

フォーマット変換を用いた HDR 画像に対するロスレスビット深度スケーラブル符号化 Lossless Bit Depth Scalable Coding for HDR Images using Format Conversion

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 基本層の圧縮データから、高ダイナミックレンジ (HDR) 入力画像からトーンマッピングされた低ダイナミックレンジ (LDR) 画像を復号でき、拡張層と合わせると浮動小数点表現された HDR 入力画像をロスレスに復号できる、二階層ビット深度スケーラブル符号化を提案する。現在の提案法では、入力画像が OpenEXR フォーマットの場合に限定されていた。今回、RGBE フォーマットの入力画像に提案法を適用すると、LDR 画像の画質が劣化してしまうことがわかった。我々はこれを抑制するため、基本層において一時的に RGBE フォーマットから OpenEXR フォーマットへのフォーマット変換を行う方法を導入する。結果、RGBE フォーマットの入力画像の場合でも、画質の劣化を低減することができた。

1 はじめに

近年、高ダイナミックレンジ (HDR) 画像 [1] が注目を集めており、例えば、RGBE フォーマット [2] や OpenEXR フォーマット [3] では、画素値が浮動小数点表現されている。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 [4]–[7]。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基本層と拡張層の二層に分けて出力する。基本層からはオリジナルの HDR からトーンマッピングされた LDR 画像を一般的な非可逆デコーダで復号し、拡張層を併せると HDR 画像を復号できる。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、浮動小数点でから整数への一対一対応の写像を導入することにより、従来のアプローチをロスレスへとサ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 画像の画素値に可逆対数写像を適用し、圧縮効率を改善できることを確認した (以下、提案法タイプ 1)。

本報告では、これらの報告に改善と分析を追加する。提案法タイプ 1 では、OpenEXR フォーマットの入力画像に対しては問題ないが、RGBE フォーマットの入力画像に対しては、復号された LDR 画像の画質が低下することがわかった。この問題を回避するため、我々は一時的にシステム内部で、画像を RGBE フォーマットから OpenEXR フォーマットに変換する (以下、提案法タイプ 2)。結果、提案法は、OpenEXR フォーマットと RGBE フォーマットの両方で、従来法の基本層の符号化性能を維持していることを確認した。

2 ロスレスなビット深度スケーラブル符号化

以下に、浮動小数点データフォーマットの HDR 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 浮動小数点データフォーマット A

浮動小数点画素値の HDR 画像には、OpenEXR と RGBE という 2 つの代表的なデータフォーマットがある。我々の以前の報告では、前者のフォーマットに焦点を当てている [8], [9]。

OpenEXR フォーマットでは、オリジナルの HDR 画像の画素値 $x_{H,c}$ は、色成分 $C \in \{R, G, B\}$ とすると、

$$x_{H,c} = \begin{cases} (1 + x_{M,c} \cdot 2^{-10}) \cdot 2^{-15+x_{E,c}}, & x_{E,c} \in [1, 30] \\ (1 + x_{M,c} \cdot 2^{-10}) \cdot 2^{-14} & , x_{E,c} = 0 \end{cases} \quad (1)$$

として、仮数部 $x_{M,c}$ と指数部 $x_{E,c}$ により与えられる。こ

ここで、仮数部と指数部の値は、

$$x_{M,c} \in [0, 2^{10}), \quad x_{E,c} \in [0, 2^5) \quad (2)$$

の範囲の整数として与えられる。仮数部と指数部のビット深度はそれぞれ 10[bit] と 5[bit] である。また、このフォーマットは 1[bit] の符号部 $x_{S,c}$ も持つ。したがって、HDR 画像の画素値が各色成分につき $10+5+1=16$ [bit] で表される (合計 48[bit]) [3]。

本報告では、各色成分の指数部および仮数部を

$$\begin{cases} \mathbf{x}_E = [x_{E,R} & x_{E,G} & x_{E,B}]^T \\ \mathbf{x}_M = [x_{M,R} & x_{M,G} & x_{M,B}]^T \\ \mathbf{x}_S = [x_{S,R} & x_{S,G} & x_{S,B}]^T \end{cases} \quad (3)$$

とし、HDR 画像データを

$$\mathbf{x}_D = [\mathbf{x}_E \ \mathbf{x}_M \ \mathbf{x}_S] \quad (4)$$

と定義する。これらのベクトルを使用して、式 (1) を

$$\mathbf{x}_H = \text{Flt}_A(\mathbf{x}_D) \quad (5)$$

$$\mathbf{x}_H = [x_{H,R} \ x_{H,G} \ x_{H,B}]^T \quad (6)$$

と表す。以後、OpenEXR フォーマットをフォーマット A とする。

2.2 浮動小数点データフォーマット B

RGBA フォーマットでは、オリジナルの HDR 画像の画素値 $x_{H,c}$ は、色成分 $C \in \{R, G, B\}$ とすると、

$$x_{H,c} = \frac{x_{M,c} + 0.5}{256} \cdot 2^{x_{E,0} - 128} \quad (7)$$

として、仮数部 $x_{M,c}$ と指数部 $x_{E,0}$ により与えられる。指数部、仮数部は

$$x_{M,c} \in [0, 2^8), \quad x_{E,0} \in [0, 2^8) \quad (8)$$

のように、どちらも 8[bit] である。指数部 $x_{E,0}$ は、一般的に三色に同じものが使用されており、このフォーマットでは画素値が合計 32[bit] で表される [2]。本報告では、以前の報告 [9] の分析に、このフォーマットを追加する。式 (3) と式 (4) のベクトルを使用して、式 (7) を

$$\mathbf{x}_H = \text{Flt}_B(\mathbf{x}_D) \quad (9)$$

と表す。以後、RGBA フォーマットをフォーマット B とする。

2.3 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、ロスレスエンコーダへの入力

$$\mathbf{e}_I = \mathbf{y}_I - \mathbf{x}_I \quad (10)$$

は整数でなければならない。したがって、浮動小数点値 \mathbf{x}_H から整数 \mathbf{x}_I への変換 (図 1 中の 'Int') が導入される。これは、入力 HDR 画像がフォーマット A のとき、

$$\mathbf{x}_I = \text{Int}_A(\mathbf{x}_H) \quad (11)$$

として定義される。ただし、このとき、 $C \in \{R, G, B\}$ とすると、

$$\mathbf{x}_I = [x_{I,R} \ x_{I,G} \ x_{I,B}]^T \quad (12)$$

$$x_{I,c} = \begin{cases} x_{H,c} \cdot 2^{25-C_{min}^*} - 2^{10} & , C_{min}^* > 0 \\ x_{H,c} \cdot 2^{24} & , C_{min}^* = 0 \end{cases} \quad (13)$$

である。 \mathbf{x}_E の最小値 C_{min}^* が格納され、ビットストリームに埋め込まれている。同様に、入力 HDR 画像がフォーマット B のとき、

$$\mathbf{x}_I = \text{Int}_B(\mathbf{x}_H) \quad (14)$$

として定義される。ただし、このとき、 $C \in \{R, G, B\}$ とすると、

$$x_{I,c} = (x_{H,c} - 2^{-1}) \cdot 2^{128+8-C_{min}^*} \quad (15)$$

である。

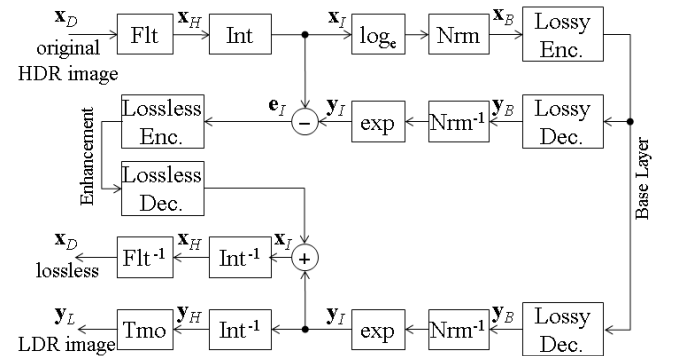


図 1: 従来法

図 1 において、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 \mathbf{x}_I を、対数関数 \log_e によりレンジ圧縮する ('log_e')。但し、汎用的なエンコーダで圧縮するため、画素値の範囲を $[0, 255]$ に正規化し ('NRM')、丸め処理をして LDR 画像 \mathbf{x}_B を生成する。得られた LDR 画像 \mathbf{x}_B は、汎用的なロッシー符号化 ('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。復号処理においては、対数関数の逆関数（'EXP'）により、まず HDR 整数画像の近似値 \mathbf{y}_I が復号される。そこから、浮動小数点値へ変換し（'Int⁻¹'）、HDR 画像 \mathbf{y}_H が生成される。2.5 で定義するトーンマッピング処理（'TMO'）により、LDR 画像 \mathbf{y}_L が生成される。しかし、復号された HDR 画像 \mathbf{y}_H と LDR 画像 \mathbf{y}_L の両方に、ロッシーエンコーダによる符号化誤差が含まれている。

2.4 従来法の問題点

一方、拡張層は、復号された HDR 整数画像の近似値 \mathbf{y}_I を得て、オリジナルの HDR 整数画像 \mathbf{x}_I との差分である、式 (10) の整数 e_I をロスレス圧縮（'Lossless Enc'）した結果を、ビットストリームとする。

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、 e_I は、非常に大きなビット深度、並びに拡張層のデータ量を必要とし、また、残差であるため画素間の相関が弱い傾向があるので、そのデータ量を圧縮することが非常に困難である。

2.5 トーンマッピング処理

本報告では、HDR カラー画像に対して、Hill 関数に基づいたトーンマッピング処理を使用する。この処理は、

$$\mathbf{y}_L = Tmo(\mathbf{y}_H) \quad (16)$$

として定義される。ただし、このとき、色成分 $C \in \{R, G, B\}$ とすると、

$$\begin{cases} \mathbf{y}_H = [y_{H,R} & y_{H,G} & y_{H,B}]^T \\ \mathbf{y}_L = [y_{L,R} & y_{L,G} & y_{L,B}]^T \end{cases} \quad (17)$$

$$\begin{cases} y_{L,c} = 255 \cdot y_{H,c} \cdot (y_{L,Y}/y_{H,Y}) \\ y_{L,Y} = Hill(y_{H,Y}/\bar{y}_{H,Y}) \\ y_{H,Y} = 0.27y_{H,R} + 0.67y_{H,G} + 0.06y_{H,B} \end{cases} \quad (18)$$

である。Hill 関数は

$$Hil(x) = \frac{x^a}{x^a + b^a}, \quad \bar{x} = \exp\left(\frac{E}{x > 0} [\log_e(x)]\right) \quad (19)$$

のように定義される。ここで、 $E[\]$ の演算は、入力画像の全画素の算術的平均を意味する。

3 提案法

以下、拡張層のデータ量を低減するために導入される可逆対数写像について説明する。本報告では、'RGBE' 形式の画像のための特別な処理を、我々の以前の報告 [9] に追加する。

3.1 可逆対数写像

図 2 には、本報告における提案法タイプ 1 を示す。まず、整数 \mathbf{x}_R を生成するため、図 1 の従来法の 'FLT' と 'Int' の代わりに可逆対数写像 'rev' を適用する。この手順自体は [10] に報告されている。これは、入力 HDR 画像がフォーマット A のとき、

$$\mathbf{x}_R = Rev_A(\mathbf{x}_D) \quad (20)$$

として定義される。ただし、このとき、 $C \in \{R, G, B\}$ とすると、

$$\mathbf{x}_R = [x_{R,R} \ x_{R,G} \ x_{R,B}]^T \quad (21)$$

$$x_{R,c} = (x_{E,c} - C_{min}^*) \cdot 2^{10} + x_{M,c} \quad (22)$$

である。同様に、入力 HDR 画像がフォーマット B のとき、

$$\mathbf{x}_R = Rev_B(\mathbf{x}_D) \quad (23)$$

として定義される。ただし、このとき、 $C \in \{R, G, B\}$ とすると、

$$x_{R,c} = (x_{E,0} - C_{min}^*) \cdot 2^8 + x_{M,c} \quad (24)$$

である。

可逆対数写像は、整数を他の整数に写像し、その写像後の整数のダイナミックレンジを減少させる。また、その逆写像により、損失なしでオリジナルの整数を復号できる点で、今回の提案法に適している。我々はすでに、

$$\mathbf{e}_R = \mathbf{y}_R - \mathbf{x}_R \quad (25)$$

で表される拡張層の予測誤差のビット深度がより小さくなることを [9] で確認した。

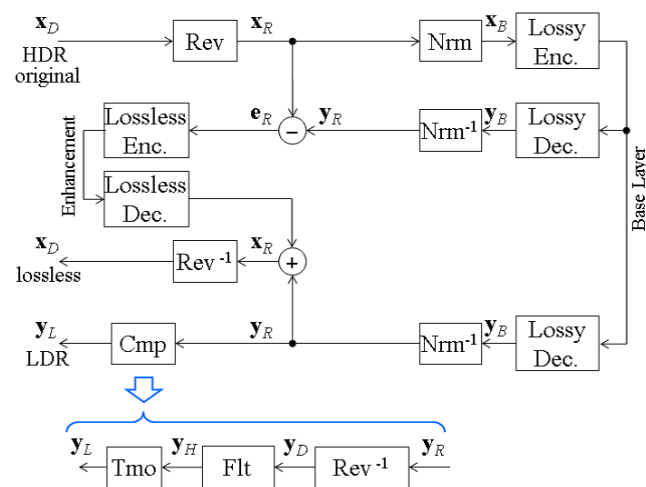


図 2: 提案法タイプ 1

3.2 RGBE 画像のフォーマット変換

以下、フォーマット B の入力画像に対する特別な処理を提案する。我々の以前の報告書 [9] では、フォーマット A の入力画像に対して、式 (20) の ' REV_A ' と式 (5) の ' FLT_A ' は、図 2 の ' Rev ' と ' Flt ' として使用されていた。それを、式 (23) の ' REV_B ' と式 (9) の ' FLT_B ' で置き換えることで、フォーマット B に拡張することができる。以下では、これを提案法タイプ 1 とする。

しかし、5.1 で示すように、その時の LDR 画像 y_L の画質は低下する。本報告では、符号化性能を向上させるため、図 3 に示すように、フォーマット変換 ' CNV ' を導入する。以下では、これを提案法タイプ 2 とする。フォーマット変換 ' CNV ' は、図 4 のように、' REV_A '、' FLT_A ' と ' REV_B ' の逆変換で構成されている。この処理は、システム内部でフォーマット B の画像をフォーマット A に一時的に変換する。

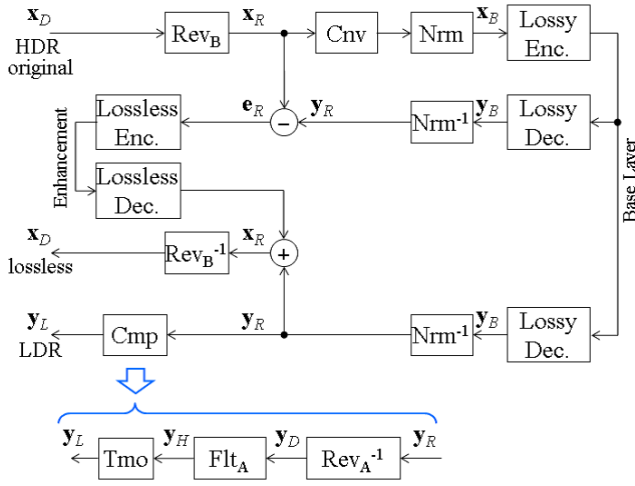


図 3: 提案法タイプ 2

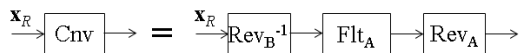


図 4: フォーマット変換 ' Cnv '

4 理論解析

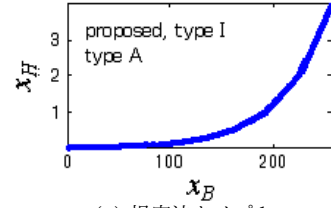
3.2 でフォーマット変換を導入する根拠を説明する。

3.2 のフォーマット変換により、フォーマット B の入力画像に対する、基本層の LDR 画質の劣化が軽減されることを示す。フォーマット B の入力画像に対して、提案法タイプ 1 (図 2) は、

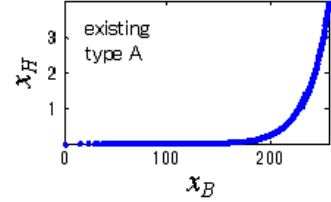
$$\mathbf{y}_H = Flt_B(Rev_B^{-1}(\mathbf{y}_R)) \quad (26)$$

のように \mathbf{y}_H を生成する。したがって、式 (23) と式 (24) より、

$$x_{E,0} = (x_{R,c} - x_{M,c})/256 + C_{min}^* \quad (27)$$

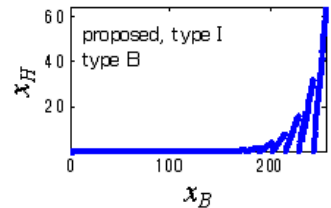


(a) 提案法タイプ 1

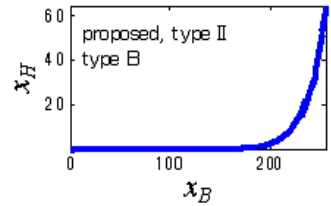


(b) 従来法

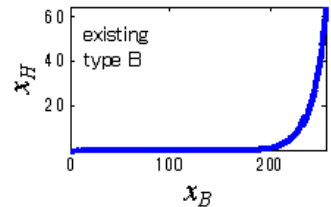
図 5: フォーマット A の画像「Canon」への写像



(a) 提案法タイプ 1



(b) 提案法タイプ 2



(c) 従来法

図 6: フォーマット B の画像「Belgium」への写像

を得る。式 (7) に式 (27) を代入し、

$$\begin{aligned} x_{H,c} &= \frac{x_{M,c} + 0.5}{256} \cdot 2^{(x_{R,c} - x_{M,c})/256 + C_{min}^* - 128} \\ &= f_{Ia}(x_{R,c}) \cdot f_{Ib}(x_{M,c}) \cdot 2^{C_{min}^* - 128} \end{aligned} \quad (28)$$

を得る。ただし、このとき、

$$\begin{cases} f_{Ia}(x_{R,c}) = 2^{x_{R,c}/256} \\ f_{Ib}(x_{M,c}) = \frac{\delta_A + 2^{-9}}{2^{\delta_A}} \in [0.002, 0.5] \\ \delta_A = \frac{x_{M,c}}{256} \in [0, 1] \end{cases} \quad (29)$$

とする。

一方、提案法タイプ 2 (図 3) は、

$$\mathbf{y}_H = Flt_A(Rev_A^{-1}(\mathbf{y}_R)) \quad (30)$$

より \mathbf{y}_H を生成する。したがって、式 (20) と式 (22) より、

$$x_{E,c} = (x_{R,c} - x_{M,c})/1024 + C_{min}^* \quad (31)$$

を得る。 $x_{E,c} > 0$ のとき、式 (1) に式 (31) を代入し、

$$\begin{aligned} x_{H,c} &= \frac{x_{M,c} + 1024}{256} \cdot 2^{(x_{R,c} - x_{M,c})/1024 + C_{min}^* - 15} \\ &= f_{IIa}(x_{R,c}) \cdot f_{IIb}(x_{M,c}) \cdot 2^{C_{min}^* - 15} \end{aligned} \quad (32)$$

を得る。ただし、このとき、

$$\begin{cases} f_{IIa}(x_{R,c}) = 2^{x_{R,c}/1024} \\ f_{IIb}(x_{M,c}) = \frac{\delta_B + 2^0}{2^{\delta_B}} \in [1, 1.06] \\ \delta_B = \frac{x_{M,c}}{1024} \in [0, 1] \end{cases} \quad (33)$$

とする。

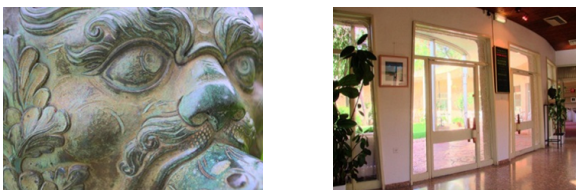
ここで、式 (28) と式 (32) を使用して、提案法タイプ 1 と提案法タイプ 2 を比較する。 $t \in \{ \text{提案法タイプ 1}, \text{提案法タイプ 2} \}$ として、両者を

$$\begin{aligned} \mathbf{y}_H &= Flt_t(Rev_t^{-1}(\mathbf{y}_R)) \\ &= Flt_t(Rev_t^{-1}(Nrm_t^{-1}(\mathbf{y}_R))) \\ &= f_t(\mathbf{y}_B) \end{aligned} \quad (34)$$

のように書き換える。図 5(a) に、フォーマット A の入力画像に対する t =提案法タイプ 1 の式 (34) を示す。これは、式 (29) の δ_A により、区分的線形である。すなわち、不連続である。一方、従来法では、図 5 のように連続している。図 6(a) には、フォーマット B の入力画像に対する t =提案法タイプ 1 の式 (34) を示す。この場合、図 5(a) は、より不連続である。図 6(b) に示すように、これは提案法タイプ 2 で緩和される。結果は、図 6(c) の従来法に近接している。

5 結果

図 7 には、フォーマット A の「Cannon」(566 × 780 画素) とフォーマット B の「Belgium」(769 × 1025 画素) をトーンマッピングした画像 \mathbf{y}_L を示す。以下、オリジナルの HDR 画像としてこの画像を、基本層のロッキー符号化と拡張層のロスレス符号化には JPEG2000 国際標準のロッキーモードとロスレスモードを、それぞれ使用した。



(a)フォーマットA「Cannon」 (b)フォーマットB「Belgium」

図 7: 実験で評価した浮動小数点画像

5.1 基本層

図 8 には、フォーマット A の入力画像に対する、提案法タイプ 1 の基本層におけるレート歪み曲線を示す。ビットレートは、色成分ごとの画素当たりのビット数 [bpp] である。歪みは、

$$PSNR = 10 \log_{10} \frac{255^2}{E[(\mathbf{y}_L - \mathbf{x}_L)^2]} [dB] \quad (35)$$

として定義された peak signal to noise ratio (PSNR) [dB] である。ここで、 $E[\cdot]$ は画像内の全画素の集合平均である。

提案法タイプ 1 と従来法との差は、基本層のビットレートが例えば 3.1[bpp] のとき、0.46[dB] である。これにより、提案法が従来法の符号化性能を維持していることが確認できる。

図 9(a) には、フォーマット B の入力画像に対する、提案法タイプ 1 の基本層におけるレート歪み曲線を示す。これにより、提案法タイプ 1 (=我々の以前の手法 [9]) では、LDR 画像の品質が著しく劣化することが確認できる。図 9(b) には、提案法タイプ 2 の基本層におけるレート歪み曲線を示す。これにより、提案法タイプ 2 は従来法より若干優れていることが確認できる。

その結果、提案法は、従来法の基本層の符号化性能を

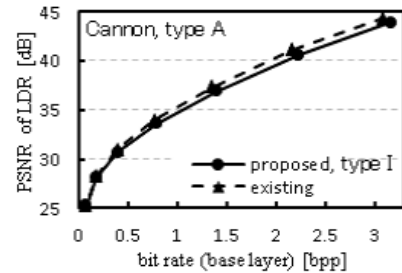
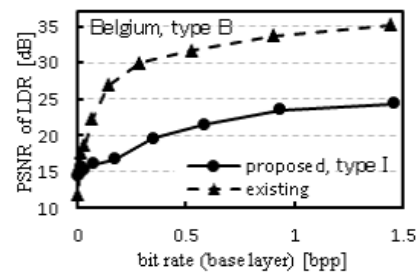
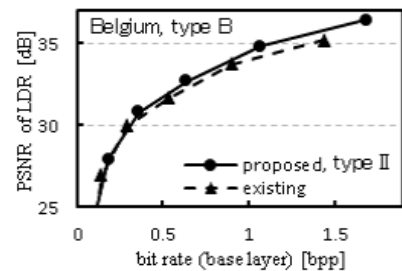


図 8: 基本層のレート歪み曲線 (入力: フォーマット A)



(a) 提案法タイプ 1



(b) 提案法タイプ 2

図 9: 基本層のレート歪み曲線 (入力: フォーマット B)

フォーマット A とフォーマット B の両方で維持していることが確認された。

5.2 拡張層

図 10 は、フォーマット A の入力画像に対する、提案法タイプ 1 の拡張層の符号化性能を示す。縦軸は、拡張層のビットレート [bpp] である。横軸は、復号された LDR 画像の PSNR を示す。復号された HDR 画像の PSNR は、ロスレスであるため無限である。図 10 より、提案法は従来法に比べて、ビットレートを 3.5[bpp] 以上低減していることが確認できる。

図 11(a) は、フォーマット B の入力画像に対する、提案法タイプ 1 の符号化性能を示す。この場合、図 9(a) に示すように、提案法タイプ 1 では LDR 画像が高画質に再生されない。図 11(b) は、フォーマット B の入力画像に対する、提案法タイプ 2 の符号化性能を示す。

図 10 の場合と同様に、提案法は従来法に比べて、ビットレートを 5.8[bpp] を以上低減していることが確認できる。

その結果、提案法は、フォーマット A とフォーマット B の両方で拡張層のビットレートを低減することが確認された。

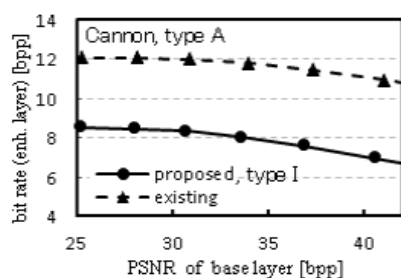
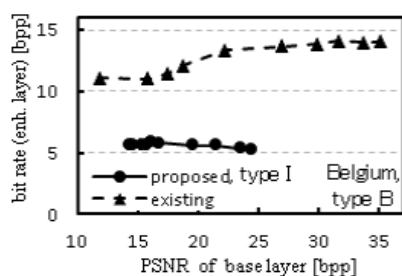
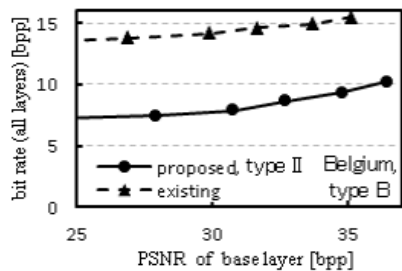


図 10: 拡張層の符号化性能 (入力: フォーマット A)



(a) 提案法タイプ 1



(b) 提案法タイプ 2

図 11: 拡張層の符号化性能 (入力: フォーマット B)

6 まとめ

本報告では、浮動小数点データフォーマットの HDR 画像のビット深度スケーラブル符号化を提案した。これは、圧縮データの一部から LDR 画像を復号することができる。ほとんどの従来のスケーラブル符号化方式とは異なり、提案手法では、損失無く (=ロスレス)、オリジナルの HDR 画像を復号することができる。可逆対数写像を導入した提案法は、従来法の基本層の符号化性能を維持したまま、拡張層のビット深度とビットレートを低減する。我々の調査は”差”に基づく手法に限定されているので、例えば、[4] のように”比”に基づく手法を含むことが必要である。

参考文献

- [1] E. Reinhard, G. Ward, S. Pattanaik, P. Debevec, W. Heidrich and K. Myszkowski, “High Dynamic Range Imaging - Acquisition, Display and Image based Lighting,” Morgan Kaufmann, 2010.
- [2] G. Ward, “Real Pixels, Graphics Gems II,” Academic Press, Boston, 1991.
- [3] R. Bogart, F. Kainz, and D. Hess, “OpenEXR Image File Format,” ACM International Conference and Exhibition on Computer Graphics and Interactive Techniques, Sketches & Applications, 2003.
- [4] G. Ward, and M. Simmons, “JPEG-HDR: A backwards-compatible, high dynamic range extension to JPEG,” Thirteenth Color Imaging Conference, Nov. 2005.
- [5] R. Mantiuk, A. Efremov, K. Myszkowski and H. P. Seidel, “Backward compatible high dynamic range MPEG video compression,” ACM Trans. Graphics, International Conference and Exhibition on Computer Graphics and Interactive Techniques, vol.25, issue 3, pp.713-723, July 2006.
- [6] M. Winken, D. Marpe, H. Schwarz and T. Wiegand, “Bit-depth scalable video coding,” IEEE International Conference on Image Processing, no.I, pp.5-8, Sept. 2007.
- [7] T. Jinno, M. Okuda and N. Adami, “New Local tone mapping and two-layer coding for HDR Images,” IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.765-768, March 2012.
- [8] M. Iwahashi and H. Kiya, “Efficient lossless bit depth scalable coding for HDR images,” APSIPA Annual Summit and Conference, OS.49-IVM.17, Dec. 2012.
- [9] C. Y. Ping, T. Shibata, M. Iwahashi and H. Kiya, “Lossless bit depth scalable coding for floating point images,” International Workshop on Advanced Image Technology, pp.169-174, Jan. 2013.
- [10] James F. Blinn, “Floating -point tricks,” IEEE Computer Graphics and Applications, pp.81-84, August 1997.