

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7146407号
(P7146407)

(45)発行日 令和4年10月4日(2022.10.4)

(24)登録日 令和4年9月26日(2022.9.26)

(51)国際特許分類

H 0 4 N	5/92 (2006.01)	F I	H 0 4 N	5/92	
G 1 1 B	20/10 (2006.01)		G 1 1 B	20/10	3 0 1 Z
G 1 1 B	27/00 (2006.01)		G 1 1 B	27/00	D
H 0 4 N	19/85 (2014.01)		H 0 4 N	19/85	
H 0 4 N	5/926(2006.01)		H 0 4 N	5/926	

請求項の数 22 (全31頁)

(21)出願番号 特願2018-26308(P2018-26308)
 (22)出願日 平成30年2月16日(2018.2.16)
 (65)公開番号 特開2019-145918(P2019-145918)
 A)
 (43)公開日 令和1年8月29日(2019.8.29)
 審査請求日 令和3年2月10日(2021.2.10)

(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
 (74)代理人 100126240
 弁理士 阿部 琢磨
 100124442
 弁理士 黒岩 創吾
 小林 達
 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
 秋吉 秀信
 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
 田邊 章弘
 東京都大田区下丸子3丁目30番2号キ
 最終頁に続く

(54)【発明の名称】 撮像装置、表示制御装置及び表示制御装置の制御方法

(57)【特許請求の範囲】**【請求項1】**

撮像センサにより画像を撮影して、撮影した画像を記録する撮像装置であって、前記撮像センサにより撮影して得られた画像データを符号化する符号化手段と、前記撮像センサより撮影して得られた画像データのうち、再生対象となる再生領域よりも大きい領域を符号化領域として設定し、前記符号化領域の画像データを前記符号化手段により符号化し、前記符号化された画像データを所定の記録フォーマットで記録媒体に記録するように制御する制御手段と、を有し、

前記制御手段は、前記撮像センサにより画像データを取得可能なセンサ画像領域のうち、一部の領域を前記符号化領域として設定し、当該符号化領域の画像データを前記符号化手段により符号化するように制御し、

前記センサ画像領域において、符号化開始／終了が可能な位置に制約があり、前記制御手段は、前記センサ画像領域において、前記再生領域と前記符号化開始／終了が可能な位置とに基づいて、前記符号化領域の開始位置及び終了位置を設定する、ことを特徴とする撮像装置。

【請求項2】

前記制御手段は、前記符号化領域の画像データを、複数の分割画像データに分割し、前記複数の分割画像データを前記符号化手段によりそれぞれ符号化し、前記符号化された複数の分割画像データを所定の記録フォーマットで1つの画像ファイルとして前記記録媒体に記録するように制御することを特徴とする請求項1に記載の撮像装置。

【請求項 3】

記録サイズを設定するための設定手段を有し、

前記制御手段は、前記設定手段により設定された記録サイズに応じて、前記再生領域および前記符号化領域を決定し、

前記再生領域のサイズは、前記設定された記録サイズに対応する領域であることを特徴とする請求項 1 に記載の撮像装置。

【請求項 4】

前記所定の記録フォーマットは、記録される画像データのうち、再生対象とする領域を指定可能な記録フォーマットであり、

前記制御手段は、記録される前記符号化領域よりも小さい前記再生領域を、再生対象の領域として指定することを特徴とする請求項 1 に記載の撮像装置。 10

【請求項 5】

前記制御手段は、前記再生領域の画像データに対応するサムネイル画像データを前記符号化手段により符号化し、前記符号化したサムネイル画像データを前記符号化した複数の分割画像データと共に前記所定の記録フォーマットで 1 つの画像ファイルとして記録するように制御することを特徴とする請求項 1 に記載の撮像装置。

【請求項 6】

前記符号化手段は、HEVC (High Efficiency Video Coding) 規格に従って前記画像データを圧縮符号化することを特徴とする請求項 1 に記載の撮像装置。 20

【請求項 7】

前記所定の記録フォーマットは、HEIF (High Efficiency Image File Format) であることを特徴とする請求項 1 に記載の撮像装置。

【請求項 8】

前記所定の記録フォーマットは、HEIF (High Efficiency Image File Format) であり、前記制御手段は、前記複数の分割画像データを、それぞれ画像アイテムとして記録し、前記複数の分割画像データから構成される画像を導出画像アイテムとして記録するように制御することを特徴とする請求項 2 に記載の撮像装置。

【請求項 9】

前記制御手段は、前記複数の分割画像データを、オーバーレイ形式の導出画像アイテムとして記録することを特徴とする請求項 8 に記載の撮像装置。 30

【請求項 10】

前記制御手段は、前記導出画像アイテムにおいて、前記再生領域のみが再生されるよう、再生領域を指定することを特徴とすることを特徴とする請求項 8 または 9 に記載の撮像装置。

【請求項 11】

前記制御手段は、前記導出画像アイテムにおいて、前記複数の分割画像データが、前記符号化領域に対応する位置関係となるように配置されるように、前記複数の分割画像データそれぞれの位置情報を記録するように制御することを特徴とする請求項 8 乃至 10 のいずれか 1 項に記載の撮像装置。 40

【請求項 12】

前記制御手段は、前記複数の分割画像データがそれぞれ重ならない位置となるように前記複数の分割画像データそれぞれの位置情報を記録することを特徴とする請求項 11 に記載の撮像装置。

【請求項 13】

前記制御手段は、前記所定の記録フォーマットにおいて、前記導出画像アイテムを主画像として記録することを特徴とする請求項 8 乃至 12 のいずれか 1 項に記載の撮像装置。

【請求項 14】

前記制御手段における、前記センサ画像領域における符号化開始可能な位置は、前記センサ画像領域の幅または高さ方向において、所定画素数単位の位置であることを特徴とする請求

項 1 乃至 1 3 のいずれか 1 項に記載の撮像装置。

【請求項 1 5】

前記制約において、前記センサ画像領域の幅及び高さ方向それぞれにおいて、所定画素数単位の位置が前記符号化開始可能な位置として予め決められており、

前記センサ画像領域の幅または高さ方向とで、前記所定画素数が異なる、ことを特徴とする請求項 1 4 に記載の撮像装置。

【請求項 1 6】

前記所定画素数は、前記高さ方向よりも前記幅方向のほうが大きいことを特徴とする請求項 1 5 に記載の撮像装置。

【請求項 1 7】

前記制御手段は、前記センサ画像領域内における前記再生領域の開始位置を前記所定画素数で除算したときの商に、前記所定画素数を乗じた位置を、前記符号化領域の開始位置に設定する、ことを特徴とする請求項 1 4 に記載の撮像装置。

10

【請求項 1 8】

前記制御手段は、前記センサ画像領域内における前記再生領域の開始位置が、前記センサ画像領域における符号化開始が可能な位置である場合は、前記再生領域の開始位置を前記符号化領域の開始位置に設定することを特徴とする請求項 1 乃至 1 7 のいずれか 1 項に記載の撮像装置。

【請求項 1 9】

前記制約は、前記センサ画像領域における符号化開始可能な位置の制約である開始位置制約と、前記符号化領域の幅 / 高さの制約である符号化サイズ制約とが含まれている、ことを特徴とする請求項 1 乃至 1 8 のいずれか 1 項に記載の撮像装置。

20

【請求項 2 0】

前記制御手段は、前記再生領域の開始位置と前記符号化開始が可能な位置とに基づいて前記符号化領域の開始位置を設定し、前記符号化領域の開始位置と、前記符号化サイズ制約とから、前記符号化領域の終了位置を設定する、ことを特徴とする請求項 1 9 に記載の撮像装置。

【請求項 2 1】

撮像センサにより撮影した画像を記録する記録装置の制御方法であって、

前記撮影して得られた画像データを符号化する符号化工程と、

前記撮影して得られた画像データのうち、再生対象となる再生領域よりも大きい領域を符号化領域として設定し、前記符号化領域の画像データを前記符号化工程で符号化し、前記符号化された画像データを所定の記録フォーマットで記録媒体に記録するように制御する制御工程と、を有し、

30

前記制御工程では、前記撮像センサにより画像データを取得可能なセンサ画像領域のうち、一部の領域を前記符号化領域として設定し、当該符号化領域の画像データを前記符号化工程で符号化するように制御し、

前記センサ画像領域において、符号化開始 / 終了が可能な位置に制約があり、

前記制御工程では、前記センサ画像領域において、前記再生領域と前記符号化開始 / 終了が可能な位置とに基づいて、前記符号化領域の開始位置及び終了位置を設定する、ことを特徴とする記録装置の制御方法。

40

【請求項 2 2】

コンピュータを、請求項 1 乃至 2 0 のいずれか 1 項に記載の撮像装置の各手段として機能させることを特徴とするプログラム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、撮像装置及び、表示制御装置表示制御装置の制御方法に関する。

【背景技術】

【0 0 0 2】

50

画像データを圧縮符号化する画像処理装置として撮像装置が知られている。このような画像処理装置は、撮像部によって動画像信号を取得し、取得した動画像信号を圧縮符号化し、圧縮符号化が行われた画像ファイルを記録媒体に記録する。従来、圧縮符号化前の画像データは 100 nits の輝度レベルを上限とする SDR (Standard Dynamic Range) で表現されたものであった。しかしながら、近年では、輝度レベルの上限を 10000 nits 程度にまで拡張させた HDR (High Dynamic Range) で表現され、人が知覚できる輝度範囲に近い輝度範囲を有する画像データが提供されている。

【0003】

特許文献 1 では、HDR の画像を撮影して記録する際に、HDR に非対応の装置でも、HDR 画像を詳細に確認可能な画像データを生成して記録する画像データ記録装置について記載がある。10

【先行技術文献】

【特許文献】

【0004】

【文献】特開 2017 - 139618 号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

特許文献 1 では、HDR の画像を記録することについては記載されているが、HDR 画像を記録する場合の最適な記録方式については考慮されていなかった。20

【0006】

そこで、本発明は、例えば HDR 画像のようにデータ量が大きい画像を記録する際に、記録および再生に適した記録形態で画像を記録する装置、また、その記録形態で記録された画像を再生するための表示制御装置を提供することを目的とする。

【0007】

また、本発明は、画像を符号化して記録する装置において、符号化時に各種制約があったとしても、記録および再生に適した記録形態で画像を記録する装置、また、その記録形態で記録された画像を再生するための表示制御装置を提供することを目的とする。

【課題を解決するための手段】

【0008】

上述の課題を解決するため、本発明の撮像装置は、

撮像センサにより画像を撮影して、撮影した画像を記録する撮像装置であって、撮像センサにより撮影して得られた画像データを符号化する符号化手段と、撮像センサより撮影して得られた画像データのうち、再生対象となる再生領域よりも大きい領域を符号化領域として設定し、符号化領域の画像データを符号化手段により符号化し、符号化された画像データを所定の記録フォーマットで記録媒体に記録するように制御する制御手段と、を有し、制御手段は、撮像センサにより画像データを取得可能なセンサ画像領域のうち、一部の領域を符号化領域として設定し、当該符号化領域の画像データを符号化手段により符号化するように制御し、センサ画像領域において、符号化開始 / 終了が可能な位置に制約があり、制御手段は、センサ画像領域において、再生領域と符号化開始 / 終了が可能な位置に基づいて、符号化領域の開始位置及び終了位置を設定する、ことを特徴とする。40

【0009】

また、本発明の表示制御装置は、
再生対象となる再生領域よりも大きい符号化領域の画像データを、複数の分割画像データに分割し、複数の分割画像データをそれぞれ符号化し、符号化された複数の分割画像データが所定の記録フォーマットで 1 つの画像ファイルとして記録されている記録媒体から、画像ファイルを読み出す読み出し手段と、読み出した画像ファイルに含まれる複数の分割画像データをそれぞれ複合する複合手段と、復号手段により復号された複数の分割画像データを 1 つの画像データに結合し、結合された画像データのうち、再生領域の画像データ50

を表示手段に表示するように制御する表示制御手段と、を有することを特徴とする。

【発明の効果】

【0010】

本発明によれば、データ量が大きく、かつ、高解像度の画像を記録する際に、記録および再生に適した記録形態で画像を記録する撮像装置、また、その記録形態で記録された画像を再生するための表示制御装置を提供することができる。

【0011】

本発明によれば、画像を符号化して記録する装置において、符号化時に各種制約があつたとしても、記録および再生に適した記録形態で画像を記録する装置、また、その記録形態で記録された画像を再生するための表示制御装置を提供することができる。

10

【図面の簡単な説明】

【0012】

【図1】撮像装置100の構成を示すブロック図。

【図2】HEIFファイルの構造を示す図。

【図3】HDR撮影モードでの処理を示すフローチャート。

【図4】HDR撮影モードにおける画像データの分割方法を決定する処理を示すフローチャート。

【図5】HDR画像データ記録時の画像データの符号化領域および分割方法を示す図。

【図6】HEIFファイルを構築する処理を示すフローチャート。

【図7】HEIFファイルとして記録されたHDR画像データを再生する際の表示処理を示すフローチャート。

20

【図8】オーバーレイ画像作成処理を示すフローチャート。

【図9】画像アイテムのプロパティ取得処理を示すフローチャート。

【図10】画像アイテムのデータ取得処理を示すフローチャート。

【図11】主画像を構成する画像のアイテムID取得処理を示すフローチャート。

【図12】1画像アイテムの画像作成処理を示すフローチャート。

【図13】オーバーレイ画像と分割画像の位置関係を示す図。

【発明を実施するための形態】

【0013】

本発明の実施の形態を、撮像装置100を例に添付の図面を参照して以下に詳細に説明するが、本発明は、以下の実施形例に限定されるものではない。

30

【0014】

<撮像装置の構成>

図1は、撮像装置100を示すブロック図である。図1に示すように、撮像装置100は、CPU101と、メモリ102と、不揮発性メモリ103と、操作部104と、撮像部112と、画像処理部113と、符号化処理部114と、表示制御部115と、表示部116とを有している。更に、撮像装置100は、通信制御部117と、通信部118と、記録媒体制御部119と、内部バス130とを有している。撮像装置100は、撮影レンズ111を用いて被写体の光学像を撮像部112の画素アレイに結像するが、撮影レンズ111は、撮像装置100のボディ(筐体、本体)から、着脱不能であってもよいし、着脱可能であってもよい。また、撮像装置100は、記録媒体制御部119を介して画像データの書き込み及び読み出しを記録媒体120に対して行う。記録媒体120は、撮像装置100に着脱可能であってもよいし、着脱不能であってもよい。

40

【0015】

CPU101は、不揮発性メモリ103に記憶されているコンピュータプログラムを実行することによって、内部バス130を介して撮像装置100の各部(各機能ブロック)の動作を制御する。

【0016】

メモリ102は、書き換え可能な揮発性メモリである。メモリ102は、撮像装置100の各部の動作を制御するためのコンピュータプログラム、撮像装置100の各部の動作

50

に関するパラメータ等の情報、通信制御部 117 によって受信される情報等を一時的に記録する。また、メモリ 102 は、撮像部 112 によって取得された画像、画像処理部 113、符号化処理部 114 等によって処理された画像及び情報を一時的に記録する。メモリ 102 は、これらを一時的に記録するために十分な記憶容量を備えている。

【0017】

不揮発性メモリ 103 は、電気的に消去及び記録が可能なメモリであり、例えば EEPROM 等が用いられる。不揮発性メモリ 103 は、撮像装置 100 の各部の動作を制御するコンピュータプログラム及び撮像装置 100 の各部の動作に関するパラメータ等の情報を記憶する。当該コンピュータプログラムにより、撮像装置 100 によって行われる各種動作が実現される。

10

【0018】

操作部 104 は、撮像装置 100 を操作するためのユーザインターフェースを提供する。操作部 104 は、電源ボタン、メニューボタン、撮影ボタン等の各種ボタンを含んでおり、各種ボタンはスイッチ、タッチパネル等により構成される。CPU 101 は、操作部 104 を介して入力されたユーザの指示に従って撮像装置 100 を制御する。なお、ここでは、操作部 104 を介して入力される操作に基づいて CPU 101 が撮像装置 100 を制御する場合を例に説明したが、これに限定されるものではない。例えば、不図示のリモートコントローラ、不図示の携帯端末等から通信部 118 を介して入力される要求に基づいて、CPU 101 が撮像装置 100 を制御してもよい。

【0019】

撮影レンズ（レンズユニット）111 は、ズームレンズ、フォーカスレンズ等を含む不図示のレンズ群、不図示のレンズ制御部、不図示の絞り等によって構成される。撮影レンズ 111 は、画角を変更するズーム手段として機能し得る。レンズ制御部は、CPU 101 から送信される制御信号により、焦点の調整及び絞り値（F 値）の制御を行う。撮像部 112 は、動画像を構成する複数の画像を順次取得する取得手段として機能し得る。撮像部 112 としては、例えば、例えば CCD（電荷結合素子）、CMOS（相補型金属酸化膜半導体）素子等でのエリアイメージセンサが用いられる。撮像部 112 は、被写体の光学像を電気信号に変換する不図示の光電変換部が行列状、すなわち、2 次元的に配列された不図示の画素アレイを有している。当該画素アレイには、被写体の光学像が撮影レンズ 111 によって結像される。撮像部 112 は、撮像した画像を画像処理部 113 又はメモリ 102 に出力する。なお、撮像部 112 は、静止画像を取得することも可能である。

20

【0020】

画像処理部 113 は、撮像部 112 から出力される画像データ、又は、メモリ 102 から読み出された画像データに対し、所定の画像処理を行う。当該画像処理の例としては、補間処理、縮小処理（リサイズ処理）、色変換処理等が挙げられる。また、画像処理部 113 は、撮像部 112 によって取得された画像データを用いて、露光制御、測距制御等のための所定の演算処理を行う。画像処理部 113 による演算処理によって得られた演算結果に基づいて、露光制御、測距制御等が CPU 101 によって行われる。具体的には、A E（自動露出）処理、A W B（オートホワイトバランス）処理、A F（オートフォーカス）処理等が CPU 101 によって行われる。

30

【0021】

符号化処理部 114 は、画像データに対してフレーム内予測符号化（画面内予測符号化）、フレーム間予測符号化（画面間予測符号化）等を行うことによって、画像データのサイズを圧縮する。符号化処理部 114 は、例えば、半導体素子等により構成された符号化装置である。符号化処理部 114 は、撮像装置 100 の外部に設けられた符号化装置であつてもよい。符号化処理部 114 は、例えば、H.265（ITU H.265 又は ISO / IEC 23008 - 2）方式によって符号化処理を行う。

40

【0022】

表示制御部 115 は、表示部 116 を制御する。表示部 116 は、不図示の表示画面を備える。表示制御部 115 は、画像データに対してリサイズ処理、色変換処理等を行うこ

50

とにより、表示部 116 の表示画面に表示可能な画像を生成し、当該画像、すなわち、画像信号を表示部 116 に出力する。表示部 116 は、表示制御部 115 から送られてくる画像信号に基づいて、表示画面に画像を表示する。表示部 116 は、表示画面にメニュー等の設定画面を表示する機能である OSD (On Screen Display) 機能を備えている。表示制御部 115 は、画像信号に OSD 画像を重畠して表示部 116 に画像信号を出力し得る。表示部 116 は、液晶ディスプレイ、有機 EL ディスプレイ等により構成されており、表示制御部 115 から送られてきた画像信号を表示する。表示部 116 は、例えばタッチパネルであってもよい。表示部 116 がタッチパネルである場合、表示部 116 は、操作部 104 としても機能し得る。

【0023】

通信制御部 117 は、CPU 101 に制御される。通信制御部 117 は、IEEE 802.11 等によりあらかじめ定められた無線通信規格に適合する変調信号を生成して、当該変調信号を通信部 118 に出力する。また、通信制御部 117 は、無線通信規格に適合する変調信号を、通信部 118 を介して受信し、受信した変調信号を復号し、復号した信号に応じた信号を CPU 101 に出力する。通信制御部 117 は、通信設定を記憶するためのレジスタを備えている。通信制御部 117 は、CPU 101 からの制御によって、通信時の送受信感度を調整し得る。通信制御部 117 は、所定の変調方式で送受信を行うことができる。通信部 118 は、通信制御部 117 から供給される変調信号を撮像装置 100 の外部にある情報通信機器等の外部機器 127 へ出力し、また、外部機器 127 からの変調信号を受信するアンテナを備えている。また、通信部 118 には、通信用の回路等が備えられている。なお、ここでは、通信部 118 によって無線通信が行われる場合を例に説明したが、通信部 118 によって行われる通信は無線通信に限定されるものではない。例えば、配線等を用いた電気的な接続によって通信部 118 と外部機器 127 とが接続されてもよい。

【0024】

記録媒体制御部 119 は、記録媒体 120 を制御する。記録媒体制御部 119 は、CPU 101 からの要求に基づいて、記録媒体 120 を制御するための制御信号を記録媒体 120 に出力する。記録媒体 120 としては、例えば不揮発性メモリや磁気ディスク等が用いられる。記録媒体 120 は、上述したように、着脱可能であってもよいし、着脱不能であってもよい。記録媒体 120 は、符号化された画像データ等を記録する。記録媒体 120 のファイルシステムに適合した形式で画像データ等がファイルとして保存される。ファイルとしては、例えば、MP4 ファイル (ISO / IEC 14496 - 14 : 2003)、MXF (Material Exchange Format) ファイル等が挙げられる。各々の機能ブロック 101 ~ 104、112 ~ 115、117、119 は、内部バス 130 を介して互いにアクセス可能となっている。

【0025】

ここで、本実施例の撮像装置 100 の通常の動作について説明する。

【0026】

撮像装置 100 は、ユーザが操作部 104 の電源ボタンを操作すると、操作部 104 から CPU 101 に起動の指示が出される。この指示を受けて、CPU 101 は、不図示の電源供給部を制御して、撮像装置 100 の各ブロックに対して電源を供給させる。電源が供給されると、CPU 101 は、例えば、操作部 104 のモード切り替えスイッチが、例えば、静止画撮影モード、再生モード等のどのモードであるかを操作部 102 からの指示信号により確認する。

【0027】

通常の静止画撮影モードでは、撮像装置 100 は、撮影待機状態でユーザが操作部 104 の静止画記録ボタンを操作することで撮影処理を行う。撮影処理では、撮像部 112 で撮影した静止画の画像データについて、画像処理部 113 で画像処理を施し、符号化処理部 114 で符号化処理し、記録媒体制御部 119 により符号化された画像データを画像ファイルとして記録媒体 120 に記録する。なお、撮影待機状態では、撮像部 112 で所定

10

20

30

40

50

のフレームレートで画像を撮影し、画像処理部で表示用の画像処理を施して表示制御部 115により表示部 116に表示させることによりライブビュー画像の表示を行う。

【0028】

再生モードでは、記録媒体 120に記録されている画像ファイルを記録媒体制御部 119により読み出し、読み出した画像ファイルの画像データを符号化処理部 114で復号する。つまり、符号化処理部 114はデコーダの機能も有している。そして、画像処理部 113で表示用の処理を施し、表示制御部 115により表示部 116に表示させる。

【0029】

通常の静止画の撮影および再生については、上記のように実行されるが、本実施形例の撮像装置では、通常の静止画だけでなく、HDR静止画を撮影するためのHDR撮影モードを有し、また、撮影したHDR静止画を再生することができる。

10

【0030】

以降、HDR静止画像の撮影および再生の処理について説明する。

【0031】

<ファイル構造>

まずは、HDR静止画像を記録する際のファイル構造について説明する。

【0032】

最近、High Efficiency Image File Format(以下、HEIFと称す)と呼ばれる静止画ファイルフォーマットが策定された。(ISO/IEC 23008-12:2017)

20

【0033】

これは、JPEGのような従来の静止画ファイルフォーマットと比べ、以下のような特徴を持つ。

- ・ISOベースメディアファイルフォーマット(以下、ISOBMFFと称す)に準拠したファイル形式である。(ISO/IEC 14496-14:2003)
- ・単一の静止画像だけでなく、複数の静止画像を格納できる。
- ・HEVC/H.265やAVC/H.264など、動画の圧縮に用いられる圧縮形式で圧縮された静止画を格納できる。

【0034】

本実施形例では、HDR静止画像の記録ファイルとして、HEIFを採用する。

30

【0035】

最初に、HEIFが格納するデータについて説明する。

【0036】

HEIFは、格納する個々のデータを、アイテムという単位で管理する。

【0037】

各アイテムは、データ自身の他に、ファイル内で一意となる整数値アイテムID(item_ID)と、アイテムの種別を表すアイテムタイプ(item_type)を持つ。

【0038】

アイテムは、データが画像を表す画像アイテム(image_item)と、データがメタデータであるメタデータアイテム(metadata_item)に分けることができる。

40

【0039】

画像アイテムには、データが符号化された画像データである符号化画像アイテム(encoded_image_item)と、1個以上の他の画像アイテムを操作した結果の画像を表す、導出画像アイテム(derived_image_item)がある。

【0040】

導出画像アイテムの例として、オーバーレイ形式の導出画像アイテムであるオーバーレイ画像がある。これは、ImageOverlay構造体(オーバーレイ情報)用いて、任意個の画像アイテムを任意の位置に配置して合成された結果のオーバーレイ画像である。

【0041】

50

メタデータアイテムの例として、`E x i f` データを格納することができる。

【0042】

前述のように、`H E I F` は複数の画像アイテムを格納することができる。

【0043】

複数の画像間に関係がある場合、それらの関係を記述することができる。

【0044】

複数の画像間の関係の例は、導出画像アイテムとそれを構成する画像アイテムの関係、本画像とサムネイル画像の関係、などである。

【0045】

また、画像アイテムとメタデータアイテムの関係も、同様に記述することができる。 10

【0046】

`H E I F` 形式は、`I S O B M F F` 形式に準拠している。そのため、まず、`I S O B M F` Fについて、簡単に説明する。

【0047】

`I S O B M F F` 形式は、ボックスと呼ばれる構造でデータを管理する。

【0048】

ボックスは、4バイトのデータ長フィールドと、4バイトのデータタイプフィールドから始まり、それ以降に任意長のデータを持つデータ構造である。

【0049】

データ部の構造は、データタイプによって決定する。`I S O B M F F` の仕様書や`H E I F` の仕様書では、いくつかのデータタイプ及びそのデータ部の構造が規定されている。 20

【0050】

また、ボックスは、別のボックスをデータに持つこともある。すなわち、ボックスの入れ子が可能である。ここで、あるボックスのデータ部に入れ子にされたボックスを、サブボックスと呼ぶ。

【0051】

また、サブボックスではないボックスを、ファイルレベルのボックスと呼ぶ。これは、ファイルの先頭から、順次アクセス可能なボックスである。

【0052】

図2を用いて、`H E I F` 形式のファイルについて説明する。 30

【0053】

まず、ファイルレベルのボックスについて説明する。

【0054】

データタイプが '`f t y p`' であるファイルタイプボックスには、ファイルの互換性に関する情報を格納する。`I S O B M F F` 準拠のファイル仕様は、その仕様が規定するファイルの構造や、ファイルが格納するデータを、ブランドと呼ぶ4バイトのコードで宣言し、それらをファイルタイプボックスに格納する。ファイルの先頭にファイルタイプボックスを配置することにより、ファイルのリーダは、ファイルタイプボックスの内容をチェックすることで、さらにファイルの内容を読み出して解釈することなく、ファイルの構造を知ることができる。 40

【0055】

`H E I F` の仕様では、「`m i f 1`」というブランドでそのファイル構造を表す。また、格納する符号化画像データが`H E V C` 圧縮されたデータの場合、`H E V C` 圧縮のプロファイルに従って、「`h e i c`」または「`h e i x`」というブランドで表す。

【0056】

データタイプが '`m e t a`' であるメタデータボックスは、いろいろなサブボックスを含み、各アイテムに関するデータを格納する。内訳については、後述する。

【0057】

データタイプが '`m d a t`' であるメディアデータボックスには、各アイテムのデータを格納する。例えば、符号化画像アイテムの符号化画像データや、メタデータアイテムの E 50

`xif` データ、である。

【0058】

次に、メタデータボックスのサブボックスについて説明する。

【0059】

データタイプが '`hdlr`' であるハンドラボックスは、メタデータボックスが管理するデータのタイプを表す情報を格納する。HEIF仕様では、ハンドラボックスの `handler_type` は '`pict`' である。

【0060】

データタイプが '`dinf`' であるデータ情報ボックスは、このファイルが対象とするデータが存在するファイルを指定する。ISOBMFFでは、あるファイルが対象とするデータを、そのファイル以外のファイルに格納することが可能である。その場合、データ情報ボックス内のデータリファレンスに、データが存在するファイルの URL を記述したデータエントリ URL ボックスを格納する。対象とするデータが同じファイルに存在する場合、それを表すフラグのみを有するデータエントリ URL ボックスを格納する。

10

【0061】

データタイプが '`pitm`' であるプライマリアイテムボックスは、主画像を表す画像アイテムのアイテム ID を格納する。

【0062】

データタイプが '`inff`' であるアイテム情報ボックスは、以下の、アイテム情報エントリを格納するためのボックスである。

20

【0063】

データタイプが '`infe`' であるアイテム情報エントリは、各アイテムの、アイテム ID やアイテムタイプ、フラグを格納する。

【0064】

符号化画像データが HEVC 圧縮されたデータである画像アイテムのアイテムタイプは '`hvc1`' であり、画像アイテムである。

【0065】

オーバーレイ形式の導出画像アイテム、つまり、Image Overlay 構造体（オーバーレイ情報）のアイテムタイプは '`iov1`' であり、画像アイテムに分類である。

30

【0066】

`Exif` 形式のメタデータアイテムのアイテムタイプは '`Exif`' であり、メタデータアイテムである。

【0067】

また、画像アイテムには、フラグフィールドの最下位ビットが立っている場合、その画像アイテムは隠し画像 (`hidden image`) であることを指定することができる。このフラグが立っている場合、その画像アイテムは、再生時に表示対象として扱われず、非表示となる。

【0068】

データタイプが '`iref`' であるアイテム参照ボックスは、各アイテム間の参照関係を、参照関係の種類、参照元アイテムのアイテム ID、1 個以上の参照先アイテムのアイテム ID、で格納する。

40

【0069】

オーバーレイ形式の導出画像アイテムの場合、参照関係の種類に '`dimg`'、参照元アイテムのアイテム ID に導出画像アイテムのアイテム ID、参照先アイテムのアイテム ID にオーバーレイを構成する各画像アイテムのアイテム ID、を格納する。

【0070】

サムネイル画像の場合、参照関係の種類に '`thmb`'、参照元アイテムのアイテム ID にサムネイル画像のアイテム ID、参照先アイテムのアイテム ID に本画像のアイテム ID、を格納する。

【0071】

50

データタイプが'iprp'であるアイテムプロパティボックスは、以下の、アイテムプロパティコンテナボックスとアイテムプロパティ関連ボックスを格納するためのボックスである。

【0072】

データタイプが'ipco'であるアイテムプロパティコンテナボックスは、個々のプロパティデータのボックスを格納するボックスである。

【0073】

各画像アイテムは、その画像の特徴、属性を表すプロパティデータを持つことができる。

【0074】

プロパティデータボックスには、以下のようなものがある。

10

【0075】

デコーダ構成・初期化データ(Decoder configuration and initialization。HEVCの場合、タイプは'hvcc')は、デコーダの初期化に使われるデータである。HEVCのパラメータセットデータ(Video Parameter Set、Sequence Parameter Set、Picture Parameter Set)は、ここに格納する。

【0076】

画像空間範囲(Image spatial extents。タイプは'ispe')は、画像の大きさ(幅、高さ)である。

【0077】

色空間情報(Colour information。タイプは'colr')は、画像の色空間情報である。

20

【0078】

画像回転情報(Image rotation。タイプは'irot')が、画像を回転して表示する際の、回転方向である。

【0079】

画像のピクセル情報(タイプは'pixi')は、画像を構成するデータのビット数を示す情報である。

【0080】

また、本実施例では、HDRメタデータとして、マスターディスプレイカラーボリューム情報(Mastering display colour volume。タイプは'MDCV')や、コンテンツライトレベル情報(Contents light level information。タイプは'CLLI')を、プロパティデータボックスとして格納している。

30

【0081】

上記以外プロパティも存在するが、ここでは明記しない。

【0082】

データタイプが'ipma'であるアイテムプロパティ関連ボックスは、各アイテムとプロパティの関連付けを、アイテムIDと、関連するプロパティの'ipco'におけるインデックスの配列、の形式で格納する。

40

【0083】

データタイプが'idat'であるアイテムデータボックスは、データサイズが小さいアイテムデータを格納するボックスである。

【0084】

オーバーレイ形式の導出画像アイテムのデータは、構成画像の位置情報などを格納したImageOverlay構造体(オーバーレイ情報)を'idat'に格納することができる。オーバーレイ情報には、背景色情報であるcanvas_fill_valueパラメータや、オーバーレイで合成した際の最終的な合成後の再生画像サイズであるout_put_width、out_put_heightパラメータ等がある。さらに、合成要素となる画像毎に、合成画像内での配置位置を水平位置座標、垂直位置座標を示すhor

50

`horizontal_offset`、`vertical_offset` パラメータがある。オーバーレイ情報に含まれるこれらのパラメータを用いることにより、指定した背景色、サイズの 1 つの画像内の任意の位置に複数の画像を配置した合成画像を再生することができる。

【0085】

データタイプが '`iloc`' であるアイテムロケーションボックスは、各アイテムのデータの位置情報を、オフセット基準 (`construction_method`)、オフセット基準からのオフセット値、長さ、の形式で格納する。

【0086】

オフセット基準は、ファイルの先頭、または、'`idat`'、である。

【0087】

なお、メタデータボックスには上記以外のボックスも存在するが、ここでは明記しない。

【0088】

図 2 では、2 つの符号化画像アイテムがオーバーレイ形式の画像を構成している構造を図示している。オーバーレイ形式の画像を構成する符号化画像アイテム数は 2 に限定されない。オーバーレイ形式の画像を構成する符号化画像アイテム数が増加すると、以下のボックス、項目、がその分増加する。

- ・アイテム情報ボックスに、符号化画像アイテムの、アイテム情報エントリが追加。
- ・アイテム参照ボックスの参照関係種類 '`dimg`' の参照先アイテム ID に、符号化画像アイテムのアイテム ID が増加。
- ・アイテムプロパティコンテナボックスにおける、デコーダ構成・初期化データ、画像空間範囲、など。
- ・アイテムプロパティ関連ボックスに、符号化画像アイテムとその関連プロパティのインデックスの項目が追加。
- ・アイテムロケーションボックスに符号化画像アイテムの位置情報の項目が追加。
- ・メディアデータボックスに、符号化画像データが追加。

【0089】

< HDR 画像の撮影 >

次に、HDR 画像を撮影して記録する際の撮像装置 100 での処理について説明する。

【0090】

記録する画像の幅や高さが昨今大きくなっているが、あまりに大きいサイズを符号化すると他機種での復号時に互換性が失われたり、システムの規模が大きくなったりする。具体的に本実施例では符号・復号に H . 265 を用いた例を説明する。H . 265 には規格として、`Profile` や `Level` といったパラメータがあり、画像符号化方法や、画像サイズが大きくなるとこれらのパラメータが変化する。これらの値は復号時の機器で再生対象の可否を判断するパラメータであり、このパラメータを判定して再生不可として弾かれる場合がある。本実施例では大きな画像サイズを記録する場合、複数の画像表示領域に分割して、HEIF 規格が謳うオーバーレイの方法を用い、それらの符号データを HEIF ファイルに格納して管理する方法について述べる。このように複数の画像表示領域に分けることで 1 つあたりの画像表示領域のサイズが小さくなり他の機種での再生互換性が高まる。以下、符号化領域の 1 辺のサイズを 4096 以下と定めて説明する。なお符号化形式は H . 265 以外のものであってもよいし、1 辺のサイズを 4096 以外のサイズに規定してもよい。

【0091】

次に符号化領域の開始位置や符号化領域の幅・高さ、また、再生領域の開始位置や再生領域の幅・高さについてのアライメント制約について説明する。撮像装置としては、分割後の画像を符号化する際に、符号化領域の垂直・水平符号化開始アライメントや、再生時の再生幅・高さアライメント等がハードウェア制約として一般的に存在する。つまり 1 つ以上の符号化領域のそれぞれの符号化開始位置、再生領域の開始位置や幅・高さを計算して符号化する必要がある。撮像装置はユーザからの指示で複数の画像サイズを切り替えて

10

20

30

40

50

記録するため、画像サイズごとにそれらを切り替える必要がある。

【0092】

本実施例で説明する符号化処理部114は符号化時に横方向（以後x方向）に64画素単位、縦方向（以後y方向）に1画素単位の開始位置のアライメント制約がある。また符号化する幅、高さのアライメント制約にそれぞれ32画素単位、16画素単位の制約がある。再生領域の開始位置はx方向に2画素単位、y方向に1画素単位の開始位置のアライメント制約がある。また、再生領域の幅・高さのアライメント制約はそれぞれ、2画素、1画素単位の制約がある。なお、ここでアライメント制約について例を述べたが、一般的にはこれ以外のアライメント制約であってもよい。

【0093】

次にHDR記録モード（HDR撮影モード）でHDR画像データを撮影し、撮影したHDR画像データをHEIFファイルとして記録するまでのフローについて図3を用いて説明する。HDRモードでは撮像部112と画像処理部113においてHDRを示す色域BT.2020と、PQガンマカーブを用いてHDRの色空間を表現する処理を行う。HDRのための記録については本実施例では割愛する。

10

【0094】

S101ではCPU101がユーザによるSW2の押下の有無を判定する。ユーザが被写体を捉え、SW2まで押下した場合（S101のYES）、CPU101はSW2が押されたことを検知し、S102に遷移する。SW2が押されるまでCPU101はこの検出を続ける（S101のNO）。

20

【0095】

S102ではCPU101が、ユーザが捉えた画角の画像をHEIFファイルに保存するため、符号化領域を縦横方向にいくつに分割するか（分割数N）、どのように分割するかを決定する。この結果をメモリ102に保存する。S102については図4を用いながら後述する。S103に進む。

【0096】

S103ではCPU101がメモリ102にある変数MをM=1に初期化する。その後S104に進む。

【0097】

S104ではCPU101が、S102で決定した分割数N回分ループしたかを監視する。ループ回数が分割数Nに到達した場合（S104のYES）S109に進み、そうでなければ（S104のNO）S105に進む。

30

【0098】

S105ではCPU101が分割画像Mの符号開始領域、再生領域の情報を先のメモリ102から読み出しておく。S106に進む。

【0099】

S106ではCPU101が符号化処理部114にメモリ102に配置された画像全体のうち、どの領域を符号化するかを伝え、符号化させる。S107に進む。

【0100】

S107では符号化処理部114が符号化した結果の符号データと、符号時に発生する付随情報をCPU101がメモリ102に一時保存する。具体的にH.265の例では、のちにHEIFファイルに格納しなければならない、VPS、SPS、PPS等のH.265規格情報のことである。本実施例とは直接関係ないため割愛するが、H.265のVPS、SPS、PPSは、符号データのサイズや、ビット深度、表示・非表示領域の指定、フレーム情報等復号に必要な様々な情報である。その後S108に進む。

40

【0101】

S108ではCPU101が変数MをインクリメントしS104に戻る。

【0102】

S109ではCPU101がHEIFファイルに格納するメタデータを生成する。CPU101は、撮像部112、画像処理部113、符号化処理部114等から、再生時に必

50

要な情報を引出し、メタデータとしてメモリ102に一旦保存する。具体的には Exif 形式に成形されたデータである。 Exif データの詳細についてはすでに一般的であるため本実施例での説明は割愛する。S110に進む。

【0103】

S110ではCPU101がサムネイルを符号化する。メモリ102にある画像全体を画像処理部113が縮小しサムネイルサイズにして一旦メモリ102に配置する。このメモリ102にあるサムネイル画像をCPU101が符号化処理部114に指示して符号化する。ここでサムネイルは先に述べたアライメント制約を満たすようなサイズで符号化するが、符号化領域と再生領域とが異なる方式でもよい。このサムネイルもH.265で符号化される。S111に進む。

10

【0104】

S111では符号化処理部114がサムネイルを符号化した結果の符号データと、符号時に発生する付随情報をCPU101がメモリ102に一時保存する。先のS107の説明と同様にVPS、SPS、PPS等のH.265規格情報のことである。次にS112に進む。

【0105】

S112では、ここまでステップでCPU101が各種データをメモリ102に保存した。CPU101はこのメモリ102に保存したデータを順序良く結合しHEIFファイルを完成させ、メモリ102に保存する。このHEIFファイルを完成させるためのフローは、図6を用いてのちに説明する。S113に進む。

20

【0106】

S113でCPU101は、メモリ102にあるHEIFファイルを記録媒体制御部119によって記録媒体120に書き出すよう指示する。その後S101に戻る。

【0107】

このようなステップで撮影が行われてからHEIFファイルが記録媒体120に記録される。

【0108】

次に、先に説明したS102での分割の方法決定について、図4、図5(a)を用いて説明する。以下センサ画像領域の左上を原点(0, 0)とし、座標を(x, y)、幅高さをそれぞれw, hで表す。また領域については開始座標の(x, y)と幅高さのw, hを合わせて[x, y, w, h]と表現する。

30

【0109】

S121でCPU101は撮影時の記録モードをメモリ102から取得する。記録モードとは例えば画像サイズ、画像のアスペクト比、画像の圧縮率などが決定した記録方法を決めたもので、ユーザはこの記録モードから1つを選択して撮影を行う。例えば、Lサイズ、3:2アスペクトなどの設定のことである。記録モードに応じて撮影画角が異なるため、撮影の画角を決定する。S122に進む。

【0110】

S122ではセンサ画像領域[0, 0, H, V]の中から、再生画像の開始座標位置(x0, y0)を取得する。次にS123に進む。

40

【0111】

S123では再生画像の領域[x0, y0, Hp, Vp]をメモリ102から取得する。次にS124に進む。

【0112】

S124では再生画像の領域を含むように符号化する必要がある。先に述べたように符号開始位置のアライメントはx方向に64画素のアライメントをとるため、再生画像の領域を含むように符号化しようとすると(Ex0, E y0)の位置から符号化しなければならない。このようにしてL_{ex}のサイズが求まる。このL_{ex}のオフセットが0であれば(S124のNO)S126に進み、必要であれば(S124のYES)S125に進む。

【0113】

50

S 1 2 5 では、 L_{ex} を求める。 L_{ex} は先に述べたように再生画像領域と符号化開始位置のアライメントによって決定される。CPU101は L_{ex} を以下のようにして決定する。 x_0 を水平方向の符号化開始位置の画素アライメント64画素で除し、その商に先の64画素を乗じて求める。余りを含めず計算するため、再生画像領域の開始位置よりも左側に位置する座標が得られる。よって L_{ex} は、再生画像領域の x 座標から先の計算によって求めた符号化開始位置の x 座標の差で求まる。S 1 2 6 に進む。

【0 1 1 4】

S 1 2 6 では再生画像の最上位ラインの右端の最終位置の座標を計算する。これを(x_N , y_0)とする。 x_N は x_0 に H_p を加算して求める。S 1 2 7 に進む。

【0 1 1 5】

S 1 2 7 では再生画像の右端を含むように符号化領域の右端を求める。符号化領域の右端 w の求めるためには、符号化の幅のアライメント制約が必要である。先に述べたように符号化の幅・高さの制約はそれぞれ32画素の倍数の制約がある。この制約と(x_N , y_0)との関係から再生画像の右端より先に R_{ex} のオフセットが必要かどうかをCPU101が計算する。 R_{ex} が必要であれば(S 1 2 7 のYES) S 1 2 8 に進み、不要であれば(S 1 2 7 のNO) S 1 2 9 に進む。

【0 1 1 6】

S 1 2 8 では、CPU101が再生画像右端座標(x_N , y_0)の右側に32画素のアライメントになるように R_{ex} を付加し、符号化の終了位置(E_{xN} , E_{y0})を求める。S 1 2 9 に進む。

【0 1 1 7】

S 1 2 9 では、CPU101が表示領域の右下の座標を求める。再生画像のサイズが自明のためこの座標は(x_N , y_N)となる。S 1 3 0 に進む。

【0 1 1 8】

S 1 3 0 では再生画像の下端を含むように符号化領域の下端を求める。符号化領域の下端を求めるためには、符号化の高さのアライメント制約が必要である。先に述べたように符号化の高さの制約はそれぞれ16画素の倍数の制約がある。この制約と再生画像の右下座標(x_N , y_N)との関係から再生画像の下端より先に V_{ex} のオフセットが必要かどうかをCPU101が計算する。 V_{ex} が必要であれば(S 1 3 0 のYES) S 1 3 1 に進み、不要であれば(S 1 3 0 のNO) S 1 3 2 に進む。

【0 1 1 9】

S 1 3 1 ではCPU101が V_{ex} を先の制約と再生画像の右下座標(x_N , y_N)との関係から計算し、符号化領域の右下の座標(E_{xN} , E_{yN})を求める。 V_{ex} の計算方法は、垂直方向の符号化開始位置である y_0 から y_N を含むように符号化の高さアライメントである16画素の倍数の位置に設定する。具体的には、 $y_0 = 0$ であったとして、 V_p を符号化の高さアライメントの16画素で除し、商と余りを求める。再生領域を含むように符号化領域をとらなければならないため、余りがある場合は商に1を加算し、先の16画素を乗じる。これによって16画素の倍数でありながら V_p を含む符号化領域の下端の y 座標の E_{yN} が得られる。 V_{ex} は E_{yN} から y_N の差で求まる。なお、 E_{yN} は y_N から V_{ex} だけ下方向にオフセットさせた値であり、 E_{xN} はS 1 2 8 で既に求めた値である。S 1 3 2 に進む。

【0 1 2 0】

S 1 3 2 では、CPU101が、符号化領域のサイズ $H_p' \times V_p'$ を以下のように求める。水平サイズ H_p' に、アライメント制約から求めた L_{ex} と R_{ex} を加算して、 H_p' を求める。また、垂直サイズ V_p' に、アライメント制約から求めた V_{ex} を加算して、 V_p' を求める。 H_p' と V_p' から符号化領域のサイズ $H_p' \times V_p'$ が求まる。S 1 3 3 に進む。

【0 1 2 1】

S 1 3 3 では、CPU101が符号化対象の水平サイズ H_p' が分割すべき4096画素を超えるか否かを判断し、超える場合は(S 1 3 3 のYES) S 1 3 4 に進み、超えない場合(S 1 3 3 のNO)はS 1 3 5 に進む。

10

20

30

40

50

【 0 1 2 2 】

S 1 3 4 では、符号化対象の水平サイズ $H p'$ を4 0 9 6画素以下になるように2つ以上の領域に分割する。例えば6 0 0 0画素であれば2分割であるし、9 0 0 0画素であれば3分割となる。

【 0 1 2 3 】

2分割の場合は、先の符号化の幅に関するアライメント3 2画素を満足するおおよその中心位置で分割する。均等にならない場合は左の分割画像を大きくする。また3分割に分割する場合で均等にならない場合、先の符号化の幅に関するアライメントに注意しながら、A、B、Cと分割領域を分け、分割領域AとBのサイズが等しくなるようにし、分割領域Cをそれよりも若干小さくなるように分割する。3分割以上であっても同じアルゴリズムによって分割位置、サイズを決定する。S 1 3 5 に進む。

10

【 0 1 2 4 】

S 1 3 5 では、C P U 1 0 1 が符号化対象の垂直サイズ $V p'$ が分割すべき4 0 9 6画素を超えるか否かを判断し、超える場合(S 1 3 5 のY E S)はS 1 3 6 に進み、超えない場合(S 1 3 5 のN O)は処理を終了する。

【 0 1 2 5 】

S 1 3 6 では、S 1 3 4 と同様に $V p'$ を複数の分割領域に分割し処理を終了する。

【 0 1 2 6 】

S 1 0 2 での分割の例について、図5(b)、(c)、(d)、(e)を用いて具体的な数値で説明する。図5は、画像データの分割方法だけでなく、符号化領域および再生領域の関係も示している。

20

【 0 1 2 7 】

図5(b)について説明する。図5(b)は左右二つの領域に画像を分割し再生領域の左右と下端にL e x、R e x、V e xが付与されるケースである。まず、再生画像領域は[2 5 6 + 4 8、0、4 8 6 4、3 2 4 2]が撮像装置で記録したい最終的な画像の領域である。この再生画像領域の左上の座標は(2 5 6 + 4 8、0)であり、先に述べた符号化の開始アライメントである6 4画素単位の倍数ではない。よって、ここより4 8画素左側にL e xを設け、符号化開始位置の左端は $x = 2 5 6$ の位置となる。次に再生領域の幅については $w = 4 8 + 4 8 6 4$ となるが、これは符号化の幅のアライメントの3 2画素の倍数になっておらず符号化時の幅の制約アライメントを鑑みて、再生画像領域の右端に1 6画素R e xを付与する。同様に垂直方向も計算すると $V e x = 4$ 画素となる。この再生画像は横方向のサイズが4 0 9 6を超えていたため、横方向を符号化の幅のアライメントを鑑みつつおおよそ中心の位置で2分割し、結果として分割画像1の符号化領域は[2 5 6、0、2 4 9 6、3 2 4 2 + 8]、分割画像2の符号化画像領域は[2 5 6 + 2 4 9 6、0、2 4 3 2、3 2 4 2 + 8]が求まる。

30

【 0 1 2 8 】

図5(c)について説明する。図5(c)は左右二つの領域に画像を分割し、再生領域の右と下端にR e x、V e xが付与されるケースである。再生画像領域のx位置が $x = 0$ のため、符号化領域の開始アライメントである6 4画素とも一致するためL e x = 0となる。以後、先に述べた計算方法と同様であり、結果として分割画像1の符号化領域は[0、0、2 2 4 0、3 2 4 5 6 + 8]、分割画像2の符号化画像領域は[2 2 4 0、0、2 1 4 4、2 4 5 6 + 8]が求まる。

40

【 0 1 2 9 】

図5(d)について説明する。図5(d)は左右二つの領域に画像を分割し、再生領域の左右にL e x、R e xが付与されるケースである。再生画像領域の縦サイズ $h = 3 6 4 8$ であり、これは符号化時の画像高さの制約である1 6画素の倍数となるため、 $V e x = 0$ となる。以後、先に述べた計算方法と同様であり、結果として分割画像1の符号化領域は[2 5 6 + 4 8、0、2 2 4 0、3 6 4 8]、分割画像2の符号化画像領域は[2 5 6 + 2 2 4 0、0、2 1 4 4、3 6 4 8]が求まる。

【 0 1 3 0 】

50

図5(d)について説明する。図5(d)は上下左右に4つの領域に画像を分割し、再生領域の左右にL e x、R e xが付与されるケースである。このケースは符号化領域が、縦方向も横方向も4096を超えるため、田の字に分割を実施する。以後、先に述べた計算方法と同様であり、結果として分割画像1の符号化領域は[0、0、4032、3008]、分割画像2の符号化領域は[4032、0、4032、3008]、分割画像3の符号化領域は[0、3008、4032、2902]、分割画像4の符号化領域は[4032、3008、4032、2902]が求まる。

【0131】

このようにしてCPU101は記録モードに応じて再生画像の領域が決定し、それごとに分割領域の計算を行う必要がある。他の方法として、これらの計算をあらかじめ実施しておき、その結果をメモリ102保存しておく。CPU101は記録モードに応じた分割領域、再生領域の情報をメモリ102から読みだして符号化処理部114に設定してもよい。

10

【0132】

本実施例のように撮影画像を複数の分割画像に分割して記録する場合、記録した画像から撮影画像を復元するには、複数の表示領域を合成する必要がある。合成画像を生成する際に、各々の分割画像に重複領域がないほうが、余分なデータの符号化や記録をしなくてよいため、効率的となる。そのため、本実施例では分割画像間で接する境界には重なる部分（重複領域）を持たないように、各再生領域、各符号化領域を設定した。ただし、ハードウェアのアライメント制約などの条件によっては分割画像間に重複領域を設けても構わない。

20

【0133】

次に、先のS112で述べたHEIFファイルの構築方法について図6を用いながら説明する。HEIFファイルは図2で示す構造であるから、CPU101はファイルの先頭から順にメモリ102上に構築する。

【0134】

S141で'f t y p'ボックスを生成する。このボックスはファイルの互換性を知るためのボックスでファイルの先頭におかれる。CPU101はこのボックスのメジャーブランドに'heic'、コンパチブルブランドに'mif1'、'heic'を格納する。本実施例ではこれらの値を使用するがこれ以外であってもよい。S142に進む。

30

【0135】

S142で'meta'ボックスを生成する。このボックスは以下に記載の複数のボックスを格納するボックスである。CPU101はこのボックスを生成してS143に進む。

【0136】

S143では、「inf」ボックス内に格納する'hdr'ボックスを生成する。このボックスは先に述べた'meta'ボックスの属性を示すものである。CPU101はこのタイプに'pict'を格納して、S144に進む。なお'pict'は、メタデータボックスが管理するデータのタイプを表す情報であり、HEIF仕様では、ハンドラボックスのhandler_typeは'pict'と規定されている。

40

【0137】

S144では、「inf」ボックス内に格納する'dinf'ボックスを生成する。このボックスはこのファイルが対象とするデータの存在位置を示すものである。CPU101はこのボックスを生成してS145に進む。

【0138】

S145では、「inf」ボックス内に格納する'pitm'ボックスを生成する。このボックスは、主画像を表す画像アイテムのアイテムIDを格納する。主画像となるのはオーバーレイで合成されたオーバーレイ画像であるからCPU101はオーバーレイ情報アイテムIDを、主画像のアイテムIDとして格納する。S146に進む。

【0139】

S146では、「meta」ボックス内に格納する'inf'ボックスを生成する。このボ

50

ツクスはアイテムのリストを管理するボックスである。ここではデータ長フィールドに初期値を入れ、データタイプフィールドに' i i n f 'を保存し S 1 4 7 に進む。

【 0 1 4 0 】

S 1 4 7 では、' i i n f ' ボックス内に格納する' i n f e ' ボックスを生成する。この' i n f e ' ボックスはファイル内に格納される各アイテムのアイテム情報を登録するボックスであり、アイテム毎に' i n f e ' ボックスを生成する。分割画像については、それぞれの分割画像が1つのアイテムとしてこのボックスに登録される。他にも、複数の分割画像から主画像を構成するためのオーバーレイ情報や、E x i f 情報、サムネイル画像が、それぞれアイテムとして登録される。この際、前述したとおり、オーバーレイ情報、分割画像、サムネイル画像は、画像アイテムとして登録する。画像アイテムについては、隠し画像を示すフラグを付加することが可能であり、このフラグが付加することにより、再生時に画像が表示されないことになる。つまり、画像アイテムにこの隠し画像フラグを付加することにより、再生時に非表示とする画像を指定することができる。そこで、分割画像に関しては、隠し画像を示すフラグを立てておき、オーバーレイ情報のアイテムにはそのフラグを立てないでおく。これにより、個々の分割画像自体は表示対象外となり、オーバーレイ情報により複数の分割画像がオーバーレイ合成された結果の画像のみが表示対象となる。それぞれのアイテムについて個別に' i n f e ' ボックスを生成し、それらを先の' i i n f ' 内に格納する。S 1 4 8 に進む。10

【 0 1 4 1 】

S 1 4 8 では、' i i n f ' ボックス内に格納する' i r e f ' ボックスを生成する。このボックスはオーバーレイで構成される画像（主画像）、その画像を構成する分割画像の関係を示す情報を格納する。先に説明した S 1 0 2 で分割方法が決定しているため、決定された分割方法に基づいて C P U 1 0 1 はこのボックスを生成する。S 1 4 9 に進む。20

【 0 1 4 2 】

S 1 4 9 では、' i i n f ' ボックス内に格納する' i p r p ' ボックスを生成する。このボックスはアイテムのプロパティを格納するボックスであり、S 1 5 0 で生成される' i p c o ' ボックスや、S 1 5 1 で生成される' i p m a ' ボックスを格納するボックスである。S 1 5 0 に進む。50

【 0 1 4 3 】

S 1 5 0 では、' i p r p ' ボックス内に格納する' i p c o ' ボックスを生成する。このボックスはアイテムのプロパティコンテナボックスであり、各種プロパティを保存する。プロパティは複数存在し、C P U 1 0 1 は以下のプロパティコンテナボックスを生成して' i p c o ' ボックス内に格納する。プロパティは、画像アイテム毎に生成されるものと複数の画像アイテムに共通して生成されるものがある。主画像となるオーバーレイ情報により構成されるオーバーレイ画像および分割画像とで共通する情報として、' c o l r ' ボックスを生成する。' c o l r ' ボックスには、主画像（オーバーレイ画像）、分割画像の色空間情報として、H D R のガンマカーブ等の色情報を格納する。分割画像については、それ、' h v c C ' ボックス、' i s p e ' ボックスを生成する。先に説明した S 1 0 7 や S 1 1 1 で、メモリ 1 0 2 に格納してある符号化時の付随情報を C P U 1 0 1 が読み出し、' h v c C ' のプロパティコンテナボックスを生成する。この' h v c C ' に格納する付随情報は、40 符号化領域を符号化したときの情報や、符号化領域のサイズ（幅、高さ）だけでなく、再生領域のサイズ（幅、高さ）や、符号化領域中の再生領域の位置情報等も含む。これらの付随情報がゴロム圧縮されて' h v c C ' のプロパティコンテナボックスに記録される。ゴロム圧縮については一般的な圧縮方法のためここでは説明を割愛する。また、' i s p e ' ボックスには、分割画像の再生領域のサイズ（幅、高さ）情報（例として 2 2 4 0 × 2 4 5 0 等）を格納する。主画像となるオーバーレイ画像の回転を表す情報を格納した' i r o t ' ボックス、画像のデータのビット数を示す' p i x i ' ボックスを生成する。' p i x i ' ボックスについては、主画像（オーバーレイ画像）と分割画像とで個別に生成してもよいが、本実施例においてはオーバーレイ画像と分割画像は同じビット数、つまり 1 0 ビットであるので 1 つだけ生成する。また H D R の補足情報が格納される' C L L I ' や' M D C V ' 50

ボックスも生成する。さらに、サムネイル画像のプロパティとして、主画像（オーバーレイ画像）とは別に、色空間情報を格納した'color'ボックス、符号化時の情報を格納した'hvcC'ボックス、画像サイズを格納した'ispe'ボックス、画像データのビット数の情報を格納した'pixi'ボックス、HDRの補足情報を格納した'CLLI'ボックスを生成する。これらのプロパティコンテナボックスを生成し、'ipco'ボックス内に格納する。S151に進む。

【0144】

S151で'ipma'ボックスを生成する。このボックスはアイテムとプロパティの関係性を示すボックスであり、アイテム毎に先に述べたどのプロパティと関連性があるのかを示す。CPU101はメモリ102に保存された各種データからアイテムとプロパティの関係性を判断し、このボックスを生成する。S152に進む。10

【0145】

S152で'idat'ボックスを生成する。このボックスには各分割画像の再生領域をどのように配置してオーバーレイ画像を生成するかを示すオーバーレイ情報を格納する。オーバーレイ情報には、背景色情報であるcanvas_fill_valueパラメータや、オーバーレイ画像全体のサイズであるoutput_width、output_heightパラメータがある。また、合成要素となる分割画像一つずつに対して、分割画像を合成する水平位置座標、垂直位置座標を示すhorizontal_offset、vertical_offsetパラメータがある。これらのパラメータに対してCPU101は、S102で決定した分割方法に基づいてこれらの情報を記載する。具体的には、オーバーレイ画像のサイズを示すoutput_width、output_heightパラメータに、再生領域全体のサイズを記載する。そして、各分割画像の位置情報を示すhorizontal_offset、vertical_offsetパラメータには、再生領域左上の開始座標位置(x0、y0)から、分割画像の左上の位置までの幅方向および高さ方向のオフセット値をそれぞれ記載する。このようにしてオーバーレイ情報を生成することにより、画像の再生時に、分割画像をhorizontal_offset、vertical_offsetパラメータに基づいて配置して分割画像を合成することにより、分割前の状態の画像を再生することができるようになる。本実施例では、分割画像に重複領域を設けていないため、各分割画像が重ならないように配置されるように位置情報が記載されることになる。そして、分割画像を合成して得られた画像のうち、output_width、output_heightパラメータにより表示される領域を指定することにより、再生領域のみを表示対象として再生することができる。このように生成されたオーバーレイ情報が格納された'idat'ボックスを生成する。S153に進む。2030

【0146】

S153で'iloc'ボックスを生成する。このボックスは各種データがファイルのどの位置に配置されるかを示すボックスである。各種情報がメモリ102に保存されているため、これらのサイズからこのボックスを生成する。具体的に、オーバーレイを示す情報は先の'idat'に格納するが、この'idat'内部の位置とサイズ情報である。また、サムネイルデータ、符号データ12は'mdat'ボックスに格納するが、この位置とサイズ情報を格納する。S154に進む。40

【0147】

S154で'mdat'ボックスを生成する。このボックスは以下に記載の複数のボックスを含むボックスである。CPU101は'mdat'ボックスを生成しS155に進む。

【0148】

S155でExifデータを'mdat'ボックスに格納する。先のS109で、メタデータのExifがメモリ102に保存されているので、CPU101はこれをメモリ102から読み出し'mdat'ボックスに追記する。S156に進む。

【0149】

S156でサムネイルデータを'mdat'ボックスに格納する。先のS110で、サム50

ネイルデータはメモリ 102 に保存されているので、CPU101 はこれをメモリ 102 から読み出し、「mdat」ボックスに追記する。S157 に進む。

【0150】

S157 で符号化画像 1 のデータを「mdat」ボックスに格納する。先の S106 のループ 1 回目で、分割画像 1 のデータはメモリ 102 に保存されているので、CPU101 はこれをメモリ 102 から読み出し、「mdat」ボックスに追記する。これを符号化画像 N まで繰り返して符号化画像 1 - N をすべて「mdat」ボックスに追記する。このようなステップで CPU101 は HEIF ファイルを構築する。

【0151】

以上の処理により、図 2 のような HEIF フォーマットで HDR 画像を記録媒体 120 に記録することができる。

10

【0152】

< HDR 画像の再生 >

次に、記録媒体 120 に HEIF フォーマットで記録された HDR 画像ファイルを再生する際の撮像装置 100 での処理について説明する。なお、本実施形例では、撮像装置 100 で再生する場合について説明するが、撮像部を有さない画像処理装置において、記録媒体 120 に記録された HEIF フォーマットで記録された HDR 画像を再生する場合に、同様の処理を実現するようにしてもよい。

【0153】

図 7 を用いて、主画像がオーバーレイ画像時の HEIF の再生（表示）処理について説明する。

20

【0154】

ステップ S701 で、CPU101 は、記録媒体制御部 119 を使用して、記録媒体 120 に存在する指定ファイルの先頭部分をメモリ 102 に読み出す。そして、読み出したファイルの先頭部分に、正しい構造のファイルタイプボックスが存在するか、さらに、その中のブランドに、HEIF を表す「mif1」が存在するかをチェックする。

【0155】

また、独自のファイル構造に対応したブランドを記録してある場合、そのブランドの存在チェックを行う。そのブランドが特定の構造を保証する限りにおいて、このブランドのチェックにより、これ以降のいくつかの構造チェック、例えば、ステップ S703、ステップ S704、を省略することも可能である。

30

【0156】

ステップ S702 で、CPU101 は、記録媒体 120 から指定ファイルのメタデータボックスをメモリ 102 に読み出す。

【0157】

ステップ S703 で、CPU101 は、ステップ S702 で読み出したメタデータボックスからハンドラボックスを探し、構造をチェックする。HEIF の場合、ハンドラタイプは「pict」でなければならない。

【0158】

ステップ S704 で、CPU101 は、ステップ S702 で読み出したメタデータボックスからデータ情報ボックスを探し、構造をチェックする。本実施例では、同一ファイル内にデータが存在することを前提としているため、データエントリ URL ボックスに、そのむねを表すフラグが立っていることをチェックする。

40

【0159】

ステップ S705 で、CPU101 は、ステップ S702 で読み出したメタデータボックスからプライマリアイテムボックスを探し、主画像のアイテム ID を取得する。

【0160】

ステップ S706 で、CPU101 は、ステップ S702 で読み出したメタデータボックスからアイテム情報ボックスを探し、ステップ S705 で取得した主画像のアイテム ID に対応したアイテム情報エントリを取得する。上述の HDR 画像の撮影処理により記録

50

されたHEIF形式の画像ファイルの場合、オーバーレイ情報が主画像として指定されることになる。つまり、主画像のアイテムIDに対応したアイテム情報エントリには、そのアイテムタイプがオーバーレイを表す‘iov1’となっている。

【0161】

ステップS707で、CPU101は、オーバーレイ画像作成処理を実行する。これについて、図8を用いて、後に説明する。

【0162】

ステップS708で、CPU101は、ステップS707で作成したオーバーレイ画像を、表示制御部115を経由して、表示部116に表示する。

【0163】

なお、作成したオーバーレイ画像を表示する際には、画像の色空間情報を指定する必要があることがある。画像の色空間情報は、色空間プロパティ‘color’のcolour_primariesで色域が、transfer_characteristicsで変換特性（ガンマ相当）が指定されているので、それらの値を用いる。HDR画像の例として、色域がRec.ITU-R BT.2020、変換特性がRec.ITU-R BT.2100(PQ)、などが用いられる。

【0164】

また、アイテムプロパティコンテナボックスに、‘MDCV’、‘CLL1’などのHDR用メタデータが存在する場合、それを使用することも可能である。

【0165】

次に、図8を用いて、オーバーレイ画像作成処理について説明する。

【0166】

ステップS801で、CPU101は、オーバーレイ画像のプロパティ取得処理を実行する。これについては、図9を用いて、後に説明する。

【0167】

ステップS802で、CPU101は、オーバーレイ画像のオーバーレイ情報取得処理を実行する。オーバーレイ情報は、‘idat’ボックスに記録されているので、‘idat’ボックスに格納されているデータを取得することになる。

【0168】

オーバーレイ情報には、オーバーレイ画像の画像サイズ(‘output_width’、‘output_height’パラメータ)、オーバーレイ画像を構成する個々の画像アイテム(分割画像アイテム)の位置情報(‘horizontal_offset’、‘vertical_offset’)、などが存在する。

【0169】

オーバーレイ情報は、アイテムタイプがオーバーレイ‘iov1’である画像アイテムの、画像アイテムデータ扱いである。

【0170】

この取得処理については、図10を用いて、後に説明する。

【0171】

ステップS803で、CPU101は、分割画像アイテムのすべてのアイテムID取得処理を実行する。これについては、図11を用いて、後に説明する。

【0172】

ステップS804で、CPU101は、ステップS802で取得したオーバーレイ情報のオーバーレイ画像サイズを用いて、そのサイズの画像データを格納するメモリを確保する。このメモリ領域をキャンバスと呼ぶこととする。

【0173】

ステップS805で、CPU101は、分割画像カウンタnを0に初期化する。

【0174】

ステップS806で、CPU101は、分割画像カウンタnが分割画像アイテム数と等しいかチェックする。等しい場合、ステップS2210に進む。等しくない場合、ステッ

10

20

30

40

50

PS 807に進む。

【0175】

ステップS807で、CPU101は、n番目の分割画像アイテムに関して、1個の画像アイテムの画像作成処理を実行する。これについては、図12を用いて、後に説明する。

【0176】

ステップS808で、CPU101は、ステップS807で作成した、n番目の分割画像アイテムの画像（分割画像）を、ステップS802で取得したオーバーレイ情報に従い、ステップS804で確保したキャンバス上に配置する。

【0177】

オーバーレイでは、オーバーレイを構成する分割画像は、キャンバス上の任意の位置に配置することが可能であり、その位置はオーバーレイ情報に記述されている。しかし、オーバーレイ画像の外側は表示対象外である。そのため、ここで、分割画像をキャンバスに配置する際に、分割画像の符号化領域のうち、オーバーレイ画像の領域と重複する領域、すなわち、分割画像の再生領域、のみを配置するようにする。

【0178】

以下、図13を用いて、説明する。

【0179】

座標系は、オーバーレイ画像の左上を原点(0, 0)とし、X座標は右方向に増加、Y座標は下方向に増加するとする。

【0180】

オーバーレイ画像の大きさを、幅W_o、高さH_oとする(0 < W_o、0 < H_o)。

【0181】

従って、オーバーレイ画像は、左上の座標が(0, 0)で、右下の座標が(W_o - 1, H_o - 1)である。

【0182】

また、分割画像の大きさを、幅W_n、高さH_nとする(0 < W_n、0 < H_n)。

【0183】

分割画像の左上の位置を、(X_n, Y_n)とする。

【0184】

従って、分割画像は、左上座標が(X_n, Y_n)、右下座標が(X_n + W_n - 1, Y_n + H_n - 1)である。

【0185】

分割画像における、オーバーレイ画像（キャンバス）との重複領域は、以下の方法で求めることができる。

【0186】

以下の場合は、重複領域が存在しない場合である。

W_o - 1 < X_n (分割画像の左端が、オーバーレイ画像の右端より右)

X_n + W_n - 1 < 0 (分割画像の右端が、オーバーレイ画像の左端より左)

H_o - 1 < Y_n (分割画像の上端が、オーバーレイ画像の下端より下)

Y_n + H_n - 1 < 0 (分割画像の下端が、オーバーレイ画像の上端より上)

この場合、この画像は、処理対象外となる。

【0187】

また、以下の場合、分割画像全体が、重複領域となる。

0 ≤ X_nかつ(X_n + W_n - 1) ≤ (W_o - 1)かつ0 ≤ Y_nかつ(Y_n + H_n - 1) ≤ (H_o - 1)

この場合、分割画像全体を、キャンバス上の指定位置(X_n, Y_n)に配置する。

【0188】

上記以外の場合、分割画像の一部が配置対象となる。

重複領域の左上座標を(X_l, Y_t)、右下座標を(X_r, Y_b)とする。

左端X_lは、以下の通り。

10

20

30

40

50

```

i f ( 0 < = X n )
    X l = X n ;
e l s e
    X l = 0 ;
右端 X r は、以下の通り。
i f ( X n + W n - 1 < = W o - 1 )
    X r = X n + W n - 1 ;
e l s e
    X r = W o - 1 ;
上端 Y t は、以下の通り。
i f ( 0 < = Y n )
    Y t = Y n ;
e l s e
    Y t = 0 ;
下端 Y b は、以下の通り。
i f ( Y n + H n - 1 < = H o - 1 )
    Y b = Y n + H n - 1 ;
e l s e
    Y b = H o - 1 ;
この重複領域の大きさは、幅 X r - X l + 1 、高さ Y b - Y t + 1 、である。

```

【0189】

この重複領域の左上座標 (X l 、 Y t) 、右下座標 (X r 、 Y b) は、前述のように、キャンバスの左上を原点 (0 , 0) とした座標系である。

【0190】

重複領域の左上座標 (X l 、 Y t) を、分割画像の左上を原点とする座標系で表すと、以下のようなになる。

```

X l ' = X l - X n ;
Y t ' = Y t - Y n ;

```

【0191】

まとめると、以下のようなになる。

【0192】

分割画像の左上から (X l ' 、 Y t ') 離れた位置から幅 (X r - X l + 1) 、高さ (Y b - Y t + 1) の矩形を切り出し、キャンバスの (X l 、 Y t) の位置に配置する。

【0193】

これにより、分割画像のうちの再生領域を、キャンバス上の適切な位置に配置することが可能である。

【0194】

ステップ S 809 で、 C P U 1 0 1 は、分割画像カウンタ n に 1 加算し、ステップ S 806 に戻る。

【0195】

ステップ S 810 で、 C P U 1 0 1 は、ステップ S 801 で取得したオーバーレイ画像のプロパティ中に回転プロパティ ' i r o t ' が存在するか調べ、存在する場合、回転角を調べる。回転プロパティが存在しない場合、または、回転プロパティが存在するが回転角が 0 の場合、処理を終了する。回転プロパティが存在し、回転角が 0 以外の場合、ステップ S 811 に進む。

【0196】

ステップ S 811 で、 C P U 1 0 1 は、ステップ S 806 ~ ステップ S 809 で作成したキャンバスの画像を、ステップ S 810 で取得した角度だけ回転し、それを作成画像とする。

【0197】

10

20

30

40

50

これにより、オーバーレイ画像を作成することが可能である。

【0198】

なお、上記説明では、ステップS806～ステップS809の処理を分割画像数分ループで繰り返しているが、並行処理可能な環境では、ステップS806～ステップS809の処理を、分割画像数分並行処理しても良い。

【0199】

次に、図9を用いて、画像アイテムのプロパティ取得処理について説明する。

【0200】

ステップS901で、CPU101は、メタデータボックス内のアイテムプロパティボックス内のアイテムプロパティ関連ボックスから、指定画像アイテムIDのエントリを探し、そこに格納されている、プロパティインデックスの配列を取得する。
10

【0201】

ステップS902で、CPU101は、配列カウンタnを0に初期化する。

【0202】

ステップS903で、CPU101は、配列カウンタnが配列要素数と等しいかチェックする。等しい場合、終了する。等しくない場合、ステップS904に進む。

【0203】

ステップS904で、CPU101は、メタデータボックス内のアイテムプロパティボックス内のアイテムプロパティコンテナボックスから、配列のn番目の要素のインデックスのプロパティを取得する。
20

【0204】

ステップS905で、CPU101は、配列カウンタnに1加算し、ステップS903に戻る。

【0205】

次に、図10を用いて、画像アイテムのデータ取得処理について説明する。

【0206】

ステップS1001で、CPU101は、メタデータボックス内のアイテムロケーションボックスから、指定画像アイテムIDのエントリを探し、オフセット基準(const`construction_method`)、オフセット、長さ、を取得する。

【0207】

ステップS1002で、CPU101は、ステップS1001で取得したオフセット基準をチェックする。オフセット基準は、値0がファイルの先頭からのオフセット、値1がアイテムデータボックス内のオフセット、を表す。オフセット基準が0の場合、ステップS1003に進む。オフセット基準が1の場合、ステップS1004に進む。
30

【0208】

ステップS1003で、CPU101は、ファイルの先頭からオフセットバイトの位置から長さバイト分を、メモリ102に読み出す。

【0209】

ステップS1004で、CPU101は、メタデータボックス内のアイテムデータボックスのデータ部の先頭からオフセットバイトの位置から長さバイト分を、メモリ102に読み出す。
40

【0210】

次に、図11を用いて、主画像を構成する画像のアイテムID取得処理について説明する。

【0211】

ステップS1101で、CPU101は、メタデータボックス内のアイテム参照ボックスから、参照タイプが‘d i n g’で、参照元アイテムIDが主画像のアイテムIDであるエントリを探す。

【0212】

ステップS1102で、CPU101は、ステップS1101で取得したエントリの、
50

参照先アイテム I D の配列を取得する。

【0213】

次に、図12を用いて、1個の符号化画像アイテムの画像作成処理について説明する。

【0214】

ステップS1201で、CPU101は、その画像アイテムのプロパティを取得する。これは、図9を用いて、説明済みである。

【0215】

ステップS1202で、CPU101は、その画像アイテムのデータを取得する。これは、図10を用いて、説明済みである。画像アイテムのデータとは、符号化された画像データである。

10

【0216】

ステップS1203で、CPU101は、ステップS1201で取得したプロパティのうちのデコーダ構成・初期化データを用いて、デコーダを初期化する。

【0217】

ステップS1204で、CPU101は、ステップS1202で取得した符号化された画像データをデコーダでデコードし、デコード結果を取得する。

【0218】

ステップS1205で、CPU101は、ピクセル形式変換が必要かどうかチェックする。

【0219】

不要な場合、終了する。必要な場合、ステップS1206に進む。

20

【0220】

デコーダの出力データのピクセル形式と、表示装置がサポートしている画像のピクセル形式が異なる場合、ピクセル形式変換が必要である。

【0221】

例えば、デコーダの出力データのピクセル形式がY C b C r（輝度色差）形式であり、表示装置の画像のピクセル形式がR G B形式の場合、Y C b C r形式からR G B形式へのピクセル形式変換が必要である。また、同じY C b C r形式でも、ビット深度（8ビット、10ビット、など）や色差サンプル（4：2：0、4：2：2、など）が異なる場合も、ピクセル形式変換が必要である。

30

【0222】

なお、色空間情報プロパティ'color'のmatrix_coefficientsで、R G B形式からY C b C r形式へ変換する際の係数が指定されているので、Y C b C r形式からR G B形式へ変換するためには、その係数の逆数を用いればよい。

【0223】

ステップS1206で、CPU101は、ステップS1204で取得したデコーダ出力データを、所望のピクセル形式に変換する。

【0224】

これにより、指定された符号化画像アイテムの画像を作成することが可能である。

【0225】

以上の処理により、記録媒体120に図2のようなH E I Fフォーマットで記録されたH D R画像を再生することができる。

40

【0226】

<他の実施形態>

以上、本発明について実施の形態に基づいて説明したが、本発明は、これらの実施の形態に限定されるものではなく、この発明の要旨を逸脱しない範囲の様々な形態も本発明に含まれる。

【0227】

また、上記の実施の形態の機能を制御方法として、この制御方法を画像処理装置に実行させるようにすればよい。また、上述の実施の形態の機能を有するプログラムを制御プロ

50

グラムとして、当該制御プログラムを画像処理装置が備えるコンピュータに実行させるようにしてよい。なお、制御プログラムは、例えば、コンピュータに読み取り可能な記憶媒体に記録される。

【 0 2 2 8 】

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する。そして、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ー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。

10

20

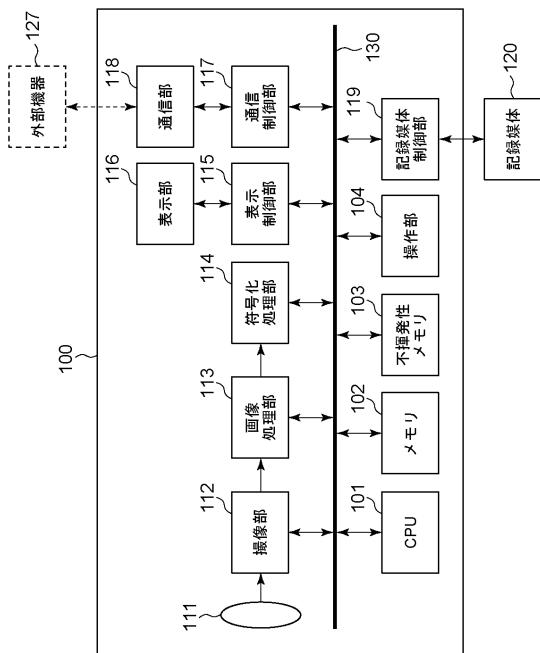
30

40

50

【図面】

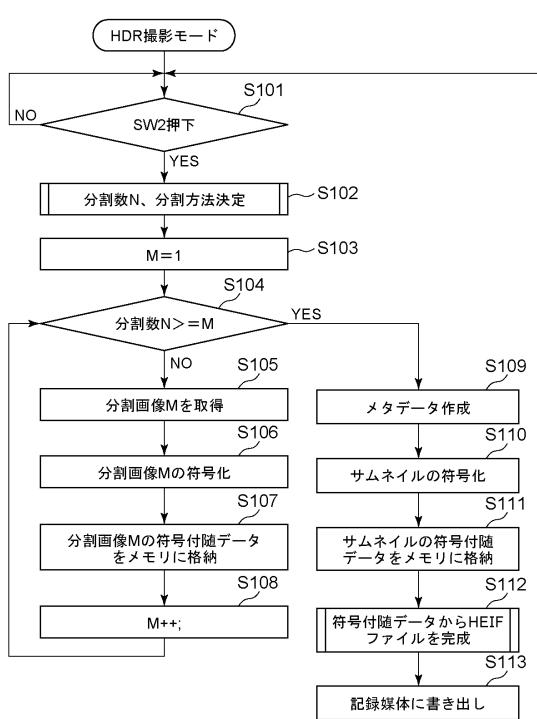
【図 1】



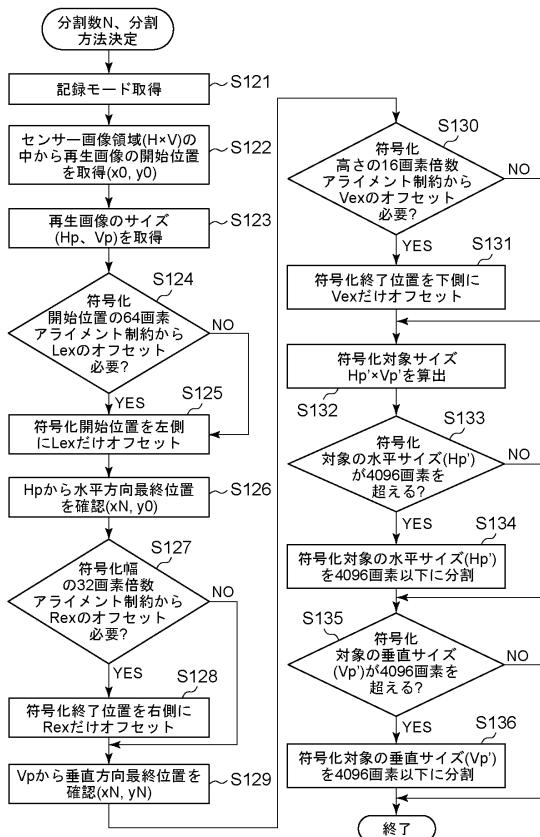
【図 2】

	File Type Box
'meta'	Metadata Box
'hdlr'	Handler Box
'dinf'	Data information Box
'dref'	Data Reference
'url'	DataEntryURLBox
'pitm'	Primary Item Box
'iinf'	符号化画像のアイテム情報エントリ
'infe'	符号化画像2のアイテム情報エントリ
'infe'	主画像のアイテム情報エントリ
'infe'	サムネイル画像のアイテム情報エントリ
'infe'	メタデータのアイテム情報エントリ
'iref'	Item Reference Box
'dimg'	主画像と符号化画像1、2の関係
'thmb'	主画像とサムネイル画像の関係
'cdsc'	主画像とメタデータの関係
'iprp'	Item Properties Box
'ipco'	ItemPropertyContainerBox
'colr'	主画像、符号化画像1、2の色空間情報
'hvcC'	符号化画像1のDecoder Configuration Record
'ispe'	符号化画像のサイズ情報
'pixi'	主画像、符号化画像のピクセル情報
'hvcC'	符号化画像2のDecoder Configuration Record
'ispe'	符号化画像のサイズ情報
'pixi'	サムネイル画像のピクセル情報
'CLLT'	サムネイル画像のContent Light Level Info(HDR静的のメタ)
'colr'	サムネイル画像の色空間情報
'hvcC'	サムネイル画像のDecoder Configuration Record
'ispe'	サムネイル画像のサイズ情報
'pixi'	サムネイル画像のピクセル情報
'CLLT'	サムネイル画像のContent Light Level Info(HDR静的のメタ)
'MDCV'	Mastering Display Color Volume(HDR静的のメタ)
'ipra'	Item Property Association
'ipba'	符号化画像のプロパティのインデックス配列
'ipba'	符号化画像2のプロパティのインデックス配列
'ipba'	主画像のプロパティのインデックス配列
'ipba'	サムネイル画像のプロパティのインデックス配列
'idat'	Image Overlay 構造体
'iloc'	Item Location Box
'iloc'	符号化画像1のデータ(HEVC)の位置情報
'iloc'	符号化画像2のデータ(HEVC)の位置情報
'iloc'	主画像のデータ(Image Overlay構造体)の位置情報
'iloc'	サムネイル画像のデータ(HEVC)の位置情報
'iloc'	メタデータ(Exifデータ)の位置情報
'mdat'	Media Data Box
'mdat'	メタデータ(Exifデータ)
'mdat'	サムネイル画像のデータ(HEVC)
'mdat'	符号化画像1のデータ(HEVC)
'mdat'	符号化画像2のデータ(HEVC)

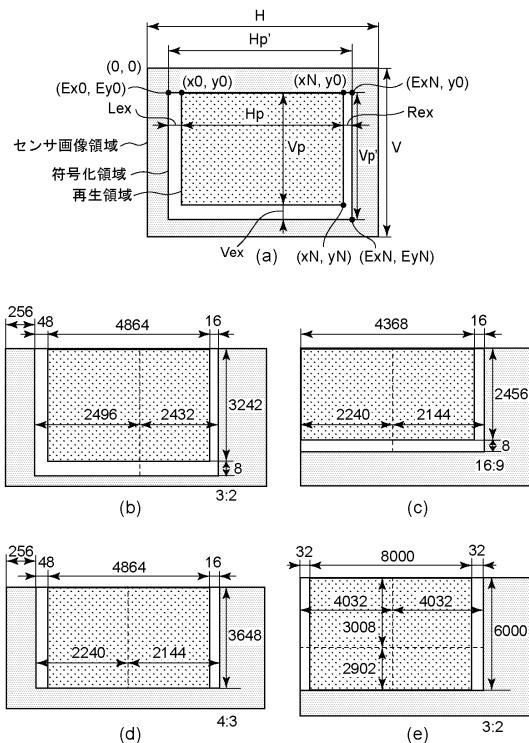
【図 3】



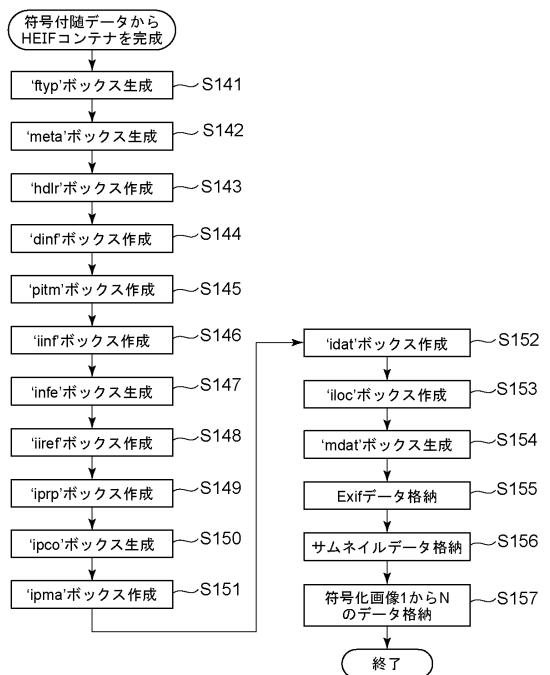
【図 4】



【図5】



【図6】



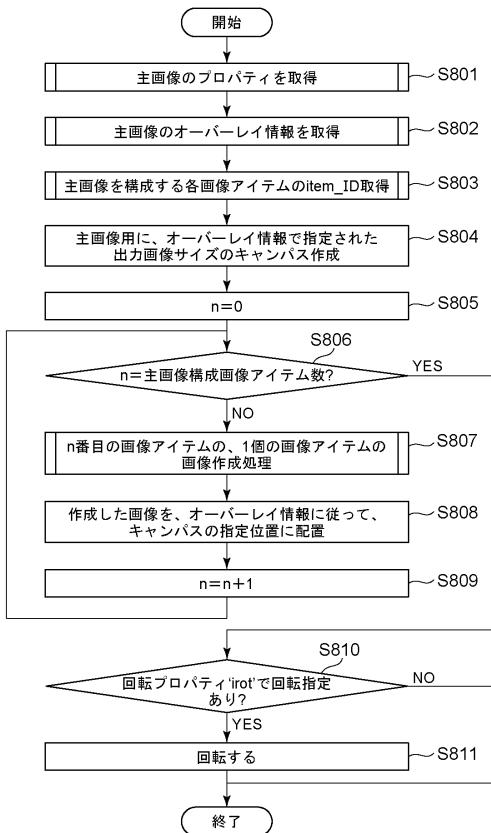
10

20

【図7】



【図8】

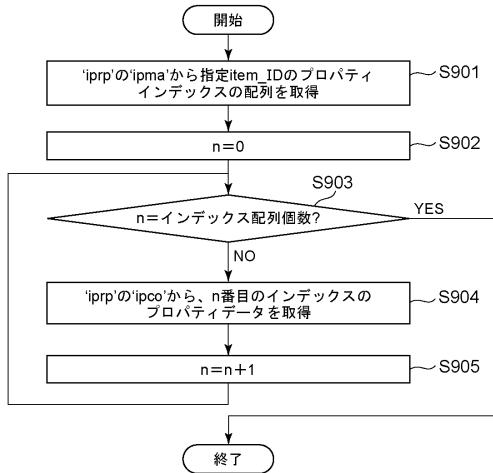


30

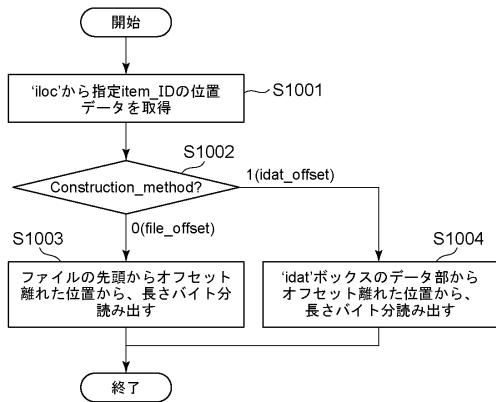
40

50

【図 9】



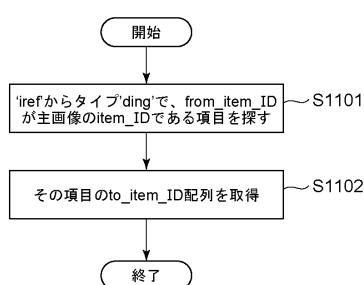
【図 10】



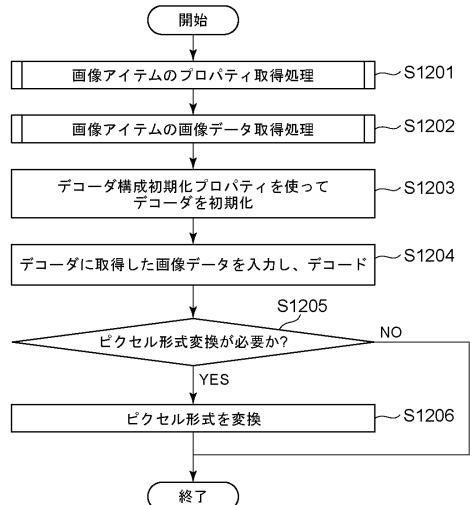
10

20

【図 11】



【図 12】

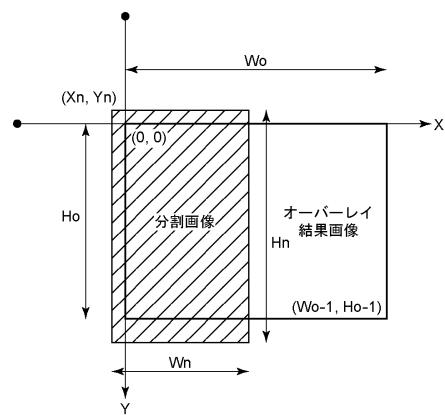


30

40

50

【図 1 3】



10

20

30

40

50

フロントページの続き

ヤノン株式会社内

審査官 川中 龍太

- (56)参考文献
- 特開2018-011197(JP,A)
特開2012-080432(JP,A)
米国特許出願公開第2017/0308116(US,A1)
米国特許第06466254(US,B1)
特開2017-139618(JP,A)
HEIF-Utility-Native-DLL , liuziangexit , 2018年02月07日 , [令和4年2月21日検索] , インターネット < URL: <https://github.com/liuziangexit/HEIF-Utility-Native-DLL/blob/master/EnglishIntro/README.md> >
Davide Concion , High Efficiency Image File Format #WWDC17 Session 513 , Apple Inc. , 2017年06月 , [令和4年2月18日検索] , インターネット < https://devstreaming-cdn.apple.com/videos/wwdc/2017/513fzgbviu23l/513/513_high_efficiency_image_file_format.pdf?dl=1 >
Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 12: Image File Format , INTERNATIONAL STANDARD ISO/IEC 23008-12 , ISO/IEC , 2017年12月
- (58)調査した分野 (Int.Cl. , DB名)
- H 04 N 5 / 76 - 5 / 956
H 04 N 19 / 00 - 19 / 98
G 11 B 20 / 10 - 20 / 16
G 11 B 27 / 00 - 27 / 34