

固定小数点トーンマッピング処理のための HDR 画像フォーマット制約の緩和

田代 敦[†] 室伏 辰耶[†] 土橋 俊之[†] 岩橋 政宏^{††} 貴家 仁志[†]

[†] 首都大学東京システムデザイン学部 〒191-0065 東京都日野市旭が丘 6-6

^{††} 長岡技術科学大学 〒940-2188 新潟県長岡市上富岡町 1603-1

E-mail: †{tashiro-atsushi1,murofushi-tatsuya,dobashi-toshiyuki}@ed.tmu.ac.jp,kiya@tmu.ac.jp,
††iwahashi@vos.nagaokaut.ac.jp

あらまし 本稿では、浮動小数点数で表現されたハイダイナミックレンジ (HDR) 画像を対象に、整数化トーンマッピング処理の固定小数点演算法を提案する。トーンマッピング処理は HDR 画像のダイナミックレンジを圧縮し、ローダイナミックレンジ (LDR) 画像を生成する処理である。一般に、HDR 画像の画素値は浮動小数点数で表現されるため、トーンマッピング処理は浮動小数点演算で行われる。このため、トーンマッピング処理には膨大な演算コストが要求される。浮動小数点数を指数部と仮数部に分割する整数化処理および固定小数点演算法を用いて処理を行う方法が提案された。しかし、先行研究では、HDR 画像フォーマットが限定されており、種々のフォーマットを有する HDR 画像に対しては適用することが困難である。提案法では中間フォーマットを用いることで、フォーマットに依存しないトーンマッピング処理を行うことを可能とする。実験では、複数の形式で表現された HDR 画像に対して提案法が精度を保ちつつ演算コストを低減していることを示した。

キーワード ハイダイナミックレンジ, トーンマッピング, 固定小数点演算, 浮動小数点数表現

Relaxation of Restriction on HDR Image Formats for A Fixed-Point Tone Mapping Operation

Atsushi TASHIRO[†], Tatsuya MUROFUSHI[†], Toshiyuki DOBASHI[†], Masahiro IWAHASHI^{††}, and
Hitoshi KIYA[†]

[†] Tokyo Metropolitan University, Asahigaoka 6-6, Hino-shi, Tokyo, 191-0065 Japan

^{††} Nagaoka University of Technology, Kamitomiokamachi 1603-1, Nagaoka-shi, Niigata, 940-2188 Japan

E-mail: †{tashiro-atsushi1,murofushi-tatsuya,dobashi-toshiyuki}@ed.tmu.ac.jp,kiya@tmu.ac.jp,
††iwahashi@vos.nagaokaut.ac.jp

Abstract This paper proposes an integer tone mapping operation (TMO) with fixed-point arithmetic for high dynamic range (HDR) images expressed in floating-point data. A TMO generates a low dynamic range (LDR) image from HDR image by compressing its dynamic range. In general, it uses floating-point arithmetic because values of HDR images are expressed in floating-point data. For this, it requires huge computational costs, even though LDR images are expressed in integer data. The method with fixed-point arithmetic and integer processing in which a floating-point number is divided into the exponent and the mantissa was proposed. However, in the previous research, HDR image formats are restricted, so it is difficult to apply integer processing to HDR images having various formats. By using an intermediate format, the proposed TMO is available for any HDR images. The results show reducing the computational cost while maintaining the numerical accuracy.

Key words tone mapping, high dynamic range, fixed-point arithmetic, floating-point data

1. はじめに

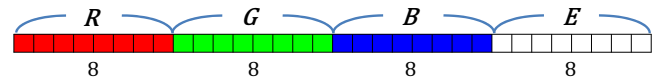
ハイダイナミックレンジ (HDR) 画像は、コンピュータグラフィックスや写真の分野で主に用いられており、医用画像や車載カメラなど様々な分野でも普及し始めている [1]~[3]。これまで、RGBA フォーマットや OpenEXR フォーマットなど、いくつかの HDR 画像フォーマットが提案されてきた [1]。これらの画像フォーマットは、浮動小数点数で表現されており、非常に広いダイナミックレンジを持つ情報が保存できる。しかし、広いダイナミックレンジを表示できるような表示機器の普及は、まだ十分ではない。そこで、従来の表示機器で表示するために、ダイナミックレンジを圧縮し、ローダイナミックレンジ (LDR) 画像を生成するトーンマッピング処理が重要である。

トーンマッピング処理に関しても、これまでに多くの手法が提案されている [1], [4]~[11]。トーンマッピング処理の手法は、大きくグローバルオペレータ [4]~[6] とローカルオペレータ [4], [9]~[11] の 2 種類に分類される。グローバルオペレータは、すべての画素に同一の処理を行う手法であり、演算コストが低い。ローカルオペレータは、局所的に異なる処理を行う手法であり、細部の表現に優れるが演算コストが高い。どちらの手法を用いる場合においても、入力データが浮動小数点数表現であるため、浮動小数点演算になってしまう。実際の画像処理や組み込みシステムでは、実装規模が小さく、高速な演算が可能な固定小数点演算が用いられる [12]~[14]。しかし、HDR 画像のように広いダイナミックレンジを持つデータに対して固定小数点演算を実現するのは困難である。そのため、演算に用いる各入出力データのダイナミックレンジを削減する必要がある。本稿では、整数化トーンマッピング処理 [15] を行い、演算におけるダイナミックレンジを削減することで、固定小数点演算での実装を可能としている。

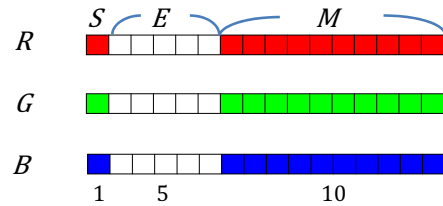
整数化トーンマッピング処理 [15] は、HDR 画像に対するトーンマッピング処理の各入出力を整数値に分解する処理である。文献 [16] ではこの整数化トーンマッピング処理の手法を基に RGBA フォーマットに関する固定小数点演算でのトーンマッピング処理を実現している。しかし、この手法 [16] は RGBA フォーマットのみを対象にしているため、画素値の格納形式が異なる他の HDR 画像フォーマットに関して結果は保証されていない。

本稿では、文献 [16] の手法を基に、浮動小数点数表現された HDR 画像に対して、中間フォーマットを用いた整数化トーンマッピング処理を固定小数点演算で実行する方法を提案する。中間フォーマットを用いることで、画素値の格納形式にとらわれず、浮動小数点数で表現された HDR 画像を対象にトーンマッピング処理を行うことができる。

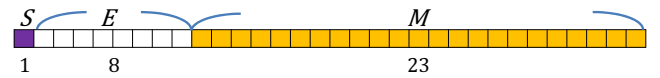
実験では、RGBA フォーマットと OpenEXR フォーマットの HDR 画像を用いて、浮動小数点演算で処理する従来法 [4] と提案法の比較を行い、提案法が従来のトーンマッピング処理と同等の精度を保ちつつ、メモリコストや処理速度といった演算に必要なコストを削減していることを示す。



(a) RGBA フォーマット



(b) OpenEXR フォーマット



(c) IEEE754 単精度形式

図 1: HDR 画像フォーマット

2. 準備

本節では、対象となる HDR 画像フォーマットの構成 [1]、代表的なトーンマッピング処理 [4] の手順、および課題点について説明する。

2.1 HDR 画像フォーマット

ここでは、使用した HDR 画像フォーマットの構成を紹介する。図 1 に構成を示す。

(a) RGBA フォーマット

RGBA フォーマットの構成は、RGB 各チャンネルごとに仮数部 8 ビット、そして RGB 共通の指数部が 8 ビットの 1 ピクセルあたり 32 ビットである。画素値 $F_1(p)$ と指数部 $F_{1E}(p)$ と仮数部 $F_{1M}(p)$ の関係は、以下の通りである。

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

(b) OpenEXR フォーマット

OpenEXR フォーマットの構成は、RGB 各チャンネルごとに符号部 1 ビット、指数部 5 ビット、仮数部 10 ビットの 1 ピクセルあたり 48 ビットである。RGBA と異なり、OpenEXR には符号部が存在し、負の値を表現することができるが、幾何平均値を取る過程を考慮し、本稿では負の値はゼロ値として取り扱う。

また、OpenEXR における数は、正規化数と非正規化数が存在する。非正規化数は、正規化数では表現できない絶対値の小さい数値を表現するための数である。OpenEXR には、正規化数と非正規化数に対応した 2 種類の表現が存在する。正規化数における画素値 F_2 と指数部 F_{2E} 、仮数部 F_{2M} の関係は、

$$F_2 = (-1)^{sign} \cdot (1 + F_{2M} \cdot 2^{-10}) \cdot 2^{F_{2E} - 15} \quad (2)$$

と表す。一方、非正規化数においては、

$$F_2 = (-1)^{sign} \cdot (F_{2M} \cdot 2^{-10}) \cdot 2^{-14} \quad (3)$$

と表される。

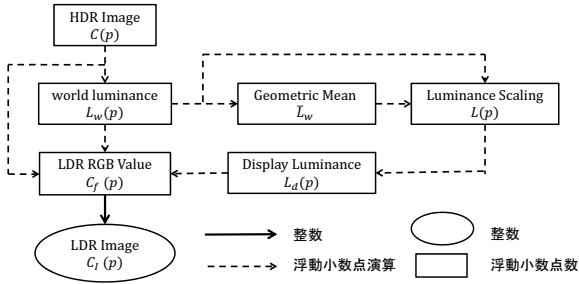


図 2: 従来法 [4] のブロック図

(c) IEEE754 形式

一般に、HDR 画像の表現形式は浮動小数点数形式である。IEEE754 で定められている単精度浮動小数点数は、RGB の各チャンネルごとに符号部 1 ビット、指数部 8 ビット、仮数部 23 ビットの 1 ピクセルあたり 96 ビットで構成されている。画素値 F_3 と指数部 F_{3E} 、仮数部 F_{3M} の関係は以下の通りである。

$$F_3 = (-1)^{\text{sign}} \cdot (1 + F_{3M} \cdot 2^{-23}) \cdot 2^{F_{3E}-127} \quad (4)$$

2.2 従来法とその課題点

トーンマッピング処理は、浮動小数点表現された HDR 画像から、整数表現された LDR 画像を生成する処理である。トーンマッピング処理手法にはグローバルオペレータとローカルオペレータがあるが、本稿ではグローバルオペレータを取り扱う。代表的なトーンマッピング処理 [4] の手順を図 2 に示す。

一般に、HDR 画像に対するトーンマッピング処理は、浮動小数点数表現された RGB 値 $C(p) \in \{R(p), G(p), B(p)\}$ から始まる。各画素 p において HDR 画像の RGB 値 $C(p)$ から HDR 画像の輝度値 $L_w(p)$ を以下の通り計算する。

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

次に、HDR 画像の輝度値 $L_w(p)$ の幾何平均値 \bar{L}_w を以下の通り計算する。

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

ここで、 N は HDR 画像の総ピクセル数である。式 (3) において、文献 [4] では微小値を用いて、画素値がゼロのピクセルに起因する特異性を回避している。しかし、RGBE フォーマットや OpenEXR フォーマットといった HDR 画像は非常に小さい数値まで取り扱うことができるため、この微小値が出力画像に与える影響を無視できなくなる。本稿では、微小値を使用せずに、 $L_w(p) \neq 0$ のピクセルのみを用いて処理を行うこととする。

次に、スケーリングされた輝度 $L(p)$ を以下の通り計算する。

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

ここで、 $k \in [0, 1]$ は "key value" と呼ばれるパラメータである。

次に、トーンマッピング関数 $y()$ を用いて、ディスプレイ輝度 $L_d(p)$ を以下の通り計算する。

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

出力として得られる結果は、ここでのトーンマッピング関数の選択に依存する。代表的なトーンマッピング関数の一つである Reinhard のグローバルオペレータ [4] は以下の通り定義されている。

$$y_R(L(p)) = \frac{L(p)}{1 + L(p)} \quad (9)$$

本稿では、従来法として式 (9) を用いている。次に、浮動小数点数表現された LDR 画像 $C_f(p)$ を以下の通り求める。

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

$$C(p) \in \{R(p), G(p), B(p)\}$$

$$C_f(p) \in \{R_f(p), G_f(p), B_f(p)\}$$

最後に、 $C_f(p)$ を 255 倍して整数値に丸め込むことにより、LDR 画像の RGB 値 $C_l(p)$ を 8 ビットのカラー画像として生成する。

$$C_l(p) = \text{round}(C_f(p) \cdot 255) \quad (11)$$

$$C_l(p) \in \{R_l(p), G_l(p), B_l(p)\}$$

ここで、 $\text{round}(x)$ は x を最も近い整数値に丸め込むことを意味する。

以上の手順では、最終的に得られる LDR 画像の RGB 値は整数であるにもかかわらず、それ以外の入出力と内部演算はすべて浮動小数点数である。

文献 [16] では、この手法を固定小数点演算で実現しているが、入力として RGBE フォーマットのみを対象としているため、OpenEXR のような表現形式の異なる HDR 画像フォーマットに対する動作が保証できないという制約が存在する。提案法では、入力に中間フォーマットを用いることで、この入力フォーマットの制約を緩和している。

3. 提案法

本節では、入力として用いる中間フォーマットの構成と中間フォーマットを用いた整数化処理の流れ、および固定小数点演算における注意点を示す。

3.1 中間フォーマット

図 3 は、本稿で提案する中間フォーマットの説明図である。本稿では、各種の HDR 画像を指数部と仮数部をそれぞれ 8 ビットに設定した中間フォーマットに変換することを提案する。このフォーマットを用いることで、形式の異なる HDR 画像フォーマットに対して、共通の整数化トーンマッピング処理を行うことができる。RGB 各チャンネルごとに指数部を独立することで、この中間フォーマットに変換する際の誤差を軽減することができる。中間フォーマットの構成は RGB 各チャンネルごとに仮数部 8 ビット、指数部 8 ビットの 1 ピクセルあたり 48 ビットである。

画素値 F と指数部 F_E および仮数部 F_M の関係は以下の通りである。

$$F = (F_M + 0.5) \cdot 2^{F_E-136} \quad (12)$$

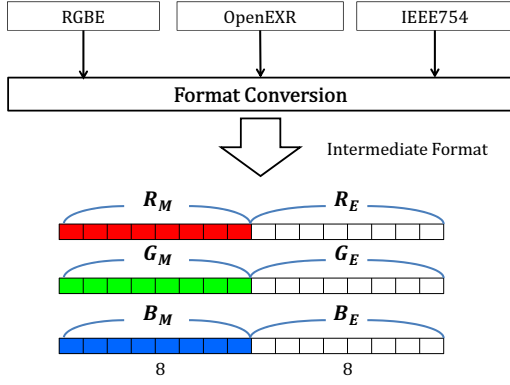


図 3: 中間フォーマットへのフォーマット変換

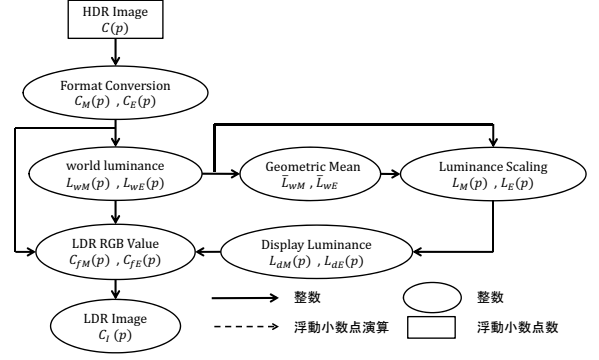


図 4: 提案法のブロック図

また、逆の関係として、指数部 F_E および仮数部 F_M は、

$$F_E = \lceil \log_2 F + 128 \rceil \quad (13)$$

$$F_M = \lfloor F \cdot 2^{136 - F_E} \rfloor \quad (14)$$

のように求められる。

式 (1) の F_1 , 式 (2) および (3) の F_2 , 式 (4) の F_3 を式 (13) および (14) の F としてそれぞれ用いることで、中間フォーマットの指数部と仮数部を一意に求めることができる。

3.2 整数化トーンマッピング処理

提案法は、整数化トーンマッピング処理 [15] に基づいている。整数化トーンマッピング処理は、トーンマッピングの各入出力を整数値にすることで、リソースの削減を図る手法である。この手法は図 4 のようにまとめられる。ただし、文献 [15] においては、各手順の内部演算は浮動小数点演算で行っている。

提案法では、最初に 2.1 節のように与えられる HDR 画像の RGB 値 $C(p)$ に対して、3.1 節に示した中間フォーマットへの変換を行う。(図 3)

これにより、 $C(p) \in R(p), G(p), B(p)$ の指数部 $C_E(p)$ および仮数部 $C_M(p)$ が

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

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

$$C_E(p) \in \{R_E(p), G_E(p), B_E(p)\}$$

$$C_M(p) \in \{R_M(p), G_M(p), B_M(p)\}$$

のように計算される。

次に、各処理に関する具体的な計算を示す。各計算式は従来法 [4] の式 (5)-(11) および中間フォーマットの指数部と仮数部の関係式 (12) から得られる。また、中間フォーマットへの変換を含めた各入出力値の範囲は $0 \leq x \leq 255$ である。

まず、HDR 画像の輝度 $L_w(p)$ の指数部 $L_{wE}(p)$ と仮数部 $L_{wM}(p)$ は以下の通り計算する。

$$L_{wE}(p) = \lceil \log_2 ML(p) + 128 \rceil \quad (17)$$

$$L_{wM}(p) = \lfloor ML(p) \cdot 2^{136 - L_{wE}(p)} \rfloor \quad (18)$$

$$ML(p) = 0.27(R_M(p)) \cdot 2^{R_E(p) - 136} \\ + 0.67(G_M(p)) \cdot 2^{G_E(p) - 136}$$

$$+ 0.06(B_M(p)) \cdot 2^{B_E(p) - 136}$$

$ML(p)$ の計算は、RGB がすべてゼロ値の場合は $ML(p) = 0$ とし、以降の計算では該当するピクセルはすべてゼロとして扱う。

得られた $L_{wE}(p)$ と $L_{wM}(p)$ から、輝度の幾何平均値 \bar{L}_w の指数部 \bar{L}_{wE} と仮数部 \bar{L}_{wM} を次のように計算する。

$$\bar{L}_{wE} = \lceil SL_{wE} + SL_{wM} + 128 \rceil \quad (19)$$

$$\bar{L}_{wM} = \lfloor 2^{SL_{wE} + SL_{wM} - \bar{L}_{wE} + 136} \rfloor \quad (20)$$

$$SL_{wE} = \frac{1}{N} \sum_p (L_{wE}(p) - 136) \quad (21)$$

$$SL_{wM} = \frac{1}{N} \sum_p \log_2(L_{wM}(p) + 0.5) \quad (22)$$

この時、 $L_{wE}(p)$ が非ゼロの画素のみを用いて計算する。

これまでに求めた $L_{wE}(p), L_{wM}(p), \bar{L}_{wE}, \bar{L}_{wM}$ から、スケールリングされた輝度 $L(p)$ の指数部 $L_E(p)$ と仮数部 $L_M(p)$ が以下のように計算される。

$$L_E(p) = \lceil \log_2 AL_w(p) + L_{wE}(p) - \bar{L}_{wE} + 128 \rceil \quad (23)$$

$$L_M(p) = \lfloor AL_w(p) \cdot 2^{L_{wE}(p) - L_E(p) - \bar{L}_{wE} + 136} \rfloor \quad (24)$$

$$AL_w(p) = k \cdot \frac{L_{wM}(p) + 0.5}{\bar{L}_{wM} + 0.5}$$

次に、ディスプレイ輝度 $L_d(p)$ の指数部 $L_{dE}(p)$ と仮数部 $L_{dM}(p)$ の計算を行う。この計算は、選択したトーンマッピング関数によって計算式が異なる。提案法では、式 (9) のトーンマッピング関数を選択する。

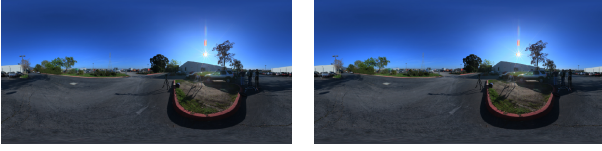
$$FL(p) = \frac{L_M(p) + 0.5}{L_M(p) + 0.5 + 2^{136 - L_E(p)}} \quad (25)$$

$$L_{dE}(p) = \lceil \log_2(FL(p)) + 128 \rceil \quad (26)$$

$$L_{dM}(p) = \lfloor FL(p) \cdot 2^{136 - L_{dE}(p)} \rfloor \quad (27)$$

最終的に 8 ビットカラー画像として生成する LDR 画像の RGB 値 $C_I(p)$ は、

$$C_I(p) = \text{round} \left(RL(p) \cdot 2^{C_E(p) + L_{dE}(p) - L_{wE}(p) - 136} \cdot 255 \right) \quad (28)$$



(a) 従来法 [4] (b) 提案法

図 5: 生成した LDR 画像

$$RL(p) = \frac{(L_{dM}(p) + 0.5)(C_M(p) + 0.5)}{L_{wM}(p) + 0.5}$$

によって計算することができる。

以上の計算を用いることで、トーンマッピング処理の各入出力を、すべて指数部と仮数部に対応する整数値で実現することができる。これにより、演算時のダイナミックレンジを大きく削減できるため、式 (25) 以外の演算は固定小数点演算による実装が可能になる。3.3 節では、固定小数点演算で式 (25) を計算する方法を述べる。提案法では、2.1 節 (a)-(c) で示した各形式から式 (15)(16) における中間フォーマットへの変換による誤差と、整数化トーンマッピング処理の各手順における量子化誤差が発生するが、4 節で行った実験により、この誤差が十分に小さく、視覚的な品質に影響を及ぼさないことを示す。

3.3 固定小数点演算

3.2 節で述べた整数化トーンマッピング処理によって、各入出力の値は整数化されている。そのため、ほとんどの計算は固定小数点演算が可能である。しかし、式 (25) においては、分母の値の範囲が非常に広いため、固定小数点で演算を行うのが難しい。提案法では、固定小数点演算を行うために、式 (25) を次のように変形する。

$$FL(p) = \frac{1}{1 + \frac{2^{136-L_E(p)}}{L_M(p)+0.5}} \quad (29)$$

さらに、式 (29) の分母の値に応じて、以下のように場合分けを行う。

(i) 式 (29) において、 $136 - L_E(p) > 22$ の場合、分母の右側部分が非常に大きくなるため、分母の 1 は無視することができる。これより、式 (25)(27) は次のように近似できる。

$$FL(p) = \frac{L_M(p) + 0.5}{2^{136-L_E(p)}} \quad (30)$$

$$L_{dE}(p) = \lceil \log_2(L_M(p) + 0.5) - (136 - L_E(p)) + 128 \rceil \quad (31)$$

$$L_{dM}(p) = \lfloor (L_M(p) + 0.5) \cdot 2^{L_E(p) - L_{dE}(p)} \rfloor \quad (32)$$

(ii) 式 (29) において、 $136 - L_E(p) < -16$ の場合、分母の右部分は非常に小さくなるため、無視することができる。これより、式 (25)-(27) は、次のように近似できる。

$$FL(p) = 1 \quad (33)$$

$$L_{dE}(p) = 128 \quad (34)$$

$$L_{dM}(p) = 255 \quad (35)$$

(iii) それ以外の場合は、固定小数点演算で計算可能である。

表 1: 従来法 [4] と提案法の PSNR

入力フォーマット	PSNR[dB]		
	最大値	最小値	平均値
RGBE	57.43	52.28	55.67
OpenEXR	69.26	48.89	57.27

表 2: 従来法 [4] と提案法の処理速度の比

入力形式	従来法 [4] / 提案法 [倍]
RGBE	13.17
OpenEXR	10.77
IEEE754	7.39

(i)-(iii) の場合分けに加えて、提案法では式 (20)(22) に関するテーブルを用いる。

以上で、トーンマッピング処理のすべての手順を固定小数点演算で行うことが可能となる。

4. 実験と評価

本稿では、複数のフォーマットの HDR 画像のトーンマッピング処理を固定小数点演算のみで行う手法を提案しており、これによって演算コストの削減を図っている。しかし、固定小数点演算を用いることにより、精度の低下が懸念される。

演算コストと精度を確認するため、浮動小数点演算を用いる従来法 [4] と提案法の間で、処理速度の比較、生成する LDR 画像の比較、およびメモリ使用量に関する比較評価を行った。32 枚の RGBE フォーマットと 42 枚の OpenEXR フォーマットを対象に実験を行い、実装には C 言語を用いた。提案法では、8 ビットの整数データと 32 ビットの固定小数点演算を用い、従来法 [4] では、64 ビットの浮動小数点データと 64 ビットの浮動小数点演算をそれぞれ用いた。

4.1 LDR 画像の比較

表 1 は従来法 [4] と提案法を用いて生成された LDR 画像の PSNR の値である。また、図 5 は従来法 [4] と提案法で生成した LDR 画像である。2 枚の画像の視覚的な違いを見つけることは難しい。更に、PSNR も高い値を示しており、RGBE フォーマットは平均 55.67 dB であり、OpenEXR フォーマットでは平均 57.27 dB である。これらの結果から、提案法が精度を維持したトーンマッピング処理を行っていることが示された。

4.2 処理時間の比較

表 2 は従来法 [4] と提案法の処理時間に対する比である。本実験では、入力として 2.1 節の (a)-(c) の形式をそれぞれ入力として用いた場合の処理時間を計測した。計測に用いた画像は RGBE, OpenEXR とともに 393216 pixel の画像であり、使用した CPU は Marvell PXA270 624MHz(ARM プロセッサ) である。表 2 から提案法が従来法 [4] に対して、7.39 倍~13.17 倍高速な処理を行うことを示している。この結果から、提案法が固定小数点演算を用いることで、演算コストを削減していることが確認された。

表 3: 従来法 [4] と提案法のメモリコスト

各入出力 データ	メモリ使用量 [bits]	
	従来法 [4]	提案法
HDR 画像の RGB 値	$M \times N \times 192$	$M \times N \times 48$
輝度値	$M \times N \times 64$	$M \times N \times 16$
輝度の幾何平均値	64	16
LDR 画像の RGB 値	$M \times N \times 24$	$M \times N \times 24$
テーブル	—	8192

4.3 メモリコストの比較

表 3 は従来法 [4] と提案法の確保するメモリ量について示している。ここで、 M, N は画像サイズである。表中に記載されていない項目はトーンマッピング処理において、一時データとして処理することができるため、評価には含まない。

表 3 の結果より、提案法がトーンマッピング処理に必要なメモリ量を、従来法 [4] の 31.4%まで削減していることを確認した。

5. まとめ

本稿では、複数の HDR 画像フォーマットを対象に、固定小数点演算によるトーンマッピング処理を実現する方法を提案し、演算コストの削減を確認した。提案法は、文献 [16] における入力画像フォーマットの制約を、RGBE フォーマットを拡張した中間フォーマットを用いることで緩和している。これにより、FPU を内蔵していないプロセッサや少ないメモリを搭載したシステムでも、フォーマット選択の自由度が高いトーンマッピング処理が可能となっている。

実験と評価から、提案法は RGBE だけでなく、OpenEXR および IEEE754 形式で表現された HDR 画像に対して精度を保ちつつ、メモリ使用量を 31.4%まで削減し、さらに最大で 13.17 倍高速なトーンマッピング処理を行っていることを確認した。

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] M. Iwahashi and H. Kiya, "Efficient Lossless Bit Depth Scalable Coding for HDR Images" *APSIPA ASC 2012*, no.OS.49-IVM.17-5, December 2012.
- [3] M. Iwahashi and H. Kiya, "Two Layer Lossless Coding of HDR Images" *IEEE ICASSP 2013*, pp.267-276, Vancouver, BC, Canada, 28th may, 2013.
- [4] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic Tone Reproduction for Digital Images" *SIGGRAPH '02*, pp.267-276, New York, NY, USA, July 2002. ACM.
- [5] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive logarithmic mapping for displaying high dynamic contrast scenes" *Computer Graphics Forum*, Vol.22, p.419-426, 2003.
- [6] S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg, "Time-Dependent Visual Adaption for Fast Realistic Image Display" *SIGGRAPH '00*, pp.47-54, New York, NY, USA, July 2000. ACM.
- [7] J. Duan, and G. Qiu, "Fast Tone Mapping with for High Dynamic Range Images" *Proc. International Conference Pat-*

tern Recognition (ICPR), Vol.2, pp.847-850, Aug 2004.

- [8] A. Koz, and F. Dudaux, "Optimized Tone Mapping with LDR Image Quality Constraint for Backward Compatible High Dynamic Range Image and Video Coding" *ICIP 2013*, pp.1762-1766, 2013.
- [9] K. Chiu, C. H. Shirley, P. Shirley, S. Swamy, C. Wang and K. Zimmerman, "Spartially Nonuniform Scaling Functions for High Contrast Images" *Processing of Graphics Interface '93*, pp.245-253, 1996.
- [10] F. Durand, and J. Dorsey, "Fast Bilateral Filtering for the Display of High-Dynamic-Range Images" *SIGGRAPH '02*, pp.257-266, New York, NY, USA, July 2002. ACM.
- [11] R. Fattal, D. Lischinski, and M. Werman, "Gradient Domain High Dynamic Range Compression" *SIGGRAPH '02*, pp.249-256, New York, NY, USA, July 2002. ACM.
- [12] T. Viitanen, P. Jaaskelainen, O. Esko, and J. Takala, "Simplified Floating-point Division and Square Root" *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp.2707-2711, 2013.
- [13] C. H. Lampert, and O. Wirjadi, "Anisotropic Gaussian Filtering Using Fixed Point Arithmetic" *ICIP 2006*, pp.1565-1568, Atlanta, GA, Oct 2006.
- [14] K. J. Hass, "Synthesizing Optimal Fixed -Point Arithmetic for Embedded Signal Processing" *MWSCAS 2010*, pp.61-64, Aug 2010.
- [15] T. Murofushi, M. Iwahashi, and H. Kiya, "An Integer Tone Mapping Operation for HDR Images Expressed in Floating Point Data" *Proc. IEEE Int. Conf. Acoustics, Speech and Processing*, pp.2479-2483, 2013.
- [16] T. Dobashi, T. Murofushi, M. Iwahashi, and H. Kiya, "A Fixed-Point Tone Mapping Operation for HDR Images in the RGBE Format" *APSIPA ASC 2013*, no.OS.37-IVM. 16-4, November 2013.