

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6000763号
(P6000763)

(45) 発行日 平成28年10月5日(2016.10.5)

(24) 登録日 平成28年9月9日(2016.9.9)

(51) Int.Cl. F I
 HO4N 19/85 (2014.01) HO4N 19/85
 HO4N 5/91 (2006.01) HO4N 5/91 N

請求項の数 10 (全 18 頁)

(21) 出願番号	特願2012-192292 (P2012-192292)	(73) 特許権者	000001007
(22) 出願日	平成24年8月31日 (2012.8.31)		キヤノン株式会社
(65) 公開番号	特開2014-49980 (P2014-49980A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成26年3月17日 (2014.3.17)	(74) 代理人	100076428
審査請求日	平成27年8月27日 (2015.8.27)		弁理士 大塚 康徳
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治
		(74) 代理人	100134175
			弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 画像処理装置及び画像処理方法

(57) 【特許請求の範囲】

【請求項1】

第1の動画像と第2の動画像とを結合する処理を行う装置であって、
 前記第1の動画像と前記第2の動画像の結合点の前後に隣接する第1のフレーム及び第2のフレームのピクチャ識別子とスライスタイプとを取得する取得手段と、
 前記取得手段により取得された前記第1のフレームのピクチャ識別子と前記第2のフレームのピクチャ識別子とが一致するか否かを判定し、一致する場合に、前記第2のフレームのピクチャ識別子を、前記第1のフレームのピクチャ識別子と異なる値に変更する変更手段とを備え、

前記変更手段は、前記第2のフレームのスライスタイプが第1のスライスタイプと一致する場合、前記第2のフレームのスライスタイプを前記第1のスライスタイプよりもビット長が短い第2のスライスタイプへ変更する処理を行うとともに、前記第2のフレームのピクチャ識別子を、前記一致するピクチャ識別子よりも長いビット長に変更する処理を行うことを特徴とする画像処理装置。

10

【請求項2】

前記変更手段は、前記第2のフレームのスライスタイプが前記第1のスライスタイプと一致する場合であって、前記第2のフレームのピクチャ識別子のビット長が第1のビット長以下の場合に、前記第2のフレームのピクチャ識別子を、前記一致するピクチャ識別子よりも長いビット長に変更する処理を行い、

前記第1のビット長は、前記第1のスライスタイプのビット長と前記第2のスライスタ

20

イプのビット長との差分と、前記ピクチャ識別子が取り得る最大ビット長とに基づいて決定されることを特徴とする請求項 1 に記載の画像処理装置。

【請求項 3】

前記変更手段は、前記第 2 のフレームのスライスタイプが前記第 1 のスライスタイプと一致しない場合、前記第 2 のフレームのピクチャ識別子を前記一致するピクチャ識別子よりも長いビット長に変更する処理を行わないことを特徴とする請求項 1 に記載の画像処理装置。

【請求項 4】

第 1 の動画像と第 2 の動画像とを結合する処理を行う装置であって、

前記第 1 の動画像と前記第 2 の動画像の結合点の前後に隣接する第 1 のフレーム及び第 2 のフレームのピクチャ識別子とスライスタイプとを取得する取得手段と、

前記取得手段により取得された前記第 1 のフレームのピクチャ識別子と前記第 2 のフレームのピクチャ識別子とが一致するか否かを判定し、一致する場合に、前記第 2 のフレームのピクチャ識別子を、前記第 1 のフレームのピクチャ識別子と異なる値に変更する変更手段とを備え、

前記変更手段は、前記第 2 のフレームのスライスタイプが第 2 のスライスタイプと一致する場合、前記第 2 のフレームのスライスタイプを前記第 2 のスライスタイプよりもビット長が長い第 1 のスライスタイプへ変更する処理を行うとともに、前記第 2 のフレームのピクチャ識別子を、前記一致するピクチャ識別子よりも短いビット長に変更する処理を行うことを特徴とする画像処理装置。

【請求項 5】

前記変更手段は、前記第 2 のフレームのスライスタイプが前記第 2 のスライスタイプと一致する場合であって、前記第 2 のフレームのピクチャ識別子のビット長が、第 2 のビット長以上の場合に、前記第 2 のフレームのピクチャ識別子を、前記一致するピクチャ識別子よりも短いビット長に変更する処理を行い、

前記第 2 のビット長は、前記第 1 のスライスタイプのビット長と前記第 2 のスライスタイプのビット長との差分と、前記ピクチャ識別子が取り得る最小ビット長とに基づいて決定されることを特徴とする請求項 4 に記載の画像処理装置。

【請求項 6】

前記変更手段は、前記第 2 のフレームのスライスタイプが前記第 2 のスライスタイプと一致しない場合、前記第 2 のフレームのピクチャ識別子を前記一致するピクチャ識別子よりも短いビット長に変更する処理を行わないことを特徴とする請求項 4 に記載の画像処理装置。

【請求項 7】

前記変更手段は、前記第 2 のフレームのピクチャ識別子を、前記第 2 のフレームに続く第 3 のフレームのピクチャ識別子とも更に異なる値に変更することを特徴とする請求項 1 から 6 の何れか 1 項に記載の画像処理装置。

【請求項 8】

前記第 1 の動画像と前記第 2 の動画像は、H.264 規格で定められる IDR ピクチャのフレームを含み、

前記第 1 のフレームと前記第 2 のフレームは、前記 IDR ピクチャであって、

前記ピクチャ識別子は、前記 IDR ピクチャのスライスヘッダの `idr_pic_id` であり、前記スライスタイプは、前記スライスヘッダの `slice_type` であることを特徴とする請求項 1 から 7 のいずれか 1 項に記載の画像処理装置。

【請求項 9】

第 1 の動画像と第 2 の動画像とを結合する処理を行う画像処理方法であって、

前記第 1 の動画像と前記第 2 の動画像の結合点の前後に隣接する第 1 のフレーム及び第 2 のフレームのピクチャ識別子とスライスタイプとを取得する取得工程と、

前記取得工程において取得された前記第 1 のフレームのピクチャ識別子と前記第 2 のフレームのピクチャ識別子とが一致するか否かを判定し、一致する場合に、前記第 2 のフレ

10

20

30

40

50

ームのピクチャ識別子を、前記第1のフレームのピクチャ識別子と異なる値に変更する変更工程とを含み、

前記変更工程では、前記第2のフレームのスライスタイプが第1のスライスタイプと一致する場合、前記第2のフレームのスライスタイプを前記第1のスライスタイプよりもビット長が短い第2のスライスタイプへ変更する処理を行うとともに、前記第2のフレームのピクチャ識別子を、前記一致するピクチャ識別子よりも長いビット長に変更する処理を行うことを特徴とする画像処理方法。

【請求項10】

第1の動画像と第2の動画像とを結合する処理を行う画像処理方法であって、

前記第1の動画像と前記第2の動画像の結合点の前後に隣接する第1のフレーム及び第2のフレームのピクチャ識別子とスライスタイプとを取得する取得工程と、

前記取得工程において取得された前記第1のフレームのピクチャ識別子と前記第2のフレームのピクチャ識別子とが一致するか否かを判定し、一致する場合に、前記第2のフレームのピクチャ識別子を、前記第1のフレームのピクチャ識別子と異なる値に変更する変更工程とを含み、

前記変更工程では、前記第2のフレームのスライスタイプが第2のスライスタイプと一致する場合、前記第2のフレームのスライスタイプを前記第2のスライスタイプよりもビット長が長い第1のスライスタイプへ変更する処理を行うとともに、前記第2のフレームのピクチャ識別子を、前記一致するピクチャ識別子よりも短いビット長に変更する処理を行うことを特徴とする画像処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像処理装置及び画像処理方法に関するものである。

【背景技術】

【0002】

近年、撮影後の編集の容易さから画面内符号化（イントラ符号化）を利用した動画を撮影できる機種が増えてきている。画面内符号化された画像は、画面間符号化（インター符号化）と異なり、いわゆる参照画像が必要ない。符号化された1枚の画像データがあれば復号することが可能で、動画像から静止画像を切り出す、といった編集作業が容易である。

【0003】

ただし、便利な反面、画面間符号化を使用する場合に比べて、符号化効率が悪くなってしまうという欠点もある。画面間符号化を使用して動画像を符号化するMPEGに比べて、画面内符号化のみのMotion JPEGの符号量は一般的に多くなる。そのため、編集作業の手間はあるものの、動画像の符号化にはMPEGが用いられることが多かった。

【0004】

その一方でH.264といったJPEGやMPEGよりも効率のよい画像符号化が可能な方式が普及することにより、再び画面内符号化の動画が注目されている。H.264には画面内予測や予測ブロックサイズを適応的に変えることができる等、画面内符号化に関してだけでも効率を上げるツールがたくさん採用されている。

【0005】

H.264においては、IDR（Instantaneous Decoding Refresh）と呼ばれる特殊な画面内符号化ピクチャ（以下、IDRピクチャ）が用いられる。IDRピクチャが挿入されると、ストリームを復号するために必要な全ての状態がリセットされる（H.264のストリームの先頭は、必ずIDRでなければならない）。逆にIDRピクチャに関してその他の制約はなく、以降、どのタイミングでIDRピクチャが挿入されてもよい。即ち、全てのフレームがIDRピクチャであってもよいし、先頭以外、一切IDRピクチャがなくてもよい。

【0006】

10

20

30

40

50

画面内符号化されたフレームのピクチャタイプには、I D Rピクチャ以外に通常のIピクチャもある。IピクチャはM P E Gに採用されているIピクチャと同じで、俗に言うG O Pの先頭という意味で捉えることができる。ストリームの画像データの部分において、I D RピクチャとIピクチャは、符号化効率が変わるなどの差はない。

【 0 0 0 7 】

よって、画面内符号化のみを用いた画像符号化方式（以下、A L L - Iピクチャ）で動画像を作成する場合、全てI D Rピクチャにするか、Iピクチャも混ぜる（または先頭以外全てIピクチャなど）かの2通りの方法がある。しかし、例えばI D Rピクチャを含む動画の途中のストリームを削除して、削除した前後のストリームを接続する、といった場合に別の問題が発生する。

【 0 0 0 8 】

I D Rピクチャには各ピクチャにピクチャ識別子(`i d r _ p i c _ i d`)というパラメータがあり、ヘッダ内に書き込む必要がある。ピクチャ識別子は、H . 2 6 4ではゴロム符号化という可変長符号化方式を用いて、I D Rピクチャのピクチャ画像と共に符号化されている。

【 0 0 0 9 】

図8の表を参照して、上記ピクチャ識別子を可変長符号化している手法の例としてゴロム符号化について説明を行う。図8の表の左側の列がゴロム符号（二進数）、右側の列が表現できるデータ値（十進数）の範囲である。図8の表のゴロム符号の1の部分にはセパレータ、このセパレータを中心に左側の0がプリフィックス、右側がサフィックスと呼ばれている。サフィックスは図8の表ではxで表現されていて、xは0, 1いずれかの値が入るという意味である。プリフィックスの0の数はサフィックスのビット長を表わしていて、セパレータを中心に、左右のビット長が同じになる。

【 0 0 1 0 】

例えば、プリフィックスの0の数が2であった場合、サフィックスのビット長も2である。

【 0 0 1 1 】

図8の表でみると、上から3行目にあたり、表現できる値は3から6の4通りである。具体的にいうと、以下となる。

ゴロム符号：0 0 1 0 0 = データ値：3

ゴロム符号：0 0 1 0 1 = データ値：4

ゴロム符号：0 0 1 1 0 = データ値：5

ゴロム符号：0 0 1 1 1 = データ値：6

データ値0のときだけ例外で、プリフィックスとサフィックスが存在しない。このように数値の変換を可変長とすることで、符号化効率が向上する。

【 0 0 1 2 】

ピクチャ識別子はH . 2 6 4勧告によると、0から6 5 5 3 5までの任意の値をとってよいが、I D Rピクチャが隣り合う場合、同じピクチャ識別子が続いてはいけないという規定がある。隣接したI D Rピクチャで、同じピクチャ識別子が続けて付与されない限りは、同じピクチャ識別子が一つの動画の中で使用されても構わない。

【 0 0 1 3 】

しかしながら編集作業が行われた場合、編集後に同じピクチャ識別子が連続してしまう可能性がある。編集後に同じピクチャ識別子が連続する場合、どちらかのピクチャ識別子を修正すればよいのだが、このピクチャ識別子は前述の通り可変長符号化（ゴロム符号化）されており、その値だけの編集ができない場合がある。

【 0 0 1 4 】

例えば、2つの連続するI D Rピクチャのピクチャ識別子が2つとも0という値だった場合、ゴロム符号は1ビットの1という値で表現されている。同じピクチャ識別子が連続してはいけない、という規定から、どちらかのピクチャ識別子を変更しなければならないが、ゴロム符号化されているビットは1以外とることができない。

10

20

30

40

50

【 0 0 1 5 】

また、2つの連続するIDRピクチャのピクチャ識別子が1でさらにその前後のピクチャ識別子が2だった場合も問題が生じる。ピクチャ識別子が1の場合のゴロム符号は、図8の表より"010"である。このビット長で表現できる1以外の数は、ゴロム符号"011"で表される2という値である。ピクチャ識別子が重なったIDRピクチャの一方を1以外の数字で置き換えなければならないが、この場合は1以外の値は2しか選択できない。しかしながら、2と変更すると、さらにその隣のIDRピクチャのピクチャ識別子と重なり、さらにそこで修整しなければならない。

【 0 0 1 6 】

このように多数のIDRピクチャのピクチャ識別子を検索し、修整する必要がある。場合によっては、重複を避けることが不可能な状況もありうる。このような場合は、再符号化するか、編集する場所を所望の位置から変える必要がある。

【 0 0 1 7 】

上記の問題を解決する手法として、ピクチャ識別子のみの編集ができない場合は、パディングビット等を削除してピクチャ識別子へのビット長の割り当てを増やす方法が提案されている(特許文献1)。

【 先行技術文献 】

【 特許文献 】

【 0 0 1 8 】

【 特許文献 1 】 特許第 4 7 5 7 8 4 0 号 公 報

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 9 】

図9を用いて特許文献1にて提案されている実施例を説明する。ピクチャ識別子編集前のIDRピクチャは、

ヘッダ内のピクチャ識別子 9 0 1 (i d r _ p i c _ i d)

ストリームデータのビット長を整えるために挿入されるパディングビット 9 0 2 (c a b a c _ a l i g n m e n t _ o n e _ b i t)

データの最後に挿入されるパディングビット 9 0 5 (t r a i l i n g _ z e r o _ 8 b i t s)

を含む。

【 0 0 2 0 】

また、ピクチャ識別子編集後のピクチャは、

ピクチャ識別子 9 0 1 を変更可能なようにNビット拡張したピクチャ識別子 9 0 3

パディングビット 9 0 2 からNビット削減したパディングビット 9 0 4 (c a b a c _ a l i g n m e n t _ o n e _ b i t)

ビット長を整えるためにデータの最後に挿入されるパディングビット 9 0 6 (t r a i l i n g _ z e r o _ 8 b i t s)

を含む。

【 0 0 2 1 】

ここではピクチャ識別子のみの編集ができない場合、変更できるゴロム符号になるように、ピクチャ識別子 9 0 1 のビット長をNビット増やして、Nビット拡張したピクチャ識別子 9 0 3 としている。しかし、そのままではピクチャ全体のビット長がNビット増加してしまい、H.264ストリームとして成立しなくなる可能性がある。そこでビット量を調整するため、c a b a c _ a l i g n m e n t _ o n e _ b i t 9 0 2 から増加したNビット分削減し、9 0 4 とすることで、全体のビット量の増加が抑えられる。しかし、このc a b a c _ a l i g n m e n t _ o n e _ b i t 9 0 2 は、画像依存のデータであり、削減させたいNビット分の余裕がない可能性がある。その場合は、ストリームデータの最後についている t r a i l i n g _ z e r o _ 8 b i t s 9 0 5 から増加分を削減して、9 0 6 にする。また、t r a i l i n g _ z e r o _ 8 b i t s も画像依存なので、存

10

20

30

40

50

在しない場合がある。そのときは、次のスライスデータでつじつまを合わせる、としている。このような方法では、データ先頭にピクチャ識別子を編集するためのビット長を調整できるパラメータが存在する場合は比較的容易に編集できるが、存在しない場合はストリームを解析してパディングビットを調整しなければならないため処理が複雑化するおそれがある。

【0022】

そこで、本発明の目的は、動画像の編集の際に、編集後の動画像においてピクチャ識別子の連続を防止するためのピクチャ識別子の変更をより簡易とし、編集処理時の負担を軽減することを可能とする。

【課題を解決するための手段】

【0023】

上記目的を達成するために、本発明は、

第1の動画像と第2の動画像とを結合する処理を行う装置であって、

前記第1の動画像と前記第2の動画像の結合点の前後に隣接する第1のフレーム及び第2のフレームのピクチャ識別子とスライスタイプとを取得する取得手段と、

前記取得手段により取得された前記第1のフレームのピクチャ識別子と前記第2のフレームのピクチャ識別子とが一致するか否かを判定し、一致する場合に、前記第2のフレームのピクチャ識別子を、前記第1のフレームのピクチャ識別子と異なる値に変更する変更手段とを備え、

前記変更手段は、前記第2のフレームのスライスタイプが第1のスライスタイプと一致する場合、前記第2のフレームのスライスタイプを前記第1のスライスタイプよりもビット長が短い第2のスライスタイプへ変更する処理を行うとともに、前記第2のフレームのピクチャ識別子を、前記一致するピクチャ識別子よりも長いビット長に変更する処理を行うことを特徴とする。

【発明の効果】

【0024】

本発明によれば、単一のIDRピクチャのヘッダ内のデータを変更することにより、編集後の動画像においてピクチャ識別子の連続を防止するためのピクチャ識別子の変更をより簡易とし、編集処理時の負担を軽減することが可能となる。

【図面の簡単な説明】

【0025】

【図1】本発明の実施形態1における動画像編集装置101の概略構成を示すブロック図である。

【図2】本発明の実施形態1における削除符号化画像を説明するための図である。

【図3】本発明の実施形態1における処理の一例を示すフローチャートである。

【図4】本発明の実施形態1におけるヘッダ解析処理とヘッダ書換処理を説明するための図である。

【図5】本発明の実施形態2における動画像編集装置501の概略構成を示すブロック図である。

【図6】本発明の実施形態2における結合符号化画像を説明するための図である。

【図7】本発明の実施形態2における処理の一例を示すフローチャートである。

【図8】ゴロム符号とデータ値の範囲を示した図である。

【図9】従来の装置におけるストリーム編集の説明図である。

【図10】本発明の実施形態におけるピクチャ識別子のビット長の拡張を説明するための図である。

【図11】本発明の実施形態におけるピクチャ識別子のビット長の縮小を説明するための図である。

【図12】本発明の実施形態3における処理の一例を示すフローチャートである。

【図13】本発明の実施形態3におけるヘッダ解析処理とヘッダ書換処理を説明するための図である。

10

20

30

40

50

【発明を実施するための形態】**【0026】**

以下に、本発明の実施形態を、添付の図面に基づいて詳細に説明する。

【0027】**[実施形態1]**

以下、図1を参照して、本発明の実施形態1における動画像編集装置について説明する。動画像編集装置101には、全てのフレームのピクチャタイプがIDRピクチャである動画像が入力される。以下、当該動画像を本明細書では「符号化画像」と呼ぶ。動画像編集装置101には符号化画像102が入力され、編集符号化画像110が出力される。動画像編集装置101は、符号化画像解析部103、外部からの設定入力を受け付ける外部設定入力104、削除範囲設定部105、符号化画像102から削除符号化画像107を生成する符号化画像削除部106、ヘッダ解析部108、ヘッダ書換部109を有する。符号化画像解析部103、削除範囲設定部105、符号化画像削除部106により動画像編集部として機能する。

10

【0028】

なお、図1の動画像編集装置101において、各ブロックは専用ロジック回路やメモリを用いてハードウェア的に構成されてもよい。或いは、メモリに記憶されている処理プログラムをCPU等のコンピュータが実行することにより、ソフトウェア的に構成されてもよい。

【0029】

20

次に、図1に示す各ブロックの動作について説明する。本実施形態の動画像編集装置101は、「ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding」（以後「H.264」という）で符号化された動画を編集することができる。符号化画像解析部103は、入力された符号化画像102の構成情報としてストリーム中の各ピクチャの位置情報を、削除範囲設定部105へ出力する。削除範囲設定部105は、符号化画像102からの構成情報を元に、ユーザからの指定等の外部設定入力104に従って、符号化画像102の削除範囲のピクチャ位置情報を符号化画像削除部106へ出力する。

【0030】

符号化画像削除部106は、削除範囲設定部105で指示された削除開始点から削除終了点までの削除範囲のフレームを符号化画像102から削除して、削除符号化画像107をヘッダ解析部108とヘッダ書換部109へ出力する。

30

【0031】

ヘッダ解析部108は、削除符号化画像107の削除開始点の直前のフレームのヘッダと削除終了点の直後のフレームのヘッダ、即ち、削除により新たに隣接し合う（連続することとなった）フレームのヘッダを解析する。なお、当該解析に当たり、削除範囲設定部105から上述の削除範囲のピクチャ位置情報を取得しても良い。ヘッダ解析部108は、各フレームのヘッダ内のピクチャ識別子である`idr_pic_id`とスライスタイプである`slice_type`を抽出し、ヘッダ書換部109に出力する。

【0032】

ヘッダ書換部109は、取得したピクチャ識別子である`idr_pic_id`と`slice_type`をチェックする。削除開始点の直前のフレーム（第1のフレーム）の`idr_pic_id`と削除終了点の直後のフレーム（第2のフレーム）の`idr_pic_id`とが一致した場合、削除符号化画像107のヘッダ中の`idr_pic_id`および`slice_type`を書き換えて、編集符号化画像110を出力する。なお、ヘッダ書換部109における具体的なヘッダの書き換え方法は後述する。

40

【0033】

続いて、実施形態1における削除符号化画像107について、図2を参照して説明を行う。図1の符号化画像削除部106は、削除範囲設定部105で指示された削除範囲（例えば、削除開始点201から削除終了点202までの範囲）の符号化画像を削除して、削除符号化画像107を生成する。

50

【0034】

ここで `idr_pic_id` を重複しない値に変更できるようにするために、`idr_pic_id` の値は少なくとも5ビット以上必要である。しかし、`idr_pic_id` が必ず5ビット以上であるという保証はない。そこで本実施形態では、`idr_pic_id` が5ビット以上の値をとれないことがないように、ゴロム符号化されている `slice_type` を変更することでビットを余らせ、その分 `idr_pic_id` に割り振る。

【0035】

H.264規格によると、全てのフレームのピクチャタイプがIDRである場合、`slice_type` が取りうる値は、2もしくは7である。ここで、“2”とは、「スライスヘッダの設定はそのスライスだけに有効」という意味である。一方“7”とは「スライスヘッダの設定は、フレーム内のスライスに有効」という意味である。フレーム内に1つしかスライスが存在しない場合や、マルチスライスであってもピクチャ内の設定を変えない場合（即ち、複数スライスの設定が同一の場合）は“7”でよい。そこで、本発明においては、ストリーム生成時に `slice_type = 7`（第1のスライスタイプ）と設定し、編集時に `slice_type = 2`（第2のスライスタイプ）に変更することで余裕ビットを生み出す。

【0036】

図8に示されるように、データ値“7”はゴロム符号化において7ビットで表される。データ値“2”は、ゴロム符号化において3ビットで表される。そこで、ヘッダの書き換えが生じた場合、`slice_type` を“7”から“2”に変更することで4ビット削減され、その削減された4ビットを `idr_pic_id` へと振り分け可能である。

【0037】

次に、図3と図4を参照して、図1のヘッダ解析部108におけるヘッダ解析処理と、ヘッダ書換部109におけるヘッダ書換処理動作について詳細に説明する。図3は、ヘッダ解析部108およびヘッダ書換部109の制御動作の流れを示したフローチャートである。該フローチャートに対応する処理は、例えば、ヘッダ解析部108及びヘッダ書換部109として機能するプロセッサが対応するプログラム（装置のROM等に格納）を実行することにより実現できる。図4は、図2で示した符号化画像102と削除符号化画像107と編集符号化画像110のストリームの中身をフレーム単位で示した一構成例である。

【0038】

図3におけるS301では、ヘッダ書換部109はスライス削除開始点の直前のフレームの `idr_pic_id` をヘッダ解析部108から取得する。ここで、取得した `idr_pic_id` の値を「n」とする。ここで、符号化画像102の構成を図4に示すものと仮定すると、図4の符号化画像102における削除開始点201の直前のフレーム401の `idr_pic_id` の値は「2」、即ち $n = 2$ となる。

【0039】

図3のS302では、ヘッダ書換部109は削除終了点の直後のフレームの `slice_type` をヘッダ解析部108から取得する。ここで取得した `slice_type` の値を「a」とする。例えば、図4における削除終了点202の直後のフレーム402の `slice_type` は7、即ち $a = 7$ である。

【0040】

S303では、ヘッダ書換部109はステップ302で取得した削除終了点の直後のフレームの `slice_type` の値 a が、7であるか否かを判定する。もし削除終了点の直後のフレームの `slice_type` の値 a が7である場合（S303で「YES」）、ステップ304に進む。一方、削除終了点の直後のフレームの `slice_type` の値 a が7でない場合（S303で「NO」）、ステップ310に進む。例えば、図4における削除終了点202の直後のフレーム402の `slice_type` は7であるため、ステップ304に進む。

10

20

30

40

50

【0041】

S304では、ヘッダ書換部109は削除終了点の直後のフレームの*idr_pic_id*をヘッダ解析部108から取得する。取得した*idr_pic_id*の値を「m」とする。例えば、図4における削除終了点202の直後のフレーム402の*idr_pic_id*を取得する。ここで取得した*idr_pic_id*は2、即ちm=2である。

【0042】

S305では、ヘッダ書換部109はS301で取得した削除開始点の直前のフレームの*idr_pic_id*の値nと、S304で取得した削除終了点の直後のフレームの*idr_pic_id*の値mとを比較する。もし、nとmとが不一致の場合（S305で「NO」）、*idr_pic_id*を修正する必要がないので処理を終了する。一方、nとmとが一致する場合（S305で「YES」）、ステップ306に進む。例えば、図4における削除符号化画像107の削除開始点201の直前のフレーム401の*idr_pic_id*と、削除終了点202の直後のフレーム402の*idr_pic_id*とは、共に「2」で一致しているため、S306に進む。

【0043】

S306では、ヘッダ書換部109は削除終了点の直後のフレームの*idr_pic_id*のビット長をヘッダ解析部108から取得する。取得した*idr_pic_id*のビット長を「b」とする。例えば、図4における削除終了点202の直後のフレーム402の*idr_pic_id*のビット長を取得する。取得した*idr_pic_id*のビット長は、図8のゴロム符号とデータ値の範囲の関係より3ビットであることが分かる。

【0044】

S307では、ヘッダ書換部109はS306で取得した削除終了点の直後のフレームの*idr_pic_id*のビット長bが第1のビット長である29ビット以下であるか否かを判定する。H.264規格による*idr_pic_id*の最大ビット長は33ビットであり、現行が30ビット以上であった場合は4ビット拡張できない。即ち、第1のビット長は、2つのスライスタイプのビット長の差分と、*idr_pic_id*の最大ビット長とに基づいて決定される値である。S307では、処理対象の符号化画像102が、このような場合に該当するか否かを判定する。もし、削除終了点の直後のフレームの*idr_pic_id*のビット長bが第1のビット長以下である場合（S307で「YES」）ステップ308に進む。S307では、ヘッダ書換部109は削除終了点の直後のフレームの*idr_pic_id*のビット長のbが29ビットよりも大きい場合（S307で「NO」）ステップ318に進む。例えば、図4における削除終了点202の直後のフレーム402の*idr_pic_id*のビット長は3ビットであり、29ビット以下であるためS308に進む。

【0045】

S308では、ヘッダ書換部109は削除終了点の直後のフレームの*slice_type*を7から2に変更する。この処理を行うことで、4ビットの余裕ができる。例えば、図4における削除終了点202の直後のフレーム402の*slice_type*のみ7から2に変更する。続くS309では、ヘッダ書換部109は削除終了点の直後のフレームの*idr_pic_id*のビット長を4ビット拡張する。例えば、図4における削除終了点202の直後のフレーム402の*idr_pic_id*のビット長を4ビット拡張して、3ビットから7ビットにする。その後、S318に進む。

【0046】

S303でNOと判定された後、S310では、ヘッダ書換部109はS302で取得した削除終了点の直後のフレームの*slice_type*の値aが2であるか否かを判定する。この時考慮すべきは、符号化画像102を作成時に*slice_type*が7に設定されていた点である。即ち、*slice_type*が2であった場合には、当該フレームは過去に編集されているフレームと考えることができる。つまり、すでに*idr_pic_id*が4ビット拡張されていると考えてよい。そこで、削除終了点の直後のフレームの*slice_type*の値aが2である場合（S310で「YES」）、ステップ31

10

20

30

40

50

1に進む。一方、削除終了点の直後のフレームの `slice_type` の値 `a` が 2 でない場合 (S310 で「NO」)、本発明適用外の符号化画像 102 として処理を終了する。

【0047】

続く S311 では、ヘッダ書換部 109 は削除終了点の直後のフレームの `idr_pic_id` をヘッダ解析部 108 から取得する。取得した `idr_pic_id` を `m` とする。S312 では、ヘッダ書換部 109 はステップ 301 で取得した削除開始点の直前のフレームの `idr_pic_id` の `n` と、S311 で取得した削除終了点の直後のフレームの `idr_pic_id` の `m` を比較する。不一致の場合 (S312 で「NO」) は、`idr_pic_id` を修正する必要がないので処理を終了する。一方、一致の場合 (S312 で「YES」)、S313 に進む。

10

【0048】

S313 では、ヘッダ書換部 109 は削除終了点の直後のフレームの `idr_pic_id` のビット長をヘッダ解析部 108 から取得する。取得した `idr_pic_id` のビット長を `b` とする。続く S314 では、ヘッダ書換部 109 は S313 で取得した削除終了点の直後のフレームの `idr_pic_id` のビット長の `b` が第 2 のビット長である 5 ビット以上であるか否かを判定する。これは `idr_pic_id` が取り得る最小ビット長は 1 ビットだからである。即ち、第 2 のビット長は、2 つのスライスタイプのビット長の差分と、`idr_pic_id` の最小ビット長とに基づいて決定される値である。

【0049】

削除終了点の直後のフレームの `idr_pic_id` のビット長 `b` が第 2 のビット長以上である場合 (S314 で「YES」) はステップ 315 に進む。一方、削除終了点の直後のフレームの `idr_pic_id` のビット長の `b` が 5 よりも小さい場合 (S314 で「NO」)、S317 に進む。

20

【0050】

S315 では、ヘッダ書換部 109 は削除終了点の直後のフレームの `slice_type` を 2 から 7 に変更する。続く S316 では、ヘッダ書換部 109 は削除終了点の直後のフレームの `idr_pic_id` のビット長を 4 ビット縮小する。一方、S317 では、ヘッダ書換部 109 は S313 で取得した削除終了点の直後のフレームの `idr_pic_id` のビット長 `b` が 3 ビット以下であるか否かを判定する。削除終了点の直後のフレームの `idr_pic_id` のビット長 `b` が 3 ビット以下である場合 (S317 で「YES」)、本発明適用外の符号化画像 102 として処理を終了する。一方、削除終了点の直後のフレームの `idr_pic_id` のビット長 `b` が 3 ビットよりも大きい場合 (S317 で「NO」)、S318 に進む。この場合は、ビット長 `b` は 4 ビット (第 3 のビット長) である。

30

【0051】

S318 では、ヘッダ書換部 109 は削除終了点の直後のフレームの `idr_pic_id` を変更する。但し、削除終了点の直後のフレームの `idr_pic_id` を変更する値は、削除開始点の直前のフレームの `idr_pic_id` の値 `n` とは異なる値に変更する。

【0052】

例えば、図 4 における削除終了点 202 の直後のフレーム 402 の `idr_pic_id` を変更する場合、変更する値は、削除開始点の直前のフレームの `idr_pic_id` の 2 とは異なる 7 に変更する。本実施形態では、`idr_pic_id` を 2 から 7 へと変更しているが、変更後の値は 2 以外の値であれば良い。`idr_pic_id` を 4 ビット拡張することにより、図 8 で示されているデータ値の範囲は "1 - 2" の 2 種類から "7 - 14" の 8 種類へ増える。よって、本実施形態によれば "7 - 14" の 8 種類の `idr_pic_id` が選択可能となる。その結果、`idr_pic_id` が 2、`slice_type` が 7 のフレーム 402 を、`idr_pic_id` が 7、`slice_type` が 2 のフレーム 404 に変更した編集符号化画像 110 が生成される。なお、図 10 は、`idr_pic_id` のビット拡張前後のゴロム符号及びデータ値の例を示している。図 10 で

40

50

は、4ビット拡張によりデータ値「2」から「11」へ変更されている。また、図11は、idr_pic_idのビット縮小前後のゴロム符号及びデータ値の例を示している。図11では、4ビット縮小によりデータ値「12」から「2」へ変更されている。

【0053】

以上の説明では、S305およびS312において、idr_pic_idの値を比較しているが、比較自体を行わないという方法もある。つまり、idr_pic_idの値が一致していない場合でも、次の処理S306またはS313の処理へと進んでもよい。このように動作することで常にslice_typeを修正することになるが、例外判定処理が減ることで処理ステップが減少する利点が得られる。

【0054】

なお、以上の説明では、符号化画像がすべてIDRで構成されていた場合(ALL-IDRピクチャ)を説明したが、発明の実施形態は当該構成に限定されるものではない。編集後の削除符号化画像107において隣り合うIDRピクチャが存在する可能性があれば、本願発明が適用可能である。即ち、IDRピクチャを含む動画像であれば、同様に本発明を適用可能であり、本願発明の適用対象が全てIDRピクチャで構成される動画像だけに限定されることはない。

【0055】

以上の本実施形態によれば、符号化画像の編集後にピクチャ識別子が同一のIDRピクチャが連続してしまう場合であっても、単一のフレームのヘッダ内のデータを変更することにより、効率的にピクチャ識別子の変更を行うことができる。

【0056】

[実施形態2]

実施形態1では、入力された動画の一部のシーケンスを削除することにより編集を行う場合について説明した。これに対し実施形態2では、2つの動画を結合することにより新たな動画を生成する編集を行う場合について説明する。

【0057】

図5は、発明の実施形態2に対応する動画像編集装置の構成を示す。動画像編集装置501には、全てのフレームのピクチャタイプがIDRピクチャである符号化画像502及び503が入力され、編集符号化画像506が出力される。動画像編集装置501は、符号化画像502、503から結合符号化画像505を生成する符号化画像結合部504、ヘッダ解析部108、ヘッダ書換部109を有する。本実施形態では、符号化画像結合部504が動画像編集部として機能する。なお、図5の動画像編集装置501において、各ブロックは専用ロジック回路やメモリを用いてハードウェア的に構成されてもよい。或いは、メモリに記憶されている処理プログラムをCPU等のコンピュータが実行することにより、ソフトウェア的に構成されてもよい。

【0058】

次に、図5に示す各ブロックの動作について説明する。符号化画像結合部504は、入力される符号化画像502と符号化画像503とを結合して、結合符号化画像505を出力する。ヘッダ解析部108は、結合符号化画像505の結合点直前のフレームのヘッダと結合点直後のフレームのヘッダ、即ち、結合により新たに隣接し合う(連続する)こととなったフレームのヘッダを解析する。ヘッダ解析部108は、結合点前後の各IDRピクチャのヘッダ内のピクチャ識別子であるidr_pic_idと結合点前後の各フレームのヘッダ内のスライスタイプであるslice_typeを抽出し、ヘッダ書換部109に出力する。

【0059】

ヘッダ書換部109は、取得したidr_pic_idとslice_typeをチェックする。結合点直前のフレーム(第1のフレーム)のidr_pic_idと、結合点直後のフレーム(第2のフレーム)のidr_pic_idとが一致した場合に、結合符号化画像505のヘッダ中のidr_pic_idおよびslice_typeを書き換えて、編集符号化画像506を出力する。なお、ヘッダ書換部109における具体的なヘ

10

20

30

40

50

ッダの書き換え方法は後述する。

【0060】

続いて、実施形態2における結合符号化画像505について、図6を参照して説明を行う。まず、符号化画像502は、ストリームA(601)で構成され、符号化画像503は、ストリームB(602)で構成されている。本実施形態では、符号化画像結合部504が、ストリームA(601)の後にストリームB(602)を結合して結合符号化画像505を生成する場合を説明する。なお、結合符号化画像505のストリームA(601)とストリームB(602)の境界を、結合点603とする。

【0061】

次に図7を参照して、図5のヘッダ解析部108におけるヘッダ解析処理と、ヘッダ書換部109におけるヘッダ書換処理動作について詳細に説明する。図7は、ヘッダ解析部108およびヘッダ書換部109の制御動作の流れを示したフローチャートである。該フローチャートに対応する処理は、例えば、ヘッダ解析部108及びヘッダ書換部109として機能するプロセッサが対応するプログラム(装置のROM等に格納)を実行することにより実現できる。なお、図7に示す各ステップのうち、図3と同一の参照番号が振られているステップは、図3で説明した処理と同一処理が実行されるステップである。

10

【0062】

図7におけるS701では、ヘッダ書換部109は結合点の直前のフレームの*idr_pic_id*をヘッダ解析部108から取得する。ここで取得した*idr_pic_id*の値を「n」とする。続くS702では、ヘッダ書換部109は結合点の直後のフレームの*slice_type*を取得する。取得した*slice_type*の値を「a」とする。

20

【0063】

ステップ303では、ヘッダ書換部109がステップ702で取得した結合点の直後のフレームの*slice_type*の値aが7であるか否かを判定する。もし、結合点の直後のフレームの*slice_type*の値aが7である場合(S303で「YES」)、ステップ704に進む。一方、結合点の直後のフレームの*slice_type*の値aが7でない場合(S303で「NO」)、ステップ310に進む。

【0064】

S704では、ヘッダ書換部109は結合点の直後のフレームの*idr_pic_id*をヘッダ解析部108から取得する。ここで取得した*idr_pic_id*の値を「m」とする。続くステップ305では、ヘッダ書換部109はS701で取得した結合点の直前のフレームの*idr_pic_id*の値nと、ステップ704で取得した結合点の直後のフレームの*idr_pic_id*の値mとを比較する。もし、nとmとが不一致の場合(S305で「NO」)、*idr_pic_id*を修正する必要がないので処理を終了する。一方、nとmとが一致する場合(S305で「YES」)、S706に進む。

30

【0065】

S706では、ヘッダ書換部109は結合点の直後のフレームの*idr_pic_id*のビット長をヘッダ解析部108から取得する。取得した*idr_pic_id*のビット長を「b」とする。続くS307では、ヘッダ書換部109はS706で取得した結合点の直後のフレームの*idr_pic_id*のビット長のbが29ビット以下であるか否かを判定する。もし、結合点の直後のフレームの*idr_pic_id*のビット長のbが29ビット以下である場合(S307で「YES」)、ステップ708に進む。一方、結合点の直後のフレームの*idr_pic_id*のビット長のbが29ビットよりも大きい場合(S307で「NO」)、S718に進む。

40

【0066】

S708では、ヘッダ書換部109は結合点の直後のフレームの*slice_type*を7から2に変更する。この処理を行うことで、4ビットの余裕ができる。続くS709では、ヘッダ書換部109は結合点の直後のフレームの*idr_pic_id*のビット長を4ビット拡張する。その後、S718に進む。

50

【0067】

S303でNOと判定された後、Sステップ310では、ヘッダ書換部109はステップ702で取得した結合点の直後のフレームのslice_typeのaが2であるか否かを判定する。このとき考慮すべきは、符号化画像502および符号化画像503を作成時に、slice_typeが7に設定されていた点である。即ち、slice_typeが2であった場合には、当該フレームは過去に編集されているフレームと考えることができる。つまり、すでにidr_pic_idが4ビット拡張されていると考えてよい。そこで、結合点の直後のフレームのslice_typeの値aが2である場合(S310で「YES」)はS711に進む。一方、結合点の直後のフレームのslice_typeの値aが2でない場合(S310で「NO」)、本発明適用外の符号化画像502および符号化画像503として処理を終了する。

10

【0068】

続くS711では、ヘッダ書換部109は結合点の直後のフレームのidr_pic_idをヘッダ解析部108から取得する。取得したidr_pic_idをmとする。ステップ312では、ヘッダ書換部109はS701で取得した結合点の直前のフレームのidr_pic_idのnと、ステップ711で取得した結合点の直後のフレームのidr_pic_idのmとを比較する。不一致の場合(S312で「NO」)、idr_pic_idを修正する必要がないので処理を終了する。一方、一致の場合(S312で「YES」)、S713に進む。

20

【0069】

S713では、ヘッダ書換部109は結合点の直後のフレームのidr_pic_idのビット長をヘッダ解析部108から取得する。取得したidr_pic_idのビット長をbとする。続くステップ314では、ヘッダ書換部109はS713で取得した結合点の直後のフレームのidr_pic_idのビット長のbが5ビット以上であるか否かを判定する。結合点の直後のフレームのidr_pic_idのビット長のbが5ビット以上である場合(S314で「YES」)、S715に進む。結合点の直後のフレームのidr_pic_idのビット長のbが5ビットよりも小さい場合(S314で「NO」)、S317に進む。

【0070】

S715では、ヘッダ書換部109は結合点の直後のフレームのslice_typeを2から7に変更する。続くS716では、ヘッダ書換部109は結合点の直後のフレームのidr_pic_idのビット長を4ビット縮小する。一方、S317では、ヘッダ書換部109はS713で取得した結合点の直後のフレームのidr_pic_idのビット長のbが3ビット以下であるか否かを判定する。結合点の直後のフレームのidr_pic_idのビット長のbが3ビット以下である場合(S317で「YES」)、本発明適用外の符号化画像502および符号化画像503として処理を終了する。一方、結合点の直後のフレームのidr_pic_idのビット長のbが3ビットよりも大きい場合(S317で「NO」)、S718に進む。なお、この場合はビット長bは4ビットである。S718では、結合点の直後のフレームのidr_pic_idを変更する。

30

【0071】

なお、上記S305および312においてidr_pic_idの値を比較しているが、比較自体を行わないという方法もある。つまり、idr_pic_idの値が一致していない場合でも、次の処理S706またはS713の処理へと進んでもよい。このように動作することで常にslice_typeを修正することになるが、例外判定処理が減ることで処理ステップが減少する利点が見られる。

40

【0072】

なお、以上の説明では、符号化画像がすべてIDRで構成されていた場合(ALL-IDRピクチャ)を説明したが、発明の実施形態は当該構成に限定されるものではない。編集後の結合符号化画像505において隣り合うIDRピクチャが存在する可能性があれば、本願発明が適用可能である。即ち、IDRピクチャを含む動画像であれば、同様に本発

50

明を適用可能であり、本願発明の適用対象が全てIDRピクチャで構成される動画像だけに限定されることはない。

【0073】

以上の本実施形態によれば、符号化画像の結合後にピクチャ識別子が同一のIDRピクチャが連続してしまう場合であっても、単一のフレームのヘッダ内のデータを変更することにより、効率的にピクチャ識別子の変更を行うことができる。

【0074】

[実施形態3]

次に、図12及び図13を参照して実施形態1の変形例として、実施形態3を説明する。図13は、ヘッダ解析部108およびヘッダ書換部109の制御動作の流れを示したフローチャートである。図中、図3と同一の参照番号が振られているステップは、図3で説明した処理と同一処理が実行されるステップである。図12では、図3のS318の代わりに、S1201とS1202とを設けた点で処理が異なる。

10

【0075】

図13は、実施形態1の処理を行った後の編集符号化画像110を再度編集する場合の、削除符号化画像1305と再編集符号化画像1306のストリームの中身をフレーム単位で示した図である。図中、図4に示した要素に対応する要素については説明が共通するので同一の参照番号を付した。図13における編集符号化画像110は実施形態1の処理が既に行われた編集符号化画像である。図13における符号化画像中の削除範囲は、削除開始点1303から削除終了点1304である。図13における削除開始点1303の直前のフレーム(第1のフレーム)1301の*idr_pic_id*は2、削除終了点の直後のフレーム(第2のフレーム)401の*idr_pic_id*も2である。以下、実施形態1と異なる箇所のみ説明を行う。

20

【0076】

図12において、S1201では、ヘッダ書換部109は削除終了点の直後から2番目のフレーム(第3のフレーム)の*idr_pic_id*をヘッダ解析部108から取得する。取得した*idr_pic_id*をpとする。例えば、図13における削除終了点1304の直後から2番目のフレーム404の*idr_pic_id*を取得する。取得した*idr_pic_id*は7である。続くS1202では、ヘッダ書換部109は削除終了点の直後のフレームの*idr_pic_id*を変更する。但し、削除終了点の直後のフレームの*idr_pic_id*を変更する値は、削除開始点の直前のフレームの*idr_pic_id*のn、及び、削除終了点の直後から2番目のフレームの*idr_pic_id*のpとは異なる値に変更する。

30

【0077】

例えば、図13における削除終了点1304の直後のフレーム401の*idr_pic_id*を変更する場合を考える。このとき、変更後の値は、削除開始点の直前のフレーム1301の*idr_pic_id*の2、及び、削除終了点の直後から2番目のフレーム404の*idr_pic_id*の7とは異なる値でなくてはならず、図13では8に変更した場合を示している。以上の結果、*idr_pic_id*が2、*slice_type*が7のフレーム401を、*idr_pic_id*が8、*slice_type*が2のフレーム1302に変更した再編集符号化画像1306が生成される。

40

【0078】

なお、実施形態2においても、S718の代わりにS1201及びS1202を実行することで、結合点の直後のフレームの*idr_pic_id*と、その後続くフレームの*idr_pic_id*との重複を回避することができる。

【0079】

以上の本実施形態によれば、編集後の符号化画像において、ヘッダが書き換えられるフレームの後続フレームのピクチャ識別子との重複も防止することができる。

【0080】

以上、本発明の好ましい実施形態について説明したが、本発明はこれらの実施形態に限

50

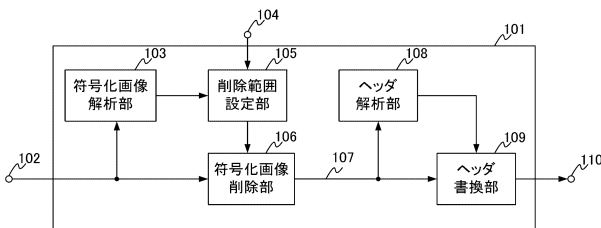
定されず、その要旨の範囲内で種々の変形及び変更が可能である。なお、実施形態1から3において説明した画像編集装置は、例えば、カメラ、携帯電話、パソコンなどの動画を取り扱うことができるものであればどのような装置であっても良い。

【0081】

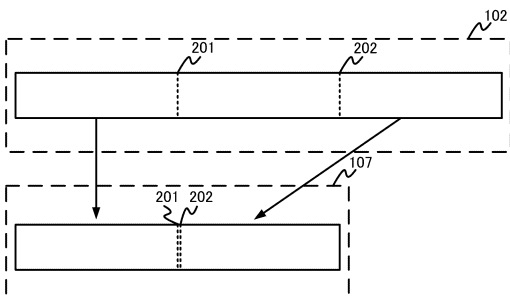
(その他の実施例)

また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア(プログラム)を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ(またはCPUやMPU等)がプログラムを読み出して実行する処理である。

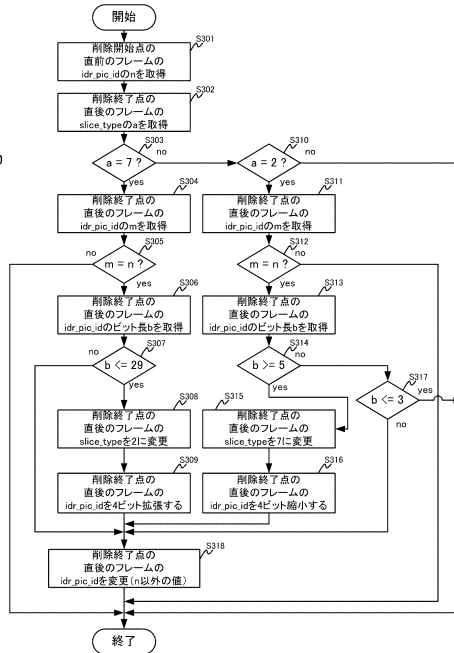
【図1】



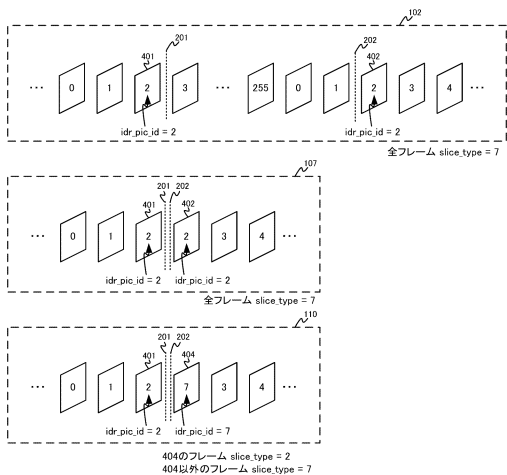
【図2】



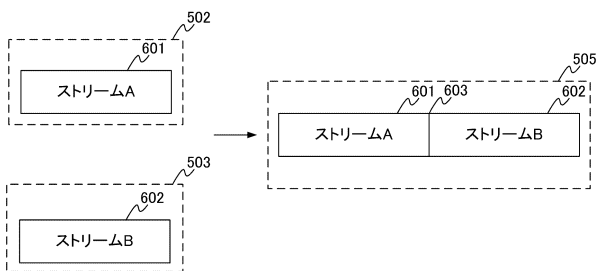
【図3】



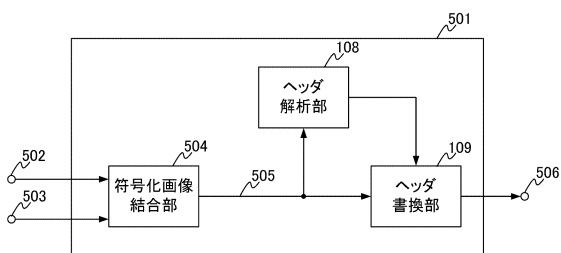
【図4】



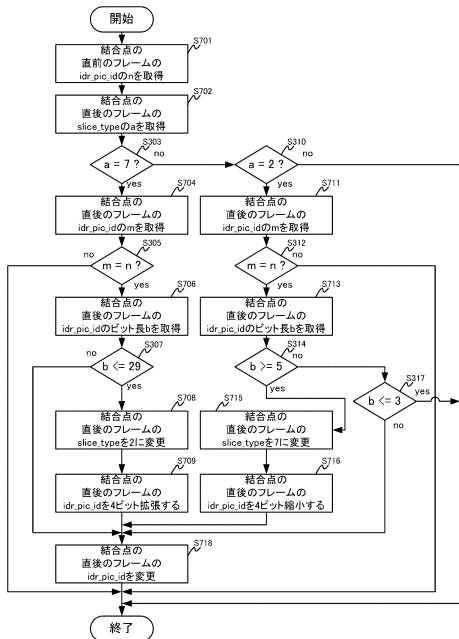
【図6】



【図5】



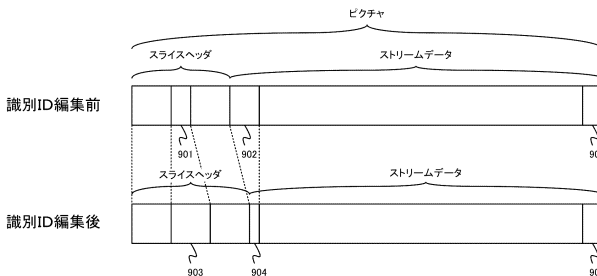
【図7】



【図8】

ゴロム符号	データ値の範囲
1	0
0 1 x	1-2
0 0 1 x x	3-6
0 0 0 1 x x x	7-14
0 0 0 0 1 x x x x	15-30
0 0 0 0 0 1 x x x x x	31-62
0 0 0 0 0 0 1 x x x x x x	63-126
0 0 0 0 0 0 0 1 x x x x x x x	127-254
0 0 0 0 0 0 0 0 1 x x x x x x x x	255-510

【図9】



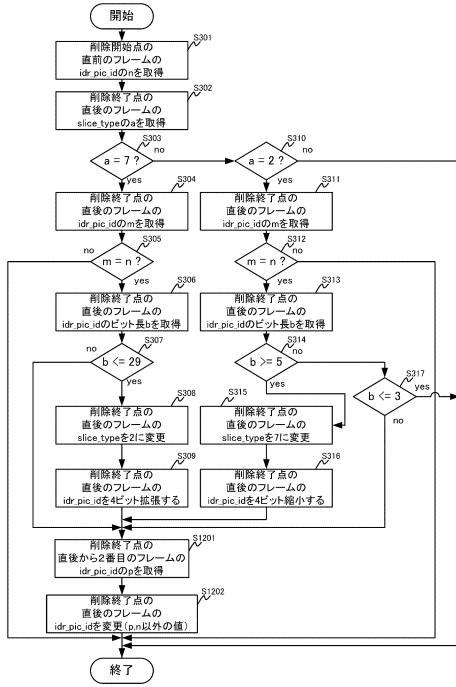
【図10】

	ゴロム符号	データ値
4ビット拡張前	0 1 1	2
4ビット拡張後	0 0 0 1 1 0 0	11

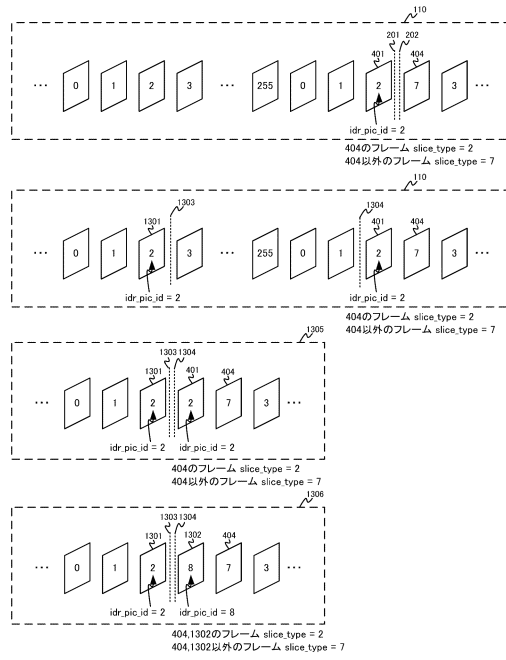
【図11】

	ゴロム符号	データ値
4ビット縮小前	0 0 0 1 1 0 1	12
4ビット縮小後	0 1 1	2

【図12】



【図13】



フロントページの続き

- (72)発明者 田中 英一
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 林 良典
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 大川 浩司
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 堀井 啓明

- (56)参考文献 特開2008-011518(JP,A)
特開2010-081227(JP,A)

- (58)調査した分野(Int.Cl., DB名)
H04N19/00-19/98