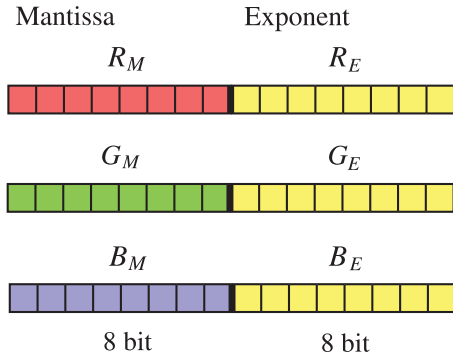
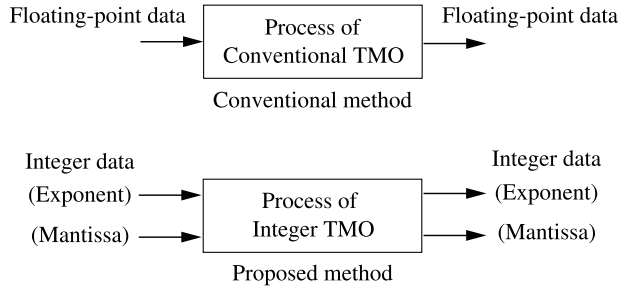**Fig. 5.** The bit allocation of the proposed intermediate format.

Fig. 6. The difference between the conventional method [1] and the proposed integer TMO.

the exponent part of each RGB channel in this format is independent, and it reduces the error of the format conversion. As an example, this section describes the case of 8-bit mantissa and 8-bit exponent; the bit length of the intermediate format will be discussed in the later section. The encode functions that yield the exponent part F_E and the mantissa part F_M of each RGB channel F are defined as

$$F_E = \lceil \log_2 F + 128 \rceil, \quad (12)$$

$$F_M = \lfloor F \cdot 2^{136 - F_E} \rfloor, \quad (13)$$

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

$$F = (F_M + 0.5) \cdot 2^{F_E - 136}. \quad (14)$$

B) 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 6 shows the difference between the conventional method [1] and the proposed integer TMO. The integer TMO defines new processes and replaces each tone mapping process by them. These new processes are composite functions shown in Fig. 7. Each process of the proposed method is described as follows.
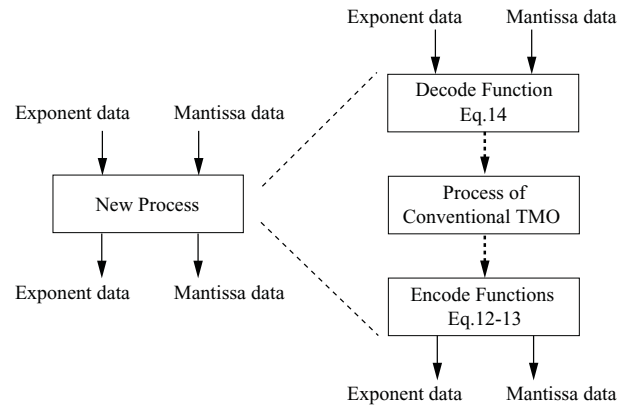

Fig. 7. A new process defined in the proposed integer TMO.

The proposed integer TMO converts RGB values $C(p)$ into the intermediate format described in Section IIIA 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

$$C_E(p) = \lceil \log_2 C(p) + 128 \rceil, \quad (15)$$

$$C_M(p) = \lfloor C(p) \cdot 2^{136 - C_E(p)} \rfloor. \quad (16)$$

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

$$L_{w_E}(p) = \lceil \log_2 ML(p) - 8 \rceil, \quad (17)$$

$$L_{w_M}(p) = \lfloor ML(p) \cdot 2^{-L_{w_E}(p)} \rfloor, \quad (18)$$

$$\begin{aligned} ML(p) = {}& 0.27(R_M(p) + 0.5) \cdot 2^{R_E(p)} \\ & + 0.67(G_M(p) + 0.5) \cdot 2^{G_E(p)} \\ & + 0.06(B_M(p) + 0.5) \cdot 2^{B_E(p)}, \end{aligned} \quad (19)$$

where $0 \leq L_{w_E}(p) \leq 255$ and $0 \leq L_{w_M}(p) \leq 255$. 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

$$\bar{L}_{w_E} = \lceil SL_{w_M} + SL_{w_E} + 128 \rceil, \quad (20)$$

$$\bar{L}_{w_M} = \lfloor 2^{SL_{w_M} + SL_{w_E} - \bar{L}_{w_E} + 136} \rfloor, \quad (21)$$

$$SL_{w_E} = \frac{1}{N} \sum_p (L_{w_E}(p) - 136), \quad (22)$$

$$SL_{w_M} = \frac{1}{N} \sum_p \log_2 (L_{w_M}(p) + 0.5), \quad (23)$$

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 using only non-zero $L_{w_E}(p)$'s.

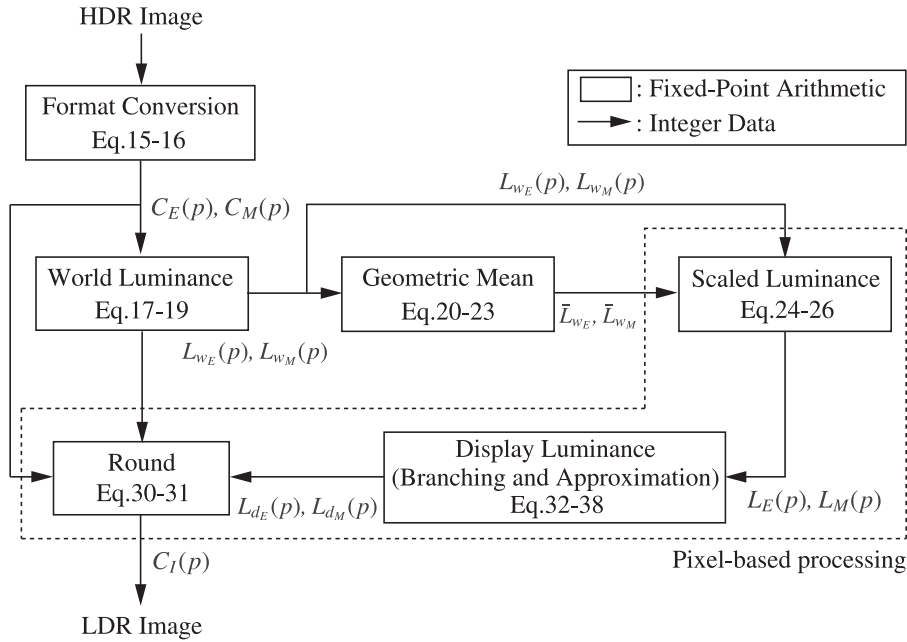Then, the exponent part $L_E(p)$ and the mantissa part $L_M(p)$ of the scaled luminance $L(p)$ of the HDR image are
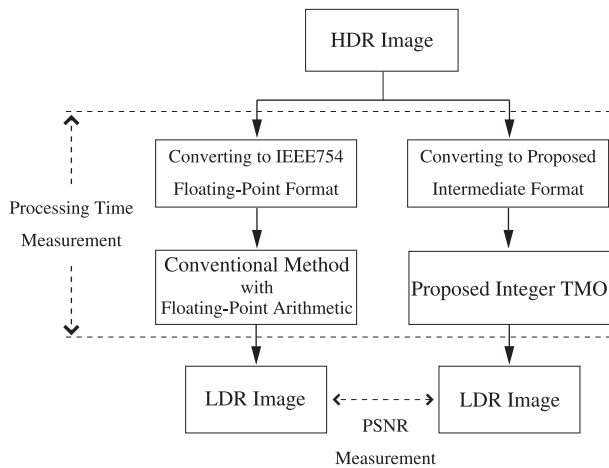
Fig. 8. The outline of the proposed method.

Fig. 9. The block diagram of the experiments.

calculated as

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

$$(24)$$

$$L_M(p) = \lfloor AL_w(p) \cdot 2^{136 + L_{w_E}(p) - L_E(p) - \bar{L}_{w_E}} \rfloor, \quad (25)$$

$$AL_w(p) = k \cdot \frac{L_{w_M}(p) + 0.5}{\bar{L}_{w_M} + 0.5}. \quad (26)$$

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

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 equation (9) is used as an example,

$$L_{d_E}(p) = \lceil \log_2(FL(p)) + 128 \rceil, \quad (27)$$

$$L_{d_M}(p) = \lfloor FL(p) \cdot 2^{136 - L_{d_E}(p)} \rfloor, \quad (28)$$

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

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

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

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

$$(30)$$

$$RL(p) = \frac{(L_{d_M}(p) + 0.5)(C_M(p) + 0.5)}{L_{w_M}(p) + 0.5}. \quad (31)$$

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.

C) 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 [22]. Nevertheless, equation (29) is difficult to be calculated without floating-point arithmetic because the range of value of the denominator is very

(a)
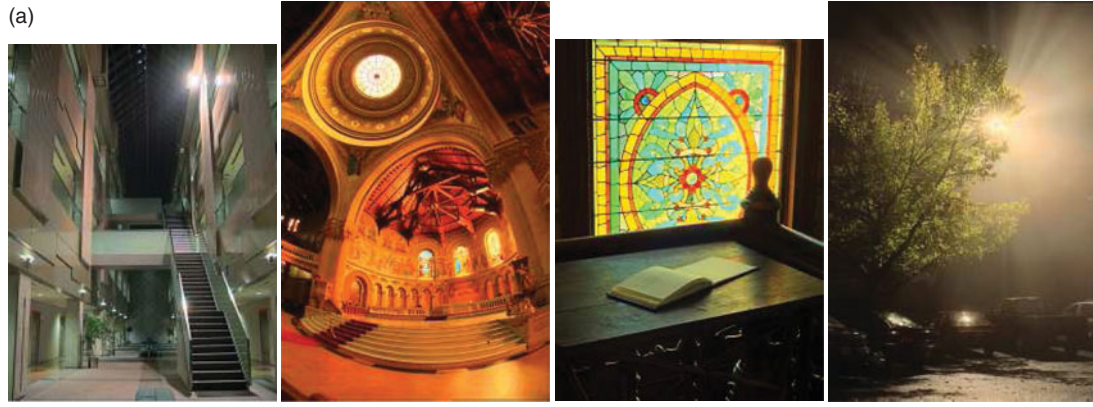
The examples of HDR images in the RGBE format.

(b)
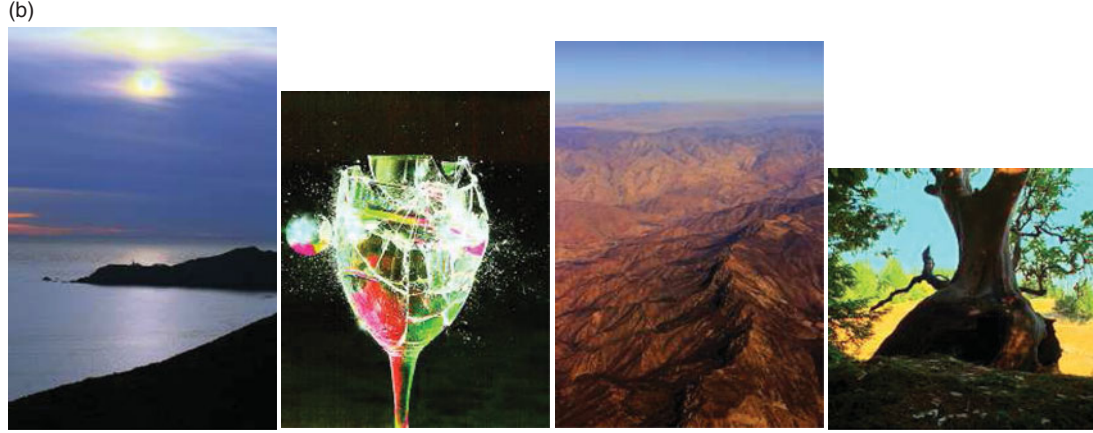
The examples of HDR images in the OpenEXR format.

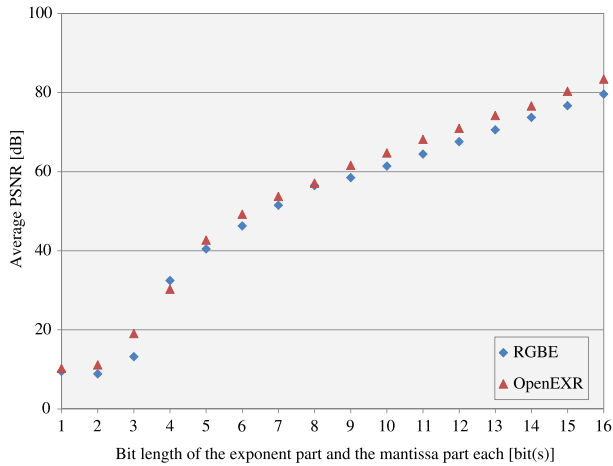**Fig. 10.** The examples of HDR images used in the experiments.

Fig. 11. The relation between the bit length of the intermediate format and the average PSNR. The exponent part and the mantissa part have same bit length.

wide. Because of this, the method deforms equation (29) as follows:

$$FL(p) = \frac{1}{1 + \frac{2^{136 - L_E(p)}}{L_M(p) + 0.5}}. \tag{32}$$

Furthermore, the method branches equation (32) into three cases and approximates it based on the power of two in the denominator as follows.

Table 2. The conditions in the experiments and the evaluation.

	Arithmetic	Data
Proposed	32-bit Fixed-point	8-bit Integer
Conventional	64-bit Floating-point	64-bit Floating-point

Case 1: If $136 - L_E(p) > 15$ in equation (32), "1" in the denominator can be ignored because the right part of the denominator is very large, and so it is approximated as

$$FL(p) = \frac{L_M(p) + 0.5}{2^{136 - L_E(p)}}, \tag{33}$$

$$L_{d_E}(p) = \lceil \log_2(L_M(p) + 0.5) - (136 - L_E(p)) + 128 \rceil, \tag{34}$$

$$L_{d_M}(p) = \lfloor (L_M(p) + 0.5) \cdot 2^{L_E(p) - L_{d_E}(p)} \rfloor. \tag{35}$$

Case 2: If $136 - L_E(p) < -8$ in equation (32), the right part of the denominator can be ignored because it is very small, and so it is approximated as

$$FL(p) = 1, \tag{36}$$

Table 3. The PSNR between the proposed and the conventional methods [1] (RGBE).

Image no.	PSNR (dB)	Image no.	PSNR (dB)
1	56.74	17	56.05
2	56.49	18	58.06
3	56.31	19	56.69
4	56.59	20	54.99
5	57.48	21	55.02
6	55.23	22	54.16
7	58.23	23	55.38
8	54.94	24	55.54
9	55.92	25	56.16
10	56.34	26	54.71
11	57.26	27	54.66
12	52.56	28	55.74
13	56.70	29	56.07
14	55.09	30	55.16
15	56.32	31	57.61
16	54.52	32	57.32

Table 4. The PSNR between the proposed and the conventional methods [1] (OpenEXR).

Image no.	PSNR (dB)	Image no.	PSNR (dB)
1	57.13	22	56.62
2	55.15	23	48.89
3	60.05	24	62.27
4	53.23	25	59.35
5	69.26	26	59.78
6	56.72	27	56.87
7	55.53	28	56.71
8	68.23	29	60.52
9	55.04	30	56.41
10	57.14	31	60.52
11	54.48	32	55.31
12	61.43	33	54.29
13	56.34	34	55.68
14	57.48	35	59.26
15	55.03	36	57.94
16	55.53	37	60.83
17	53.86	38	49.01
18	54.68	39	50.66
19	61.70	40	61.73
20	57.64	41	55.05
21	56.62	42	55.88

Table 5. The maximum, minimum, and average PSNR between the proposed and the conventional methods [1].

Input formats	PSNR (dB)		
	Maximum	Minimum	Average
RGBE	57.43	52.28	55.67
OpenEXR	69.26	48.89	57.27

$$L_{d_E}(p) = 128, \qquad (37)$$

$$L_{d_M}(p) = 255. \qquad (38)$$

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 equation (21)) and $\log_2$ (in equation (23)). Each table consists of 16×256 bits. In equation (15)–(38), division operations are simply done by division, not right shift. Moreover, 2^x and $\log_2$ are conducted using simple bit shift operation except equations (21) and (23). The method can calculate all equations of the TMO with only fixed-point arithmetic by these branching, approximation, and tables. Note that the conventional method [1] consists of floating-point data and floating-point arithmetic. In contrast, the proposed method consists of integer data and fixed-point arithmetic.

IV. EXPERIMENTAL AND EVALUATION RESULTS

The proposed method introduces the intermediate format to relieve the limitation of formats. Moreover, the computational cost of the TMO is reduced using fixed-point arithmetic instead of floating-point arithmetic in the proposed method. 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) of the resulting LDR images and processing time of the TMO, and evaluation of memory usage. Figure 9 shows the block diagram of these experiments. The HDR images in the RGBE format and the OpenEXR format were used as input images. Figure 10 shows examples of these images. The input HDR image is converted to IEEE754 floating-point format in the conventional method [1] at the first step. On the other hand, it is converted to the proposed intermediate format in the proposed method. Negative values in the OpenEXR format were set to zero. Both the proposed method and the conventional method [1] were implemented in C-language. The proposed method was compared with [1] in Sections IVB–IVD, and with the other conventional methods [20–22] in Section IVE.

A) The relation between the bit length and the tone mapped LDR image quality

This experiment was carried out to examine the relation between the bit length of the intermediate format and the tone mapped LDR image quality. In this experiment, 32 images in the RGBE format and 42 images in the OpenEXR format were used. Figure 11 shows the relation between the bit length of the intermediate format and the average PSNR, where the exponent part and the mantissa part have same bit length. It indicates that the PSNR values get better with increase in the bit length, and they can be a little <60 dB at 8-bit. Therefore, the other experiments and evaluation used the intermediate format with 8-bit exponent part and 8-bit mantissa part.
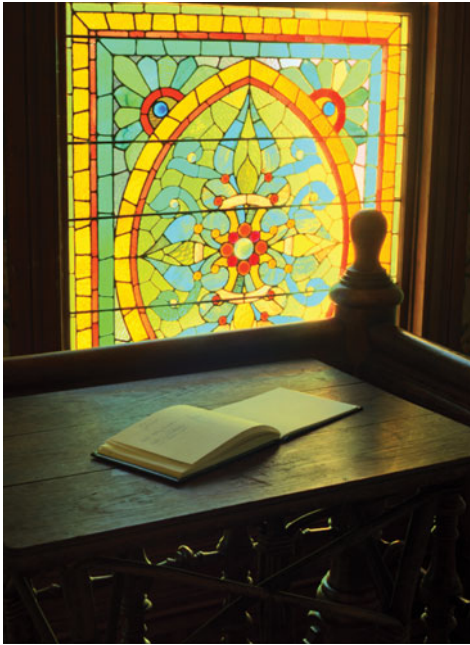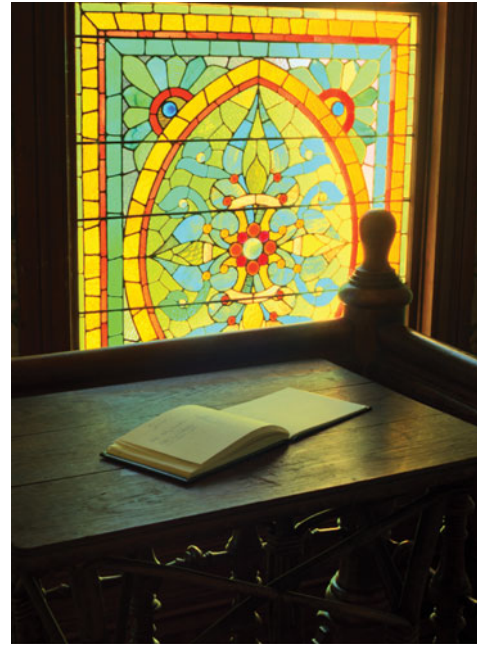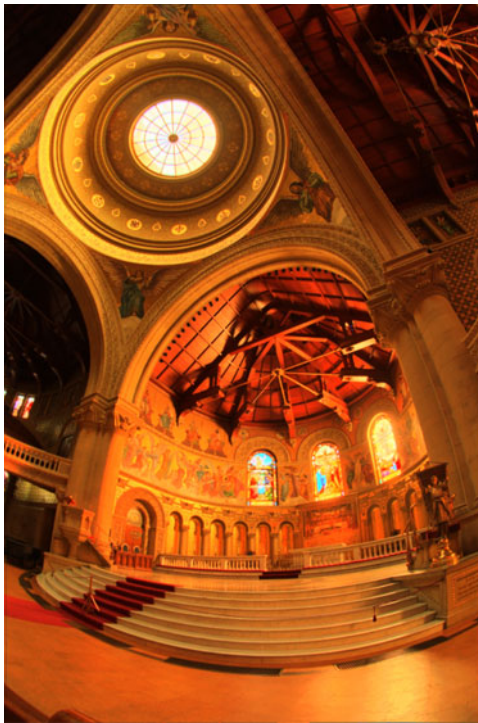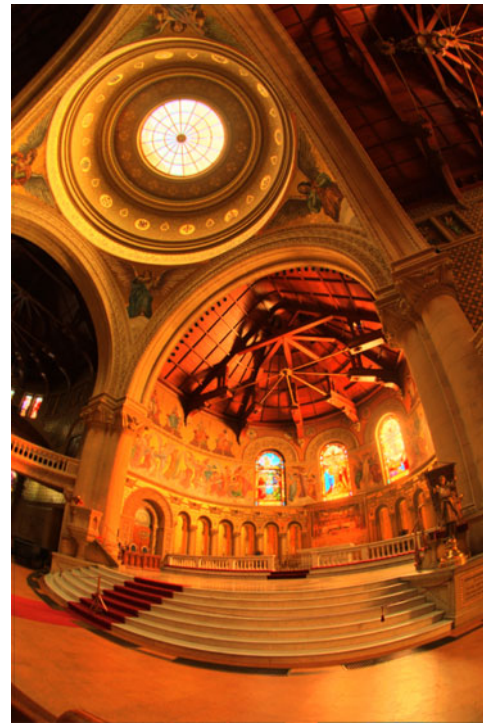
| Conventional method [1] | Proposed method |

Fig. 12. LDR images comparison (OpenEXR).

| Conventional method [1] | Proposed method |

Fig. 13. LDR images comparison (RGBE).

B) Comparison of tone mapped LDR images

This experiment applied tone mapping for 32 HDR images in the RGBE format and 42 images in the OpenEXR format using the proposed method and the conventional method [1] , and measured the PSNR of the tone mapped LDR images. Table 2 shows that condition of this experiment. The parameter k was set to 0.5.

Tables 3 and 4 show the PSNR of each LDR image. Table 5 shows the maximum, minimum, and average PSNR. In all cases, high PSNR values were obtained in the proposed method. The proposed method involves many rounding operations; however, the PSNR results still indicated high values. Therefore, the errors with accumulative rounding do not significantly affect the resulting LDR images. Figures 12 and 13 show LDR images obtained by the proposed method
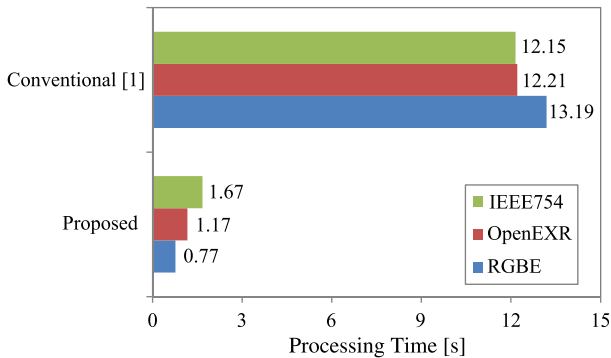
Fig. 14. The processing time of the proposed method and the conventional method [1].

Table 6. The memory usage of the conventional method [1] and the proposed method.

The data used in the methods	Memory usage (bits)	
	Conventional [1]	Proposed
An HDR image	$A \times B \times 192$	$A \times B \times 48$
World luminance	$A \times B \times 64$	$A \times B \times 16$
Geometric mean	64	16
Table (for $\log_2$ and 2^x)	−	8192

and the conventional 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 can execute the TMO with high accuracy, even though it involves the format conversion and the fixed-point arithmetic.

C) Comparison of the processing time

This experiment applied tone mapping for HDR images with 393216 pixels in the IEEE754, the OpenEXR, and the RGBE format using the proposed method and the conventional method [1], and measured the processing time of the methods. The experimental environment was with Marvell PXA270 ARM Processor 624MHz and 128MB RAM. Note that this processor does not have a FPU. The proposed method used 32-bit fixed-point arithmetic, and the conventional method [1] used 64-bit floating-point arithmetic in this experiment. Table 2 shows that condition of this experiment.

Figure 14 compares the processing time of the proposed method and the conventional method [1]. The proposed method was 7.26, 10.44, and 17.13 times faster than the conventional method when the input HDR image formats were the IEEE754, the OpenEXR, and the RGBE, respectively. Therefore, this experiment confirmed that the proposed method reduced the computational cost using fixed-point arithmetic.

D) Comparison of the memory usage

Table 6 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 this table can be conducted per pixel, and it is indicated in Fig. 8. The memory usage which depends on the image size is reduced by 75.0% in the proposed method. The method uses two pre-calculated tables, and each table consists of 16×256 bits. Therefore, the total memory usage of the proposed method which includes the tables is $A \times B \times 64 + 8208$ bits. The 8208 bits are canceled out if the size of the input HDR image is larger than 42 pixels.

E) Comparison with the other conventional methods

The proposed method was compared with the other conventional methods [20–22].

1) COMPARISON OF THE PROCESSING TIME

The processing times of the proposed method and [20–22] during tone mapping for HDR image (393216 pixels) were 0.77, 0.47, 78.42, and 0.55 s, respectively. Note that the CPUs used for the proposed method and [21, 22] were PXA270 ARM processor 624 MHz, and that of [20] was Pentium4 1800 MHz. The processing speed of [21] was very slow without FPU because it used floating-point for its internal arithmetic. The same in [22] gave high-speed processing because it is designed for the RGBE format exclusive use. The proposed method was a little slower than [22]; however, it can be applied for various formats. Moreover, considering the performance of CPUs, it is expected that the proposed method can be processed faster than [20].

2) COMPARISON OF THE PSNR VALUES

The [21, 22] are basis methods of the proposed method, and these methods are designed for the RGBE format. The experiment in this section applied tone mapping for 32 HDR images in the RGBE format, and compared the average PSNRs between these methods and the conventional method [1]. The PSNR values of the proposed method, [21, 22] were 55.67 dB, 58.78, and 56.18 dB, respectively. The proposed method and [22] used fixed-point arithmetic. On the other hand, [21] used floating-point arithmetic. Errors occurred using fixed-point arithmetic instead of floating-point arithmetic, and it was cause of PSNR drop. Comparing with [22], the PSNR dropped a little because a part of internal process of the proposed method was changed in order to adapt to the intermediate format. However, the average PSNR still maintained over 55 dB, and it is sufficiently high value.

V. CONCLUSION

This paper proposed a TMO with fixed-point arithmetic and low-memory. The proposed method can apply the TMO for multiple HDR image formats using an intermediate format. Using the intermediate format, the method can apply the tone mapping to two 8-bit integer numbers, which correspond to the exponent part and the mantissa part, separately. The method reduces the memory cost using

8-bit integer numbers instead of a 64-bit floating-point number. Furthermore, the method performs the TMO with only fixed-point arithmetic to reduce the computational cost. As a result, the method is effective on a computer with low-memory and a low-performance processor. The experimental results confirmed that the proposed method can execute the TMO with high accuracy, even though it is with fixed-point arithmetic and integer data. The future work is applying the method to local TMOs.

REFERENCES

[1] Reinhard, E.; Stark, M.; Shirley, P.; Ferwerda, J.: Photographic tone reproduction for digital images. ACM Trans. Graph., **21**(3) (2002), 267–276.

[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, MA, USA, 2010.

[3] Drago, F.; Myszkowski, K.; Annen, T.; Chiba, N.: Adaptive logarithmic mapping for displaying high contrast scenes. Comput. Graph. Forum, **22**(3) (2003), 419–426.

[4] Fattal, R.; Lischinski, D.; Werman, M.: Gradient domain high dynamic range compression. ACM Trans. Graph., **21**(3) (2002), 249–256.

[5] Iwahashi, M.; Kiya, H.: Efficient lossless bit depth scalable coding for HDR images, in *2013 Signal and Information Processing Association Annual Summit and Conf. (APSIPA)*, No. OS.37-IVM.16–4.

[6] Iwahashi, M.; Kiya, H.: Two layer lossless coding of HDR images, in *2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1340–1344.

[7] Xu, R.; Pattanaik, S.N.; Hughes, C.E.: High-dynamic-range still image encoding in JPEG2000. IEEE Trans. Comput. Graph. Appl., **25**(6) (2005), 57–64.

[8] Zhang, Y.; Reinhard, E.; Bull, D.: Perception-based high dynamic range video compression with optimal bit-depth transformation, in 2011, *IEEE Int. Conf. on Image Processing (ICIP)*, 1321–1324.

[9] Ward, G.; Simmons, M.: JPEG–HDR: a backwards-compatible, high dynamic range extension to JPEG, in *2006 ACM SIGGRAPH Courses*, No. 3.

[10] Kainz, F.; Bogart, R.; Hess, D.: The OpenEXR image file format, in *2003 ACM SIGGRAPH Technical Sketches and Applications*.

[11] Ward, G.: Real Pixels. Graphic Gems 2, *Academic Press*, San Diego, CA, USA, 1992, 80–83.

[12] Viitanen, T.; Jaaskelainen, P.; Esko, O.; Takala, J.: Simplified floating-point division and square root, in *2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2707–2711.

[13] Lampert, C.H.; Wirjadi, O.: Anisotropic Gaussian filtering using fixed point arithmetic, in *2006 IEEE Int. Conf. on Image Processing (ICIP)*, 1565–1568.

[14] Hass, K.J.: Synthesizing optimal fixed-point arithmetic for embedded signal processing, in *2010 IEEE Int. Midwest Symp. on Circuits and Systems (MWSCAS)*, 61–64.

[15] Cho, C.S.; Kim, Y.H.; Shin, H.S.; Choi, B.H.: Implementation of a fixed-point MPEG-4 scalable lossless coding encoder, in *2010 IEEE Int. Symp. on Broadband Multimedia Systems and Broadcasting (BMSB)*, 1–5.

[16] Marrot, J.; Bourennane, S.: Fast tensor signal filtering using fixed point algorithm, in *2008 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 921–924.

[17] Shen, J.; Jin, X.; Sun, H.: High dynamic range image tone mapping and retexturing using fast trilateral filtering. Vis. Comput. **23**(9–11) (2007), 641–650.

[18] Shen, J.; Fang, S.; Zhao, H.; Jin, X.; Sun, H.: Fast approximation of trilateral filter for tone mapping using a signal processing approach. Signal Process. **89**(5) (2009), 901–907.

[19] Duan, J.; Qiu, G: Fast tone mapping for high dynamic range images, in *2004 Int. Conf. on Pattern Recognition (ICPR)*, 847–850.

[20] Thakur, S.K.; Sivasubramanian, M.; Nallaperumal, K.; Marappan, K.; Vishwanath, N.: Fast tone mapping for high dynamic range images, in *2013 IEEE Int. Conf. on Computational Intelligence and Computing Research (ICCIC)*, 1–4

[21] Murofushi, T.; Iwahashi, M.; Kiya, H.: An integer tone mapping operation for hdr images expressed in floating point data, in *2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2479–2483.

[22] Dobashi, T.; Murofushi, T.; Iwahashi, M.; Kiya, H.: A Fixed-point tone mapping operation for HDR images in the RGBE format, in *2013 Signal and Information Processing Association Annual Summit and Conf. (APSIPA)*, No. OS.37-IVM.16–4.

[23] "Information technology – Microprocessor Systems – Floating-Point arithmetic," ISO/IEC/IEEE 60559, 2011.

Toshiyuki Dobashi received his B. Eng. and M. Eng. degrees from Tokyo Metropolitan University, Japan in 2012 and 2014, respectively. He joined Canon Inc. in 2014. His research interests include image processing.

Atsushi Tashiro received his B. Eng. degree from Tokyo Metropolitan University, Japan in 2014. He joined Hitachi Co., Ltd. in 2014. His research interests include image processing.

Masahiro Iwahashi received his B.Eng., M.Eng., and D.Eng. degrees in Electrical Engineering from Tokyo Metropolitan University in 1988, 1990, and 1996, respectively. In 1990, he joined Nippon Steel Co. Ltd. From 1991 to 1992, he was dispatched to Graphics Communication Technology Co. Ltd. In 1993, he joined Nagaoka University of Technology, where he is currently a Professor of Department of Electrical Engineering, Faculty of Technology. From 1995 to 2001, he served concurrently as a lecturer of Nagaoka Technical College. From 1998 to 2001, he was dispatched to Thammasat University, Thailand, and to the Electronic Engineering Polytechnic Institute of Surabaya, Indonesia, as a JICA expert. His research interests are in the area of digital signal processing, multi-rate systems, and image compression. He served as an editorial committee member of the transaction on fundamentals of IEICE from 2007 to 2011.

Hitoshi Kiya received his B.Eng. and M.Eng. degrees from Nagaoka University of Technology, Japan, in 1980 and 1982, respectively, and his D.Eng. degree from Tokyo Metropolitan University in 1987. In 1982, he joined Tokyo Metropolitan University as an Assistant Professor, where he became a Full Professor in 2000. From 1995 to 1996, he attended the University of Sydney, Australia as a Visiting Fellow. He currently serves as the Chair of IEEE SPS Japan Chapter, an Associate Editor for IEEE Trans. Image Processing and IEEE Trans. Information Forensics and Security, respectively. He also served as the President of

IEICE Engineering Sciences Society (ESS), an Associate Editor of IEEE Trans. Signal Processing, the Editor-in-Chief for IEICE Fundamentals Review, a Vice President of APSIPA, a Member of the Board of Governors of APSIPA, and the Editor-in-Chief for IEICE ESS publications. His research interests are in the area of signal and image processing including multirate signal processing, wavelets, video coding, compressed-domain video manipulation, and security for multimedia. He received the IWAIT Best Paper Award in 2014, the ITE Niwa-Takayanagi Best Paper Award in 2012, the Telecommunications Advancement Foundation Award in 2011, the IEICE ESS Contribution Award in 2010, and the IEICE Best Paper Award in 2008. He is a Fellow Member of the IEICE and the ITE, and a Senior Member of the IEEE.