

浮動小数点画像に対するロスレスなビット深度スケーラブル符号化

Lossless Bit Depth Scalable Coding for Floating Point Images

伊藤麗良[†] 岩橋政宏[†] 貴家仁志[‡]

[†]長岡技術科学大学 [‡]首都大学東京

Reira ITO[†]

Masahiro IWAHASHI[†]

Hitoshi KIYA[‡]

[†]Nagaoka University of Technology

[‡]Tokyo Metropolitan University

アブストラクト 浮動小数点表現された入力画像をロスレスに復号できる、ビット深度スケーラブル符号化を提案する。浮動小数点表現された画素値を、整数値に一对一に写像することで、入力画像のロスレス再生が可能となる。しかし、写像後のビット深度が長くなり圧縮に適さない。そこで、写像後のヒストグラムが疎であることに着目し、これをパッキングして密にする。これにより、拡張層の圧縮効率を改善できる。一方、標準的なビット深度を持つ基本層の画像は、入力画像を対数関数によりレンジ圧縮する。この場合、パッキングが対数関数と等価であるため、特に処理を付加することなく、汎用的なデコーダにより高画質に再生できることを示す。

1 はじめに

近年、高ダイナミックレンジ (HDR) 画像が注目を集めており、例えば、RGBE 形式や OpenEXR 形式 [1] では、画素の値が浮動小数点により表現されている。一方、汎用的な画像表示装置や処理システムでは、画素値が整数として表現された低ダイナミックレンジ (LDR) 画像を扱うように設計されている。このため、HDR 用の画像システムを開発する際、LDR 用の画像システムとの互換性を考慮することが重要となる。

このような画像システムとして、ビット深度スケーラブル符号化が提案されている [2]–[6]。HDR 画像のデータ量を圧縮する際、エンコーダは圧縮データを二層に分けて出力し、基本層からは LDR 画像を、拡張層を併せると HDR 画像を、それぞれ復号できる機能を持つ。しかし、これらの従来法では、再生された HDR 画像には若干の誤差が含まれる。本報告では、オリジナルな HDR 画像を一切の誤差なく完全に復元できる、ロスレスなビット深度スケーラブル符号化を提案する。

まず、HDR 画像のロスレス符号化を実現するため、浮動小数点で表現された画素値を、情報のロスが生じないように整数値に一对一に写像する。次に、得られた HDR

整数画像の画素値を対数関数によりレンジ圧縮 [7] することで、LDR 画像を生成する。基本層は、この LDR 画像をロッシェー符号化した結果 (圧縮データ) とする。一方、拡張層は、LDR 画像を逆レンジ圧縮して、HDR 整数画像の近似値を得て、オリジナルの HDR 整数画像との差分をロスレス圧縮した結果とする。以上により、従来のアプローチをロスレスへとシンプルに拡張できる (以下、従来法)。

しかし、HDR 整数画像のレンジ (値域=最大-最小+1) が非常に広くなり、このままではデータ圧縮に適さない。そこで我々は、HDR 整数画像の画素値として、使われていない値が多く存在する (ヒストグラムが疎である) ことに着目し、ヒストグラム・パッキング [8], [9] を適用してレンジ圧縮することで、拡張層の圧縮効率を改善できることを既に示した [10]。

本報告では特に、浮動小数点表現された HDR 画像からヒストグラム・パッキングされた HDR 整数画像への画素値の一对一の写像が、LDR 画像を生成する際に適用する対数関数 [7] をよく近似することを解析的に示す。これにより、提案法による基本層からは、復号時に何ら特別な処理を付加することなく、汎用的なロッシェー・デコーダにより高画質な LDR を再生できることを示す。

2 ロスレスなビット深度スケーラブル符号化

以下に、浮動小数点画像を対象とした、従来法とその問題点についてまとめる。

2.1 浮動小数点表現された HDR 画像

本報告では、オリジナルの HDR 画像の画素値 x_F が、仮数部 x_M と指数部 x_E により、

$$x_F = (1 + x_M \cdot 2^{-D_M}) \cdot 2^{x_E - E_0} = f_I(x_M, x_E) \quad (1)$$

として浮動小数点表現される場合について考察する。ここで、仮数部と指数部の値域は、

$$\begin{cases} x_M \in [Min_M, Max_M] \in [0, 2^{D_M} - 1] \\ x_E \in [Min_E, Max_E] \in [0, 2^{D_E} - 1] \end{cases} \quad (2)$$

として、それぞれのビット深度である D_M と D_E により与えられる。なお、 Max_E と Min_E は、画像内の全ての x_E のうち、最大と最小をそれぞれ意味する。同様に、 Max_M と Min_M は x_M の最大と最小である。

例えば、OpenEXR 形式では D_M は 10[bit]、 D_E は 5[bit]、定数 E_0 は、 $1 \leq x_E \leq 30$ に対して 15 と定義される [1]。また、正負符号を持ち、Red、Green、Blue の 3 成分で構成されるが、本報告では以下、議論を輝度のみ（白黒画像）に限定して進める。

2.2 従来法の処理手順

図 1 には、本報告における従来法を示す。まず、HDR 画像のロスレス符号化を実現するため、浮動小数点で表現された式 (1) の画素値 x_F に定数を乗じることで、

$$\begin{aligned} x_I &= x_F \cdot 2^{D_M + E_0 - Min_E} \\ &= (x_M + 2^{D_M}) \cdot 2^{x_E - Min_E} \end{aligned} \quad (3)$$

により、整数値 x_I に一対一に写像する（図 1(a) の 'Float to Integer'）。ここで、この逆写像である、

$$\begin{cases} x_M = x_I \cdot 2^{-x_E + Min_E} - 2^{D_M} \\ x_E = \text{Floor}[\log_2 x_I] + Min_E - D_M \end{cases} \quad (4)$$

により、整数 x_I からもとの x_F および (x_M, x_E) へとロスレスに復元できる（図 1(b) の 'Integer to Float'）。これにより、HDR 画像のロスレス符号化が可能となる。

基本層（base layer）の圧縮データを得るため、得られた HDR 整数画像の画素値 x_I を、対数関数、

$$x_T = \log_e x_I \quad (5)$$

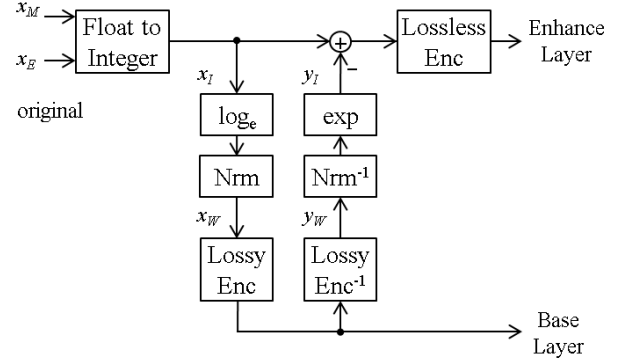
によりレンジ圧縮する（図 1(a) 中の ' \log_e '）。但し、汎用的なエンコーダで圧縮するため、

$$x_W = \text{Round} \left[\frac{x_T - Min_T}{Max_T - Min_T} \cdot 255 \right] \quad (6)$$

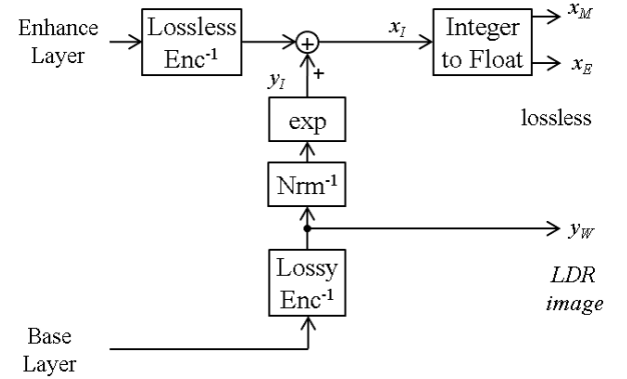
により LDR のビット深度（この場合は 8 [bit]）に正規化（図 1(a) 中の ' Nrm '）することで、LDR 画像 x_W を生成する [7]。なお、 $\text{Round}[x]$ は、 x に近い整数への丸めを表す。式 (5) と式 (6) より、

$$\begin{aligned} x_W &= \text{Round} \left[\frac{\log_e x_I - \log_e Min_I}{\log_e Max_I - \log_e Min_I} \cdot 255 \right] \\ &= f_R(x_I) \\ x_I &\in [Min_I, Max_I], \quad Min_I > 0 \end{aligned} \quad (7)$$

となり、得られた LDR 画像 x_W は汎用的なロシー符号化（図 1(a) 中の 'Lossy Enc'）により圧縮されて、基本層のビットストリームとなる。本報告では、JPEG 2000 のロシー符号化を用いた [11]。



(a) エンコーダ



(b) デコーダ

図 1: 従来法

2.3 従来法の問題点

一方、拡張層は、符号化側で復号された（符号化誤差を含む）LDR 画像 y_W を逆レンジ圧縮して HDR 整数画像の近似値 y_I を得て、オリジナルの HDR 整数画像 x_I との差分である、

$$r_I = x_I - y_I \quad (8)$$

をロスレス圧縮（図 1(a) 中の 'Lossless Enc'）した結果を、ビットストリームとする。ここで、 y_I は、

$$y_I = \text{Round} \left[\exp \left(\frac{Max_T - Min_T}{255} y_W + Min_T \right) \right] \quad (9)$$

となる。

以上の処理では、従来のアプローチがロスレスへとシンプルに拡張されている。しかし、HDR 整数画像のレンジ（値域 = 最大 - 最小 + 1）が非常に広くなり、このままではデータ圧縮に適さない。

3 提案法とその解析

以下、HDR 画像がロスレス再生されるという条件下で、拡張層のビット深度（= \log_2 レンジ）を、効果的に低減する方法について説明する。また、拡張層のビット深度と基本層の画質について解析する。

3.1 提案法の処理手順

図 2 には、本報告における提案法を示す。提案法では、ロスレスな写像（図中の 'Rev.Log'）、

$$\begin{aligned} x_L &= x_M + 2^{D_M} \cdot (x_E - Min_E) \\ &= f_L(x_M, x_E) \end{aligned} \quad (10)$$

により、整数画像 x_L を生成する。この処理自体は、[12] においても紹介されている。表 1 には、 $D_M = D_E = 2$ 、 $Min_E = 0$ の場合について、この写像（以下、可逆対数写像）の例を示す。

従来法における x_I は、二進数では $(00100)_2$ から $(11100)_2$ まで、十進数では 4 から 56 まで、すなわち、レンジは $56 - 4 + 1 = 53$ となっている。一方、提案法における x_L は、レンジが 16 に削減されている。どちらも 16 個の非ゼロのビンを持っているが、レンジが異なっている。これは、HDR 整数画像 x_I の画素値において、使われていない値（ヒストグラムの零ビン）が多く存在することを意味する。従って、 x_I にヒストグラム・パッキング [8], [9] を適用することで、レンジが圧縮され、拡張層

表 1: 可逆対数写像 'Rev.Log' の例
($D_M = D_E = 2, Min_E = 0$)

floating point		fixed point		
x_M	$\times 2^{(x_E - E_0)}$	x_F	x_I	x_L
1.11	$\times 2^{(11-10)}$	=11.100	111000	11 11
1.10	$\times 2^{(11-10)}$	=11.000	111000	11 10
1.01	$\times 2^{(11-10)}$	=10.100	111000	11 01
1.00	$\times 2^{(11-10)}$	=10.000	111000	11 00
1.11	$\times 2^{(10-10)}$	=1.1100	011100	10 11
1.10	$\times 2^{(10-10)}$	=1.1000	011100	10 10
1.01	$\times 2^{(10-10)}$	=1.0100	011100	10 01
1.00	$\times 2^{(10-10)}$	=1.0000	011100	10 00
1.11	$\times 2^{(01-10)}$	=0.1110	011100	01 11
1.10	$\times 2^{(01-10)}$	=0.1100	011100	01 10
1.01	$\times 2^{(01-10)}$	=0.1010	011100	01 01
1.00	$\times 2^{(01-10)}$	=0.1000	011100	01 00
1.11	$\times 2^{(00-10)}$	=0.0111	011100	00 11
1.10	$\times 2^{(00-10)}$	=0.0110	011100	00 10
1.01	$\times 2^{(00-10)}$	=0.0101	011100	00 01
1.00	$\times 2^{(00-10)}$	=0.0100	011100	00 00

の圧縮効率が改善される [10]。

なお、逆方向の可逆対数写像（図 2(b) 中の 'Rev.Log⁻¹'）は、

$$\begin{cases} x_M = x_L - 2^{D_M} \cdot (x_E - Min_E) \\ x_E = \text{Floor}[x_L \cdot 2^{-D_M}] + Min_E \end{cases} \quad (11)$$

と定義され、整数 x_L から元の (x_M, x_E) をロスレスに復号できる。また、図 2(a) 中の 'Nrm' は、

$$\begin{aligned} x_N &= \text{Round} \left[\frac{x_L - Min_L}{Max_L - Min_L} \cdot 255 \right] \\ &= f_N(x_L) \end{aligned} \quad (12)$$

であり、8 ビット深度への正規化を意味する。

3.2 拡張層のビット深度

以下、式 (10) の可逆対数写像 (Rev.Log) により、拡張層のビット深度が低減されることを示す。任意の整数値 x_X のビット深度を、

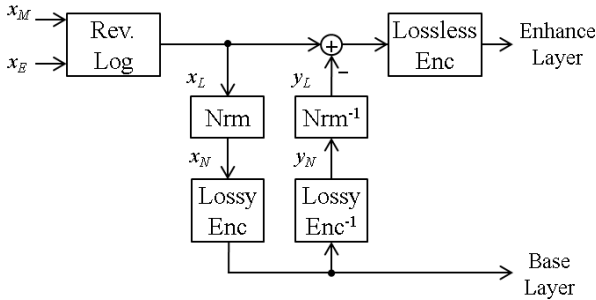
$$B_{dp}(x_X) = \log_2(Max_X - Min_X + 1) \quad (13)$$

と定義する。このとき、従来法における x_I のビット深度は、 Min_M と Max_M をそれぞれ 0 と $R_M - 1$ とすると、

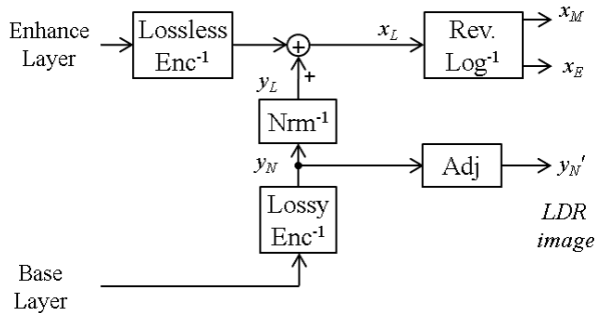
$$\begin{aligned} B_{dp}(x_I) &= \log_2 \left((-1 + 2^{1+D_M}) 2^{Max_E - Min_E} - 2^{D_M} + 1 \right) \\ &< Max_E - Min_E + 1 + D_M \\ &< 2^{D_E} + D_M \end{aligned} \quad (14)$$

となる。一方、提案法における x_L のビット深度は、

$$\begin{aligned} B_{dp}(x_L) &= \log_2 \left(2^{D_M} - 1 + 2^{D_M} Max_E - 2^{D_M} Min_E + 1 \right) \\ &< Max_E - Min_E + 1 + D_M \\ &< \log_2(Max_E - Min_E + 1) + D_M \\ &< D_E + D_M \end{aligned} \quad (15)$$



(a) エンコーダ



(b) デコーダ

図 2: 提案法

となる。式 (14) と式 (15) を比較すると、 x_L のビット深度が x_I のビット深度よりも縮小されている。以上より、式 (10) の 'Rev.Log' により、拡張層のビット深度を低減できることが示された。

3.3 LDR 画像の画質

次に、浮動小数点表現された HDR 画像からヒストグラム・パッキングされた HDR 整数画像への可逆対数写像 (Rev.Log) が、LDR 画像を生成する際に適用した対数関数 [7] とよく近似することを解析的に示す。これは、図 1(a) の x_W と図 2(a) の x_N がほぼ同じになることを意味する。

従来法と提案法のそれぞれについて、レンジが $[0,1]$ の範囲に正規化された LDR 画像を、

$$y_{ex} = \frac{x_W}{255}, \quad y_{pr} = \frac{x_N}{255} \quad (16)$$

とする。以下、両者の誤差が僅かであることを解析的に示す。式 (8) より、従来法の y_{ex} は、

$$\begin{aligned} y_{ex} &= \frac{\log_e x_I - \log_e \text{Min}_I}{\log_e \text{Max}_I - \log_e \text{Min}_I} \\ &= \frac{1}{\gamma_0} \log_e \frac{\text{Min}_I + (\text{Max}_I - \text{Min}_I)x}{\text{Min}_I} \\ &= \frac{\log_e(1 + \alpha_0 x)}{\gamma_0} \end{aligned} \quad (17)$$

となる。ここで、 α_0 、 γ_0 は、

$$\alpha_0 = \frac{\text{Max}_I}{\text{Min}_I} - 1, \quad \gamma_0 = \log_e \frac{\text{Max}_I}{\text{Min}_I} \quad (18)$$

である。同様に、式 (3) と式 (10) より、

$$\begin{aligned} x_L &= x_M + 2^{D_M} \cdot \log_2 \frac{x_I}{x_M + 2^{D_M}} \\ &= \frac{2^{D_M}}{\log_e 2} \log_e(x_I \delta_{x_I}) - D_M 2^{D_M} \end{aligned} \quad (19)$$

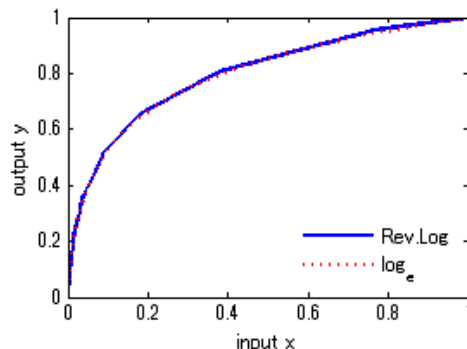
となる。ここで、 δ_x 、 ε_x は、

$$\begin{cases} \delta_x = \frac{2^{\varepsilon_x}}{1 + \varepsilon_x} \in [0.9421, 1) \\ \varepsilon_x = x_M \cdot 2^{-D_M} \in [0, 1) \end{cases} \quad (20)$$

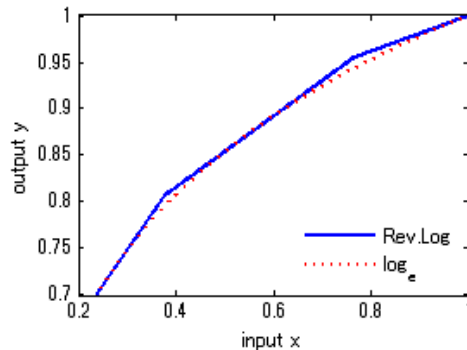
である。

よって、提案法の y_{pr} は、

$$\begin{aligned} y_{pr} &= \frac{\log_e x_I \delta_{x_I} - \log_e \text{Min}_I \delta_{\text{Min}_I}}{\log_e \text{Max}_I \delta_{\text{Max}_I} - \log_e \text{Min}_I \delta_{\text{Min}_I}} \\ &= \frac{\log_e \frac{x_I}{\text{Min}_I} + \log_e \frac{\delta_{x_I}}{\delta_{\text{Min}_I}}}{\log_e \frac{\text{Max}_I}{\text{Min}_I} + \log_e \frac{\delta_{\text{Max}_I}}{\delta_{\text{Min}_I}}} \\ &= \frac{\log_e(1 + \alpha_0 x) + \Delta y}{\gamma_0 + \Delta \gamma} \end{aligned} \quad (21)$$



(a) 全体



(b) 拡大

図 3: 'Rev.Log' と ' \log_e '

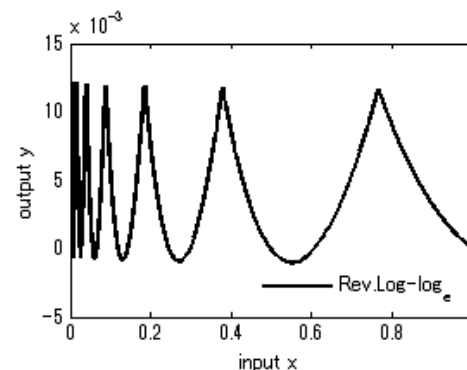


図 4: 'Rev.Log' と ' \log_e ' の差

となる。以上より、式 (17) の ' \log_e ' による LDR 画像と、式 (21) の 'Rev.Log' による LDR 画像との差分は、

$$\begin{cases} \Delta y = \log_e \frac{\delta_{x_I}}{\delta_{\text{Min}_I}} \in [-0.0596, 0.0596] \\ \Delta \gamma = \log_e \frac{\delta_{\text{Max}_I}}{\delta_{\text{Min}_I}} \in [-0.0596, 0.0596] \end{cases} \quad (22)$$

となり、非常に小さいことが解析的に示された。

図 3 には、 $D_M = D_E = 4$ および $\text{Min}_E = 0$ を例として、式 (17) の ' \log_e ' と式 (21) の 'Rev.Log' をそれぞれ示す。図 4 には、両者の差を示す。同図からは、'Rev.Log' は ' \log_e ' をよく近似することがわかる。これにより、提案法による基本階層からは、復号時に何ら特別な処理を付加することなく、汎用的なロッシー・デコーダにより、高画質な LDR 画像を再生できることになる。

3.4 LDR 画像の微調整

ここで、デコーダ側に若干の追加処理 (図 2(b) 中の 'Adj') を付加することで、式 (22) の差が埋まり、更に高画質な LDR 画像が得られる。この補正は、

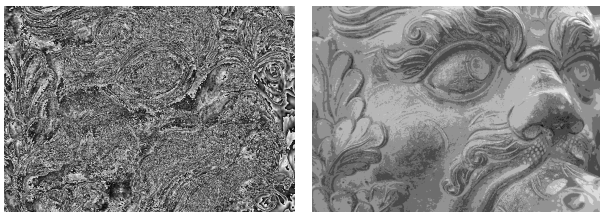
$$f_{Adj}(y_N) = f_R \circ f_I \circ f_L^{-1} \circ f_N^{-1}(y_N) \quad (23)$$

であり、これにより得られる LDR 画像 $y_{N'}$ は、従来法の LDR 画像 y_W と一致する。

なお、今回はレンジ圧縮に対数関数を用いたが、それ以外の関数については文献 [13]、[14] に記載した。

4 実験結果

図 5 には、OpenEXR 形式 (544 × 768 ピクセル) の画像「Cannon」の赤成分の仮数部 x_M と指数部 x_E を示す。元データのビット深度はそれぞれ 10[bit]、5[bit] であるが、8[bit] に正規化して表示してある。以下、オリジナルの HDR 画像の輝度値としてこの画像を、基本層のロッシェン符号化には実数 9/7 DWT (5 stage, 最適ビット配分) を、拡張層のロスレス符号化には整数 5/3 DWT (5 stage) を、それぞれ用いた。



(a) 仮数部 x_M (b) 指数部 x_E

図 5: 実験で評価した浮動小数点画像



(a) 'log_e' (b) 'Rev.Log'

図 6: 写像の結果

4.1 基本層

まず、LDR 画像の画質について調べる。図 6(a) には、図 5 の画像に、対数関数によるレンジ圧縮を適用した結果である x_W を示す。図 6(b) には、可逆対数写像 'Rev.Log' で写像された画像 x_N を示す。両者の違いは極僅かであり、'Rev.Log' が 'log_e' をよく近似することが確認できる。このときの両者の差を peak signal to noise ratio (PSNR)、

$$PSNR = 10 \log_{10} \frac{255^2}{E[(x_N - x_W)^2]} [dB] \quad (24)$$

により測定した。ここで、 $E[\]$ は、画像内の全画素の集合平均である。結果は 48 [dB] であった。この値は、デコーダでの微調整 (Adj) を用いずに基本層から復号される LDR 画像の、PSNR の上限を意味する。

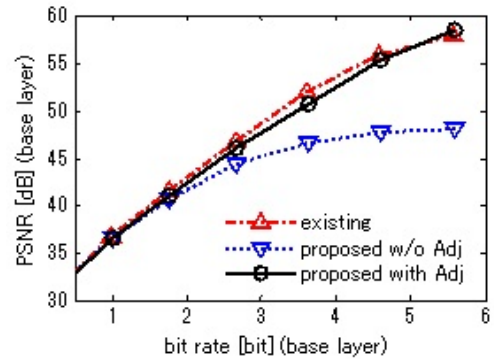


図 7: 基本層のレート-歪み曲線

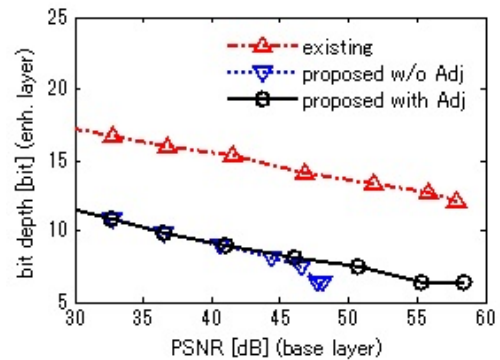


図 8: 拡張層のビット深度

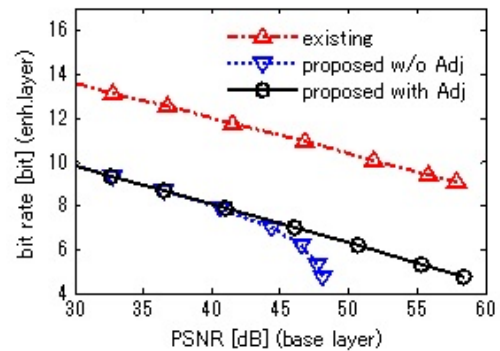


図 9: 拡張層のビットレート

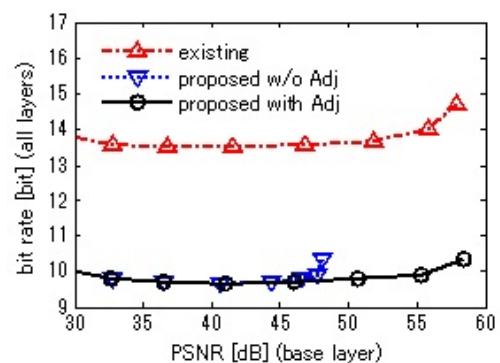


図 10: 全階層のビットレート

次に、基本層の符号化性能について調べる。図7には、基本層におけるレート歪み曲線を示す。レートは、基本層のビットストリームのデータ量を、1画素あたりに換算したときの平均符号長 [bit] である。歪みは、式 (24) によって測定された PSNR [dB] である。提案法と従来法との差は、基本層のビットレートが、例えば 3.6 [bit] のとき、 $51.86 - 46.56 = 5.30$ [dB] となった。また、**3.3** に示した通り、提案法では、補正を加えなくても (図中の 'w/o Adj'), 最高で 48 [dB] 程度の高い画質が得られることが、実験的にも確認された。

提案法に補正を加えた場合 (図中の 'with Adj'), 従来法との差は、例えばビットレートが 3.6 [bit] のとき、 $51.86 - 50.60 = 1.26$ [dB] であった。補正することで、上述した 48 [dB] の上限が取り払われていることが確認できる。

4.2 拡張層

図8には、拡張層のビット深度を示す。基本層の PSNR が、例えば 41 [dB] のとき、 $15.27 - 9.03 = 6.24$ [bit] も削減されている。図9には、拡張層のビットレートを示す。上記の場合、 $11.78 - 7.85 = 3.93$ [bit] 削減されている。図10には、基本層と拡張層の合計のビットレートを示す。このとき、 $13.54 - 9.61 = 3.93$ [bit] 削減されている。

以上より、**3.2** に示した理論的考察が実験的にも裏付けられ、拡張層の圧縮効率を、提案法により効果的に改善できることが結論づけられた。

5 結論

拡張層の圧縮データから、浮動小数点表現された HDR 画像をロスレスに復号でき、基本層からは LDR 画像が復号できる、ビット深度スケラブル符号化を提案した。HDR 画像をロスレスに再生するため、画素値を整数値へ一対一に写像している。結果、ヒストグラムが疎であるため、これをパッキングすることで、拡張層の圧縮効率を改善した。LDR 画像は、HDR 画像を対数関数によりレンジ圧縮している。パッキングが対数関数と等価であるため、汎用的なデコーダにより、高画質な LDR を再生できることを、理論的にも実験的にも示した。今後は、カラー画像へ適用する。

参考文献

[1] E.Reinhard, G.Ward, S.Pattanaik, P.Debevec, W.Heidrich, K.Myszkowski, High Dynamic Range Imaging - Acquisition, Display and Image based Lighting, Morgan Kaufmann, 2010

[2] G. Ward, M. Simmons, JPEG-HDR: A Backwards-Compatible, High Dynamic Range Extension to

JPEG, SIGGRAPH '05 Courses, ACM Article No. 2, 2005

[3] H.Kikuchi, W.Otake, M.Iwahashi, "Bit Rate Reduction of Enhancement Layer in Bit Depth Scalable Coding", ISPACS, pp.264-267, Dec. 2009

[4] S.Yea, W.A.Pearlman, "A Wavelet-Based Two-Stage Near-Lossless Coder" *IEEE transactions on Image processing*, Vol. 15, Issure. 11, pp.3488-3500, 2006

[5] Ishtiaq Rasool Khan, "Two Layer Scheme for Encoding of High Dynamic Range Images", IEEE ICASSP, pp.1169-1172, 2008

[6] A.Boschetti, N.Adami, R.Leonardi, M.Okuda, "Flexible and effective High Dynamic Range image coding" IEEE ICIP, pp.3145- 3148, 2010

[7] R. Xu, S. N. Pattanaik, C. E. Hughes, "High-Dynamic-Range Still Image Encoding in JPEG2000" *IEEE Computer Graphics and Applications*, Vol. 25, No. 6, pp.57-64, 2005

[8] J. Pinho, "An Online Preprocessing Technique for Improving the Lossless Compression of Images with Sparse Histograms", Signal Processing Letters, IEEE, vol. 9, no.1, pp. 5-7, 2002

[9] M. Iwahashi, H. Kobayashi, H. Kiya, "Lossy Compression of Sparse Histogram Image", IEEE ICASSP, pp.1361-1364, 2012

[10] 岩橋政宏, 貴家仁志, "ハイダイナミックレンジ画像のための二階層ロスレス符号化", 映像情報メディア学会, 技術報告, vol.36, no.21, pp.17-20, June, 2012

[11] ISO/IEC 15444-1:2004, "Information technology - JPEG 2000 image coding system: Core coding system", pp.1-194, 2004

[12] James F. Blinn, "Floating - Point Tricks", IEEE Computer Graphics and Applications, pp.81-84, August 1997

[13] M.Iwahashi, H.Kiya, "Efficient Lossless Bit Depth Scalable Coding for HDR Images", APSIPA Annual Summit and Conference, Dec. 2012 (採録決定).

[14] 室伏辰耶, 岩橋政宏, 貴家仁志, "HDR 画像におけるトーンマッピング関数の多段表現", 第 27 回 信号処理シンポジウム, Nov. 2012.