

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6936699号
(P6936699)

(45) 発行日 令和3年9月22日 (2021.9.22)

(24) 登録日 令和3年8月31日 (2021.8.31)

(51) Int.Cl. F I
HO 4 N 1/413 (2006.01) HO 4 N 1/413
HO 4 N 19/36 (2014.01) HO 4 N 19/36
HO 4 N 19/436 (2014.01) HO 4 N 19/436

請求項の数 10 (全 18 頁)

(21) 出願番号	特願2017-209563 (P2017-209563)	(73) 特許権者	000001007
(22) 出願日	平成29年10月30日 (2017.10.30)		キヤノン株式会社
(65) 公開番号	特開2019-83401 (P2019-83401A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	令和1年5月30日 (2019.5.30)	(74) 代理人	110003281
審査請求日	令和2年10月22日 (2020.10.22)		特許業務法人大塚国際特許事務所
		(72) 発明者	榎田 幸
			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
		(72) 発明者	岸 裕樹
			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
		審査官	西谷 憲人

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及びプログラム

(57) 【特許請求の範囲】

【請求項 1】

HDR(High Dynamic Range)画像を符号化する画像符号化装置において、
 符号化対象のHDR画像のダイナミックレンジを圧縮してLDR (Low Dynamic Range)画像を
 生成する第1の生成手段と、

前記LDR画像を符号化する第1の符号化手段と、

前記LDR画像と前記HDR画像から、前記LDR画像と前記HDR画像との差分を表すHDR差分画
 像を生成する第2の生成手段と、

前記HDR差分画像を符号化する第2の符号化手段と、

少なくとも前記第2の生成手段及び前記第2の符号化手段を制御する制御手段とを有し、
 前記第2の生成手段は、

前記LDR画像と前記HDR画像とに基づいて、当該HDR画像とLDR画像の差分を表す実
 数の成分値を導出手段と、

前記実数の成分値を、設定された実数範囲に基づいて整数化し、得られた整数の
 成分値を前記HDR差分画像の成分として前記第2の符号化手段に供給する整数化手段とを有
 し、

前記制御手段は、

前記HDR画像の符号化に先立って前記整数化手段に設定した初期の実数範囲を一度
 も外れる実数の成分値を導出しないまま、前記導出手段が前記HDR差分画像の生成を終了
 した場合、前記第2の符号化手段で得られた符号化データを前記HDR差分画像の符号化デー

10

20

タとして決定し、

前記導出手段が、前記初期の実数範囲を一度でも外れる実数の成分値を導出した場合、前記第2の生成手段及び前記第2の符号化手段による再符号化を行わせる

ことを特徴とする画像符号化装置。

【請求項 2】

前記制御手段は、

前記導出手段による実数の成分値の導出を前記HDR差分画像の最終の画素に至るまで継続させ、前記実数範囲を超える成分値が導出される度に当該実数範囲を更新すると共に、最後に更新した位置を記憶する更新手段とを有し、

前記導出手段による前記HDR差分画像の最終の画素の実数の成分値を導出するまでに、前記更新手段による前記実数範囲を更新があった場合、前記導出手段及び前記第2の符号化手段を制御して、前記HDR差分画像の先頭の画素から前記更新手段で記憶された位置までの再符号化を実行させる

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】

前記制御手段は、

前記HDR画像の符号化を開始してから、前記実数範囲を最初に更新した以降は、前記第2の符号化手段による符号化を停止させ、

前記導出手段による前記HDR差分画像の最終の画素の実数の成分値を導出した場合に、前記導出手段及び前記第2の符号化手段を制御して、前記HDR差分画像の再符号化させる

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】

前記制御手段は、

前記導出手段が導出する実数の成分値を、前記HDR画像の符号化に先立って前記整数化手段に設定した初期の実数範囲にクリッピングさせる

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 5】

前記制御手段は、

前記導出手段が導出する実数の成分値が、前記HDR画像の符号化に先立って前記整数化手段に設定した初期の実数範囲を超えた場合、当該実数の成分値に基づき実数範囲を更新すると共に、前記実数の成分値と位置とを符号化データの一部として生成する手段を含む

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 6】

前記第2の生成手段と前記第2の符号化手段の組を複数有し、

各組が前記符号化対象のHDR画像を分割したバンドについて並列に符号化を行い、

かつ、各組が有する前記整数化手段に設定する初期の実数範囲を異ならせる

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 7】

前記導出手段は、前記HDR画像と前記LDR画像の差分を表す、実数の輝度比を表すYレシオ、及び、実数の2つの色差差分であるCb_e、Cr_eを導出することを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の画像符号化装置。

【請求項 8】

1成分につき8ビットのデータを符号化するJPEG符号化手段を前記第1、第2の符号化手段として利用することを特徴とする請求項 1 乃至 7 のいずれか1項に記載の画像符号化装置

。

【請求項 9】

HDR(High Dynamic Range)画像を符号化する画像符号化装置の制御方法において、

符号化対象のHDR画像のダイナミックレンジを圧縮してLDR (Low Dynamic Range) 画像を

10

20

30

40

50

生成する第1の生成工程と、

前記LDR画像を符号化する第1の符号化工程と、

前記LDR画像と前記HDR画像から、前記LDR画像と前記HDR画像との差分を表すHDR差分画像を生成する第2の生成工程と、

前記HDR差分画像を符号化する第2の符号化工程と、

少なくとも前記第2の生成工程及び前記第2の符号化工程を制御する制御工程とを有し、

前記第2の生成工程は、

前記LDR画像と前記HDR画像とに基づいて、当該HDR画像とLDR画像の差分を表す実数の成分値を導出工程と、

前記実数の成分値を、設定された実数範囲に基づいて整数化し、得られた整数の成分値を前記HDR差分画像の成分として前記第2の符号化工程に供給する整数化工程とを有し、

前記制御工程は、

前記HDR画像の符号化に先立って前記整数化工程に設定した初期の実数範囲を一度も外れる実数の成分値を導出しないまま、前記導出工程が前記HDR差分画像の生成を終了した場合、前記第2の符号化工程で得られた符号化データを前記HDR差分画像の符号化データとして決定し、

前記導出工程が、前記初期の実数範囲を一度でも外れる実数の成分値を導出した場合、前記第2の生成工程及び前記第2の符号化工程による再符号化を行わせる

ことを特徴とする画像符号化装置の制御方法。

【請求項 10】

コンピュータが読み込み実行することで、前記コンピュータに、請求項9に記載の方法の各工程を実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は画像データ、特に、高ダイナミックレンジ画像の符号化技術に関するものである。

【背景技術】

【0002】

現在、静止画像データが様々な形式で流通している。その代表的な形式がJPEG圧縮形式である。JPEG圧縮形式の画像は、デジタルカメラの出力ファイルとしても利用されており、ウェブブラウザを始め、画像を表示・処理することができるアプリケーションソフトウェアのほぼ100%がJPEG圧縮方式の画像をサポートしている。一般的にJPEG圧縮方式がサポートしている静止画像は1コンポーネント当り8ビットである。つまり、1コンポーネント当り256階調までしか再現できない。このような1コンポーネント当たり8ビットの画像をLDR(Low Dynamic Range)画像と呼ぶ。

【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JPEG圧縮方式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、例えば32bit程度のデータを持つ高ダイナミックレンジ(High Dynamic Range;以下、HDR)画像が注目されている。

【0004】

HDR画像の符号化方法は種々提案されている。特許文献1、2はその一例である。特許文献1、2では、HDR画像データからLDR画像データを生成し、それをJPEG符号化する。そして、LDR画像からHDR画像に戻すための差分情報を作成し、この差分情報を前記JPEG符号列内のアプリケーションマーカセグメントに書き込む。この符号化方法の長所は、従来のJPEG圧縮方式をサポートしている画像ビューワーでLDR画像データを復号・表示できる点にある。

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特表2007-534238号公報

【特許文献2】特開2011-193511号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

HDR画像データの格納方法は様々な方法がある。ここではHDR画像データをLDR画像データと差分情報で表現する場合で説明する。この差分情報の生成・符号化方法としても、様々な方法が考えられるが、ここではISO/IEC JTC 1/SC 29/WG1のJPEG HDR Ad Hoc Groupで検討されている方式をベースにする。

10

【0007】

この方式の概要は、入力されるHDR画像データに対し、各コンポーネントのダイナミックレンジを8ビットに圧縮するトーンマッピング処理を行い、LDR画像データを生成する。このLDR画像データをJPEG符号化する。その後、LDR画像データからHDR画像データを復元するために必要なHDR差分データを生成し、このHDR差分データを符号化する。これら2種類の符号化データを1つのHDR符号として、ファイル等に出力する。

【0008】

上記HDR差分データは、LDR画像データと元のHDR画像データの輝度成分比（Yレシオ）を計算し、このYレシオを使ってLDR画像データをHDR画像データのダイナミックレンジに戻し、元のHDR画像データとの色差成分の差分を計算する。このYレシオと色差成分の差分を8ビット化して得たHDR差分データを、LDR画像データの符号化と同じJPEG符号化する。

20

【0009】

図4は、上記のHDR差分データ生成装置のブロック構成図である。この装置は、輝度成分比算出部401、対数変換部402、バッファB403、最大値／最小値導出部A404、整数化部A405、バッファD406、入力画像除算部407、LDR相当画像／LDR画像差分算出部408、色変換部409、ノーマライズ部410、バッファC411、最大値／最小値導出部B412、整数化部B413、バッファE414を有する。

【0010】

HDR差分データ生成装置の概要は次の通りである。まずHDR画像とLDR画像のそれぞれの輝度成分から、画素毎に“HDR輝度成分値／LDR輝度成分値”を演算し、この結果を輝度成分比（Yレシオ）とする。次に、HDR画像のRGBの各色成分に対し上記輝度成分比で除算することで、LDR相当のダイナミックレンジを持つ画像（LDR相当画像）を生成する。その後LDR画像とLDR相当画像の間で、画素毎にRGBの各色成分の差分を求める。この差分データに対して、YCbCr色変を行い、Cb差分、Cr差分のデータを生成する。こうして、Yレシオ、Cb差分、Cr差分が得られる。Yレシオ、Cb差分、Cr差分を、1画素を構成する3成分と見立てた1枚の画像を、LDR画像符号化部に送り符号化することで、HDR差分データの符号化データを生成できる。

30

【0011】

図5のフローチャートは、図4の構成の処理手順をして表したものである。以下、上記HDR差分データの符号化方法の詳細を図5を用いて説明する。なお、この処理を行うに先立って、HDR画像並びにLDR画像が既に所定のメモリに格納されているものとする。

40

【0012】

ステップS501の輝度成分比算出は、輝度成分比算出部401の処理である。輝度成分比算出部401は、所定のメモリに格納されているHDR画像とLDR画像を取得する。そして、輝度成分比算出部401は、以下の式に従い、HDR画像の輝度成分Y_hdrとLDR画像の輝度成分Y_ldr、並びにYレシオ（Ratio）を算出する。

$$Y_hdr = 0.299 * R_hdr + 0.587 * G_hdr + 0.114 * B_hdr$$

$$Y_ldr = 0.299 * R_ldr + 0.587 * G_ldr + 0.144 * B_ldr$$

$$Ratio = Y_hdr / Y_ldr$$

50

上記において、R_Hdr,G_hdr,B_HdrはHDR画像におけるR,G,Bの成分値であり、R_ldr,G_ldr,B_ldrはLDR画像におけるR,G,B成分値である。

【 0 0 1 3 】

ここで、輝度成分比算出部401は、画素毎に算出されるRatio (Yレシオ) をバッファE414へ格納する。

【 0 0 1 4 】

ステップS502の対数変換は対数変換部402の処理である。対数変換部402は、バッファE414からYレシオを取得し、底を2とする対数変換を行う。これは、Yレシオを最終的にJPEGコーデック部で符号化されるため、8ビットの整数 (0~255) で表現する必要があるためである。この対数表現により、対数表現を行わない場合と比べ、微妙な輝度変化を再現する事ができる。対数変換部402は、この対数表現されたYレシオ (Log Yレシオ) を一旦バッファB403に格納する。

10

【 0 0 1 5 】

ステップS503は、最大値 / 最小値導出部404の処理である。最大値 / 最小値導出部404は、画像全体のLog Yレシオにおける、最大値と最小値 (Max_LogYRatio、Min_LogYRatio) を算出し、この値を整数化部405に出力する。

【 0 0 1 6 】

ステップS504の整数化は整数化部405の処理である。整数化部405は、Max_LogYRatio、Min_LogYRatioを使い、バッファB403に格納されているLog Yレシオを8ビット化し、その8ビットで表す整数値をバッファD406へ出力する。またこのMax_LogYRatio、Min_LogYRatioは復号時に8ビット整数を元の実数表現に戻す際に必須である。そのため、最大値 / 最小値導出部A404は、HDR差分データの符号内に保存するために所定のメモリに、Max_LogYRatio、Min_LogYRatioの値を格納する。

20

【 0 0 1 7 】

ステップS505の入力画像除算は、入力画像除算部407の処理である。入力画像除算部407は、HDR画像の色成分に対する処理である。このステップS505にて、入力画像除算部407は、所定のメモリに格納されているRGB色空間のHDR画像に対し、バッファE414に格納されているYレシオで除算し、RGB色空間のLDR相当画像を生成する。

【 0 0 1 8 】

ステップS506の差分算出は、LDR相当画像 / LDR画像差分算出部408の処理である。LDR相当画像 / LDR画像差分算出部408は、RGB色空間における、先のステップS505で求めたLDR相当画像と、所定のメモリに格納されているLDR画像との画素毎の差分 (R_e、G_e、B_e) を求める。

30

【 0 0 1 9 】

ステップS507の色変換は、色変換部409の処理である。色変換部409は、ステップS506で求めたRGB色空間での画素毎の差分値に対して、以下の式を用いて色変換を行うことで、色差差分Cb_e、Cr_eを得る。

$$Cb_e = -0.1687 * R_e - 0.3313 * G_e + 0.5 * B_e$$

$$Cr_e = 0.5 * R_e - 0.4187 * G_e - 0.0813 * B_e$$

【 0 0 2 0 】

ステップS508のノーマライズはノーマライズ部410の処理である。ノーマライズ部410は、以下の式を用いて色差差分Cb_e、Cr_eをY_ldrでノーマライズし、バッファC 411に格納する。

$$Cb_e = Cb_e / Y_ldr$$

$$Cr_e = Cr_e / Y_ldr$$

【 0 0 2 1 】

ステップS509の最大値 / 最小値算出は、最大値 / 最小値導出部B 412の処理である。最大値 / 最小値導出部B 412は、バッファC 411に保存されている画像全体のCb_e、Cr_eからそれぞれの最大値 (Max_Cb_e、Max_Cr_e) と、それぞれの最小値 (Min_Cb_e、Min_Cr_e) を導出し、整数化部413へ出力する。また、このMax_Cb_e、Max_Cr_e、Min_Cb_e、Min_Cr_e

40

50

eは復号時に必須となる。それ故、最大値／最小値導出部B 412は、HDR差分データの符号内に保存するため、これらMax_Cb_e、Max_Cr_e、Min_Cb_e、Min_Cr_eを所定のメモリへ格納する。

【 0 0 2 2 】

ステップS510の整数化は整数化部413の処理である。整数化部413は、Max_Cb_e、Max_Cr_e、Min_Cb_e、Min_Cr_eを元に、バッファC411に保存されているCb_e、Cr_eを8ビット化してバッファD406に格納する。

【 0 0 2 3 】

ステップS511にて、バッファD406に、1画像分のLogYレシオ、Cb_e、Cr_eの各8ビット整数データが揃った段階で、この3種類の値を3成分と見立てた1枚の画像（HDR差分データ）として、不図示のJPEGコーデック部に供給する。JPEGコーデック部は供給されたHDR差分データをJPEG符号化し、差分画像符号を生成することになる。

【 0 0 2 4 】

以上説明した処理の中で、最大値／最小値導出部A404、並びに最大値／最小値導出部B412は、画像の全ての入力データに対する処理を終えないと、その画像内の最大値・最小値を求める事ができない点に注意されたい。つまり、図4で示した4つのバッファのうち、バッファB403、バッファC411は、符号化対象のHDR画像の画像サイズ分のメモリが必要となる。更にHDR画像データは、各色成分を32ビット以上のダイナミックレンジを持っているものであるから、そのメモリサイズは膨大なものとなるのが理解できよう。

【 0 0 2 5 】

また、バッファ容量を少なくすると、今度は画像全体の処理を複数パス形式で行わざるを得ず、最終的なHDR画像の符号化データを得るまでに多くの時間を必要するという問題も発生する。

【 0 0 2 6 】

本発明はかかる問題に鑑みなされたものであり、これまでよりも十分に少ないメモリでありながら、高速にHDR画像の符号化を行う技術を提供しようとするものである。

【課題を解決するための手段】

【 0 0 2 7 】

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(High Dynamic Range)画像を符号化する画像符号化装置において、

符号化対象のHDR画像のダイナミックレンジを圧縮してLDR (Low Dynamic Range)画像を生成する第1の生成手段と、

前記LDR画像を符号化する第1の符号化手段と、

前記LDR画像と前記HDR画像から、前記LDR画像と前記HDR画像との差分を表すHDR差分画像を生成する第2の生成手段と、

前記HDR差分画像を符号化する第2の符号化手段と、

少なくとも前記第2の生成手段及び前記第2の符号化手段を制御する制御手段とを有し、

前記第2の生成手段は、

前記LDR画像と前記HDR画像とに基づいて、当該HDR画像とLDR画像の差分を表す実数の成分値を導出手段と、

前記実数の成分値を、設定された実数範囲に基づいて整数化し、得られた整数の成分値を前記HDR差分画像の成分として前記第2の符号化手段に供給する整数化手段とを有し、

前記制御手段は、

前記HDR画像の符号化に先立って前記整数化手段に設定した初期の実数範囲を一度も外れる実数の成分値を導出しないまま、前記導出手段が前記HDR差分画像の生成を終了した場合、前記第2の符号化手段で得られた符号化データを前記HDR差分画像の符号化データとして決定し、

前記導出手段が、前記初期の実数範囲を一度でも外れる実数の成分値を導出した場

10

20

30

40

50

合、前記第2の生成手段及び前記第2の符号化手段による再符号化を行わせることを特徴とする。

【発明の効果】

【0028】

本発明によれば、少ないメモリでありながら、高速にHDR画像の符号化を行うことが可能になる。

【図面の簡単な説明】

【0029】

【図1】第1の実施形態におけるResidualデータ符号化処理フロー図。

【図2】第1の実施形態における再処理対象Blockの図。

10

【図3】実施形態における画像処理システムのブロック図。

【図4】HDR差分データの生成処理のブロック図。

【図5】HDR差分データの生成処理の処理フロー図。

【図6】第2の実施形態におけるResidualデータ符号化処理フロー図。

【図7】第2の実施形態におけるResidualデータ符号化処理フロー図。

【図8】第3の実施形態におけるクリッピング処理の例を示す図。

【図9】第3の実施形態の変形例におけるResidualデータ符号化処理フロー図。

【図10】第3の実施形態の変形例における符号化データの出力例を示す図。

【図11】実施形態におけるHDR差分データ符号化部の構成を示す図。

20

【発明を実施するための形態】

【0030】

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。

【0031】

図3は実施形態におけるHDR(High Dynamic Range)画像を符号化する画像符号化装置のブロック構成図である。本装置は、CPU301、入力部302、HDR差分データ符号部303、LDR画像生成部304、符号列生成部305、LDR画像符号部306、表示部307、メモリ308、通信I/F部309、及び、蓄積部310を有する。

【0032】

CPU301は、メモリ308に格納されたプログラムを実行することで、本装置の全体の制御を行う。入力部302は、ユーザからの指示や、HDR画像データなどを入力する部分であり、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。HDR差分データ符号部303はHDR差分画像データ（以下、単にHDR差分データ）の生成処理と、HDR差分データの符号化処理を行う。LDR画像生成部304は符号化対象のHDR画像からLDR画像を生成する。LDR画像符号部306は、LDR画像を符号化する。このLDR符号化部306は、JPEG符号化と同じであり、1成分当たり8ビットで表される画像を、8*8画素のブロック単位にDCT、量子化、エントロピー符号化を行う。表示部307は、液晶ディスプレイ等で構成され、画像の表示やユーザに提示する情報を表示するものである。メモリ308は、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CPU301に提供する。通信I/F309は外部（例えばネットワーク）との通信を行うためのインターフェースである。蓄積部310は、符号化対象のHDR画像データ、符号化HDR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。符号列生成部305は、LDR画像符号化部306、HDR差分データ符号化部303それぞれが生成した符号を、予め設定されたフォーマットにして、1つのファイルとして生成し、蓄積部310に格納する。

30

40

【0033】

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310に格納されているか、メモリ308のROMに格納されているものとする。蓄積部310に格納されている場合は、一旦メモリ308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、本発明の主眼ではないので、その説明は省略する。

【0034】

50

上記構成において、本実施形態の特徴とする部分はHDR差分データを符号化するHDR差分データ符号化部303にある。そこで、以下では、HDR差分データ符号化部303に的を絞って説明する。

【 0 0 3 5 】

図11は実施形態におけるHDR差分データ符号化部303のブロック構成図である。図4と同じ構成については同じ参照符号を付し、その説明については省略する。

【 0 0 3 6 】

図4のバッファB403、バッファC411は、1画像に含まれる画素数の容量を有することが必要であったが、図11に示すバッファB403'、バッファC411'、バッファD406'は8*8画素分の容量である。ここで示す“8”はJ P E G符号化における符号化単位の画素ブロックが8*8画素であることに由来する。

【 0 0 3 7 】

制御部1101は、図11に示す各構成要素を制御するものであり、図3のCPU301が代替処理しても構わない。

【 0 0 3 8 】

本実施形態では、対数変換部402により得られた輝度比を表すYレシオ、並びに、ノーマライズ部410で生成した色差差分を表すCb_e、Cr_eについて、予め設定された実数の最小値、最大値の初期値を決めて処理を開始する。そして、処理の途中で、この最小値と最大値で示される実数範囲（以下、単に範囲という）を外れる値が出現した場合、最小値、最大値を更新（範囲を更新）して処理を継続する。このとき、この最小値・最大値を更新した場所（画素位置）を記憶保持する。そして符号化対象の画像の全画素について処理を終えた段階で、最小値・最大値を更新した場所が存在する場合は、最終の最小値・最大値で処理をしていない領域を特定し、その領域を最終の最小値・最大値で再符号化するものである。

【 0 0 3 9 】

そのため、初期パラメータ保持部1102は、Yレシオの初期の最小値Ini_Min_LogYratioと最大値Ini_Max_LogYratio、並びに、色差差分の初期の最小値Ini_Min_Cb_e、Ini_Min_Cr_eと最大値Ini_Max_Cb_e、Ini_Max_Cr_eを保持する。これらの初期値は、統計的に求めた値であるものとするが、ユーザ自信が設定しても良い。ユーザ自身が設定する場合には、いくつかのサンプルHDR画像を指定して学習させれば良い。

【 0 0 4 0 】

また、説明を簡単なものとするため、入力されるHDR画像の画像サイズは水平方向80画素、垂直方向120画素であるものとする。そして、8*8画素を1ブロックとした場合の各ブロックの配列状態を図2(a)に示す。図示において、左上隅のブロック200のブロック番号を"0"とし、ラスタースキャン順にブロック番号を割り振る。結果、右下隅の最後のブロックのブロック番号は"149"となる。また、符号化対象のHDR画像、並びに、LDR画像生成部304が生成したLDR画像がメモリ308に既に格納されているものとする。

【 0 0 4 1 】

また図11において、最小値、最大値を導出して整数化する部分は、Yレシオと、Cb_e、Cr_eの色差の差分データに対する部分の2ヶ所あり、入力するコンポーネントが異なるだけで処理のアルゴリズム自身は同じである。それ故、以下では、Yレシオに着目して説明する。

【 0 0 4 2 】

以下、図1のフローチャートに従って、図11の構成を参照しながら制御部1101の処理手順を説明する。

【 0 0 4 3 】

まず、ステップS100にて、制御部1101は、HDR差分データの符号化に先立ち、整数化部405への設定値を初期化する。具体的には、制御部1101は、初期パラメータ保持部1102に保持された値Ini_Max_LogYratio及びIni_Min_LogYratioを、Max_LogYratio及びMin_LogYratioとして整数化部405に設定する。

【 0 0 4 4 】

次にステップS101にて、制御部1101は、画像入力部1106を制御し、メモリ308に格納されたHDR画像から、ブロック番号順に8*8画素のブロックの入力を行わせ、対数変換部402に実数の成分値であるLogYの生成処理を行わせる。

【 0 0 4 5 】

ステップS102にて、制御部1101は、バッファB403'内の着目ブロック内の全logYレシオが、その時点で整数化部405に設定したMin_LogYratioとMax_LogYratioが規定する範囲内にあるか否かを判定する。

【 0 0 4 6 】

範囲内にある場合には、制御部1101は処理をステップS105に進め、整数化部405に対して、その時点で設定されているMax_LogYratio及びMin_LogYratioに基づく整数化を行わせ、その結果をバッファD406'に格納していく。そして、制御部1101は、バッファD406'に格納された着目ブロックのYレシオを符号化部1103に供給する。符号化部1103は、供給された8*8個のYレシオを8*8個の輝度と見なし、JPEG符号化を行し、その符号をバッファ1104に出力する。

10

【 0 0 4 7 】

一方、バッファ403'に格納された着目ブロック内に、その時点で整数化部405に設定したMin_LogYratioとMax_LogYratioが規定する範囲外となるYレシオが存在した場合、制御部1101は処理をステップS102からステップS103に進める。このステップS103にて、制御部1101は範囲外となったlogYでMin_LogYratio、又は/及び、Max_LogYratioを更新する。例えば、着目ブロック内に、Max_LogYratioを超える1以上のLogYが存在した場合には、制御部1101は、その中の最大値を更新後の更新後のMax_LogYratioとして整数化部405に設定する。また、着目ブロック内に、Min_LogYratioを下回る1以上のLogYが存在した場合、制御部1101は、その中の最小値を更新後のMin_LogYratioとして整数化部405に設定する。そして、制御部1101は、整数化部405に対し、バッファB403'のLogYに対する整数化処理を行わせる。そして、ステップS104にて、制御部1101は、着目ブロックのブロック番号、更新後のMin_LogYratio及びMax_LogYratioを、例えばメモリ308の予め設定されたアドレス位置に記憶し、処理をステップS105に進める。なお、符号化を行うに先立って、このアドレス位置には、無効であると判別可能なブロック番号（例えば"-1"や最大ブロック番号を超える値等）を格納するものとする。また、ステップS101の判定でNoとなるたびに、このアドレス位置に格納された値は上書きされるものとする。

20

30

【 0 0 4 8 】

ステップS106にて、制御部1101は、着目ブロックが最終ブロックであるか否かを判定する。着目ブロックが最終ブロックである場合、制御部1101は処理をステップS107に進める。また、着目ブロックが最終ブロックでない場合、次のブロックの符号化を行うため、ブロック番号を1インクリメントした上で、処理をステップS101に戻す。

【 0 0 4 9 】

さて、ステップS107にて、制御部1101は、上記のYレシオの符号化処理にて、一度でもステップS103の更新処理を行ったか否かを判断する。具体的には、メモリ308の所定アドレスに格納されたブロック番号が有効か無効かを判定すればよい。メモリ308の所定アドレスに無効な値が格納されている場合、ステップS103に処理を一度も行っていないことを意味する。つまり、この段階でバッファ1104に格納されている符号化データは全てが有効であることになる。それ故、制御部1101はステップS109に処理を進め、出力部1105を制御し、バッファ1104に格納された符号化データをHDR差分データのYレシオの符号化データとして出力する。

40

【 0 0 5 0 】

一方、ステップS107にて、制御部1101がメモリ308の所定アドレス位置に有効なブロック番号"N"（0 N 149）が格納されていると判断した場合、バッファ1104に格納されている符号化データのうち、ブロック番号"N"～"149"までのブロックの符号化データは有効な符号化データであり、ブロック番号"0"から"N-1"までのブロックが無効な符号化データで

50

あることを意味する。それ故、制御部1101は、処理をステップS108に進める。このステップS1108において、制御部1101は、バッファ1104に格納されている無効な符号化データを破棄、もしくは上書き可能とする。そして、制御部1101は、その時点で整数化部405に設定されている最終Min_LogYratioとMax_LogYratioに従って、ブロック番号"0"から"N-1"までの再符号化を行わせるため、画像入力部1106に対して符号化対象のHDR画像におけるブロック番号"0"から"N-1"までのブロックを再入力を行わせ、その範囲のブロックの再符号化を行わせる。この結果、バッファ1104には、全ブロックの有効な符号化データが揃うことになる。それ故、制御部1101はステップS109にて、出力部1105を制御し、ブロック番号"0"乃至"N-1"、及び、ブロック番号"N"乃至"149"の符号化データの出力を行わせる。

【 0 0 5 1 】

10

上記の処理における具体例を図2(a)-(d)を参照して説明する。

【 0 0 5 2 】

符号化に先立ち、制御部1101は、整数化部405にLogYの最小値、最大値の初期値Ini_Min_LogYratio及びIni_Max_LogYratioを初期パラメータ保持部1102から読み出し、Max_LogYratio及びMin_LogYratioとして整数化部405に設定する。そして、左上隅のブロック200から符号化を開始する(図2(a))。

【 0 0 5 3 】

そして、符号化を行っている最中のブロック番号"23"のブロック202内に、Max_LogYratioを超える値Max_1が存在し、ブロック番号"55"のブロック203内にMin_LogYratioを下回るMin_1が存在したとする(図2(b))。

20

【 0 0 5 4 】

この場合、ブロック番号"0"からブロック"22"までブロック領域204は、{Min_LogYratio, Max_LogYratio} = {(Ini_Min_LogYratio, Ini_Max_LogYratio)}として処理される。

【 0 0 5 5 】

そして、ブロック番号"23"のブロック202を処理をしている時に、整数化部405に設定されるMax_LogYratioがMax_1で更新され、そのブロック番号"23"がメモリ308に保存される。そして、ブロック番号"23"からブロック番号"54"までのブロック領域205が{Min_LogYratio, Max_LogYratio} = {Ini_Min_LogYratio, Max_1}として処理される。そして、ブロック番号"55"のブロック内に、整数化部405に設定されるMin_LogYratioを下回る値Min_1が存在するので、整数化部405に設定されるMin_LogYratioがMin_1で更新され、且つ、ブロック番号"55"がメモリ308に保存される。そして、ブロック番号"55"以降、最終ブロックまでが、{Min_LogYratio, Max_LogYratio} = {Min_1, Max_1}として処理されることになる。

30

【 0 0 5 6 】

上記の状況では、ブロック番号"55"から"149"までは、{Min_LogYratio, Max_LogYratio} = {Min_1, Max_1}として有効なブロックの符号化データを得る。そして、先頭のブロック番号"0"からブロック番号"54"までが無効であるので、それらブロックについて{Min_LogYratio, Max_LogYratio} = {Min_1, Max_1}として整数化させ、再符号化処理が適用される。

【 0 0 5 7 】

40

以上から理解できるように、対数変換部402が生成するLogYの全てが、初期値{Ini_Min_LogYratio, Ini_Max_LogYratio}が規定する範囲内にある場合、最初のパスで生成される全ブロックの符号化データが有効なものとなる。しかも、図4に示したバッファB403、バッファC411のように大容量のメモリも必要としない。

【 0 0 5 8 】

また、仮に最初のパスのいずれかの、対数変換部402が、初期値{Ini_Min_LogYratio, Ini_Max_LogYratio}で規定する範囲外となるLogYを生成したとしても、整数化部405への設定値を最後に更新したブロック以降は有効な符号化データである。つまり、再符号化は、先頭から最後に更新したブロックの1つ前までの区間の部分的なものとしてでき、2パスになったとしても処理の遅延量は、これまでよりも少なくできる。

50

【 0 0 5 9 】

なお、図11では、バッファB403', バッファC411'を設ける例を示したが、これらを削除しても良い。図2(b)と同様、着目ブロックが、ブロック番号"23"のブロック202であった場合、更新パラメータに従った整数化が着目ブロックの次のブロック（ブロック番号"24"）から適用させればよい。これは、ブロック番号"55"についても同じである。つまり、最終的には、先頭のブロック番号"0"からブロック番号"55"までのブロック領域を、{Min_LogYratio、Max_LogYratio}={Min_1、Max_1}として再符号化すればよい。

【 0 0 6 0 】

上記はYレシオに対する説明であったが、図11におけるノーマライズ部410で得られる色差差分Cb_eやCr_eについても同じである。これは、以下に説明する実施形態でも同じである。

10

【 0 0 6 1 】

なお、上記実施形態では、HDR差分データ符号化部303内の符号化部1103がYレシオ、色差差分Cb_e、Cr_eを符号化するものとしたが、符号化部1103が結局のところ、これら3種類のデータが8ビットとしてJPEG符号化を行うものである。故に、符号化部1103と、LDR画像符号化部306を別々に設けるのではなく、一方のみを選択的に私用するようにしても良い。係る点は、以下の実施形態でも同じである。

【 0 0 6 2 】

[第2の実施形態]

上記第1の実施形態では、最小値・最大値の更新が発生した場合、逐次更新を行い処理を続行し、最後に再符号化が必要な領域に対して、最終の最小値・最大値で再符号化する事を説明した。本第2の実施形態では、最小値・最大値の更新が発生した場合、最小値・最大値の更新処理のみを、残りの領域に対して行う場合について説明する。

20

【 0 0 6 3 】

図6に本第2の実施形態での処理フローを示す。なお、図1の処理フローと同じ処理をする部分については、同じ番号を付けてある。また同じ処理でフロー上で異なる場所にあるものについては、枝番を付けてある。

【 0 0 6 4 】

本第2の実施形態におけるステップS100からステップS102までは、第1の実施形態と同じである。ステップS102の判断処理で範囲外となった場合の処理が第1の実施形態と異なる。ステップS102の判断処理で着目ブロック内に範囲外となるLogYがあると判断した場合、ステップS103に分岐し、以下の処理を行う。

30

【 0 0 6 5 】

まずステップS103では、制御部1101は、符号化部1103による符号化処理を停止させる。そして、制御部1101は、着目ブロック内のLogyで、その時点でのMin_LogYratio、Max_LogYratioを更新する。次にステップS106-2で最終Blockか否かを判断する。この判断方法は、ステップS106と同じである。ステップS106-2の判断で最終Blockではない場合、ステップS101-2に移る。ステップS101-2はステップS101と同じ処理である。次にステップS102-2はステップS101-2の演算結果が新しい最小値・最大値の範囲に有るか否かを判断する。判断方法は、ステップS102と同じである。範囲外の場合は、ステップS103-2に移動し最小値・最大値の更新を行ってステップS106-2に戻る。ステップS106-2で最終Blockまで処理が終わるとステップS108で、新しい最小値・最大値を初期値として、本フローを再処理する。

40

【 0 0 6 6 】

以上の結果、初期設定したMin_LogYratio、Max_LogYratioのいずれかが一度でも更新された場合は、最小値・最大値のみを更新し、符号化は行わないので、演算コストを下げる事ができる。また初期設定したMin_LogYratio、Max_LogYratioが更新されない場合には、1パス目で有効な符号化データを得ることができる。

【 0 0 6 7 】

[第3の実施形態]

50

上記第1、第2の実施形態では、再符号化する場合を説明したが、本実施形態では、再符号化をしない場合を説明する。

【0068】

図7を用いて説明する。図1と同じ処理については、同じ番号を付けてある。第1の実施形態との差は、ステップS102での範囲内か否かの判断の結果、範囲外となった場合の処理である。この場合、制御部1101は、ステップS700に処理を進め、強制的に、設定値の最小値或いはノ及び最大値でクリッピングする。

【0069】

この具体例を図8に示す。この例の場合は、最大値が範囲外になる場合を説明している。

10

【0070】

図8(a)はLogYデータの例である。例えば、最大値Max_LogYratioの初期値が「8.0」の場合、ブロック内の画素番号"0"から"25"までは最大値以下であるためそのまま出力する。画素番号"26"から"29"と"33"から"36"までは、最大値8.0を超えているためステップS700の処理でクリッピング処理され、これらの値は「8.0」になる。よってこのBlockの結果は、図8(b)に示すようになる。

【0071】

上の説明では最大値の場合を説明したが、最小値でも同じであり、最小値より小さい値が発生した場合は、最小値でクリッピングすれば良い。

【0072】

20

この処理により、第1、第2の実施形態で説明したような再符号化を含む再処理が必要なくなり、高速に処理することが可能となると共に、処理結果を溜めておくバッファも必要なくなる。

【0073】

[第3の実施形態の変形例]

上記第3の実施形態では、初期設定したMin_LogYratio、Max_LogYratioの範囲を越えるLogYが発生した場合、その範囲の上限または下限にクリッピングを行う事を説明したが、クリッピング処理をしない方法もある。例えば、第1の実施形態では、参照値Min_LogYratio、最大値Max_LogYratio、更新場所のデータは、再処理のためだけに使用し、最終の最小値・最大値で処理した結果を符号化するため、最終的なHDR差分符号データの中に格納する必要がなかった。しかし、まず更新した最小値・最大値と、更新したBlock番号・画素番号をHDR差分符号データ内の別の場所に保持する。次にデコード時にこの情報を読み出しながら、更新場所で最小値・最大値を更新しながらデコードすることを行うことで、クリッピングによる画質劣化を防ぐとができる。

30

【0074】

この時の処理フローを図9に示す。第1の実施形態等との差が有る部分のみを説明する。

【0075】

ステップS102の判断処理で、制御部1101は、範囲外と判断した場合に処理をステップS103に移り、最小値・最大値の更新を行う。次にステップS900でステップS103で更新した値と、更新の場所としてブロック番号と画素番号をメモリ308等に出力する。またステップS106の判断文で最終ブロックと判断された場合は、そのままこの処理を終了する。図示しないがメモリ308に出力されたLDR符号データ、HDR差分符号データを1つの符号データにする処理の時に、上記ステップS900で出力した値を符号データの一部として別の領域に書き出す。例えばHDR符号データがJPEGフォーマットの場合には、COMマーカやAPPマーカなどを使えばよい。これによりアプリケーション固有のデータとして管理する事ができ、デコードがこの情報を読み出し、上記の処理をすることで、最小値・最大値を更新した場合でもデコードすることが可能となる。

40

【0076】

この出力例を図10に示す。この例は、全体としてはJPEG形式で、LDR符号を主画像とするフォーマットである。図中の参照符号1000はS01マーカ、同1004はE01マーカでJPEG符号

50

データ列の先頭と最終を示すものである。参照符号1001は今回の更新情報である。参照符号1002はHDR差分符号データ列で、例えばAPPマーカを使ってLDR符号データ列の中に格納している。参照符号1003はLDR符号データ列であり、LDR符号データの画像サイズや符号化条件と実際の符号列が格納されている。この形式を取ることで、HDR差分データの存在を知らないアプリ等でも、LDR符号部分のみはデコードでき表示することができる。

【 0 0 7 7 】

更新情報1000の詳細が図示のAPPマーカ1005から示す部分である。参照符号1006～1007が各更新情報であり、最小値・最大値の更新が行われた個数(N)格納される。更新情報の内部は、参照符号1008～1011までであり、Block番号1008、画素番号1009、最小値1010、最大値1011である。

10

【 0 0 7 8 】

[第4の実施形態]

実施形態3までは、HDR差分データのデータ符号化を1組だけ使用したシーケンシャルに処理する事を説明した。本実施形態では、符号化対象のHDR差分データを副走査方向で複数の領域に分割(バンド分割)し、各バンドで独立して並列処理を行うことで処理の高速化を図る場合を説明する。つまり、図11におけるHDR差分画像の符号化に係る構成を複数有する。

【 0 0 7 9 】

例えば2個にバンド分割する場合は、最小値Min_LogYratio、最大値Max_LogYratioを2組(2バンド分)用意し、処理を行う。この2組の最小値・最大値の初期値は、同じ値でも良いし、異なる値でも良い。異なる値を入れる場合は、(Min_s0, Max_s0)と(Min_s1, Max_s1)で

20

$\text{Min_s1} < \text{Min_s0}, \text{Max_s1} > \text{Max_s0}$

の関係を持つ、すなわち(Min_s1, Max_s1)が(Min_s0, Max_s0)の範囲を包含するように設定する。この結果、Min_s1, Max_s1を初期値として設定するバンドの方で再符号化が発生する確率を下げる事ができる。そのため、再処理に必要な処理時間などを短縮することが可能となる。本実施形態の処理フローは、これまでの実施形態で説明した処理フローを必要な数、例えば2組、用意し、2組の処理部は独立して処理することが可能である。

【 0 0 8 0 】

また各処理部に与える最小値・最大値の初期値の決め方もこれに限るものではなく、様々な方法が考えられる。

30

【 0 0 8 1 】

なお領域分割の方法は、上記の方法に限るものではなく、例えばブロック番号0のブロックを処理部0、ブロック番号1を処理部1、ブロック番号2を処理部0、...のように交互に処理することも可能であり、様々な分割方法、処理方法が有る。

【 0 0 8 2 】

また最小値・最大値の初期値の決め方も、これまで説明した方法に限るものではなく、例えば、入力データを少しだけプリスキャンして、入力データの特徴を抽出してから決定するなど、色々な方法がある。プリスキャンする場所も入力データの先頭部分だけでないことは容易に推察できる。

40

【 0 0 8 3 】

以上説明したように、HDR差分データの符号化に利用するメモリ容量をこれまでよりも大幅に削減でき、しかも、符号化に係る処理時間を短縮させることができる。

【 0 0 8 4 】

(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、ASIC)によっても実現可能である。

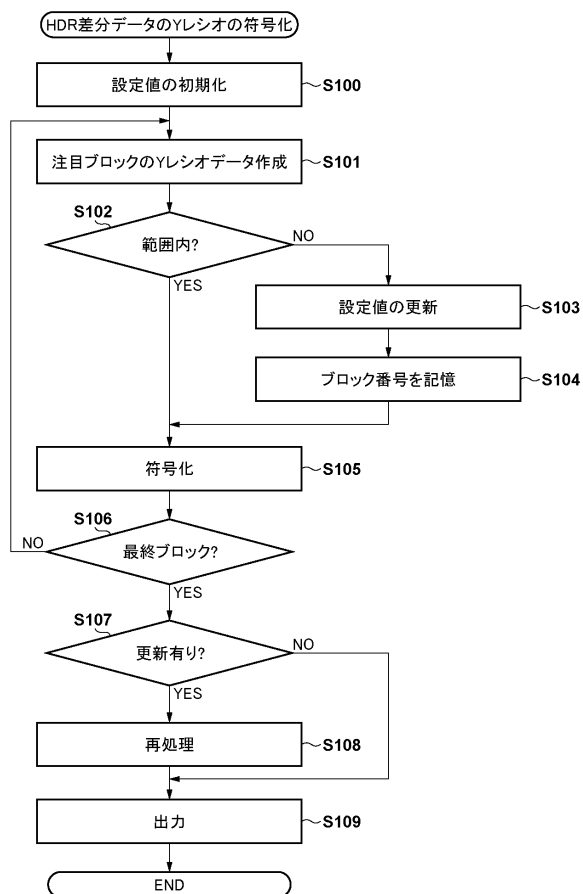
【符号の説明】

50

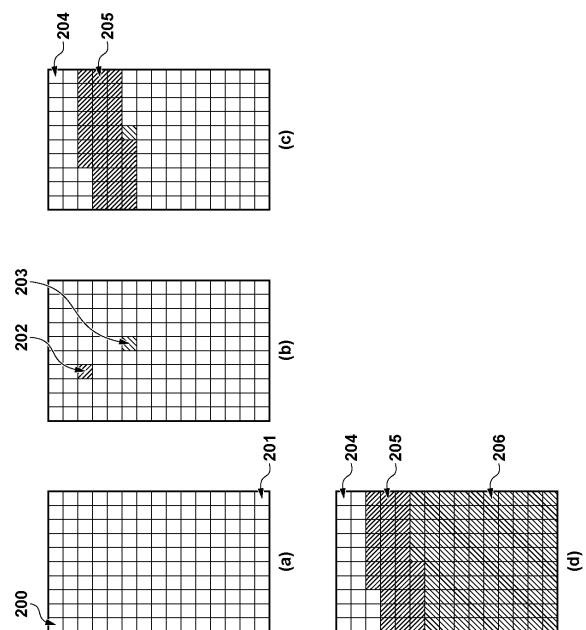
【 0 0 8 5 】

1101...制御部、1102...初期パラメータ保持部、1103...符号化部、1104...バッファ、1105...出力部、1106...画像入力部、401...輝度成分比算出部、402...対数変換部、405,413...整数化部、410...ノーマライズ部

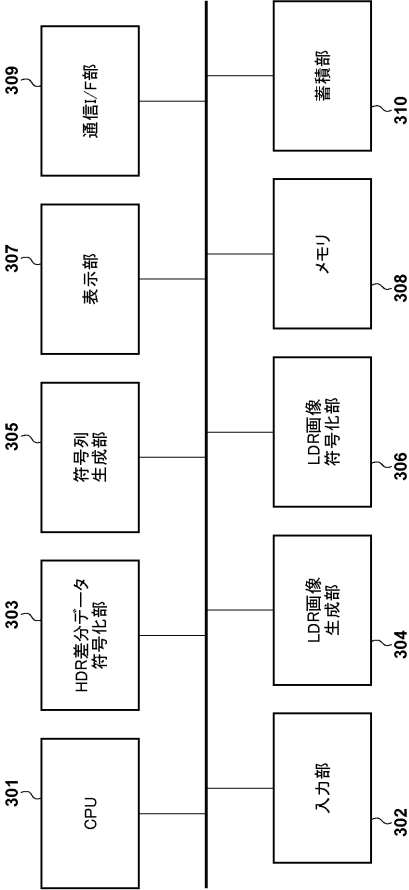
【 図 1 】



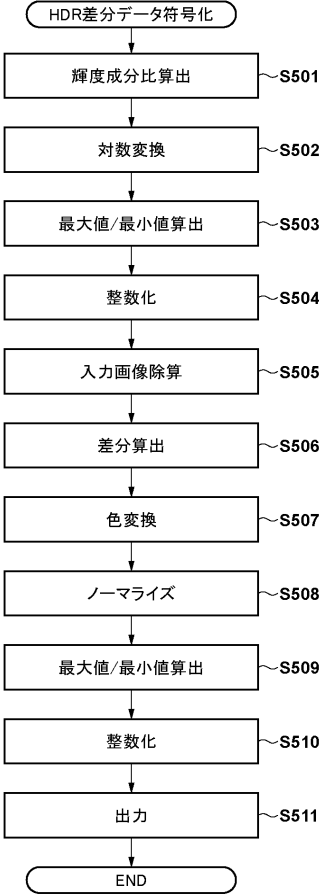
【 図 2 】



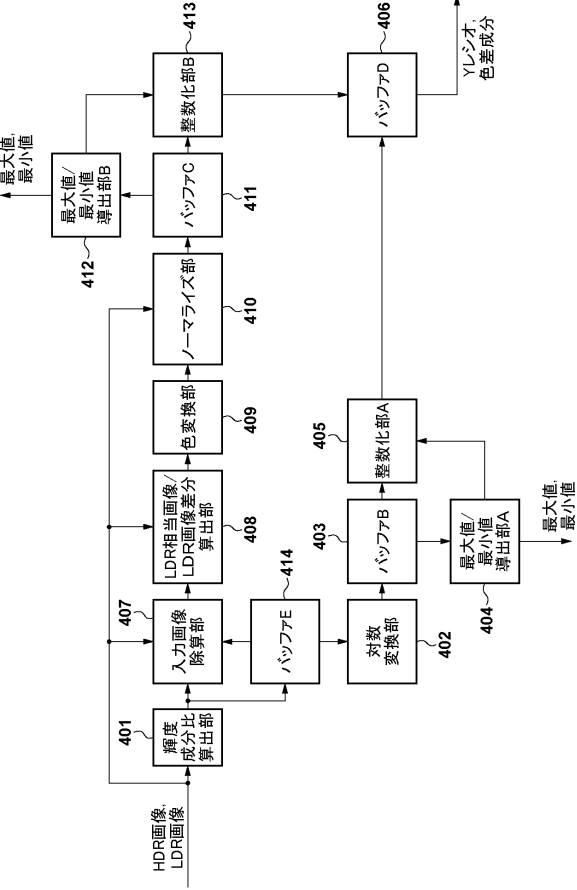
【図 3】



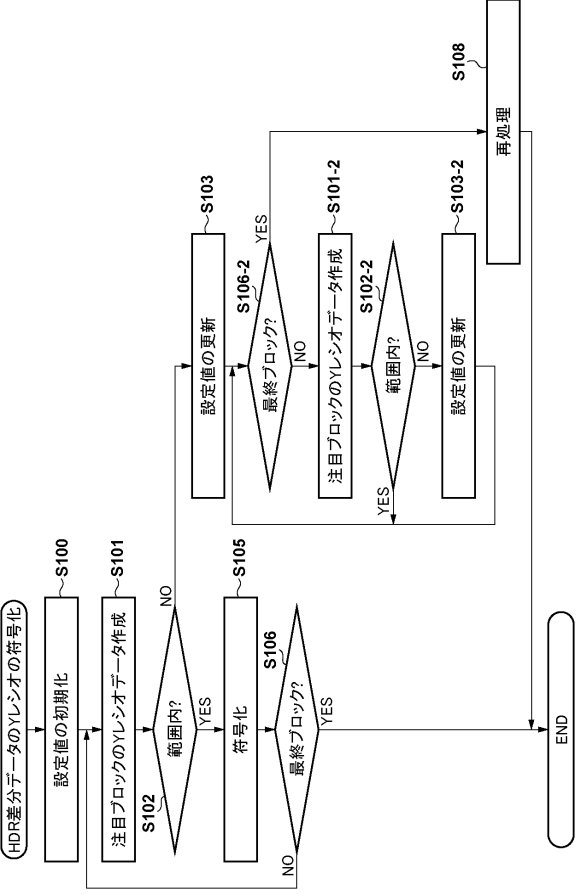
【図 5】



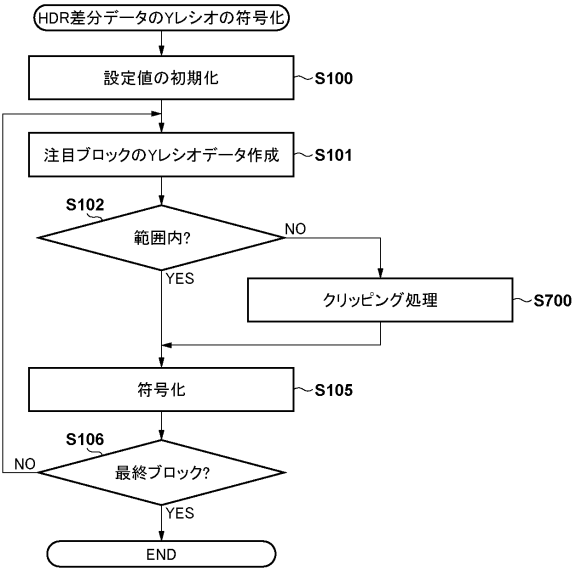
【図 4】



【図 6】



【図 7】



【図 8】

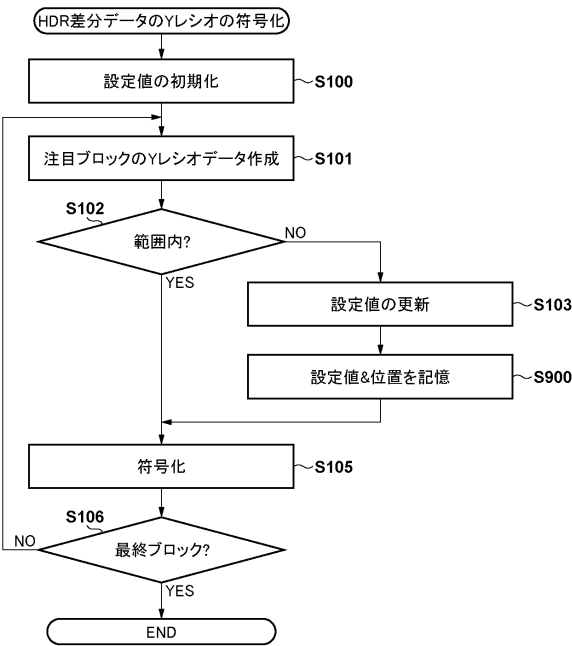
画素番号	LogY
0	1.54
1	1.82
2	2.15
...	...
25	7.77
26	8.89
27	9.94
28	11.01
29	8.55
30	7.65
31	7.79
32	7.88
33	10.19
34	11.22
35	9.32
36	8.57
37	7.23
...	...
62	7.65
63	6.21

(a)

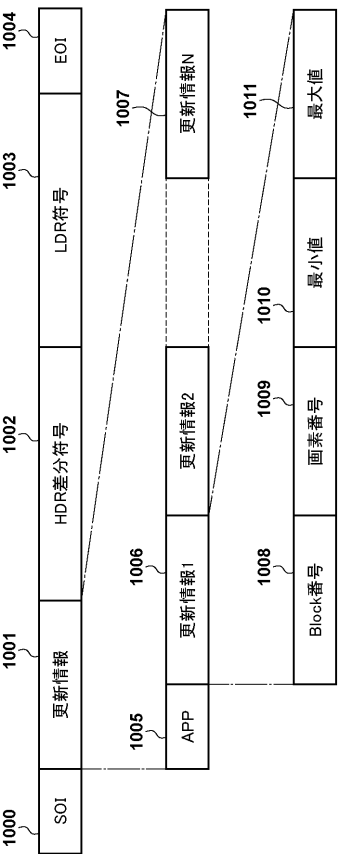
画素番号	LogY
0	1.54
1	1.82
2	2.15
...	...
25	7.77
26	8.00
27	8.00
28	8.00
29	8.00
30	7.65
31	7.79
32	7.88
33	8.00
34	8.00
35	8.00
36	8.00
37	7.23
...	...
62	7.65
63	6.21

(b)

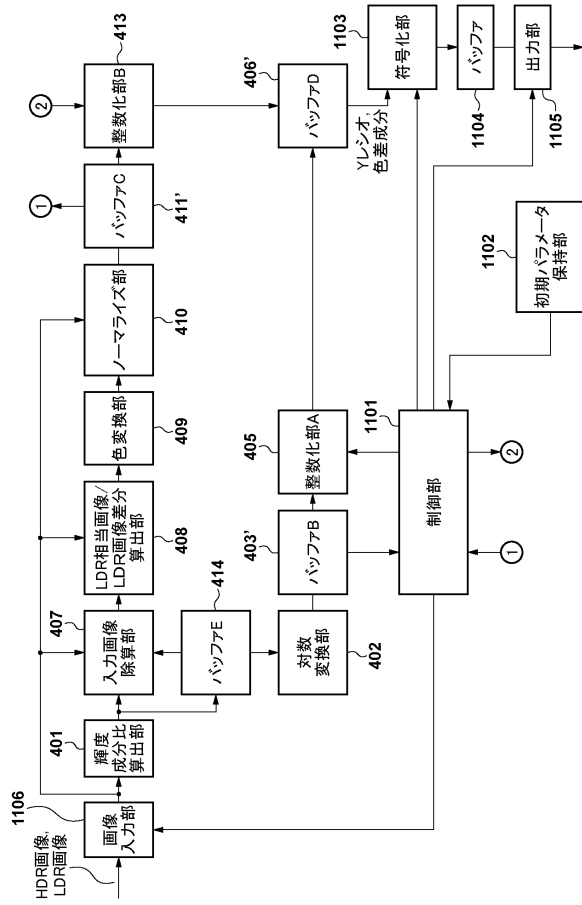
【図 9】



【図 10】



【図 11】



フロントページの続き

(56)参考文献 特開 2 0 1 5 - 0 1 9 2 8 6 (J P , A)
特表 2 0 1 7 - 5 0 8 3 3 2 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

H 0 4 N 1 / 4 1 3

H 0 4 N 1 9 / 3 6

H 0 4 N 1 9 / 4 3 6