

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7295231号
(P7295231)

(45)発行日 令和5年6月20日(2023.6.20)

(24)登録日 令和5年6月12日(2023.6.12)

(51)国際特許分類	F I
H 0 4 N 19/513 (2014.01)	H 0 4 N 19/513
H 0 4 N 19/58 (2014.01)	H 0 4 N 19/58
H 0 4 N 19/70 (2014.01)	H 0 4 N 19/70
H 0 4 N 19/52 (2014.01)	H 0 4 N 19/52

請求項の数 22 (全51頁)

(21)出願番号	特願2021-523115(P2021-523115)	(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2021-530940(P2021-530940 A)		Beijing Bytedance Network Technology Co., Ltd.
(43)公表日	令和3年11月11日(2021.11.11)		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/IB2019/055595		Room B-0035, 2/F, No.30 Building, No.30, Shixing Road, Shijingshan District Beijing 100041 China
(87)国際公開番号	WO2020/003284		
(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】 LUTとAMVPの間の相互作用

(57)【特許請求の範囲】

【請求項1】

1または複数のテーブルを維持することであって、各テーブルは、動き候補のセットを含み、各動き候補は、対応する動き情報に関連付けられ、前の映像ブロックから導出され、前記テーブルにおける前記動き候補の配列は、前記テーブルへの前記動き候補の追加の順に基づく、ことと、

第1の映像ブロックに対する動き候補を導出するための動き候補リストの導出処理を実行することであって、前記動き候補リストの導出処理は、順に、前記1または複数のテーブルのうちのテーブルにおける1または複数の動き候補を選択的にチェックすることを含む、ことと、

前記動き候補リストに基づいて、動きベクトル予測因子として用いられる動き情報を導出することと、

前記動き情報、および、動きベクトルと前記動きベクトル予測因子との間のMVD (Motion Vector Difference)に基づいて、前記第1の映像ブロックと前記映像ブロックを含む映像のビットストリームの間の変換を実行することであって、前記MVDは、前記ビットストリームにて示される、ことと、
を有し、

前記動き候補リストを更新するために用いられる前記チェックされた動き候補のうちの少なくとも1つの動き候補は、前記第1の映像ブロックの参照ピクチャと同じ参照ピクチャを有する映像処理方法。

【請求項 2】

前記変換を行うことは、前記第 1 の映像ブロックを前記ビットストリームに符号化することを含む、請求項 1 に記載の方法。

【請求項 3】

前記変換を行うことは、前記第 1 の映像ブロックを前記ビットストリームから復号化することを含む、請求項 1 に記載の方法。

【請求項 4】

前記テーブルにおける前記 1 または複数の動き候補をチェックすることは、前記テーブルに少なくとも 1 つの動き候補がある場合に有効化され、または、

前記テーブルにおける前記 1 または複数の動き候補をチェックすることは、TMVP (Temporal Motion Vector Prediction) の動き候補を導出するために、前記第 1 の映像ブロックを有するピクチャとは異なるピクチャの時間的なブロックをチェックした後に、前記動き候補リストが満杯でない場合に有効化され、または、

前記テーブルにおける前記 1 または複数の動き候補をチェックすることは、ある動き候補を挿入した後に前記動き候補リストが満杯でない場合に有効化され、または、

前記テーブルにおける前記 1 または複数の動き候補をチェックすることは、i) スケーリングを行うことなく上の近傍のブロックからの動き候補が無い場合、および/または、ii) スケーリングを行うことなく左の近傍のブロックからの動き候補が無い場合に、有効化され、または、

前記テーブルにおける前記 1 または複数の動き候補をチェックすることは、空間的または時間的なブロックから導出された動き候補、AMVP 動き候補、SMVD 動き候補、SMV 動き候補、または、アフィンインター動き候補の少なくとも 1 つを含む他の動き候補をチェックする前に有効化される、請求項 1 に記載の方法。

【請求項 5】

前記動き候補リストは、AMVP (Advanced Motion Vector Prediction) の候補リストである、請求項 1 に記載の方法。

【請求項 6】

前記 1 または複数のチェックされた動き候補を用いて前記動き候補リストを更新するか否かは、前記チェックの結果に基づく、請求項 1 に記載の方法。

【請求項 7】

前記動き候補リストを更新することは、前記チェックされた動き候補に関連付けられた動きベクトルを前記動き候補リストに追加することを有する、請求項 1 に記載の方法。

【請求項 8】

前記変換を行うことは、いくつかの前記動きベクトルの差分を用いる SMVD (Symmetric Motion Vector Difference) 処理、または、いくつかの動きベクトルを用いる SMV (Symmetric Motion Vector) 処理の少なくとも 1 つを実行することを含む、請求項 1 に記載の方法。

【請求項 9】

前記テーブルにおける前記 1 または複数の動き候補をチェックすることは、第 1 の参照ピクチャリストの参照ピクチャをチェックし、その後、第 2 の参照ピクチャリストの参照リストをチェックする、請求項 1 に記載の方法。

【請求項 10】

前記第 1 の参照ピクチャリストは、現在の対象の参照ピクチャリストである、請求項 9 に記載の方法。

【請求項 11】

チェック対象の前記動き候補は、双方向予測動き候補である、請求項 9 に記載の方法。

【請求項 12】

現在の参照ピクチャと前記テーブルにおいて同一の参照ピクチャを有する動き候補がチェックされる、請求項 1 に記載の方法。

10

20

30

40

50

【請求項 13】

前記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が更にチェックされ、前記同一の参照ピクチャを有する前記動き候補をチェックすることは、前記異なる参照ピクチャを有する前記動き候補をチェックすることの前に行われる、請求項 12 に記載の方法。

【請求項 14】

前記動き候補リストの構成処理は、前記テーブルにおける少なくとも 1 つのチェックされた動き候補に基づいて前記動き候補を更新する前のプルーニング工程を有する、請求項 1 に記載の方法。

【請求項 15】

前記プルーニング工程は、チェック対象の動き候補と、前記動き候補リストにおける利用可能な動き候補の一部または全部とを比較することを含む、請求項 14 に記載の方法。

【請求項 16】

前記プルーニング工程は、複数の工程を含み、その数は、複数の空間的または時間的な動き候補の関数である、請求項 14 に記載の方法。

【請求項 17】

前記テーブルにおける動き候補は、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、動きベクトル差分精度、イントラモード情報、IC (illumination compensation) パラメータ、フィルタリング処理にて用いられるフィルタパラメータ、または、動きベクトル差分値、の少なくとも 1 つの含む動き情報に関連付けられる、請求項 1 に記載の方法。

【請求項 18】

前記第 1 の映像ブロックの前記動き情報に基づく動き候補を用いて、1 または複数のテーブルのうちテーブルを更新することを更に有し、

前記第 1 の映像ブロックの前記動き情報に対応する前記テーブルにおける前記動き候補のインデックスは、前記テーブルにおける他の動き候補よりも大きなインデックスを有する、請求項 1 に記載の方法。

【請求項 19】

前記テーブルにおける前記 1 または複数の動き候補は、前記 1 または複数の動き候補の 1 または複数のインデックスの順にてチェックされる、請求項 1 に記載の方法。

【請求項 20】

プロセッサと、命令を有する非一時的メモリを有する、映像データを処理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、

1 または複数のテーブルを維持することであって、各テーブルは、動き候補のセットを含み、各動き候補は、対応する動き情報に関連付けられ、前の映像ブロックから導出され、前記テーブルにおける前記動き候補の配列は、前記テーブルへの前記動き候補の追加の順に基づく、ことと、

第 1 の映像ブロックに対する動き候補を導出するための動き候補リストの導出処理を実行することであって、前記動き候補リストの導出処理は、順に、前記 1 または複数のテーブルのうちテーブルにおける 1 または複数の動き候補を選択的にチェックすることを含む、ことと、

前記動き候補リストに基づいて、動きベクトル予測因子として用いられる動き情報を導出することと、

前記動き情報、および、動きベクトルと前記動きベクトル予測因子との間の MVD (Motion Vector Difference) に基づいて、前記第 1 の映像ブロックと前記映像ブロックを含む映像のビットストリームの間の変換を実行することであって、前記 MVD は、前記ビットストリームにて示される、ことと、
を行わせ、

前記動き候補リストを更新するために用いられる前記チェックされた動き候補のうちの

10

20

30

40

50

少なくとも1つの動き候補は、前記第1の映像ブロックの参照ピクチャと同じ参照ピクチャを有する、装置。

【請求項21】

プロセッサに、

1または複数のテーブルを維持することであって、各テーブルは、動き候補のセットを含み、各動き候補は、対応する動き情報に関連付けられ、前の映像ブロックから導出され、前記テーブルにおける前記動き候補の配列は、前記テーブルへの前記動き候補の追加の順に基づく、ことと、

第1の映像ブロックに対する動き候補を導出するための動き候補リストの導出処理を実行することであって、前記動き候補リストの導出処理は、順に、前記1または複数のテーブルのうちのテーブルにおける1または複数の動き候補を選択的にチェックすることを含む、ことと、

前記動き候補リストに基づいて、動きベクトル予測因子として用いられる動き情報を導出することと、

前記動き情報、および、動きベクトルと前記動きベクトル予測因子との間のMVD (Motion Vector Difference) に基づいて、前記第1の映像ブロックと前記映像ブロックを含む映像のビットストリームの間の変換を実行することであって、前記MVDは、前記ビットストリームにて示される、ことと、
を行わせ、

前記動き候補リストを更新するために用いられる前記チェックされた動き候補のうちの少なくとも1つの動き候補は、前記第1の映像ブロックの参照ピクチャと同じ参照ピクチャを有す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22】

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、

1または複数のテーブルを維持することであって、各テーブルは、動き候補のセットを含み、各動き候補は、対応する動き情報に関連付けられ、前の映像ブロックから導出され、前記テーブルにおける前記動き候補の配列は、前記テーブルへの前記動き候補の追加の順に基づく、ことと、

第1の映像ブロックに対する動き候補を導出するための動き候補リストの導出処理を実行することであって、前記動き候補リストの導出処理は、順に、前記1または複数のテーブルのうちのテーブルにおける1または複数の動き候補を選択的にチェックすることを含む、ことと、

前記動き候補リストに基づいて、動きベクトル予測因子として用いられる動き情報を導出することと、

前記動き情報、および、動きベクトルと前記動きベクトル予測因子との間のMVD (Motion Vector Difference) に基づいて、前記第1の映像ブロックから前記ビットストリームを生成することであって、前記MVDは、前記ビットストリームにて示される、ことと、

前記ビットストリームを非一時的コンピュータ可読媒体に格納することと、
を有し、

前記動き候補リストを更新するために用いられる前記チェックされた動き候補のうちの少なくとも1つの動き候補は、前記第1の映像ブロックの参照ピクチャと同じ参照ピクチャを有する、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、2019年1月1

10

20

30

40

50

6日出願の国際特許出願PCT/CN2019/072058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、および国際特許出願第PCT/CN2019/072058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。

10

【発明の概要】

【0004】

本明細書は、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、映像復号化の方法は、テーブルを維持することによって、各テーブルは、動き候補のセットを含み、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、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であって、変換を行うことは、動き候補のセットのうちの少なくとも一部を予測因子として使用して第1の映像ブロックの動き情報を処理する、こととを含むように提供される。

20

【0006】

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。

【0007】

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る。

30

【図面の簡単な説明】

【0008】

【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: Transform Block)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。

40

【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。

【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。

50

- 【図 1 2】 $N \times 2N$ 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 1 3】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 1 4】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。
- 【図 1 5】結合双方向予測マージ候補の例を示す。
- 【図 1 6】動きベクトル予測候補の導出処理の例を示す。
- 【図 1 7】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 1 8】CU の動き予測のための例示的な ATMVP (Alternative Temporal Motion Vector Prediction) を示す。
- 【図 1 9】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 2 0】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。 10
- 【図 2 1】バイラテラルマッチングの例を示す。
- 【図 2 2】テンプレートマッチングの例を示す。
- 【図 2 3】FRUC (Frame Rate Up Conversion) におけるユニラテラル動き推定 (ME: Motion Estimation) の例を示す。
- 【図 2 4】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 2 5】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 2 6】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 2 7 A】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 2 7 B】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。 20
- 【図 2 8】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。
- 【図 2 9】映像ビットストリーム処理の別の例示的な方法のフローチャートである。
- 【図 3 0】提案される HMVP 方法による復号化フローチャートの一例を示す。
- 【図 3 1】提案される HMVP 方法を用いたテーブルの更新例を示す。
- 【図 3 2 A】冗長性除去に基づく LUT 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 3 2 B】冗長性除去に基づく LUT 更新方法 (1 つの冗長性動き候補を除去する) の例を示す。 30
- 【図 3 3 A】冗長性除去に基づく LUT 更新方法 (複数の冗長性動き候補を除去する) の例を示す。
- 【図 3 3 B】冗長性除去に基づく LUT 更新方法 (複数の冗長性動き候補を除去する) の例を示す。
- 【図 3 4】タイプ 1 のブロックとタイプ 2 のブロックとの相違点の一例を示す。
- 【発明を実施するための形態】
- 【0009】
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。
- 【0010】 40
1. 導入
- 【0011】
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、AMVP モード) に関する。HEVC 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。
- 【0012】
簡単な説明
- 【0013】 50

映像符号化規格は、主に周知のITU-TおよびISO/IEC規格の開発によって発展してきた。ITU-TはH.261とH.263を作り、ISO/IECはMPEG-1とMPEG-4 Visualを作り、両団体はH.262/MPEG-2 VideoとH.264/MPEG-4 AVC (Advanced Video Coding)とH.265/HEVC規格を共同で作った。H.262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HEVCエンコーダフレームワークの一例を図1に示す。

【0014】

2.1 パーティション構造

【0015】

2.1.1 H.264/AVCにおけるパーティションツリー構造

【0016】

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。

【0017】

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。

【0018】

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たり1つの動きベクトル(MV)のみが許可される。

【0019】

2.1.2 HEVCにおけるパーティションツリー構造

【0020】

HEVCにおいて、CTUは、様々な局所的特徴に適應するように、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)予測を使用した、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。

【0021】

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。

【0022】

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、エンコーダ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。

【0023】

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

10

20

30

40

50

、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つの変換ユニットのツリー (TU) とを有する。

【0024】

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、 64×64 から 4×4 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。

10

【0025】

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 4×4 、 8×8 、 16×16 、および 32×32 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 4×4 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。

20

【0026】

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

【0027】

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

【0028】

残差符号化の場合、 CB は、変換ブロック (TB) 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 CB および TB の分割のみを指定する。サイズ $M \times M$ の所与の輝度 CB 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、 SPS 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。エンコーダは、それが使用することになる最大および最小輝度 TB サイズを示す。 CB サイズが最大 TB 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 TB サイズとなる場合、分割を行わないことが、非明示的に行われる。輝度 TB サイズが 4×4 である場合を除き、彩度 TB サイズは、各次元において輝度 TB サイズの半分であり、この場合、4つの 4×4 輝度 TB によって覆われる領域には1つの 4×4 彩度 TB が使用される。イントラピクチャ予測 CU の場合、最近の近傍の TB (CB 内または CB 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。

30

40

【0029】

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

【0030】

50

2.1.2.2 親子ノード

【0031】

CTB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、CTB）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。

10

【0032】

2.1.3 JEMにおけるより大きいCTUを有する4分木+2分木ブロック構造

【0033】

HEVCを超えた将来の映像符号化技術を探索するため、2015年には、VCEGとMPEGが共同でJVET (Joint Video Exploration Team) を設立した。それ以来、多くの新しい方法がJVETによって採用され、JEM (Joint Exploration Mode) と呼ばれる参照ソフトウェアに組み込まれてきた。

【0034】

2.1.3.1 QTBTブロックの分割構造

20

【0035】

HEVCとは異なり、QTBT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、CU、PU、TUのコンセプトの切り離しを取り除き、CUパーティションの形状の柔軟性を向上させる。QTBTブロック構造において、CUは正方形または長方形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、QTBT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。

30

【0036】

QTBT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

40

【0037】

QTBT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64x64ブロックの彩度サンプルを有する128x128の輝度サンプルとして設定し、MinQTSizeを16x16として設定し、MaxBTSizeを64x64として設定し、MinBTSize (幅および高さの両方について) を4x4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、16x16 (即ち、MinQTSize) から128x128 (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

50

ノードが 128×128 である場合、サイズが $MaxBTSize$ (すなわち、 64×64) 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) 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、 256×256 個の輝度サンプルである。

10

【0038】

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がない。

【0039】

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。

20

【0040】

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、 4×8 および 8×4 ブロックのために双予測はサポートされず、 4×4 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。

30

【0041】

2.1.4 VVCの3分木

【0042】

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(d)、(e)に示すように、3分木(TT)パーティションを2つ以上、すなわち、水平および垂直中心側の3分木を導入する。

【0043】

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。

40

【0044】

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葉は、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葉が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。

【0045】

50

2.1.5 分割構造

【0046】

この応答で使用されるツリー構造は、マルチツリータイプ (Multi-Tree Type: MTT) と呼ばれ、QTTを一般化したものである。QTT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。

【0047】

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。

【0048】

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。

【0049】

1つの領域ツリーは、1つのCTUを4×4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。

【0050】

2.2 HEVC/H.265におけるインター予測

【0051】

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。

【0052】

1つのCUがスキップモードにて符号化される場合、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。

【0053】

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。

【0054】

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スライスのみにも双方向予測が利用可能である。

【0055】

10

20

30

40

50

以下、H E V C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 6 】

2 . 2 . 1 マージモード

【 0 0 5 7 】

2 . 2 . 1 . 1 マージモードの候補の導出

【 0 0 5 8 】

マージモードを使用してP U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、H E V C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

【 0 0 5 9 】

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P Uごとに一定数の候補を想定しているため、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(Max Num Merge Cand)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ため、最良マージ候補のインデックスは、短縮された単項2値化(TU: truncated unary binarization)を使用して符号化される。C Uのサイズが8に等しい場合、現在のC UのすべてのP Uは、2 N × 2 N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。

【 0 0 6 0 】

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。

【 0 0 6 1 】

2 . 2 . 1 . 2 空間的候補の導出

【 0 0 6 2 】

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A₁、B₁、B₀、A₀、B₂である。位置A₁、B₁、B₀、A₀のいずれかのP Uが利用可能でない場合(例えば、別のスライスまたはタイ尔に属しているため)、またはイントラ符号化された場合にのみ、位置B₂が考慮される。位置A₁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、2 N × 2 Nとは異なるパーティションに関連付けられた「第2のP U」である。一例として、図12は、それぞれN × 2 Nおよび2 N × Nの場合の第2のP Uを示す。現在のP UをN × 2 Nに分割する場合、リスト構築に位置A₁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P Uのみを有するためには冗長である。同様に、現在のP Uを2 N × Nに分割する場合、位置B₁は考慮されない。

【 0 0 6 3 】

10

20

30

40

50

2.2.1.3 時間的候補の導出

【0064】

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PUに基づいて、スケール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 (co-located 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ル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ルしたものである。 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。このスケール処理の実際的な実現については、HEVC仕様に記載されている。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ル処理の説明。

10

【0065】

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_0 と候補 C_1 との間で時間的候補の位置を選択する。位置 C_0 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_1 が使用される。そうでない場合、位置 C_0 が時間的マージ候補の導出に使用される。

20

【0066】

2.2.1.4 追加候補の挿入

【0067】

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、 mvL_0 および $refIdxL_0$ 、または mvL_1 および $refIdxL_1$ 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。ここで規定される、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。

30

【0068】

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。

40

【0069】

2.2.1.5 並列処理のための動き推定領域

【0070】

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

50

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、「log2_parallel_merge_level_minus2」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。

7.3.2.3 ピクチャパラメータセットRBS P構文

7.3.2.3.1 一般ピクチャパラメータセットRBS P構文

【0071】

【表1】

10

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

20

30

【0072】

log2_parallel_merge_level_minus2 + 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Log2ParMrgLevelの値を指定する。log2_parallel_merge_level_minus2の値は、0 ~ CtbLog2SizeY - 2を含む範囲内とする。

変数Log2ParMrgLevelは、以下のように導出される。

$$\text{Log2ParMrgLevel} = \text{log2_parallel_merge_level_minus2} + 2 \quad (7-37)$$

注3: Log2ParMrgLevel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、Log2ParMrgLevelが6に等しい場合、64x64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。

40

【0073】

2.2.2 AMVPモードにおける動きベクトル予測

【0074】

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

50

ーダは、候補リストから最良の予測因子を選択し、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。この場合の符号化対象の最大値は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。

【0075】

2.2.2.1 動きベクトル予測候補の導出

【0076】

図16に、動きベクトル予測候補の導出処理をまとめる。

【0077】

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を導出するために、図11に示したように、5つの異なる位置にある各PU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。

【0078】

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。

【0079】

2.2.2.2 空間的動きベクトル候補

【0080】

空間的動きベクトル候補の導出において、図11に示したような位置にあるPU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PUの左側のための導出の順序は、 A_0 、 A_1 、スケーリングされた A_0 、スケーリングされた A_1 として規定される。現在のPUの上側のための導出の順序は、 B_0 、 B_1 、 B_2 、スケーリングされた B_0 、スケーリングされた B_1 、スケーリングされた B_2 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。

・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POC)

- (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POC)

・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POC)

- (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POC)

【0081】

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、POCが近傍のPUの参照ピクチャと現在のPU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PU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MV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。

【0082】

空間的スケーリング処理において、図17に示すように、時間的スケーリングと同様に

10

20

30

40

50

して、近傍のPUの動きベクトルをスケーリングする。主な違いは、現在のPU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。

【0083】

2.2.2.3 時間的動きベクトル候補

【0084】

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）。参照ピクチャインデックスはデコーダに信号通知される。

【0085】

2.2.2.4 AMVP情報の信号通知

【0086】

AMVP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、MVD、およびmv予測因子候補インデックスを信号通知することができる。

構文テーブル：

【0087】

10

20

30

40

50

【表 2】

prediction_unit(x0, y0, nPbW, nPbH) {	記述子	
if(cu_skip_flag[x0][y0]) {		
if(MaxNumMergeCand > 1)		
merge_idx [x0][y0]	ae(v)	
} else { /* MODE_INTER */		
merge_flag [x0][y0]	ae(v)	
if(merge_flag[x0][y0]) {		10
if(MaxNumMergeCand > 1)		
merge_idx [x0][y0]	ae(v)	
} else {		
if(slice_type == B)		
inter_pred_idc [x0][y0]	ae(v)	
if(inter_pred_idc[x0][y0] != PRED_L1) {		
if(num_ref_idx_l0_active_minus1 > 0)		
ref_idx_l0 [x0][y0]	ae(v)	20
mvd_coding(x0, y0, 0)		
mvp_l0_flag [x0][y0]	ae(v)	
}		
if(inter_pred_idc[x0][y0] != PRED_L0) {		
if(num_ref_idx_l1_active_minus1 > 0)		
ref_idx_l1 [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		30
} else		
mvd_coding(x0, y0, 1)		
mvp_l1_flag [x0][y0]	ae(v)	
}		
}		
}		
}		
}		
}		

【 0 0 8 8 】

7 . 3 . 8 . 9 動きベクトル差構文

【 0 0 8 9 】

40

50

【表 3】

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

10

20

【 0 0 9 0 】

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

【 0 0 9 1 】

2.3.1 サブCUに基づく動きベクトル予測

30

【 0 0 9 2 】

QTBT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。エンコーダにおいて、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隣接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。

40

【 0 0 9 3 】

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。

【 0 0 9 4 】

2.3.1.1 代替の時間的動きベクトル予測

【 0 0 9 5 】

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

50

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）。

【0096】

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。

【0097】

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。最初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。

【0098】

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。

【0099】

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。

【0100】

2.3.1.2 空間的 - 時間的動きベクトル予測

【0101】

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 4×4 サブCUであるA、B、C、およびDを含む 8×8 CUを考える。現在のフレームの近傍の 4×4 ブロックには、a、b、c、d というラベルが付けられている。

【0102】

サブCUのAの動きの導出は、その2つの空間的近傍を特定することによって始まる。第1の近傍は、サブCUの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の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（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、HEVCに規定

10

20

30

40

50

されているTMVP (Temporal Motion Vector Predictor) 導出と同様の手順に従って、サブブロックAのTMVP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CUの動きベクトルとする。

【0103】

図20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CUの例を示す。

【0104】

2.3.1.3 サブCU動き予測モード信号通知

【0105】

サブCUモードは追加のマー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。ATMVPモードおよびSTMVPモードを表すように、各CUのマーージ候補リストに2つの追加のマーージ候補を加える。シーケンスパラメータセットがATMVPおよびSTMVPが有効であることを示す場合、7個までのマーージ候補を使用する。追加のマーージ候補の符号化ロジックは、HMにおけるマー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CUについて、2つの追加のマーージ候補に対して2回以上のRDチェックが必要となる。

【0106】

JEMにおいて、マーージインデックスのすべてのピンは、CABACによって符号化されたコンテキストである。一方、HEVC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。

【0107】

2.3.2 適応型動きベクトル差解像度

【0108】

HEVC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差分(MVD: Motion Vector Difference)(動きベクトルとPUの予測動きベクトルとの差)が信号通知される。JEMにおいて、LAMVR (Locally Adaptive Motion Vector Resolution)が導入される。JEMにおいて、MVD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復号化できる。MVD解像度は符号化ユニット(CU)レベルで制御され、MVD解像度フラグは、少なくとも1つの非ゼロMVDの構成要素を有する各CUに対して条件付きで信号通知される。

【0109】

少なくとも1つの非ゼロMVDの構成要素を有するCUの場合、 $1/4$ 輝度サンプルMV精度がCU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MV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MV精度が使用されるかまたは4輝度サンプルMV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。

【0110】

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

【0111】

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

10

20

30

40

50

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、以下のエン符号化方式が適用される。

【0112】

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。

【0113】

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。

【0114】

2.3.3 パターンマッチング動きベクトル導出

【0115】

PMMVD (Pattern Matched Motion Vector Derivation) モードは、FRUC (Frame-Rate Up Conversion) 技術に基づく特殊な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。

【0116】

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。

【0117】

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コストの選択に基づく。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。

【0118】

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。

【0119】

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、 $M \times M$ 個のサブ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、予め規定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。そして、各サブCUのMVを導出する。

【0120】

10

20

30

40

50

【数 1】

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

【0121】

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。

10

【0122】

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。

20

【0123】

2 . 3 . 1 CU レベル MV 候補セット

【0124】

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

30

(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

【0125】

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 をスケールリングすることで MV b を導出する。

40

【0126】

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。具体的には、現在の CU の (0 , 0)、(W / 2 , 0)、(0 , H / 2)、(W / 2 , H / 2) の位置の補間 MV を加算する。

【0127】

AMVP モードで FRUC を適用する場合、元の AMVP 候補を CU レベル MV 候補セットにも加える。

50

【 0 1 2 8 】

C Uレベルにおいて、A M V P C Uのための最大15個のM VおよびマージC Uのための最大13個のM Vを候補リストに加える。

【 0 1 2 9 】

2 . 3 . 3 . 2 サブC UレベルM V候補セット

【 0 1 3 0 】

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

10

【 0 1 3 1 】

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 2 】

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 3 】

サブC Uレベルにおいて、最大17個のM Vが候補リストに追加される。

20

【 0 1 3 4 】

2 . 3 . 3 . 3 補間M Vフィールドの生成

【 0 1 3 5 】

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 6 】

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、 4×4 ブロックレベルでトラバースされる。各 4×4 ブロックにおいて、現在のピクチャ(図23に示す)の 4×4 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。 4×4 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。

30

【 0 1 3 7 】

2 . 3 . 3 . 4 補間およびマッチングコスト

【 0 1 3 8 】

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、動き補償補間が必要である。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プレートマッチングの両方に双線形補間を使用する。

40

【 0 1 3 9 】

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 0 】

【 数 2 】

50

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

【0141】

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。

【0142】

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。

10

【0143】

2.3.3.5 MV の改良

【0144】

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

【0145】

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

【0146】

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、そのような限定はない。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

$costBi \ factor * \min(cost0, cost1)$ 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。

そうでない場合、

$list1$ からの単一予測を用いる。

40

【0147】

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。

【0148】

2.3.4 デコーダ側動きベクトル改良

【0149】

50

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双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、デコーダ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。

【0150】

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。

【0151】

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。

【0152】

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

【0153】

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。

【0154】

- ブロック1~4の空間的候補

【0155】

- ブロック1~4の外挿アフィン候補

【0156】

- ATMVP

【0157】

- STMVP

【0158】

- 仮想アフィン候補

【0159】

- 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)。

【0160】

- 外挿アフィン候補(ブロック5)

10

20

30

40

50

【 0 1 6 1 】

- 時間的候補 (H E V C のように導出)

【 0 1 6 2 】

- 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 3 】

- 結合候補

【 0 1 6 4 】

- ゼロ候補

【 0 1 6 5 】

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前に挿入する。

10

【 0 1 6 6 】

いくつかの実施形態において、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で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 7 】

2 . 3 . 6 共有マージリスト J V E T - M 0 1 7 0

【 0 1 6 8 】

小さなスキップ / マージ符号化された C U を並列処理することを有効にするために、 C U 分割木における 1 つの祖先ノードのすべての葉の符号化ユニット (C U) に対して同じマージ候補リストを共有することが提案される。祖先ノードをマージ共有ノードと呼ぶ。マージ共有ノードが葉 C U であるように見せかけて、マージ共有ノードにおいて共有マージ候補リストを生成する。

20

【 0 1 6 9 】

T y p e - 2 の定義において、復号化の構文解析段階において、 C T U 内部の C U ごとにマージ共有ノードを決定する。また、マージ共有ノードは、葉 C U の祖先ノードであり、以下の 2 つの基準を満たさなければならない。

【 0 1 7 0 】

マージ共有ノードのサイズは、サイズ閾値以上であること。

30

【 0 1 7 1 】

マージ共有ノードにおいて、子 C U のサイズは、サイズ閾値よりも小さいこと。

【 0 1 7 2 】

さらに、マージ共有ノードのサンプルがピクチャ境界の外側でないことを保証する必要がある。構文解析段階において、祖先ノードが基準 (1) および (2) を満たすが、ピクチャ境界の外側にいくつかのサンプルを有する場合、この祖先ノードはマージ共有ノードではないので、先に進んでその子 C U のためのマージ共有ノードを見出す。

【 0 1 7 3 】

図 3 5 に、 T y p e - 1 と T y p e - 2 の定義の違いの一例を示す。本例において、親ノードは、 3 つの子 C U に 3 分割される。親ノードのサイズは 1 2 8 である。 T y p e - 1 定義の場合、 3 つの子 C U は別々のマージ共有ノードである。しかし、 T y p e - 2 定義の場合、親ノードはマージ共有ノードである。

40

【 0 1 7 4 】

提案した共用マージ候補リストアルゴリズムは、並進マージ (マージモードおよびトライアングルマージモードを含む、履歴に基づく候補もサポートされる) およびサブブロックに基づくマージモードをサポートする。すべての種類のマージモードにおいて、共有マージ候補リストアルゴリズムの挙動は基本的に同じに見え、マージ共有ノードが葉 C U であるように見せるだけの候補をマージ共有ノードに生成する。それには 2 つの大きな利点がある。第 1 の利点は、マージモードのための並列処理を有効にすることであり、第 2 の利点は、すべての葉 C U のすべての計算をマージ共有ノードに共有することである。その

50

ため、ハードウェアコーデックのためのすべてのマージモードのハードウェアコストを大幅に低減することができる。提案した共有マージ候補リストアルゴリズムにより、エンコーダとデコーダはマージモードの並列符号化に容易に対応でき、マージモードのサイクルバジェット問題を軽減する。

【0175】

2.3.7 タイル群

【0176】

J V E T - L 0 6 8 6 では、タイルグループに代えるためスライスが削除され、H E V C 構文要素 `slice_address` がタイルグループの最初のタイルのアドレスとして `tile_group_header` 内の `tile_group_address` に置き換えられる（ピクチャ内に複数のタイルがある場合）。

10

【0177】

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

【0178】

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。

【0179】

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 4×4 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。

20

【0180】

4. いくつかの例

【0181】

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、より高い符号化効率を有する映像符号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。

30

【0182】

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/復号化処理を行うことを可能にするため、L U T に基づく動きベクトル予測は、H M V P H i s t o r y - b a s e d M o t i o n V e c t o r P r e d i c t i o n) 法と呼ぶこともできる。L U T に基づく動きベクトル予測方法において、以前に符号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/復号化の間、維持される。L U T に記憶されたこれらの動き候補を H M V P 候補と称する。1つのブロックの符号化/復号化の間、L U T 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/ A M V P 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、L U T を使用してもよい。更新された L U T は、その後、後続のブロックを符号化するために用いられる。このように、L U T 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

40

50

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。

【0183】

いくつか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。

【0184】

以下の実施例は、一般的な概念を説明する。

10

【0185】

ルックアップテーブルの例

【0186】

例A1：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。

a．動き候補の動き情報は、ここでは、予測方向、参照インデックス/ピクチャ、動きベクトル、LICフラグ、アフィンフラグ、MVD精度、MVD値の一部または全部を含んでもよい。

b．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、ブロック位置情報および/またはブロック形状をさらに含んでもよい。

20

【0187】

LUTの選択

【0188】

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とき、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。例B2：ルックアップテーブルの選択は、ブロックの位置に依存してもよい。

【0189】

ルックアップテーブルの使用法

30

【0190】

例C1：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。

【0191】

例C2：1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。

40

c．それらは、サブブロックマージモード、アフィンマージモード、トライアングルマージモード、インター-イントラマージモード、MMVD (Merge with MVD) モードのような他の種類のマージモードに使用してもよい。

d．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、マージ候補リストが満杯になっていない。

ii．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。

iii．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。

50

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、HEVCデザイン、またはJEM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、TMVP、STMVP、ATMVPなどを含む）マージ候補の数が、最大許容のマージ候補から、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、SPS / PPS / 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。

10

4 . 他の例において、既にマージ候補リストに含まれていない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、HEVC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。

vi . 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、c1) 動き情報を追加しない、または、c2) 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。

20

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、TMVP、STMVP、ATMVP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、ATMVP、STMVP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。

30

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（ $K \leq M$ ）の候補とのみ比較することができる。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、エンコーダからデコーダに信号通知されてもよい。

40

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テーブルから選択し、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）。

50

e . 隣接 / 非隣接の空間的または時間的ブロックから導出されるような、他のマージ（またはアフィンマージまたは他のインター符号化方法）候補をチェックする前に、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。

f . ルックアップテーブルに少なくとも1つの動き候補がある場合、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。

【 0 1 9 2 】

例 C 3 : 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ための予測因子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く、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 . それらは、MVD の一部のみを符号化する SMVD (Symmetric Motion Vector Difference) 符号化に使用してもよい（例えば、1つの参照ピクチャリストに対して信号通知された MVD のみであり、別の参照ピクチャリストから導出される）。

c . それらは、MV の一部のみを符号化する SMV (Symmetric Motion Vector) 符号化に使用してもよい（例えば、1つの参照ピクチャリストに対して信号通知されたもののみであり、別の参照ピクチャリストから導出される）。

d . 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 候補をチェックまたは挿入した後、AMVP 候補リストが満杯になっていない。

ii . AMVP 候補リストが、空間的近傍から選択し、ブ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 . 上側の近傍のブロックからの 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、左側の近傍のブロックからの AMVP 候補がスケーリング無しで存在しない場合。

iv . 特定の AMVP 候補を挿入した後、AMVP 候補リストが満杯になっていない。

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

vi . 実施例 C 2 の vi . 3 および 4 に記載されたものと同様の規則は、AMVP モードに適用されてよい。

e . 隣接 / 非隣接の空間的または時間的ブロックから導出されるような、他の AMVP（または SMVD / SMV / アフィンインターまたは他のインター符号化方法）候補をチェックする前に、動き候補のチェックを有効にしてもよい。

f . ルックアップテーブルに少なくとも1つの動き候補がある場合、動き候補のチェックを有効にしてもよい。

g . 現在の参照ピクチャと同一の参照ピクチャを有する（すなわち、POC (Picture - Order - Count) が同一である）動き候補をチェックする。すなわち、動き候補が現在の参照ピクチャと同一の参照ピクチャを含む場合、対応する動きベクトルは、AMVP 候補リスト構成処理を考慮してもよい。

i . 代替的に、更に、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 にて）チェックする。すなわち、動き候補が現在の参照ピクチャとは異なる参照ピクチャを有する場合、対応する動きベクトルは、AMVP 候補リスト構築処理を考慮してもよい。

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。すなわち、同一の参照ピクチャを有する動き候補に対して、より高い優先順位を割り当てる。

iii . 代替的に、動き候補は、マージにおいて同様にチェックされる。

iv . 1つの動き候補が双方向予測候補である場合、まず参照ピクチャリスト X の参照ピクチャ（例えば、参照ピクチャの参照ピクチャインデックスまたはピクチャ順序カウ

10

20

30

40

50

ンタ)をチェックし、次に現在の参照対象ピクチャリストがXである場合、参照ピクチャリストY ($Y \neq X$ 、例えば、 $Y = 1 - X$)の参照ピクチャをチェックしてもよい。

v. 代替的に、1つの動き候補が双方向予測候補である場合、まず参照ピクチャリストY ($Y \neq X$ 、例えば、 $Y = 1 - X$)の参照ピクチャ(例えば、参照ピクチャの参照ピクチャインデックスまたはピクチャ順序カウンタ)をチェックし、次に現在の参照対象ピクチャリストがXである場合、参照ピクチャリストXの参照ピクチャをチェックしてもよい。

vi. 代替的に、参照ピクチャリストの参照ピクチャが、すべてのチェック対象の動き候補に関連付けられたY ($Y \neq X$ 、例えば、 $Y = 1 - X$)である前に、参照ピクチャリストの参照ピクチャが、すべてのチェック対象の動き候補に関連付けられたXであるかどうかをチェックしてもよい。

10

【0193】

例C4: ルックアップテーブルにおける動き候補のチェック順序は、以下のように規定される(K (K - 1)個の動き候補をチェックすることができるとする):

a. ルックアップテーブルにおける最後のK個の動き候補。

b. 最初のK % L 候補であって、Lは、K / Lである場合のルックアップテーブルのサイズである。

c. K / Lである場合、ルックアップテーブルにおけるすべての候補(L個の候補)。

d. 代替的に、さらに、動き候補インデックスの降順に基づいてもよい。

e. 代替的に、さらに、動き候補インデックスの昇順に基づいてもよい。

20

f. 代替的に、動き候補に関連付けられた位置と現在のブロックの距離などの候補情報に基づいて、K個の動き候補を選択する。

i. 一例において、K個の最も近い動き候補を選択する。

ii. 一例において、距離を算出する際に、候補情報は、ブロックの形状をさらに考慮されてもよい。

g. 一例において、L個の候補を含むテーブルからの動き候補のKのチェック順序は、次のように規定されてもよい: a_0 、 $a_0 + T_0$ 、 $a_0 + T_0 + T_1$ 、 $a_0 + T_0 + T_1 + T_2$ 、...、 $a_0 + T_0 + T_1 + T_2 + \dots + T_{K-1}$ のインデックスを持つ候補を順に選択し、 a_0 と T_i (i は $0 \dots K-1$ である)は整数値である。

i. 一例において、 a_0 は、0 (すなわち、テーブルにおける動き候補の最初のエントリ)に設定される。代替的に、 a_0 は $(K - L / K)$ に設定される。演算「/」は、結果をゼロに切り捨てる整数除算として規定される。代替的に、 a_0 は、0と L / K との間の任意の整数に設定される。

30

1. 代替的に、 a_0 の値は、現在のブロックおよび近傍のブロックの符号化情報に依存してもよい。

ii. 一例において、すべての間隔 T_i (i は $0 \dots K-1$ である)は同じであり、例えば L / K である。演算「/」は、結果をゼロに切り捨てる整数除算として規定される。

iii. 一例において、 (K, L, a_0, T_i) は、 $(4, 16, 0, 4)$ 、または $(4, 12, 0, 3)$ 、または $(4, 8, 0, 1)$ 、または $(4, 16, 3, 4)$ 、または $(4, 12, 2, 3)$ 、または $(4, 8, 1, 2)$ に設定される。 T_i はすべての i について同じである。

40

iv. このような方法は、KがLよりも小さい場合にのみ適用されてもよい。

v. 代替的に、Kが閾値以上である場合、例C4のc部を適用してもよい。閾値は、Lとして規定されてもよく、またはKに依存してもよく、またはブロックごとに適応的に変更されてもよい。一例において、閾値は、ルックアップテーブルから新しい動き候補を追加する前のリストにおける利用可能な動き候補の数に依存してもよい。

h. 一例において、L個の候補を含むテーブルからの動き候補のKのチェック順序は、次のように規定されてもよい: a_0 、 $a_0 - T_0$ 、 $a_0 - T_0 - T_1$ 、 $a_0 - T_0 - T_1 - T_2$ 、...、 $a_0 - T_0 - T_1 - T_2 - \dots - T_{K-1}$ のインデックスを持つ候補を順に選択し、 a_0 と T_i (i は $0 \dots K-1$ である)は整数値である。

50

i . 一例において、 a_0 は、 $L - 1$ （すなわち、テーブルにおける動き候補の最後のエントリ）に設定される。代替的に、 a_0 は、 $L - 1 - L / K$ と $L - 1$ との間の任意の整数に設定される。

ii . 一例において、すべての間隔 T_i （ i は $0 \dots K - 1$ である）は同じであり、例えば L / K である。

iii . 一例において、 (K, L, a_0, T_i) は、 $(4, 16, L - 1, 4)$ 、または $(4, 12, L - 1, 3)$ 、または $(4, 8, L - 1, 1)$ 、または $(4, 16, L - 4, 4)$ 、または $(4, 12, L - 3, 3)$ 、または $(4, 8, L - 2, 2)$ に設定される。 T_i はすべての i について同じである。

iv . このような方法は、 K が L よりも小さい場合にのみ適用されてもよい。

v . 代替的に、更に、 K が閾値以上である場合、例C4のc部を適用してもよい。閾値は、 L として規定されてもよく、または K に依存してもよく、またはブロックごとに適応的に変更されてもよい。一例において、閾値は、ルックアップテーブルから新しい動き候補を追加する前のリストにおける利用可能な動き候補の数に依存してもよい。

i . ルックテーブルから動き候補を選択する数および/または方法は、符号化情報、例えばブロックサイズ/ブロック形状に依存してもよい。

i . 一例において、より小さいブロックサイズの場合、最後の K 個の動き候補を選択する代わりに、（最後から始まらない）他の K 個の動き候補を選択してもよい。

ii . 一例において、符号化情報は、AMVPモードであってもよいし、マージモードであってもよい。

iii . 一例において、符号化情報は、アフィンモードまたは非アフィンAMVPモードまたは非アフィンマージモードであってもよい。

iv . 一例において、符号化情報は、アフィンAMVP（インター）モード、またはアフィンマージモード、または非アフィンAMVPモード、または非アフィンマージモードであってもよい。

v . 一例において、符号化情報は、CPR（Current Picture Reference）モードであってもよいし、CPRモードでなくてもよい。

vi . 代替的に、ルックアップテーブルから動き候補を選択する方法は、ルックアップテーブルにおける動き候補の数、および/または、ルックアップテーブルから新しい動き候補を追加する前のリストにおける利用可能な動き候補の数にさらに依存してもよい。

j . 一例において、チェック対象のルックアップテーブルにおける利用可能な動き候補の最大数（すなわち、マージ/AMVP候補リストに追加されてもよい）は、ルックアップテーブルにおける利用可能な動き候補の数（ $N_{\text{available}}^{\text{inLUT}}$ によって示される）、および/または、追加対象の（これは、予め規定されていてもよいし、信号通知されていてもよい）最大許容動き候補の数（ N_{maxMC} によって示される）、および/または、ルックアップテーブルから候補をチェックする前の候補リストにおける利用可能な動き候補の数（ $N_{\text{available}}$ によって示される）に依存していてもよい。

i . 一例において、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最大数は、 $(N_{\text{available}}^{\text{inLUT}}, N_{\text{maxMC}}, N_{\text{available}})$ の最小値に設定される。

ii . 代替的に、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最大数は、 $(N_{\text{available}}^{\text{inLUT}}, N_{\text{maxMC}} - N_{\text{available}})$ の最小値に設定される。

iii . 一例において、 $N_{\text{available}}$ は、空間的または時間的（隣接および/または非隣接）な近傍のブロックから導出された挿入候補の数を示す。代替的に、サブブロック候補（AMTP、STMVPなど）の数は、 $N_{\text{available}}$ にカウントされない。

iv . N_{maxMC} は、符号化モードに依存してもよく、例えば、マージモードおよびAMVPモードにおいて、 N_{maxMC} は異なる値に設定されてもよい。一例において、マージモードの場合、 N_{maxMC} は、4、6、8、10などに設定してもよく、AMVPモードの場合、 N_{maxMC} は、1、2、4などに設定してもよい。

v . 代替的に、 N_{maxMC} は、ブロックサイズ、ブロック形状、スライスタイプなどのような他の符号化情報に依存してもよい。

10

20

30

40

50

k . 異なるルックアップテーブルのチェック順序は、次のサブセクションのルックアップテーブルの使用法で規定されている。

l . 一旦、マージ / A M V P 候補リストが、最大許容候補数に達すると、このチェック処理は終了する。

m . 一旦、マージ / A M V P 候補リストが、最大許容候補数から閾値 (T h) を減算した値に達すると、このチェック処理は終了する。一例において、 T h は、例えば、 1、 2、または 3 など、正の整数値として予め規定されてもよい。代替的に、 T h は、ブロックごとに適応的に変更されてもよい。代替的に、 T h は、 S P S / P P S / スライスヘッダ等において信号通知されてもよい。代替的に、 T h は、さらに、ブロック形状 / ブロックサイズ / 符号化モードなどに依存してもよい。代替的に、 T h は、 L U T からの動き候補を追加する前の利用可能な候補の数に依存してもよい。

10

n . 代替的に、追加された動き候補の数が最大許容動き候補数に達すると、それは終了する。最大許容動き候補数は、信号通知されてもよく、または予め規定されてもよい。代替的に、最大許容動き候補数は、ブロック形状 / ブロックサイズ / 符号化モード等にさらに依存してもよい。

o . テーブルサイズを示す 1 つのシンタックス要素ならびにチェック可能な動き候補の数 (すなわち、 $K = L$) は、 S P S、 P P S、スライスヘッダ、タイルヘッダにおいて信号通知されてもよい。

【 0 1 9 4 】

いくつかの実装形態において、ルックアップテーブルにおける動き候補は他の候補を導出するために用いられてもよく、導出された候補はブロックを符号化するために用いられてもよい。

20

【 0 1 9 5 】

いくつかの実装形態において、 1 つのブロックの動き情報符号化のためのルックアップテーブルの使用の有効 / 無効は、 S P S、 P P S、スライスヘッダ、タイルヘッダ、 C T U、 C T B、 C U、または P U、複数の C T U / C T B / C U / P U を覆う領域において信号通知されてもよい。

【 0 1 9 6 】

いくつかの実装形態において、ルックアップテーブルからの予測を適用するかどうかは、さらに符号化情報に依存してもよい。 1 つのブロックに適用しないと推測される場合、予測の指示の追加の信号通知はスキップされる。代替的に、 1 つのブロックに適用しないと推測される場合、ルックアップテーブルの動き候補にアクセスする必要はなく、関連する動き候補のチェックは省略される。

30

【 0 1 9 7 】

いくつかの実装形態において、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。

a . 一例において、現在のブロックの参照ピクチャに関連付けられたルックアップテーブルのみを、現在のブロックを符号化するために利用してもよい。

b . 一例において、現在のブロックを符号化するために、現在のブロックの同じスライスタイプおよび / または同じ量子化パラメータを有するピクチャに関連付けられたルックアップテーブルのみを利用してもよい。

40

【 0 1 9 8 】

ルックアップテーブルの更新

【 0 1 9 9 】

動き情報を有するブロックを符号化した後 (すなわち、 I n t r a B C 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。

【 0 2 0 0 】

上述したすべての例および実装形態において、ルックアップテーブルは、符号化情報、または以前符号化されたブロックからの符号化情報から導出された情報を復号化順に示す。

50

a. ルックアップテーブルは、並進動き情報、またはアフィン動き情報、またはアフィンモデルパラメータ、またはイントラモード情報、または照明補償情報等を含んでもよい。

b. 代替的に、ルックアップテーブルは、並進動き情報、またはアフィン動き情報、またはアフィンモデルパラメータ、またはイントラモード情報、または照明補償情報等のような情報を少なくとも2種類含んでもよい。

【0201】

追加の例示的な実施形態

【0202】

以前符号化されたブロックの動き情報としてHMVP候補を規定する、HMVP(History-based MVP)方法が提案される。符号化/復号化処理中、複数のHMVP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HMVP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30に示す。

10

【0203】

一例において、テーブルサイズはL(例えば、L=16または6、または44)に設定され、これは、L個までのHMVP候補をテーブルに追加することができることを示す。

【0204】

1つの実施形態(例11.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HMVP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(FIFO: First-In-First-Out)規則が適用される。図31は、FIFO規則を適用してHMVP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

【0205】

別の実施形態(発明11.g.iii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、LUT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。

【0206】

いくつかの例を以下に示す。

30

【0207】

図32Aは、新しい動き候補を追加する前に、LUTが満杯であった場合の例を示す。

【0208】

図32Bは、新しい動き候補を追加する前には、LUTが満杯でなかった場合の例を示す。

【0209】

図32Aおよび図32Aは、ともに、冗長性除去に基づくLUT更新方法(1つの冗長性動き候補を除去する)の例を示す。

【0210】

図33Aおよび図33Bは、冗長性除去に基づくLUT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つの場合の実装例を示す。

40

【0211】

図33Aは、新しい動き候補を追加する前に、LUTが満杯であった場合の例を示す。

【0212】

図33Bは、新しい動き候補を追加する前に、LUTが満杯でなかった場合の例を示す。

【0213】

HMVP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。TMVP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K0のHMVP、例えば、K0=16または6)までのすべてのHMVP候補を挿入する。HMVP候補に対してプ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

50

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、LUTからの動き候補の取り出しを終了する。

【0214】

同様に、HMVP候補は、AMVP候補リスト構築処理において使用されてもよい。TMVP候補の後に、テーブルにおける最後のK1個のHMVP候補の動きベクトルを挿入する。AMVP対象参照ピクチャと同じ参照ピクチャを有するHMVP候補のみを用いて、AMVP候補リストを構築する。HMVP候補に対してプルーニングを適用する。一例において、K1は4に設定される。

【0215】

図28は、映像処理装置2800のブロック図である。装置2800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2800は、スマートフォン、タブレット、コンピュータ、IoT(Internet of Things)受信機等により実装されてもよい。装置2800は、1つ以上のプロセッサ2802と、1つ以上のメモリ2804と、映像処理ハードウェア2806と、を含んでもよい。1つまたは複数のプロセッサ2802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2804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2806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よい。

10

【0216】

図29は、映像復号化方法2900の例のフローチャートである。方法2900は、テーブルを維持すること(2902)を含み、各テーブルは、動き候補のセットを含み、各動き候補は、対応する動き情報に関連付けられる。方法2900は、さらに、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(2904)を行うことを含み、変換を行うことは、動き候補のセットのうち少なくとも一部を予測因子として使用して第1の映像ブロックの動き情報を処理することを含む。

20

【0217】

方法2900に対して、いくつかの実施形態において、動き情報は、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少なくとも1つを含む。さらに、動き情報は、動き情報がどこから来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。いくつかの実施形態において、映像ブロックはCUまたはPUであり、映像の一部は1つ以上の映像スライスまたは1つ以上の映像ピクチャに対応してよい。

30

【0218】

いくつかの実施形態において、各LUTは、関連するカウンタを含み、カウンタは、映像の部分の最初においてゼロ値に初期化され、映像の部分における各符号化された映像領域ごとに増加される。映像領域は、符号化ツリーユニット、符号化ツリーブロック、符号化ユニット、符号化ブロックまたは予測ユニットのうちの一つを含む。いくつかの実施形態において、カウンタは、対応するLUTに対して、対応するLUTから除去された動き候補の数を示す。いくつかの実施形態において、動き候補のセットはすべてのLUTに対して同じサイズを有していてもよい。いくつかの実施形態において、映像の一部は1つの映像スライスに対応し、LUTの数は $N * P$ に等しく、 N は1つの復号化スレッド当たりのLUTを表す整数であり、 P は1つの映像スライスにおける最大符号化ユニットの行の数またはタイルの数を表す整数である。方法2900のさらなる詳細は、第4章に提供される実施例および以下に列挙される実施例に記載される。

40

【0219】

上述した方法/技術の特徴および実施形態を以下に説明する。

【0220】

1. テーブルを維持することであって、各テーブルは、動き候補のセットを含み、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、第1の映像ブロックと、第1の

50

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であって、変換を行うことは、動き候補のセットのうち少なくとも一部を予測因子として使用して第1の映像ブロックの動き情報を処理することを含む、ことと、を有する映像処理方法。

【0221】

2. 前記テーブルは、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項に記載の方法。

【0222】

3. 前記変換を行うことは、動き候補の前記セットの少なくとも一部を使用して、AMVP (Advanced Motion Vector Prediction) 候補リスト導出処理を行うことを含む、第1項に記載の方法。

10

【0223】

4. 前記AMVP候補リスト導出処理は、1つ以上のテーブルからの動き候補をチェックすることを含む、第3項に記載の方法。

【0224】

5. 前記変換を行うことは、動き候補をチェックすることを含み、前記チェックされた動き候補に関連付けられた動きベクトルは、前記第1の映像ブロックの前記動きベクトルを符号化するための動きベクトル予測因子として使用する、第1～4項のいずれか1項に記載の方法。

【0225】

6. チェックされた動き候補に関連付けられた動きベクトルは、前記AMVP動き候補リストに追加される、第4項に記載の方法。

20

【0226】

7. 前記変換を行うことは、規則に基づいて動き候補の少なくとも一部をチェックすることを含む、第1項に記載の方法。

【0227】

8. 前記規則は、TMVP (Temporal Motion Vector Prediction) 候補をチェックした後、AMVP候補リストが満杯でない場合、前記チェックを有効にする、第7項に記載の方法。

【0228】

9. 前記規則は、空間的近傍から選択し、ブルーニングした後、TMVP候補を挿入する前に、AMVP候補リストが満杯でない場合、前記チェックを有効にすることを有効にする、第7項に記載の方法。

30

【0229】

10. 前記規則は、i) 上側の近傍のブロックからのAMVP候補がスケール無しで存在しない場合、または、ii) 左側の近傍のブロックからのAMVP候補がスケール無しで存在しない場合に、前記チェックを有効にする、第7項に記載の方法。

【0230】

11. 規則は、AMVP候補リストに動き候補を追加する前に、ブルーニングを適用する場合、前記チェックを有効にする、第7項に記載の方法。

【0231】

12. 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、第1項に記載の方法。

40

【0232】

13. 現在の参照ピクチャと異なる参照ピクチャを有する動き候補をさらにチェックする、第12項に記載の方法。

【0233】

14. 同一の参照ピクチャを有する動き候補をチェックすることは、異なる参照ピクチャを有する動き候補をチェックすることに先立って実行される、第13項に記載の方法。

【0234】

15. 動き候補から動きベクトルをテーブルに追加する前に、ブルーニング工程を含む

50

AMVP候補リスト構成処理をさらに含む、第1項に記載の方法。

【0235】

16. 前記ブルーニング工程は、動き候補を、AMVP候補リストにおける利用可能な動き候補の少なくとも一部と比較することを含む、第15項に記載の方法。

【0236】

17. 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AMVP候補の関数である、第15項に記載の方法。

【0237】

18. 前記複数の工程は、AMVP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、前記ブルーニングが、K個のAMVP候補にのみ適用され、 $K < M$ であり、KおよびMは整数である、第17項に記載の方法。

10

【0238】

19. 前記変換を行うことは、いくつかの動きベクトルの差分を用いて、SMVD (Symmetric Motion Vector Difference) 処理を行うことを含む、第1項に記載の方法。

【0239】

20. 前記変換を行うことは、いくつかの動きベクトルを用いて、SMV (Symmetric Motion Vector) 処理を行うことを含む、第1項に記載の方法。

【0240】

21. 前記規則は、あるAMVP候補を挿入した後に、AMVP候補リストが満杯でない場合、前記チェックを有効とする、第7項に記載の方法。

20

【0241】

22. 前記テーブルにおける動き候補のチェックを有効とすることを更に有し、前記チェックは、空間的または時間的ブロックから導出した他の候補をチェックする前に有効にされ、他の候補は、AMVP候補、SMVD候補、SMV候補、またはアフィンインター候補を含む、第1項に記載の方法。

【0242】

23. テーブルにおける動き候補のチェックを有効とすることを更に有し、前記チェックは、前記テーブルに少なくとも1つの動き候補がある場合に有効になる、第1項に記載の方法。

30

【0243】

24. 双方向予測候補である動き候補の場合、第2の参照ピクチャリストの参照ピクチャがチェックされる前に、第1の参照ピクチャリストの参照ピクチャがチェックされ、前記第1の参照ピクチャリストは現在の参照対象ピクチャリストである、第1項に記載の方法。

【0244】

25. 双方向予測候補である動き候補の場合、第2の参照ピクチャリストの参照ピクチャがチェックされる前に、第1の参照ピクチャリストの参照ピクチャがチェックされ、前記第2の参照ピクチャリストは現在の参照対象ピクチャリストである、第1または2項に記載の方法。

40

【0245】

26. 第1の参照ピクチャリストの参照ピクチャは、第2の参照ピクチャリストの参照ピクチャの前にチェックされる、第1項に記載の方法。

【0246】

27. 前記変換を行うことは、前記第1の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項に記載の方法。

【0247】

28. 前記変換を行うことは、前記第1の映像ブロックをビットストリーム表現から生成することを含む、第1項に記載の方法。

【0248】

50

29．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少なくとも1つを含む動き情報に関連付けられる、第1～28項のいずれか1項に記載の方法。

【0249】

30．1つの動き候補は、イントラ符号化されたブロックに用いられるイントラ予測モードに関連付けられる、第1～29項のいずれか1項に記載の方法。

【0250】

31．1つの動き候補は、IC符号化されたブロックに用いられる複数のIC (Illumination Compensation) パラメータに関連付けられる、第1～29項のいずれか1項に記載の方法。

10

【0251】

32．1つの動き候補は、フィルタリング処理に用いられるフィルタパラメータに関連付けられる、第1～29項のいずれか1項に記載の方法。

【0252】

33．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1～29項のいずれか1項に記載の方法。

【0253】

34．1つ以上のテーブルを前記更新することは、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33項に記載の方法。

20

【0254】

35．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34項に記載の方法。

【0255】

36．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、前記命令は、前記プロセッサにより実行された際に、前記プロセッサに、第1項～35項のいずれか一項に記載の方法を実施させる装置。

【0256】

37．第1～35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、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。

30

【0257】

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。

【0258】

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む、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

40

50

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。

【0263】

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。

【0264】

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。

10

20

30

40

50

【 図 面 】
【 図 1 】

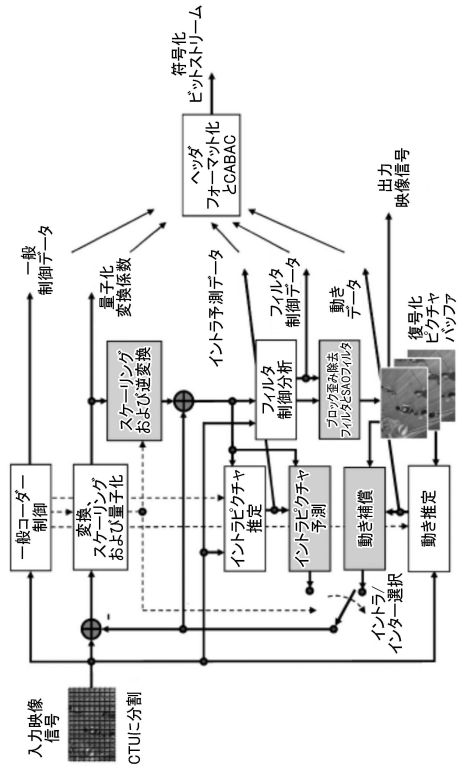
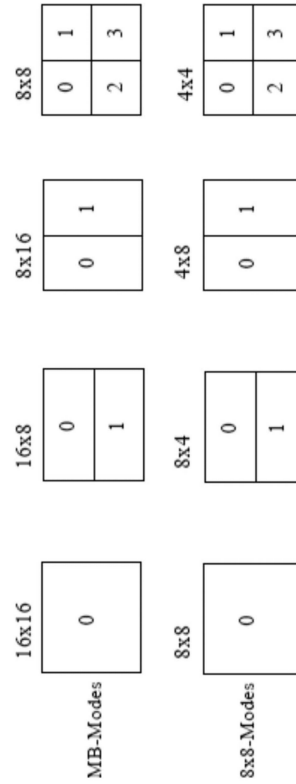


図1 典型的なHEVC映像エンコーダ (デコーダモデリング要素はライトグレーの影が付けられている)

【 図 2 】



10

20

【 図 3 】

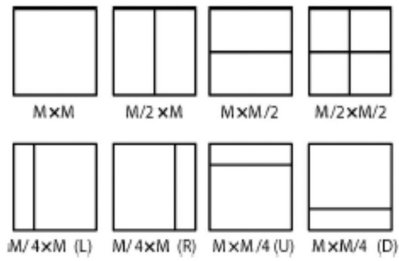
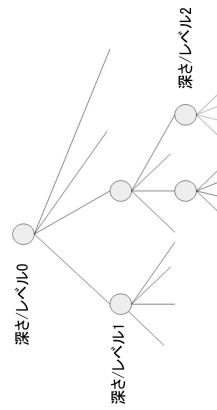


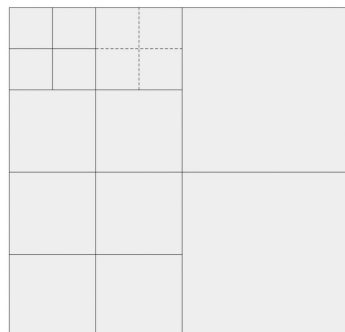
図3 特定のサイズ制限に従って、CBをPBに分割するためのモード。イントラピクチャ予測CBでは、M x MとM/2 x M/2だけがサポートされる

【 図 4 】



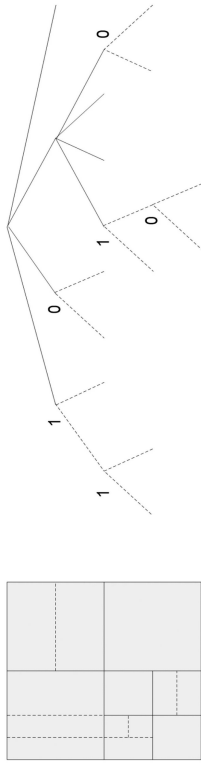
30

40

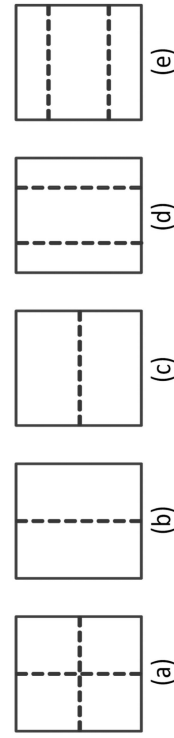


50

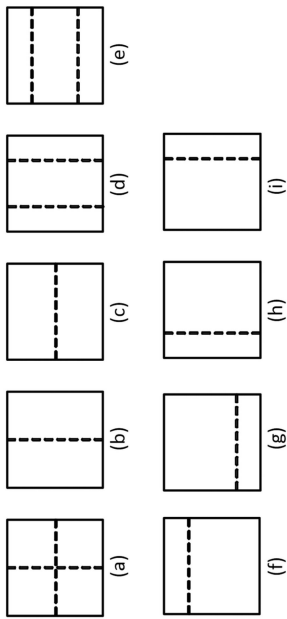
【 図 5 】



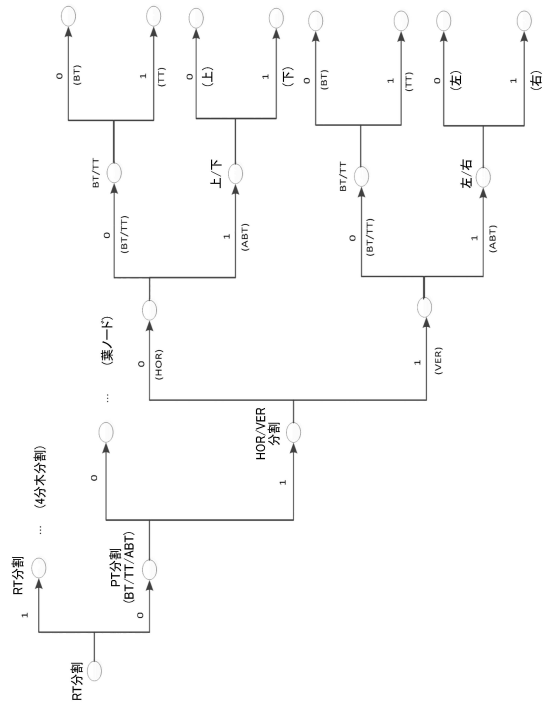
【 図 6 】



【 図 7 】



【 図 8 】



10

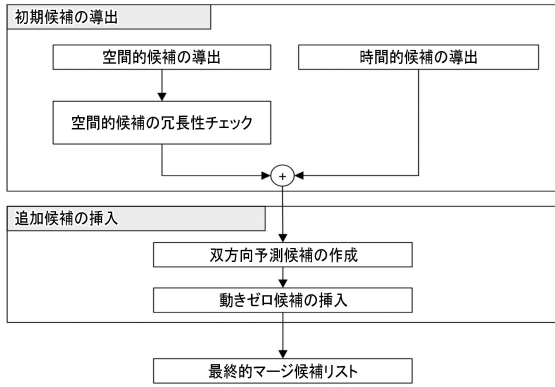
20

30

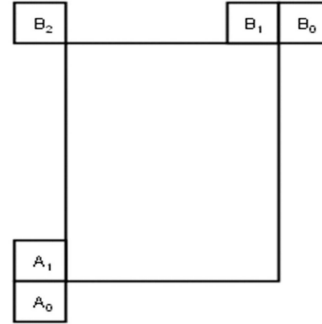
40

50

【図 9】

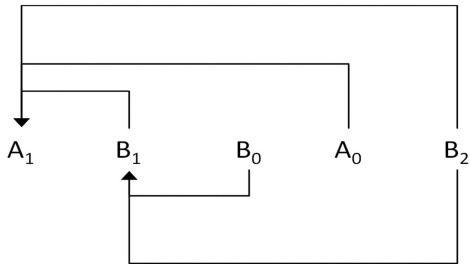


【図 10】

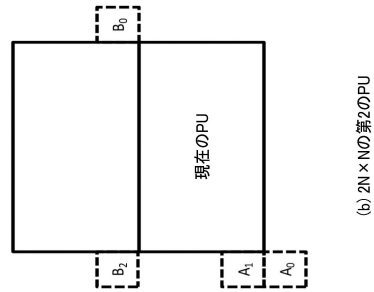


10

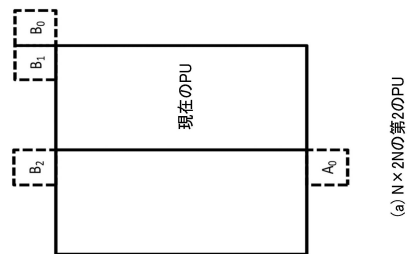
【図 11】



【図 12】



20

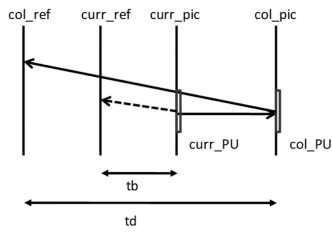


30

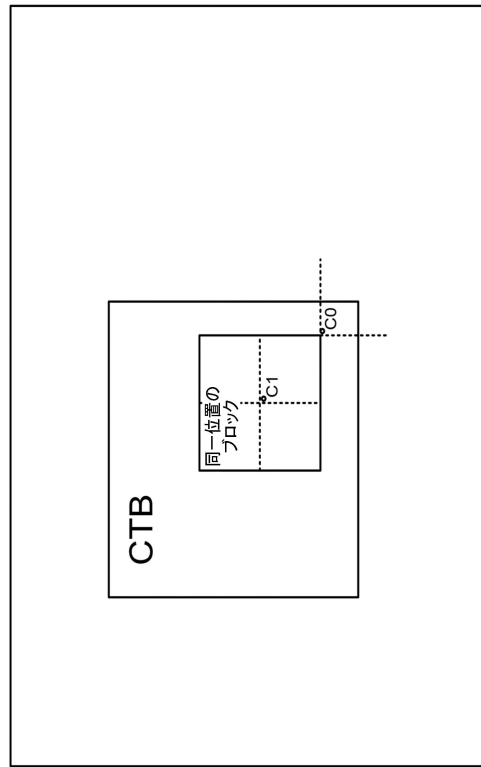
40

50

【 図 1 3 】



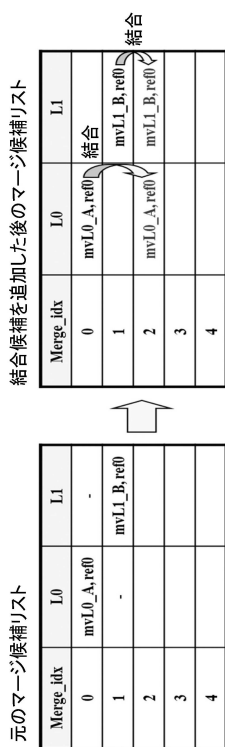
【 図 1 4 】



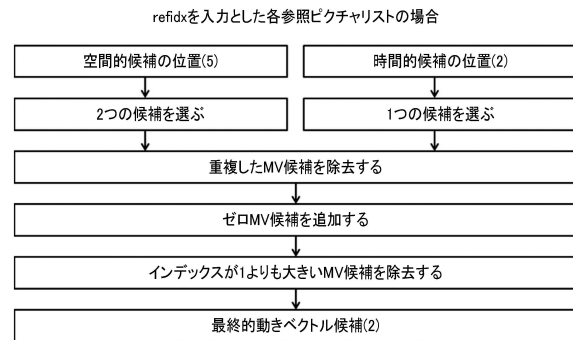
10

20

【 図 1 5 】



【 図 1 6 】

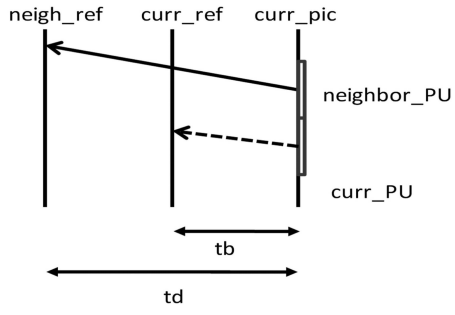


30

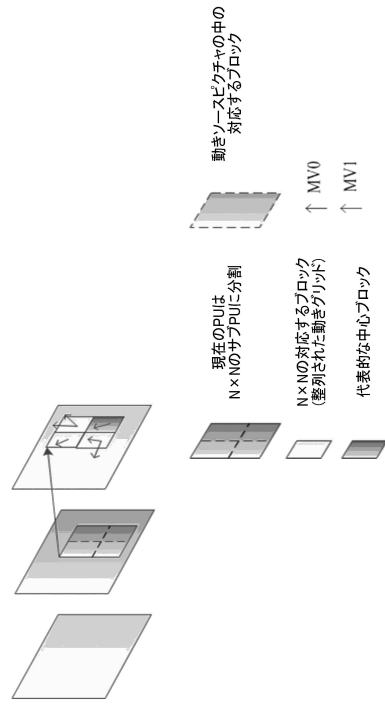
40

50

【図 17】



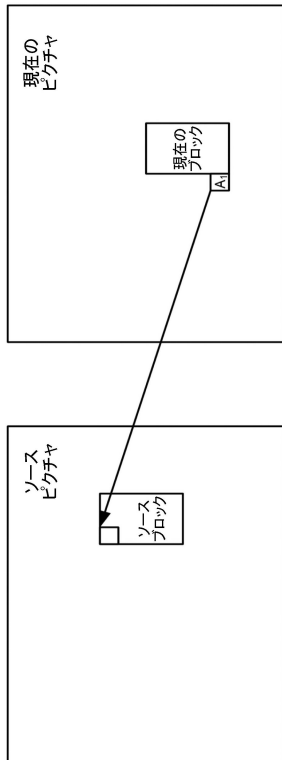
【図 18】



10

20

【図 19】



【図 20】

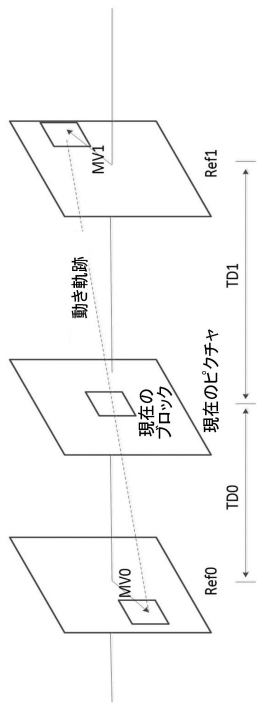


30

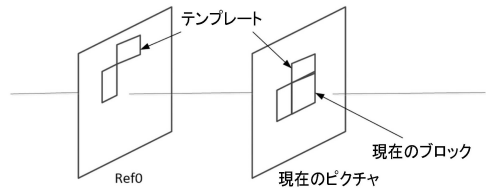
40

50

【図 2 1】



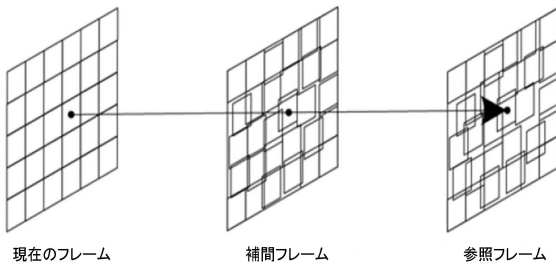
【図 2 2】



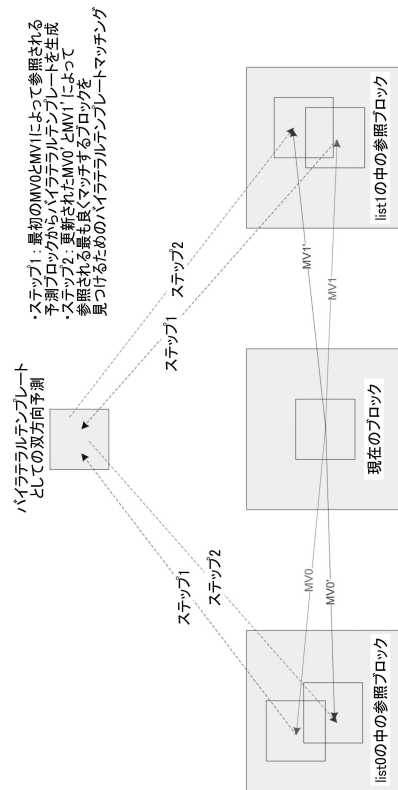
10

20

【図 2 3】



【図 2 4】



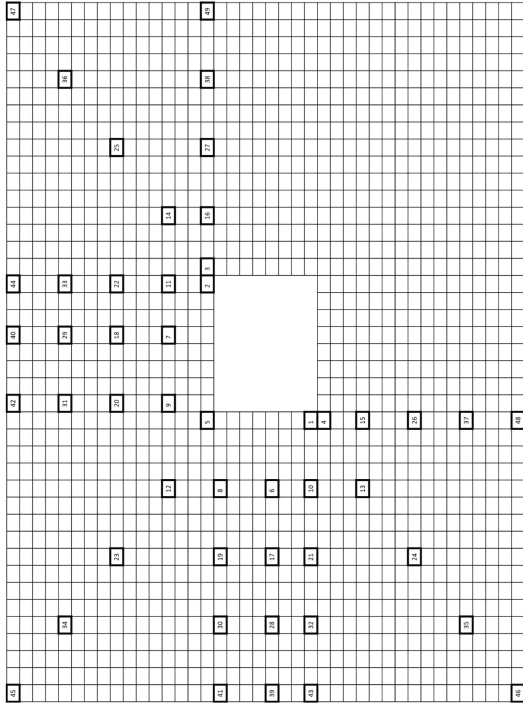
・ステップ1: 最初のMV0とMV1によって参照される予測ブロックからバイディレクショナルテンプレートを作成
 ・ステップ2: 更新されたMV0とMV1によって参照される最も良くマッチするブロックを見つけて、バイディレクショナルテンプレートマッチング

30

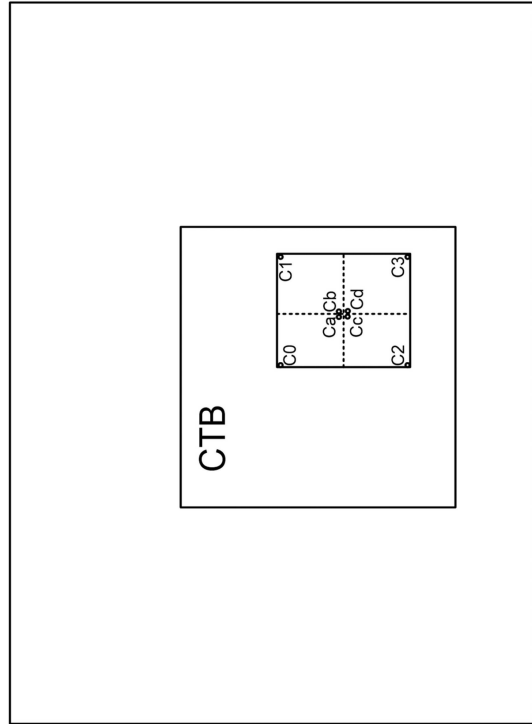
40

50

【図 2 5】



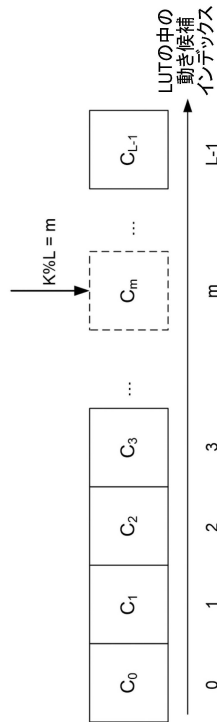
【図 2 6】



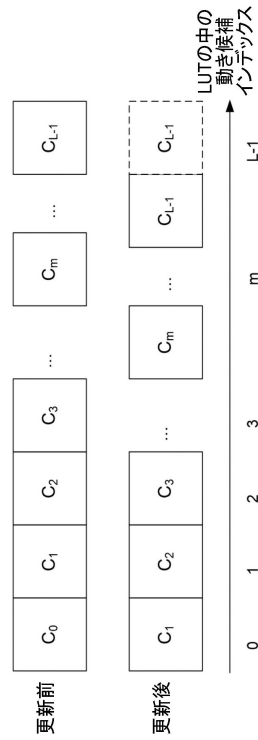
10

20

【図 2 7 A】



【図 2 7 B】

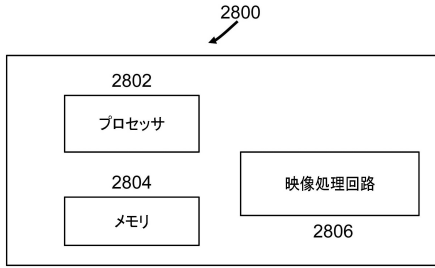


30

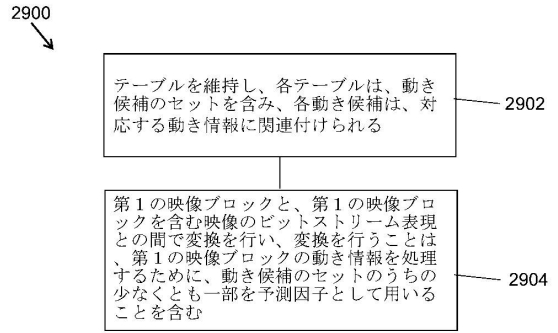
40

50

【図 28】

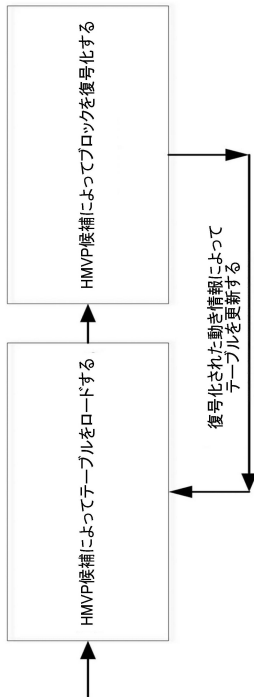


【図 29】

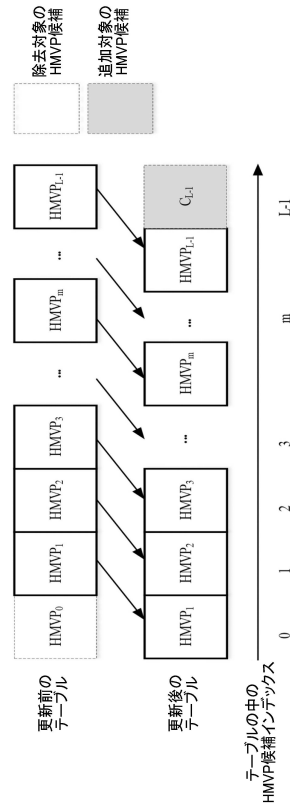


10

【図 30】



【図 31】



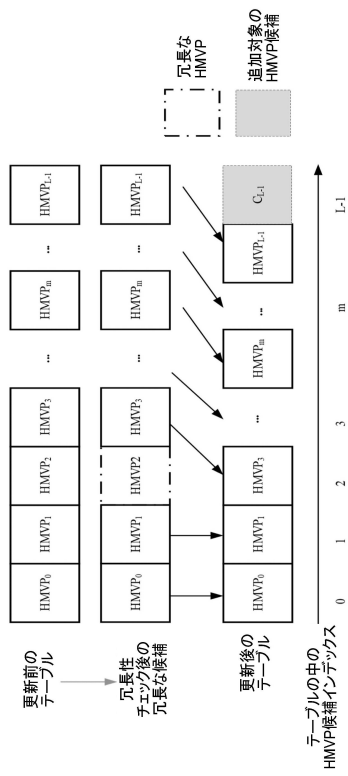
20

30

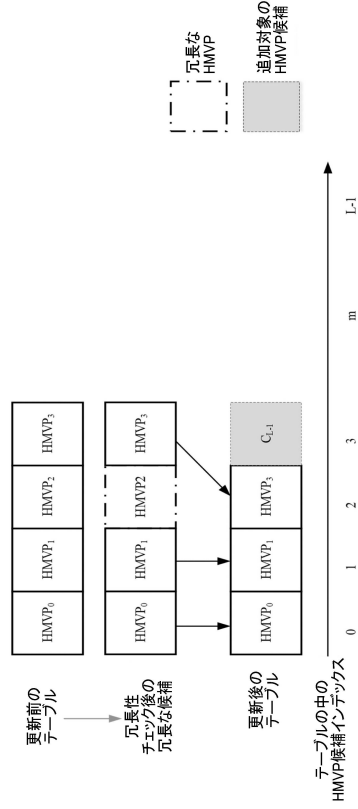
40

50

【図 3 2 A】



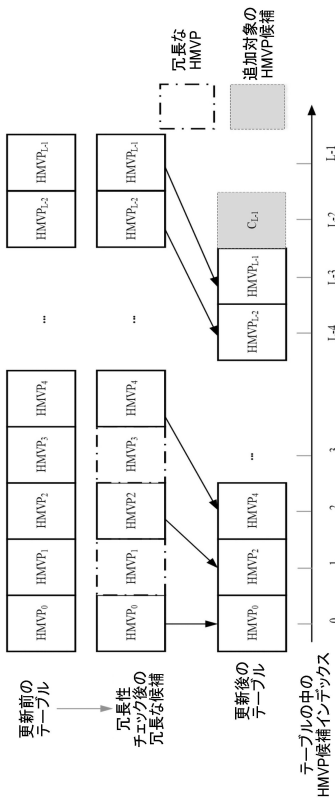
【図 3 2 B】



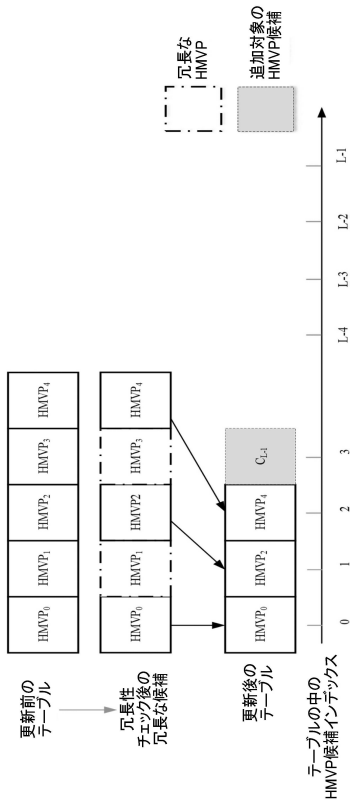
10

20

【図 3 3 A】



【図 3 3 B】

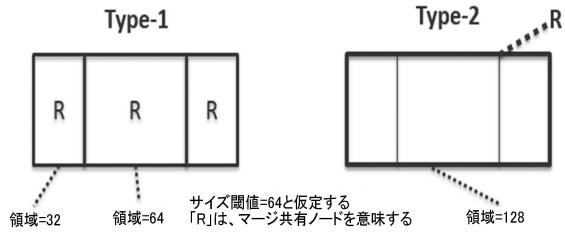


30

40

50

【 図 3 4 】



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/CN2019/072058

(32)優先日 平成31年1月16日(2019.1.16)

(33)優先権主張国・地域又は機関

中国(CN)

前置審査

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 フロアー, 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

弁理士法人栄光事務所

(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 베이징, 하이ディアнь ディストリクト, チーチュン 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 베이징, 하이ディアнь ディストリクト, チーチュン 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 間宮 嘉誉

(56)参考文献 国際公開第2018/231700(WO, A1)

米国特許出願公開第2011/0194609(US, A1)

国際公開第2015/100726(WO, A1)

米国特許出願公開第2012/0257678(US, A1)

国際公開第2017/197126(WO, A1)

国際公開第2012/095467(WO, A1)

(58)調査した分野 (Int.Cl., DB名)

H04N 7/12

H04N 19/00 - 19/98