

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号

特許第7100200号

(P7100200)

(45)発行日 令和4年7月12日(2022.7.12)

(24)登録日 令和4年7月4日(2022.7.4)

(51)国際特許分類

F I

H 0 4 N 19/52 (2014.01)

H 0 4 N 19/52

請求項の数 17 (全45頁)

(21)出願番号	特願2021-523110(P2021-523110)	(73)特許権者	520476341
(86)(22)出願日	令和1年7月1日(2019.7.1)		北京字節跳動網絡技術有限公司
(65)公表番号	特表2021-530184(P2021-530184 A)		Beijing Bytedance Network Technology Co., Ltd.
(43)公表日	令和3年11月4日(2021.11.4)		中華人民共和國100041北京市石景山区実興大街30号院3号楼2層B-0035房間
(86)国際出願番号	PCT/IB2019/055582		Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District Beijing 100041 China
(87)国際公開番号	WO2020/003275		
(87)国際公開日	令和2年1月2日(2020.1.2)		
審査請求日	令和3年2月24日(2021.2.24)		
(31)優先権主張番号	PCT/CN2018/093663	(73)特許権者	520477474
(32)優先日	平成30年6月29日(2018.6.29)		バイトダンス インコーポレイテッド
(33)優先権主張国・地域又は機関	中国(CN)		最終頁に続く
(31)優先権主張番号	PCT/CN2018/105193		
(32)優先日	平成30年9月12日(2018.9.12)		
	最終頁に続く		

(54)【発明の名称】 マージ/AMVPにHMVP候補を追加する際の一部/全部ブルーニング

## (57)【特許請求の範囲】

## 【請求項1】

映像処理のための方法であって、

テーブルを維持することであって、前記テーブルは、コード化された1または複数の映像ブロックから導出された1または複数の候補を含み、前記テーブルにおける前記候補の配列は、前記テーブルへの前記候補の追加の順序に基づく、ことと、

候補リストを構成することであって、前記テーブルにおける少なくとも1つの第1の候補は、前記構成の間、チェックされ、ブルーニング工程が、前記少なくとも1つの第1の候補に利用されて前記第1の候補を前記候補リストに追加するか否かを判定する、ことと、前記候補リストを用いて動き情報を導出することと、

前記動き情報に基づいて現在の映像ブロックをコード化することと

を有し、

前記少なくとも1つの第1の候補は、前記候補リストに含まれている少なくとも1つの第2の候補にてブルーニングされ、

前記少なくとも1つの第2の候補の各候補は、前記各候補が導出された位置に基づいて前記候補リストから選択され、

前記第2の候補は、前記現在の映像ブロックの所定の空間的に近傍のブロックから導出される空間的候補であり、

前記少なくとも1つの第2の候補の総数は、Kであり、

Kは、2以下である、方法。

**【請求項 2】**

前記少なくとも 1 つの第 2 の候補は、前記候補リストにおける最初の 1 つの候補または N 個の候補を含み、

N は、1 よりも大きい整数である、請求項 1 に記載の方法。

**【請求項 3】**

前記少なくとも 1 つの第 2 の候補は、ブロック A 1 から導出された空間的候補、および、ブロック B 1 から導出された空間的候補の少なくとも一方を含み、

ブロック A 1 は、前記現在の映像ブロックに対する左の近傍のブロックであり、

ブロック B 1 は、前記現在の映像ブロックに対する上の近傍のブロックである、

請求項 1 に記載の方法。

10

**【請求項 4】**

前記少なくとも 1 つの第 2 の候補は、前記候補リストに存在する候補の一部である、請求項 1 に記載の方法。

**【請求項 5】**

前記候補リストの前記候補の一部は、空間的候補の一部である、請求項 4 に記載の方法。

**【請求項 6】**

前記少なくとも 1 つの第 1 の候補のうちの 1 つの候補が前記少なくとも 1 つの第 2 の候補のいずれとも異なる場合、当該候補は前記候補リストに追加される、請求項 1 に記載の方法。

**【請求項 7】**

前記少なくとも 1 つの第 2 の候補は、サブブロックベースの候補を除く、請求項 1 に記載の方法。

20

**【請求項 8】**

前記テーブルにおける前記候補は、予測方向、参照ピクチャインデックス、動きベクトル値、強度補償フラグ、アフィンフラグ、動きベクトル差分精度、および動きベクトル差分値の少なくともいずれかを含む動き情報に関連付けられている、請求項 1 に記載の方法。

**【請求項 9】**

前記現在の映像ブロックの前記動き情報を用いて前記テーブルを更新すること  
を更に有する、請求項 1 に記載の方法。

**【請求項 10】**

前記テーブルが、前記動き情報に対応する候補が前記候補リストに追加される前に満杯になった場合、前記テーブルの候補が除去される、請求項 9 に記載の方法。

30

**【請求項 11】**

前記候補リストは、マージ候補リストである、請求項 1 に記載の方法。

**【請求項 12】**

前記コード化することは、前記現在の映像ブロックをビットストリームに符号化することを有する、請求項 1 に記載の方法。

**【請求項 13】**

前記コード化することは、前記現在の映像ブロックをビットストリームから復号化することを有する、請求項 1 に記載の方法。

40

**【請求項 14】**

前記候補リストにおける候補の数が、（前記候補リストの最大許容候補数 - 閾値）に達した場合、チェックが終了され、

前記閾値は、正の整数である、請求項 1 に記載の方法。

**【請求項 15】**

プロセッサと、命令を有する非一時的メモリを有する、映像データをコード化するための装置であって、

前記命令は、前記プロセッサによって実行された際に、前記プロセッサに、

テーブルを維持することであって、前記テーブルは、現在の映像ブロックの前にコード化された 1 または複数の前にコード化された映像ブロックから導出された 1 または複数の候

50

補を含み、前記テーブルにおける前記候補の配列は、前記テーブルへの前記候補の追加の順序に基づく、ことと、

候補リストを構成することであって、前記テーブルを用いた前記構成の間、前記テーブルにおける少なくとも1つの第1の候補がチェックされ、ブルーニング工程が、前記テーブルにおける少なくとも1つの第1の候補に利用されて前記第1の候補を前記候補リストに追加するか否かを判定する、ことと、

前記候補リストを用いて動き情報を導出することと、

前記動き情報に基づいて前記現在の映像ブロックをコード化することと  
を行わせ、

前記少なくとも1つの第1の候補は、前記候補リストに含まれている少なくとも1つの第2の候補にてブルーニングされ、

前記少なくとも1つの第2の候補の各候補は、前記各候補が導出された位置に基づいて前記候補リストから選択され、

前記第2の候補は、前記現在の映像ブロックの所定の空間的に近傍のブロックから導出される空間的候補であり、

前記少なくとも1つの第2の候補の総数は、Kであり、  
Kは、2以下である、装置。

#### 【請求項16】

プロセッサに、

テーブルを維持することであって、前記テーブルは、現在の映像ブロックの前にコード化された1または複数の前にコード化された映像ブロックから導出された1または複数の候補を含み、前記テーブルにおける前記候補の配列は、前記テーブルへの前記候補の追加の順序に基づく、ことと、

候補リストを構成することであって、前記テーブルを用いた前記構成の間、前記テーブルにおける少なくとも1つの第1の候補がチェックされ、ブルーニング工程が、前記テーブルにおける少なくとも1つの第1の候補に利用されて前記第1の候補を前記候補リストに追加するか否かを判定する、ことと、

前記候補リストを用いて動き情報を導出することと、

前記動き情報に基づいて現在の映像ブロックを符号化することと  
を行わせ、

前記少なくとも1つの第1の候補は、前記候補リストに含まれている少なくとも1つの第2の候補にてブルーニングされ、

前記少なくとも1つの第2の候補の各候補は、前記各候補が導出された位置に基づいて前記候補リストから選択され、

前記第2の候補は、前記現在の映像ブロックの所定の空間的に近傍のブロックから導出される空間的候補であり、

前記少なくとも1つの第2の候補の総数は、Kであり、  
Kは、2以下である、

命令を格納した非一時的コンピュータ可読記憶媒体。

#### 【請求項17】

映像のビットストリームを格納するための方法であって、

前記方法は、

テーブルを維持することであって、前記テーブルは、コード化された1または複数の映像ブロックから導出された1または複数の候補を含み、前記テーブルにおける前記候補の配列は、前記テーブルへの前記候補の追加の順序に基づく、ことと、

候補リストを構成することであって、前記テーブルにおける少なくとも1つの第1の候補は、前記構成の間、チェックされ、ブルーニング工程が、前記少なくとも1つの第1の候補に利用されて前記第1の候補を前記候補リストに追加するか否かを判定する、ことと、  
前記候補リストを用いて動き情報を導出することと、

前記動き情報に基づいて、現在の映像ブロックからビットストリームを生成することと

10

20

30

40

50

を有し、  
前記少なくとも１つの第１の候補は、前記候補リストに含まれている少なくとも１つの第２の候補にてブルーニングされ、  
前記少なくとも１つの第２の候補の各候補は、前記各候補が導出された位置に基づいて前記候補リストから選択され、  
前記第２の候補は、前記現在の映像ブロックの所定の空間的に近傍のブロックから導出される空間的候補であり、  
前記少なくとも１つの第２の候補の総数は、 $K$ であり、  
 $K$ は、２以下である、方法。

【発明の詳細な説明】

【技術分野】

【０００１】

関連出願の相互参照

パリ条約に基づく適用可能な特許法および／または規則に基づいて、本願は、２０１８年６月２９日出願の国際特許出願 PCT/CN2018/093663 号、２０１８年９月１２日出願の国際特許出願 PCT/CN2018/105193 号、２０１８年１１月２７日出願の国際特許出願 PCT/CN2018/117627、および２０１９年１月１０日出願の国際特許出願 PCT/CN2019/071214 号の優先権および利益を適時に主張することを目的とする。米国の法律の下、あらゆる目的のために、国際特許出願 PCT/CN2018/093663 号、国際特許出願第 PCT/CN2018/105193 号、国際特許出願 PCT/CN2018/117627 号および国際特許出願第 PCT/CN2019/071214 の開示の全文は、本願の開示の一部として参照により援用される。

【０００２】

この特許明細書は、映像符号化および復号化技術、デバイスおよびシステムに関する。

【背景技術】

【０００３】

映像圧縮の進歩にもかかわらず、デジタル映像は、依然として、インターネットおよび他のデジタル通信ネットワークにおいて最大の帯域幅の使用量を占めている。映像の受信および表示が可能な接続されたユーザ機器の数が増加するにつれ、デジタル映像の使用に対する帯域幅需要は増大し続けることが期待される。

【発明の概要】

【０００４】

本明細書は、動きベクトルのマージリストを使用してデジタル映像を符号化および復号化するための方法、システム、およびデバイスを開示する。

【０００５】

１つの例示的な態様において、映像処理方法は、テーブルのセットを維持することであって、各テーブルは、動き候補を含み、各動き候補は、対応する動き情報に関連付けられる、ことと、動き候補へのブルーニング工程を使用して１つ以上のテーブルで動き候補に基づいて候補リストを更新することと、構成した候補リストを用いて、第１の映像ブロックと、前記第１の映像ブロックを含む映像のビットストリーム表現との間の変換を行うことと、を有する。

【０００６】

別の態様において、映像処理方法は、第１の映像ブロックを含む映像のビットストリーム表現を受信することと、候補リストを更新するために、１つまたは複数のテーブルにおける動き候補に対してブルーニング工程を適用することであって、各テーブルが動き候補を含み、各動き候補が対応する動き情報に関連付けられる、ことと、構成された候補リストを用いて、ビットストリーム表現と前記第１の映像ブロックとの間の変換を行うことと、を有するように提供される。

【０００７】

10

20

30

40

50

さらに別の代表的な態様では、本明細書で説明される様々な技法は、非一時的なコンピュータ可読媒体に記憶されるコンピュータプログラム製品として実施され得る。このコンピュータプログラム製品は、本明細書に記載の方法を実行するためのプログラムコードを含む。

【0008】

さらに別の代表的な態様では、映像符号化または復号化装置は、本明細書で説明される方法を実装し得る。

【0009】

1つ以上の実装形態の詳細は、添付の添付ファイル、図面、および以下の説明に記載されている。他の特徴は、説明および図面、並びに特許請求の範囲の記載から明らかとなろう。

10

【図面の簡単な説明】

【0010】

【図1】映像エンコーダの実装形態の例を示すブロック図である。

【図2】H.264映像符号化規格におけるマクロブロックの分割を示す。

【図3】符号化ブロック(CB: Coding Block)を予測ブロック(PU: Prediction Block)に分割する例を示す。

【図4】CTBをCBおよび変換ブロック(TB)に細分するための例示的な実装形態を示す。実線はCB境界を示し、点線はTB境界を示し、その分割を含むCTBの例、および対応する4分木を含む。

【図5】映像データを分割するための4分木2分木(QTBT: Quad Tree Binary Tree)構造の一例を示す。

20

【図6】映像ブロックの分割の例を示す。

【図7】4分木の分割の例を示す。

【図8】ツリー型信号通知の例を示す。

【図9】マージ候補リスト構築のための導出処理の一例を示す。

【図10】空間的マージ候補の位置の例を示す。

【図11】空間的マージ候補の冗長性チェックに考慮される候補対の例を示す。

【図12】 $N \times 2N$ および $2N \times N$ パーティションの第2のPUの位置の例を示す。

【図13】時間的マージ候補のための動きベクトルのスケーリングを示す。

【図14】時間的マージ候補の候補位置とその同一位置のピクチャを示す。

30

【図15】結合双方向予測マージ候補の例を示す。

【図16】動きベクトル予測候補の導出処理の例を示す。

【図17】空間的動きベクトル候補のための動きベクトルのスケーリングの例を示す。

【図18】CUの動き予測のための例示的なATMVP(Alternative Temporal Motion Vector Prediction)を示す。

【図19】ソースブロックおよびソースピクチャの識別の一例を絵で示す。

【図20】4つのサブブロックおよび近傍のブロックを有する1つのCUの例を示す。

【図21】バイラテラルマッチングの例を示す。

【図22】テンプレートマッチングの例を示す。

【図23】FRUC(Frame Rate Up Conversion)における一方の動き推定(ME: Motion Estimation)の例を示す。

40

【図24】バイラテラルテンプレートマッチングに基づくDMVRの例を示す。

【図25】空間的マージ候補を導出するために使用する空間的に近傍のブロックの例を示す。

【図26】ルックアップテーブル更新のための代表的な位置の選択方法の一例を示す。

【図27A】新しい動き情報のセットでルックアップテーブルを更新する例を示す。

【図27B】新しい動き情報のセットでルックアップテーブルを更新する例を示す。

【図28】本明細書に記載されるビジュアルメディアの復号化またはビジュアルメディアの符号化技術を実装するためのハードウェアプラットフォームの一例を示すブロック図である。

50

【図 29】映像処理方法の一例を示すフローチャートである。

【図 30】別の例示の映像処理方法を示すフローチャートである。

【図 31】提案された H M V P 方法による復号化フローチャートの一例を示す。

【図 32】提案される H M V P 方法を用いたテーブルの更新の例を示す。

【図 33 A】冗長性除去に基づく L U T 更新方法（1つの冗長性動き候補を除去する）の例を示す。

【図 33 B】冗長性除去に基づく L U T 更新方法（1つの冗長性動き候補を除去する）の例を示す。

【図 34 A】冗長性除去に基づく L U T 更新方法（複数の冗長性動き候補を除去する）の例を示す。

10

【図 34 B】冗長性除去に基づく L U T 更新方法（複数の冗長性動き候補を除去する）の例を示す。

【発明を実施するための形態】

【0011】

映像の圧縮率を改善するために、研究者らは、映像を符号化する新しい技術を絶えず求めている。

【0012】

1. 導入

【0013】

本明細書は、映像符号化技術に関する。具体的には、映像符号化における動き情報の符号化（例えば、マージモード、A M V P モード）に関する。H E V C のような既存の映像符号化規格に適用してもよいし、規格（Versatile Video Coding）を確定させるために適用してもよい。本発明は、将来の映像符号化規格または映像コーデックにも適用可能である。

20

【0014】

簡単な説明

【0015】

映像符号化規格は、主に周知の I T U - T および I S O / I E C 規格の開発によって発展してきた。I T U - T は H . 2 6 1 と H . 2 6 3 を作り、I S O / I E C は M P E G - 1 と M P E G - 4 V i s u a l を、両団体は H . 2 6 2 / M P E G - 2 V i d e o と H . 2 6 4 / M P E G - 4 A V C ( A d v a n c e d V i d e o C o d i n g ) と H . 2 6 5 / H E V C 規格を共同で作った。H . 2 6 2 以来、映像符号化規格は、時間予測と変換符号化が利用されるハイブリッド映像符号化構造に基づく。典型的な H E V C エンコーダフレームワークの一例を図 1 に示す。

30

【0016】

2.1 パーティション構造

【0017】

2.1.1 H . 2 6 4 / A V C におけるパーティションツリー構造

【0018】

以前の規格における符号化層のコアは、16×16ブロックの輝度サンプルを含み、通常の4:2:0カラーサンプリングの場合、2つの対応する8×8ブロックの彩度サンプル含むマクロブロックであった。

40

【0019】

イントラ符号化されたブロックは、画素間の空間的相関を利用するために空間予測を使用する。2つのパーティションを規定する。16×16および4×4である。

【0020】

インター符号化されたブロックは、ピクチャ間の動きを推定することで、空間的予測の代わりに時間予測を用いる。動きは、16×16マクロブロックまたはそのサブマクロブロックパーティションのいずれかに対して独立して推定できる。16×8、8×16、8×8、8×4、4×8、4×4（図2参照）。1つのサブマクロブロックパーティション当

50

たり1つの動きベクトル (MV: Motion Vector) のみが許可される。

【0021】

## 2.1.2 HEVCにおけるパーティションツリー構造

【0022】

HEVCにおいて、CTUは、様々な局所的特徴に適応するように、符号化ツリーと呼ばれる4分木構造を用いてCUに分割される。インターピクチャ(時間的)予測またはイントラピクチャ(空間的)予測を使用する、ピクチャ領域を符号化するかどうかの決定は、CUレベルで行われる。各CUは、PU分割タイプに応じて1つ、2つまたは4つのPUに更に分割することができる。1つのPUの内部では、同じ予測処理が適用され、PU単位で関連情報がデコーダに送信される。PU分割タイプに基づく予測処理を適用して残差ブロックを得た後、CUのための符号化ツリーに類似した別の4分木構造に基づいて、CUを変換ユニット(TU)に分割することができる。HEVC構造の重要な特徴の1つは、CU、PU、TUを含む複数のパーティション概念を有することである。

10

【0023】

以下、HEVCを使用したハイブリッド映像符号化に関連する様々な特徴に焦点を当てる。

【0024】

1) 符号化ツリーユニットおよび符号化ツリーブロック(CTB)構造。HEVCにおける類似した構造は、符号化ツリーユニット(CTU)であり、この符号化ツリーユニットは、エンコードによって選択されたサイズを有し、従来のマクロブロックよりも大きくてもよい。CTUは、輝度CTBと、対応する彩度CTBおよび構文要素とからなる。輝度CTBのサイズ $L \times L$ は、 $L = 16$ 、 $32$ 、または $64$ のサンプルとして選択することができ、より大きいサイズは、一般的に、より優れた圧縮を有効にする。HEVCは、次いで、ツリー構造および4分木様の信号通知を使用して、CTBをより小さなブロックに分割することをサポートする。

20

【0025】

2) 符号化ユニット(CU)および符号化ブロック(CB): CTUの4分木の構文は、その輝度および彩度CBのサイズおよび位置を指定する。4分木のルートはCTUに関連付けられる。従って、輝度CTBのサイズは、輝度CBに対してサポートされる最大のサイズである。CTUを輝度CBおよび彩度CBに分割することは、共に信号通知されることである。1つの輝度CBおよび通常2つの彩度CBは、関連する構文と共に、1つの符号化ユニット(CU)を形成する。CTBは、1つのCUのみを含んでもよく、または複数のCUを形成するように分割されてもよく、各CUは、それに関連付けられた予測ユニット(PU)への分割と、1つの変換ユニット(TU)のツリーとを有する。

30

【0026】

3) 予測ユニットおよび予測ブロック(PB): インターピクチャまたはイントラピクチャ予測を使用してピクチャ領域を符号化するかどうかの決定は、CUレベルで行われる。PUの分割構造は、そのルートがCUレベルにある。基本的な予測タイプの決定に基づいて、次に、輝度および彩度CBのサイズをさらに分割し、輝度および彩度予測ブロック(PB)から予測することができる。HEVCは、 $64 \times 64$ から $4 \times 4$ までの可変PBサイズのサンプルをサポートする。図3は、 $M \times M$ のCUのための許可されたPBの例を示す。

40

【0027】

4) TUおよび変換ブロック: 予測残差は、ブロック変換を使用して符号化される。TUツリー構造は、そのルートがCUレベルにある。この輝度CB残差は、輝度変換ブロック(TB)と同一であってもよいし、小さな輝度TBにさらに分割されてもよい。彩度TBについても同様である。正方形TBサイズ $4 \times 4$ 、 $8 \times 8$ 、 $16 \times 16$ 、および $32 \times 32$ に対して、離散コサイン変換(DCT)の整数基底関数に類似した整数基底関数が規定される。輝度イントラピクチャ予測残差の $4 \times 4$ 変換のために、離散サイン変換(DST)の形式から導出される整数変換が代替的に指定される。

【0028】

50

図 4 は、C T B を C B [ 及び変換ブロック ( T B ) ] に細分する例を示す。実線は C B 境界を示し、点線は T B 境界を示す。( a ) C T B とその分割 ( b ) 対応する 4 分木。

【 0 0 2 9 】

2 . 1 . 2 . 1 変換ブロックおよびユニットへのツリー構造の分割

【 0 0 3 0 】

残差符号化の場合、C B は、変換ブロック ( T B ) に再帰的に分割することができる。この分割は、残差 4 分木によって信号通知される。図 4 に示すように、1 つのブロックを再帰的に象限に分割することができるように、正方形の C B および T B の分割のみを指定する。サイズ  $M \times M$  の所与の輝度 C B に対して、フラグは、それがサイズ  $M / 2 \times M / 2$  の 4 つのブロックに分割されるかどうかを信号通知する。さらなる分割が可能である場合、S P S に示される残留 4 分木の最大深さによって信号通知されるように、各象限には、それが 4 つの象限に分割されているかどうかを示すフラグが割り当てられる。残差 4 分木の結果得られる葉ノードブロックは、変換符号化によってさらに処理される変換ブロックである。エンコードは、それが使用することになる最大輝度 T B サイズおよび最小輝度 T B サイズを示す。C B サイズが最大 T B サイズよりも大きい場合、分割は非明示的に行われる。分割により、示された最小値よりも小さい輝度 T B サイズとなる場合、分割は、非明示的に行われない。輝度 T B サイズが  $4 \times 4$  である場合を除き、彩度 T B サイズは、各次元において輝度 T B サイズの半分であり、この場合、4 つの  $4 \times 4$  輝度 T B によって覆われる領域には 1 つの  $4 \times 4$  彩度 T B が使用される。イントラピクチャ予測 C U の場合、最近の近傍の T B ( C B 内または C B 外 ) の復号サンプルを、イントラピクチャ予測のための参照データとして用いる。

【 0 0 3 1 】

従来の規格とは対照的に、H E V C 設計により、インターピクチャ予測 C U のために 1 つの T B が複数の P B にまたがるのが可能となり、4 分木構造の T B の分割の潜在的な符号化効率の利点が最大となる。

【 0 0 3 2 】

2 . 1 . 2 . 2 親子ノード

【 0 0 3 3 】

C T B は、4 分木構造に基づいて分割され、そのノードは符号化ユニットである。4 分木構造における複数のノードは、葉ノードおよび非葉ノードを含む。葉ノードは、ツリー構造内に子ノードを持たない (すなわち、葉ノードはそれ以上分割されない)。非葉ノードは、ツリー構造のルートノードを含む。ルートノードは、映像データの最初の映像ブロック (例えば、C T B ) に対応する。複数のノードのうちのそれぞれの非ルートノードにおいて、それぞれの非ルートノードは、それぞれの非ルートノードのツリー構造における親ノードに対応する映像ブロックのサブブロックである映像ブロックに対応する。複数の非葉ノードのそれぞれの非葉ノードは、ツリー構造において 1 つ以上の子ノードを有する。

【 0 0 3 4 】

2 . 1 . 3 J E M におけるより大きい C T U を有する 4 分木 + 2 分木ブロック構造

【 0 0 3 5 】

H E V C を超えた将来の映像符号化技術を探索するため、2 0 1 5 年には、V C E G と M P E G が共同で J V E T ( J o i n t V i d e o E x p l o r a t i o n T e a m ) を設立した。それ以来、多くの新しい方法が J V E T によって採用され、J E M ( J o i n t E x p l o r a t i o n M o d e ) と呼ばれる参照ソフトウェアに組み込まれてきた。

【 0 0 3 6 】

2 . 1 . 3 . 1 Q T B T ブロックの分割構造

【 0 0 3 7 】

H E V C とは異なり、Q T B T 構造は、複数のパーティションタイプ概念を削除する。すなわち、C U、P U、T U のコンセプトの切り離しを取り除き、C U パーティションの形状の柔軟性を向上させる。Q T B T ブロック構造において、C U は正方形または長方形

10

20

30

40

50



のいずれかを有することができる。図 5 に示すように、まず、符号化ツリーユニット (CTU) を 4 分木構造で分割する。4 分木の葉ノードは、2 分木構造によってさらに分割される。2 分木の分割には、対称水平分割と対称垂直分割の 2 つの分割タイプがある。2 分木の葉ノードは、符号化ユニット (CU) と呼ばれ、このセグメント化は、それ以上の分割を行うことなく、予測および変換処理に使用される。これは、QTB T の符号化されたブロック構造において、CU、PU および TU が同じブロックサイズを有することを意味する。JEM において、CU は、しばしば異なる色成分の符号化ブロック (CB) からなり、例えば、4 : 2 : 0 彩度フォーマットの P および B スライスの場合、1 つの CU は 1 つの輝度 CB および 2 つの彩度 CB を含み、また、CU は、しばしば単一の成分の CB からなり、例えば、I スライスの場合、1 つの CU は、1 つの輝度 CB のみ、または、2 つの彩度 CB のみを含む。

10

#### 【0038】

QTB T 分割スキームに対して以下のパラメータを規定する。

- CTU のサイズ：1 つの 4 分木のルートノードのサイズ、HEVC と同じ概念
- MinQTSize：最小許容の 4 分木の葉ノードサイズ
- MaxBTSize：最大許容の 2 分木のルートノードサイズ
- MaxBTDepth：最大許容の 2 分木の深さ
- MinBTSize：最小許容の 2 分木の葉ノードのサイズ

#### 【0039】

QTB T の分割構造の一例において、CTU のサイズを、2 つの対応する  $64 \times 64$  ブロックの彩度サンプルを有する  $128 \times 128$  の輝度サンプルとして設定し、MinQTSize を  $16 \times 16$  として設定し、MaxBTSize を  $64 \times 64$  として設定し、MinBTSize (幅および高さの両方について) を  $4 \times 4$  として設定し、MaxBTDepth を 4 として設定する。4 分木の分割は、まず CTU に適用され、4 分木の葉ノードを生成する。4 分木の葉ノードは、 $16 \times 16$  (即ち、MinQTSize) から  $128 \times 128$  (即ち、CTU サイズ) までのサイズを有することが可能である。葉 4 分木のノードが  $128 \times 128$  である場合、サイズが MaxBTSize (すなわち、 $64 \times 64$ ) を超えるため、2 分木によってさらに分割されない。そうでない場合、葉 4 分木のノードは、2 分木によってさらに分割されてもよい。従って、この 4 分木の葉ノードは、2 分木のルートノードでもあり、その 2 分木の深さは 0 である。2 分木の深さが MaxBTDepth (すなわち、4) に達した場合、それ以上の分割は考慮されない。2 分木のノードの幅が MinBTSize (すなわち、4 である) に等しい場合、それ以上の水平分割は考慮されない。同様に、2 分木のノードの高さが MinBTSize に等しい場合、それ以上の垂直分割は考慮されない。2 分木の葉ノードは、さらに分割することなく、予測および変換処理によってさらに処理される。JEM において、最大 CTU サイズは、 $256 \times 256$  個の輝度サンプルである。

20

30

#### 【0040】

図 5 (左) は QTB T を用いたブロックの分割の例を示し、図 5 (右) は対応するツリー表現を示す。実線は 4 分木の分割を表し、点線は 2 分木の分割を表す。2 分木の各分割 (即ち、非葉) ノードにおいて、1 つのフラグが、どの分割タイプ (即ち、水平または垂直) が使用されるかを示すために信号通知される。ここで、0 は、水平分割を表し、1 は、垂直分割を表す。4 分木の分割の場合、4 分木の分割は常にブロックを水平および垂直に分割し、等分したサイズの 4 つのサブブロックを生成するため、分割タイプを示す必要はない。

40

#### 【0041】

さらに、QTB T 方式は、輝度および彩度が別個の QTB T 構造を有する能力をサポートする。現在、P および B スライスの場合、1 つの CTU における輝度および彩度 CTB は、同じ QTB T 構造を共有する。しかしながら、I スライスの場合、輝度 CTB は QTB T 構造によって CU に分割され、彩度 CTB は別の QTB T 構造によって彩度 CU に分割される。これは、1 つの I スライスにおける 1 つの CU が 1 つの輝度成分の 1 つの符号化

50

ブロックまたは2つの彩度成分の1つの符号化ブロックからなり、1つのPまたはBスライスにおける1つのCUが3つの色成分すべての符号化ブロックからなることを意味する。

【0042】

HEVCにおいて、小さなブロックのためのインター予測は、動き補償のメモリアクセスを低減するために制限され、その結果、 $4 \times 8$ および $8 \times 4$ ブロックのために双方向予測はサポートされず、 $4 \times 4$ ブロックのためにインター予測はサポートされない。JEMのQTB Tにおいて、これらの制限は取り除かれる。

【0043】

2.1.4 VVCの3分木

【0044】

いくつかの実装形態において、4分木および2分木以外のツリータイプがサポートされる。本実装形態において、図6の(d)、(e)部分に示すように、3分木(TT)分割を2つ以上、すなわち、水平および垂直の中心側の3分木を導入する。

【0045】

図6は、(a)4分木分割、(b)垂直2分木分割、(c)水平2分木分割、(d)垂直中心側3分木分割、(e)水平中心側3分木分割を示す。

【0046】

いくつかの実装形態において、2つのレベルのツリー、すなわち、領域ツリー(4分木)および予測ツリー(2分木または3分木)がある。CTUは、まず、領域ツリー(RT)によって分割される。RTリーフは、予測ツリー(PT)によってさらに分割されてもよい。PT葉はまた、最大PT深さに達するまで、PTでさらに分割されてもよい。PTは基本符号化ユニットである。便宜上、ここでもCUと呼ぶ。1つのCUをさらに分割することはできない。予測および変換は両方ともJEMと同様にCUに適用される。パーティション構造全体を「マルチタイプツリー」と呼ぶ。

【0047】

2.1.5 [8]の分割構造

【0048】

この応答で使用するツリー構造は、マルチツリータイプ(Multi-Tree Type: MTT)と呼ばれ、QTB Tを一般化したものである。QTB Tにおいて、図5に示すように、まず、符号化ツリーユニット(CTU)を4分木構造で分割する。4分木の葉ノードは、2分木構造によってさらに分割される。

【0049】

MTTの基本構造は、2つのタイプのツリーノードを構成する。図7に示すように、領域ツリー(RT)および予測ツリー(PT)は、9つのタイプのパーティションをサポートする。

【0050】

図7は、(a)4分木分割、(b)垂直2分木分割、(c)水平2分木分割、(d)垂直3分木分割、(e)水平3分木分割、(f)水平上方非対称2分木分割、(g)水平下方非対称2分木分割、(h)垂直左非対称2分木分割、(i)垂直右非対称2分木分割を示す。

【0051】

1つの領域ツリーは、1つのCTUを $4 \times 4$ サイズの領域ツリーの葉ノードになるように正方形のブロックに再帰的に分割することができる。領域ツリーにおける各ノードにおいて、予測ツリーは、2分木(BT)、3分木(TT)、および非対称2分木(ABT)の3つのツリータイプのうちの1つから形成されることができる。PT分割において、予測ツリーの枝に4分木のパーティションを有することは禁止される。JEMにおけるように、輝度ツリーおよび彩度ツリーは、I個のスライスに分けられる。RTおよびPTの信号通知方法を図8に示す。

【0052】

2.2 HEVC/H.265におけるインター予測

10

20

30

40

50

## 【 0 0 5 3 】

各インター予測されたPUは、1つまたは2つの参照ピクチャリストのための動きパラメータを有する。動きパラメータは、動きベクトルおよび参照ピクチャインデックスを含む。2つの参照ピクチャリストのうちの1つの参照ピクチャリストの使用は、`inter_pred_idc`を使用して信号通知されてもよい。動きベクトルは、予測因子に関連する差分として明確に符号化されてもよく、このような符号化モードは、AMVPモードと呼ばれる。

## 【 0 0 5 4 】

1つのCUがスキップモードで符号化される場合、1つのPUがこのCUに関連付けられ、有意な残差係数がなく、符号化された動きベクトル差分も参照ピクチャインデックスもない。マージモードを指定し、これにより、現在のPUのための動きパラメータを、空間的および時間的候補を含む近傍のPUから取得する。マージモードは、スキップモードのためだけでなく、任意のインター予測されたPUに適用することができる。マージモードの代替としては、動きパラメータの明確な送信であり、各参照ピクチャリストおよび参照ピクチャリストの使用に対する参照ピクチャインデックスに対応する動きベクトルをPUごとに明確に信号通知することである。

10

## 【 0 0 5 5 】

2つの参照ピクチャリストのうちの1つを使用することを信号通知が示す場合、サンプルのうちの1つのブロックからPUを生成する。これを「単一予測」と呼ぶ。PスライスおよびBスライスの両方に対して単一予測が利用可能である。

20

## 【 0 0 5 6 】

両方の参照ピクチャリストを使用することを信号通知が示す場合、サンプルのうちの2つのブロックからPUを生成する。これを「双方向予測」と呼ぶ。Bスライスのみに双方向予測が利用可能である。

## 【 0 0 5 7 】

以下、HEVCに規定されるインター予測モードについて詳細に説明する。まず、マージモードについて説明する。

## 【 0 0 5 8 】

## 2.2.1 マージモード

## 【 0 0 5 9 】

## 2.2.1.1 マージモードの候補の導出

30

## 【 0 0 6 0 】

マージモードを使用してPUを予測する場合、ビットストリームからマージ候補リストにおけるエントリを指すインデックスを構文解析し、これを使用して動き情報を検索する。このリストの構成は、HEVC規格で規定されており、以下のステップのシーケンスに基づいてまとめることができる。

- ・ステップ1：初期候補の導出
  - ステップ1.1：空間的候補の導出
  - ステップ1.2：空間的候補の冗長性チェック
  - ステップ1.3：時間的候補の導出
- ・ステップ2：追加候補の挿入
  - ステップ2.1：双方向予測候補の作成
  - ステップ2.2：動きゼロ候補の挿入

40

## 【 0 0 6 1 】

これらのステップは図9にも概略的に示されている。空間的マージ候補導出のために、5つの異なる位置にある候補の中から最大4つのマージ候補を選択する。時間的マージ候補導出のために、2つの候補の中から最大1つのマージ候補を選択する。デコーダ側ではPUごとに一定数の候補を想定しているので、候補数がスライスヘッダで信号通知されるマージ候補の最大数 (`MaxNumMergeCand`) に達しない場合、追加候補を生成する。候補の数は一定であるので、最良マージ候補のインデックスは、短縮された単項2

50

値化 (TU: truncated unary binarization) を使用して符号化される。CUのサイズが8に等しい場合、現在のCUのすべてのPUは、 $2N \times 2N$  予測ユニットのマージ候補リストと同じ1つのマージ候補リストを共有する。

【0062】

以下、上述したステップに関連付けられた動作を詳しく説明する。

【0063】

2.2.1.2 空間的候補の導出

【0064】

空間的マージ候補の導出において、図10に示す位置にある候補の中から、最大4つのマージ候補を選択する。導出の順序はA<sub>1</sub>、B<sub>1</sub>、B<sub>0</sub>、A<sub>0</sub>、B<sub>2</sub>である。位置A<sub>1</sub>、B<sub>1</sub>、B<sub>0</sub>、A<sub>0</sub>のいずれかのPUが利用可能でない場合（例えば、別のスライスまたはタイルに属しているため）、またはイントラ符号化された場合にのみ、位置B<sub>2</sub>が考慮される。位置A<sub>1</sub>の候補を加えた後、残りの候補を加えると、冗長性チェックを受け、それにより、同じ動き情報を有する候補を確実にリストから排除でき、符号化効率を向上させることができる。計算の複雑性を低減するために、前述の冗長性チェックにおいて、考えられる候補対のすべてを考慮することはしない。代わりに、図11において矢印でリンクされた対のみを考慮し、冗長性チェックに使用される対応する候補が同じ動き情報を有していない場合にのみ、その候補をリストに加える。重複した動き情報の別のソースは、 $2N \times 2N$ とは異なるパーティションに関連付けられた「第2のPU」である。一例として、図12は、それぞれ $N \times 2N$ および $2N \times N$ の場合の第2のPUを示す。現在のPUを $N \times 2N$ に分割する場合、リスト構成に位置A<sub>1</sub>の候補は考慮されない。実際、この候補を加えることにより、同じ動き情報を有する2つの予測ユニットが導かれることとなり、1つの符号化ユニットに1つのPUのみを有するためには冗長である。同様に、現在のPUを $2N \times N$ に分割する場合、位置B<sub>1</sub>は考慮されない。

【0065】

2.2.1.3 時間的候補の導出

【0066】

このステップにおいて、1つの候補のみがリストに追加される。具体的には、この時間的マージ候補の導出において、所与の参照ピクチャリストにおける現在のピクチャとの間に最小のPOC差を有するピクチャに属する同一位置のPU (co-located PU) に基づいて、スケーリングされた動きベクトルを導出する。スライスヘッダにおいて、同一位置のPUの導出に用いられる参照ピクチャリストが明確に信号通知される。図13に点線で示すように、時間的マージ候補のスケーリングされた動きベクトルが得られる。これは、POC距離 $t_b$ および $t_d$ を利用して、同一位置のPUの動きベクトルからスケーリングしたものである。 $t_b$ は、現在のピクチャの参照ピクチャと現在のピクチャのPOC差として規定され、 $t_d$ は、同一位置のPUの参照ピクチャと同一位置のピクチャのPOC差として規定する。時間的マージ候補の参照ピクチャインデックスをゼロに等しく設定する。このスケーリング処理の実際的な実現については、HEVC仕様[1]に記載されている。Bスライスの場合、2つの動きベクトル、即ち、1つは参照ピクチャリスト0のためのもの、もう1つは参照ピクチャリスト1のためのものを取得し、これらを組み合わせることによって、双方向予測マージ候補を形成する。時間的マージ候補のための動きベクトルのスケーリングの説明。

【0067】

参照フレームに属する同一位置のPU (Y) において、図14に示すように、候補C<sub>0</sub>と候補C<sub>1</sub>との間で時間的候補の位置を選択する。位置C<sub>0</sub>のPUが利用可能でない場合、イントラ符号化されている場合、または現在のCTUの外側にある場合、位置C<sub>1</sub>が使用される。そうでない場合、位置C<sub>0</sub>が時間的マージ候補の導出に使用される。

【0068】

2.2.1.4 追加候補の挿入

【0069】

10

20

30

40

50

空間的 - 時間的マージ候補の他に、2つの追加のタイプのマージ候補、すなわち、結合双方向予測マージ候補およびゼロマージ候補がある。空間的 - 時間的マージ候補を利用して、結合双方向予測マージ候補を生成する。結合双方向予測マージ候補は、Bスライスのみで使用される。最初の候補の第1の参照ピクチャリスト動きパラメータと別の候補の第2の参照ピクチャリスト動きパラメータとを組み合わせることで、結合双方向予測候補を生成する。これら2つのタプルが異なる動きの仮説を提供する場合、これらのタプルは、新しい双方向予測候補を形成する。一例として、図15は、オリジナルリスト（左側）における、 $mvL0$ および $refIdxL0$ 、または $mvL1$ および $refIdxL1$ を有する2つの候補を用いて、最終リスト（右側）に加えられる結合双方向予測マージ候補を生成する場合を示す。これらの追加のマージ候補を生成するために考慮される組み合わせについては、様々な規則が存在する。

10

#### 【0070】

ゼロ動き候補を挿入し、マージ候補リストにおける残りのエントリを埋めることにより、 $MaxNumMergeCand$ 容量にヒットする。これらの候補は、空間的変位がゼロであり、新しいゼロ動き候補をリストに加える度にゼロから始まり増加する参照ピクチャインデックスを有する。これらの候補が使用する参照フレームの数は、それぞれ、一方向予測の場合は1つ、双方向予測の場合は2つである。最終的には、これらの候補に対して冗長性チェックは行われない。

#### 【0071】

##### 2.2.1.5 並列処理のための動き推定領域

20

#### 【0072】

符号化処理を高速化するために、動き推定を並列に行うことができ、それによって、所与の領域内のすべての予測ユニットの動きベクトルを同時に導出する。1つの予測ユニットは、その関連する動き推定が完了するまで、隣接するPUから動きパラメータを導出することができないので、空間的近傍からのマージ候補の導出は、並列処理に干渉する可能性がある。符号化効率と処理待ち時間との間のトレードオフを緩和するために、HEVCは、動き推定領域（MER: Motion Estimation Region）を規定し、そのサイズは、「 $\log_2\_parallel\_merge\_level\_minus\_2$ 」構文要素を使用してピクチャパラメータセットにおいて信号通知される。1つのMERを規定するとき、同じ領域にあるマージ候補は使用不可としてマークされ、それゆえにリスト構築においては考慮されない。

30

##### 7.3.2.3 ピクチャパラメータセットRBSP構文

##### 7.3.2.3.1 一般ピクチャパラメータセットRBSP構文

#### 【0073】

40

50

【表 1】

<b>pic_parameter_set_rbsp()</b> {	記述子
<b>pps_pic_parameter_set_id</b>	ue(v)
<b>pps_seq_parameter_set_id</b>	ue(v)
<b>dependent_slice_segments_enabled_flag</b>	u(1)
...	
<b>pps_scaling_list_data_present_flag</b>	u(1)
if( pps_scaling_list_data_present_flag )	
scaling_list_data()	
<b>lists_modification_present_flag</b>	u(1)
<b>log2_parallel_merge_level_minus2</b>	ue(v)
<b>slice_segment_header_extension_present_flag</b>	u(1)
<b>pps_extension_present_flag</b>	u(1)
...	
rbsp_trailing_bits()	
}	

10

20

## 【0074】

$\log 2\_parallel\_merge\_level\_minus 2 + 2$  は、8.5.3.2.2.2 節で指定されたマージモードの輝度動きベクトルの導出処理と、8.5.3.2.3 節で指定された空間的マージ候補の導出処理で使用される変数  $\log 2\_Parallel\_Merge\_Level$  の値を指定する。 $\log 2\_parallel\_merge\_level\_minus 2$  の値は、 $0 \sim CtbLog 2SizeY - 2$  を含む範囲内とする。

変数  $\log 2\_Parallel\_Merge\_Level$  は、以下のように導出される。

30

$$\log 2\_Parallel\_Merge\_Level = \log 2\_parallel\_merge\_level\_minus 2 + 2 \quad (7 - 37)$$

注3： $\log 2\_Parallel\_Merge\_Level$  の値は、マージ候補リストを並列に導出する組み込み能力を示す。例えば、 $\log 2\_Parallel\_Merge\_Level$  が6に等しい場合、 $64 \times 64$  ブロックに含まれたすべての予測ユニット (PU) および符号化ユニット (CU) のためのマージ候補リストを並列に導出することができる。

## 【0075】

2.2.2 AMVPモードにおける動きベクトル予測

## 【0076】

動きベクトル予測は、動きベクトルと近傍のPUとの間の空間的 - 時間的相関を利用し、これを動きパラメータの明確な伝送に用いる。まず、左側、上側の時間的に近傍のPU位置の可用性をチェックし、冗長な候補を取り除き、ゼロベクトルを加えることで、候補リストの長さを一定にすることで、動きベクトル候補リストを構築する。次いで、エンコーダは、候補リストから最良の予測子を選択し、選択された候補を示す対応するインデックスを送信することができる。マージインデックスの信号通知と同様に、最良の動きベクトル候補のインデックスは、短縮された単項を使用して符号化される。この場合のエンコーディング対象の最大値は2である (例えば、図2～図8)。以下の章では、動きベクトル予測候補の導出処理の詳細を説明する。

40

## 【0077】

2.2.2.1 動きベクトル予測候補の導出

50

## 【 0 0 7 8 】

図 1 6 に、動きベクトル予測候補の導出処理をまとめる。

## 【 0 0 7 9 】

動きベクトル予測において、空間的動きベクトル候補と時間的動きベクトル候補という 2 つのタイプの動きベクトル候補が考慮される。空間的動きベクトル候補の導出のために、図 1 1 に示したように、5 つの異なる位置にある各 P U の動きベクトルに基づいて、最終的には 2 つの動きベクトル候補を導出する。

## 【 0 0 8 0 】

時間的動きベクトル候補の導出のために、2 つの異なる同一位置の配置に基づいて導出された 2 つの候補から 1 つの動きベクトル候補を選択する。空間的 - 時間的候補の最初のリストを作成した後、リストにおける重複した動きベクトル候補を除去する。可能性のある候補の数が 2 よりも多い場合、関連づけられた参照ピクチャリストにおける参照ピクチャインデックスが 1 よりも大きい動きベクトル候補をリストから削除する。空間的 時間的動きベクトル候補の数が 2 未満である場合は、追加のゼロ動きベクトル候補をリストに加える。

10

## 【 0 0 8 1 】

2 . 2 . 2 . 2 空間的動きベクトル候補

## 【 0 0 8 2 】

空間的動きベクトル候補の導出において、図 1 1 に示したような位置にある P U から導出された 5 つの可能性のある候補のうち、動きマージと同じ位置にあるものを最大 2 つの候補を考慮する。現在の P U の左側のための導出の順序は、A<sub>0</sub>、A<sub>1</sub>、スケーリングされた A<sub>0</sub>、スケーリングされた A<sub>1</sub> として規定される。現在の P U の上側のための導出の順序は、B<sub>0</sub>、B<sub>1</sub>、B<sub>2</sub>、スケーリングされた B<sub>0</sub>、スケーリングされた B<sub>1</sub>、スケーリングされた B<sub>2</sub> として規定される。そのため、辺ごとに、動きベクトル候補として使用できる場合が 4 つ、すなわち空間的スケーリングを使用する必要がない 2 つの場合と、空間的スケーリングを使用する 2 つの場合とがある。4 つの異なる場合をまとめると、以下のようになる。

20

・空間的スケーリングなし

- ( 1 ) 同じ参照ピクチャリスト、かつ、同じ参照ピクチャインデックス ( 同じ P O C )

- ( 2 ) 異なる参照ピクチャリスト、かつ、同じ参照ピクチャ ( 同じ P O C )

30

・空間的スケーリング

- ( 3 ) 同じ参照ピクチャリスト、かつ、異なる参照ピクチャ ( 異なる P O C )

- ( 4 ) 異なる参照ピクチャリスト、かつ、異なる参照ピクチャ ( 異なる P O C )

## 【 0 0 8 3 】

最初に非空間的スケーリングの場合をチェックし、次に空間的スケーリングを行う。参照ピクチャリストにかかわらず、P O C が近傍の P U の参照ピクチャと現在の P U の参照ピクチャとで異なる場合、空間的スケーリングを考慮する。左側候補のすべての P U が利用可能でないか、またはイントラ符号化されている場合、上側の動きベクトルのスケーリングは、左側および上側の M V 候補の並列導出に役立つ。そうでない場合、上側の動きベクトルに対して空間的スケーリングは許可されない。

40

## 【 0 0 8 4 】

空間的スケーリング処理において、図 1 7 に示すように、時間的スケーリングと同様にして、近傍の P U の動きベクトルをスケーリングする。主な違いは、現在の P U の参照ピクチャリストおよびインデックスを入力として与え、実際のスケーリング処理は時間的スケーリングと同じであることである。

## 【 0 0 8 5 】

2 . 2 . 2 . 3 時間的動きベクトル候補

## 【 0 0 8 6 】

参照ピクチャインデックスを導出する以外は、時間的マージ候補を導出するための処理は、すべて、空間的動きベクトル候補を導出するための処理と同じである ( 図 6 参照 ) 。参

50

照ピクチャインデックスはデコーダに信号通知される。

【 0 0 8 7 】

2 . 2 . 2 . 4 A M V P 情報の信号通知

【 0 0 8 8 】

A M V P モードの場合、ビットストリームにおいて、4つの部分、すなわち、予測方向、参照インデックス、M V D、およびm v 予測因子候補インデックスが信号通知される。

構文テーブル：

【 0 0 8 9 】

【 表 2 】

prediction_unit( x0, y0, nPbW, nPbH ) {	記述子
if( cu_skip_flag[ x0 ][ y0 ] ) {	
if( MaxNumMergeCand > 1 )	
<b>merge_idx</b> [ x0 ][ y0 ]	ae(v)
} else { /* MODE_INTER */	
<b>merge_flag</b> [ x0 ][ y0 ]	ae(v)
if( merge_flag[ x0 ][ y0 ] ) {	
if( MaxNumMergeCand > 1 )	
<b>merge_idx</b> [ x0 ][ y0 ]	ae(v)
} else {	
if( slice_type == B )	
<b>inter_pred_idc</b> [ x0 ][ y0 ]	ae(v)
if( inter_pred_idc[ x0 ][ y0 ] != PRED_L1 ) {	
if( num_ref_idx_l0_active_minus1 > 0 )	
<b>ref_idx_l0</b> [ x0 ][ y0 ]	ae(v)
mvd_coding( x0, y0, 0 )	
<b>mvp_l0_flag</b> [ x0 ][ y0 ]	ae(v)
}	
if( inter_pred_idc[ x0 ][ y0 ] != PRED_L0 ) {	
if( num_ref_idx_l1_active_minus1 > 0 )	
<b>ref_idx_l1</b> [ x0 ][ y0 ]	ae(v)
if( mvd_l1_zero_flag && inter_pred_idc[ x0 ][ y0 ] == PRED_BI )	
{	
MvdL1[ x0 ][ y0 ][ 0 ] = 0	
MvdL1[ x0 ][ y0 ][ 1 ] = 0	
} else	
mvd_coding( x0, y0, 1 )	
<b>mvp_l1_flag</b> [ x0 ][ y0 ]	ae(v)
}	
}	
}	
}	

【 0 0 9 0 】



## 7.3.8.9 動きベクトル差構文

【0091】

【表3】

<b>mvd_coding( x0, y0, refList ) {</b>	記述子
<b>abs_mvd_greater0_flag[ 0 ]</b>	ae(v)
<b>abs_mvd_greater0_flag[ 1 ]</b>	ae(v)
<b>if( abs_mvd_greater0_flag[ 0 ] )</b>	
<b>abs_mvd_greater1_flag[ 0 ]</b>	ae(v)
<b>if( abs_mvd_greater0_flag[ 1 ] )</b>	
<b>abs_mvd_greater1_flag[ 1 ]</b>	ae(v)
<b>if( abs_mvd_greater0_flag[ 0 ] ) {</b>	
<b>if( abs_mvd_greater1_flag[ 0 ] )</b>	
<b>abs_mvd_minus2[ 0 ]</b>	ae(v)
<b>mvd_sign_flag[ 0 ]</b>	ae(v)
<b>}</b>	
<b>if( abs_mvd_greater0_flag[ 1 ] ) {</b>	
<b>if( abs_mvd_greater1_flag[ 1 ] )</b>	
<b>abs_mvd_minus2[ 1 ]</b>	ae(v)
<b>mvd_sign_flag[ 1 ]</b>	ae(v)
<b>}</b>	
<b>}</b>	

【0092】

2.3 JEM ( Joint Exploration Model ) における新しいインター予測方法

【0093】

2.3.1 サブCUに基づく動きベクトル予測

【0094】

Q T B Tを有するJEMにおいて、各CUは、各予測方向に対して最大1つの動きパラメータのセットを有することができる。エンコードにおいて、大きなCUをサブCUに分割し、大きなCUのすべてのサブCUの動き情報を導出することにより、2つのサブCUレベルの動きベクトル予測方法を考慮する。ATMVP ( Alternative Temporal Motion Vector Prediction ) 方法により、各CUが、配列された参照ピクチャにおける現在のCUよりも小さい複数のブロックから複数の動き情報のセットをフェッチすることが可能となる。STMVP ( Spatial - Temporal Motion Vector Prediction ) 法において、時間的動きベクトル予測因子および空間的近傍動きベクトルを使用して、サブCUの動きベクトルを再帰的に導出する。

【0095】

サブCU動き予測のためにより正確な動きフィールドを維持するために、参照フレームの動き圧縮は現在無効にされている。

【0096】

2.3.1.1 代替の時間的動きベクトル予測

【0097】

ATMVP ( Alternative Temporal Motion Vector

Prediction)において、TMVP(Temporal Motion Vector Prediction)法は、現在のCUより小さいブロックから複数セットの動き情報(動きベクトルおよび参照インデックスを含む)をフェッチすることで修正される。図18に示すように、サブCUは、正方形の $N \times N$ ブロックである(デフォルトでは、 $N$ は4に設定される)。

#### 【0098】

ATMVPは、CU内のサブCUの動きベクトルを2つのステップで予測する。第1のステップは、参照ピクチャにおける対応するブロックを、いわゆる時間的ベクトルで特定することである。この参照ピクチャを動きソースピクチャと呼ぶ。第2のステップは、図18に示すように、現在のCUをサブCUに分割し、各サブCUに対応するブロックから各サブCUの動きベクトルならびに参照インデックスを取得する。

10

#### 【0099】

第1のステップにおいて、現在のCUの空間的に近傍のブロックの動き情報によって、参照ピクチャおよび対応するブロックを決定する。近傍のブロックの繰り返し走査処理を回避するために、現在のCUのマージ候補リストにおける最初のマージ候補を用いる。第1の利用可能な動きベクトルおよびその関連する参照インデックスを、時間的ベクトルおよび動きソースピクチャのインデックスに設定する。このように、ATMVPでは、TMVPに比べて、対応するブロックをより正確に特定することができ、対応するブロック(配列されたブロックと呼ばれることがある)は、常に現在のCUに対して右下または中心位置にある。1つの例において、最初のマージ候補が左側の近傍のブロック(即ち、図19の $A_1$ )からのものである場合、関連するMVおよび参照ピクチャを利用して、ソースブロックおよびソースピクチャを特定する。

20

#### 【0100】

図19は、ソースブロックおよびソースピクチャの特定の例を示す。

#### 【0101】

第2のステップにおいて、現在のCUの座標に時間ベクトルを加えることで、動きソースピクチャにおける時間的ベクトルによって、サブCUの対応するブロックを特定する。サブCUごとに、その対応するブロックの動き情報(中心サンプルを覆う最小の動きグリッド)を使用して、サブCUの動き情報を導出する。対応する $N \times N$ ブロックの動き情報を特定した後、HEVCのTMVPと同様に、現在のサブCUの動きベクトルおよび参照インデックスに変換され、動きスケーリングや他の手順が適用される。例えば、デコーダは、低遅延条件(すなわち、現在のピクチャのすべての参照ピクチャのPOCが現在のピクチャのPOCよりも小さい)が満たされているかどうかをチェックし、場合によっては、動きベクトル $MV_x$ (参照ピクチャリスト $X$ に対応する動きベクトル)を使用して、各サブCUの動きベクトル $MV_y$ ( $X$ が0または1に等しく、 $Y$ が $1 - X$ に等しい)を予測する。

30

#### 【0102】

### 2.3.1.2 空間的 - 時間的動きベクトル予測

#### 【0103】

この方法において、サブCUの動きベクトルは、ラスタスキャンの順に沿って再帰的に導出される。図20にこの概念を示す。4つの $4 \times 4$ サブCUである、A、B、C、およびDを含む $8 \times 8$ CUを考える。現在のフレームの近傍の $4 \times 4$ ブロックには、a、b、c、dというラベルが付けられている。

40

#### 【0104】

サブCU Aの動き導出は、その2つの空間的近傍を特定することで始まる。第1の近傍は、サブCU Aの上の $N \times N$ ブロックである(ブロックc)。このブロックcが利用可能でないか、またはイントラ符号化されている場合、サブCU Aより上の他の $N \times N$ 個のブロックをチェックする(ブロックcから始まり、左から右へ)。第2の近傍は、サブCU Aの左側のブロックである(ブロックb)。ブロックbが利用可能でないか、またはイントラ符号化されている場合、サブCU Aの左側の他のブロックをチェックする(

50

ブロック b から始まり、上から下へ)。各リストの近傍のブロックから得られた動き情報を、所与のリストの第 1 の参照フレームにスケーリングする。次に、H E V C に規定されている T M V P ( T e m p o r a l M o t i o n V e c t o r P r e d i c t o r ) 導出と同様の手順に従って、サブブロック A の T M V P を導出する。位置 D における配列されたブロックの動き情報をフェッチし、それに応じてスケーリングする。最後に、動き情報を検索し、スケーリングした後、参照リストごとにすべての利用可能な動きベクトル ( 3 まで ) を別々に平均する。この平均化された動きベクトルを現在のサブ C U の動きベクトルとする。

【 0 1 0 5 】

図 2 0 は、4 つのサブブロック ( A - D ) およびその近傍のブロックを有する 1 つの C U の例を示す。

【 0 1 0 6 】

2 . 3 . 1 . 3 サブ C U 動き予測モード信号通知

【 0 1 0 7 】

サブ C U モードは追加のマージ候補として有効とされ、モードを信号通知するために追加の構文要素は必要とされない。A T M V P モードおよび S T M V P モードを表すように、各 C U のマージ候補リストに 2 つの追加のマージ候補を加える。シーケンスパラメータセットが A T M V P および S T M V P が有効であることを示す場合、7 個までのマージ候補を使用する。追加のマージ候補の符号化ロジックは、H M におけるマージ候補の場合と同じであり、つまり、P または B スライスにおける各 C U について、2 つの追加のマージ候補に対して 2 回以上の R D チェックが必要となる。

【 0 1 0 8 】

J E M において、マージインデックスのすべてのピンは、C A B A C によって符号化されたコンテキストである。一方、H E V C においては、最初のピンのみが符号化されたコンテキストであり、残りのピンはバイパス符号化されたコンテキストである。

【 0 1 0 9 】

2 . 3 . 2 適応型動きベクトル差分解像度

【 0 1 1 0 】

H E V C において、`use_integer_mv_flag` がスライスヘッダにおいて 0 であるとき、 $1 / 4$  輝度サンプルの単位で動きベクトルの差分 ( M V D : M o t i o n V e c t o r D i f f e r e n c e ) ( 動きベクトルと P U の予測動きベクトルとの差 ) が信号通知される。J E M において、L A M V R ( L o c a l l y A d a p t i v e M o t i o n V e c t o r R e s o l u t i o n ) が導入される。J E M において、M V D は、 $1 / 4$  輝度サンプル、整数輝度サンプルまたは 4 つの輝度サンプルの単位でコーディングできる。M V D 解像度は符号化ユニット ( C U ) レベルで制御され、M V D 解像度フラグは、少なくとも 1 つの非ゼロ M V D の構成要素を有する各 C U に対して条件付きで信号通知される。

【 0 1 1 1 】

少なくとも 1 つの非ゼロ M V D の構成要素を有する C U の場合、 $1 / 4$  輝度サンプル M V 精度が C U において使用されるか否かを示すために、第 1 のフラグが信号通知される。第 1 のフラグ ( 1 に等しい ) が、 $1 / 4$  輝度サンプル M V 精度が使用されていないことを示す場合、整数輝度サンプル M V 精度が使用されるかまたは 4 輝度サンプル M V 精度が使用されるかを示すために、別のフラグが信号通知される。

【 0 1 1 2 】

C U の第 1 の M V D 解像度フラグがゼロであるか、または C U に対して符号化されていない ( つまり、C U におけるすべての M V D がゼロである ) 場合、C U に対して  $1 / 4$  輝度サンプル M V 解像度が使用される。C U が整数輝度サンプル M V 精度または 4 輝度サンプル M V 精度を使用する場合、C U の A M V P 候補リストにおける M V P を対応する精度に丸める。

【 0 1 1 3 】

10

20

30

40

50

エンコーダにおいて、CUレベルのRDチェックは、どのMVD解像度をCUに用いるかを決定するために用いられる。すなわち、1つのMVD解像度ごとに3回、CUレベルのRDチェックを行う。エンコーダの速度を速めるために、JEMにおいては、以下の符号化方式が適用される。

【0114】

通常の1/4輝度サンプルMVD解像度を有するCUのRDチェック中、現在のCUの動き情報（整数輝度サンプル精度）が記憶される。整数輝度サンプルおよび4輝度サンプルのMVD解像度を有する同じCUのRDチェック中に、記憶された動き情報（丸められた後）は、更なる小範囲の動きベクトル改良の開始点として使用されるので、時間がかかる動き推定処理が3回重複しない。

10

【0115】

4輝度サンプルMVD解像度を有するCUのRDチェックを条件付きで呼び出す。CUの場合、整数輝度サンプルMVD解像度のRDコストが1/4輝度サンプルMVD解像度のそれよりもはるかに大きい場合、CUのための4輝度サンプルMVD解像度のRDチェックは省略される。

【0116】

2.3.3 パターンマッチング動きベクトル導出

【0117】

PMMVD (Pattern Matched Motion Vector Derivation) モードは、FRUC (Frame-Rate Up Conversion) 技術に基づく特殊マージモードである。このモードでは、ブロックの動き情報は信号通知されず、デコーダ側で導出される。

20

【0118】

そのマージフラグが真である場合、FRUCフラグは、CUに信号通知される。FRUCフラグが偽である場合、マージインデックスは信号通知され、通常のマージモードが使用される。FRUCフラグが真である場合、追加のFRUCモードフラグを信号通知して、どの方法（バイラテラルマッチングまたはテンプレートマッチング）を使用してブロックの動き情報を導出するかを示す。

【0119】

エンコーダ側では、CUのためにFRUCマージモードを使用するかどうかの決定は、通常のマージ候補に対して行われるのと同じように、RDコスト選択に基づく。つまり、RDコスト選択を使用して、1つのCUに対して2つのマッチングモード（バイラテラルマッチングおよびテンプレートマッチング）を両方チェックする。最小コストに導くものが、更に、他のCUモードと比較される。FRUCマッチングモードが最も効率的なものである場合、CUに対してFRUCフラグを真に設定し、関連するマッチングモードを使用する。

30

【0120】

FRUCマージモードにおける動き導出処理は、2つのステップを有する。まず、CUレベルの動き探索を実行し、次に、サブCUレベルの動き改良を実行する。CUレベルでは、バイラテラルマッチングまたはテンプレートマッチングに基づいて、CU全体のための初期の動きベクトルを導出する。まず、MV候補のリストを生成し、最小マッチングコストに導く候補を、さらなるCUレベル改善の開始点として選択する。そして、開始点付近のバイラテラルマッチングまたはテンプレートマッチングに基づく局所検索を行い、最小マッチングコストとなるMV結果をCU全体のMVとする。続いて、導出されたCU動きベクトルを開始点として、サブCUレベルでの動き情報をさらに改良する。

40

【0121】

例えば、 $W \times H$  CU動き情報導出のために、以下の導出処理を行う。第1のステージにおいて、 $W \times H$  CU全体のためのMVが導出される。第2のステージにおいて、CUは、 $M \times M$ のサブCUにさらに分割される。Mの値は、(16)のように計算されるが、Dは、予め定義された分割深さであり、JEMにおいてデフォルトで3に設定される。そし

50

て、各サブＣＵのＭＶを導出する。

【 0 1 2 2 】

【 数 1 】

$$M = \max\{4, \min\{\frac{M}{2^D}, \frac{N}{2^D}\}\} \quad (1)$$

【 0 1 2 3 】

図 2 1 に示すように、このバイラテラルマッチングは、2つの異なる参照ピクチャにおける現在のＣＵの動き軌跡に沿った2つのブロック間の最も近いマッチングを見出すことにより、現在のＣＵの動き情報を導出するために用いられる。連続した動き軌跡を仮定すると、2つの参照ブロックを指す動きベクトルＭＶ 0 およびＭＶ 1 は、現在のピクチャと2つの参照ピクチャとの間の時間的距離、例えばＴＤ 0 およびＴＤ 1 に比例する。特殊なケースとしては、現在のピクチャが時間的に2つの参照ピクチャの間にあり、現在のピクチャから2つの参照ピクチャまでの時間的な距離が同じである場合、バイラテラルマッチングはミラーに基づく双方向ＭＶとなる。

【 0 1 2 4 】

図 2 2 に示すように、現在のピクチャにおけるテンプレート（現在のＣＵの上側および／または左側の近傍のブロック）と、参照ピクチャにおけるブロック（テンプレートと同じサイズ）との間の最も近いマッチングを見出すことで、テンプレートマッチングを使用して、現在のＣＵの動き情報を導出する。前述のＦＲＵＣマージモード以外に、テンプレートマッチングは、ＡＭＶＰモードにも適用される。ＪＥＭにおいて、ＨＥＶＣと同様、ＡＭＶＰは2つの候補を有する。テンプレートマッチング法を用いることで、新しい候補を導出する。テンプレートマッチングによって新規に導出された候補が、第 1 の既存のＡＭＶＰ候補と異なる場合、ＡＭＶＰ候補リストの最初に挿入し、次に、リストサイズを2（第 2 の既存のＡＭＶＰ候補を取り除くことを意味する）に設定する。ＡＭＶＰモードに適用される場合、ＣＵレベル検索のみが適用される。

【 0 1 2 5 】

2 . 3 . 3 . 1    ＣＵレベルＭＶ候補セット

【 0 1 2 6 】

ＣＵレベルのＭＶ候補セットは、以下からなる。

（ i ）現在のＣＵがＡＭＶＰモードになっている場合の元のＡＭＶＰ候補

（ i i ）すべてのマージ候補、

（ i i i ）補間ＭＶフィールド内の複数のＭＶ。

（ i v ）上と左の近傍の動きベクトル

【 0 1 2 7 】

バイラテラルマッチングを使用する場合、マージ候補の各有効なＭＶを入力として使用して、バイラテラルマッチングを仮定してＭＶ対を生成する。例えば、マージ候補の1つの有効なＭＶは、参照リストＡにおいて（ＭＶ a , r e f a ）である。そして、その対をなすバイラテラルＭＶの参照ピクチャ r e f b が他の参照リストＢにおいて見出され、r e f a および r e f b は、時間的に現在のピクチャの異なる側にある。参照リストＢにおいてこのような r e f b が利用可能でない場合、r e f b を r e f a とは異なる参照として決定し、現在のピクチャとの時間的距離はリストＢにおける最小値である。r e f b を決定した後、現在のピクチャと r e f a , r e f b との時間距離に基づいてＭＶ a をスケーリングすることでＭＶ b を導出する。

【 0 1 2 8 】

補間されたＭＶフィールドからの4つのＭＶもＣＵレベル候補リストに追加する。より具体的には、現在のＣＵの位置（ 0 , 0 ）、（ W / 2 , 0 ）、（ 0 , H / 2 ）、（ W / 2 , H / 2 ）の補間ＭＶを加算する。

10

20

30

40

50

## 【 0 1 2 9 】

A M V P モードで F R U C を適用する場合、元の A M V P 候補を C U レベル M V 候補セットにも加える。

## 【 0 1 3 0 】

C U レベルにおいて、A M V P C U のための最大 1 5 個の M V およびマージ C U のための最大 1 3 個の M V を候補リストに加える。

## 【 0 1 3 1 】

2 . 3 . 3 . 2 サブ C U レベル M V 候補セット

## 【 0 1 3 2 】

サブ C U レベルの M V 候補セットは、以下からなる。

( i ) C U レベルの検索から決定された M V 、

( i i ) 上、左、左上、右上の近傍の M V 、

( i i i ) 参照ピクチャからの並置された M V のスケーリングされたバージョン、

( i v ) 最大 4 つの A T M V P 候補、

( v ) 最大 4 つの S T M V P 候補

## 【 0 1 3 3 】

参照ピクチャからのスケーリングされた M V は、以下のように導出される。両方のリストにおける参照ピクチャをすべてトラバースする。参照ピクチャにおけるサブ C U の配列位置にある M V は、開始 C U レベル M V の参照に対してスケーリングされる。

## 【 0 1 3 4 】

A T M V P および S T M V P の候補は、最初の 4 つの候補に限定される

## 【 0 1 3 5 】

サブ C U レベルにおいて、最大 1 7 個の M V が候補リストに追加される。

## 【 0 1 3 6 】

2 . 3 . 3 . 3 補間 M V フィールドの生成

## 【 0 1 3 7 】

フレームを符号化する前に、一方の M E に基づいてピクチャ全体に対して補間動きフィールドを生成する。そして、この動きフィールドを後に C U レベルまたはサブ C U レベルの M V 候補として使用してもよい。

## 【 0 1 3 8 】

まず、両方の参照リストにおける各参照ピクチャの動きフィールドは、 $4 \times 4$  ブロックレベルでトラバースされる。各  $4 \times 4$  ブロックにおいて、現在のピクチャ ( 図 2 3 に示す ) の  $4 \times 4$  ブロックを通過するブロックに関連する動きで、補間動きがまだ割り当てられていない場合、時間的距離 T D 0 および T D 1 に基づいて ( H E V C における T M V P の M V スケーリングと同様に ) 、参照ブロックの動きを現在のピクチャにスケーリングし、スケーリングされた動きを現在のフレームのブロックに割り当てる。  $4 \times 4$  ブロックにスケーリングされた M V が割り当てられていない場合、ブロックの動きは、補間された動きフィールドにおいて利用不可能であるとマークされる。

## 【 0 1 3 9 】

2 . 3 . 3 . 4 補間およびマッチングコスト

## 【 0 1 4 0 】

1 つの動きベクトルが 1 つの分数のサンプル位置を指す場合、動きの補償された補間が必要である。複雑性を低減するために、通常の 8 タップ H E V C 補間の代わりに、バイラテラルマッチングおよびテンプレートマッチングの両方に双線形補間を使用する。

## 【 0 1 4 1 】

マッチングコストの計算は、異なるステップでは少し異なる。C U レベルの候補セットから候補を選択する場合、マッチングコストは、バイラテラルマッチングまたはテンプレートマッチングの差分の絶対値の和 ( S A D ) である。開始 M V を決定した後、サブ C U レベル検索におけるバイラテラルマッチングのマッチングコスト C を以下のように算出する。

## 【 0 1 4 2 】

10

20

30

40

50

【数 2】

$$C = SAD + w \cdot (|MV_x - MV_x^s| + |MV_y - MV_y^s|) \quad (2)$$

【0143】

ここで、 $w$ は、経験的に4に設定された重み係数であり、 $MV$ および $MV^s$ は、それぞれ、現在の $MV$ および開始 $MV$ を示す。 $SAD$ は、依然として、サブCUレベル検索におけるテンプレートマッチングのマッチングコストとして使用される。

10

【0144】

FRUCモードにおいて、 $MV$ は、輝度サンプルのみを使用することによって導出される。導出された動きは、MCインター予測のために、輝度および彩度の両方に使用される。 $MV$ を決定した後、輝度用の8タップ補間フィルタおよび彩度用の4タップ補間フィルタを使用して、最終的なMCを行う。

【0145】

2.3.3.5  $MV$ の改良

【0146】

$MV$ 改良は、バイラテラルマッチングコストまたはテンプレートマッチングコストの基準を有するパターンに基づく $MV$ 検索である。JEMでは、2つの検索パターン、即ち、UCBDS (Unrestricted Center-Biased Diamond Search) およびCUレベルおよびサブCUレベルでの $MV$ 改良のための適応的横断検索をそれぞれサポートする。CUおよびサブCUレベルの $MV$ 改善の両方のために、 $MV$ は、 $1/4$ 輝度サンプル $MV$ の正確度で直接検索され、これに続いて $1/8$ 輝度サンプル $MV$ の改良が行われる。CUおよびサブCUステップのための $MV$ 改良の検索範囲は、8つの輝度サンプルに等しく設定される。

20

【0147】

2.3.3.6 テンプレートマッチングFRUCマージモードにおける予測方向の選択

【0148】

バイラテラルマッチングマージモードにおいては、2つの異なる参照ピクチャにおける現在のCUの動き軌跡に沿った2つのブロック間の最も近いマッチングに基づいて、CUの動き情報を導出するため、双方向予測が常に適用される。テンプレートマッチングマージモードについては、そのような制限はない。テンプレートマッチングマージモードにおいて、エンコーダは、 $list0$ からの単一予測、 $list1$ からの単一予測、またはCUのための双方向予測のうちから選択することができる。選択は、テンプレートマッチングコストに基づいて、以下のように行う。

30

$costBiFactor * \min(cost0, cost1)$  の場合

双方向予測を用いる。

それ以外の場合において、 $cost0 < cost1$  の場合

$list0$ からの単一予測を用いる。

40

そうでない場合、

$list1$ からの単一予測を用いる。

【0149】

ここで、 $cost0$ は $list0$ テンプレートマッチングの $SAD$ であり、 $cost1$ は $list1$ テンプレートマッチングの $SAD$ であり、 $costBi$ は双方向予測テンプレートマッチングの $SAD$ である。 $factor$ の値が1.25である場合、選択処理が双方向予測に偏っていることを意味する。このインター予測方向選択は、CUレベルのテンプレートマッチング処理にのみ適用される。

【0150】

2.3.4 デコーダ側動きベクトル改良

50

## 【 0 1 5 1 】

双方向予測演算において、1つのブロック領域を予測するために、`list 0`の動きベクトル(MV)および`list 1`のMVをそれぞれ使用して構成される2つの予測ブロックを組み合わせ、1つの予測信号を形成する。DMVR(Decoder-side Motion Vector Refinement)方法において、バイラテラルテンプレートマッチング処理によって、双方向予測の2つの動きベクトルをさらに改良する。追加の動き情報を送信することなく改良されたMVを得るために、デコードにおいてバイラテラルテンプレートマッチングを適用し、バイラテラルテンプレートと参照ピクチャにおける再構成サンプルとの間の歪みに基づく検索を行う。

## 【 0 1 5 2 】

DMVRにおいて、図23に示すように、`list 0`の最初のMV0と`list 1`のMV1とから、それぞれ2つの予測ブロックの重み付け結合(すなわち、平均)としてバイラテラルテンプレートを生成する。テンプレートマッチング操作は、生成されたテンプレートと参照ピクチャにおけるサンプル領域(最初の予測ブロックの付近)との間のコスト尺度を計算することからなる。2つの参照ピクチャの各々について、テンプレートコストが最小となるMVを、そのリストの更新されたMVと見なし、元のMVに置き換える。JEMにおいて、各リストに対して9つのMV候補を検索する。9つのMV候補は、元のMVと、水平または垂直方向のいずれかまたは両方向に元のMVに対してオフセットしている1つの輝度サンプルを有する8つの周囲のMVを含む。最後に、2つの新しいMV、即ち、図24に示すようなMV0'およびMV1'を使用して、最終的な双方向予測結果を生成する。差分の絶対値の和(SAD)をコスト尺度として使用する。

## 【 0 1 5 3 】

DMVRは、追加の構文要素を送信することなく、過去の参照ピクチャからの1つのMVと、将来の参照ピクチャからの1つのMVとの間の双方向予測のマージモードに適用される。JEMにおいて、CUに対してLIC、アフィン動き、FRUC、またはサブCUマージ候補が有効である場合、DMVRは適用されない。

## 【 0 1 5 4 】

## 2.3.5 バイラテラルマッチングの改良を伴うマージ/スキップモード

## 【 0 1 5 5 】

まず、利用可能な候補の数が最大候補サイズ19に達するまで、空間的に近傍のブロックおよび時間的に近傍のブロックの動きベクトルおよび参照インデックスを冗長性チェック付き候補リストに挿入することで、マージ候補リストを構築する。マージ/スキップモードのマージ候補リストは、予め規定された挿入順に基づいて、HEVC(結合候補およびゼロ候補)に用いられる空間的候補(図11)、時間的候補、アフィン候補、ATMVP(Advanced Temporal MVP)候補、STMVP(Spatial Temporal MVP)候補、および追加候補を挿入することで構築される。

## 【 0 1 5 6 】

- ブロック1~4の空間的候補

## 【 0 1 5 7 】

- ブロック1~4の外挿アフィン候補

## 【 0 1 5 8 】

- ATMVP

## 【 0 1 5 9 】

- STMVP

## 【 0 1 6 0 】

- 仮想アフィン候補

## 【 0 1 6 1 】

- 空間的候補(ブロック5)(利用可能な候補の数が6よりも少ない場合にのみ使用される)。

## 【 0 1 6 2 】

10

20

30

40

50



- 外挿アフィン候補 ( ブロック 5 )

【 0 1 6 3 】

- 時間的候補 ( H E V C のように導出 )

【 0 1 6 4 】

- 外挿アフィン候補に続く非隣接空間的候補 ( 図 2 5 に示すブロック 6 ~ 4 9 )。

【 0 1 6 5 】

- 結合候補

【 0 1 6 6 】

- ゼロ候補

【 0 1 6 7 】

10

なお、I C フラグは、S T M V P およびアフィンを除き、マージ候補から継承される。また、最初の 4 つの空間的候補について、双方向予測のものを単一予測のものの前に挿入する。

【 0 1 6 8 】

[ 8 ] において、現在のブロックに接続されていないブロックにアクセスすることができる。非隣接ブロックが非イントラモードにて符号化されている場合、関連する動き情報を追加のマージ候補として追加してもよい。

【 0 1 6 9 】

3 . 本明細書に開示される実施形態が解決しようとする課題の例

【 0 1 7 0 】

20

現在の H E V C 設計は、動き情報をよりよく符号化するために、現在のブロックの近傍のブロック ( 現在のブロックの隣 ) の相関をとることができる。しかしながら、近傍のブロックが、異なる動き軌跡を有する異なる対象に対応する可能性がある。この場合、その近傍のブロックからの予測は効率的ではない。

【 0 1 7 1 】

非隣接ブロックの動き情報からの予測は、全ての動き情報 ( 一般的には  $4 \times 4$  レベル ) をキャッシュに記憶するコストをかけることになり、付加的な符号化利得をもたらし、ハードウェア実装の複雑性を大幅に増大させる。

【 0 1 7 2 】

4 . 技術的解決策のいくつかの例

30

【 0 1 7 3 】

本開示の技術の実施形態は、既存の実装の欠点を克服し、それにより、より高いコード化効率を有する映像コード化を提供する。本願の説明において、L U T は、テーブル、リスト、配列またはインデックス付きエントリの他の配列であってもよい。

【 0 1 7 4 】

既存の実装形態の欠点を克服するために、様々な実施形態において、ブロックの動き情報を予測するために、少なくとも 1 つの動き候補が記憶された 1 つ以上のテーブル ( 例えばルックアップテーブル ) を使用する L U T に基づく動きベクトル予測技術を実装し、より高いコード化効率を有する映像コード化を提供することができる。ルックアップテーブルは、ブロックの動き情報を予測するために動き候補を含める際に使用できるテーブルの一例であり、他の実装形態も可能である。各 L U T は、それぞれが対応する動き情報に関連付けられた 1 つ以上の動き候補を含んでもよい。動き候補の動き情報は、予測方向、参照インデックス / ピクチャ、動きベクトル、L I C フラグ、アフィンフラグ、M V D ( M o t i o n V e c t o r D e r i v a t i o n ) 精度、および / または M V D 値の一部または全部を含んでもよい。動き情報は、動き情報がどこに由来しているかを示すために、ブロック位置情報をさらに含んでもよい。

40

【 0 1 7 5 】

開示される技術に基づいた L U T に基づく動きベクトル予測は、既存のおよび将来の映像符号化規格の両方を向上させることができ、様々な実施形態のために以下の例で説明される。L U T は、履歴データ ( 例えば、既に処理されたブロック ) に基づいて符号化 / 復号

50

化処理を行うことを可能にするため、LUTに基づく動きベクトル予測は、HMVP (History-based Motion Vector Prediction) 法と呼ぶこともできる。LUTに基づく動きベクトル予測方法において、以前にコード化されたブロックからの動き情報を有する1つまたは複数のテーブルは、符号化/復号化処理の間、維持される。LUTに記憶されたこれらの動き候補をHMVP候補と称する。1つのブロックの符号化/復号化の間、LUTにおける関連付けられた動き情報を動き候補リスト(例えば、マージ/AMVP候補リスト)に追加して、1つのブロックを符号化/復号化した後に、LUTを使用してもよい。更新されたLUTは、その後、後続のブロックを復号化するために用いられる。このように、LUTにおける動き候補の更新は、ブロックの符号化/復号化の順に基づく。以下の例は、一般的な概念を説明するための例であると考えられるべきである。

10

#### 【0176】

以下の実施例は、一般的な概念を説明するための例であると考えられるべきである。これらの例は狭い意味で解釈されるべきではない。さらに、これらの例は、任意の方法で組み合わせることができる。

#### 【0177】

一部の実施形態において、1つのブロックの動き情報を予測するために、少なくとも1つの動き候補が記憶された1つ以上のルックアップテーブルを用いてもよい。実施形態は、動き候補を用いて、ルックアップテーブルに記憶された動き情報のセットを示すことができる。従来のAMVPまたはマージモードの場合、実施形態では、動き情報を記憶するためにAMVPまたはマージ候補を使用してもよい。

20

#### 【0178】

以下の実施例は、一般的な概念を説明する。

#### 【0179】

ルックアップテーブルの例

#### 【0180】

各ルックアップテーブルは、各候補がその動き情報に関連付けられた1つ以上の動き候補を含んでもよい。

#### 【0181】

LUTの選択

30

#### 【0182】

例B1: 1つのブロックを符号化する場合、1つのルックアップテーブルからの動き候補の一部または全部を順にチェックすることができる。1つのブロックを符号化する間に1つの動き候補をチェックする際に、この動き候補を動き候補リスト(例えば、AMVP、マージ候補リスト)に加えてもよい。

#### 【0183】

ルックアップテーブルの使用法

#### 【0184】

チェック対象のルックアップテーブルにおける動き候補の総数は、予め規定されてもよい。

a. それは、符号化情報、ブロックサイズ、ブロック形状等にさらに依存してもよい。

40

1. 例えば、AMVPモードの場合、m個の動き候補のみをチェックし、マージモードの場合、n個の動き候補をチェックしてもよい(例えば、m=2、n=44)。

b. 一例において、チェック対象の動き候補の総数は、VPS (Video Parameter Set)、SPS (Sequence Parameter Set)、PPS (Picture Parameter Set)、スライスヘッダ、タイルヘッダ、CTU (Coding Tree Unit)、CTB (Coding Tree Block)、CU (Coding Unit) またはPU (Prediction Unit)、複数のCTU/CTB/CU/PUを含む領域で信号通知されてもよい。

#### 【0185】

例C1: 1つのルックアップテーブルに含まれる1つ以上の動き候補は、1つのブロック

50

によって直接継承されてもよい。

a . それらをマージモード符号化に使用してもよい。すなわち、マージ候補リスト導出処理において動き候補をチェックしてもよい。

b . これらは、アフィンマージモード符号化に使用してもよい。

i . アフィンフラグが1である場合、ルックアップテーブルにおける動き候補をアフィンマージ候補として加えることができる。

c . 以下の場合、ルックアップテーブルにおける動き候補のチェックを有効にしてもよい。

i . T M V P 候補を挿入した後、マージ候補リストが満杯になっていない。

i i . 空間的マージ候補導出のために特定の空間的に近傍のブロックをチェックした後、マージ候補リストが満杯になっていない。

i i i . すべての空間的マージ候補の後、マージ候補リストが満杯になっていない。

i v . 結合双方向予測マージ候補の後、マージ候補リストが満杯になっていない。

v . 他の符号化方式（例えば、H E V C デザイン、またはJ E M デザインのマージ導出処理）からマージ候補リストに入れられた空間的または時間的（例えば、隣接空間および非隣接空間、T M V P、S T M V P、A T M V P などを含む）マージ候補の数が、最大許容されるマージ候補から、所与の閾値を引いた数よりも少ない場合。

1 . 一例において、閾値は、1 または 0 に設定される。

2 . 代替的に、閾値は、S P S / P P S / シーケンス、ピクチャ、スライスヘッダ / タイルにおいて信号通知されてもよく、または予め規定されてもよい。

3 . 代替的に、閾値は、ブロックごとに適応的に変更されてもよい。例えば、それは、ブロックサイズ / ブロック形状 / スライスタイプのような符号化されたブロック情報に依存してもよく、および / または利用可能な空間的または時間的マージ候補の数に依存してもよい。

4 . 他の例において、既にマージ候補リストに追加されたある種のマージ候補の数が、最大許容マージ候補から、所与の閾値を引いた数未満である場合。「ある種のマージ候補」は、H E V C のような空間的候補であってもよいし、隣接しないマージ候補であってもよい。

v i . マージ候補リストに動き候補を追加する前に、プルーニングを適用してもよい。本特許明細書に開示されたこの例および他の例の様々な実装形態において、プルーニングは、a ) 動き情報と既存のエントリとを一意性のために比較すること、または、b ) 一意である場合、動き情報をリストに追加すること、または、c ) 一意でない場合、c 1 ) 動き情報を追加しない、または、c 2 ) 動き情報を追加し、一致した既存のエントリを削除することを含んでもよい。いくつかの実装形態において、テーブルから候補リストに動き候補を追加する際に、プルーニング工程は実行されない。

1 . 一例において、動き候補は、マージ候補リストの他の符号化方法から利用可能な空間的または時間的（例えば、隣接空間および非隣接空間、T M V P、S T M V P、A T M V P 等を含む）マージ候補の全部または一部にプルーニングされてもよい。

2 . 動き候補は、サブブロックに基づく動き候補、例えば、A T M V P、S T M V P にプルーニングされなくてもよい。

3 . 一例において、現在の動き候補は、マージ候補リストにおける利用可能な動き候補（現在の動き候補の前に挿入された）の全部または一部にプルーニングされてもよい。

4 . 動き候補に関連するプルーニング工程の数（例えば、動き候補をマージリストにおける他の候補と比較する必要がある回数）は、利用可能な空間的または時間的マージ候補の数に依存してもよい。例えば、新しい動き候補をチェックする際に、マージリストに利用可能な候補が M 個ある場合、新しい動き候補を最初の K 個（ $K \leq M$ ）の候補とのみ比較してもよい。プルーニング関数が偽を返す（例えば、最初の K 個の候補のいずれとも同一でない）場合、この新しい動き候補は、M 個の候補のすべてと異なると見なされ、マージ候補リストに追加され得る。一例において、K は、 $\min(K, 2)$  に設定される。

5 . 一例において、新しく付加された動き候補とマージ候補リストにおける最初の N 個の候補とを比較するだけである。例えば、 $N = 3$ 、4 または 5 である。N は、エンコーダが

10

20

30

40

50

らデコーダに信号通知されてもよい。

6．一例において、チェック対象の新しい動き候補は、マージ候補リストにおける最後のN個の候補と比較されるのみである。例えば、N = 3、4または5である。Nは、エンコーダからデコーダに信号通知されてもよい。

7．一例において、以前リストに追加された候補を選択し、テーブルからの新しい動き候補と比較する方法は、前回追加された候補がどこから導出されたかに依存してもよい。

a．一例において、ルックアップテーブルにおける動き候補を、所与の時間的および/または空間的に近傍のブロックから導出された候補と比較してもよい。

b．一例において、ルックアップテーブルにおける動き候補の異なるエントリを、以前追加された異なる候補と比較してもよい(すなわち、異なる位置から導出された)。

10

【0186】

例C2：ルックアップテーブルに含まれる動き候補は、ブロックの動き情報を符号化する予測モジュールとして用いられてもよい使用してもよい。

a．それらをAMVPモード符号化に使用してもよい。すなわち、AMVP候補リスト導出処理において動き候補をチェックしてもよい。

b．以下の場合、ルックアップテーブルにおける動き候補のチェックを有効にしてもよい。

i．TMVP候補を挿入した後、AMVP候補リストが満杯になっていない。

ii．AMVP候補リストは、空間的近傍から選択し、プルーニングした後で、TMVP候補を挿入する直前には、満杯になっていない。

iii．上側の隣接ブロックからのAMVP候補がスケーリング無しで存在しない場合、および/または、左側の隣接ブロックからのAMVP候補がスケーリング無しで存在しない場合。

20

iv．AMVP候補リストに動き候補を追加する前に、プルーニングを適用してもよい。

c．現在の参照ピクチャと同一の参照ピクチャを有する動き候補をチェックする。

i．代替的に、現在の参照ピクチャとは異なる参照ピクチャを有する動き候補も(スケーリングされたMVと併せて)チェックする。

ii．代替的に、まず、現在の参照ピクチャと同一の参照ピクチャを有するすべての動き候補をチェックし、次に、現在の参照ピクチャとは異なる参照ピクチャを有する動き候補をチェックする。

iii．代替的に、マージでも同様に動き候補をチェックする。

30

【0187】

例C3：以前符号化されたフレーム/スライス/タイルにおけるルックアップテーブルの動き候補を使用して、異なるフレーム/スライス/タイルにおけるブロックの動き情報を予測してもよい。

【0188】

ルックアップテーブルの更新

【0189】

例D1：動き情報を有するブロックを符号化した後(すなわち、IntraBCモード、インター符号化モード)に、1つ以上のルックアップテーブルを更新してもよい。

【0190】

40

例D2：1つのブロックが1つのピクチャ/スライス/タイルの境界に位置する場合、ルックアップテーブルの更新は常に許可されない。

【0191】

例D3：現在のLCU行を符号化するために、上記LCU行の動き情報を無効にしてもよい。

【0192】

例D4：新しい時間層インデックスを使用してスライス/タイルを符号化する開始時に、利用可能な動き候補の数を0にリセットすることができる。

【0193】

例D5：ルックアップテーブルは、同じ時間層インデックスを有する1つのスライス/タ

50

イル / LCU の行 / スライスで連続的に更新されてもよい。

【 0 1 9 4 】

例 D 6 : ルックアップテーブル更新プロセスは、異なる手順内で呼び出されてもよい。

【 0 1 9 5 】

追加の例示的な実施形態

【 0 1 9 6 】

以前コード化されたブロックの動き情報として H M V P 候補を規定する、H M V P ( H i s t o r y - b a s e d M V P ) 方法が提案される。符号化 / 復号化処理中、複数の H M V P 候補を有するテーブルが維持される。新しいスライスに遭遇した場合、テーブルは空になる。インター符号化されたブロックがあるときはいつでも、関連する動き情報を新しい H M V P 候補としてテーブルの最後のエントリに加える。全体の符号化フローを図 3 1 に示す。

10

【 0 1 9 7 】

一例において、テーブルサイズは L ( 例えば、L = 1 6 または 6、または 4 4 ) に設定され、これは、L 個までの H M V P 候補をテーブルに追加することができることを示す。

【 0 1 9 8 】

1 つの実施形態 ( 例 1 1 . g . i に対応する ) において、以前符号化されたブロックからの H M V P 候補が L 個よりも多く存在する場合、テーブルが常に最新の以前符号化された L 個の動き候補を含むように、先入れ先出し ( F I F O : F i r s t - I n - F i r s t - O u t ) 規則が適用される。図 3 2 は、F I F O 規則を適用して H M V P 候補を除去し、提案される方法で使用されるテーブルに新しいものを追加する例を示す。

20

【 0 1 9 9 】

別の実施形態 ( 発明 1 1 . g . i i i に対応する ) において、新しい動き候補を追加するときはいつでも ( 例えば、現在のブロックがインター符号化され、非アフィンモードであるなど )、まず、冗長性チェック処理を適用し、L U T に同じまたは類似した動き候補があるかどうかを識別する。

【 0 2 0 0 】

いくつかの例を以下に示す。

【 0 2 0 1 】

図 3 3 A は、新しい動き候補を追加する前に、L U T が満杯であった場合の例を示す。

30

【 0 2 0 2 】

図 3 3 B は、新しい動き候補を追加する前に、L U T が満杯でない場合の例を示す。

【 0 2 0 3 】

図 3 3 A および図 3 3 B は、ともに、冗長性除去に基づく L U T 更新方法 ( 1 つの冗長性動き候補を除去する ) の例を示す。

【 0 2 0 4 】

図 3 4 A および図 3 4 B は、冗長性除去に基づく L U T 更新方法 ( 複数の冗長性動き候補を除去する、図では 2 つの候補を示す ) の 2 つの場合の例示の実装形態を示す。

【 0 2 0 5 】

図 3 4 A は、新しい動き候補を追加する前に、L U T が満杯であった場合の例を示す。

40

【 0 2 0 6 】

図 3 4 B は、新しい動き候補を追加する前に、L U T が満杯でない場合の例を示す。

【 0 2 0 7 】

H M V P 候補は、マージ候補リスト構築処理において使用され得る。T M V P 候補の後に、テーブルにおける最後のエントリから最初のエントリ ( または最後の K 0 の H M V P、例えば、K 0 = 1 6 または 6 ) までのすべての H M V P 候補を挿入する。H M V P 候補に対してブルーニングを適用する。利用可能なマージ候補の総数が信号通知された最大許容マージ候補に達すると、マージ候補リスト構築処理を終了する。代替的に、加算された動き候補の総数が、所与の値に達すると、L U T からの動き候補のフェッチを終了する。

【 0 2 0 8 】

50

同様に、H M V P 候補は、A M V P 候補リスト構築処理において使用されてもよい。T M V P 候補の後に、テーブルにおける最後の K 1 個の H M V P 候補の動きベクトルを挿入する。A M V P 対象参照ピクチャと同じ参照ピクチャを有する H M V P 候補のみを用いて、A M V P 候補リストを構築する。H M V P 候補に対してブルーニングを適用する。一例において、K 1 は 4 に設定される。

#### 【 0 2 0 9 】

図 2 8 は、映像処理装置 2 8 0 0 のブロック図である。装置 2 8 0 0 は、本明細書に記載の方法の 1 つ以上を実装するために使用してもよい。装置 2 8 0 0 は、スマートフォン、タブレット、コンピュータ、I o T ( I n t e r n e t o f T h i n g s ) 受信機等により実装されてよい。装置 2 8 0 0 は、1 つ以上のプロセッサ 2 8 0 2 と、1 つ以上のメモリ 2 8 0 4 と、映像処理ハードウェア 2 8 0 6 と、を含んでよい。1 つまたは複数のプロセッサ 2 8 0 2 は、本明細書に記載される 1 つ以上の方法を実装するように構成されてもよい。1 つまたは複数のメモリ 2 8 0 4 は、本明細書で説明される方法および技術を実装するために使用されるデータおよびコードを記憶するために使用してもよい。映像処理ハードウェア 2 8 0 6 は、本明細書に記載される技術をハードウェア回路にて実装するために用いられてもよい。

10

#### 【 0 2 1 0 】

図 2 9 は、映像処理方法 2 9 0 0 の例のフローチャートである。方法 2 9 0 0 は、ステップ 2 9 0 2 でテーブルのセットを維持することを含み、各テーブルは、動き候補を含み、各動き候補は、対応する動き情報に関連付けられる。方法 2 9 0 0 は、ステップ 2 9 0 4 において、動き候補へのブルーニング工程を使って、1 つ以上のテーブルにおける動き候補に基づいて動き候補リストを更新することをさらに含む。方法 2 9 0 0 は、さらに、ステップ 2 9 0 6 において、構築された動き候補リストを使って、第 1 の映像ブロックと、第 1 の映像ブロックを含む映像のビットストリーム表現との間で変換を行うことと、を含む。

20

#### 【 0 2 1 1 】

図 3 0 は、映像処理方法 3 0 0 0 の例のフローチャートである。方法 3 0 0 0 は、ステップ 3 0 0 2 において、映像データの第 1 の映像ブロックを含む映像のビットストリーム表現を受信することを含む。方法 3 0 0 0 は、ステップ 3 0 0 4 において、1 つの候補リストを更新するために、1 つ以上のテーブルにおける動き候補に対してブルーニング工程を適用することをさらに含む、各テーブルは、動き候補を含み、各動き候補は、対応する動き情報に関連付けられる。方法 3 0 0 0 は、ステップ 3 0 0 6 において、構築された候補リストを使用してビットストリーム表現と第 1 の映像ブロックとの間で変換を行うことをさらに含む。

30

#### 【 0 2 1 2 】

上述した方法 / 技術の特徴および実施形態を、項目に基づくフォーマットを使用して以下に説明する。

#### 【 0 2 1 3 】

1 . テーブルのセットを維持することであって、各テーブルは、動き候補を含み、各動き候補は、対応する動き情報に関連付けられる、ことと、動き候補へのブルーニング工程を使用して 1 つ以上のテーブルで動き候補に基づいて候補リストを更新することと、第 1 の映像ブロックと、構成した候補リストを用いて、前記第 1 の映像ブロックを含む映像のビットストリーム表現との間で変換を行うことと、を有する、映像処理方法。

40

#### 【 0 2 1 4 】

2 . 第 1 の映像ブロックを含む映像のビットストリーム表現を受信することと、候補リストを構成するために 1 つまたは複数のテーブルにおける動き候補に対してブルーニング工程を適用することであって、各テーブルが動き候補を含み、各動き候補が対応する動き情報に関連付けられている、ことと、構成された候補リストを用いて、ビットストリーム表現と前記第 1 の映像ブロックとの間の変換を行うことと、を有する、映像処理方法。

#### 【 0 2 1 5 】

50

3. 少なくとも1つのテーブルは、前記第1の映像ブロックの前に復号化された、以前復号化された映像ブロックから導出した動き候補を含む、第1または2項に記載の方法。

【0216】

4. 前記ブルーニング工程は、テーブルからの動き候補を前記候補リストに追加する前に行われる、第1または2項に記載の方法。

【0217】

5. 前記候補リストは、マージ候補リストに対応する、第3項に記載の方法。

【0218】

6. 前記ブルーニング工程は、テーブルからの動き候補と、利用可能な空間的または時間的マージ候補の少なくとも一部とを比較することを含む、第1または2項に記載の方法。

10

【0219】

7. 前記ブルーニング工程は、テーブルからの動き候補とすべての利用可能な空間的マージ候補とを比較することを含む、第1または2項に記載の方法。

【0220】

8. 前記ブルーニング工程は、動き候補をテーブルからサブブロックに基づく動き候補に変更することなく行われる、第1、2または4項に記載の方法。

【0221】

9. 前記ブルーニング工程は、動き候補を加える前に、テーブルからの動き候補をマージ候補リストにおける利用可能な動き候補の少なくとも1部と比較することを含む、第1または2項に記載の方法。

20

【0222】

10. 前記ブルーニング工程は、複数の工程を含み、その数は、複数の空間的または時間的マージ候補の関数である、第1または2項に記載の方法。

【0223】

11. 前記複数の工程は、マージ候補リストにおいてM個の候補が利用可能である場合に、ブルーニングが、K個のマージ候補にのみ適用され、 $K \leq M$ であり、KおよびMが整数である、第9項に記載の方法。

【0224】

12. Kを $\min(K, 2)$ に設定する、第10項に記載の方法。

【0225】

30

13. ブルーニング工程は、動き候補を候補リストに加える前に、テーブルからの動き候補と候補リストにおける第1のN個の候補とを比較することを含み、Nが整数である、第1項に記載の方法。

【0226】

14. 前記ブルーニング工程は、動き候補を前記候補リストに加える前に、チェック対象のテーブルからの動き候補と、候補リストにおける最後のN個の候補とを比較することを含み、Nが整数である、第1または2項に記載の方法。

【0227】

15. Nの値を信号通知することをさらに含む、第12または13項に記載の方法。

【0228】

40

16. ブルーニング工程は、候補がどのように導出されたかに基づいて、テーブルからの動き候補と、候補リストにおけるある候補とを比較することを含む、第1項に記載の方法。

【0229】

17. ブルーニング工程は、テーブルからの動き候補と、所与の時間的または空間的に隣接ブロックから導出された候補とを比較することを含む、第15項に記載の方法。

【0230】

18. テーブルにおける異なる動き候補のエントリを、異なる位置から導出された、以前に追加された異なる候補と比較する、第15項に記載の方法。

【0231】

19. テーブルから前記候補リストに動き候補を追加する際に、ブルーニング工程を呼び

50

出さないことをさらに含む、第 1 項に記載の方法。

【 0 2 3 2 】

2 0 . 前記変換を行うことは、前記映像ブロックから前記ビットストリーム表現を生成することを含む、第 1 ~ 1 9 項のいずれか 1 項に記載の方法。

【 0 2 3 3 】

2 1 . 前記変換を行うことは、前記ビットストリーム表現から前記映像ブロックを生成することを含む、第 1 ~ 1 9 項のいずれか 1 項に記載の方法。

【 0 2 3 4 】

2 2 . 動き候補は、予測方向、参照ピクチャインデックス、動きベクトル値、強度補償フラグ、アフィンフラグ、動きベクトル差精度、または動きベクトル差分値のうちの少なくとも 1 つを含む動き情報に関連付けられる、第 1 ~ 2 1 項のいずれか 1 項に記載の方法。

【 0 2 3 5 】

2 3 . 前記動き候補は、イントラモード符号化のためのイントラ予測モードの動き候補に対応する、第 1 ~ 2 2 項のいずれか 1 項に記載の方法。

【 0 2 3 6 】

2 4 . 前記動き候補は、I C パラメータ符号化のための照明補償パラメータを含む動き候補に対応する、第 1 ~ 2 2 項のいずれか 1 項に記載の方法。

【 0 2 3 7 】

2 5 . 前記変換に基づいて、1 つ以上のテーブルを更新することをさらに含む、第 1 項 ~ 第 2 4 項のいずれか 1 項に記載の方法。

【 0 2 3 8 】

2 6 . 1 つ以上のテーブルを更新することは、前記変換を行った後、前記第 1 の映像ブロックの動き情報に基づいて 1 つ以上のテーブルを更新することを含む、第 2 5 項に記載の方法。

【 0 2 3 9 】

2 7 . 前記更新されたテーブルに基づいて、前記映像の後続の映像ブロックと前記映像のビットストリーム表現との間で変換を行うことをさらに含む、第 2 6 項に記載の方法。

【 0 2 4 0 】

2 8 . プロセッサと、命令が記憶された非一時的メモリとを備える装置であって、命令は、前記プロセッサによって実装された際に、前記プロセッサに、第 1 項から第 2 7 項のいずれか 1 項に記載の方法を実施させる装置。

【 0 2 4 1 】

2 9 . 非一時的なコンピュータ可読媒体に記憶されたコンピュータプログラム製品であって、第 1 ~ 2 7 項のいずれか 1 項に記載の方法を実行するためのプログラムコードを含む、コンピュータプログラム製品。

【 0 2 4 2 】

以上、説明の目的で本開示の技術の特定の実施形態を説明したが、本発明の範囲から逸脱することなく様々な修正が可能であることは、理解されるであろう。従って、本開示の技術は、添付の特許請求の範囲による場合を除き、限定されない。

【 0 2 4 3 】

本明細書に記載された開示された、およびその他の実施形態、モジュール、および機能操作の実装形態は、本明細書に開示された構造およびその構造的等価物を含め、デジタル電子回路、またはコンピュータソフトウェア、ファームウェア、若しくはハードウェアで実施されてもよく、またはそれらの 1 つ以上の組み合わせで実施してもよい。開示された、およびその他の実施形態は、1 つ以上のコンピュータプログラム製品、すなわち、データ処理装置によって実装されるため、またはデータ処理装置の操作を制御するために、コンピュータ可読媒体上に符号化されたコンピュータプログラム命令の 1 つ以上のモジュールとして実施することができる。このコンピュータ可読媒体は、機械可読記憶装置、機械可読記憶基板、記憶装置、機械可読伝播信号をもたらす物質の組成物、またはこれらの 1 つ以上の組み合わせであってもよい。「データ処理装置」という用語は、例えば、プログラ

10

20

30

40

50



マブルプロセッサ、コンピュータ、または複数のプロセッサ、若しくはコンピュータを含め、データを処理するためのすべての装置、デバイス、および機械を含む。この装置は、ハードウェアの他に、当該コンピュータプログラムの実行環境を作るコード、例えば、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、またはこれらの1つ以上の組み合わせを構成するコードを含んでもよい。伝播信号は、人工的に生成した信号、例えば、機械で生成した電気、光、または電磁信号であり、適切な受信装置に送信するための情報を符号化するために生成される。

【0244】

コンピュータプログラム（プログラム、ソフトウェア、ソフトウェアアプリケーション、スクリプト、またはコードとも呼ばれる）は、コンパイルされた言語または解釈された言語を含む任意の形式のプログラミング言語で記述することができ、それは、スタンドアロンプログラムとして、またはコンピューティング環境で使用するのに適したモジュール、コンポーネント、サブルーチン、または他のユニットとして含む任意の形式で展開することができる。コンピュータプログラムは、必ずしもファイルシステムにおけるファイルに対応するとは限らない。プログラムは、他のプログラムまたはデータを保持するファイルの一部（例えば、マークアップ言語文書に格納された1つ以上のスクリプト）に記録されていてもよいし、当該プログラム専用の単一のファイルに記憶されていてもよいし、複数の調整ファイル（例えば、1つ以上のモジュール、サブプログラム、またはコードの一部を格納するファイル）に記憶されていてもよい。1つのコンピュータプログラムを、1つのサイトに位置する1つのコンピュータ、または複数のサイトに分散され通信ネットワークによって相互接続される複数のコンピュータで実行させるように展開可能である。

【0245】

本明細書に記載されたプロセスおよびロジックフローは、入力データ上で動作し、出力を生成することによって機能を実行するための1つ以上のコンピュータプログラムを実行する1つ以上のプログラマブルプロセッサによって行うことができる。プロセスおよびロジックフローはまた、特別目的のロジック回路、例えば、FPGA（Field Programmable Gate Array）またはASIC（Application Specific Integrated Circuit）によって実行することができ、装置はまた、特別目的のロジック回路として実装することができる。

【0246】

コンピュータプログラムの実行に適したプロセッサは、例えば、汎用および専用マイクロプロセッサの両方、並びに任意の種類のデジタルコンピュータの任意の1つ以上のプロセッサを含む。一般的に、プロセッサは、読み出し専用メモリまたはランダムアクセスメモリまたはその両方から命令およびデータを受信する。コンピュータの本質的な要素は、命令を実行するためのプロセッサと、命令およびデータを記憶するための1つ以上の記憶装置とである。一般的に、コンピュータは、データを記憶するための1つ以上の大容量記憶デバイス、例えば、磁気、光磁気ディスク、または光ディスクを含んでもよく、またはこれらの大容量記憶デバイスからデータを受信するか、またはこれらにデータを転送するように動作可能に結合されてもよい。しかしながら、コンピュータは、このようなデバイスを有する必要はない。コンピュータプログラム命令およびデータを記憶するのに適したコンピュータ可読媒体は、あらゆる形式の不揮発性メモリ、媒体、および記憶装置を含み、例えば、EPROM、EEPROM、フラッシュ記憶装置、磁気ディスク、例えば内部ハードディスクまたはリムーバブルディスク、光磁気ディスク、およびCD-ROMおよびDVD-ROMディスク等の半導体記憶装置を含む。プロセッサおよびメモリは、専用ロジック回路によって補完されてもよく、または専用ロジック回路に組み込まれてもよい。

【0247】

この特許明細書は多くの詳細を含むが、これらは、任意の発明の範囲または特許請求の範囲を限定するものと解釈されるべきではなく、むしろ、特定の発明の特定の実施形態に特有であり得る特徴の説明と解釈されるべきである。本特許明細書において別の実施形態の文脈で説明されている特定の特徵は、1つの例において組み合わせて実装してもよい。逆

10

20

30

40

50

に、単一の例の文脈で説明された様々な特徴は、複数の実施形態において別個にまたは任意の適切なサブコンビネーションで実装してもよい。さらに、特徴は、特定の組み合わせで作用するものとして上記に記載され、最初にそのように主張されていてもよいが、主張された組み合わせからの1つ以上の特徴は、場合によっては、組み合わせから抜粋されることができ、主張された組み合わせは、サブ組み合わせまたはサブ組み合わせのバリエーションに向けられてもよい。

【0248】

同様に、動作は図面において特定の順番で示されているが、これは、所望の結果を達成するために、このような動作が示された特定の順番でまたは連続した順番で実行されること、または示された全ての操作が実行されることを必要とするものと理解されるべきではない。また、本特許明細書に記載されている例における様々なシステムの構成要素の分離は、全ての実施形態においてこのような分離を必要とするものと理解されるべきではない。

10

【0249】

いくつかの実装形態および例のみが記載されており、この特許明細書に記載され図示されている内容に基づいて、他の実施形態、拡張および変形が可能である。

20

30

40

50

【図面】

【 図 1 】

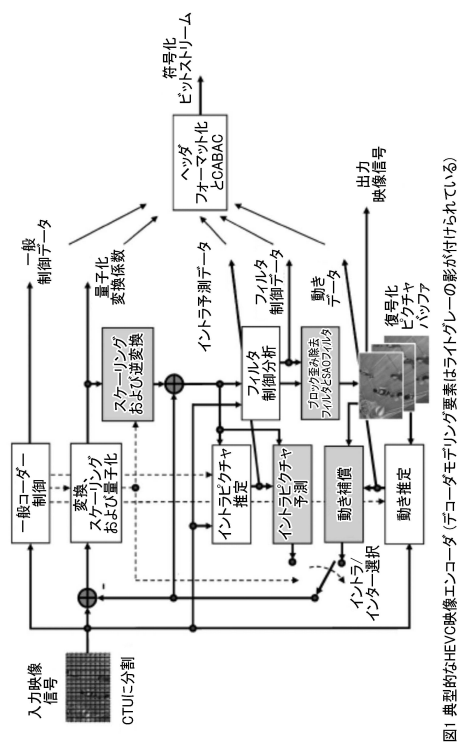
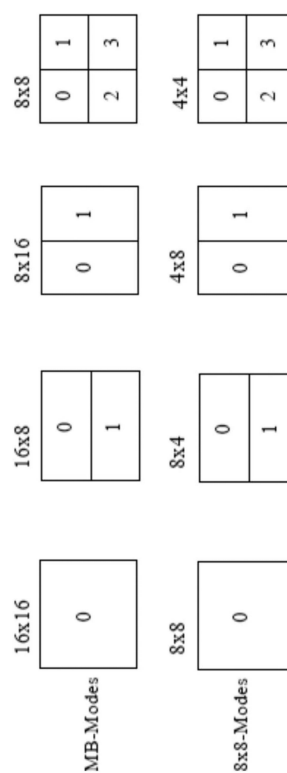


図1 典型的なHEVC映像エンコーダ（デコーダ）モジュール要素はライトグレーの影が付けられている

## 【圖 2】



【 図 3 】

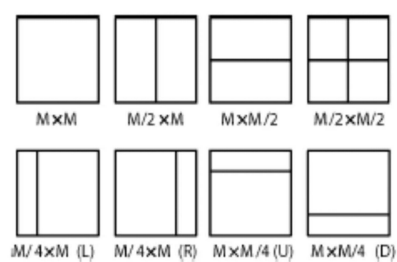
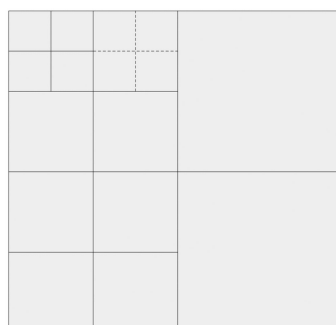
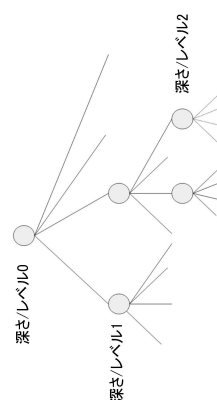
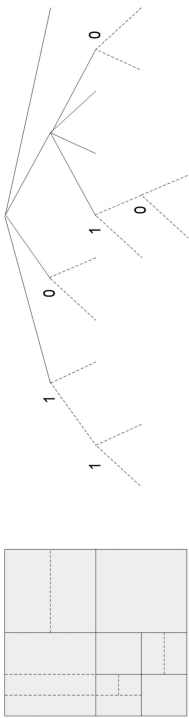


図3 特定のサイズ制限に従って、CBをPBIに分割するためのモード。  
イントラピクチャ予測CBでは、 $M \times M$ と $M/2 \times M/2$ だけがサポートされる

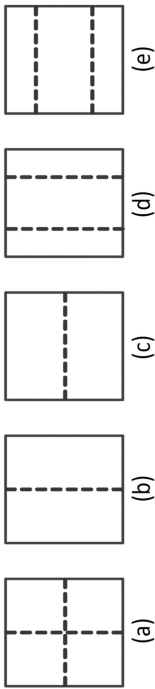
【圖 4】



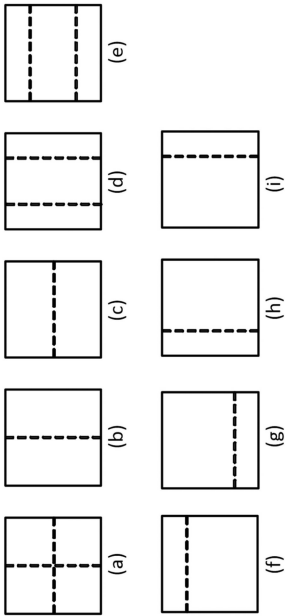
【図 5】



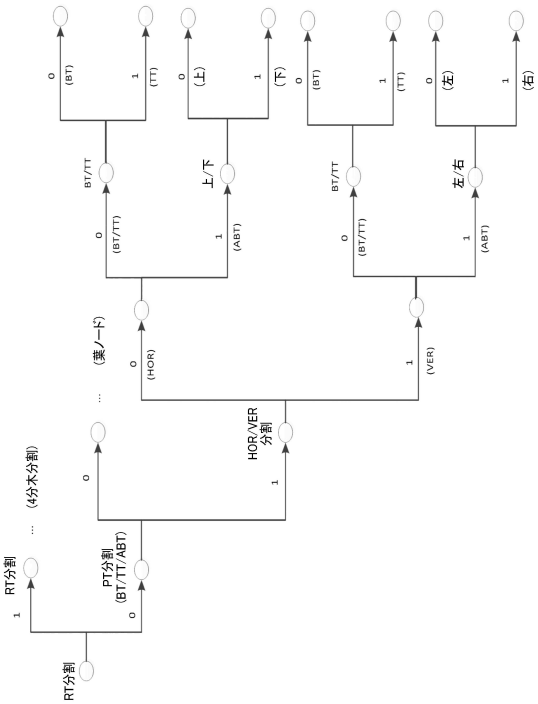
【図 6】



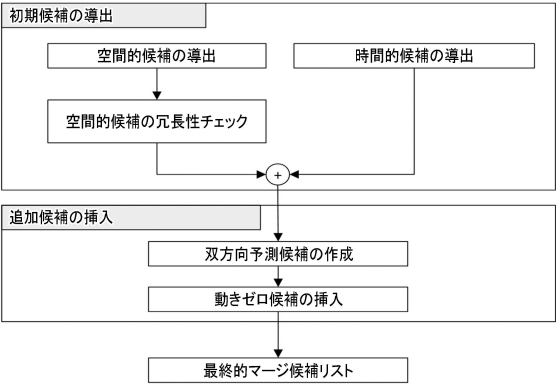
【図 7】



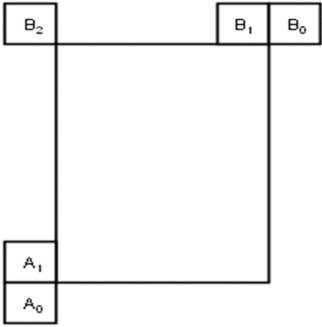
【図 8】



【図 9】

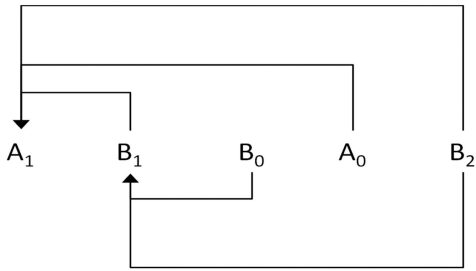


【図 10】

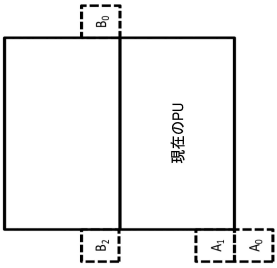


10

【図 11】

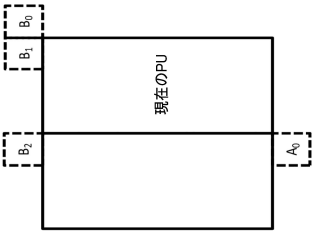


【図 12】



(b)  $2N \times N$  の第2のPU

20



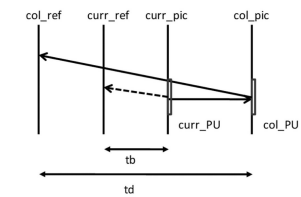
(a)  $N \times 2N$  の第2のPU

30

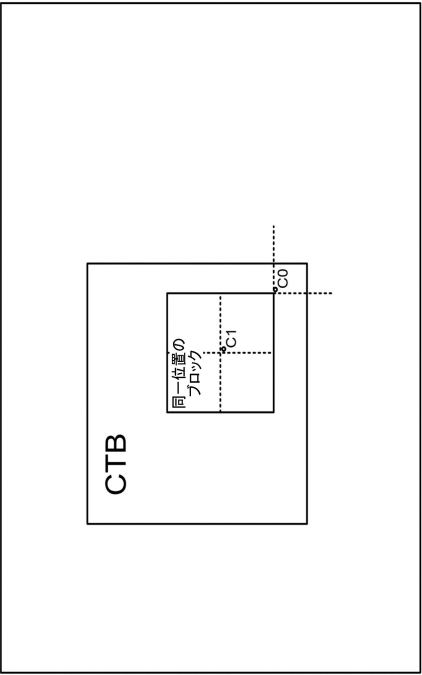
40

50

【図 1 3】



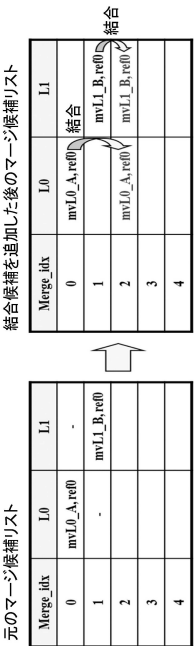
【図 1 4】



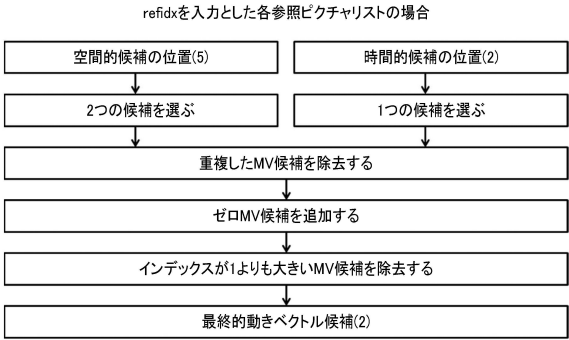
10

20

【図 1 5】



【図 1 6】

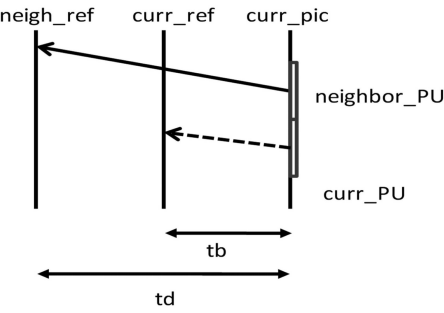


30

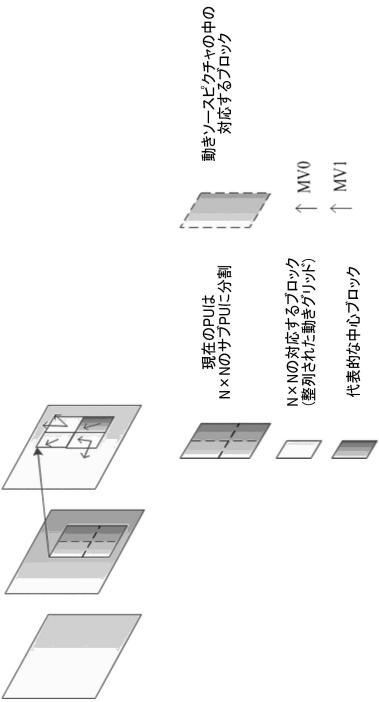
40

50

【図 17】



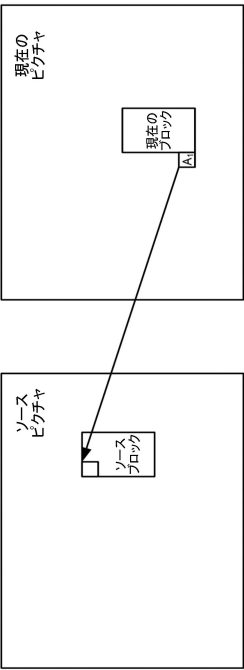
【図 18】



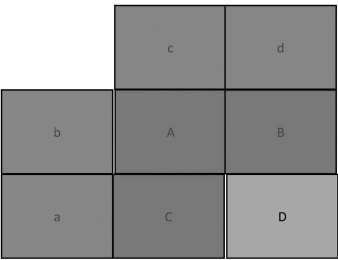
10

20

【図 19】



【図 20】

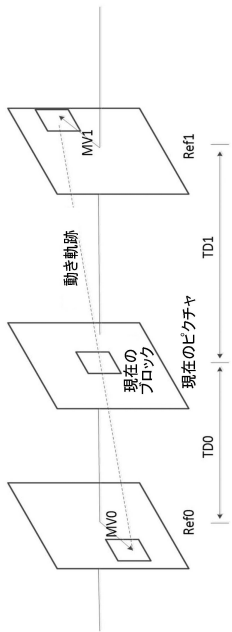


30

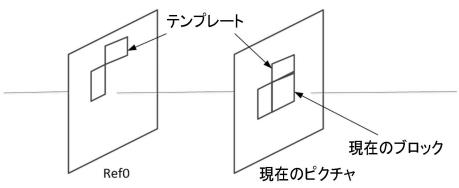
40

50

【図 2 1】



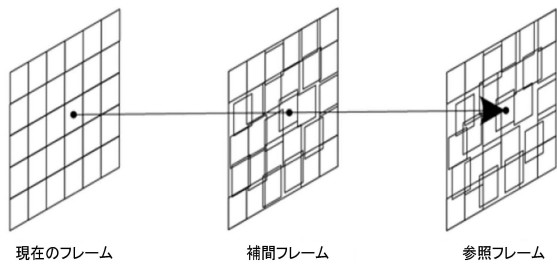
【図 2 2】



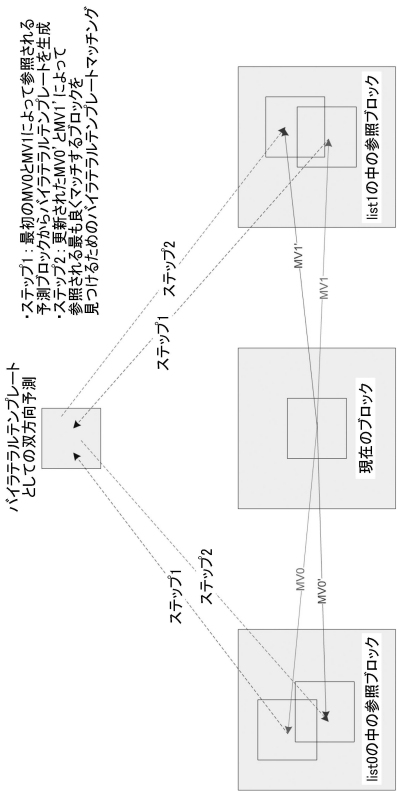
10

20

【図 2 3】



【図 2 4】



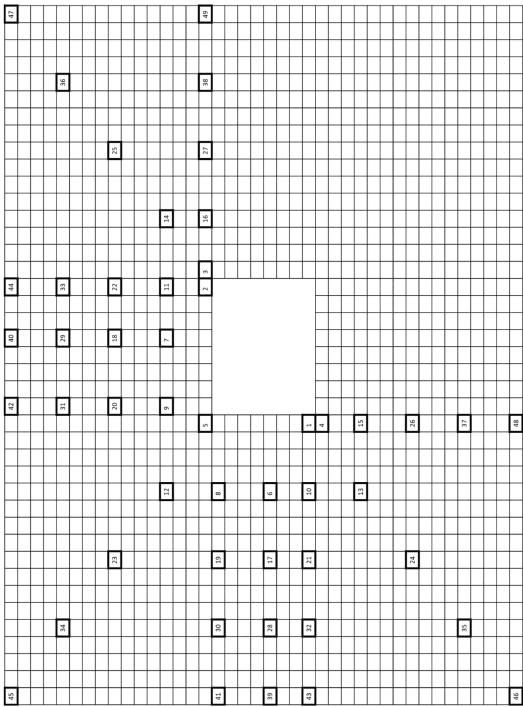
30

40

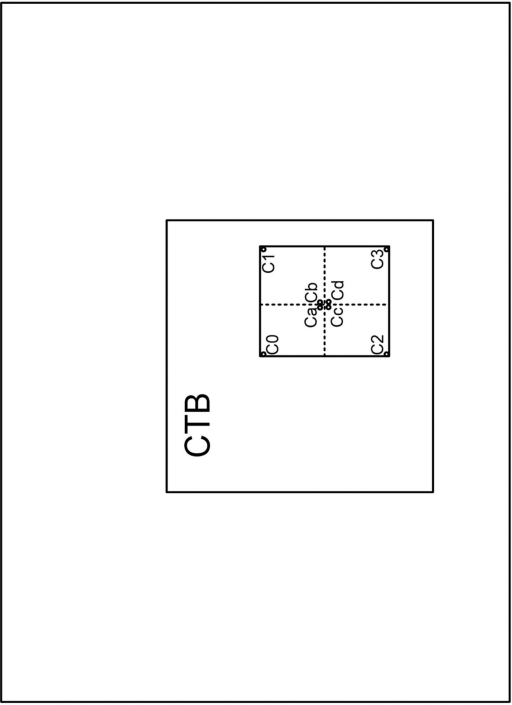
50



【図 2 5】



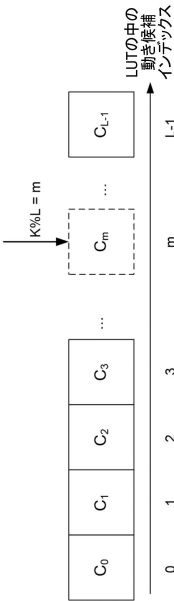
【図 2 6】



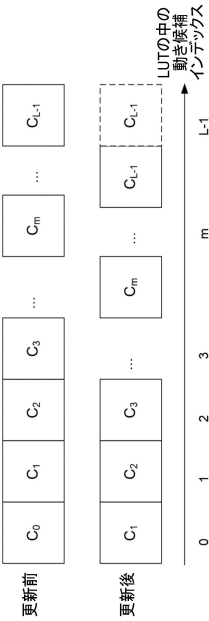
10

20

【図 2 7 A】



【図 2 7 B】

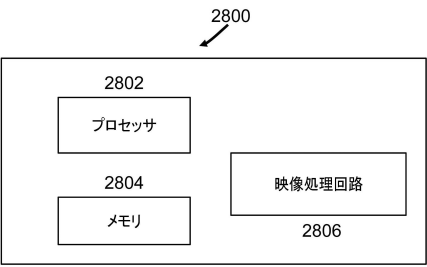


30

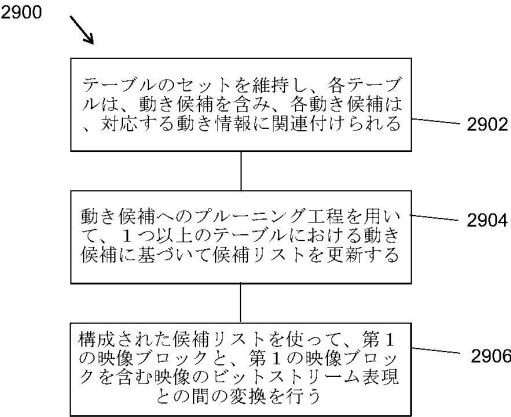
40

50

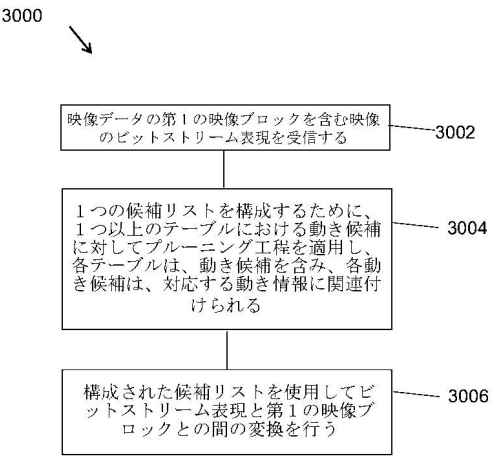
【図 28】



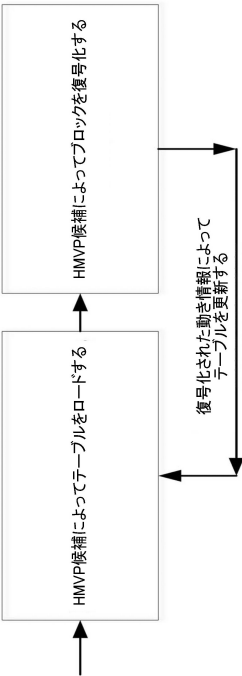
【図 29】



【図 30】



【図 31】



10

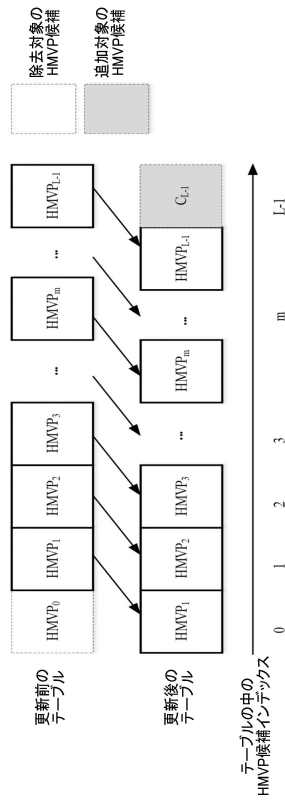
20

30

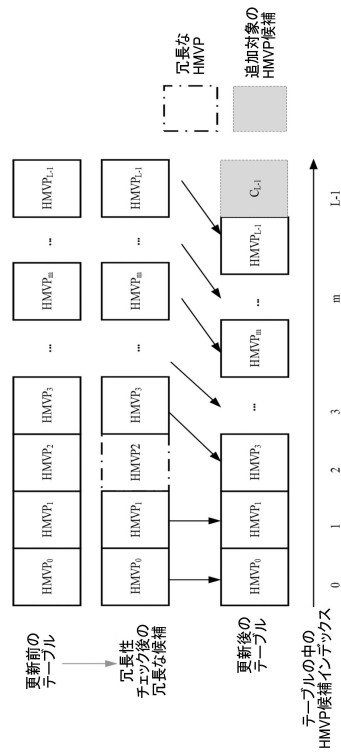
40

50

【図 3 2】



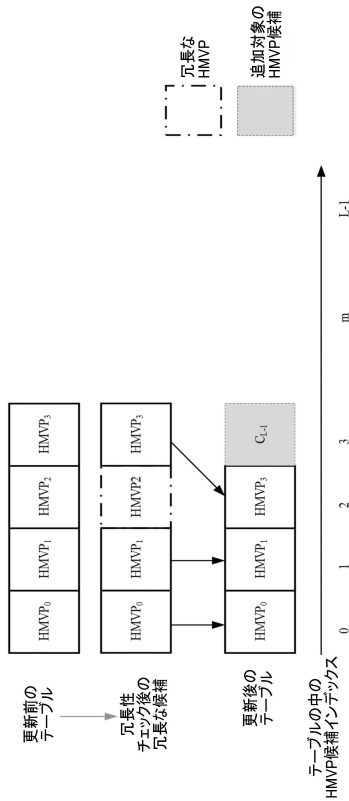
【図 3 3 A】



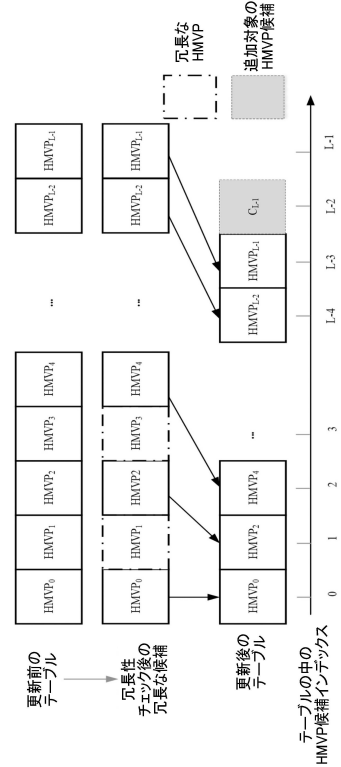
10

20

【図 3 3 B】



【図 3 4 A】

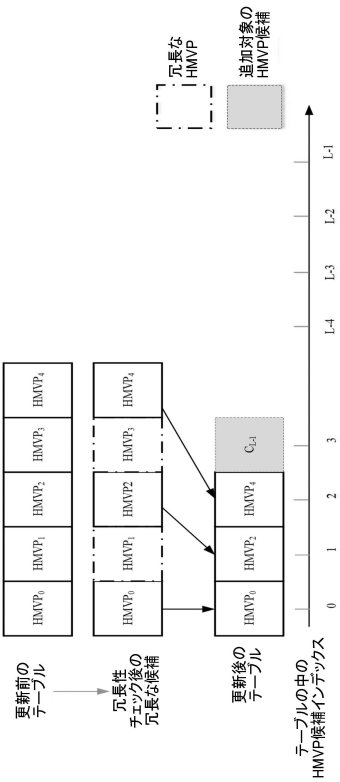


30

40

50

【図 3 4 B】



10

20

30

40

50

## フロントページの続き

(33)優先権主張国・地域又は機関

中国(CN)

(31)優先権主張番号 PCT/CN2018/117627

(32)優先日 平成30年11月27日(2018.11.27)

(33)優先権主張国・地域又は機関

中国(CN)

(31)優先権主張番号 PCT/CN2019/071214

(32)優先日 平成31年1月10日(2019.1.10)

(33)優先権主張国・地域又は機関

中国(CN)

## 早期審査対象出願

BYTEDANCE INC.

アメリカ合衆国 カリフォルニア州 90066, ロサンゼルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロア, スイート ナンバー・137  
12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137 Los Angeles, California 90066 United States of America

(74)代理人 110002000

特許業務法人栄光特許事務所

(72)発明者 ジャン リー

アメリカ合衆国 90066 カリフォルニア州, ロサンゼルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロア, スイート ナンバー137

(72)発明者 ジャン カイ

アメリカ合衆国 90066 カリフォルニア州, ロサンゼルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロア, スイート ナンバー137

(72)発明者 リウ ホンビン

中華人民共和国 100080 ベイジン, ハイディアン ディストリクト, デーチュン ロード, チャイナ サテライト コミュニケーションズ タワー, ナンバー63, ジンリトウティアオ ポストオフィス

(72)発明者 ワン ユエ

中華人民共和国 100080 ベイジン, ハイディアン ディストリクト, デーチュン ロード, チャイナ サテライト コミュニケーションズ タワー, ナンバー63, ジンリトウティアオ ポストオフィス

審査官 坂東 大五郎

(56)参考文献 特表2022-507682(JP, A)

特開2021-052373(JP, A)

特表2021-513795(JP, A)

特表2021-510265(JP, A)

Li Zhang et al., CE4-related: History-based Motion Vector Prediction, Joint Video Experts Team (JVET), 2018年07月18日, [JVET-K0104-v5] (version 5)

(58)調査した分野 (Int.Cl., DB名)

H04N 19/00 - 19/98