

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7249441号
(P7249441)

(45)発行日 令和5年3月30日(2023.3.30)

(24)登録日 令和5年3月22日(2023.3.22)

(51)国際特許分類

F I

H 0 4 N	19/105 (2014.01)	H 0 4 N	19/105
H 0 4 N	19/577 (2014.01)	H 0 4 N	19/577
H 0 4 N	19/537 (2014.01)	H 0 4 N	19/537
H 0 4 N	19/52 (2014.01)	H 0 4 N	19/52

請求項の数 12 (全84頁)

(21)出願番号	特願2021-574322(P2021-574322)	(73)特許権者	502032105
(86)(22)出願日	令和2年6月10日(2020.6.10)		エルジー エレクトロニクス インコーポ レイティド
(65)公表番号	特表2022-536532(P2022-536532 A)		L G E L E C T R O N I C S I N C .
(43)公表日	令和4年8月17日(2022.8.17)		大韓民国, ソウル, ヨンドゥンポーク, ヨイーデロ, 1 2 8
(86)国際出願番号	PCT/KR2020/007521		1 2 8, Yeoui-daero, Y eongdeungpo-gu, 07
(87)国際公開番号	WO2020/251257		3 3 6 Seoul, Republic of Korea
(87)国際公開日	令和2年12月17日(2020.12.17)		
審査請求日	令和4年1月21日(2022.1.21)	(74)代理人	100099759
(31)優先権主張番号	62/861,986		弁理士 青木 篤
(32)優先日	令和1年6月14日(2019.6.14)	(74)代理人	100123582
(33)優先権主張国・地域又は機関	米国(US)		弁理士 三橋 真二
		(74)代理人	100165191

最終頁に続く

(54)【発明の名称】 予測サンプルを生成するための加重値インデックス情報を導出する映像デコーディング方法、及びその装置

(57)【特許請求の範囲】

【請求項1】

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

前記インター予測モード情報に基づいて、現在ブロックに対する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のうち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に対する加重値インデックスに基づいて、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、前記加重値インデックスは、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、を含み、

前記候補は、構成されたアフィンマージ候補を含み、

前記構成されたアフィンマージ候補は、C P M V (Control Point Motion Vector)を含み、

前記アフィンマージ候補が C P 0 (Control Point 0) に対する C P M V 0 を含む場合に
基づいて、前記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 に対す

10

20

る加重値インデックスと等しく設定され、

前記C P 0は、前記現在ブロックの左上側角に関連し、

前記構成されたアフィンマージ候補が前記C P 0に対する前記C P M V 0を含まない場合に基づいて、前記アフィンマージ候補に対する前記加重値インデックスは、C P 1 (Control Point 1) に対する加重値インデックスと等しく設定され、

前記C P 1は、前記現在ブロックの右上側角に関連する、映像デコーディング方法。

【請求項2】

前記C P 0に対する前記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、前記C P M V 0の導出のために使用されるブロックに対する加重値インデックスと等しく、

前記C P 0の前記周辺ブロックは、前記現在ブロックの左上側角の周辺ブロック、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック、及び前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含む、請求項1に記載の映像デコーディング方法。

【請求項3】

前記C P 1に対する前記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、C P M V 1の導出のために使用されるブロックに対する加重値インデックスと等しく、

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

【請求項4】

前記候補は、S b T M V P (subblock-based temporal motion vector prediction) 候補を含み、

前記S b T M V P 候補に対する加重値インデックスは、前記現在ブロックの左側の周辺ブロックに対する加重値インデックスに基づいて導出される、請求項1に記載の映像デコーディング方法。

【請求項5】

前記候補は、S b T M V P 候補を含み、

前記S b T M V P 候補に対する加重値インデックスは、0で導出される、請求項1に記載の映像デコーディング方法。

【請求項6】

前記候補は、S b T M V P 候補を含み、

前記S b T M V P 候補に対する加重値インデックスは、同一位置ブロック内のセンターブロックに対する加重値インデックスに基づいて導出され、

前記同一位置ブロックは、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の位置と同じ位置のブロックを含み、

前記センターブロックは、前記同一位置ブロックの中央に位置する4個のサブブロックのうち、右下側のサブブロックを含む、請求項1に記載の映像デコーディング方法。

【請求項7】

エンコーディング装置によって実行される映像エンコーディング方法において、

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

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

前記マージ候補リストに含まれた候補のうちの一つの候補を示す選択情報を生成するステップと、

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

前記候補は、構成されたアフィンマージ候補を含み、

前記構成されたアフィンマージ候補は、C P M V (Control Point Motion Vector) を含み、

10

20

30

40

50

前記構成されたアフィンマージ候補がC P 0 (Control Point 0) に対するC P M V 0を含む場合に基づいて、前記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に対する加重値インデックスと等しく設定され、
前記C P 0は、前記現在ブロックの左上側角に関連し、

前記構成されたアフィンマージ候補が前記C P 0に対する前記C P M V 0を含まない場合に基づいて、前記構成されたアフィンマージ候補に対する前記加重値インデックスは、C P 1 (Control Point 1) に対する加重値インデックスと等しく設定され、
前記C P 1は、前記現在ブロックの右上側角に関連する、映像エンコーディング方法。

【請求項 8】

前記構成されたアフィンマージ候補が前記C P 0に対する前記C P M V 0を含む前記場合に基づいて、前記構成されたアフィンマージ候補に対する前記加重値インデックスは、前記C P 1に対する前記加重値インデックス、及びC P 2 (control point 2) に対する加重値インデックスにかかわらず、前記C P 0に対する前記加重値インデックスと等しく設定され、

前記C P 2は、前記現在ブロックの左下側角に関連する、請求項 1 に記載の映像デコーディング方法。

【請求項 9】

前記構成されたアフィンマージ候補が前記C P 0に対する前記C P M V 0を含む前記場合に基づいて、前記構成されたアフィンマージ候補に対する前記加重値インデックスは、前記C P 0に対する前記加重値インデックスを、前記C P 1に対する前記加重値インデックス、又はC P 2に対する加重値インデックスと比較することなく、前記C P 0に対する前記加重値インデックスと等しく設定され、

前記C P 2は、前記現在ブロックの左下側角に関連する、請求項 1 に記載の映像デコーディング方法。

【請求項 10】

前記構成されたアフィンマージ候補が前記C P 0に対する前記C P M V 0を含む前記場合に基づいて、前記構成されたアフィンマージ候補に対する前記加重値インデックスは、前記C P 1に対する前記加重値インデックス、及びC P 2 (control point 2) に対する加重値インデックスにかかわらず、前記C P 0に対する前記加重値インデックスと等しく設定され、

前記C P 2は、前記現在ブロックの左下側角に関連する、請求項 7 に記載の映像エンコーディング方法。

【請求項 11】

前記構成されたアフィンマージ候補が前記C P 0に対する前記C P M V 0を含む前記場合に基づいて、前記構成されたアフィンマージ候補に対する前記加重値インデックスは、前記C P 0に対する前記加重値インデックスを、前記C P 1に対する前記加重値インデックス、又はC P 2に対する加重値インデックスと比較することなく、前記C P 0に対する前記加重値インデックスと等しく設定され、

前記C P 2は、前記現在ブロックの左下側角に関連する、請求項 7 に記載の映像エンコーディング方法。

【請求項 12】

映像のためのデータの送信方法であって、

前記映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、前記現在ブロックに対するマージ候補リストを生成するステップと、

前記マージ候補リストに含まれた候補のうちの一つの候補を示す選択情報を生成するステップと、

10

20

30

40

50

前記インター予測モード情報及び前記選択情報を含む映像情報をエンコーディングするステップと、
に基づいて生成される、ステップを含み、
前記候補は、構成されたアフィンマージ候補を含み、
前記構成されたアフィンマージ候補は、C P M V (Control Point Motion Vector) を含み、
前記構成されたアフィンマージ候補がC P 0 (Control Point 0) に対するC P M V 0 を含む場合に基づいて、前記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 に対する加重値インデックスと等しく設定され、
前記C P 0 は、前記現在ブロックの左上側角に関連し、
前記構成されたアフィンマージ候補が前記C P 0 に対する前記C P M V 0 を含まない場合に基づいて、前記構成されたアフィンマージ候補に対する前記加重値インデックスは、C P 1 (Control Point 1) に対する加重値インデックスと等しく設定され、
前記C P 1 は、前記現在ブロックの右上側角に関連する、送信方法。

10

【発明の詳細な説明】

【技術分野】

【0001】

本技術は、予測サンプルを生成するための加重値インデックス情報を導出する映像デコーディング方法、及びその装置に関する。

【背景技術】

20

【0002】

近年、4 Kまたは8 K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。

30

【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、映像コーディングの効率を高める方法、及び装置を提供することにある。

40

【0006】

本文書の別の技術的課題は、インター予測で予測サンプルを生成するための加重値インデックス情報を誘導する方法、及び装置を提供することにある。

【0007】

本文書のまた別の技術的課題は、双予測の際にアフィンマージ候補リスト内の候補のための加重値インデックス情報を誘導する方法、及び装置を提供することにある。

【課題を解決するための手段】

【0008】

本文書の一実施形態に係ると、デコーディング装置によって実行される映像デコーディ

50

ング方法が提供される。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を含む映像情報を受信するステップと、前記インター予測モード情報に基づいて、現在ブロックのマージ候補リストを生成するステップと、前記マージ候補リストのうち、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、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、CPMV (Control Point Motion Vector) を含み、前記アフィンマージ候補が前記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に対する第0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前記現在ブロックの左上側に位置するCP0に対するCPMV0を含まない場合、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、前記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加重値インデックス情報に基づいて導出される。

【0009】

本文書の別の実施形態に係ると、エンコーディング装置によって実行されるビデオエンコーディング方法を提供する。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示すインター予測モード情報を生成するステップと、前記インター予測モードに基づいて前記現在ブロックのマージ候補リストを生成するステップと、前記マージ候補リストに含まれた候補のうちの一つの候補を示す選択情報を生成するステップと、前記インター予測モード情報及び前記選択情報を含む映像情報をエンコーディングするステップと、を含み、前記候補は、アフィンマージ候補を含み、前記アフィンマージ候補は、CPMV (Control Point Motion Vector) を含み、前記アフィンマージ候補が前記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に対する第0加重値インデックス情報に基づいて示され、前記アフィンマージ候補が前記現在ブロックの左上側に位置するCP0に対するCPMV0を含まない場合、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、前記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加重値インデックス情報に基づいて示される。

【0010】

本文書のまた別の実施形態に係ると、デコーディング装置に映像デコーディング方法を行うように引き起こす映像情報を含むビットストリームが保存されたコンピュータ読み取り可能なデジタル保存媒体を提供する。前記映像デコーディング方法は、ビットストリームを介してインター予測モード情報を含む映像情報を取得するステップと、前記インター予測モード情報に基づいて現在ブロックのマージ候補リストを生成するステップと、前記マージ候補リストに含まれた候補のうちの一つの候補を選択するステップと、前記選択された候補に基づいて前記現在ブロックの動き情報を導出するステップと、前記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、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CPMV (Control Point Motion Vector) を含み、前記アフィンマージ候補が前記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に対する第0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前記現在ブロックの左上側に位置するCP0に対するCPMV0を含まない場合、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、前記現在ブロックの右上側に位置するC P 1 (C o n t r o l P o i n t 1) に対する第1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。

10

【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】インター予測の手順を例示的に示す。

【図5】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面である。

20

【図6】アフィン動きモデルを介して表現される動きを例示的に示す。

【図7a】アフィン動き予測のためのC P M Vを例示的に示す。

【図7b】アフィン動き予測のためのC P M Vを例示的に示す。

【図8】アフィンM V Fがサブブロックの単位で決定される場合を例示的に示す。

【図9】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面である。

【図10】アフィンマージモードにおいて候補の位置を説明するための図面である。

【図11】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面である。

【図12】本文書の実施形態に係るビデオ／映像エンコーディング方法及び関連のコンポーネントの一例を概略的に示す。

【図13】本文書の実施形態に係るビデオ／映像エンコーディング方法及び関連のコンポーネントの一例を概略的に示す。

30

【図14】本文書の実施形態に係る映像／ビデオデコーディング方法及び関連のコンポーネントの一例を概略的に示す。

【図15】本文書の実施形態に係る映像／ビデオデコーディング方法及び関連のコンポーネントの一例を概略的に示す。

【図16】本文書において開示された実施形態が適用できるコンテンツストリーミングシステムの例を示す。

【発明を実施するための形態】

【0015】

本開示は、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。ところ、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、本開示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、明細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。

40

【0016】

50

一方、本開示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、本文書の本質から逸脱しない限り、本開示の権利範囲に含まれる。

【0017】

本明細書において「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、本明細書において「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でき

10

【0018】

本明細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、

【0019】

本明細書において「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、本明細書において「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

20

【0020】

また、本明細書において「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

30

【0021】

また、本明細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)”で表示された場合、“予測”の一例として“イントラ予測”が提案されたものである。他の表現としては、本明細書の“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る。また、“予測 (即ち、イントラ予測)”で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る。

40

【0022】

本明細書において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。

【0023】

以下、添付した図面を参照して、本開示の望ましい実施形態をより詳細に説明しようとする。以下、図面上の同じ構成要素に対しては、同じ参照符号を使用し、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

【0024】

50

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。

【0025】

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。

【0026】

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。

【0027】

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。

【0028】

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。

【0029】

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。

【0030】

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。

【0031】

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。

【0032】

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 au

10

20

30

40

50

dio 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。

【0033】

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。

【0034】

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。

【0035】

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 height equal to the height of the picture and a width specified by syntax elements in the picture parameter set)。前記タイル行は、CTUの四角領域であり、前記四角領域は、ピクチャパラメータセット内のシンタックス要素によって明示される幅を有し、高さは、前記ピクチャの高さと同一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。

【0036】

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)。

【0037】

10

20

30

40

50

ピクセル (pixel) またはペル (pel) は、1つのピクチャ (または、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。

【0038】

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。

【0039】

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とは、画像エンコード装置を含むことができる。

【0040】

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。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。

【0041】

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 (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

10

20

30

40

50

づいて下位 (d e e p e r) 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。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 (r e c u
r s i v e l y) 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。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 (P U : P r e d i c t i o n
U n i t) または変換ユニット (T U : T r a n s f o r m U n i t) 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。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 (r e s i d u a l s i g n a l) を
導く単位であることができる。

【0042】

ユニットは、場合によって、ブロック (b l o c k) または領域 (a r e a) などの用
語と混用して使用されることができ。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 (t r a n s f o r m c o e f f i c i e n t)
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 (l u m a) 成分のピクセル／ピクセル値のみを示すことができ、彩
度 (c h r o m a) 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 (p i x e l) またはペル (p e l) に対応
する用語として使用することができる。

【0043】

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 (r e s i d u a l s i
g n a l、残余ブロック、残余サンプルアレイ) を生成でき、生成されたレジデュアル信
号は、変換部 232 に送信される。この場合、図示されたように、エンコード装置 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 (p r e d i c t e d b l o
c k) 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。

【0044】

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 (n e i g h b o r) 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 (P l a n a r 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。

【0045】

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。

【0046】

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。

【0047】

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。

10

【0048】

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)、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／又はシンタックス要素は、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る。ここで、ネットワークは、放送網及び／又は通信網などを含むことができ、デジタル格納媒体は、USB、SD、CD、DVD、ブルーレイ、HDD、SSD など、様々な格納媒体を含むことができる。エントロピーエンコード部 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。

20

30

40

【0049】

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

50

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る。加算部 250 は、復元部または復元ブロック生成部と呼ばれることができる。生成された復元信号は、現在ピクチャ内の次の処理対象ブロックのイントラ予測のために使用されることができ、後述するように、フィルタリングを経て次のピクチャのインター予測のために使用されることができる。

【0050】

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る。

【0051】

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る。

【0052】

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。

【0053】

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。

【0054】

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。

【0055】

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ルリングされた (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

10

20

30

40

50

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。

【0056】

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。

【0057】

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（memoery）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もできる。前記ハードウェアコンポーネントは、メモリ360を内／外部コンポーネントとしてさらに備えることもできる。

【0058】

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。

【0059】

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ビンを受信し、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／ビ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ビン（bin）の発生確率を予測し、ビンの算術デコード（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／ビンの文脈モデルのためにデコードされたシンボル／ビ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。

【0060】

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。

【0061】

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。

【0062】

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。

【0063】

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。

10

【0064】

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。

【0065】

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。

20

30

【0066】

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。

40

【0067】

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。

【0068】

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。

【0069】

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

50

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。

【0070】

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。

【0071】

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。

【0072】

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。

【0073】

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。

【0074】

図 4 は、インター予測の手順を例示的に示す。

【0075】

図4を参照すると、インター予測の手順は、インター予測モード決定ステップと、決定された予測モードによる動き情報導出ステップと、導出された動き情報に基づいた予測実行（予測サンプルの生成）ステップと、を含むことができる。前記インター予測の手順は、前述したように、エンコーディング装置及びデコーディング装置で実行されることができる。本文書において、コーディング装置とは、エンコーディング装置及び／又はデコーディング装置を含むことができる。

【0076】

図4を参照すると、コーディング装置は、現在ブロックに対するインター予測モードを決定する（S400）。ピクチャ内の現在ブロックの予測のために多様なインター予測モードが使用できる。例えば、マージモード、スキップモード、MVP (motion vector prediction) モード、アフィン (Affine) モード、サブブロックマージモード、MMVD (merge with MVD) モード等の多様なモードが使用できる。DMVR (Decoder side motion vector refinement) モード、AMVR (adaptive motion vector resolution) モード、Bi-prediction with CU-level weight (BCW)、Bi-directional optical flow (BDOF) 等が付随的なモードとしてさらに、又は代わりに使用できる。アフィンモードは、アフィン動き予測 (affine motion prediction) モードとも呼ばれ得る。MVPモードは、AMVP (advanced motion vector prediction) モードとも呼ばれ得る。本文書において、一部モード及び／又は一部モードにより導出された動き情報の候補は、他のモードの動き情報に関する候補のうちの一つに含まれることもある。例えば、HMVP候補は、前記マージ／スキップモードのマージ候補に追加されることがあり、或いは前記MVPモードのmvp候補に追加されることがある。前記HMVP候補が前記マージモード又はスキップモードの動き情報の候補に使用される場合、前記HMVP候補は、HMVPマージ候補とも呼ばれ得る。

【0077】

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

【0078】

コーディング装置は、前記現在ブロックに対する動き情報を導出する（S410）。前記動き情報の導出を前記インター予測モードに基づいて導出されることができる。

【0079】

コーディング装置は、現在ブロックの動き情報を用いてインター予測を実行することができる。エンコーディング装置は、動き推定 (motion estimation) 手順を介して、現在ブロックに対する最適の動き情報を導出することができる。例えば、エンコーディング装置は、現在ブロックに対する原本ピクチャ内の原本ブロックを用いて、相関性の高い類似する参照ブロックを参照ピクチャ内の決められた探索範囲内で分数ピクセル単位で探索することができ、これを介して動き情報を導出することができる。ブロッ

クの類似性は、位相 (p h a s e) ベースのサンプル値の差に基づいて導出することができる。例えば、ブロックの類似性は、現在ブロック (又は現在ブロックのテンプレート) と参照ブロック (又は参照ブロックのテンプレート) との間の S A D に基づいて計算されることができる。この場合、探索領域内の S A D が最も小さい参照ブロックに基づいて動き情報を導出することができる。導出された動き情報は、インター予測モードに基づいて、様々な方法によってデコーディング装置にシグナリングされることができる。

【 0 0 8 0 】

コーディング装置は、前記現在ブロックに対する動き情報に基づいてインター予測を実行する (S 4 2 0) 。コーディング装置は、前記動き情報に基づいて、前記現在ブロックに対する予測サンプルを導出することができる。前記予測サンプルを含む現在ブロックは、予測されたブロックと呼ばれ得る。

10

【 0 0 8 1 】

図 5 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面である。

【 0 0 8 2 】

マージモード (m e r g e m o d e) 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 (r e g u l a r m e r g e m o d e) と呼ばれることができる。例えば、前記マージモードは、 r e g u l a r _ m e r g e _ f l a g 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。

20

【 0 0 8 3 】

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 (m e r g e c a n d i d a t e b l o c k) をサーチ (s e a r c h) 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

【 0 0 8 4 】

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。

【 0 0 8 5 】

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 (s p a t i a l m e r g e c a n d i d a t e) と 1 個の時間的マージ候補 (t e m p o r a l m e r g e c a n d i d a t e) 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 4 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 M V P 候補は、S M V P 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 M V P 候補は、T M V P と呼ばれることができる。

40

【 0 0 8 6 】

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。

【 0 0 8 7 】

コーディング装置 (エンコーディング装置 / デコーディング装置) は、現在ブロックの空間的周辺ブロックを探索し、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角の周

50

辺ブロック、左側の周辺ブロック、右上側角の周辺ブロック、上側の周辺ブロック、左上側角の周辺ブロックを含むことができる。但し、これは例示であって、前述した空間的周辺ブロック以外にも、右側の周辺ブロック、下側の周辺ブロック、右下側の周辺ブロック等、追加的な周辺ブロックがさらに前記空間的周辺ブロックとして使用できる。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として導出できる。例えば、エンコーディング装置又はデコーディング装置は、図5に示された5個のブロックをA1→B1→B0→A0→B2のように順序通り探索して、利用可能な候補を順次にインデキシングし、マージ候補リストで構成することができる。

【0088】

コーディング装置は、前記現在ブロックの時間的周辺ブロックを探索し、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。

【0089】

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。

【0090】

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。

10

【0091】

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。

【0092】

前記選択されたマージ候補の動き情報が前記現在ブロックの動き情報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。

20

【0093】

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。

30

【0094】

一方、前記ペアワイズ平均マージ候補（*pair-wise average merge candidate*）は、ペアワイズ平均候補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。

40

【0095】

例えば、ペアワイズ平均マージ候補が追加された後にも、マージ候補リストが満たされ

50

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。

【0096】

一方、既存にはコーディングブロックの動きを表現するために、ただ一つの動きベクトルを使用することができた。すなわち、トランスレーション (translation) 動きモデルが使用できた。但し、このような方法がブロック単位で最適の動きを表現するかもしれないが、実際に各サンプルの最適の動きではなく、サンプル単位で最適の動きベクトルを決定することができれば、コーディング効率を高めることができる。このため、アフィン動きモデル (affine motion model) が使用できる。アフィン動きモデルを使用してコーディングするアフィン動き予測方法は、後述する内容のように4つの動きを効率的に表現できる。

【0097】

図6は、アフィン動きモデルを介して表現される動きを例示的に示す。

【0098】

図6を参照すると、前記アフィン動きモデルを介して表現され得る動きは、並進 (translate) 動き、スケール (scale) 動き、回転 (rotate) 動き及びせん断 (shear) 動きを含むことができる。すなわち、図6に示した時間の流れに応じて映像 (の一部) が平面移動する並進動きだけでなく、時間の流れに応じて映像 (の一部) がスケール (scale) されるスケール動き、時間の流れに応じて映像 (の一部) が回転する回転動き、時間の流れに応じて映像 (の一部) が平行四辺形変形されるせん断動きを前記アフィン動き予測を介して効率的に表現できる。

【0099】

エンコーディング装置／デコーディング装置は、前記アフィン動き予測を介して、現在ブロックのコントロールポイント (control point、CP) での動きベクトルに基づいて映像の歪曲形態を予測することができ、これを介して予測の正確度を高めることによって映像の圧縮性能を向上させることができる。また、前記現在ブロックの周辺ブロックの動きベクトルを用いて、前記現在ブロックの少なくとも一つのコントロールポイントに対する動きベクトルが誘導できるが、追加される付加情報に対するデータ量の負担を減らし、インター予測の効率をかなり向上させることができる。

【0100】

アフィン動きモデルが表現し得る動きのうちの3つの動き (トランスレーション (translation)、スケール (scale)、回転 (rotate)) を表現するアフィン動きモデルは類似 (similarity) (又は簡素化された (simplified)) アフィン動きモデルと呼ばれ得る。しかし、アフィン動きモデルが前述の動きモデルに限定されるわけではない。

【0101】

アフィン動きの予測方法は、2個、3個又は4個の動きベクトルを用いて、ブロックの各サンプル単位で動きベクトルを表現することができる。

【0102】

図7a及び図7bは、アフィン動き予測のためのCPMVを例示的に示す。

【0103】

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することができる。そのとき、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) と示すことができる。

【0104】

例えば、図7aは、2個のCPMVが用いられる場合を示すことができ、これは、4-

10

20

30

40

50

パラメータアフィンモデルと呼ばれ得る。この場合、 (x, y) のサンプル位置での動きベクトルは、例えば、数式 1 のように決定されることができる。

【0105】

【数1】

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

10

【0106】

例えば、図 7 b は、3 個の CPMV が用いられる場合を示すことができ、これは、6-パラメータアフィンモデルと呼ばれ得る。この場合、 (x, y) サンプル位置での動きベクトルは、例えば、数式 2 のように決定されることができる。

【0107】

【数2】

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

20

【0108】

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

30

【0109】

図 8 は、アフィン MVF がサブブロックの単位で決定される場合を例示的に示す。

【0110】

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

40

【0111】

図 8 の場合、アフィン MVF が 4×4 サブブロックの単位で決定されるが、サブブロックのサイズは多様に変形されることができる。

【0112】

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

50

ne 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。

【0113】

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。

【0114】

図9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面である。

【0115】

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。

【0116】

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。

【0117】

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。

【0118】

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。

【0119】

10

20

30

40

50

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。

【0120】

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、プ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。

10

【0121】

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示すことができる。例えば、図8を参照すると、左下側 (bottom-left) の周辺ブロックAがアフィン予測モードでコーディングされた場合、周辺ブロックAの左上側 (top-left) 角、右上側 (top-right) 角及び左下側 (bottom-left) 角の動きベクトル $v2$ 、 $v3$ 及び $v4$ が取得できる。周辺ブロックAが4-パラメータアフィン動きモデルでコーディングされた場合、現在ブロックの2個のCPMVPが $v2$ 及び $v3$ によって算出されることができる。周辺ブロックAが6-パラメータアフィン動きモデルでコーディングされた場合、現在ブロックの3個のCPMVP $v2$ 、 $v3$ 及び $v4$ によって算出されることができる。

20

【0122】

図10は、アフィンマージモードにおいて候補の位置を説明するための図面である。

【0123】

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。CPMVP k ($k=0, 1, 2, 3$) は、 k 番目のコントロールポイントを示すことができる。

30

【0124】

図10を参照すると、CPMVP0のために $B2 \rightarrow B3 \rightarrow A2$ の順序によってブロックがチェックでき、第一に利用可能なブロックの動きベクトルが使用できる。CPMVP1のために $B1 \rightarrow B0$ の順序によってブロックがチェックでき、CPMVP2のために $A1 \rightarrow A0$ の順序によってブロックがチェックできる。TMVP (temporal motion vector predictor) は、利用可能な場合、CPMVP3に使用されることができる。

【0125】

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、{CPMVP0、CPMVP1、CPMVP2}、{CPMVP0、CPMVP1、CPMVP3}、{CPMVP0、CPMVP2、CPMVP3}、{CPMVP1、CPMVP2、CPMVP3}、{CPMVP0、CPMVP1}、及び{CPMVP0、CPMVP2}のうち、いずれか1つに該当することができる。

40

【0126】

3個のCPMVP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CPMVP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

50

きる。

【0127】

図11は、インター予測においてSbTMVPを説明するための図面である。

【0128】

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

10

【0129】

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

【0130】

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。

20

【0131】

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。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、動きベクトルは、適用される動きシフトで選択されることができ。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。

【0132】

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col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。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。例えば、動きシフトが図4のA1の動きとして設定されることができ。この場合、各サブブロックに対してcolピクチャ内の対応ブロックの動き情報は、サブブロックの動き情報を導出するために使用されることができ。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。

30

【0133】

SbTMVP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。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。SbTMVPモードは、SPS (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。SbTMVPモードが利用可能である場合、SbTMVP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のエントリー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。

40

【0134】

SbTMVPで使用されるサブCU（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、SbTMVP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。追加的なSbTMVP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

50

、PまたはBスライス内の各CUに対して追加的なRD (rate-distortion) コストを利用するRDチェックがSbTMVP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。

【0135】

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MVに基づいて予測サンプルを生成できる。双(Bi)予測が適用される場合、L0予測(すなわち、参照ピクチャリストL0内の参照ピクチャとMVL0を用いた予測)に基づいて導出された予測サンプルとL1予測(すなわち、参照ピクチャリストL1内の参照ピクチャとMVL1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、L0方向の動きベクトルは、L0動きベクトルまたはMVL0と呼ばれることができ、L1方向の動きベクトルは、L1動きベクトルまたはMVL1と呼ばれることができる。双(Bi)予測が適用される場合、L0予測に用いられた参照ピクチャとL1予測に用いられた参照ピクチャとが現在ピクチャを基準に互いに異なる時間的方向に位置する場合(すなわち、双予測でありながら、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。

【0136】

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。

【0137】

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。

【0138】

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MVL0に基づいたL0予測サンプルとL1参照ピクチャ及びMVL1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る。

【0139】

【数3】

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

【0140】

数式3において、Pbi-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、 $P0$ 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、 $P1$ 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。

【0141】

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。

【0142】

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。

【0143】

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動きベクトル精度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(affine)と組み合わせられるとき(インター予測モードとしてアフィン予測モードが利用されるとき)、アフィン予測モードが現在ベストモード(best mode)と選択された場合、アフィンME(Motion Estimation)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(quantization parameter)、及び時間的レベル(temporal level)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。

【0144】

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(context coded bin)及び後続するバイパスコーディングされたビン(bypass coded bin)を用いてコーディングされることができる。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。

【0145】

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。

【0146】

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

10

20

30

40

50

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。

【 0 1 4 7 】

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1のとおりでありうる。

【 0 1 4 8 】

【表1－1】

8.4.2.2 Derivation process for luma motion vectors for merge mode

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

Inputs to this process are:

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

Outputs of this process are:

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

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

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

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

【 0 1 4 9 】

10

20

30

40

50

【表 1 - 2】

3. The derivation process for temporal luma motion vector prediction as specified in in clause 8.4.2.11 is invoked with the luma location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the variable $refIdxL0Col$ as inputs, and the output being the availability flag $availableFlagL0Col$ and the temporal motion vector $mvL0Col$. The variables $availableFlagCol$, $predFlagL0Col$ and $predFlagL1Col$ are derived as follows:

$$availableFlagCol = availableFlagL0Col \quad (8-283)$$

$$predFlagL0Col = availableFlagL0Col \quad (8-284)$$

$$predFlagL1Col = 0 \quad (8-285)$$

$$gbIdxCol = 0$$

(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 (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the variable $refIdxL1Col$ as inputs, and the output being the availability flag $availableFlagL1Col$ and the temporal motion vector $mvL1Col$. The variables $availableFlagCol$ and $predFlagL1Col$ are derived as follows:

$$availableFlagCol = availableFlagL0Col \ || \ availableFlagL1Col \quad (8-286)$$

10

【 0 1 5 0 】

【表 1 - 3】

$$predFlagL1Col = availableFlagL1Col \quad (8-287)$$

5. The merging candidate list, $mergeCandList$, is constructed as follows:

$$\begin{aligned} & i = 0 \\ & \text{if}(availableFlagA_1) \\ & \quad mergeCandList[i++] = A_1 \\ & \text{if}(availableFlagB_1) \\ & \quad mergeCandList[i++] = B_1 \\ & \text{if}(availableFlagB_0) \\ & \quad mergeCandList[i++] = B_0 \\ & \text{if}(availableFlagA_0) \\ & \quad mergeCandList[i++] = A_0 \\ & \text{if}(availableFlagB_2) \\ & \quad mergeCandList[i++] = B_2 \\ & \text{if}(availableFlagCol) \\ & \quad mergeCandList[i++] = Col \end{aligned} \quad (8-288)$$

30

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 $numCurrMergeCand$.

40

8. The derivation process for pairwise average merging candidates specified in clause 8.4.2.4 is invoked with $mergeCandList$, the reference indices $refIdxL0N$ and $refIdxL1N$, the prediction list utilization flags $predFlagL0N$ and $predFlagL1N$, the motion vectors $mvL0N$ and $mvL1N$ of every candidate N in $mergeCandList$, $numCurrMergeCand$ and $numOrigMergeCand$ as inputs, and the output is assigned to $mergeCandList$, $numCurrMergeCand$, the reference indices $refIdxL0avgCand_k$ and $refIdxL1avgCand_k$, the prediction list utilization flags $predFlagL0avgCand_k$ and $predFlagL1avgCand_k$ and the motion vectors $mvL0avgCand_k$ and $mvL1avgCand_k$ of every new candidate $avgCand_k$ being added into $mergeCandList$. The bi-prediction weight index $gbIdx$ of every new candidate $avgCand_k$ being added into $mergeCandList$ is set equal to 0. The number of candidates being added, $numAvgMergeCand$, is set equal to $(numCurrMergeCand - numOrigMergeCand)$. When $numAvgMergeCand$ is greater than 0, k ranges from 0 to $numAvgMergeCand - 1$, inclusive.

【 0 1 5 1 】

50

【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 refIdxL0zeroCand_m and refIdxL1zeroCand_m, the prediction list utilization flags predFlagL0zeroCand_m and predFlagL1zeroCand_m and the motion vectors mvL0zeroCand_m and mvL1zeroCand_m of every new candidate zeroCand_m being added into mergeCandList. The bi-prediction weight index gbIdx of every new candidate zeroCand_m being added into mergeCandList is set equal to 0. The number of candidates being added, numZeroMergeCand, is set equal to (numCurrMergeCand - numOrigMergeCand - numAvgMergeCand). When numZeroMergeCand is greater than 0, m ranges from 0 to numZeroMergeCand - 1, inclusive.
10. The variable mergeIdxOffset is set equal to 0.
11. When mmvd_flag[xCb][yCb] is equal to 1, the variable mmvdCnt is set equal to 0 and the following applies until mmvdCnt is greater than (merge_idx[xCb][yCb] + mergeIdxOffset) or mmvdCnt is equal to MaxNumMergeCand:
- When candidate mergeCandList[mmvdCnt] uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1.
 - The variable mmvdCnt is incremented by 1.

10

【 0 1 5 2 】

【表 1 - 5】

12. The following assignments are made with N being the candidate at position merge_idx[xCb][yCb] + mergeIdxOffset in the merging candidate list mergeCandList (N = mergeCandList[merge_idx[xCb][yCb] + mergeIdxOffset]) and X being replaced by 0 or 1:

$$\text{refIdxLX} = \text{refIdxLXN} \quad (8-289)$$

$$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-290)$$

$$\text{mvLX}[0][0][0] = \text{mvLXN}[0] \quad (8-291)$$

$$\text{mvLX}[0][0][1] = \text{mvLXN}[1] \quad (8-292)$$

$$\text{gbIdx} = \text{gbIdxN} \quad (8-293)$$

13. When mmvd_flag[xCb][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 mvL0[0][0], mvL1[0][0], the reference indices refIdxL0, refIdxL1 and the prediction list utilization flags predFlagL0[0][0] and 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:

$$\text{mvLX}[0][0][0] += \text{mMvdLX}[0] \quad (8-294)$$

$$\text{mvLX}[0][0][1] += \text{mMvdLX}[1] \quad (8-295)$$

20

30

【 0 1 5 3 】

前記表 1 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（8.4.2.2の目次3）で前記g b i I d x C o l 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。

40

【 0 1 5 4 】

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（c o l l o c a t e d）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（c o l）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

50

、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 2 のとおりでありうる。

【 0 1 5 5 】

【表 2 - 1】

8.4.2.2 Derivation process for luma motion vectors for merge mode

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

Inputs to this process are:

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

Outputs of this process are:

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

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

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

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

【 0 1 5 6 】

10

20

30

40

50

【表 2 - 2】

3.	The derivation process for temporal luma motion vector prediction as specified in in clause 8.4.2.11 is invoked with the luma location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the variable $refIdxL0Col$ as inputs, and the output being the availability flag $availableFlagL0Col$ and the temporal motion vector $mvL0Col$. The variables $availableFlagCol$, $predFlagL0Col$ and $predFlagL1Col$ are derived as follows:	
	$availableFlagCol = availableFlagL0Col$	(8-283)
	$predFlagL0Col = availableFlagL0Col$	(8-284)
	$predFlagL1Col = 0$	(8-285)
	$gbIdxCol = 0$	(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 (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the variable $refIdxL1Col$ as inputs, and the output being the availability flag $availableFlagL1Col$ and the temporal motion vector $mvL1Col$. The variables $availableFlagCol$ and $predFlagL1Col$ are derived as follows:	
	$availableFlagCol = availableFlagL0Col \ \ availableFlagL1Col$	(8-286)

10

【 0 1 5 7 】

【表 2 - 3】

	$predFlagL1Col = availableFlagL1Col$	(8-287)
	$gbIdxCol = gbIdxCol$	(x-xxx)
5.	The merging candidate list, $mergeCandList$, is constructed as follows:	
	$i = 0$ if($availableFlagA_1$) $mergeCandList[i++] = A_1$ if($availableFlagB_1$) $mergeCandList[i++] = B_1$ if($availableFlagB_0$) $mergeCandList[i++] = B_0$ if($availableFlagA_0$) $mergeCandList[i++] = A_0$ if($availableFlagB_2$) $mergeCandList[i++] = B_2$ if($availableFlagCol$) $mergeCandList[i++] = Col$	(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 $(MaxNumMergeCand - 1)$ 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 $numCurrMergeCand$.	

20

30

40

【 0 1 5 8 】

50

【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.
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.
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:

10

20

【 0 1 5 9 】

【表 2 - 5】

- When candidate mergeCandList[mmvdCnt] uses the current decoded picture as its reference picture, mergeIdxOffset is incremented by 1.
 - The variable mmvdCnt is incremented by 1.
12. The following assignments are made with N being the candidate at position merge_idx[xCb][yCb] + mergeIdxOffset in the merging candidate list mergeCandList (N = mergeCandList[merge_idx[xCb][yCb] + mergeIdxOffset]) and X being replaced by 0 or 1:
- $$\text{refIdxLX} = \text{refIdxLXN} \quad (8-289)$$
- $$\text{predFlagLX}[0][0] = \text{predFlagLXN} \quad (8-290)$$
- $$\text{mvLX}[0][0][0] = \text{mvLXN}[0] \quad (8-291)$$
- $$\text{mvLX}[0][0][1] = \text{mvLXN}[1] \quad (8-292)$$
- $$\text{gbiIdx} = \text{gbiIdxN} \quad (8-293)$$
13. When mmvd_flag[xCb][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 mvL0[0][0], mvL1[0][0], the reference indices refIdxL0, refIdxL1 and the prediction list utilization flags predFlagL0[0][0] and 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:
- $$\text{mvLX}[0][0][0] += \text{mMvdLX}[0] \quad (8-294)$$
- $$\text{mvLX}[0][0][1] += \text{mMvdLX}[1] \quad (8-295)$$

30

40

【 0 1 6 0 】

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

50

、`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。

【0161】

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。

【0162】

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。

【0163】

10

20

30

40

50

【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 $gbIdx$.

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

1. When $sps_sbtmvp_enabled_flag$ is equal to 1, the following applies:
 - The derivation process for merging candidates from neighbouring coding units as specified in clause 8.4.2.3 is invoked with the luma coding block location (x_{Cb} , y_{Cb}), the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ and the luma coding block width as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1.

【 0 1 6 4 】

【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 $gbIdxSbCol$, the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$, the reference indices $refIdxLXSbCol$, the luma motion vectors $mvLXSbCol[xSbIdx][ySbIdx]$ and the prediction list utilization flags $predFlagLXSbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$ and X being 0 or 1.
2. When $sps_affine_enabled_flag$ is equal to 1, the sample locations (x_{NbA_0} , y_{NbA_0}), (x_{NbA_1} , y_{NbA_1}), (x_{NbA_2} , y_{NbA_2}), (x_{NbB_0} , y_{NbB_0}), (x_{NbB_1} , y_{NbB_1}), (x_{NbB_2} , y_{NbB_2}), (x_{NbB_3} , y_{NbB_3}), and the variables $numSbX$ and $numSbY$ are derived as follows:

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

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

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

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

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

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

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

【 0 1 6 5 】

10

20

30

40

50

【表 3 - 3】

numSbX = cbWidth >> 2	(8-543)	
numSbY = cbHeight >> 2	(8-544)	
3. When <code>sps_affine_enabled_flag</code> is equal to 1, the variable <code>availableFlagA</code> is set equal to FALSE and the following applies for (<code>xNbA_k</code> , <code>yNbA_k</code>) from (<code>xNbA₀</code> , <code>yNbA₀</code>) to (<code>xNbA₁</code> , <code>yNbA₁</code>):		
– The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (<code>xCurr</code> , <code>yCurr</code>) set equal to (<code>xCb</code> , <code>yCb</code>) and the neighbouring luma location (<code>xNbA_k</code> , <code>yNbA_k</code>) as inputs, and the output is assigned to the block availability flag <code>availableA_k</code> .		
– When <code>availableA_k</code> is equal to TRUE and <code>MotionModelIdc[xNbA_k][yNbA_k]</code> is greater than 0 and <code>availableFlagA</code> is equal to FALSE, the following applies:		10
– The variable <code>availableFlagA</code> is set equal to TRUE, <code>motionModelIdcA</code> is set equal to <code>MotionModelIdc[xNbA_k][yNbA_k]</code> , (<code>xNb</code> , <code>yNb</code>) is set equal to (<code>CbPosX[xNbA_k][yNbA_k]</code> , <code>CbPosY[xNbA_k][yNbA_k]</code>), <code>nbW</code> is set equal to <code>CbWidth[xNbA_k][yNbA_k]</code> , <code>nbH</code> is set equal to <code>CbHeight[xNbA_k][yNbA_k]</code> , <code>numCpMv</code> is set equal to <code>MotionModelIdc[xNbA_k][yNbA_k] + 1</code> , and <code>gbiIdxA</code> is set equal to <code>GbiIdx[xNbA_k][yNbA_k]</code> .		
– For X being replaced by either 0 or 1, the following applies:		
– When <code>PredFlagLX[xNbA_k][yNbA_k]</code> is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (<code>xCb</code> , <code>yCb</code>), the luma coding block width and height (<code>cbWidth</code> , <code>cbHeight</code>), the neighbouring luma coding block location (<code>xNb</code> , <code>yNb</code>), the neighbouring luma coding block width and height (<code>nbW</code> , <code>nbH</code>), and the number of control point motion vectors <code>numCpMv</code> as input, the control point motion vector predictor candidates <code>cpMvLXA[cpIdx]</code> with <code>cpIdx = 0 .. numCpMv - 1</code> as output.		20
– 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 6 6 】

【表 3 - 4】

4. When <code>sps_affine_enabled_flag</code> is equal to 1, the variable <code>availableFlagB</code> is set equal to FALSE and the following applies for (<code>xNbB_k</code> , <code>yNbB_k</code>) from (<code>xNbB₀</code> , <code>yNbB₀</code>) to (<code>xNbB₂</code> , <code>yNbB₂</code>):		30
– The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (<code>xCurr</code> , <code>yCurr</code>) set equal to (<code>xCb</code> , <code>yCb</code>) and the neighbouring luma location (<code>xNbB_k</code> , <code>yNbB_k</code>) as inputs, and the output is assigned to the block availability flag <code>availableB_k</code> .		
– When <code>availableB_k</code> is equal to TRUE and <code>MotionModelIdc[xNbB_k][yNbB_k]</code> is greater than 0 and <code>availableFlagB</code> is equal to FALSE, the following applies:		
– The variable <code>availableFlagB</code> is set equal to TRUE, <code>motionModelIdcB</code> is set equal to <code>MotionModelIdc[xNbB_k][yNbB_k]</code> , (<code>xNb</code> , <code>yNb</code>) is set equal to (<code>CbPosX[xNbAB][yNbB_k]</code> , <code>CbPosY[xNbB_k][yNbB_k]</code>), <code>nbW</code> is set equal to <code>CbWidth[xNbB_k][yNbB_k]</code> , <code>nbH</code> is set equal to <code>CbHeight[xNbB_k][yNbB_k]</code> , <code>numCpMv</code> is set equal to <code>MotionModelIdc[xNbB_k][yNbB_k] + 1</code> , and <code>gbiIdxB</code> is set equal to <code>GbiIdx[xNbB_k][yNbB_k]</code> .		
– For X being replaced by either 0 or 1, the following applies:		40
– When <code>PredFlagLX[xNbB_k][yNbB_k]</code> 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 (<code>xCb</code> , <code>yCb</code>), the luma coding block width and height (<code>cbWidth</code> , <code>cbHeight</code>), the neighbouring luma coding block location (<code>xNb</code> , <code>yNb</code>), the neighbouring luma coding block width and height (<code>nbW</code> , <code>nbH</code>), and the number of control point motion vectors <code>numCpMv</code> as input, the control point motion vector predictor candidates <code>cpMvLXB[cpIdx]</code> with <code>cpIdx = 0 .. numCpMv - 1</code> as output.		
– The following assignments are made:		

【 0 1 6 7 】

【表 3 - 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 = 1..6</code> , <code>cpIdx = 0..2</code> as outputs and <code>gbIdxConstK</code> is set equal to 0 with <code>K = 1..6</code> .		
6.	The initial subblock merging candidate list, <code>subblockMergeCandList</code> , is constructed as follows:		
	<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)	10
			20

【 0 1 6 8 】

【表 3 - 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:		
-	The reference indices, the prediction list utilization flags and the motion vectors of <code>zeroCand_m</code> with <code>m</code> equal to (<code>numCurrMergeCand - numOrigMergeCand</code>) are derived as follows:		
	$\text{refIdxL0ZeroCand}_m = 0$	(8-550)	
	$\text{predFlagL0ZeroCand}_m = 1$	(8-551)	
	$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero}$	(8-552)	
	$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero}$	(8-553)	
	$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero}$	(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)	40
	$\text{cpMvL1ZeroCand}_m[0] = \text{mvZero}$	(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)	

【 0 1 6 9 】

【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:	
$\text{subblockMergeCandList}[\text{numCurrMergeCand}++] = \text{zeroCand}_m \quad (8-562)$	
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 7 0】

【表 3 - 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 7 1】

【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.	
...	

10

20

【0172】

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。

30

【0173】

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。

40

【0174】

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 $gbIdx$.

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 1 7 5 】

【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 $gbIdxSbCol$, the number of luma coding subblocks in horizontal direction $numSbX$ and in vertical direction $numSbY$, the reference indices $refIdxLXSbCol$, the luma motion vectors $mvLXSbCol[xSbIdx][ySbIdx]$ and the prediction list utilization flags $predFlagLXSbCol[xSbIdx][ySbIdx]$ with $xSbIdx = 0..numSbX - 1$, $ySbIdx = 0..numSbY - 1$ and X being 0 or 1.
- 2. When $sps_affine_enabled_flag$ is equal to 1, the sample locations (x_{NbA_0} , y_{NbA_0}), (x_{NbA_1} , y_{NbA_1}), (x_{NbA_2} , y_{NbA_2}), (x_{NbB_0} , y_{NbB_0}), (x_{NbB_1} , y_{NbB_1}), (x_{NbB_2} , y_{NbB_2}), (x_{NbB_3} , y_{NbB_3}), and the variables $numSbX$ and $numSbY$ are derived as follows:

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

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

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

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

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

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

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

【 0 1 7 6 】

10

20

30

40

50

【表 5 - 3】

numSbX = cbWidth >> 2	(8-543)	
numSbY = cbHeight >> 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 (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbA _k , yNbA _k) as inputs, and the output is assigned to the block availability flag availableA _k .		
– When availableA _k is equal to TRUE and MotionModelIdc[xNbA _k][yNbA _k] is greater than 0 and availableFlagA is equal to FALSE, the following applies:		10
– The variable availableFlagA is set equal to TRUE, motionModelIdcA is set equal to MotionModelIdc[xNbA _k][yNbA _k], (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 gbiIdx is set equal to GbiIdx[xNbA _k][yNbA _k].		
– 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 (xCb, yCb), the luma coding block width and height (cbWidth, cbHeight), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates cpMvLXA[cpIdx] with cpIdx = 0 .. numCpMv - 1 as output.		20
– The following assignments are made:		
predFlagLXA = PredFlagLX[xNbA _k][yNbA _k]	(8-545)	
refIdxLXA = RefIdxLX[xNbA _k][yNbA _k]	(8-546)	

【 0 1 7 7 】

【表 5 - 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 ₂ , yNbB ₂):		30
– The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbB _k , yNbB _k) as inputs, and the output is assigned to the block availability flag availableB _k .		
– When availableB _k is equal to TRUE and 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 gbiIdx is set equal to GbiIdx[xNbB _k][yNbB _k].		
– For X being replaced by either 0 or 1, the following applies:		40
– 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 (xCb, yCb), the luma coding block width and height (cbWidth, cbHeight), the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors numCpMv as input, the control point motion vector predictor candidates cpMvLXB[cpIdx] with cpIdx = 0 .. numCpMv - 1 as output.		
– The following assignments are made:		

【 0 1 7 8 】

【表 5 - 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:	
	<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> .	

【 0 1 7 9】

【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)

【 0 1 8 0】

10

20

30

40

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 8 1】

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

【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 8 3 】

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

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

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

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

$$y_{Ctr} = y_{Cb} + (cbHeight / 2) \quad (8-580)$$
 2. The luma location ($x_{ColCtrCb}$, $y_{ColCtrCb}$) is set equal to the top-left sample of the collocated luma coding block covering the location given by (x_{Ctr} , y_{Ctr}) inside $ColPic$ relative to the top-left luma sample of the collocated picture specified by $ColPic$.

【 0 1 8 4 】

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 (xColCtrCb, yColCtrCb), 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 8 5 】

30

40

50

【表 6 - 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:	
$xSb = xCb + xSbIdx * sbWidth \quad (8-586)$	
$ySb = yCb + ySbIdx * sbHeight \quad (8-587)$	
- The location ($xColSb, yColSb$) of the collocated subblock inside ColPic is derived as follows.	
$xColSb = Clip3(xCtb, \\ Min(CurPicWidthInSamplesY - 1, xCtb + (1 \ll CtbLog2SizeY) + 3), \\ xSb + (tempMv[0] \gg 4)) \quad (8-588)$	
$yColSb = Clip3(yCtb, \\ Min(CurPicHeightInSamplesY - 1, yCtb + (1 \ll CtbLog2SizeY) - 1), \\ ySb + (tempMv[1] \gg 4)) \quad (8-589)$	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 ($(xColSb \gg 3) \ll 3, (yColSb \gg 3) \ll 3$) inside the ColPic.	
- The luma location ($xColCb, yColCb$) is set equal to the top-left sample of the collocated luma coding block specified by colCb relative to the top-left luma sample of the collocated picture specified by ColPic.	
- The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, ($xColCb, yColCb$), refIdxL0 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock $mvL0SbCol[xSbIdx][ySbIdx]$ and availableFlagL0SbCol.	20
- 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:	
$mvLXSbCol[xSbIdx][ySbIdx] = ctrMvLX \quad (8-590)$	
$predFlagLXSbCol[xSbIdx][ySbIdx] = ctrPredFlagLX \quad (8-591)$	

【 0 1 8 6 】

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.

10

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 $gbIdxSbCol$.

20

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,

30

【 0 1 8 7 】

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. 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.	10
	20

【0 1 8 8】

【表 7 - 3】

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. - 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.	30
	40

【0 1 8 9】

【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:	
<code>ctrPredFlagL0 = 0</code>	(8-596)
<code>ctrPredFlagL1 = 0</code>	(8-597)

10

【0190】

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

【0191】

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（`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

【0192】

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 $gbIdx$.	10
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.	20

【 0 1 9 3 】

【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.		30
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:		40
(x_{A_0} , y_{A_0}) = ($x_{Cb} - 1$, $y_{Cb} + cbHeight$)	(8-536)	40
(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 9 4 】

【表 8 - 3】

	numSbX = cbWidth >> 2	(8-543)	
	numSbY = cbHeight >> 2	(8-544)	
3.	When <code>sps_affine_enabled_flag</code> is equal to 1, the variable <code>availableFlagA</code> is set equal to FALSE and the following applies for $(xNbA_k, yNbA_k)$ from $(xNbA_0, yNbA_0)$ to $(xNbA_1, yNbA_1)$:		
	- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location $(xCurr, yCurr)$ set equal to (xCb, yCb) and the neighbouring luma location $(xNbA_k, yNbA_k)$ as inputs, and the output is assigned to the block availability flag <code>availableA_k</code>. - When <code>availableA_k</code> is equal to TRUE and <code>MotionModelIdc[xNbA_k][yNbA_k]</code> is greater than 0 and <code>availableFlagA</code> is equal to FALSE, the following applies: - The variable <code>availableFlagA</code> is set equal to TRUE, <code>motionModelIdcA</code> is set equal to <code>MotionModelIdc[xNbA_k][yNbA_k]</code>, (xNb, yNb) is set equal to $(CbPosX[xNbA_k][yNbA_k], CbPosY[xNbA_k][yNbA_k])$, <code>nbW</code> is set equal to <code>CbWidth[xNbA_k][yNbA_k]</code>, <code>nbH</code> is set equal to <code>CbHeight[xNbA_k][yNbA_k]</code>, <code>numCpMv</code> is set equal to <code>MotionModelIdc[xNbA_k][yNbA_k] + 1</code>, and <code>gbIdxA</code> is set equal to <code>GbIdx[xNbA_k][yNbA_k]</code>. - For X being replaced by either 0 or 1, the following applies: - When <code>PredFlagLX[xNbA_k][yNbA_k]</code> is equal to 1, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height $(cbWidth, cbHeight)$, the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors <code>numCpMv</code> as input, the control point motion vector predictor candidates <code>cpMvLXA[cpIdx]</code> with <code>cpIdx = 0 .. numCpMv - 1</code> as output. - The following assignments are made: $\text{predFlagLXA} = \text{PredFlagLX}[xNbA_k][yNbA_k] \quad (8-545)$ $\text{refIdxLXA} = \text{RefIdxLX}[xNbA_k][yNbA_k] \quad (8-546)$	10	20

【 0 1 9 5】

【表 8 - 4】

4.	When <code>sps_affine_enabled_flag</code> is equal to 1, the variable <code>availableFlagB</code> is set equal to FALSE and the following applies for $(xNbB_k, yNbB_k)$ from $(xNbB_0, yNbB_0)$ to $(xNbB_2, yNbB_2)$:		
	- The availability derivation process for a block as specified in clause 6.4.X [Ed. (BB): Neighbouring blocks availability checking process tbd] is invoked with the current luma location $(xCurr, yCurr)$ set equal to (xCb, yCb) and the neighbouring luma location $(xNbB_k, yNbB_k)$ as inputs, and the output is assigned to the block availability flag <code>availableB_k</code>. - When <code>availableB_k</code> is equal to TRUE and <code>MotionModelIdc[xNbB_k][yNbB_k]</code> is greater than 0 and <code>availableFlagB</code> is equal to FALSE, the following applies: - The variable <code>availableFlagB</code> is set equal to TRUE, <code>motionModelIdcB</code> is set equal to <code>MotionModelIdc[xNbB_k][yNbB_k]</code>, (xNb, yNb) is set equal to $(CbPosX[xNbB_k][yNbB_k], CbPosY[xNbB_k][yNbB_k])$, <code>nbW</code> is set equal to <code>CbWidth[xNbB_k][yNbB_k]</code>, <code>nbH</code> is set equal to <code>CbHeight[xNbB_k][yNbB_k]</code>, <code>numCpMv</code> is set equal to <code>MotionModelIdc[xNbB_k][yNbB_k] + 1</code>, and <code>gbIdxB</code> is set equal to <code>GbIdx[xNbB_k][yNbB_k]</code>. - For X being replaced by either 0 or 1, the following applies: - When <code>PredFlagLX[xNbB_k][yNbB_k]</code> is equal to TRUE, the derivation process for luma affine control point motion vectors from a neighbouring block as specified in clause 8.4.4.5 is invoked with the luma coding block location (xCb, yCb), the luma coding block width and height $(cbWidth, cbHeight)$, the neighbouring luma coding block location (xNb, yNb), the neighbouring luma coding block width and height (nbW, nbH), and the number of control point motion vectors <code>numCpMv</code> as input, the control point motion vector predictor candidates <code>cpMvLXB[cpIdx]</code> with <code>cpIdx = 0 .. numCpMv - 1</code> as output. - The following assignments are made:	30	40

【 0 1 9 6】

【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 = 1..6</code> , <code>cpIdx = 0..2</code> as outputs and <code>gbIdxConstK</code> is set equal to 0 with <code>K = 1..6</code> .	
6.	The initial subblock merging candidate list, <code>subblockMergeCandList</code> , is constructed as follows:	10
	<pre> i = 0 if(availableFlagSbCol) subblockMergeCandList[i++] = SbCol if(availableFlagA && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = A if(availableFlagB && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = B if(availableFlagConst1 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const1 if(availableFlagConst2 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const2 if(availableFlagConst3 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const3 if(availableFlagConst4 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const4 if(availableFlagConst5 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const5 if(availableFlagConst6 && i < MaxNumSubblockMergeCand) subblockMergeCandList[i++] = Const6 </pre>	(8-549)
		20

【 0 1 9 7 】

【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> .	30
8.	When <code>numCurrMergeCand</code> is less than <code>MaxNumSubblockMergeCand</code> , the following is repeated until <code>numCurrMergeCand</code> is equal to <code>MaxNumSubblockMergeCand</code> , with <code>mvZero[0]</code> and <code>mvZero[1]</code> both being equal to 0:	
—	The reference indices, the prediction list utilization flags and the motion vectors of <code>zeroCand_m</code> with <code>m</code> equal to (<code>numCurrMergeCand - numOrigMergeCand</code>) are derived as follows:	
	$\text{refIdxL0ZeroCand}_m = 0$	(8-550)
	$\text{predFlagL0ZeroCand}_m = 1$	(8-551)
	$\text{cpMvL0ZeroCand}_m[0] = \text{mvZero}$	(8-552)
	$\text{cpMvL0ZeroCand}_m[1] = \text{mvZero}$	(8-553)
	$\text{cpMvL0ZeroCand}_m[2] = \text{mvZero}$	(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 1 9 8 】

【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:

【0 1 9 9】

【表 8 - 8】

-	The following assignments are made with N being the candidate at position merge_subblock_idx[xCb][yCb] in the subblock merging candidate list subblockMergeCandList (N = subblockMergeCandList[merge_subblock_idx[xCb][yCb]]):	
	refIdxLX = refIdxLXN	(8-568)
	predFlagLX[0][0] = predFlagLXN	(8-569)
	cpMvLX[0] = cpMvLXN[0]	(8-570)
	cpMvLX[1] = cpMvLXN[1]	(8-571)
	cpMvLX[2] = cpMvLXN[2]	(8-572)
	numCpMv = motionModelIdxN + 1	(8-573)
	gbIdx = gbIdxN	(8-574)
-	For xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1, the following applies:	
	predFlagLX[xSbIdx][ySbIdx] = predFlagLX[0][0]	(8-575)
-	When predFlagLX[0][0] is equal to 1, the derivation process for motion vector arrays from affine control point motion vectors as specified in subclause 8.4.4.9 is invoked with the luma coding block location (xCb, yCb), the luma coding block width cbWidth, the luma prediction block height cbHeight, the number of control point motion vectors numCpMv, the control point motion vectors cpMvLX[cpIdx] with cpIdx being 0..2, and the number of luma coding subblocks in horizontal direction numSbX and in vertical direction numSbY as inputs, the luma subblock motion vector array mvLX[xSbIdx][ySbIdx] and the chroma subblock motion vector array mvCLX[xSbIdx][ySbIdx] with xSbIdx = 0..numSbX - 1, ySbIdx = 0..numSbY - 1 as outputs.	
-	The following assignment is made for x = xCb..xCb + cbWidth - 1 and y = yCb..yCb + cbHeight - 1:	
	MotionModelIdx[x][y] = numCpMv - 1	(8-576)

【0 2 0 0】

【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 0 1】

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

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

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

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

$$y_{Ctr} = y_{Cb} + (cbHeight / 2) \quad (8-580)$$
 2. The luma location ($x_{ColCtrCb}$, $y_{ColCtrCb}$) is set equal to the top-left sample of the collocated luma coding block covering the location given by (x_{Ctr} , y_{Ctr}) inside $ColPic$ relative to the top-left luma sample of the collocated picture specified by $ColPic$.

【 0 2 0 2】

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.	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 2 0 3 】

【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. $\begin{aligned} \text{xColSb} = & \text{Clip3}(\text{xCtb}, \\ & \text{Min}(\text{CurPicWidthInSamplesY} - 1, \text{xCtb} + (1 \ll \text{CtbLog2SizeY}) + 3), \\ & \text{xSb} + (\text{tempMv}[0] \gg 4)) \end{aligned} \quad (8-588)$ $\begin{aligned} \text{yColSb} = & \text{Clip3}(\text{yCtb}, \\ & \text{Min}(\text{CurPicHeightInSamplesY} - 1, \text{yCtb} + (1 \ll \text{CtbLog2SizeY}) - 1), \\ & \text{ySb} + (\text{tempMv}[1] \gg 4)) \end{aligned} \quad (8-589)$	
– 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 0 4 】

【表 9 – 5】

– The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, $(\text{xColCb}, \text{yColCb})$, refIdxL0 set equal to 0 and sbFlag set equal to 1 as inputs and the output being assigned to the motion vector of the subblock $\text{mvL0SbCol}[\text{xSbIdx}][\text{ySbIdx}]$ and $\text{availableFlagL0SbCol}$.	
– The derivation process for collocated motion vectors as specified in clause 8.4.2.12 is invoked with currCb, colCb, $(\text{xColCb}, \text{yColCb})$, 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 $\text{mvL1SbCol}[\text{xSbIdx}][\text{ySbIdx}]$ and $\text{availableFlagL1SbCol}$.	
– When $\text{availableFlagL0SbCol}$ and $\text{availableFlagL1SbCol}$ are both equal to 0, the following applies for X being 0 and 1:	
$\text{mvLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrMvLX} \quad (8-590)$	10
$\text{predFlagLXSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrPredFlagLX} \quad (8-591)$	
$\text{gbIdxSbCol}[\text{xSbIdx}][\text{ySbIdx}] = \text{ctrgbIdx} \quad (\text{x-xxx})$	

【 0 2 0 5】

【表 1 0 – 1】

8.4.4.4 Derivation process for subblock-based temporal merging base motion data	
Inputs to this process are:	
– the location $(\text{xCtb}, \text{yCtb})$ of the top-left sample of the luma coding tree block that contains the current coding block, – the location $(\text{xColCtb}, \text{yColCtb})$ 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.	20
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 ctrgbIdx.	30
The variable tempMv is set as follows:	
$\text{tempMv}[0] = 0 \quad (8-592)$	
$\text{tempMv}[1] = 0 \quad (8-593)$	
The variable currPic specifies the current picture.	
The variable availableFlagN is set equal to FALSE, and the following applies:	
– When availableFlagA_1 is equal to 1, the following applies: – availableFlagN is set equal to TRUE,	40

【 0 2 0 6】

【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.

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.

20

【0 2 0 7】

【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.

30

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

```

xColCb = Clip3( xCtb,
                Min( CurPicWidthInSamplesY - 1, xCtb + ( 1 << CtbLog2SizeY ) + 3 ), (8-59
4)
                xColCtCb + ( tempMv[0] >> 4 ) )

yColCb = Clip3( yCtb,
                Min( CurPicHeightInSamplesY - 1, yCtb + ( 1 << CtbLog2SizeY ) - 1 ), (8-5
95)
                yColCtCb + ( tempMv[1] >> 4 ) )

```

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

40

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.

【0 2 0 8】

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:	
ctrPredFlagL0 = 0	(8-596)
ctrPredFlagL1 = 0	(8-597)

10

【0209】

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、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、サブブロックが利用可能でない場合、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（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

【0210】

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

【0211】

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。

【0212】

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。

【0213】

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

【0214】

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

【0215】

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

【0216】

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。

【0217】

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

【0218】

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。

【0219】

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} のうち少なくとも一つを含み得る。CPMV0、CPMV1、CPMV2及びCPMV3は、CP0、CP1、CP2及びCP3それぞれに対する動きベクトルに該当し得る。

【0220】

一実施形態において、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に対する第0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。前記第0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、前記CPMV0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該当し得る。そのとき、前記CP0の周辺ブロックは、現在ブロックの左上側角の周辺ブロック、左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック及び左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含むことができる。

【0221】

一方、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に対するCPMV0を含まない場合、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、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。前記第1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、前記CPMV1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該当し得る。そのとき、前記CP1の周辺ブロックは、現在ブロックの右上側角の周辺ブロック及び右上側角の周辺ブロックの左側に隣接した上側の周辺ブロックを含むことができる。

【0222】

前述の方法によると、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} それぞれに対して、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。

【0223】

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別の実施形態に係ると、現在ブロックの左上側に位置するCP0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CP1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CP0に対する第0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。前記第0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、前記CPMV0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該当し得る。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はない場合、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 予測サンプルのそれぞれに同一の加重値を付与する加重値インデックス情報に該当し得る。

【 0 2 2 4 】

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。

【 0 2 2 5 】

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。

【 0 2 2 6 】

図 1 2 及び 1 3 は、本文書の実施形態に係るビデオ／映像エンコーディング方法及び関連のコンポーネントの一例を概略的に示す。

【 0 2 2 7 】

図 1 2 において開示された方法は、図 2 又は図 1 3 において開示されたエンコーディング装置によって実行されることができる。具体的に、例えば、図 1 2 の S 1 2 0 0 乃至 S 1 2 2 0 は、図 1 3 の前記エンコーディング装置 2 0 0 の予測部 2 2 0 によって実行されることができる。図 1 2 の S 1 2 3 0 は、図 1 3 の前記エンコーディング装置 2 0 0 のエントロピーエンコーディング部 2 4 0 によって実行されることができる。また、図 1 2 で示していないが、図 1 2 において前記エンコーディング装置 2 0 0 の予測部 2 2 0 により予測サンプル又は予測関連情報を導出することができ、前記エンコーディング装置 2 0 0 のレジデュアル処理部 2 3 0 により原本サンプル又は予測サンプルからレジデュアル情報が導出でき、前記エンコーディング装置 2 0 0 のエントロピーエンコーディング部 2 4 0 によりレジデュアル情報又は予測関連情報からビットストリームが生成できる。図 1 2 で開示された方法は、本文書で前述した実施形態を含むことができる。

【 0 2 2 8 】

図 1 2 を参照すると、エンコーディング装置は、現在ブロックのインター予測モードを決定し、インター予測モードを示すインター予測モード情報を生成することができる (S 1 2 0 0) 。例えば、エンコーディング装置は、現在ブロックに適用するインター予測モードとして、マージモード、アフィン (マージ) モード又はサブブロックマージモードを決定することができ、これを示すインター予測モード情報を生成することができる。

【 0 2 2 9 】

エンコーディング装置は、インター予測モードに基づいて、現在ブロックのマージ候補リストを生成することができる (S 1 2 1 0) 。例えば、エンコーディング装置は、決定したインター予測モードによってマージ候補リストを生成することができる。ここで、決定したインター予測モードがアフィンマージモード又はサブブロックマージモードである場合、前記マージ候補リストは、アフィンマージ候補リスト又はサブブロックマージ候補

10

20

30

40

50

リスト等のように呼ばれ得るが、簡略にマージ候補リストとも呼ばれ得る。

【0230】

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。

10

【0231】

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。

20

【0232】

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、`SbTMVP`で導出された候補（または、`SbTMVP`候補）も含まれることができる。例えば、`SbTMVP`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。

30

40

【0233】

エンコーディング装置は、マージ候補リストに含まれた候補のうちの一つの候補を示す選択情報を生成することができる（S1220）。例えば、マージ候補リストは、空間的

50

マージ候補、時間的マージ候補、ペアワイズ候補又は零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一つの候補を選択することができる。或いは、例えば、サブブロックのマージ候補リストは、承継されたアフィンマージ候補、構成されたアフィンマージ候補、S b T M V P 候補又は零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一つの候補を選択することができる。

【0234】

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

【0235】

また、エンコーディング装置は、現在ブロックのインター予測タイプを双予測 (b i p r e d i c t i o n) で示すインター予測タイプ情報を生成することができる。例えば、前記現在ブロックのインター予測タイプは、L 0 予測、L 1 予測又は双 (b i) 予測のうち、双予測と決定されることができ、これを示すインター予測タイプ情報を生成することができる。ここで、L 0 予測は、参照ピクチャリスト0ベースの予測を示すことができ、L 1 予測は、参照ピクチャリスト1ベースの予測を示すことができ、双予測は、参照ピクチャリスト0及び参照ピクチャリスト1ベースの予測を示すことができる。例えば、エンコーディング装置は、前記インター予測タイプに基づいて、インター予測タイプ情報を生成することができる。例えば、前記インター予測タイプ情報は、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。

【0236】

エンコーディング装置は、インター予測モード情報及び選択情報を含む映像情報をエンコーディングすることができる (S 1 2 3 0)。例えば、前記映像情報は、ビデオ情報とも呼ばれ得る。前記映像情報は、本文書の前述した実施形態に係る多様な情報を含むことができる。例えば、映像情報は、予測関連情報又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、エンコーディング装置は、前述した情報 (又はシンタックス要素) の全部又は一部を含む映像情報をエンコーディングし、ビットストリーム又はエンコーディングされた情報を生成することができる。或いは、ビットストリームの形態で出力することができる。また、前記ビットストリーム又はエンコーディングされた情報は、ネットワーク又は保存媒体を介してデコーディング装置に送信されることができる。

【0237】

図12に示していないが、例えば、エンコーディング装置は、現在ブロックの予測サンプルを生成することができる。或いは、例えば、エンコーディング装置は選択された候補に基づいて現在ブロックの予測サンプルを生成することができる。又は、例えば、エンコーディング装置は、選択された候補に基づいて動き情報を導出することができ、動き情報に基づいて現在ブロックの予測サンプルを生成することができる。例えば、エンコーディング装置は、双予測によってL 0 予測サンプル及びL 1 予測サンプルを生成することができ、前記L 0 予測サンプル及び前記L 1 予測サンプルに基づいて、現在ブロックの予測サンプルを生成することができる。この場合、双予測のための加重値インデックス情報 (又は加重値情報) を用いて、前記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することができる。ここで、加重値情報は、前記加重値インデックス情報に基づいて示されることができる。

【0238】

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

10

20

30

40

50

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。

【0239】

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。

【0240】

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。

【0241】

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。

【0242】

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。

【0243】

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。

【0244】

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。

【0245】

例えば、前記 CP が前記現在ブロックの左上側に位置する CP 0 であり得る。この場合、前記周辺ブロックは、前記現在ブロックの左上側角の周辺ブロック、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック及び前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図 10 において、A₂ ブロック、B₂ ブロック又は B₃ ブロックを含むことができる。

【0246】

又は、例えば、前記 CP が前記現在ブロックの右上側に位置する CP 1 であり得る。この場合、前記周辺ブロックは、前記現在ブロックの右上側角の周辺ブロック及び前記右上側角の周辺ブロックの左側に隣接した上側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図 10 において、B₀ ブロック又は B₁ ブロックを含むことができ

10

20

30

40

50

る。

【0247】

又は、例えば、前記CPが前記現在ブロックの左下側に位置するCP2であり得る。この場合、前記周辺ブロックは、前記現在ブロックの左下側角の周辺ブロック及び前記左下側角の周辺ブロックの上側に隣接した左側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図10において、A₀ブロック又はA₁ブロックを含むことができる。

【0248】

又は、例えば、前記CPが前記現在ブロックの右下側に位置するCP3であり得る。ここで、CP3はRBとも呼ばれ得る。この場合、前記周辺ブロックは、前記現在ブロックの
10 コル(col)ブロック又は前記コルブロックの右下側角の周辺ブロックを含むことができる。ここで、前記コルブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができる。或いは、前記周辺ブロックは、図10においてTブロックを含むことができる。

【0249】

または、例えば、前記候補は、SbTMVP候補を含むことができる。例えば、前記選択された候補が前記SbTMVP候補である場合、前記現在ブロックの左側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
20 できる。

【0250】

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。

【0251】

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
30 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。

【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、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
40 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
50 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
50 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
50 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。

【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、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。

【0254】

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。

【0255】

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。

【0256】

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。

【0257】

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、CPMV (Control Point Motion Vector) を含むことができる。

【0258】

例えば、前記アフィンマージ候補が前記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に対する第0加重値インデックス情報に基づいて示されることができる。前記アフィンマージ候補が前記現在ブロックの左上側に位置するCP0に対するCPMV0を含まない場合、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、前記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加重値インデックス情報に基づいて示されることができる。

【0259】

前記第0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、前記CPMV

10

20

30

40

50

0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角の周辺ブロック、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック及び前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含むことができる。

【0260】

前記第1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、前記C P M V 1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角の周辺ブロック及び前記右上側角の周辺ブロックの左側に隣接した上側の周辺ブロックを含むことができる。

【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、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表されることができる。すなわち、S b T M V P 候補のための加重値インデックス情報は、前記左側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。

【0262】

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に表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。

【0263】

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表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。

【0264】

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表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表されることができる。

【0265】

或いは、図12で示していないが、例えば、エンコーディング装置は、前記予測サンプルと原本サンプルに基づいてレジデュアルサンプルを導出することができる。この場合、前記レジデュアルサンプルに基づいてレジデュアル関連情報が導出できる。前記レジデュアル関連情報に基づいてレジデュアルサンプルが導出できる。前記レジデュアルサンプル及び前記予測サンプルに基づいて、復元サンプルが生成できる。前記復元サンプルに基づいて、復元ブロック及び復元ピクチャが導出できる。又は、例えば、エンコーディング装置は、レジデュアル関連情報又は予測関連情報を含む映像情報をエンコーディングすることができる。

【0266】

例えば、エンコーディング装置は、前述した情報（又はシンタックス要素）の全部又は一部を含む映像情報をエンコーディングし、ビットストリーム又はエンコーディングされた情報を生成することができる。或いは、ビットストリームの形態で出力することができる。また、前記ビットストリーム又はエンコーディングされた情報は、ネットワーク又は

10

20

30

40

50

保存媒体を介してデコーディング装置に送信されることができる。或いは、前記ビットストリーム又はエンコーディングされた情報は、コンピュータ読み取り可能な保存媒体に保存されることができ、前記ビットストリーム又は前記エンコーディングされた情報は、前述した映像エンコーディング方法により生成されることができる。

【0267】

図14及び図15は、本文書の実施形態に係るビデオ／映像デコーディング方法及び関連のコンポーネントの一例を概略的に示す。

【0268】

図14において開示された方法は、図3又は図15において開示されたデコーディング装置によって実行されることができる。具体的に、例えば、図14のS1400は、図15において前記デコーディング装置300のエントロピーデコーディング部310によって実行されることができ、図14のS1410乃至S1440は、図15において前記デコーディング装置300の予測部330によって実行されることができる。また、図14で示していないが、図15において前記デコーディング装置300のエントロピーデコーディング部310によりビットストリームから予測関連情報又はレジデュアル情報が導出でき、前記デコーディング装置300のレジデュアル処理部320によりレジデュアル情報からレジデュアルサンプルが導出でき、前記デコーディング装置300の予測部330により予測関連情報から予測サンプルが導出でき、前記デコーディング装置300の加算部340によりレジデュアルサンプル又は予測サンプルから復元ブロック又は復元ピクチャが導出できる。図14において開示された方法は、本文書で前述した実施形態を含むことができる。

【0269】

図14を参照すると、デコーディング装置はビットストリームを介してインター予測モード情報を含む映像情報を受信することができる(S1400)。例えば、前記映像情報は、ビデオ情報とも呼ばれ得る。前記映像情報は、本文書の前述した実施形態に係る多様な情報を含むことができる。例えば、映像情報は、予測関連情報又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。

【0270】

例えば、前記予測関連情報は、インター予測モード情報又はインター予測タイプ情報を含むことができる。例えば、前記インター予測モード情報は、多様なインター予測モードのうち少なくとも一部を示す情報を含むことができる。例えば、マージモード、スキップモード、MVP (motion vector prediction) モード、アフィン (Affine) モード、サブブロックマージモード又はMMVD (merge with MVD) モード等の多様なモードが使用できる。また、DMVR (Decoder side motion vector refinement) モード、AMVR (adaptive motion vector resolution) モード、BCW (Bi-prediction with CU-level weight) 又はBDOPF (Bi-directional optical flow) 等が付随的なモードとしてさらに、又は代わりに使用できる。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。或いは、前記インター予測タイプ情報は、L0予測、L1予測又は双 (bi) 予測のいずれかを示す情報を含むことができる。

【0271】

デコーディング装置は、インター予測モード情報に基づいて、現在ブロックのマージ候補リストを生成することができる(S1410)。例えば、デコーディング装置は、前記インター予測モード情報に基づいて、現在ブロックのインター予測モードをマージモード、アフィン (マージ) モード又はサブブロックマージモードと決定することができ、決定したインター予測モードによってマージ候補リストを生成することができる。ここで、インター予測モードがアフィンマージモード又はサブブロックマージモードと決定された場合、前記マージ候補リストは、アフィンマージ候補リスト又はサブブロックマージ候補リスト等のように呼ばれ得るが、簡略にマージ候補リストとも呼ばれ得る。

【0272】

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。

10

【0273】

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。

20

【0274】

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、`SbTMVP`で導出された候補（または、`SbTMVP`候補）も含まれることができる。例えば、`SbTMVP`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。

30

40

【0275】

デコーディング装置は、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することができる（S1420）。例えば、マージ候補リストは、空間的マージ候補、時間的マージ候補、ペアワイズ候補又は零ベクトル候補のうち少なく

50

とも一部を含むことができ、現在ブロックのインター予測のために、このような候補のうちの一つの候補を選択することができる。或いは、例えば、サブブロックマージ候補リストは、承継されたアフィンマージ候補、構成されたアフィンマージ候補、S b T M V P 候補又は零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の一つの候補を選択することができる。例えば、前記選択された候補は、選択情報に基づいて前記マージ候補リストから選択されることができる。例えば、前記選択情報は、前記マージ候補リスト内の選択された候補を示すインデックス情報を含むことができる。例えば、前記選択情報は、マージインデックス情報又はサブブロックマージインデックス情報とも呼ばれ得る。例えば、前記選択情報は、前記映像情報に含まれることがある。或いは、前記選択情報は、前記インター予測モード情報に含まれることもある。

10

【0276】

デコーディング装置は、動き情報に基づいて、現在ブロックのL0予測サンプル及びL1予測サンプルを生成することができる(S1430)。例えば、デコーディング装置は、インター予測タイプが双予測で導出される場合、選択された候補に基づいてL0動き情報及びL1動き情報を導出することができる。デコーディング装置は、インター予測タイプ情報に基づいて、現在ブロックのインター予測タイプを双予測(bi-prediction)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、L0予測、L1予測又は双(bi)予測のうち、双予測で導出されることができる。ここで、L0予測は、参照ピクチャリスト0ベースの予測を示すことができ、L1予測は、参照ピクチャリスト1ベースの予測を示すことができ、双予測は、参照ピクチャリスト0及び参照ピクチャリスト1ベースの予測を示すことができる。例えば、前記インター予測タイプ情報は、inter_pred_idcシタックス要素を含むことができる。

20

【0277】

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動きベクトル等を含むことができ、L1動き情報は、L1参照ピクチャインデックス及びL1動きベクトル等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で参照ピクチャを示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で参照ピクチャを示す情報を含むことができる。

30

【0278】

例えば、前記候補は、空間的マージ候補を含むことができる。例えば、前記選択された候補が前記空間的マージ候補である場合、前記空間的マージ候補に基づいてL0動き情報及びL1動き情報が導出され得るし、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。

【0279】

例えば、前記候補は、時間的マージ候補を含むことができる。例えば、前記選択された候補が前記時間的マージ候補である場合、前記時間的マージ候補に基づいてL0動き情報及びL1動き情報が導出され得るし、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。

40

【0280】

例えば、前記候補は、ペアワイズ(pair-wise)候補を含むことができる。例えば、前記選択された候補がペアワイズ候補である場合、前記ペアワイズ候補に基づいてL0動き情報及びL1動き情報が導出され得るし、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。

【0281】

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

50

マージ候補と呼ばれることもできる。

【0282】

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。

【0283】

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。

【0284】

例えば、前記CPが前記現在ブロックの左上側に位置するCP0であり得る。この場合、前記周辺ブロックは、前記現在ブロックの左上側角の周辺ブロック、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック及び前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図10において、A₂ブロック、B₂ブロック又はB₃ブロックを含むことができる。

【0285】

又は、例えば、前記CPが前記現在ブロックの右上側に位置するCP1であり得る。この場合、前記周辺ブロックは、前記現在ブロックの右上側角の周辺ブロック及び前記右上側角の周辺ブロックの左側に隣接した上側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図10において、B₀ブロック又はB₁ブロックを含むことができる。

【0286】

又は、例えば、前記CPが前記現在ブロックの左下側に位置するCP2であり得る。この場合、前記周辺ブロックは、前記現在ブロックの左下側角の周辺ブロック及び前記左下側角の周辺ブロックの上側に隣接した左側の周辺ブロックを含むことができる。或いは、前記周辺ブロックは、図10において、A₀ブロック又はA₁ブロックを含むことができる。

【0287】

又は、例えば、前記CPが前記現在ブロックの右下側に位置するCP3であり得る。ここで、CP3はRBとも呼ばれ得る。この場合、前記周辺ブロックは、前記現在ブロックのコール (col) ブロック又は前記コールブロックの右下側角の周辺ブロックを含むことができる。ここで、前記コールブロックは、前記現在ブロックが位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができる。或いは、前記周辺ブロックは、図10においてTブロックを含むことができる。

【0288】

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

【0289】

デコーディング装置は、L0予測サンプル、L1予測サンプル及び加重値情報に基づいて、現在ブロックの予測サンプルを生成することができる (S1440)。例えば、前記

10

20

30

40

50

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 予測サンプル又は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示すことができ、前記加重平均は、加重値インデックス情報に基づいて実行されることができる。例えば、加重値インデックス情報は、3 個又は5 個の加重値のいずれかの加重値を示す情報を含むことができる。例えば、前記加重平均は、BCW (Bi-prediction with CU-level Weight) 又はBWA (Bi-prediction with Weighted Average) での加重平均を示すことができる。

10

【0290】

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。

【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、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。

20

【0292】

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。

30

【0293】

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。

40

【0294】

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

50

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。

【0295】

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。

【0296】

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、CPMV (Control Point Motion Vector) を含むことができる。

【0297】

例えば、前記アフィンマージ候補が前記現在ブロックの左上側に位置する CP 0 (Control Point 0) に対する CPMV 0 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 CP 0 に対する第 0 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。前記アフィンマージ候補が前記現在ブロックの左上側に位置する CP 0 に対する CPMV 0 を含まない場合、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、前記現在ブロックの右上側に位置する CP 1 (Control Point 1) に対する第 1 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。

【0298】

前記第 0 加重値インデックス情報は、前記 CP 0 の周辺ブロックのうち、前記 CPMV 0 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該当し、前記 CP 0 の周辺ブロックは、前記現在ブロックの左上側角の周辺ブロック、前記左上側角の周辺ブロックの下側に隣接した左側の周辺ブロック、及び前記左上側角の周辺ブロックの右側に隣接した上側の周辺ブロックを含むことができる。

【0299】

前記第 1 加重値インデックス情報は、前記 CP 1 の周辺ブロックのうち、前記 CPMV 1 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該当し、前記 CP 1 の周辺ブロックは、前記現在ブロックの右上側角の周辺ブロック及び前記右上側角の周辺ブロックの左側に隣接した上側の周辺ブロックを含むことができる。

【0300】

或いは、例えば、前記候補は S b T M V P 候補を含むことができ、前記 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の周辺ブロックの加重値インデックス情報に基づいて導出されることができる。すなわち、S b T M V P 候補のための加重値インデックス情報は、前記左側の周辺ブロックの加重値インデックス情報に基づいて導出されることができる。

【0301】

又は、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 で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 で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示すことができる。

【0302】

或いは、例えば、前記候補は、S b T M V P 候補を含むことができ、前記加重値インデックス情報は、コル (col) ブロック内のセンタースタックの加重値インデックス情報に基づいて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 (col) ブロック内のセンタースタックの加重値インデックス情報に基づいて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

10

20

30

40

50

位置する現在ピクチャと異なる参照ピクチャ内に前記現在ブロックと同じ位置のブロックを含むことができ、前記センターブロックは、前記コルブロックの中央に位置する4個のサブブロックのうち、右下側のサブブロックを含むことができる。

【0303】

又は、例えば、前記候補はSbTMVP候補を含むことができ、前記加重値インデックス情報は、コル(col)ブロックのサブブロックそれぞれの加重値インデックス情報に基づいて導出されることができる。すなわち、SbTMVP候補のための加重値インデックス情報は、コル(col)ブロックのサブブロックそれぞれの加重値インデックス情報に基づいて導出されることができる。

【0304】

図14に示していないが、例えば、デコーディング装置は、前記映像情報に含まれたレジデュアル関連情報に基づいてレジデュアルサンプルを導出することができる。また、デコーディング装置は、前記予測サンプル及び前記レジデュアルサンプルに基づいて復元サンプルを生成することができる。前記復元サンプルに基づいて、復元ブロック及び復元ピクチャが導出できる。

【0305】

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映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。

【0306】

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。

【0307】

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、TV、コンピュータ、スマートフォン、セットトップボックス、ディスプレイ装置などの映像処理を実行する装置に含まれることができる。

【0308】

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、メモリカード、格納媒体及び／又は他の格納装置を含むことができる。即ち、本文書で説明した実施形態は、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、各図面で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。

【0309】

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

10

20

30

40

50

ィア放送送受信装置、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（V o D）サービス提供装置、O T Tビデオ（O v e r t h e t o p v i d e o）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（3 D）ビデオ装置、V R（v i r t u a l r e a l i t y）装置、A R（a r g u m e n t e r e a l i t y）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、O T Tビデオ（O v e r t h e t o p v i d e o）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、D V R（D i g i t a l V i d e o R e c o r d e r）などを含むことができる。

10

【0310】

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（B D）、汎用直列バス（U S B）、R O M、P R O M、E P R O M、E E P R O M、R A M、C D－R O M、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。

20

【0311】

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。

30

【0312】

図16は、本文書において開示された実施形態が適用できるコンテンツストリーミングシステムの例を示す。

【0313】

図16を参照すると、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きくエンコーディングサーバ、ストリーミングサーバ、ウェブサーバ、メディアストレージ、ユーザ装置、及びマルチメディア入力装置を含むことができる。

【0314】

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

40

【0315】

前記ビットストリームは、本文書の実施形態に適用されるエンコード方法またはビットストリーム生成方法により生成されることができ、前記ストリーミングサーバは、前記ビットストリームを送信または受信する過程で一時的に前記ビットストリームを格納することができる。

【0316】

50

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

【0317】

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

10

【0318】

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

20

【0319】

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

【0320】

本明細書に記載された請求項は、多様な方式で組み合わせることができる。例えば、本明細書の方法請求項の技術的特徴が組み合わせて装置で実現されることができ、本明細書の装置請求項の技術的特徴が組み合わせて方法で実現されることができ、また、本明細書の方法請求項の技術的特徴と装置請求項の技術的特徴が組み合わせて装置で実現されることができ、本明細書の方法請求項の技術的特徴と装置請求項の技術的特徴が組み合わせて方法で実現されることができる。

30

40

50

【図 5】

【図 6】

[5]

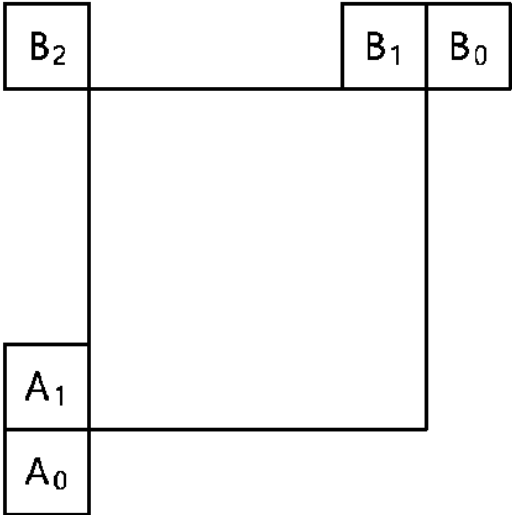
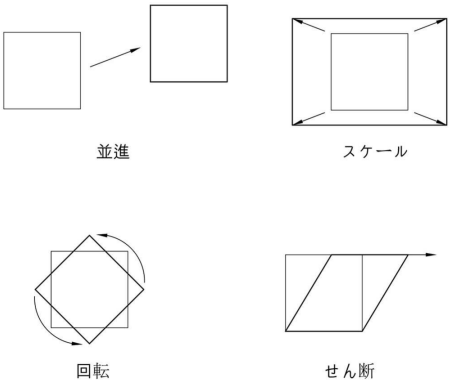


図 6



10

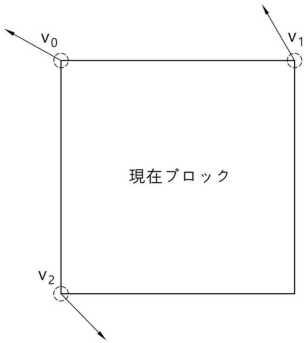
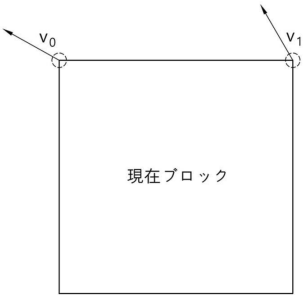
20

【図 7 a】

【図 7 b】

図 7a

図 7b



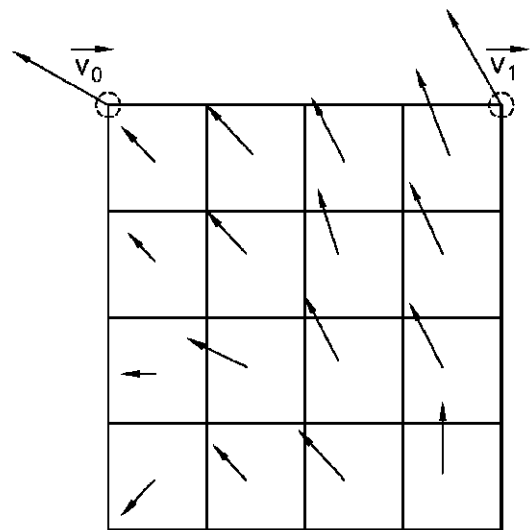
30

40

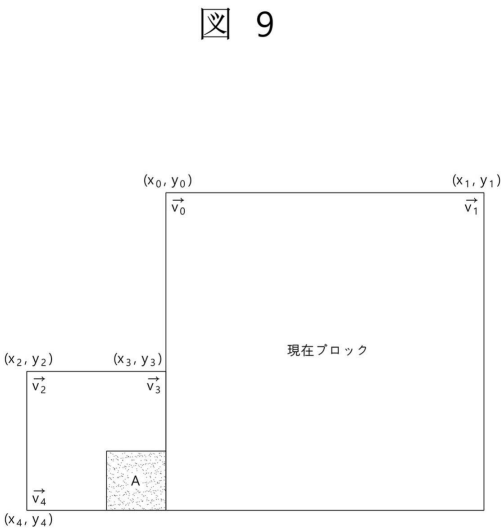
50

【図 8】

[58]



【図 9】

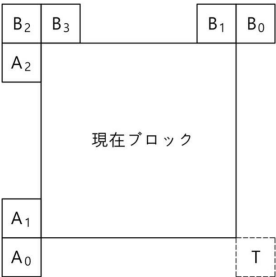


10

20

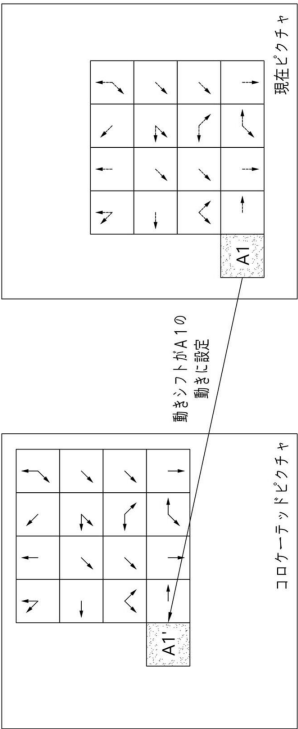
【図 10】

図 10



【図 11】

図 11



30

40

50

【図 1 2】

【図 1 3】

図 12

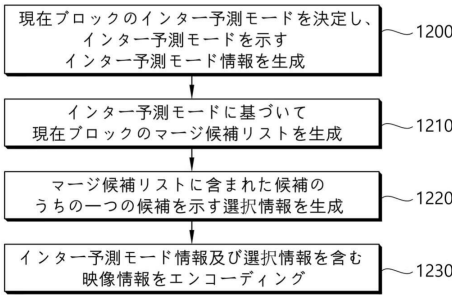
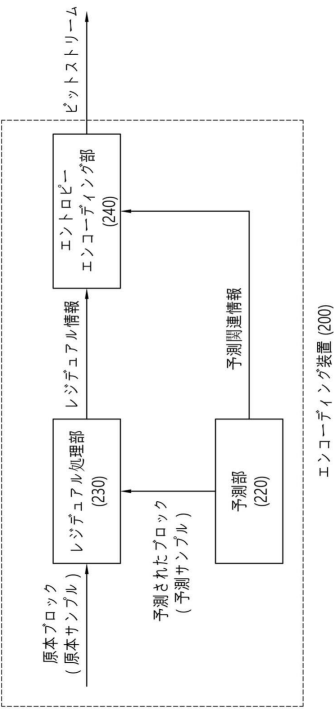


図 13



【図 1 4】

【図 1 5】

図 14

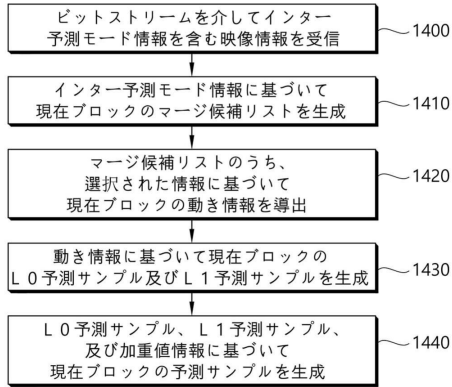
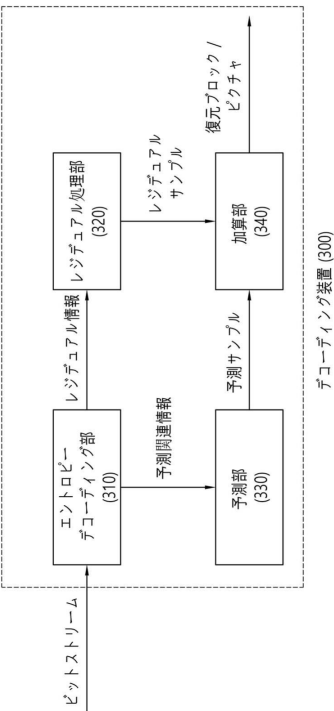


図 15



10

20

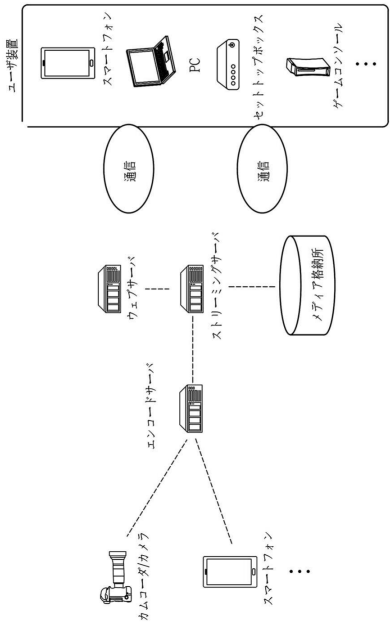
30

40

50

【図 16】

図 16



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-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-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-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 大濱 宏之
(56)参考文献 特表2022-522149 (JP, A)
特表2022-522398 (JP, A)
韓国公開特許第10-2017-0124079 (KR, A)
(58)調査した分野 (Int.Cl., DB名)
H04N 19/00-19/98