

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7256290号
(P7256290)

(45)発行日 令和5年4月11日(2023.4.11)

(24)登録日 令和5年4月3日(2023.4.3)

(51)国際特許分類 F I
H 0 4 N 19/54 (2014.01) H 0 4 N 19/54
H 0 4 N 19/70 (2014.01) H 0 4 N 19/70

請求項の数 9 (全80頁)

(21)出願番号	特願2021-554683(P2021-554683)	(73)特許権者	502032105
(86)(22)出願日	令和2年3月10日(2020.3.10)		エルジー エレクトロニクス インコーポ レイティド
(65)公表番号	特表2022-524432(P2022-524432 A)		L G E L E C T R O N I C S I N C .
(43)公表日	令和4年5月2日(2022.5.2)		大韓民国, ソウル, ヨンドゥンポーク, ヨイーデロ, 1 2 8
(86)国際出願番号	PCT/KR2020/003323		1 2 8, Yeoui-daero, Y eongdeungpo-gu, 07
(87)国際公開番号	WO2020/184953		3 3 6 Seoul, Republic of Korea
(87)国際公開日	令和2年9月17日(2020.9.17)		
審査請求日	令和3年9月29日(2021.9.29)	(74)代理人	100099759
(31)優先権主張番号	62/817,513		弁理士 青木 篤
(32)優先日	平成31年3月12日(2019.3.12)	(74)代理人	100123582
(33)優先権主張国・地域又は機関	米国(US)		弁理士 三橋 真二
		(74)代理人	100165191

最終頁に続く

(54)【発明の名称】 双予測のための加重値インデックス情報を誘導するビデオ又は映像コーディング

(57)【特許請求の範囲】

【請求項1】

デコーディング装置によって行われる映像デコーディング方法において、
ビットストリームを介してインター予測モード情報を含む映像情報を受信するステップ
と、

前記インター予測モード情報に基づいて、現在ブロックのマージ候補リストを生成する
ステップと、

前記マージ候補リストに含まれた候補のうちの一つの候補を選択するステップと、
前記選択された候補に基づいて、前記現在ブロックの動き情報を導出するステップと、
前記導出された動き情報に基づいて、前記現在ブロックのL 0 予測サンプル及びL 1 予
測サンプルを生成するステップと、

前記L 0 予測サンプル、前記L 1 予測サンプル、及び加重値情報に基づいて前記現在ブ
ロックの予測サンプルを生成し、前記加重値情報は、前記選択された候補に対する加重値
インデックス情報に基づいて導出される、ステップと、を含み、

前記候補は、承継アフィン候補と構成されたアフィン候補を含み、前記構成されたアフ
ィン候補は、CPMV (Control Point Motion Vector) を含
み、

前記構成されたアフィン候補がCP0 (Control Point 0) に対するCPMV
を含むこと、及び双予測が前記現在ブロックに適用されることに基づいて、前記構成さ
れたアフィン候補に対する加重インデックス情報は、前記現在ブロックの前記CP0の周

辺ブロックのうち、特定のブロックの加重値インデックス情報に等しいように固定され、
前記C P 0は前記現在ブロックの左上側角に関連し、

前記特定のブロックは、前記C P 0に対する前記C P M Vの導出のために使用されるブロックである、映像デコーディング方法。

【請求項2】

前記周辺ブロックは、前記現在ブロックの左上側角の周辺ブロック、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック、及び前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含む、請求項1に記載の映像デコーディング方法。

【請求項3】

前記候補は、ペアワイズ (pair-wise) 候補を含み、

前記ペアワイズ候補は、前記候補のうち異なる2個の候補に基づいて導出され、

前記ペアワイズ候補に対する加重値インデックス情報は、前記2個の候補のうちの一つの加重値インデックス情報に基づいて導出される、請求項1に記載の映像デコーディング方法。

【請求項4】

前記候補は、サブブロックベーステンポラルマージ (subblock-based temporal merge) 候補を含み、

前記サブブロックベーステンポラルマージ候補に対する加重値インデックス情報は、前記現在ブロックの左側の周辺ブロックの加重値インデックス情報に基づいて導出される、請求項1に記載の映像デコーディング方法。

【請求項5】

前記候補は、サブブロックベーステンポラルマージ候補を含み、

前記サブブロックベーステンポラルマージ候補に対する加重値インデックス情報の値は、0として導出される、請求項1に記載の映像デコーディング方法。

【請求項6】

前記候補は、サブブロックベーステンポラルマージ (subblock-based temporal merge) 候補を含み、

前記サブブロックベーステンポラルマージ候補に対する加重値インデックス情報は、コル (col) ブロック内のセンタブロックの加重値インデックス情報に基づいて導出され、

前記コルブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含み、

前記センタブロックは、前記コルブロックの中央に位置する4個のサブブロックのうち右下側のサブブロックである、請求項1に記載の映像デコーディング方法。

【請求項7】

前記候補は、サブブロックベーステンポラルマージ (subblock-based temporal merge) 候補を含み、

前記サブブロックベーステンポラルマージ候補に対する加重値インデックス情報は、コル (col) ブロックのサブブロックそれぞれの加重値インデックス情報に基づいて導出され、

前記コルブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含む、請求項1に記載の映像デコーディング方法。

【請求項8】

エンコーディング装置によって行われる映像エンコーディング方法において、

現在ブロックのインター予測モードを決定し、前記インター予測モードを示すインター予測モード情報を生成するステップと、

前記インター予測モードに基づいて、前記現在ブロックのマージ候補リストを生成するステップと、

前記マージ候補リストに含まれた候補のうちの一つの候補を選択し、前記選択された候

10

20

30

40

50

補を示す選択情報を生成するステップと、

前記インター予測モード情報及び前記選択情報を含む映像情報をエンコーディングするステップと、を含み、

前記候補は、承継されたアフィン候補と構成されたアフィン候補を含み、前記構成されたアフィン候補は、CPMV (Control Point Motion Vector) を含み、

前記構成されたアフィン候補がCP0 (Control Point 0) に対するCPMVを含むこと、及び双予測が前記現在ブロックに適用されることに基づいて、前記構成されたアフィン候補に対する加重インデックス情報は、前記現在ブロックのCP0 (Control Point 0) の周辺ブロックのうち、特定のブロックの加重値インデックス情報に等しいように固定され、前記CP0は前記現在ブロックの左上側角に関連し、

前記特定のブロックは、前記CP0に対するCPMVの導出のために使用されるブロックである、映像エンコーディング方法。

【請求項9】

映像データに対する送信方法であって、

エンコードされた映像情報を取得するステップであって、前記エンコードされた映像情報は、現在ブロックのインター予測モードを決定し、前記インター予測モードを示すインター予測モード情報を生成し、前記インター予測モードに基づいて前記現在ブロックのマージ候補リストを生成し、前記マージ候補リストに含まれる候補の一つを選択し、前記選択された候補を示す選択情報を生成し、前記インター予測モード情報と前記選択情報を含む映像情報をエンコーディングすることを実行することにより生成される、ステップと、

前記エンコードされた映像情報に対する前記映像データを送信するステップを含み、

前記候補は、承継されたアフィン候補と構成されたアフィン候補を含み、前記構成されたアフィン候補は、CPMV (Control Point Motion Vector) を含み、

前記構成されたアフィン候補がCP0 (Control Point 0) に対するCPMVを含むこと、及び双予測が前記現在ブロックに適用されることに基づいて、前記構成されたアフィン候補に対する加重インデックス情報は、前記現在ブロックのCP0 (Control Point 0) の周辺ブロックのうち、特定ブロックの加重値インデックス情報に等しいように固定され、前記CP0は前記現在ブロックの左上側角に関連し、

前記特定ブロックは、前記CP0に対するCPMVの導出のために使用されるブロックである、映像データに対する送信方法。

【発明の詳細な説明】

【技術分野】

【0001】

本技術は、双予測のための加重値インデックス情報を誘導するビデオ又は映像コーディングに関する。

【背景技術】

【0002】

近年、4Kまたは8K以上のUHD (Ultra High Definition) 映像／ビデオのような高解像度、高品質の映像／ビデオに対する需要が様々な分野で増加している。映像／ビデオデータが高解像度、高品質になるほど、既存の映像／ビデオデータに比べて相対的に送信される情報量またはビット量が増加するので、既存の有無線広帯域回線のような媒体を利用して映像データを送信するか、既存の格納媒体を利用して映像／ビデオデータを格納する場合、送信費用と格納費用が増加される。

【0003】

また、近年、VR (Virtual Reality)、AR (Artificial Reality) コンテンツやホログラムなどの実感メディア (Immersive Media) に対する関心及び需要が増加しており、ゲーム映像のように、現実映像と異なる映像特性を有する映像／ビデオに対する放送が増加している。

【0004】

これにより、上記のような様々な特性を有する高解像度・高品質の映像／ビデオの情報を効果的に圧縮して送信するか、格納し、再生するために高効率の映像／ビデオ圧縮技術が求められる。

【発明の概要】

【課題を解決するための手段】

【0005】

本文書の一実施例によれば、映像／ビデオコーディングの効率を高める方法及び装置を提供する。

【0006】

本文書の一実施例によれば、映像コーディングの際、加重値ベースの双予測を用いる方法及び装置を提供する。

【0007】

本文書の一実施例によれば、インター予測で双予測のための加重値インデックス情報を誘導する方法及び装置を提供する。

【0008】

本文書の一実施例によれば、双予測の際、マージ候補リスト又はアフィンマージ候補リスト内の候補のための加重値インデックス情報を誘導する方法及び装置を提供する。

【0009】

本文書の一実施例によれば、デコーディング装置により行われるビデオ／映像デコーディング方法を提供する。

【0010】

本文書の一実施例によれば、ビデオ／映像デコーディングを行うデコーディング装置を提供する。

【0011】

本文書の一実施例によれば、エンコーディング装置により行われるビデオ／映像エンコーディング方法を提供する。

【0012】

本文書の一実施例によれば、ビデオ／映像エンコーディングを行うエンコーディング装置を提供する。

【0013】

本文書の一実施例によれば、本文書の実施例の少なくとも一つに開示されたビデオ／映像エンコーディング方法によって生成されたエンコーディングされたビデオ／映像情報が格納されたコンピュータ読み取り可能なデジタル格納媒体を提供する。

【0014】

本文書の一実施例によれば、デコーディング装置により本文書の実施例の少なくとも一つに開示されたビデオ／映像デコーディング方法を行うように引き起こすエンコーディングされた情報又はエンコーディングされたビデオ／映像情報が格納されたコンピュータ読み取り可能なデジタル格納媒体を提供する。

【発明の効果】

【0015】

本文書の一実施例によれば、全般的な映像／ビデオ圧縮の効率を高めることができる。

【0016】

本文書の一実施例によれば、インター予測の際に動きベクトルの候補を効率的に構成することができる。

【0017】

本文書の一実施例によれば、効率的に加重値ベースの双予測を行うことができる。

【図面の簡単な説明】

【0018】

【図1】本文書の実施例が適用できるビデオ／映像コーディングシステムの例を概略的に

10

20

30

40

50

示す。

【図 2】本文書の実施例が適用できるビデオ／映像エンコーディング装置の構成を概略的に説明する図である。

【図 3】本文書の実施例が適用できるビデオ／映像デコーディング装置の構成を概略的に説明する図である。

【図 4】インター予測でマージモードを説明するための図である。

【図 5 A】アフィン動き予測のための C P M V を例示的に示す。

【図 5 B】アフィン動き予測のための C P M V を例示的に示す。

【図 6】アフィン M V F がサブブロックの単位で決定される場合を例示的に示す。

【図 7】インター予測でアフィンマージモードを説明するための図である。

【図 8】アフィンマージモードで候補の位置を説明するための図である。

【図 9】インター予測で S b T M V P を説明するための図である。

【図 10】本文書の実施例に係るビデオ／映像エンコーディング方法、及び関連のコンポーネントの一例を概略的に示す。

【図 11】本文書の実施例に係るビデオ／映像エンコーディング方法、及び関連のコンポーネントの一例を概略的に示す。

【図 12】本文書の実施例に係る映像／ビデオデコーディング方法、及び関連のコンポーネントの一例を概略的に示す。

【図 13】本文書の実施例に係る映像／ビデオデコーディング方法、及び関連のコンポーネントの一例を概略的に示す。

【図 14】本文書で開示された実施例が適用できるコンテンツストリーミングシステムの例を示す。

【発明を実施するための形態】

【0019】

本文書の開示は、様々な変更を加えることができ、様々な実施例を有することができるが、特定の実施例を図面に例示し、詳細に説明しようとする。しかし、これは、本開示を特定の実施例に限定しようとするわけではない。本文書で使用する用語は、単に特定の実施例を説明するために使用されたものであって、本文書における実施例の技術的思想を限定しようとする意図に使用されるものではない。単数の表現は、文脈上明らかに異なる意味ではない限り、複数の表現を含む。本文書において、「含む」又は「有する」等の用語は、文書上に記載された特徴、数字、段階、動作、構成要素、部品又はこれらを組み合わせたものが存在することを指定しようとするものであり、1つ又はそれ以上の異なる特徴や、数字、段階、動作、構成要素、部品又はこれらを組み合わせたものの存在又は付加の可能性を予め排除しないものと理解されるべきである。

【0020】

一方、本文書で説明される図面上の各構成は、互いに異なる特徴的な機能に関する説明の便宜のために独立して示すものであって、各構成が互いに別個のハードウェアや別個のソフトウェアで具現されるということの意味するのではない。例えば、各構成のうち、2つ以上の構成を合わせて1つの構成をなすこともあり、1つの構成が複数の構成に分けられることもある。各構成が統合及び／又は分離された実施例も、本文書の開示範囲に含まれる。

【0021】

以下、添付図を参照として、本文書の実施例を説明しようとする。以下、図面上の同じ構成要素に対しては同じ参照符号を使用し得、同じ構成要素に対して重複した説明は省略され得る。

【0022】

図 1 は、本文書の実施例が適用できるビデオ／映像コーディングシステムの例を概略的に示す。

【0023】

図 1 に示すように、ビデオ／映像コーディングシステムは、第 1 の装置（ソースデバイ

10

20

30

40

50

ス) 及び第2の装置(受信デバイス)を備えることができる。ソースデバイスは、エンコーディングされたビデオ(v i d e o) / 映像(i m a g e) 情報またはデータをファイルまたはストリーミング形態でデジタル格納媒体またはネットワークを介して受信デバイスに伝達することができる。

【0024】

前記ソースデバイスは、ビデオソース、エンコーディング装置、送信部を備えることができる。前記受信デバイスは、受信部、デコーディング装置、及びレンダラを備えることができる。前記エンコーディング装置は、ビデオ / 映像エンコーディング装置と呼ばれることができ、前記デコーディング装置は、ビデオ / 映像デコーディング装置と呼ばれることができる。送信機は、エンコーディング装置に備えられることができる。受信機は、デコーディング装置に備えられることができる。レンダラは、ディスプレイ部を備えることができ、ディスプレイ部は、別個のデバイスまたは外部コンポーネントで構成されることもできる。

10

【0025】

ビデオソースは、ビデオ / 映像のキャプチャ、合成、または生成過程などを介してビデオ / 映像を取得できる。ビデオソースは、ビデオ / 映像キャプチャデバイス及び / 又はビデオ / 映像生成デバイスを含むことができる。ビデオ / 映像キャプチャデバイスは、例えば、1つ以上のカメラ、以前にキャプチャされたビデオ / 映像を含むビデオ / 映像アーカイブなどを含むことができる。ビデオ / 映像生成デバイスは、例えば、コンピュータ、タブレット、及びスマートフォンなどを含むことができ、(電子的に) ビデオ / 映像を生成できる。例えば、コンピュータなどを介して仮想のビデオ / 映像が生成され得るし、この場合、関連データが生成される過程にビデオ / 映像キャプチャ過程が代替されることができる。

20

【0026】

エンコーディング装置は、入力ビデオ / 映像をエンコーディングすることができる。エンコーディング装置は、圧縮及びコーディング効率のために、予測、変換、量子化など、一連の手順を行うことができる。エンコーディングされたデータ(エンコーディングされたビデオ / 映像情報) は、ビットストリーム(b i t s t r e a m) 形態で出力されることができる。

【0027】

送信部は、ビットストリーム形態で出力されたエンコーディングされたビデオ / 映像情報またはデータをファイルまたはストリーミング形態でデジタル格納媒体またはネットワークを介して受信デバイスの受信部に伝達することができる。デジタル格納媒体は、USB、SD、CD、DVD、ブルーレイ、HDD、SSDなど、様々な格納媒体を含むことができる。送信部は、予め決められたファイルフォーマットを介してメディアファイルを生成するためのエレメントを含むことができ、放送 / 通信ネットワークを介しての送信のためのエレメントを含むことができる。受信部は、前記ビットストリームを受信 / 抽出してデコーディング装置に伝達することができる。

30

【0028】

デコーディング装置は、エンコーディング装置の動作に対応する逆量子化、逆変換、予測など、一連の手順を行ってビデオ / 映像をデコーディングすることができる。

40

【0029】

レンダラは、デコーディングされたビデオ / 映像をレンダリングすることができる。レンダリングされたビデオ / 映像は、ディスプレイ部を介してディスプレイされることができる。

【0030】

本文書は、ビデオ / 映像コーディングに関する。例えば、本文書で開示された方法 / 実施例は、VVC(v e r s a t i l e v i d e o c o d i n g) 標準に開示される方法に適用されることができる。また、本文書で開示された方法 / 実施例は、EVC(e s s e n t i a l v i d e o c o d i n g) 標準、AV1(AOMedia Video

50

1) 標準、AVS2 (2nd generation of audio video coding standard) 又は次世代ビデオ／映像コーディング標準 (ex. 267 又は H. 268 等) に開示される方法に適用されることができる。

【0031】

本文書では、ビデオ／映像コーディングに関する多様な実施例を提示し、別の言及がない限り、前記実施例は互いに組み合わせて行うこともある。

【0032】

本文書で、ビデオ (video) は、時間の流れによる一連の映像 (image) の集合を意味することができる。ピクチャ (picture) は、一般的に特定の時間帯の1つの映像を示す単位を意味し、スライス (slice) ／タイル (tile) は、コーディングにおいてピクチャの一部を構成する単位である。スライス／タイルは、1つ以上のCTU (coding tree unit) を含むことができる。1つのピクチャは、1つ以上のスライス／タイルで構成されることができる。タイルは、ピクチャ内の特定のタイル列及び特定のタイル列以内のCTUの四角領域である (A tile is a rectangular region of CTUs within a particular tile column and a particular tile row in a picture)。前記タイル列は、CTUの四角領域であり、前記四角領域は、前記ピクチャの高さと同じ高さを有し、幅はピクチャパラメータセット内のシンタックス要素により明示され得る (The tile column is a rectangular region of CTUs having a height equal to the height of the picture and a width specified by syntax elements in the picture parameter set)。前記タイル行はCTUの四角領域であり、前記四角領域はピクチャパラメータセット内のシンタックス要素によって明示される幅を有し、高さは前記ピクチャの高さと同じであり得る (The tile row is a rectangular region of CTUs having a height specified by syntax elements in the picture parameter set and a width equal to the width of the picture)。タイルのスキャンはピクチャをパーティショニングするCTUの特定の順次オーダリングを示し得、前記CTUはタイル内のCTUラスタースキャンに連続的に整列され得、ピクチャ内のタイルは、前記ピクチャの前記タイルのラスタースキャンに連続的に整列され得る (A tile scan is a specific sequential ordering of CTUs partitioning a picture in which the CTUs are ordered consecutively in CTU raster scan in a tile whereas tiles in a picture are ordered consecutively in a raster scan of the tiles of the picture)。スライスは単一のNALユニットに排他的に含まれ得る、整数個の完全なタイル又はピクチャのタイル内の整数個の連続的な完全なCTU行を含み得る (A slice includes an integer number of complete tiles or an integer number of consecutive complete CTU rows within a tile of a picture that may be exclusively contained in a single NAL unit)

【0033】

一方、1つのピクチャは、2つ以上のサブピクチャに区分されることができる。サブピクチャは、ピクチャ内の1つ以上のスライスの四角領域であり得る (a rectangular region of one or more slices within a picture)。

【0034】

10

20

30

40

50

ピクセル (pixel) またはペル (pel) は、1つのピクチャ (または、映像) を構成する最小の単位を意味することができる。また、ピクセルに対応する用語として、「サンプル (sample)」が使用され得る。サンプルは、一般的にピクセルまたはピクセルの値を示すことができ、ルマ (luma) 成分のピクセル／ピクセル値のみを示すことができ、クロマ (chroma) 成分のピクセル／ピクセル値のみを示すこともできる。
【0035】

ユニット (unit) は、映像処理の基本単位を示すことができる。ユニットは、ピクチャの特定領域及び当該領域に関連した情報のうち、少なくとも1つを含むことができる。1つのユニットは、1つのルマブロック及び2つのクロマ (例えば、cb、cr) ブロックを含むことができる。ユニットは、場合によって、ブロック (block) または領域 (area) などの用語と混用して使用されることができる。一般的な場合、M×Nブロックは、M個の列とN個の行とからなるサンプル (または、サンプルアレイ) または変換係数 (transform coefficient) の集合 (または、アレイ) を含むことができる。

10

【0036】

本文書において、「A又はB (A or B)」は、「ただA」、「ただB」、又は「A及びB両方」を意味し得る。言い換えると、本文書において、「A又はB (A or B)」は、「A及び／又はB (A and/or B)」と解釈され得る。例えば、本文書において、「A、B又はC (A, B or C)」は、「ただA」、「ただB」、「ただC」、又は「A、B及びCの任意の全ての組み合わせ (any combination of A, B and C)」を意味し得る。

20

【0037】

本文書で使用されるスラッシュ (／) や休止符 (comma) は、「及び／又は (and/or)」を意味し得る。例えば、「A／B」は、「A及び／又はB」を意味し得る。これによって、「A／B」は、「ただA」、「ただB」、又は「A及びB両方」を意味し得る。例えば、「A、B、C」は、「A、B又はC」を意味し得る。

【0038】

本文書において、「少なくとも一つのA及びB (at least one of A and B)」は、「ただA」、「ただB」又は「A及びB両方」を意味し得る。また、本文書において、「少なくとも一つのA又はB (at least one of A or B)」や「少なくとも一つのA及び／又はB (at least one of A and/or B)」という表現は、「少なくとも一つのA及びB (at least one of A and B)」と同様に解釈され得る。

30

【0039】

また、本文書において、「少なくとも一つのA、B及びC (at least one of A, B and C)」は、「ただA」、「ただB」、「ただC」、又は「A、B及びCの任意の全ての組み合わせ (any combination of A, B and C)」を意味し得る。また、「少なくとも一つのA、B又はC (at least one of A, B or C)」や「少なくとも一つのA、B及び／又はC (at least one of A, B and/or C)」は、「少なくとも一つのA、B及びC (at least one of A, B and C)」を意味し得る。

40

【0040】

また、本文書で使用される括弧は、「例えば (for example)」を意味し得る。具体的に、「予測 (イントラ予測)」と表示された場合、「予測」の一例として「イントラ予測」が提案されたものであり得る。言い換えると、本文書の「予測」は、「イントラ予測」に制限 (limit) されず、「イントラ予測」が「予測」の一例として提案されたものであり得る。また、「予測 (すなわち、イントラ予測)」と表示された場合にも、「予測」の一例として「イントラ予測」が提案されたものであり得る。

【0041】

本文書において一つの図面内で個別的に説明される技術的特徴は、個別的に具現されて

50

もよく、同時に具現されてもよい。

【0042】

図2は、本文書の実施例が適用できるビデオ／映像エンコーディング装置の構成を概略的に説明する図面である。以下、エンコーディング装置というのは、映像エンコーディング装置及び／又はビデオエンコーディング装置を含むことができる。

【0043】

図2に示すように、エンコーディング装置200は、映像分割部(image partitioner)210、予測部(predictor)220、レジデュアル処理部(residual processor)230、エントロピーエンコーディング部(entropy encoder)240、加算部(adder)250、フィルタリング部(filter)260、及びメモリ(memory)270を備えて構成されることができる。予測部220は、インター予測部221及びイントラ予測部222を備えることができる。レジデュアル処理部230は、変換部(transformer)232、量子化部(quantizer)233、逆量子化部(dequantizer)234、逆変換部(inverse transformer)235を備えることができる。レジデュアル処理部230は、減算部(subtractor、231)をさらに備えることができる。加算部250は、復元部(reconstructor)または復元ブロック生成部(reconstructed block generator)と呼ばれることができる。上述した映像分割部210、予測部220、レジデュアル処理部230、エントロピーエンコーディング部240、加算部250、及びフィルタリング部260は、実施形態によって1つ以上のハードウェアコンポーネント(例えば、エンコーダチップセットまたはプロセッサ)によって構成されることができる。また、メモリ270は、DPB(decoded picture buffer)を備えることができ、デジタル格納媒体によって構成されることもできる。前記ハードウェアコンポーネントは、メモリ270を内／外部コンポーネントとしてさらに備えることもできる。

【0044】

映像分割部210は、エンコーディング装置200に入力された入力映像(または、ピクチャ、フレーム)を1つ以上の処理ユニット(processing unit)に分割することができる。一例として、前記処理ユニットは、コーディングユニット(coding unit、CU)と呼ばれることができる。この場合、コーディングユニットは、コーディングツリーユニット(coding tree unit、CTU)または最大コーディングユニット(largest coding unit、LCU)からQTBT(T Quad-tree binary-tree ternary-tree)構造によって再帰的に(recursively)分割されることができる。例えば、1つのコーディングユニットは、クアッドツリー構造、バイナリツリー構造、及び／又はターナリ構造に基づいて下位(deeper)デプスの複数のコーディングユニットに分割されることができる。この場合、例えば、クアッドツリー構造が先に適用され、バイナリツリー構造及び／又はターナリ構造がその後に適用されることができる。または、バイナリツリー構造が先に適用されることがもできる。それ以上分割されない最終コーディングユニットに基づいて本開示に係るコーディング手順が行われ得る。この場合、映像特性によるコーディング効率などに基づいて、最大コーディングユニットが最終コーディングユニットとして使用されることができ、または、必要に応じてコーディングユニットは、再帰的に(recursively)もっと下位デプスのコーディングユニットに分割されて最適のサイズのコーディングユニットが最終コーディングユニットとして使用されることができる。ここで、コーディング手順とは、後述する予測、変換、及び復元などの手順を含むことができる。他の例として、前記処理ユニットは、予測ユニット(PU: Prediction Unit)または変換ユニット(TU: Transform Unit)をさらに備えることができる。この場合、前記予測ユニット及び前記変換ユニットは、各々上述した最終コーディングユニットから分割またはパーティショニングされることができる。前記予測ユニットは、サンプル予測の単位であることができ、前記変換ユニットは、変換係

10

20

30

40

50

数を導く単位及び／又は変換係数からレジデュアル信号 (residual signal) を導く単位であることができる。

【0045】

ユニットは、場合によって、ブロック (block) または領域 (area) などの用語と混用して使用されることができる。一般的な場合、 $M \times N$ ブロックは、 M 個の列と N 個の行からなるサンプルまたは変換係数 (transform coefficient) 等の集合を示すことができる。サンプルは、一般的にピクセルまたはピクセルの値を示すことができ、輝度 (luma) 成分のピクセル／ピクセル値のみを示すことができ、彩度 (chroma) 成分のピクセル／ピクセル値のみを示すこともできる。サンプルは、1つのピクチャ (または、映像) をピクセル (pixel) またはペル (pel) に対応する用語として使用することができる。

10

【0046】

エンコーディング装置 200 は、入力映像信号 (原本ブロック、原本サンプルアレイ) でインター予測部 221 またはイントラ予測部 222 から出力された予測信号 (予測されたブロック、予測サンプルアレイ) を減算してレジデュアル信号 (residual signal、残余ブロック、残余サンプルアレイ) を生成でき、生成されたレジデュアル信号は、変換部 232 に送信される。この場合、図示されたように、エンコーダ 200 内において入力映像信号 (原本ブロック、原本サンプルアレイ) で予測信号 (予測ブロック、予測サンプルアレイ) を減算するユニットは、減算部 231 と呼ばれることができる。予測部は、処理対象ブロック (以下、現在ブロックという) に対する予測を行い、前記現在ブロックに対する予測サンプルを含む予測されたブロック (predicted block) を生成できる。予測部は、現在ブロックまたはCU単位でイントラ予測が適用されるか、またはインター予測が適用されるか決定することができる。予測部は、各予測モードについての説明で後述するように、予測モード情報など、予測に関する様々な情報を生成してエントロピーエンコーディング部 240 に伝達することができる。予測に関する情報は、エントロピーエンコーディング部 240 でエンコーディングされてビットストリーム形態で出力されることができる。

20

【0047】

イントラ予測部 222 は、現在ピクチャ内のサンプルを参照して現在ブロックを予測できる。前記参照されるサンプルは、予測モードによって前記現在ブロックの周辺 (neighbor) に位置することができ、または、離れて位置することもできる。イントラ予測で予測モードは、複数の非方向性モードと複数の方向性モードとを含むことができる。非方向性モードは、例えば、DCモード及びプラナーモード (Planarモード) を含むことができる。方向性モードは、予測方向の細かい程度によって、例えば、33個の方向性予測モードまたは65個の方向性予測モードを含むことができる。ただし、これは、例示であり、設定によってそれ以上またはそれ以下の個数の方向性予測モードが使用され得る。イントラ予測部 222 は、周辺ブロックに適用された予測モードを用いて、現在ブロックに適用される予測モードを決定することもできる。

30

【0048】

インター予測部 221 は、参照ピクチャ上で動きベクトルにより特定される参照ブロック (参照サンプルアレイ) に基づいて、現在ブロックに対する予測されたブロックを導くことができる。このとき、インター予測モードで送信される動き情報の量を減らすために、周辺ブロックと現在ブロックとの間の動き情報の相関性に基づいて、動き情報をブロック、サブブロック、またはサンプル単位で予測することができる。前記動き情報は、動きベクトル及び参照ピクチャインデックスを含むことができる。前記動き情報は、インター予測方向 (L0予測、L1予測、Bi予測等) 情報をさらに含むことができる。インター予測の場合に、周辺ブロックは、現在ピクチャ内に存在する空間的周辺ブロック (spatial neighboring block) と参照ピクチャに存在する時間的周辺ブロック (temporal neighboring block) とを含むことができる。前記参照ブロックを含む参照ピクチャと前記時間的周辺ブロックを含む参照ピクチャと

40

50

は同じであることができ、異なることもできる。前記時間的周辺ブロックは、同一位置参照ブロック (collocated reference block)、同一位置CU (col CU) などの名前と呼ばれることができ、前記時間的周辺ブロックを含む参照ピクチャは、同一位置ピクチャ (collocated picture、colPic) と呼ばれることもできる。例えば、インター予測部221は、周辺ブロックに基づいて動き情報候補リストを構成し、前記現在ブロックの動きベクトル及び／又は参照ピクチャインデックスを導出するために、どの候補が使用されるかを指示する情報を生成できる。様々な予測モードに基づいてインター予測が行われ得るし、例えば、スキップモードとマージモードとの場合に、インター予測部221は、周辺ブロックの動き情報を現在ブロックの動き情報として利用することができる。スキップモードの場合、マージモードとは異なり、レジデュアル信号が送信されないことがある。動き情報予測 (motion vector prediction、MVP) モードの場合、周辺ブロックの動きベクトルを動きベクトル予測子 (motion vector predictor) として用い、動きベクトル差分 (motion vector difference) をシグナリングすることにより、現在ブロックの動きベクトルを指示することができる。

【0049】

予測部220は、後述する様々な予測方法に基づいて予測信号を生成できる。例えば、予測部は、1つのブロックに対する予測のために、イントラ予測またはインター予測を適用できるだけでなく、イントラ予測とインター予測とを同時に適用することができる。これは、combined inter and intraprediction (CIIP) と呼ばれることができる。また、予測部は、ブロックに対する予測のために、イントラブロックコピー (intra block copy、IBC) 予測モードに基づくことができ、またはパレットモード (palette mode) に基づくこともできる。前記IBC予測モードまたはパレットモードは、例えば、SCC (screen content coding) などのように、ゲームなどのコンテンツ映像／動画コーディングのために使用されることができる。IBCは、基本的に現在ピクチャ内で予測を行うが、現在ピクチャ内で参照ブロックを導出する点においてインター予測と同様に行われることができる。すなわち、IBCは、本文書において説明されるインター予測技法のうち、少なくとも1つを用いることができる。パレットモードは、イントラコーディングまたはイントラ予測の一例と見なすことができる。パレットモードが適用される場合、パレットテーブル及びパレットインデックスに関する情報に基づいてピクチャ内のサンプル値をシグナリングすることができる。

【0050】

前記予測部 (インター予測部221及び／又は前記イントラ予測部222を含む) を介して生成された予測信号は、復元信号を生成するために用いられるか、レジデュアル信号を生成するために用いられることができる。変換部232は、レジデュアル信号に変換技法を適用して、変換係数 (transform coefficients) を生成することができる。例えば、変換技法は、DCT (Discrete Cosine Transform)、DST (Discrete Sine Transform)、GBT (Graph-Based Transform)、又はCNT (Conditionally Non-linear Transform) の少なくとも一つを含むことができる。ここで、GBTは、ピクセル間の関係情報をグラフで表現するという際に、このグラフから得られた変換を意味する。CNTは、以前に復元された全てのピクセル (all previously reconstructed pixel) を用いて予測信号を生成し、それに基づいて獲得される変換を意味する。また、変換過程は正方形の同じサイズを有するピクセルブロックに適用されてもよく、正方形ではない可変サイズのブロックに適用されてもよい。

【0051】

量子化部233は、変換係数を量子化してエントロピーエンコーディング部240に送信され、エントロピーエンコーディング部240は、量子化された信号 (量子化された変

10

20

30

40

50

換係数に関する情報)をエンコーディングしてビットストリームに出力することができる。前記量子化された変換係数に関する情報は、レジデュアル情報と呼ばれることができる。量子化部233は、係数スキャン順序(scan order)に基づいてブロック形態の量子化された変換係数を1次元ベクトル形態で再整列することができ、前記1次元ベクトル形態の量子化された変換係数に基づいて前記量子化された変換係数に関する情報を生成することもできる。エントロピーエンコーディング部240は、例えば、指数ゴロム(exponential Golomb)、CAVLC(context-adaptive variable length coding)、CABAC(context-adaptive binary arithmetic coding)などのような様々なエンコーディング方法を行うことができる。エントロピーエンコーディング部240は、量子化された変換係数の他に、ビデオ／イメージ復元に必要な情報(例えば、シンタックス要素(syntax elements)の値等)を共にまたは別にエンコーディングすることもできる。エンコーディングされた情報(例えば、エンコーディングされたビデオ／映像情報)は、ビットストリーム形態でNAL(network abstraction layer)ユニット単位で送信または格納されることができる。前記ビデオ／映像情報は、アダプテーションパラメータセット(APS)、ピクチャパラメータセット(PPS)、シーケンスパラメータセット(PPS)、またはビデオパラメータセット(VPS)等、様々なパラメータセットに関する情報をさらに含むことができる。また、前記ビデオ／映像情報は、一般制限情報(general constraint information)をさらに含むことができる。本文書において、エンコーディング装置からデコーディング装置に伝達／シグナリングされる情報及び／又はシンタックス要素は、ビデオ／映像情報に含まれることができる。前記ビデオ／映像情報は、上述したエンコーディング手順を介してエンコーディングされて前記ビットストリームに含まれることができる。前記ビットストリームは、ネットワークを介して送信されることができ、またはデジタル格納媒体に格納されることができる。ここで、ネットワークは、放送網及び／又は通信網などを含むことができ、デジタル格納媒体は、USB、SD、CD、DVD、ブルーレイ、HDD、SSDなど、様々な格納媒体を含むことができる。エントロピーエンコーディング部240から出力された信号は、送信する送信部(図示せず)及び／又は格納する格納部(図示せず)がエンコーディング装置200の内／外部エレメントとして構成されることができ、または送信部は、エントロピーエンコーディング部240に含まれることもできる。

【0052】

量子化部233から出力された量子化された変換係数は、予測信号を生成するために用いられることができる。例えば、量子化された変換係数に逆量子化部234及び逆変換部235を介して逆量子化及び逆変換を適用することにより、レジデュアル信号(レジデュアルブロックまたはレジデュアルサンプル)を復元できる。加算部155は、復元されたレジデュアル信号をインター予測部221またはイントラ予測部222から出力された予測信号に加えることにより、復元(reconstructed)信号(復元ピクチャ、復元ブロック、復元サンプルアレイ)が生成され得る。スキップモードが適用された場合のように、処理対象ブロックに対するレジデュアルがない場合、予測されたブロックが復元ブロックとして使用されることができる。加算部250は、復元部または復元ブロック生成部と呼ばれることができる。生成された復元信号は、現在ピクチャ内の次の処理対象ブロックのイントラ予測のために使用されることができ、後述するように、フィルタリングを経て次のピクチャのインター予測のために使用されることもできる。

【0053】

一方、ピクチャエンコーディング及び／又は復元過程でLMCS(luma mapping with chroma scaling)が適用されることもできる。

【0054】

フィルタリング部260は、復元信号にフィルタリングを適用して主観的／客観的画質を向上させることができる。例えば、フィルタリング部260は、復元ピクチャに様々な

フィルタリング方法を適用して修正された(modified)復元ピクチャを生成でき、前記修正された復元ピクチャをメモリ270、具体的に、メモリ270のDPBに格納することができる。前記様々なフィルタリング方法は、例えば、デブロッキングフィルタリング、サンプル適応的オフセット(sample adaptive offset)、適応的ループフィルタ(adaptive loop filter)、両方向フィルタ(bilateral filter)などを含むことができる。フィルタリング部260は、各フィルタリング方法についての説明で後述するように、フィルタリングに関する様々な情報を生成してエントロピーエンコーディング部240に伝達することができる。フィルタリングに関する情報は、エントロピーエンコーディング部240でエンコーディングされてビットストリーム形態で出力されることができる。

10

【0055】

メモリ270に送信された修正された復元ピクチャは、インター予測部221で参照ピクチャとして使用されることができる。エンコーディング装置は、これを介してインター予測が適用される場合、エンコーディング装置100とデコーディング装置における予測ミスマッチを避けることができ、符号化効率も向上させることができる。

【0056】

メモリ270のDPBは、修正された復元ピクチャをインター予測部221における参照ピクチャとして使用するために格納することができる。メモリ270は、現在ピクチャ内の動き情報が導出された(または、エンコーディングされた)ブロックの動き情報及び／又は既に復元されたピクチャ内のブロックの動き情報を格納することができる。前記格納された動き情報は、空間的周辺ブロックの動き情報または時間的周辺ブロックの動き情報として活用するために、インター予測部221に伝達することができる。メモリ270は、現在ピクチャ内の復元されたブロックの復元サンプルを格納することができ、イントラ予測部222に伝達することができる。

20

【0057】

図3は、本文書の実施例が適用できるビデオ／映像デコーディング装置の構成を概略的に説明する図面である。以下、デコーディング装置というのは、映像デコーディング装置及び／又はビデオデコーディング装置を含むことができる。

【0058】

図3に示すように、デコーディング装置300は、エントロピーデコーディング部(entropy decoder)310、レジデュアル処理部(residual processor)320、予測部(predictor)330、加算部(add)340、フィルタリング部(filter)350、及びメモリ(memory)360を備えて構成されることができる。予測部330は、インター予測部331及びイントラ予測部332を備えることができる。レジデュアル処理部320は、逆量子化部(dequantizer)321及び逆変換部(inverse transformer)321を備えることができる。上述したエントロピーデコーディング部310、レジデュアル処理部320、予測部330、加算部340、及びフィルタリング部350は、実施形態によって1つのハードウェアコンポーネント(例えば、デコーダチップセットまたはプロセッサ)により構成されることができる。また、メモリ360は、DPB(decoded picture buffer)を備えることができ、デジタル格納媒体により構成されることもできる。前記ハードウェアコンポーネントは、メモリ360を内／外部コンポーネントとしてさらに備えることもできる。

30

40

【0059】

ビデオ／映像情報を含むビットストリームが入力されれば、デコーディング装置300は、図3のエンコーディング装置でビデオ／映像情報が処理されたプロセスに対応して映像を復元できる。例えば、デコーディング装置300は、前記ビットストリームから取得したブロック分割関連情報に基づいてユニット／ブロックを導出できる。デコーディング装置300は、エンコーディング装置で適用された処理ユニットを用いてデコーディングを行うことができる。したがって、デコーディングの処理ユニットは、例えば、コーディ

50

ングユニットであることができ、コーディングユニットは、コーディングツリーユニットまたは最大コーディングユニットからクアッドツリー構造、バイナリツリー構造、及び／又はターナリツリー構造にしたがって分割されることができる。コーディングユニットから1つ以上の変換ユニットが導出され得る。そして、デコーディング装置300を介してデコーディング及び出力された復元映像信号は、再生装置を介して再生されることができる。

【0060】

デコーディング装置300は、図3のエンコーディング装置から出力された信号をビットストリーム形態で受信することができ、受信された信号は、エン트로ピーデコーディング部310を介してデコーディングされることができる。例えば、エン트로ピーデコーディング部310は、前記ビットストリームをパースングして映像復元（または、ピクチャ復元）に必要な情報（例えば、ビデオ／映像情報）を導出できる。前記ビデオ／映像情報は、アダプテーションパラメータセット（APS）、ピクチャパラメータセット（PPS）、シーケンスパラメータセット（SPS）、またはビデオパラメータセット（VPS）など、様々なパラメータセットに関する情報をさらに含むことができる。また、前記ビデオ／映像情報は、一般制限情報（general constraint information）をさらに含むことができる。デコーディング装置は、前記パラメータセットに関する情報及び／又は前記一般制限情報に基づいてさらにピクチャをデコーディングすることができる。本文書において後述されるシグナリング／受信される情報及び／又はシンタックス要素は、前記デコーディング手順を介してデコーディングされて、前記ビットストリームから取得されることができる。例えば、エン트로ピーデコーディング部310は、指数ゴロム符号化、CAVLC、またはCABACなどのコーディング方法を基にビットストリーム内の情報をデコーディングし、映像復元に必要なシンタックス要素の値、レジデュアルに関する変換係数の量子化された値を出力できる。より具体的に、CABACエン트로ピーデコーディング方法は、ビットストリームで各シンタックス要素に該当するピンを受信し、デコーディング対象のシンタックス要素情報と周辺及びデコーディング対象ブロックのデコーディング情報あるいは以前ステップでデコーディングされたシンボル／ピンの情報を利用して文脈（context）モデルを決定し、決定された文脈モデルによってピン（bin）の発生確率を予測してピンの算術デコーディング（arithmetic decoding）を行い、各シンタックス要素の値に該当するシンボルを生成できる。このとき、CABACエン트로ピーデコーディング方法は、文脈モデル決定後、次のシンボル／ピンの文脈モデルのためにデコーディングされたシンボル／ピンの情報を利用して文脈モデルをアップデートすることができる。エン트로ピーデコーディング部310でデコーディングされた情報のうち、予測に関する情報は、予測部（インター予測部332及びイントラ予測部331）に提供され、エン트로ピーデコーディング部310でエン트로ピーデコーディングが行われたレジデュアル値、すなわち、量子化された変換係数及び関連パラメータ情報は、レジデュアル処理部320に入力されることができる。レジデュアル処理部320は、レジデュアル信号（レジデュアルブロック、レジデュアルサンプル、レジデュアルサンプルアレイ）を導出できる。また、エン트로ピーデコーディング部310でデコーディングされた情報のうち、フィルタリングに関する情報は、フィルタリング部350に提供されることができる。一方、エンコーディング装置から出力された信号を受信する受信部（図示せず）がデコーディング装置300の内／外部エレメントとしてさらに構成されることができ、または受信部は、エン트로ピーデコーディング部310の構成要素であることもできる。一方、本文書に係るデコーディング装置は、ビデオ／映像／ピクチャデコーディング装置と呼ばれることができ、前記デコーディング装置は、情報デコーダ（ビデオ／映像／ピクチャ情報デコーダ）及びサンプルデコーダ（ビデオ／映像／ピクチャサンプルデコーダ）に区分することもできる。前記情報デコーダは、前記エン트로ピーデコーディング部310を備えることができ、前記サンプルデコーダは、前記逆量子化部321、逆変換部322、加算部340、フィルタリング部350、メモリ360、インター予測部332、及びイントラ予測部331のうち、少なくとも

10

20

30

40

50

も1つを備えることができる。

【0061】

逆量子化部321では、量子化された変換係数を逆量子化して変換係数を出力できる。逆量子化部321は、量子化された変換係数を2次元のブロック形態で再整列することができる。この場合、前記再整列は、エンコーディング装置で行われた係数スキャン順序に基づいて再整列を行うことができる。逆量子化部321は、量子化パラメータ（例えば、量子化ステップサイズ情報）を用いて量子化された変換係数に対する逆量子化を行い、変換係数（transform coefficient）を取得できる。

【0062】

逆変換部322では、変換係数を逆変換してレジデュアル信号（レジデュアルブロック、レジデュアルサンプルアレイ）を取得するようになる。

【0063】

予測部は、現在ブロックに対する予測を行い、前記現在ブロックに対する予測サンプルを含む予測されたブロック（predicted block）を生成できる。予測部は、エントロピーデコーディング部310から出力された前記予測に関する情報に基づいて、前記現在ブロックにイントラ予測が適用されるか、またはインター予測が適用されるか決定することができ、具体的なイントラ／インター予測モードを決定できる。

【0064】

予測部320は、後述する様々な予測方法に基づいて予測信号を生成できる。例えば、予測部は、1つのブロックに対する予測のために、イントラ予測またはインター予測を適用できるだけでなく、イントラ予測とインター予測とを同時に適用することができる。これは、combined inter and intra prediction（CIIP）と呼ばれることができる。また、予測部は、ブロックに対する予測のために、イントラブロックコピー（intra block copy、IBC）予測モードに基づくことができ、またはパレットモード（palette mode）に基づくこともできる。前記IBC予測モードまたはパレットモードは、例えば、SCC（screen content coding）などのように、ゲームなどのコンテンツ映像／動画コーディングのために使用されることができる。IBCは、基本的に現在ピクチャ内で予測を行うが、現在ピクチャ内で参照ブロックを導出する点においてインター予測と同様に行われることができる。すなわち、IBCは、本文書において説明されるインター予測技法のうち、少なくとも1つを利用することができる。パレットモードは、イントラコーディングまたはイントラ予測の一例と見なすことができる。パレットモードが適用される場合、パレットテーブル及びパレットインデックスに関する情報が前記ビデオ／映像情報に含まれてシグナリングされることができる。

【0065】

イントラ予測部331は、現在ピクチャ内のサンプルを参照して現在ブロックを予測できる。前記参照されるサンプルは、予測モードによって前記現在ブロックの周辺（neighbor）に位置することができ、または離れて位置することができる。イントラ予測において予測モードは、複数の非方向性モードと複数の方向性モードとを含むことができる。イントラ予測部331は、周辺ブロックに適用された予測モードを用いて、現在ブロックに適用される予測モードを決定することもできる。

【0066】

インター予測部332は、参照ピクチャ上で動きベクトルにより特定される参照ブロック（参照サンプルアレイ）に基づいて、現在ブロックに対する予測されたブロックを導くことができる。このとき、インター予測モードで送信される動き情報の量を減らすために、周辺ブロックと現在ブロックとの間の動き情報の相関性に基づいて動き情報をブロック、サブブロック、またはサンプル単位で予測することができる。前記動き情報は、動きベクトル及び参照ピクチャインデックスを含むことができる。前記動き情報は、インター予測方向（L0予測、L1予測、Bi予測等）情報をさらに含むことができる。インター予測の場合に、周辺ブロックは、現在ピクチャ内に存在する空間的周辺ブロック（spat

10

20

30

40

50

ial neighboring block) と参照ピクチャに存在する時間的周辺ブロック (temporal neighboring block) とを備えることができる。例えば、インター予測部 332 は、周辺ブロックに基づいて動き情報候補リストを構成し、受信した候補選択情報に基づいて前記現在ブロックの動きベクトル及び／又は参照ピクチャインデックスを導出できる。様々な予測モードに基づいてインター予測が行われ得るし、前記予測に関する情報は、前記現在ブロックに対するインター予測のモードを指示する情報を含むことができる。

【0067】

加算部 340 は、取得されたレジデュアル信号を予測部 (インター予測部 332 及び／又はイントラ予測部 331 を備える) から出力された予測信号 (予測されたブロック、予測サンプルアレイ) に加えることにより、復元信号 (復元ピクチャ、復元ブロック、復元サンプルアレイ) を生成できる。スキップモードが適用された場合のように、処理対象ブロックに対するレジデュアルがない場合、予測されたブロックが復元ブロックとして使用されることができる。

10

【0068】

加算部 340 は、復元部または復元ブロック生成部と呼ばれることができる。生成された復元信号は、現在ピクチャ内の次の処理対象ブロックのイントラ予測のために使用されることができ、後述するように、フィルタリングを経て出力されることができ、または次のピクチャのインター予測のために使用されることもできる。

【0069】

一方、ピクチャデコーディング過程で LMCS (luma mapping with chroma scaling) が適用されることもできる。

20

【0070】

フィルタリング部 350 は、復元信号にフィルタリングを適用して主観的／客観的画質を向上させることができる。例えば、フィルタリング部 350 は、復元ピクチャに様々なフィルタリング方法を適用して修正された (modified) 復元ピクチャを生成でき、前記修正された復元ピクチャをメモリ 360、具体的に、メモリ 360 の DPB に送信することができる。前記様々なフィルタリング方法は、例えば、デブロッキングフィルタリング、サンプル適応的オフセット (sample adaptive offset)、適応的ループフィルタ (adaptive loop filter)、両方向フィルタ (bilateral filter) などを含むことができる。

30

【0071】

メモリ 360 の DPB に格納された (修正された) 復元ピクチャは、インター予測部 332 で参照ピクチャとして使用されることができる。メモリ 360 は、現在ピクチャ内の動き情報が導出された (または、デコーディングされた) ブロックの動き情報及び／又は既に復元されたピクチャ内のブロックの動き情報を格納できる。前記格納された動き情報は、空間的周辺ブロックの動き情報または時間的周辺ブロックの動き情報として活用するために、インター予測部 260 に伝達することができる。メモリ 360 は、現在ピクチャ内の復元されたブロックの復元サンプルを格納でき、イントラ予測部 331 に伝達することができる。

40

【0072】

本明細書において、エンコーディング装置 200 のフィルタリング部 260、インター予測部 221、及びイントラ予測部 222 で説明された実施形態等は、各々デコーディング装置 300 のフィルタリング部 350、インター予測部 332、及びイントラ予測部 331 にも同一または対応するように適用されることができる。

【0073】

インター予測が適用される場合、エンコーディング装置／デコーディング装置の予測部は、ブロック単位でインター予測を行って予測サンプルを導出することができる。インター予測は、現在ピクチャ以外のピクチャのデータ要素 (ex. サンプル値、又は動き情報) に依存的な方法で導出される予測を示すことができる (Inter predicti

50

on can be a prediction derived in a manner that is dependent on data elements (ex. sample values or motion information) of picture(s) other than the current picture)。現在ブロックにインター予測が適用される場合、参照ピクチャのインデックスの指す参照ピクチャ上で動きベクトルにより特定される参照ブロック（参照サンプルアレイ）に基づいて、現在ブロックに対する予測されたブロック（予測サンプルアレイ）を誘導することができる。このとき、インター予測モードで送信される動き情報の量を減らすために、周辺ブロックと現在ブロックとの間の動き情報の相関性に基づいて、現在ブロックの動き情報をブロック、サブブロック又はサンプル単位で予測することができる。前記動き情報は、動きベクトル及び参照ピクチャのインデックスを含むことができる。前記動き情報は、インター予測タイプ（L0予測、L1予測、Bi予測等）の情報を更に含むことができる。インター予測が適用される場合、周辺ブロックは、現在ピクチャ内に存在する空間的周辺ブロック（spatial neighboring block）と、参照ピクチャに存在する時間的周辺ブロック（temporal neighboring block）とを含むことができる。前記参照ブロックを含む参照ピクチャと、前記時間的周辺ブロックを含む参照ピクチャとは同一であってもよく、異なってもよい。前記時間的周辺ブロックは、同じ位置参照ブロック（collocated reference block）、同じ位置CU（colCU）等の名称で呼ばれ得、前記時間的周辺ブロックを含む参照ピクチャは、同じ位置ピクチャ（collocated picture、colPic）と呼ばれることもある。例えば、現在ブロックの周辺ブロックに基づいて動き情報の候補リストが構成されることができ、前記現在ブロックの動きベクトル及び／又は参照ピクチャのインデックスを導出するために、どの候補が選択（使用）されるかを指示するフラグ又はインデックス情報がシグナリングできる。様々な予測モードに基づいてインター予測が行われ、例えば、スキップモードとマージモードの場合、現在ブロックの動き情報は、選択された周辺ブロックの動き情報を同一であり得る。スキップモードの場合、マージモードと異なり、レジデュアル信号が送信されないことがある。動き情報予測（motion vector prediction、MVP）モードの場合、選択された周辺ブロックの動きベクトルを動きベクトル予測子（motion vector predictor）として利用し、動きベクトル差分（motion vector difference）はシグナリングされることができる。この場合、前記動きベクトル予測子及び動きベクトル差分の和を用いて、前記現在ブロックの動きベクトルを導出することができる。

【0074】

前記動き情報は、インター予測タイプ（L0予測、L1予測、Bi予測等）に応じて、L0動き情報及び／又はL1動き情報を含むことができる。L0方向の動きベクトルは、L0動きベクトル又はMVL0と呼ばれ得、L1方向の動きベクトルは、L1動きベクトル又はMVL1と呼ばれ得る。L0動きベクトルに基づいた予測は、L0予測と呼ばれ得、L1動きベクトルに基づいた予測をL1予測と呼ばれ得、前記L0動きベクトル及び前記L1動きベクトルの両方に基づいた予測を双（Bi）予測と呼ばれ得る。ここで、L0動きベクトルは、参照ピクチャリストL0（L0）に関連した動きベクトルを示すことができ、L1動きベクトルは、参照ピクチャリストL1（L1）に関連した動きベクトルを示すことができる。参照ピクチャリストL0は、前記現在ピクチャよりも、出力順序上、以前のピクチャを参照ピクチャに含むことができ、参照ピクチャリストL1は、前記現在ピクチャよりも、出力順序上、以後のピクチャを含むことができる。前記以前のピクチャは、順方向（参照）ピクチャと呼ばれ得、前記以後のピクチャは、逆方向（参照）ピクチャと呼ばれ得る。前記参照ピクチャリストL0は、前記現在ピクチャよりも、出力順序上、以後のピクチャを参照ピクチャにさらに含むことができる。この場合、前記参照ピクチャリストL0内で前記以前のピクチャが先にインデキシングされ、前記以後のピクチャは、その後にインデキシングされ得る。前記参照ピクチャリストL1は、前記現在ピクチャよりも、出力順序上、以前のピクチャを参照ピクチャにさらに含むことができる。この場

10

20

30

40

50

合、前記参照ピクチャリスト1内で前記以後のピクチャが先にインデキシングされ、前記以前のピクチャは、その後インデキシングされ得る。ここで、出力順序は、POC (picture order count) 順序 (order) に対応し得る。

【0075】

ピクチャ内の現在ブロックの予測のために多様なインター予測モードが使用できる。例えば、マージモード、スキップモード、MVP (motion vector prediction) モード、アフィン (affine) モード、サブブロックマージモード、MMVD (merge with MVD) モード等の多様なモードが使用できる。DMVR (Decoder side motion vector refinement) モード、AMVR (adaptive motion vector resolution) モード、Bi-prediction with CU-level weight (BCW)、Bi-directional optical flow (BDOF) 等が付随的モードとしてさらに、又は代わりに使用できる。アフィンモードは、アフィン動き予測 (affine motion prediction) モードと呼ばれることもある。MVPモードは、AMVP (advanced motion vector prediction) モードと呼ばれることもある。本文書で、一部のモード及び／又は一部のモードにより導出された動き情報の候補は、他のモードの動き情報に関する候補の一つに含まれることもある。例えば、HMVP候補は、前記マージ／スキップモードのマージ候補として追加されることがあり、又は前記MVPモードのmv p候補として追加されることがある。前記HMVP候補が前記マージモード又はスキップモードの動き情報の候補として使用される場合、前記HMVP候補は、HMVPマージ候補と呼ばれ得る。

【0076】

現在ブロックのインター予測モードを指す予測モード情報がエンコーディング装置からデコーディング装置にシグナリングされることができる。前記予測モード情報は、ビットストリームに含まれ、デコーディング装置に受信されることができる。前記予測モード情報は、多数の候補モードのうちの一つを指示するインデックス情報を含むことができる。或いは、フラグ情報の階層的シグナリングを通じて、インター予測モードを指示することもある。この場合、前記予測モード情報は、一つ以上のフラグを含むことができる。例えば、スキップフラグをシグナリングしてスキップモードの適用可否を指示し、スキップモードが適用されない場合に、マージフラグをシグナリングしてマージモードの適用可否を指示し、マージモードが適用されない場合に、MVPモードが適用されるものと指示するか、追加的な区分のためのフラグをさらにシグナリングすることもある。アフィンモードは、独立したモードとしてシグナリングされることもあり、又はマージモード又はMVPモード等に従属的なモードとしてシグナリングされることもある。例えば、アフィンモードは、アフィンマージモード及びアフィンMVPモードを含むことができる。

【0077】

一方、現在ブロックに、前述したlist0 (L0) 予測、list1 (L1) 予測、又は双 (BI) 予測が現在ブロック (現在コーディングユニット) に使用されるか否かを示す情報がシグナリングできる。前記情報は、動き予測方向情報、インター予測方向情報又はインター予測指示情報と呼ばれ得、例えば、inter_pred_idcのシンタックス要素の形態で構成／エンコーディング／シグナリングされることができる。すなわち、inter_pred_idcのシンタックス要素は、前述したL0予測、L1予測、又は双予測が現在ブロック (現在コーディングユニット) に使用されるか否かを示すことができる。本文書では、説明の便宜のために、inter_pred_idcのシンタックス要素が指すインター予測タイプ (L0予測、L1予測、又はBI予測) は、動き予測方向と表示されることができる。例えば、L0予測はpred_L0、L1予測はpred_L1、双予測はpred_BIで表されることがある。

【0078】

前述したように、1つのピクチャは、一つ以上のスライス (slice) を含むことができる。スライスは、Iスライス (intra slice)、Pスライス (predi

10

20

30

40

50

ctive slice) 及び B スライス (bi-predictive slice) を含むスライスタイプのうちの一つのタイプを有することができる。前記スライスタイプは、スライスタイプ情報に基づいて指示されることができる。I スライス内のブロックに対しては、予測のためにインター予測は使用されず、イントラ予測のみ使用されることができる。もちろん、この場合にも、予測なしで原本サンプル値をコーディングしてシグナリングすることもある。P スライス内のブロックに対しては、イントラ予測又はインター予測が使用でき、インター予測が使用される場合には、単 (uni) 予測のみ使用できる。一方、B スライス内のブロックに対しては、イントラ予測又はインター予測が使用でき、インター予測が使用される場合には、最大の双 (bi) 予測まで使用できる。すなわち、B スライス内のブロックに対しては、インター予測が使用される場合、単予測又は双予測が使用できる。

10

【0079】

L0 及び L1 は、現在ピクチャよりも以前にエンコーディング／デコーディングされた参照ピクチャを含むことができる。ここで、L0 は、参照ピクチャリスト 0 を示すことができ、L1 は参照ピクチャリスト 1 を示すことができる。例えば、L0 は、POC (picture order count) 順序上、現在ピクチャよりも以前及び／又は以後の参照ピクチャを含むことができ、L1 は、POC 順序上、現在ピクチャよりも以後及び／又は以前の参照ピクチャを含むことができる。この場合、L0 には POC 順序上、現在ピクチャよりも以前の参照ピクチャに相対的にさらに低い参照ピクチャのインデックスが割り当てられることがあり、L1 には POC 順序上、現在ピクチャよりも以後の参照ピクチャに相対的にさらに低い参照ピクチャのインデックスが割り当てられることがある。B スライスの場合、双予測が適用でき、この場合にも、単方向の双予測が適用でき、又は両方向の双予測が適用できる。両方向の双予測は、真 (true) の双予測と呼ばれ得る。

20

【0080】

一方、現在ブロックの動き情報を用いてインター予測が行われ得る。エンコーディング装置は、動き推定 (motion estimation) 手順を通じて、現在ブロックに対する最適の動き情報を導出することができる。例えば、エンコーディング装置は、現在ブロックに対する原本ピクチャ内の原本ブロックを用いて、相関性の高い類似する参照ブロックを参照ピクチャ内の決められた探索範囲内で分数ピクセル単位で探索することができる。これを通じて、動き情報を導出することができる。ブロックの類似性は、位相 (phase) ベースのサンプル値の差に基づいて導出することができる。例えば、ブロックの類似性は、現在ブロック (又は現在ブロックのテンプレート) と参照ブロック (又は参照ブロックのテンプレート) との間の SAD (sum of absolute differences) に基づいて計算されることができる。この場合、探索領域内の SAD が最も小さい参照ブロックに基づいて動き情報を導出することができる。導出された動き情報は、インター予測モードに基づいて、様々な方法によってデコーディング装置にシグナリングされることができる。

30

【0081】

図 4 は、インター予測でマージモードを説明するための図である。

【0082】

40

マージモード (merge mode) が適用される場合、現在予測ブロックの動き情報が直接的に送信されず、周辺予測ブロックの動き情報を用いて、前記現在予測ブロックの動き情報を誘導することになる。従って、マージモードを用いたことを知らせるフラグ情報及び周辺のどの予測ブロックを用いたかを知らせるマージインデックスを送信することによって、現在予測ブロックの動き情報を指示することができる。前記マージモードは、レギュラーマージモード (regular merge mode) と呼ばれ得る。例えば、前記マージモードは、regular_merge_flag のシンタックス要素の値が 1 である場合に適用されることができる。

【0083】

エンコーディング装置は、マージモードを行うために、現在予測ブロックの動き情報を

50

誘導するために用いられるマージ候補ブロック (merge candidate block) をサーチ (search) しなければならない。例えば、前記マージ候補ブロックは、最大5個まで用いられることができるが、本文書の実施例はこれに限定されない。また、前記マージ候補ブロックの最大の個数はスライスヘッダ又はタイルグループヘッダで送信されることができ、本文書の実施例はこれに限定されない。前記マージ候補ブロックを見つけた後、エンコーディング装置はマージ候補リストを生成することができ、これらのうち最も小さい費用を有するマージ候補ブロックを最終のマージ候補ブロックとして選択することができる。

【0084】

本文書は、前記マージ候補リストを構成するマージ候補ブロックに対する多様な実施例を提供することができる。

10

【0085】

例えば、前記マージ候補リストは5個のマージ候補ブロックを用いることができる。例えば、4個の空間的マージ候補 (spatial merge candidate) と1個の時間的マージ候補 (temporal merge candidate) とを用いることができる。具体的な例として、空間的マージ候補の場合、図4に示すブロックを空間的マージ候補に用いることができる。以下、前記空間的マージ候補又は後述する空間的MVP候補はSMVPと呼ばれ得、前記時間的マージ候補又は後述する時間的MVP候補はTMVPと呼ばれ得る。

【0086】

前記現在ブロックに対するマージ候補リストは、例えば、次のような手続に基づいて構成されることができる。

20

【0087】

コーディング装置 (エンコーディング装置/デコーディング装置) は、現在ブロックの空間的周辺ブロックを探索して、導出された空間的マージ候補をマージ候補リストに挿入できる。例えば、前記空間的周辺ブロックは、前記現在ブロックの左下側角の周辺ブロック、左側の周辺ブロック、右上側角の周辺ブロック、上側の周辺ブロック、左上側角の周辺ブロックを含むことができる。但し、これは例示であって、前述した空間的周辺ブロック以外にも、右側の周辺ブロック、下側の周辺ブロック、右下側の周辺ブロック等、追加的な周辺ブロックがさらに前記空間的周辺ブロックとして使用されることができる。コーディング装置は、前記空間的周辺ブロックを優先順位に基づいて探索し、可用なブロックを検出し、検出されたブロックの動き情報を前記空間的マージ候補として導出することができる。例えば、エンコーディング装置又はデコーディング装置は、図4に示された5個のブロックをA1→B1→B0→A0→B2のように順序通りに探索し、可用な候補を順次にインデキシングしてマージ候補リストで構成することができる。

30

【0088】

コーディング装置は、前記現在ブロックの時間的周辺ブロックを探索して、導出された時間的マージ候補を前記マージ候補リストに挿入できる。前記時間的周辺ブロックは、前記現在ブロックが位置する現在ピクチャと異なるピクチャである参照ピクチャ上に位置し得る。前記時間的周辺ブロックが位置する参照ピクチャは、コロケートド (collocated) ピクチャ又はコル (col) ピクチャと呼ばれ得る。前記時間的周辺ブロックは、前記コル (col) ピクチャ上での前記現在ブロックに対する同じ位置ブロック (co-located block) の右下側角の周辺ブロック及び右下側中心のブロックの順に探索されることができる。一方、動き情報の圧縮 (motion data compression) が適用される場合、前記コル (col) ピクチャに一定の格納単位毎に特定の動き情報を代表の動き情報として格納することができる。この場合、前記一定の格納単位内の全てのブロックに対する動き情報を格納する必要がなく、これを通じて、動き情報の圧縮 (motion data compression) 効果が得られる。この場合、一定の格納単位は、例えば、16x16のサンプル単位、又は8x8のサンプル単位等で予め決められることもあり、又はエンコーディング装置からデコーディング装置

40

50

に前記一定の格納単位に対するサイズ情報がシグナリングされることもある。前記動き情報の圧縮 (motion data compression) が適用される場合、前記時間的周辺ブロックの動き情報は、前記時間的周辺ブロックが位置する前記一定の格納単位の代表の動き情報に代替されることができる。すなわち、この場合、具現の側面で見ると、前記時間的周辺ブロックの座標に位置する予測ブロックではなく、前記時間的周辺ブロックの座標 (左上のサンプルポジション) に基づいて、一定値だけ算術的右シフトの後、算術的左シフトした位置をカバーする予測ブロックの動き情報に基づいて前記時間的マージ候補が導出できる。例えば、前記一定の格納単位が $2n \times 2n$ のサンプル単位である場合、前記時間的周辺ブロックの座標が (x_{Tnb}, y_{Tnb}) とすれば、修正された位置である $((x_{Tnb} > n) < n)$ 、 $((y_{Tnb} > n) < n)$ に位置する予測ブロックの動き情報が前記時間的マージ候補のために使用できる。具体的に、例えば、前記一定の格納単位が 16×16 のサンプル単位である場合、前記時間的周辺ブロックの座標が (x_{Tnb}, y_{Tnb}) とすれば、修正された位置である $((x_{Tnb} > 4) < 4)$ 、 $((y_{Tnb} > 4) < 4)$ に位置する予測ブロックの動き情報が前記時間的マージ候補のために使用できる。或いは、例えば、前記一定の格納単位が 8×8 のサンプル単位である場合、前記時間的周辺ブロックの座標が (x_{Tnb}, y_{Tnb}) とすれば、修正された位置である $((x_{Tnb} > 3) < 3)$ 、 $((y_{Tnb} > 3) < 3)$ に位置する予測ブロックの動き情報が前記時間的マージ候補のために使用できる。

【0089】

コーディング装置は、現在のマージ候補の個数が最大のマージ候補の個数よりも小さいか否かを確認することができる。前記最大のマージ候補の個数は、予め定義されるか、エンコーディング装置からデコーディング装置にシグナリングされることができる。例えば、エンコーディング装置は、前記最大のマージ候補の個数に関する情報を生成し、エンコーディングして、ビットストリームの形態で前記デコーダに伝達できる。前記最大のマージ候補の個数がすべて満たされると、以降の候補追加の過程は行わなくてもよい。

【0090】

前記確認の結果、前記現在のマージ候補の個数が前記最大のマージ候補の個数よりも小さい場合、コーディング装置は、追加のマージ候補を前記マージ候補リストに挿入できる。例えば、前記追加のマージ候補は、後述する履歴ベースマージ候補 (history based merge candidate(s))、ペアワイズ平均マージ候補 (pair-wise average merge candidate(s))、ATMVP、組み合わせられた双予測 (combined bi-predictive) マージ候補 (現在スライス／タイルグループのスライス／タイルグループのタイプが B タイプである場合) 及び／又はゼロベクトルマージ候補の少なくとも一つを含むことができる。

【0091】

前記確認の結果、前記現在のマージ候補の個数が前記最大のマージ候補の個数よりも小さくない場合、コーディング装置は、前記マージ候補リストの構成を終了し得る。この場合、エンコーディング装置は、RD (rate-distortion) コスト (cost) に基づき、前記マージ候補リストを構成するマージ候補のうち、最適のマージ候補を選択することができる。前記選択されたマージ候補を指す選択情報 (ex. merge index) をデコーディング装置にシグナリングすることができる。デコーディング装置は、前記マージ候補リスト及び前記選択情報に基づいて、前記最適のマージ候補を選択することができる。

【0092】

前記選択されたマージ候補の動き情報が前記現在ブロックの動き情報に使用でき、前記現在ブロックの動き情報に基づいて、前記現在ブロックの予測サンプルを導出することができることは前述した通りである。エンコーディング装置は、前記予測サンプルに基づいて前記現在ブロックのレジデュアルサンプルを導出することができ、前記レジデュアルサンプルに関するレジデュアル情報をデコーディング装置にシグナリングできる。デコーディング装置は、前記レジデュアル情報に基づいて導出されたレジデュアルサンプル、及び

前記予測サンプルに基づいて復元サンプルを生成することができ、これに基づいて、復元ピクチャを生成することができることは前述した通りである。

【0093】

スキップモード (skip mode) が適用される場合、前記でマージモードが適用される場合と同じ方法で前記現在ブロックの動き情報を導出することができる。但し、スキップモードが適用される場合、該当ブロックに対するレジデュアル信号が省略され、よって、予測サンプルが直ぐに復元サンプルに用いられることができる。前記スキップモードは、例えば、cu__skip__flagのシンタックス要素の値が1である場合に適用されることができる。

【0094】

一方、前記ペアワイズ平均マージ候補 (pair-wise average merge candidate) は、ペアワイズ平均候補又はペアワイズ候補と呼ばれ得る。ペアワイズ平均候補は、既存の (existing) マージ候補リストで予め定義された候補のペアを平均して生成されることができる。また、予め定義されたペアは、{(0, 1)、(0, 2)、(1, 2)、(0, 3)、(1, 3)、(2, 3)}のように定義できる。ここで、数字は、マージ候補リストに対するマージインデックスを示すことができる。平均化された動きベクトル (averaged motion vector) は、各参照リストに対して別に計算されることができる。例えば、2個の動きベクトルが1つのリスト内で利用可能な場合、前記2つの動きベクトルは、互いに異なる参照ピクチャを指しても、平均化され得る。例えば、1つの動きベクトルのみが利用可能な場合、直ちに (directly) 1つが使用できる。例えば、利用可能な動きベクトルがない場合、リストを有効ではない状態で維持できる。

【0095】

例えば、ペアワイズ平均マージ候補が追加された後にも、マージ候補リストが満たされていない場合、即ち、マージ候補リスト内に現在のマージ候補の個数が最大のマージ候補の個数よりも小さい場合、最大のマージ候補の番号が示されるまでゼロベクトル (zero MVP) が最後に挿入され得る。すなわち、マージ候補リスト内の現在のマージ候補の個数が最大のマージ候補の個数になるまでゼロベクトルを挿入し得る。

【0096】

一方、既存にはコーディングブロックの動きを表現するために、ただ1つの動きベクトルを使用することができた。すなわち、トランスレーション (translation) 動きモデルが使用できた。但し、このような方法がブロック単位で最適の動きを表現しているかもしれないが、実際の各サンプルの最適の動きではなく、サンプル単位で最適の動きベクトルを決定することができれば、コーディングの効率を高めることができる。このため、アフィン動きモデル (affine motion model) が使用できる。アフィン動きモデルを使用してコーディングするアフィン動き予測方法は、次の通りである。

【0097】

アフィン動き予測方法は、2個、3個又は4個の動きベクトルを用いて、ブロックの各サンプル単位で動きベクトルを表現することができる。例えば、アフィン動きモデルは、4つの動きを表現することができる。アフィン動きモデルが表現し得る動きのうち、3つの動き (トランスレーション (translation)、スケール (scale)、回転 (rotate)) を表現するアフィン動きモデルは、類似 (similarity) (又は簡素化した (simplified)) アフィン動きモデルと呼ばれ得、これを基準に説明するが、前述した動きモデルに限定されるわけではない。

【0098】

図5a及び図5bは、アフィン動き予測のためのCPMVを例示的に示す。

【0099】

アフィン動き予測は、2つ以上のコントロールポイント動きベクトル (CPMV: Control Point Motion Vector) を用いて、ブロックが含むサンプル位置の動きベクトルを決定することができる。このとき、動きベクトルの集合をアフィ

10

20

30

40

50

ン動きベクトルフィールド（MVF：Motion Vector Field）と示すことができる。

【0100】

例えば、図5aは、2つのCPMVが用いられる場合を示すことができ、これは、4パラメータアフィンモデルと呼ばれ得る。この場合、(x, y)のサンプル位置での動きベクトルは、例えば、数式1のように決定されることができる。

【0101】

【数1】

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{1y}-mv_{0y}}{W}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{1x}-mv_{0x}}{W}y + mv_{0y} \end{cases}$$

10

【0102】

例えば、図5bは、3個のCPMVが用いられる場合を示すことができ、これは、6パラメータアフィンモデルと呼ばれ得る。この場合、(x, y)のサンプル位置での動きベクトルは、例えば、数式2のように決定されることができる。

【0103】

【数2】

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{2x}-mv_{0x}}{H}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{2y}-mv_{0y}}{H}y + mv_{0y} \end{cases}$$

20

【0104】

数式1及び数式2において、 $\{v_x, v_y\}$ は、(x, y)位置での動きベクトルを示すことができる。また、 $\{v_{0x}, v_{0y}\}$ は、コーディングブロックの左上側角の位置のコントロールポイント（CP：Control Point）のCPMVを示すことができ、 $\{v_{1x}, v_{1y}\}$ は、右上側角の位置のCPのCPMVを示すことができ、 $\{v_{2x}, v_{2y}\}$ は、左下側角の位置のCPのCPMVを示すことができる。また、Wは、現在ブロックの幅（width）を示すことができ、Hは、現在ブロックの高さ（height）を示すことができる。

30

【0105】

図6は、アフィンMVFがサブブロック単位で決定される場合を例示的に示す。

【0106】

エンコーディング／デコーディング過程で、アフィンMVFはサンプル単位又は既に定義されたサブブロック単位で決定されることができる。例えば、サンプル単位で決定する場合、各サンプル値を基準に動きベクトルが得られる。或いは、例えば、サブブロック単位で決定する場合、サブブロックの中央（センターの右下側、すなわち、中央の4個のサンプルのうちの右下側サンプル）サンプル値を基準に、該当ブロックの動きベクトルが得られる。すなわち、アフィン動き予測で、現在ブロックの動きベクトルは、サンプル単位又はサブブロック単位で導出されることができる。

40

【0107】

実施例では、アフィンMVFが4x4のサブブロック単位で決定される場合を仮定して説明し得るが、これは、説明の便宜のためであるので、サブブロックのサイズは多様に変形され得る。

【0108】

すなわち、アフィン予測が可能な場合、現在ブロックに適用可能な動きモデルは、3つ

50

(平行移動動きモデル (translational motion model)、4-パラメータアフィン動きモデル (4-parameter affine motion model)、6-パラメータアフィン動きモデル (6-parameter affine motion model)) を含むことができる。ここで、トランスレーション動きモデルは、既存のブロック単位の動きベクトルが使用されるモデルを示すことができ、4-パラメータアフィン動きモデルは、2個のCPMVが使用されるモデルを示すことができ、6-パラメータアフィン動きモデルは、3個のCPMVが使用されるモデルを示すことができる。

【0109】

一方、アフィン動き予測は、アフィンMVP (又はアフィンインター) モード又はアフィンマージモードを含むことができる。

【0110】

図7は、インター予測でアフィンマージモードを説明するための図である。

【0111】

例えば、アフィンマージモードで、CPMVはアフィン動き予測でコーディングされた周辺ブロックのアフィン動きモデルによって決定されることができる。例えば、サーチ (search) 順序上のアフィン動き予測でコーディングされた周辺ブロックがアフィンマージモードのために使用できる。すなわち、周辺ブロックの少なくとも一つがアフィン動き予測でコーディングされた場合、現在ブロックはアフィンマージモードでコーディングされることができる。ここで、アフィンマージモードは、AF_MERGEと呼ばれ得る。

【0112】

アフィンマージモードが適用される場合、周辺ブロックのCPMVを用いて、現在ブロックのCPMVが導出できる。この場合、周辺ブロックのCPMVがそのまま現在ブロックのCPMVとして使用されることもあり、周辺ブロックのCPMVが前記周辺ブロックのサイズ及び前記現在ブロックのサイズ等に基づいて修正され、現在ブロックのCPMVとして使用されることもある。

【0113】

一方、サブブロック単位で動きベクトル (MV: Motion Vector) が導出されるアフィンマージモードの場合には、サブブロックマージモードと呼ばれ得、これは、サブブロックマージフラグ (又はmerge_subblock_flagのシンタックス要素) に基づいて指示されることができる。或いは、merge_subblock_flagのシンタックス要素の値が1である場合、サブブロックマージモードが適用されることが指示され得る。この場合、後述するアフィンマージ候補リストは、サブブロックのマージ候補リストと呼ばれることもある。この場合、前記サブブロックのマージ候補リストには、後述するSbTMVPとして導出された候補がさらに含まれ得る。この場合、前記SbTMVPとして導出された候補は、前記サブブロックのマージ候補リストの0番のインデックスの候補として用いられ得る。言い換えると、前記SbTMVPとして導出された候補は、前記サブブロックのマージ候補リスト内で、後述する承継されたアフィン候補 (inherited affine candidate) 又は構成されたアフィン候補 (constructed affine candidate) より前に位置し得る。

【0114】

アフィンマージモードが適用される場合、現在ブロックに対するCPMVの導出のためにアフィンマージ候補リストが構成できる。例えば、アフィンマージ候補リストは、次の候補の少なくとも一つを含むことができる。1) 承継された (inherited) アフィンマージ候補。2) 構成された (constructed) アフィンマージ候補。3) ゼロ動きベクトル候補 (又はゼロベクトル)。ここで、前記承継されたアフィンマージ候補は、周辺ブロックがアフィンモードでコーディングされた場合、周辺ブロックのCPMVに基づいて導出される候補であり、前記構成されたアフィンマージ候補は、各CPM

10

20

30

40

50

V単位で該当CPの周辺ブロックのMVに基づいてCPMV_sを構成して導出された候補であり、ゼロ動きベクトル候補は、その値が0であるCPMVで構成された候補を示すことができる。

【0115】

前記アフィンマージ候補リストは、例えば、次のように構成されることができる。

【0116】

最大2個の承継されたアフィン候補があり得、承継されたアフィン候補は、周辺ブロックのアフィン動きモデルから導出されることができる。周辺ブロックは一つの左側の周辺ブロックと上側の周辺ブロックとを含むことができる。候補ブロックは、図4のように位置し得る。左側予測子(left predictor)のためのスキャン順序はA1→A0であり得、上側予測子(above predictor)のためのスキャン順序は、B1→B0→B2であり得る。左側及び上側それぞれから一つの承継された候補のみが選択できる。二つの承継された候補間にはブルーニングチェック(pruning check)が行われないことがある。

【0117】

周辺アフィンブロックが確認される場合、確認したブロックのコントロールポイント動きベクトルが現在ブロックのアフィンマージリスト内のCPMVP候補を導出するために使用されることができる。ここで、周辺のアフィンブロックは、現在ブロックの周辺ブロックのうち、アフィン予測モードでコーディングされたブロックを示すことができる。例えば、図7を参照すると、左下側(bottom-left)の周辺ブロックAがアフィン予測モードでコーディングされた場合、周辺ブロックAの左上側(top-left)角、右上側(top-right)角及び左下側(bottom-left)角の動きベクトル v_2 、 v_3 及び v_4 が獲得できる。周辺ブロックAが4-パラメータアフィン動きモデルでコーディングされた場合、現在ブロックの2個のCPMVが v_2 及び v_3 によって算出できる。周辺ブロックAが6-パラメータアフィン動きモデルでコーディングされた場合、現在ブロックの3個のCPMVが v_2 、 v_3 及び v_4 によって算出されることができる。

【0118】

図8は、アフィンマージモードで候補の位置を説明するための図である。

【0119】

構成された(constructed)アフィン候補は、各コントロールポイントの周辺の平行移動(translational)動き情報を組み合わせて構成される候補を意味し得る。コントロールポイントの動き情報は、特定された空間的周辺及び時間的周辺から導出されることができる。CPMV_k($k=1, 2, 3, 4$)は、 k 番目のコントロールポイントを示すことができる。

【0120】

図8を参照すると、CPMV₁のためにB2→B3→A2の順によってブロックがチェックでき、一番目に利用可能なブロックの動きベクトルが使用できる。CPMV₂のために、B1→B0の順によってブロックがチェックでき、CPMV₃のために、A1→A0の順によってブロックがチェックできる。TMVP(temporal motion vector predictor)は、利用可能な場合、CPMV₄として使用されることができる。

【0121】

4個のコントロールポイントの動きベクトルが獲得された後、アフィンマージ候補は獲得した動き情報に基づいて構成されることができる。コントロールポイント動きベクトルの組み合わせは、{CPMV₁, CPMV₂, CPMV₃}、{CPMV₁, CPMV₂, CPMV₄}、{CPMV₁, CPMV₃, CPMV₄}、{CPMV₂, CPMV₃, CPMV₄}、{CPMV₁, CPMV₂}及び{CPMV₁, CPMV₃}のように構成されることができ、羅列した順序によって構成されることができる。

【0122】

10

20

30

40

50

3個のCPMVの組み合わせは、6パラメータアフィンマージ候補を構成することができ、2個のCPMVの組み合わせは、4パラメータアフィンマージ候補を構成することができる。動きのスケーリング過程を回避するために、コントロールポイントの参照インデックスが互いに異なる場合、コントロールポイント動きベクトルの関連した組み合わせは捨てられ得る。

【0123】

図9は、インター予測でSbTMVPを説明するための図である。

【0124】

一方、SbTMVP (subblock-based temporal motion vector prediction) 方法が用いられることもある。例えば、SbTMVPは、ATMVP (advanced temporal motion vector prediction) と呼ばれることもある。SbTMVPは、動きベクトル予測及び現在ピクチャ内のCUに対するマージモードを向上するために、コロケートッドピクチャ (collocated picture) 内の動きフィールドを用いることができる。ここで、コロケートッドピクチャは、コル (col) ピクチャと呼ばれることもある。

【0125】

例えば、SbTMVPは、サブブロック (又はサブCU) レベルで動きを予測することができる。また、SbTMVPは、コルピクチャから時間的動き情報をフェッチング (fetching) する前に動きシフト (shift) を適用することができる。ここで、動きシフトは、現在ブロックの空間的周辺ブロックのうちの一つの動きベクトルから獲得されることができる。

【0126】

SbTMVPは、二つのステップによって、現在ブロック (又はCU) 内のサブブロック (又はサブCU) の動きベクトルを予測することができる。

【0127】

第1ステップで、空間的周辺ブロックは、図4のA1、B1、B0及びA0の順によって試験され得る。コル (col) ピクチャを自分の参照ピクチャとして使用する動きベクトルを有する一番目の空間的周辺ブロックが確認でき、動きベクトルは、適用される動きシフトとして選択されることができる。空間的周辺ブロックからこのような動きが確認されない場合、動きシフトは (0, 0) に設定されることができる。

【0128】

第2ステップでは、第1ステップで確認された動きシフトがcolピクチャからサブブロックレベルの動き情報 (動きベクトル及び参照インデックス) を獲得するために適用されることができる。例えば、動きシフトが現在ブロックの座標に追加できる。例えば、動きシフトが図4のA1の動きに設定されることができる。この場合、各サブブロックに対して、colピクチャ内の対応ブロックの動き情報は、サブブロックの動き情報を導出するために使用されることができる。時間的動きのスケーリングは、時間的動きベクトルの参照ピクチャと現在ブロックの参照ピクチャとを整理するために適用されることができる。

【0129】

SbTMVP候補及びアフィンマージ候補をすべて含む組み合わせられたサブブロックベースのマージリストは、アフィンマージモードのシグナリングに使用されることができる。ここで、アフィンマージモードは、サブブロックベースのマージモードと呼ばれ得る。SbTMVPモードは、SPS (sequence parameter set) に含まれたフラグにより利用可能、又は利用可能ではないことがある。SbTMVPモードが利用可能な場合、SbTMVP予測子は、サブブロックベースのマージ候補のリストの第1エントリに追加されることがあり、アフィンマージ候補が次についてくることがある。アフィンマージ候補リストの最大に許容されるサイズは5個であり得る。

【0130】

SbTMVPで使用するサブCU (又はサブブロック) のサイズは8x8に固定され得、アフィンマージモードと同様にSbTMVPモードは幅及び高さが全て8以上のプロ

10

20

30

40

50

ックにのみ適用され得る。追加的なS b T M V P マージ候補のエンコーディングのロジックは、他のマージ候補と同一であり得る。すなわち、P又はBスライス内の各CUに対して、追加的なRD (rate-distortion) コストを用いるRDチェックがS b T M V P 候補を用いるかを決定するために行われ得る。

【0131】

一方、予測モードによって導出された動き情報に基づいて、現在ブロックに対する予測されたブロックを導出することができる。前記予測されたブロックは、前記現在ブロックの予測サンプル（予測サンプルアレイ）を含むことができる。現在ブロックの動きベクトルが分数のサンプル単位を指す場合、補間 (interpolation) 手続が行われ得、これを通じて、参照ピクチャ内で分数のサンプル単位の参照サンプルに基づいて前記現在ブロックの予測サンプルが導出できる。現在ブロックにアフィンインター予測（アフィン予測モード）が適用される場合、サンプル／サブブロック単位のMVに基づいて予測サンプルを生成することができる。双 (Bi) 予測が適用される場合、L0予測（すなわち、参照ピクチャリストL0内の参照ピクチャとMV L0を用いた予測）に基づいて導出された予測サンプルと、L1予測（すなわち、参照ピクチャリストL1内の参照ピクチャとMV L1を用いた予測）に基づいて導出された予測サンプルの（位相による）加重和 (weighted sum)、又は加重平均 (weighted average) を通じて導出された予測サンプルが現在ブロックの予測サンプルに用いられることができる。ここで、L0方向の動きベクトルは、L0動きベクトル又はMV L0と呼ばれ得、L1方向の動きベクトルは、L1動きベクトル又はMV L1と呼ばれ得る。双 (Bi) 予測が適用される場合、L0予測に用いられた参照ピクチャとL1予測に用いられた参照ピクチャとが、現在ピクチャを基準に互いに異なる時間的方向に位置する場合（すなわち、双予測であるととも両方向予測に該当する場合）、これを真 (true) の双予測と呼ばれ得る。

【0132】

また、導出された予測サンプルに基づいて復元サンプル及び復元ピクチャが生成でき、以降のインループフィルタリング等の手続が行われ得ることは前述した通りである。

【0133】

一方、現在ブロックに双予測が適用される場合、加重平均に基づいて予測サンプルを導出することができる。例えば、加重平均を用いた双予測は、BCW (Bi-prediction with CU-level Weight)、BWA (Bi-prediction with Weighted Average) 又は加重平均双予測 (weighted averaging bi-prediction) と呼ばれ得る。

【0134】

既存に双予測信号（すなわち、双予測サンプル）は、L0予測信号（L0予測サンプル）とL1予測信号（L1予測サンプル）の単純平均を通じて導出されることができた。すなわち、双予測のサンプルは、L0参照ピクチャ及びMV L0に基づいたL0予測サンプルとL1参照ピクチャ及びMV L1に基づいたL1予測サンプルの平均として導出された。但し、双予測が適用される場合、次のようにL0予測信号とL1予測信号の加重平均を通じて、双予測信号（双予測サンプル）を導出することもある。例えば、双予測信号（双予測サンプル）は、数式3のように導出されることができる。

【0135】

【数3】

$$P_{bi-pred} = ((8 - w) * P_0 + w * P_1 + 4) \gg 3$$

【0136】

数式3において、 $P_{bi-pred}$ は、双予測信号の値、すなわち、双予測が適用されて導出された予測サンプル値を示すことができ、 w は、加重値を示すことができる。また、 P_0 は、L0予測信号の値、すなわち、L0予測が適用されて導出された予測サンプル値を示すことができ、 P_1 は、L1予測信号の値、すなわち、L1予測が適用されて導出さ

れた予測サンプル値を示すことができる。

【0137】

例えば、加重平均の双予測では、5個の加重値が許容され得る。例えば、前記5個の加重値(w)は、-2、3、4、5、又は10を含むことができる。すなわち、加重値(w)は、-2、3、4、5、又は10を含む加重値の候補のうち一つで決定されることができる。双予測が適用される各CUに対して、加重値wは2つの方法のうち一つにより決定されることができる。第一の方法は、併合されないCUに対して、加重値インデックスが動きベクトル差分(motion vector difference)後にシグナリングされることができる。第二の方法は、併合されたCUに対して、加重値インデックスがマージ候補インデックスに基づいて周辺ブロックから推論されることができる。

10

【0138】

例えば、加重平均の双予測は、256個以上のルマサンプルを有するCUに適用されることができる。すなわち、CUの幅及び高さの積が256よりも大きいか等しい場合に、加重平均の双予測が適用され得る。低遅延(low-delay)ピクチャの場合、5個の加重値が使用され得、低遅延ではない(non-low-delay)ピクチャの場合、3個の加重値が使用され得る。例えば、前記3個の加重値は、3、4又は5を含むことができる。

【0139】

例えば、エンコーディング装置で、早いサーチアルゴリズムは、エンコーディング装置の複雑性(complexity)を大きく増加させないながら、加重値インデックスを見つけるために適用されることができる。このようなアルゴリズムは、次のように要約できる。例えば、AMVR(adaptive motion vector resolution)と組み合わせられるとき(インター予測モードとしてAMVRが用いられるとき)、現在ピクチャが低遅延のピクチャである場合、同一ではない加重値は、1-pel及び4-pelの動きベクトルの精密度に対して条件部としてチェックされることができる。例えば、アフィン(affine)と組み合わせられるとき(インター予測モードとしてアフィン予測モードが用いられるとき)、アフィン予測モードが現在のベストモード(best mode)として選択された場合、アフィンME(Motion Estimation)は同一ではない加重値に対して行われ得る。例えば、双予測の2個の参照ピクチャが同一である場合、同一ではない加重値が条件部としてチェックされることができる。例えば、現在ピクチャと参照ピクチャとの間のPOC距離、コーディングQP(quantization parameter)及び時間的レベル(temporal level)に依存して特定の条件が満たされる場合、同一ではない加重値がサーチされないことがある。

20

30

【0140】

例えば、BCW加重値インデックス(又は加重値インデックス)は、一つのコンテキストコーディングされたビン(context coded bin)及び続くバイパスコーディングされたビン(bypass coded bin)を用いてコーディングされることができる。一番目のコンテキストコーディングされたビンは、同一の加重値が使用されるか否かを示すことができる。一番目のコンテキストコーディングされたビンに基づいて、同一ではない加重値が使用される場合、使用される同一ではない加重値を指示するために、バイパスコーディングを使用して追加的なビンがシグナリングできる。

40

【0141】

一方、本文書の一実施例によれば、マージモードのための動きベクトルの候補の構成時、時間的動きベクトルの候補が双予測を使用するときに加重平均のための加重値インデックスを誘導又は導出できる。すなわち、インター予測タイプが双(bi)予測である場合、マージ候補リスト内の時間的マージ候補(又は時間的動きベクトル候補)のための加重値インデックス情報を誘導又は導出できる。

【0142】

例えば、時間的動きベクトルの候補に対して、加重平均のための加重値インデックスは

50

常時 0 として導出されることができる。ここで、加重値インデックスが 0 であることは、各参照方向（すなわち、双予測で L 0 予測方向及び L 1 予測方向）の加重値が同一であることを意味し得る。例えば、この場合、マージモードのためのルマ成分の動きベクトルを導出する手続は、次の表の通りである。

【 0 1 4 3 】

【表 1】

8.4.2.2 Derivation process for luma motion vectors for merge mode

This process is only invoked when `merge_flag[ xCb ][ yCb ]` is equal to 1, where (`xCb`, `yCb`) specify the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture.

Inputs to this process are:

- a luma location (`xCb`, `yCb`) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable `cbWidth` specifying the width of the current coding block in luma samples,
- a variable `cbHeight` specifying the height of the current coding block in luma samples.

Outputs of this process are:

- the luma motion vectors in 1/16 fractional-sample accuracy `mvL0[ 0 ][ 0 ]` and `mvL1[ 0 ][ 0 ]`,
- the reference indices `refIdxL0` and `refIdxL1`,
- the prediction list utilization flags `predFlagL0[ 0 ][ 0 ]` and `predFlagL1[ 0 ][ 0 ]`,
- the bi-prediction weight index `gbIdx`.

The bi-prediction weight index `gbIdx` is set equal to 0.

The motion vectors `mvL0[ 0 ][ 0 ]` and `mvL1[ 0 ][ 0 ]`, the reference indices `refIdxL0` and `refIdxL1` and the prediction list utilization flags `predFlagL0[ 0 ][ 0 ]` and `predFlagL1[ 0 ][ 0 ]` are derived by the following ordered steps:

1. The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (`xCb`, `yCb`), the luma coding block width `cbWidth`, and the luma coding block height `cbHeight` as inputs, and the output being the availability flags `availableFlagA0`, `availableFlagA1`, `availableFlagB0`, `availableFlagB1` and `availableFlagB2`, the reference indices `refIdxLXA0`, `refIdxLXA1`, `refIdxLXB0`, `refIdxLXB1` and `refIdxLXB2`, the prediction list utilization flags `predFlagLXA0`, `predFlagLXA1`, `predFlagLXB0`, `predFlagLXB1` and `predFlagLXB2`, and the motion vectors `mvLXA0`, `mvLXA1`, `mvLXB0`, `mvLXB1` and `mvLXB2`, with `X` being 0 or 1, and the bi-prediction weight indices `gbIdxA0`, `gbIdxA1`, `gbIdxB0`, `gbIdxB1`, `gbIdxB2`.
2. The reference indices, `refIdxLXCol`, with `X` being 0 or 1, and the bi-prediction weight index `gbIdxCol` for the temporal merging candidate `Col` are set equal to 0.
3. The derivation process for temporal luma motion vector prediction as specified in in clause 8.4.2.11 is invoked with the luma location (`xCb`, `yCb`), the luma coding block width `cbWidth`, the luma coding block height `cbHeight` and the variable `refIdxL0Col` as inputs, and the output being the availability flag `availableFlagL0Col` and the temporal motion vector `mvL0Col`. The variables `availableFlagCol`, `predFlagL0Col` and `predFlagL1Col` are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \quad (8-283)$$

$$\text{predFlagL0Col} = \text{availableFlagL0Col} \quad (8-284)$$

$$\text{predFlagL1Col} = 0 \quad (8-285)$$

$$\text{gbIdxCol} = 0$$

(8-xxx)

4. When `tile_group_type` is equal to B, the derivation process for temporal luma motion vector prediction as specified in clause 8.4.2.11 is invoked with the luma location (`xCb`, `yCb`), the luma coding block width `cbWidth`, the luma coding block height `cbHeight` and the variable `refIdxL1Col` as inputs, and the output being the availability flag `availableFlagL1Col` and the temporal motion vector `mvL1Col`. The variables `availableFlagCol` and `predFlagL1Col` are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \mid \mid \text{availableFlagL1Col} \quad (8-286)$$

$$\text{predFlagL1Col} = \text{availableFlagL1Col} \quad (8-287)$$

【 0 1 4 4 】

10

20

30

40

50

【表 2】

5. The merging candidate list, mergeCandList, is constructed as follows:	<pre> i = 0 if(availableFlagA₁) mergeCandList[i++] = A₁ if(availableFlagB₁) mergeCandList[i++] = B₁ if(availableFlagB₀) mergeCandList[i++] = B₀ if(availableFlagA₀) mergeCandList[i++] = A₀ if(availableFlagB₂) mergeCandList[i++] = B₂ if(availableFlagCol) mergeCandList[i++] = Col </pre> (8-288)	10
6. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the mergeCandList.		
7. When numCurrMergeCand is less than (MaxNumMergeCand - 1) and NumHmvpCand is greater than 0, the following applies:	– The derivation process of history-based merging candidates as specified in 8.4.2.6 is invoked with mergeCandList, and numCurrMergeCand as inputs, and modified mergeCandList and numCurrMergeCand as outputs. – numOrigMergeCand is set equal to numCurrMergeCand.	
8. The derivation process for pairwise average merging candidates specified in clause 8.4.2.4 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList, numCurrMergeCand and numOrigMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0avgCand _k and refIdxL1avgCand _k , the prediction list utilization flags predFlagL0avgCand _k and predFlagL1avgCand _k and the motion vectors mvL0avgCand _k and mvL1avgCand _k of every new candidate avgCand _k being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate avgCand _k being added into mergeCandList is set equal to 0. The number of candidates being added, numAvgMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand). When numAvgMergeCand is greater than 0, k ranges from 0 to numAvgMergeCand - 1, inclusive.		20
9. The derivation process for zero motion vector merging candidates specified in clause 8.4.2.5 is invoked with the mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList and numCurrMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0zeroCand _m and refIdxL1zeroCand _m , the prediction list utilization flags predFlagL0zeroCand _m and predFlagL1zeroCand _m and the motion vectors mvL0zeroCand _m and mvL1zeroCand _m of every new candidate zeroCand _m being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate zeroCand _m being added into mergeCandList is set equal to 0. The number of candidates being added, numZeroMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand - numAvgMergeCand). When numZeroMergeCand is greater than 0, m ranges from 0 to numZeroMergeCand - 1, inclusive.		30
10. The variable mergeIdxOffset is set equal to 0.		
11. When mmvd_flag[xCb][yCb] is equal to 1, the variable mmvdCnt is set equal to 0 and the following applies until mmvdCnt is greater than (merge_idx[xCb][yCb] + mergeIdxOffset) or mmvdCnt is equal to MaxNumMergeCand:	– When candidate mergeCandList[mmvdCnt] uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1. – The variable mmvdCnt is incremented by 1.	40

【 0 1 4 5 】

【表 3】

12. The following assignments are made with N being the candidate at position $\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset}$ in the merging candidate list mergeCandList ($N = \text{mergeCandList}[\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset}]$) and X being replaced by 0 or 1:	
$\text{refIdxLX} = \text{refIdxLXN}$	(8-289)
$\text{predFlagLX}[0][0] = \text{predFlagLXN}$	(8-290)
$\text{mvLX}[0][0][0] = \text{mvLXN}[0]$	(8-291)
$\text{mvLX}[0][0][1] = \text{mvLXN}[1]$	(8-292)
$\text{gbiIdx} = \text{gbiIdxN}$	(8-293)
13. When $\text{mmvd_flag}[\text{xCb}][\text{yCb}]$ is equal to 1, the following applies:	
- The derivation process for merge motion vector difference as specified in 8.4.2.7 is invoked with the luma location (xCb, yCb), the luma motion vectors $\text{mvL0}[0][0]$, $\text{mvL1}[0][0]$, the reference indices refIdxL0, refIdxL1 and the prediction list utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$ as inputs, and the motion vector differences mMvdL0 and mMvdL1 as outputs. - The motion vector difference mMvdLX is added to the merge motion vectors mvLX for X being 0 and 1 as follows:	
$\text{mvLX}[0][0][0] += \text{mMvdLX}[0]$	(8-294)
$\text{mvLX}[0][0][1] += \text{mMvdLX}[1]$	(8-295)

10

20

【0146】

前記表 1 乃至表 3 は、一つの手続を示し得、表の順序に沿って前記手続が連続して行われ得る。前記手続は、マージモードのためのルマ成分の動きベクトルを導出する手続（8.4.2.2）を含むことができる。

【0147】

前記表 1 乃至表 3 を参照すると、 gbiIdx は、双予測加重値インデックスを示すことができ、 gbiIdxCol は、時間的マージ候補（例えば、マージ候補リスト内の時間的動きベクトルの候補）のための双予測加重値インデックスを示すことができ、前記マージモードのためのルマ成分の動きベクトルを導出する手続（8.4.2.2 で第 3 ステップ）で、前記 gbiIdxCol は 0 として導出されることができる。すなわち、時間的動きベクトルの候補の加重値インデックスは 0 として導出されることができる。

30

【0148】

或いは、例えば、時間的動きベクトルの候補に対する加重平均のための加重値インデックスは、コロケートッド（*collocated*）ブロックの加重値インデックスとして導出されることができる。ここで、コロケートッドブロックは、コル（*col*）ブロック、同じ位置ブロック又は同じ位置参照ブロックと呼ばれ得、コルブロックは参照ピクチャ上で現在ブロックと同じ位置のブロックを示すことができる。例えば、この場合、マージモードのためのルマ成分の動きベクトルを導出する手続は、次の表の通りである。

40

【0149】

50

【表 4】

8.4.2.2 Derivation process for luma motion vectors for merge mode

This process is only invoked when $\text{merge_flag}[\text{xCb}][\text{yCb}]$ is equal to 1, where (xCb, yCb) specify the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture.

Inputs to this process are:

- a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable cbWidth specifying the width of the current coding block in luma samples,
- a variable cbHeight specifying the height of the current coding block in luma samples.

Outputs of this process are:

- the luma motion vectors in 1/16 fractional-sample accuracy $\text{mvL0}[0][0]$ and $\text{mvL1}[0][0]$,
- the reference indices refIdxL0 and refIdxL1 ,
- the prediction list utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$,
- the bi-prediction weight index gbIdx .

The bi-prediction weight index gbIdx is set equal to 0.

The motion vectors $\text{mvL0}[0][0]$ and $\text{mvL1}[0][0]$, the reference indices refIdxL0 and refIdxL1 and the prediction utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$ are derived by the following ordered steps:

1. The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (xCb, yCb), the luma coding block width cbWidth, and the luma coding block height cbHeight as inputs, and the output being the availability flags availableFlagA_0 , availableFlagA_1 , availableFlagB_0 , availableFlagB_1 and availableFlagB_2 , the reference indices refIdxLXA_0 , refIdxLXA_1 , refIdxLXB_0 , refIdxLXB_1 and refIdxLXB_2 , the prediction list utilization flags predFlagLXA_0 , predFlagLXA_1 , predFlagLXB_0 , predFlagLXB_1 and predFlagLXB_2 , and the motion vectors mvLXA_0 , mvLXA_1 , mvLXB_0 , mvLXB_1 and mvLXB_2 , with X being 0 or 1, and the bi-prediction weight indices gbIdxA_0 , gbIdxA_1 , gbIdxB_0 , gbIdxB_1 , gbIdxB_2 .
2. The reference indices, refIdxLXCol , with X being 0 or 1, and the bi-prediction weight index gbIdxCol for the temporal merging candidate Col are set equal to 0.
3. The derivation process for temporal luma motion vector prediction as specified in in clause 8.4.2.11 is invoked with the luma location (xCb, yCb), the luma coding block width cbWidth, the luma coding block height cbHeight and the variable refIdxL0Col as inputs, and the output being the availability flag $\text{availableFlagL0Col}$ and the temporal motion vector mvL0Col . The variables availableFlagCol , predFlagL0Col and predFlagL1Col are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \quad (8-283)$$

$$\text{predFlagL0Col} = \text{availableFlagL0Col} \quad (8-284)$$

$$\text{predFlagL1Col} = 0 \quad (8-285)$$

$$\text{gbIdxCol} = 0 \quad (8-xxx)$$

4. When tile_group_type is equal to B, the derivation process for temporal luma motion vector prediction as specified in clause 8.4.2.11 is invoked with the luma location (xCb, yCb), the luma coding block width cbWidth, the luma coding block height cbHeight and the variable refIdxL1Col as inputs, and the output being the availability flag $\text{availableFlagL1Col}$ and the temporal motion vector mvL1Col . The variables availableFlagCol and predFlagL1Col are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \mid \mid \text{availableFlagL1Col} \quad (8-286)$$

$$\text{predFlagL1Col} = \text{availableFlagL1Col} \quad (8-287)$$

【 0 1 5 0 】

10

20

30

40

50

【表 5】

gbIdxCol = gbIdxCol	(x-xxx)
5. The merging candidate list, mergeCandList, is constructed as follows: <pre> i = 0 if(availableFlagA₁) mergeCandList[i++] = A₁ if(availableFlagB₁) mergeCandList[i++] = B₁ if(availableFlagB₀) mergeCandList[i++] = B₀ if(availableFlagA₀) mergeCandList[i++] = A₀ if(availableFlagB₂) mergeCandList[i++] = B₂ if(availableFlagCol) mergeCandList[i++] = Col </pre> 6. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the mergeCandList. 7. When numCurrMergeCand is less than (MaxNumMergeCand – 1) and NumHmvpCand is greater than 0, the following applies: – The derivation process of history-based merging candidates as specified in 8.4.2.6 is invoked with mergeCandList, and numCurrMergeCand as inputs, and modified mergeCandList and numCurrMergeCand as outputs. – numOrigMergeCand is set equal to numCurrMergeCand. 8. The derivation process for pairwise average merging candidates specified in clause 8.4.2.4 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList, numCurrMergeCand and numOrigMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0avgCand_k and refIdxL1avgCand_k, the prediction list utilization flags predFlagL0avgCand_k and predFlagL1avgCand_k and the motion vectors mvL0avgCand_k and mvL1avgCand_k of every new candidate avgCand_k being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate avgCand_k being added into mergeCandList is set equal to 0. The number of candidates being added, numAvgMergeCand, is set equal to (numCurrMergeCand – numOrigMergeCand). When numAvgMergeCand is greater than 0, k ranges from 0 to numAvgMergeCand – 1, inclusive. 9. The derivation process for zero motion vector merging candidates specified in clause 8.4.2.5 is invoked with the mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N of every candidate N in mergeCandList and numCurrMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0zeroCand_m and refIdxL1zeroCand_m, the prediction list utilization flags predFlagL0zeroCand_m and predFlagL1zeroCand_m and the motion vectors mvL0zeroCand_m and mvL1zeroCand_m of every new candidate zeroCand_m being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate zeroCand_m being added into mergeCandList is set equal to 0. The number of candidates being added, numZeroMergeCand, is set equal to (numCurrMergeCand – numOrigMergeCand – numAvgMergeCand). When numZeroMergeCand is greater than 0, m ranges from 0 to numZeroMergeCand – 1, inclusive. 10. The variable mergeIdxOffset is set equal to 0. 11. When mmvd_flag[xCb][yCb] is equal to 1, the variable mmvdCnt is set equal to 0 and the following applies until mmvdCnt is greater than (merge_idx[xCb][yCb] + mergeIdxOffset) or mmvdCnt is equal to MaxNumMergeCand: – When candidate mergeCandList[mmvdCnt] uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1.	

【 0 1 5 1 】

10

20

30

40

50

【表 6】

- The variable mmvdCnt is incremented by 1.	
12. The following assignments are made with N being the candidate at position $\text{merge_idx}[x\text{Cb}][y\text{Cb}] + \text{mergeIdxOffset}$ in the merging candidate list mergeCandList ($N = \text{mergeCandList}[\text{merge_idx}[x\text{Cb}][y\text{Cb}] + \text{mergeIdxOffset}]$) and X being replaced by 0 or 1:	
$\text{refIdxLX} = \text{refIdxLXN}$	(8-289)
$\text{predFlagLX}[0][0] = \text{predFlagLXN}$	(8-290)
$\text{mvLX}[0][0][0] = \text{mvLXN}[0]$	(8-291)
$\text{mvLX}[0][0][1] = \text{mvLXN}[1]$	(8-292)
$\text{gbiIdx} = \text{gbiIdxN}$	(8-293)
13. When $\text{mmvd_flag}[x\text{Cb}][y\text{Cb}]$ is equal to 1, the following applies:	
- The derivation process for merge motion vector difference as specified in 8.4.2.7 is invoked with the luma location ($x\text{Cb}, y\text{Cb}$), the luma motion vectors $\text{mvL0}[0][0]$, $\text{mvL1}[0][0]$, the reference indices refIdxL0, refIdxL1 and the prediction list utilization flags $\text{predFlagL0}[0][0]$ and $\text{predFlagL1}[0][0]$ as inputs, and the motion vector differences mMvdL0 and mMvdL1 as outputs. - The motion vector difference mMvdLX is added to the merge motion vectors mvLX for X being 0 and 1 as follows:	
$\text{mvLX}[0][0][0] += \text{mMvdLX}[0]$	(8-294)
$\text{mvLX}[0][0][1] += \text{mMvdLX}[1]$	(8-295)

10

20

【0152】

前記表 4 乃至表 6 は、一つの手続を示し得、表の順序に沿って前記手続が連続して行われ得る。前記手続は、マージモードのためのルマ成分の動きベクトルを導出する手続（8.4.2.2）を含むことができる。

【0153】

前記表 4 乃至表 6 を参照すると、 gbiIdx は、双予測加重値インデックスを示すことができ、 gbiIdxCol は、時間的マージ候補（例えば、マージ候補リスト内の時間的動きベクトルの候補）のための双予測加重値インデックスを示すことができ、前記マージモードのためのルマ成分の動きベクトルを導出する手続（8.4.2.2 で第 3 ステップ）で、前記 gbiIdxCol は 0 として導出されることができるが、スライスのタイプ又はタイルグループのタイプが B である場合（8.4.2.2 で第 4 ステップ）には、前記 gbiIdxCol が gbiIdxCol として導出されることができる。すなわち、時間的動きベクトルの候補の加重値インデックスは、コル（col）ブロックの加重値インデックスとして導出されることができる。

30

【0154】

一方、本文書の別の実施例によれば、サブブロック単位のマージモードのための動きベクトルの候補の構成時、時間的動きベクトルの候補が双予測を使用するとき、加重平均のための加重値インデックスを誘導又は導出できる。ここで、サブブロック単位のマージモードは、（サブブロック単位の）アフィンマージモードと呼ばれ得、時間的動きベクトルの候補は、サブブロックベースの時間的動きベクトルの候補を示すことができ、 SbTMVP 候補と呼ばれることもある。すなわち、インター予測タイプが双（bi）予測である場合、アフィンマージ候補リスト又はサブブロックのマージ候補リスト内の SbTMVP 候補（又はサブブロックベースの時間的動きベクトルの候補）のための加重値インデックス情報を誘導又は導出できる。

40

【0155】

例えば、サブブロックベースの時間的動きベクトルの候補に対する加重平均のための加

50

重値インデックスは、常時 0 として導出されることができる。ここで、加重値インデックスが 0 であることは、各参照方向（すなわち、双予測で L0 予測方向及び L1 予測方向）の加重値が同一であることを意味し得る。例えば、この場合、サブブロックマージモード内の動きベクトル及び参照インデックスを導出する手続又はサブブロックベースの時間的マージ候補を導出する手続は、次の表の通りである。

【 0 1 5 6 】

【表 7】

8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode

Inputs to this process are:

- a luma location (x_{Cb}, y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables $cbWidth$ and $cbHeight$ specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flag arrays $predFlagL0[xSbIdx][ySbIdx]$ and $predFlagL1[xSbIdx][ySbIdx]$,
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy $mvL0[xSbIdx][ySbIdx]$ and $mvL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy $mvCL0[xSbIdx][ySbIdx]$ and $mvCL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbIdx$.

The variables $numSbX$, $numSbY$ and the subblock merging candidate list, $subblockMergeCandList$ are derived by the following ordered steps:

1. When $sps_sbtmvp_enabled_flag$ is equal to 1, the following applies:

- The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (x_{Cb}, y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the luma coding block width as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1.
- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (x_{Cb}, y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$, the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ as inputs and the output being the availability flag $availableFlagSbCol$, the bi-prediction weight index $gbIdxSbCol$, the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$, the reference indices $refIdxLXSbCol$, the luma motion vectors $mvLXSbCol[xSbIdx][ySbIdx]$ and the prediction list utilization flags $predFlagLXSbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$ and X being 0 or 1.

2. When $sps_affine_enabled_flag$ is equal to 1, the sample locations (x_{NbA_0}, y_{NbA_0}), (x_{NbA_1}, y_{NbA_1}), (x_{NbA_2}, y_{NbA_2}), (x_{NbB_0}, y_{NbB_0}), (x_{NbB_1}, y_{NbB_1}), (x_{NbB_2}, y_{NbB_2}), (x_{NbB_3}, y_{NbB_3}), and the variables $numSbX$ and $numSbY$ are derived as follows:

$$(x_{A_0}, y_{A_0}) = (x_{Cb} - 1, y_{Cb} + cbHeight) \quad (8-536)$$

$$(x_{A_1}, y_{A_1}) = (x_{Cb} - 1, y_{Cb} + cbHeight - 1) \quad (8-537)$$

$$(x_{A_2}, y_{A_2}) = (x_{Cb} - 1, y_{Cb}) \quad (8-538)$$

$$(x_{B_0}, y_{B_0}) = (x_{Cb} + cbWidth, y_{Cb} - 1) \quad (8-539)$$

$$(x_{B_1}, y_{B_1}) = (x_{Cb} + cbWidth - 1, y_{Cb} - 1) \quad (8-540)$$

$$(x_{B_2}, y_{B_2}) = (x_{Cb} - 1, y_{Cb} - 1) \quad (8-541)$$

$$(x_{B_3}, y_{B_3}) = (x_{Cb}, y_{Cb} - 1) \quad (8-542)$$

$$numSbX = cbWidth >> 2 \quad (8-543)$$

【 0 1 5 7 】

10

20

30

40

50

【表 8】

$\text{numSbY} = \text{cbHeight} \gg 2 \quad (8-544)$	
3. When $\text{sps_affine_enabled_flag}$ is equal to 1, the variable availableFlagA is set equal to FALSE and the following applies for ($\text{xNbA}_k, \text{yNbA}_k$) from ($\text{xNbA}_0, \text{yNbA}_0$) to ($\text{xNbA}_1, \text{yNbA}_1$): - The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location ($\text{xCurr}, \text{yCurr}$) set equal to (xCb, yCb) and the neighbouring luma location ($\text{xNbA}_k, \text{yNbA}_k$) as inputs, and the output is assigned to the block availability flag availableA_k. - When availableA_k is equal to TRUE and $\text{MotionModelIdc}[\text{xNbA}_k][\text{yNbA}_k]$ is greater than 0 and availableFlagA is equal to FALSE, the following applies: - The variable availableFlagA is set equal to TRUE, motionModelIdcA is set equal to $\text{MotionModelIdc}[\text{xNbA}_k][\text{yNbA}_k]$, (xNb, yNb) is set equal to ($\text{CbPosX}[\text{xNbA}_k][\text{yNbA}_k], \text{CbPosY}[\text{xNbA}_k][\text{yNbA}_k]$), nbW is set equal to $\text{CbWidth}[\text{xNbA}_k][\text{yNbA}_k]$, nbH is set equal to $\text{CbHeight}[\text{xNbA}_k][\text{yNbA}_k]$, numCpMv is set equal to $\text{MotionModelIdc}[\text{xNbA}_k][\text{yNbA}_k] + 1$, and gbIdxA is set equal to $\text{GbIdx}[\text{xNbA}_k][\text{yNbA}_k]$. - For X being replaced by either 0 or 1, the following applies: - When $\text{PredFlagLX}[\text{xNbA}_k][\text{yNbA}_k]$ is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height ($\text{cbWidth}, \text{cbHeight}$), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates $\text{cpMvLXA}[\text{cpIdx}]$ with $\text{cpIdx} = 0 \dots \text{numCpMv} - 1$ as output. - The following assignments are made: $\text{predFlagLXA} = \text{PredFlagLX}[\text{xNbA}_k][\text{yNbA}_k] \quad (8-545)$ $\text{refIdxLXA} = \text{RefIdxLX}[\text{xNbA}_k][\text{yNbA}_k] \quad (8-546)$	10 20
4. When $\text{sps_affine_enabled_flag}$ is equal to 1, the variable availableFlagB is set equal to FALSE and the following applies for ($\text{xNbB}_k, \text{yNbB}_k$) from ($\text{xNbB}_0, \text{yNbB}_0$) to ($\text{xNbB}_2, \text{yNbB}_2$): - The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location ($\text{xCurr}, \text{yCurr}$) set equal to (xCb, yCb) and the neighbouring luma location ($\text{xNbB}_k, \text{yNbB}_k$) as inputs, and the output is assigned to the block availability flag availableB_k. - When availableB_k is equal to TRUE and $\text{MotionModelIdc}[\text{xNbB}_k][\text{yNbB}_k]$ is greater than 0 and availableFlagB is equal to FALSE, the following applies: - The variable availableFlagB is set equal to TRUE, motionModelIdcB is set equal to $\text{MotionModelIdc}[\text{xNbB}_k][\text{yNbB}_k]$, (xNb, yNb) is set equal to ($\text{CbPosX}[\text{xNbB}_k][\text{yNbB}_k], \text{CbPosY}[\text{xNbB}_k][\text{yNbB}_k]$), nbW is set equal to $\text{CbWidth}[\text{xNbB}_k][\text{yNbB}_k]$, nbH is set equal to $\text{CbHeight}[\text{xNbB}_k][\text{yNbB}_k]$, numCpMv is set equal to $\text{MotionModelIdc}[\text{xNbB}_k][\text{yNbB}_k] + 1$, and gbIdxB is set equal to $\text{GbIdx}[\text{xNbB}_k][\text{yNbB}_k]$. - For X being replaced by either 0 or 1, the following applies: - When $\text{PredFlagLX}[\text{xNbB}_k][\text{yNbB}_k]$ is equal to TRUE, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height ($\text{cbWidth}, \text{cbHeight}$), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates $\text{cpMvLXB}[\text{cpIdx}]$ with $\text{cpIdx} = 0 \dots \text{numCpMv} - 1$ as output. - The following assignments are made: $\text{predFlagLXB} = \text{PredFlagLX}[\text{xNbB}_k][\text{yNbB}_k] \quad (8-547)$	30 40

【 0 1 5 8 】

【表 9】

	$\text{refIdxLXB} = \text{RefIdxLX}[\text{xNbB}_k][\text{yNbB}_k]$	(8-548)	
5.	When <code>sps_affine_enabled_flag</code> is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location (<code>xCb</code> , <code>yCb</code>), the luma coding block width and height (<code>cbWidth</code> , <code>cbHeight</code>), the availability flags <code>availableA₀</code> , <code>availableA₁</code> , <code>availableA₂</code> , <code>availableB₀</code> , <code>availableB₁</code> , <code>availableB₂</code> , <code>availableB₃</code> as inputs, and the availability flags <code>availableFlagConstK</code> , the reference indices <code>refIdxLXConstK</code> , prediction list utilization flags <code>predFlagLXConstK</code> , motion model indices <code>motionModelIdxConstK</code> and <code>cpMvpLXConstK[cpIdx]</code> with <code>X</code> being 0 or 1, <code>K</code> = 1..6, <code>cpIdx</code> = 0..2 as outputs and <code>gbIdxConstK</code> is set equal to 0 with <code>K</code> = 1..6..		
6.	The initial subblock merging candidate list, <code>subblockMergeCandList</code> , is constructed as follows:		10
	<pre> i = 0 if(availableFlagSbCol) subblockMergeCandList[i++] = SbCol if(availableFlagA && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = A if(availableFlagB && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = B if(availableFlagConst1 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const1 if(availableFlagConst2 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const2 if(availableFlagConst3 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const3 if(availableFlagConst4 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const4 if(availableFlagConst5 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const5 if(availableFlagConst6 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const6 </pre>	(8-549)	20
7.	The variable <code>numCurrMergeCand</code> and <code>numOrigMergeCand</code> are set equal to the number of merging candidates in the <code>subblockMergeCandList</code> .		
8.	When <code>numCurrMergeCand</code> is less than <code>MaxNumSubblockMergeCand</code> , the following is repeated until <code>numCurrMergeCand</code> is equal to <code>MaxNumSubblockMergeCand</code> , with <code>mvZero[0]</code> and <code>mvZero[1]</code> both being equal to 0:		
–	The reference indices, the prediction list utilization flags and the motion vectors of <code>zeroCand_m</code> with <code>m</code> equal to (<code>numCurrMergeCand</code> – <code>numOrigMergeCand</code>) are derived as follows:		
	$\text{refIdxL0ZeroCand}_m = 0$	(8-550)	
	$\text{predFlagL0ZeroCand}_m = 1$	(8-551)	
	$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero}$	(8-552)	30
	$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero}$	(8-553)	
	$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero}$	(8-554)	
	$\text{refIdxL1ZeroCand}_m = (\text{tile_group_type} == B) ? 0 : -1$	(8-555)	
	$\text{predFlagL1ZeroCand}_m = (\text{tile_group_type} == B) ? 1 : 0$	(8-556)	
	$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero}$	(8-557)	
	$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero}$	(8-558)	
	$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero}$	(8-559)	
	$\text{motionModelIdxZeroCand}_m = 1$	(8-560)	
	$\text{gbIdxZeroCand}_m = 0$	(8-561)	
–	The candidate <code>zeroCand_m</code> with <code>m</code> equal to (<code>numCurrMergeCand</code> – <code>numOrigMergeCand</code>) is added at the end of <code>subblockMergeCandList</code> and <code>numCurrMergeCand</code> is incremented by 1 as follows:		40

【 0 1 5 9 】

【表 1 0】

$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$	
The variables refIdxL0, refIdxL1, $\text{predFlagL0}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{predFlagL1}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvL0}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvL1}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvCL0}[\text{xSbIdx}][\text{ySbIdx}]$, and $\text{mvCL1}[\text{xSbIdx}][\text{ySbIdx}]$ with $\text{xSbIdx} = 0..\text{numSbX} - 1$, $\text{ySbIdx} = 0..\text{numSbY} - 1$ are derived as follows:	
– If $\text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$ is equal to SbCol, the bi-prediction weight index gbiIdx is set equal to 0 and the following applies with X being 0 or 1:	
$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$	
– For $\text{xSbIdx} = 0..\text{numSbX} - 1$, $\text{ySbIdx} = 0..\text{numSbY} - 1$, the following applies:	
$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] \quad (8-564)$	
$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][0] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][0] \quad (8-565)$	
$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][1] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][1] \quad (8-566)$	
– When $\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}]$ is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$ and refIdxLX as inputs, and the output being $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$.	
– The following assignment is made for $x = \text{xCb}..\text{xCb} + \text{cbWidth} - 1$ and $y = \text{yCb}..\text{yCb} + \text{cbHeight} - 1$:	
$\text{MotionModelIdx}[x][y] = 0 \quad (8-567)$	
– Otherwise ($\text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$ is not equal to SbCol), the following applies with X being 0 or 1:	
– The following assignments are made with N being the candidate at position $\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]$ in the subblock merging candidate list $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$):	
$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$	
$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$	
$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$	
$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$	
$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$	
$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$	
$\text{gbiIdx} = \text{gbiIdxN} \quad (8-574)$	
– For $\text{xSbIdx} = 0..\text{numSbX} - 1$, $\text{ySbIdx} = 0..\text{numSbY} - 1$, the following applies:	
$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLX}[0][0] \quad (8-575)$	
– When $\text{predFlagLX}[0][0]$ is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location (xCb, yCb), the luma coding block width cbWidth, the luma prediction block height cbHeight, the number of control point motion vectors numCpMv, the control point motion vectors $\text{cpMvLX}[\text{cpIdx}]$ with cpIdx being 0..2, and the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY as inputs, the luma subblock motion vector array $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$ and the chroma subblock motion vector array $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$ with $\text{xSbIdx} = 0..\text{numSbX} - 1$, $\text{ySbIdx} = 0..\text{numSbY} - 1$ as outputs.	
– The following assignment is made for $x = \text{xCb}..\text{xCb} + \text{cbWidth} - 1$ and $y = \text{yCb}..\text{yCb} + \text{cbHeight} - 1$:	
$\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1 \quad (8-576)$	

【0 1 6 0】

10

20

30

40

50

【表 1 1】

8.4.4.3 Derivation process for subblock-based temporal merging candidates

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable $cbWidth$ specifying the width of the current coding block in luma samples,
- a variable $cbHeight$ specifying the height of the current coding block in luma samples.
- the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, and $availableFlagB_1$ of the neighbouring coding units,
- the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, and $refIdxLXB_1$ of the neighbouring coding units,
- the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, and $predFlagLXB_1$ of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, and $mvLXB_1$ of the neighbouring coding units.

Outputs of this process are:

- the availability flag $availableFlagSbCol$,
- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxLOSbCol$ and $refIdxLSbCol$,
- the luma motion vectors in 1/16 fractional-sample accuracy $mvLOSbCol[xSbIdx][ySbIdx]$ and $mvLSbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the prediction list utilization flags $predFlagLOSbCol[xSbIdx][ySbIdx]$ and $predFlagLSbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbiIdxSbCol$.

The $gbiIdxSbCol$ is set equal to 0.

...

【0 1 6 1】

前記表 7 乃至表 1 1 は、2つの手続を示し得、表の順序に沿って前記手続がそれぞれ連続して行われ得る。前記手続は、サブブロックマージモード内の動きベクトル及び参照インデックスを導出する手続（8.4.4.2）又はサブブロックベースの時間的マージ候補を導出する手続（8.4.4.3）を含むことができる。

【0 1 6 2】

前記表 7 乃至表 1 1 を参照すると、 $gbiIdx$ は、双予測加重値インデックスを示すことができ、 $gbiIdxSbCol$ は、サブブロックベースの時間的マージ候補（例えば、サブブロックベースのマージ候補リスト内の時間的動きベクトルの候補）のための双予測加重値インデックスを示すことができ、前記サブブロックベースの時間的マージ候補を導出する手続（8.4.4.3）で、前記 $gbiIdxSbCol$ は 0 として導出されることができる。すなわち、サブブロックベースの時間的動きベクトルの候補の加重値インデックスは 0 として導出されることができる。

【0 1 6 3】

或いは、例えば、サブブロックベースの時間的動きベクトルの候補に対する加重平均のための加重値インデックスは、時間的センター（center）ブロックの加重値インデックスとして導出されることができる。例えば、前記時間的センターブロックは、コル（col）ブロック又はコル（col）ブロックのセンターに位置したサブブロック又はサンプルを示すことができ、具体的には、コルブロックの中央 4 個のサブブロック又はサン

ブルのうち右下側に位置したサブブロック又はサンプルを示すことができる。例えば、この場合、サブブロックマージモード内の動きベクトル及び参照インデックスを導出する手続、サブブロックベースの時間的マージ候補を導出する手続又はサブブロックベースの時間的マージのためのベース動き情報を導出する手続は、次の表の通りである。

【 0 1 6 4 】

【表 1 2】

8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode

Inputs to this process are:

- a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables cbWidth and cbHeight specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY,
- the reference indices refIdxL0 and refIdxL1,
- the prediction list utilization flag arrays predFlagL0[xSbIdx][ySbIdx] and predFlagL1[xSbIdx][ySbIdx],
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy mvL0[xSbIdx][ySbIdx] and mvL1[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy mvCL0[xSbIdx][ySbIdx] and mvCL1[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1,
- the bi-prediction weight index gbIdx.

The variables numSbX, numSbY and the subblock merging candidate list, subblockMergeCandList are derived by the following ordered steps:

1. When sps_sbtmvp_enabled_flag is equal to 1, the following applies:

- The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (xCb, yCb), the luma coding block width cbWidth, the luma coding block height cbHeight and the luma coding block width as inputs, and the output being the availability flags availableFlagA0, availableFlagA1, availableFlagB0, availableFlagB1 and availableFlagB2, the reference indices refIdxLXA0, refIdxLXA1, refIdxLXB0, refIdxLXB1 and refIdxLXB2, the prediction list utilization flags predFlagLXA0, predFlagLXA1, predFlagLXB0, predFlagLXB1 and predFlagLXB2, and the motion vectors mvLXA0, mvLXA1, mvLXB0, mvLXB1 and mvLXB2, with X being 0 or 1.
- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (xCb, yCb), the luma coding block width cbWidth, the luma coding block height cbHeight, the availability flags availableFlagA0, availableFlagA1, availableFlagB0, availableFlagB1, the reference indices refIdxLXA0, refIdxLXA1, refIdxLXB0, refIdxLXB1, the prediction list utilization flags predFlagLXA0, predFlagLXA1, predFlagLXB0, predFlagLXB1 and the motion vectors mvLXA0, mvLXA1, mvLXB0, mvLXB1 as inputs and the output being the availability flag availableFlagSbCol, the bi-prediction weight index gbIdxSbCol, the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY, the reference indices refIdxLXSbCol, the luma motion vectors mvLXSbCol[xSbIdx][ySbIdx] and the prediction list utilization flags predFlagLXSbCol[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1 and X being 0 or 1.

2. When sps_affine_enabled_flag is equal to 1, the sample locations (xNbA0, yNbA0), (xNbA1, yNbA1), (xNbA2, yNbA2), (xNbB0, yNbB0), (xNbB1, yNbB1), (xNbB2, yNbB2), (xNbB3, yNbB3), and the variables numSbX and numSbY are derived as follows:

$$(xA_0, yA_0) = (xCb - 1, yCb + cbHeight) \quad (8-536)$$

$$(xA_1, yA_1) = (xCb - 1, yCb + cbHeight - 1) \quad (8-537)$$

$$(xA_2, yA_2) = (xCb - 1, yCb) \quad (8-538)$$

$$(xB_0, yB_0) = (xCb + cbWidth, yCb - 1) \quad (8-539)$$

$$(xB_1, yB_1) = (xCb + cbWidth - 1, yCb - 1) \quad (8-540)$$

$$(xB_2, yB_2) = (xCb - 1, yCb - 1) \quad (8-541)$$

$$(xB_3, yB_3) = (xCb, yCb - 1) \quad (8-542)$$

$$\text{numSbX} = \text{cbWidth} \gg 2 \quad (8-543)$$

【 0 1 6 5 】

10

20

30

40

50

【表 1 3】

$\text{numSbY} = \text{cbHeight} >> 2 \quad (8-544)$	
3. When $\text{sps_affine_enabled_flag}$ is equal to 1, the variable availableFlagA is set equal to FALSE and the following applies for (xNbA_k, yNbA_k) from (xNbA_0, yNbA_0) to (xNbA_1, yNbA_1): – The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbA_k, yNbA_k) as inputs, and the output is assigned to the block availability flag availableA_k. – When availableA_k is equal to TRUE and $\text{MotionModelIdc}[\text{xNbA}_k][\text{yNbA}_k]$ is greater than 0 and availableFlagA is equal to FALSE, the following applies: – The variable availableFlagA is set equal to TRUE, motionModelIdcA is set equal to $\text{MotionModelIdc}[\text{xNbA}_k][\text{yNbA}_k]$, (xNb, yNb) is set equal to ($\text{CbPosX}[\text{xNbA}_k][\text{yNbA}_k]$, $\text{CbPosY}[\text{xNbA}_k][\text{yNbA}_k]$), nbW is set equal to $\text{CbWidth}[\text{xNbA}_k][\text{yNbA}_k]$, nbH is set equal to $\text{CbHeight}[\text{xNbA}_k][\text{yNbA}_k]$, numCpMv is set equal to $\text{MotionModelIdc}[\text{xNbA}_k][\text{yNbA}_k] + 1$, and gbIdxA is set equal to $\text{GbIdx}[\text{xNbA}_k][\text{yNbA}_k]$. – For X being replaced by either 0 or 1, the following applies: – When $\text{PredFlagLX}[\text{xNbA}_k][\text{yNbA}_k]$ is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height (cbWidth, cbHeight), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates $\text{cpMvLXA}[\text{cpIdx}]$ with $\text{cpIdx} = 0 \dots \text{numCpMv} - 1$ as output. – The following assignments are made: $\text{predFlagLXA} = \text{PredFlagLX}[\text{xNbA}_k][\text{yNbA}_k] \quad (8-545)$ $\text{refIdxLXA} = \text{RefIdxLX}[\text{xNbA}_k][\text{yNbA}_k] \quad (8-546)$	10 20
4. When $\text{sps_affine_enabled_flag}$ is equal to 1, the variable availableFlagB is set equal to FALSE and the following applies for (xNbB_k, yNbB_k) from (xNbB_0, yNbB_0) to (xNbB_2, yNbB_2): – The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbB_k, yNbB_k) as inputs, and the output is assigned to the block availability flag availableB_k. – When availableB_k is equal to TRUE and $\text{MotionModelIdc}[\text{xNbB}_k][\text{yNbB}_k]$ is greater than 0 and availableFlagB is equal to FALSE, the following applies: – The variable availableFlagB is set equal to TRUE, motionModelIdcB is set equal to $\text{MotionModelIdc}[\text{xNbB}_k][\text{yNbB}_k]$, (xNb, yNb) is set equal to ($\text{CbPosX}[\text{xNbAB}][\text{yNbB}_k]$, $\text{CbPosY}[\text{xNbB}_k][\text{yNbB}_k]$), nbW is set equal to $\text{CbWidth}[\text{xNbB}_k][\text{yNbB}_k]$, nbH is set equal to $\text{CbHeight}[\text{xNbB}_k][\text{yNbB}_k]$, numCpMv is set equal to $\text{MotionModelIdc}[\text{xNbB}_k][\text{yNbB}_k] + 1$, and gbIdxB is set equal to $\text{GbIdx}[\text{xNbB}_k][\text{yNbB}_k]$. – For X being replaced by either 0 or 1, the following applies: – When $\text{PredFlagLX}[\text{xNbB}_k][\text{yNbB}_k]$ is equal to TRUE, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height (cbWidth, cbHeight), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates $\text{cpMvLXB}[\text{cpIdx}]$ with $\text{cpIdx} = 0 \dots \text{numCpMv} - 1$ as output. – The following assignments are made: $\text{predFlagLXB} = \text{PredFlagLX}[\text{xNbB}_k][\text{yNbB}_k] \quad (8-547)$	30 40

【 0 1 6 6 】

【表 1 4】

	$\text{refIdxLXB} = \text{RefIdxLX}[\text{xNbB}_k][\text{yNbB}_k]$	(8-548)
5.	When <code>sps_affine_enabled_flag</code> is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location (<code>xCb</code> , <code>yCb</code>), the luma coding block width and height (<code>cbWidth</code> , <code>cbHeight</code>), the availability flags <code>availableA₀</code> , <code>availableA₁</code> , <code>availableA₂</code> , <code>availableB₀</code> , <code>availableB₁</code> , <code>availableB₂</code> , <code>availableB₃</code> as inputs, and the availability flags <code>availableFlagConstK</code> , the reference indices <code>refIdxLXConstK</code> , prediction list utilization flags <code>predFlagLXConstK</code> , motion model indices <code>motionModelIdxConstK</code> and <code>cpMvpLXConstK[cpIdx]</code> with <code>X</code> being 0 or 1, <code>K</code> = 1..6, <code>cpIdx</code> = 0..2 as outputs and <code>gbIdxConstK</code> is set equal to 0 with <code>K</code> = 1..6..	
6.	The initial subblock merging candidate list, <code>subblockMergeCandList</code> , is constructed as follows:	
	<pre> i = 0 if(availableFlagSbCol) subblockMergeCandList[i++] = SbCol if(availableFlagA && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = A if(availableFlagB && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = B if(availableFlagConst1 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const1 if(availableFlagConst2 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const2 if(availableFlagConst3 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const3 if(availableFlagConst4 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const4 if(availableFlagConst5 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const5 if(availableFlagConst6 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const6 </pre>	(8-549)
7.	The variable <code>numCurrMergeCand</code> and <code>numOrigMergeCand</code> are set equal to the number of merging candidates in the <code>subblockMergeCandList</code> .	
8.	When <code>numCurrMergeCand</code> is less than <code>MaxNumSubblockMergeCand</code> , the following is repeated until <code>numCurrMergeCand</code> is equal to <code>MaxNumSubblockMergeCand</code> , with <code>mvZero[0]</code> and <code>mvZero[1]</code> both being equal to 0:	
–	The reference indices, the prediction list utilization flags and the motion vectors of <code>zeroCand_m</code> with <code>m</code> equal to (<code>numCurrMergeCand</code> – <code>numOrigMergeCand</code>) are derived as follows:	
	$\text{refIdxL0ZeroCand}_m = 0$	(8-550)
	$\text{predFlagL0ZeroCand}_m = 1$	(8-551)
	$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero}$	(8-552)
	$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero}$	(8-553)
	$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero}$	(8-554)
	$\text{refIdxL1ZeroCand}_m = (\text{tile_group_type} == B) ? 0 : -1$	(8-555)
	$\text{predFlagL1ZeroCand}_m = (\text{tile_group_type} == B) ? 1 : 0$	(8-556)
	$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero}$	(8-557)
	$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero}$	(8-558)
	$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero}$	(8-559)
	$\text{motionModelIdxZeroCand}_m = 1$	(8-560)
	$\text{gbIdxZeroCand}_m = 0$	(8-561)
–	The candidate <code>zeroCand_m</code> with <code>m</code> equal to (<code>numCurrMergeCand</code> – <code>numOrigMergeCand</code>) is added at the end of <code>subblockMergeCandList</code> and <code>numCurrMergeCand</code> is incremented by 1 as follows:	

【 0 1 6 7 】

10

20

30

40

【表 1 5】

$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$	
The variables refIdxL0, refIdxL1, $\text{predFlagL0}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{predFlagL1}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvL0}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvL1}[\text{xSbIdx}][\text{ySbIdx}]$, $\text{mvCL0}[\text{xSbIdx}][\text{ySbIdx}]$, and $\text{mvCL1}[\text{xSbIdx}][\text{ySbIdx}]$ with $\text{xSbIdx} = 0..\text{numSbX} - 1$, $\text{ySbIdx} = 0..\text{numSbY} - 1$ are derived as follows:	
– If $\text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$ is equal to SbCol, the bi-prediction weight index gbiIdx is set equal to 0 and the following applies with X being 0 or 1:	
$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$	
– For $\text{xSbIdx} = 0..\text{numSbX} - 1$, $\text{ySbIdx} = 0..\text{numSbY} - 1$, the following applies:	
$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] \quad (8-564)$	10
$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][0] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][0] \quad (8-565)$	
$\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}][1] = \text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}][1] \quad (8-566)$	
– When $\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}]$ is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$ and refIdxLX as inputs, and the output being $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$.	
– The following assignment is made for $\text{x} = \text{xCb}..\text{xCb} + \text{cbWidth} - 1$ and $\text{y} = \text{yCb}..\text{yCb} + \text{cbHeight} - 1$:	
$\text{MotionModelIdx}[\text{x}][\text{y}] = 0 \quad (8-567)$	
– Otherwise ($\text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$ is not equal to SbCol), the following applies with X being 0 or 1:	20
– The following assignments are made with N being the candidate at position $\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]$ in the subblock merging candidate list $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$):	
$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$	
$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$	
$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$	
$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$	
$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$	
$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$	
$\text{gbiIdx} = \text{gbiIdxN} \quad (8-574)$	
– For $\text{xSbIdx} = 0..\text{numSbX} - 1$, $\text{ySbIdx} = 0..\text{numSbY} - 1$, the following applies:	30
$\text{predFlagLX}[\text{xSbIdx}][\text{ySbIdx}] = \text{predFlagLX}[0][0] \quad (8-575)$	
– When $\text{predFlagLX}[0][0]$ is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location (xCb, yCb), the luma coding block width cbWidth, the luma prediction block height cbHeight, the number of control point motion vectors numCpMv, the control point motion vectors $\text{cpMvLX}[\text{cpIdx}]$ with cpIdx being 0..2, and the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY as inputs, the luma subblock motion vector array $\text{mvLX}[\text{xSbIdx}][\text{ySbIdx}]$ and the chroma subblock motion vector array $\text{mvCLX}[\text{xSbIdx}][\text{ySbIdx}]$ with $\text{xSbIdx} = 0..\text{numSbX} - 1$, $\text{ySbIdx} = 0..\text{numSbY} - 1$ as outputs.	
– The following assignment is made for $\text{x} = \text{xCb}..\text{xCb} + \text{cbWidth} - 1$ and $\text{y} = \text{yCb}..\text{yCb} + \text{cbHeight} - 1$:	
$\text{MotionModelIdx}[\text{x}][\text{y}] = \text{numCpMv} - 1 \quad (8-576)$	40

【0 1 6 8】

【表 1 6】

8.4.4.3 Derivation process for subblock-based temporal merging candidates

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable $cbWidth$ specifying the width of the current coding block in luma samples,
- a variable $cbHeight$ specifying the height of the current coding block in luma samples.
- the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, and $availableFlagB_1$ of the neighbouring coding units,
- the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, and $refIdxLXB_1$ of the neighbouring coding units,
- the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, and $predFlagLXB_1$ of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, and $mvLXB_1$ of the neighbouring coding units.

Outputs of this process are:

- the availability flag $availableFlagSbCol$,
- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0SbCol$ and $refIdxL1SbCol$,
- the luma motion vectors in 1/16 fractional-sample accuracy $mvL0SbCol[xSbIdx][ySbIdx]$ and $mvL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the prediction list utilization flags $predFlagL0SbCol[xSbIdx][ySbIdx]$ and $predFlagL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbIdxSbCol$.

The availability flag $availableFlagSbCol$ is derived as follows.

- If one or more of the following conditions is true, $availableFlagSbCol$ is set equal to 0.
 - $tile_group_temporal_mvp_enable_flag$ is equal to 0.
 - sps_sbtmvp_flag is equal to 0.
 - $cbWidth$ is less than 8.
 - $cbHeight$ is less than 8.
- Otherwise, the following ordered steps apply:
 1. The location (x_{Ctb} , y_{Ctb}) of the top-left sample of the luma coding tree block that contains the current coding block and the location (x_{Ctr} , y_{Ctr}) of the below-right center sample of the current luma coding block are derived as follows:

$$x_{Ctb} = (x_{Cb} \gg CtuLog2Size) \ll CtuLog2Size \quad (8-577)$$

$$y_{Ctb} = (y_{Cb} \gg CtuLog2Size) \ll CtuLog2Size \quad (8-578)$$

$$x_{Ctr} = x_{Cb} + (cbWidth / 2) \quad (8-579)$$

$$y_{Ctr} = y_{Cb} + (cbHeight / 2) \quad (8-580)$$
 2. The luma location ($x_{ColCtrCb}$, $y_{ColCtrCb}$) is set equal to the top-left sample of the collocated luma coding block covering the location given by (x_{Ctr} , y_{Ctr}) inside $ColPic$ relative to the top-left luma sample of the collocated picture specified by $ColPic$.
 3. The derivation process for subblock-based temporal merging base motion data as specified in clause 8.4.4.4 is invoked with the location (x_{Ctb} , y_{Ctb}), the location ($x_{ColCtrCb}$, $y_{ColCtrCb}$),

【 0 1 6 9 】

10

20

30

40

50

【表 1 7】

	the availability flags availableFlagA₀, availableFlagA₁, availableFlagB₀ and availableFlagB₁, and the prediction list utilization flags predFlagLXA₀, predFlagLXA₁, predFlagLXB₀ and predFlagLXB₁, and the reference indices refIdxLXA₀, refIdxLXA₁, refIdxLXB₀ and refIdxLXB₁, and the motion vectors mvLXA₀, mvLXA₁, mvLXB₀ and mvLXB₁, with X being 0 and 1 as inputs and the motion vectors ctrMvLX, the prediction list utilization flags ctrPredFlagLX and the reference indices ctrRefIdxLX of the collocated block, with X being 0 and 1, the bi-prediction weight index gbiIdxSbCol, and the temporal motion vector tempMV as outputs.	
4.	The variable availableFlagSbCol is derived as follows: – If both ctrPredFlagL0 and ctrPredFlagL1 are equal to 0, availableFlagSbCol is set equal to 0. – Otherwise, availableFlagSbCol is set equal to 1.	
	When availableFlagSbCol is equal to 1, the following applies:	10
–	The variables numSbX, numSbY, sbWidth, sbHeight and refIdxLXSbCol are derived as follows:	
	$\text{numSbX} = \text{cbWidth} \gg 3 \quad (8-581)$	
	$\text{numSbY} = \text{cbHeight} \gg 3 \quad (8-582)$	
	$\text{sbWidth} = \text{cbWidth} / \text{numSbX} \quad (8-583)$	
	$\text{sbHeight} = \text{cbHeight} / \text{numSbY} \quad (8-584)$	
	$\text{refIdxLXSbCol} = 0 \quad (8-585)$	
–	For xSbIdx = 0..numSbX – 1 and ySbIdx = 0..numSbY – 1, the motion vectors mvLXSbCol[xSbIdx][ySbIdx] and prediction list utilization flags predFlagLXSbCol[xSbIdx][ySbIdx] are derived as follows:	
	– The luma location (xSb, ySb) specifying the top-left sample of the current coding subblock relative to the top-left luma sample of the current picture is derived as follows:	20
	$\text{xSb} = \text{xCb} + \text{xSbIdx} * \text{sbWidth} \quad (8-586)$	
	$\text{ySb} = \text{yCb} + \text{ySbIdx} * \text{sbHeight} \quad (8-587)$	
	– The location (xColSb, yColSb) of the collocated subblock inside ColPic is derived as follows.	
	$\text{xColSb} = \text{Clip3}(\text{xCtb},$	
	$\text{Min}(\text{CurPicWidthInSamplesY} - 1, \text{xCtb} + (1 \ll \text{CtbLog2SizeY}) + 3), (8-588)$	
	$\text{xSb} + (\text{tempMv}[0] \gg 4))$	
	$\text{yColSb} = \text{Clip3}(\text{yCtb},$	
	$\text{Min}(\text{CurPicHeightInSamplesY} - 1, \text{yCtb} + (1 \ll \text{CtbLog2SizeY}) - 1), (8-589)$	30
	$\text{ySb} + (\text{tempMv}[1] \gg 4))$	
	– The variable currCb specifies the luma coding block covering the current coding subblock inside the current picture.	
	– The variable colCb specifies the luma coding block covering the modified location given by ((xColSb >> 3) << 3, (yColSb >> 3) << 3) inside the ColPic.	
	– The luma location (xColCb, yColCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.	
	– The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), refIdxL0 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock mvL0SbCol[xSbIdx][ySbIdx] and availableFlagL0SbCol.	40

【 0 1 7 0 】

【表 1 8】

– The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), refIdxL1 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock mvL1SbCol[xSbIdx][ySbIdx] and availableFlagL1SbCol. – When availableFlagL0SbCol and availableFlagL1SbCol are both equal to 0, the following applies for X being 0 and 1: $\text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrMvLX} \quad (8-590)$ $\text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrPredFlagLX} \quad (8-591)$	
8.4.4.4 Derivation process for subblock-based temporal merging base motion data Inputs to this process are: – the location (xCtb, yCtb) of the top-left sample of the luma coding tree block that contains the current coding block, – the location (xColCtb, yColCtb) of the top-left sample of the collocated luma coding block that covers the below-right center sample. – the availability flags availableFlagA₀, availableFlagA₁, availableFlagB₀, and availableFlagB₁ of the neighbouring coding units, – the reference indices refIdxLXA₀, refIdxLXA₁, refIdxLXB₀, and refIdxLXB₁ of the neighbouring coding units, – the prediction list utilization flags predFlagLXA₀, predFlagLXA₁, predFlagLXB₀, and predFlagLXB₁ of the neighbouring coding units, – the motion vectors in 1/16 fractional-sample accuracy mvLXA₀, mvLXA₁, mvLXB₀, and mvLXB₁ of the neighbouring coding units. Outputs of this process are: – the motion vectors ctrMvL0 and ctrMvL1, – the prediction list utilization flags ctrPredFlagL0 and ctrPredFlagL1, – the reference indices ctrRefIdxL0 and ctrRefIdxL1, – the temporal motion vector tempMV, – the bi-prediction weight index gbiIdxSbCol. The variable tempMv is set as follows: $\text{tempMv}[0] = 0 \quad (8-592)$ $\text{tempMv}[1] = 0 \quad (8-593)$ The variable currPic specifies the current picture. The variable availableFlagN is set equal to FALSE, and the following applies: – When availableFlagA₁ is equal to 1, the following applies: – availableFlagN is set equal to TRUE, – refIdxLXN is set equal to refIdxLXA₀ and mvLXN is set equal to mvLXA₀, for X being replaced by 0 and 1. – When availableFlagN is equal to FALSE and availableFlagLB₁ is equal to 1, the following applies: – availableFlagN is set equal to TRUE, – refIdxLXN is set equal to refIdxLXB₀ and mvLXN is set equal to mvLXB₀, for X being replaced by 0 and 1.	10 20 30 40

【 0 1 7 1 】

10

20

30

40

50

【表 19】

- When availableFlagN is equal to FALSE and availableFlagB₀ is equal to 1, the following applies: - availableFlagN is set equal to TRUE. - refIdxLXN is set equal to refIdxLXB₁ and mvLXN is set equal to mvLXB₁, for X being replaced by 0 and 1. - When availableFlagN is equal to FALSE and availableFlagA₀ is equal to 1, the following applies: - availableFlagN is set equal to TRUE. - refIdxLXN is set equal to refIdxLXA₁ and mvLXN is set equal to mvLXA₁, for X being replaced by 0 and 1.	
When availableFlagN is equal to TRUE, the following applies: - If all of the following conditions are true, tempMV is set equal to mvLIN: - predFlagLIN is equal to 1, - DiffPicOrderCnt(ColPic, RefPicList1[refIdxLIN]) is equal to 0, - DiffPicOrderCnt(aPic, currPic) is less than or equal to 0 for every picture aPic in every reference picture list of the current tile group, - tile_group_type is equal to B, - collocated_from_l0_flag is equal to 0. - Otherwise if all of the following conditions are true, tempMV is set equal to mvLON: - predFlagLON is equal to 1, - DiffPicOrderCnt(ColPic, RefPicList0[refIdxLON]) is equal to 0.	10
The location (xColCb, yColCb) of the collocated block inside ColPic is derived as follows. $xColCb = Clip3(xCtb, Min(CurPicWidthInSamplesY - 1, xCtb + (1 \ll CtbLog2SizeY) + 3), (8-594) xColCtCb + (tempMv[0] \gg 4))$ $yColCb = Clip3(yCtb, Min(CurPicHeightInSamplesY - 1, yCtb + (1 \ll CtbLog2SizeY) - 1), (8-595) yColCtCb + (tempMv[1] \gg 4))$ The array colPredMode is set equal to the prediction mode array CuPredMode of the collocated picture specified by ColPic. The motion vectors ctrMvL0 and ctrMvL1, the prediction list utilization flags ctrPredFlagL0 and ctrPredFlagL1, and the reference indices ctrRefIdxL0 and ctrRefIdxL1 are derived as follows:	20
- If colPredMode[xColCb][yColCb] is equal to MODE_INTER, the following applies: - The variable currCb specifies the luma coding block covering (xCtCb, yCtCb) inside the current picture. - The variable colCb specifies the luma coding block covering the modified location given by ((xColCb >> 3) << 3, (yColCb >> 3) << 3) inside the ColPic. - The luma location (xColCb, yColCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic. - The gbiIdxSbCol is set equal to gbiIdxcolCb. - The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), centerRefIdxL0, and sbFlag set equal to 1 as inputs and the output being assigned to ctrMvL0 and ctrPredFlagL0. - The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), centerRefIdxL1, and sbFlag set equal to 1 as inputs and the output being assigned to ctrMvL1 and ctrPredFlagL1.	30
	40

【0172】

【表 20】

– Otherwise, the following applies:	
ctrPredFlagL0 = 0	(8-596)
ctrPredFlagL1 = 0	(8-597)

【0173】

前記表12乃至表20は、3つの手続を示し得、表の順序に沿って前記手続がそれぞれ連続して行われ得る。前記手続は、サブブロックマージモード内の動きベクトル及び参照インデックスを導出する手続（8.4.4.2）、サブブロックベースの時間的マージ候補を導出する手続（8.4.4.3）又はサブブロックベースの時間的マージのためのベース動き情報を導出する手続（8.4.4.4）を含むことができる。

【0174】

前記表12乃至表20を参照すると、g b i I d x は、双予測加重値インデックスを示すことができ、g b i I d x S b C o l は、サブブロックベースの時間的マージ候補（例えば、サブブロックベースのマージ候補リスト内の時間的動きベクトル候補）のための双予測加重値インデックスを示すことができ、サブブロックベースの時間的マージのためのベース動き情報を導出する手続（8.4.4.4）で、前記 g b i I d x S b C o l は、g b i I d x c o l C b として導出されることができる。すなわち、サブブロックベースの時間的動きベクトル候補の加重値インデックスは、時間的センターブロックとして導出されることができる。例えば、前記時間的センターブロックは、コル（c o l）ブロック又はコル（c o l）ブロックのセンターに位置したサブブロック又はサンプルを示すことができ、具体的には、コルブロックの中央4個のサブブロック又はサンプルのうち右下側に位置したサブブロック又はサンプルを示すことができる。

【0175】

或いは、例えば、サブブロックベースの時間的動きベクトルの候補に対する加重平均のための加重値インデックスは、それぞれのサブブロック単位の加重値インデックスとして導出されることができ、サブブロックが利用可能ではない場合、時間的センター（c e n t e r）ブロックの加重値インデックスとして導出されることができる。例えば、前記時間的センターブロックは、コル（c o l）ブロック又はコル（c o l）ブロックのセンターに位置したサブブロック又はサンプルを示すことができ、具体的には、コルブロックの中央4個のサブブロック又はサンプルのうち右下側に位置したサブブロック又はサンプルを示すことができる。例えば、この場合、サブブロックマージモード内の動きベクトル及び参照インデックスを導出する手続、サブブロックベースの時間的マージ候補を導出する手続又はサブブロックベースの時間的マージのためのベース動き情報を導出する手続は、次の表の通りである。

【0176】

10

20

30

40

50

【表 2 1】

8.4.4.2 Derivation process for motion vectors and reference indices in subblock merge mode

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- two variables $cbWidth$ and $cbHeight$ specifying the width and the height of the luma coding block.

Outputs of this process are:

- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flag arrays $predFlagL0[xSbIdx][ySbIdx]$ and $predFlagL1[xSbIdx][ySbIdx]$,
- the luma subblock motion vector arrays in 1/16 fractional-sample accuracy $mvL0[xSbIdx][ySbIdx]$ and $mvL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the chroma subblock motion vector arrays in 1/32 fractional-sample accuracy $mvCL0[xSbIdx][ySbIdx]$ and $mvCL1[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbIdx$.

The variables $numSbX$, $numSbY$ and the subblock merging candidate list, $subblockMergeCandList$ are derived by the following ordered steps:

1. When $sps_sbmvp_enabled_flag$ is equal to 1, the following applies:

- The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the luma coding block width as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1.
- The derivation process for subblock-based temporal merging candidates as specified in clause 8.4.4.3 is invoked with the luma location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$, the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ as inputs and the output being the availability flag $availableFlagSbCol$, the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$, the reference indices $refIdxLXSbCol$, the bi-prediction weight index $gbIdxSbCol[xSbIdx][ySbIdx]$, the luma motion vectors $mvLXSbCol[xSbIdx][ySbIdx]$ and the prediction list utilization flags $predFlagLXSbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$ and X being 0 or 1.

2. When $sps_affine_enabled_flag$ is equal to 1, the sample locations (x_{NbA_0} , y_{NbA_0}), (x_{NbA_1} , y_{NbA_1}), (x_{NbA_2} , y_{NbA_2}), (x_{NbB_0} , y_{NbB_0}), (x_{NbB_1} , y_{NbB_1}), (x_{NbB_2} , y_{NbB_2}), (x_{NbB_3} , y_{NbB_3}), and the variables $numSbX$ and $numSbY$ are derived as follows:

$$(x_{A_0}, y_{A_0}) = (x_{Cb} - 1, y_{Cb} + cbHeight) \quad (8-536)$$

$$(x_{A_1}, y_{A_1}) = (x_{Cb} - 1, y_{Cb} + cbHeight - 1) \quad (8-537)$$

$$(x_{A_2}, y_{A_2}) = (x_{Cb} - 1, y_{Cb}) \quad (8-538)$$

$$(x_{B_0}, y_{B_0}) = (x_{Cb} + cbWidth, y_{Cb} - 1) \quad (8-539)$$

$$(x_{B_1}, y_{B_1}) = (x_{Cb} + cbWidth - 1, y_{Cb} - 1) \quad (8-540)$$

$$(x_{B_2}, y_{B_2}) = (x_{Cb} - 1, y_{Cb} - 1) \quad (8-541)$$

$$(x_{B_3}, y_{B_3}) = (x_{Cb}, y_{Cb} - 1) \quad (8-542)$$

$$numSbX = cbWidth >> 2 \quad (8-543)$$

【0 1 7 7】

10

20

30

40

50

【表 2 2】

numSbY = cbHeight >> 2	(8-544)	
3. When <code>sps_affine_enabled_flag</code> is equal to 1, the variable <code>availableFlagA</code> is set equal to <code>FALSE</code> and the following applies for (<code>xNbA_k</code> , <code>yNbA_k</code>) from (<code>xNbA₀</code> , <code>yNbA₀</code>) to (<code>xNbA₁</code> , <code>yNbA₁</code>):		
- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (<code>xCurr</code>, <code>yCurr</code>) set equal to (<code>xCb</code>, <code>yCb</code>) and the neighbouring luma location (<code>xNbA_k</code>, <code>yNbA_k</code>) as inputs, and the output is assigned to the block availability flag <code>availableA_k</code>.		
- When <code>availableA_k</code> is equal to <code>TRUE</code> and <code>MotionModelIdc[xNbA_k][yNbA_k]</code> is greater than 0 and <code>availableFlagA</code> is equal to <code>FALSE</code>, the following applies:		
- The variable <code>availableFlagA</code> is set equal to <code>TRUE</code>, <code>motionModelIdcA</code> is set equal to <code>MotionModelIdc[xNbA_k][yNbA_k]</code>, (<code>xNb</code>, <code>yNb</code>) is set equal to (<code>CbPosX[xNbA_k][yNbA_k]</code>, <code>CbPosY[xNbA_k][yNbA_k]</code>), <code>nbW</code> is set equal to <code>CbWidth[xNbA_k][yNbA_k]</code>, <code>nbH</code> is set equal to <code>CbHeight[xNbA_k][yNbA_k]</code>, <code>numCpMv</code> is set equal to <code>MotionModelIdc[xNbA_k][yNbA_k] + 1</code>, and <code>gbIdxA</code> is set equal to <code>GbiIdx[xNbA_k][yNbA_k]</code>.	10	
- For X being replaced by either 0 or 1, the following applies:		
- When <code>PredFlagLX[xNbA_k][yNbA_k]</code> is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (<code>xCb</code>, <code>yCb</code>), the luma coding block width and height (<code>cbWidth</code>, <code>cbHeight</code>), the neighbouring luma coding block location (<code>xNb</code>, <code>yNb</code>), the neighbouring luma coding block width and height (<code>nbW</code>, <code>nbH</code>), and the number of control point motion vectors <code>numCpMv</code> as input, the control point motion vector predictor candidates <code>cpMvLXA[cpIdx]</code> with <code>cpIdx = 0 .. numCpMv - 1</code> as output.		
- The following assignments are made:		20
$\text{predFlagLXA} = \text{PredFlagLX}[\text{xNbA}_k][\text{yNbA}_k]$	(8-545)	
$\text{refIdxLXA} = \text{RefIdxLX}[\text{xNbA}_k][\text{yNbA}_k]$	(8-546)	
4. When <code>sps_affine_enabled_flag</code> is equal to 1, the variable <code>availableFlagB</code> is set equal to <code>FALSE</code> and the following applies for (<code>xNbB_k</code> , <code>yNbB_k</code>) from (<code>xNbB₀</code> , <code>yNbB₀</code>) to (<code>xNbB₂</code> , <code>yNbB₂</code>):		
- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (<code>xCurr</code>, <code>yCurr</code>) set equal to (<code>xCb</code>, <code>yCb</code>) and the neighbouring luma location (<code>xNbB_k</code>, <code>yNbB_k</code>) as inputs, and the output is assigned to the block availability flag <code>availableB_k</code>.		
- When <code>availableB_k</code> is equal to <code>TRUE</code> and <code>MotionModelIdc[xNbB_k][yNbB_k]</code> is greater than 0 and <code>availableFlagB</code> is equal to <code>FALSE</code>, the following applies:		
- The variable <code>availableFlagB</code> is set equal to <code>TRUE</code>, <code>motionModelIdcB</code> is set equal to <code>MotionModelIdc[xNbB_k][yNbB_k]</code>, (<code>xNb</code>, <code>yNb</code>) is set equal to (<code>CbPosX[xNbAB][yNbB_k]</code>, <code>CbPosY[xNbB_k][yNbB_k]</code>), <code>nbW</code> is set equal to <code>CbWidth[xNbB_k][yNbB_k]</code>, <code>nbH</code> is set equal to <code>CbHeight[xNbB_k][yNbB_k]</code>, <code>numCpMv</code> is set equal to <code>MotionModelIdc[xNbB_k][yNbB_k] + 1</code>, and <code>gbIdxB</code> is set equal to <code>GbiIdx[xNbB_k][yNbB_k]</code>.	30	
- For X being replaced by either 0 or 1, the following applies:		
- When <code>PredFlagLX[xNbB_k][yNbB_k]</code> is equal to <code>TRUE</code>, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (<code>xCb</code>, <code>yCb</code>), the luma coding block width and height (<code>cbWidth</code>, <code>cbHeight</code>), the neighbouring luma coding block location (<code>xNb</code>, <code>yNb</code>), the neighbouring luma coding block width and height (<code>nbW</code>, <code>nbH</code>), and the number of control point motion vectors <code>numCpMv</code> as input, the control point motion vector predictor candidates <code>cpMvLXB[cpIdx]</code> with <code>cpIdx = 0 .. numCpMv - 1</code> as output.		
- The following assignments are made:		
$\text{predFlagLXB} = \text{PredFlagLX}[\text{xNbB}_k][\text{yNbB}_k]$	(8-547)	40

【 0 1 7 8 】

【表 2 3】

	$\text{refIdxLXB} = \text{RefIdxLX}[\text{xNbB}_k][\text{yNbB}_k]$	(8-548)	
5.	When <code>sps_affine_enabled_flag</code> is equal to 1, the derivation process for constructed affine control point motion vector merging candidates as specified in clause 8.4.4.6 is invoked with the luma coding block location (<code>xCb</code> , <code>yCb</code>), the luma coding block width and height (<code>cbWidth</code> , <code>cbHeight</code>), the availability flags <code>availableA₀</code> , <code>availableA₁</code> , <code>availableA₂</code> , <code>availableB₀</code> , <code>availableB₁</code> , <code>availableB₂</code> , <code>availableB₃</code> as inputs, and the availability flags <code>availableFlagConstK</code> , the reference indices <code>refIdxLXConstK</code> , prediction list utilization flags <code>predFlagLXConstK</code> , motion model indices <code>motionModelIdcConstK</code> and <code>cpMvpLXConstK[cpIdx]</code> with <code>X</code> being 0 or 1, <code>K = 1..6</code> , <code>cpIdx = 0..2</code> as outputs and <code>gbIdxConstK</code> is set equal to 0 with <code>K = 1..6</code> .		
6.	The initial subblock merging candidate list, <code>subblockMergeCandList</code> , is constructed as follows:		10
	<pre> i = 0 if(availableFlagSbCol) subblockMergeCandList[i++] = SbCol if(availableFlagA && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = A if(availableFlagB && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = B if(availableFlagConst1 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const1 if(availableFlagConst2 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const2 if(availableFlagConst3 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const3 if(availableFlagConst4 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const4 if(availableFlagConst5 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const5 if(availableFlagConst6 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const6 </pre>	(8-549)	20
7.	The variable <code>numCurrMergeCand</code> and <code>numOrigMergeCand</code> are set equal to the number of merging candidates in the <code>subblockMergeCandList</code> .		
8.	When <code>numCurrMergeCand</code> is less than <code>MaxNumSubblockMergeCand</code> , the following is repeated until <code>numCurrMergeCand</code> is equal to <code>MaxNumSubblockMergeCand</code> , with <code>mvZero[0]</code> and <code>mvZero[1]</code> both being equal to 0:		
–	The reference indices, the prediction list utilization flags and the motion vectors of <code>zeroCand_m</code> with <code>m</code> equal to (<code>numCurrMergeCand – numOrigMergeCand</code>) are derived as follows:		
	$\text{refIdxL0ZeroCand}_m = 0$	(8-550)	
	$\text{predFlagL0ZeroCand}_m = 1$	(8-551)	
	$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero}$	(8-552)	
	$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero}$	(8-553)	30
	$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero}$	(8-554)	
	$\text{refIdxL1ZeroCand}_m = (\text{tile_group_type} == B) ? 0 : -1$	(8-555)	
	$\text{predFlagL1ZeroCand}_m = (\text{tile_group_type} == B) ? 1 : 0$	(8-556)	
	$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero}$	(8-557)	
	$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero}$	(8-558)	
	$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero}$	(8-559)	
	$\text{motionModelIdcZeroCand}_m = 1$	(8-560)	
	$\text{gbIdxZeroCand}_m = 0$	(8-561)	
–	The candidate <code>zeroCand_m</code> with <code>m</code> equal to (<code>numCurrMergeCand – numOrigMergeCand</code>) is added at the end of <code>subblockMergeCandList</code> and <code>numCurrMergeCand</code> is incremented by 1 as follows:		40

【 0 1 7 9 】

【表 2 4】

subblockMergeCandList[numCurrMergeCand++] = zeroCand _m (8-562)	
The variables refIdxL0, refIdxL1, predFlagL0[xSbIdx][ySbIdx], predFlagL1[xSbIdx][ySbIdx], mvL0[xSbIdx][ySbIdx], mvL1[xSbIdx][ySbIdx], mvCL0[xSbIdx][ySbIdx], and mvCL1[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1 are derived as follows:	
- If subblockMergeCandList[merge_subblock_idx[xCb][yCb]] is equal to SbCol, the bi-prediction weight index gbiIdx is set equal to 0 and the following applies with X being 0 or 1:	
refIdxLX = refIdxLXSbCol (8-563)	
- For xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1, the following applies:	
predFlagLX[xSbIdx][ySbIdx] = predFlagLXSbCol[xSbIdx][ySbIdx] (8-564)	
mvLX[xSbIdx][ySbIdx][0] = mvLXSbCol[xSbIdx][ySbIdx][0] (8-565)	
mvLX[xSbIdx][ySbIdx][1] = mvLXSbCol[xSbIdx][ySbIdx][1] (8-566)	
- When predFlagLX[xSbIdx][ySbIdx], is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with mvLX[xSbIdx][ySbIdx] and refIdxLX as inputs, and the output being mvCLX[xSbIdx][ySbIdx].	
- The following assignment is made for x = xCb ..xCb + cbWidth - 1 and y = yCb..yCb + cbHeight - 1:	
MotionModelIdx[x][y] = 0 (8-567)	
- Otherwise (subblockMergeCandList[merge_subblock_idx[xCb][yCb]] is not equal to SbCol), the following applies with X being 0 or 1:	
- The following assignments are made with N being the candidate at position merge_subblock_idx[xCb][yCb] in the subblock merging candidate list subblockMergeCandList (N = subblockMergeCandList[merge_subblock_idx[xCb][yCb]]):	
refIdxLX = refIdxLXN (8-568)	
predFlagLX[0][0] = predFlagLXN (8-569)	
cpMvLX[0] = cpMvLXN[0] (8-570)	
cpMvLX[1] = cpMvLXN[1] (8-571)	
cpMvLX[2] = cpMvLXN[2] (8-572)	
numCpMv = motionModelIdxN + 1 (8-573)	
gbiIdx = gbiIdxN (8-574)	
- For xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1, the following applies:	
predFlagLX[xSbIdx][ySbIdx] = predFlagLX[0][0] (8-575)	
- When predFlagLX[0][0] is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location (xCb, yCb), the luma coding block width cbWidth, the luma prediction block height cbHeight, the number of control point motion vectors numCpMv, the control point motion vectors cpMvLX[cpIdx] with cpIdx being 0..2, and the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY as inputs, the luma subblock motion vector array mvLX[xSbIdx][ySbIdx] and the chroma subblock motion vector array mvCLX[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX - 1, ySbIdx = 0 .. numSbY - 1 as outputs.	
- The following assignment is made for x = xCb ..xCb + cbWidth - 1 and y = yCb..yCb + cbHeight - 1:	
MotionModelIdx[x][y] = numCpMv - 1 (8-576)	

【 0 1 8 0 】

10

20

30

40

【表 2 5】

8.4.4.3 Derivation process for subblock-based temporal merging candidates

Inputs to this process are:

- a luma location (x_{Cb} , y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- a variable $cbWidth$ specifying the width of the current coding block in luma samples,
- a variable $cbHeight$ specifying the height of the current coding block in luma samples.
- the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, and $availableFlagB_1$ of the neighbouring coding units,
- the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, and $refIdxLXB_1$ of the neighbouring coding units,
- the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, and $predFlagLXB_1$ of the neighbouring coding units,
- the motion vectors in 1/16 fractional-sample accuracy $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, and $mvLXB_1$ of the neighbouring coding units.

Outputs of this process are:

- the availability flag $availableFlagSbCol$,
- the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$,
- the reference indices $refIdxL0SbCol$ and $refIdxL1SbCol$,
- the luma motion vectors in 1/16 fractional-sample accuracy $mvL0SbCol[xSbIdx][ySbIdx]$ and $mvL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$,
- the bi-prediction weight index $gbiIdxSbCol[xSbIdx][ySbIdx]$, the prediction list utilization flags $predFlagL0SbCol[xSbIdx][ySbIdx]$ and $predFlagL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$.

The availability flag $availableFlagSbCol$ is derived as follows.

- If one or more of the following conditions is true, $availableFlagSbCol$ is set equal to 0.
 - $tile_group_temporal_mvp_enable_flag$ is equal to 0.
 - sps_sbtmvp_flag is equal to 0.
 - $cbWidth$ is less than 8.
 - $cbHeight$ is less than 8.
- Otherwise, the following ordered steps apply:
 1. The location (x_{Ctb} , y_{Ctb}) of the top-left sample of the luma coding tree block that contains the current coding block and the location (x_{Ctr} , y_{Ctr}) of the below-right center sample of the current luma coding block are derived as follows:

$$x_{Ctb} = (x_{Cb} \gg CtuLog2Size) \ll CtuLog2Size \quad (8-577)$$

$$y_{Ctb} = (y_{Cb} \gg CtuLog2Size) \ll CtuLog2Size \quad (8-578)$$

$$x_{Ctr} = x_{Cb} + (cbWidth / 2) \quad (8-579)$$

$$y_{Ctr} = y_{Cb} + (cbHeight / 2) \quad (8-580)$$
 2. The luma location ($x_{ColCtrCb}$, $y_{ColCtrCb}$) is set equal to the top-left sample of the collocated luma coding block covering the location given by (x_{Ctr} , y_{Ctr}) inside $ColPic$ relative to the top-left luma sample of the collocated picture specified by $ColPic$.
 3. The derivation process for subblock-based temporal merging base motion data as specified in clause 8.4.4.4 is invoked with the location (x_{Ctb} , y_{Ctb}), the location ($x_{ColCtrCb}$, $y_{ColCtrCb}$), the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$ and $availableFlagB_1$, and

【 0 1 8 1 】

10

20

30

40

50

【表 2 6】

the prediction list utilization flags predFlagLXA_0, predFlagLXA_1, predFlagLXB_0 and predFlagLXB_1, and the reference indices refIdxLXA_0, refIdxLXA_1, refIdxLXB_0 and refIdxLXB_1, and the motion vectors mvLXA_0, mvLXA_1, mvLXB_0 and mvLXB_1, with X being 0 and 1 as inputs and the motion vectors ctrMvLX, the prediction list utilization flags ctrPredFlagLX and the reference indices ctrRefIdxLX of the collocated block, with X being 0 and 1, the bi-prediction weight index ctrbiIdx, and the temporal motion vector tempMV as outputs.	
4. The variable $\text{availableFlagSbCol}$ is derived as follows: – If both ctrPredFlagL0 and ctrPredFlagL1 are equal to 0, $\text{availableFlagSbCol}$ is set equal to 0. – Otherwise, $\text{availableFlagSbCol}$ is set equal to 1.	
When $\text{availableFlagSbCol}$ is equal to 1, the following applies:	
– The variables numSbX, numSbY, sbWidth, sbHeight and refIdxLXSbCol are derived as follows:	10
$\text{numSbX} = \text{cbWidth} >> 3$	(8-581)
$\text{numSbY} = \text{cbHeight} >> 3$	(8-582)
$\text{sbWidth} = \text{cbWidth} / \text{numSbX}$	(8-583)
$\text{sbHeight} = \text{cbHeight} / \text{numSbY}$	(8-584)
$\text{refIdxLXSbCol} = 0$	(8-585)
– For $\text{xSbIdx} = 0.. \text{numSbX} - 1$ and $\text{ySbIdx} = 0.. \text{numSbY} - 1$, the motion vectors $\text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}]$ and prediction list utilization flags $\text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}]$ are derived as follows:	20
– The luma location (xSb, ySb) specifying the top-left sample of the current coding subblock relative to the top-left luma sample of the current picture is derived as follows:	
$\text{xSb} = \text{xCb} + \text{xSbIdx} * \text{sbWidth}$	(8-586)
$\text{ySb} = \text{yCb} + \text{ySbIdx} * \text{sbHeight}$	(8-587)
– The location (xColSb, yColSb) of the collocated subblock inside ColPic is derived as follows.	
$\begin{aligned} \text{xColSb} = & \text{Clip3}(\text{xCtb}, \\ & \text{Min}(\text{CurPicWidthInSamplesY} - 1, \text{xCtb} + (1 << \text{CtbLog2SizeY}) + 3), (8-58 \\ & 8) \\ & \text{xSb} + (\text{tempMv}[0] >> 4)) \end{aligned}$	
$\begin{aligned} \text{yColSb} = & \text{Clip3}(\text{yCtb}, \\ & \text{Min}(\text{CurPicHeightInSamplesY} - 1, \text{yCtb} + (1 << \text{CtbLog2SizeY}) - 1), (8-5 \\ & 89) \\ & \text{ySb} + (\text{tempMv}[1] >> 4)) \end{aligned}$	30
– The variable currCb specifies the luma coding block covering the current coding subblock inside the current picture.	
– The variable colCb specifies the luma coding block covering the modified location given by (($\text{xColSb} >> 3$) $<< 3$, ($\text{yColSb} >> 3$) $<< 3$) inside the ColPic.	
– The luma location (xColCb, yColCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.	
– The $\text{gbIdxSbCol}[\text{xSbIdx}][\text{ySbIdx}]$ is set equal to gbIdxcolCb.	
– The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, (xColCb, yColCb), refIdxL0 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock $\text{mvL0SbCol}[\text{xSbIdx}][\text{ySbIdx}]$ and $\text{availableFlagL0SbCol}$.	40

【 0 1 8 2 】

【表 2 7】

– The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with <code>currCb</code>, <code>colCb</code>, (<code>xColCb</code>, <code>yColCb</code>), <code>refIdxL1</code> set equal to 0 and <code>sbFlag</code> set equal to 1 as inputs and the output being assigned to the motion vector of the subblock <code>mvL1SbCol[xSbIdx][ySbIdx]</code> and <code>availableFlagL1SbCol</code>. – When <code>availableFlagL0SbCol</code> and <code>availableFlagL1SbCol</code> are both equal to 0, the following applies for <code>X</code> being 0 and 1: $\text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrMvLX} \quad (8-590)$ $\text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrPredFlagLX} \quad (8-591)$ $\text{gbIdxSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrgbIdx} \quad (\text{x-xxx})$	10
8.4.4.4 Derivation process for subblock-based temporal merging base motion data	
Inputs to this process are:	
– the location (<code>xCtb</code>, <code>yCtb</code>) of the top-left sample of the luma coding tree block that contains the current coding block, – the location (<code>xColCtb</code>, <code>yColCtb</code>) of the top-left sample of the collocated luma coding block that covers the below-right center sample. – the availability flags <code>availableFlagA₀</code>, <code>availableFlagA₁</code>, <code>availableFlagB₀</code>, and <code>availableFlagB₁</code> of the neighbouring coding units, – the reference indices <code>refIdxLXA₀</code>, <code>refIdxLXA₁</code>, <code>refIdxLXB₀</code>, and <code>refIdxLXB₁</code> of the neighbouring coding units, – the prediction list utilization flags <code>predFlagLXA₀</code>, <code>predFlagLXA₁</code>, <code>predFlagLXB₀</code>, and <code>predFlagLXB₁</code> of the neighbouring coding units, – the motion vectors in 1/16 fractional-sample accuracy <code>mvLXA₀</code>, <code>mvLXA₁</code>, <code>mvLXB₀</code>, and <code>mvLXB₁</code> of the neighbouring coding units.	20
Outputs of this process are:	
– the motion vectors <code>ctrMvL0</code> and <code>ctrMvL1</code>, – the prediction list utilization flags <code>ctrPredFlagL0</code> and <code>ctrPredFlagL1</code>, – the reference indices <code>ctrRefIdxL0</code> and <code>ctrRefIdxL1</code>, – the temporal motion vector <code>tempMV</code>, – the bi-prediction weight index <code>ctrgbIdx</code>.	
The variable <code>tempMv</code> is set as follows:	30
$\text{tempMv}[0] = 0 \quad (8-592)$	
$\text{tempMv}[1] = 0 \quad (8-593)$	
The variable <code>currPic</code> specifies the current picture.	
The variable <code>availableFlagN</code> is set equal to FALSE, and the following applies:	
– When <code>availableFlagA₁</code> is equal to 1, the following applies: – <code>availableFlagN</code> is set equal to TRUE, – <code>refIdxLXN</code> is set equal to <code>refIdxLXA₀</code> and <code>mvLXN</code> is set equal to <code>mvLXA₀</code>, for <code>X</code> being replaced by 0 and 1. – When <code>availableFlagN</code> is equal to FALSE and <code>availableFlagLB₁</code> is equal to 1, the following applies: – <code>availableFlagN</code> is set equal to TRUE,	40

【 0 1 8 3 】

【表 2 8】

- refIdxLXN is set equal to refIdxLXB₀ and mvLXN is set equal to mvLXB₀, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagB₀ is equal to 1, the following applies:
 - availableFlagN is set equal to TRUE.
 - refIdxLXN is set equal to refIdxLXB₁ and mvLXN is set equal to mvLXB₁, for X being replaced by 0 and 1.
- When availableFlagN is equal to FALSE and availableFlagA₀ is equal to 1, the following applies:
 - availableFlagN is set equal to TRUE.
 - refIdxLXN is set equal to refIdxLXA₁ and mvLXN is set equal to mvLXA₁, for X being replaced by 0 and 1.

10

When availableFlagN is equal to TRUE, the following applies:

- If all of the following conditions are true, tempMV is set equal to mvL1N:
 - predFlagL1N is equal to 1,
 - DiffPicOrderCnt(ColPic, RefPicList1[refIdxL1N]) is equal to 0,
 - DiffPicOrderCnt(aPic, currPic) is less than or equal to 0 for every picture aPic in every reference picture list of the current tile group,
 - tile_group_type is equal to B,
 - collocated_from_l0_flag is equal to 0.
- Otherwise if all of the following conditions are true, tempMV is set equal to mvL0N:
 - predFlagL0N is equal to 1,
 - DiffPicOrderCnt(ColPic, RefPicList0[refIdxL0N]) is equal to 0.

20

The location (xColCb, yColCb) of the collocated block inside ColPic is derived as follows.

$$\begin{aligned} \text{xColCb} &= \text{Clip3}(\text{xCtb}, \\ &\quad \text{Min}(\text{CurPicWidthInSamplesY} - 1, \text{xCtb} + (1 \ll \text{CtbLog2SizeY}) + 3), (8-59) \\ &\quad 4) \\ &\quad \text{xColCtrCb} + (\text{tempMv}[0] \gg 4)) \\ \text{yColCb} &= \text{Clip3}(\text{yCtb}, \\ &\quad \text{Min}(\text{CurPicHeightInSamplesY} - 1, \text{yCtb} + (1 \ll \text{CtbLog2SizeY}) - 1), (8-5) \\ &\quad 95) \\ &\quad \text{yColCtrCb} + (\text{tempMv}[1] \gg 4)) \end{aligned}$$

The array colPredMode is set equal to the prediction mode array CuPredMode of the collocated picture specified by ColPic.

30

The motion vectors ctrMvL0 and ctrMvL1, the prediction list utilization flags ctrPredFlagL0 and ctrPredFlagL1, and the reference indices ctrRefIdxL0 and ctrRefIdxL1 are derived as follows:

- If colPredMode[xColCb][yColCb] is equal to MODE_INTER, the following applies:
 - The variable currCb specifies the luma coding block covering (xCtrCb, yCtrCb) inside the current picture.
 - The variable colCb specifies the luma coding block covering the modified location given by ((xColCb >> 3) << 3, (yColCb >> 3) << 3) inside the ColPic.
 - The luma location (xColCb, yColCb) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.
 - The gbiIdxSbCol is set equal to ctrgbiIdx.

40

【 0 1 8 4】

【表 2 9】

- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with <code>currCb</code>, <code>colCb</code>, (<code>xColCb</code>, <code>yColCb</code>), <code>centerRefIdxL0</code>, and <code>sbFlag</code> set equal to 1 as inputs and the output being assigned to <code>ctrMvL0</code> and <code>ctrPredFlagL0</code>.	
- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with <code>currCb</code>, <code>colCb</code>, (<code>xColCb</code>, <code>yColCb</code>), <code>centerRefIdxL1</code>, and <code>sbFlag</code> set equal to 1 as inputs and the output being assigned to <code>ctrMvL1</code> and <code>ctrPredFlagL1</code>.	
- Otherwise, the following applies:	
$\text{ctrPredFlagL0} = 0$	(8-596)
$\text{ctrPredFlagL1} = 0$	(8-597)

10

【0185】

前記表 2 1 乃至表 2 9 は、3つの手続を示し得、表の順序に沿って前記手続がそれぞれ連続して行われ得る。前記手続は、サブブロックマージモード内の動きベクトル及び参照インデックスを導出する手続（8.4.4.2）、サブブロックベースの時間的マージ候補を導出する手続（8.4.4.3）又はサブブロックベースの時間的マージのためのベース動き情報を導出する手続（8.4.4.4）を含むことができる。

【0186】

20

前記表 2 1 乃至表 2 9 を参照すると、`gbiIdx` は、双予測加重値インデックスを示すことができ、`gbiIdxSbCol` は、サブブロックベースの時間的マージ候補（例えば、サブブロックベースのマージ候補リスト内の時間的動きベクトル候補）のための双予測加重値インデックスを示すことができ、サブブロックベースの時間的マージのためのベース動き情報を導出する手続（8.4.4.3）で、前記 `gbiIdxSbCol` は、`gbiIdxcolCb` として導出されることができる。或いは、条件（例えば、`availableFlagL0SbCol` 及び `availableFlagL1SbCol` が全て 0 である場合）によって、サブブロックベースの時間的マージのためのベース動き情報を導出する手続（8.4.4.3）で、前記 `gbiIdxSbCol` は、`ctrgbiIdx` として導出されることができ、サブブロックベースの時間的マージのためのベース動き情報を導出する手続（8.4.4.4）で、前記 `ctrgbiIdx` は、`gbiIdxSbCol` として導出されることができる。すなわち、サブブロックベースの時間的動きベクトルの候補の加重値インデックスは、それぞれのサブブロック単位の加重値インデックスとして導出されることができ、サブブロックが利用可能ではない場合、時間的センターブロックとして導出されることができる。例えば、前記時間的センターブロックは、コル（`col`）ブロック又はコル（`col`）ブロックのセンターに位置したサブブロック又はサンプルを示すことができ、具体的には、コルブロックの中央 4 個のサブブロック又はサンプルのうち右下側に位置したサブブロック又はサンプルを示すことができる。

30

【0187】

一方、本文書のまた別の実施例によれば、マージモードのための動きベクトルの候補の構成時、ペアワイズ（`pairwise`）候補の加重値インデックスを誘導又は導出することができる。言い換えると、マージ候補リストにペアワイズ候補が含まれ得、この場合、前記ペアワイズ候補の加重平均のための加重値インデックスが導出できる。例えば、前記ペアワイズ候補は、前記マージ候補リスト内の他のマージ候補に基づいて導出されることができ、前記ペアワイズ候補が双予測を使用する場合、加重平均のための加重値インデックスを導出することができる。すなわち、インター予測タイプが双（`bi`）予測である場合、マージ候補リスト内のペアワイズ候補のための加重値インデックス情報を誘導又は導出することができる。

40

【0188】

例えば、前記ペアワイズ候補が前記マージ候補リスト内の 2 個のマージ候補（例えば、

50

c a n d 0 及び c a n d 1) に基づいて導出されることができ、前記ペアワイズ候補が双予測を使用する場合、前記ペアワイズ候補の加重値インデックスは、前記マージ候補 c a n d 0 及び／又はマージ候補 c a n d 1 の加重値インデックスに基づいて導出されることができる。言い換えると、前記ペアワイズ候補の加重値インデックスは、前記ペアワイズ候補を導出するのに用いられたマージ候補のいずれかのマージ候補（例えば、マージ候補 c a n d 0 又はマージ候補 c a n d 1) の加重値インデックスとして導出されることができる。或いは、例えば、前記ペアワイズ候補の加重値インデックスは、前記ペアワイズ候補を導出するのに用いられたマージ候補（例えば、マージ候補 c a n d 0 及びマージ候補 c a n d 1) の加重値インデックスの特定の割合で導出されることができる。ここで、特定の割合は 1 : 1 であり得るが、異なる割合によって導出されることがもある。例えば、特定の割合は、デフォルト (d e f a u l t) 割合又はデフォルト値で決定されることができるが、これに限定されるわけではなく、デフォルト割合は 1 : 1 の割合で定義されることもあるが、異なる割合で定義されることもある。或いは、例えば、マージ候補の加重値インデックスの特定の割合に基づいてペアワイズ候補の加重値インデックスを導出する場合、特定の割合によって、前述したように、前記ペアワイズ候補の加重値インデックスがマージ候補のいずれかのマージ候補の加重値インデックスとして導出されることと同じ結果が導出されることもある。

【0189】

一方、本文書のまた別の実施例によれば、サブブロック単位のマージモードのための動きベクトルの候補の構成時、(代表) 動きベクトルの候補が双予測を使用するとき、加重平均のための加重値インデックスを誘導又は導出することができる。すなわち、インター予測タイプが双 (b i) 予測である場合、アフィンマージ候補リスト又はサブブロックのマージ候補リスト内の候補 (又はアフィンマージ候補) のための加重値インデックス情報を誘導又は導出することができる。

【0190】

例えば、アフィンマージ候補のうち、構成された (c o n s t r u c t e d) アフィンマージ候補は、現在ブロックの空間的 (s p a t i a l) に隣接したブロック (又は空間的周辺ブロック) 又は時間的 (t e m p o r a l) に隣接したブロック (又は時間的周辺ブロック) の動き情報に基づいて、C P 0、C P 1、C P 2 又は R B 候補を導出し、アフィンモデルとして M V F を導出する候補を示すことができる。例えば、C P 0 は、現在ブロックの左上側サンプルポジションで位置するコントロールポイントを示すことができ、C P 1 は、現在ブロックの右上側サンプルポジションに位置するコントロールポイントを示すことができ、C P 2 は、現在ブロックの左下側サンプルポジションに位置するコントロールポイントを示すことができる。また、R B は、現在ブロックの右下側サンプルポジションに位置するコントロールポイントを示すことができる。

【0191】

例えば、(代表) 動きベクトルの候補が構成されたアフィンマージ候補 (又は (現在) アフィンマージ候補) である場合、(現在) アフィンマージ候補の加重値インデックスは、C P 0 候補ブロックのうち、C P 0 での動きベクトルで決定されたブロックの加重値インデックスとして導出されることができる。或いは、(現在) アフィンマージ候補の加重値インデックスは、C P 1 候補ブロックのうち、C P 1 での動きベクトルで決定されたブロックの加重値インデックスとして導出されることができる。或いは、(現在) アフィンマージ候補の加重値インデックスは、C P 2 候補ブロックのうち、C P 2 での動きベクトルで決定されたブロックの加重値インデックスとして導出されることができる。或いは、(現在) アフィンマージ候補の加重値インデックスは、R B 候補ブロックのうち、R B での動きベクトルで決定されたブロックの加重値インデックスとして導出されることができる。或いは、(現在) アフィンマージ候補の加重値インデックスは、C P 0 での動きベクトルで決定されたブロックの加重値インデックス、C P 1 での動きベクトルで決定されたブロックの加重値インデックス、C P 2 での動きベクトルで決定されたブロックの加重値インデックス、又は R B での動きベクトルで決定されたブロックの加重値インデックスの

少なくとも一つに基づいて導出されることもある。例えば、複数の加重値インデックスに基づいて、前記（現在）アフィンマージ候補の加重値インデックスを導出する場合、前記複数の加重値インデックスに対する特定の割合が用いられることもある。ここで、特定の割合は、1：1、1：1：1又は1：1：1：1であり得るが、別の割合によって導出されることもある。例えば、特定の割合は、デフォルト（default）割合又はデフォルト値で決定されることができ、これに限定されるわけではなく、デフォルト割合は1：1の割合で定義され得るが、別の割合で定義されることもある。

【0192】

或いは、例えば、前記（現在）アフィンマージ候補の加重値インデックスは、各候補の加重値インデックスのうち、発生頻度が高い候補の加重値インデックスとして導出されることができ、例えば、CP0候補ブロックのうち、CP0での動きベクトルで決定された候補ブロックの加重値インデックス、CP1候補ブロックのうち、CP1での動きベクトルで決定された候補ブロックの加重値インデックス、CP2候補ブロックのうち、CP2での動きベクトルで決定された候補ブロックの加重値インデックス及び／又はRB候補ブロックのうち、RBでの動きベクトルで決定された候補ブロックの加重値インデックスのうち、最も多く重複する加重値インデックスを前記（現在）アフィンマージ候補の加重値インデックスとして導出することができる。

【0193】

例えば、前記コントロールポイントとしてCP0及びCP1が用いられてもよく、CP0、CP1及びCP2が用いられてもよく、RBは用いられなくてもよい。但し、例えば、アフィンブロック（アフィン予測モードでコーディングされるブロック）のRB候補を活用しようとする際には、前述した実施例で説明した時間的候補ブロックでの加重値インデックスを誘導又は導出する方法が活用できる。例えば、CP0、CP1、又はCP2は、現在ブロックの空間的周辺ブロックに基づいて候補を導出し、候補の中からCP0、CP1又はCP2での動きベクトル（すなわち、CPMV1、CPMV2又はCPMV3）として用いるブロックを決定することができる。或いは、例えば、RBは、現在ブロックの時間的周辺ブロックに基づいて候補を導出し、候補の中からRBでの動きベクトルとして用いるブロックを決定することができる。

【0194】

或いは、例えば、（代表）動きベクトルの候補がSbTMVP（又はATMVP）候補である場合、前記SbTMVP候補の加重値インデックスは、現在ブロックの左側の周辺ブロックの加重値インデックスとして導出されることができ、すなわち、SbTMVP（又はATMVP）に誘導された候補が双予測を使用する場合、現在ブロックの左側の周辺ブロックの加重値インデックスをサブブロックベースのマージモードのための加重値インデックスとして導出できる。すなわち、インター予測タイプが双（bi）予測である場合、アフィンマージ候補リスト又はサブブロックのマージ候補リスト内のSbTMVP候補のための加重値インデックス情報を誘導又は導出することができる。

【0195】

例えば、SbTMVP候補は、現在ブロックの空間的に隣接した左側ブロック（又は左側の周辺ブロック）を基準にコル（col）ブロックを導出することができるので、前記左側の周辺ブロックの加重値インデックスは信頼を受けることができると見ることができる。これによって、前記SbTMVP候補の加重値インデックスは、左側の周辺ブロックの加重値インデックスとして導出されることができ、

【0196】

図10及び11は、本文書の実施例に係るビデオ／映像エンコーディング方法及び関連のコンポーネントの一例を概略的に示す。

【0197】

図10で開示された方法は、図2又は図11で開示されたエンコーディング装置によって行われ得る。具体的に、例えば、図10のS1000乃至S1030は、図11の前記エンコーディング装置200の予測部220によって行われ得、図10のS1040は、

10

20

30

40

50

図11の前記エンコーディング装置200のエントロピーエンコーディング部240によって行われ得る。また、図10で示していないが、図11で前記エンコーディング装置200の予測部220により予測サンプル又は予測関連の情報を導出することができ、前記エンコーディング装置200のレジデュアル処理部230により原本サンプル又は予測サンプルからレジデュアル情報が導出でき、前記エンコーディング装置200のエントロピーエンコーディング部240により、レジデュアル情報又は予測関連の情報からビットストリームが生成できる。図10で開示された方法は、本文書で前述した実施例を含むことができる。

【0198】

図10を参照すると、エンコーディング装置は、現在ブロックのインター予測モードを決定し、インター予測モードを示すインター予測モード情報を生成することができる（S1000）。例えば、エンコーディング装置は、現在ブロックに適用するインター予測モードとして、マージモード、アフィン（マージ）モード又はサブブロックマージモードを決定することができ、これを示すインター予測モード情報を生成することができる。

【0199】

エンコーディング装置は、インター予測モードに基づいて現在ブロックのマージ候補リストを生成することができる（S1010）。例えば、エンコーディング装置は、決定したインター予測モードによってマージ候補リストを生成することができる。ここで、決定したインター予測モードがアフィンマージモード又はサブブロックマージモードである場合、前記マージ候補リストは、アフィンマージ候補リスト又はサブブロックのマージ候補リスト等のように呼ばれ得るが、簡略にマージ候補リストと呼ばれることもある。

【0200】

例えば、前記マージ候補リスト内の候補の数が最大の候補数になるまでマージ候補リストに候補が挿入できる。ここで、候補は、現在ブロックの動き情報（又は動きベクトル）を導出するための候補又は候補ブロックを示すことができる。例えば、候補ブロックは、現在ブロックの周辺ブロックに対する探索を通じて導出されることができる。例えば、周辺ブロックは、現在ブロックの空間的周辺ブロック及び／又は時間的周辺ブロックを含むことができ、空間的周辺ブロックが優先的に探索され、（空間的マージ）候補が導出でき、その後、時間的周辺ブロックが探索され、（時間的マージ）候補として導出されることができる。導出された候補は、前記マージ候補リストに挿入できる。例えば、前記マージ候補リストは、前記候補を挿入した後にも、前記マージ候補リスト内の候補の数が最大の候補数よりも少ない場合、追加の候補を挿入することができる。例えば、追加の候補は、履歴ベースマージ候補（history based merge candidate(s)）、ペアワイズ平均マージ候補（pair-wise average merge candidate(s)）、ATMVP、組み合わせられた双予測（combined bi-predictive）マージ候補（現在スライス／タイルグループのスライス／タイルグループのタイプがBタイプである場合）及び／又はゼロベクトルマージ候補の少なくとも一つを含むことができる。

【0201】

或いは、例えば、アフィンマージ候補リスト内の候補の数が最大の候補数になるまでアフィンマージ候補リストに候補が挿入できる。ここで、候補は、現在ブロックのCPMV（Control Point Motion Vector）を含むことができる。又は、前記候補は、前記CPMVを導出するための候補又は候補ブロックを示すこともできる。前記CPMVは、現在ブロックのCP（Control Point）での動きベクトルを示すことができる。例えば、CPは、2個、3個又は4個であり得、現在ブロックの左上側（又は左上側角）、右上側（又は右上側角）、左下側（又は左下側角）又は右下側（又は右下側角）の少なくとも一部に位置し得、（各）位置毎に一つのCPのみが存在し得る。

【0202】

例えば、候補は、現在ブロックの周辺ブロック（又は現在ブロックのCPの周辺ブロッ

10

20

30

40

50

ク) に対する探索を通じて導出されることができる。例えば、アフィンマージ候補リストは、承継された (*inherited*) アフィンマージの候補、構成された (*constructed*) アフィンマージの候補又はゼロ (*zero*) 動きベクトルの候補の少なくとも一つを含むことができる。例えば、アフィンマージ候補リストは、とりあえず前記承継されたアフィンマージ候補を挿入することができ、以降、構成されたアフィンマージ候補を挿入することができる。また、アフィンマージ候補リストに構成されたアフィンマージ候補まで挿入したが、前記アフィンマージ候補リスト内の候補の数が最大の候補数よりも小さい場合、残りはゼロ動きベクトルの候補で満たされることができる。ここで、ゼロ動きベクトルの候補は、ゼロベクトルと呼ばれることもある。例えば、アフィンマージ候補リストは、サンプル単位で動きベクトルが導出されるアフィンマージモードによるリストであってもよいが、サブブロック単位で動きベクトルが導出されるアフィンマージモードによるリストであってもよい。この場合、アフィンマージ候補リストは、サブブロックのマージ候補リストと呼ばれることもあり、サブブロックのマージ候補リストは、*SbTMVP* として導出された候補 (又は *SbTMVP* 候補) も含まれ得る。例えば、*SbTMVP* 候補がサブブロックのマージ候補リストに含まれる場合、サブブロックのマージ候補リスト内で承継されたアフィンマージ候補及び構成されたアフィンマージ候補より前に位置し得る。

【0203】

エンコーディング装置は、マージ候補リストに含まれた候補のうちの一つの候補を選択し、選択された候補を示す選択情報を生成することができる (*S1020*)。例えば、マージ候補リストは、空間的マージ候補、時間的マージ候補、ペアワイズ候補又はゼロベクトル候補の少なくとも一部を含むことができ、現在ブロックのインター予測のためにこのような候補のうちの一つの候補を選択することができる。或いは、例えば、サブブロックのマージ候補リストは、承継されたアフィンマージ候補、構成されたアフィンマージ候補、*SbTMVP* 候補又はゼロベクトル候補の少なくとも一部を含むことができ、現在ブロックのインター予測のために、このような候補のうちの一つの候補を選択することができる。

【0204】

例えば、前記選択情報は、前記マージ候補リスト内の選択された候補を示すインデックス情報を含むことができる。例えば、前記選択情報は、マージインデックス情報又はサブブロックマージインデックス情報と呼ばれることもある。

【0205】

エンコーディング装置は、現在ブロックのインター予測タイプを双予測 (*bi-prediction*) で示すインター予測タイプ情報を生成することができる (*S1030*)。例えば、前記現在ブロックのインター予測タイプは、*L0* 予測、*L1* 予測又は双 (*bi*) 予測のうち、双予測で決定されることができ、これを示すインター予測タイプ情報を生成することができる。ここで、*L0* 予測は、参照ピクチャリスト0ベースの予測を示すことができ、*L1* 予測は、参照ピクチャリスト1ベースの予測を示すことができ、双予測は、参照ピクチャリスト0及び参照ピクチャリスト1ベースの予測を示すことができる。例えば、エンコーディング装置は、前記インター予測タイプに基づいてインター予測タイプ情報を生成することができる。例えば、前記インター予測タイプ情報は、*inter_pred_idc* のシンタックス要素を含むことができる。

【0206】

エンコーディング装置は、インター予測モード情報、選択情報及びインター予測タイプ情報を含む映像情報をエンコーディングすることができる (*S1040*)。例えば、前記映像情報は、ビデオ情報と呼ばれることもある。前記映像情報は、本文書の前述した実施例に係る多様な情報を含むことができる。例えば、映像情報は、予測関連の情報又はレジデュアル関連の情報の少なくとも一部を含むことができる。例えば、前記予測関連の情報は、前記インター予測モード情報、選択情報及びインター予測タイプ情報の少なくとも一部を含むことができる。例えば、エンコーディング装置は、前述した情報 (又はシンタッ

10

20

30

40

50

クス要素)の全部または一部を含む映像情報をエンコーディングし、ビットストリーム又はエンコーディングされた情報を生成することができる。或いは、ビットストリームの形態で出力することができる。また、前記ビットストリーム又はエンコーディングされた情報は、ネットワーク又は格納媒体を介してデコーディング装置に送信されることができる。

【0207】

図10に示していないが、例えば、エンコーディング装置は、現在ブロックの予測サンプルを生成することができる。或いは、例えば、エンコーディング装置は、選択された候補に基づいて、現在ブロックの予測サンプルを生成することができる。或いは、例えば、エンコーディング装置は、選択された候補に基づいて動き情報を導出することができ、動き情報に基づいて現在ブロックの予測サンプルを生成することができる。例えば、エンコーディング装置は、双予測によってL0予測サンプル及びL1予測サンプルを生成することができ、前記L0予測サンプル及び前記L1予測サンプルに基づいて、現在ブロックの予測サンプルを生成することができる。この場合、双予測のための加重値インデックス情報(又は加重値情報)を用いて、前記L0予測サンプル及び前記L1予測サンプルから現在ブロックの予測サンプルを生成することができる。ここで、加重値情報は、前記加重値インデックス情報に基づいて示されることができる。

10

【0208】

言い換えると、例えば、エンコーディング装置は、選択された候補に基づいて、現在ブロックのL0予測サンプル及びL1予測サンプルを生成することができる。例えば、現在ブロックのインター予測タイプが双予測と決定された場合、現在ブロックの予測のために、参照ピクチャリスト0及び参照ピクチャリスト1が用いられることができる。例えば、前記L0予測サンプルは、前記参照ピクチャリスト0に基づいて導出された現在ブロックの予測サンプルを示すことができ、前記L1予測サンプルは、前記参照ピクチャリスト1に基づいて導出された現在ブロックの予測サンプルを示すことができる。

20

【0209】

例えば、前記候補は、空間的マージ候補を含むことができる。例えば、前記選択された候補が前記空間的マージ候補である場合、前記空間的マージ候補に基づいて、L0動き情報及びL1動き情報が導出でき、これに基づいて、前記L0予測サンプル及び前記L1予測サンプルが生成できる。

【0210】

例えば、前記候補は、時間的マージ候補を含むことができる。例えば、前記選択された候補が前記時間的マージ候補である場合、前記時間的マージ候補に基づいてL0動き情報及びL1動き情報が導出でき、これに基づいて、前記L0予測サンプル及び前記L1予測サンプルが生成できる。

30

【0211】

例えば、前記候補は、ペアワイズ(pair-wise)候補を含むことができる。例えば、前記選択された候補がペアワイズ候補である場合、前記ペアワイズ候補に基づいて、L0動き情報及びL1動き情報が導出でき、これに基づいて、前記L0予測サンプル及び前記L1予測サンプルが生成できる。例えば、前記ペアワイズ候補は、前記マージ候補リストに含まれた候補のうち、他の2個の候補に基づいて導出されることができる。

40

【0212】

或いは、例えば、前記マージ候補リストは、サブブロックのマージ候補リストであり得、アフィンマージ候補、サブブロックのマージ候補又はSBTMVP候補が選択されることもある。ここで、サブブロック単位のアフィンマージ候補は、サブブロックのマージ候補と呼ばれることもある。

【0213】

例えば、前記候補は、サブブロックのマージ候補を含むことができる。例えば、前記選択された候補が前記サブブロックのマージ候補である場合、前記サブブロックのマージ候補に基づいてL0動き情報及びL1動き情報が導出でき、これに基づいて、前記L0予測サンプル及び前記L1予測サンプルが生成できる。例えば、前記サブブロックのマージ候

50

補は、CPMV (Control Point Motion Vector) を含むことができ、前記L0予測サンプル及び前記L1予測サンプルは、前記CPMVに基づいてサブブロック単位で予測が行われて生成できる。

【0214】

ここで、CPMVは、前記現在ブロックのCP (Control Point) の周辺ブロックのうちの一つのブロックに基づいて示されることができる。例えば、CPは、2個、3個又は4個であり得、現在ブロックの左上側（又は左上側角）、右上側（又は右上側角）、左下側（又は左下側角）又は右下側（又は右下側角）の少なくとも一部に位置し得、（各）位置毎に一つのCPのみが存在し得る。

【0215】

例えば、前記CPが前記現在ブロックの左上側に位置するCP0であり得る。この場合、前記周辺ブロックは、前記現在ブロックの左上側角の周辺ブロック、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック及び前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図8で、A₂ブロック、B₂ブロック又はB₃ブロックを含むことができる。

【0216】

或いは、例えば、前記CPが前記現在ブロックの右上側に位置するCP1であり得る。この場合、前記周辺ブロックは、前記現在ブロックの右上側角の周辺ブロック及び前記右上側角の周辺ブロックの左側に隣接した上側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図8で、B₀ブロック又はB₁ブロックを含むことができる。

【0217】

或いは、例えば、前記CPが前記現在ブロックの左下側に位置するCP2であり得る。この場合、前記周辺ブロックは、前記現在ブロックの左下側角の周辺ブロック及び前記左下側角の周辺ブロックの上側に隣接した左側の周辺ブロックを含むことができる。若しくは、前記周辺ブロックは、図8で、A₀ブロック又はA₁ブロックを含むことができる。

【0218】

或いは、例えば、前記CPが前記現在ブロックの右下側に位置するCP3であり得る。ここで、CP3はRBと呼ばれることもある。この場合、前記周辺ブロックは、前記現在ブロックのコール (col) ブロック又は前記コールブロックの右下側角の周辺ブロックを含むことができる。ここで、前記コールブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができる。或いは、前記周辺ブロックは、図8で、Tブロックを含むことができる。

【0219】

或いは、例えば、前記候補は、SbTMVP候補を含むことができる。例えば、前記選択された候補が前記SbTMVP候補である場合、前記現在ブロックの左側の周辺ブロックに基づいてL0動き情報及びL1動き情報が導出でき、これに基づいて、前記L0予測サンプル及び前記L1予測サンプルが生成できる。例えば、前記L0予測サンプル及び前記L1予測サンプルは、サブブロック単位で予測が行われて生成されることができる。

【0220】

例えば、L0動き情報は、L0参照ピクチャのインデックス及びL0動きベクトル等を含むことができ、L1動き情報は、L1参照ピクチャのインデックス及びL1動きベクトル等を含むことができる。L0参照ピクチャのインデックスは、参照ピクチャリスト0で参照ピクチャを示す情報を含むことができ、L1参照ピクチャのインデックスは、参照ピクチャリスト1で参照ピクチャを示す情報を含むことができる。

【0221】

例えば、エンコーディング装置は、L0予測サンプル、L1予測サンプル及び加重値情報に基づいて、現在ブロックの予測サンプルを生成することができる。例えば、前記加重値情報は、加重値インデックス情報に基づいて示されることができる。前記加重値インデックス情報は、双予測のための加重値インデックス情報を示すことができる。例えば、前記加重値情報は、L0予測サンプル又はL1予測サンプルの加重平均のための情報を含む

10

20

30

40

50

ことができる。すなわち、前記加重値インデックス情報は、前記加重平均に用いられた加重値に対するインデックス情報を示すことができ、前記加重平均に基づいて予測サンプルを生成する手続で、加重値インデックス情報を生成することもできる。例えば、加重値インデックス情報は、3個又は5個の加重値のいずれかの加重値を示す情報を含むことができる。例えば、前記加重平均は、BCW (Bi-prediction with CU-level Weight) 又はBWA (Bi-prediction with Weighted Average) での加重平均を示すことができる。

【0222】

例えば、前記候補は、時間的マージ候補を含むことができ、前記加重値インデックス情報は0で示されることができる。すなわち、時間的マージ候補のための加重値インデックス情報は0で示されることができる。ここで、0である加重値インデックス情報は、各参照方向（すなわち、双予測でL0予測方向及びL1予測方向）の加重値が同一であることを示すことができる。或いは、例えば、前記候補は、時間的マージ候補を含むことができ、前記加重値インデックス情報は、コル (col) ブロックの加重値インデックス情報に基づいて示されることができる。すなわち、時間的マージ候補のための加重値インデックス情報は、コル (col) ブロックの加重値インデックス情報に基づいて示されることができる。ここで、前記コルブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができる。

【0223】

或いは、例えば、前記候補は、ペアワイズ (pair-wise) 候補を含むことができ、前記加重値インデックス情報は、前記ペアワイズ候補を導出するのに用いられたマージ候補リスト内の他の2個の候補のうちの一つの加重値インデックス情報で示されることができる。すなわち、前記ペアワイズ候補のための加重値インデックス情報は、前記ペアワイズ候補を導出するのに用いられたマージ候補リスト内の他の2個の候補のうちの一つの加重値インデックス情報で示されることができる。或いは、例えば、前記加重値インデックス情報は、前記2個の候補の加重値インデックス情報に基づいて示されることができる。

【0224】

或いは、例えば、前記マージ候補リストは、サブブロックのマージ候補リストであり得、アフィンマージ候補、サブブロックのマージ候補又はSbTMVP候補が選択されることもある。ここで、サブブロック単位のアフィンマージ候補は、サブブロックのマージ候補と呼ばれることもある。

【0225】

例えば、前記候補は、サブブロックのマージ候補を含むことができ、前記加重値インデックス情報は、現在ブロックのCPの周辺ブロックのうち、特定のブロックの加重値インデックス情報に基づいて示されることができる。すなわち、前記サブブロックのマージ候補のための加重値インデックス情報は、現在ブロックのCPの周辺ブロックのうち、特定のブロックの加重値インデックス情報に基づいて示されることができる。ここで、前記特定のブロックは、前記CPに対するCPMVの導出のために使用されるブロックであり得る。或いは、現在ブロックのCPの周辺ブロックのうち、CPMVとして用いられるMVを有するブロックであり得る。

【0226】

例えば、前記CPが前記現在ブロックの左上側に位置するCP0であり得る。この場合、前記加重値インデックス情報は、前記現在ブロックの左上側角の周辺ブロックの加重値インデックス情報、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロックの加重値インデックス情報、又は前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックの加重値インデックス情報に基づいて示されることができる。或いは、前記加重値インデックス情報は、図8で、A₂ブロックの加重値インデックス情報、B₂ブロックの加重値インデックス情報又はB₃ブロックの加重値インデックス情報に基づいて示されることができる。

【0227】

或いは、例えば、前記CPが前記現在ブロックの右上側に位置するCP1であり得る。この場合、前記加重値インデックス情報は、前記現在ブロックの右上側角の周辺ブロックの加重値インデックス情報、又は前記右上側角の周辺ブロックの左側に隣接した上側の周辺ブロックの加重値インデックス情報に基づいて示されることができる。或いは、前記加重値インデックス情報は、図8で、B₀ブロックの加重値インデックス情報、又はB₁ブロックの加重値インデックス情報に基づいて示されることができる。

【0228】

或いは、例えば、前記CPが前記現在ブロックの左下側に位置するCP2であり得る。この場合、前記加重値インデックス情報は、前記現在ブロックの左下側角の周辺ブロックの加重値インデックス情報、又は前記左下側角の周辺ブロックの上側に隣接した左側の周辺ブロックの加重値インデックス情報に基づいて示されることができる。或いは、前記加重値インデックス情報は、図8で、A₀ブロックの加重値インデックス情報、又はA₁ブロックの加重値インデックス情報に基づいて示されることができる。

【0229】

或いは、例えば、前記CPが前記現在ブロックの右下側に位置するCP3であり得る。ここで、CP3はRBと呼ばれることもある。この場合、前記加重値インデックス情報は、前記現在ブロックの科尔(c o l)ブロックの加重値インデックス情報、又は前記科尔ブロックの右下側角の周辺ブロックの加重値インデックス情報に基づいて示されることができる。ここで、前記科尔ブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができる。或いは、前記加重値インデックス情報は、図8で、Tブロックの加重値インデックス情報に基づいて示されることができる。

【0230】

或いは、例えば、前記CPは、複数のCPを含むことができる。例えば、複数のCPは、CP₀、CP₁、CP₂又はRBの少なくとも二つを含むことができる。この場合、前記加重値インデックス情報は、前記CPMVそれぞれの導出のために使用された特定のブロックの加重値インデックス情報のうち、最も多く重複する加重値インデックス情報に基づいて示されることができる。或いは、前記加重値インデックス情報は、前記特定のブロックの加重値インデックス情報のうち、発生頻度が最も高い加重値インデックス情報に基づいて示されることができる。すなわち、前記加重値インデックス情報は、複数のCPそれぞれに対するCPMVを導出するために使用された特定のブロックの加重値インデックス情報に基づいて示されることができる。

【0231】

或いは、例えば、前記候補は、S b T M V P 候補を含むことができ、前記加重値インデックス情報は、前記現在ブロックの左側の周辺ブロックの加重値インデックス情報に基づいて示されることができる。すなわち、S b T M V P 候補のための加重値インデックス情報は、前記左側の周辺ブロックの加重値インデックス情報に基づいて示されることができる。或いは、例えば、前記候補は、S b T M V P 候補を含むことができ、前記加重値インデックス情報は、0で示されることができる。すなわち、S b T M V P 候補のための加重値インデックス情報は、0で示されることができる。ここで、0である加重値インデックス情報は、各参照方向(すなわち、双予測でL₀予測方向及びL₁予測方向)の加重値が同一であることを示すことができる。或いは、例えば、前記候補は、S b T M V P 候補を含むことができ、前記加重値インデックス情報は、科尔(c o l)ブロック内のセンタースタックの加重値インデックス情報に基づいて示されることができる。すなわち、S b T M V P 候補のための加重値インデックス情報は、科尔(c o l)ブロック内のセンタースタックの加重値インデックス情報に基づいて示されることができる。ここで、前記科尔ブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができ、前記センタースタックは、前記科尔ブロックの中央に位置する4個のサブブロックのうち、右下側のサブブロックを含むことが

できる。或いは、例えば、前記候補は、S b T M V P 候補を含むことができ、前記加重値インデックス情報は、コル (c o l) ブロックのサブブロックそれぞれの加重値インデックス情報に基づいて示されることができる。すなわち、S b T M V P 候補のための加重値インデックス情報は、コル (c o l) ブロックのサブブロックそれぞれの加重値インデックス情報に基づいて示されることができる。

【0232】

或いは、図10に示していないが、例えば、エンコーディング装置は、前記予測サンプルと原本サンプルとに基づいて、レジデュアルサンプルを導出することができる。この場合、前記レジデュアルサンプルに基づいて、レジデュアル関連の情報が導出できる。前記レジデュアル関連の情報に基づいて、レジデュアルサンプルが導出できる。前記レジデュアルサンプル及び前記予測サンプルに基づいて、復元サンプルが生成できる。前記復元サンプルに基づいて、復元ブロック及び復元ピクチャが導出できる。或いは、例えば、エンコーディング装置は、レジデュアル関連の情報又は予測関連の情報を含む映像情報をエンコーディングすることができる。

【0233】

例えば、エンコーディング装置は、前述した情報（又はシンタックス要素）の全部又は一部を含む映像情報をエンコーディングし、ビットストリーム又はエンコーディングされた情報を生成することができる。或いは、ビットストリームの形態で出力できる。また、前記ビットストリーム又はエンコーディングされた情報は、ネットワーク又は格納媒体を介してデコーディング装置に送信されることができる。或いは、前記ビットストリーム又はエンコーディングされた情報は、コンピュータ読み取り可能な格納媒体に格納されることができる。前記ビットストリーム又は前記エンコーディングされた情報は、前述した映像エンコーディング方法により生成できる。

【0234】

図12及び図13は、本文書の実施例に係るビデオ／映像デコーディング方法及び関連のコンポーネントの一例を概略的に示す。

【0235】

図12で開示された方法は、図3又は図13で開示されたデコーディング装置によって行われ得る。具体的に、例えば、図12のS1200は、図13で前記デコーディング装置300のエントロピーデコーディング部310によって行われ得、図12のS1210乃至S1260は、図13で前記デコーディング装置300の予測部330によって行われ得る。また、図12で示していないが、図13で前記デコーディング装置300のエントロピーデコーディング部310によりビットストリームから予測関連の情報又はレジデュアル情報が導出でき、前記デコーディング装置300のレジデュアル処理部320によりレジデュアル情報からレジデュアルサンプルが導出でき、前記デコーディング装置300の予測部330により予測関連の情報から予測サンプルが導出でき、前記デコーディング装置300の加算部340によりレジデュアルサンプル又は予測サンプルから復元ブロック又は復元ピクチャが導出できる。図12で開示された方法は、本文書で前述した実施例を含むことができる。

【0236】

図12を参照すると、デコーディング装置は、ビットストリームを介してインター予測モード情報及びインター予測タイプ情報を含む映像情報を受信することができる（S1200）。例えば、前記映像情報は、ビデオ情報と呼ばれることもある。前記映像情報は、本文書の前述した実施例に係る多様な情報を含むことができる。例えば、映像情報は、予測関連の情報又はレジデュアル関連の情報の少なくとも一部を含むことができる。

【0237】

例えば、前記予測関連の情報は、インター予測モード情報又はインター予測タイプ情報を含むことができる。例えば、前記インター予測モード情報は、多様なインター予測モードの少なくとも一部を示す情報を含むことができる。例えば、マージモード、スキップモード、MVP (motion vector prediction) モード、アフィン (

10

20

30

40

50

A f f i n e) モード、サブブロックマージモード又はMMVD (merge with MVD) モード等の多様なモードが使用され得る。また、DMVR (Decoder side motion vector refinement) モード、AMVR (adaptive motion vector resolution) モード、BCW (Bi-prediction with CU-level weight) 又はBDOPF (Bi-directional optical flow) 等が付随的モードとして、さらに又は代わりに使用され得る。例えば、前記インター予測タイプ情報は、inter__pred__idcのシンタックス要素を含むことができる。或いは、前記インター予測タイプ情報は、L0予測、L1予測又は双(bi)予測のいずれかを示す情報を含むことができる。

10

【0238】

デコーディング装置は、インター予測モード情報に基づいて、現在ブロックのマージ候補リストを生成することができる(S1210)。例えば、デコーディング装置は、前記インター予測モード情報に基づいて、現在ブロックのインター予測モードをマージモード、アフィン(マージ)モード又はサブブロックマージモードと決定することができ、決定したインター予測モードによってマージ候補リストを生成することができる。ここで、インター予測モードがアフィンマージモード又はサブブロックマージモードと決定された場合、前記マージ候補リストは、アフィンマージ候補リスト又はサブブロックのマージ候補リスト等のように呼ばれ得るが、簡略にマージ候補リストと呼ばれることもある。

【0239】

例えば、前記マージ候補リスト内の候補の数が最大の候補数になるまでマージ候補リストに候補が挿入できる。ここで、候補は、現在ブロックの動き情報(又は動きベクトル)を導出するための候補又は候補ブロックを示すことができる。例えば、候補ブロックは、現在ブロックの周辺ブロックに対する探索を通じて導出できる。例えば、周辺ブロックは、現在ブロックの空間的周辺ブロック及び／又は時間的周辺ブロックを含むことができ、空間的周辺ブロックが優先的に探索され、(空間的マージ)候補が導出でき、以降、時間的周辺ブロックが探索され、(時間的マージ)候補として導出でき、導出された候補は、前記マージ候補リストに挿入されることができる。例えば、前記マージ候補リストは、前記候補を挿入した後も前記マージ候補リスト内の候補の数が最大の候補数よりも少ない場合、追加の候補を挿入することができる。例えば、追加の候補は、履歴ベースマージ候補(history based merge candidate(s))、ペアワイズ平均マージ候補(pair-wise average merge candidate(s))、ATMVP、組み合わせられた双予測(combined bi-predictive)マージ候補(現在スライス／タイルグループのスライス／タイルグループのタイプがBタイプである場合)及び／又はゼロベクトルマージ候補の少なくとも一つを含むことができる。

20

30

【0240】

或いは、例えば、アフィンマージ候補リスト内の候補の数が最大の候補数になるまでアフィンマージ候補リストに候補が挿入できる。ここで、候補は、現在ブロックのCPMV(Control Point Motion Vector)を含むことができる。或いは、前記候補は、前記CPMVを導出するための候補又は候補ブロックを示すこともできる。前記CPMVは、現在ブロックのCP(Control Point)での動きベクトルを示すことができる。例えば、CPは、2個、3個又は4個であり得、現在ブロックの左上側(又は左上側角)、右上側(又は右上側角)、左下側(又は左下側角)又は右下側(又は右下側角)の少なくとも一部に位置し得、(各)位置毎に一つのCPのみが存在し得る。

40

【0241】

例えば、候補ブロックは、現在ブロックの周辺ブロック(又は現在ブロックのCPの周辺ブロック)に対する探索を通じて導出できる。例えば、アフィンマージ候補リストは、承継された(inherited)アフィンマージ候補、構成された(constructed)

50

t e d) アフィンマージ候補又はゼロ (z e r o) 動きベクトル候補の少なくとも一つを含むことができる。例えば、アフィンマージ候補リストは、とりあえず前記承継されたアフィンマージ候補を挿入することができ、以降、構成されたアフィンマージ候補を挿入することができる。また、アフィンマージ候補リストに構成されたアフィンマージ候補まで挿入したが、前記アフィンマージ候補リスト内の候補の数が最大の候補数よりも小さい場合、残りはゼロ動きベクトルの候補で満たされることができる。ここで、ゼロ動きベクトルの候補は、ゼロベクトルと呼ばれることもある。例えば、アフィンマージ候補リストは、サンプル単位で動きベクトルが導出されるアフィンマージモードによるリストであり得るが、サブブロック単位で動きベクトルが導出されるアフィンマージモードによるリストであり得る。この場合、アフィンマージ候補リストは、サブブロックのマージ候補リストと呼ばれることもあり、サブブロックのマージ候補リストは、S b T M V P として導出された候補 (又は S b T M V P 候補) も含まれ得る。例えば、S b T M V P 候補がサブブロックのマージ候補リストに含まれる場合、サブブロックのマージ候補リスト内で承継されたアフィンマージ候補及び構成されたアフィンマージ候補よりも前に位置し得る。

【 0 2 4 2 】

デコーディング装置は、マージ候補リストに含まれた候補のうちの一つの候補を選択することができる (S 1 2 2 0) 。例えば、マージ候補リストは、空間的マージ候補、時間的マージ候補、ペアワイズ候補又はゼロベクトル候補の少なくとも一部を含むことができ、現在ブロックのインター予測のために、このような候補のうちの一つの候補を選択することができる。或いは、例えば、サブブロックのマージ候補リストは、承継されたアフィンマージ候補、構成されたアフィンマージ候補、S b T M V P 候補又はゼロベクトル候補の少なくとも一部を含むことができ、現在ブロックのインター予測のために、このような候補のうちの一つの候補を選択することができる。例えば、前記選択された候補は、選択情報に基づいて、前記マージ候補リストから選択されることができる。例えば、前記選択情報は、前記前記マージ候補リスト内の選択された候補を示すインデックス情報を含むことができる。例えば、前記選択情報は、マージインデックス情報又はサブブロックマージインデックス情報と呼ばれることもある。例えば、前記選択情報は、前記映像情報に含まれ得る。或いは、前記選択情報は、前記インター予測モード情報に含まれることもある。

【 0 2 4 3 】

デコーディング装置は、インター予測タイプ情報に基づいて、現在ブロックのインター予測タイプを双予測 (b i - p r e d i c t i o n) として導出できる (S 1 2 3 0) 。例えば、前記現在ブロックのインター予測タイプは、前記インター予測タイプ情報に基づいて、L 0 予測、L 1 予測又は双 (b i) 予測のうち、双予測として導出されることができる。ここで、L 0 予測は、参照ピクチャリスト 0 ベースの予測を示すことができ、L 1 予測は、参照ピクチャリスト 1 ベースの予測を示すことができ、双予測は、参照ピクチャリスト 0 及び参照ピクチャリスト 1 ベースの予測を示すことができる。例えば、前記インター予測タイプ情報は、i n t e r _ p r e d _ i d c のシンタックス要素を含むことができる。

【 0 2 4 4 】

デコーディング装置は、選択された候補に基づいて、現在ブロックの動き情報を導出することができる (S 1 2 4 0) 。例えば、デコーディング装置は、インター予測タイプが双予測として導出されることによって、選択された候補に基づいて、L 0 動き情報及びL 1 動き情報を導出することができる。例えば、L 0 動き情報は、L 0 参照ピクチャのインデックス及びL 0 動きベクトル等を含むことができ、L 1 動き情報は、L 1 参照ピクチャのインデックス及びL 1 動きベクトル等を含むことができる。L 0 参照ピクチャのインデックスは、参照ピクチャリスト 0 で参照ピクチャを示す情報を含むことができ、L 1 参照ピクチャインデックスは、参照ピクチャリスト 1 で参照ピクチャを示す情報を含むことができる。

【 0 2 4 5 】

デコーディング装置は、動き情報に基づいて、現在ブロックのL 0 予測サンプル及びL

10

20

30

40

50

1 予測サンプルを生成することができる (S 1 2 5 0)。例えば、現在ブロックのインター予測タイプが双予測として導出された場合、現在ブロックの予測のために、参照ピクチャリスト 0 及び参照ピクチャリスト 1 が用いられ得る。例えば、前記 L 0 予測サンプルは、前記参照ピクチャリスト 0 に基づいて導出された現在ブロックの予測サンプルを示すことができ、前記 L 1 予測サンプルは、前記参照ピクチャリスト 1 に基づいて導出された現在ブロックの予測サンプルを示すことができる。

【0 2 4 6】

例えば、前記候補は、空間的マージ候補を含むことができる。例えば、前記選択された候補が前記空間的マージ候補である場合、前記空間的マージ候補に基づいて、L 0 動き情報及び L 1 動き情報が導出でき、これに基づいて、前記 L 0 予測サンプル及び前記 L 1 予測サンプルが生成できる。

10

【0 2 4 7】

例えば、前記候補は、時間的マージ候補を含むことができる。例えば、前記選択された候補が前記時間的マージ候補である場合、前記時間的マージ候補に基づいて、L 0 動き情報及び L 1 動き情報が導出でき、これに基づいて、前記 L 0 予測サンプル及び前記 L 1 予測サンプルが生成できる。

【0 2 4 8】

例えば、前記候補は、ペアワイズ (pair-wise) 候補を含むことができる。例えば、前記選択された候補がペアワイズ候補である場合、前記ペアワイズ候補に基づいて、L 0 動き情報及び L 1 動き情報が導出でき、これに基づいて、前記 L 0 予測サンプル及び前記 L 1 予測サンプルが生成できる。例えば、前記ペアワイズ候補は、前記マージ候補リストに含まれた候補のうち、他の 2 個の候補に基づいて導出されることができる。

20

【0 2 4 9】

或いは、例えば、前記マージ候補リストは、サブブロックのマージ候補リストであり得、アフィンマージ候補、サブブロックのマージ候補又は S b T M V P 候補が選択されることもある。ここで、サブブロック単位のアフィンマージ候補は、サブブロックのマージ候補と呼ばれることもある。

【0 2 5 0】

例えば、前記候補は、サブブロックのマージ候補を含むことができる。例えば、前記選択された候補が前記サブブロックのマージ候補である場合、前記サブブロックのマージ候補に基づいて、L 0 動き情報及び L 1 動き情報が導出でき、これに基づいて、前記 L 0 予測サンプル及び前記 L 1 予測サンプルが生成できる。例えば、前記サブブロックのマージ候補は、CPMV (Control Point Motion Vector) を含むことができ、前記 L 0 予測サンプル及び前記 L 1 予測サンプルは、前記 CPMV に基づいてサブブロック単位で予測が行われて生成されることができる。

30

【0 2 5 1】

ここで、CPMV は、前記現在ブロックの CP (Control Point) の周辺ブロックのうちの一つのブロックに基づいて導出されることができる。例えば、CP は、2 個、3 個又は 4 個であり得、現在ブロックの左上側 (又は左上側角)、右上側 (又は右上側角)、左下側 (又は左下側角) 又は右下側 (又は右下側角) の少なくとも一部に位置し得、(各) 位置毎に一つの CP のみが存在し得る。

40

【0 2 5 2】

例えば、前記 CP が前記現在ブロックの左上側に位置する CP 0 であり得る。この場合、前記周辺ブロックは、前記現在ブロックの左上側角の周辺ブロック、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック及び前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図 8 で、A₂ ブロック、B₂ ブロック又は B₃ ブロックを含むことができる。

【0 2 5 3】

或いは、例えば、前記 CP が前記現在ブロックの右上側に位置する CP 1 であり得る。この場合、前記周辺ブロックは、前記現在ブロックの右上側角の周辺ブロック及び前記右

50

上側角の周辺ブロックの左側に隣接した上側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図8で、 B_0 ブロック又は B_1 ブロックを含むことができる。

【0254】

或いは、例えば、前記CPが前記現在ブロックの左下側に位置するCP2であり得る。この場合、前記周辺ブロックは、前記現在ブロックの左下側角の周辺ブロック及び前記左下側角の周辺ブロックの上側に隣接した左側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図8で、 A_0 ブロック又は A_1 ブロックを含むことができる。

【0255】

或いは、例えば、前記CPが前記現在ブロックの右下側に位置するCP3であり得る。ここで、CP3はRBと呼ばれることもある。この場合、前記周辺ブロックは、前記現在ブロックのCOLブロック又は前記COLブロックの右下側角の周辺ブロックを含むことができる。ここで、前記COLブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができる。或いは、前記周辺ブロックは、図8で、Tブロックを含むことができる。

【0256】

或いは、例えば、前記候補は、SbTMVP候補を含むことができる。例えば、前記選択された候補が前記SbTMVP候補である場合、前記現在ブロックの左側の周辺ブロックに基づいて、L0動き情報及びL1動き情報が導出でき、これに基づいて、前記L0予測サンプル及び前記L1予測サンプルが生成できる。例えば、前記L0予測サンプル及び前記L1予測サンプルは、サブブロック単位で予測が行われて生成されることができる。

【0257】

デコーディング装置は、L0予測サンプル、L1予測サンプル及び加重値情報に基づいて、現在ブロックの予測サンプルを生成することができる(S1260)。例えば、前記加重値情報は、加重値インデックス情報に基づいて導出されることができる。例えば、前記加重値情報は、L0予測サンプル又はL1予測サンプルの加重平均のための情報を含むことができる。すなわち、前記加重値インデックス情報は、前記加重平均に用いられた加重値に対するインデックス情報を示すことができ、前記加重平均は、加重値インデックス情報に基づいて行われ得る。例えば、加重値インデックス情報は、3個又は5個の加重値のいずれかの加重値を示す情報を含むことができる。例えば、前記加重平均は、BCW (Bi-prediction with CU-level Weight) 又はBWA (Bi-prediction with Weighted Average) での加重平均を示すことができる。

【0258】

例えば、前記候補は、時間的マージ候補を含むことができ、前記加重値インデックス情報は、0として導出されることができる。すなわち、時間的マージ候補のための加重値インデックス情報は、0として導出されることができる。ここで、0である加重値インデックス情報は、各参照方向(すなわち、双予測でL0予測方向及びL1予測方向)の加重値が同一であることを示すことができる。或いは、例えば、前記候補は、時間的マージ候補を含むことができ、前記加重値インデックス情報は、COL(c o l)ブロックの加重値インデックス情報に基づいて導出されることができる。すなわち、時間的マージ候補のための加重値インデックス情報は、COL(c o l)ブロックの加重値インデックス情報に基づいて導出されることができる。ここで、前記COLブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができる。

【0259】

或いは、例えば、前記候補は、ペアワイズ(pair-wise)候補を含むことができ、前記加重値インデックス情報は、前記ペアワイズ候補を導出するのに用いられたマージ候補リスト内の他の2個の候補のうちの1つの加重値インデックス情報として導出されることができる。すなわち、前記ペアワイズ候補のための加重値インデックス情報は、前記ペアワイズ候補を導出するのに用いられたマージ候補リスト内の他の2個の候補のうち

の一つの加重値インデックス情報として導出されることができる。或いは、例えば、前記加重値インデックス情報は、前記2個の候補の加重値インデックス情報に基づいて、導出されることもある。

【0260】

或いは、例えば、前記マージ候補リストは、サブブロックのマージ候補リストであり得、アフィンマージ候補、サブブロックのマージ候補又はS b T M V P候補が選択されることもある。ここで、サブブロック単位のアフィンマージ候補は、サブブロックのマージ候補と呼ばれることもある。

【0261】

例えば、前記候補は、サブブロックのマージ候補を含むことができ、前記加重値インデックス情報は、現在ブロックのC Pの周辺ブロックのうち、特定のブロックの加重値インデックス情報に基づいて導出されることができる。すなわち、前記サブブロックのマージ候補のための加重値インデックス情報は、現在ブロックのC Pの周辺ブロックのうち、特定のブロックの加重値インデックス情報に基づいて導出されることができる。ここで、前記特定のブロックは、前記C Pに対するC P M Vの導出のために使用されるブロックであり得る。或いは、現在ブロックのC Pの周辺ブロックのうち、C P M Vとして用いられるM Vを有するブロックであり得る。

【0262】

例えば、前記C Pが前記現在ブロックの左上側に位置するC P 0であり得る。この場合、前記加重値インデックス情報は、前記現在ブロックの左上側角の周辺ブロックの加重値インデックス情報、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロックの加重値インデックス情報又は前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックの加重値インデックス情報に基づいて導出されることができる。或いは、前記加重値インデックス情報は、図8で、A₂ブロックの加重値インデックス情報、B₂ブロックの加重値インデックス情報又はB₃ブロックの加重値インデックス情報に基づいて導出されることができる。

【0263】

或いは、例えば、前記C Pが前記現在ブロックの右上側に位置するC P 1であり得る。この場合、前記加重値インデックス情報は、前記現在ブロックの右上側角の周辺ブロックの加重値インデックス情報又は前記右上側角の周辺ブロックの下側に隣接した上側の周辺ブロックの加重値インデックス情報に基づいて導出されることができる。或いは、前記加重値インデックス情報は、図8で、B₀ブロックの加重値インデックス情報又はB₁ブロックの加重値インデックス情報に基づいて導出されることができる。

【0264】

或いは、例えば、前記C Pが前記現在ブロックの左下側に位置するC P 2であり得る。この場合、前記加重値インデックス情報は、前記現在ブロックの左下側角の周辺ブロックの加重値インデックス情報又は前記左下側角の周辺ブロックの上側に隣接した左側の周辺ブロックの加重値インデックス情報に基づいて導出されることができる。或いは、前記加重値インデックス情報は、図8で、A₀ブロックの加重値インデックス情報又はA₁ブロックの加重値インデックス情報に基づいて導出されることができる。

【0265】

或いは、例えば、前記C Pが前記現在ブロックの右下側に位置するC P 3であり得る。ここで、C P 3はR Bと呼ばれることもある。この場合、前記加重値インデックス情報は、前記現在ブロックのコール（c o l）ブロックの加重値インデックス情報又は前記コールブロックの右下側角の周辺ブロックの加重値インデックス情報に基づいて導出されることができる。ここで、前記コールブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができる。或いは、前記加重値インデックス情報は、図8で、Tブロックの加重値インデックス情報に基づいて導出されることができる。

【0266】

或いは、例えば、前記CPは、複数のCPを含むことができる。例えば、複数のCPは、CP0、CP1、CP2又はRBの少なくとも二つを含むことができる。この場合、前記加重値インデックス情報は、前記CPMVそれぞれの導出のために使用された特定のブロックの加重値インデックス情報のうち最も多く重複する加重値インデックス情報に基づいて導出されることができる。或いは、前記加重値インデックス情報は、前記特定のブロックの加重値インデックス情報のうち、発生頻度が最も高い加重値インデックス情報に基づいて導出されることができる。すなわち、前記加重値インデックス情報は、複数のCPそれぞれに対するCPMVを導出するために使用された特定のブロックの加重値インデックス情報に基づいて導出されることができる。

【0267】

或いは、例えば、前記候補は、SbTMVP候補を含むことができ、前記加重値インデックス情報は、前記現在ブロックの左側の周辺ブロックの加重値インデックス情報に基づいて導出されることができる。すなわち、SbTMVP候補のための加重値インデックス情報は、前記左側の周辺ブロックの加重値インデックス情報に基づいて導出されることができる。或いは、例えば、前記候補は、SbTMVP候補を含むことができ、前記加重値インデックス情報は、0として導出されることができる。すなわち、SbTMVP候補のための加重値インデックス情報は、0として導出されることができる。ここで、0である加重値インデックス情報は、各参照方向（すなわち、双予測でL0予測方向及びL1予測方向）の加重値が同一であることを示すことができる。或いは、例えば、前記候補は、SbTMVP候補を含むことができ、前記加重値インデックス情報は、コル（col）ブロック内のセンターブロックの加重値インデックス情報に基づいて導出されることができる。すなわち、SbTMVP候補のための加重値インデックス情報は、コル（col）ブロック内のセンターブロックの加重値インデックス情報に基づいて導出されることができる。ここで、前記コルブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができ、前記センターブロックは、前記コルブロックの中央に位置する4個のサブブロックのうち、右下側のサブブロックを含むことができる。或いは、例えば、前記候補は、SbTMVP候補を含むことができ、前記加重値インデックス情報は、コル（col）ブロックのサブブロックそれぞれの加重値インデックス情報に基づいて導出されることができる。すなわち、SbTMVP候補のための加重値インデックス情報は、コル（col）ブロックのサブブロックそれぞれの加重値インデックス情報に基づいて導出されることができる。

【0268】

図12に示していないが、例えば、デコーディング装置は、前記映像情報に含まれたレジデュアル関連の情報に基づいて、レジデュアルサンプルを導出することができる。また、デコーディング装置は、前記予測サンプル及び前記レジデュアルサンプルに基づいて、復元サンプルを生成することができる。前記復元サンプルに基づいて、復元ブロック及び復元ピクチャが導出できる。

【0269】

例えば、デコーディング装置は、ビットストリーム又はエンコーディングされた情報をデコーディングし、前述した情報（又はシンタックス要素）の全部又は一部を含む映像情報を獲得することができる。また、前記ビットストリーム又はエンコーディングされた情報は、コンピュータ読み取り可能な格納媒体に格納されることができ、前述したデコーディング方法が行われるように引き起こすことができる。

【0270】

前述した実施例において、方法は、一連のステップ又はブロックとしてフローチャートに基づいて説明されているが、該当実施例は、ステップの順序に限定されるわけではなく、あるステップは、前述したところと異なるステップと異なる順序で、又は同時に発生し得る。また、当業者であれば、フローチャートに示されているステップが排他的ではなく、異なるステップが含まれるか、フローチャートの一つ又はそれ以上のステップが本文書の実施例の範囲に影響を与えずに削除され得ることを理解することができる。

【0271】

前述した本文書の実施例に係る方法は、ソフトウェアの形態で具現されることができ、本文書に係るエンコーディング装置及び／又はデコーディング装置は、例えば、TV、コンピュータ、スマートフォン、セットトップボックス、ディスプレイ装置等の映像処理を行う装置に含まれ得る。

【0272】

本文書で、実施例がソフトウェアで具現される際、前述した方法は、前述した機能を行うモジュール（過程、機能等）で具現されることができる。モジュールはメモリに格納され、プロセッサにより実行されることができる。メモリは、プロセッサの内部又は外部にあってもよく、よく知られている様々な手段でプロセッサと連結されてもよい。プロセッサは、ASIC (application-specific integrated circuit)、他のチップセット、論理回路及び／又はデータ処理装置を含むことができる。メモリは、ROM (read-only memory)、RAM (random access memory)、フラッシュメモリ、メモリカード、格納媒体及び／又は他の格納装置を含むことができる。即ち、本文書で説明した実施例は、プロセッサ、マイクロプロセッサ、コントローラ又はチップ上で具現されて行われ得る。例えば、各図面で示している機能ユニットは、コンピュータ、プロセッサ、マイクロプロセッサ、コントローラ又はチップ上で具現されて行われ得る。この場合、具現のための情報 (ex. information on instructions) 又はアルゴリズムがデジタル格納媒体に格納されることができる。

【0273】

また、本文書の実施例が適用されるデコーディング装置及びエンコーディング装置は、マルチメディア放送受信装置、モバイル通信端末、ホームシネマビデオ装置、デジタルシネマビデオ装置、監視用カメラ、ビデオ対話装置、ビデオ通信のようなリアルタイム通信装置、モバイルストリーミング装置、格納媒体、カムコーダ、オーダーメイド型ビデオ (VoD) サービス提供装置、OTTビデオ (Over the top video) 装置、インターネットストリーミングサービス提供装置、3次元 (3D) ビデオ装置、VR (virtual reality) 装置、AR (argumente reality) 装置、映像電話ビデオ装置、運送手段端末 (ex. 車両 (自律走行車両含む) 端末、飛行機端末、船舶端末など) 及び医療用ビデオ装置等に含まれ得、ビデオ信号又はデータ信号を処理するために使用され得る。例えば、OTTビデオ (Over the top video) 装置としては、ゲームコンソール、ブルーレイプレーヤー、インターネットアクセスTV、ホームシアターシステム、スマートフォン、タブレットPC、DVR (Digital Video Recorder) 等を含み得る。

【0274】

また、本文書の実施例が適用される処理方法は、コンピュータで実行されるプログラムの形態で生産されることができ、コンピュータが読み取ることができる記録媒体に格納されることができる。本文書の実施例に係るデータ構造を有するマルチメディアデータもまた、コンピュータが読み取ることができる記録媒体に格納されることができる。前記コンピュータが読み取ることができる記録媒体は、コンピュータで読み取られるデータが格納される全ての種類の格納装置及び分散格納装置を含む。前記コンピュータが読み取ることができる記録媒体は、例えば、ブルーレイディスク (BD)、汎用直列バス (USB)、ROM、PROM、EPROM、EEPROM、RAM、CD-ROM、磁気テープ、フロッピーディスク、及び光学的データ格納装置を含み得る。また、前記コンピュータが読み取ることができる記録媒体は、搬送波 (例えば、インターネットを介した送信) の形態で具現されたメディアを含む。また、エンコーディング方法で生成されたビットストリームが、コンピュータが読み取られる記録媒体に格納されるか、有無線通信ネットワークを介して送信されることができる。

【0275】

また、本文書の実施例は、プログラムコードによるコンピュータプログラム製品で具現

されることができ、前記プログラムコードは、本文書の実施例によってコンピュータで実行されることができる。前記プログラムコードは、コンピュータによって読み取り可能なキャリア上に格納されることができる。

【0276】

図14は、本文書で開示された実施例が適用できるコンテンツストリーミングシステムの例を示す。

【0277】

図14を参照すると、本文書の実施例が適用されるコンテンツストリーミングシステムは、大きくエンコーディングサーバ、ストリーミングサーバ、ウェブサーバ、メディアストレージ、ユーザ装置、及びマルチメディア入力装置を含むことができる。

10

【0278】

前記エンコーディングサーバは、スマートフォン、カメラ、カムコーダ等のようなマルチメディア入力装置から入力されたコンテンツをデジタルデータに圧縮してビットストリームを生成し、これを前記ストリーミングサーバに送信する役割をする。別の例として、スマートフォン、カメラ、カムコーダ等のようなマルチメディア入力装置がビットストリームを直接生成する場合、前記エンコーディングサーバは省略され得る。

【0279】

前記ビットストリームは、本文書の実施例が適用されるエンコーディング方法又はビットストリームの生成方法により生成されることができ、前記ストリーミングサーバは、前記ビットストリームを送信又は受信する過程で、一時的に前記ビットストリームを格納することができる。

20

【0280】

前記ストリーミングサーバは、ウェブサーバを介してのユーザ要請に基づいてマルチメディアデータをユーザ装置に送信し、前記ウェブサーバは、ユーザにどのようなサービスがあるかを知らせる媒介体役割をする。ユーザが前記ウェブサーバに所望のサービスを要請すれば、前記ウェブサーバは、これをストリーミングサーバに伝達し、前記ストリーミングサーバは、ユーザにマルチメディアデータを送信する。このとき、前記コンテンツストリーミングシステムは、別の制御サーバを含むことができ、この場合、前記制御サーバは、前記コンテンツストリーミングシステム内の各装置間の命令／応答を制御する役割をする。

30

【0281】

前記ストリーミングサーバは、メディア格納所及び／又はエンコーディングサーバからコンテンツを受信することができる。例えば、前記エンコーディングサーバからコンテンツを受信ようになる場合、前記コンテンツをリアルタイムに受信することができる。この場合、円滑なストリーミングサービスを提供するために、前記ストリーミングサーバは、前記ビットストリームを一定時間の間、格納することができる。

【0282】

前記ユーザ装置の例では、携帯電話、スマートフォン (smart phone)、ノートブックコンピュータ (laptop computer)、デジタル放送用端末機、PDA (personal digital assistants)、PMP (portable multimedia player)、ナビゲーション、スレートPC (slate PC)、タブレットPC (tablet PC)、ウルトラブック (ultrabook)、ウェアラブルデバイス (wearable device、例えば、ウォッチ型端末機 (smartwatch)、グラス型端末機 (smart glass)、HMD (head mounted display))、デジタルTV、デスクトップコンピュータ、デジタルサイニジなどがありうる。

40

【0283】

前記コンテンツストリーミングシステム内の各サーバは、分散サーバとして運営されることができ、この場合、各サーバで受信するデータは、分散処理されることができる。

【0284】

50

本明細書に記載された請求項は、多様な方式で組み合わせることができる。例えば、本明細書の方法請求項の技術的特徴が組み合わせられて装置として具現されることができ、本明細書の装置請求項の技術的特徴が組み合わせられて方法として具現されることができ、本明細書の方法請求項の技術的特徴と装置請求項の技術的特徴とが組み合わせられて装置として具現されることができ、本明細書の方法請求項の技術的特徴と装置請求項の技術的特徴とが組み合わせられて方法として具現されることができ。

【図面】

【図 1】

【図 2】

図 1

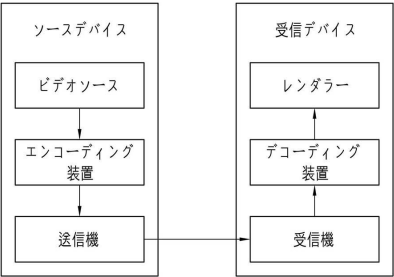
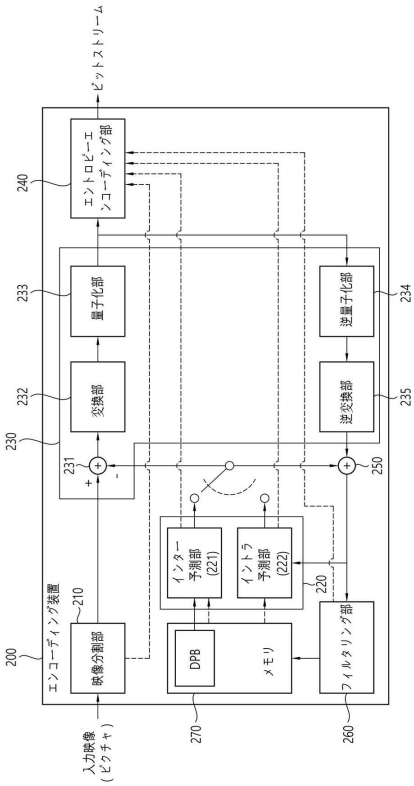


図 2



10

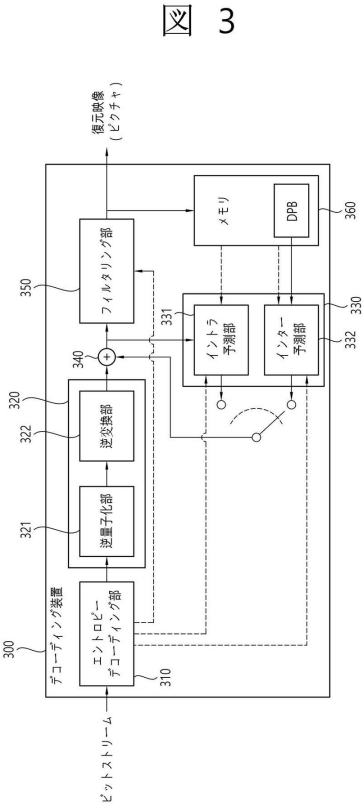
20

30

40

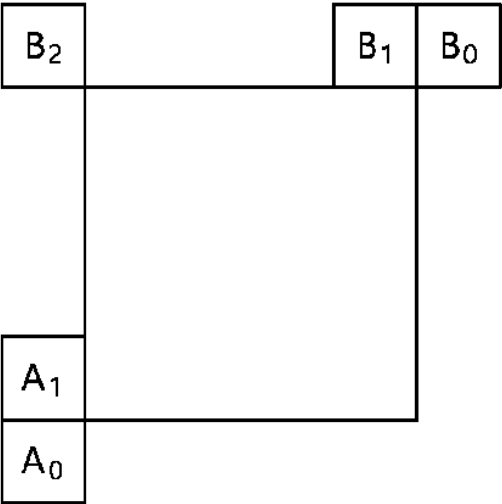
50

【図 3】



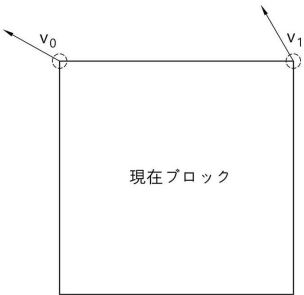
【図 4】

[図4]



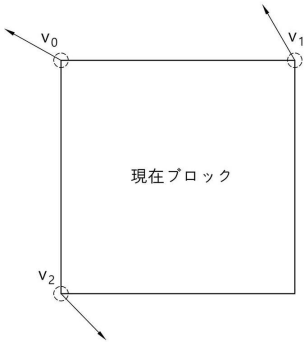
【図 5 A】

図 5A



【図 5 B】

図 5B



10

20

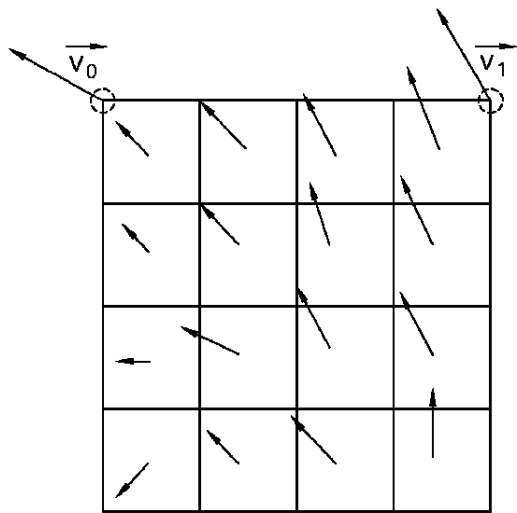
30

40

50

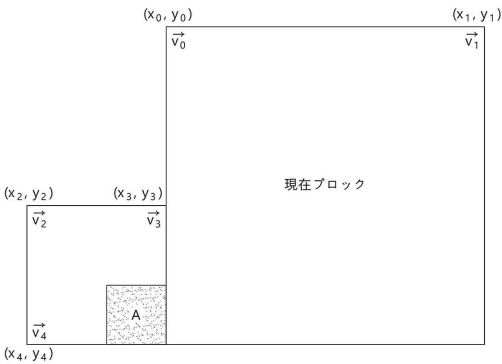
【図 6】

[図6]



【図 7】

図 7



10

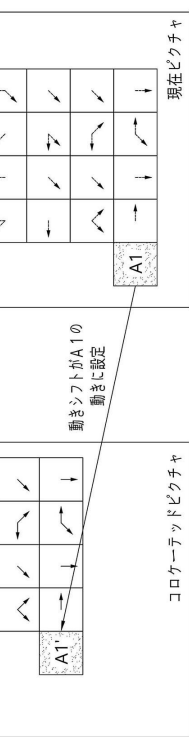
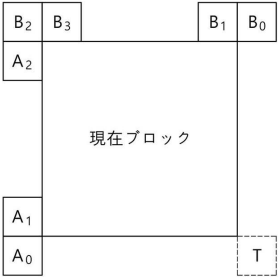
20

【図 8】

【図 9】

図 9

図 8



30

40

50

【図 1 0】

【図 1 1】

図 10

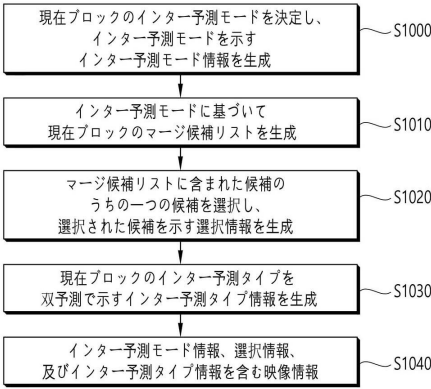
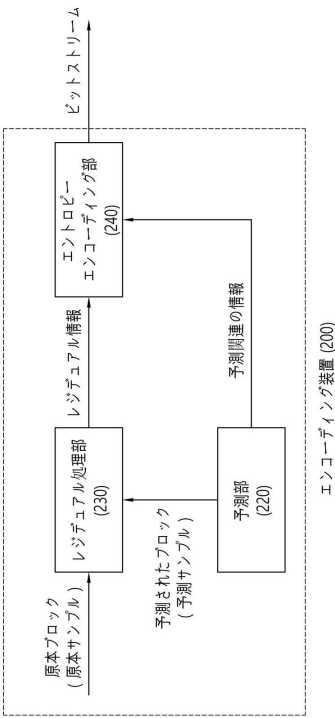


図 11



【図 1 2】

【図 1 3】

図 12

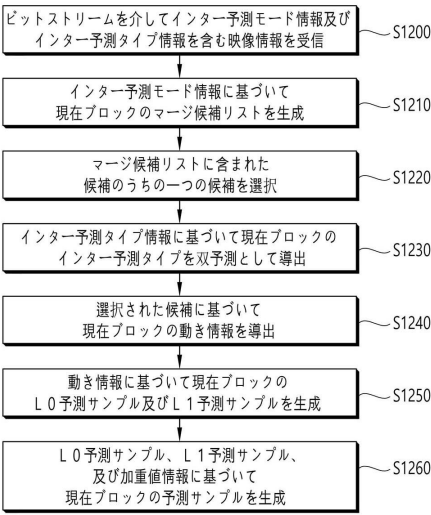
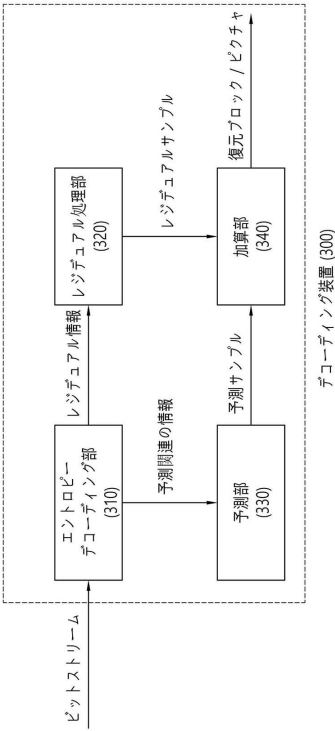


図 13



10

20

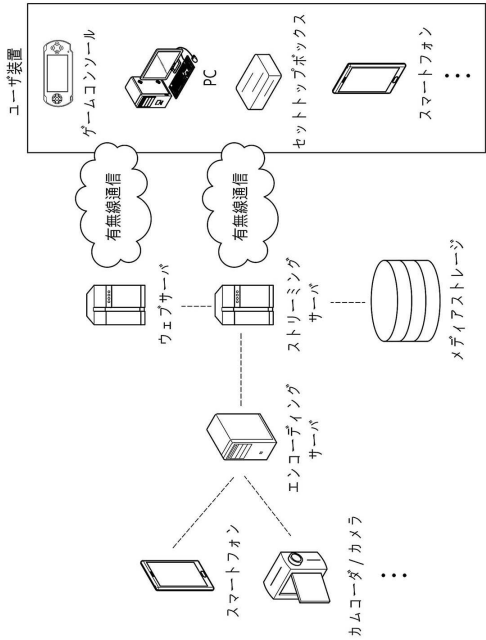
30

40

50

【図 14】

図 14



10

20

30

40

50

フロントページの続き

- 弁理士 河合 章
(74)代理人 100114018
弁理士 南山 知広
(74)代理人 100159259
弁理士 竹本 実
(72)発明者 バク ネリ
大韓民国, ソウル 0 6 7 7 2, ソチョーク, ヤンジェーデロ 1 1 -ギル, 1 9, エルジー エレクトロニクス インコーポレイティド, アイピー センター
(72)発明者 ナム チョンハク
大韓民国, ソウル 0 6 7 7 2, ソチョーク, ヤンジェーデロ 1 1 -ギル, 1 9, エルジー エレクトロニクス インコーポレイティド, アイピー センター
(72)発明者 チャン ヒョンムン
大韓民国, ソウル 0 6 7 7 2, ソチョーク, ヤンジェーデロ 1 1 -ギル, 1 9, エルジー エレクトロニクス インコーポレイティド, アイピー センター
審査官 岩井 健二
(56)参考文献 Huanbang Chen, Haitao Yang, and Jianle Chen, CE4: Common base for affine merge mode (Test 4.2.1), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L0366-v1, 12th Meeting: Macao, CN, 2018年09月, pp.1-3
Yu-Chi Su, et al., CE4-related: Generalized bi-prediction improvements combined from JVE T-L0197 and JVET-L0296, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO /IEC JTC 1/SC 29/WG 11, JVET-L0646-v5, 12th Meeting: Macao, CN, 2018年10月, pp. 1-6
Jianle Chen Yan Ye Seung Hwan Kim, Algorithm description for Versatile Video Coding and Test Model 3 (VTM 3), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L1002-v1, 12th Meeting: Macao, CN, 2018年12月, pp.1-5, 18-30
Naeri Park, et al., Non-CE4 : Simplifications on BCW index derivation process, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-O0366 -r1, 15th Meeting: Gothenburg, SE, 2019年07月, pp.1-5
(58)調査した分野 (Int.Cl., D B名)
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8