

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7510550号
(P7510550)

(45)発行日 令和6年7月3日(2024.7.3)

(24)登録日 令和6年6月25日(2024.6.25)

(51)国際特許分類		F I	
H O 4 N	19/52 (2014.01)	H O 4 N	19/52
H O 4 N	19/105 (2014.01)	H O 4 N	19/105
H O 4 N	19/54 (2014.01)	H O 4 N	19/54
H O 4 N	19/577 (2014.01)	H O 4 N	19/577
H O 4 N	19/70 (2014.01)	H O 4 N	19/70
請求項の数 3 (全85頁)			
(21)出願番号	特願2023-126949(P2023-126949)	(73)特許権者	502032105
(22)出願日	令和5年8月3日(2023.8.3)		エルジー エレクトロニクス インコーポ レイティド
(62)分割の表示	特願2021-574299(P2021-574299)の分割		L G E L E C T R O N I C S I N C .
原出願日	令和2年6月10日(2020.6.10)		大韓民国, ソウル, ヨンドゥンポーク, ヨイーデロ, 1 2 8
(65)公開番号	特開2023-145702(P2023-145702 A)		1 2 8 , Y e o u i - d a e r o , Y
(43)公開日	令和5年10月11日(2023.10.11)		e o n g d e u n g p o - g u , 0 7
審査請求日	令和5年8月3日(2023.8.3)		3 3 6 S e o u l , R e p u b l i c
(31)優先権主張番号	62/861,988		o f K o r e a
(32)優先日	令和1年6月14日(2019.6.14)	(74)代理人	100099759
(33)優先権主張国・地域又は機関	米国(US)		弁理士 青木 篤
		(74)代理人	100123582
			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 (b i - p r e d i c t i o n) が適用される場合、加重平均のための加重値イン
デックス情報を導出する画像デコード方法及びその装置

(57)【特許請求の範囲】

【請求項1】

デコード装置によって行われる画像デコード方法において、
ビットストリームを介してインター予測モード情報及びレジデュアル情報を含む画像情
報を取得するステップと、
前記レジデュアル情報に基づいてレジデュアルサンプルを生成するステップと、
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内の候補から選択された候補に基づいて前記現在ブロックに関す
る動き情報を導出するステップと、
前記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、及び前記現在ブロックのための加重値
情報に基づいて双予測を実行することによって前記現在ブロックの予測サンプルを生成す
るステップであって、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
クスに基づいて導出されるステップと、
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ス
テップと、
を含み、
前記候補は、継承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を含み、

前記継承されたアフィンマージ候補は、前記現在ブロックの周辺ブロックのCPMV (control point motion vector) に基づいて導出され、

前記構成されたアフィンマージ候補のそれぞれは、CP0 (Control Point 0) に対するCPMV、CP1 (Control Point 1) に対するCPMV、CP2 (Control Point 2) に対するCPMV、又はCP3 (Control Point 3) に対するCPMVの、少なくとも2個を含み、

前記CP0は、前記現在ブロックの左上側コーナーに関連し、

前記CP1は、前記現在ブロックの右上側コーナーに関連し、

前記CP2は、前記現在ブロックの左下側コーナーに関連し、

前記CP3は、前記現在ブロックの右下側コーナーに関連し、

第1の構成されたアフィンマージ候補が{CP0、CP1、CP2}に基づいて生成されることと、前記双予測が前記現在ブロックに適用されることとに応じて、前記第1の構成されたアフィンマージ候補に対する加重値インデックスは、前記CP0の周辺ブロックのうち、特定ブロックに対する加重値インデックスに等しいように固定され、
前記CP0の前記周辺ブロックのうち、前記特定ブロックは、前記CP0に対する前記CPMVを導出するために使用されるブロックであり、

第2の構成されたアフィンマージ候補が{CP1、CP2、CP3}に基づいて生成されることと、前記双予測が前記現在ブロックに適用されることとに応じて、前記第2の構成されたアフィンマージ候補に対する加重値インデックスは、前記CP1の周辺ブロックのうち、特定ブロックに対する加重値インデックスに等しいように固定され、
前記CP1の前記周辺ブロックのうち、前記特定ブロックは、前記CP1に対する前記CPMVの導出のために使用されるブロックであり、

前記構成されたアフィンマージ候補は、前記マージ候補リスト中の前記継承されたアフィンマージ候補の後に挿入される、画像デコード方法。

【請求項2】

エンコード装置によって行われる画像エンコード方法において、

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1つを示す選択情報を生成するステップと、

前記現在ブロックのレジデュアルサンプルに基づいてレジデュアル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、前記選択情報、及び前記レジデュアル情報を含む画像情報をエンコードするステップと、

を含み、

前記候補は、継承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を含み、前記継承されたアフィンマージ候補は、前記現在ブロックの周辺ブロックのCPMV (control point motion vector) に基づいて導出され、

前記構成されたアフィンマージ候補のそれぞれは、CP0 (Control Point 0) に対するCPMV、CP1 (Control Point 1) に対するCPMV、CP2 (Control Point 2) に対するCPMV、又はCP3 (Control Point 3) に対するCPMVの、少なくとも2個を含み、

前記CP0は、前記現在ブロックの左上側コーナーに関連し、

前記CP1は、前記現在ブロックの右上側コーナーに関連し、

前記CP2は、前記現在ブロックの左下側コーナーに関連し、

前記CP3は、前記現在ブロックの右下側コーナーに関連し、

第1の構成されたアフィンマージ候補が{CP0、CP1、CP2}に基づいて生成されることと、双予測が前記現在ブロックに適用されることとに応じて、前記第1の構成されたアフィンマージ候補に対する加重値インデックスは、前記CP0の周辺ブロックのう

ち、特定ブロックに対する加重値インデックスに等しいように固定され、
前記C P 0の前記周辺ブロックのうち、前記特定ブロックは、前記C P 0に対する前記C P M Vを導出するために使用されるブロックであり、

第2の構成されたアフィンマージ候補が{C P 1、C P 2、C P 3}に基づいて生成されることと、前記双予測が前記現在ブロックに適用されることとに応じて、前記第2の構成されたアフィンマージ候補に対する加重値インデックスは、前記C P 1の周辺ブロックのうち、特定ブロックに対する加重値インデックスに等しいように固定され、

前記C P 1の前記周辺ブロックのうち、前記特定ブロックは、前記C P 1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り、

前記構成されたアフィンマージ候補は、前記マージ候補リスト中の前記継承されたアフィンマージ候補の後に挿入される、画像エンコード方法。

10

【請求項3】

画像のためのデータの送信方法であって、

前記画像のためのビットストリームを取得するステップであって、前記ビットストリームは、

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1つを示す選択情報を生成するステップと、

20

前記現在ブロックのレジデュアルサンプルに基づいてレジデュアル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、前記選択情報、及び前記レジデュアル情報を含む画像情報をエンコードするステップと、

に基づいて生成される、ステップと、

前記ビットストリームを含む前記データを送信するステップと、を含み、

前記候補は、継承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を含み、前記継承されたアフィンマージ候補は、前記現在ブロックの周辺ブロックのC P M V (control point motion vector) に基づいて導出され、

前記構成されたアフィンマージ候補のそれぞれは、C P 0 (Control Point 0) に対するC P M V、C P 1 (Control Point 1) に対するC P M V、C P 2 (Control Point 2) に対するC P M V、又はC P 3 (Control Point 3) に対するC P M Vの、少なくとも2個を含み、

30

前記C P 0は、前記現在ブロックの左上側コーナーに関連し、

前記C P 1は、前記現在ブロックの右上側コーナーに関連し、

前記C P 2は、前記現在ブロックの左下側コーナーに関連し、

前記C P 3は、前記現在ブロックの右下側コーナーに関連し、

第1の構成されたアフィンマージ候補が{C P 0、C P 1、C P 2}に基づいて生成されることと、双予測が前記現在ブロックに適用されることとに応じて、前記第1の構成されたアフィンマージ候補に対する加重値インデックスは、前記C P 0の周辺ブロックのうち、特定ブロックに対する加重値インデックスに等しいように固定され、

40

前記C P 0の前記周辺ブロックのうち、前記特定ブロックは、前記C P 0に対する前記C P M Vを導出するために使用されるブロックであり、

第2の構成されたアフィンマージ候補が{C P 1、C P 2、C P 3}に基づいて生成されることと、前記双予測が前記現在ブロックに適用されることとに応じて、前記第2の構成されたアフィンマージ候補に対する加重値インデックスは、前記C P 1の周辺ブロックのうち、特定ブロックに対する加重値インデックスに等しいように固定され、

前記C P 1の前記周辺ブロックのうち、前記特定ブロックは、前記C P 1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り、

前記構成されたアフィンマージ候補は、前記マージ候補リスト中の前記継承されたアフィ

50

ンマージ候補の後に挿入される、送信方法。

【発明の詳細な説明】

【技術分野】

【0001】

本技術は、双予測（*bi-prediction*）が適用される場合、加重平均のための加重値インデックス情報を導出する画像デコード方法及びその装置に関する。

【背景技術】

【0002】

近年、4Kまたは8K以上のUHD（*Ultra High Definition*）画像／ビデオのような高解像度、高品質の画像／ビデオに対する需要が様々な分野で増加している。画像／ビデオデータが高解像度、高品質になるほど、既存の画像／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画像データを送信するか、既存の格納媒体を利用して画像／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、ゲーム画像のように、現実画像と異なる画像特性を有する画像／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・高品質の画像／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画像／ビデオ圧縮技術が求められる。

【発明の概要】

【発明が解決しようとする課題】

【0005】

本文書の技術的課題は、画像コーディング効率を上げる方法及び装置を提供することにある。

【0006】

本文書の他の技術的課題は、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導く方法及び装置を提供することにある。

【0007】

本文書のさらに他の技術的課題は、双予測が適用される場合、加重平均のための加重値インデックス情報を導く方法及び装置を提供することにある。

【課題を解決するための手段】

【0008】

本文書の一実施形態によれば、デコード装置によって行われる画像デコード方法が提供される。前記方法は、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レジデュアル情報を含む画像情報を取得するステップと、前記レジデュアル情報に基づいてレジデュアルサンプルを生成するステップと、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のうち、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、前記動き情報に基づいて前記現在ブロックのL0予測サンプル及びL1予測サンプルを生成するステップと、前記L0予測サンプル、前記L1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、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ステップとを含み、前記候補は、アフィンマージ候補を含み、前記アフィンマージ候補は、前記現在ブロックの左上側に位置するCP0（*Control Point 0*）、前記現在ブロックの右上側に位置するCP1（*Control Point 1*）、前記現在ブロックの左下側に位置するCP2（*Control Point 2*）

trol Point 2)、及び前記現在ブロックの右下側に位置するCP3 (Control Point 3)の各々に対する動きベクトルのうち、少なくとも1つを含み、前記アフィンマージ候補が{CP0、CP1、CP2}、{CP01、CP1、CP3}、{CP0、CP2、CP3}、{CP0、CP1}、及び{CP0、CP2}のうち、いずれか1つに基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記CP0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基づいて導出され、前記アフィンマージ候補が{CP1、CP2、CP3}に基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記CP1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基づいて導出される。
【0009】

10

本文書の他の一実施形態によれば、エンコード装置によって行われるビデオエンコード方法を提供する。前記方法は、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表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1つの候補を表す選択情報を生成するステップと、前記現在ブロックに対するレジデュアルサンプルに基づいてレジデュアル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、前記選択情報、及び前記レジデュアル情報を含む画像情報をエンコードするステップとを含み、前記候補は、アフィンマージ候補を含み、前記アフィンマージ候補は、前記現在ブロックの左上側に位置するCP0 (Control Point 0)、前記現在ブロックの右上側に位置するCP1 (Control Point 1)、前記現在ブロックの左下側に位置するCP2 (Control Point 2)、及び前記現在ブロックの右下側に位置するCP3 (Control Point 3)の各々に対する動きベクトルのうち、少なくとも1つを含み、前記アフィンマージ候補が{CP0、CP1、CP2}、{CP0、CP1、CP3}、{CP0、CP2、CP3}、{CP0、CP1}、及び{CP0、CP2}のうち、いずれか1つに基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記CP0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基づいて表され、前記アフィンマージ候補が{CP1、CP2、CP3}に基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記CP1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基づいて表される。

20

30

【0010】

本文書のさらに他の一実施形態によれば、デコード装置をして画像デコード方法を行うように引き起こす画像情報を含むビットストリームが格納されたコンピュータ読み取り可能デジタル格納媒体を提供する。前記画像デコード方法は、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レジデュアル情報を含む画像情報を取得するステップと、前記レジデュアル情報に基づいてレジデュアルサンプルを生成するステップと、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のうち、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、前記動き情報に基づいて前記現在ブロックのL0予測サンプル及びL1予測サンプルを生成するステップと、前記L0予測サンプル、前記L1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、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ステップとを含み、前記候補は、アフィンマージ候補を含み、前記アフィンマージ候補は、前記現在ブロックの左上側に位置するCP0 (Control Point 0)、前記現在ブロックの右上側に位置するCP1 (Control Point 1)、前記現在ブロックの左下側に位置するCP2 (Control Point 2)、及び前記現在ブロックの右下側に位置するCP3 (Control Point 3)の各々に対する動きベクトルのうち、少なくとも1つを含み、前記アフィンマージ候補が{CP0、CP1、CP2}、{CP0、CP1、CP3}、{CP0、CP2、CP3}、{CP0、CP1}、

40

50

及び {C P 0、C P 2} のうち、いずれか 1 つに基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記 C P 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基づいて導出され、前記アフィンマージ候補が {C P 1、C P 2、C P 3} に基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記 C P 1 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基づいて導出される。

【発明の効果】

【0011】

本文書によれば、全般的な画像／ビデオ圧縮効率を上げることができる。

【0012】

本文書によれば、インター予測の際、動きベクトル候補を効率的に構成することができる。

【0013】

本文書によれば、効率的に加重値基盤の双予測を行うことができる。

【図面の簡単な説明】

【0014】

【図 1】本文書の実施形態が適用され得るビデオ／画像コーディングシステムの例を概略的に示す。

【図 2】本文書の実施形態が適用され得るビデオ／画像エンコード装置の構成を概略的に説明する図である。

【図 3】本文書の実施形態が適用され得るビデオ／画像デコード装置の構成を概略的に説明する図である。

【図 4】インター予測に基づいたビデオ／画像エンコード方法の一例を示す図である。

【図 5】エンコード装置内のインター予測部を概略的に示す図である。

【図 6】インター予測に基づいたビデオ／画像デコード方法の一例を示す図である。

【図 7】デコード装置内のインター予測部を概略的に示す図である。

【図 8】インター予測においてマージモードを説明するための図である。

【図 9 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 9 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 10】アフィン M V F がサブブロック単位で決定される場合を例示的に示す。

【図 11】インター予測においてアフィンマージモードを説明するための図である。

【図 12】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 13】インター予測において S b T M V P を説明するための図である。

【図 14】本文書の実施形態（等）に係るビデオ／画像エンコード方法及び関連コンポーネントの一例を概略的に示す。

【図 15】本文書の実施形態（等）に係るビデオ／画像エンコード方法及び関連コンポーネントの一例を概略的に示す。

【図 16】本文書の実施形態（等）に係る画像／ビデオデコード方法及び関連コンポーネントの一例を概略的に示す。

【図 17】本文書の実施形態（等）に係る画像／ビデオデコード方法及び関連コンポーネントの一例を概略的に示す。

【図 18】本文書において開示された実施形態が適用され得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】

【0015】

本文書の開示は、様々な変更を加えることができ、様々な実施形態を有することができる。特定の実施形態を図面に例示し、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形態に限定しようとするものではない。本文書で常用する用語は、単に特定の実施形態を説明するために使用されたものであって、本文書の実施形態等の技術的思想を限定しようとする意図で使用されるものではない。単数の表現は、文脈上明らか

10

20

30

40

50

に異なるように意味しない限り、複数の表現を含む。本明細書において、「含む」又は「有する」などの用語は、文書上に記載された特徴、数字、ステップ、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つ又はそれ以上の他の特徴や、数字、ステップ、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可能性を予め排除しないことと理解されるべきである。

【0016】

一方、本文書において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的に図示されたものであって、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実現されるということを意味するものではない。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があり、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形態も、本文書の開示範囲に含まれる。

10

【0017】

本文書は、ビデオ／画像コーディングに関するものである。例えば、本文書において開示された方法／実施形態は、VVC (versatile video coding) 標準に開示される方法に適用されることができる。また、本文書において開示された方法／実施形態は、EVC (essential video coding) 標準、AV1 (AOMedia Video 1) 標準、AVS2 (2nd generation of audio video coding standard)、または次世代ビデオ／画像コーディング標準 (ex. H. 267 or H. 268 等) に開示される方法に適用されることができる。

20

【0018】

本文書では、ビデオ／画像コーディングに関する様々な実施形態を提示し、他の言及がない限り、前記実施形態等は、互いに組み合わせられて行われることもできる。

【0019】

以下、添付した図面を参照して、本文書の実施形態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でき、同じ構成要素について重複した説明は省略されることができる。

【0020】

図1は、本開示が適用され得るビデオ／画像コーディングシステムの例を概略的に示す。

30

【0021】

図1に示すように、ビデオ／画像コーディングシステムは、第1の装置（ソースデバイス）及び第2の装置（受信デバイス）を含むことができる。ソースデバイスは、エンコードされたビデオ (video)／画像 (image) 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。

【0022】

前記ソースデバイスは、ビデオソース、エンコード装置、送信部を含むことができる。前記受信デバイスは、受信部、デコード装置、及びレンダラを含むことができる。前記エンコード装置は、ビデオ／画像エンコード装置と呼ばれることができ、前記デコード装置は、ビデオ／画像デコード装置と呼ばれることができる。送信機は、エンコード装置に含まれることができる。受信機は、デコード装置に含まれることができる。レンダラは、ディスプレイ部を含むこともでき、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。

40

【0023】

ビデオソースは、ビデオ／画像のキャプチャ、合成または生成過程などを介してビデオ／画像を取得することができる。ビデオソースは、ビデオ／画像キャプチャデバイス及び／又はビデオ／画像生成デバイスを含むことができる。ビデオ／画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／画像を含むビデオ／画像アーカイブなどを含むことができる。ビデオ／画像生成デバイスは、例えば、コンピュ

50

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／画像を生成できる。例えば、コンピュータなどを介して仮想のビデオ／画像が生成されることができ、この場合、ビデオ／画像キャプチャ過程を関連データが生成される過程に代替されることができる。

【0024】

エンコード装置は、入力ビデオ／画像をエンコードすることができる。エンコード装置は、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、一連の手順を実行することができる。エンコードされたデータ（エンコードされたビデオ／画像情報）は、ビットストリーム（*bit stream*）形態で出力されることができる。

【0025】

送信部は、ビットストリーム形態で出力されたエンコードされたビデオ／画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た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／抽出してデコード装置に伝達できる。

【0026】

デコード装置は、エンコード装置の動作に対応する逆量子化、逆変換、予測など、一連の手順を実行してビデオ／画像をデコードすることができる。

【0027】

レンダラは、デコードされたビデオ／画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／画像は、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。

【0028】

この文書は、ビデオ／画像コーディングに関するものである。例えば、この文書において開示された方法／実施形態は、VVC（*versatile video coding*）標準、EVC（*essential video coding*）標準、AV1（*AOMedia Video 1*）標準、AVS2（*2nd generation of audio video coding standard*）、または次世代ビデオ／画像コーディング標準（*ex. H. 267 or H. 268*等）に開示される方法に適用されることができる。

【0029】

この文書では、ビデオ／画像コーディングに関する様々な実施形態を提示し、他の言及がない限り、前記実施形態は、互いに組み合わせて実行されることもできる。

【0030】

この文書において、ビデオ（*video*）は、時間の流れによる一連の画像（*image*）の集合を意味することができる。ピクチャ（*picture*）は、一般的に特定時間帯の1つの画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。

【0031】

タイルは、特定タイル列及び特定タイル列以内の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って明示されることができる（*The tile column is a rectangular region of CTUs having a h*

10

20

30

40

50

eight 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ることができる (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の特定の順次的オーダリングを表すことができ、前記CTUは、タイル内のCTUラスタースキャンで連続的に整列されることができる、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で連続的に整列されることができる (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、複数の完全なタイルまたは1つのNALユニットに含まれることができるピクチャの1つのタイル内の複数の連続的なCTU行を含むことができる。この文書においてタイルグループとスライスとは混用されることができる。例えば、本文書においてtile group/tile group headerは、slice/slice headerと呼ばれることができる。

【0032】

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ることができる (an rectangular region of one or more slices within a picture)。

【0033】

ピクセル (pixel) またはペル (pel) は、1つのピクチャ (または、画像) 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もでき、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。または、サンプルは、空間ドメインでのピクセル値を意味することもでき、このようなピクセル値が周波数ドメインに変換されると、周波数ドメインでの変換係数を意味することもできる。

【0034】

ユニット (unit) は、画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。

【0035】

本文書において「AまたはB (A or B)」は、「Aのみ」、「Bのみ」、または「AとBの両方」を意味することができる。他に表現すれば、本文書において「AまたはB (A or B)」は、「A及び/又はB (A and/or B)」と解釈されることが

10

20

30

40

50

できる。例えば、本明細書において「A、B、またはC (A、B or C)」は、「Aのみ」、「Bのみ」、「Cのみ」、または「A、B、及びCの任意の全ての組み合わせ (any combination of A、B and C)」を意味することができる。

【0036】

本文書において使用されるスラッシュ (/) やコンマ (comma) は、「及び/又は (and/or)」を意味することができる。例えば、「A/B」は、「A及び/又はB」を意味することができる。これにより、「A/B」は、「Aのみ」、「Bのみ」、または「AとBの両方」を意味することができる。例えば、「A、B、C」は、「A、B、またはC」を意味することができる。

【0037】

本文書において「少なくとも1つのA及びB (at least one of A and B)」は、「Aのみ」、「Bのみ」、または「AとBの両方」を意味することができる。また、本文書において「少なくとも1つのAまたはB (at least one of A or B)」や「少なくとも1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1つのA及びB (at least one of A and B)」と同様に解釈されることができる。

【0038】

また、本文書において「少なくとも1つのA、B、及びC (at least one of A、B and C)」は、「Aのみ」、「Bのみ」、「Cのみ」、または「A、B、及びCの任意の全ての組み合わせ (any combination of A、B and C)」を意味することができる。また、「少なくとも1つのA、B、またはC (at least one of A、B or C)」や「少なくとも1つのA、B、及び/又はC (at least one of A、B and/or C)」は、「少なくとも1つのA、B、及びC (at least one of A、B and C)」を意味することができる。

【0039】

また、本文書で使用される括弧は、「例えば (for example)」を意味することができる。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うる。言い換えれば、本文書の「予測」は、「イントラ予測」に制限 (limit) されるものではなく、「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うる。また、「予測 (即ち、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うる。

【0040】

本文書において1つの図面内で個別的に説明される技術的特徴は、個別的に実現されることができ、同時に実現されることもできる。

【0041】

図2は、本開示が適用され得るビデオ/画像エンコード装置の構成を概略的に説明する図である。以下、ビデオエンコード装置は、画像エンコード装置を含むことができる。

【0042】

図2に示すように、エンコード装置200は、画像分割部 (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ド部 (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。

10

20

30

40

50

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。上述した画像分割部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ド部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。

【0043】

画像分割部210は、エンコード装置200に入力された入力画像(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(TT(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、画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。

【0044】

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、画像)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。

【0045】

エンコード装置200は、入力画像信号(原本ブロック、原本サンプルアレイ)から、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

10

20

30

40

50

号は、変換部 232 に送信される。この場合、図示されたように、エンコード装置 200 内で入力画像信号（原本ブロック、原本サンプルアレイ）から予測信号（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ド部 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ド部 240 でエンコード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

【0046】

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することができる。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。

20

【0047】

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。

30

40

【0048】

50

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 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画像／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類似して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。

10

【0049】

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 を備える) 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) を生成できる。例えば、変換技法は、DCT (Discrete Cosine Transform)、DST (Discrete Sine Transform)、KLT (Karhunen-Loeve Transform)、GBT (Graph-Based Transform)、またはCNT (Conditionally Non-linear Transform) のうち、少なくとも1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するとき、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、それに基づいて取得される変換を意味する。また、変換過程は、正四角形の同じサイズを有するピクセルブロックに適用されることができ、正四角形でない可変サイズのブロックにも適用されることができる。

20

30

【0050】

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ド部 240 に送信され、エントロピーエンコード部 240 は、量子化された信号 (量子化された変換係数に関する情報) をエンコード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ド部 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ド方法を行うことができる。エントロピーエンコード部 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) とともに、または別にエンコードすることもできる。エンコードされた情報 (ex. エンコードされたビデオ／画像情報) 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／画像情報は、アダプテーションパラメータセット (APS

40

50

）、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／画像情報は、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ド装置からデコード装置に伝達／シグナリングされる情報及び／又はシタックス要素は、ビデオ／画像情報に含まれることができる。前記ビデオ／画像情報は、上述したエンコード手順を介してエンコード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、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ド部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）がエンコード装置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ド部240に備えられることもできる。

【0051】

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算することによって復元（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、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ができ、

【0052】

一方、ピクチャエンコード及び／又は復元過程でLMCS（luma mapping with chroma scaling）が適用されることができ、

【0053】

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、復元ピクチャに多様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することができ、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。前記多様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対する説明で後述するように、フィルタリングに関する多様な情報を生成してエントロピーエンコード部240に伝達できる。フィルタリングに関する情報は、エントロピーエンコード部240でエンコードされてビットストリーム形態で出力されることができ、

【0054】

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われることができる。エンコード装置は、これを介してインター予測が適用される場合、エンコード装置200とデコード装置での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。

【0055】

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での参照ピ

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、エンコードされた）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に伝達できる。

【0056】

一方、本文書において量子化／逆量子化及び／又は変換／逆変換のうち、少なくとも1つは省略されることができる。前記量子化／逆量子化が省略される場合、前記量子化された変換係数は、変換係数と呼ばれることができる。前記変換／逆変換が省略される場合、前記変換係数は、係数またはレジデュアル係数と呼ばれることができ、または、表現の統一性のために、変換係数と依然として呼ばれることもできる。

【0057】

また、本文書において量子化された変換係数及び変換係数は、各々変換係数及びスケーリングされた（scaled）変換係数と称されることができる。この場合、レジデュアル情報は、変換係数（等）に関する情報を含むことができ、前記変換係数（等）に関する情報は、レジデュアルコーディングシンタックスを介してシグナリングされることができる。前記レジデュアル情報（または、前記変換係数（等）に関する情報）に基づいて変換係数が導出され得るし、前記変換係数に対する逆変換（スケーリング）を介してスケーリングされた変換係数が導出され得る。前記スケーリングされた変換係数に対する逆変換（変換）に基づいてレジデュアルサンプルが導出され得る。これは、本文書の他の部分でも同様に適用／表現されることができる。

【0058】

図3は、本開示が適用され得るビデオ／画像デコード装置の構成を概略的に説明する図である。

【0059】

図3に示すように、デコード装置300は、エントロピーデコード部（entropy decoder）310、レジデュアル処理部（residual processor）320、予測部（predictor）330、加算部（adder）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、イントラ予測部331及びインター予測部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）322を備えることができる。上述したエントロピーデコード部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ができる。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。

【0060】

ビデオ／画像情報を含むビットストリームが入力されれば、デコード装置300は、図2のエンコード装置でビデオ／画像情報が処理されたプロセスに対応して画像を復元することができる。例えば、デコード装置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。デコード装置300は、エンコード装置で適用された処理ユニットを用いてデコードを行うことができる。したがって、デコードの処理ユニットは、例えば、コーディ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

10

20

30

40

50

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、デコード装置300を介してデコード及び出力された復元画像信号は、再生装置を介して再生されることができる。

【0061】

デコード装置300は、図3のエンコード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、エントロピーデコード部310を介してデコードされることができる。例えば、エントロピーデコード部310は、前記ビットストリームをパースングして画像復元（または、ピクチャ復元）に必要な情報（ex. ビデオ／画像情報）を導出できる。前記ビデオ／画像情報は、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／画像情報は、一般制限情報（general constraint information）をさらに含むことができる。デコード装置は、さらに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ピクチャをデコードすることができる。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、前記デコード手順を介してデコードされて、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、エントロピーデコード部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ドし、画像復元に必要なシンタックスエレメントの値、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。より詳細に、CABACエントロピーデコード方法は、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、デコード対象シンタックス要素情報と周辺及びデコード対象ブロックのデコード情報、あるいは以前ステップでデコードされたシンボル／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、ピンの算術デコード（arithmetic decoding）を行って、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エントロピーデコード方法は、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ドされたシンボル／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。エントロピーデコード部310でデコードされた情報のうち、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、エントロピーデコード部310でエントロピーデコード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、エントロピーデコード部310でデコードされた情報のうち、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、エンコード装置から出力された信号を受信する受信部（図示せず）がデコード装置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、エントロピーデコード部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ド装置は、ビデオ／画像／ピクチャデコード装置と呼ばれることができ、前記デコード装置は、情報デコーダ（ビデオ／画像／ピクチャ情報デコーダ）及びサンプルデコーダ（ビデオ／画像／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、前記エントロピーデコード部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1つを備えることができる。

【0062】

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することができる。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、エンコード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

10

20

30

40

50

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）を取得することができる。

【0063】

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。

【0064】

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）を生成することができる。予測部は、エントロピーデコード部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を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することができる。

【0065】

予測部330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画像／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類似して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／画像情報に含まれてシグナリングされることができる。

【0066】

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することができる。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）に位置でき、または離れて位置することもできる。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利用し、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。

【0067】

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、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。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。多様な予測モードに基づいてインター予測が実行されるこ

10

20

30

40

50

とができ、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。

【0068】

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）に加算することによって復元信号（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得る。

【0069】

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われることができ、後述するように、フィルタリングを経て出力されることもでき、または、次のピクチャのインター予測のために使われることもできる。

【0070】

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

【0071】

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、復元ピクチャに多様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することができ、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に送信できる。前記多様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。

【0072】

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われることができる。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、デコードされた）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332に伝達できる。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することができ、イントラ予測部331に伝達できる。

【0073】

本明細書において、エンコード装置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は、各々デコード装置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。

【0074】

一方、前述したように、ビデオコーディングを実行するにあたって圧縮効率を上げるために予測を実行する。それによって、コーディング対象ブロックである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を生成することができる。ここで、前記予測されたブロックは、空間ドメイン（または、ピクセルドメイン）での予測サンプルを含む。前記予測されたブロックは、エンコード装置及びデコード装置で同様に導出され、前記エンコード装置は、原本ブロックの原本サンプル値自体でない前記原本ブロックと前記予測されたブロックとの間のレジデュアルに関する情報（レジデュアル情報）をデコード装置にシグナリングすることで画像コーディング効率を上げることができる。デコード装置は、前記レジデュアル情報に基づいてレジデュアルサンプルを含むレジデュアルブロックを導出し、前記レジデュアルブロックと前記予測されたブロックを加算して復元サンプル

10

20

30

40

50

を含む復元ブロックを生成することができ、復元ブロックを含む復元ピクチャを生成することができる。

【0075】

前記レジデュアル情報は、変換及び量子化手順を介して生成されることができる。例えば、エンコード装置は、前記原本ブロックと前記予測されたブロックとの間のレジデュアルブロックを導出し、前記レジデュアルブロックに含まれているレジデュアルサンプル（レジデュアルサンプルアレイ）に変換手順を実行して変換係数を導出し、前記変換係数に量子化手順を実行して量子化された変換係数を導出することで、関連したレジデュアル情報を（ビットストリームを介して）デコード装置にシグナリングすることができる。ここで、前記レジデュアル情報は、前記量子化された変換係数の値情報、位置情報、変換技法、変換カーネル、量子化パラメータなどの情報を含むことができる。デコード装置は、前記レジデュアル情報に基づいて逆量子化／逆変換手順を実行してレジデュアルサンプル（または、レジデュアルブロック）を導出できる。デコード装置は、予測されたブロックと前記レジデュアルブロックに基づいて復元ピクチャを生成することができる。また、エンコード装置は、以後ピクチャのインター予測のための参照のために量子化された変換係数を逆量子化／逆変換してレジデュアルブロックを導出し、これに基づいて復元ピクチャを生成することができる。

【0076】

図4は、インター予測に基づいたビデオ／画像エンコード方法の一例を示し、図5は、エンコード装置内のインター予測部を概略的に示した一例である。図5のエンコード装置内のインター予測部は、上述した図2のエンコード装置200のインター予測部221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。

【0077】

図4及び図5に示すように、エンコード装置は、現在ブロックに対するインター予測を行う（S400）。エンコード装置は、現在ブロックのインター予測モード及び動き情報を導出し、現在ブロックの予測サンプルを生成できる。ここで、インター予測モード決定、動き情報導出、及び予測サンプル生成手順は、同時に行われることができ、いずれか1つの手順が他の手順より先に行われることもできる。

【0078】

例えば、エンコード装置のインター予測部221は、予測モード決定部221__1、動き情報導出部221__2、予測サンプル導出部221__3を備えることができ、予測モード決定部221__1で前記現在ブロックに対する予測モードを決定し、動き情報導出部221__2で前記現在ブロックの動き情報を導出し、予測サンプル導出部221__3で前記現在ブロックの予測サンプルを導出できる。例えば、エンコード装置のインター予測部221は、動き推定（motion estimation）を介して参照ピクチャの一定領域（サーチ領域）内で現在ブロックと類似したブロックをサーチし、現在ブロックとの差が最小または一定基準以下である参照ブロックを導出できる。これに基づいて参照ブロックが位置する参照ピクチャを指す参照ピクチャインデックスを導出し、参照ブロックと現在ブロックとの位置差に基づいて動きベクトルを導出できる。エンコード装置は、様々な予測モードのうち、現在ブロックに対して適用されるモードを決定できる。エンコード装置は、様々な予測モードに対するRD costを比較し、現在ブロックに対する最適の予測モードを決定できる。

【0079】

例えば、エンコード装置は、現在ブロックにスキップモードまたはマージモードが適用される場合、マージ候補リストを構成し、マージ候補リストに含まれたマージ候補が指す参照ブロックのうち、現在ブロックとの差が最小または一定基準以下である参照ブロックを導出できる。この場合、導出された参照ブロックと連関したマージ候補が選択され、選択されたマージ候補を指すマージインデックス情報が生成されてデコード装置にシグナリングされることができる。選択されたマージ候補の動き情報を利用して現在ブロックの動き情報が導出され得る。

【0080】

他の例として、エンコード装置は、現在ブロックに（A）MVPモードが適用される場合、（A）MVP候補リストを構成し、（A）MVP候補リストに含まれたmvp（motion vector predictor）候補のうち、選択されたmvp候補の動きベクトルを現在ブロックのmvpとして用いることができる。この場合、例えば、上述した動き推定によって導出された参照ブロックを指す動きベクトルが現在ブロックの動きベクトルとして用いられることができ、mvp候補のうち、現在ブロックの動きベクトルとの差が最も小さい動きベクトルを有するmvp候補が選択されたmvp候補になることができる。現在ブロックの動きベクトルからmvpを減算した差分であるMVD（motion vector difference）が導出され得る。この場合、MVDに関する情報がデコード装置にシグナリングされることができる。また、（A）MVPモードが適用される場合、参照ピクチャインデックスの値は、参照ピクチャインデックス情報で構成されて別にデコード装置にシグナリングされることができる。

10

【0081】

エンコード装置は、予測サンプルに基づいてレジデュアルサンプルを導出できる（S410）。エンコード装置は、現在ブロックの原本サンプルと予測サンプルとの比較を介してレジデュアルサンプルを導出できる。

【0082】

エンコード装置は、予測情報及びレジデュアル情報を含む画像情報をエンコードする（S420）。エンコード装置はエンコードされた画像情報をビットストリーム形態で出力することができる。予測情報は、予測手順に関連した情報であって、予測モード情報（ex.skip flag、merge flag or mode index等）及び動き情報に関する情報を含むことができる。動き情報に関する情報は、動きベクトルを導出するための情報である候補選択情報（ex.merge index）を含むことができる。また、動き情報に関する情報は、L0予測、L1予測、または双（bi）予測が適用されるか否かを表す情報を含むことができる。レジデュアル情報は、レジデュアルサンプルに関する情報である。レジデュアル情報は、レジデュアルサンプルに対する量子化された変換係数に関する情報を含むことができる。

20

【0083】

出力されたビットストリームは、（デジタル）格納媒体に格納されてデコード装置に伝達されることができ、または、ネットワークを介してデコード装置に伝達されることもできる。

30

【0084】

また、上述したように、エンコード装置は、参照サンプル及びレジデュアルサンプルに基づいて復元ピクチャ（復元サンプル及び復元ブロックを含む）を生成できる。これは、デコード装置で行われることと同一の予測結果をエンコード装置で導出するためであり、これを介してコーディング効率を上げることができるためである。したがって、エンコード装置は、復元ピクチャ（または、復元サンプル、復元ブロック）をメモリに格納し、インター予測のための参照ピクチャとして活用することができる。復元ピクチャにインループフィルタリング手順などがさらに適用され得ることは、上述したとおりである。

40

【0085】

図6は、インター予測に基づいたビデオ／画像デコード方法の一例を示し、図7は、デコード装置内のインター予測部を概略的に示した一例である。図7のデコード装置内のインター予測部は、上述した図3のデコード装置300のインター予測部332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。

【0086】

図6及び図7に示すように、デコード装置は、前記エンコード装置で行われた動作と対応する動作を行うことができる。デコード装置は、受信された予測情報に基づいて現在ブロックに予測を行い、予測サンプルを導出できる。

【0087】

50

具体的に、デコード装置は、受信された予測情報に基づいて現在ブロックに対する予測モードを決定できる（S600）。デコード装置は、予測情報内の予測モード情報に基づいて現在ブロックにどのインター予測モードが適用されるか決定することができる。

【0088】

インター予測モード候補は、スキップモード、マージモード、及び／又は（A）MVPモードを含むことができ、または、様々なインター予測モードを含むことができる。

【0089】

デコード装置は、決定されたインター予測モードに基づいて現在ブロックの動き情報を導出する（S610）。例えば、デコード装置は、現在ブロックにスキップモードまたはマージモードが適用される場合、マージ候補リストを構成し、マージ候補リストに含まれたマージ候補のうち1つのマージ候補を選択できる。ここで、選択は、上述した選択情報（merge index）に基づいて行われることができる。選択されたマージ候補の動き情報を利用して現在ブロックの動き情報が導出され得る。選択されたマージ候補の動き情報が現在ブロックの動き情報として利用されることができる。

【0090】

他の例として、デコード装置は、現在ブロックに（A）MVPモードが適用される場合、（A）MVP候補リストを構成し、（A）MVP候補リストに含まれたmvp（motion vector predictor）候補のうち、選択されたmvp候補の動きベクトルを現在ブロックのmvpとして用いることができる。ここで、選択は、上述した選択情報（mvp flag or mvp index）に基づいて行われることができる。この場合、前記MVDに関する情報に基づいて現在ブロックのMVDを導出でき、現在ブロックのmvpとMVDとに基づいて現在ブロックの動きベクトルを導出できる。また、参照ピクチャインデックス情報に基づいて現在ブロックの参照ピクチャインデックスを導出できる。現在ブロックに関する参照ピクチャリスト内で参照ピクチャインデックスが指すピクチャが現在ブロックのインター予測のために参照される参照ピクチャとして導出されることができる。

【0091】

一方、候補リスト構成なしに現在ブロックの動き情報が導出され得るし、この場合、予測モードで開示された手順によって現在ブロックの動き情報が導出され得る。この場合、上述したような候補リスト構成は省略されることができる。

【0092】

デコード装置は、現在ブロックの動き情報に基づいて現在ブロックに対する予測サンプルを生成できる（S620）。この場合、現在ブロックの参照ピクチャインデックスに基づいて参照ピクチャを導出し、現在ブロックの動きベクトルが参照ピクチャ上で指す参照ブロックのサンプルを用いて現在ブロックの予測サンプルを導出できる。このとき、場合によって、現在ブロックの予測サンプルのうち、全部または一部に対する予測サンプルフィルタリング手順がさらに行われ得る。

【0093】

例えば、デコード装置のインター予測部332は、予測モード決定部332__1、動き情報導出部332__2、予測サンプル導出部332__3を備えることができ、予測モード決定部332__1で受信された予測モード情報に基づいて現在ブロックに対する予測モードを決定し、動き情報導出部332__2で受信された動き情報に関する情報に基づいて現在ブロックの動き情報（動きベクトル及び／又は参照ピクチャインデックス等）を導出し、予測サンプル導出部332__3で現在ブロックの予測サンプルを導出できる。

【0094】

デコード装置は、受信されたレジデュアル情報に基づいて現在ブロックに対するレジデュアルサンプルを生成する（S630）。デコード装置は、予測サンプル及びレジデュアルサンプルに基づいて現在ブロックに対する復元サンプルを生成し、これに基づいて復元ピクチャを生成できる（S640）。その後、復元ピクチャにインループフィルタリング手順などがさらに適用され得ることは、上述したとおりである。

【0095】

上述したように、インター予測手順は、インター予測モード決定ステップ、決定された予測モードによる動き情報導出ステップ、導出された動き情報に基づいた予測実行（予測サンプル生成）ステップを含むことができる。インター予測手順は、上述したように、エンコード装置及びデコード装置で行われることができる。

【0096】

一方、現在ブロックの動き情報を導出するにあたり、空間的周辺ブロック（等）及び時間的周辺ブロック（等）に基づいて動き情報候補（等）を導出し、導出された動き情報候補（等）に基づいて現在ブロックのための動き情報候補を選択できる。このとき、選択された動き情報候補を現在ブロックの動き情報として使用することができる。

10

【0097】

図8は、インター予測においてマージモードを説明するための図である。

【0098】

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に、周辺予測ブロックの動き情報を利用して前記現在予測ブロックの動き情報を導くようになる。したがって、マージモードを用いたことを報知す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報知す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）と呼ばれることができる。例えば、前記マージモードは、regular_merge_flag 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ことができる。

20

【0099】

エンコード装置は、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導く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、本文書の実施形態（等）は、これに限定されない。そして、前記マージ候補ブロックの最大個数は、スライスヘッダまたはタイルグループヘッダで送信されることができるが、本文書の実施形態（等）は、これに限定されない。前記マージ候補ブロックを探した後、エンコード装置は、マージ候補リストを生成でき、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マージ候補ブロックとして選択することができる。

30

【0100】

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様々な実施形態を提供できる。

【0101】

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）とを利用できる。具体的な例として、空間的マージ候補の場合、図8に示されたブロックを空間的マージ候補として用いることができる。以下、前記空間的マージ候補または後述する空間的MVP候補は、SMVPと呼ばれることができ、前記時間的マージ候補または後述する時間的MVP候補は、TMVPと呼ばれることができる。

40

【0102】

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

【0103】

コーディング装置（エンコード装置／デコード装置）は、現在ブロックの空間的周辺ブロックを探索し、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することができる。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コーナー周辺ブロック、左側周辺ブロック、右上側コーナー周辺ブロック、上側周辺ブロック、左上側コーナ

50

一周辺ブロックを含むことができる。ただし、これは例示であって、上述した空間的周辺ブロックの他にも、右側周辺ブロック、下側周辺ブロック、右下側周辺ブロックなど、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て利用可能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で導出することができる。例えば、エンコード装置またはデコード装置は、図8に示された5個のブロックを $A1 \rightarrow B1 \rightarrow B0 \rightarrow A0 \rightarrow B2$ のように順番に探索し、利用可能な候補を順次インデクシングしてマージ候補リストとして構成することができる。

【0104】

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、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することができる。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することができる。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) ピクチャまたはコル (col) ピクチャと呼ばれることができる。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一位置ブロック (co-located block) の右下側コーナー周辺ブロック及び右下側センターブロックの順に探索されることができる。一方、動き情報圧縮 (motion data compression) が適用される場合、前記コル (col) ピクチャに一定格納単位毎に特定動き情報を代表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、これを介して動き情報圧縮 (motion data compression) 効果を得ることができる。この場合、一定格納単位は、例えば、 16×16 サンプル単位、または 8×8 サンプル単位などで予め決められることができ、またはエンコード装置からデコード装置に前記一定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できる。前記動き情報圧縮 (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格納単位の代表動き情報に代替されることができる。すなわち、この場合、実現の側面から見ると、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ない、前記時間的周辺ブロックの座標 (左上端サンプルポジション) に基づいて一定値の分だけ算術的右側シフト後、算術的左側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され得る。例えば、前記一定格納単位が $2n \times 2n$ 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されることができる。具体的に、例えば、前記一定格納単位が 16×16 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されることができる。または、例えば、前記一定格納単位が 8×8 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されることができる。

【0105】

コーディング装置は、現在マージ候補の個数が最大マージ候補の個数より小さいか否かを確認できる。前記最大マージ候補の個数は、予め定義されるか、エンコード装置からデコード装置にシグナリングされることができる。例えば、エンコード装置は、前記最大マージ候補の個数に関する情報を生成し、エンコードしてビットストリーム形態で前記デコーダに伝達することができる。前記最大マージ候補の個数が全て満たされれば、以後の候補追加過程は、進行しないことができる。

【0106】

前記確認結果、前記現在マージ候補の個数が前記最大マージ候補の個数より小さい場合、コーディング装置は、追加マージ候補を前記マージ候補リストに挿入することができる。例えば、前記追加マージ候補は、後述するヒストリー基盤のマージ候補（等）（*history based merge candidate(s)*）、ペアワイズ平均マージ候補（等）（*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うち、少なくとも1つを含むことができる。

【0107】

前記確認結果、前記現在マージ候補の個数が前記最大マージ候補の個数より小さくない場合、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できる。この場合、エンコード装置は、RD（*rate-distortion*）コスト（*cost*）基盤で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でき、前記選択されたマージ候補を指す選択情報（*ex.merge index*）をデコード装置にシグナリングすることができる。デコード装置は、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できる。

【0108】

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として使用されることができ、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できることは、上述したとおりである。エンコード装置は、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ド装置にシグナリングすることができる。デコード装置は、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でき、これに基づいて復元ピクチャを生成できることは、上述したとおりである。

【0109】

スキップモード（*skip mode*）が適用される場合、上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できる。ただし、スキップモードが適用される場合、当該ブロックに対するレジデュアル信号が省略され、したがって、予測サンプルが直ちに復元サンプルとして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* 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。

【0110】

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またはペアワイズ候補と呼ばれることができる。ペアワイズ平均候補（等）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）} のように定義されることができる。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表すことができる。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である場合、前記2個の動きベクトルは、互いに異なる参照ピクチャを指しても平均化されることができる。例えば、1つの動きベクトルのみが利用可能である場合、直ちに（*directly*）1つが使用され得る。例えば、利用可能な動きベクトルがない場合、リストを有効でない状態に維持することができる。

【0111】

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なかった場合、すなわち、マージ候補リスト内に現在マージ候補の個数が最大マージ候補

10

20

30

40

50

の個数より小さい場合、最大マージ候補番号が表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マージ候補の個数が最大マージ候補個数になるまでゼロベクトルを挿入できる。

【0112】

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、移し（translation）動きモデルが使用されることができた。ただし、このような方法がブロック単位で最適の動きを表現したこともあり得るが、実際、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できるならば、コーディング効率を上げることができる。このために、アフィン動きモデル（affine motion model）が使用され得る。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、次のとおりであることができる。

10

【0113】

アフィン動き予測方法は、2個、3個、または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できる。例えば、アフィン動きモデルは、4つの動きを表現できる。アフィン動きモデルが表現できる動きのうち、3つの動き（移し（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）（または、簡素化された（simplified））アフィン動きモデルと呼ばれることができる。しかし、アフィン動きモデルが上述した動きモデルに限定されるものではない。

20

【0114】

図9a及び図9bは、アフィン動き予測のためのCPMVを例示的に示す。

【0115】

アフィン動き予測は、2個以上のコントロールポイント動きベクトル（Control Point Motion Vector、CPMV）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できる。このとき、動きベクトルの集合をアフィン動きベクトルフィールド（MVF：Motion Vector Field）と表すことができる。

【0116】

例えば、図9aは、2個のCPMVが利用される場合を表すことができ、これは、4パラメータアフィンモデルと呼ばれることができる。この場合、（x、y）サンプル位置における動きベクトルは、例えば、数式1のように決定されることができる。

30

【0117】

【数1】

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

【0118】

例えば、図9bは、3個のCPMVが利用される場合を表すことができ、これは、6パラメータアフィンモデルと呼ばれることができる。この場合、（x、y）サンプル位置における動きベクトルは、例えば、数式2のように決定されることができる。

40

【0119】

【数2】

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

50

【0120】

数式1及び数式2において、 $\{v_x, v_y\}$ は、 (x, y) 位置における動きベクトルを表すことができる。また、 $\{v_{0x}, v_{0y}\}$ は、コーディングブロックの左上端コーナー位置のコントロールポイント(CP: Control Point)のCPMVを表すことができ、 $\{v_{1x}, v_{1y}\}$ は、右上端コーナー位置のCPのCPMVを表すことができ、 $\{v_{2x}, v_{2y}\}$ は、左下端コーナー位置のCPのCPMVを表すことができる。また、 W は、現在ブロックの幅($width$)を表すことができ、 H は、現在ブロックの高さ($height$)を表すことができる。

【0121】

図10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。

10

【0122】

エンコード／デコード過程でアフィンMVFは、サンプル単位あるい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ことができる。または、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(センター右下側、すなわち、中央の4個のサンプルのうち、右下側サンプル)サンプル値を基準に当該ブロックの動きベクトルを得ることができる。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、サンプル単位またはサブブロック単位で導出されることができる。

【0123】

図10の場合、アフィンMVFが 4×4 サブブロック単位で決定されるが、サブブロックのサイズは、様々に変形されることができる。

20

【0124】

すなわち、アフィン予測が利用可能な場合、現在ブロックに適用可能な動きモデルは、3つ(移し動きモデル($translational\ motion\ model$))、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、移し動きモデルは、既存のブロック単位動きベクトルが使用されるモデルを表すことができ、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表すことができ、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表すことができる。

30

【0125】

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

【0126】

図11は、インター予測においてアフィンマージモードを説明するための図である。

【0127】

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されることができる。すなわち、周辺ブロックのうち、少なくとも1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 と呼ばれることができる。

40

【0128】

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を利用して現在ブロックのCPMVが導出され得る。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ができ、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などに基づいて修正されて現在ブロックのCPMVとして使用されることもできる。

【0129】

50

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ることができ、これは、サブブロックマージフラグ (または、 $merge_subblock_flag$ シンタックス要素) に基づいて指示されることができる。または、 $merge_subblock_flag$ 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、サブブロックマージ候補リストと呼ばれることもできる。この場合、前記サブブロックマージ候補リストには、後述する $SbTMVP$ で導出された候補がさらに含まれることができる。この場合、前記 $SbTMVP$ で導出された候補は、前記サブブロックマージ候補リストの0番インデックスの候補として利用されることができる。言い換えれば、前記 $SbTMVP$ で導出された候補は、前記サブブロックマージ候補リスト内で後述する継承されたアフィン候補 ($inherited\ affine\ candidate$) または構成されたアフィン候補 ($constructed\ affine\ candidate$) より前に位置することができる。

【0130】

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 $CPMV$ 導出のために、アフィンマージ候補リストが構成され得る。例えば、アフィンマージ候補リストは、次の候補のうち、少なくとも1つを含むことができる。1) 継承された ($inherited$) 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 (または、ゼロベクトル)。ここで、前記継承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 $CPMV$ s 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 $CPMV$ 単位で当該 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表すことができる。

【0131】

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。

【0132】

最大2個の継承されたアフィン候補がありうるし、継承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、1つの左側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のように位置することができる。左側予測子 ($left\ predictor$) のためのスキャン順序は、 $A1 \rightarrow A0$ であることができ、上側予測子 ($above\ predictor$) のためのスキャン順序は、 $B1 \rightarrow B0 \rightarrow B2$ であることができる。左側及び上側の各々から1つの継承された候補のみが選択され得る。2つの継承された候補間には、ブルーニングチェック ($pruning\ check$) が行われなければならないことがある。

【0133】

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表すことができる。例えば、図7に示すように、左下側 ($bottom-left$) 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 ($top-left$) コーナー、右上側 ($top-right$) コーナー、及び左下側 ($bottom-left$) コーナーの動きベクトルの $v2$ 、 $v3$ 、及び $v4$ が取得され得る。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 $v2$ 及び $v3$ によって算出され得る。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 $CPMV$ の $v2$ 、 $v3$ 、及び $v4$ によって算出されることができる。

【0134】

図12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。

【0135】

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 translational 動き情報を組み合わせて構成される候補を意味することができる。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 ($k=0, 1, 2, 3$) は、 k 番目のコントロールポイントを表すことができる。

【0136】

図12に示すように、CPMV0のために、 $B2 \rightarrow B3 \rightarrow A2$ 順序によってブロックがチェックされ得るし、1番目に利用可能なブロックの動きベクトルが使用され得る。CPMV1のために、 $B1 \rightarrow B0$ 順序によってブロックがチェックされ得るし、CPMV2のために、 $A1 \rightarrow A0$ 順序によってブロックがチェックされ得る。TMVP (temporal motion vector predictor) は、利用可能な場合、CPMV3で使用されることができる。

【0137】

4個のコントロールポイントの動きベクトルが取得された後、アフィンマージ候補は、取得した動き情報に基づいて生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、{CPMV0、CPMV1、CPMV2}、{CPMV0、CPMV1、CPMV3}、{CPMV0、CPMV2、CPMV3}、{CPMV1、CPMV2、CPMV3}、{CPMV0、CPMV1}、及び {CPMV0、CPMV2} のうち、いずれか1つに該当することができる。

【0138】

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できる。動きスケール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ることができる。

【0139】

図13は、インター予測において SbTMVP を説明するための図である。

【0140】

一方、SbTMVP (subblock-based temporal motion vector prediction) 方法が利用されることもできる。例えば、SbTMVPは、ATMVP (advanced temporal motion vector prediction) と呼ばれることができる。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利用できる。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できる。

【0141】

例えば、SbTMVPは、サブブロック（または、サブCU）レベルで動きを予測できる。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、1つの動きベクトルから取得されることができる。

【0142】

SbTMVPは、2つのステップによって現在ブロック（または、CU）内のサブブロック（または、サブCU）の動きベクトルを予測できる。

【0143】

第1のステップで空間的周辺ブロックは、図4のA1、B1、B0、及びA0の順序によって試験されることができる。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1番目の空間的周辺ブロックが確認され得るし、動きベクトル

10

20

30

40

50

ルは、適用される動きシフトで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。

【0144】

第2のステップでは、第1のステップで確認された動きシフトがc o lピクチャからサブブロックレベル動き情報(動きベクトル及び参照インデックス)を取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されることができる。例えば、動きシフトが図8のA1の動きとして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c o l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列するために適用されることができる。

10

【0145】

S b T V M P候補及びアフィンマージ候補を全て含む組み合わせられたサブブロック基盤のマージリストは、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、サブブロック基盤のマージモードと呼ばれることができる。S b T V M Pモードは、S P S (sequence parameter set)に含まれたフラグにより利用可能または利用不可能であることができる。S b T V M Pモードが利用可能である場合、S b T V M P予測子は、サブブロック基盤のマージ候補のリストの第1のエントリとして追加されることができ、アフィンマージ候補が次にくることができる。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個であることができる。

20

【0146】

S b T V M Pで使用されるサブC U (または、サブブロック)のサイズは、8×8に固定されることができ、アフィンマージモードと同様に、S b T V M Pモードは、幅及び高さが共に8以上であるブロックにのみ適用されることができる。追加的なS b T V M Pマージ候補のエンコードロジックは、他のマージ候補と同一であることができる。すなわち、PまたはBスライス内の各C Uに対して追加的なR D (rate-distortion)コストを利用するR DチェックがS b T V M P候補を利用するか決定するために行われることができる。

【0147】

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サンプル単位を指す場合、補間(interpolation)手順が行われ得るし、これを介して参照ピクチャ内で分数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され得る。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M Vに基づいて予測サンプルを生成できる。双(B i)予測が適用される場合、L 0予測(すなわち、参照ピクチャリストL 0内の参照ピクチャとM V L 0を用いた予測)に基づいて導出された予測サンプルとL 1予測(すなわち、参照ピクチャリストL 1内の参照ピクチャとM V L 1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)加重合計(weighted sum)または加重平均(weighted average)を介し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として用いられることができる。ここで、L 0方向の動きベクトルは、L 0動きベクトルまたはM V L 0と呼ばれることができ、L 1方向の動きベクトルは、L 1動きベクトルまたはM V L 1と呼ばれることができる。双(B i)予測が適用される場合、L 0予測に用いられた参照ピクチャとL 1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、双予測でありながら、両方向予測に該当する場合)、これをトゥルー(true)双予測と呼ぶことができる。

30

40

【0148】

50

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され得るし、その後、インループフィルタリングなどの手順が行われ得ることは、前述したとおりである。

【0149】

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できる。例えば、加重平均を利用した双予測は、BCW (Bi-prediction with CU-level Weight)、BWA (Bi-prediction with Weighted Average)、または加重平均双予測 (weighted averaging bi-prediction) と呼ばれることができる。

【0150】

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) との単純平均によって導出されることができた。すなわち、双予測サンプルは、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 予測サンプルとの平均で導出された。ただし、双予測が適用される場合、次のように、L0 予測信号と L1 予測信号との加重平均によって双予測信号 (双予測サンプル) を導出することもできる。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

【0151】

【数 3】

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

【0152】

数式 3 において、 $P_{\text{bi-pred}}$ 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表すことができ、 w は、加重値を表すことができる。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表すことができ、 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表すことができる。

【0153】

例えば、加重平均双予測では、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、または 10 を含むことができる。すなわち、加重値 (w) は、-2、3、4、5、または 10 を含む加重値候補のうち 1 つに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、1 つにより決定されることができる。1 番目の方法は、併合されていない 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) 以後にシグナリングされることができる。2 番目の方法は、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。

【0154】

例えば、加重平均双予測は、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 より大きいか、同一である場合に加重平均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るし、低遅延でない (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、または 5 を含むことができる。

【0155】

例えば、エンコード装置において、早いサーチアルゴリズムは、エンコード装置の複雑性 (complexity) を大きく増加させないながら、加重値インデックスを探す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、次のように要約されることができる。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 が利用される

とき)、現在ピクチャが低遅延ピクチャである場合、同一でない加重値は、 $1 - pel$ 及び $4 - pel$ 動きベクトル精密度に対して条件付きでチェックされることができる。例えば、アフィン (a f f i n e) 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利用されるとき)、アフィン予測モードが現在ベストモード (b e s t m o d e) と選択された場合、アフィンME (M o t i o n E s t i m a t i o n) は、同一でない加重値に対して行われることができる。例えば、双予測の2個の参照ピクチャが同一である場合、同一でない加重値が条件付きで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 (q u a n t i z a t i o n p a r a m e t e r) 、及び時間的レベル (t e m p o r a l l e v e l) に依存して特定条件が満たされる場合、同一でない加重値がサーチされないことがある。
【0156】

10

例えば、BCW加重値インデックス (または、加重値インデックス) は、1つのコンテキストコーディングされたビン (c o n t e x t c o d e d b i n) 及び後続するバイパスコーディングされたビン (b y p a s s c o d e d b i n) を用いてコーディングされることができる。1番目のコンテキストコーディングされたビンは、同じ加重値が使用されるか否かを表すことができる。1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ない加重値が使用される場合、使用される同一で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され得る。

【0157】

一方、双予測が適用される場合、予測サンプルを生成するのに利用される加重値情報は、マージ候補リストに含まれた候補のうち、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ことができる。

20

【0158】

本文書の一実施形態によれば、マージモードのための動きベクトル候補構成の際、時間的動きベクトル候補に対する加重値インデックス情報は、下記のように導出することができる。例えば、時間的動きベクトル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情報を導出できる。すなわち、インター予測タイプが双 (b i) 予測である場合、マージ候補リスト内の時間的マージ候補 (または、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導出できる。

【0159】

30

例えば、時間的動きベクトル候補に対して加重平均のための加重値インデックス情報は、常に0に導出されることができる。ここで、加重値インデックス情報が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することができる。例えば、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順は、次の表のとおりでありうる。

【0160】

40

50

【表 1 - 1】

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.

【0 1 6 1】

【表 1 - 2】

3. 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 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:

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

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

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

$$\text{gbIdxCol} = 0$$

(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:

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

【0 1 6 2】

10

20

30

40

50

【表 1 - 3】

	$\text{predFlagL1Col} = \text{availableFlagL1Col}$	(8-287)
5.	The merging candidate list, mergeCandList , is constructed as follows: $i = 0$ if(availableFlagA_1) $\text{mergeCandList}[i++] = A_1$ if(availableFlagB_1) $\text{mergeCandList}[i++] = B_1$ if(availableFlagB_0) $\text{mergeCandList}[i++] = B_0$ if(availableFlagA_0) $\text{mergeCandList}[i++] = A_0$ if(availableFlagB_2) $\text{mergeCandList}[i++] = B_2$ if(availableFlagCol) $\text{mergeCandList}[i++] = \text{Col}$	(8-288)
6.	The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the mergeCandList .	
7.	When numCurrMergeCand is less than $(\text{MaxNumMergeCand} - 1)$ 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 $\text{predFlagL0avgCand}_k$ and $\text{predFlagL1avgCand}_k$ 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 $(\text{numCurrMergeCand} - \text{numOrigMergeCand})$. When numAvgMergeCand is greater than 0, k ranges from 0 to $\text{numAvgMergeCand} - 1$, inclusive.	

【0 1 6 3】

【表 1 - 4】

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 $\text{refIdxL0zeroCand}_m$ and $\text{refIdxL1zeroCand}_m$, the prediction list utilization flags $\text{predFlagL0zeroCand}_m$ and $\text{predFlagL1zeroCand}_m$ 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 $(\text{numCurrMergeCand} - \text{numOrigMergeCand} - \text{numAvgMergeCand})$. When numZeroMergeCand is greater than 0, m ranges from 0 to $\text{numZeroMergeCand} - 1$, inclusive.	
10.	The variable mergeIdxOffset is set equal to 0.	
11.	When $\text{mmvd_flag}[\text{xCb}][\text{yCb}]$ is equal to 1, the variable mmvdCnt is set equal to 0 and the following applies until mmvdCnt is greater than $(\text{merge_idx}[\text{xCb}][\text{yCb}] + \text{mergeIdxOffset})$ or mmvdCnt is equal to MaxNumMergeCand : - When candidate $\text{mergeCandList}[\text{mmvdCnt}]$ uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1. - The variable mmvdCnt is incremented by 1.	

【0 1 6 4】

10

20

30

40

50

【表 1 - 5】

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)
$\text{gbIdx} = \text{gbIdxN}$	(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

【0 1 6 5】

前記表 1 を参照すれば、 gbIdx は、双予測加重値インデックスを表すことができ、 gbIdxCol 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候補）に対する双予測加重値インデックスを表すことができる。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順（8.4.2.2 の目次 3）で前記 gbIdxCol は、0 に導出されることができる。すなわち、時間的動きベクトル候補の加重値インデックスは、0 に導出されることができる。

【0 1 6 6】

または、時間的動きベクトル候補に対する加重平均のための加重値インデックスは、コロケートド（collocated）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、コロケートドブロックは、コル（col）ブロック、同一位置ブロック、または同一位置参照ブロックと呼ばれることができ、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表すことができる。例えば、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順は、次の表のとおりでありうる。

30

【0 1 6 7】

40

50

【表 2 - 1】

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.

【0 1 6 8】

【表 2 - 2】

3. 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 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:

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

【0 1 6 9】

10

20

30

40

50

【表 2 - 3】

predFlagL1Col = availableFlagL1Col	(8-287)
gbiIdxCol = gbiIdxCol	(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>	(8-288)
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.	

【 0 1 7 0 】

【表 2 - 4】

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 gbiIdx 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.	30
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 gbiIdx 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.	40
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:	

【 0 1 7 1 】

10

20

30

40

50

【表 2 - 5】

- When candidate <code>mergeCandList[mmvdCnt]</code> uses the current decoded picture as its reference picture, <code>mergeIdxOffset</code> is incremented by 1. - The variable <code>mmvdCnt</code> is incremented by 1.	
12. The following assignments are made with <code>N</code> being the candidate at position <code>merge_idx[xCb][yCb] + mergeIdxOffset</code> in the merging candidate list <code>mergeCandList</code> (<code>N = mergeCandList[merge_idx[xCb][yCb] + mergeIdxOffset]</code>) and <code>X</code> being replaced by 0 or 1:	
$\text{refIdxLX} = \text{refIdxLXN} \quad (8-289)$	10
$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-290)$	
$\text{mvLX}[0][0][0] = \text{mvLXN}[0] \quad (8-291)$	
$\text{mvLX}[0][0][1] = \text{mvLXN}[1] \quad (8-292)$	
$\text{gbIdx} = \text{gbIdxN} \quad (8-293)$	
13. When <code>mmvd_flag[xCb][yCb]</code> 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 (<code>xCb, yCb</code>), the luma motion vectors <code>mvL0[0][0]</code>, <code>mvL1[0][0]</code>, the reference indices <code>refIdxL0</code>, <code>refIdxL1</code> and the prediction list utilization flags <code>predFlagL0[0][0]</code> and <code>predFlagL1[0][0]</code> as inputs, and the motion vector differences <code>mMvdL0</code> and <code>mMvdL1</code> as outputs.	
- The motion vector difference <code>mMvdLX</code> is added to the merge motion vectors <code>mvLX</code> for <code>X</code> being 0 and 1 as follows:	20
$\text{mvLX}[0][0][0] += \text{mMvdLX}[0] \quad (8-294)$	
$\text{mvLX}[0][0][1] += \text{mMvdLX}[1] \quad (8-295)$	

【 0 1 7 2 】

前記表 2 を参照すれば、`g b i I d x` は、双予測加重値インデックスを表すことができ、`g b i I d x C o l` 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候補）に対する双予測加重値インデックスを表すことができる。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順でスライスのタイプまたはタイルグループのタイプが B である場合（8.4.2.2 の目次 4）には、前記 `g b i I d x C o l` が `g b i I d x C o l` で導出されることができる。すなわち、時間的動きベクトル候補の加重値インデックスは、コル（`c o l`）ブロックの加重値インデックスで導出されることができる。

【 0 1 7 3 】

一方、本文書の他の実施形態によれば、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候補構成の際、時間的動きベクトル候補に対する加重平均のための加重値インデックスを導出できる。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ることができる。時間的動きベクトル候補は、サブブロック基盤の時間的動きベクトル候補を表すことができ、`S b T M V P`（または、`A T M V P`）候補と呼ばれることもできる。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またはサブブロックマージ候補リスト内の `S b T M V P` 候補（または、サブブロック基盤の時間的動きベクトル候補）に対する加重値インデックス情報を導出できる。

【 0 1 7 4 】

例えば、サブブロック基盤の時間的動きベクトル候補に対する加重平均のための加重値インデックス情報は、常に 0 に導出されることができる。ここで、加重値インデックス情報が 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することができる。例えば、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順及びサブブロック基盤の時間的マ

10

20

30

40

50

ージ候補を導出する手順は、各々次の表 3 及び表 4 のとおりでありうる。

【 0 1 7 5 】

【表 3 - 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 $gbiIdx$.	10
The variables $numSbX$, $numSbY$ and the subblock merging candidate list, $subblockMergeCandList$ are derived by the following ordered steps:	20
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.	

【 0 1 7 6 】

【表 3 - 2】

– 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 $gbiIdxSbCol$, 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:	40
$(x_{A_0}, y_{A_0}) = (x_{Cb} - 1, y_{Cb} + cbHeight)$	(8-536)
$(x_{A_1}, y_{A_1}) = (x_{Cb} - 1, y_{Cb} + cbHeight - 1)$	(8-537)
$(x_{A_2}, y_{A_2}) = (x_{Cb} - 1, y_{Cb})$	(8-538)
$(x_{B_0}, y_{B_0}) = (x_{Cb} + cbWidth, y_{Cb} - 1)$	(8-539)
$(x_{B_1}, y_{B_1}) = (x_{Cb} + cbWidth - 1, y_{Cb} - 1)$	(8-540)
$(x_{B_2}, y_{B_2}) = (x_{Cb} - 1, y_{Cb} - 1)$	(8-541)
$(x_{B_3}, y_{B_3}) = (x_{Cb}, y_{Cb} - 1)$	(8-542)

【 0 1 7 7 】

10

20

30

40

50

【表 3 - 3】

	$\text{numSbX} = \text{cbWidth} \gg 2$	(8-543)
	$\text{numSbY} = \text{cbHeight} \gg 2$	(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:	
	$\text{predFlagLXA} = \text{PredFlagLX}[\text{xNbA}_k][\text{yNbA}_k]$	(8-545)
	$\text{refIdxLXA} = \text{RefIdxLX}[\text{xNbA}_k][\text{yNbA}_k]$	(8-546)

【0 1 7 8】

【表 3 - 4】

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:	

【0 1 7 9】

10

20

30

40

50

【表 3 - 5】

$$\text{predFlagLXB} = \text{PredFlagLX}[xNbB_k][yNbB_k] \quad (8-547)$$

$$\text{refIdxLXB} = \text{RefIdxLX}[xNbB_k][yNbB_k] \quad (8-548)$$

5. When `sps_affine_enabled_flag` 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 (`xCb`, `yCb`), the luma coding block width and height (`cbWidth`, `cbHeight`), the availability flags `availableA0`, `availableA1`, `availableA2`, `availableB0`, `availableB1`, `availableB2`, `availableB3` as inputs, and the availability flags `availableFlagConstK`, the reference indices `refIdxLXConstK`, prediction list utilization flags `predFlagLXConstK`, motion model indices `motionModelIdxConstK` and `cpMvpLXConstK[cpIdx]` with `X` being 0 or 1, `K` = 1..6, `cpIdx` = 0..2 as outputs and `gbIdxConstK` is set equal to 0 with `K` = 1..6..
6. The initial subblock merging candidate list, `subblockMergeCandList`, is constructed as follows:

```

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

```

【 0 1 8 0 】

【表 3 - 6】

7. The variable `numCurrMergeCand` and `numOrigMergeCand` are set equal to the number of merging candidates in the `subblockMergeCandList`.
8. When `numCurrMergeCand` is less than `MaxNumSubblockMergeCand`, the following is repeated until `numCurrMergeCand` is equal to `MaxNumSubblockMergeCand`, with `mvZero[0]` and `mvZero[1]` both being equal to 0:
- The reference indices, the prediction list utilization flags and the motion vectors of `zeroCandm` with `m` equal to (`numCurrMergeCand` - `numOrigMergeCand`) are derived as follows:

$$\text{refIdxL0ZeroCand}_m = 0 \quad (8-550)$$

$$\text{predFlagL0ZeroCand}_m = 1 \quad (8-551)$$

$$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero} \quad (8-552)$$

$$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero} \quad (8-553)$$

$$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero} \quad (8-554)$$

$$\text{refIdxL1ZeroCand}_m = (\text{tile_group_type} == B) ? 0 : -1 \quad (8-555)$$

$$\text{predFlagL1ZeroCand}_m = (\text{tile_group_type} == B) ? 1 : 0 \quad (8-556)$$

$$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero} \quad (8-557)$$

$$\text{cpMvL1ZeroCand}_m[1] = \text{mvZero} \quad (8-558)$$

$$\text{cpMvL1ZeroCand}_m[2] = \text{mvZero} \quad (8-559)$$

$$\text{motionModelIdxZeroCand}_m = 1 \quad (8-560)$$

$$\text{gbIdxZeroCand}_m = 0 \quad (8-561)$$

【 0 1 8 1 】

10

20

30

40

50

【表 3 - 7】

The candidate zeroCand_m with m equal to $(\text{numCurrMergeCand} - \text{numOrigMergeCand})$ is added at the end of $\text{subblockMergeCandList}$ and numCurrMergeCand is incremented by 1 as follows:	(8-562)
The variables refIdxL0, refIdxL1, $\text{predFlagL0}[xSbIdx][ySbIdx]$, $\text{predFlagL1}[xSbIdx][ySbIdx]$, $\text{mvL0}[xSbIdx][ySbIdx]$, $\text{mvL1}[xSbIdx][ySbIdx]$, $\text{mvCL0}[xSbIdx][ySbIdx]$, and $\text{mvCL1}[xSbIdx][ySbIdx]$ with $xSbIdx = 0 \dots \text{numSbX} - 1$, $ySbIdx = 0 \dots \text{numSbY} - 1$ are derived as follows:	
If $\text{subblockMergeCandList}[\text{merge_subblock_idx}[xCb][yCb]]$ is equal to SbCol, the bi-prediction weight index gbIdx is set equal to 0 and the following applies with X being 0 or 1:	(8-563)
For $xSbIdx = 0 \dots \text{numSbX} - 1$, $ySbIdx = 0 \dots \text{numSbY} - 1$, the following applies:	
$\text{predFlagLX}[xSbIdx][ySbIdx] = \text{predFlagLXSbCol}[xSbIdx][ySbIdx]$	(8-564)
$\text{mvLX}[xSbIdx][ySbIdx][0] = \text{mvLXSbCol}[xSbIdx][ySbIdx][0]$	(8-565)
$\text{mvLX}[xSbIdx][ySbIdx][1] = \text{mvLXSbCol}[xSbIdx][ySbIdx][1]$	(8-566)
When $\text{predFlagLX}[xSbIdx][ySbIdx]$ is equal to 1, the derivation process for chroma motion vectors in clause 8.4.2.13 is invoked with $\text{mvLX}[xSbIdx][ySbIdx]$ and refIdxLX as inputs, and the output being $\text{mvCLX}[xSbIdx][ySbIdx]$.	
The following assignment is made for $x = xCb \dots xCb + \text{cbWidth} - 1$ and $y = yCb \dots yCb + \text{cbHeight} - 1$:	(8-567)
Otherwise ($\text{subblockMergeCandList}[\text{merge_subblock_idx}[xCb][yCb]]$ is not equal to SbCol), the following applies with X being 0 or 1:	

10

20

【0 1 8 2】

【表 3 - 8】

The following assignments are made with N being the candidate at position $\text{merge_subblock_idx}[xCb][yCb]$ in the subblock merging candidate list $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[xCb][yCb]]$):	(8-568)
$\text{refIdxLX} = \text{refIdxLXN}$	(8-569)
$\text{predFlagLX}[0][0] = \text{predFlagLXN}$	(8-570)
$\text{cpMvLX}[0] = \text{cpMvLXN}[0]$	(8-571)
$\text{cpMvLX}[1] = \text{cpMvLXN}[1]$	(8-572)
$\text{cpMvLX}[2] = \text{cpMvLXN}[2]$	(8-573)
$\text{numCpMv} = \text{motionModelIdxN} + 1$	(8-574)
$\text{gbIdx} = \text{gbIdxN}$	(8-575)
For $xSbIdx = 0 \dots \text{numSbX} - 1$, $ySbIdx = 0 \dots \text{numSbY} - 1$, the following applies:	(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 $\text{mvLX}[xSbIdx][ySbIdx]$ and the chroma subblock motion vector array $\text{mvCLX}[xSbIdx][ySbIdx]$ with $xSbIdx = 0 \dots \text{numSbX} - 1$, $ySbIdx = 0 \dots \text{numSbY} - 1$ as outputs.	(8-576)
The following assignment is made for $x = xCb \dots xCb + \text{cbWidth} - 1$ and $y = yCb \dots yCb + \text{cbHeight} - 1$:	(8-576)
$\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1$	(8-576)

30

40

【0 1 8 3】

50

【表 4】

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 $gbiIdxSbCol$.
The $gbiIdxSbCol$ is set equal to 0.	
...	

【0184】

前記表 3 及び表 4 を参照すれば、 $gbiIdx$ は、双予測加重値インデックスを表すことができ、 $gbiIdxSbCol$ は、サブブロック基盤の時間的マージ候補（例えば、サブブロック基盤のマージ候補リスト内の時間的動きベクトル候補）に対する双予測加重値インデックスを表すことができ、前記サブブロック基盤の時間的マージ候補を導出する手順（8.4.4.3）で前記 $gbiIdxSbCol$ は、0 に導出されることができる。すなわち、サブブロック基盤の時間的動きベクトル候補の加重値インデックスは、0 に導出されることができる。

【0185】

または、サブブロック基盤の時間的動きベクトル候補に対する加重平均のための加重値インデックス情報は、時間的センター（center）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。例えば、前記時間的センターブロックは、コル（col）ブロックまたはコル（col）ブロックのセンターに位置したサブブロックまたはサンプルを表すことができ、具体的には、コルブロックの中央の 4 個のサブブロックまたはサンプルのうち、右下側に位置したサブブロックまたはサンプルを表すことができる。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順、サブブロック基盤の時間的マージ候補を導出する手順、及びサブブロック基盤の時間的マージのためのベース動き情報を導出する手順は、各々次の表 5、表 6、及び表 7 のとおりでありうる。

【0186】

10

20

30

40

50

【表 5 - 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 $gbiIdx$.

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

1. When $sps_sbtmp_enabled_flag$ 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.

【 0 1 8 7 】

【表 5 - 2】

- 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 $gbiIdxSbCol$, 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:

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

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

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

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

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

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

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

【 0 1 8 8 】

10

20

30

40

50

【表 5 - 3】

	$\text{numSbX} = \text{cbWidth} \gg 2$	(8-543)
	$\text{numSbY} = \text{cbHeight} \gg 2$	(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:	
	$\text{predFlagLXA} = \text{PredFlagLX}[\text{xNbA}_k][\text{yNbA}_k]$	(8-545)
	$\text{refIdxLXA} = \text{RefIdxLX}[\text{xNbA}_k][\text{yNbA}_k]$	(8-546)

【 0 1 8 9 】

【表 5 - 4】

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 $(\text{CbPosX}[\text{xNbAB}][\text{yNbB}_k], \text{CbPosY}[\text{xNbB}_k][\text{yNbB}_k])$, 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 $(\text{cbWidth}, \text{cbHeight})$, 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:	

【 0 1 9 0 】

10

20

30

40

50

【表 5 - 5】

	$\text{predFlagLXB} = \text{PredFlagLX}[xNbB_k][yNbB_k]$	(8-547)
	$\text{refIdxLXB} = \text{RefIdxLX}[xNbB_k][yNbB_k]$	(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> .	

10

20

【 0 1 9 1】

【表 5 - 6】

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)

30

40

【 0 1 9 2】

50

【表 5 - 7】

- The candidate zeroCand_m with m equal to $(\text{numCurrMergeCand} - \text{numOrigMergeCand})$ is added at the end of $\text{subblockMergeCandList}$ and numCurrMergeCand is incremented by 1 as follows:	
$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$	
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 gbIdx is set equal to 0 and the following applies with X being 0 or 1:	
$\text{refIdxLX} = \text{refIdxLXSbCol} \quad (8-563)$	10
- 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 $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$:	20
$\text{MotionModelIdx}[x][y] = 0 \quad (8-567)$	
- 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:	

【0 1 9 3】

【表 5 - 8】

- 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 $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$):	
$\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$	30
$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$	
$\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$	
$\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$	
$\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$	
$\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$	
$\text{gbIdx} = \text{gbIdxN} \quad (8-574)$	
- 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.	40
- 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 9 4】

【表 6 – 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 $refIdxL0SbCol$ and $refIdxL1SbCol$,
- the luma motion vectors in 1/16 fractional-sample accuracy $mvL0SbCol[xSbIdx][ySbIdx]$ and $mvL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0 \dots numSbX - 1$, $ySbIdx = 0 \dots numSbY - 1$,
- the prediction list utilization flags $predFlagL0SbCol[xSbIdx][ySbIdx]$ and $predFlagL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0 \dots numSbX - 1$, $ySbIdx = 0 \dots numSbY - 1$,
- the bi-prediction weight index $gbIdxSbCol$.

The availability flag $availableFlagSbCol$ is derived as follows.

【 0 1 9 5 】

【表 6 – 2】

- 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_sbtmp_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 $ColPic$.

【 0 1 9 6 】

10

20

30

40

50

【表 6 - 3】

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 (xCtb, yCtb), the location (xColCtCb, yColCtCb), 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 gbIdxSbCol, 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.	10
When availableFlagSbCol is equal to 1, the following applies:	
- 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:	20

【 0 1 9 7 】

30

40

50

【表 6 - 4】

- The luma location (x_{Sb}, y_{Sb}) 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:	
$x_{Sb} = x_{Cb} + x_{SbIdx} * sbWidth \quad (8-586)$	
$y_{Sb} = y_{Cb} + y_{SbIdx} * sbHeight \quad (8-587)$	
- The location (x_{ColSb}, y_{ColSb}) of the collocated subblock inside ColPic is derived as follows.	
$x_{ColSb} = Clip3(x_{Ctb},$ $Min(CurPicWidthInSamplesY - 1, x_{Ctb} + (1 \ll CtbLog2SizeY) + 3),$ $x_{Sb} + (tempMv[0] \gg 4)) \quad (8-588)$	
$y_{ColSb} = Clip3(y_{Ctb},$ $Min(CurPicHeightInSamplesY - 1, y_{Ctb} + (1 \ll CtbLog2SizeY) - 1),$ $y_{Sb} + (tempMv[1] \gg 4)) \quad (8-589)$	10
- 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 (($x_{ColSb} \gg 3$) $\ll$ 3, ($y_{ColSb} \gg 3$) $\ll$ 3) inside the ColPic.	
- The luma location (x_{ColCb}, y_{ColCb}) 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, (x_{ColCb}, y_{ColCb}), 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 $mvL0SbCol[x_{SbIdx}][y_{SbIdx}]$ and availableFlagL0SbCol.	20
- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, (x_{ColCb}, y_{ColCb}), 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 $mvL1SbCol[x_{SbIdx}][y_{SbIdx}]$ and availableFlagL1SbCol.	
- When availableFlagL0SbCol and availableFlagL1SbCol are both equal to 0, the following applies for X being 0 and 1:	
$mvLXSbCol[x_{SbIdx}][y_{SbIdx}] = ctrMvLX \quad (8-590)$	
$predFlagLXSbCol[x_{SbIdx}][y_{SbIdx}] = ctrPredFlagLX \quad (8-591)$	

【 0 1 9 8 】

10

20

30

40

50

【表 7 - 1】

8.4.4.4 Derivation process for subblock-based temporal merging base motion data

Inputs to this process are:

- the location (x_{Ctb} , y_{Ctb}) of the top-left sample of the luma coding tree block that contains the current coding block,
- the location ($x_{ColCtrCb}$, $y_{ColCtrCb}$) of the top-left sample of the collocated luma coding block that covers the below-right center sample,
- 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 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:

$$tempMv[0] = 0 \quad (8-592)$$

$$tempMv[1] = 0 \quad (8-593)$$

The variable $currPic$ specifies the current picture.

The variable $availableFlagN$ is set equal to FALSE, and the following applies:

- When $availableFlagA_1$ is equal to 1, the following applies:
 - $availableFlagN$ is set equal to TRUE,
 - $refIdxLXN$ is set equal to $refIdxLXA_0$ and $mvLXN$ is set equal to $mvLXA_0$, for X being replaced by 0 and 1.
- When $availableFlagN$ is equal to FALSE and $availableFlagLB_1$ is equal to 1, the following applies:
 - $availableFlagN$ is set equal to TRUE,

【 0 1 9 9 】

10

20

30

40

50

【表 7 - 2】

- 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

【0 2 0 0】

【表 7 - 3】

The location (xColCb, yColCb) of the collocated block inside ColPic is derived as follows.	
$xColCb = Clip3(xCtb, Min(CurPicWidthInSamplesY - 1, xCtb + (1 \ll CtbLog2SizeY) + 3), (8-59$ $4) xColCtrCb + (tempMv[0] \gg 4))$	30
$yColCb = Clip3(yCtb, Min(CurPicHeightInSamplesY - 1, yCtb + (1 \ll CtbLog2SizeY) - 1), (8-5$ $95) yColCtrCb + (tempMv[1] \gg 4))$	
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:	
- 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 gbiIdxcolCb.	40

【0 2 0 1】

【表 7 - 4】

- The derivation process for temporal motion vector prediction in subclause 8.4.2.12 is invoked with <code>currCb</code>, <code>colCb</code>, <code>(xCb, yCb)</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>(xCb, yCb)</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

【0202】

前記表 5、表 6、及び表 7 を参照すれば、`gbiIdx` は、双予測加重値インデックスを表すことができ、`gbiIdxSbCol` は、サブブロック基盤の時間的マージ候補（例えば、サブブロック基盤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表すことができる。サブブロック基盤の時間的マージのためのベース動き情報を導出する手順（8.4.4.4）で前記 `gbiIdxSbCol` は、`gbiIdxColCb` で導出されることができる。すなわち、サブブロック基盤の時間的動きベクトル候補の加重値インデックスは、時間的センタブロックの加重値インデックスで導出されることができる。例えば、前記時間的センタブロックは、コル（`col`）ブロックまたはコル（`col`）ブロックのセンターに位置したサブブロックまたはサンプルを表すことができ、具体的には、コルブロックの中央の 4 個のサブブロックまたはサンプルのうち、右下側に位置したサブブロックまたはサンプルを表すことができる。

20

【0203】

または、サブブロック基盤の時間的動きベクトル候補に対する加重平均のための加重値インデックス情報は、それぞれのサブブロック単位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、サブブロックが利用可能でない場合、時間的センター（`center`）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。例えば、前記時間的センタブロックは、コル（`col`）ブロックまたはコル（`col`）ブロックのセンターに位置したサブブロックまたはサンプルを表すことができ、具体的には、コルブロックの中央の 4 個のサブブロックまたはサンプルのうち、右下側に位置したサブブロックまたはサンプルを表すことができる。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順、サブブロック基盤の時間的マージ候補を導出する手順、及びサブブロック基盤の時間的マージのためのベース動き情報を導出する手順は、次の表 8、表 9、及び表 10 のとおりでありうる。

30

【0204】

40

50

【表 8 - 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 $sps_sbtmp_enabled_flag$ 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.	

【 0 2 0 5 】

【表 8 - 2】

– 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}), and the variables $numSbX$ and $numSbY$ are derived as follows:		
(x_{A_0} , y_{A_0}) = ($x_{Cb} - 1$, $y_{Cb} + cbHeight$)	(8-536)	
(x_{A_1} , y_{A_1}) = ($x_{Cb} - 1$, $y_{Cb} + cbHeight - 1$)	(8-537)	
(x_{A_2} , y_{A_2}) = ($x_{Cb} - 1$, y_{Cb})	(8-538)	
(x_{B_0} , y_{B_0}) = ($x_{Cb} + cbWidth$, $y_{Cb} - 1$)	(8-539)	
(x_{B_1} , y_{B_1}) = ($x_{Cb} + cbWidth - 1$, $y_{Cb} - 1$)	(8-540)	
(x_{B_2} , y_{B_2}) = ($x_{Cb} - 1$, $y_{Cb} - 1$)	(8-541)	
(x_{B_3} , y_{B_3}) = (x_{Cb} , $y_{Cb} - 1$)	(8-542)	

【 0 2 0 6 】

10

20

30

40

50

【表 8 - 3】

$$\text{numSbX} = \text{cbWidth} \gg 2$$

(8-543)

$$\text{numSbY} = \text{cbHeight} \gg 2$$

(8-544)

3. When `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 (x_{Curr}, y_{Curr}) set equal to (x_{Cb}, y_{Cb}) and the neighbouring luma location $(xNbA_k, yNbA_k)$ as inputs, and the output is assigned to the block availability flag `availableAk`.

– When `availableAk` is equal to TRUE and `MotionModelIdc[ xNbAk ][ yNbAk ]` 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 `MotionModelIdc[ xNbAk ][ yNbAk ]`, (xNb, yNb) is set equal to $(CbPosX[xNbA_k][yNbA_k], CbPosY[xNbA_k][yNbA_k])$, `nbW` is set equal to `CbWidth[ xNbA_k ][ yNbA_k ]`, `nbH` is set equal to `CbHeight[ xNbA_k ][ yNbA_k ]`, `numCpMv` is set equal to `MotionModelIdc[ xNbA_k ][ yNbA_k ] + 1`, and `gbIdxA` is set equal to `GbiIdx[ xNbA_k ][ yNbA_k ]`.

10

– For `X` being replaced by either 0 or 1, the following applies:

– When `PredFlagLX[ xNbA_k ][ 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 (x_{Cb}, y_{Cb}) , 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 `cpMvLXA[ cpIdx ]` with `cpIdx = 0 .. numCpMv - 1` as output.

20

– The following assignments are made:

$$\text{predFlagLXA} = \text{PredFlagLX}[xNbA_k][yNbA_k]$$

(8-545)

$$\text{refIdxLXA} = \text{RefIdxLX}[xNbA_k][yNbA_k]$$

(8-546)

【 0 2 0 7 】

【表 8 - 4】

4. When `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 $(xNbB_1, yNbB_1)$:

– 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 (x_{Curr}, y_{Curr}) set equal to (x_{Cb}, y_{Cb}) and the neighbouring luma location $(xNbB_k, yNbB_k)$ as inputs, and the output is assigned to the block availability flag `availableBk`.

30

– When `availableBk` is equal to TRUE and `MotionModelIdc[ xNbB_k ][ 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 `MotionModelIdc[ xNbB_k ][ yNbB_k ]`, (xNb, yNb) is set equal to $(CbPosX[xNbAB][yNbB_k], CbPosY[xNbB_k][yNbB_k])$, `nbW` is set equal to `CbWidth[ xNbB_k ][ yNbB_k ]`, `nbH` is set equal to `CbHeight[ xNbB_k ][ yNbB_k ]`, `numCpMv` is set equal to `MotionModelIdc[ xNbB_k ][ yNbB_k ] + 1`, and `gbIdxB` is set equal to `GbiIdx[ xNbB_k ][ yNbB_k ]`.

– For `X` being replaced by either 0 or 1, the following applies:

– When `PredFlagLX[ xNbB_k ][ 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 (x_{Cb}, y_{Cb}) , 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 `cpMvLXB[ cpIdx ]` with `cpIdx = 0 .. numCpMv - 1` as output.

40

– The following assignments are made:

【 0 2 0 8 】

50

【表 8 - 5】

	$\text{predFlagLXB} = \text{PredFlagLX}[\text{xNbB}_k][\text{yNbB}_k]$	(8-547)
	$\text{refIdxLXB} = \text{RefIdxLX}[\text{xNbB}_k][\text{yNbB}_k]$	(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

【 0 2 0 9】

【表 8 - 6】

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:	30
—	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)
		40

【 0 2 1 0】

【表 8 - 7】

The candidate zeroCand_m with m equal to $(\text{numCurrMergeCand} - \text{numOrigMergeCand})$ is added at the end of $\text{subblockMergeCandList}$ and numCurrMergeCand is incremented by 1 as follows: $\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$	
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 gbIdx 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:	

10

20

【0 2 1 1】

【表 8 - 8】

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 $\text{subblockMergeCandList}$ ($N = \text{subblockMergeCandList}[\text{merge_subblock_idx}[\text{xCb}][\text{yCb}]]$): $\text{refIdxLX} = \text{refIdxLXN} \quad (8-568)$ $\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-569)$ $\text{cpMvLX}[0] = \text{cpMvLXN}[0] \quad (8-570)$ $\text{cpMvLX}[1] = \text{cpMvLXN}[1] \quad (8-571)$ $\text{cpMvLX}[2] = \text{cpMvLXN}[2] \quad (8-572)$ $\text{numCpMv} = \text{motionModelIdxN} + 1 \quad (8-573)$ $\text{gbIdx} = \text{gbIdxN} \quad (8-574)$	
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 $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$: $\text{MotionModelIdx}[x][y] = \text{numCpMv} - 1 \quad (8-576)$	

30

40

【0 2 1 2】

50

【表 9 – 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 $refIdxL0SbCol$ 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 $gbIdxSbCol[xSbIdx][ySbIdx]$, the prediction list utilization flags $predFlagL0SbCol[xSbIdx][ySbIdx]$ and $predFlagL1SbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$.

【 0 2 1 3 】

【表 9 – 2】

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_sbtmp_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 $ColPic$.

【 0 2 1 4 】

10

20

30

40

50

【表 9 - 3】

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 (xCtb, yCtb), the location (xColCtCb, yColCtCb), 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 ctrgbIdx, 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: – 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:	10
	20

【 0 2 1 5 】

【表 9 - 4】

- 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: $\text{xSb} = \text{xCb} + \text{xSbIdx} * \text{sbWidth} \quad (8-586)$ $\text{ySb} = \text{yCb} + \text{ySbIdx} * \text{sbHeight} \quad (8-587)$	30
- The location (xColSb, yColSb) of the collocated subblock inside ColPic is derived as follows. $\begin{aligned} \text{xColSb} = & \text{Clip3}(\text{xCtb}, \\ & \text{Min}(\text{CurPicWidthInSamplesY} - 1, \text{xCtb} + (1 \ll \text{CtbLog2SizeY}) + 3), (8-588) \\ & \text{xSb} + (\text{tempMv}[0] \gg 4)) \end{aligned}$ $\begin{aligned} \text{yColSb} = & \text{Clip3}(\text{yCtb}, \\ & \text{Min}(\text{CurPicHeightInSamplesY} - 1, \text{yCtb} + (1 \ll \text{CtbLog2SizeY}) - 1), (8-589) \\ & \text{ySb} + (\text{tempMv}[1] \gg 4)) \end{aligned}$	
- 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 gbiIdxSbCol[xSbIdx][ySbIdx] is set equal to gbiIdxcolCb.	40

【 0 2 1 6 】

【表 9 – 5】

– 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.		
– 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:		
$mvLXSbCol[xSbIdx][ySbIdx] = ctrMvLX$	(8-590)	10
$predFlagLXSbCol[xSbIdx][ySbIdx] = ctrPredFlagLX$	(8-591)	
$gbIdxSbCol[xSbIdx][ySbIdx] = ctrgbIdx$	(x-xxx)	

【 0 2 1 7 】

【表 1 0 – 1】

8.4.4.4 Derivation process for subblock-based temporal merging base motion data		
Inputs to this process are:		20
– the location (xCtb, yCtb) of the top-left sample of the luma coding tree block that contains the current coding block,		
– the location (xColCtrCb, yColCtrCb) of the top-left sample of the collocated luma coding block that covers the below-right center sample.		
– 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:		30
– 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 ctrgbIdx.		
The variable tempMv is set as follows:		
$tempMv[0] = 0$	(8-592)	
$tempMv[1] = 0$	(8-593)	
The variable currPic specifies the current picture.		40
The variable availableFlagN is set equal to FALSE, and the following applies:		
– When availableFlagA ₁ is equal to 1, the following applies:		
– availableFlagN is set equal to TRUE,		

【 0 2 1 8 】

【表 1 0 - 2】

- 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.
 - 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 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.

10

20

【0 2 1 9】

【表 1 0 - 3】

- 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.
- The location (xColCb, yColCb) of the collocated block inside ColPic is derived as follows.
- $$\begin{aligned} xColCb = & \text{Clip3}(xCtb, \\ & \text{Min}(\text{CurPicWidthInSamplesY} - 1, xCtb + (1 \ll \text{CtbLog2SizeY}) + 3), (8-59 \\ & 4) \\ & xColCtCb + (\text{tempMv}[0] \gg 4)) \\ yColCb = & \text{Clip3}(yCtb, \\ & \text{Min}(\text{CurPicHeightInSamplesY} - 1, yCtb + (1 \ll \text{CtbLog2SizeY}) - 1), (8-5 \\ & 95) \\ & yColCtCb + (\text{tempMv}[1] \gg 4)) \end{aligned}$$
- The array colPredMode is set equal to the prediction mode array CuPredMode of the collocated picture specified by ColPic.
- 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 (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.

30

40

【0 2 2 0】

50

【表 10 - 4】

- 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. - 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. - Otherwise, the following applies:	
$\text{ctrPredFlagL0} = 0$	(8-596)
$\text{ctrPredFlagL1} = 0$	(8-597)

10

【0221】

前記表 8、表 9、及び表 10 を参照すれば、g b i I d x は、双予測加重値インデックスを表すことができ、g b i I d x S b C o l は、サブブロック基盤の時間的マージ候補（例えば、サブブロック基盤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表すことができる。サブブロック基盤の時間的マージのためのベース動き情報を導出する手順（8.4.4.3）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 で導出されることができる。または、条件（例えば、a v a i l a b l e F l a g L 0 S b C o l 及び a v a i l a b l e F l a g L 1 S b C o l が共に 0 である場合）によってサブブロック基盤の時間的マージのためのベース動き情報を導出する手順（8.4.4.3）で前記 g b i I d x S b C o l は、c t r g b i I d x で導出されることができる。サブブロック基盤の時間的マージのためのベース動き情報を導出する手順（8.4.4.4）で前記 c t r g b i I d x は、g b i I d x S b C o l で導出されることができる。すなわち、サブブロック基盤の時間的動きベクトル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で導出されることができる。例えば、前記時間的センタブロックは、コル（c o l）ブロックまたはコル（c o l）ブロックのセンターに位置したサブブロックまたはサンプルを表すことができ、具体的には、コルブロックの中央の 4 個のサブブロックまたはサンプルのうち、右下側に位置したサブブロックまたはサンプルを表すことができる。

20

30

【0222】

一方、本文書のさらに他の実施形態によれば、マージモードのための動きベクトル候補構成の際、ペアワイズ（p a i r - w i s e）候補に対する加重値インデックス情報を導出できる。例えば、マージ候補リストにペアワイズ候補が含まれ得るし、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情報が導出され得る。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、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導出できる。

40

【0223】

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 個のマージ候補（例えば、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る。

【0224】

例えば、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 個のマージ候補のうち、いずれか 1 つのマージ候補（例えば、マージ候補 c a n d 0 またはマージ候補

50

c a n d 1) 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。例えば、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 個のマージ候補のうち、双予測を使用する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ができる。

【0225】

または、前記他の 2 個のマージ候補のそれぞれの加重値インデックス情報が第 1 の加重値インデックス情報と同一である場合、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、前記第 1 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。一方、前記他の 2 個のマージ候補のそれぞれの加重値インデックス情報が互いに同一でない場合、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、デフォルト (D E F A U L T) 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。デフォルト加重値インデックス情報は、L 0 予測サンプル及び L 1 予測サンプルの各々に同じ加重値を付与する加重値インデックス情報に該当することができる。

10

【0226】

または、前記他の 2 個のマージ候補のそれぞれの加重値インデックス情報が第 1 の加重値インデックス情報と同一である場合、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、前記第 1 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。一方、前記他の 2 個のマージ候補のそれぞれの加重値インデックス情報が互いに同一でない場合、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、前記他の 2 個の候補のそれぞれの加重値インデックス情報のうち、デフォルト (D E F A U L T) 加重値インデックス情報でな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。デフォルト加重値インデックス情報は、L 0 予測サンプル及び L 1 予測サンプルの各々に同じ加重値を付与する加重値インデックス情報に該当することができる。

20

【0227】

一方、本文書のさらに他の実施形態によれば、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候補構成の際、時間的動きベクトル候補に対する加重平均のための加重値インデックス情報を導出できる。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ることができる。時間的動きベクトル候補は、サブブロック基盤の時間的動きベクトル候補を表すことができ、S b T M V P (または、A T M V P) 候補と呼ばれることもできる。前記 S b T M V P 候補に対する加重値インデックス情報は、現在ブロックの左側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、S b T M V P で導かれた候補が双予測を使用する場合、現在ブロックの左側周辺ブロックの加重値インデックスをサブブロック基盤のマージモードのための加重値インデックスで導出することができる。

30

【0228】

例えば、S b T M V P 候補は、現在ブロックの空間的に隣接した左側ブロック (または、左側周辺ブロック) を基準にコル (c o l) ブロックを導出できるので、前記左側周辺ブロックの加重値インデックスは、信頼されることができるとみなすことができる。これにより、前記 S b T M V P 候補に対する加重値インデックスは、左側周辺ブロックの加重値インデックスで導出されることができる。

【0229】

一方、本文書のさらに他の実施形態によれば、アフィンマージモードのための動きベクトル候補構成の際、アフィンマージ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情報を導出できる。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またはサブブロックマージ候補リスト内の候補に対する加重値インデックス情報を導出できる。

40

【0230】

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 (または、空間的周辺ブロック) または時間的 (t e m p o r a l) に隣接したブロック (または、時間的周辺ブロック) の動き情報に基づき、C P 0、C P 1、C P 2、または C P 3 候補

50

を導出してアフィンモデルでMVFを導出する候補を表すことができる。例えば、CP0は、現在ブロックの左上端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表すことができ、CP1は、現在ブロックの右上端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表すことができ、CP2は、現在ブロックの左下端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表すことができる。また、CP3は、現在ブロックの右下端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表すことができる。

【0231】

例えば、アフィンマージ候補のうち、構成された (constructed) アフィンマージ候補は、下記のように、現在ブロックの各コントロールポイントに対する動きベクトルの組み合わせで生成されることができる。アフィンマージ候補は、前記現在ブロックの左上側に位置するCP0 (Control Point 0)、前記現在ブロックの右上側に位置するCP1 (Control Point 1)、前記現在ブロックの左下側に位置するCP2 (Control Point 2)、及び前記現在ブロックの右下側に位置するCP3 (Control Point 3)の各々に対する動きベクトルのうち、少なくとも1つを含むことができる。

【0232】

例えば、アフィンマージ候補のうち、構成された (constructed) アフィンマージ候補は、{CP0、CP1、CP2}、{CP0、CP1、CP3}、{CP0、CP2、CP3}、{CP1、CP2、CP3}、{CP0、CP1}、及び{CP0、CP2}のように、現在ブロックの各コントロールポイントの組み合わせに基づいて生成されることができる。例えば、アフィンマージ候補は、{CPMV0、CPMV1、CPMV2}、{CPMV0、CPMV1、CPMV3}、{CPMV0、CPMV2、CPMV3}、{CPMV1、CPMV2、CPMV3}、{CPMV0、CPMV1}、及び{CPMV0、CPMV2}のうち、少なくとも1つを含むことができる。CPMV0、CPMV1、CPMV2、及びCPMV3は、CP0、CP1、CP2、及びCP3の各々に対する動きベクトルに該当することができる。

【0233】

一実施形態において、前記アフィンマージ候補が{CP0、CP1、CP2}、{CP0、CP1、CP3}、{CP0、CP2、CP3}、{CP0、CP1}、及び{CP0、CP2}のうち、いずれか1つに基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、CP0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基づいて導出されることができる。すなわち、前記アフィンマージ候補が現在ブロックの左上側に位置するCP0 (Control Point 0)に対するCPMV0を含む場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記CP0に対し第0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。一方、前記アフィンマージ候補が{CP1、CP2、CP3}に基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、CP1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基づいて導出されることができる。すなわち、前記アフィンマージ候補が{CP1、CP2、CP3}に基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、現在ブロックの右上側に位置するCP1 (Control Point 1)に対する第1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。

【0234】

上述した方法によれば、前記アフィンマージ候補に対する加重値インデックス情報は、{CPMV0、CPMV1、CPMV2}、{CPMV0、CPMV1、CPMV3}、{CPMV0、CPMV2、CPMV3}、{CPMV1、CPMV2、CPMV3}、{CPMV0、CPMV1}、及び{CPMV0、CPMV2}の各々に対して1番目のCPMVの導出のために使用される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。

【0235】

前記アフィンマージ候補に対する加重値インデックス情報を導出する他の実施形態によれば、現在ブロックの左上側に位置するC P 0に対する加重値インデックス情報と現在ブロックの右上側に位置するC P 1に対する加重値インデックス情報とが同一である場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記C P 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基づいて導出されることができる。一方、現在ブロックの左上側に位置するC P 0に対する加重値インデックス情報と現在ブロックの右上側に位置するC P 1に対する加重値インデックス情報とが同一でない場合、前記アフィンマージ候補に対する加重値インデックス情報は、デフォルト（D E F A U L T）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。前記デフォルト加重値インデックス情報は、前記L 0予測サンプル及び前記L 1予測サンプルの各々に同じ加重値を付与する加重値インデックス情報に該当することができる。

10

【0236】

前記アフィンマージ候補に対する加重値インデックス情報を導出するさらに他の実施形態によれば、前記アフィンマージ候補に対する加重値インデックス情報は、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で導出されることができる。例えば、C P 0候補ブロックのうち、C P 0における動きベクトルと決定された候補ブロックの加重値インデックス、C P 1候補ブロックのうち、C P 1における動きベクトルと決定された候補ブロックの加重値インデックス、C P 2候補ブロックのうち、C P 2における動きベクトルと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C P 3候補ブロックのうち、C P 3における動きベクトルと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アフィンマージ候補の加重値インデックスで導出することができる。

20

【0237】

例えば、前記コントロールポイントとしてC P 0及びC P 1が用いられ得るし、C P 0、C P 1、及びC P 2が用いられ得るし、C P 3は用いられないこともある。ただし、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C P 3候補を活用しようとするときには、上述した実施形態等で説明した時間的候補ブロックにおける加重値インデックスを導出する方法が活用され得る。

【0238】

図14及び図15は、本文書の実施形態（等）に係るビデオ／画像エンコード方法及び関連コンポーネントの一例を概略的に示す。

30

【0239】

図14において開示された方法は、図2または図15において開示されたエンコード装置によって行われることができる。具体的に、例えば、図14のS1400ないしS1420は、図15の前記エンコード装置200の予測部220によって行われることができ、図14のS1430は、図15の前記エンコード装置200のレジデュアル処理部230によって行われることができ、図14のS1440は、図15の前記エンコード装置200のエントロピーエンコード部240によって行われることができる。また、図14において図示していないが、図15において前記エンコード装置200の予測部220によって予測サンプルまたは予測関連情報を導出でき、前記エンコード装置200のレジデュアル処理部230によって原本サンプルまたは予測サンプルからレジデュアル情報が導出され得るし、前記エンコード装置200のエントロピーエンコード部240によってレジデュアル情報または予測関連情報からビットストリームが生成され得る。図14において開示された方法は、本文書において上述した実施形態を含むことができる。

40

【0240】

図14に示すように、エンコード装置は、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表すインター予測モード情報を生成できる（S1400）。例えば、エンコード装置は、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、またはサブブロックマージモードを決定でき、これを表すインター予測モード情報を生成できる。また、エンコード装置は、現在ブロックのインタ

50

一予測タイプを双予測 (bi-prediction) で表すインター予測タイプ情報を生成できる。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、または双 (bi) 予測のうち、双予測と決定されることができ、これを表すインター予測タイプ情報を生成できる。ここで、L0 予測は、参照ピクチャリスト0基盤の予測を表すことができ、L1 予測は、参照ピクチャリスト1基盤の予測を表すことができ、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基盤の予測を表すことができる。例えば、エンコード装置は、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できる。例えば、前記インター予測タイプ情報は、inter_pred_idc シンタックス要素を含むことができる。

【0241】

エンコード装置は、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できる (S1410)。例えば、エンコード装置は、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また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またはサブブロックマージ候補リストなどのように呼ばれることができるが、簡略にマージ候補リストと呼ばれることもできる。

【0242】

例えば、前記マージ候補リスト内の候補の個数が最大候補個数になるまでマージ候補リストに候補が挿入され得る。ここで、候補は、現在ブロックの動き情報 (または、動きベクトル) を導出するための候補または候補ブロックを表すことができる。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介し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先に探索されて (空間的マージ) 候補が導出され得るし、その後、時間的周辺ブロックが探索されて (時間的マージ) 候補と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にも、前記マージ候補リスト内の候補の個数が最大候補個数より少ない場合、追加候補を挿入できる。例えば、追加候補は、history based merge candidate (s)、pair-wise average merge candidate (s)、ATMVP、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うち、少なくとも1つを含むことができる。

【0243】

または、例えば、アフィンマージ候補リスト内の候補の個数が最大候補個数になるまでアフィンマージ候補リストに候補が挿入され得る。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。または、前記候補は、前記CPMVを導出するための候補または候補ブロックを表すこともできる。前記CPMVは、現在ブロックのCP (Control Point) における動きベクトルを表すことができる。例えば、CPは、2個、3個、または4個であることができ、現在ブロックの左上側 (または、左上側コーナー)、右上側 (または、右上側コーナー)、左下側 (または、左下側コーナー)、または右下側 (または、右下側コーナー) のうち、少なくとも一部に位置することができ、各位置毎に1つのCPのみが存在しうる。

【0244】

例えば、候補は、現在ブロックの周辺ブロック (または、現在ブロックのCPの周辺ブロック) に対する探索を介し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、継承された (inherited) アフィンマージ候補、構成された (constructed) アフィンマージ候補、またはゼロ (zero) 動きベクトル候補のうち、少なくとも1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、先に前記継承されたアフィンマージ候補を挿入でき、その後、構成されたアフィンマージ候補を挿

10

20

30

40

50

入できる。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個数が最大候補個数より小さい場合、残りは、ゼロ動きベクトル候補で満たすことができる。ここで、ゼロ動きベクトル候補は、ゼロベクトルと呼ばれることもできる。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ることができるが、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ることもできる。この場合、アフィンマージ候補リストは、サブブロックマージ候補リストと呼ばれることができ、サブブロックマージ候補リストは、S b T M V Pで導出された候補（または、S b T M V P候補）も含まれることができる。例えば、S b T M V P候補がサブブロックマージ候補リストに含まれる場合、サブブロックマージ候補リスト内で継承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することができる。

10

【0245】

エンコード装置は、マージ候補リストに含まれた候補のうち1つの候補を表す選択情報を生成できる（S 1 4 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、またはゼロベクトル候補のうち、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1つの候補を選択できる。または、例えば、サブブロックマージ候補リストは、継承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、またはゼロベクトル候補のうち、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1つの候補を選択できる。

20

【0246】

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表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また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できる。

【0247】

エンコード装置は、現在ブロックに対するレジデュアルサンプルに基づいてレジデュアル情報を生成できる（S 1 4 3 0）。例えば、エンコード装置は、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できる。例えば、エンコード装置は、レジデュアルサンプルの量子化された変換係数を表すレジデュアル情報を生成できる。レジデュアル情報は、指数ゴロム、C A V L C、C A B A Cなどのような様々なエンコード方法を介して生成されることができる。

30

【0248】

エンコード装置は、インター予測モード情報、選択情報、及びレジデュアル情報を含む画像情報をエンコードすることができる（S 1 4 4 0）。例えば、前記画像情報は、ビデオ情報と呼ばれることもできる。前記画像情報は、本文書の上述した実施形態（等）に係る様々な情報を含むことができる。例えば、画像情報は、予測関連情報またはレジデュアル関連情報のうち、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うち、少なくとも一部を含むことができる。例えば、エンコード装置は、上述した情報（または、シンタックス要素）の全部または一部を含む画像情報をエンコードしてビットストリームまたはエンコードされた情報を生成できる。または、ビットストリーム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またはエンコードされた情報は、ネットワークまたは格納媒体を介してデコード装置に送信されることができる。

40

【0249】

図14に図示していないが、例えば、エンコード装置は、現在ブロックの予測サンプルを生成できる。または、例えば、エンコード装置は、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できる。または、例えば、エンコード装置は、選択された候補に基づいて動き情報を導出でき、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できる。例えば、エンコード装置は、双予測によってL 0予測サンプル及びL 1予測サン

50

ブルを生成でき、前記 L 0 予測サンプル及び前記 L 1 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（または、加重値情報）を利用して前記 L 0 予測サンプル及び前記 L 1 予測サンプルから現在ブロックの予測サンプルを生成できる。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表されることができる。

【0250】

言い換えれば、例えば、エンコード装置は、選択された候補に基づいて現在ブロックの 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 が利用され得る。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表すことができ、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表すことができる。

10

【0251】

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 動き情報が導出され得るし、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され得る。

【0252】

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 動き情報が導出され得るし、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され得る。

20

【0253】

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 動き情報が導出され得るし、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され得る。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。

【0254】

または、例えば、前記マージ候補リストは、サブブロックマージ候補リストであることができ、アフィンマージ候補、サブブロックマージ候補、または SbTMVP 候補が選択されることもできる。ここで、サブブロック単位のアフィンマージ候補は、サブブロックマージ候補と呼ばれることもできる。

30

【0255】

例えば、前記候補は、サブブロック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マージ候補である場合、前記サブブロックマージ候補に基づいて L 0 動き情報及び L 1 動き情報が導出され得るし、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され得る。例えば、前記サブブロックマージ候補は、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。

40

【0256】

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、1 つのブロックに基づいて表されることができる。例えば、CP は、2 個、3 個、または 4 個であることができ、現在ブロックの左上側（または、左上側コーナー）、右上側（または、右上側コーナー）、左下側（または、左下側コーナー）、または右下側（または、右下側コーナー）のうち、少なくとも一部に位置することができ、各位置毎に 1 つの CP のみが存在しうる。

【0257】

50

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であることができる。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コーナー周辺ブロック、前記左上側コーナー周辺ブロックの下側に隣接した左側周辺ブロック、及び前記左上側コーナー周辺ブロックの右側に隣接した上側周辺ブロックを含むことができる。または、前記周辺ブロックは、図12においてA₂ブロック、B₂ブロック、またはB₃ブロックを含むことができる。

【0258】

または、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であることができる。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コーナー周辺ブロック及び前記右上側コーナー周辺ブロックの左側に隣接した上側周辺ブロックを含むことができる。または、前記周辺ブロックは、図12においてB₀ブロックまたはB₁ブロックを含むことができる。

10

【0259】

または、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であることができる。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コーナー周辺ブロック及び前記左下側コーナー周辺ブロックの上側に隣接した左側周辺ブロックを含むことができる。または、前記周辺ブロックは、図12においてA₀ブロックまたはA₁ブロックを含むことができる。

【0260】

または、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であることができる。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(c o l)ブロックまたは前記コールブロックの右下側コーナー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。または、前記周辺ブロックは、図12においてTブロックを含むことができる。

20

【0261】

または、例えば、前記候補は、S b T M V P 候補を含むことができる。例えば、前記選択された候補が前記S b T M V P 候補である場合、前記現在ブロックの左側周辺ブロックに基づいてL₀動き情報及びL₁動き情報が導出され得るし、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され得る。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。

30

【0262】

例えば、L₀動き情報は、L₀参照ピクチャインデックス及びL₀動きベクトルなどを含むことができ、L₁動き情報は、L₁参照ピクチャインデックス及びL₁動きベクトルなどを含むことができる。L₀参照ピクチャインデックスは、参照ピクチャリスト0で参照ピクチャを表す情報を含むことができ、L₁参照ピクチャインデックスは、参照ピクチャリスト1で参照ピクチャを表す情報を含むことができる。

【0263】

例えば、エンコード装置は、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表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表すことができる。例えば、前記加重値情報は、L₀予測サンプルまたは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表すことができ、前記加重平均に基づいて予測サンプルを生成する手順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個または5個の加重値のうち、いずれか1つの加重値を表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) またはBWA (Bi-prediction with Weighted Average) における加重平均を表すことができる。

40

50

【0264】

例えば、前記候補は、時間的マージ候補を含むことができ、前記時間的マージ候補に対する加重値インデックス情報は、0に表されることができる。すなわち、時間的マージ候補に対する加重値インデックス情報は、0に表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表すことができる。または、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表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表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。

10

【0265】

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利用されたマージ候補リスト内の他の2個の候補のうち1つの加重値インデックス情報に基づいて表されることができる。すなわち、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、前記ペアワイズ候補を導出するのに利用されたマージ候補リスト内の他の2個の候補のうち1つの加重値インデックス情報に基づいて表されることができる。

【0266】

例えば、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、他の2個の候補に基づいて表されることができる。前記他の2個の候補のそれぞれの加重値インデックス情報が第1の加重値インデックス情報と互いに同一である場合、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、前記第1の加重値インデックス情報に基づいて表されることができる。前記他の2個の候補のそれぞれの加重値インデックス情報が互いに同一でない場合、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、デフォルト（DEFAULT）加重値インデックス情報に基づいて表されることができる。このとき、前記デフォルト加重値インデックス情報は、前記L0予測サンプル及び前記L1予測サンプルの各々に同じ加重値を付与する加重値インデックス情報に該当することができる。

20

【0267】

例えば、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、他の2個の候補に基づいて表されることができる。前記他の2個の候補のそれぞれの加重値インデックス情報が第1の加重値インデックス情報と互いに同一である場合、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、前記第1の加重値インデックス情報に基づいて表されることができる。前記他の2個の候補のそれぞれの加重値インデックス情報が互いに同一でない場合、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、前記他の2個の候補のそれぞれの加重値インデックス情報のうち、デフォルト（DEFAULT）加重値インデックス情報でない加重値インデックス情報に基づいて表されることができる。前記デフォルト加重値インデックス情報は、前記L0予測サンプル及び前記L1予測サンプルの各々に同じ加重値を付与する加重値インデックス情報に該当することができる。

30

【0268】

例えば、前記マージ候補リストは、サブブロックマージ候補リストであることができ、アフィンマージ候補、サブブロックマージ候補、またはSbTMVP候補が選択されることもできる。ここで、サブブロック単位のアフィンマージ候補は、サブブロックマージ候補と呼ばれることもできる。

40

【0269】

例えば、前記候補は、アフィンマージ候補を含むことができ、前記アフィンマージ候補は、前記現在ブロックの左上側に位置するCP0（Control Point 0）、前

50

記現在ブロックの右上側に位置するCP1 (Control Point 1)、前記現在ブロックの左下側に位置するCP2 (Control Point 2)、及び前記現在ブロックの右下側に位置するCP3 (Control Point 3)の各々に対する動きベクトルのうち、少なくとも1つを含むことができる。

【0270】

例えば、前記アフィンマージ候補が {CP0、CP1、CP2}、{CP01、CP1、CP3}、{CP0、CP2、CP3}、{CP0、CP1}、及び{CP0、CP2}のうち、いずれか1つに基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記CP0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基づいて表されることができる。前記アフィンマージ候補が {CP1、CP2、CP3}に基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記CP1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基づいて表されることができる。

10

【0271】

前記CP0の周辺ブロックのうち、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に該当し、前記CP0の周辺ブロックは、前記現在ブロックの左上側コーナー周辺ブロック、前記左上側コーナー周辺ブロックの下側に隣接した左側周辺ブロック、及び前記左上側コーナー周辺ブロックの右側に隣接した上側周辺ブロックを含むことができる。

【0272】

前記CP1の周辺ブロックのうち、特定ブロックは、前記CP1に対するCPMVの導出のために使用されるブロックに該当し、前記CP1の周辺ブロックは、前記現在ブロックの右上側コーナー周辺ブロック及び前記右上側コーナー周辺ブロックの左側に隣接した上側周辺ブロックを含むことができる。

20

【0273】

または、例えば、前記候補は、SbTMVP候補を含むことができ、前記SbTMVP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周辺ブロックの加重値インデックス情報に基づいて表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、前記左側周辺ブロックの加重値インデックス情報に基づいて表されることができる。

30

【0274】

または、例えば、前記候補は、SbTMVP候補を含むことができ、前記SbTMVP候補に対する加重値インデックス情報は、0に表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に表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表すことができる。

【0275】

または、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ブロックの加重値インデックス情報に基づいて表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ブロックの加重値インデックス情報に基づいて表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、右下側サブブロックを含むことができる。

40

【0276】

または、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のそれぞれの加重値インデックス情報に基づいて表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のそれぞれの加重値インデックス

50

ス情報に基づいて表されることができる。

【0277】

例えば、エンコード装置は、上述した情報（または、シンタックス要素）の全部または一部を含む画像情報をエンコードしてビットストリームまたはエンコードされた情報を生成できる。または、ビットストリーム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またはエンコードされた情報は、ネットワークまたは格納媒体を介してデコード装置に送信されることができる。または、前記ビットストリームまたはエンコードされた情報は、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、前記ビットストリームまたは前記エンコードされた情報は、上述した画像エンコード方法により生成されることができる。

10

【0278】

図16及び図17は、本文書の実施形態（等）に係るビデオ／画像デコード方法及び関連コンポーネントの一例を概略的に示す。

【0279】

図16において開示された方法は、図3または図17において開示されたデコード装置によって行われることができる。例えば、図16のS1600は、図17において前記デコード装置300のエントロピーデコード部310によって行われることができ、図16のS1610は、図17において前記デコード装置300のレジデュアル処理部320によって行われることができる。また、図16のS1620ないしS1660は、図17において前記デコード装置300の予測部330によって行われることができ、図16のS1670は、図17において前記デコード装置300の加算部340によって行われることができる。また、図16において図示していないが、図17において前記デコード装置300のエントロピーデコード部310によってビットストリームから予測関連情報またはレジデュアル情報が導出され得る。図16において開示された方法は、本文書において上述した実施形態を含むことができる。

20

【0280】

図16に示すように、デコード装置は、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レジデュアル情報を含む画像情報を受信することができる（S1600）。例えば、前記画像情報は、ビデオ情報と呼ばれることもできる。前記画像情報は、本文書の上述した実施形態（等）に係る様々な情報を含むことができる。例えば、画像情報は、予測関連情報またはレジデュアル関連情報のうち、少なくとも一部を含むことができる。

30

【0281】

例えば、前記予測関連情報は、インター予測モード情報また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、様々なインター予測モードのうち、少なくとも一部を表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（Affine）モード、サブブロックマージモード、またはMMVD（merge with MVD）モードなど、様々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）、またはBDOF（Bi-directional optical flow）などが付随的なモードとしてさらにまたはその代わりに使用されることができる。例えば、前記インター予測タイプ情報は、inter_pred_idcシンタックス要素を含むことができる。または、前記インター予測タイプ情報は、L0予測、L1予測、または双（bi）予測のうち、いずれか1つを表す情報を含むことができる。

40

【0282】

デコード装置は、レジデュアル情報に基づいてレジデュアルサンプルを生成できる（S1610）。デコード装置は、デコード装置は、前記レジデュアル情報に基づいて量子化された変換係数を導出でき、前記変換係数に対する逆変換手順に基づいてレジデュアルサ

50

ンプルを導出できる。

【0283】

デコード装置は、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できる（S1620）。例えば、デコード装置は、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、また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また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またはサブブロックマージ候補リストなどのように呼ばれることができるが、簡略に、マージ候補リストと呼ばれることもできる。

10

【0284】

例えば、前記マージ候補リスト内の候補の個数が最大候補個数になるまでマージ候補リストに候補が挿入され得る。ここで、候補は、現在ブロックの動き情報（または、動きベクトル）を導出するための候補または候補ブロックを表すことができる。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介し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先に探索されて（空間的マージ）候補が導出され得るし、その後、時間的周辺ブロックが探索されて（時間的マージ）候補と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にも、前記マージ候補リスト内の候補の個数が最大候補個数より少ない場合、追加候補を挿入できる。例えば、追加候補は、history based merge candidate(s)、pair-wise average merge candidate(s)、ATMVP、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うち、少なくとも1つを含むことができる。

20

【0285】

または、例えば、アフィンマージ候補リスト内の候補の個数が最大候補個数になるまでアフィンマージ候補リストに候補が挿入され得る。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。または、前記候補は、前記CPMVを導出するための候補または候補ブロックを表すこともできる。前記CPMVは、現在ブロックのCP（Control Point）における動きベクトルを表すことができる。例えば、CPは、2個、3個、または4個であることができ、現在ブロックの左上側（または、左上側コーナー）、右上側（または、右上側コーナー）、左下側（または、左下側コーナー）、または右下側（または、右下側コーナー）のうち、少なくとも一部に位置することができ、各位置毎に1つのCPのみが存在しうる。

30

【0286】

例えば、候補ブロックは、現在ブロックの周辺ブロック（または、現在ブロックのCPの周辺ブロック）に対する探索を介し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、継承された（inherited）アフィンマージ候補、構成された（constructed）アフィンマージ候補、またはゼロ（zero）動きベクトル候補のうち、少なくとも1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、先に前記継承されたアフィンマージ候補を挿入でき、その後、構成されたアフィンマージ候補を挿入できる。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個数が最大候補個数より小さい場合、残りは、ゼロ動きベクトル候補で満たすことができる。ここで、ゼロ動きベクトル候補は、ゼロベクトルと呼ばれることもできる。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ることができるが、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

40

50

リストであることもできる。この場合、アフィンマージ候補リストは、サブブロックマージ候補リストと呼ばれることもでき、サブブロックマージ候補リストは、S b T M V P で導出された候補（または、S b T M V P 候補）も含まれることができる。例えば、S b T M V P 候補がサブブロックマージ候補リストに含まれる場合、サブブロックマージ候補リスト内で継承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することができる。

【0287】

デコード装置は、マージ候補リストのうち、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できる（S 1 6 3 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、またはゼロベクトル候補のうち、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1つの候補を選択できる。または、例えば、サブブロックマージ候補リストは、継承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、またはゼロベクトル候補のうち、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1つの候補を選択できる。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表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また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できる。例えば、前記選択情報は、前記画像情報に含まれることができる。または、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できる。

【0288】

デコード装置は、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 1 予測サンプルを生成できる（S 1 6 4 0）。例えば、デコード装置は、インター予測タイプが双予測で導出される場合、選択された候補に基づいてL 0 動き情報及びL 1 動き情報を導出できる。デコード装置は、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）で導出することができる。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、または双（b i）予測のうち、双予測で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基盤の予測を表すことができ、L 1 予測は、参照ピクチャリスト1基盤の予測を表すことができ、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基盤の予測を表すことができる。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 シンタックス要素を含むことができる。

【0289】

例えば、L 0 動き情報は、L 0 参照ピクチャインデックス及びL 0 動きベクトルなどを含むことができ、L 1 動き情報は、L 1 参照ピクチャインデックス及びL 1 動きベクトルなどを含むことができる。L 0 参照ピクチャインデックスは、参照ピクチャリスト0で参照ピクチャを表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で参照ピクチャを表す情報を含むことができる。

【0290】

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 動き情報が導出され得るし、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され得る。

【0291】

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 動き情報が導出され得るし、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され得る。

【0292】

例えば、前記候補は、ペアワイズ (pairwise) 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 動き情報が導出され得るし、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され得る。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。

【0293】

または、例えば、前記マージ候補リストは、サブブロックマージ候補リストであることができ、アフィンマージ候補、サブブロックマージ候補、または SbTMVP 候補が選択されることもできる。ここで、サブブロック単位のアフィンマージ候補は、サブブロックマージ候補と呼ばれることもできる。

【0294】

例えば、前記候補は、アフィン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アフィンマージ候補である場合、前記アフィンマージ候補に基づいて L0 動き情報及び L1 動き情報が導出され得るし、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され得る。例えば、前記アフィンマージ候補は、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されることができる。

【0295】

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、1 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、または 4 個であることができ、現在ブロックの左上側 (または、左上側コーナー)、右上側 (または、右上側コーナー)、左下側 (または、左下側コーナー)、または右下側 (または、右下側コーナー) のうち、少なくとも一部に位置することができ、各位置毎に 1 つの CP のみが存在しうる。

【0296】

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 であることができる。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コーナー周辺ブロック、前記左上側コーナー周辺ブロックの下側に隣接した左側周辺ブロック、及び前記左上側コーナー周辺ブロックの右側に隣接した上側周辺ブロックを含むことができる。または、前記周辺ブロックは、図 12 において A₂ ブロック、B₂ ブロック、または B₃ ブロックを含むことができる。

【0297】

または、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 であることができる。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コーナー周辺ブロック及び前記右上側コーナー周辺ブロックの左側に隣接した上側周辺ブロックを含むことができる。または、前記周辺ブロックは、図 12 において B₀ ブロックまたは B₁ ブロックを含むことができる。

【0298】

または、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 であることができる。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コーナー周辺ブロック及び前記左下側コーナー周辺ブロックの上側に隣接した左側周辺ブロックを含むことができる。または、前記周辺ブロックは、図 12 において A₀ ブロックまたは A₁ ブロックを含むことができる。

【0299】

または、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 であることができる。ここで、CP3 は、RB と呼ばれることもできる。この場合、前記周辺ブロックは、前記現在ブロックのコル (col) ブロックまたは前記コルブロックの右下側コーナー周辺ブロックを含むことができる。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

10

20

30

40

50

を含むことができる。または、前記周辺ブロックは、図 12 において T ブロックを含むことができる。

【0300】

または、例えば、前記候補は、S b T M V P 候補を含むことができる。例えば、前記選択された候補が前記 S b T M V P 候補である場合、前記現在ブロックの左側周辺ブロックに基づいて L 0 動き情報及び L 1 動き情報が導出され得るし、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され得る。例えば、前記 L 0 予測サンプル及び前記 L 1 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。

【0301】

デコード装置は、L 0 予測サンプル、L 1 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できる (S 1260)。例えば、前記加重値情報は、マージ候補リストに含まれた候補のうち、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ことができる。例えば、前記加重値情報は、L 0 予測サンプルまたは L 1 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表すことができ、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ることができる。例えば、加重値インデックス情報は、3 個または 5 個の加重値のうち、いずれか 1 つの加重値を表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) または BWA (Bi-prediction with Weighted Average) における加重平均を表すことができる。

【0302】

例えば、前記候補は、時間的マージ候補を含むことができ、前記時間的マージ候補に対する加重値インデックス情報は、0 に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 に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 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表すことができる。

【0303】

例えば、前記候補は、時間的マージ候補を含むことができ、前記時間的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、時間的マージ候補に対する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る。

【0304】

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利用されたマージ候補リスト内の他の 2 個の候補のうち 1 つの加重値インデックス情報で導出されることができる。すなわち、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、前記ペアワイズ候補を導出するのに利用されたマージ候補リスト内の他の 2 個の候補のうち 1 つの加重値インデックス情報で導出されることができる。

【0305】

例えば、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、他の 2 個の候補に基づいて導出されることができる。前記他の 2 個の候補のそれぞれの加重値インデックス情報が第 1 の加重値インデックス情報と互いに同一である場合、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、前記第 1 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。前記他の 2 個の候補のそれぞれの加重値インデックス情報が互いに同一でない場合、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、デフォルト (DEFAULT) 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。このとき、前記デフォルト加重値インデックス情報は、前

10

20

30

40

50

記 L 0 予測サンプル及び前記 L 1 予測サンプルの各々に同じ加重値を付与する加重値インデックス情報に該当することができる。

【0306】

例えば、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、他の2個の候補に基づいて導出されることができる。前記他の2個の候補のそれぞれの加重値インデックス情報が第1の加重値インデックス情報と互いに同一である場合、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、前記第1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。前記他の2個の候補のそれぞれの加重値インデックス情報が互いに同一でない場合、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、前記他の2個の候補のそれぞれの加重値インデックス情報のうち、デフォルト (DEFAULT) 加重値インデックス情報でな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。前記デフォルト加重値インデックス情報は、前記 L 0 予測サンプル及び前記 L 1 予測サンプルの各々に同じ加重値を付与する加重値インデックス情報に該当することができる。

10

【0307】

例えば、前記マージ候補リストは、サブブロックマージ候補リストであることができ、アフィンマージ候補、サブブロックマージ候補、または S b T M V P 候補が選択されることもできる。ここで、サブブロック単位のアフィンマージ候補は、サブブロックマージ候補と呼ばれることもできる。

【0308】

例えば、前記候補は、アフィンマージ候補を含むことができ、前記アフィンマージ候補は、前記現在ブロックの左上側に位置する C P 0 (Control Point 0)、前記現在ブロックの右上側に位置する C P 1 (Control Point 1)、前記現在ブロックの左下側に位置する C P 2 (Control Point 2)、及び前記現在ブロックの右下側に位置する C P 3 (Control Point 3) の各々に対する動きベクトルのうち、少なくとも1つを含むことができる。

20

【0309】

例えば、前記アフィンマージ候補が {C P 0、C P 1、C P 2}、{C P 0、C P 1、C P 3}、{C P 0、C P 2、C P 3}、{C P 0、C P 1}、及び {C P 0、C P 2} のうち、いずれか1つに基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記 C P 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基づいて表されることができる。前記アフィンマージ候補が {C P 1、C P 2、C P 3} に基づいて生成される場合、前記アフィンマージ候補に対する加重値インデックス情報は、前記 C P 1 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基づいて表されることができる。

30

【0310】

前記 C P 0 の周辺ブロックのうち、特定ブロックは、前記 C P 0 に対する C P M V の導出のために使用されるブロックに該当し、前記 C P 0 の周辺ブロックは、前記現在ブロックの左上側コーナー周辺ブロック、前記左上側コーナー周辺ブロックの下側に隣接した左側周辺ブロック、及び前記左上側コーナー周辺ブロックの右側に隣接した上側周辺ブロックを含むことができる。

40

【0311】

前記 C P 1 の周辺ブロックのうち、特定ブロックは、前記 C P 1 に対する C P M V の導出のために使用されるブロックに該当し、前記 C P 1 の周辺ブロックは、前記現在ブロックの右上側コーナー周辺ブロック及び前記右上側コーナー周辺ブロックの左側に隣接した上側周辺ブロックを含むことができる。

【0312】

または、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 S b T M V P 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、S b T M V P 候補のた

50

めの加重値インデックス情報は、前記左側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。

【0313】

または、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 S b T M V P 候補に対する加重値インデックス情報は、0に導出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に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表すことができる。

【0314】

または、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、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、右下側サブブロックを含むことができる。

【0315】

または、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の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、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の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。

【0316】

デコード装置は、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できる（S 1 6 6 0）。例えば、デコード装置は、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でき、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され得る。

【0317】

例えば、デコード装置は、ビットストリームまたはエンコードされた情報をデコードして前述した情報（または、シンタックス要素）の全部または一部を含む画像情報を取得することができる。また、前記ビットストリームまたはエンコードされた情報は、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、前述したデコード方法が実行されるようにすることができる。

【0318】

前述した実施形態において、方法は、一連のステップまたはブロックで流れ図に基づいて説明されているが、該当実施形態は、ステップの順序に限定されるものではなく、あるステップは、前述と異なるステップと、異なる順序にまたは同時に発生できる。また、当業者であれば、流れ図に示すステップが排他的でなく、他のステップが含まれ、または、流れ図の1つまたはそれ以上のステップが本文書の実施形態の範囲に影響を及ぼさずに削除可能であることを理解することができる。

【0319】

前述した本文書の実施形態による方法は、ソフトウェア形態で実現されることができ、本文書によるエンコード装置及び／又はデコード装置は、例えば、T V、コンピュータ、スマートフォン、セットトップボックス、ディスプレイ装置などの画像処理を実行する装置に含まれることができる。

【0320】

本文書において、実施形態がソフトウェアで実現される時、前述した方法は、前述した機能を遂行するモジュール（過程、機能など）で実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

10

20

30

40

50

の内部または外部にあり、よく知られた多様な手段でプロセッサと連結されることができる。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리카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形態は、プロセッサ、マイクロプロセッサ、コントローラまたはチップ上で実現されて実行されることができる。例えば、各図面を示す機能ユニットは、コンピュータ、プロセッサ、マイクロプロセッサ、コントローラまたはチップ上で実現されて実行されることができる。この場合、実現のための情報 (例えば、information on instructions) また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。

10

【0321】

また、本文書の実施形態が適用されるデコード装置及びエンコード装置は、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、注文型ビデオ (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、画像電話ビデオ装置、運送手段端末 (例えば、車両 (自律走行車両を含む) 端末、飛行機端末、船舶端末等)、及び医療用ビデオ装置などに含まれることができ、ビデオ信号またはデータ信号を処理するために使われることができる。例えば、OTTビデオ (Over the top video) 装置として、ゲームコンソール、ブルーレイプレーヤ、インターネット接続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) などを含むことができる。

20

【0322】

また、本文書の実施形態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り可能な記録媒体に格納されることができる。また、本文書の実施形態によるデータ構造を有するマルチメディアデータもコンピュータが読み取り可能な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り可能な記録媒体は、コンピュータで読みだすことができ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り可能な記録媒体は、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むことができる。また、前記コンピュータが読み取り可能な記録媒体は、搬送波 (例えば、インターネットを介した送信) の形態で実現されたメディアを含む。また、エンコード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り可能な記録媒体に格納され、または、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。

30

【0323】

また、本文書の実施形態は、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実現されることができ、前記プログラムコードは、本文書の実施形態により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、コンピュータにより読み取り可能なキャリア上に格納されることができる。

40

【0324】

図18は、本文書において開示された実施形態が適用され得るコンテンツストリーミングシステムの例を示す。

【0325】

図18に示すように、本文書の実施形態が適用されるコンテンツストリーミングシステムは、大別してエンコードサーバ、ストリーミングサーバ、ウェブサーバ、メディア格納

50

【 0 3 2 6 】

【 0 3 2 7 】

10

【 0 3 2 8 】

20

【 0 3 2 9 】

【 0 3 3 0 】

30

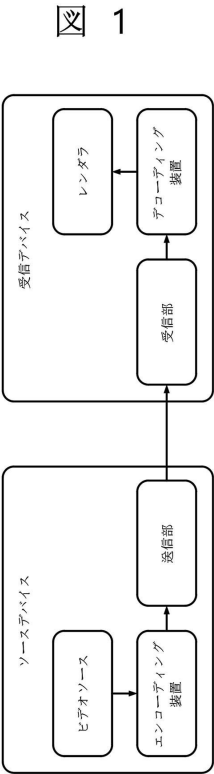
【 0 3 3 1 】

40

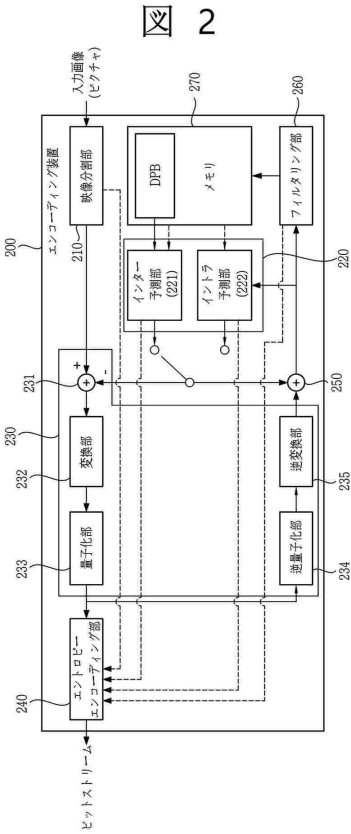
【 0 3 3 2 】

50

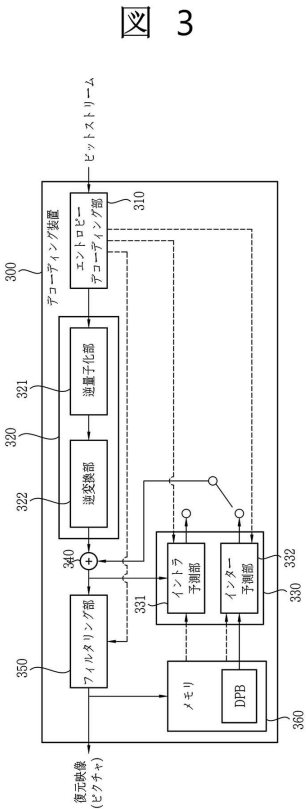
【図面】
【図 1】



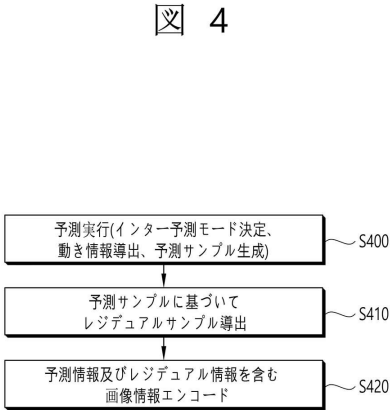
【図 2】



【図 3】



【図 4】



10

20

30

40

50

【図 5】

【図 6】

図 5

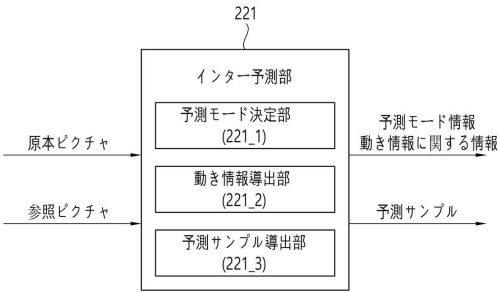
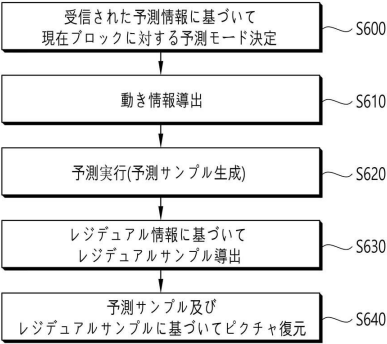


図 6



【図 7】

【図 8】

図 7

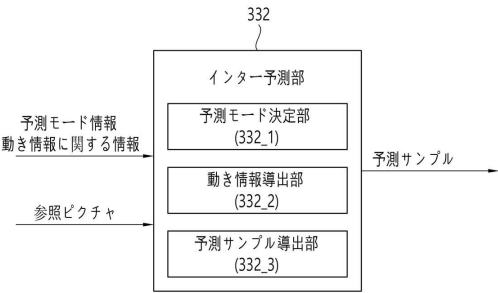
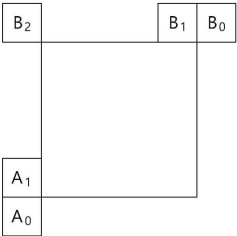


図 8



10

20

30

40

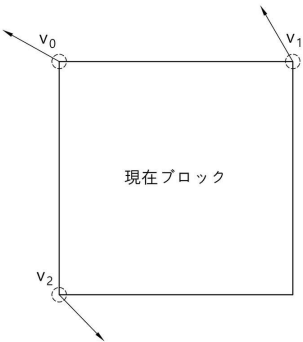
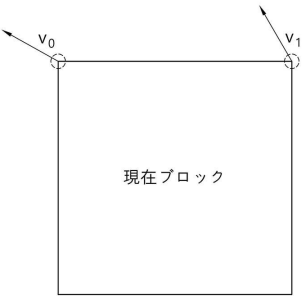
50

【図 9 a】

【図 9 b】

図 9a

図 9b



10

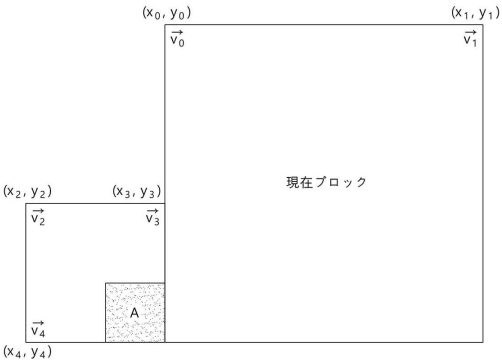
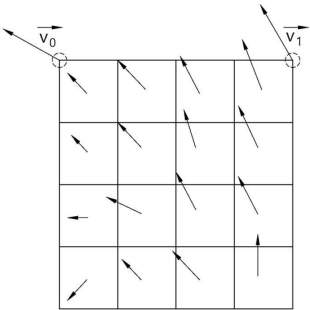
20

【図 1 0】

【図 1 1】

図 10

図 11



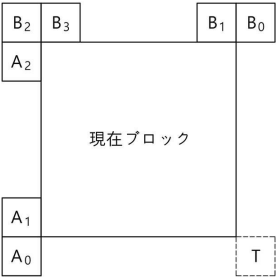
30

40

50

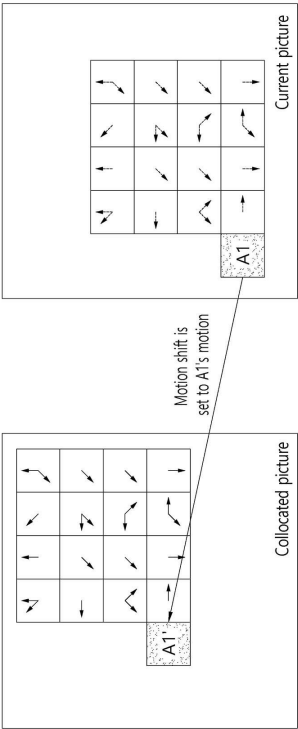
【図 1 2】

図 12



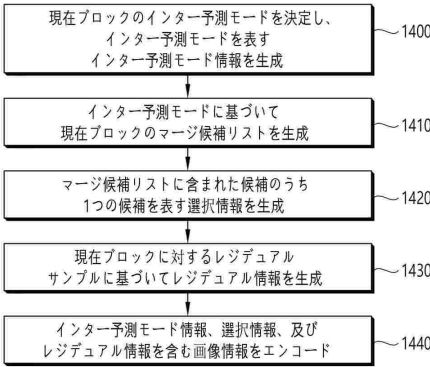
【図 1 3】

図 13



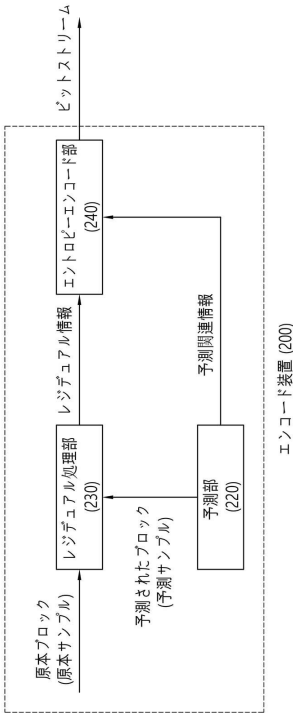
【図 1 4】

図 14



【図 1 5】

図 15



10

20

30

40

50

【図 16】

【図 17】

図 16

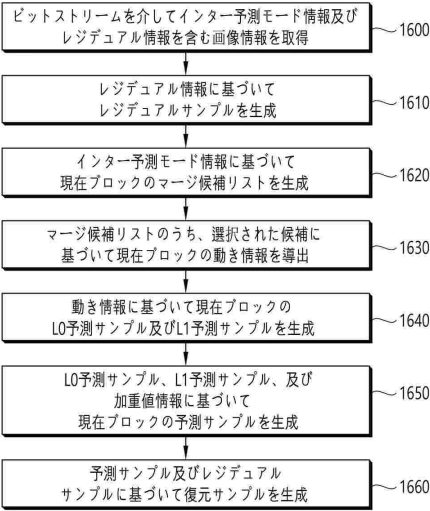
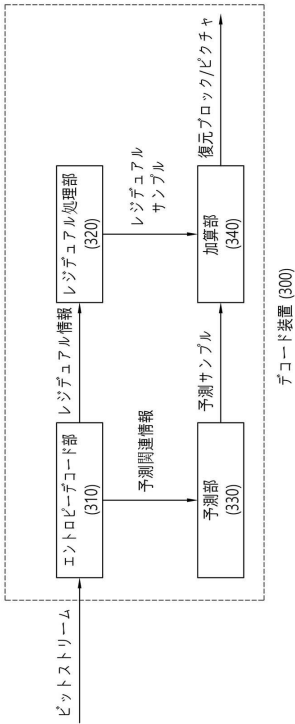
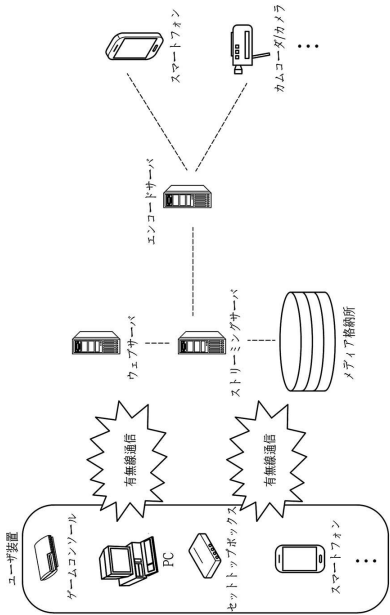


図 17



【図 18】

図 18



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 06772, ソチョーク, ヤンジェーデロ 11ーギル, 19, 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 06772, ソチョーク, ヤンジェーデロ 11ーギル, 19, 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 06772, ソチョーク, ヤンジェーデロ 11ーギル, 19, 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 国際公開第2020/244658 (WO, A1)
国際公開第2020/185555 (WO, A1)
Zhi-Yi Lin, et al., CE2-2.1: Simplification of constructed affine merging candidate derivation, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-N0075-v1, 14th Meeting: Geneva, CH, 2019年03月, pp.1-4
Jie Chen, Ruling Liao, and Yan Ye, Non-CE2: Gbi inheritance for constructed affine merge candidate, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-N481-v6, 14th Meeting: Geneva, CH, 2019年03月, pp.1-4
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., DB名)
H04N 19/00 - 19/98