

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7391958号
(P7391958)

(45)発行日 令和5年12月5日(2023.12.5)

(24)登録日 令和5年11月27日(2023.11.27)

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

請求項の数 12 (全47頁)

(21)出願番号	特願2021-523795(P2021-523795)	(73)特許権者	516227559
(86)(22)出願日	令和1年11月8日(2019.11.8)		オッポ広東移動通信有限公司
(65)公表番号	特表2022-506429(P2022-506429 A)		GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP., LTD.
(43)公表日	令和4年1月17日(2022.1.17)		中華人民共和国広東省東莞市長安鎮烏沙海浜路18号
(86)国際出願番号	PCT/KR2019/015194		No. 18 Haibin Road, Wusha, Chang'an, Dongguan, Guangdong 523860 China
(87)国際公開番号	WO2020/096425		
(87)国際公開日	令和2年5月14日(2020.5.14)		
審査請求日	令和4年10月13日(2022.10.13)	(74)代理人	100091487
(31)優先権主張番号	10-2018-0136261		弁理士 中村 行孝
(32)優先日	平成30年11月8日(2018.11.8)	(74)代理人	100120031
(33)優先権主張国・地域又は機関	韓国(KR)		弁理士 宮嶋 学

最終頁に続く

(54)【発明の名称】 ビデオ信号符号化／復号方法及び前記方法に用いられる機器

(57)【特許請求の範囲】

【請求項1】

ビデオ復号方法であって、
 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るマージ候補の中から、前記現在ブロックのマージ候補を指定するステップと、
 前記現在ブロックのオフセットベクトルを導出するステップと、
 前記オフセットベクトルに前記マージ候補の動きベクトルを加算することで前記現在ブロックの動きベクトルを導出するステップと、を含み、
前記オフセットベクトルの大きさは、動き大きさ候補のうちのいずれか1つを指定する第1インデックス情報に基づいて、決定され、
動き大きさ候補の最大値又は最小値のうちの少なくとも1つは、前記現在ブロックの動きベクトル精度に基づいて、異なるように設定される、ビデオ復号方法。

【請求項2】

前記動き大きさ候補の最大値又は最小値のうちの少なくとも1つは、前記動き大きさ候補の範囲を示すフラグの値に基づいて、異なるように設定されることを特徴とする
 請求項1に記載のビデオ復号方法。

【請求項3】

前記フラグは、画像レベルで、シグナルリングされることを特徴とする
 請求項2に記載のビデオ復号方法。

【請求項 4】

前記オフセットベクトルの大きさは、前記第 1 インデックス情報により指定される前記動き大きさ候補で表される値にシフト演算を適用することで得られることを特徴とする
請求項 1 に記載のビデオ復号方法。

【請求項 5】

前記オフセットベクトルの方向は、ベクトル方向候補のうちのいずれか 1 つを指定する第 2 インデックス情報に基づいて、決定されることを特徴とする
請求項 1 に記載のビデオ復号方法。

【請求項 6】

動きベクトルオフセット大きさ候補の範囲は、前記動きベクトル精度に基づいて、異なるように設定されることを特徴とする
請求項 1 に記載のビデオ復号方法。

【請求項 7】

前記現在ブロックの動きベクトル精度が小数画素である場合、インデックス情報の値に対応する変数の値を 1、2、4、8 又は 16 とし、または、
前記現在ブロックの動きベクトル精度が整数画素である場合、インデックス情報の値に対応する変数の値を 4、8、16、32 又は 64 とすることを特徴とする
請求項 1 に記載のビデオ復号方法。

【請求項 8】

前記小数画素は $1/4$ 画素であることを特徴とする
請求項 7 に記載のビデオ復号方法。

【請求項 9】

前記動きベクトル精度を決定するための情報は、ビットストリームの画像レベルでシグナリングされることを特徴とする
請求項 1 に記載のビデオ復号方法。

【請求項 10】

ビデオ符号化方法であって、
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るマージ候補の中から、前記現在ブロックのマージ候補を選択するステップと、
前記現在ブロックのオフセットベクトルを導出するステップと、
前記オフセットベクトルに前記マージ候補の動きベクトルを加算することで前記現在ブロックの動きベクトルを導出するステップと、を含み、
第 1 インデックス情報に対して符号化を行うステップを更に含み、前記第 1 インデックス情報は、複数の動き大きさ候補のうち、前記オフセットベクトルの大きさの動き大きさ候補を指定するためのものであり、
動き大きさ候補の最大値又は最小値のうちの少なくとも 1 つは、前記現在ブロックの動きベクトル精度に基づいて、異なるように設定される、ビデオ符号化方法。

【請求項 11】

前記動き大きさ候補の範囲を示すフラグを符号化するステップを更に含み、前記符号化されたフラグの値に基づいて、前記動き大きさ候補の最大値又は最小値のうちの少なくとも 1 つが、異なるように設定され、又は、
前記ビデオ符号化方法は、第 2 インデックス情報に対して符号化を行うステップを更に含み、前記第 2 インデックス情報は、複数のベクトル方向候補のうち、前記オフセットベクトルの方向を示すベクトル方向候補を指定するためのものであることを特徴とする
請求項 10 に記載のビデオ符号化方法。

【請求項 12】

ビデオ復号機器であって、現在ブロックのマージ候補リストを生成し、前記マージ候補リストに含まれるマージ候補の中から、前記現在ブロックのマージ候補を指定し、前記現在ブロックのオフセットベクトルを導出し、前記オフセットベクトルに前記マージ候補の

10

20

30

40

50

動きベクトルを加算することで前記現在ブロックの動きベクトルを導出するように構成されるインター予測部を備え、前記オフセットベクトルの大きさは、動き大きさ候補のうちのいずれか1つを指定する第1インデックス情報に基づいて、決定され、動き大きさ候補の最大値又は最小値のうちの少なくとも1つは、前記現在ブロックの動きベクトル精度に基づいて、異なるように設定される、ビデオ復号機器。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ビデオ信号符号化／復号方法及び前記方法に用いられる機器に関する。

【背景技術】

【0002】

表示パネルがますます大きくなっているにつれて、画質がより高いビデオサービスは必要になっている。高精細度ビデオサービスの最大の問題は、データ量の大幅な増加である。このような問題を解決するために、ビデオ圧縮率の向上に関する検討を積極的に進行している。代表的な例として、2009年に、動画像専門家グループ（Motion Picture Experts Group：MPEG）及び国際電気通信連合－電気通信（International Telecommunication Union－Telecommunication：ITU－T）傘下のビデオ符号化専門家グループ（Video Coding Experts Group：VCEG）は、ビデオ符号化共同チーム（Joint Collaborative Team on Video Coding：JCT－VC）を確立した。JCT－VCは、ビデオ圧縮規格HEVC（高効率ビデオ符号化：High Efficiency Video Coding）を提出し、該規格が2013年1月25日に承認された。その圧縮性能は、H.264／AVCの約2倍である。高精細度ビデオサービスの急激な成長に伴い、HEVCの性能上の制限も次第に現れている。

【発明の概要】

【発明が解決しようとする課題】

【0003】

本発明の目的は、ビデオ信号に対して符号化／復号を行う場合に、オフセットベクトルに基づいて、マージ候補から導出された動きベクトルをリファインする方法及び該方法を実行するための機器を提供することである。

【0004】

本発明の目的は、ビデオ信号に対して符号化／復号を行う場合に、オフセットベクトルのシグナルリング方法、及び該方法を実行するための機器を提供することである。

【0005】

本発明が実現しようとする技術的課題は、上記言及した技術的課題に限定されず、本発明が属する技術分野において通常の知識を有する者は、下記説明により、言及されていない他の技術的課題を明瞭に理解することができる。

【課題を解決するための手段】

【0006】

本発明のビデオ信号復号／符号化方法は、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るマージ候補の中から、前記現在ブロックのマージ候補を指定するステップと、前記現在ブロックのオフセットベクトルを導出するステップと、前記オフセットベクトルに前記マージ候補の動きベクトルを加算することで前記現在ブロックの動きベクトルを導出するステップと、を含む。

【0007】

本発明によるビデオ信号復号／符号化方法において、前記オフセットベクトルの大きさは、動き大きさ候補のうちのいずれか1つを指定する第1インデックス情報に基づいて決定されることができる。

【0008】

10

20

30

40

50

本発明によるビデオ信号復号／符号化方法において、前記動き大きさ候補の最大値又は最小値のうちの少なくとも1つは、前記動き大きさ候補の範囲を示すフラグの値に基づいて異なるように設定されることができる。

【0009】

本発明によるビデオ信号復号／符号化方法において、前記フラグは、画像レベルでシグナルリングされることができる。

【0010】

本発明によるビデオ信号復号／符号化方法において、前記動き大きさ候補の最大値又は最小値のうちの少なくとも1つは、前記現在ブロックの動きベクトル精度に基づいて異なるように設定されることができる。

【0011】

本発明によるビデオ信号復号／符号化方法において、前記オフセットベクトルの大きさは、前記第1インデックス情報により指定される前記動き大きさ候補で表される値にシフト演算を適用することで得られることができる。

【0012】

本発明によるビデオ信号復号／符号化方法において、前記オフセットベクトルの方向は、ベクトル方向候補のうちのいずれか1つを指定する第2インデックス情報に基づいて、決定されることができる。

【0013】

本発明の上記簡単な要約の特徴は、下記説明される本発明の詳細な記載の例示的な実施形態だけであり、本発明の範囲を限定するものではない。

【発明の効果】

【0014】

本発明によれば、オフセットベクトルに基づいてマージ候補の動きベクトルをリファインすることでインター予測効率を向上させることができる。

【0015】

本発明によれば、オフセットベクトルの大きさ及び方向を適応的に決定することで、インター予測効率を向上させることができる。

【0016】

本発明で取得可能な効果は、上記効果に限定されず、本発明が属する技術分野において通常の知識を有する者は、下記記述により、言及されていない他の効果を明瞭に理解することができる。

【図面の簡単な説明】

【0017】

【図1】本発明の実施例によるビデオエンコーダ（encoder）を示すブロック図である。

【図2】本発明の実施例によるビデオデコーダ（decoder）を示すブロック図である。

【図3】本発明の実施例による基本符号化ツリーユニットを示す図である。

【図4】符号化ブロックの複数の分割タイプを示す図である。

【図5】符号化ツリーユニットの分割を示す図である。

【図6】本発明の実施例によるインター予測方法を示すフローチャートである。

【図7】対象の非線形動きを示す図である。

【図8】本発明の実施例によるアフィン動きに基づくインター予測方法を示すフローチャートである。

【図9】各アフィン動きモデルのアフィンシードベクトルの例を示す図である。

【図10】4パラメータ動きモデルにおけるサブブロックのアフィンベクトルの例を示す図である。

【図11】マージモードで現在ブロックの動き情報を導出するプロセスを示すフローチャートである。

10

20

30

40

50

【図12】マージ候補を導出するための候補ブロックの例を示す図である。

【図13】参照サンプルの位置を示す図である。

【図14】マージ候補を導出するための候補ブロックの例を示す図である。

【図15】参照サンプルの位置の変動例を示す図である。

【図16】参照サンプルの位置の変動例を示す図である。

【図17】オフセットベクトルの大きさを示す `distance_idx` 及びオフセットベクトルの方向を示す `direction_idx` の値に基づいたオフセットベクトルを示す図である。

【図18】オフセットベクトルの大きさを示す `distance_idx` 及びオフセットベクトルの方向を示す `direction_idx` の値に基づいたオフセットベクトルを示す図である。

10

【発明を実施するための形態】

【0018】

以下、図面を参照しながら、本発明の実施例を詳しく説明する。

【0019】

ビデオの符号化及び復号は、ブロックを単位として実行される。例えば、符号化ブロック、変換ブロック又は予測ブロックに対して、変換、量子化、予測、ループフィルタリング又は再構成などの符号化／復号処理を行うことができる。

【0020】

以下、符号化／復号されるべきブロックを「現在ブロック」と呼ぶ。例えば、現在符号化／復号処理ステップによれば、現在ブロックは、符号化ブロック、変換ブロック又は予測ブロックを表すことができる。

20

【0021】

なお、本明細書で用いられる用語「ユニット」は、特定の符号化／復号プロセスを実行するための基本ユニットを表し、また、「ブロック」は、所定の大きさのサンプルアレイを表すと理解されてもよい。特に明記しない限り、「ブロック」と「ユニット」は、相互交換可能に用いられる。例えば、後述される実施例において、符号化ブロック及び符号化ユニットは、同様な意味を有すると理解されてもよい。

【0022】

図1は、本発明の実施例によるビデオエンコーダ (encoder) を示すブロック図である。

30

【0023】

図1を参照すると、ビデオ符号化機器100は、画像分割部110、予測部120、125、変換部130、量子化部135、再配列部160、エントロピー符号化部165、逆量子化部140、逆変換部145、フィルタ部150及びメモリ155を備えてもよい。

【0024】

図1に示す各部材は、単独で示されてビデオ符号化機器における互いに異なる特徴的機能を表すが、各部材が分離したハードウェア又は単一のソフトウェアアセンブリで構成されることを表さない。つまり、各部材について、説明しやすくするために、各部材を配列することで、各部材のうちの少なくとも2つの部材を組み合わせる1つの部材を構成するか又は1つの部材を複数の部材に分けて、これにより機能を実行する。本発明の本質を逸脱しない限り、このような各部材を整合する実施例及び各部材を分離させる実施例も本発明の権利範囲に属する。

40

【0025】

なお、一部の構成要素は、本発明における本質機能を実行するための必須な構成要素ではなく、性能を向上させるための選択可能な構成要素だけである。本発明は、性能を向上させるだけに用いられる構成要素以外の、本発明の本質の実現に必要な部材のみを含むことで実行されてもよく、また、性能を向上させるだけに用いられる選択的な構成要素以外の必須な構成要素のみを含む構造も本発明の権利範囲に属する。

【0026】

50

画像分割部 110 は、入力された画像を少なくとも 1 つの処理ユニットに分割することができる。この場合、処理ユニットは、予測ユニット (Prediction Unit : PU) であってもよく、変換ユニット (Transform Unit : TU) であってもよく、更に符号化ユニット (Coding Unit : CU) であってもよい。画像分割部 110 は、1 つの画像を複数の符号化ユニット、予測ユニット及び変換ユニットの組み合わせに分割する。所定の基準 (例えば、コスト関数) に基づいて符号化ユニット、予測ユニット及び変換ユニットの組み合わせを選択して画像を符号化することができる。

【0027】

例えば、1 つの画像を複数の符号化ユニットに分割することができる。画像を符号化ユニットに分割するために、四分木構造 (Quad Tree Structure) のような再帰的ツリー構造を用いて、1 つのビデオ又は最大符号化ユニット (largest coding unit) をルートとして符号化ユニットを他の符号化ユニットに分割することができる。前記他の符号化ユニットは、分割された符号化ユニットの数と同じである数の子ノードを有してもよい。幾つかの制限によって分割されない符号化ユニットは、リーフノードとなる。つまり、1 つの符号化ユニットが正方形分割だけを実現できると仮定する場合、1 つの符号化ユニットを最大限 4 つの他の符号化ユニットに分割することができる。

【0028】

以下、本発明の実施例において、符号化ユニットは、符号化を実行するユニットを指してもよく、復号を実行するユニットを指してもよい。

【0029】

1 つの符号化ユニットにおける予測ユニットを大きさが同じである正方形又は矩形のうちの少なくとも 1 つなどの形状に分割することができ、1 つの符号化ユニットにおける 1 つの予測ユニットを形状及び／又は大きさがもう 1 つの予測ユニットと異なるものに分割することもできる。

【0030】

符号化ユニットに基づいてイントラ予測を実行する予測ユニットが最小符号化ユニットではない場合、複数の予測ユニット $N \times N$ に分割する必要がなく、イントラ予測を実行することができる。

【0031】

予測部 120、125 は、インター予測を実行するインター予測部 120 及びイントラ予測を実行するイントラ予測部 125 を含んでもよい。予測ユニットに対して、インター予測を用いるかそれともイントラ予測を用いるかを決定することができ、また、各予測方法に基づいて、具体的な情報 (例えば、イントラ予測モード、動きベクトル、参照画像など) を決定することができる。この場合、予測を実行する処理ユニットは、予測方法及び具体的なコンテンツを決定する処理ユニットと異なることがある。例えば、予測ユニットにより、予測方法及び予測モードなどを決定することができ、また、変換ユニットにより予測を実行することができる。生成された予測ブロックとオリジナルブロックとの間の残差値 (残差ブロック) を変換部 130 に入力することができる。なお、予測のための予測モード情報、動きベクトル情報などを残差値と共にエントロピー符号化部 165 において符号化してデコーダに伝送することができる。特定の符号化モードを用いる場合、予測部 120、125 により予測ブロックを生成することなく、直接的にオリジナルブロックを符号化してデコーダに伝送することもできる。

【0032】

インター予測部 120 は、現在画像の直前の画像又は直後の画像のうちの少なくとも 1 つの画像の情報に基づいて予測ユニットを予測することができる。幾つかの場合に、現在画像における符号化された一部の領域の情報に基づいて予測ユニットを予測することもできる。インター予測部 120 は、参照画像補間部、動き予測部、動き補償部を含んでもよい。

【0033】

10

20

30

40

50

参照画像補間部は、メモリ155から参照画像情報を受信し、また、参照画像から、整数画素又はその以下の画素情報を生成することができる。輝度画素について言えば、 $1/4$ 画素を単位として整数画素以下の画素情報を生成するために、フィルタ係数が異なるDCTに基づく8タップ補間フィルタ(DCT-based Interpolation Filter)を用いることができる。色度信号について言えば、 $1/8$ 画素を単位として整数画素以下の画素情報を生成するために、フィルタ係数が異なるDCTに基づく4タップ補間フィルタ(DCT-based Interpolation Filter)を用いることができる。

【0034】

動き予測部は、参照画像補間部により補間された参照画像に基づいて、動き予測を実行することができる。動きベクトルを算出するための方法として、全探索ブロックマッチングアルゴリズム(FBMA: Full search-based Block Matching Algorithm)、3段階探索法(TSS: Three Step Search)、新たな3段階探索アルゴリズム(NTS: New Three-Step Search Algorithm)等の複数の方法を用いることができる。補間された画素によれば、動きベクトルは、 $1/2$ 画素又は $1/4$ 画素を単位とした動きベクトル値を有してもよい。動き予測部において、異なる動き予測方法を用いることで現在予測ユニットを予測することができる。動き予測方法として、スキップ(Skip)方法、マージ(Merge)方法、アドバンスド動きベクトル予測(AMVP: Advanced Motion Vector Prediction)方法、イントラブロック複製(Intra Block Copy)方法などの複数の方法を用いることができる。

【0035】

イントラ予測部125は、現在画像における画素情報である、現在ブロックの周辺の参照画素情報に基づいて、予測ユニットを生成することができる。現在予測ユニットの隣接ブロックは、インター予測を実行したブロックであり、且つ参照画素は、インター予測を実行した画素である場合、インター予測を実行したブロックに含まれる参照画素を、周辺のイントラ予測を実行したブロックの参照画素情報とすることができる。つまり、参照画素が利用不可能である場合、利用不可能な参照画素情報の代わりに、利用可能な参照画素のうちの少なくとも1つの参照画素を用いることができる。

【0036】

イントラ予測において、予測モードは、予測方向に基づいて参照画素情報を用いる角度予測モード及び予測を実行する場合に方向情報を使用しない非角度モードを有してもよい。輝度情報を予測するためのモードは、色度情報を予測するためのモードと異なってもよい。色度情報を予測するために、輝度情報を予測するためのイントラ予測モード情報又は予測された輝度信号情報を用いることができる。

【0037】

イントラ予測を実行する場合、予測ユニットの大きさが変換ユニットの大きさと同じであると、予測ユニットの左側に位置する画素、左上に位置する画素、上方に位置する画素に基づいて、予測ユニットに対してイントラ予測を実行することができる。しかしながら、イントラ予測を実行する場合、予測ユニットの大きさが変換ユニットの大きさと異なると、変換ユニットの参照画素に基づいてイントラ予測を実行することができる。なお、最小符号化ユニットのみに対して、 $N \times N$ 分割を用いたイントラ予測を適用することができる。

【0038】

予測モードに基づいて参照画素に対して適応的イントラ平滑化(AIS: Adaptive Intra Smoothing)フィルタを適用した後、イントラ予測方法で、予測ブロックを生成することができる。参照画素に適用される適応的イントラ平滑化フィルタのタイプは異なる可能性がある。イントラ予測方法を実行するために、現在予測ユニットの周辺に位置する予測ユニットのイントラ予測モードに基づいて、現在予測ユニットのイントラ予測モードを予測することができる。周辺予測ユニットから予測されたモード情

10

20

30

40

50

報を用いて現在予測ユニットの予測モードを予測する場合、現在予測ユニットのイントラ予測モードが周辺予測ユニットのイントラ予測モードと同じであると、所定のフラグ情報を用いて、現在予測ユニットの予測モードが周辺予測ユニットと同じであることを表す情報を伝送することができる。現在予測ユニットの予測モードが周辺予測ユニットの予測モードと異なると、エントロピー符号化を実行することで、現在ブロックの予測モード情報を符号化することができる。

【0039】

なお、予測部120、125で生成された予測ユニットに基づいて予測を実行する予測ユニットと予測ユニットのオリジナルブロックとの差分値である残差値情報を含む残差ブロックを生成することができる。生成された残差ブロックを変換部130に入力することができる。

10

【0040】

変換部130において、離散コサイン変換(DCT: Discrete Cosine Transform)、離散サイン変換(DST: Discrete Sine Transform)などの変換方法で、オリジナルブロックと、予測部120及び125により生成された予測ユニット間の残差値(residual)情報を含む残差ブロックと、に対して変換を行うことができる。ここで、DCT変換カーネルは、DCT2又はDCT8のうちの少なくとも1つを含み、DSTcは、DST7を含む。残差ブロックを変換するためにDCTを適用するかそれともDSTを適用するかについて、残差ブロックを生成するための予測ユニットのイントラ予測モード情報に基づいて決定することができる。残差ブロックに対する変換をスキップすることもできる。残差ブロックに対する変換をスキップするかどうかを表すフラグを符号化することができる。大きさが閾値以下である残差ブロック、輝度成分又は色度成分(4:4:4フォーマット以下)について、変換をスキップすることを許容できる。

20

【0041】

量子化部135は、変換部130で周波数領域に変換された値を量子化することができる。量子化係数は、ブロック又は画像の重要度によって変わってもよい。量子化部135で算出された値は、逆量子化部140及び再配列部160に提供されてもよい。

【0042】

再配列部160は、量子化された残差値に対して係数値の再配列を実行することができる。

30

【0043】

再配列部160は、係数走査(Coefficient Scanning)方法で、2次元のブロック形状係数を1次元のベクトル形態に変更することができる。例えば、再配列部160は、ジグザグ走査(Zig-Zag Scan)方法で、DC係数ないし高周波数領域の係数を走査し、それを1次元ベクトル形態に変換することができる。変換ユニットの大きさ及びイントラ予測モードによれば、ジグザグ走査の代わりに、列方向に沿って2次元ブロック形状係数を走査する垂直走査及び行方向に沿って2次元ブロック形状係数を走査する水平走査を用いることもできる。つまり、変換ユニットの大きさ及びイントラ予測モードに基づいて、ジグザグ走査、垂直方向走査及び水平方向走査のうちのどちらを用いるかを決定することができる。

40

【0044】

エントロピー符号化部165は、再配列部160で算出された値に基づいてエントロピー符号化を実行することができる。例えば、エントロピー符号化は、指数ゴロムコード(Exponential Golomb)、コンテキスト適応型可変長符号化(CAVLC: Context-Adaptive Variable Length Coding)、コンテキスト適応型二値算術符号化(CABAC: Context-Adaptive Binary Arithmetic Coding)等の複数の符号化方法を用いることができる。

【0045】

50

エントロピー符号化部 165 は、再配列部 160 及び予測部 120、125 からの符号化ユニットの残差値係数情報及びブロックタイプ情報、予測モード情報、分割ユニット情報、予測ユニット情報及び伝送ユニット情報、動きベクトル情報、参照フレーム情報、ブロックの補間情報、フィルタリング情報などの複数の情報を符号化することができる。

【0046】

エントロピー符号化部 165 は、再配列部 160 から入力された符号化ユニットの係数値に対してエントロピー符号化を行うことができる。

【0047】

逆量子化部 140 及び逆変換部 145 は、量子化部 135 により量子化された複数の値に対して逆量子化を行い、変換部 130 により変換された値に対して逆変換を行う。逆量子化部 140 及び逆変換部 145 で生成された残差値と、予測部 120、125 に含まれる動き予測部、動き補償部及びイントラ予測部により予測された予測ユニットと、をマージすることで、再構成ブロック (Reconstructed Block) を生成することができる。

【0048】

フィルタ部 150 は、デブロッキングフィルタ、オフセット補正部 (offset correction unit)、適応型ループフィルタ (ALF: Adaptive Loop Filter) のうちの少なくとも 1 つを含んでもよい。

【0049】

デブロッキングフィルタは、ブロック間の境界により、再構成された画像で生成されたブロック歪みを除去することができる。デブロッキングを実行するかどうかを判定するために、ブロックに含まれる幾つかの列又は行に含まれる画素に基づいて、現在ブロックに対してデブロッキングフィルタを適用するかどうかを判定することができる。ブロックに対してデブロッキングフィルタを適用する場合、必要なデブロッキングフィルタリング強度に基づいて強フィルタ (Strong Filter) 又は弱フィルタ (Weak Filter) を適用することができる。なお、デブロッキングフィルタを用いる過程において、垂直フィルタリング又は水平フィルタリングを実行する場合、水平方向フィルタリング及び垂直方向フィルタリングを同期して実行することができる。

【0050】

オフセット補正部は、画素単位で、デブロッキングされるビデオにおける、オリジナルビデオとのオフセットを補正することができる。下記方式で、指定した画像に対してオフセット補正を行うことができる。つまり、ビデオに含まれる画素を所定の数の領域に分割した後、オフセット実行を必要とする領域を決定し、対応する領域に対してオフセットを適用するか又は各画素の縁情報を考慮してオフセットを適用する。

【0051】

フィルタリングされた再構成画像とオリジナルビデオを比較した値に基づいて、適応的ループフィルタリング (ALF: Adaptive Loop Filtering) を実行することができる。ビデオに含まれる画素を所定のグループに分割した後、該当するグループに用いられる 1 つのフィルタを決定することで、各グループに対してフィルタリングを差別的に実行することができる。適応的ループフィルタリングを適用するかどうかに関わる情報及び輝度情報を符号化ユニット (Coding Unit: CU) に応じて伝送することができる。適用される適応的ループフィルタの形状及びフィルタ係数は、各ブロックによって異なる。なお、適用されるブロックの特性に関わらず、同一タイプ (一定のタイプ) の適応的ループフィルタを適用することもできる。

【0052】

メモリ 155 は、フィルタ部 150 により算出された再構成ブロック又は画像を記憶することができる。インター予測を実行する場合、記憶された再構成ブロック又は画像を予測部 120、125 に提供することができる。

【0053】

図 2 は、本発明の実施例によるビデオデコーダ (decoder) を示すブロック図で

10

20

30

40

50

ある。

【0054】

図2を参照すると、ビデオデコーダ200は、エントロピー復号部210、再配列部215、逆量子化部220、逆変換部225、予測部230、予測部235、フィルタ部240、メモリ245を備えてもよい。

【0055】

ビデオエンコーダからビデオビットストリームを入力する場合、ビデオエンコーダによるステップと逆であるステップで、入力されたビットストリームを復号することができる。

【0056】

エントロピー復号部210は、ビデオエンコーダのエントロピー符号化部でエントロピー符号化を実行するステップと逆であるステップで、エントロピー復号を実行することができる。例えば、ビデオエンコーダで実行される方法に対応するように、指数ゴロムコード(Exponential Golomb)、コンテキスト適応型可変長符号化(CAVLC: Context-Adaptive Variable Length Coding)、コンテキスト適応型二値算術符号化(CABAC: Context-Adaptive Binary Arithmetic Coding)等の複数の方法を適用することができる。

10

【0057】

エントロピー復号部210は、エンコーダにより実行されたイントラ予測及びインター予測に関わる情報に対して復号を行うことができる。

20

【0058】

再配列部215は、符号化部においてエントロピー復号部210によりエントロピー復号されたビットストリームを再配列する方法で再配列を実行することができる。1次元ベクトル形態で表される複数の係数を2次元ブロック形状の係数に再構成することで再配列を行うことができる。再配列部215は、符号化部により実行された係数走査に関わる情報を受信し、該当する符号化部により実行された走査順に応じて逆走査を行う方法で、再配列を実行することができる。

【0059】

逆量子化部220は、エンコーダにより提供された量子化パラメータ及び再配列されたブロックの係数値に基づいて逆量子化を実行することができる。

30

【0060】

逆変換部225は、ビデオエンコーダにより実行された量子化結果に対して、逆離散コサイン変換、逆離散サイン変換を実行することができる。前記逆離散コサイン変換、逆離散サイン変換は、変換部により実行された変換の逆変換であり、つまり、離散コサイン変換、離散サイン変換の逆変換である。ここで、DCT変換カーネルは、DCT2又はDCT8のうちの少なくとも1つを含んでもよく、DST変換カーネルは、DST7を含んでもよい。代替的には、ビデオエンコーダにおいて変換をスキップすると、逆変換部225は、逆変換を実行しなくてもよい。逆変換は、ビデオエンコーダで決定された伝送ユニットにより実行されてもよい。ビデオデコーダの逆変換部225において、予測方法、現在ブロックの大きさ及び予測方向などの複数の情報に基づいて、変換方法(例えば、DCT又はDST)を選択的に実行することができる。

40

【0061】

予測部230、235は、エントロピー復号部210により提供された、予測ブロックの生成に関わる情報及びメモリ245により提供された先に復号されたブロック又は画像情報に基づいて、予測ブロックを生成することができる。

【0062】

上述したように、ビデオエンコーダにおける操作方式と同じである方式でイントラ予測を実行する場合、予測ユニットの大きさが変換ユニットの大きさと同じであると、予測ユニットの左側に位置する画素、左上に位置する画素、上方に位置する画素に基づいて、予測ユニットに対してイントラ予測を実行する。イントラ予測を実行する場合、予測ユニッ

50

トの大きさが変換ユニットの大きさと異なると、変換ユニットの参照画素に基づいてイントラ予測を実行することができる。なお、最小符号化ユニットのみに対して、 $N \times N$ 分割を用いたイントラ予測を適用することができる。

【0063】

予測部230、235は、予測ユニット判定部、インター予測部及びイントラ予測部を含んでもよい。予測ユニット判定部は、エントロピー復号部210から入力された予測ユニット情報、イントラ予測方法の予測モード情報、インター予測方法の動き予測関連情報などの複数の情報を受信し、現在符号化ユニットに基づいて、予測ユニットを分類し、予測ユニットがインター予測を実行しているかそれともイントラ予測を実行しているかを決定する。インター予測部230は、ビデオエンコーダにより提供された、現在予測ユニットのインター予測の実行に必要な情報を用いて、現在予測ユニットが属する現在画像の直前の画像又は直後の画像のうちの少なくとも1つの画像に含まれる情報に基づいて、現在予測ユニットに対してインター予測を実行することができる。代替的には、現在予測ユニットが属する現在画像における再構成された一部の領域の情報に基づいて、インター予測を実行することもできる。

【0064】

インター予測を実行するために、符号化ユニットに基づいて、対応する符号化ユニットに含まれる予測ユニットの動き予測方法がスキップモード (Skip Mode)、マージモード (Merge Mode)、アドバンスド動きベクトル予測モード (AMVP Mode)、イントラブロック複製モードのうちのどちらであるかを判定することができる。

【0065】

イントラ予測部235は、現在画像における画素情報に基づいて予測ブロックを生成することができる。予測ユニットが、イントラ予測を実行した予測ユニットである場合、ビデオエンコーダから提供された予測ユニットのイントラ予測モード情報に基づいて、イントラ予測を実行することができる。イントラ予測部235は、適応的イントラ平滑化 (AIS: Adaptive Intra Smoothing) フィルタ、参照画素補間部、DCフィルタを含んでもよい。適応的イントラ平滑化フィルタは、現在ブロックの参照画素に対してフィルタリングを実行する部分であり、また、現在予測ユニットの予測モードに基づいて、フィルタを適用するかどうかを決定することができる。ビデオエンコーダから提供された予測ユニットの予測モード及び適応的イントラ平滑化フィルタ情報を用いて、現在ブロックの参照画素に対して適応的イントラ平滑化フィルタリングを実行することができる。現在ブロックの予測モードが、適応的イントラ平滑化フィルタリングを実行しないモードであると、適応的イントラ平滑化フィルタを適用しなくてもよい。

【0066】

参照画素補間部について言えば、予測ユニットの予測モードが、参照画素に対して補間を行う画素値に基づいてイントラ予測を実行する予測ユニットであると、参照画素に対して補間を行うことで、整数値又はその以下の画素単位の参照画素を生成することができる。現在予測ユニットの予測モードが、参照画素に対して補間を行わない方式で予測ブロックを生成する予測モードであると、参照画素に対して補間を行わなくてもよい。現在ブロックの予測モードがDCモードであると、DCフィルタは、フィルタリングにより予測ブロックを生成することができる。

【0067】

再構成されたブロック又は画像をフィルタ部240に提供することができる。フィルタ部240は、デブロッキングフィルタ、オフセット補正部、ALFを含んでもよい。

【0068】

ビデオエンコーダから、該当するブロック又は画像に対してデブロッキングフィルタを適用するかどうかに関わる情報及びデブロッキングフィルタを適用する場合に強フィルタを用いるかそれとも弱フィルタを用いるかに関わる情報を受信することができる。ビデオデコーダのデブロッキングフィルタは、ビデオエンコーダから提供された、デブロッキングフィルタに関わる情報を受信することができ、また、ビデオデコーダは、該当するプロ

ックに対してデブロッキングフィルタリングを実行することができる。

【0069】

オフセット補正部は、符号化を行う場合にビデオに適用されるオフセット補正のタイプ及びオフセット情報に基づいて、再構成された画像に対してオフセット補正を実行することができる。

【0070】

エンコーダから提供された、ALFを適用するかどうかに関わる情報、ALF係数情報等に基づいて、ALFを符号化ユニットに適用することができる。このようなALF情報は、特定のパラメータ集合に含まれることで提供されてもよい。

【0071】

メモリ245は、再構成された画像又はブロックを記憶し、前記画像又はブロックを参照画像又は参照ブロックとして利用できるようになり、また、再構成された画像を出力部に提供することができる。

【0072】

図3は、本発明の実施例による基本符号化ツリーユニットを示す図である。

【0073】

大きさが最も大きい符号化ブロックを符号化ツリーブロックと定義することができる。1つの画像は、複数の符号化ツリーユニット（符号化ツリーユニットは、大きさがCoding Tree Unit:CTU）に分割されてもよい。最も大きい符号化ユニットであり、最大符号化ユニット（LCU: Largest Coding Unit）と呼ばれてもよい。図3は、1つの画像を複数の符号化ツリーユニットに分割する例を示す。

【0074】

符号化ツリーユニットの大きさは、画像レベル又はシーケンスレベルで定義されてもよい。このため、画像パラメータ集合又はシーケンスパラメータ集合により、信号を用いて、符号化ツリーユニットの大きさを示す情報を送信することができる。

【0075】

例えば、シーケンスにおける画像全体の符号化ツリーユニットの大きさを 128×128 とすることができる。代替的には、画像レベルの 128×128 又は 256×256 のうちのいずれか1つを符号化ツリーユニットの大きさと決定することができる。例えば、第1画像における符号化ツリーユニットの大きさを 128×128 とし、また、第2画像における符号化ツリーユニットの大きさを 256×256 とすることができる。

【0076】

符号化ツリーユニットの分割により符号化ブロックを生成することができる。符号化ブロックは、符号化／復号処理を行うための基本ユニットを表す。例えば、異なる符号化ブロックに応じて予測又は変換を実行することができ、又は、異なる符号化ブロックに応じて予測符号化モードを決定することができる。ここで、予測符号化モードは、予測画像を生成する方法を表す。例えば、予測符号化モードは、画面内予測（Intra Prediction:イントラ予測）、画面間予測（Inter Prediction:インター予測）、現在画像参照（Current Picture Referencing: CPR）、イントラブロック複製（Intra Block Copy:IBC）又は組み合わせ予測（Combined Prediction）を含んでもよい。符号化ブロックについて、イントラ予測、インター予測、現在画像参照又は組み合わせ予測のうちの少なくとも1つの予測符号化モードを用いて、符号化ブロックに関わる予測ブロックを生成することができる。

【0077】

ビットストリームを介して、現在ブロックの予測符号化モードを表す情報をシグナルリングすることができる。例えば、前記情報は、予測符号化モードがイントラモードであるかそれともインターモードであるかを示す1ビットフラグであってもよい。現在ブロックの予測符号化モードがインターモードであると決定された場合のみ、現在画像参照又は組み合わせ予測を用いることができる。

10

20

30

40

50

【0078】

現在画像参照は、現在画像を参照画像とし、現在画像における符号化／復号された領域から現在ブロックの予測ブロックを取得するために用いられる。ここで、現在画像は、現在ブロックを含む画像を意味する。ビットストリームを介して、現在ブロックに対して現在画像参照を適用するかどうかを表す情報をシグナルリングすることができる。例えば、前記情報は、1ビットフラグであってもよい。前記フラグが真である場合、現在ブロックの予測符号化モードを現在画像参照と決定することができる。前記フラグが偽である場合、現在ブロックの予測モードをインター予測と決定することができる。

【0079】

代替的には、参照画像インデックスに基づいて、現在ブロックの予測符号化モードを決定することができる。例えば、参照画像インデックスが現在画像を指し示す場合、現在ブロックの予測符号化モードを現在画像参照と決定することができる。参照画像インデックスが現在の画像ではない他の画像を指し示す場合、現在ブロックの予測符号化モードをインター予測と決定することができる。つまり、現在画像参照は、現在画像における符号化／復号された領域の情報をを用いる予測方法であり、また、インター予測は、符号化／復号された他の画像の情報をを用いる予測方法である。

【0080】

組み合わせ予測は、イントラ予測、インター予測及び現在画像参照のうちの2つ以上を組み合わせることで形成された符号化モードである。例えば、組み合わせ予測を適用する場合、イントラ予測、インター予測又は現在画像参照のうちの1つに基づいて、第1予測ブロックを生成することができ、また、もう1つに基づいて、第2予測ブロックを生成することができる。第1予測ブロック及び第2予測ブロックを生成する場合、第1予測ブロック及び第2予測ブロックの平均演算及び加重加算により最終的予測ブロックを生成することができる。ビットストリームを介して、組み合わせ予測を適用するかどうかを表す情報をシグナルリングすることができる。前記情報は、1ビットフラグであってもよい。

【0081】

図4は、符号化ブロックの複数の分割タイプを示す図である。

【0082】

四分木分割、二分木分割又は三分木分割に基づいて、符号化ブロックを複数の符号化ブロックに分割することができる。また、四分木分割、二分木分割又は三分木分割に基づいて、分割された符号化ブロックを更に複数の符号化ブロックに分割することもできる。

【0083】

四分木分割は、現在ブロックを4つのブロックに分割する分割技術である。四分木分割の結果として、現在ブロックを4つの正方形パーティション（図4の（a）における「SPLIT__QT」を参照する）に分割することができる。

【0084】

二分木分割は、現在ブロックを2つのブロックに分割する分割技術である。垂直方向に沿って（即ち、現在ブロックを横断する垂直線を用いる）、現在ブロックを2つのブロックに分割する過程を垂直方向二分木分割と呼び、また、水平方向に沿って（即ち、現在ブロックを横断する水平線を用いる）、現在ブロックを2つのブロックに分割する過程を水平方向二分木分割と呼ぶことができる。二分木分割を行った後に、現在ブロックを2つの非正方形パーティションに分割することができる。図4の（b）における「SPLIT__BT__VER」は、垂直方向二分木分割結果を表し、また、図4（c）における「SPLIT__BT__HOR」は、水平方向二分木分割結果を表す。

【0085】

三分木分割は、現在ブロックを3つのブロックに分割する分割技術である。垂直方向に沿って（即ち、現在ブロックを横断する2本の垂直線を用いる）、現在ブロックを3つのブロックに分割する過程を垂直方向三分木分割と呼び、また、水平方向に沿って（即ち、現在ブロックを横断する2本の水平線）、現在ブロックを3つのブロックに分割する過程を水平方向三分木分割と呼ぶことができる。三分木分割を行った後、現在ブロックを3つ

の非正方形パーティションに分割することができる。この場合、現在ブロックの中心に位置するパーティションの幅／高さは、他のパーティションの幅／高さの2倍であってもよい。図4(d)における「SPLIT__TT__VER」は、垂直方向三分木分割結果を表し、また、図4(e)における「SPLIT__TT__HOR」は、水平方向三分木分割結果を表す。

【0086】

符号化ツリーユニットの分割回数を分割深度(Partitioning Depth)と定義することができる。シーケンス又は画像レベルで、符号化ツリーユニットの最大分割深度を決定することができる。従って、符号化ツリーユニットの最大分割深度は、シーケンス又は画像によって異なってもよい。

【0087】

代替的には、複数の分割技術のそれぞれに対して最大分割深度を単独で決定することができる。例えば、四分木分割を許容する最大分割深度は、二分木分割及び／又は三分木分割を許容する最大分割深度と異なってもよい。

【0088】

エンコーダは、ビットストリームを介して、シグナルリング現在ブロックの分割形状又は分割深度のうちの少なくとも1つを表す情報をシグナルリングすることができる。デコーダは、ビットストリームから解析された情報に基づいて、符号化ツリーユニットの分割形状及び分割深度を決定することができる。

【0089】

図5は、符号化ツリーユニットの分割例を示す図である。

【0090】

四分木分割、二分木分割及び／又は三分木分割などの分割技術で符号化ブロックを分割する過程を多分木分割(Multi Tree Partitioning)と呼ぶことができる。

【0091】

符号化ブロックに対して多分木分割を適用することで生成された符号化ブロックを複数の下位符号化ブロックと呼ぶことができる。符号化ブロックの分割深度がkである場合、複数の下位符号化ブロックの分割深度をk+1とする。

【0092】

一方で、分割深度がk+1である複数の符号化ブロックについて、分割深度がkである符号化ブロックを上位符号化ブロックと呼ぶことができる。

【0093】

上位符号化ブロックの分割形状又は隣接符号化ブロックの分割タイプのうちの少なくとも1つに基づいて、現在符号化ブロックの分割タイプを決定することができる。ここで、隣接符号化ブロックは、現在符号化ブロックに隣接し、それは、現在符号化ブロックの上隣接ブロック、左隣接ブロック又は左上隅に隣接する隣接ブロックのうちの少なくとも1つを含んでもよい。ここで、分割タイプは、四分木分割を行うかどうか、二分木分割を行うかどうか、二分木分割方向、三分木分割を行うかどうか又は三分木分割方向のうちの少なくとも1つを含んでもよい。

【0094】

符号化ブロックの分割形状を決定するために、ビットストリームを介して、符号化ブロックが分割されたかどうかを表す情報をシグナルリングすることができる。前記情報は、1ビットフラグ「split__cu__flag」であり、前記フラグが真である場合、多分木分割技術で符号化ブロックを分割することを示す。

【0095】

「split__cu__flag」が真である場合、ビットストリームを介して、符号化ブロックが四分木分割されたかどうかを表す情報をシグナルリングすることができる。前記情報は、1ビットフラグ「split__qt__flag」であり、前記フラグが真である場合、符号化ブロックは、4つのブロックに分割されてもよい。

10

20

30

40

50

【0096】

例えば、図5に示す例において、符号化ツリーユニットが四分木分割されたことにより、分割深度が1である4つの符号化ブロックを生成することが図示される。なお、四分木分割結果として生成された4つの符号化ブロックのうちの1番目の符号化ブロック及び4番目の符号化ブロックに対して再び四分木分割を適用することを図示する。最終的に、分割深度が2である4つの符号化ブロックを生成することができる。

【0097】

なお、分割深度が2である符号化ブロックに対して再び四分木分割を適用することで分割深度が3である符号化ブロックを生成することができる。

【0098】

符号化ブロックに対して四分木分割を適用していない場合、符号化ブロックの大きさ、符号化ブロックが画像の境界に位置するかどうか、最大分割深度又は隣接ブロックの分割形状のうちの少なくとも1つを考慮することで、前記符号化ブロックに対して二分木分割を実行するかそれとも三分木分割を実行するかを決定することができる。前記符号化ブロックに対して二分木分割又は三分木分割を実行すると決定した場合、ビットストリームを介して、分割方向を表す情報をシグナルリングすることができる。前記情報は、1ビットフラグ「`mtt_split_cu_vertical_flag`」であってもよい。前記フラグに基づいて、分割方向が垂直方向であるかそれとも水平方向であるかを決定することができる。なお、ビットストリームを介して、前記符号化ブロックに対して二分木分割又は三分木分割のうちのどちらを適用するかを表す情報をシグナルリングすることができる。前記情報は、1ビットフラグ「`mtt_split_cu_binary_flag`」であってもよい。前記フラグに基づいて、前記符号化ブロックに対して二分木分割を適用するかそれとも三分木分割を適用するかを決定することができる。

【0099】

例えば、図5に示す例において、分割深度が1である符号化ブロックに対して垂直方向二分木分割を適用することを図示する。前記分割結果として生成された符号化ブロックのうちの左側符号化ブロックに対して垂直方向三分木分割を適用し、右側符号化ブロックに対して垂直方向二分木分割を適用する。

【0100】

インター予測は、直前の画像の情報をを用いて現在ブロックを予測する予測符号化モードである。例えば、直前の画像における、現在ブロックと同一の位置でのブロック（以下、コロケートッドブロックと呼ぶ。`Collocated block`）を現在ブロックの予測ブロックとすることができる。以下、位置が現在ブロックと同じであるブロックに基づいて生成された予測ブロックをコロケートッド予測ブロック（`Collocated Prediction Block`）と呼ぶ。

【0101】

一方で、直前の画像に存在する対象が現在画像における他の位置に移動した場合、対象の動きにより、現在ブロックを効果的に予測することができる。例えば、直前の画像と現在画像を比較することで対象の移動方向及び大きさを了解できると、対象の動き情報を考慮して現在ブロックの予測ブロック（又は予測画像）を生成することができる。以下、動き情報を用いることで生成された予測ブロックを動き予測ブロックと呼ぶことができる。

【0102】

現在ブロックから予測ブロックを減算することで残差ブロック（`residual block`）を生成することができる。この場合、対象の動きが存在する場合、コロケートッド予測ブロックの代わりに動き予測ブロックを用いることで、残差ブロックのエネルギーを減少させ、残差ブロックの圧縮性能を向上させることができる。

【0103】

上述したように、動き情報を用いて予測ブロックを生成する過程を動き補償予測と呼ぶことができる。大部分のインター予測において、動き補償予測に基づいて予測ブロックを生成することができる。

10

20

30

40

50

【0104】

動き情報は、動きベクトル、参照画像インデックス、予測方向又は双方向加重値インデックスのうちの少なくとも1つを含んでもよい。動きベクトルは、対象の移動方向及び大きさを表す。参照画像インデックスは、参照画像リストに含まれる複数の参照画像のうちの、現在ブロックの参照画像を指定する。予測方向は、一方向L0予測、一方向L1予測又は双方向予測（L0予測及びL1予測）のうちのいずれか1つを指す。現在ブロックの予測方向に基づいて、L0方向の動き情報又はL1方向の動き情報のうちの少なくとも1つを用いることができる。双方向加重値インデックスは、L0予測ブロックに適用される加重値及びL1予測ブロックに適用される加重値を指定する。

【0105】

図6は、本発明の実施例によるインター予測方法を示すフローチャートである。

【0106】

図6を参照すると、インター予測方法は、現在ブロックのインター予測モードを決定するステップ（S601）と、決定されたインター予測モードに基づいて、現在ブロックの動き情報を得るステップ（S602）と、得られた動き情報に基づいて、現在ブロックに対して動き補償予測を実行するステップ（S603）と、を含む。

【0107】

ここで、インター予測モードは、現在ブロックの動き情報を決定するための複数の技術を表し、また、並進（Translation）動き情報を用いたインター予測モード及びアフィン（Affine）動き情報を用いたインター予測モードを含んでもよい。例えば、並進動き情報を用いたインター予測モードは、マージモード及びアドバンスド動きベクトル予測モードを含んでもよい。アフィン動き情報を用いたインター予測モードは、アフィンマージモード及びアフィン動きベクトル予測モードを含んでもよい。インター予測モードによれば、現在ブロックに隣接する隣接ブロック又はビットストリームから解析された情報に基づいて、現在ブロックの動き情報を決定することができる。

【0108】

以下、アフィン動き情報を用いたインター予測方法を詳しく説明する。

【0109】

図7は、対象の非線形動きを示す図である。

【0110】

ビデオ内の対象の動きは、非線形動きである可能性がある。例えば、図7に示す例のように、カメラ拡大（Zoom-in）、縮小（Zoom-out）、回転（Rotation）又はアフィン変換などの対象の非線形動きを発生する可能性がある。対象の非線形動きを発生した場合、並進動きベクトルにより対象の動きを有効に表すことができない。従って、対象の非線形動きを発生した部分において、並進動きの代わりに、アフィン動きを用いることで、符号化効率を向上させることができる。

【0111】

図8は、本発明の実施例によるアフィン動きに基づくインター予測方法を示すフローチャートである。

【0112】

ビットストリームから解析された情報に基づいて、現在ブロックに対してアフィン動きに基づくインター予測技術を適用するかどうかを決定することができる。具体的には、現在ブロックに対してアフィンマージモードを適用するかどうかを示すフラグ又は現在ブロックに対してアフィン動きベクトル予測モードを適用するかどうかを示すフラグのうちの少なくとも1つに基づいて、現在ブロックに対してアフィン動きに基づくインター予測技術を適用するかどうかを決定することができる。

【0113】

アフィン動きに基づくインター予測技術を現在ブロックに適用する場合、現在ブロックのアフィン動きモデルを決定することができる（S801）。アフィン動きモデルは、6パラメータアフィン動きモデル又は4パラメータアフィン動きモデルのうちの少なくとも

10

20

30

40

50

1つにより決定されてもよい。6パラメータアフィン動きモデルは、6つのパラメータでアフィン動きを表し、また、4パラメータアフィン動きモデルは、4つのパラメータでアフィン動きを表す。

【0114】

式1は、6つのパラメータでアフィン動きを表す場合である。アフィン動きは、アフィンシードベクトルにより決定された所定領域の並進動きを表す。

【数1】

【式1】

$$\begin{aligned} v_x &= ax - by + e \\ v_y &= cx + dy + f \end{aligned}$$

10

【0115】

6つのパラメータでアフィン動きを表す場合、複雑な動きを表すことができるが、各パラメータに対して符号化を行う場合に必要なビット数は増加するため、符号化効率を低下させる。従って、4つのパラメータで、アフィン動きを表すこともできる。式2は、4つのパラメータで、アフィン動きを表す場合である。

20

【数2】

【式2】

$$\begin{aligned} v_x &= ax - by + e \\ v_y &= bx + ay + f \end{aligned}$$

【0116】

現在ブロックのアフィン動きモデルを決定するための情報に対して符号化を行い、ビットストリームを介して、シグナルリングすることができる。例えば、前記情報は、1ビットフラグ「`affine_type_flag`」であってもよい。前記フラグの値が0である場合、4パラメータアフィン動きモデルを適用することを表す。前記フラグの値が1である場合、6パラメータアフィン動きモデルを適用することを表す。前記フラグは、スライス、画像ブロック又はブロック（例えば、符号化ブロック又は符号化ツリーユニット）を単位として符号化を行うことができる。スライスレベルで、信号を用いてフラグを送信する場合、前記スライスに属する全てのブロックに対して、前記スライスレベルで決定されたアフィン動きモデルを適用することができる。

30

【0117】

代替的には、現在ブロックのアフィンインター予測モードに基づいて、現在ブロックのアフィン動きモデルを決定することができる。例えば、アフィンマージモードを適用する場合、現在ブロックのアフィン動きモデルを4パラメータ動きモデルと決定することができる。一方で、アフィン動きベクトル予測モードを適用する場合、現在ブロックのアフィン動きモデルを決定するための情報に対して符号化を行い、ビットストリームを介してシグナルリングすることができる。例えば、現在ブロックに対してアフィン動きベクトル予測モードを適用する場合、1ビットフラグ「`affine_type_flag`」に基づいて、現在ブロックのアフィン動きモデルを決定することができる。

40

【0118】

次に、現在ブロックのアフィンシードベクトルを導出することができる（S802）。

50

4パラメータアフィン動きモデルを選択した場合、現在ブロックの2つの制御点での動きベクトルを導出することができる。なお、6パラメータアフィン動きモデルを選択した場合、現在ブロックの3つの制御点での動きベクトルを導出することができる。制御点での動きベクトルをアフィンシードベクトルと呼ぶことができる。制御点は、現在ブロックの左上隅、右上隅又は左下隅のうちの少なくとも1つを含んでもよい。

【0119】

図9は、各アフィン動きモデルのアフィンシードベクトルの例を示す図である。

【0120】

4パラメータアフィン動きモデルにおいて、左上隅、右上隅又は左下隅のうちの2つに関わるアフィンシードベクトルを導出することができる。例えば、図9の(a)に示す例のように、4パラメータアフィン動きモデルを選択した場合、現在ブロックの左上隅(例えば、左上サンプル(x_0, y_0))に関わるアフィンシードベクトル sv_0 及び現在ブロックの右上隅(例えば、右上サンプル(x_1, y_1))に関わるアフィンシードベクトル sv_1 を用いてアフィンベクトルを導出することができる。また、左上隅に関わるアフィンシードベクトルの代わりに、左下隅に関わるアフィンシードベクトルを用いることもできる。又は、右上隅に関わるアフィンシードベクトルの代わりに、左下隅に関わるアフィンシードベクトルを用いることもできる。

【0121】

6パラメータアフィン動きモデルにおいて、左上隅、右上隅及び左下隅に関わるアフィンシードベクトルを導出することができる。例えば、図9の(b)に示す例のように、6パラメータアフィン動きモデルを選択した場合、現在ブロックの左上隅(例えば、左上サンプル(x_0, y_0))に関わるアフィンシードベクトル sv_0 、現在ブロックの右上隅(例えば、右上サンプル(x_1, y_1))に関わるアフィンシードベクトル sv_1 及び現在ブロックの左下隅(例えば、左下サンプル(x_2, y_2))に関わるアフィンシードベクトル sv_2 を用いてアフィンベクトルを導出することができる。

【0122】

後述する実施例において、4パラメータアフィン動きモデルで、左上制御点及び右上制御点のアフィンシードベクトルをそれぞれ第1アフィンシードベクトル及び第2アフィンシードベクトルと呼ぶ。後述する、第1アフィンシードベクトル及び第2アフィンシードベクトルを用いる実施例において、第1アフィンシードベクトル及び第2アフィンシードベクトルのうちの少なくとも1つを左下制御点のアフィンシードベクトル(第3アフィンシードベクトル)又は右下制御点のアフィンシードベクトル(第4アフィンシードベクトル)に置き換えることができる。

【0123】

なお、6パラメータアフィン動きモデルで、左上制御点、右上制御点及び左下制御点のアフィンシードベクトルをそれぞれ第1アフィンシードベクトル、第2アフィンシードベクトル及び第3アフィンシードベクトルと呼ぶ。後述する、第1アフィンシードベクトル、第2アフィンシードベクトル及び第3アフィンシードベクトルを用いる実施例において、第1アフィンシードベクトル、第2アフィンシードベクトル及び第3アフィンシードベクトルのうちの少なくとも1つを右下制御点のアフィンシードベクトル(第4アフィンシードベクトル)に置き換えることができる。

【0124】

アフィンシードベクトルを用いて、異なるサブブロックに応じて、アフィンベクトルを導出することができる(S803)。ここで、アフィンベクトルは、アフィンシードベクトルに基づいて導出された並進動きベクトルを表す。サブブロックのアフィンベクトルをアフィンサブブロック動きベクトル又はサブブロック動きベクトルと呼ぶことができる。

【0125】

図10は、4パラメータ動きモデルにおけるサブブロックのアフィンベクトルの例を示す図である。

【0126】

制御点の位置、サブブロックの位置及びアフィンシードベクトルに基づいてサブブロックのアフィンベクトルを導出することができる。例えば、式3は、アフィンサブブロックベクトルの導出例を示す。

【数3】

【式3】

$$v_x = \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)}(x - x_0) - \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)}(y - y_0) + sv_{0x}$$

$$v_y = \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)}(x - x_0) - \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)}(y - y_0) + sv_{0y}$$

10

【0127】

前記式3において、 (x, y) は、サブブロックの位置を表す。ここで、サブブロックの位置は、サブブロックに含まれる参照サンプルの位置を表す。参照サンプルは、サブブロックの左上隅に位置するサンプルであってもよく、又は x 軸又は y 軸座標における少なくとも1つの、中心位置に位置するサンプルであってもよい。 (x_0, y_0) は、第1制御点の位置を表し、また、 (sv_{0x}, sv_{0y}) は、第1アフィンシードベクトルを表す。なお、 (x_1, y_1) は、第2制御点の位置を表し、また、 (sv_{1x}, sv_{1y}) は、第2アフィンシードベクトルを表す。

20

【0128】

第1制御点及び第2制御点はそれぞれ現在ブロックの左上隅及び右上隅に対応する場合、 $x_1 - x_0$ を現在ブロックの幅と同じである値と設定することができる。

【0129】

続いて、各サブブロックのアフィンベクトルを用いて各サブブロックに対して動き補償予測を実行することができる(S804)。動き補償予測を実行した後、各サブブロックに関わる予測ブロックを生成することができる。サブブロックの予測ブロックを現在ブロックの予測ブロックと設定することができる。

30

【0130】

続いて、並進動き情報を用いたインター予測方法を詳しく説明する。

【0131】

現在ブロックの他のブロックの動き情報から現在ブロックの動き情報を導出することができる。ここで、他のブロックは、現在ブロックよりも優先してインター予測で符号化／復号を行うブロックであってもよい。現在ブロックの動き情報を他のブロックの動き情報と同じくする場合をマージモードと定義することができる。また、他のブロックの動きベクトルを現在ブロックの動きベクトルの予測値と設定する場合を動きベクトル予測モードと定義することができる。

【0132】

図11は、マージモードで現在ブロックの動き情報を導出するプロセスを示すフローチャートである。

40

【0133】

現在ブロックのマージ候補を導出することができる(S1101)。現在ブロックのマージ候補は、現在ブロックの前の、インター予測で符号化／復号されたブロックから導出される。

【0134】

図12は、マージ候補を導出するための候補ブロックの例を示す図である。

【0135】

候補ブロックは、現在ブロックに隣接するサンプルを含む隣接ブロック又は現在ブロッ

50

クに隣接しないサンプルを含む非隣接ブロックのうちの少なくとも1つを含んでもよい。以下に、候補ブロックを決定するためのサンプルを参照サンプルと指定する。なお、現在ブロックに隣接する参照サンプルを隣接参照サンプルと呼び、現在ブロックに隣接しない参照サンプルを非隣接参照サンプルと呼ぶ。

【0136】

隣接参照サンプルは、現在ブロックの最左側列の隣接列、又は現在ブロックの最上行の隣接行に含まれてもよい。例えば、現在ブロックの左上サンプルの座標が $(0, 0)$ であると、 $(-1, H-1)$ 位置での参照サンプルを含むブロック、 $(W-1, -1)$ 位置での参照サンプルを含むブロック、 $(W, -1)$ 位置での参照サンプルを含むブロック、 $(-1, H)$ 位置での参照サンプルを含むブロック又は $(-1, -1)$ 位置での参照サンプルを含むブロックのうちの1つは、候補ブロックとして用いられてもよい。図面に示すように、インデックスが0から4である隣接ブロックを候補ブロックとして用いることができる。

10

【0137】

非隣接参照サンプルは、現在ブロックに隣接する参照サンプルとのx軸距離又はy軸距離のうちの少なくとも1つが事前定義された値を有するサンプルを表す。例えば、左側参照サンプルとのx軸距離が事前定義された値である参照サンプルを含むブロック、上参照サンプルとのy軸距離が事前定義された値である非隣接サンプルを含むブロック、又は左上参照サンプルとのx軸距離及びy軸距離が事前定義された値である非隣接サンプルを含むブロックのうちの1つを候補ブロックとして用いることができる。事前定義された値は、4、8、12、16等の整数であってもよい。図面に示すように、インデックスが5から26であるブロックのうちの少なくとも1つを候補ブロックとして用いることができる。

20

【0138】

隣接参照サンプルと同一の垂直線、水平線又は対角線に位置しないサンプルを非隣接参照サンプルとすることができる。

【0139】

図13は、参照サンプルの位置を示す図である。

【0140】

図13に示す例のように、上非隣接参照サンプルのx座標を上隣接参照サンプルのx座標と異なるように設定することができる。例えば、上隣接参照サンプルの位置が $(W-1, -1)$ である場合、y軸に沿って上隣接参照サンプルからNだけ離れる上非隣接参照サンプルの位置を $((W/2) - 1, -1 - N)$ とし、また、y軸に沿って上隣接参照サンプルから2Nだけ離れる上非隣接参照サンプルの位置を $(0, -1 - 2N)$ とすることができる。つまり、非隣接参照サンプルの位置は、隣接参照サンプルの位置及び隣接参照サンプルとの距離に基づいて決定されてもよい。

30

【0141】

以下、候補ブロックのうちの、隣接参照サンプルを含む候補ブロックを隣接ブロックと呼び、非隣接参照サンプルを含むブロックを非隣接ブロックと呼ぶ。

【0142】

現在ブロックと候補ブロックとの距離が閾値以上である場合、前記候補ブロックを、マージ候補として利用不可能であるものと設定することができる。前記閾値は、符号化ツリーユニットの大きさによって決まってもよい。例えば、前記閾値を符号化ツリーユニットの高さ(ctu_height)とするか又は符号化ツリーユニットの高さに対してオフセット値を加算又は減算することで得られた値(例えば、 $\text{ctu_height} \pm N$)とすることができる。オフセット値Nは、エンコーダ及びデコーダで事前定義された値であり、4、8、16、32又はctu_heightとしてもよい。

40

【0143】

現在ブロックのy軸座標と候補ブロックに含まれるサンプルのy軸座標との差分値が閾値より大きい場合、候補ブロックを、マージ候補として利用不可能であるものと決定することができる。

50

【0144】

代替的には、現在ブロックと同一の符号化ツリーユニットに属しない候補ブロックを、マージ候補として利用不可能であるものと設定することができる。例えば、参照サンプルは現在ブロックが属する符号化ツリーユニットの上境界を越えた場合、参照サンプルを含む候補ブロックを、マージ候補として利用不可能であるものと設定することができる。

【0145】

現在ブロックの上境界が符号化ツリーユニットの上境界に隣接する場合、複数の候補ブロックを、マージ候補として利用不可能であるものと設定すれば、現在ブロックの符号化／復号効率を低下させる。上記問題を解決するために、候補ブロックを設定することで、現在ブロックの上方に位置する候補ブロックの数を現在ブロックの左側に位置する候補ブロックの数より多くすることができる。

10

【0146】

図14は、マージ候補を導出するための候補ブロックの例を示す図である。

【0147】

図14に示す例のように、現在ブロックの上方のN個のブロック列に属する上ブロック及び現在ブロックの左側のM個のブロック列に属する左側ブロックを候補ブロックとすることができる。この場合、MをNより大きいものとすることで、左側候補ブロックの数を上候補ブロックの数より大きいものとすることができる。

【0148】

例えば、現在ブロック内の参照サンプルのy軸座標と候補ブロックとして利用可能である上ブロックのy軸座標との差分値を現在ブロックの高さのN倍以下にすることができる。なお、現在ブロック内の参照サンプルのx軸座標と候補ブロックとして利用可能である左側ブロックのx軸座標との差分値を現在ブロックの幅のM倍以下にすることができる。

20

【0149】

例えば、図14に示す例において、現在ブロックの上方の2つのブロック列に属するブロック及び現在ブロックの左側の5つのブロック列に属するブロックを候補ブロックとすることを図示する。

【0150】

もう1つの例として、候補ブロックは、現在ブロックと同一の符号化ツリーユニットに属しない場合、前記候補ブロックの代わりに、現在ブロックと同一の符号化ツリーユニットに属するブロック、又は前記符号化ツリーユニットの境界に隣接する参照サンプルを含むブロックを用いて、マージ候補を導出することができる。

30

【0151】

図15は、参照サンプルの位置の変動例を示す図である。

【0152】

参照サンプルが現在ブロックと異なる符号化ツリーユニットに含まれ、且つ前記参照サンプルが前記符号化ツリーユニットの境界に隣接しない場合、前記参照サンプルの代わりに、前記符号化ツリーユニットの境界に隣接する参照サンプルを用いて候補ブロック参照サンプルを決定することができる。

【0153】

40

例えば、図15(a)及び図15(b)に示す例において、現在ブロックの上境界と符号化ツリーユニットの上境界が互いに接触する場合、現在ブロックの上方の参照サンプルは、現在ブロックと異なる符号化ツリーユニットに属する。符号化ツリーユニットの上境界に隣接するサンプルで、現在ブロックと異なる符号化ツリーユニットに属する参照サンプルのうちの、符号化ツリーユニットの上境界に隣接しない参照サンプルを置き換えることができる。

【0154】

例えば、図15(a)の例に示すように、位置6での参照サンプルは、符号化ツリーユニットの上境界の位置6'に位置するサンプルに置き換えられる。図15(b)の例に示すように、位置15での参照サンプルは、符号化ツリーユニットの上境界の位置15'に位置

50

するサンプルに置き換えられる。この場合、置換えサンプルの y 座標は、符号化ツリーユニットの隣接位置に変わってもよく、置換えサンプルの x 座標は、参照サンプルと同じであるように設定されてもよい。例えば、位置 6' でのサンプルは、位置 6 でのサンプルと同一の x 座標を有してもよく、位置 15' でのサンプルは、位置 15 でのサンプルと同一の x 座標を有してもよい。

【0155】

代替的には、参照サンプルの x 座標に対してオフセット値を加算又は減算することで得られた値を置換えサンプルの x 座標とすることができる。例えば、現在ブロックの上方に位置する隣接参照サンプルと非隣接参照サンプルの x 座標が同じである場合、参照サンプルの x 座標に対してオフセット値を加算又は減算することで得られた値を置換えサンプルの x 座標とすることができる。その目的は、非隣接参照サンプルを置き換えるための置換えサンプルが他の非隣接参照サンプル又は隣接参照サンプルと同一の位置にあることを防止することである。

【0156】

図 16 は、参照サンプルの位置の変動例を示す図である。

【0157】

符号化ツリーユニットの境界に位置するサンプルで、現在ブロックと異なる符号化ツリーユニットに含まれて且つ符号化ツリーユニットの境界に隣接しない参照サンプルを置き換える場合、参照サンプルの x 座標に対してオフセット値を加算又は減算することで得られた値を置換えサンプルの x 座標とすることができる。

【0158】

例えば、図 16 に示す例において、位置 6 での参照サンプル及び位置 15 での参照サンプルはそれぞれ、符号化ツリーユニットの上境界に隣接する行と同一の y 座標を有する、位置 6' でのサンプル及び位置 15' でのサンプルに置き換えられてもよい。この場合、位置 6' でのサンプルの x 座標は、位置 6 での参照サンプルの x 座標との差分値が $W/2$ であるものと設定されてもよく、位置 15' でのサンプルの x 座標は、位置 15 での参照サンプルの x 座標との差分値が $W-1$ であるものと設定されてもよい。

【0159】

図 15 及び図 16 に示す例と異なっており、現在ブロックの最上行の上方に位置する行の y 座標又は符号化ツリーユニットの上境界の y 座標は更に、置換えサンプルの y 座標と設定されてもよい。

【0160】

図示されていないが、符号化ツリーユニットの左境界に基づいて、参照サンプルを代替するためのサンプルを決定することができる。例えば、参照サンプルは、現在ブロックと同一の符号化ツリーユニットに含まれず、且つ符号化ツリーユニットの左境界に隣接しない場合、前記参照サンプルを符号化ツリーユニットの左境界に隣接するサンプルに置き換えることができる。この場合、置換えサンプルは、参照サンプルと同一の y 座標を有してもよく、又は、参照サンプルの y 座標に対してオフセット値を加算又は減算することで得られた y 座標を有してもよい。

【0161】

続いて、置換えサンプルを含むブロックを候補ブロックとし、前記候補ブロックに基づいて、現在ブロックのマージ候補を導出することができる。

【0162】

現在ブロックと異なる画像に含まれる時間的隣接ブロックからマージ候補を導出することもできる。例えば、コロケーテッド画像に含まれるコロケーテッドブロックからマージ候補を導出することができる。

【0163】

マージ候補の動き情報を候補ブロックの動き情報と同じであるように設定することができる。例えば、候補ブロックの動きベクトル、参照画像インデックス、予測方向又は双方向加重値インデックスのうちの少なくとも 1 つをマージ候補の動き情報とすることができる。

10

20

30

40

50

る。

【0164】

マージ候補を含むマージ候補リストを生成することができる（S1102）。前記マージ候補は、現在ブロックに隣接する隣接ブロックから導出された隣接マージ候補、及び非隣接ブロックから導出された非隣接マージ候補に分類されてもよい。

【0165】

所定の順番に応じて、マージ候補リスト内の複数のマージ候補のインデックスを割り当てることができる。例えば、隣接マージ候補に割り当てられるインデックスは、非隣接マージ候補に割り当てられるインデックスよりも小さい値を有してもよい。代替的には、図12又は図14に示す各ブロックのインデックスに基づいて、インデックスを各マージ候補に割り当てることができる。

10

【0166】

マージ候補リストに複数のマージ候補が含まれる場合、前記複数のマージ候補のうちの少なくとも1つを選択することができる（S1103）。この場合、ビットストリームを介して、現在ブロックの動き情報が隣接マージ候補から導出されたものであるかどうかを示す情報をシグナルリングすることができる。前記情報は、1ビットフラグであってもよい。例えば、ビットストリームを介して、現在ブロックの動き情報が隣接マージ候補から導出されたものであるかどうかを示すシンタックス要素 `isAdjacentMergeFlag` をシグナルリングすることができる。シンタックス要素 `isAdjacentMergeFlag` の値が1である場合、現在ブロックの動き情報は、隣接マージ候補に基づいて導出されてもよい。一方で、シンタックス要素 `isAdjacentMergeFlag` の値が0である場合、現在ブロックの動き情報は、非隣接マージ候補に基づいて導出されてもよい。

20

【0167】

表1は、シンタックス要素 `isAdjacentMergeFlag` を含むシンタックステーブルを示す。

30

40

50

【表 1】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	記述子
if(slice_type != I) {	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_luma_mpm_flag[x0][y0]	
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
} else {	
if (cu_skip_falg[x0][y0]) {	
if (MaxNumMergeCand > 1){	
isAdjacentMergeflag	ae(v)
if (isAdjcanetMergeflag){	
merge_idx[x0][y0]	ae(v)
} else{	
NA_merge_idx[x0][y0]	ae(v)
}	
}	
} else { /* MODE_INTER*/	
merge_flag[x0][y0]	ae(v)
if (merge_flag[x0][y0]){	
if (MaxNumMergeCand > 1){	
isAdjacentMergeflag	ae(v)
if (isAdjcanetMergeflag){	
merge_idx[x0][y0]	ae(v)
} else{	
NA_merge_idx[x0][y0]	ae(v)

【 0 1 6 8 】

10

20

30

40

50

}	
}	
}	
if(CuPredMode[x0][y0] != MODE_INTRA)	
cu_cbf	ae(v)
if(cu_cbf) {	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	

10

【0169】

ビットストリームを介して、複数のマージ候補のうちのいずれかを指定するための情報をシグナルリングすることができる。例えば、ビットストリームを介して、マージ候補リストに含まれるマージ候補のうちのいずれか1つのインデックスを示す情報をシグナルリングすることができる。

【0170】

`isAdjacentMergeFlag`が1である場合、信号を用いて、隣接マージ候補のうちのいずれか1つを決定するためのシンタックス要素`merge_idx`を送信することができる。シンタックス要素`merge_idx`の最大値を隣接マージ候補の数との差分値が1である値とすることができる。

20

【0171】

`isAdjacentMergeFlag`が0である場合、信号を用いて、非隣接マージ候補のうちのいずれか1つを決定するためのシンタックス要素`NA_merge_idx`を送信することができる。シンタックス要素`NA_merge_idx`は、非隣接マージ候補のインデックスと隣接マージ候補の数との差を求めることで得られた値を示す。デコーダは、`NA_merge_idx`に基づいてインデックスに隣接マージ候補の数を加算することで、非隣接マージ候補を選択することができる。

30

【0172】

マージ候補リストに含まれるマージ候補の数が閾値未満である場合、インター動き情報リストに含まれるマージ候補をマージ候補リストに追加することができる。ここで、閾値は、マージ候補リストに含まれる最大マージ候補数又は最大マージ候補数からオフセット値を減算した値であってもよい。オフセット値は、1又は2等の整数であってもよい。インター動き情報リストは、現在ブロックの前に符号化／復号されたブロックに基づいて導出されたマージ候補を含んでもよい。

【0173】

インター動き情報リストは、現在画像におけるインター予測に基づいて符号化／復号されたブロックから導出されたマージ候補を含む。例えば、インター動き情報リストに含まれるマージ候補の動き情報を、インター予測に基づいて符号化／復号されたブロックの動き情報に等しいように設定することができる。ここで、動き情報は、動きベクトル、参照画像インデックス、予測方向又は双方向加重値インデックスのうちの少なくとも1つを含んでもよい。解釈の便宜上、インター動き情報リストに含まれるマージ候補をインターマージ候補と呼ぶ。

40

【0174】

現在ブロックのマージ候補を選択する場合、前記マージ候補の動きベクトルは、初期動きベクトルと設定される。また、初期動きベクトルに対してオフセットベクトルを加算又は減算することで、現在ブロックの動き補償予測を実行することができる。マージ候補の動きベクトルに対してオフセットベクトルを加算又は減算することで新たな動きベクトル

50

を導出することは、マージ動き差分値符号化方法と定義されてもよい。

【0175】

ビットストリームを介して、マージオフセット符号化方法を用いるかどうかを示す情報をシグナルリングすることができる。前記情報は、1ビットのフラグ `merge__offset__vector__flag` であってもよい。例えば、`merge__offset__vector__flag` の値が1である場合、マージ動き差分値符号化方法を現在ブロックに適用することを示す。マージ動き差分値符号化方法を現在ブロックに適用する場合、マージ候補の動きベクトルに対してオフセットベクトルを加算又は減算することで、現在ブロックの動きベクトルを導出することができる。`merge__offset__vector__flag` の値が0である場合、マージ動き差分値符号化方法が現在ブロックに適用されていないことを示す。マージオフセット符号化方法が適用されていない場合、マージ候補の動きベクトルを現在ブロックの動きベクトルとすることができる。

10

【0176】

スキップモードを適用するかどうかを示すスキップフラグの値が真である場合又はマージモードを適用するかどうかを示すマージフラグの値が真である場合にのみ、シグナルリングされることが可能である。例えば、スキップモードを現在ブロックに適用するかどうかを示す `skip__flag` の値が1である場合、又は、マージモードを現在ブロックに適用するかどうかを示す `merge__flag` の値が1である場合、`merge__offset__vector__flag` に対して符号化を行い、シグナルリングすることができる。

20

【0177】

マージオフセット符号化方法を現在ブロックに適用すると決定した場合、別途、マージ候補リストに含まれるマージ候補のうちのいずれか1つを指定する情報、オフセットベクトルの大きさを示す情報、及びオフセットベクトルの方向を示す情報のうちの少なくとも1つをシグナルリングすることができる。

【0178】

ビットストリームを介して、マージ候補リストに含まれることが可能であるマージ候補の最大数を決定するための情報をシグナルリングすることができる。例えば、マージ候補リストに含まれることが可能であるマージ候補の最大数を6又はより小さい整数とすることができる。

30

【0179】

マージオフセット符号化方法を現在ブロックに適用すると決定した場合、所定の最大数のマージ候補のみを現在ブロックの初期動きベクトルとすることができる。つまり、マージオフセット符号化方法を適用するかどうかに基づいて、現在ブロックに利用可能なマージ候補の数を適応的に決定することができる。例えば、`merge__offset__vector__flag` の値を0とした場合、現在ブロックに利用可能なマージ候補の最大数をMとしてもよい。`merge__offset__vector__flag` の値を1とした場合、現在ブロックに利用可能なマージ候補の最大数をNとしてもよい。ここで、Mは、マージ候補リストに含まれるマージ候補の最大数を表し、また、Nは、M以下の整数を表す。

40

【0180】

例えば、Mが6であり、且つNが2である場合、マージ候補リストに含まれるマージ候補のうち最小インデックスを有する2つのマージ候補を、現在ブロックに利用可能であるものと設定することができる。従って、インデックス値が0であるマージ候補の動きベクトル又はインデックス値が1であるマージ候補の動きベクトルを現在ブロックの初期動きベクトルと設定することができる。MとNが同じである場合（例えば、M及びNが2である場合）、マージ候補リストに含まれる全てのマージ候補を、現在ブロックに利用可能であるものと設定することができる。

【0181】

代替的には、マージ動き差分値符号化方法を現在ブロックに適用するかどうかに基づい

50

て、隣接ブロックをマージ候補として利用できるかどうかを決定することができる。例えば、`merge_offset_vector_flag`の値が1である場合、現在ブロックの右上隅に隣接する隣接ブロック、左下隅に隣接する隣接ブロック、及び左下隅に隣接する隣接ブロックのうちの少なくとも1つを、マージ候補として利用不可能なものと設定することができる。従って、マージ動き差分値符号化方法を現在ブロックに適用する場合、現在ブロックの右上隅に隣接する隣接ブロック、左下隅に隣接する隣接ブロック、及び左下隅に隣接する隣接ブロックのうちの少なくとも1つの動きベクトルを、初期動きベクトルと設定することができない。代替的には、`merge_offset_vector_flag`の値が1である場合、現在ブロックの時間的隣接ブロックを、マージ候補として利用不可能であるものと設定することができる。

10

【0182】

マージ動き差分値符号化方法を現在ブロックに適用する場合、対をなすマージ候補及びゼロマージ候補のうちの少なくとも1つを使用しないように設定することができる。従って、`merge_offset_vector_flag`の値が1である場合、マージ候補リストに含まれるマージ候補の数が最大数未満であっても、対をなすマージ候補又はゼロマージ候補のうちの少なくとも1つをマージ候補リストに追加しない可能性もある。

【0183】

マージ候補の動きベクトルを、現在ブロックの初期動きベクトルと設定することができる。この場合、現在ブロックに利用可能なマージ候補の数が複数である場合、ビットストリームを介して、複数のマージ候補のうちのいずれか1つを指定する情報をシグナルリングすることができる。例えば、マージ候補リストに含まれるマージ候補の最大数が1より大きい場合、ビットストリームを介して、複数のマージ候補のうちのいずれか1つを示す情報`merge_idx`をシグナルリングすることができる。つまり、マージオフセット符号化方法で、複数のマージ候補のうちのいずれか1つを指定するための情報`merge_idx`により、マージ候補を指定することができる。現在ブロックの初期動きベクトルは、`merge_idx`で示されるマージ候補の動きベクトルと設定されてもよい。

20

【0184】

一方で、現在ブロックに利用可能なマージ候補の数が1である場合、マージ候補を指定するための情報をシグナルリングすることを省略してもよい。例えば、マージ候補リストに含まれることが可能であるマージ候補の最大数が1以下である場合、マージ候補を指定するための情報`merge_idx`をシグナルリングすることを省略してもよい。つまり、マージオフセット符号化方法で、マージ候補リストに1つのマージ候補が含まれる時、マージ候補を指定するための情報`merge_idx`の符号化を省略し、また、マージ候補リストに含まれるマージ候補に基づいて、初期動きベクトルを決定することができる。マージ候補の動きベクトルは、現在ブロックの初期動きベクトルと設定されてもよい。

30

【0185】

もう1つの例として、現在ブロックのマージ候補を決定した後、マージ動き差分値符号化方法を現在ブロックに適用するかどうかを決定することができる。例えば、マージ候補リストに含まれることが可能であるマージ候補の最大数が1より大きい場合、マージ候補のうちのいずれか1つを指定するための情報`merge_idx`をシグナルリングすることができる。`merge_idx`に基づいて、マージ候補を選択した後、マージ動き差分値符号化方法を現在ブロックに適用するかどうかを示す`merge_offset_vector_flag`に対して復号を行うことができる。表2は、上記実施例によるシンタックステーブルを示す図である。

40

【表 2】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	記述子
if(slice_type != I) {	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_luma_mpm_flag[x0][y0]	
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
} else { /* MODE_INTER */	
if(cu_skip_flag[x0][y0]) {	
if(merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx[x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if(merge_idx < 2 && merge_offset_vector_flag) {	
distance_idx[x0][y0]	ae(v)
direction_idx[x0][y0]	ae(v)
}	
}	
} else {	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	

10

20

30

40

【 0 1 8 6 】

50

<code>merge_idx[x0][y0]</code>	<code>ae(v)</code>
<code>merge_offset_vector_flag</code>	<code>ae(v)</code>
<code>if (merge_idx < 2 && merge_offset_vector_flag) {</code>	
<code>distance_idx[x0][y0]</code>	<code>ae(v)</code>
<code>direction_idx[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>}</code>	
<code>} else {</code>	
<code>if(slice_type == B)</code>	
<code>inter_pred_idc[x0][y0]</code>	<code>ae(v)</code>
<code>if(sps_affine_enabled_flag && cbWidth >= 16</code> <code>&& cbHeight >= 16) {</code>	
<code>inter_affine_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(sps_affine_type_flag &&</code> <code>inter_affine_flag[x0][y0])</code>	
<code>cu_affine_type_flag[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>}</code>	

10

20

【0187】

もう1つの例として、現在ブロックのマージ候補を決定した後、決定されたマージ候補インデックスが、マージ動き差分値符号化方法を適用する場合に利用可能なマージ候補の最大数より小さい場合のみに、マージ動き差分値符号化方法を現在ブロックに適用するかどうかを決定することができる。例えば、インデックス情報`merge_idx`の値がN未満である場合のみに、現在ブロックに対してマージ動き差分値符号化方法を適用するかどうかを示す`merge_offset_vector_flag`に対して符号化を行い、シグナルリングすることができる。インデックス情報`merge_idx`の値がN以上である場合、`merge_offset_vector_flag`に対する符号化を省略することができる。`merge_offset_vector_flag`に対する符号化を省略すると、マージ動き差分値符号化方法が現在ブロックに適用されていないと決定することができる。

30

【0188】

代替的には、現在ブロックのマージ候補を決定した後、決定されたマージ候補が双方向動き情報を有するかそれとも一方向動き情報を有するかを考慮して、マージ動き差分値符号化方法を現在ブロックに適用するかどうかを決定することができる。例えば、インデックス情報`merge_idx`の値がN未満であって、且つインデックス情報に基づいて選択されたマージ候補が双方向動き情報を有する場合のみに、マージ動き差分値符号化方法を現在ブロックに適用するかどうかを示す`merge_offset_vector_flag`に対して符号化を行い、シグナルリングすることができる。任意選択的に、インデックス情報`merge_idx`の値がN未満であって、インデックス情報に基づいて選択されたマージ候補が一方向動き情報を有する場合のみに、マージ動き差分値符号化方法を現在ブロックに適用するかどうかを示す`merge_offset_vector_flag`に対して符号化を行い、シグナルリングする。

40

50

【0189】

代替的には、現在ブロックの大きさ、現在ブロックの形状、及び現在ブロックが符号化ツリーユニットの境界に接触するかどうかのうちの少なくとも1つに基づいて、マージ動き差分値符号化方法を適用するかどうかを決定することができる。現在ブロックの大きさ、現在ブロックの形状、及び現在ブロックが符号化ツリーユニットの境界に接触するかどうかのうちの少なくとも1つが所定の条件を満たさない場合、マージ動き差分値符号化方法を現在ブロックに適用するかどうかを示すmerge_offset_vector_flagに対する符号化を省略してもよい。

【0190】

マージ候補を選択する場合、マージ候補の動きベクトルを現在ブロックの初期動きベクトルとすることができる。続いて、オフセットベクトルの大きさを示す情報及びオフセットベクトルの方向を示す情報を復号し、オフセットベクトルを決定することができる。オフセットベクトルは、水平方向成分又は垂直方向成分を有することができる。

【0191】

オフセットベクトルの大きさを示す情報は、動き大きさ候補のうちのいずれか1つを示すインデックス情報であってもよい。例えば、ビットストリームを介して、動き大きさ候補のうちのいずれか1つを示すインデックス情報distance_idxをシグナルリングすることができる。表3は、インデックス情報distance_idxの二値化、distance_idxに基づいてオフセットベクトルの大きさを決定するための変数DistFromMergeMVの値を示す。

【表3】

distance_idx[x][y]	二値化	DistFromMergeMV[x0][y0]
0	0	1
1	10	2
2	110	4
3	1110	8
4	11110	16
5	111110	32
6	1111110	64
7	1111111	128

【0192】

変数DistFromMergeMVを所定値で除算することで、オフセットベクトルの大きさを導出することができる。式4は、オフセットベクトルの大きさを決定する例を示す。

【数4】

【式4】

$$abs(offsetMV) = DistFromMergeMV \ll 2$$

【0193】

式4によれば、変数DistFromMergeMVを4で除算することで得られた値又は変数DistFromMergeMVを左へ2だけシフトすることで得られた値をオフ

セットベクトルの大きさとしてすることができる。

【0194】

表3に示す例よりも多い動き大きさ候補又はより少ない動き大きさ候補を用いることができ、又は、動きベクトルオフセット大きさ候補の範囲を、表3に示す例と異なるように設定することができる。例えば、オフセットベクトルの水平方向成分又は垂直方向成分の大きさを、2個のサンプル距離以下とすることができる。表4は、インデックス情報 `distance_idx` の二値化、`distance_idx` に基づいてオフセットベクトルの大きさを決定するための変数 `DistFromMergeMV` の値を示す。

【表4】

<code>distance_idx[x][y]</code>	二値化	<code>DistFromMergeMV[x0][y0]</code>
0	0	1
1	10	2
2	110	4
3	111	8

10

【0195】

代替的には、動きベクトル精度に基づいて、動きベクトルオフセット大きさ候補の範囲を、異なるように設定することができる。例えば、現在ブロックの動きベクトル精度が小数画素 (`fractional-pel`) である場合、インデックス情報 `distance_idx` の値に対応する変数 `DistFromMergeMV` の値を1、2、4、8、16などとすることができる。ここで、小数画素は、 $1/16$ 画素、 $1/8$ 画素 (`octo-pel`)、 $1/4$ 画素 (`quarter-pel`) 又は $1/2$ 画素 (`half-pel`) のうちの少なくとも1つを含む。一方で、現在ブロックの動きベクトル精度が整数画素である場合、インデックス情報 `distance_idx` の値に対応する変数 `DistFromMergeMV` の値を4、8、16、32、64などとすることができる。つまり、現在ブロックの動きベクトル精度に基づいて、変数 `DistFromMergeMV` を決定するための表を、異なるように設定することができる。

20

【0196】

例えば、現在ブロック又はマージ候補の動きベクトル精度が $1/4$ 画素である場合、表3を用いて、`distance_idx` で表される変数 `DistFromMergeMV` を導出することができる。一方で、現在ブロック又はマージ候補の動きベクトル精度が整数画素である場合、表3における `distance_idx` で示される変数 `DistFromMergeMV` の値をN倍 (例えば、4倍) にすることで導出された値を変数 `DistFromMergeMV` の値とすることができる。

30

【0197】

ビットストリームを介して、動きベクトル精度を決定するための情報をシグナルリングすることができる。例えば、シーケンスレベル、画像レベル、スライス (`slice`) レベル又はブロックレベルで、信号により情報を送信することができる。従って、ビットストリームを介して、シグナルリングされた、動きベクトル精度に関わる情報に基づいて、動き大きさ候補の範囲を、異なるように設定することができる。代替的には、現在ブロックのマージ候補に基づいて、動きベクトル精度を決定することができる。例えば、現在ブロックの動きベクトル精度を、マージ候補の動きベクトル精度と同じであるように設定することができる。

40

【0198】

代替的には、ビットストリームを介して、オフセットベクトルの検索範囲を決定するための情報をシグナルリングすることができる。検索範囲に基づいて、動き大きさ候補の数

50

、動き大きさ候補の最小値及び動き大きさ候補の最大値のうちの少なくとも1つを決定することができる。例えば、ビットストリームを介して、オフセットベクトルの検索範囲を決定するためのフラグ `merge_offset_vector_flag` をシグナルリングすることができる。前記情報は、シーケンスヘッダ、画像ヘッダ又はスライスヘッダを介して、シグナルリングされてもよい。

【0199】

例えば、`merge_offset_extend_range_flag` の値が0である場合、オフセットベクトルの大きさを2以下とすることができる。従って、`DistFromMergeMV` の最大値を8とすることができる。一方で、`merge_offset_extend_range_flag` の値が1である場合、オフセットベクトルの大きさを、32個のサンプル距離とすることができる。従って、`DistFromMergeMV` の最大値を128とすることができる。

【0200】

オフセットベクトルの大きさが閾値より大きいかどうかを示すフラグを用いて、オフセットベクトルの大きさを決定することができる。例えば、ビットストリームを介して、オフセットベクトルの大きさが閾値より大きいかどうかを示すフラグ `distance_flag` をシグナルリングすることができる。閾値は、1、2、4、8又は16であってもよい。例えば、`distance_flag` が1である場合、オフセットベクトルの大きさが4より大きいことを示す。一方で、`distance_flag` が0である場合、オフセットベクトルの大きさが4以下であることを示す。

【0201】

オフセットベクトルの大きさが閾値より大きい場合、インデックス情報 `distance_idx` を用いて、オフセットベクトルの大きさと閾値との差分値を導出することができる。代替的には、オフセットベクトルの大きさが閾値以下である場合、インデックス情報 `distance_idx` を用いて、オフセットベクトルの大きさを決定することができる。表5は、`distance_flag` 及び `distance_idx` を符号化するプロセスのシンタックステーブルを示す。

10

20

30

40

50

【表 5】

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	記述子
if(slice_type != I) {	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_luma_mpm_flag[x0][y0]	
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
} else { /* MODE_INTER */	
if(cu_skip_flag[x0][y0]) {	
if(merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	
merge_idx[x0][y0]	ae(v)
merge_offset_vector_flag	ae(v)
if(merge_idx < 2 && merge_offset_vector_flag) {	
distance_flag[x0][y0]	ae(v)
distance_idx[x0][y0]	
direction_idx[x0][y0]	ae(v)
}	
}	
}	
} else {	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(merge_affine_flag[x0][y0] == 0 && MaxNumMergeCand > 1) {	

10

20

30

40

【 0 2 0 2 】

50

<code>merge_idx[x0][y0]</code>	<code>ae(v)</code>
<code>merge_offset_vector_flag</code>	<code>ae(v)</code>
<code>if (merge_idx < 2 && merge_offset_vector_flag) {</code>	
<code>distance_flag[x0][y0]</code>	<code>ae(v)</code>
<code>distance_idx[x0][y0]</code>	<code>ae(v)</code>
<code>direction_idx[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>}</code>	
<code>} else {</code>	
<code>if(slice_type == B)</code>	
<code>inter_pred_idc[x0][y0]</code>	<code>ae(v)</code>
<code>if(sps_affine_enabled_flag && cbWidth >= 16</code> <code>&& cbHeight >= 16) {</code>	
<code>inter_affine_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(sps_affine_type_flag &&</code> <code>inter_affine_flag[x0][y0])</code>	
<code>cu_affine_type_flag[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>}</code>	

10

20

【0203】

式5は、`distance_flag`及び`distance_idx`を用いて、オフセットベクトルの大きさを決定するための変数`DistFromMergeMV`を導出する例を示す。

30

【数5】

【式5】

$$DistFromMergeMV = N * distance_flag + (1 \ll distance_idx)$$

【0204】

式5において、`distance_flag`の値を1又は0とすることができる。`distance_idx`の値を1、2、4、8、16、32、64、128等とすることができる。Nは、閾値により決定された係数を表す。例えば、閾値が4である場合、Nを16とすることができる。

40

【0205】

オフセットベクトルの方向を示す情報は、ベクトル方向候補のうちのいずれか1つを示すインデックス情報であってもよい。例えば、ビットストリームを介して、ベクトル方向候補のうちのいずれか1つを示すインデックス情報`direction_idx`をシグナルリングすることができる。表6は、インデックス情報`direction_idx`の二値化、`direction_idx`によるオフセットベクトルの方向を示す。

50

【表 6】

direction_idx[x][y]	二値化	sign[x][y][0]	sign[x][y][1]
0	00	+1	0
1	01	-1	0
2	10	0	+1
3	11	0	-1

10

【0206】

表 6 において、 $sign[0]$ は、水平方向を示し、また、 $sign[1]$ は、垂直方向を示す。 $+1$ は、オフセットベクトルの x 成分又は y 成分の値が正数（+）であることを示し、また、 -1 は、オフセットベクトルの x 成分又は y 成分の値が負数（-）であることを示す。式 6 は、オフセットベクトルの大きさ及び方向に基づいて、オフセットベクトルを決定する例を示す。

【数 6】

【式 6】

20

$$\begin{aligned} offsetMV[0] &= abs(offsetMV) * sign[0] \\ offsetMV[1] &= abs(offsetMV) * sign[1] \end{aligned}$$

【0207】

式 6 において、 $offsetMV[0]$ は、オフセットベクトルの垂直方向成分を示し、 $offsetMV[1]$ は、オフセットベクトルの水平方向ベクトルを示す。

30

【0208】

図 17 は、オフセットベクトルの大きさを示す $distance_idx$ 及びオフセットベクトルの方向を示す $direction_idx$ の値に基づいたオフセットベクトルを示す図である。

【0209】

図 7 に示す例において、 $distance_idx$ 及び $direction_idx$ の値に基づいて、オフセットベクトルの大きさ及び方向を決定することができる。オフセットベクトルの最大大きさを、閾値以下に設定することができる。ここで、閾値は、エンコーダ及びデコーダにより事前定義された値を有してもよい。例えば、閾値は、32 個のサンプル距離であってもよい。代替的には、初期動きベクトルの大きさに基づいて閾値を決定することができる。例えば、初期動きベクトルの水平成分の大きさに基づいて、水平方向の閾値を設定することができ、初期動きベクトルの垂直成分に基づいて、垂直方向の閾値を設定することができる。

40

【0210】

マージ候補が双方向動き情報を有する場合、マージ候補の $L0$ 動きベクトルを現在ブロックの $L0$ 初期動きベクトルとすることができ、また、マージ候補の $L1$ 動きベクトルを現在ブロックの $L1$ 初期動きベクトルとすることができる。この場合、マージ候補の $L0$ 参照画像と現在画像との出力順差分値（以下、 $L0$ 差分値と呼ぶ）及びマージ候補の $L1$ 参照画像と現在画像との出力順差分値（以下、 $L1$ 差分値と呼ぶ）を考慮することで、 $L0$ オフセットベクトル及び $L1$ オフセットベクトルを決定することができる。

50

【0211】

まず、L0 差分値と L1 差分値のシンボルが同じである場合、L0 オフセットベクトルと L1 オフセットベクトルを、同じであるように設定することができる。一方で、L0 差分値と L1 差分値のシンボルが異なる場合、L0 オフセットベクトルと逆である方向で、L1 オフセットベクトルを設定することができる。

【0212】

L0 オフセットベクトルの大きさと L1 オフセットベクトルの大きさを同じであるように設定することができる。代替的には、L0 差分値及び L1 差分値に基づいて L0 オフセットベクトルを縮小することで、L1 オフセットベクトルの大きさを決定することができる。

10

【0213】

例えば、式 7 は、L0 差分値と L1 差分値のシンボルが同じである場合の L0 オフセットベクトル及び L1 オフセットベクトルを示す。

【数 7】

【式 7】

$$\begin{aligned} offsetMVL0[0] &= abs(offsetMV) * sign[0] \\ offsetMVL0[1] &= abs(offsetMV) * sign[1] \\ offsetMVL1[0] &= abs(offsetMV) * sign[0] \\ offsetMVL1[1] &= abs(offsetMV) * sign[1] \end{aligned}$$

20

【0214】

式 7 において、offsetMVL0[0] は、L0 オフセットベクトルの水平方向成分を示し、offsetMVL0[1] は、L0 オフセットベクトルの垂直方向成分を示す。offsetMVL1[0] は、L1 オフセットベクトルの水平方向成分を示し、offsetMVL1[1] は、L1 オフセットベクトルの垂直方向成分を示す。

30

【0215】

式 8 は、L0 差分値と L1 差分値のシンボルが異なる場合の L0 オフセットベクトル及び L1 オフセットベクトルを示す。

【数 8】

【式 8】

$$\begin{aligned} offsetMVL0[0] &= abs(offsetMV) * sign[0] \\ offsetMVL0[1] &= abs(offsetMV) * sign[1] \\ offsetMVL1[0] &= -1 * abs(offsetMV) * sign[0] \\ offsetMVL1[1] &= -1 * abs(offsetMV) * sign[1] \end{aligned}$$

40

【0216】

4 つ以上のベクトル方向候補を定義することもできる。表 7 及び表 8 は、8 個のベクトル方向候補を定義する例を示す。

50

【表 7】

direction_idx[x][y]	二値化	sign[x][y][0]	sign[x][y][1]
0	000	+1	0
1	001	-1	0
2	010	0	+1
3	011	0	-1
4	100	+1	+1
5	101	+1	-1
6	110	-1	+1
7	111	-1	-1

10

【表 8】

direction_idx[x][y]	二値化	sign[x][y][0]	sign[x][y][1]
0	000	+1	0
1	001	-1	0
2	010	0	+1
3	011	0	-1
4	100	+1/2	+1/2
5	101	+1/2	-1/2
6	110	-1/2	+1/2
7	111	-1/2	-1/2

20

30

【0217】

表 7 及び表 8 において、 $sign[0]$ 及び $sign[1]$ の絶対値が 0 より大きい場合、オフセットベクトルが対角線方向にあることを示す。表 6 を用いる場合、対角線オフセットベクトルの x 軸成分及び y 軸成分の大きさを $abs(offsetMV)$ とし、表 7 を用いる場合、対角線オフセットベクトルの x 軸成分及び y 軸成分の大きさを $abs(offsetMV/2)$ とする。

【0218】

図 18 は、オフセットベクトルの大きさを示す $distance_idx$ 及びオフセットベクトルの方向を示す $direction_idx$ の値に基づいたオフセットベクトルを示す図である。

40

【0219】

図 18 (a) は、表 6 を適用する場合の例であり、図 18 (b) は、表 7 を適用する場合の例である。

【0220】

ビットストリームを介して、ベクトル方向候補の数又は大きさのうちの少なくとも 1 つを決定するための情報をシグナルリングすることができる。例えば、ビットストリームを介して、ベクトル方向候補を決定するためのフラグ $merge_offset_direction_range_flag$ をシグナルリングすることができる。シーケンスレベ

50

ル、画像レベル又はスライスレベルで、信号によりフラグを送信することができる。例えば、フラグの値が0である場合、表6に示す4つのベクトル方向候補を用いることができる。一方で、フラグの値が1である場合、表7又は表8に示す8個のベクトル方向候補を用いることができる。

【0221】

代替的には、オフセットベクトルの大きさに基づいて、ベクトル方向候補の数又は大きさのうちの少なくとも1つを決定することができる。例えば、オフセットベクトルの大きさを決定するための変数 $DistFromMergeMV$ の値が閾値以下である場合、表7又は表8に示す8個のベクトル方向候補を用いることができる。一方で、変数 $DistFromMergeMV$ の値が閾値より大きい場合、表6に示す4個のベクトル方向候補を用いることができる。

10

【0222】

代替的には、初期動きベクトルの x 成分の値 MV_x 及び y 成分の値 MV_y に基づいて、ベクトル方向候補の数又は大きさのうちの少なくとも1つを決定することができる。例えば、 MV_x と MV_y との差又は差の絶対値が閾値以下である場合、表7又は表8に示す8個のベクトル方向候補を用いることができる。一方で、 MV_x と MV_y との差又は差の絶対値が閾値より大きい場合、表6に示す4個のベクトル方向候補を用いることができる。

【0223】

オフセットベクトルに初期動きベクトルを加算することで、現在ブロックの動きベクトルを導出することができる。式9は、現在ブロックの動きベクトルを決定する例を示す。

20

【数9】

【式9】

$$\begin{aligned} mvL0[0] &= mergeMVL0[0] + offsetMVL0[0] \\ mvL0[1] &= mergeMVL0[1] + offsetMVL0[1] \\ mvL1[0] &= mergeMVL1[0] + offsetMVL1[0] \\ mvL1[1] &= mergeMVL1[1] + offsetMVL1[1] \end{aligned}$$

30

【0224】

式9において、 $mvL0$ は、現在ブロックの $L0$ 動きベクトルを示し、 $mvL1$ は、現在ブロックの $L1$ 動きベクトルを示す。 $mergeMVL0$ は、現在ブロックの $L0$ 初期動きベクトル（即ち、マージ候補の $L0$ 動きベクトル）を示し、 $mergeMVL1$ は、現在ブロックの $L1$ 初期動きベクトルを示す。 $[0]$ は、動きベクトルの水平方向成分を示し、 $[1]$ は、動きベクトルの垂直方向成分を示す。

【0225】

40

イントラ予測は、現在ブロックの周辺の符号化／復号された再構成サンプルを用いて現在ブロックを予測することである。この場合、現在ブロックのイントラ予測は、インループフィルタを適用する前の再構成サンプルを用いることができる。

【0226】

イントラ予測技術は、マトリックス ($Matrix$) に基づくイントラ予測及び周辺の再構成サンプルとの方向性を考慮した一般的なイントラ予測を含む。ビットストリームを介して、現在ブロックのイントラ予測技術を示す情報をシグナルリングすることができる。前記情報は、1ビットフラグであってもよい。代替的には、現在ブロックの位置、大きさ、形状又は隣接ブロックのイントラ予測技術のうちの少なくとも1つに基づいて、現在ブロックのイントラ予測技術を決定することができる。例えば、現在ブロックが画像の境

50

界に跨って存在する場合、現在ブロックを、マトリックスに基づくイントラ予測を適用しないように設定する。

【0227】

マトリックスに基づくイントラ予測は、エンコーダ及びデコーダに記憶されたマトリックスと現在ブロックの周辺の再構成サンプルとのマトリックス乗算に基づいて現在ブロックの予測ブロックを得る方法である。ビットストリームを介して、記憶された複数のマトリックスのうちのいずれか1つを示すための情報をシグナルリングすることができる。デコーダは、前記情報及び現在ブロックの大きさに基づいて、現在ブロックのイントラ予測に用いられるマトリックスを決定することができる。

【0228】

一般的なイントラ予測は、非角度イントラ予測モード又は角度イントラ予測モードに基づいて現在ブロックに関わる予測ブロックを得る方法である。

【0229】

オリジナル画像と予測画像を減算することで、導出される残差画像を導出することができる。この場合、残差画像を周波数領域に変更する場合、周波数成分のうちの高周波数成分を除去しても、ビデオの主観画質を大幅に低下させることがない。従って、高周波数成分の値を小さくするか又は高周波数成分の値を0とすると、明らかな視覚的歪みを引き起こすことなく、圧縮効率を向上させるという効果を有する。上記特性を反映するために、現在ブロックを変換することで残差画像を2次元周波数成分に分解することができる。離散コサイン変換(DCT: Discrete Cosine Transform)又は離散サイン変換(DST: Discrete Sine Transform)等の変換技術を用いて前記変換を実行することができる。

【0230】

DCT又はDSTを用いて現在ブロックを変換した後、変換された現在ブロックに対して再び変換を行うことができる。この場合、DCT又はDSTに基づく変換を第1変換と定義し、第1変換を適用したブロックに対して再び変換を行うプロセスを第2変換と呼ぶ。

【0231】

第1変換は、複数の変換カーネル候補のうちのいずれか1つを用いて実行することができる。例えば、DCT2、DCT8又はDCT7のうちのいずれか1つを用いて第1変換を実行することができる。

【0232】

水平方向及び垂直方向に対して、異なる変換カーネルを用いることもできる。ビットストリームを介して、水平方向の変換カーネルと垂直方向の変換カーネルの組み合わせを表す情報をシグナルリングすることもできる。

【0233】

第1変換及び第2変換の実行ユニットは、異なる。例えば、8×8ブロックに対して第1変換を実行し、また、変換された8×8ブロックにおける大きさが4×4であるサブブロックに対して第2変換を実行することができる。この場合、第2変換を実行しない余剰領域の変換係数を0とすることもできる。

【0234】

代替的には、4×4ブロックに対して第1変換を実行し、また、変換された4×4ブロックを含む大きさが8×8である領域に対して第2変換を実行することができる。

【0235】

ビットストリームを介して、第2変換を実行するかどうかを表す情報をシグナルリングすることができる。

【0236】

デコーダにおいて、第2変換の逆変換(第2逆変換)を実行し、また、その結果に対して第1変換の逆変換(第1逆変換)を実行することができる。第2逆変換及び第1逆変換の実行結果として、現在ブロックの残差信号を得ることができる。

【0237】

量子化は、ブロックのエネルギーの減少に用いられ、また、量子化プロセスは、変換係数を特定の定数で除算するプロセスを含む。前記定数は、量子化パラメータにより導出されてもよく、また、量子化パラメータは、1から63の値と定義されてもよい。

【0238】

エンコーダにおいて変換及び量子化を実行すると、デコーダは、逆量子化及び逆変換により、残差ブロックを得ることができる。デコーダにおいて、予測ブロックと残差ブロックを加算することで、現在ブロックの再構成ブロックを得ることができる。

【0239】

現在ブロックの再構成ブロックを得ると、インループフィルタリング (In-loop filtering) により、量子化及び符号化プロセスにおいて発生した情報紛失を減少させることができる。インループフィルタは、デブロッキングフィルタ (Deblocking filter)、サンプル適応型オフセットフィルタ (Sample Adaptive Offset filter: SAO) 又は適応型ループフィルタ (Adaptive Loop Filter: ALF) のうちの少なくとも1つを含んでもよい。

【0240】

復号プロセス又は符号化プロセスに重点を置いて説明する実施例を符号化プロセス又は復号プロセスに用いるものも本発明の範囲内に含まれる。説明された順番と異なる順番で、所定の順番で説明された複数の実施例を変更したものも本発明の範囲内に含まれる。

【0241】

一連のステップ又はフローチャートに基づいて実施例を説明したが、これは、発明の時間的順番を限定しない。また、必要に応じて、同時に実行してもよい又は他の順番に応じて実行してもよい。なお、上記実施例において、ブロック図を構成する構成要素 (例えば、ユニット、モジュール等) は、それぞれ、ハードウェア機器又はソフトウェアとして実現されてもよい。また、複数の構成要素を組み合わせる単一のハードウェア機器又はソフトウェアとして実行されてもよい。前記実施例は、プログラム命令の形態で実行してもよい。前記プログラム命令は、種々のコンピュータ部材により実行されてコンピュータ可読記憶媒体に記録されてもよい。前記コンピュータ可読記憶媒体は、プログラム命令、データファイル、データ構造などを単独で含んでもよい又はその組み合わせを含んでもよい。コンピュータ可読記憶媒体の例として、ハードディスク、フレキシブルディスク及び磁気テープなどの磁気媒体、CD-ROM、DVD等の光学記録媒体、フロッピーディスク (floptical disk) のような磁気-光学媒体 (magneto-optical media) 及びROM、RAM、フラッシュメモリ等のような、プログラム命令を記憶して該命令を実行する方式で特に配置されたハードウェア機器を含んでもよい。前記ハードウェア機器は、1つ又は複数のソフトウェアモジュールとして操作を行い、本発明による処理を実行するように構成されてもよく、その逆も同様である。

【産業上の利用可能性】

【0242】

本発明は、ビデオに対して符号化／復号を行う電子機器に適用可能である。

10

20

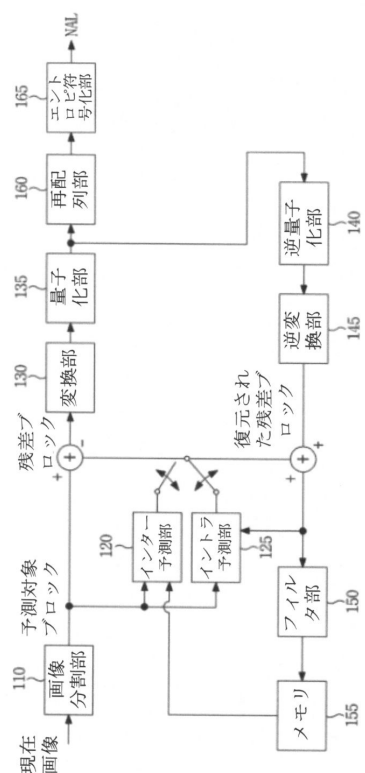
30

40

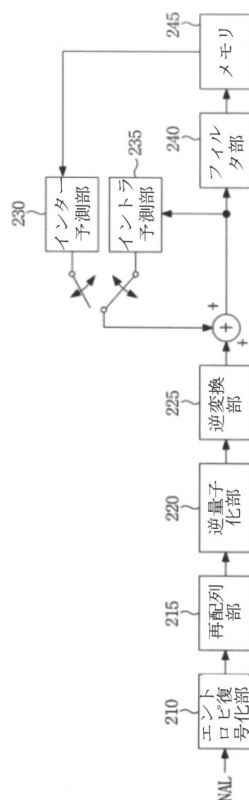
50

【図面】

【図 1】

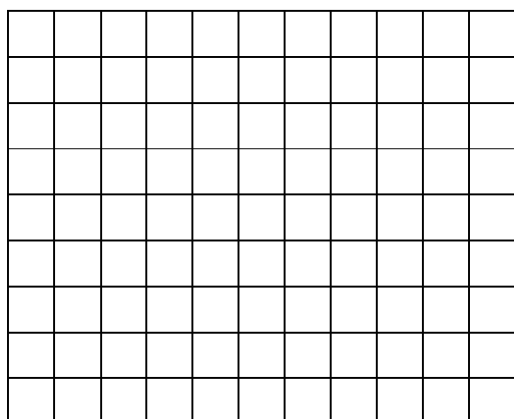


【図 2】



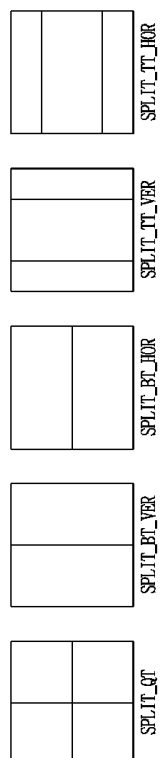
【図 3】

[도3]



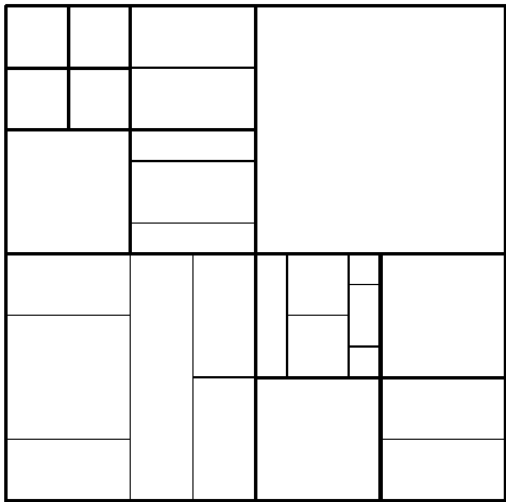
【図 4】

[54]

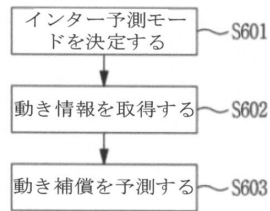


【図 5】

[図5]

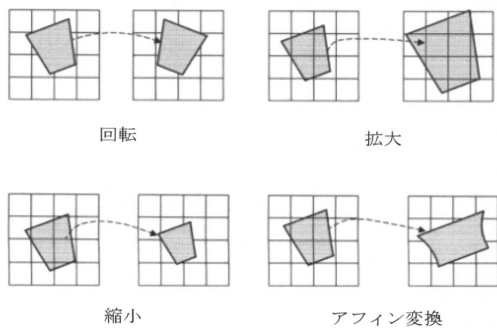


【図 6】

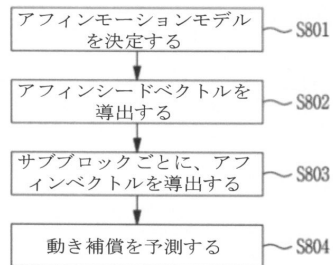


10

【図 7】



【図 8】



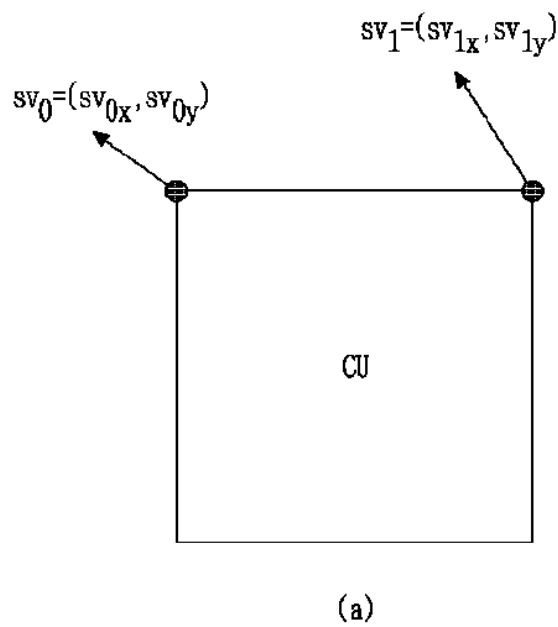
20

30

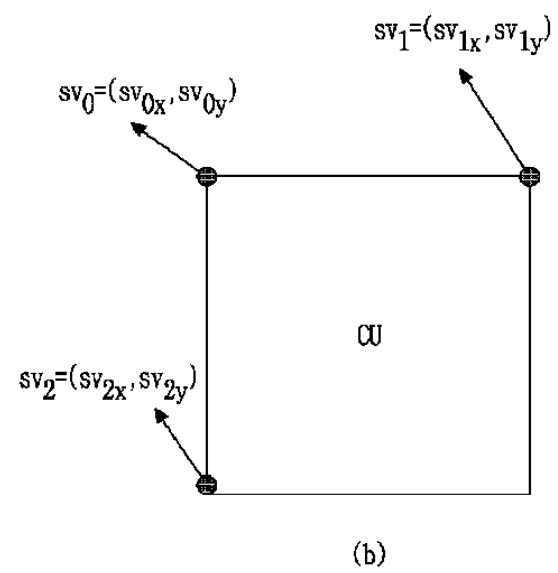
40

50

【図 9 (a) 】



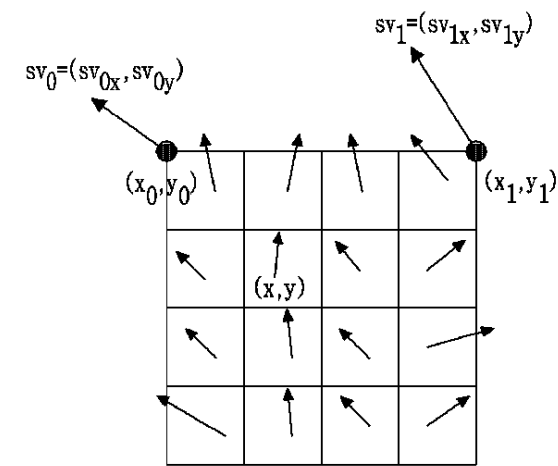
【図 9 (b) 】



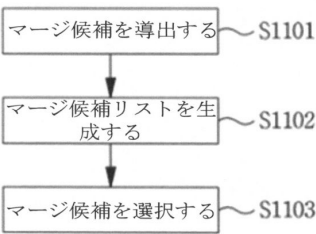
10

【図 1 0】

[図 10]



【図 1 1】



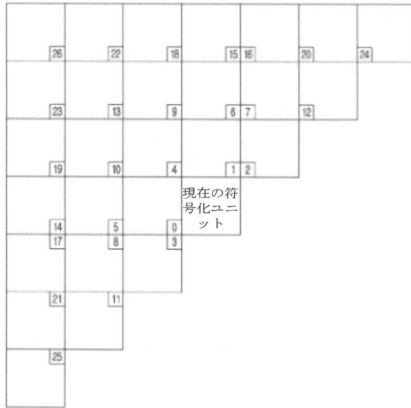
20

30

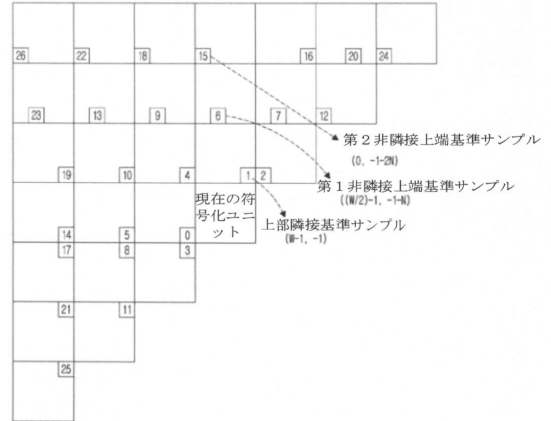
40

50

【図 1 2】

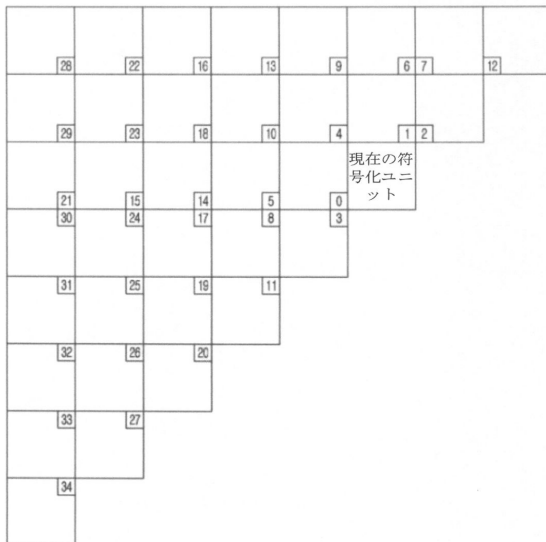


【図 1 3】

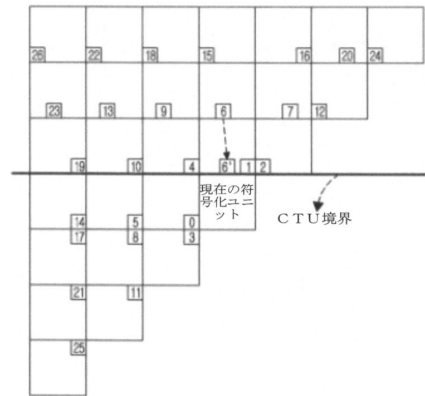


10

【図 1 4】



【図 1 5 a】



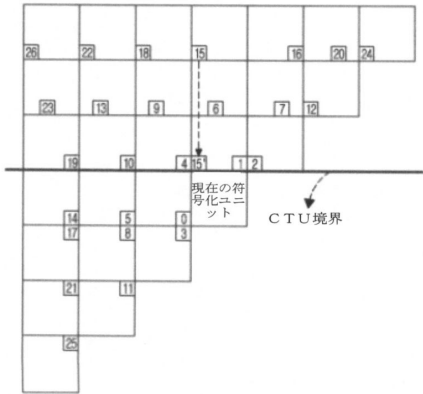
20

30

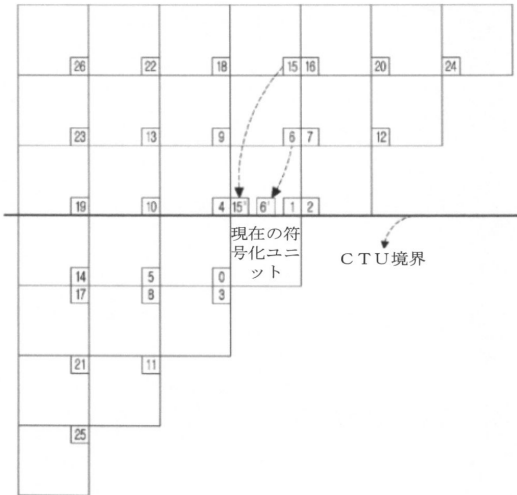
40

50

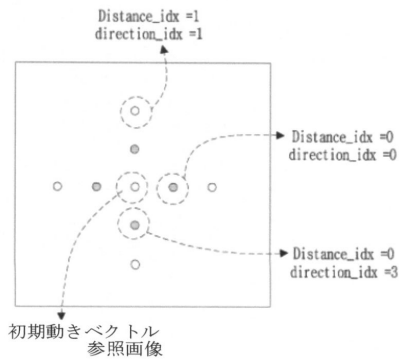
【図 15 b】



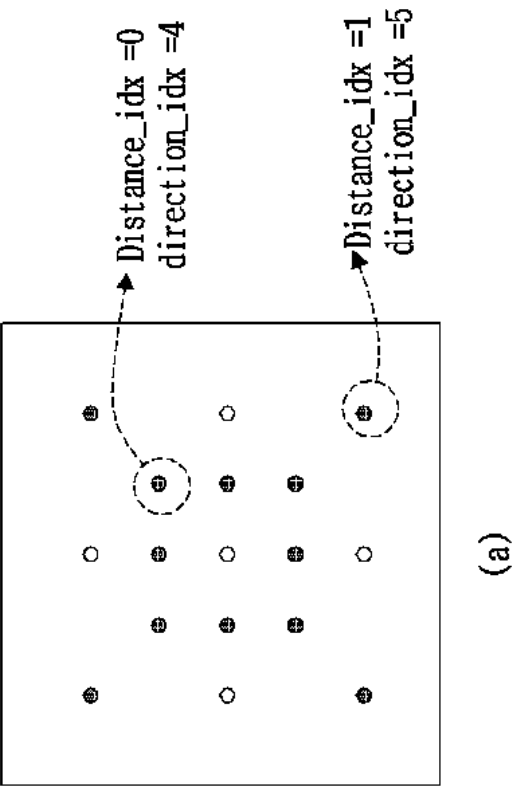
【図 16】



【図 17】



【図 18 (a)】



10

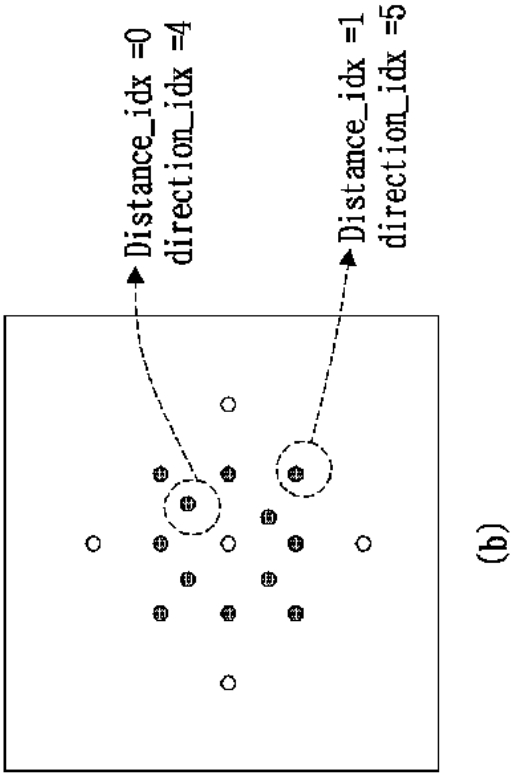
20

30

40

50

【図 18 (b)】



10

20

30

40

50

フロントページの続き

- (74)代理人 100107582
弁理士 関根 毅
- (74)代理人 100152205
弁理士 吉田 昌司
- (74)代理人 100137523
弁理士 出口 智也
- (72)発明者 リー、ベグン
大韓民国キョンギード、ソンナムーシ、ブンダンーグ、パンギョーロ、393、214-1202
- 審査官 岩井 健二
- (56)参考文献 国際公開第2020/017367 (WO, A1)
Seungsoo Jeong, et al., Proposed WD for CE4 Ultimate motion vector expression (Test 4.5.4), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L0054_WD, 12th Meeting: Macao, CN, 2018年10月, pp.1-12
Tomonori Hashimoto, Eiichi Sasaki, and Tomohiro Ikai, Non-CE4: Enhanced ultimate motion vector expression, Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L0355, 12th Meeting: Macau, CH, 2018年10月, pp.1-5
Li Jingya, Ru-Ling Liao, and Chong Soon Lim, CE4-related: Improvement on ultimate motion vector expression, Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L0408-v2, 12th Meeting: Macao, CN, 2018年10月, pp.1-4
Kai Zhang, et al., Non-CE4: Harmonization of MMVD and AMVR, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-M0267-v2, 13th Meeting: Marrakech, MA, 2019年01月, pp.1-2
- (58)調査した分野 (Int.Cl., DB名)
H04N 19/00 - 19/98