

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6867774号
(P6867774)

(45) 発行日 令和3年5月12日 (2021.5.12)

(24) 登録日 令和3年4月13日 (2021.4.13)

(51) Int.Cl.		F I	
H O 4 N	1/41	(2006.01)	H O 4 N 1/41
H O 4 N	19/36	(2014.01)	H O 4 N 19/36
H O 4 N	19/46	(2014.01)	H O 4 N 19/46

請求項の数 12 (全 26 頁)

(21) 出願番号	特願2016-203813 (P2016-203813)	(73) 特許権者	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22) 出願日	平成28年10月17日 (2016.10.17)	(74) 代理人	110003281 特許業務法人大塚国際特許事務所
(65) 公開番号	特開2018-67758 (P2018-67758A)	(72) 発明者	菊地 智恵 東京都大田区下丸子3丁目30番2号 キ ヤノン株式会社内
(43) 公開日	平成30年4月26日 (2018.4.26)		
審査請求日	令和1年10月4日 (2019.10.4)	審査官	松永 隆志

最終頁に続く

(54) 【発明の名称】 画像符号化装置及びその制御方法

(57) 【特許請求の範囲】

【請求項 1】

H D R 画像を符号化し、L D R 画像を H D R 画像の精度に戻すための補助データを生成する画像符号化装置であって、

符号化対象の H D R 画像から L D R 画像を生成する L D R 画像生成手段と、

生成した L D R 画像を非可逆符号化する符号化手段と、

符号化した L D R 画像を復号する復号手段と、

H D R 画像と L D R 画像に基づき、L D R 画像を H D R 画像の精度に戻すための補助データを生成する補助データ生成手段と、

H D R 画像における画素の輝度値に基づき、前記補助データ生成手段が利用する L D R 画像として、前記 L D R 画像生成手段が生成した L D R 画像、前記復号手段による復号で得た L D R 画像のいずれを用いるかを判定する判定手段と

を有することを特徴とする画像符号化装置。

【請求項 2】

前記符号化手段は、マクロブロックを単位として、色変換、D C T 変換、量子化、及び、エントロピー符号化を行う J P E G コーデック手段であって、

前記復号手段は、前記エントロピー符号化を行う前の係数に対して逆量子化、逆 D C T、逆色変換を行う手段であることを特徴とする請求項 1 に記載の画像符号化装置。

【請求項 3】

前記 L D R 画像生成手段はマクロブロックを単位に L D R 画像を生成し、

10

20

前記復号手段はマクロブロックを単位に復号し、

前記判定手段はマクロブロックが示す画像を単位に判定することを特徴とする請求項 2 に記載の画像符号化装置。

【請求項 4】

前記判定手段は、

H D R 画像における画素のうち、予め設定された輝度値より低い輝度を持つ画素が占める割合を求め、

当該割合が予め設定された閾値を超える場合、前記復号手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定し、

前記割合が予め設定された閾値以下の場合、前記 L D R 画像生成手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の画像符号化装置。

【請求項 5】

前記 L D R 画像生成手段が生成する際に利用するトーンカーブとして、予め設定された固定のトーンカーブを用いるか、符号化対象の H D R 画像内の最大輝度、最小輝度に依存したトーンカーブを用いるのかを設定する設定手段を更に有し、

前記判定手段は、

前記設定手段により符号化対象の H D R 画像内の最大輝度、最小輝度に依存したトーンカーブを用いると設定された場合、前記復号手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定し、

前記設定手段により前記固定のトーンカーブを用いると設定された場合、前記 L D R 画像生成手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定する

ことを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の画像符号化装置。

【請求項 6】

前記 L D R 画像生成手段が生成する際のトーンマッピングとして、グローバルトーンマッピング、ローカルトーンマッピングのいずれを利用するかを設定する設定手段を更に有し、

前記判定手段は、

前記設定手段によりローカルトーンマッピングを利用すると設定された場合、前記復号手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定し、

前記設定手段によりグローバルトーンマッピングを利用すると設定された場合、前記 L D R 画像生成手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定する

ことを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の画像符号化装置。

【請求項 7】

前記判定手段は、

前記符号化手段における量子化で用いる量子化ステップが予め設定された値を超える場合、前記復号手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定し、

前記符号化手段における量子化で用いる量子化ステップが前記予め設定された値以下である場合、前記 L D R 画像生成手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定することを特徴とする請求項 2 又は 3 に記載の画像符号化装置。

【請求項 8】

前記判定手段は、

前記符号化手段における色変換による色差成分について間引きが存在する場合、前記復号手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定し、

前記符号化手段における色変換による色差成分について間引きが存在しない場合、

前記 L D R 画像生成手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定することを特徴とする請求項 2 又は 3 に記載の画像符号化装置。

【請求項 9】

前記判定手段は、

前記 D C T 変換を行って得た係数と、前記復号手段による逆量子化後の係数との差分を求め、当該差分の絶対値が予め設定した値を超える個数の占める割合と、予め設定された閾値とを比較し、

前記割合が前記閾値より大きい場合、前記復号手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定し、

前記割合が前記閾値以下の場合、前記 L D R 画像生成手段で得た L D R 画像を前記補助データ生成手段により利用するものとして決定する

ことを特徴とする請求項 2 又は 3 に記載の画像符号化装置。

【請求項 10】

前記量子化で用いる量子化ステップに基づいて、前記 D C T 変換を行うマクロブロックにおける、前記割合を求める範囲を決定する手段を更に有することを特徴とする請求項 9 に記載の画像符号化装置。

【請求項 11】

H D R 画像を符号化し、L D R 画像を H D R 画像の精度に戻すための補助データを生成する画像符号化装置の制御方法であって、

符号化対象の H D R 画像から L D R 画像を生成する L D R 画像生成工程と、

生成した L D R 画像を非可逆符号化する符号化工程と、

符号化した L D R 画像を復号する復号工程と、

H D R 画像と L D R 画像に基づき、L D R 画像を H D R 画像の精度に戻すための補助データを生成する補助データ生成工程と、

H D R 画像における画素の輝度値に基づき、前記補助データ生成工程が利用する L D R 画像として、前記 L D R 画像生成工程が生成した L D R 画像、前記復号工程による復号で得た L D R 画像のいずれを用いるかを判定する判定工程と

を有することを特徴とする画像符号化装置の制御方法。

【請求項 12】

コンピュータが読み込み実行することで、前記コンピュータを、請求項 1 乃至 10 のいずれか 1 項に記載の画像符号化装置が有する各手段として機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は画像符号化技術、特に、高ダイナミックレンジ (High Dynamic Range; H D R) を持つデジタル画像 (以下、H D R 画像) の符号化技術に関するものである。

【背景技術】

【0002】

静止画像の流通のためのデータ形式の代表的なものに J P E G がある。J P E G は、デジタルカメラの出力ファイルのデータ形式として利用されていることもあって、それを利用したソフトウェアは、ウェブブラウザを始め、多数存在する。J P E G 圧縮の対象となる静止画像は、1 コンポーネント当り 8 ビット (2⁵ 6 階調) の画像データである。

【0003】

一方、人間の視覚は 1 : 10000 までのコントラスト比を認識できるため、従来の J P E G 画像の階調表現では実際の見た目と大きな差ができる。そこで、1 コンポーネント当り 8 ビットを超えるビット数 (例えば 32 ビット) で表される H D R 画像が注目されている。最近のデジタルカメラやスマートフォンでも H D R モードを搭載したものが一般的になっている。また、簡単にコンポーネント当り 32 ビットのデータを持つ H D R 画像を作成できるソフトウェアアプリケーションも増えてきている。

【0004】

10

20

30

40

50

1 コンポーネント当たり32ビットのデータを持つHDR画像を、従来のJPEGビューアで見ることができる技術が、ISO/IEC 18477-2 (JPEG XT Part 2) で規定されている。また、特許文献1にも同様の技術が開示されている。HDR画像を8 [ビット/コンポーネント] の、より小さいダイナミックレンジ画像 (Low Dynamic Range: LDR画像) にトーンマッピングし、JPEG圧縮する。そして、原画であるHDR画像とLDR画像の比率を別途HDR補助データとして作成し、HDR補助データとLDR画像のJPEG符号データを、1つのファイルに保存するという方法である。HDR補助データを保存する場所は、JPEGのアプリケーションマーカ (APP11) 内である。一般的なJPEGビューアは、サポートしないAPPマーカを無視してデコードする。そのため、たとえば、APP11を、HDR補助データを保存しているマーカとして認識できないJPEGビューアがHDR補助データをもつJPEGファイルを受け取ると、LDR画像のみをデコードし表示する。一方、APP11にHDR補助データが保存されていることを認識できるHDR画像ビューアは、LDR画像とAPP11マーカ内のHDR補助データを組み合わせて、元のHDR画像をデコードし表示できる、という仕組みである。

10

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特許第05086067号公報

【発明の概要】

【発明が解決しようとする課題】

20

【0006】

ここで、HDR補助データの作成方法について考察する。まず、ISO/IEC 18477-2 に準拠したHDR画像の符号化処理について、図1Aを参照して説明する。なお、それぞれの処理については、ISO/IEC 18477-2 に記載されているため、詳細な説明は割愛し、ここでは簡単な説明にとどめる。

【0007】

まず、LDR画像のJPEG符号データの作成までを説明する。入力としてHDR画像101が与えられると、HDR画像101はトーンマッピング部102により、1コンポーネントあたり0-255の256階調の線形色空間のLDR画像に変換される。色変換部112は、線形色空間の線形LDR画像を、sRGB色空間に変換する。そして、sRGB空間の画像データは、RGB/YCC変換部113およびサブサンプリング部114による処理を経て、JPEG圧縮部115に供給され、JPEG符号化される。

30

【0008】

次に、HDR補助データの作成の手順を説明する。Y算出部103は、HDR画像101から輝度成分 (Y成分) を算出する。一方、Y算出部105も、トーンマッピング102で作成された線形LDR画像からも輝度成分 (Y成分) が算出する。ノイズ付加部106は、Y算出部105で算出された各輝度成分にノイズを加算する。そして、Yratio算出部107は、HDRのY成分をノイズが付加されたLDRのY成分で割る (除算) ことで、輝度成分の比率 (Yratio) を算出する。Yratioマッピング部108は、算出されたYratioを、0-255の範囲にマッピングし、HDR補助データのY成分として、JPEG圧縮部109に供給する。さらに、Yratio算出部107は、算出したYratioをRGBスケール変換部104に供給する。RGBスケール変換部104は、HDR画像のRGBの各色成分を、Yratioを用いてスケール変換する。差分取得部110は、スケール変換されたHDR画像のRGB値と、トーンマッピング102から出力される線形LDR画像のRGB値との差分値を取得する。色差算出111は、この差分値によってHDR補助データの色差成分を算出し、JPEG圧縮部109に供給する。JPEG圧縮部109は、Yratioマッピング部108から入力された輝度成分と、色差算出111から入力される色差成分とを使って、HDR補助データのJPEG符号データを作成する。

40

【0009】

以上より、LDRのJPEG符号データと、HDR補助データのJPEG符号データが

50

生成される。出力部 116 は、両符号データと LDR 画像から HDR 画像へ伸長するための必要な情報と共に、1つの JPEG ファイルを作成し、JPEG XT データ 117 として出力する。

【0010】

以上が ISO/IEC 18477-2 の方法による HDR 画像を符号化処理である。HDR 補助データ作成時に使用される LDR 画像は、トーンマッピング部 102 から得られたデータである。しかし、最終的に、JPEG XT データ 117 として出力される LDR 画像は、非可逆圧縮である JPEG 圧縮部 115 から出力されるものであって、劣化後のデータである。従って、JPEG XT データ 117 を受信して、復号して得られる HDR 画像に対しては、高い精度が期待できない。

10

【0011】

このような問題に対して、特許文献 1 は、JPEG 圧縮部 115 で符号化された LDR 画像を一度復号して得られる LDR 画像を、HDR 補助データ作成のために使用する方法を開示している。図 1B が、特許文献 1 に開示された構成である。図 1A と同様の構成については同符号を付し、その説明は省略する。図 1A との違いは、LDR 画像の JPEG 圧縮部 115 の結果を、JPEG 復号部 119 によって復号する。そして、復号された LDR 画像が色変換部 118 により線形色空間の LDR 画像に変換され、その結果が Y 算出部 105 と差分取得部 110 へ供給されている。

【0012】

このように、HDR 画像の符号化側で、JPEG 圧縮された LDR 画像を JPEG 復号部 119 によってローカルデコードすることで、HDR 補助データ作成時の LDR 画像と、HDR 復号時に使用する LDR 画像の差異が解消される。したがって、JPEG XT データの受信側（符号化 HDR 画像の復号装置）では、より正確な HDR 画像を復号することができる。

20

【0013】

しかし、このように毎回ローカルデコードすることは、HDR 画像ファイル作成時の処理負荷を増大させるという問題が生じる。特に、デジタルカメラやスマートフォンなどのモバイル機器において HDR 画像を合成・圧縮するようなユースケースを考えた場合、処理負荷の軽減と高画質な HDR 画像ファイルの生成の両立が大きな問題となる。

【0014】

本発明は上記に鑑みてなされたものであり、符号化処理の負荷と、復号される HDR 画像の画質とのバランスを考慮し、より効率的に HDR 画像を符号化する技術を提供しようとするものである。

30

【課題を解決するための手段】

【0015】

この課題を解決するため、例えば本発明の画像符号化装置は以下の構成を備える。すなわち、

HDR 画像を符号化し、LDR 画像を HDR 画像の精度に戻すための補助データを生成する画像符号化装置であって、

符号化対象の HDR 画像から LDR 画像を生成する LDR 画像生成手段と、

40

生成した LDR 画像を非可逆符号化する符号化手段と、

符号化した LDR 画像を復号する復号手段と、

HDR 画像と LDR 画像に基づき、LDR 画像を HDR 画像の精度に戻すための補助データを生成する補助データ生成手段と、

HDR 画像における画素の輝度値に基づき、前記補助データ生成手段が利用する LDR 画像として、前記 LDR 画像生成手段が生成した LDR 画像、前記復号手段による復号で得た LDR 画像のいずれを用いるかを判定する判定手段とを有する。

【発明の効果】

【0016】

本発明によれば、HDR 画像を符号化する際の、LDR 画像を HDR 画像の精度に戻す

50

ための補助データを作成する際に利用するLDR画像として、ローカルデコードで得たものを利用するか否かを適応的に切り換える。この結果、符号化にかかる処理負荷と画質のいずれにも対応させることが可能となる。

【図面の簡単な説明】

【0017】

【図1A】HDR画像符号化処理のブロック図。

【図1B】HDR画像符号化処理のブロック図。

【図2】第1の実施形態におけるHDR画像符号化装置のブロック図。

【図3】第1の実施形態におけるHDR画像の圧縮フローを示す図。

【図4】第1の実施形態における線形LDR画像の作成フローを示す図。

【図5】第1の実施形態におけるHDR補助データの作成フローを示す図。

【図6】第1の実施形態におけるトーンカーブを示す図。

【図7】第1の実施形態の変形例におけるHDR画像の圧縮フローを示す図。

【図8】第1の実施形態の変形例におけるマクロブロック単位のLDR画像圧縮フローを示す図。

【図9】第1の実施形態の変形例におけるマクロブロック単位の線形LDR画像の作成フローを示す図。

【図10】第2の実施形態におけるHDR画像の圧縮フローを示す図。

【図11】第3の実施形態におけるHDR画像の圧縮フローを示す図。

【図12】第3の実施形態の変形例におけるHDR画像の圧縮フローを示す図。

【図13】第3の実施形態の変形例におけるマクロブロック単位のLDR画像圧縮フローを示す図。

【図14】第4の実施形態におけるHDR画像の圧縮フローを示す図。

【図15】第4の実施形態の変形例におけるHDR画像の圧縮フローを示す図。

【図16】第5の実施形態におけるマクロブロック単位のLDR画像圧縮フローを示す図。

【図17】第5の実施形態の変形例におけるDCT係数の比較範囲を示す図。

【図18】第5の実施形態の変形例におけるマクロブロック単位のLDR画像圧縮フローを示す図。

【発明を実施するための形態】

【0018】

以下、添付図面に従って本発明に係る実施形態を詳細に説明する。なお、実施形態では、説明を容易にするため、符号化対象のHDR画像データは1920×1080画素のRGBカラーデータとし、各コンポーネント(色)は16ビットの整数値で表されるものとして説明する。但し、RGB以外のコンポーネントの構成(例えばグレイ単色やYMC等)でも構わず、色空間の種類、コンポーネントの個数は問わない。また、1成分のビット数も16ビットに限らず、12ビットや32ビットなど、8ビットを超えるビット数であれば、ビット数に特に制限はない。また、整数値だけに限定されず、32ビットの実数値などでもよい。つまり、HDRのコントラスト比を表現できれば、入力画像の形式はどのようなものでもよい。上記の数値は、あくまで、理解を容易にする具体例を示すものと理解されたい。

【0019】

[第1の実施形態]

図2は、本実施形態におけるHDR画像の画像符号化装置のブロック図である。同図で、201はCPUであり、各構成の処理の全てに関わる。202は入力部であり、ユーザからの指示や、符号化対象のHDR画像データなどを入力する部分である。これは、キーボードやマウスなどのポインティングシステムを含む。符号化対象のHDR画像データの発生源は撮像装置とするが、入力部202にその撮像装置が接続されても良いし、符号化対象のHDR画像データを格納した記憶媒体が入力部202に接続されても構わない。場合によっては、HDR画像データを保持するサーバがネットワーク上に存在し、入力部2

10

20

30

40

50

02はそのサーバからダウンロードしても構わない。03はHDR補助データ生成部、04はLDR画像生成部である。05はJPEGコーデック部（非可逆符号化部）であり、1コンポーネント当り8bitの非圧縮画像データをJPEG符号データに変換したり、JPEG符号データを復号し非圧縮画像データに変換する。06はメモリであり、ROMやRAMを含み、処理に必要なプログラム、データ、作業領域などをCPU201に提供する。07は表示部であり、通常は液晶ディスプレイなどが用いられる。08は蓄積部であり、画像データ、プログラムなどを蓄積する部分で、通常はハードディスクなどが用いられる。また、以降のフローチャートの処理に必要な制御プログラムは、蓄積部08に格納されているか、メモリ06のROMに格納されているものとする。蓄積部08に格納されている場合は、一旦メモリ06内のRAMに読み込まれてから実行される。なお、システム構成については、上記以外にも様々な構成要素が存在するが、本実施形態の主眼ではないので、その説明は省略する。

10

【0020】

本実施形態におけるHDR補助データの生成方法は、ISO/IEC 18477-2で規定されている方法を基本とする。すなわち、HDR画像データとLDR画像データの輝度成分比（Yレシオ；Yratio）を利用する方法である。Yレシオを使って、LDR画像データをHDRのレンジに戻して、オリジナルのHDR画像データとの色差成分の差分を生成する。そして、Yレシオと色差成分を8ビット化してJPEG符号化する方法である。

【0021】

20

符号化装置に入力されたHDR画像データをLDR画像とHDR補助データに分けて、圧縮する方法について図3を用いて説明する。なお、ISO/IEC 18477-2で規定されている処理と同様の部分については、ここでの詳しい説明は割愛する。

【0022】

ステップS301にて、CPU201は、入力部202を介して符号化対象のHDR画像データを入力し、メモリ06に格納する。ステップS302にて、CPU201は、LDR画像生成部04を制御し、入力したHDR画像データに対してトーンマッピング処理を行わせる。LDR画像生成部04は生成した線形LDR画像データを、メモリ06に保存する。また、LDR画像生成部04は、線形LDR画像を作成する際に、低輝度の画素数をカウントし、保持するものとする。この低輝度画素数のカウント処理の詳細は後述する。

30

【0023】

ステップS303にて、CPU201は、上記で得た低輝度画素数から、低輝度画素数の割合RLPを算出する。すなわち、画像全体の低輝度画素数をC、入力されたHDR画像の幅がw画素、高さがh画素とすると、 $RLP = C / (w \times h)$ として算出できる（実施形態では、 $w = 1920$ 、 $h = 1080$ ）。

【0024】

ステップS304にて、CPU201は、算出して得られたRLPと、予め決められた閾値 Th_RLP を比較する。RLPが閾値 Th_RLP より大きければステップS305へ処理を進め、そうでなければステップS313へ処理を進める。実施形態では、閾値 Th_RLP は“0.5”とする。すなわち、低輝度画素が画像全体の半分以上を占める場合には、符号化対象のHDR画像は低輝度画素の割合が大きい画像として判断し、ステップS305へ進む。ステップS305およびステップS313では、HDR補助データ作成時にローカルデコードしたLDR画像を使用するかどうかを示す情報を、メモリ06に予め確保したフラグFlagに設定する。具体的には、ステップS305にて、CPU201は、HDR補助データ作成時にローカルデコードしたLDR画像を使用することを示す値“1”をフラグFlagに設定する。ステップS313にて、CPU201は、ステップS302で作成した線形LDR画像を用いてHDR補助データを作成することを示す値“0”をフラグFlagに設定する。従って、S313に進んだ場合、LDR画像のローカルデコードは行わないことになる。

40

50

【 0 0 2 5 】

次に、LDR画像をローカルデコードする場合のステップS305以降の処理を説明する。ステップS305にて、CPU201はフラグFlagを“1”に設定すると、処理をステップS306に進める。ステップS306にて、CPU201は、LDR画像の符号化条件として、サブサンプリング情報および量子化ステップに関する情報を取得する。次に、ステップS307にて、CPU201は線形LDR画像の色変換を行う。すなわち、CPU201は、線形色空間のLDR画像にガンマ処理を施し、sRGB色空間の画像データに変換し、その後、RGB/YCC変換を行う。また、CPU201は、ステップS306において取得した情報によって、サブサンプリングが必要な場合は、色差成分Cb, Crのサブサンプリングを行う。これらの処理については、一般的な処理であり、ISO/IEC 18477-2にも記載されているので、ここでの詳細な説明は割愛する。ステップS308にて、CPU201は、JPEGコーデック部205を制御し、ステップS307で得られたLDR画像のYCCデータをマクロブロックに分割させ、DCT変換及び量子化処理を行わせる。これらの処理についても一般的な処理であるため、ここでの詳細な説明を割愛する。ステップS309にて、CPU201は、JPEGコーデック部205を制御し、量子化されたDCT係数を、JPEG (ISO/IEC 10918) に規定された方法でエントロピー符号化する。JPEGコーデック部205は、生成した符号化データを、一旦、メモリ206に格納する。

10

【 0 0 2 6 】

次に、ステップS310にて、CPU201は、フラグFlagの値が“1”であるかどうかを判定する。すなわち、ステップS304でローカルデコードしたLDR画像を使用すると判定された場合、Flagの値が1であるので、CPU201は処理をステップS311に進める。また、Flagの値が0である場合には、CPU201はステップS317に処理を進める。ここでの説明では、ステップS305を経ているので、ステップS311に処理が進むものとして説明する。

20

【 0 0 2 7 】

ステップS311にて、CPU201は、JPEGコーデック部206を制御し、ステップS308で得られた量子化後のDCT係数を逆量子化、逆DCTすることで、ローカルデコード後のLDR画像のYCCデータを取得する。なお、ローカルデコードでは、符号化データから開始してしても良いが、JPEGでのエントロピー（ハフマン符号化）そのものは可逆であるので、途中の量子化後の係数からローカルデコードを行っている。本処理についても、DCT/量子化と同様に詳細な説明は割愛する。ステップS312にて、CPU201は、ステップS311で取得したYCCデータに逆色変換をし、線形LDR画像を生成し、メモリ206に保存し、処理をステップS314に進める。すなわち、CPU201は、YCC/RGB変換によってローカルデコードしたRGB画像データを取得し、さらに、逆ガンマ処理を行うことで、ローカルデコードした線形RGB画像データを取得する。なお、メモリ206には、既にステップS302で作成した線形LDR画像が保存されているが、そのデータをローカルデコードした結果で上書きすることになる。ステップS314にて、CPU201は、HDR補助データ生成部203を制御し、ステップS312で得られたローカルデコードの線形RGB画像を入力として、LDRからHDRの精度（8ビットから16ビット）に戻すためのHDR補助データを作成させる。この処理の詳細は、図5を用いて後述する。次に、ステップS315にて、CPU201は、ステップS314で作成したHDR補助データを、ISO/IEC 18477-2記載の方法で出力する。そして、ステップS316にて、フラグFlagが“1”（ローカルデコードを行うことを示している）であるかどうかを判定する。フラグFlagの値が“1”であれば、既にLDR画像のJPEG符号データがメモリ206に作成されているものと判断し、CPU201は処理をステップS317に進める。また、フラグFlagの値が“0”であれば、処理をステップS306に進め、LDR画像のJPEG符号データの作成処理に進む。ステップS317にて、CPU201は、LDR画像の符号化データを、ISO/IEC 18477-1記載の方法で出力し、このフローを終了する。

30

40

50

【 0 0 2 8 】

次に、ステップ S 3 0 4 にて、符号化対象の H D R 画像が、低輝度画素の割合が小さい画像として判断された場合を説明する。

【 0 0 2 9 】

ステップ S 3 1 3 にて、C P U 2 0 1 は、フラグ F l a g に “ 0 ” を設定、すなわち、ローカルデコードしないとして設定する。そして、ステップ S 3 1 4、3 1 5 にて、C P U 2 0 1 は、H D R 補助データ生成部 2 0 3 を制御し、ステップ S 3 0 2 で得られた線形 L D R 画像を入力として、H D R 補助データを作成、出力させる。次にステップ S 3 1 6 にて、C P U 2 0 1 は、フラグ F l a g の値が “ 1 ” であるか否かを判定し、“ 1 ” であればステップ S 3 1 7、“ 0 ” であればステップ S 3 0 6 に処理を進める。ここでの説明では、ステップ S 3 1 3 からの説明であるので、処理はステップ S 3 0 6 へ進み、C P U 2 0 1 は、L D R 画像の符号化処理を行う。ステップ S 3 0 6 からステップ S 3 0 9 までは、上述の説明と同じである。ステップ S 3 1 0 では、再びフラグ F l a g を参照する。今回は、ステップ S 3 1 3 の処理を経ているため、フラグ F l a g は “ 0 ” である。したがって、ローカルデコードをする必要がないため、ステップ S 3 1 7 へ進み、L D R 画像の符号化データを出力し、このフローを終了する。

10

【 0 0 3 0 】

次に、ステップ S 3 0 2 における L D R 画像生成部 2 0 4 による線形 L D R 画像の作成処理を、図 4 を参照して説明する。

【 0 0 3 1 】

20

ステップ S 4 0 1 にて、L D R 画像生成部 2 0 4 は、符号化対象の H D R 画像データから L D R 画像データを生成するためのトーンカーブを取得する。トーンカーブとは、H D R 画像を所定のダイナミックレンジに圧縮するトーンマッピング処理に利用する計算式やテーブルである。トーンマッピング処理は、人間の視覚特性や表示デバイス特性を考慮した手法など様々な手法が提案されている。本実施形態では、各色に対して、図 6 (a) のようなトーンカーブを描く数式を取得するものとする。他にもルックアップテーブルや、図 6 (a) のトーンカーブ上のサンプリングされた数点の値など、トーンカーブを特定できる情報であれば、どのようなものでも構わない。

【 0 0 3 2 】

ステップ S 4 0 2 にて、L D R 画像生成部 2 0 4 は、画像の幅 w と高さ h を取得する。本実施形態の場合、幅 $w = 1920$ [画素]、高さ $h = 1080$ [画素] である。ステップ S 4 0 3 にて、L D R 画像生成部 2 0 4 は、低輝度の画素値を数えるためのカウンタ C に、ゼロを代入して初期化する。そして、ステップ S 4 0 4 にて、L D R 画像生成部 2 0 4 は、画像の高さ方向をカウントするための変数 y に、ゼロを代入して初期化する。また、ステップ S 4 0 5 にて、L D R 画像生成部 2 0 4 は、画像の幅方向をカウントするための変数 x にゼロを代入して初期化する。なお、本実施形態では、画像の左上隅 (0 , 0) からラスト順に以下の処理を行う。

30

【 0 0 3 3 】

ステップ S 4 0 6 では、L D R 画像生成部 2 0 4 は、H D R 画像内の (x , y) で表わされる画素位置の画素値 P_H を取得する。本実施形態では、画素値 P_H はそれぞれ 16 ビットの R , G , B の値で構成される。ステップ S 4 0 7 では、ステップ S 4 0 1 で取得したトーンカーブを使って、画素値 P_H を L D R 画像の画素値 P_L (8 ビット) にマッピングする。そして、L D R 画像生成部 2 0 4 は、メモリ 2 0 6 に確保した、線形 L D R 画像用の領域内の画素位置 (x , y) に、画素値 P_L を保存する。次に、ステップ S 4 0 8 にて、L D R 画像生成部 2 0 4 は、画素値 P_H の値から、画素位置 (x , y) の輝度値 Y_{hdr} を算出する。本実施形態では、輝度 Y_{hdr} は 8 ビットの J P E G 圧縮と同様に、I S O / I E C 18477 - 1 で規定された変換式で算出される。ステップ S 4 0 9 にて、L D R 画像生成部 2 0 4 は、ステップ S 4 0 7 で算出した Y_{hdr} と予め決められた輝度の閾値 Th_Y とを比較する。輝度値 Y_{hdr} がしきい値 Th_Y より小さければステップ S 4 1 0 へ、そうでなければステップ S 4 1 1 へ進む。本実施形態では、図

40

50

6に示すように、マッピングしたLDRの画素値が“96”となる値“4550”を閾値 Th_Y とする。したがって、輝度値 Y_hdr が“4550”よりも小さい値であれば、処理はステップS410に進み、LDR画像生成部204は低輝度画素を数えるカウンタCに“1”を加算する。また、輝度値 Y_hdr が4550以上であれば、処理はステップS410の処理は行わず、ステップS411に処理が進む。このステップS411にて、LDR画像生成部204は、画素位置(x, y)に対してトーンマップ処理および低輝度画素の判定が終了したので、画像の水平方向のカウンタxに“1”を加算する。そして、ステップS412にて、画像の横方向のカウンタxが画像幅wと同じ値であるか判定する。同じ値であれば、この画像の右端まで処理したものと判定し、ステップS413へ進む。そうでなければ、まだ右端に到達していないものと判定し、ステップS406へ処理を戻す。ステップS413にて、LDR画像生成部204は、画像の高さ方向の変数yに“1”を加算する。そして、ステップS414にて、LDR画像生成部204は、変数yと画像の高さhを比較し、同じであれば、この画像のすべての画素の処理を終えたものとして、このフローを終了する。異なる場合には、ステップS405に処理を戻して、残りの画素の処理を続ける。

10

【0034】

以上の処理により、HDR画像と同じ大きさの線形色空間のLDR画像(線形LDR画像)と、HDR画像の輝度成分 Y_hdr 、および低輝度画素数Cを得ることができる。なお、線形LDR画像は1コンポーネント当り8ビットの整数値(0~255)の画像データである。

20

【0035】

次に、図5を参照して、ステップS314における、HDR補助データ生成部203によるHDR補助データ作成処理を説明する。HDR補助データの作成方法については、ISO/IEC 18477-2で記載されている方法をそのまま使用すれば良いので、詳細な説明は割愛する。

【0036】

まず、ステップS501にて、HDR補助データ生成部203は、ステップS301で取得したHDR画像の輝度値 Y_hdr を取得する。輝度値 Y_hdr は、ステップS302の線形LDR画像作成時に既に計算されているので、その値を保存していれば、ここで計算する必要はない。ステップS502にて、HDR補助データ生成部203は、メモリ206の線形LDR画像用の領域から線形LDR画像データを取得する。フラグFlagに“0”が設定されている場合は、メモリ203にはステップS302で作成された線形LDR画像が保存されている。一方、フラグFlagが“1”に設定されている場合は、メモリ203には、ステップS302で作成された線形LDR画像が、ステップS312の処理によってローカルデコード結果の線形LDR画像によって上書きされた画像が保存されている。ステップS503にて、HDR補助データ生成部203は、ステップS502で取得した線形LDR画像の各画素について輝度値 Y_ldr を算出する。輝度値の計算方法は、ISO/IEC 18477-2で規定された変換式を用いる。ステップS504にて、HDR補助データ生成部203は、ISO/IEC 18477-2で規定されている方法に従って、ステップS501で取得した Y_hdr とステップS503で取得した Y_ldr を用いて、次式に従い全画素のYレシオ(Y_ratio)を計算する。

30

$$Y_ratio(x,y)=Y_hdr(x,y)/(Y_ldr(x,y)+nf)$$

ここで、nfは、ノイズ付加部106により付加されるノイズである。これにより、ゼロ割を防ぐことができる。

40

【0037】

ステップS505にて、HDR補助データ生成部203は、ISO/IEC 18477-2で規定されている方法にしたがって、HDR画像と線形LDR画像の各画素位置毎にRGB値の差分 $DifRGB$ を算出する。ステップS506にて、HDR補助データ生成部203は、ステップS505で算出した画素位置毎の $DifRGB$ を、ステップS5

50

04で算出した同一画素位置の Y_ratio で割ることでスケール変換する。そして、HDR補助データ生成部203は、ISO/IEC 18477-2で規定されている方法に従って、各画素位置に対応するHDR補助データの色差成分 CC_hdr を算出する。ステップS507にて、HDR補助データ生成部203は、 CC_hdr をステップS503で算出した Y_ldr で正規化し、 CC_norm を取得する。ステップS508にて、HDR補助データ生成部203は、ステップS504で取得した Y_ratio の対数を取り、 Y_log を算出する。ステップS509にて、HDR補助データ生成部203は、すべての画素の CC_norm の値から、最大値 Max_CC (Max_Cb と Max_Cr)と最小値 Min_CC (Min_Cb と Min_Cr)を取得する。そして、HDR補助データ生成部203は、 Max_CC と Min_CC の範囲で CC_norm を量子化し、 CC_norm を8ビット化した CC_xt を取得する。ステップS511にて、HDR補助データ生成部203は、ステップS508で取得した Y_log について、全ての画素の最大値 Max_log と最小値 Min_log を取得する。そして、HDR補助データ生成部203は、 Max_log と Min_log の範囲で Y_log を量子化し、8ビット化した Y_xt を取得する。そして、ステップS513において、 Y_xt と CC_xt をJPEG方式で符号化し、このフローを終了する。

10

【0038】

本実施形態では、LDR画像において輝度値が96以下となる画素数が、画像全体の半数を超える場合に、LDR画像をローカルデコードした結果を使ってHDR補助データを作成する。つまり、低輝度画素数が非常に多い場合のみ処理負荷が高いローカルデコード処理を実施し、復号時のHDR画像の画質向上を図る。これは人間の視覚特性上、低輝度に対する感度が高いためである。そのため、処理負荷が高くても、復号時のHDR画像の画質を向上する効果大きい。一方、低輝度画素の割合が多くない画像の場合には、復号時のHDR画像の画質を向上させる効果が少ないため、ローカルデコード処理を行わず、符号化にかかる処理負荷を低減させることができる。したがって、本実施形態によれば、復号時の画質向上の効果が大きい時のみ、符号化時の処理負荷が大きくなるローカルデコード処理を実行することが容易になる。

20

【0039】

[第1の実施形態の変形例]

上記の第1の実施形態では、符号化対象のHDR画像全体に対する低輝度画素値が占める割合を用いて、ローカルデコード処理の有無を決定した。図3のステップS302で、画像全面分、すなわち 1920×1080 画素の線形LDR画像を作成すると共に、低輝度画素値をカウントした。そのため、線形LDR画像作成後に、ローカルデコードの有無を判定できた。しかし、画像全体の情報が必要になるため、並列化処理ができず、メモリ使用量も大きくなってしまふ。

30

【0040】

たとえば、マクロブロック(8×8 画素)の高さと同じ8ラインずつに画像を分割し、HDR画像の圧縮フローを実行し、並列処理で高速化することを考える。LDR画像の作成を終えた8ライン分の符号化を行うと同時に、次の8ラインのLDR画像の作成を並列して実行する。しかし、画像全体における低輝度画素の割合が分からないと、ローカルデコードの有無が決まらない。そのため、HDR補助データの作成は、ステップS302の処理が終わるまで始めることができず、並列処理できない。さらに、ローカルデコードの有無が決定するまで、ステップS308で得られるLDR画像の量子化値をすべて保存しておく必要がある。そのため、LDR画像のすべての量子化値を保持しなければならない。あるいは、LDR画像の符号データだけを保持しておき、ローカルデコードが必要になった時点で復号し、線形LDR画像を作成することもできる。この場合、LDR画像のメモリ使用量は符号データ分と、8ライン分のメモリのみで良いが、エン트로ピー符号化を復号する時間が必要になり、処理全体が遅くなる。

40

【0041】

上述の問題を解決する方法として、本変形例では、マクロブロック単位でローカルデコ

50

ードの有無を切り替えるものである。以下、本変形例の具体的な処理を、図7のフローを使って説明する。なお、本変形例において、第1の実施形態と同じ動作の処理には、同符号を付与し、その詳細な説明は割愛する。また、以下の説明では、HDR補助データ生成部203、LDR画像生成部204、JPEGコーデック部205等の各種処理部が、CPU201によって実行される例を説明する。

【0042】

CPU201は、ステップS306にて、LDR画像の符号化条件を取得する。次に、ステップS701にて、CPU201は、HDR画像のサイズ、すなわち幅 w と高さ h を取得する。本実施形態の場合、幅 $w = 1920$ 画素、高さ $h = 1080$ 画素である。ステップS702にて、CPU201は、処理すべきマクロブロック数 N を算出する。本実施形態では、画像の幅 w が1920画素であるので、 $N = 1920 \div 8 = 240$ と計算できる。続くステップS703にて、CPU201は、HDR画像データを先頭から順に8ライン取得する。8ラインとは、マクロブロックの高さと一致するライン数である。ステップS704にて、CPU201は、処理済みのマクロブロックをカウントするためのカウンタ M にゼロを代入し初期化する。ステップS705にて、CPU201は、マクロブロック単位でLDR画像の圧縮処理を行う。この処理については、後に図8を使って説明する。ステップS706にて、CPU201は、HDR補助データ作成に必要なマクロブロックの CC_norm と Y_log を作成する。この処理は、第1の実施形態で説明した図5のステップS501からステップS508を、マクロブロックに対して適用した処理である。第1の実施形態で説明済みのため、本変形例における処理フローとその説明は割愛する。次にステップS707にて、CPU201は、マクロブロックのカウンタ M に“1”を加算する。ステップS708にて、CPU201は、カウンタ M とステップS703で算出した処理すべきマクロブロック数 N を比較する。 $M = N$ であれば、現時点で入力された8ライン分のHDR画像の処理が終了したと判断し、CPU201は処理をステップS709へ進める。そうでなければ、CPU201はステップS705へ処理を戻す。ステップS709にて、CPU201は、HDR画像の最後のラインまで処理が終わったかどうかを判定する。本実施形態では、ステップS703で入力されたライン数を加算しておき、ステップS701で取得したHDR画像の高さと同じになった場合、最後のラインまで終了したと判定する。最後のラインまで終了した場合、CPU201は処理をステップS710へ進める。そうでなければ、CPU201は、ステップS703へ処理を戻し、次の8ラインを取得する。

【0043】

ステップS710にて、CPU201は、ステップS706で得られた CC_norm と Y_log を使って、HDR補助データの圧縮処理を行う。この処理は、第1の実施形態で説明した図5のフローチャートにおけるステップS509からステップS513までの処理と同じであるので、その説明は割愛する。なお、図5のフローのステップS509以降の処理において、画像全体における最大値および最小値を取得する必要がある。そのため、LDR画像の作成を8ラインずつ実施する本変形例においては、ステップS508までと、それ以降で処理を分割する必要がある。次にステップS315にて、CPU201は、ステップS710で作成されたHDR補助データを出力する。そして、ステップS317において、CPU201は、ステップS705で作成されたLDRの圧縮データをLDR情報として出力する。ステップS315とステップS317の処理は第1の実施形態と同じであるので、ここでの説明は省略する。

【0044】

次に、ステップS705のマクロブロック単位のLDR画像圧縮処理を図8のフローチャートに従って説明する。

【0045】

ステップS801にて、CPU201は、マクロブロック単位で線形LDR画像を作成する。この方法については、後に図9を用いて説明する。ステップS802にて、CPU201は、ステップS801で線形LDR画像を作成する際にカウントした低輝度画素の

10

20

30

40

50

数から、着目マクロブロックの低輝度画素の割合 RLP_m を算出する。マクロブロックのサイズは 8×8 画素である。故に、ステップ $S801$ で得られる低輝度画素数を C とすると、 $RLP_m = C / 64$ となる。ステップ $S803$ にて、 $CPU201$ は、 RLP_m と、あらかじめ決められた閾値 Th_RLP を比較する。 RLP_m が閾値 Th_RLP より大きければ、 $CPU201$ は処理をステップ $S305$ へ進める。本変形例では、閾値 Th_RLP は 0.5 とする。したがって、本変形例の場合、1つのマクロブロックに低輝度の画素が 33 以上あれば、処理はステップ $S305$ へ進むことになる。ステップ $S305$ にて、 $CPU201$ は、ローカルデコードを行うことを示すためフラグ $Flag$ に “1” を代入する。一方、 RLP_m が閾値以下となって、 $S313$ に処理が進んだ場合、 $CPU201$ は、ローカルデコードを行わないことを示すためフラグ $Flag$ に “0” を代入する。ステップ $S307$ からステップ $S310$ までは第1の実施形態と同様の動作をする。ステップ $S310$ にて、 $CPU201$ は、フラグ $Flag$ が “1” であるか否かを判定する。フラグ $Flag$ が “1” の場合、 $CPU201$ は処理をステップ $S311$ へ進め、そうでなければ、本フローを終了する。ステップ $S311$ およびステップ $S312$ は第1の実施形態と同様の動作をする。ただし、 $S311$ 、 $S312$ における処理対象は、着目マクロブロックとなる。ステップ $S804$ では、ステップ $S312$ で得られたローカルデコード後の画像データで、ステップ $S801$ で作成した着目マクロブロックの線形 LDR 画像を上書きし、このフローを終了する。

【0046】

次に、ステップ $S801$ のマクロブロック単位の線形 LDR 画像の作成処理を図9のフローを参照して説明する。

【0047】

基本的には、第1の実施形態の図4で画像全体に対して行った処理を、本変形例に合わせて1つのマクロブロックに対して行うことになる。

【0048】

ステップ $S401$ にて、 $CPU201$ は、トーンマッピング処理に用いるトーンカーブを取得し、ステップ $S403$ に処理を進める。ステップ $S403$ からステップ $S411$ までは、第1の実施形態と同様の処理である。ステップ $S901$ にて、 $CPU201$ は、変数 x とマクロブロックの幅 $= 8$ を比較する。変数 $x = 8$ であれば、 $CPU201$ は、このマクロブロックの右端まで処理を終えたものと判断し、処理をステップ $S413$ へ進める。また、変数 x がマクロブロック幅に達していない場合、 $CPU201$ は着目画素の右隣の画素を処理するためにステップ $S406$ へ処理を戻す。ステップ $S413$ にて、 $CPU201$ は、マクロブロックの高さ方向の変数 y に “1” を加算する。ステップ $S902$ にて、 $CPU201$ は、変数 y とマクロブロックの高さ $= 8$ と比較する。 y が 8 であればこの処理を終了し、そうでなければ次のラインの先頭から処理を進めるために、ステップ $S405$ へ処理を戻す。

【0049】

本変形例の処理によれば、マクロブロック単位でローカルデコードの有無を切り替えることができる。そのため、マクロブロック境界にアーチファクトが発生するリスクは高くなる。しかし、 LDR の符号化に要するメモリ使用量を削減でき、並列処理による高速化も行いやすいという効果が期待できる。

【0050】

なお、本実施形態では、マクロブロックと同じ高さである8ラインずつデータを読み込み処理したが、マクロブロックの高さ ($= 8$) の倍数であれば、何ラインであっても構わない。

【0051】

[第2の実施形態]

第1の実施形態では、図6(a)のようにHDR画像のデータがとり得る範囲すべてをカバーするトーンカーブを使用した。このように画像によらず一意に決まるトーンカーブを適用するトーンマッピング処理は、計算コストが小さく、リアルタイム性が要求される

10

20

30

40

50

場合には好まれる。しかし、入力されるH D R画像がこの範囲をすべて使用していないことがある。実際、個々のH D R画像のダイナミックレンジは様々であり、取り得るデータ範囲の一部しか使っていない画像がほとんどである。たとえば、第1の実施形態では、入力されたH D R画像の画素値が“ 4 5 5 0 ”より小さい値であれば低輝度画素としてカウントしたが、入力H D R画像の最小画素値が“ 4 5 5 0 ”であると、ローカルデコードが必要と判断されることが全くない。

【 0 0 5 2 】

一方、符号化対処のH D R画像のダイナミックレンジに依存したトーンカーブを用いてトーンマッピングを行うことも考えられる。たとえば、入力H D R画像の画素値の最小値が“ 4 5 5 0 ”で、最大値が“ 5 0 5 3 0 ”である場合、図6 (b) のようなトーンカーブを生成し、L D R画像を作成する。このように個々の符号化対象のH D R画像のダイナミックレンジに合わせたトーンカーブを使用する方が、計算コストはかかるが、より綺麗（高品位）なL D R画像を復号できる符号データを作成することができる。つまり、計算コストよりも画質を重視していると判断できる。

【 0 0 5 3 】

本第2の実施形態では、H D R画像の画素値の取り得る全範囲に対応する固定トーンカーブ（図6 (a) ）を用いて符号化するのか、符号化対象のH D R画像に依存したトーンカーブを生成した上で符号化するのかを、ユーザが入力部2 0 2を介して設定する例を説明する。そして、その設定に応じてローカルデコードの有無を切り換える。

【 0 0 5 4 】

なお、本第2の実施形態におけるL D R画像生成部2 0 4は、固定トーンカーブを用いることが設定された場合には、図6 (a) に示すトーンカーブを用いてL D R画像を生成する。一方、符号化対象のH D R画像に依存するトーンカーブを用いることが設定された場合、L D R画像生成部2 0 4は、符号化対象のH D R画像の画素の最大値と最小値とを求め、その範囲の出力値を0乃至2 5 5とする曲線を設定した上で、L D R画像を生成するものとする。

【 0 0 5 5 】

以下、本第2の実施形態の処理を図1 0のフローチャートに従って説明する。なお、本第2の実施形態において、第1の実施形態と同様の処理には、同じ番号を付与し、その説明は省略する。

【 0 0 5 6 】

ステップS 3 0 1にて、C P U 2 0 1はH D R画像のデータを取得し、メモリ2 0 6に格納する。次に、ステップS 3 0 2にて、C P U 2 0 1はL D R画像生成部2 0 4を制御し、線形L D R画像を作成させる。生成された線形L D R画像はメモリ2 0 6に格納される。なお、L D R画像の生成手順は先に説明した通りである。次に、ステップS 1 0 0 1にて、C P U 2 0 1は、ステップS 3 0 2で使用したトーンカーブが、符号化対象のH D R画像に依存したトーンカーブであったかを判定する。この判定は、ユーザの設定から判定しても良い。

【 0 0 5 7 】

符号化対象のH D R画像に依存したトーンカーブを用いたと判定された場合、C P U 2 0 1は画質重視であると判断し、処理をステップS 3 0 5へ進める。ステップS 3 0 5にて、C P U 2 0 1は、ローカルデコードしたL D R画像をH D R補助データ作成時に使用するようにするため、フラグF l a gに“ 1 ”を設定する。一方、ステップS 3 0 2で使用したトーンカーブが、符号化対象のH D R画像に依存しない、固定トーンカーブであった場合、C P U 2 0 1は、ステップS 1 0 0 1からステップS 3 1 3に処理を進める。ステップS 3 1 3にて、C P U 2 0 1は、L D R画像のローカルデコードをしないことを示すため、フラグF l a gに“ 0 ”を設定する。ステップS 3 0 6以降は、第1の実施形態と同様の動作であるので、ここでの説明は割愛する。

【 0 0 5 8 】

以上の本第2の実施形態によれば、画質優先か否かに応じて、適用するトーンカーブを

10

20

30

40

50

決定し、LDR画像のローカルデコードをするか、しないかを切り分けることができる。

【0059】

〔第2の実施形態の変形例〕

上記第2の実施形態では、符号化対象のHDR画像に依存したトーンカーブを使用するか、固定トーンカーブを使用するかによって、ローカルデコードをする／しないを切り替えた。しかし、符号化対象のHDR画像の輝度値の最小値と最大値から、ローカルデコードをする／しないを切り替えてもよい。つまり、入力HDR画像の最大輝度値が予め設定された閾値より小さい場合には、低輝度の画像であると判定し、ローカルデコードを行う。逆に、入力HDR画像の最小輝度値が予め設定された閾値以上の場合には、ローカルデコードを行わない。本変形例の場合、第1の実施形態のように低輝度画素の割合を計算するわけではない。そのため、ほとんどの画素の輝度が低いにも関わらず、ノイズなどで1画素でも最大輝度値が大きくなると、ローカルデコードされない。しかし、ローカルデコードの有無をより簡易に判定できる。さらに、入力HDR画像のダイナミックレンジに応じたトーンカーブを使用しながらも、HDR補助データ作成にかかる計算コストを小さく抑えることができる。

10

【0060】

〔第3の実施形態〕

トーンマッピング手法としては、大きく2つの手法がある。1つは、第1、第2の実施形態で示したように、画像全体に一樣な輝度変換を行うグローバルトーンマッピングである。もうひとつは、画像の局所領域毎に、異なった輝度変換を行うローカルトーンマッピングである。

20

【0061】

グローバルトーンマッピングは、画像全体で一樣な変換関数を用いるため、画像中のテクスチャ情報の視認性が低下する。一方、ローカルトーンマッピングは、グローバルトーンマッピングに比べて計算コストが高い。しかし、局所領域におけるコントラストを維持することが可能になり、テクスチャ情報の視認性低下を防ぐことができる。

【0062】

この2つのトーンマッピング手法を使い分ける場合に、ローカルデコードをする／しないを切り替える方法について、図11を用いて説明する。なお、トーンマッピングを行う場合のグローバルトーンマッピング、ローカルトーンマッピングのいずれを用いるのかは、入力部202によりユーザが設定するものとする。

30

【0063】

図11において、図10と同じ処理には同じ番号を付与し、その部分の説明は省略する。

【0064】

ステップS301にて、CPU201は、符号化対象のHDR画像のデータを取得し、メモリ206に格納する。次に、ステップS302にて、CPU201は、LDR画像生成部204を制御し、メモリ206に格納されたHDR画像から線形LDR画像を作成させ、作成した線形LDR画像をメモリ206に格納させる。次に、ステップS1101にて、ステップS302で使用したトーンマッピングの手法が、ローカルトーンマッピングだったかどうかを判定する。この判定は、ユーザからの設定から行うものとするが、例えば、1つのHDR画像に対して、複数のトーンカーブを使用した場合に、ローカルトーンマッピングであったと判定しても良い。ローカルトーンマッピングである場合は、CPU201は、画質重視であると判断し、処理をステップS305へ進める。また、ローカルトーンマッピングではない場合には、CPU201は計算コスト重視と判断し、処理をステップS313に進める。以下の処理は第2の実施形態と同様である。

40

【0065】

本第3の実施形態よれば、トーンマッピングの手法により画質重視か、計算コスト重視かを容易に判断でき、LDR画像のローカルデコードをするか、しないかを簡単に切り分けることができる。

50

【 0 0 6 6 】

[第 3 の実施形態の変形例]

上記第 3 の実施形態では、ローカルトーンマッピングが選択された場合に、LDR 画像の全領域についてローカルデコードした結果を用いた。

【 0 0 6 7 】

本変形例では、すべてのマクロブロックに対してローカルトーンマッピングを実施するが、マクロブロック毎にローカルデコードをする / しないを切り替える場合について考える。この場合、人間の視覚特性は、低輝度に対する感度が大きいことを考慮すると、暗い領域として分類されたマクロブロックのみ、ローカルデコードすることで、計算コストを軽減することができる。このような処理について、図 1 2 を用いて説明する。なお、第 1 乃至第 3 の実施形態と同じ処理には同じ番号を付与し、詳細な説明は割愛する。

10

【 0 0 6 8 】

ステップ S 3 0 1 にて、CPU 2 0 1 は、符号化対象の HDR 画像データを取得し、メモリ 2 0 6 に格納する。次いで、ステップ S 1 2 0 1 にて、CPU 2 0 1 は、HDR 画像を、暗い領域と明るい領域の 2 つに分類し、領域分割する。領域分割の方法は、明るい領域と暗い領域に 2 分できるのであれば、どのような方法を用いても構わない。

【 0 0 6 9 】

本変形例では、各画素の輝度値を見て、16 ビットで取り得る輝度範囲 (0 ~ 6 5 5 3 5) の真ん中 (4 3 6 7 8) 未満であれば暗い領域の画素として分類する。それ以外であれば明るい領域の画素として分類する。その後、例えば 1 2 8 画素 × 1 2 8 画素未満の領域は隣接領域と統合し、領域分割を終了するものとする。

20

【 0 0 7 0 】

ステップ S 7 0 2 にて、CPU 2 0 1 は、画像全体に含まれるマクロブロック数 N を算出する。実施形態では、画像サイズが 1 9 2 0 × 1 0 8 0 画素であるので、 $N = (1 9 2 0 / 8) \times (1 0 8 0 / 8) = 3 2 4 0 0$ と算出できる。ステップ S 7 0 4 にて、CPU 2 0 1 は、マクロブロック用のカウンタ M にゼロを代入して初期化する。ステップ S 1 2 0 2 にて、CPU 2 0 1 は、マクロブロック単位の LDR 画像圧縮を行う。この処理については、後に図 1 3 を用いて説明する。

【 0 0 7 1 】

ステップ S 7 0 7 にて、CPU 2 0 1 は、マクロブロック用カウンタ M に “ 1 ” を加算する。ステップ S 7 0 8 にて、CPU 2 0 1 は、マクロブロック用カウンタ M の値と、マクロブロック数 N とを比較する。同じであれば、CPU 2 0 1 は、すべてのマクロブロックについて LDR 画像圧縮を終えたものと判断し、処理をステップ S 3 1 4 へ進める。そうでなければ、CPU 2 0 1 は処理をステップ S 1 2 0 2 へ戻す。ステップ S 3 1 4、ステップ S 3 1 5、ステップ S 3 1 7 は第 1 の実施形態と同様の動作を行い、このフローを終了する。

30

【 0 0 7 2 】

次に、ステップ S 1 2 0 2 の処理を、図 1 3 を参照して説明する。なお、図 1 3 において、図 9 と同様の動作には同じ番号を付与し、その説明は省略する。

【 0 0 7 3 】

ステップ S 8 0 1 にて、CPU 2 0 1 は、マクロブロック単位の線形 LDR 画像を作成する。ステップ S 1 3 0 1 にて、CPU 2 0 1 は、現在処理中のマクロブロック内に、ステップ S 1 2 0 2 の結果、暗い領域として判定された画素が 3 2 画素以上 (半数以上) 含まれているかどうか判定する。32 画素以上が暗い領域であれば、CPU 2 0 1 はステップ S 3 0 5 へ処理を進め、このマクロブロックはローカルデコードするものとして、フラグ F l a g に “ 1 ” を代入する。一方、ステップ S 3 1 3 に処理が進んだ場合、CPU 2 0 1 は、ローカルデコードしないものとするため、フラグ F l a g に “ 0 ” を代入する。ステップ S 3 0 7 からステップ S 8 0 4 は、第 1 の実施形態およびその変形例で説明したのと同様の処理を行い、このフローを終了する。

40

【 0 0 7 4 】

50

本変形例によれば、低輝度領域に分類されるマクロブロックのみローカルデコードを実施する。そのため、H D R画像デコード時に処理が切り替わったマクロブロックの境界部分にアーチファクトが出る可能性はある。しかし、ローカルトーンマッピングを行う場合であっても、ローカルデコードするマクロブロックを限定することで、H D R補助データ作成にかかる計算コストを削減することができる。

【 0 0 7 5 】

なお、本変形例では、すべてのマクロブロックに対してローカルトーンマッピングする方法について説明したが、第 3 の実施形態のように、グローバルトーンマッピングとローカルトーンマッピングのどちらかを選択する場合に適用してもよい。

【 0 0 7 6 】

10

[第 4 の実施形態]

ローカルデコードした方がよい場合は、符号化対象のH D R画像をトーンマッピングして得られるL D R画像と、最終的にJ P E G X Tファイルに書き込まれた符号化データを復号して得られる復号後L D R画像の差異が大きい場合である。L D R画像の量子化ステップが大きいほど、量子化による誤差が大きくなるので、上記の差異が大きくなると考えてよい。したがって、本第 4 の実施形態では、L D R画像の量子化ステップを基準にローカルデコードをする / しないを切り替える方法について説明する。

【 0 0 7 7 】

以下、本第 4 の実施形態における処理を、図 1 4 を参照して説明する。

【 0 0 7 8 】

20

ステップS 3 0 1、3 0 2では、C P U 2 0 1は、第 1 の実施形態と同様に、符号化対象のH D R画像データを取得し、線形L D R画像を作成する。ステップS 3 0 6にて、C P U 2 0 1は、L D R画像の符号化条件を取得する。符号化条件には、L D R画像のQパラメータの情報も含まれる。Qパラメータは、1 から 1 0 0 までの整数値を取り、1 0 0 を指定すると量子化ステップがもっとも細くなり、1 を指定すると、量子化ステップが最も粗くなる。ステップS 1 4 0 1にて、C P U 2 0 1は、ステップS 3 0 6で取得したL D R画像の符号化条件からQパラメータの値を参照する。そして、C P U 2 0 1は、Qパラメータと、あらかじめ決められた値 Th_q を比較する。そしてQパラメータが Th_q 未満であれば、C P U 2 0 1は処理をステップS 3 0 5、そうでなければステップS 3 1 3に進める。本実施形態の場合、 Th_q は6 0 とする。したがって、L D R画像のQパラメータが6 0 未満であれば、ステップS 3 0 5に進み、そうでなければステップS 3 1 3へ進む。ステップS 3 0 5以降の処理は、第 1 の実施形態と同様の処理であるので、ここでの詳細な説明は割愛する。

30

【 0 0 7 9 】

上記の結果、L D R画像を符号化する際のQパラメータから、L D R画像の劣化具合を予測し、劣化の程度が大きいと判断した場合にはローカルデコードを行い、劣化の程度が許容できる場合にはローカルデコードは行わないとする処理が実現できる。

【 0 0 8 0 】

[第 4 の実施形態の変形例]

J P E G X Tでは、J P E G圧縮する際に、輝度のサブサンプリングは行わず、色差成分のみサブサンプリングを定義している。C b、C rのx軸方向およびy軸方向のサンプリング比をそれぞれ $S b x$ 、 $S b y$ 、 $S r x$ 、 $S r y$ とすると、J P E G X Tでは以下の4つのサブサンプリング比が定義されている。

40

(1) ($S b x$, $S b y$, $S r x$, $S r y$) = (1 , 1 , 1 , 1)

(2) ($S b x$, $S b y$, $S r x$, $S r y$) = (1 , 2 , 1 , 2)

(3) ($S b x$, $S b y$, $S r x$, $S r y$) = (2 , 1 , 2 , 1)

(4) ($S b x$, $S b y$, $S r x$, $S r y$) = (2 , 2 , 2 , 2)

上記の(1)の場合、間引きされる色差成分はないので、画質の劣化は少ないと考えて良い。そこで、本第 4 の変形例では、成分C b、C rのサブサンプリングが上記の(2) ~ (4)のいずれかの場合、ローカルデコードをする。本変形例のH D R画像の圧縮処理

50

を、図 15 のフローチャートに従って説明する。

【 0 0 8 1 】

図 15 と図 14 との違いは、図 14 におけるステップ S 1 4 0 1 がステップ S 1 5 0 1 で置き換わった点である。図 15 のステップ S 1 5 0 1 にて、CPU 2 0 1 は、ステップ S 3 0 6 で取得した L D R 画像の符号化条件から、サブサンプリング情報を参照する。そして、CPU 2 0 1 は、サブサンプリングが上記 (2) ~ (4) のいずれかであったと判断した場合、処理をステップ S 3 0 5 に、そうでなければ、処理をステップ S 3 1 3 に進める。このようにすることで、サブサンプリングされる場合には、L D R 画像をローカルデコードした結果を用いて H D R 補助データが作成されることになる。

【 0 0 8 2 】

[第 5 の実施形態]

上記第 1 乃至第 3 の実施形態、並びにその変形例では、H D R 画像の情報や L D R 画像の符号化条件などで、ローカルデコードをする / しないを切り替えた。しかし、L D R 画像の劣化を直接評価することでローカルデコードを切り替えても良い。そこで本第 5 の実施形態では、量子化前後の値を直接比較することで、その劣化の程度を評価し、ローカルデコードする / しないをマクロブロック単位で決定する例を説明する。

【 0 0 8 3 】

本第 5 の実施形態の場合、第 1 の実施形態の変形例とほぼ同じ動作をするが、図 7 におけるステップ S 7 0 5 のマクロブロック単位の L D R 画像圧縮の方法が異なる。そこで、本第 5 の実施形態におけるマクロブロック単位の L D R 画像圧縮の方法について図 1 6 を用いて説明する。なお、図 1 6 において図 8 と同じ動作の処理には、同じ番号を付与し、その説明は省略する。

【 0 0 8 4 】

ステップ S 8 0 1 にて、CPU 2 0 1 は、マクロブロック単位の線形 L D R 画像の作成を行う。この処理については、第 1 の実施形態の変形例と同じである。次にステップ S 3 0 7 にて、CPU 2 0 1 は、ステップ S 8 0 1 で作成した線形 L D R 画像に、第 1 の実施形態と同様に、色変換を行う。ステップ S 1 6 0 1 にて、CPU 2 0 1 は、ステップ S 3 0 7 で得られた L D R 画像のマクロブロックに D C T を実行し、その時の輝度の D C T 係数を例えばメモリ 2 0 6 に保存する。ステップ S 1 6 0 2 にて、CPU 2 0 1 は、ステップ S 1 6 0 1 で得られた Y , C b , C r の D C T 係数の量子化処理を行う。ステップ S 1 6 0 3 にて、CPU 2 0 1 は、輝度 Y の成分だけ逆量子化する。ステップ S 3 0 9 にて、CPU 2 0 1 は、ステップ S 1 6 0 2 で得られた量子化後の D C T 係数をエントロピー符号化する。ステップ S 1 6 0 4 にて、CPU 2 0 1 は、ステップ S 1 6 0 1 で保存した量子化前の輝度の D C T 係数と、ステップ S 1 6 0 3 で得られた逆量子化後の輝度の D C T 係数を、マクロブロックの 6 4 個の係数についてそれぞれ差分値を算出する。そして、ステップ S 1 6 0 5 にて、CPU 2 0 1 は、ステップ S 1 6 0 4 で得られた 6 4 個の差分値のうち、その絶対値があらかじめ決められた Th_dc 以上の係数の数 K をカウントする。本実施形態では、 $Th_dc = 64$ とする。ステップ S 1 6 0 6 にて、CPU 2 0 1 は、ステップ S 1 6 0 5 で得られた K と、あらかじめ決められた値 Th_diff を比較する。本実施形態では、 $Th_diff = 32$ とする。K が Th_diff より大きければ、CPU 2 0 1 は、このマクロブロックにおける量子化前後の D C T 係数の差異が大きいと判断し、処理をステップ S 1 6 0 7 に進め、ローカルデコードする。そうでなければ、ローカルデコードをせずにこのフローを終了する。すなわち、1 マクロブロックの半数以上の D C T 係数が量子化により、 $Th_dc = 64$ 以上の差が出る場合には、ローカルデコードすることになる。

【 0 0 8 5 】

ステップ S 1 6 0 7 にて、CPU 2 0 1 は、ステップ S 1 6 0 2 で得られた量子化後の D C T 係数を逆量子化する。輝度 Y に関してはステップ S 1 6 0 3 で逆量子化されているので、ここでは色差の C b , C r だけ逆量子化すれば良い。ステップ S 3 1 2 にて、CPU 2 0 1 は、逆色変換をして、線形 L D R 画像を得る。ステップ S 8 0 4 にて、CPU 2

10

20

30

40

50

01は、マクロブロックの線形LDR画像を、ローカルデコードして得られたLDR画像で上書きして、このフローを終了する。

【0086】

本第5の実施形態によれば、すべてのDCT係数に対して量子化前後の値を比較することから、計算コストが大きくなる。しかし、LDR画像の符号化による劣化の度合いを見ることができるため、劣化が大きいところは確実にローカルデコードした画像を使ってHDR補助データを作成することができることになる。

【0087】

[第5の実施形態の変形例]

上記第5の実施形態では、一律にすべてのDCT係数を比較し、ローカルデコードをする/しないを切り替えたが、LDR画像の符号化条件であるQパラメータによって、比較するDCT係数の数およびTh_diffを切り替えてもよい。たとえば、Th_diffは比較するDCT係数の半数とすればよい。図17にDCT係数を模式的に示す。左上隅がDC成分であり、右下がAC成分であるとする。LDR画像のQパラメータが大きい場合は、図17の符号1701で示す、高周波側のDCT係数6個のみを比較する。逆にLDR画像のQパラメータが小さい場合は、図17の符号1701~1704を合わせた43個のDCT係数を比較対象とする。

【0088】

本変形例の処理フローを図18に示す。なお、図16と同じ処理をするステップには、図16と同じ番号を付与し、その説明は省略する。

【0089】

ステップS801にて、CPU201は、マクロブロック単位の線形LDR画像を作成する。ステップS307にて、CPU201は、ステップS801で作成した線形LDR画像の色変換を行う。ステップS1601にて、CPU201は、色変換後のLDR画像にDCTをかけ、その輝度の係数をメモリ206に保存する。ステップS1602にて、CPU201は、LDR画像のQパラメータに応じて、DCT係数の量子化を行う。ステップS309にて、CPU201はエントロピー符号化を行う。ステップS1801にて、CPU201は、LDR画像のQパラメータが10以上であるか判定する。10以上であれば、ステップS1802へ進む。10未満であれば、CPU201は、量子化による差異が大きい(量子化ステップが大きい)と判断し、ステップS1607へ処理を進め、ローカルデコード処理を実行する。ステップS1702にて、CPU201は、LDR画像のQパラメータが90未満であるか判定する。90未満であれば、CPU201はステップS1803へ処理を進める。また、90以上であれば、CPU201は、量子化による差異は小さいと判断し、ローカルデコードは行わず、この処理フローを終了する。ステップS1803にて、CPU201は、Qパラメータに応じた比較対象となるDCT係数の数と、しきい値Th_diffを取得する。本実施形態では以下のような値を取得するものとする。

(1) Qが70以上90未満の場合

比較対象のDCT係数は、図17の1701で示す6係数

Th_diff = 3

(2) Qが50以上70未満の場合

比較対象のDCT係数は、図17の1701および1702で示す15係数

Th_diff = 8

(3) Qが30以上50未満の場合

比較対象のDCT係数は、図17の1701~1703で示す28係数

Th_diff = 14

(4) Qが10以上30未満の場合

比較対象のDCT係数は、図17の1701~1704で示す43係数

Th_diff = 22

次に、CPU201は、処理をステップS1603に進める。このステップS1603

以降は第 5 の実施形態と同様の処理であるので、ここでの説明は省略する。

【 0 0 9 0 】

(その他の実施例)

本発明は、上述の実施形態の 1 以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける 1 つ以上のプロセッサがプログラムを読み出し実行する処理でも実現可能である。また、1 以上の機能を実現する回路 (例えば、A S I C) によっても実現可能である。

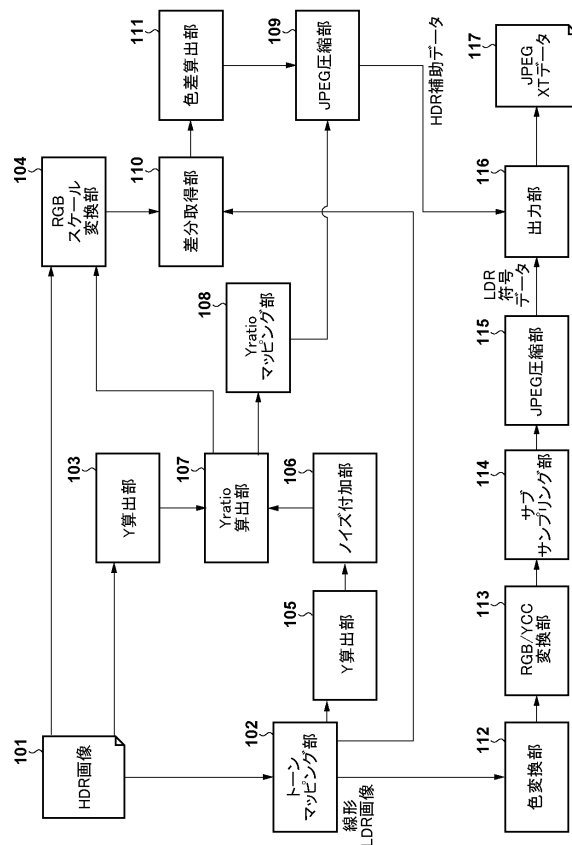
【 符号の説明 】

【 0 0 9 1 】

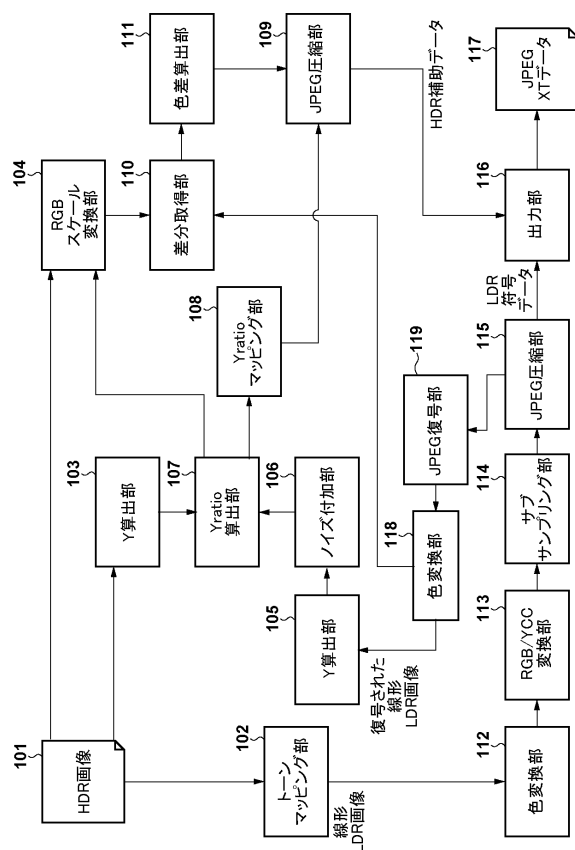
2 0 1 ... C P U、2 0 2 ... 入力部、2 0 3 ... H D R 補助データ生成部、2 0 4 ... L D R 画像生成部、2 0 5 ... J P E G コーデック部、2 0 6 ... メモリ、2 0 7 ... 表示部、2 0 8 ... 蓄積部

10

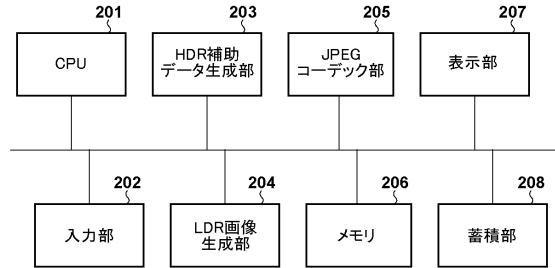
【 図 1 A 】



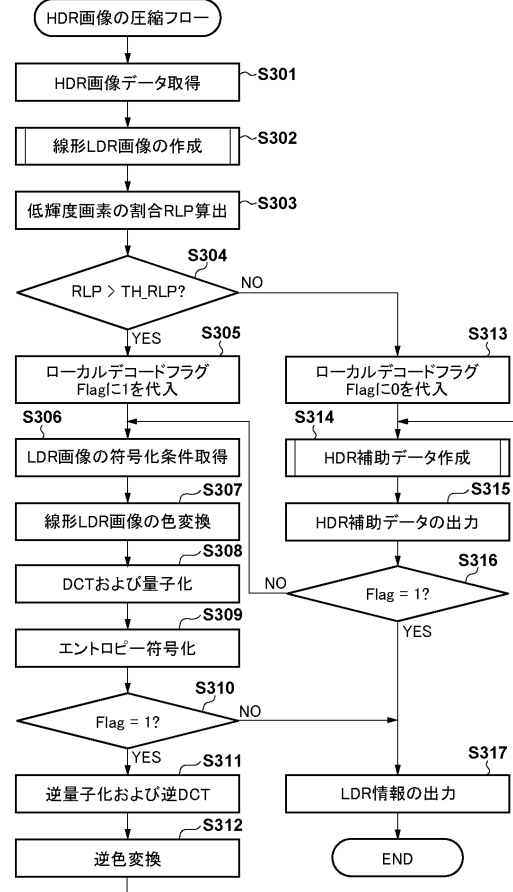
【 図 1 B 】



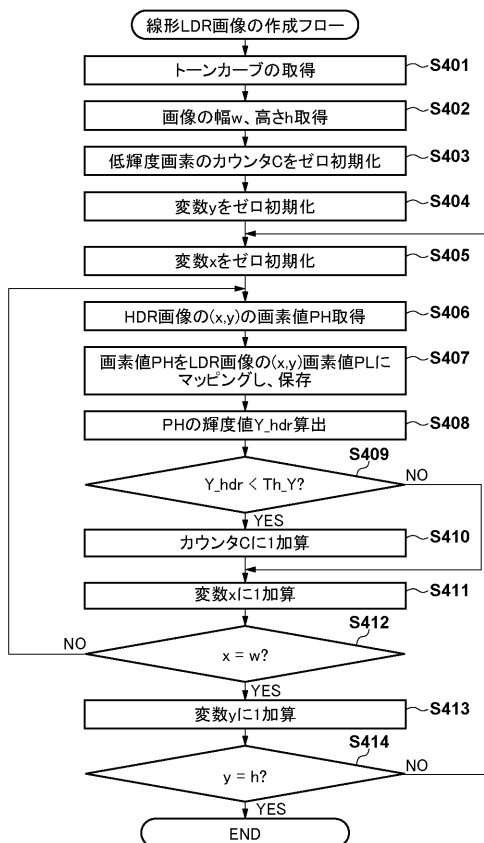
【図 2】



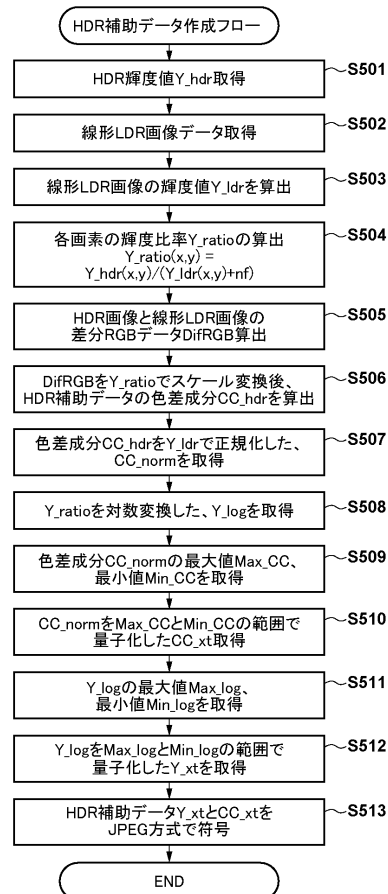
【図 3】



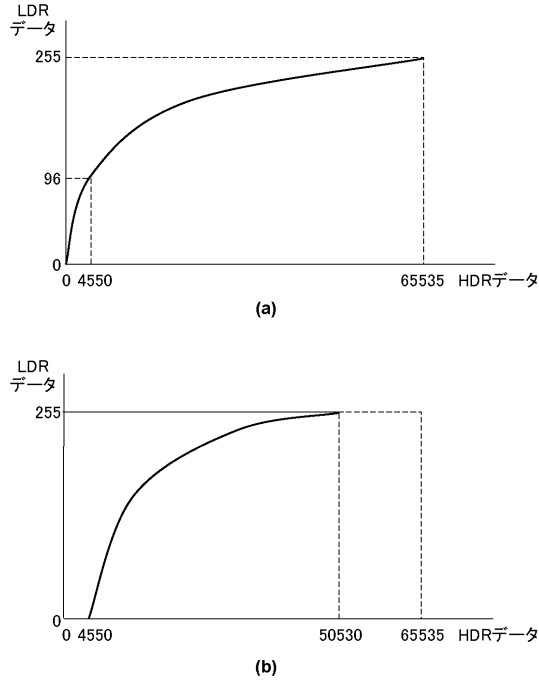
【図 4】



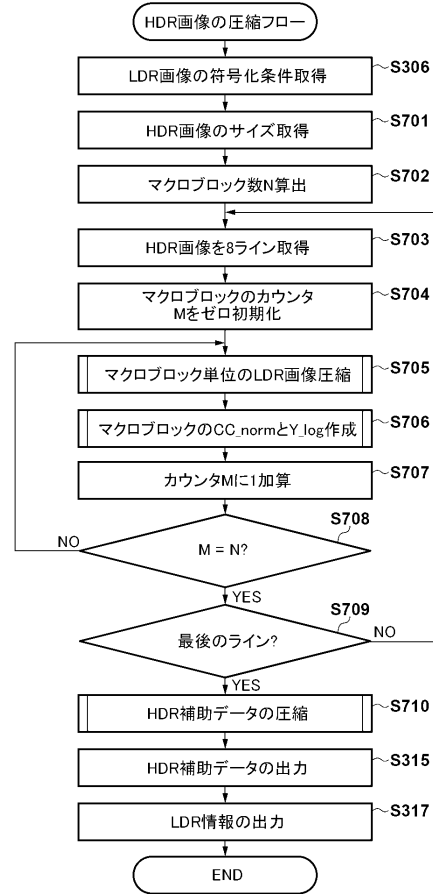
【図 5】



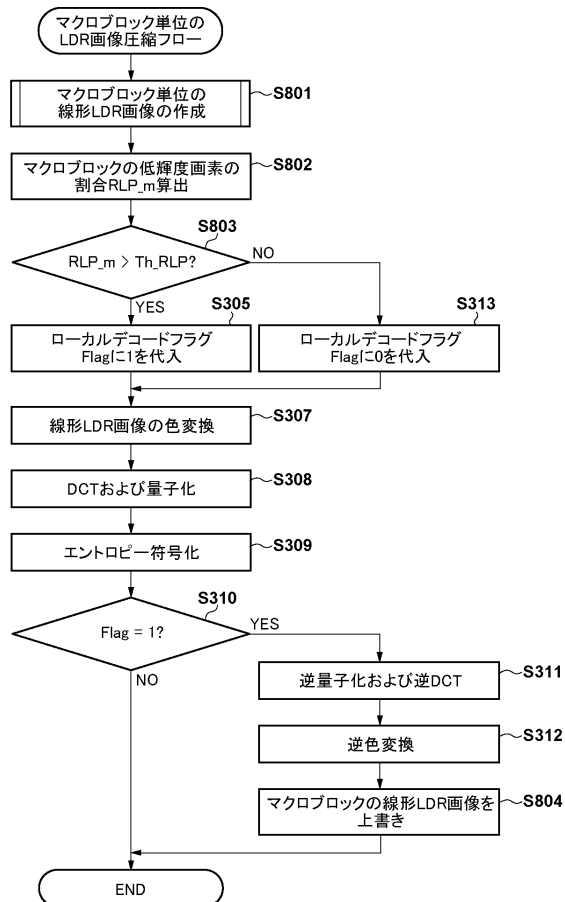
【図 6】



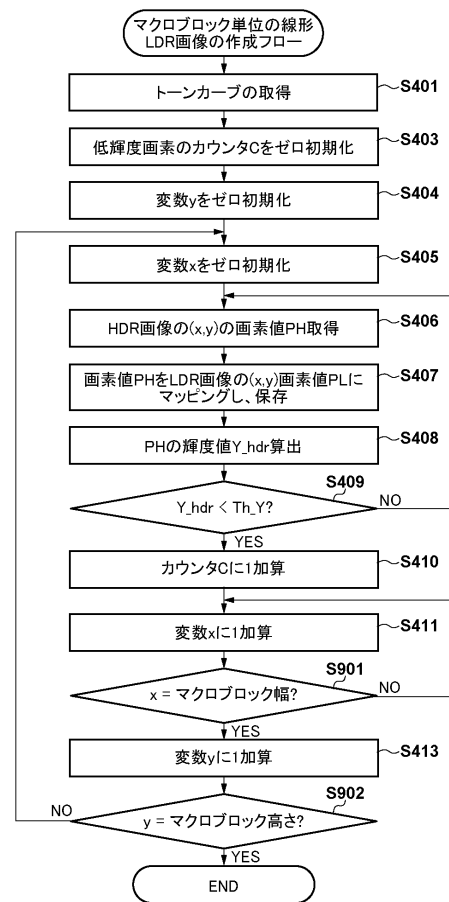
【図 7】



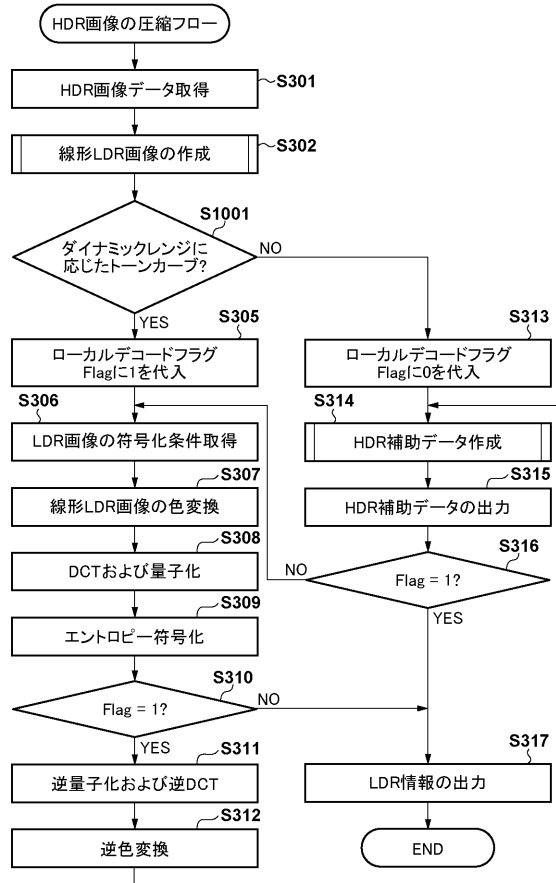
【図 8】



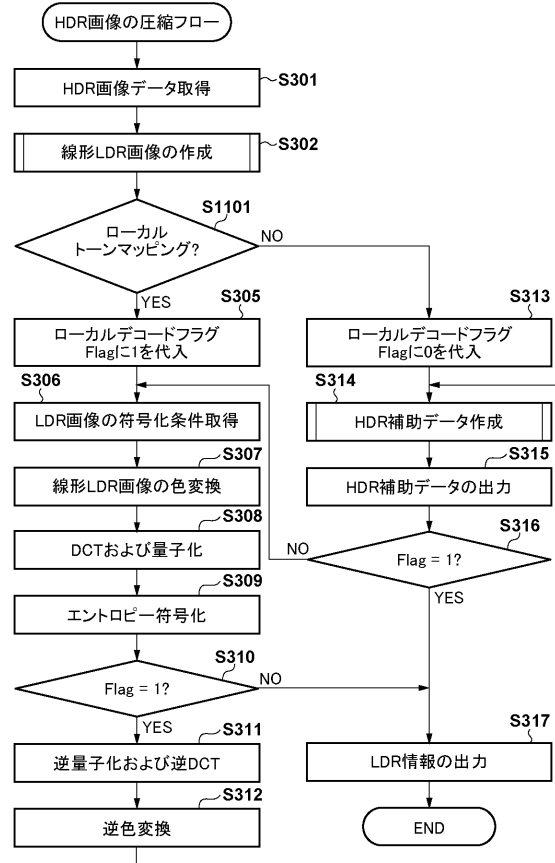
【図 9】



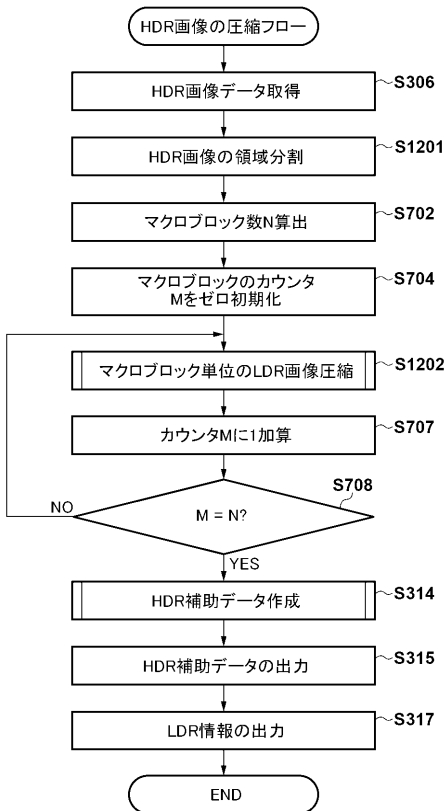
【図 10】



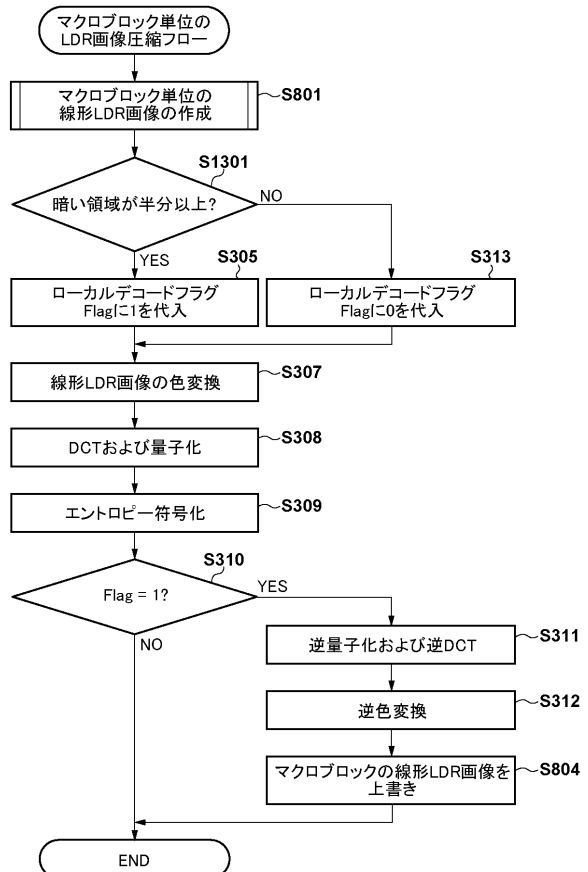
【図 11】



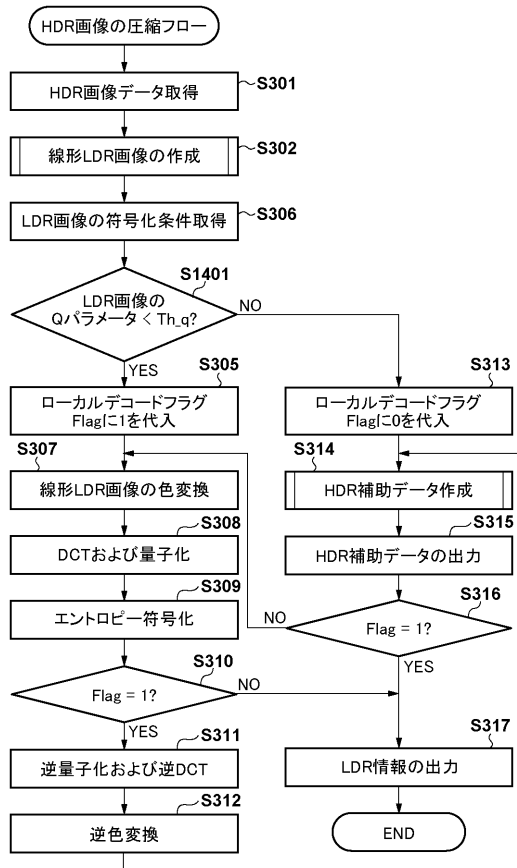
【図 12】



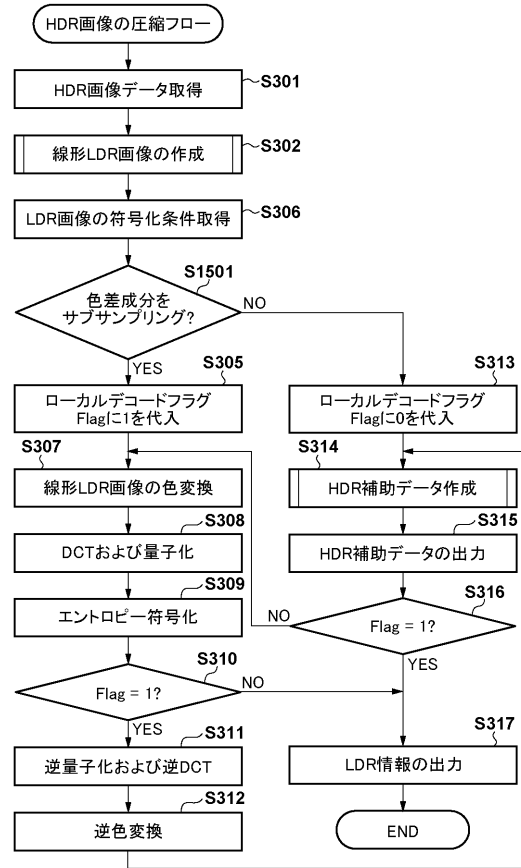
【図 13】



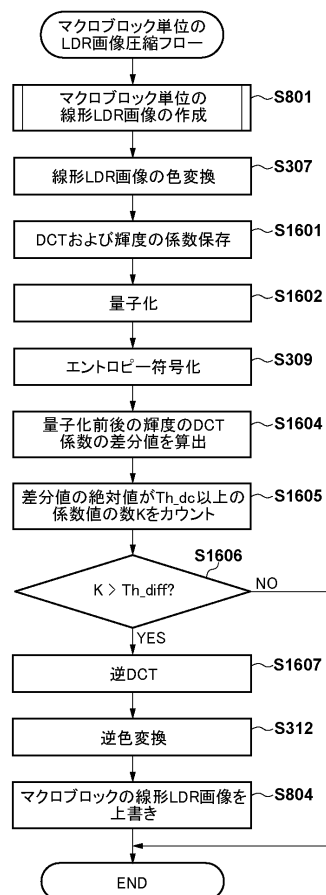
【 図 1 4 】



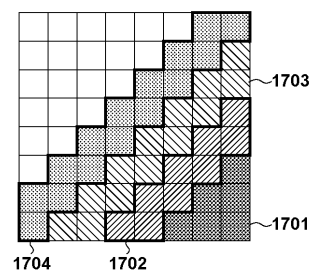
【 図 1 5 】



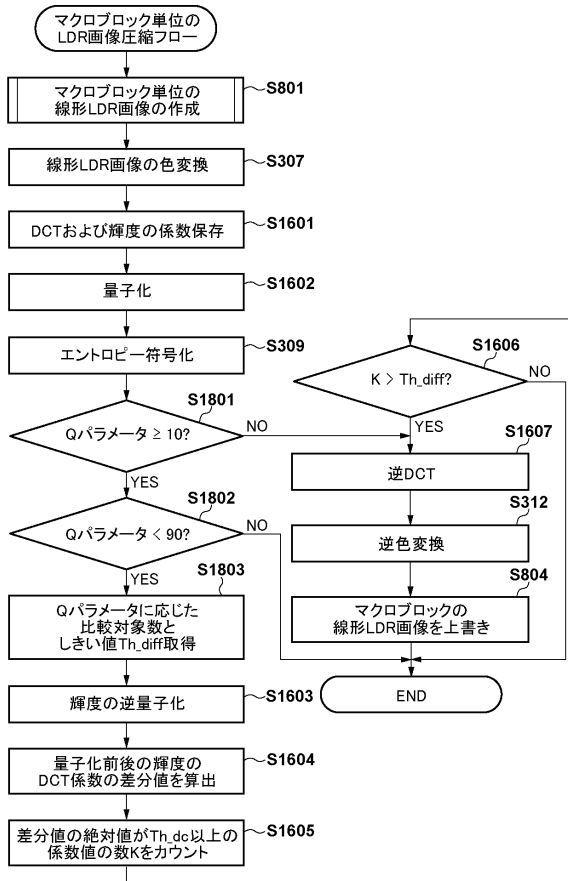
【 図 1 6 】



【 図 1 7 】



【図 18】



フロントページの続き

(56)参考文献 特開 2015 - 019286 (JP, A)

特開 2015 - 019287 (JP, A)

特表 2011 - 509536 (JP, A)

鈴木 亮 ほか 3 名, 量子化誤差の影響を考慮した J P E G X T エンコーダー, 電子情報通信学会技術研究報告, 一般社団法人電子情報通信学会, 2015 年 11 月 26 日, 第 115 巻、第 348 号, p.29 ~ 34

(58)調査した分野(Int.Cl., DB 名)

H04N 1/00 - 1/64

H04N 19/00 - 19/98