

## RESEARCH

# Bit-Depth Scalable Lossless Coding for High Dynamic Range Images

Masahiro Iwahashi<sup>1\*</sup>, Taichi Yoshida<sup>1</sup>, Norrima Binti Mokhtar<sup>2</sup> and Hitoshi Kiya<sup>3</sup>

## Abstract

In this paper, we propose a bit-depth scalable lossless coding method for high dynamic range (HDR) images based on a reversible logarithmic mapping. HDR images are generally expressed as floating-point data, such as in the OpenEXR or RGBE formats. Our bit-depth scalable coding approach outputs base layer data and enhancement layer data. It can reconstruct the low dynamic range (LDR) image from the base layer data, and reconstructs the HDR image by adding the enhancement layer data. Most previous two-layer methods have focused on the lossy coding of HDR images. Unfortunately, the extension of previous lossy methods to lossless coding does not significantly compress the enhancement layer data. This is because the bit depth becomes very large, especially for HDR images in floating-point data format. To tackle this problem, we apply a reversible logarithmic mapping to the input HDR data. Moreover, we introduce a format conversion to avoid any degradation in the quality of the reconstructed LDR image. The proposed method is effective for both OpenEXR and RGBE formats. Through a series of experiments, we confirm that the proposed method decreases the volume of compressed data while maintaining the visual quality of the reconstructed LDR images.

**Keywords:** high dynamic range imaging; lossless coding; bit-depth scalable coding

## 1 Introduction

Image data compression technologies, such as the JPEG 2000 international standard [1, 2], allow high quality images to be transmitted via worldwide digital communication networks. Digital cinema and 4K images are remarkable examples of such technology [3, 4]. These images require a huge number of pixels to express fine textures at high spatial resolutions.

Recently, high dynamic range (HDR) images have attracted considerable attention [5]. These images have a high resolution of pixel values, i.e., numerous pixel tones. Compared with the current standard for low dynamic range (LDR) images, which are expressed in 8 bits, HDR images have an extremely long bit depth and high dynamic range of pixel values. To fully utilize this dynamic range under limited memory space, the pixel values are expressed as floating-point data, such as in OpenEXR or RGBE format [6, 7]. This paper focuses on the compression of HDR images in these data formats. Moreover, the proposed method, referred to as bit-depth scalable coding, is backward compatible with a standard coding method for LDR images.

Bit-depth scalable coding outputs compressed data in two layers, a base layer and an enhancement layer. From the bit stream in the base layer, the LDR image is decoded with a standard lossy decoder. By adding the bit stream in the enhancement layer, the original HDR image can be decoded without any loss. This scalable coding system has the advantage that it can directly accommodate both HDR and LDR users. Therefore, the system has attracted many researchers, and a number of variations have been reported [8–15].

Ward et al. [8] proposed a backward compatible bit-depth scalable coding method in which the original HDR color image is tone mapped in the base layer to produce an LDR image that is compressed by the JPEG international standard encoder. The enhancement layer then embeds the luminance ratio of the LDR and HDR images. The original HDR color image is decoded by multiplying the luminance ratio in the enhancement layer and the LDR color image in the base layer. This method has been extended to video signals, and has attracted attention as a bit-depth scalable video coding method in international standardization activities [9–11, 16]. For still images, Khan introduced a piecewise linear model of a tone mapping [12]. Jinno et al. improved the coding efficiency in the enhancement layer by replacing the ratio with a low-

\* Correspondence: iwahashi@vos.nagaokaut.ac.jp

<sup>1</sup> Department of Electrical, Electronics and Information Engineering, Nagaoka University of Technology, Nagaoka, Niigata, Japan  
Full list of author information is available at the end of the article

pass-filtered HDR image [15]. However, these reports focused on ‘lossy’ coding for HDR images.

Unlike these previous reports, we discuss the ‘lossless’ coding of HDR images under a scalable coding scheme that is compatible with lossy LDR image coding. The lossless coding of HDR images is especially important for storing and archiving original visual data such as medical, artistic, and astrograph images. Such data can be used for diagnosis based on medical images, analysis of astrograph images, art preservation, and bio-medical detections [17, 18].

First, we discuss a baseline method [19], that was simply extended to lossless scalable coding from a non-scalable HDR image coding method [20]. Although the baseline method is straightforward and easy to implement, the coding efficiency in the enhancement layer is not satisfactory. To cope with this problem, we introduced a reversible logarithmic mapping and reduced the dynamic range of the HDR images [19, 21]. This approach was shown to be effective for compressing data in the enhancement layer. However, the method was limited to the OpenEXR format [6]. Another representative format, referred to as RGBE [7], has been ignored.

In this paper, we improve on our previous conference papers [19, 21] and add some theoretical analysis. First, we show that a simple extension of the reversible logarithmic mapping (Rev) to the RGBE format degrades the visual quality of the decoded LDR images. To avoid this problem, we introduce a format conversion (Cnv) to the system. We demonstrate that simply extending Rev magnifies the quantization error added by the lossy coding in the base layer. Second, we analyze the theoretical basis for why our method improves the coding efficiency of the system. We estimate how the bit depth of the residual image to be encoded in the enhancement layer is reduced by Rev. We also explain why the simply extended Rev degrades the LDR images, and why Cnv improves their quality in the RGBE format.

This paper is organized as follows. In Sec. 2, we describe two floating-point data formats and a non-scalable HDR image coding method. A baseline scalable coding method that simply extends the non-scalable coding approach is then summarized in Sec. 3, and the concept and implementation of the proposed method are introduced in Sec. 4. The theoretical analysis is described in Sec. 5, and our experimental results are summarized in Sec. 6. Finally, we present our conclusions in Sec. 7.

## 2 Data format and non-scalable coding

We first describe two floating-point data formats for HDR images. A non-scalable lossy coding method,

which is extended to scalable lossless coding of HDR images in the next section, is also summarized.

### 2.1 Type A format of HDR images

To date, there are two well-known representative data formats for HDR images. One is the OpenEXR floating-point data format [6], and the other is the RGBE data format [7].

In the OpenEXR data format, a pixel value  $x_{H,c}$  of an HDR image is described by an exponent value  $x_{E,c}$ , mantissa value  $x_{M,c}$ , and sign value  $x_{S,c}$  as

$$\begin{cases} x_{H,c} = (-1)^{x_{S,c}}(1 + 2^{-10}x_{M,c})2^{-15+x_{E,c}} \\ \text{if } x_{E,c} \neq 0 \end{cases} \quad (1)$$

and

$$\begin{cases} x_{H,c} = (-1)^{x_{S,c}}(0 + 2^{-10}x_{M,c})2^{-14} \\ \text{if } x_{E,c} = 0 \end{cases} \quad (2)$$

for a color component  $c \in \{R, G, B\}$ . The exponent, mantissa, and sign values are given as integers in the ranges

$$x_{M,c} \in [0, 2^{10} - 1], \quad x_{E,c} \in [0, 2^5 - 1], \quad x_{S,c} \in [0, 1]. \quad (3)$$

The mantissa, exponent, and sign have depths of 10 bits, 5 bits, and 1 bit, respectively. Therefore, a pixel value of an HDR image is expressed in  $10 + 5 + 1 = 16$  bits for each color component. Note that in certain special cases,  $x_{E,c} = 31$  [6].

In the remainder of this paper, we denote the exponent, mantissa, and sign of each color component as a vector

$$\begin{cases} \mathbf{x}_E = [x_{E,R}, \quad x_{E,G}, \quad x_{E,B}]^T \\ \mathbf{x}_M = [x_{M,R}, \quad x_{M,G}, \quad x_{M,B}]^T \\ \mathbf{x}_S = [x_{S,R}, \quad x_{S,G}, \quad x_{S,B}]^T \end{cases} \quad (4)$$

and define the HDR image data  $\mathbf{x}_D$  as

$$\mathbf{x}_D = [\mathbf{x}_E, \quad \mathbf{x}_M, \quad \mathbf{x}_S]. \quad (5)$$

Using these vectors, we denote (1) and (2) as

$$\mathbf{x}_H = \text{Flt}_A(\mathbf{x}_D), \quad (6)$$

where the pixel value of the HDR image  $\mathbf{x}_H$  is

$$\mathbf{x}_H = [x_{H,R}, \quad x_{H,G}, \quad x_{H,B}]^T. \quad (7)$$

Hereafter, we refer to OpenEXR as the ‘type A’ format.

## 2.2 Type B format of HDR images

In the RGBE data format, a pixel value of an HDR image  $x_{H,c}$  is given as

$$x_{H,c} = \begin{cases} \frac{x_{M,c} + 0.5}{256} 2^{x_{E,0}-128} & \text{if } x_{E,0} \neq 0 \\ 0 & \text{if } x_{E,0} = 0 \end{cases} \quad (8)$$

for a color component  $c \in \{R, G, B\}$ . Both the mantissa and exponent have depths of 8 bits, i.e.,

$$x_{M,c} \in [0, 2^8 - 1], \quad x_{E,0} \in [0, 2^8 - 1]. \quad (9)$$

The exponent  $x_{E,0}$  is commonly used among three color components. In this format, a pixel value is expressed with a total of 32 bits [7]. Using the vectors, we denote (8) as

$$\mathbf{x}_H = \text{Flt}_B(\mathbf{x}_D). \quad (10)$$

Hereafter, we refer to RGBE as the ‘type B’ format. Note that, for a type B image,  $\mathbf{x}_H$  in (10) is non-negative. In contrast,  $\mathbf{x}_H$  in (6) for a type A image can be negative, zero, or positive.

## 2.3 Non-scalable lossy coding

Figure 1 illustrates the ‘HDR image coding in JPEG 2000’ reported in [20]. At the encoder, the HDR image data  $\mathbf{x}_D$  is converted into the pixel value  $\mathbf{x}_H$  by Flt, where Flt denotes Flt<sub>A</sub> in (6) for type A images and Flt<sub>B</sub> in (10) for type B images. The logarithmic function  $\log_e$  is applied to each color component of  $\mathbf{x}_H$ . Note that pixel values that are less than or equal to zero are first clipped to the minimum positive pixel value in the image. In terms of the signal-to-noise ratio (SNR) of the variances, the effect on the LDR images is almost zero. At worst, of the 9 test images considered in this paper, the SNR is less than  $10^{-2}$  [%] for a type A ‘Still Life’ image. The effect on HDR images is also limited, with an SNR of less than  $10^{-10}$  [%] for the same input image.

The pixel values are normalized to the range  $[0, 255]$  by

$$\text{Nrm}(x) = (x - \min X) \cdot \frac{255}{\max X - \min X} \quad (11)$$

for  $X = \{x | x \in \text{image}\}$ , where  $\min X$  and  $\max X$  are the minimum and maximum pixel values in the set  $X$ , respectively. Because the input values to the encoder must be integers, the results are rounded to be integers. Namely,

$$\mathbf{x}_B = \text{Rnd}(\text{Nrm}(\log_e(\text{Clp}(\mathbf{x}_H)))) \quad (12)$$

is fed into the encoder, where Rnd and Clp are the rounding and clipping operations, respectively. In the decoder, the HDR pixel values  $\mathbf{y}_H$  are recovered from the decoded image  $\mathbf{y}_B$  with the inverse of each Nrm and  $\log_e$ .

In this paper, we extend this method to the scalable lossless coding of HDR images. The tone mapping operator Tmo described in 2.4 is added to this procedure as ‘part A’ to display color LDR images with better quality.

## 2.4 Tone mapping operation

We now summarize the tone mapping operator for color images based on the Hill function [5]. A pixel value of the HDR image  $y_{H,c}$  is tone mapped to  $y_{L,c}$  of the LDR image as

$$y_{L,c} = \text{Rnd}(255 y_{H,c} \cdot y_{L,Y} / y_{H,Y}) \quad (13)$$

for  $c \in \{R, G, B\}$ , where

$$\begin{cases} y_{H,Y} = 0.27 y_{H,R} + 0.67 y_{H,G} + 0.06 y_{H,B} \\ y_{L,Y} = \text{Hill}(y_{H,Y} / \bar{Y}_{H,Y}), \end{cases} \quad (14)$$

and the Hill function is defined as

$$\text{Hill}(x) = \frac{x^a}{x^a + b^a}. \quad (15)$$

In (14),  $\bar{Y}_{H,Y}$  is defined as

$$\bar{Y}_{H,Y} = \exp(\text{Ens}(\log_e(y_{H,Y}))), \quad (16)$$

where Ens( $\cdot$ ) denotes the ensemble average over all positive values of  $y_{H,Y}$  in the image.  $a$  and  $b$  are user-set parameters. In our experiments, we use  $(a, b) = (1, 1)$ . We denote the tone mapping in (13) as

$$\mathbf{y}_L = \text{Rnd}(\text{Clp}'(\text{Tmo}(\mathbf{y}_H))), \quad (17)$$

where

$$\begin{cases} \mathbf{y}_H = [y_{H,R}, y_{H,G}, y_{H,B}]^T \\ \mathbf{y}_L = [y_{L,R}, y_{L,G}, y_{L,B}]^T \end{cases} \quad (18)$$

for color components. Because the output values of Tmp exceed 8-bit integers for color images, we clip the output values to the range  $[0, 255]$  with Clp’.

## 3 Baseline method

The baseline scalable lossless coding method is simply an extension of the non-scalable lossy coding method. We now summarize this baseline method, as well as the problem considered in this paper.

### 3.1 Scalable lossless coding

Figure 2 illustrates the baseline method, which we use as a reference in this paper. This is a simple extension of the non-scalable lossy coding in Fig. 1 to the scalable lossless coding of HDR images. ‘Part B’ denotes the processes that have been added.

To achieve the lossless coding of HDR images,  $\mathbf{x}_H$  is converted into the integer value  $\mathbf{x}_I$  by the reversible integer mapping  $\text{Int}$  detailed in Sec. 3.2. Note that the inverse mapping  $\text{Int}^{-1}$  reconstructs the original value without any loss. The procedure for generating the LDR images is almost the same as the method in Fig. 1. The bit stream needed to reconstruct the LDR image is embedded in the base layer. In the enhancement layer, the integer value  $\mathbf{y}_I$  is reconstructed from the decoded LDR image  $\mathbf{y}_B$  with the inverse normalization  $\text{Nrm}^{-1}$ , the exponential function  $\exp$ , and the rounding operation

$$\mathbf{y}_I = \text{Rnd}(\exp(\text{Nrm}^{-1}(\mathbf{y}_B))). \quad (19)$$

Finally, the residual

$$\mathbf{e}_I = \mathbf{y}_I - \mathbf{x}_I \quad (20)$$

is encoded with a lossless coding method to generate the bit stream in the enhancement layer.

### 3.2 Reversible integer mapping

The reversible integer mapping  $\text{Int}$  from the real value  $\mathbf{x}_H$  to the integer value  $\mathbf{x}_I$  was introduced in [19]. It is defined as

$$x_{I,c} = \begin{cases} (-1)^{x_{S,c}} (x_{H,c} 2^{25 - \min X_E} - 2^{10}) & \text{if } \min X_E \neq 0 \\ (-1)^{x_{S,c}} x_{H,c} 2^{24} & \text{if } \min X_E = 0 \end{cases} \quad (21)$$

where  $c \in \{R, G, B\}$  and  $X_E = \{x_{E,c} | x_{E,c} \in \text{image}\}$  for type A images. This is a simple scaling applied to the rational number  $x_{H,c}$  in (1) so that it becomes an integer. In other words, we shift the decimal point to the right. Note that the minimum  $\min X_E$  of all the pixel values  $\mathbf{x}_E$  in the image is stored and embedded into the bit stream. We denote the mapping in (21) as

$$\mathbf{x}_I = \text{Int}_A(\mathbf{x}_H) \quad (22)$$

for

$$\mathbf{x}_I = [x_{I,R}, x_{I,G}, x_{I,B}]^T. \quad (23)$$

Similarly, a mapping for type B images can be defined as

$$\mathbf{x}_{I,c} = \left( 256x_{H,c} 2^{128 - \min X_E^+} - 0.5 \right) 2 + 1 \quad (24)$$

where  $c \in \{R, G, B\}$  and  $X_E^+ = \{x_{E,c} | x_{E,c} > 0\}$ . Note that the minimum  $\min X_E^+$  of all the positive pixel values  $\mathbf{x}_E > 0$  in the image is stored and embedded into the bit stream. We denote this mapping as

$$\mathbf{x}_I = \text{Int}_B(\mathbf{x}_H) \quad (25)$$

for type B images.

Note that the inverse of this mapping recovers the original value without any loss. Therefore, the baseline method becomes lossless for the original HDR image.

### 3.3 Problem setting

In this paper, we tackle the following limitation of the baseline method. As a result of the reversible integer mapping, the residual  $\mathbf{e}_I$  in (20) requires a very large bit depth. It is somewhat difficult to compress this data volume in the high bit rate coding of the LDR image. This is because  $\mathbf{e}_I$  is a magnified version of the coding noise  $\mathbf{e}_B = \mathbf{y}_B - \mathbf{x}_B$ . In lossy coding, the noise  $\mathbf{e}_B$  is added in the base layer, and is magnified by  $\text{Nrm}^{-1}$  and  $\exp$  as indicated in (19). Because this noise tends to have a weak correlation, the difference  $\mathbf{e}_I$  also has a weak correlation. Therefore, the data volume of the enhancement layer becomes huge. Note that the correlation of  $\mathbf{e}_I$  increases in the low bit rate coding of the LDR image. This is investigated in Sec. 6.

To cope with this problem, we previously introduced the reversible logarithmic mapping (Rev) to reduce the bit depth of the residual image [19]. However, in this previous report, we only presented experimental results without any theoretical endorsement. In this paper, we theoretically compare  $\text{Int}$  and  $\text{Rev}$  in respect of the bit depth of the residual image in the enhancement layer.

In addition,  $\text{Rev}$  has been limited to the type A format, ignoring type B. In this paper, we show that a simple extension of  $\text{Rev}$  to the type B format degrades the LDR images. To avoid this problem, we introduce a format conversion (Cnv) from type B to type A in the base layer. We present a theoretical justification for why the simply extended  $\text{Rev}$  degrades the LDR images and  $\text{Cnv}$  improves its quality for type B images.

## 4 Proposed method

The reversible logarithmic mapping (Rev) is introduced to reduce the data volume of the enhancement layer. In particular, for type B format images, the format conversion (Cnv) is introduced to maintain the visual quality of the LDR images.

#### 4.1 Type I method for Type A format images

Figure 3 illustrates the proposed type I method. Instead of Flt and Int in the baseline method (Fig. 2), the reversible logarithmic mapping Rev defined in Sec. 4.2 is applied to the HDR data  $\mathbf{x}_D$  to produce  $\mathbf{x}_R$ . This is converted to an 8-bit depth integer  $\mathbf{x}_B$  as

$$\mathbf{x}_B = \text{Rnd}(\text{Nrm}(\text{Clp}(\mathbf{x}_R))) \quad (26)$$

and fed into the lossy encoder, which outputs the bit stream in the base layer. The reconstructed pixel  $\mathbf{y}_B$  given by the decoder is inversely normalized and rounded to an integer as

$$\mathbf{y}_R = \text{Rnd}(\text{Nrm}^{-1}(\mathbf{y}_B)). \quad (27)$$

Then, the difference

$$\mathbf{e}_R = \mathbf{x}_R - \mathbf{y}_R \quad (28)$$

is encoded with the lossless encoder to generate the bit stream in the enhancement layer. In the decoder,  $\mathbf{y}_R$  is added to  $\mathbf{e}_R$  to recover  $\mathbf{x}_R$ . Applying the inverse of Rev, the original HDR data  $\mathbf{x}_D$  are retrieved without any loss. Namely, they are recovered as

$$\mathbf{x}_D = \text{Rev}^{-1}(\mathbf{e}_R + \mathbf{y}_R). \quad (29)$$

The LDR image  $\mathbf{y}_L$  is reconstructed from the decoded image  $\mathbf{y}_B$  in a similar way to the baseline method with a compensation factor (Cmp). This recovers the HDR pixel value  $\mathbf{y}_H$ , and then applies the tone mapping operation Tmo as

$$\begin{aligned} \mathbf{y}_L &= \text{Rnd}(\text{Tmo}(\text{Flt}(\text{Rev}^{-1}(\mathbf{y}_R)))) \\ &= \text{Rnd}(\text{Cmp}(\mathbf{y}_R)). \end{aligned} \quad (30)$$

It is also possible to display  $\mathbf{y}_B$  as an LDR image without using Cmp. In this case,  $\mathbf{y}_B$  in the proposed method is almost the same as that of the baseline method as illustrated in Fig. 5. There are two approaches that use the Hill function in (15) to generate the LDR image  $\mathbf{y}_L$  exemplified in Fig. 6. The first introduces Cmp in the encoding process, and the second introduces Cmp in the decoding process. The former case is convenient for data receivers, because it is not necessary to add Cmp to a standard decoder. However, this increases the data volume of the enhancement layer. In this paper, we employ the latter approach.

#### 4.2 Reversible logarithmic mapping

In the proposed type I method illustrated in Fig. 3, the reversible logarithmic mapping is applied to generate

the integer value  $\mathbf{x}_R$ . This technique was originally introduced in [22]. The mapping for type A images is defined as

$$x_{R,c} = (-1)^{x_{S,c}} ((x_{E,c} - \min X_E) 2^{10} + x_{M,c}) \quad (31)$$

for  $c \in \{R, G, B\}$ . We denote this mapping as

$$\mathbf{x}_R = \text{Rev}_A(\mathbf{x}_D) \quad (32)$$

for

$$\mathbf{x}_R = [x_{R,R}, x_{R,G}, x_{R,B}]^T. \quad (33)$$

This mapping approximates the logarithm of an HDR image  $\mathbf{x}_H$ . Substituting  $x_{E,c}$  from (1), i.e.,

$$x_{E,c} = \log_2 x_{H,c} - \log_2(1 + 2^{-10} x_{M,c}) + 15, \quad (34)$$

for positive values in (31), we have

$$x_{R,c} = (\log_2 x_{H,c} + 15 - \min X_E - \epsilon_A) 2^{10} \quad (35)$$

where

$$\begin{aligned} \epsilon_A &= \log_2(2^{-10} x_{M,c} + 1) - 2^{-10} x_{M,c} \\ &= \log_2 \frac{\delta_A + 1}{2^{\delta_A}} \end{aligned} \quad (36)$$

for  $\delta_A = 2^{-10} x_{M,c}$ . As indicated in (35),  $\text{Rev}_A$  generates a good approximation of the logarithm of the HDR image [23, 24]. The approximation error is relatively small, as  $\epsilon_A$  fluctuates around 0.06 depending on the mantissa. Therefore,  $\text{Nrm}(\mathbf{x}_R)$  becomes close to  $\mathbf{x}_B$  in (12). This is encoded with a standard lossy encoder to generate the bit stream in the base layer.

The reversible logarithmic mapping is suitable for lossless scalable coding because it one-to-one maps an integer to an integer. Therefore, its inverse mapping reconstructs the original integer values without any loss, i.e., Rev is ‘reversible’. This property also reduces the dynamic range of the mapped integer values. We have experimentally confirmed [21] that the residual in the enhancement layer  $\mathbf{e}_R$  has a lower bit depth than that of  $\mathbf{e}_I$  in the baseline method. We provide the theoretical basis for this observation in Sec. 5.1.

#### 4.3 Type I method for Type B format images

For type A images,  $\text{Rev}_A$  in (32) is applied in Fig. 3. For type B images, a direct extension of  $\text{Rev}_A$  can be defined as

$$x_{R,c} = \begin{cases} x_{E,0}^* 2^8 + x_{M,c} + 1 & \text{if } x_{E,0} \neq 0 \\ 0 & \text{if } x_{E,0} = 0 \end{cases} \quad (37)$$

for  $c \in \{R, G, B\}$  and

$$x_{E,0}^* = x_{E,0} - \min X_E^+ + 1. \quad (38)$$

We denote this mapping as

$$\mathbf{x}_R = \text{Rev}_B(\mathbf{x}_D), \quad (39)$$

and apply this to type B images in the type I method.

Note that the depth of  $\mathbf{x}_R$  from  $\text{Rev}_B$  is a maximum of 16 [bit] for each color component. Therefore, it costs 48 [bit] for all the color components, which exceeds the original 32-bit data. However, using the reversible color transform (RCT) in JPEG 2000 lossless coding reduces the cost by 16 bits. The RCT is defined as

$$\begin{cases} x_1 = \lfloor (x_R + 2x_G + x_B) / 4 \rfloor \\ x_2 = x_B - x_G \\ x_3 = x_R - x_G \end{cases} \quad (40)$$

Because the 2nd and 3rd row of this RCT take the difference between the color components, the exponent term  $x_{E,0}^*$  in (37) disappears. As a result, the bit depth becomes  $48 - 16 = 32$  [bit] in total. Furthermore, the exponent term is less than 5 [bit] in the type B images tested in our experiment. Therefore, in practice, the system can compress the data volume.

In this paper, we show that the quality of LDR images is degraded in this directly extended  $\text{Rev}_B$  for type B images. Figure 4 shows some LDR images produced by the proposed type I and type II methods. The former has lower quality than the latter, with a peak SNR (PSNR) of 20.79 [dB] compared with 29.08 [dB] at the same bit rate of 5.23 [bppc] in the base layer. The reason for this is analyzed in Sec. 5.2. To solve this problem, we introduce the following format conversion.

#### 4.4 Type II method for Type B format images

Figure 7 illustrates the modified method for type B images. We refer to this as the type II method. The type II approach includes the format conversion (Cnv). First,  $\text{Rev}_B$  converts the HDR data  $\mathbf{x}_D$  into type B  $\mathbf{x}_R$ . In the figure, this is denoted as  $\mathbf{x}_R^{(B)}$  to clearly indicate the type. The conversion introduced in this paper is defined as

$$\begin{cases} \text{Cnv}(\mathbf{x}) = \text{Rev}_A(\text{Flt}_A^{-1}(\text{Flt}_B(\text{Rev}_B^{-1}(\mathbf{x})))) \\ \mathbf{x} = \text{Clp}(\mathbf{x}_R^{(B)}) \end{cases} \quad (41)$$

as illustrated in Fig. 8. In the proposed type II method,

$$\mathbf{x}_B = \text{Rnd}(\text{Nrm}(\text{Cnv}(\text{Clp}(\mathbf{x}_R^{(B)})))) \quad (42)$$

is encoded with the lossy encoder.

As a result of this conversion, the type B image is temporarily converted into a type A image in the base layer. Therefore, the problem caused by  $\text{Rev}_B$  can be avoided, and the quality of the LDR image is improved compared to that given by applying the type I method to type B images. This assertion is theoretically endorsed in Sec. 5.2. This conversion is reversible as far as a large enough bit depth is assigned to values inside the process. However, reversion is not always necessary, as the system becomes lossless for HDR images in as much as the  $\mathbf{y}_R$  are exactly the same in the encoder and the decoder, even though Cnv is not perfectly reversible.

## 5 Theoretical analysis

We now present a theoretical analysis of why the proposed method reduces the bit depth of the enhancement layer. The rationale for introducing the format conversion is also explained.

### 5.1 Bit depth of the enhancement layer

We estimate the bit depth of the residual  $\mathbf{e}_I$  in the baseline method and  $\mathbf{e}_R$  in the proposed method, and theoretically demonstrate that the bit depth of the proposed method is smaller than that of the baseline method.

The bit depth of pixel values in an image  $\mathbf{x}$  is defined as

$$B_{dp}(\mathbf{x}) = \log_2(\max X - \min X + 1), \quad (43)$$

where  $\max X$  and  $\min X$  denote the maximum and minimum pixel values in the image. We must calculate the maximum of  $\mathbf{e}_I$  in the baseline method to estimate its bit depth. In Fig. 2, the relations

$$\begin{cases} \mathbf{x}_B = \text{Rnd}(\text{Nrm}(\log_e \text{Clp}(\mathbf{x}_I))) \\ \mathbf{y}_I = \text{Rnd}(\exp(\text{Nrm}^{-1}(\mathbf{y}_B))) \end{cases} \quad (44)$$

are modeled as

$$\begin{cases} \mathbf{x}_B = \text{Nrm}(\log_e \mathbf{x}_I) + e_1 \\ \mathbf{y}_I = \exp(\text{Nrm}^{-1}(\mathbf{y}_B)) + e_2 \end{cases} \quad (45)$$

for positive values of  $\mathbf{x}_I$ , where  $e_1, e_2 \in [-0.5, 0.5]$  are rounding errors due to Rnd in (44). Therefore, the maximum of

$$\begin{aligned} \mathbf{e}_I &= \mathbf{y}_I - \mathbf{x}_I \\ &= \exp(\text{Nrm}^{-1}(\mathbf{y}_B)) - \mathbf{x}_I + e_2 \\ &= \exp(\text{Nrm}^{-1}(\mathbf{x}_B + \mathbf{e}_B)) - \mathbf{x}_I + e_2 \\ &= \exp(\text{Nrm}^{-1}(\text{Nrm}(\log_e \mathbf{x}_I) \\ &\quad + \mathbf{e}_B + e_1)) - \mathbf{x}_I + e_2 \end{aligned} \quad (46)$$

is estimated as

$$\max E_I = \max E_B \cdot \max X_I \cdot C/255 \quad (47)$$

for

$$C = \log_e(\max X_I) - \log_e(\min X_I^+), \quad (48)$$

as detailed in Appendix A. Substituting

$$\min E_I = -\max E_I \quad (49)$$

and (47) into (43), the bit depth of  $\mathbf{e}_I$  is estimated as

$$\begin{aligned} B_{dp}(\mathbf{e}_I) &= \log_2(2 \cdot \max E_I + 1) \\ &\approx \log_2(\max E_B \cdot \max X_I \cdot C/255) + 1, \end{aligned} \quad (50)$$

giving the bit depth of the residual of the baseline method. Similarly, using the model

$$\begin{cases} \mathbf{x}_B = \text{Nrm}'(\mathbf{x}_R) + e'_1 \\ \mathbf{y}_R = \text{Nrm}'^{-1}(\mathbf{y}_B) + e'_2 \end{cases} \quad (51)$$

in the proposed method, the maximum of

$$\begin{aligned} \mathbf{e}_R &= \mathbf{y}_R - \mathbf{x}_R \\ &= \text{Nrm}'^{-1}(\mathbf{x}_B + \mathbf{e}_B) - \mathbf{x}_R + e'_2 \\ &= \text{Nrm}'^{-1}(\text{Nrm}'(\mathbf{x}_R) + \mathbf{e}_B + e'_1) \\ &\quad - \mathbf{x}_R + e'_2 \end{aligned} \quad (52)$$

is given as

$$\max E_R = \max E_B \cdot \max X_R/255, \quad (53)$$

as shown in Appendix B. Substituting

$$\min E_R = -\max E_R. \quad (54)$$

and (53) into (43), the bit depth of  $\mathbf{e}_R$  can be estimated as

$$\begin{aligned} B_{dp}(\mathbf{e}_R) &= \log_2(2 \cdot \max E_R + 1) \\ &\approx \log_2(\max E_B \cdot \max X_R/255) + 1, \end{aligned} \quad (55)$$

giving the bit depth of the residual of the proposed method.

We can now compare  $\mathbf{e}_I$  and  $\mathbf{e}_R$  in terms of bit depth. The error in the base layer  $\mathbf{e}_B$  is composed of errors due to the rounding before applying the lossy encoder, as well as quantization errors added by the lossy coding. Therefore, the maximum and minimum of

$$\mathbf{e}_B = \mathbf{y}_B - \mathbf{x}_B \quad (56)$$

are

$$\max E_B = -\min E_B = Q, \quad (57)$$

where  $Q$  is determined by the quantization step size of the lossy coding in the base layer. Taking the difference between (50) and (55), we have

$$\begin{aligned} \Delta B_{dp} &= B_{dp}(\mathbf{e}_I) - B_{dp}(\mathbf{e}_R) \\ &= \log_2(\max E_B \cdot \max X_I/255 \cdot C) \\ &\quad - \log_2(\max E_B \cdot \max X_R/255), \end{aligned} \quad (58)$$

and therefore

$$\Delta B_{dp} = \log_2 \frac{\max X_I \cdot C}{\max X_R} \quad (59)$$

is the difference in bit depth. From (1), (21), and (31), the maxima of  $\mathbf{x}_I$  and  $\mathbf{x}_R$  are expressed as

$$\begin{cases} \max X_I = (2^{C^*} - 1)2^{10} \approx 2^{C^*} \cdot 2^{10} \\ \max X_R = C^* \cdot 2^{10} \end{cases} \quad (60)$$

for

$$C^* = \max X_E - \min X_E + \gamma, \quad (61)$$

where  $\gamma \in [0, 1)$  is determined according to the mantissa  $\in [0, 2^{10})$ . Substituting (60) into (59), we have the difference as

$$\Delta B_{dp} = \log_2 \frac{2^{C^*} \cdot C}{C^*} > 0, \quad (62)$$

which is always a positive value. This indicates that the bit depth of the proposed method is smaller than that of the baseline method. Thus, we have theoretically shown that the proposed method achieves bit-depth reduction in the enhancement layer.

## 5.2 Difference between type I and type II for type B format

Next, we show that the format conversion introduced in Sec. 4.4 alleviates the degradation of LDR images in the base layer. The output LDR image  $\mathbf{y}_L$  is tone mapped from the decoded HDR image  $\mathbf{y}_H$ , which is generated from  $\mathbf{y}_R$  in the proposed method. Therefore, we analyze the relation between  $\mathbf{y}_H$  and  $\mathbf{y}_R$  for the type I and type II methods.

As illustrated in Fig. 3, the proposed type I method produces  $\mathbf{y}_H$  as

$$\mathbf{y}_H = \text{Flt}_B (\text{Rev}_B^{-1}(\mathbf{y}_R)). \quad (63)$$

for type B images. For example, the exponent of the type B image data becomes

$$x_{E,0}^* = (x_{R,c} - x_{M,c} - 1) / 256 \quad (64)$$

from the inverse of (37). Substituting this equation into (8), we have

$$x_{H,0} = f_{Ia}(x_{R,c}) \cdot f_{Ib}(x_{M,c}) \cdot 2^{\min X_E^+ - 127}, \quad (65)$$

where

$$\begin{cases} f_{Ia}(x_{R,c}) = \exp(x_{R,c} \cdot 2^{-8} \log_e 2), \\ f_{Ib}(x_{M,c}) = \frac{\delta_B + 2^{-9}}{2^{\delta_B + 2^{-8}}} \in [0.002, 0.499], \\ \delta_B = \frac{x_{M,c}}{256} \in [0, 1). \end{cases}$$

This is the relation between  $\mathbf{x}_H$  and  $\mathbf{x}_R$ , and includes a function  $f_{Ia}$  that is proportional to the exponent of  $\mathbf{x}_R$ . However, note that this is chopped by the function  $f_{Ib}$ . This is confirmed by Fig. 9, which indicates the relation between  $\mathbf{x}_B$  and  $\mathbf{x}_H$  for the type B ‘Tree’ image. Note that  $\mathbf{x}_B$  is a scaled version of  $\mathbf{x}_R$ . The points ‘o’ indicate where the mapping given by the type I method becomes discontinuous.

In contrast, the proposed type II method in Fig. 7 produces  $\mathbf{y}_H$  as

$$\begin{aligned} \mathbf{y}_H &= \text{Flt}_B(\text{Rev}_B^{-1}(\text{Rnd}(\text{Cnv}^{-1}(\mathbf{y}_R^{(A)})))) \\ &\approx \text{Flt}_A(\text{Rev}_A^{-1}(\mathbf{y}_R^{(A)})) \end{aligned} \quad (66)$$

neglecting the effect of Rnd. This means that the image is converted to type A in the base layer. Therefore, taking the inverse of (35), we have

$$x_{H,c} = f_{IIa}(x_{R,c}) \cdot f_{IIb}(x_{M,c}) \cdot 2^{\min X_E - 15}, \quad (67)$$

where

$$\begin{cases} f_{IIa}(x_{R,c}) = \exp(x_{R,c} \cdot 2^{-10} \log_e 2), \\ f_{IIb}(x_{M,c}) = \frac{\delta_A + 1}{2^{\delta_A}} \in [1, 1.06), \\ \delta_A = \frac{x_{M,c}}{1024} \in [0, 1). \end{cases}$$

Similar to the type I method, the function  $f_{IIa}$  is proportional to the exponent of  $\mathbf{x}_R$ . Note that the function  $f_{IIb}$  is close to one. Therefore, unlike the type I method, the type II method gives an HDR image  $\mathbf{x}_H$  that is approximately proportional to the exponent of  $\mathbf{x}_R$ . This is confirmed by the points marked ‘x’ in Fig. 9.

Next, we investigate how the mappings in (65) and (67) magnify the quantization error  $\mathbf{e}_B$ . Denoting the mapping as  $\mathbf{x}_H = f(\mathbf{x}_B)$ , the error is magnified as

$$\mathbf{y}_H - \mathbf{x}_H = f(\mathbf{x}_B + \mathbf{e}_B) - f(\mathbf{x}_B) \quad (68)$$

$$\approx \frac{\partial f(\mathbf{x}_B)}{\partial \mathbf{x}_B} \cdot \mathbf{e}_B. \quad (69)$$

Figure 10 illustrates the absolute value of

$$\frac{\partial f(\mathbf{x}_B)}{\partial \mathbf{x}_B} \approx \frac{\mathbf{y}_H - \mathbf{x}_H}{\mathbf{y}_B - \mathbf{x}_B} = \frac{\Delta \mathbf{x}_H}{\Delta \mathbf{x}_B} \quad (70)$$

for the type I method (marked ‘o’) and the type II method (marked ‘x’). In the figure, a larger value signifies greater amplification of the error. We can see that the type I method has larger values, especially at the jump points of Fig. 10 which come from the discontinuous points of Fig. 9. This implies that the degradation in LDR image quality produced by the type I method is alleviated by the type II method, which uses the format conversion in Sec. 4.4.

## 6 Experiments

We now describe a series of experiments that tested nine HDR images, including five type A images and four type B images. For the lossy coding in the base layer and the lossless coding in the enhancement layer, we used the JPEG 2000 international standard [1] in lossy mode and lossless mode, respectively.

### 6.1 Base layer

We compared the coding performance in the base layer of the baseline method and the proposed method. In this section, the proposed method denotes the type I procedure in Fig. 3 for type A images and the type II procedure in Fig. 7 for type B images. Figure 11 compares the baseline and proposed methods for the type A ‘Cannon’ image. The horizontal axis records the data volume of the base layer in bits per pixel per color component (bppc). The vertical axis indicates the LDR image quality in terms of PSNR, which is defined by

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{Ens}((\mathbf{y}_L - \mathbf{x}_L)^2)} \quad [dB] \quad (71)$$

for

$$\mathbf{x}_L = \text{Rnd}(\text{Clp}'(\text{Tmo}(\mathbf{x}_H))), \quad (72)$$

where  $\text{Ens}(\cdot)$  denotes the ensemble average over all pixels in the image. The results indicate that the proposed method is slightly worse (by 0.46 [dB] at 3.1

[bpp]) than the baseline approach. Figure 12 indicates the rate distortion curves for the type B ‘Belgium’ image. The results are very similar to those for ‘Cannon’. The ‘Tree’ image was investigated in different formats. Figures 13 and 14 indicate the curves for this image in type A and type B formats, respectively. Figure 15 summarizes the PSNR at 1.5 [bppc] in the base layer. This indicates that the proposed method is slightly better than the baseline technique. It can be concluded that the proposed method is comparable to or slightly better than the baseline method. This is considered to be because of the similarity of  $\mathbf{x}_B$  in the baseline method and the proposed method. Both quantities represent the logarithm of the original HDR image  $\mathbf{x}_H$ .

## 6.2 Enhancement layer

Figure 16 compares the output from the proposed and baseline methods for the type A ‘Cannon’ image. The horizontal axis indicates the PSNR of the reconstructed LDR images, and the vertical axis indicates the bit rate of the bit stream in the enhancement layer. Note that, because the decoded HDR images are lossless, the PSNR is infinite. This figure indicates that the proposed method reduces the bit rate by more than 3.4 [bppc] for this image. As indicated in the figure, the bit rate in the enhancement layer decreases as the PSNR increases. However, the bit rate in the base layer increases with PSNR, which means that there is a trade-off in the bit rate in these layers.

Figure 17 shows the results for the type B ‘Belgium’ image. We can observe that the bit rate decreases by 8.03 [bppc] at 35 [dB] LDR image quality. Unlike the case in Fig. 16, the bit rate increases with the PSNR. This is because the correlation among neighboring pixels in  $\mathbf{e}_I$  increases in low PSNR (low bit rate) coding of the LDR image as indicated in Fig. 18. For this input image, the correlation is observed to be 0.14 at a PSNR of 36.9 [dB]. The correlation monotonically increases as PSNR decreases, reaching 0.80 at 18.8 [dB]. Because  $\mathbf{e}_I$  is encoded with a transform that uses this correlation, a higher correlation serves to lower the bit rate. This is why the curve of the baseline method in the figure increases monotonically. The bit depth of the enhancement layer decreases monotonically from 26.7 [bit] at a PSNR of 18.8 [dB] to 24.1 [bit] at 36.9 [dB] as indicated in Fig. 19. Furthermore, the logarithm of the variance of  $\mathbf{e}_I$  is also monotonically decreasing as indicated in Fig. 20.

The ‘Tree’ image was again investigated in different formats. Figures 21 and 22 show the bit rate for the type A and type B image formats, respectively. We can see that better PSNR in the LDR images brings about a lower bit rate in the enhancement layer. This

suggests that a higher data volume in the base layer will lead to a lower volume in the enhancement layer. Figure 23 summarizes the bit rate at 35 [dB] LDR image quality. This figure indicates that the proposed method reduces the data volume of type A images by a minimum of 3.82 [bppc] (for the ‘Cannon’ image), and by a maximum of 8.82 [bppc] (for ‘Still Life’). For type B images, the data volume is reduced by a minimum of 7.8 [bppc] for ‘Desk’. It was confirmed that the proposed method significantly reduces the data volume of the enhancement layer for both type A and type B format images.

## 7 Conclusion

In this paper, we have presented a bit-depth scalable lossless coding for HDR images in floating-point data formats. Unlike most conventional scalable coding methods, the proposed method reconstructs the original HDR image without any loss. Introducing a reversible logarithmic mapping and format conversion technique, it was confirmed that the proposed method reduces the bit depth as well as the bit rate in the enhancement layer. It was also confirmed that the proposed method maintains the LDR image quality and coding performance of the baseline method in the base layer for both the OpenEXR and RGBE formats.

As our investigation has been limited to a difference-based approach, it is necessary to include ratio-based approaches, such as [8].

## Appendix A

### Substituting

$$\begin{cases} \text{Nrm}^{-1}(\mathbf{x}_B) = \mathbf{x}_B \cdot C/255 + C_1 \\ C = C_2 - C_1 \\ C_1 = \log_e(\min X_I^+), C_2 = \log_e(\max X_I) \end{cases}$$

into

$$\mathbf{e}_I = \exp(\text{Nrm}^{-1}(\text{Nrm}(\log_e \mathbf{x}_I) + \mathbf{e}_B + e_1)) - \mathbf{x}_I + e_2,$$

we have

$$\mathbf{e}_I = \exp(x + \epsilon) - \mathbf{x}_I + e_2$$

where

$$\begin{cases} x = \log_e \mathbf{x}_I, \\ \epsilon = (\mathbf{e}_B + e_1) \cdot C/255. \end{cases}$$

When  $x$  takes its maximum value,  $\epsilon \ll x$  holds. For example, the value of  $\epsilon/x$  for all images tested in this

paper is less than  $10^{-4}$ . In this case,

$$\begin{aligned} \mathbf{e}_I &= \exp(x + \epsilon) - \mathbf{x}_I + e_2 \\ &\approx \frac{\partial \exp(x)}{\partial x} \epsilon + \exp(x) - \mathbf{x}_I + e_2 \end{aligned}$$

holds. Therefore, we have

$$\begin{aligned} \mathbf{e}_I &= \exp(x)\epsilon + \exp(x) - \mathbf{x}_I + e_2 \\ &= \exp(\log_e \mathbf{x}_I)(\mathbf{e}_B + e_1) \cdot C/255 \\ &\quad + \exp(\log_e \mathbf{x}_I) - \mathbf{x}_I + e_2 \\ &= \mathbf{x}_I(\mathbf{e}_B + e_1) \cdot C/255 + \mathbf{x}_I - \mathbf{x}_I + e_2 \\ &= \mathbf{x}_I(\mathbf{e}_B + e_1) \cdot C/255 + e_2 \end{aligned}$$

namely,

$$\max E_I = \max X_I(\max E_B + e_1) \cdot C/255 + e_2.$$

According to our experiments,  $\max E_I$  is  $7.23 \times 10^3$  in ‘Cannon’ at minimum. Therefore  $e_2$  is negligible compared to  $\max E_I$  since the maximum of  $e_2$  is 0.5. Similarly,  $\max E_B$  takes value between 3 and approximately  $2^7$  depending on the bit rate of the base layer. Therefore,  $e_1$  is negligible in low bit rate compared to  $\max E_B$  and we have

$$\max E_I = \max X_I \cdot \max E_B \cdot C/255.$$

Note that precision of this estimation slightly decreases in high bit rate coding in which  $\max E_B$  takes small value such as 3.

## Appendix B

Substituting

$$\begin{cases} \text{Nrm}^{\prime-1}(\mathbf{x}_B) = \mathbf{x}_B \cdot C'/255 + C'_1 \\ \text{Nrm}'(\mathbf{x}_R) = (\mathbf{x}_R - C'_1) \cdot 255/C' \\ C' = C'_2 - C'_1 \\ C'_1 = \min X_R, C'_2 = \max X_R \end{cases}$$

into

$$\mathbf{e}_R = \text{Nrm}^{\prime-1}(\text{Nrm}'(\mathbf{x}_R) + \mathbf{e}_B + e'_1) - \mathbf{x}_R + e'_2,$$

we have

$$\begin{aligned} \mathbf{e}_R &= \text{Nrm}^{\prime-1}((\mathbf{x}_R - C'_1) \cdot 255/C' + \mathbf{e}_B + e'_1) \\ &\quad - \mathbf{x}_R + e'_2, \\ &= \mathbf{x}_R - C'_1 + (\mathbf{e}_B + e'_1) \cdot C'/255 + C'_1 \\ &\quad - \mathbf{x}_R + e'_2, \\ &= (\mathbf{e}_B + e'_1) \cdot C'/255 + e'_2. \end{aligned}$$

According to our experiments,  $\max E_R$  is 94 in ‘Cannon’ at minimum. Therefore  $e'_2 \in [-0.5, 0.5]$  is negligible compared to  $\max E_R$ . Similarly to appendix A,  $e_1$  is negligible compared to  $\max E_B$ . As a result, we have

$$\max E_R = \max E_B \cdot \max X_R/255.$$

### Competing interests

The authors declare that they have no competing interests.

### Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 26289117.

### Author details

<sup>1</sup> Department of Electrical, Electronics and Information Engineering, Nagaoka University of Technology, Nagaoka, Niigata, Japan. <sup>2</sup> Department of Electrical Engineering, University of Malaya, Kuala Lumpur, Malaysia. <sup>3</sup> Department of Information and Communication Systems, Faculty of System Design, Tokyo Metropolitan University, Hino, Tokyo, Japan.

### References

- ISO/IEC 15444-1, *Information Technology - JPEG 2000 Image Coding System: Core coding system*
- Skodras, A., Christopoulos, C., Ebrahimi, T.: The JPEG 2000 still image compression standard. *IEEE Signal Process. Mag.* **18**, 36–58 (2001)
- Descampe, A., Devaux, F.-O., Rouvroy, G., Legat, J., Quisquater, J.-J., Macq, B.: A flexible hardware JPEG 2000 decoder for digital cinema. *IEEE Trans. Circuits Syst. Video Technol.* **16**(11), 1397–1410 (2006)
- Kaneko, K., Ohta, N.: 4K applications beyond digital cinema. In: *Proc. Int. Conf. Virtual Syst. Multimedia*, pp. 133–136 (2010)
- Reinhard, E., Ward, G., Pattanaik, S., Debevec, P.: *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, San Francisco, CA, USA (2010)
- Bogart, R., Kainz, F., Hess, D.: OpenEXR image file format. *ACM SIGGRAPH, Sketches & Applications* (2003)
- Ward, G.: *Real Pixels Graphics Gems II*. Academic Press, Boston, UK (1991)
- Ward, G., Simmons, M.: JPEG-HDR: a backwards-compatible, high dynamic range extension to JPEG. In: *Proc. ACM SIGGRAPH 2006 Courses* (2006)
- Mantiuk, R., Efremov, A., Myszkowski, K., Seidel, H.-P.: Backward compatible high dynamic range MPEG video compression. *ACM Trans. Graph.* **25**(3), 713–723 (2006)
- Winken, M., Marpe, D., Schwarz, H., Wiegand, T.: Bit-depth scalable video coding. In: *Proc. IEEE Int. Conf. Image Process.*, vol. 1, pp. 5–8 (2007)
- Park, S., Rao, K.R.: Bit-depth scalable video coding based on H.264/AVC. *IEICE Trans. Fundamentals.* **E91-A**(6), 1541–1544 (2008)
- Khan, I.R.: Two layer scheme for encoding of high dynamic range images. In: *Proc. IEEE Int. Conf. Acoustics Speech Signal Process.*, pp. 1169–1172 (2008)
- Kikuchi, H., Otake, W., Iwahashi, M.: Bit rate reduction of enhancement layer in bit-depth scalable coding. In: *Proc. Int. Symp. Intell. Signal Process. Commun. Syst.*, pp. 264–267 (2009)
- Boschetti, A., Adami, N., Leonardi, R., Okuda, M.: Flexible and effective high dynamic range image coding. In: *Proc. IEEE Int. Conf. Image Process.*, pp. 3145–3148 (2010)
- Jinno, T., Okuda, M., Adami, N.: New local tone mapping and two-layer coding for HDR images. In: *Proc. IEEE Int. Conf. Acoustics Speech Signal Process.*, pp. 765–768 (2012)
- Chiang, J.-C., Kuo, W.-T., Kao, P.-H.: Bit-depth scalable video coding with new inter-layer prediction. *EURASIP J. Adv. Signal Process.* **2011**(1) (2011)
- Zhang, Y., Bull, D.R., Reinhard, E.: Perceptually lossless high dynamic range image compression with JPEG 2000. In: *Proc. Int. Conf. Image Process.*, pp. 1057–1060 (2012)

18. Shih, Y.S., Zhang, W.C., Sheng, H., et.al.: Bio-inspired JPEG XR CODEC design for lossless HDR biomedical image. In: Proc. Int. Comp. Symp., pp. 148–153 (2010)
19. Iwahashi, M., Kiya, H.: Efficient lossless bit depth scalable coding for HDR images. In: Proc. Asia-Pacific Signal Inf. Process. Association Annual Summit Conf., pp. 1–4 (2012)
20. Xu, R., Pattanaik, S.N., Hughes, C.E.: High-dynamic-range still-image encoding in JPEG 2000. IEEE Comput. Graphics Appl. **25**(6), 57–64 (2005)
21. Ping, C.Y., Iwahashi, M., Kiya, H.: Lossless bit depth scalable coding for floating point images. In: Proc. Int. Workshop Adv. Image Technol., pp. 169–174 (2013)
22. Blinn, J.F.: Floating-point tricks. IEEE Comput. Graph. Appl. **17**(4), 80–84 (1997)
23. Iwahashi, M., Kiya, H.: Two layer lossless coding of HDR images. In: Proc. Int. Conf. Acoust. Speech Signal Process., pp. 1340–1344 (2014)
24. Pendu, M.L., Guillemot, C., Thoreau, D.: Adaptive re-quantization for high dynamic range video compression. In: Proc. Int. Conf. Acoust. Speech Signal Process., pp. 7367–7371 (2014)

#### Figures

**Figure 1 HDR image coding in JPEG 2000.** The logarithmic function is applied and normalized to 8-bit depth before lossy encoding.

**Figure 2 The baseline method.** The HDR image is reconstructed without any loss.

**Figure 3 The proposed type I method.** The bit depth of the residual  $e_R$  is reduced by the reversible logarithmic mapping Rev.

**Figure 4 LDR images given by the proposed type I and type II methods.** The LDR image from the type I method is degraded in type B format.

**Figure 5 Images decoded in the base layer.**

**Figure 6 LDR images tone mapped with the Hill function.**

**Figure 7 The proposed type II method.** Cnv converts type B to type A in the base layer.

**Figure 8 The format conversion Cnv in the proposed type II method.**

**Figure 9 Mapping in the systems for the type B image 'Tree'.**

**Figure 10 Magnification of coding errors in the systems for the type B image 'Tree'.**

**Figure 11 Coding performance in the base layer for type A image 'Cannon'.**

**Figure 12 Coding performance in the base layer for type B image 'Belgium'.**

**Figure 13 Coding performance in the base layer for type A image 'Tree'.**

**Figure 14 Coding performance in the base layer for type B image 'Tree'.**

**Figure 15 Image quality of LDR images for various images at 1.5 [bppc] enhancement layer bit stream.**

**Figure 16 Bit rate of the enhancement layer for type A image 'Cannon'.**

**Figure 17** Bit rate of the enhancement layer for type B image 'Belgium'.

**Figure 18** Correlation of the difference for type B image 'Belgium'.

**Figure 19** Bit depth of the difference for type B image 'Belgium'.

**Figure 20** Log of variance of the difference for type B image 'Belgium'.

**Figure 21** Bit rate of the enhancement layer for type A image 'Tree'.

**Figure 22** Bit rate of the enhancement layer for type B image 'Tree'.

**Figure 23** Bit rate of the enhancement layer for various images at 35 [dB] LDR image quality.

**Additional Files**

Additional file 1 — 00Main3.pdf

The PDF file of this manuscript.

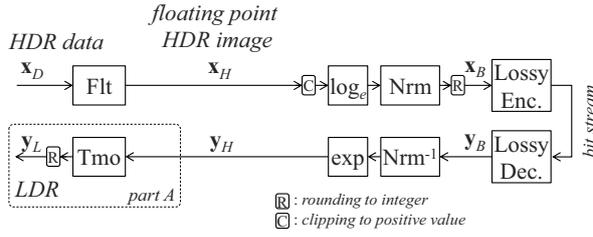


Figure 1



Figure 5

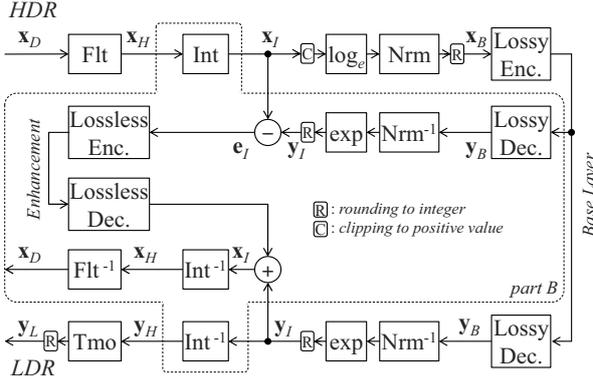


Figure 2



Figure 6

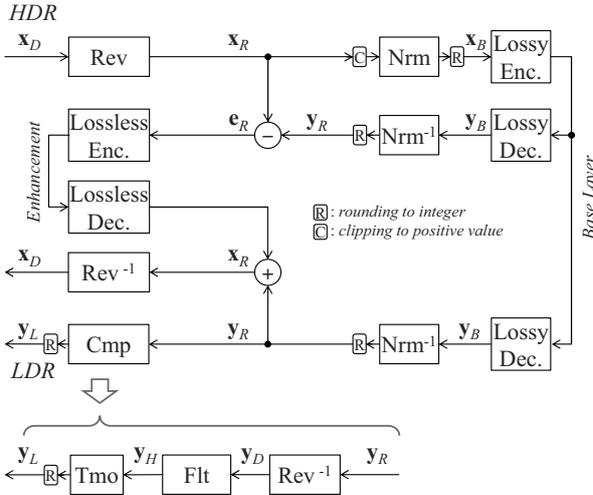


Figure 3

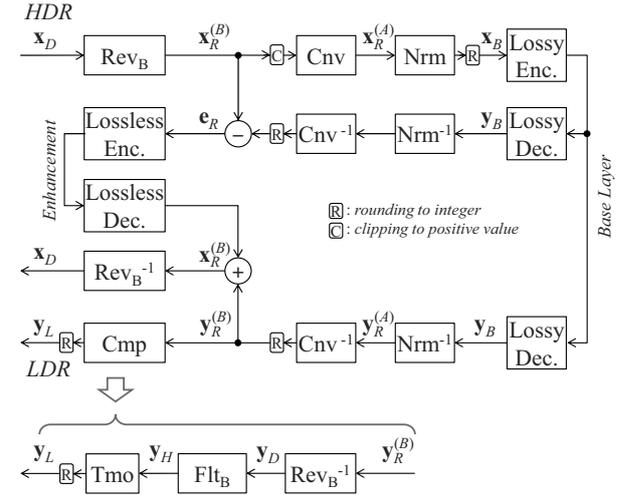


Figure 7

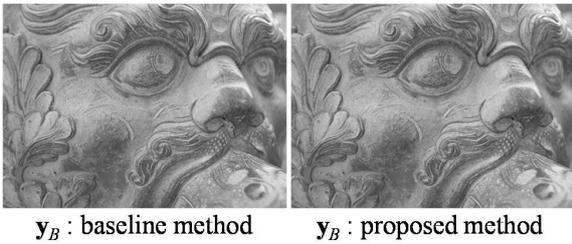


Figure 4

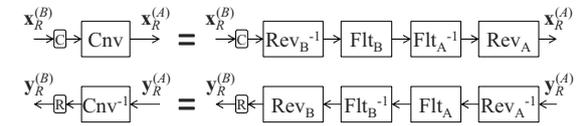


Figure 8

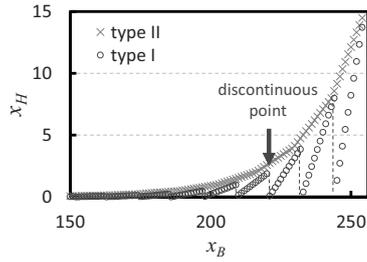


Figure 9

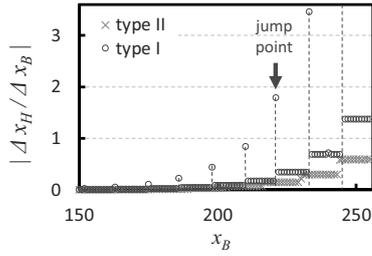


Figure 10

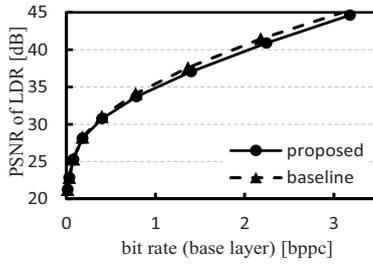


Figure 11

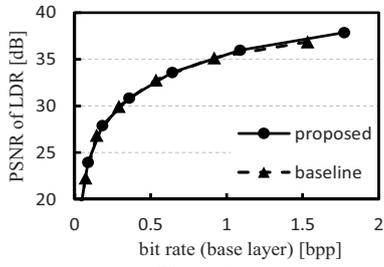


Figure 12

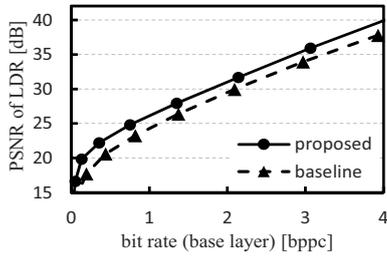


Figure 13

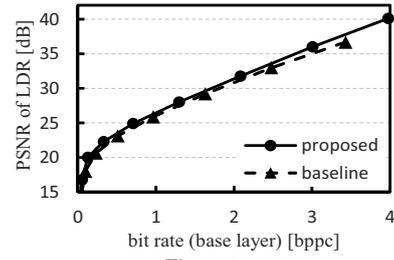


Figure 14

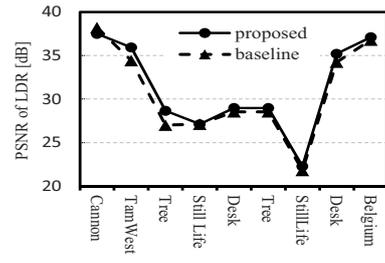


Figure 15

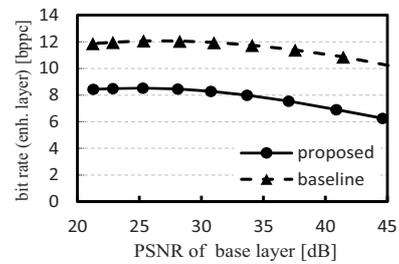


Figure 16

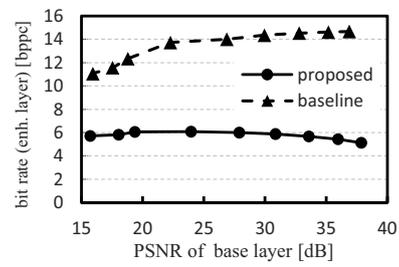


Figure 17

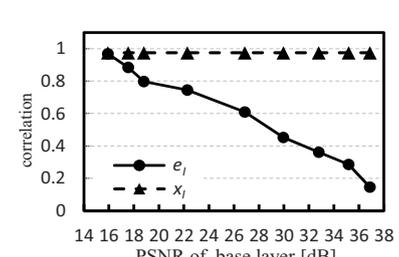


Figure 18

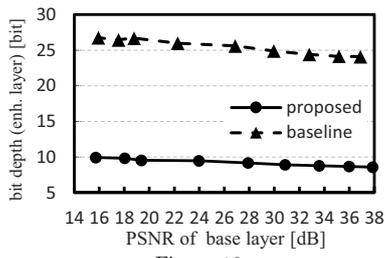


Figure 19

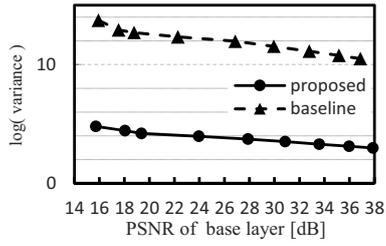


Figure 20

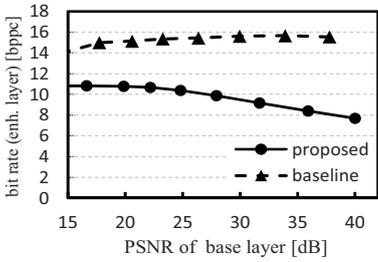


Figure 21

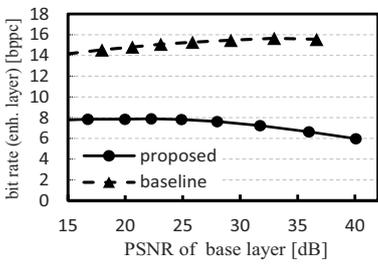


Figure 22

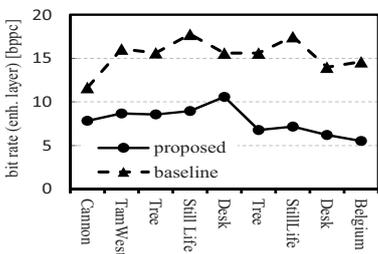


Figure 23