

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7571105号
(P7571105)

(45)発行日 令和6年10月22日(2024.10.22)

(24)登録日 令和6年10月11日(2024.10.11)

(51)国際特許分類		F I	
H O 4 N	19/433 (2014.01)	H O 4 N	19/433
H O 4 N	19/563 (2014.01)	H O 4 N	19/563
H O 4 N	19/577 (2014.01)	H O 4 N	19/577
H O 4 N	19/59 (2014.01)	H O 4 N	19/59
請求項の数 18 外国語出願 (全45頁)			
(21)出願番号	特願2022-176532(P2022-176532)	(73)特許権者	504161984
(22)出願日	令和4年11月2日(2022.11.2)		ホアウェイ・テクノロジーズ・カンパニ
(62)分割の表示	特願2020-572766(P2020-572766)		ー・リミテッド
原出願日	令和1年6月28日(2019.6.28)		中華人民共和国・518129・グァン
(65)公開番号	特開2023-14095(P2023-14095A)		ドン・シェンツェン・ロンガン・ディス
(43)公開日	令和5年1月26日(2023.1.26)		トリクト・バンティアン・(番地なし)
審査請求日	令和4年12月1日(2022.12.1)		・ホアウェイ・アドミニストレーション
(31)優先権主張番号	62/691,594	(74)代理人	・ビルディング
(32)優先日	平成30年6月28日(2018.6.28)		100110364
(33)優先権主張国・地域又は機関	米国(US)	(74)代理人	弁理士 実広 信哉
(31)優先権主張番号	62/691,582		100133569
(32)優先日	平成30年6月28日(2018.6.28)	(72)発明者	弁理士 野村 進
(33)優先権主張国・地域又は機関			セミフ・エセンリク
最終頁に続く			ドイツ・80992・ミュンヘン・リー
			スシュトラーセ・25・ホアウェイ・テ
			最終頁に続く

(54)【発明の名称】 動きベクトル精密化および動き補償のためのメモリアクセスウィンドウおよびパディング

(57)【特許請求の範囲】

【請求項1】

ビデオ画像のブロックをエンコードするためのエンコード装置であって、
前記ブロックの初期動きベクトルを取得することと、
前記初期動きベクトルに基づいて、前記ブロックの精密化動きベクトルを取得することと、

補間フィルタを用いた補間フィルタリングによって、前記精密化動きベクトルに従って前記ブロックの予測子を決定することであって、前記補間フィルタリングは、ウィンドウの内部に配置された整数サンプル位置の少なくとも1つのサンプル値と、前記ウィンドウの外部に配置された整数サンプル位置のパディングされたサンプル値とを使用することを
含み、前記パディングされたサンプル値は、前記ウィンドウの内部に配置されたサンプルの少なくとも1つのサンプル値に基づく、決定することと、

前記初期動きベクトルと、前記ブロックと前記予測子との差分（105）とをビットストリームにエンコードすることと

を行うように構成された処理回路（600）を備え、

前記ウィンドウは、前記予測子を決定するために使用される前記補間フィルタの垂直方向長さおよび／または水平方向長さに対する垂直方向サイズおよび／または水平方向サイズによって定義される、装置。

【請求項2】

前記処理回路（600）が、候補動きベクトルの探索空間に基づいて前記初期動きベクトル

ルの精密化を決定し、前記初期動きベクトルと前記精密化とに基づいて、前記精密化動きベクトルを決定するように構成される、請求項1に記載の装置。

【請求項3】

前記探索空間が、前記初期動きベクトルによって定義される位置に配置される、請求項2に記載の装置。

【請求項4】

前記処理回路（600）が、前記ブロックと前記予測子との前記差分を取得するように構成される、請求項1から3のいずれか一項に記載の装置。

【請求項5】

前記処理回路（600）が、前記予測子と、前記ブロックと前記予測子との前記差分とに基づいて、前記ブロックを再構築するようにさらに構成される、請求項4に記載の装置。

10

【請求項6】

前記パディングされたサンプル値が、対応する評価済みの整数サンプル位置の値を、前記補間フィルタの軸線に対してミラーリングすることによって決定される、請求項1から5のいずれか一項に記載の装置。

【請求項7】

前記補間フィルタが一次元フィルタまたは分離可能な二次元フィルタである、請求項1から6のいずれか一項に記載の装置。

【請求項8】

前記ウィンドウの外部に配置された前記整数サンプル位置の前記パディングされたサンプル値が、前記ウィンドウの外部に配置された前記整数サンプル位置に最も近い、前記ウィンドウの内部に配置された少なくとも1つの使用済み整数サンプル位置のサンプル値と等しくなるように決定される、請求項1から5のいずれか一項または請求項1から5のいずれか一項を引用する請求項7に記載の装置。

20

【請求項9】

パディングされる前記サンプル値の数が、前記初期動きベクトルと前記精密化動きベクトルとの差分に依存する、請求項1から8のいずれか一項に記載の装置。

【請求項10】

前記処理回路（600）が、テンプレートを用いたテンプレート整合または双方向整合によって、前記初期動きベクトルの精密化を決定するように構成され、

30

前記テンプレート整合または前記双方向整合が、
第2のウィンドウ（1120）内の整数サンプル位置にあるサンプル値にアクセスし、前記第2のウィンドウ（1120）が、前記テンプレート整合または前記双方向整合のためにアクセス可能な、探索空間と、第2の整数サンプル位置とを含み、

前記第2のウィンドウの外部に配置された整数サンプル位置で、前記第2のウィンドウ内の少なくとも1つのサンプルに基づいた、パディングされたサンプル値によって少なくとも1つのサンプル値（1030）をパディングし、

アクセスされた整数サンプル位置の値と、前記パディングされたサンプル値とを使用して、前記テンプレート整合または前記双方向整合を実行する、請求項1から9のいずれか一項に記載の装置。

40

【請求項11】

前記探索空間が、事前定義されたタップサイズのフィルタを用いた補間フィルタリングによって取得された値を伴う、少なくとも1つの分数サンプル位置（1，2，6）を含み、

前記補間フィルタリングが、前記アクセスされた整数サンプル位置の値と、少なくとも1つの前記パディングされたサンプル値とを使用して、前記少なくとも1つの分数サンプル位置の値を取得する、請求項10に記載の装置。

【請求項12】

前記テンプレート整合または前記双方向整合で使用される前記パディングされたサンプル値が、対応する前記アクセスされた整数サンプル位置の値を、前記フィルタの軸線に対してミラーリングすることによって決定される、請求項11に記載の装置。

50

【請求項 13】

前記テンプレート整合または前記双方向整合で使用される前記パディングされたサンプル値が、前記アクセスされた整数サンプル位置に最も近い値と等しくなるように決定される、請求項10または11に記載の装置。

【請求項 14】

前記第2のウィンドウ（1120）が、前記ウィンドウと同じである、請求項10から13のいずれか一項に記載の装置。

【請求項 15】

精密化が双方向整合によって決定され、前記処理回路が、異なる参照ピクチャ（Ref0, Ref1）内の位置を指し示し、かつ初期動きベクトルのペア（Pair_0, Pair_1, ..., Pair_N）を形成する、少なくとも2つの初期動きベクトルを取得するように構成される、請求項1から14のいずれか一項に記載の装置。

10

【請求項 16】

精密化がテンプレート整合によって決定され、前記処理回路が、
前記ブロックのテンプレートを取得し、
探索空間における前記テンプレートを用いたテンプレート整合によって、前記初期動きベクトルの前記精密化を決定する
ようにさらに構成される、請求項1から15のいずれか一項に記載の装置。

【請求項 17】

ビデオ画像のブロックをエンコードするためのエンコード方法であって、
前記ブロックの初期動きベクトルを取得するステップ（1310, 1410）と、
前記初期動きベクトルに基づいて、前記ブロックの精密化動きベクトルを取得するステップ（1320, 1330, 1340, 1720; 2115, 2120, 2130, 2140, 2150, 2160）と、

20

補間フィルタを用いた補間フィルタリングによって、前記精密化動きベクトルに従って前記ブロックの予測子を決定するステップ（1780）であって、前記補間フィルタリングは、ウィンドウの内部に配置された整数サンプル位置の少なくとも1つのサンプル値と、前記ウィンドウの外部に配置された整数サンプル位置のパディングされたサンプル値とを使用することを含み、前記パディングされたサンプル値は、前記ウィンドウの内部に配置されたサンプルの少なくとも1つのサンプル値に基づく、ステップ（1780）と、

30

前記初期動きベクトルと、前記ブロックと前記予測子との差分（105）とをビットストリームにエンコードするステップと

を含み、

前記ウィンドウは、前記予測子を決定するために使用される前記補間フィルタの垂直方向長さおよび／または水平方向長さに対する垂直方向サイズおよび／または水平方向サイズによって定義される、方法。

【請求項 18】

プロセッサで実行されると、前記プロセッサに請求項17に記載の方法を実行させる命令を記憶したコンピュータ可読記憶媒体。

【発明の詳細な説明】

40

【技術分野】

【0001】

本発明の実施形態は、動きベクトル決定および精密化、ならびに動き補償のための予測子決定に関し、これはビデオのエンコードおよびデコード中に使用され得る。

【背景技術】

【0002】

現在のハイブリッドビデオコーデックは、予測符号化を用いている。ビデオシーケンスのピクチャは、ピクセルのブロックに細分化され、次にこれらのブロックは符号化される。ブロックをピクセルごとに符号化する代わりに、ブロック全体が、ブロックに空間的または時間的に近接する既にエンコードされたピクセルを使用して予測される。エンコーダ

50

は、ブロックとその予測との差分のみをさらに処理する。さらなる処理は、通常、ブロックピクセルの、変換領域内の係数への変換を含む。次に、ビットストリームを形成するために、係数は量子化によってさらに圧縮され、エントロピー符号化によってさらにコンパクト化され得る。ビットストリームは、デコーダがエンコードされたビデオをデコードできるようにする、任意のシグナリング情報をさらに含む。例えば、シグナリングは、入力ピクチャのサイズ、フレームレート、量子化ステップ指示、またはピクチャのブロックに適用される予測などのエンコーダ設定に関する設定を含み得る。

【0003】

時間的予測は、ビデオのフレームとも呼ばれる、ピクチャ間の時間的な相関を使用する。時間的予測はインター予測とも呼ばれ、これは、異なるビデオフレーム間（インター）の依存関係を使用する予測であることによる。したがって、現在のブロックとも呼ばれるエンコードされるブロックは、参照ピクチャとも呼ばれる、以前にエンコードされた1つ以上のピクチャから予測される。参照ピクチャは、必ずしも現在のピクチャに先行するピクチャではなく、現在のピクチャでは、現在のブロックはビデオシーケンスの表示順に配置される。エンコーダは、表示順とは異なる符号化順でピクチャをエンコードしてもよい。現在のブロックの予測として、参照ピクチャ内の同一位置にあるブロックが決定されてもよい。同一位置にあるブロックとは、現在のピクチャ内の現在のブロックと同一の位置で、参照ピクチャ内に配置されたブロックである。このような予測は、静止したピクチャ領域、すなわち1つのピクチャから別のピクチャへと動くことのないピクチャ領域に対して正確である。

【0004】

移動を考慮した予測子、すなわち動き補償予測子を取得するために、現在のブロックの予測を決定するときは通常は動き推定が使用される。したがって、現在のブロックは、同一位置にあるブロックの位置から動きベクトルによって与えられた距離で配置された、参照ピクチャ内のブロックによって予測される。デコーダが現在のブロックの同じ予測を決定できるように、動きベクトルがビットストリーム内でシグナリングされてもよい。ブロックのそれぞれに対して動きベクトルをシグナリングすることによって生じるシグナリングオーバーヘッドをさらに低減するために、動きベクトル自体が推定されてもよい。動きベクトル推定は、空間的および／または時間的領域において、隣接するブロックの動きベクトルに基づいて実行されてもよい。

【0005】

現在のブロックの予測は、1つの参照ピクチャを使用して、または2つ以上の参照ピクチャから取得された予測に重み付けすることによって計算されてもよい。隣接するピクチャは現在のピクチャに類似している可能性がきわめて高いので、参照ピクチャは、隣接するピクチャ、すなわち表示順において現在のピクチャの直前のピクチャ、および／または直後のピクチャであってもよい。しかしながら、一般に、参照ピクチャは、表示順において現在のピクチャの前または後の、かつビットストリーム内で現在のピクチャの前（デコード順）の、任意の他のピクチャであってもよい。これは例えば、ビデオコンテンツ内でオクルージョンおよび／または非線形の動きがあった場合に利点となる場合がある。したがって参照ピクチャの識別もまた、ビットストリーム内でシグナリングされてもよい。

【0006】

インター予測の特別なモードは、現在のブロックの予測を生成する際に2つの参照ピクチャが使用される、いわゆる双予測である。具体的には、それぞれの2つの参照ピクチャで決定された2つの予測は、現在のブロックの予測信号と組み合わせられる。双予測は、片予測、すなわち、1つの参照ピクチャのみを使用する予測よりも正確な現在のブロックの予測になり得る。より正確に予測することは、現在のブロックのピクセルと予測との間の差分（「残差」とも呼ばれる）がより小さくなることにつながり、より効率的にエンコードされる、すなわちより短いビットストリームに圧縮され得る。一般に、現在のブロックを予測するために、3つ以上の参照ピクチャが、3つ以上の参照ブロックをそれぞれ見つけるのに使用されてもよく、すなわち、多参照インター予測が適用され得る。したがって、

多参照予測という用語は、双予測および3つ以上の参照ピクチャを使用する予測を含む。

【0007】

より正確な動き推定を提供するために、参照ピクチャの解像度は、ピクセル間にサンプルを補間することによって向上させてもよい。最も近いピクセルの加重平均によって、分数ピクセル補間が実行され得る。1/2ピクセル解像度の場合、例えば、通常は双線形補間が使用される。他の分数ピクセルは、それぞれの最も近いピクセルと、予測されるピクセルとの間の距離の逆数によって重み付けされた、最も近いピクセルの平均として計算される。

【0008】

動きベクトル推定は計算的に複雑なタスクであり、現在のブロックと、参照ピクチャ内の候補動きベクトルによって指し示される、対応する予測ブロック（予測子）との間の類似性が計算される。通常、探索領域はM×Mサンプルの画像を含み、M×M候補位置の各サンプル位置がテストされる。テストは、N×Nの参照ブロックCと、探索領域のテスト済みの候補位置に配置されたブロックRとの類似性尺度の計算を含む。簡単にするために、差分絶対値の和（SAD）はこの目的のためにしばしば使用される手段であり、次の式で求められる。

【数1】

$$SAD(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |R_{i,j}(x, y) - C_{i,j}|$$

【0009】

上の数式では、xおよびyは探索領域内の候補位置を定義し、指数iおよびjは、参照ブロックCおよび候補ブロックR内のサンプルを示す。候補位置はしばしばブロック変位またはオフセットと呼ばれ、探索領域内の参照ブロックのシフト、および参照ブロックCと、探索領域の重複部分との類似性の計算として、ブロック整合の表示を反映する。複雑さを低減するために、候補動きベクトルの数は、通常は、候補動きベクトルを一定の探索空間に制限することによって削減される。探索空間は、例えば、現在の画像における現在のブロックの位置に対応する参照ピクチャ内の位置を囲んでいる、ピクセルの数および／または位置によって定義されてもよい。すべてのM×Mの候補位置xおよびyに対してSADを計算した後で、最もよく整合するブロックRは、最低SADになる位置にあるブロックであり、参照ブロックCとの最大類似度に対応する。一方、候補動きベクトルは、隣接するブロックの動きベクトルによって形成された、候補動きベクトルのリストによって定義されてもよい。

【0010】

動きベクトルは、通常は少なくとも部分的にエンコーダ側で決定され、符号化されたビットストリーム内でデコーダにシグナリングされる。しかしながら、動きベクトルはデコーダで導出されてもよい。このような事例では、現在のブロックはデコーダでは使用できず、候補動きベクトルが参照ピクチャ内で指し示すブロックに対する類似性の計算に使用することはできない。したがって、現在のブロックではなく、既にデコードされたブロックのピクセルの外部で構築された、テンプレートが使用される（「テンプレート整合」）。例えば、現在のブロックに隣接する、既にデコードされたピクセルが使用されてもよい。このような動き推定によって、動きベクトルはエンコーダおよびデコーダの両方において同じ方法で導出されるのでシグナリングは必要なく、シグナリングが削減されるという利点を得られる。一方、このような動き推定の精度は低くなる場合がある。

【0011】

双予測の場合、テンプレートを使用する代わりに、候補動きベクトルが指し示す、異なる参照ピクチャ内の候補符号化ブロック同士の比較が使用されてもよく（「双方向整合」）、これについては以下でさらに詳しく説明される。

【0012】

精度とシグナリングオーバーヘッドとのトレードオフを提供するために、動きベクトル推定は、動きベクトル導出および動きベクトル精密化の2つのステップに分けられてもよい。例えば、動きベクトル導出は、候補リストからの動きベクトルの選択を含んでもよい。このような選択された動きベクトルは、例えば、探索空間内を探索することによって、さらに精密化されてもよい。探索空間内の探索は、各候補動きベクトルに対する、すなわち候補動きベクトルが指し示すブロックの各候補位置に対する、コスト関数の計算に基づく。

【0013】

文献「X. Chen, J. An, J. Zheng, JVET-D0029: Decoder-Side Motion Vector Refinement Based on Bilateral Template Matching」(サイト<http://phenix.it-sudparis.eu/jvet/>で見ることができる)では、整数ピクセル解像度の第1の動きベクトルが見つけれられ、第1の動きベクトルの周囲の探索空間における1/2ピクセル解像度での探索によってさらに精密化される、動きベクトル精密化を示す。

【0014】

動きベクトル精密化を実行するために、少なくともこれらのサンプルをメモリに記憶することが必要であり、サンプルは現在のブロックが精密化を実行するのに必要とされる、すなわち探索空間に対応するサンプル、および探索空間でテンプレート整合または双方向整合が実行されるときにアクセスされ得るサンプルである。

【0015】

外部メモリアクセスは、このハードウェアアーキテクチャおよび／またはソフトウェア実施における重要な設計パラメータである。これは、外部メモリアクセスは、内部メモリを使用するのに比べて処理が遅くなるためである。一方、例えば、チップサイズ実施のために、チップ上の内部メモリは制限される。

【発明の概要】

【課題を解決するための手段】

【0016】

本開示は、例えば、分数補間と組み合わせて実施されるとき動きベクトル精密化は、オンチップメモリサイズ、または外部メモリアクセスをより増やすことを必要とする場合があるという着眼点に基づく。この選択肢はいずれも望ましくない場合がある。分数サンプル位置を含めることによって予測子の解像度を向上させるために補間が適用される場合は、同じことが当てはまる。

【0017】

上述した問題を考慮して、本開示は、外部メモリへのアクセス数と、符号化ブロックに対する動きベクトルの動きベクトル精密化のためにアクセス可能にすることが必要なサンプル数とを考慮可能にし、ブロックの予測子を計算する、動きベクトル予測を提供する。

【0018】

これは、メモリアクセスウィンドウ内からの1つ以上のサンプルに基づく動きベクトル精密化、および／または置換値を用いた分数補間の目的で、事前定義されたメモリアクセスウィンドウの外部に配置された値をパディング(置換)することによって達成される。

【0019】

これは、独立請求項の特徴によって達成される。

【0020】

本発明の態様によれば、ビデオ画像のブロックをエンコードするための予測子を決定するためのエンコード装置が提供される。エンコード装置は、ブロックに対する初期動きベクトルを取得し、初期動きベクトルに基づいて、ブロックに対する精密化動きベクトルを取得し、精密化動きベクトルに従って、ブロックに対する予測子を決定するように構成された処理回路を備える。処理回路は、補間フィルタを用いた補間を使用して、精密化動きベクトルに従って予測子を決定し、ウィンドウの内部に配置された整数サンプル位置の、少なくとも1つのサンプル値と、補間フィルタリングのための、ウィンドウの外部に配置された整数サンプル位置に対するパディングされたサンプル値とを使用するように構成され、パディングされたサンプル値は、ウィンドウの内部に配置されたサンプルの少なくとも

10

20

30

40

50

も1つのサンプル値に基づく。

【0021】

本発明の別の態様によれば、ビデオ画像のブロックをデコードするための予測子を決定するためのデコード装置が提供される。デコード装置は、ブロックに対する初期動きベクトルを取得し、初期動きベクトルに基づいて、予測ブロックに対する精密化動きベクトルを取得し、精密化動きベクトルに従って、ブロックに対する予測子を決定するように構成された処理回路を備える。処理回路は、補間フィルタを用いた補間を使用して、精密化動きベクトルに従って予測子を決定し、ウィンドウの内部に配置された整数サンプル位置の、少なくとも1つのサンプル値と、補間フィルタリングのための、ウィンドウの外部に配置された整数サンプル位置に対するパディングされたサンプル値とを使用するように構成され、パディングされたサンプル値は、ウィンドウの内部に配置されたサンプルの少なくとも1つのサンプル値に基づく。

10

【0022】

本発明のさらに別の態様によれば、ビデオ画像のブロックをエンコードするための予測子を決定するためのエンコード方法が提供される。エンコード方法は、ブロックに対する初期動きベクトルを取得するステップと、初期動きベクトルに基づいてブロックに対する精密化動きベクトルを取得するステップと、精密化動きベクトルに従ってブロックに対する予測子を決定するステップとを含み、決定するステップは、補間フィルタを用いた補間を使用して、精密化動きベクトルに従って予測子を決定し、ウィンドウの内部に配置された整数サンプル位置の少なくとも1つのサンプル値と、補間フィルタリングのための、ウィンドウの外部に配置された整数サンプル位置に対するパディングされたサンプル値とを使用し、パディングされたサンプル値は、ウィンドウの内部に配置された整数サンプルの少なくとも1つのサンプル値に基づく。

20

【0023】

本発明のさらに別の態様によれば、ビデオ画像のブロックをデコードするための予測子を決定するためのデコード方法が提供される。デコード方法は、ブロックに対する初期動きベクトルを取得するステップと、初期動きベクトルに基づいて、予測ブロックに対する精密化動きベクトルを取得するステップと、補間フィルタを用いた補間を使用して、精密化動きベクトルに従って予測子を決定するステップとを含み、ウィンドウの内部に配置された整数サンプル位置の少なくとも1つのサンプル値と、補間フィルタリングのための、ウィンドウの外部に配置された整数サンプル位置に対するパディングされたサンプル値とを使用し、パディングされたサンプル値は、ウィンドウの内部に配置された整数サンプルの少なくとも1つのサンプル値に基づく。

30

【0024】

このような方法および装置には、補間フィルタリングによる予測子決定の目的で使用可能にされるサンプルの数を制限しながら、参照ピクチャ全体を記憶する記憶装置／（外部）メモリへのさらなるアクセスも回避するという利点がある。

【0025】

上述した態様の実施形態において、ウィンドウとは、補間フィルタリングのためのウィンドウである。

40

【0026】

実施形態によれば、補間は、ウィンドウ内の整数サンプル位置でサンプル値にアクセスすること、かつ／またはパディング値によって、ウィンドウの外部に配置された整数サンプル位置に対する少なくとも1つのサンプル値をパディングすることを含んでもよく、これはウィンドウ内の少なくとも1つのサンプルに基づき、かつ／または補間フィルタリングを実行するために、アクセスされた整数位置サンプル値およびパディングされたサンプル値を使用する。

【0027】

例えば、処理回路は、候補動きベクトルの探索空間に基づいて初期動きベクトルの精密化を決定し、初期動きベクトルと精密化とに基づいて、精密化動きベクトルを決定するよ

50

うに構成されてもよい。

【0028】

また、例えば、処理回路は、初期動きベクトルに基づいて探索空間を決定するように構成されてもよい。より詳細には、探索空間は、初期動きベクトルによって定義された位置に配置されてもよい。例えば、探索空間は、初期動きベクトルを中心としてもよい。より具体的には、例によれば、探索空間は9つの整数サンプル動きベクトル候補、すなわち初期動きベクトルと、周囲に配置された8つの動きベクトル候補とを含んでもよい。

【0029】

実施形態では、処理回路は、予測子に基づいてブロックを再構築するようにさらに構成される。再構築はデコード装置において、およびエンコード装置のデコーダループにおいて実行されてもよい。方法に関する本開示の態様は、予測子に基づく、ブロックのそれぞれの再構築ステップを含み得る。

【0030】

例えば、処理回路は、「残差」とも呼ばれる、ブロックと予測子との差分を取得するように構成されてもよい。より具体的には、処理回路は、予測子および残差に基づいて、さらに具体的には、予測子と残差との合計としてブロックを再構築するように構成されてもよい。

【0031】

例えば、パディングされたサンプル値（置換値）は、対応する使用済みの（評価済みの）整数サンプル位置の値を、補間フィルタの軸線に対してミラーリングすることによって決定される。補間フィルタリングは、一次元フィルタリング、または分離可能な二次元フィルタリングであってもよい。

【0032】

1つの例示的な実施において、ウィンドウは、初期動きベクトルによって指定された位置で、予測子を取得するためにアクセス可能なすべての整数サンプル位置によって形成される。

【0033】

ウィンドウは、例えば、初期動きベクトル位置に配置されたブロックのサイズを有するブロック、初期動きベクトル位置、または予測子を決定するために使用される補間フィルタの垂直方向長さおよび水平方向長さに対する、その垂直方向サイズおよび／または水平方向サイズによって定義される。特定の位置に「配置される」ブロックの定義は、実施形態では、例えば、ブロックの左上のサンプル位置を参照することを意味する。あるいは、別のサンプル位置が参照されてもよく、またはブロック位置は、特定の位置を中心とすることによって定義されてもよい。処理回路は、初期動きベクトル位置に従った、ウィンドウの位置および／もしくはサイズ、ブロックのサイズ、ならびに／または予測子を決定するために使用される補間フィルタの垂直方向長さおよび水平方向長さを定義または決定するように構成され得る。

【0034】

別の例として、ウィンドウの外部にある整数サンプル位置に対するパディングされたサンプル値は、ウィンドウの外部に配置された整数サンプル位置に最も近い、ウィンドウの内部に配置された少なくとも1つの使用済み整数サンプル位置のサンプル値と等しくなるように決定される。特定の例において、一次元補間フィルタの場合、ウィンドウの外部に配置された、複数の整数サンプル位置に対するパディングされたサンプル値は、補間フィルタを用いた補間を使用して予測子を決定するために使用され、ウィンドウの外部にあるすべての整数サンプル位置に対するパディングされたサンプル値は、ウィンドウの外部に配置された整数サンプル位置に最も近い、ウィンドウの内部に配置された少なくとも1つの使用済み整数サンプル位置のサンプル値と等しくなるように決定される。

【0035】

実施形態によれば、パディングされるサンプル値の数は、初期動きベクトルと精密化動きベクトルとの差分に依存する。

10

20

30

40

50

【0036】

実施形態によれば、処理回路は、テンプレート整合または双方向整合を伴うテンプレート整合によって、初期動きベクトルの精密化を決定するように構成される。テンプレート整合または双方向整合は、別のウィンドウ内の整数サンプル位置でサンプル値にアクセスする。別のウィンドウは、探索空間と、テンプレート整合または双方向整合のためにアクセス可能な別の整数サンプル位置とを含み、テンプレート整合または双方向整合は、別のウィンドウ内の少なくとも1つのサンプルに基づく、パディングされたサンプル（置換）値によって、別のウィンドウの外部に配置された整数サンプル位置で、少なくとも1つのサンプル値をさらにパディングし、アクセスされた整数サンプル位置の値およびパディングされたサンプル値を使用して、テンプレート整合または双方向整合を実行する。

10

【0037】

一実施形態において、前記探索空間は、事前定義されたタップサイズのフィルタを用いた補間フィルタリングによって取得された値を伴う、少なくとも1つの分数サンプル位置を含み、補間フィルタリングは、前記アクセスされた整数サンプル位置の値と、少なくとも1つの置換（パディングされたサンプル）値とを使用して、前記少なくとも1つの分数サンプル位置の値を取得する。

【0038】

より詳細には、テンプレート整合または双方向整合で使用されるパディングされたサンプル値は、対応する評価済みの整数サンプル位置の値を、フィルタの軸線に対してミラーリングすることによって決定される。

20

【0039】

あるいは、テンプレート整合または双方向整合で使用されるパディングされたサンプル値は、アクセスされた整数サンプル位置に最も近い値と等しくなるように決定される。

【0040】

実施形態によれば、別のウィンドウは、探索空間、初期動きベクトル位置に配置されたブロックのサイズを有するブロック、および初期動きベクトル位置、のうちの少なくとも1つに対する、その垂直方向サイズおよび／または水平方向サイズで定義される。

【0041】

好ましくは、別のウィンドウはウィンドウと同じである。

【0042】

実施形態によれば、精密化は双方向整合によって決定され、処理回路は、異なる参照ピクチャ内の位置を指し示し、初期動きベクトルのペアを形成する、少なくとも2つの初期動きベクトルを取得するように構成される。

30

【0043】

他の実施形態によれば、精密化はテンプレート整合によって決定され、処理回路は、ブロックのテンプレートを取得し、探索空間におけるテンプレートを用いたテンプレート整合によって、初期動きベクトルの精密化を決定するようにさらに構成される。

【0044】

一実施形態において、前記探索空間は、事前定義されたタップサイズのフィルタを用いた補間フィルタリングによって取得された値を伴う、少なくとも1つの分数サンプル位置を含み、補間フィルタリングは、前記使用済みの整数サンプル位置の値と、少なくとも1つのパディングされたサンプル値とを使用して、前記少なくとも1つの分数サンプル位置の値を取得する。

40

【0045】

本発明の態様によれば、プロセッサ／処理回路によって実行されると、上述したいずれかの態様もしくは実施形態、またはこれらの組み合わせによるステップを実行する命令を記憶する、非一時的コンピュータ可読記憶媒体が提供される。

【0046】

本発明の別の特定の態様によれば、予測ブロックに分割されたビデオ画像をエンコードしてビットストリームにするための、またはビデオ画像の予測ブロックをビットストリー

50

ムにエンコードするための、エンコード装置がそれぞれ提供される。エンコード装置は、予測ブロックに対する動きベクトルを決定するための処理回路を備える。処理回路は、少なくとも1つの初期動きベクトルを取得し、候補動きベクトルの探索空間に基づいて初期動きベクトルの精密化を決定して、精密化動きベクトルを取得するように構成される。探索空間は、初期動きベクトルによって与えられた位置に配置される。エンコード装置は、予測ブロックと決定された動きベクトルに基づく位置で予測ブロックによって与えられた予測子との差分をエンコードし、エンコードされた差分および初期動きベクトルを含むビットストリームを生成するためのエンコード回路（処理回路に含まれていてもよい）をさらに備える。エンコード装置は、補間フィルタを用いた補間フィルタリングを使用して、動きベクトルに従って予測子を決定するための動き予測ユニット（処理回路に含まれていてもよい）をさらに備える。補間フィルタリングは、事前定義されたウィンドウ内の整数サンプル位置でサンプル値にアクセスし、ウィンドウの外部に配置された整数サンプル位置で、ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって、少なくとも1つのサンプル値を置換し、アクセスされた整数位置サンプル値および置換値を使用して、補間フィルタリングを実行する。

10

【0047】

本発明のさらに別の特定の態様によれば、ビットストリームから、予測ブロックに分割されたビデオ画像をデコードするためのデコード装置が提供される。デコード装置は、ビットストリームから、初期動きベクトルと、予測ブロックと精密化動きベクトルによって指定された位置で予測ブロックによって与えられた予測子とのエンコードされた差分とを解析するための解析ユニット（デコード装置の処理回路に含まれていてもよい）を備える。デコード装置は、予測ブロックに対する動きベクトルを決定するための処理回路（例えば、デコード装置の処理回路）を備える。処理回路は、少なくとも1つの初期動きベクトルを取得し、候補動きベクトルの探索空間に基づいて初期動きベクトルの精密化を決定して、精密化動きベクトルを取得するように構成される。探索空間は、初期動きベクトルによって与えられた位置に配置される。さらにまた、デコード装置は、動きベクトルを決定するために処理回路によって決定された、精密化動きベクトルによって指定された位置で、解析された差分と、予測ブロックによって与えられた予測子との合計として予測ブロックを再構築するためのデコード回路（デコード装置の処理回路に含まれていてもよい）を備える。デコード回路は、補間フィルタを用いた補間フィルタリングを使用して、動きベクトルに従って予測子を決定するようにさらに構成される。補間フィルタリングは、事前定義されたウィンドウ内の整数サンプル位置でサンプル値にアクセスし、ウィンドウの外部に配置された整数サンプル位置で、ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって、少なくとも1つのサンプル値を置換し、アクセスされた整数位置サンプル値および置換値を使用して、補間フィルタリングを実行する。

20

30

【0048】

本発明のさらに別の態様によれば、予測ブロックに分割されたビデオ画像をビットストリームにエンコードするためのエンコード方法が提供される。エンコード方法は、予測ブロックに対する動きベクトルを決定するステップを含む。動きベクトルを決定するステップは、少なくとも1つの初期動きベクトルを取得するステップと、候補動きベクトルの探索空間に基づいて初期動きベクトルの精密化を決定して、精密化動きベクトルを取得するステップとを含む。探索空間は、初期動きベクトルによって与えられた位置に配置される。エンコード方法は、予測ブロックと決定した動きベクトルに基づく位置で予測ブロックによって与えられた予測子との差分をエンコードし、エンコードされた差分および初期動きベクトルを含むビットストリームを生成するステップをさらに含む。本方法は、補間フィルタを用いた補間フィルタリングを使用して、動きベクトルに従って予測子を決定するステップをさらに含む。補間フィルタリングは、事前定義されたウィンドウ内の整数サンプル位置でサンプル値にアクセスし、ウィンドウの外部に配置された整数サンプル位置で、ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって、少なくとも1つのサンプル値を置換し、アクセスされた整数位置サンプル値および置換値を使用して、補間フ

40

50

フィルタリングを実行する。

【0049】

本発明のさらに別の態様によれば、ビットストリームから、予測ブロックに分割されたビデオ画像をデコードするためのデコード方法が提供される。デコード方法は、ビットストリームから、初期動きベクトルと、予測ブロックと精密化動きベクトルによって指定された位置で予測ブロックによって与えられた予測子とのエンコードされた差分とを解析するステップを含む。本方法は、予測ブロックに対する動きベクトルを決定するステップをさらに含む。動きベクトルを決定するステップは、少なくとも1つの初期動きベクトルを取得するステップと、候補動きベクトルの探索空間に基づいて初期動きベクトルの精密化を決定して、精密化動きベクトルを取得するステップとを含む。探索空間は、初期動きベクトルによって与えられた位置に配置される。デコード方法は、動きベクトルを決定するステップによって決定された、精密化動きベクトルによって指定された位置で、解析された差分と、予測ブロックによって与えられた予測子との合計として予測ブロックを再構築するステップをさらに含む。さらに、本方法は、補間フィルタを用いた補間フィルタリングを使用して、動きベクトルに従って予測子を決定するステップを含む。補間フィルタリングは、事前定義されたウィンドウ内の整数サンプル位置でサンプル値にアクセスし、ウィンドウの外部に配置された整数サンプル位置で、ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって、少なくとも1つのサンプル値を置換し、アクセスされた整数位置サンプル値および置換値を使用して、補間フィルタリングを実行する。

【0050】

本発明の別の利点および実施形態については、従属請求項に記載されている。

【0051】

以下、例示的な実施形態が添付の図面および図を参照してより詳細に説明される。

【図面の簡単な説明】

【0052】

【図1】動きベクトル導出および精密化が使用され得るエンコーダの例示的な構造を示すブロック図である。

【図2】動きベクトル導出および精密化が使用され得るデコーダの例示的な構造を示すブロック図である。

【図3】双予測に適した例示的なテンプレート整合を示す概略図である。

【図4】片予測および双予測に適した例示的なテンプレート整合を示す概略図である。

【図5】ビットストリーム内で精密化される初期動きベクトルを提供することなく動作する、動きベクトル導出の段階を示すブロック図である。

【図6】双方向整合の概念を示す概略図である。

【図7】本発明における実施形態を実施するための例示的なハードウェアを示すブロック図である。

【図8】符号化ブロック、およびアクセス可能になるべきサンプルを含む例示的なウィンドウを示す概略図である。

【図9】反復的な探索空間を示す概略図である。

【図10】補間フィルタリングによる、メモリアクセスウィンドウの水平方向への拡張を示す概略図である。

【図11】探索空間に基づいて動きベクトル精密化を実行し、補間によって予測を得る従来の手法を示す図である。

【図12】図11の従来の例による動きベクトル導出、および予測の取得を示すフローチャートである。

【図13】本発明の例示的な実施形態に従って、探索空間に基づいて動きベクトル精密化を実行し、補間によって予測を取得する手法を示す図である。

【図14】分数位置計算を可能にするための、2つのサンプルのパディングを示す概略図である。

【図15】探索空間境界でのテンプレート整合を可能にするための、4つのサンプルのパ

ディングを示す概略図である。

【図16】補間にパディングを必要としない、例示的な分数位置を示す概略図である。

【図17】図13の例示的な実施形態による、動きベクトル精密化、および予測の取得を示すフロー図である。

【図18】予測ユニットに対するメモリアクセスウィンドウの定義を示す概略図である。

【図19】実施形態による方法を示す図である。

【図20】本発明の別の例示的な実施形態に従って、探索空間に基づいて動きベクトル精密化を実行し、補間によって予測を取得する手法を示す図である。

【図21】図20の例示的な実施形態による、動きベクトル精密化、および予測の取得を示すフロー図である。

【発明を実施するための形態】

【0053】

本開示は、参照ピクチャ内および予測子内で分数位置を取得することを目的として、動きベクトル精密化および補間を実行するためにアクセス可能にされる、サンプルの数の調節に関する。

【0054】

上述したように、外部メモリアクセスは、今日のハードウェアおよびソフトウェアのアーキテクチャにおいて最も重要な設計上の考慮事項の1つである。動きベクトル推定は、特にテンプレート整合または双方向整合を含むとき、例えば、動きベクトル精密化の場合に、探索空間の分数位置を取得するために補間フィルタリングと併用されてもよい。補間フィルタリングを使用するためには、メモリからアクセスされるべきサンプルの数を増やすことが必要とされる場合がある。しかしながら、これによって高価なオンチップメモリが増加する、または外部メモリアクセスの回数が増加することにつながる場合があり、また他方では実施が減速する。特にデコーダ側において、このような問題は運用がより高価になる、またはより減速することにつながる場合があり望ましくない。

【0055】

このような状況を避けるために、本開示は、事前定義されたウィンドウに対する外部メモリアクセスの制限を提供する。

【0056】

ウィンドウは、事前定義されたサイズであってもよい。例えば、動きベクトル精密化において、ウィンドウは、少なくとも探索空間位置をすべて含んでもよい。アクセスされるサンプルは、テンプレート整合または双方向整合を実行するために、特定のブロックおよびその探索空間用に実際に使用されるサンプルであり、例えば、テンプレートを使用したコスト関数、または双方向コスト関数の計算に使用されるすべてのサンプルである。アクセス可能なサンプルは、特定のブロックに対する特定のテンプレート整合または双方向整合中に必ずしもアクセスされないサンプルをさらに含み、それは例えば、探索空間が異なるサンプルを含むように反復的に決定されたからであり、そのサンプルに対するテンプレート整合または双方向整合が、アクセス可能なピクセルのすべてを必要とはしないからである。アクセス可能なサンプルは、任意の反復的に決定された探索空間（すなわち任意のブロックコンテンツと仮定する）から得られる可能性があるサンプルをさらに含み、おそらくは、このような考えられる探索空間でテンプレート整合を実行するためにアクセスされるサンプルをさらに含む。

【0057】

補間フィルタリングを用いて最終予測を生成する場合、従来の手法において必要なメモリアクセスウィンドウを拡張する程度は、決定した精密化動きベクトルと、初期動きベクトルとの最大差に依存する。

【0058】

ウィンドウは、すべてのアクセス可能なサンプルよりも小さく定義されてもよいことに留意されたい。ウィンドウをこのように定義することによって、オンチップメモリに記憶され外部メモリから取り込まれるサンプルの数をより少なく維持することが可能になる。

10

20

30

40

50

したがって本開示は、特定のブロックに対するテンプレート整合または双方向整合に関する動作のために記憶／キャッシュ／バッファリングされる整数サンプルの数を制限する。これは、別のメモリまたは記憶装置には通常は他のサンプルが存在しない／利用できないことを意味するものではない。上述したように、通常は参照ピクチャ全体が外部メモリに記憶され得る。しかしながら、テンプレート整合または双方向整合のためには、その一部のみが、すなわちウィンドウ内のサンプルが、ロードされてテンプレート整合に使用されてもよい。

【0059】

このように制限されたウィンドウが提供されることによって、それを使用して実施することが可能になり、実際にはウィンドウサンプルをロードするだけで済む。エンコーダおよびデコーダが確実に同様の動作をするように、ウィンドウ定義は標準的に事前定義されてもよく、またはビットストリーム内でシグナリングされてもよい。特に、参照ピクチャの形成に精密化動きベクトルが使用される場合は、エンコーダおよびデコーダの両方が同じ手法を使用しなければならない。

【0060】

図1は、ビデオストリームのフレームまたはピクチャの入力画像サンプルを受信するための入力と、エンコードされたビデオビットストリームを生成するための出力とを備えるエンコーダ100を示している。本開示における「フレーム」という用語は、ピクチャの同義語として使用される。しかしながら、本開示は、インターレースが適用される場合がある分野にも適用可能であることに留意されたい。一般に、ピクチャは $m \times n$ のピクセルを含む。これは画像サンプルに対応し、それぞれ1つ以上の色成分を含み得る。簡単にするために、以下の説明は、輝度のサンプルを意味するピクセルに言及する。しかしながら、本発明の動きベクトル探索は、クロミナンスを含む任意の色成分またはRGBなどの探索空間の成分に適用され得ることに留意されたい。一方、1つの成分に対して動きベクトル推定のみを実行し、決定された動きベクトルをより多くの（またはすべての）成分に適用することが有益な場合がある。

【0061】

符号化される入力ブロックは、必ずしも同じサイズを有さない。1つのピクチャは、異なるサイズのブロックを含む場合があり、異なるピクチャのブロックラスタも異なる場合がある。

【0062】

説明的な実施において、エンコーダ100は、予測、変換、量子化、およびエントロピー符号化をビデオストリームに適用するように構成される。変換、量子化およびエントロピー符号化は、符号化ビデオビットストリームを出力として生成するために、変換ユニット106、量子化ユニット108およびエントロピーエンコードユニット170によってそれぞれ実行される。

【0063】

ビデオストリームは複数のフレームを含む場合があり、各フレームは、イントラ符号化またはインター符号化された特定のサイズのブロックに分割される。例えばビデオストリームの第1のフレームのブロックは、イントラ予測ユニット154によってイントラ符号化される。イントラフレームは、同じフレーム内の情報のみを使用して符号化されるので、独立してデコードすることができ、ランダムアクセスのためにビットストリーム内にエントリポイントを提供することができる。ビデオストリームの他のフレームのブロックは、インター予測ユニット144によってインター符号化されてもよく、以前に符号化されたフレーム（参照フレーム）からの情報は時間的冗長性を低減するために使用され、これにより、インター符号化されたフレームの各ブロックが参照フレーム内のブロックから予測される。モード選択ユニット160は、フレームのブロックがイントラ予測ユニット154またはインター予測ユニット144のどちらによって処理されるかを選択するように構成される。このモード選択ユニット160はまた、イントラ予測またはインター予測のパラメータを制御する。画像情報のリフレッシュを可能にするために、イントラ符号化されたブロック

は、インター符号化されたフレーム内で提供されてもよい。さらに、イントラ符号化されたブロックのみを含むイントラフレームは、デコーディングのためのエントリポイント、すなわち、デコーダが以前に符号化されたフレームからの情報を有することなくデコーディングを開始できるポイントを提供するために、定期的にビデオシーケンスに挿入され得る。

【0064】

イントラ推定ユニット152およびイントラ予測ユニット154は、イントラ予測を実行するユニットである。特に、イントラ推定ユニット152は、さらにオリジナル画像の知識に基づいて予測モードを導出してもよく、イントラ予測ユニット154は、差分符号化のための対応する予測子、すなわち、選択した予測モードを使用して予測されたサンプルを提供する。空間的または時間的予測を実行するために、符号化されたブロックは、逆量子化ユニット110および逆変換ユニット112によってさらに処理され得る。ブロックの再構築後、ループフィルタリングユニット120が、デコードされた画像の品質をさらに改善するために適用される。次に、フィルタリングされたブロックは参照フレームを形成し、参照フレームはその後、デコード済みピクチャバッファ130に記憶される。エンコーダ側のこのようなデコーダループ（デコーダ）は、デコーダ側で再構築される参照ピクチャと同じ参照フレームを生成するという利点を提供する。したがって、エンコーダ側とデコーダ側は、対応する方法で動作する。ここでの「再構築」という用語は、予測ブロックをデコードされた残差ブロックに加算することによって、再構築されたブロックを取得することを指す。

【0065】

インター推定ユニット142は、インター符号化される現在のフレームまたはピクチャのブロックと、デコード済みピクチャバッファ130からの1つまたはいくつかの参照フレームとを入力として受信する。インター推定ユニット142によって動き推定が実行され、インター予測ユニット144によって動き補償が適用される。動き推定は、例えば、符号化される元の画像も使用して、特定のコスト関数に基づいて動きベクトルおよび参照フレームを取得するために使用される。例えば、動き推定ユニット142は、初期動きベクトル推定を提供し得る。次に、初期動きベクトルは直接、ベクトルの形態で、またはエンコーダおよびデコーダにおけるのと同じ方法で、事前定義されたルールに基づいて構築された候補リスト内で動きベクトル候補を参照するインデックスとして、ビットストリーム内でシグナリングされてもよい。次に、動き補償は、参照フレーム内で現在のブロックと同一位置にあるブロックを、参照フレーム内の参照ブロックに変換するものとして、すなわち動きベクトルによって、現在のブロックの予測子を導出する。インター予測ユニット144は現在のブロックに対する予測ブロックを出力し、前記予測ブロックはコスト関数を最小化する。例えば、コスト関数は、符号化される現在のブロックと、その予測ブロックとの差であってもよく、すなわちコスト関数は残差ブロックを最小化する。残差ブロックの最小化は、例えば、現在のブロックのすべてのピクセル（サンプル）と、候補参照ピクチャ内の候補ブロックとの差分絶対値の和（SAD）の計算に基づく。しかしながら、一般には、平均二乗誤差（MSE）または構造類似性メトリック（SSIM）などその他の類似性メトリックが用いられる場合がある。

【0066】

しかしながら、コスト関数はまた、そのようなインターブロックおよび／またはそのような符号化から生じる歪みを符号化するために必要なビットの数であってもよい。したがって、レート歪み最適化手順が、動きベクトルの選択、および／または一般に、ブロックに対してインター予測とイントラ予測のどちらを使用するか、およびその設定などのエンコーディングパラメータを決定するために使用されてもよい。

【0067】

イントラ推定ユニット152およびイントラ予測ユニット154は、イントラ符号化される現在のフレームまたはピクチャのブロックと、現在のフレームの既に再構築された領域からの1つまたはいくつかの参照サンプルとを入力として受信する。次に、イントラ予測は

、現在のフレームの参照サンプルの関数の観点から、現在のフレームの現在のブロックのピクセルを記述する。イントラ予測ユニット154は、現在のブロックの予測ブロックを出力し、前記予測ブロックは、好適には、符号化される現在のブロックとその予測ブロックとの差を最小化する、すなわち、それは残差ブロックを最小化する。残差ブロックの最小化は、例えばレート歪み最適化手順に基づいてもよい。特に、予測ブロックは、参照サンプルの方向補間として取得される。方向は、レート歪み最適化によって、および／またはインター予測に関連して上で言及されたような類似性尺度を計算することによって決定され得る。

【0068】

インター推定ユニット142は、ブロック、またはインター符号化される現在のフレームもしくはピクチャのより汎用的な形式の画像サンプルと、2つ以上の既にデコードされたピクチャ231とを入力として受信する。次に、インター予測は、参照ピクチャの画像サンプルを参照するために、動きベクトルの観点から現在のフレームの現在の画像サンプルを記述する。インター予測ユニット142は、現在の画像サンプルの1つ以上の動きベクトルを出力し、動きベクトルによって指し示される前記参照画像サンプルは、好適には、符号化される現在の画像サンプルとその参照画像サンプルとの差を最小化する、すなわち、それは残差画像サンプルを最小化する。次に、差分符号化のために、現在のブロックの予測子がインター予測ユニット144によって提供される。

【0069】

次いで、現在のブロックとその予測との差分、すなわち残差ブロック105が変換ユニット106によって変換される。変換係数107は、量子化ユニット108によって量子化され、エントロピーエンコードユニット170によってエントロピー符号化される。このように生成されたエンコードされたピクチャデータ171、すなわちエンコードされたビデオビットストリームは、イントラ符号化ブロックおよびインター符号化ブロックならびに対応するシグナリング（モード指示、動きベクトルの指示、および／またはイントラ予測方向など）を含む。変換ユニット106は、フーリエまたは離散コサイン変換（DFT／FFTまたはDCT）などの線形変換を適用し得る。空間周波数領域へのこのような変換は、結果として得られる係数107が通常、より低い周波数でより高い値を有するという利点を提供する。したがって、効果的な係数スキャン（ジグザグなど）および量子化の後、結果として得られる値のシーケンスは、通常、開始時にいくつかの大きな値を有し、ゼロの実行で終了する。これは、さらに効率的な符号化を可能にする。量子化ユニット108は、解像度の係数値を低下させることによって、事実上の不可逆圧縮を実行する。次いで、エントロピー符号化ユニット170が、ビットストリームを生成するために係数値にバイナリコードワードを割り当てる。エントロピー符号化ユニット170は、シグナリング情報も符号化する（図1には図示せず）。

【0070】

図2は、ビデオデコーダ200を示す。ビデオデコーダ200は、特に、デコード済みピクチャバッファ230と、インター予測ユニット244と、ブロック予測ユニットであるイントラ予測ユニット254とを備える。デコード済みピクチャバッファ230は、エンコードされたビデオビットストリームから再構築された少なくとも1つの（片予測用）、または少なくとも2つの（双予測用）参照フレームを記憶するように構成され、前記参照フレームは、エンコードされたビデオビットストリームの現在のフレーム（現在デコードされているフレーム）とは異なる。イントラ予測ユニット254は、デコードされるブロックの推定である予測ブロックを生成するように構成される。イントラ予測ユニット254は、デコード済みピクチャバッファ230から取得された参照サンプルに基づいてこの予測を生成するように構成される。

【0071】

デコーダ200は、ビデオエンコーダ100によって生成されたエンコードされたビデオビットストリームをデコードするように構成され、好ましくは、デコーダ200とエンコーダ100の両方が、エンコード／デコードされるそれぞれのブロックについて同一の予測を生

10

20

30

40

50

成する。デコード済みピクチャバッファ230、再構築されたブロック215、バッファ216、およびイントラ予測ユニット254の特徴は、図1のデコード済みピクチャバッファ130、再構築されたブロック115、バッファ116、およびイントラ予測ユニット154の特徴と同様である。

【0072】

ビデオデコーダ200は、例えば逆量子化ユニット210、逆変換ユニット212、およびループフィルタリングユニット220のような、ビデオエンコーダ100にも存在するさらなるユニットを備え、これらは、それぞれ、ビデオエンコーダ100の逆量子化ユニット110、逆変換ユニット112、およびループフィルタリングユニット120に対応する。

【0073】

エン트로ピーデコードユニット204は、受信したエンコード済みのビデオビットストリームをデコードし、それに対応して量子化された残差変換係数209、およびシグナリング情報を取得するように構成される。量子化された残差変換係数209は、残差ブロックを生成するために逆量子化ユニット210および逆変換ユニット212に供給される。残差ブロックは、予測ブロック265に追加され、追加されたものがループフィルタリングユニット220に供給され、デコードされたビデオが得られる。デコードされたビデオのフレームは、デコード済みピクチャバッファ230に記憶され、インター予測のためのデコードされたピクチャ231として機能し得る。

【0074】

一般に、図1および図2のイントラ予測ユニット154および254は、エンコードされる必要があるまたはデコードされる必要があるブロックの予測信号を生成するために、既にエンコードされた領域からの参照サンプルを使用し得る。

【0075】

エン트로ピーデコードユニット204は、その入力として、エンコードされたビットストリーム171を受信する。一般に、ビットストリームが最初に解析される、すなわち、シグナリングパラメータおよび残差がビットストリームから抽出される。典型的には、ビットストリームの構文およびセマンティクスは、エンコーダおよびデコーダが相互運用可能な方法で動作し得るように標準で定義される。上述した背景技術の節で説明したように、エンコードされたビットストリームは、予測残差のみを含んでいるわけではない。動き補償予測の場合は、ビットストリーム内で動きベクトル指示も符号化され、デコーダでそこから解析される。動きベクトル指示は、動きベクトルが提供される参照ピクチャによって、動きベクトル座標によって与えられ得る。ここまで、完全な動きベクトルの符号化について検討した。しかしながら、ビットストリーム内で、現在の動きベクトルと以前の動きベクトルとの差分のみがエンコードされてもよい。この手法では、隣接するブロック同士の動きベクトル間の冗長性を利用することができる。

【0076】

参照ピクチャを効率的に符号化するために、H.265コーデック (ITU-T, H265, Series H: Audiovisual and multimedia systems: High Efficient Video Coding) は、それぞれの参照フレームをリストインデックスに割り当てる参照ピクチャのリストを提供する。次いで参照フレームは、対応する割り当てられたリストインデックスをその中に含めることによって、ビットストリーム内でシグナリングされる。そのようなリストは、標準的に定義されるか、またはビデオもしくはいくつかのフレームのセットの先頭でシグナリングされてもよい。H.265では、L0およびL1と呼ばれる、定義された参照ピクチャの2つのリストがあることに留意されたい。次いで参照ピクチャは、リスト (L0またはL1) を示し、所望の参照ピクチャに関連付けられたそのリスト内のインデックスを示すことによって、ビットストリーム内でシグナリングされる。2つ以上のリストを提供すると、圧縮をより良好にするのに有利な場合がある。例えば、L0は、一方向にインター予測されるスライスと、双方向にインター予測されるスライスとの両方に使用されてもよく、L1は、双方向にインター予測されるスライスのためにのみ使用されてもよい。しかしながら、概して本開示は、L0およびL1リストのいかなる内容にも限定されない。

10

20

30

40

50

【0077】

リストL0およびL1は、標準的に定義され固定されてもよい。しかしながら、それらをビデオシーケンスの先頭でシグナリングすることによって、符号化／復号のより高い柔軟性を実現され得る。したがって、エンコーダは、インデックスに従って順序付けられた特定の参照ピクチャを用いてリストL0およびL1を構成してもよい。L0およびL1のリストは、同じ固定されたサイズを有してもよい。一般に、3つ以上のリストがあってもよい。動きベクトルは、参照ピクチャ内の座標によって直接シグナリングされてもよい。あるいは、H.265でも指定されているように、候補動きベクトルのリストが構築されてもよく、リスト内で特定の動きベクトルに関連付けられたインデックスが送信されてもよい。

【0078】

現在のブロックの動きベクトルは、通常は現在のピクチャ内、または先に符号化されたピクチャ内の、隣接するブロックの動きベクトルと相関する。これは、隣接するブロックは、同様の動きをする同じ移動オブジェクトに対応する可能性が高く、オブジェクトの動きは、時間とともに急激に変化する可能性は低いためである。その結果として、隣接するブロック内の動きベクトルを予測子として使用することにより、シグナリングされた動きベクトルの差分の大きさが縮小される。動きベクトル予測子(MVP)は、参照ピクチャ内で、空間的に隣接するブロックから、または時間的に隣接する、もしくは同一位置にあるブロックからの、既にエンコード／デコードされた動きベクトルから通常は導出される。H.264/AVCにおいて、これは、3つの空間的に隣接する動きベクトルの成分ごとの中央値をとることによって行われる。この手法を使用すると、予測子のシグナリングは必要とされない。参照ピクチャ内の同一位置にあるブロックからの時間的MVPは、H.264/AVCのいわゆる時間的ダイレクトモードにおいてのみ考慮される。H.264/AVCダイレクトモードは、動きベクトル以外の他の動きデータを導出するためにも使用される。したがって、それらは、HEVCにおけるブロックマーキングの概念との関連がより強い。HEVCでは、MVPを暗黙的に導出する手法は、動きベクトル導出のためにMVPのリストからどのMVPが使用されるかを明示的にシグナリングする、動きベクトル競合として知られる技法に置き換えられていた。HEVCにおける可変コーディング四分木ブロック構造は、潜在的なMVP候補としての動きベクトルを含む、いくつかの隣接するブロックを有する、1つのブロックになり得る。左隣を例にとると、最悪の場合において、 64×64 のルマ符号化ツリーブロックがさらに分割されず、左のブロックが最大深度まで分割される場合、 64×64 のルマ予測ブロックは16個の 4×4 ルマ予測ブロックを左に有する場合がある。

【0079】

このような柔軟なブロック構造を考慮するように動きベクトル競合を修正するために、進化型動きベクトル予測(AMVP)が導入された。HEVCの開発中は、符号化効率と実施しやすい設計との良好なトレードオフを提供するために、初期のAMVP設計は大幅に簡略化されていた。AMVPの初期の設計は、3つの異なるクラスの予測子からの5つのMVPを含み、すなわち空間的に隣接するものからの3つの動きベクトル、3つの空間予測子の中央値、および同一位置にある時間的に隣接するブロックからのスケーリングされた動きベクトルである。さらに、予測子のリストは、最も可能性の高い動き予測子を第1の位置に配置するように並べ替え、最小限のシグナリングオーバーヘッドを保証するために冗長な候補を削除することによって修正された。AMVP候補リスト構造の最終設計は、以下の2つのMVP候補、すなわちa) 5つの空間的に隣接するブロックから導出された2つまでの空間的候補MVP、b) 両方の空間的候補MVPが利用できないか、またはそれらが同一である場合に、2つの時間的な、同一位置にあるブロックから導出された1つの時間的候補MVPを含み、さらにc) 空間的候補、時間的候補、または両方の候補が利用できない場合のゼロの動きベクトルを含む。動きベクトル決定に関する詳細は、参照により本明細書に組み込まれる、文献「V. Sze et. al. (Ed.), High Efficiency Video Coding (HEVC) : Algorithms and Architectures, Springer, 2014」の、特にチャプター5で見ることができる。

【0080】

シグナリングオーバーヘッドをさらに増加させずに動きベクトル推定をさらに改善するた

10

20

30

40

50

めには、エンコーダ側で導出され、ビットストリーム内で提供される動きベクトルをさらに精密化することが有益であろう。動きベクトル精密化は、エンコーダからの支援なしにデコーダで実行されてもよい。そのデコーダループ内にあるエンコーダは、対応する動きベクトルを取得するために同じ精密化を用いてもよい。動きベクトル精密化は探索空間内で実行され、探索空間は、参照ピクチャの整数ピクセル位置と分数ピクセル位置とを含む。例えば、分数ピクセル位置は、 $1/2$ ピクセル位置もしくは $1/4$ ピクセル位置、または別の分数位置であってもよい。分数ピクセル位置は、双線形補間などの補間によって、整数（フルピクセル）位置から取得され得る。しかしながら、本開示は分数ピクセル位置を含んでいない探索空間にも等しく適用可能である。

【0081】

現在のブロックの双予測では、リストL0の第1の動きベクトルと、リストL1の第2の動きベクトルとをそれぞれ使用して取得された2つの予測ブロックが組み合わされて単一の予測信号になり、これによって片予測よりも、元の信号への適応をより良好にすることができ、その結果、残差情報がより少なくなり、圧縮がより効率的になる可能性がある。

【0082】

デコーダでは、現在のブロックはデコード中で使用できないため、テンプレートが動きベクトル精密化のために使用され、テンプレートは現在のブロックの推定であり、既に処理された（すなわちエンコーダ側で符号化され、デコーダ側でデコードされた）画像部分に基づいて構築される。

【0083】

最初に、第1の動きベクトルMV0の推定、および第2の動きベクトルMV1の推定は、デコーダ200での入力として受信される。エンコーダ側100において、動きベクトル推定MV0およびMV1はブロック整合によって、かつ／または（同じピクチャ内、または隣接するピクチャ内の）現在のブロックに隣接するブロックの動きベクトルによって形成された候補のリスト（マージリストなど）内を探索することによって取得され得る。MV0およびMV1は、その後、ビットストリーム内でデコーダ側に有利にシグナリングされる。しかしながら、一般に、エンコーダでの第1の決定段階もまたテンプレート整合によって実行される場合があり、これにはシグナリングオーバーヘッドを低減する利点があることに留意されたい。

【0084】

デコーダ側200において、動きベクトルMV0およびMV1は、ビットストリーム内の情報に基づいて有利に取得される。MV0およびMV1は、直接シグナリングされるか、もしくは差別的にシグナリングされるかのいずれかであり、かつ／または動きベクトルのリスト（マージリスト）内のインデックスがシグナリングされる。しかしながら、本開示は、ビットストリーム内で動きベクトルをシグナリングすることに限定されない。むしろ、動きベクトルはエンコーダの動作に対応して、既に第1の段階でテンプレート整合によって決定されてもよい。第1の段階（動きベクトル導出）のテンプレート整合は、第2の精密化段階の探索空間とは異なる探索空間に基づいて実行されてもよい。具体的には、精密化は、解像度がより高い（すなわち探索位置間の距離がより短い）探索空間で実行されてもよい。

【0085】

それぞれのMV0およびMV1が指し示す2つの参照ピクチャRefPic0およびRefPic1の指示は、同様にデコーダにも提供される。参照ピクチャは以前の処理、すなわちエンコードおよびデコードのそれぞれの結果として、エンコーダ側およびデコーダ側の両方で、デコード済みピクチャバッファ内に記憶される。これらの参照ピクチャのうちの1つが、探索によって動きベクトル精密化のために選択される。エンコーダおよびデコーダの参照ピクチャ選択ユニットはそれぞれ、MV0が指し示す第1の参照ピクチャと、MV1が指し示す第2の参照ピクチャとを選択するように構成される。選択に続いて、参照ピクチャ選択ユニットは、第1の参照ピクチャまたは第2の参照ピクチャのどちらが動きベクトル精密化の実行のために使用されるかを決定する。動きベクトル精密化を実行するために、動きベクトルMV0が指し示す候補位置の周りに、第1の参照ピクチャ内の探索領域が定義される。探

10

20

30

40

50

索領域内の候補探索空間位置は、探索空間内でテンプレート整合を実行し、差分絶対値の和（SAD）などの類似性メトリックを決定することによって、テンプレートブロックに最も類似するブロックを見つけるために分析される。探索空間の位置は、テンプレートの左上隅が一致する位置を表す。既に上述したように、左上隅は単なる慣例であり、整合位置を表すために、中心点などの探索空間の任意の点が一般的に使用される場合がある。

【0086】

上述した文献JVET-D0029によれば、デコーダ側の動きベクトル精密化（DMVR）は、入力として初期動きベクトルMV0およびMV1を有し、これらは2つの各参照ピクチャRefPict0およびRefPict1を指し示す。これらの初期動きベクトルは、RefPict0およびRefPict1の各探索空間を決定するのに使用される。さらに、動きベクトルMV0およびMV1を使用して、MV0およびMV1によって指し示される、（サンプルの）各ブロックAおよびBに基づいてテンプレートが構築され、以下になる。

テンプレート=関数（（ブロックA，ブロックB））

【0087】

関数は、サンプルに対する加重和と組み合わせたサンプルクリッピング演算であってもよい。次いで、テンプレートは、各参照ピクチャ0および1内のMV0およびMV1に基づいて決定された探索空間で、テンプレート整合を実行するために使用される。各探索空間における最良のテンプレート整合を決定するためのコスト関数はSAD（Template, Block candA'）であり、block candA'は、MV0によって与えられた位置にまたがる探索空間内で、候補MVによって指し示される候補符号化ブロックである。図3は、最もよく整合するブロックA'の決定と、その結果生じる精密化動きベクトルMV0'とを示す。これに対応して、図3に示すように、最もよく整合するブロックB'と、ブロックB'を指し示す対応する動きベクトルMV1'とを見つけるために、同じテンプレートが使用される。言い換えれば、初期動きベクトルMV0およびMV1によって指し示されたブロックAおよびBに基づいてテンプレートが構築された後に、テンプレートでRefPict0およびRefPict1を探索することによって、精密化動きベクトルMV0'およびMV1'が見つけれられる。

【0088】

動きベクトル導出技術は、フレームレートアップ変換（frame rate up-conversion、FRUC）とも呼ばれる場合がある。エンコーダおよびデコーダが動きベクトル精密化に確実に同じ初期ポイントを使用し得るように、ビットストリームでは一般に初期動きベクトルMV0およびMV1が示される。あるいは、初期動きベクトルは、1つ以上の初期候補を含む初期候補のリストを提供することによって取得されてもよい。それらのそれぞれに対して精密化動きベクトルが決定され、最終的には、コスト関数を最小化する精密化動きベクトルが選択される。

【0089】

本発明は、図3を参照して上述したテンプレート整合に限定されないことにさらに留意されたい。図4は、片予測にも適用可能な代替的なテンプレート整合を示す図である。詳細については文献JVET-A1001で見ることができ、特に、Jianle Chenらによる文献JVET-A1001の「Section 2. 4. 6. Pattern matched motion vector derivation」は「Algorithm Description of Joint Exploration Test Model 1」というタイトルがつけられ、サイト<http://phenix.it-sudparis.eu/jvet/>でアクセスすることができる。このテンプレート整合手法におけるテンプレートは、現在のフレーム内の現在のブロックに隣接するサンプルとして決定される。図1に示すように、現在のブロックの上と左の境界に隣接する、既に再構築されたサンプルが取られてもよく、「L字型テンプレート」と呼ばれる。

【0090】

図5は、さらに使用され得る、別の種類の動きベクトル導出を示す。動きベクトル導出プロセスに対する入力は、動きベクトル導出が適用されるかどうかを示すフラグである。暗黙的に、導出プロセスに対する別の入力は、（時間的または空間的に）隣接する、以前に符号化／再構築されたブロックの動きベクトルである。複数の隣接するブロックの動きベクトルが、動きベクトル導出の初期探索ステップの候補として使用される。プロセスの

出力はそれぞれ、MV0'（双予測が使用される場合はMV1'の可能性もある）、および対応する参照ピクチャインデックスrefPict0、そしておそらくはrefPict1である。次いで、動きベクトル精密化段階は、上述したようにテンプレート整合を含む。精密化された1つ（片予測）、または2つ以上（双予測／多フレーム予測）の動きベクトルを見つけた後に、（加重サンプル予測による、またはそうでない場合は精密化されたMVによって指し示されるサンプルを参照することによる、双／多予測のための）現在のブロックの予測子が構築される。

【0091】

本発明は、上述した2つの整合方法（テンプレート整合の例）に限定されない。例として、双方向整合と呼ばれる第3の整合方法（文献JVET-A1001でも説明されている）が、動きベクトル精密化に使用されてもよく、本発明が同様に適用される。双方向整合は、図6を参照して説明されている。

【0092】

双方向整合によれば、2つの異なる参照ピクチャにおいて、現在のブロックの動き軌道に沿った、2つのブロック間の最良の整合が探索される。図6に示すように、双方向整合は、2つの異なる参照ピクチャ内で、現在のブロックの動き軌道に沿って2つのブロックの最も近い整合を見つけることによって、現在のブロックの動き情報を導出するために使用される。双方向整合では、MV0によってBlock cand0'が指し示され、MV1によってBlock cand1'が指し示される場所でSAD（Block cand0'，Block cand1'）などのコスト関数が使用され得る。

【0093】

計算費用を削減するために、動きベクトルの適切な候補（候補ペア）の選択についてはコスト関数が計算されて比較され、これは以下の条件を満たすことを考慮するうえで有意義であろう。連続的な動き軌道を前提として、2つの参照ブロックを指し示している動きベクトルMV0およびMV1は、現在のピクチャと2つの参照ピクチャとの間の時間的距離、すなわちTD0およびTD1に比例するものとする。特殊な事例として、現在のピクチャが時間的に2つの参照ピクチャ間にあり、現在のピクチャから2つの参照ピクチャまでの時間的距離が同じであるときは、1つの参照ピクチャを参照している複数ペアの動きベクトルと、他の参照ピクチャを参照しているペアの各動きベクトルとの動きベクトルの差分は、互いにミラーリングするものとする。

【0094】

図7には、処理回路600が示されている。処理回路は任意のハードウェアを含んでもよく、その構成は、任意の種類プログラミング、またはこれらの両方を組み合わせたハードウェア設計によって実施されてもよい。例えば、処理回路は、上述したステップを実施する対応ソフトウェアを含む汎用プロセッサなどの、単一のプロセッサによって形成されてもよい。他方では、処理回路は、DSP（デジタル信号プロセッサ）のASIC（特定用途向け集積回路）、またはFPGA（フィールドプログラマブルゲートアレイ）などの、特殊化されたハードウェアによって実施されてもよい。

【0095】

処理回路は、上述した動きベクトル導出を実行するために相互接続された、上述したハードウェアコンポーネントの1つ以上を含んでもよい。処理回路600は、初期動きベクトル（または双予測／多予測が使用される場合は複数の初期動きベクトル）、およびテンプレート整合が使用される場合はテンプレートを取得すること610と、動きベクトル精密化620との、2つの機能を実施する計算論理を含む。これら2つの機能は同じハードウェアで実施されてもよく、または初期動きベクトルおよびテンプレート決定ユニット610、ならびに動きベクトル精密化ユニット620などの、ハードウェアの個別のユニットで実行されてもよい。処理回路600は、再構築された参照ピクチャサンプルが記憶される外部メモリ650に、通信可能に接続されてもよい。さらに、処理回路600は、外部メモリから転送され、現在処理されているブロックの動きベクトル決定に使用される、ウィンドウ内のサンプルをバッファリングする内部メモリ640をさらに含んでもよい。処理回路は、集積回路

として1つのチップに具体化されてもよい。

【0096】

処理回路は、図1および図2を参照して説明した、エンコーダおよび／またはデコーダの別の機能を実施し得ることに留意されたい。内部メモリは、キャッシュまたはラインメモリなどのオンチップメモリであってもよい。チップメモリは、好適には、計算を高速化するためにエンコーダ／デコーダチップで実施される。チップのサイズが制限されているため、オンチップメモリは通常は小型である。一方、外部メモリは非常に大きいサイズにすることができるが、外部メモリへのアクセスにより多くのエネルギーを消費し、アクセスがかなり遅くなる。通常は、計算が実行される前に、必要な情報のすべてが外部メモリからオンチップメモリに取り込まれる。最悪の場合の外部メモリアクセス（またはメモリバスの設計時に提供される必要がある帯域幅）は、フレームまたは符号化ユニットのデコード中に、外部メモリとチップとの間で生じる可能性がある最大メモリ転送量を表す。メモリ（特に外部メモリ）は、通常は事前定義されたブロックユニット内でのみアクセスされ得る。言い換えれば、通常は単一のピクセルにはアクセスできず、最小単位（例えば、 8×8 ）でアクセスされねばならない。オンチップメモリが大きいと費用が高くなるので、オンチップメモリのサイズもまた重要な設計上の考慮事項である。

10

【0097】

言い換えれば、上述した装置は、集積回路内に埋め込まれた内部メモリと、前記ウィンドウ内に配置された整数サンプルを外部メモリから内部メモリに取り込むメモリアクセスユニット（インターフェース）とをさらに備える集積回路であってもよい。

20

【0098】

上述において使用されている「予測ブロック」という用語は、予測される現在のブロックを指す。画像を等しい大きさに、または異なるサイズに細分化する（例えば、符号化ツリーユニットの階層的区分化によって、CTUをより小さいユニットブロックにする）ことによって取得され得るのは、画像内のブロックである。ブロックは、現在のエンコーダ／デコーダにも使用される典型的な形状であるため、正方形、またはより一般的には長方形であってもよい。しかしながら、本開示は、ブロックのいかなるサイズ／形状によっても限定されない。

【0099】

処理回路を備える装置はエンコーダまたはデコーダであってもよく、あるいは、そのようなエンコーダまたはデコーダ、例えば、記録デバイスおよび／または再生デバイスを備える装置であってもよい。

30

【0100】

分数サンプル位置は、図1に示すように符号化された参照ピクチャを再構築することによって得られた、実際のピクチャサンプル位置同士の間での位置である。したがって分数位置は、最も近い整数位置に基づいて、補間によって取得されねばならない。補間フィルタリングの詳細は、文献「V. Sze et. al., High Efficiency Video Coding (HEVC), Springer, 2014」の「Section 5.3 Fractional Sample Interpolation」で見ることができる。

【0101】

補間フィルタリングは、通常は、別の分数ペル（サンプル）位置を生成するために、別のフィルタに適用される。一例として、以下の1D分割可能フィルタは、H.265ビデオ圧縮標準における $1/4$ ペル位置および $1/2$ ペル位置を生成するために適用される。

40

【0102】

【表1】

位相	ルマフィルタ係数
$1/4$	$[-1, 4, -10, 58, 17, -5, 1]/64$
$1/2$	$[-1, 4, -11, 40, 40, -11, 4, -1]/64$

50

【0103】

上記の表に見られるように、補間フィルタリングは、フィルタタップ（表内の係数の数）に対応する、分数ペル位置の周囲のいくつかのサンプルを必要とする。1/2ペル位置を生成するために、上述した例示的なフィルタを使用するには、左/上、および右/下から4個の整数サンプルが必要になる。補間フィルタの長さは、1/4ペルサンプル位置（7タップ）では、1/2ペルサンプル位置（8タップ）とは異なることに留意されたい。サンプル、ペル、およびピクセルという用語は、本願では交換可能に使用され、所与の時間インスタンスにおける画像サンプル（1つ以上の色成分における）を表す。サンプルは、原則として強度値である。

【0104】

本発明のいくつかの実施形態では、事前定義されたタップサイズの補間フィルタは、前記探索空間でのテンプレート整合のためにアクセス可能な整数サンプルによって与えられたウィンドウ内でのみ、整数サンプルを評価する。ウィンドウは、一定の予測ブロックの計算に実際に使用されるよりも、ずっと多くのサンプルを含む場合がある。これは、精密化探索動作が、通常は（総当たりの探索方法とは対照的な）高速探索法を使用して実施されるためであり、これによると一部のサンプルは、探索動作の段階的な進行に応じた評価はされない。その結果、テンプレート整合の反復回数が、精密化探索動作のための計算に使用されるサンプルと同様に、予測ブロックごとに変化する場合がある。

【0105】

図8は、符号化ブロック（予測ブロック）、およびウィンドウの対応するサンプルを示す。図8に示すサンプルは参照ピクチャサンプルであり、ここでの符号化ブロックは、実際には、現在のフレーム内の現在のブロックにサイズおよび位置が対応するブロックであり、これに対して参照ピクチャ内で動きベクトルが導出されることに留意されたい。したがって、実際には図8の符号化ブロックは、要するに予測子が探索されるブロックと同一位置にあるブロックである。しかしながら、簡潔にするために、このブロックは以下「符号化ブロック」と呼ばれる。

【0106】

この例では、精密化されていない動きベクトルMV0は整数サンプル位置を指し示している。動きベクトル精密化の探索粒度は1整数サンプルであり、始点が整数サンプルなので整数サンプル点のみが探索されることを意味する。この例では、徐々に展開する探索空間で探索が実行される。これは、以前にテストされた位置のコスト関数の観点から、最良の方向に応じて新しい探索位置を追加することによって、各反復において探索空間が進むことを意味する。

【0107】

このような手法については、図9に簡略化して示されている。図9では、初期動きベクトルが中心点810を指し示している。探索空間は、初期動きベクトル位置の周囲に徐々に構築される。第1のステップでは、初期動きベクトルによって指し示されている位置810に対して、すぐ上、すぐ下、すぐ左、およびすぐ右に隣接する4つの位置が、初期動きベクトルによって指し示されている位置810と同様にテストされる。テストされた5つの点の中でコスト関数が最低になる方向に基づいて、テストされるべき別の位置が探索空間に追加される。この例では、最低コスト関数は右の点で見出されたので、第2のステップでは、探索空間は水平方向右にさらに点3つ分拡張された。第2のステップでは、最低コスト関数は、（第1のステップの最低コスト点に対して）右の点で見出され、探索空間が水平方向右に点3つ分、さらに拡張する結果となる。第3のステップでは、最低コスト関数がここでもステップ2の最低コスト点に対して右の点で見出され、探索空間が水平方向右にさらに点3つ分拡張する結果となる。図9の例によれば、上方向、上方向、右方向の順で、さらに3つのステップが実行される。例では、各反復に対して、（5つの探索点からなる）ひし形のパターンが使用され、各ステップで欠けている探索点を完成させるために探索空間が拡張される。

【0108】

探索空間決定の各反復において、探索空間は1つ以上の整数サンプル位置によって増加し得る。ここで図8に戻ると、この例では、探索の反復の最大回数は4である。最大4回の反復が可能なので、探索動作を実行するためには左に示されているすべてのサンプルがメモリから取り込まれる必要があり、この場合、探索空間の展開は徐々に左へ進む。同様に、4つのサンプルが上に拡張することが必要になる。したがって、精密化されたMVはいずれかの方向に移動できるので、探索空間は両方向（左から右、および上から下）に拡張され、ハードウェアの実施には、精密化探索が適用される前に、必要になる可能性があるすべてのサンプルが外部メモリから取り込まれることが必要になる。探索空間が下方向または右方向に展開する場合は、符号化ブロック（予測ブロック）のサイズに対応するテンプレートを用いたテンプレート整合が、これらのサンプルの一部にアクセスすることが必要になるため、さらにサンプル4つ分拡張される必要がある。さらに、ハードウェアの実施では、通常は（矩形アクセスのほうがより実行可能であって）不規則な形状は取り込めないため、隅（例えば、右上）にあるサンプルもメモリから取り込まねばならない。

【0109】

上記で説明した反復的な探索空間展開は単なる例であって、各反復において探索空間を拡張するためのルールおよび点の数は異なる場合がある、すなわち違った方法で指定される場合があることに留意されたい。

【0110】

図9もまた、上述した外部メモリアクセスルールによって生じ得るシナリオを示す。外部メモリから取り込まれるサンプルの数は、計算ステップで実際に使用されるサンプルよりもずっと多い。ここでのテンプレートが、（簡略化するために）サンプル1つ分の大きさしかないと仮定すると、白丸は外部メモリから取り込まれたサンプルを表し、影付きのサンプルは実際に使用されたものを表す。しかしながら、現在のブロックの処理が開始された時点では実際に必要なサンプルがまだわからないため、外部メモリへのアクセス数を少なく維持する必要がある場合は、このような冗長性が必要とされる。

【0111】

探索空間は別の方法で、例えば、初期動きベクトルによって指し示された位置に配置された安定した形状として、定義されてもよいことに留意されたい。形状は、正方形、長方形、ひし形などの任意の形状であってもよい。

【0112】

図10は、探索空間が分数サンプルをさらに含み得る例を示す。図8および図9では、動きベクトル探索は整数サンプルで実行され、その結果、アクセスウィンドウに含まれる実線の大きい点で示される位置が得られた。ここでは1/2ペル解像度を有するサンプル（実線の小さいほうの点）に対して探索が実行され、左側に示されている分数サンプルを生成するために、さらに3列のサンプルがメモリから取り込まねばならず、補間フィルタは対称であって8つのタップを有すると仮定される。さらに、探索動作が対称である（左と右とに反復的に移動できる）ことから、左側（ピクセル3列分の拡張）にも同じことを適用する必要があり、その結果、ウィンドウの左側に分数ピクセルが配置され得る。

【0113】

その結果、補間フィルタリングによって、メモリから取り込むべきサンプル数がさらに増加し、破線で示されているように、分数補間によって追加された位置を表す点線の円も含まれるようになる。同様に、垂直方向の位置ごとに半分ずつ探索できるようにすると、メモリからアクセスされるサンプルのウィンドウは、上側および下側で垂直方向にも拡張されることが必要になる（図10の例には示されていない）。

【0114】

メモリアクセスのウィンドウは、予測ブロック（符号化ブロック）の動きベクトル探索を実行するためにメモリから取り込まれる必要がある、すべてのサンプルを囲む長方形として定義される。メモリアクセスのウィンドウは、必要とされる実際のサンプルを含むだけでなく、動きベクトル探索動作中にアクセスされる可能性がある、すべての残りのサンプルも含んでいる。図9の例では、動きベクトル探索は右に移動していた。しかし、左方

向にも移動していた可能性があり、これは事前にはわからない。したがって、何度も外部メモリにアクセスしないように、メモリアクセスのウィンドウ（またはアクセスウィンドウ）は、各処理でアクセス可能な全サンプルを含む。

【0115】

これを実現するため、本開示は、例えば、分数サンプルを補間するために、必要な場合はメモリアクセスウィンドウの外部に配置されたサンプルのパディングを実行する。

【0116】

置換サンプル（パディングサンプル）を提供することによって、分数サンプルをさらに含む探索空間において、テンプレート整合との関連がさらに有利になる場合がある。言い換えれば、予測ブロックの探索空間は、事前定義されたタップサイズのフィルタを用いた補間フィルタリングによって取得された値を伴う、少なくとも1つの分数サンプル位置を含んでもよい。補間フィルタリングは、少なくとも1つの分数サンプル位置の値を取得するために、ウィンドウ、および少なくとも1つの置換値からのサンプル位置の値を使用する。

【0117】

図11は、先に概説したように動きベクトル導出／精密化を実行する、知られている手法の基本的なステップを示す。第1のステップ（ステップ1）において、少なくとも2つの候補動きベクトルの「探索空間」が取得される（図示されている例では、9つの候補動きベクトルがある）。任意で、探索空間を形成している候補動きベクトルの1つが、初期動きベクトルとみなされてもよい（本願で使用される用語によれば、「アンカ」動きベクトル、またはアンカ候補動きベクトルとも呼ばれる）。初期またはアンカ動きベクトルは、例えば、ビットストリームに含まれ、デコーダに対して明示的または暗黙的にシグナリングされる動きベクトルであってもよい。

【0118】

次のステップ（ステップ2）において、探索空間の最もよく整合する点を見つけるために、探索空間に整合プロセスが適用される（言い換えれば、探索空間の点はコスト関数を最小化し、最も適切な候補動きベクトルに対応する）。図示されている例では双方向整合が適用されているが、テンプレート整合などの、コスト関数を決定する他の方式も同様に適用され得る。

【0119】

図面の最後のステップ（ステップ3）は、ステップ2の処理に従って選択された最も適切な候補動きベクトルに基づく補間によって、最終予測が取得される。

【0120】

図11の概念は、図12のフロー図でさらに説明される。図12では、特に、候補動きベクトルペアの事例、すなわち双方向整合に対する図が示されている（テンプレート整合が、例えば、図3を参照して上述したような双方向予測に使用されるときにも適用可能である）。

【0121】

第1のステップ（1310）において、候補動きベクトルペアの探索空間が構築される。このステップは、図11の「ステップ1」に相当する。

【0122】

次のステップ（1320）において、補間フィルタリングを適用することによって、探索空間内の各動きベクトルペアに対する予測が生成される。次のステップ（1330）において、探索空間内の各動きベクトルペアに対して、テンプレートが構築（取得）される。上述したようにこのステップは任意であって、すなわちテンプレート整合の場合にのみ必要であり、双方向整合では必要とされない。次のステップ（1340）において、探索空間内の各動きベクトルペアに対して、ペアの2つの動きベクトル（または1つの候補動きベクトルの予測ブロック、およびテンプレート）によって指し示された予測ブロックを比較することによって、テンプレート整合コスト（または双方向整合コスト）が計算される。次のステップ（1370）では、テンプレートまたは双方向コスト関数を最小化する（より一般

的には、最もよく整合する、すなわち整合プロセスによって最も適切なものとして取得される）、結果として得られた動きベクトルペアが選択され、予測ブロックの動きベクトルペアとして出力する。上述した用語によれば、これは「精密化動きベクトルペア」と呼ばれてもよい。前述のステップ1320、1330、1340、および1370は、図11の「ステップ2」に相当する。

【0123】

次のステップ（1380）は、ステップ1370で選択した動きベクトルペアを用いた補間によって、最終予測を生成する。このステップは、図11の「ステップ3」に相当する。

【0124】

図13は、本発明における例示的な実施形態による、動きベクトル精密化および最終予測生成のための処理を示す。図13の手法と、図11に示す従来の手法との主な違いは、本発明の図13による手法は、動きベクトル精密化、および補間フィルタリングを用いた予測子の最終生成の両方に対して、サンプルパディングを使用することにある。

【0125】

（任意で）動きベクトル精密化にパディングされたサンプルを使用することが追加のステップ2に示されており、これについては図14から図18を参照しながら以下でより詳しく説明する。これによって、分数座標を有する候補動きベクトル（すなわち探索空間の分数座標）の場合であっても、双方向整合またはテンプレート整合を実行するために所望される候補ブロックは、事前定義された制限のあるアクセスウィンドウを使用することによって実行でき、これは整数点のみを含む探索空間の場合と同じである。双方向整合またはテンプレート整合を適用することによるコスト関数の実際の導出はステップ3で実行され、これは図11のステップ2と同じである。

【0126】

さらに、サンプルのパディングは、実際の予測子生成を実行するために、最後のステップ4で（ここでも、またはここでのみ）使用される。このステップは概して図11のステップ3に対応するが、パディングの概念を使用するところが異なっている。具体的には、所与の例では、最右の2列、および最下段の行に配置されたサンプル値はアクセスされないが、アクセスされた他のサンプル値に基づいて、パディングサンプル値でパディングされる。

【0127】

パディングされる（使用不可のため置換値で置換される）サンプルの数は、最後に選択された（「精密化された」）動きベクトルと、参照初期動きベクトル（「アンカ」）との差分に依存する。図示されている例では、この差分は垂直方向に2ピクセル、水平方向に1ピクセルである。これに従って、ステップ5の図では、2つの長方形の交点にあるピクセルを水平方向および垂直方向に補間するために、ピクセルの右側の2列、および最終行のピクセルのサンプル値は使用できず、上述した事前定義ルールに従って、置換値で置換（パディング）される必要があり、これについては以下でさらに詳しく説明する。

【0128】

好適には、ステップ2における探索にもパディングが使用される場合、ステップ2およびステップ4の両方の補間フィルタリング動作に同じアクセスウィンドウが使用される。

【0129】

好適には、図13のステップ2およびステップ4で使用されているメモリアksesウィンドウは、探索空間内の特定の候補動きベクトルに基づいて決定される。特定の動きベクトル（アンカ動きベクトルと呼ばれる）は、例えば、ビットストリームに含まれ、デコーダに対して明示的または暗黙的にシグナリングされる動きベクトルであってもよい。アンカ動きベクトルの選択決定は、エンコーダおよびデコーダの両方に周知されている事前定義ルールに従って実施される。

【0130】

図14は、動きベクトル精密化を目的とした、メモリアksesのウィンドウを示す。ウィンドウは、初期動きベクトルによって指し示されている位置である、中央にある点1010

10

20

30

40

50

の周囲に広がっている。さらに、分数位置1020も探索空間に属している。しかしながら、分数位置1020は、分数位置の左側の3個の整数サンプル位置と、分数位置の右側の3個の整数サンプル位置とを使用する、6個のタップフィルタによって補間される。分数位置1020は整数サンプル位置と同一直線上に配置されるので、一次元フィルタが補間に適用されてもよく、例えば、図10を参照して説明した上述の例に示したようなものになる。図14に見られるように、分数サンプル1020を補間するために2つのさらに別のサンプル位置1030が必要になり、これらはウィンドウの外部に配置される。これらのサンプルを外部メモリからさらにロードするのを避けるために、本開示によれば、ウィンドウ外位置1030の値は、ウィンドウ内のサンプルに基づいて取得された値と置換される。

【0131】

図14は、2つの整数サンプル位置同士の間中にあり、したがって $1/2$ ピクセル ($1/2$ ペル) 位置にある、分数サンプル位置1020を示す。しかしながら、本開示は、 $1/4$ または $1/8$ 位置などの他の分数位置にも等しく適用可能である。さらに、サンプル位置1020は、整数サンプル位置の行に対応する水平線上に配置され、その結果、その補間に水平1Dフィルタリングが使用され得る。しかしながら、本開示は、そのような分数位置に限定されない。分数位置は、整数サンプル位置の列に対応する垂線上に配置されてもよく、その結果、その補間に垂直1Dフィルタリングが使用され得る。さらに、分数位置は、整数サンプルと同じピクセルの直線上に配置される必要はまったくない。このような場合は、このようなサンプルを補間するために二次元フィルタリングが使用されてもよく、このような2Dフィルタリングは、垂直1Dフィルタリングと水平1Dフィルタリングとに分割可能であってもよい。

【0132】

図16に、さまざまな分数位置の例が示されている。特に、分数位置1は、水平フィルタリングおよび垂直フィルタリングの両方が適用され得る補間用の位置であり、これは、整数サンプル位置の行および列の同一線上に配置されていないためである。位置2は、水平フィルタリングのみが使用される補間用の位置であり、位置6は、垂直フィルタリングのみが使用される補間用の位置である。図16に示す分数位置はすべて、ウィンドウ内の整数位置のみを使用して取得され得る。

【0133】

メモリアクセス用のウィンドウは、さまざまな方法で定義されてもよい。図18は、メモリアクセスウィンドウが、符号化ブロックの左/右または上/下にある拡張部分EXTとして定義されている例を示す（初期動きベクトルによって与えられた符号化ブロックの位置に対応している）。拡張量は、符号化ブロックまたは予測ブロックのサイズおよび形状に依存する場合がある。図18において、拡張部分は各方向（上、下、左、右）にサンプル4個分の長さである。しかしながら、EXTは、ブロックサイズ（垂直方向および水平方向でサイズが異なる場合がある）、ならびに/または探索空間のフォームおよびサイズに応じて、異なる方向に対して異なる値をとる場合もあることに留意されたい。

【0134】

例えば、例によると、ウィンドウは、予測ブロック初期動きベクトルに対して、N整数サンプル列、およびM整数サンプル行として定義され、NおよびMは非ゼロ整数値になる。図18では、NとMとが示されているが、サイズは同じである。上述したように、NおよびMはサイズが異なってもよい。NおよびMは整数であり、そのうちの少なくとも1つは非ゼロである。パラメータNおよびM、ならびにテンプレートの形式およびサイズをとることにより、アクセスウィンドウのサイズが決定され得る。特に、テンプレートがT1行およびT2列を有する場合、メモリアクセスウィンドウのサイズは、 $(N+T1+N)$ 行および $(M+T2+M)$ 列と計算され得る。これは、探索が、サンプルN個分左または右に進むことが可能で水平に2Nサンプルとなり、サンプルM個分上または下に進むことが可能で垂直に2Mサンプルとなるからである。また、テンプレートは、 $2N \times 2M$ サンプル位置の任意の位置で、 $T1 \times T2$ サンプルと整合され得る。

【0135】

一方、図8および図9を参照して説明したような特定の探索空間構築手法に対し、メモリアクセスウィンドウは、精密化反復の最大数（探索空間構築反復）、および（各反復で達成可能な最大サンプル距離に関する）反復ステップサイズの観点から定義されてもよく、これは後に、左、右、上、および下への変位の最大量に変換され得る。したがって、メモリアクセスウィンドウは、各方向への最大変位として定義される。例えば、4回の反復において、それぞれの反復が探索空間を最大1整数サンプル位置分進め得る場合は、EXT=4となる。

【0136】

言い換えれば、この例によれば、処理回路は、直近の反復における探索空間の最もよく整合する位置の1つ（またはそれ以上）によって与えられた方向に、反復的に拡張された探索空間内での、前記テンプレートを用いたテンプレート整合によって初期動きベクトルの精密化を決定するように構成され、ウィンドウは、事前定義された反復の最大数によって定義される。双方向整合の場合にも、必要な変更を加えて同じことが適用される。

【0137】

一般に、本開示は、いかなる特定の形状もしくは形式、または探索空間の決定の種類にも限定されないことに留意されたい。別の例では、探索空間は、ウィンドウの長方形のサブウィンドウであり、またはウィンドウと等しくてもよい。対応するシグナリングパラメータ（例えば、パラメータEXT、またはパラメータNおよびM）をビットストリーム内に含める可能性があるため、メモリアクセスウィンドウの定義は重要な意味を持つ場合がある。しかしながら、メモリアクセスウィンドウのサイズは標準的に定義されてもよく、または他の符号化パラメータ（テンプレートサイズ、予測ブロックサイズ、画像解像度など）に基づいて導出可能であってもよい。図14に戻ると、実施形態によれば、精密化のためのメモリアクセスのウィンドウは、精密化されていない動きベクトルによって指し示される位置1010の周囲で定義される。ウィンドウは、動きベクトル精密化探索を実行するために、メモリからアクセスされる必要があるピクセルサンプルの最大数を識別する。ここでの探索空間は、アクセスウィンドウと一致する。言い換えれば、この例では簡略化するためにテンプレートサイズは1x1とされているが、通常はそれよりも大きくすることが可能であり、実際にそれよりも大きくなる。メモリアクセスのウィンドウは、通常は符号化ブロックの周囲の拡張部分として定義される。この例では、拡張部分は左／右から4サンプル分、かつ上／下から4サンプル分である。テンプレート整合または双方向整合中に現在テストされている探索空間位置が、メモリアクセスのウィンドウの外部からのサンプルを必要としている場合、符号化ブロックの再構築に必要なサンプルはパディングによって取得される。これは、図14の分数位置1020の事例である。テンプレート整合がこの位置で、またはこの位置を使用して実行される場合は、ウィンドウの外部に配置された別のパディング位置1030を使用して補間されねばならない。

【0138】

パディングサンプルは、例えば、以下の方法のうちの1つを使用して生成され得る。

直近サンプル複製。

補間フィルタリングの軸線に沿ったミラーリング。

【0139】

具体的には、直近サンプル複製は、アクセスされる整数サンプル位置に最も近い値と等しくなるように置換値が決定される手法を指す。例えば、図14において、2つの欠落位置の値1030は、補間される分数位置1020のすぐ右隣のサンプルの値と置換される。しかしながら、この手法は単なる例であり、本開示は近傍の単なる複製に限定されないことに留意されたい。あるいは、いくつかの直近のサンプル位置が、欠落サンプル1030を補間するのに使用されてもよい。例えば、欠落サンプルに最も近い境界上にある3つのサンプルが、例えば、3つのサンプルの欠落サンプルからの距離に反比例するように設定された重みを用いた加重平均によって、欠落サンプルを補間するために使用されてもよい。

【0140】

ミラーリングは、対応する評価済みの整数サンプル位置の値を、補間フィルタの軸線に

対してミラーリングすることによって、置換値が決定される手法を指す。例えば、図14において、2つの欠落サンプル1030は、分数位置1020の左にあるサンプルの値と置換される。具体的には、分数位置1020の右側の2つ目のサンプルは、分数位置1020の左側2つ目の位置の値に置き換えられ、分数位置1020の右側の3つ目のサンプルは、分数位置1020の左側3つ目位置の値に置き換えられる。

【0141】

欠落サンプルがミラーリングされたサンプルの関数であり、最も近いサンプルである場合は、上述した2つの手法の組み合わせも適用され得ることに留意されたい。

【0142】

実施形態によれば、ウィンドウは、前記探索空間内の整数サンプル位置において、テンプレート整合または双方向整合のためにアクセス可能なすべての整数サンプル位置によって形成される。これは例えば図14の例の場合であり、ウィンドウの外部にある追加のサンプルは、整数でない分数サンプル位置におけるテンプレート整合のためにのみ必要である。言い換えれば、図14において、上述の例では、探索される $1/2$ ピクセル座標は、メモリアクセスウィンドウの外部にあるサンプルを必要とする。結果として、使用不可のサンプルを生成するために、パディング操作が適用される。ただし、本発明はこれに限定されない。

【0143】

図15は、ウィンドウが、前記探索空間内の整数サンプル位置におけるテンプレート整合のためにアクセス可能なすべての整数サンプル位置を含まない、代替的な手法を示す。前述した例では、符号化ブロックサイズは、簡略化するために 1×1 で示されていた。図15では、サイズは 4×4 整数サンプルである。この例では、探索座標（テンプレートまたは双方向整合が計算される位置）が整数サンプルを指し示していても、ブロック内のサンプルの一部がメモリアクセスウィンドウの外部にある場合は、やはりパディングを適用することができる。この手法により、ウィンドウサイズをさらに制限することが可能になる。

【0144】

詳しく説明すると、図15は、いくつかのアクセス可能なサンプルを含む、ウィンドウ120を示している。初期動きベクトルは、符号化ブロックの左上隅をマークする位置を指し示している。探索空間は、ここではウィンドウに等しいとみなされてもよい。しかしながら、通常はウィンドウよりも小さくなる可能性があり、それも理にかなっている場合があることに留意されたく、その理由は、テンプレートが 1×1 整数サンプルよりも大きい場合は必然的に、テンプレート整合の実行に、探索空間に含まれているサンプルとは別のサンプルが必要になるためである。それにもかかわらず、探索空間がウィンドウのサイズを定義する場合は、代替的に欠落サンプルがパディングされてもよい。図15では、ウィンドウの2行10列目の例示的な探索位置が示されている。 4×4 整数サンプルの大きさのテンプレートを用いるテンプレート整合は、ウィンドウの外部にあるピクセルをさらに必要とする。これらのサンプルは、したがってパディングされる、すなわちウィンドウ内のサンプルに基づいて決定または計算された値で置換される。最近傍またはミラーリングなどの、上述したパディングの手法が適用されてもよい。

【0145】

ウィンドウは、

探索空間、または

初期動きベクトル位置に配置された予測ブロックのサイズを有するブロック、または初期動きベクトル位置、

に対するその垂直方向サイズおよび／または水平方向サイズで定義される。

【0146】

ウィンドウ定義の例は図18に示されており、上述されている。これは、初期動きベクトル位置に配置された予測ブロックのサイズに基づく、ウィンドウサイズの決定に対応する。あるいは、ウィンドウサイズは、x方向およびy方向のうちの1つまたは両方のサイズによって、初期ベクトル位置に対して定義されてもよい。例えば、ウィンドウは、初期動きベクトルによって指し示され、上、下、左、右の各方向へA整数サンプル分拡張する位置

の周囲に配置されるように定義されてもよい。あるいは、左および右にそれぞれA整数サンプル分、かつ上および下にB整数サンプル分拡張してもよい。別の定義も可能である。

【0147】

上述の例は、 $1/2$ ペル補間用に提供されていたことに留意されたい。しかしながら、本開示はこれに限定されない。一般に、 $1/4$ 、 $1/8$ などの任意の分数位置が使用されてもよく、すなわち対応する補間フィルタを使用して補間される。

【0148】

ウィンドウサイズと同じ定義が、予測子生成のための補間フィルタリング用のパディングをする事例にも適用される。具体的には、垂直方向サイズおよび水平方向サイズ（すなわちxおよびy方向）は、予測子を決定するために使用される補間フィルタの、垂直方向長さおよび水平方向長さによって決定される。

【0149】

図17は、図13に示されている処理をさらに詳細に示すフロー図である。図17のフロー図は、従来のプロセスを示す図12のフロー図とは異なっており、図12のステップ1320および1380は、それぞれステップ1720および1780にそれぞれ置き換えられる。これらのステップは、双方向整合またはテンプレート整合を実行するために（複数の）参照ピクチャ内で候補予測ブロックを生成するための補間フィルタリング、および最終予測を生成するための補間フィルタリングの両方について、置換サンプルを使用したパディングが適用されるという点が図12の各ステップとは異なる。

【0150】

図7を参照して説明した処理回路が、図1および図2に示すようなエンコーダおよび／またはデコーダで使用されてもよい。

【0151】

特に、予測ブロックに分割されたビデオ画像をビットストリームにエンコードするためのエンコード装置が提供されてもよく、エンコード装置は、処理回路を備える、上述した予測ブロックに対する動きベクトルを決定するための装置と、予測ブロックと、決定された動きベクトルによって指定された位置で予測ブロックによって与えられた予測子との差分をエンコードし、エンコードされた差分および初期動きベクトルを含むビットストリームを生成するためのエンコード回路とを備える。

【0152】

図1を参照して上述したエンコーダの別のユニットおよび機能もまた、処理回路において提供または実施され得る。

【0153】

これに対応して、デコード装置は、ビットストリームから、予測ブロックに分割されたビデオ画像をデコードするために提供され、デコード装置は、ビットストリームから、初期動きベクトルと、予測ブロックと精密化動きベクトルによって指定された位置で予測ブロックによって与えられた予測子とのエンコードされた差分とを解析する解析ユニットと、処理回路を備える、上述した予測ブロックに対する精密化動きベクトル決定のための装置と、解析した差分と、精密化動きベクトルに基づく位置で、予測ブロックによって与えられた予測子の合計として予測ブロックを再構築するためのデコード回路とを備える。例えば、予測子は、精密化動きベクトルの位置によって直接与えられてもよい。しかしながら、現在の予測ブロックの動きベクトルを取得する別の処理ステップがあってもよく、これによって動きベクトルをさらに変化させてもよい（フィルタリング、クリッピング、別の精密化など）。

【0154】

図2を参照して上述したデコーダの別のユニットおよび機能もまた、処理回路において提供または実施され得る。

【0155】

さらに、本発明の実施形態は、動きベクトル精密化を実行するための処理回路を備えた装置の観点から説明された。しかしながら、本開示はこれに限定されないだけでなく、対

10

20

30

40

50

応する方法を提供し、この方法は、上述した処理回路が構成される、実行するためのステップに対応する処理ステップを含む。

【0156】

タップは、フィルタ係数に対応する。タップサイズは、フィルタ順に対応する。ここで、フィルタは線形フィルタと仮定される。いくつかの例において、フィルタは対称であってもよく、すなわち対称係数を有する。しかしながら、本開示は、対称フィルタもしくは線形フィルタ、またはいかなる種類のフィルタにも限定されない。一般に、分数位置は、隣接するサンプルに基づいて任意の方法で取得されてもよい。

【0157】

実施形態による方法は、図19に示されている以下のステップで、予測ブロックに対する動きベクトルを決定する。ステップ1410において、予測ブロックの初期動きベクトルおよびテンプレートが取得され、次に、探索空間における前記テンプレートを用いたテンプレート整合によって、初期動きベクトルの精密化を決定する。探索空間は、初期動きベクトルによって与えられた位置に配置される。1420において、メモリアクセスウィンドウが決定される。これは、明示的に実行される必要はないことに留意されたい。メモリアクセスウィンドウは、事前定義され、知られていてもよい。したがって、このステップは、メモリアクセスウィンドウサイズの知識が知られていることを示すために図示するものである。次に、ステップ1430から1490においてテンプレート整合が実行される。

【0158】

テンプレート整合または双方向整合は、事前定義されたウィンドウ内の整数サンプル位置と、前記探索空間を含む事前定義されたウィンドウと、テンプレート整合または双方向整合のためにアクセス可能な別の整数サンプル位置とにアクセスし、ウィンドウの外部に配置された少なくとも1つの整数サンプル位置を、前記ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって置換する。アクセスされた整数サンプル位置の値および置換値は、テンプレート整合を実行するために使用される。これは図19に示されており、テンプレート整合または双方向整合は、探索空間内のすべての位置で実行される。ステップ1430において、現在位置でのテンプレート整合または双方向整合が、メモリアクセスウィンドウの外部にある1つまたは複数のサンプルを必要とするかどうかがテストされる。肯定された場合（ステップ1430がyesの場合）は、次に、ステップ1440において、ウィンドウの外部にある1つまたは複数のサンプルに対して上述したようなパディングが実行される。このステップの後、ステップ1450において、パディングされたサンプルを用いて、現在位置でテンプレート整合または双方向整合（すなわち探索）が行われる。あるいは、ウィンドウの外部に必要なサンプルがない場合（ステップ1430がNoの場合）は、ステップ1450において、使用可能なサンプルを使用して、現在位置で、テンプレート整合または双方向整合（すなわち探索）が実行される。現在位置が、探索空間におけるテストされる最後の位置である場合（ステップ1460がyesの場合）は、テンプレート整合は終了する。そうでない場合（ステップ1460がNoの場合）は、ステップ1470において次の位置が取得され、ステップ1430において、上述したように新しい位置に対してテンプレート整合が継続する。

【0159】

テンプレート整合が終了すると、（例えば、最低コストを有する記憶された位置として）最もよく整合する位置が識別される。

【0160】

図20は、本発明における実施形態の枠組み内で適用可能な、動きベクトル精密化、および予測子生成の代替的な例示的手法を示す。この手法と図13の手法との違いは、図20の場合は、パディングを伴う補間フィルタリングが、精密化動きベクトル（ペア）を使用して、予測子の最終生成のみに適用される点にある。これは図20のステップ5であり、図13のステップ4に対応する。

【0161】

一方、動きベクトル精密化（探索）、すなわち図13のステップ2には、パディングを伴

10

20

30

40

50

う補間フィルタリングは適用されない。これに代えて、図20の手法では、初期動きベクトル（または初期探索空間を形成している複数の初期候補動きベクトル）が最も近い整数点を指している場合は、最も近い整数点まで座標を丸めるために丸め演算を行う（図20のステップ2）。この丸め演算は、丸め方向を表すMV_round_diffで示される、事前定義されたベクトルの移動と同じになる。丸め演算を行うことにより、探索空間が、整数座標を含む点のみを有する、修正された探索空間を取得するように確実に修正される。結果として、双方向（またはテンプレート）整合によって、最もよく整合する整数点を取得する次のステップ3は整数座標で実行でき、その結果、テンプレート／双方向整合に対する予測の生成に補間フィルタリングは必要とされない。

【0162】

さらに、図20はステップ4を含む。このステップは任意である。ステップ4において、修正された探索空間における最もよく整合する整数点がステップ3で取得された後で、（初期の）探索空間内で最終的な精密化動きベクトルを取得するように、丸めの適用が元に戻される。言い換えれば、ステップ2とは反対方向に移動するベクトルが、ステップ3の結果生じた、最もよく整合する（精密化）動きベクトルに適用される。任意のステップ4が含まれる場合は、その後、ステップ4の結果生じた精密化動きベクトルに基づいて最終予測が取得される。あるいは、ステップ3は省略され、ステップ3の結果生じた動きベクトルにステップ5が直接適用される。さらに代替的には、ステップ4を含めるか、それとも省略するかを所定の条件に基づいて決めることができる。

【0163】

図20の手法に対応するフロー図が、図21に示されている。ここでもまた、フロー図は、単一の動きベクトルではなく動きベクトルペア用に作成されており、これらは図20では図を簡略化する目的で使用されている。

【0164】

最初のステップ1310は、図20と同じである。後続のステップ2115は、図20のステップ2の丸め演算に相当する。初期探索空間の修正が丸め演算に限定されない限りにおいて、本明細書で示す処理は一般的なものであるが、本明細書では、より一般的には、初期探索空間における各動きベクトルペアに対して、代表的な動きベクトルペアが生成されるものとする。図20の丸め演算は、代表的なものに対し、このように初期探索空間動きベクトルをマッピングする特殊な事例である。

【0165】

次のステップ2130（テンプレート整合に対して任意）および2140は、図17におけるものと同じであり、各動作が、修正済みの探索空間内の代表的な動きベクトルで実行される点のみが異なっている。ステップ2150および2160は、図20のステップ4に相当する。ステップ2150は、テンプレート整合コスト（または双方向整合コスト）をPair_best_repとして最小化する、代表的な動きベクトルペアを設定するステップを示す。

【0166】

ステップ2160（任意）は、図20のステップ4に示す、丸め演算を元に戻すことに相当する。次のステップ（1370）において、元の探索空間の動きベクトル（ペア）が、予測ブロックに対する動きベクトルとして出力される。ステップ1370および1780は、図17と同じである。

【0167】

さらに、予測ブロックに分割されたビデオ画像をビットストリームにエンコードするための、エンコード方法が提供され、エンコード方法は、上述した任意の方法に従って予測ブロックに対する動きベクトルを決定するステップと、決定した動きベクトルに基づく位置において、予測ブロックと、予測ブロックによって与えられた予測子との差分をエンコードし、エンコードされた差分と初期動きベクトルとを含むビットストリームを生成するステップとを含む。

【0168】

エンコード方法は、図1のブロックの機能を参照して説明したステップをさらに含んで

10

20

30

40

50

もよい。

【0169】

さらに、デコード方法は、ビットストリームから、予測ブロックに分割されたビデオ画像をデコードするために提供され、デコード方法は、ビットストリームから、初期動きベクトルと、予測ブロックと精密化動きベクトルによって指定された位置で予測ブロックによって与えられた予測子とのエンコードされた差分とを解析するステップと、上述した任意の方法に従って、予測ブロックに対する精密化動きベクトルを決定するステップと、解析した差分と、精密化動きベクトルによって指定された位置で予測ブロックによって与えられた予測子との合計として、予測ブロックを再構築するステップとを含む。

【0170】

デコード方法は、図2のブロックの機能を参照して説明したステップをさらに含んでもよい。

【0171】

しかしながら、図1および図2は、本開示を限定するためのものではないことに留意されたい。これらは単に、既存のエンコーダおよび／またはデコーダ内での、本発明の実施の限定されない例を提供するものである。

【0172】

上述したようなサンプルパディングを用いた動きベクトル決定は、ビデオ信号（動画）のエンコードおよび／またはデコードの一部として実施され得る。しかしながら、動きベクトル決定は、エンコード／デコードのための使用に限定されることなく、動き検出、動き解析などの、画像処理における他の目的にも使用され得る。

【0173】

動きベクトル決定は、装置として実施され得る。このような装置は、ソフトウェアとハードウェアの組み合わせであり得る。例えば、動きベクトル決定は、汎用プロセッサ、またはデジタル信号プロセッサ（DSP）、またはフィールドプログラマブルゲートアレイ（FPGA）などのチップによって実行されてもよい。しかしながら、本発明は、プログラマブルハードウェア上での実施に限定されない。それは、特定用途向け集積回路（ASIC）上でまたは上で言及されたハードウェアコンポーネントの組み合わせによって実施され得る。

【0174】

動きベクトル決定は、コンピュータ可読記憶媒体に記憶されたプログラム命令によって実施されてもよい。プログラムが実行されると、コンピュータは上述した方法のステップを実行する。コンピュータ可読記憶媒体は、DVD、CD、USB（フラッシュ）ドライブ、ハードディスク、ネットワーク経由で利用可能なサーバストレージなど、プログラムが記憶された任意の媒体であってもよい。

【0175】

エンコーダおよび／またはデコーダは、TVセット、セットトップボックス、PC、タブレット、またはスマートフォンなどのさまざまなデバイスにおいて実施されてもよく、すなわち任意の録画デバイス、符号化デバイス、コード変換デバイス、デコードデバイス、または再生デバイスなどのデバイスである。これは方法ステップを実施するソフトウェアまたはアプリであってもよく、上述したような電子機器に含まれるプロセッサで記憶／実行されてもよい。

【0176】

要約すると、本開示は、テンプレート整合または双方向整合を使用した動きベクトル決定、および動きベクトルに基づく予測子生成に関する。テンプレート整合もしくは双方向整合、および／または予測子生成は、補間フィルタリングを使用する。補間フィルタリング動作は、ウィンドウ内の整数サンプル位置にアクセスし、前記探索空間と、テンプレート整合のためにアクセス可能な別の整数サンプル位置とを含むウィンドウは、前記ウィンドウ内の少なくとも1つのサンプルに基づくパディングされたサンプル値によって、ウィンドウの外部に配置された少なくとも1つの整数サンプル位置をパディングし、テンプレート整合または双方向整合と、予測子生成とを実行するために、アクセスされた整数サン

10

20

30

40

50

ブル位置の値およびパディングされたサンプル値を使用する。

【0177】

本発明の別の実施形態が、以下で提供される。以下の項で使用されている番号付けは、前述の項で使用した番号付けに必ずしも従う必要はないことに留意されたい。

【0178】

実施形態1. ビデオ画像の予測ブロックをビットストリーム(171)にエンコードするためのエンコード装置であって、エンコード装置は、

少なくとも1つの初期動きベクトルを取得し、

候補動きベクトルの探索空間に基づいて、初期動きベクトルの精密化を決定して、精密化動きベクトルを取得し、前記探索空間は、初期動きベクトルによって与えられた位置(810)に配置される、

10

ことによって、予測ブロック(165)に対する動きベクトルを決定し、

予測ブロックと決定した動きベクトルに基づいた位置で予測ブロックによって与えられた予測子との差分(105)をエンコードして、エンコードされた差分と初期動きベクトルとを含むビットストリーム(171)を生成し、

補間フィルタを用いた補間フィルタリングを使用して、動きベクトルに従って予測子を決定し、前記補間フィルタリングは、

事前定義されたウィンドウ内の整数サンプル位置でサンプル値にアクセスし、

ウィンドウの外部に配置された整数サンプル位置で、前記ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって少なくとも1つのサンプル値を置換し、

20

アクセスされた整数位置サンプル値および置換値を使用して、補間フィルタリングを実行する、

ように構成された処理回路(600)を備える、

エンコード装置。

【0179】

実施形態2. ビットストリーム(171)から、予測ブロックに分割されたビデオ画像をデコードするためのデコード装置であって、デコード装置は、

精密化動きベクトルによって指定された位置で、ビットストリームから、初期動きベクトルと、予測ブロックと予測ブロックによって与えられた予測子とのエンコードされた差分とを解析し、

30

少なくとも1つの初期動きベクトルを取得し、

候補動きベクトルの探索空間に基づいて、初期動きベクトルの精密化を決定して、精密化動きベクトルを取得し、前記探索空間は、初期動きベクトルによって与えられた位置(810)に配置される、

ことによって、予測ブロックに対する動きベクトルを決定し、

処理回路(600)によって決定された、精密化動きベクトルによって指定された位置で、解析された差分と、予測ブロックによって与えられた予測子との合計として予測ブロックを再構築して、動きベクトルを決定する、

ように構成された処理回路を備え、

前記処理回路は、補間フィルタを用いた補間を使用して、精密化動きベクトルに従って、予測子を決定するようにさらに構成され、前記補間は、

40

事前定義されたウィンドウ内の整数サンプル位置でサンプル値にアクセスし、

ウィンドウの外部に配置された整数サンプル位置で、前記ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって、少なくとも1つのサンプル値を置換し、

アクセスされた整数サンプル位置の値および置換値を使用して、補間を実行する、

デコード装置。

【0180】

実施形態3. 置換値が、対応する評価済み整数サンプル位置の値を、補間フィルタの軸線に対してミラーリングすることによって決定される、実施形態1または2に記載の装置。

【0181】

50

実施形態4. 補間フィルタが一次元フィルタまたは分離可能な二次元フィルタである、実施形態1から3のいずれか1つに記載の装置。

【0182】

実施形態5. ウィンドウが、初期動きベクトルによって指定された位置で、予測ブロックによって与えられた予測子を取得するためにアクセス可能なすべての整数サンプル位置によって形成される、実施形態1から4のいずれか1つに記載の装置。

【0183】

実施形態6. ウィンドウが、初期動きベクトル位置に配置された予測ブロックのサイズを有するブロック、初期動きベクトル位置、および予測子を決定するために使用される、補間フィルタの垂直方向長さおよび水平方向長さのうちの少なくとも1つに対する、その垂直方向サイズおよび／または水平方向サイズによって定義される、実施形態1から5のいずれか1つに記載の装置。

【0184】

実施形態7. 置換値が、アクセスされた整数サンプル位置に最も近い値と等しくなるように決定される、実施形態1、2、または4から6のいずれか1つに記載の装置。

【0185】

実施形態8. 置換されるサンプル値の数が、初期動きベクトルと精密化動きベクトルとの差分に依存する、実施形態1から7のいずれか1つに記載の装置。

【0186】

実施形態9. 前記処理回路(600)が、テンプレートを用いたテンプレート整合または双方向整合によって、初期動きベクトルの精密化を決定するように構成され、

前記テンプレート整合または前記双方向整合が、

別の事前定義されたウィンドウ(1120)内の整数サンプル位置にあるサンプル値にアクセスし、別の事前定義されたウィンドウ(1120)が、テンプレート整合または双方向整合のためにアクセス可能な、前記探索空間と、別の整数サンプル位置とを含み、

別のウィンドウの外部に配置された整数サンプル位置で、前記別のウィンドウ内の少なくとも1つのサンプルに基づいた置換値によって少なくとも1つのサンプル値(1030)を置換し、

アクセスされた整数サンプル位置の値および置換値を使用して、テンプレート整合または双方向整合を実行する、実施形態1から8のいずれか1つに記載の装置。

【0187】

実施形態10. 前記探索空間が、事前定義されたタップサイズのフィルタを用いた補間フィルタリングによって取得された値を伴う、少なくとも1つの分数サンプル位置(1, 2, 6)を含み、

補間フィルタリングが、前記アクセスされた整数サンプル位置の値と、少なくとも1つの置換値とを使用して、前記少なくとも1つの分数サンプル位置の値を取得する、実施形態9に記載の装置。

【0188】

実施形態11. テンプレート整合または双方向整合で使用される置換値が、対応するアクセスされた整数サンプル位置の値を、フィルタの軸線に対してミラーリングすることによって決定される、実施形態10に記載の装置。

【0189】

実施形態12. テンプレート整合または双方向整合で使用される置換値が、アクセスされた整数サンプル位置に最も近い値と等しくなるように決定される、実施形態9または10に記載の装置。

【0190】

実施形態13. 別の事前定義されたウィンドウ(1120)が、探索空間、

初期動きベクトル位置に配置された予測ブロックのサイズを有するブロック、および

10

20

30

40

50

初期動きベクトル位置

のうちの少なくとも1つに対する、その垂直方向サイズおよび／または水平方向サイズで定義される、実施形態9から12のいずれか1つに記載の装置。

【0191】

実施形態14．前記別の事前定義されたウィンドウ（1120）が、前記事前定義されたウィンドウと同じである、実施形態9から13のいずれか1つに記載の装置。

【0192】

実施形態15．前記精密化が双方向整合によって決定され、前記処理回路が、異なる参照ピクチャ（Ref0, Ref1）内の位置を指し示し、初期動きベクトル（Pair_0, Pair_1, …, Pair_N）のペアを形成する、少なくとも2つの初期動きベクトルを取得するように構成される、実施形態1から14のいずれか1つに記載の装置。

10

【0193】

実施形態16．前記精密化がテンプレート整合によって決定され、前記処理回路が、予測ブロックのテンプレートを取得し、前記探索空間における前記テンプレートを用いたテンプレート整合によって、初期動きベクトルの精密化を決定する

ようにさらに構成される、実施形態1から14のいずれか1つに記載の装置。

【0194】

実施形態17．予測ブロックに分割されたビデオ画像をビットストリームにエンコードするためのエンコード方法であって、エンコード方法は、

20

少なくとも1つの初期動きベクトルを取得するステップ（1310, 1410）と、

候補動きベクトルの探索空間に基づいて、初期動きベクトルの精密化を決定して、精密化動きベクトルを取得するステップ（1320, 1330, 1340, 1720; 2115, 2120, 2130, 2140, 2150, 2160）であって、前記探索空間は、初期動きベクトルによって与えられた位置に配置される、ステップ（1320, 1330, 1340, 1720; 2115, 2120, 2130, 2140, 2150, 2160）と

を含む、予測ブロックに対する動きベクトルを決定するステップと、

決定された動きベクトルに基づく位置で、予測ブロックと予測ブロックによって与えられた予測子との差分をエンコードし、エンコードされた差分と初期動きベクトルとを含むビットストリームを生成するステップと、

30

補間フィルタを用いた補間フィルタリングを使用して、動きベクトルに従って予測子を決定するステップ（1780）であって、前記補間フィルタリングは、

事前定義されたウィンドウ内の整数サンプル位置でサンプル値にアクセスし、

ウィンドウの外部に配置された整数サンプル位置で、前記ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって少なくとも1つのサンプル値を置換し、

アクセスされた整数位置サンプル値および置換値を使用して、補間フィルタリングを実行する、ステップ（1780）と

を含む、エンコード方法。

【0195】

実施形態18．ビットストリームから、予測ブロックに分割されたビデオ画像をデコードするためのデコード方法であって、デコード方法は、

40

精密化動きベクトルによって指定された位置で、ビットストリームから、初期動きベクトルと、予測ブロックと予測ブロックによって与えられた予測子とのエンコードされた差分とを解析するステップと、

少なくとも1つの初期動きベクトルを取得するステップ（1310, 1410）と、

候補動きベクトルの探索空間に基づいて、初期動きベクトルの精密化を決定して、精密化動きベクトルを取得するステップ（1320, 1330, 1340, 1720; 2115, 2120, 2130, 2140, 2150, 2160）であって、前記探索空間は、初期動きベクトルによって与えられた位置に配置される、ステップ（1320, 1330, 1340, 1720; 2115, 2120, 2130, 2140, 2150, 2160）と

50

を含む、予測ブロックに対する動きベクトルを決定するステップと、
動きベクトルを決定するステップによって決定された、精密化動きベクトルによって指定された位置で、解析された差分と、予測ブロックによって与えられた予測子との合計として予測ブロックを再構築するステップと、
補間フィルタを用いた補間を使用して、精密化動きベクトルに従って、予測子を決定するステップ（1780）であって、前記補間は、
事前定義されたウィンドウ内の整数サンプル位置でサンプル値にアクセスし、
ウィンドウの外部に配置された整数サンプル位置で、前記ウィンドウ内の少なくとも1つのサンプルに基づく置換値によって、少なくとも1つのサンプル値を置換し、
アクセスされた整数位置サンプルおよび置換値を使用して、補間を実行する、ステップ（1780）と
を含む、デコード方法。

10

【0196】

実施形態19. 置換値が、対応する評価済みの整数サンプル位置の値を、補間フィルタの軸線に対してミラーリングすることによって決定される、実施形態17または18に記載の方法。

【0197】

実施形態20. 補間フィルタが一次元フィルタまたは分離可能な二次元フィルタである、実施形態17から19のいずれか1つに記載の方法。

【0198】

実施形態21. ウィンドウが、初期動きベクトルによって指定された位置で、予測ブロックによって与えられた予測子を取得するためにアクセス可能なすべての整数サンプル位置によって形成される、実施形態17から20のいずれか1つに記載の方法。

20

【0199】

実施形態22. ウィンドウが、
初期動きベクトル位置に配置された予測ブロックのサイズを有するブロック、
初期動きベクトル位置、および
予測子を決定するために使用される、補間フィルタの垂直方向長さおよび水平方向長さのうちの少なくとも1つに対する、その垂直方向サイズおよび／または水平方向サイズによって定義される、実施形態17から21のいずれか1つに記載の方法。

30

【0200】

実施形態23. 置換値が、アクセスされた整数サンプル位置に最も近い値と等しくなるように決定される、実施形態17、18、または20から22のいずれか1つに記載の方法。

【0201】

実施形態24. 置換されるサンプル値の数が、初期動きベクトルと精密化動きベクトルとの差分に依存する、実施形態17から23のいずれか1つに記載の方法。

【0202】

実施形態25. 精密化を決定する前記ステップ（1320, 1330, 1340, 1720; 2115, 2120, 2130, 2140, 2150, 2160）が、テンプレートをを用いたテンプレート整合または双方向整合によって、初期動きベクトルの精密化を決定し、

40

前記テンプレート整合または前記双方向整合が、
別の事前定義されたウィンドウ内の整数サンプル位置にあるサンプル値にアクセスし、
別の事前定義されたウィンドウが、テンプレート整合または双方向整合のためにアクセス可能な、前記探索空間と、別の整数サンプル位置とを含み、

別のウィンドウの外部に配置された整数サンプル位置で、前記別のウィンドウ内の少なくとも1つのサンプルに基づいた置換値によって少なくとも1つのサンプル値を置換し（1440）、

アクセスされた整数サンプル位置の値および置換値を使用して、テンプレート整合または双方向整合を実行する（1450）、実施形態17から24のいずれか1つに記載の方法。

【0203】

50

実施形態26．前記探索空間が、事前定義されたタップサイズのフィルタを用いた補間フィルタリングによって取得された値を伴う、少なくとも1つの分数サンプル位置を含み、補間フィルタリングが、前記アクセスされた整数サンプル位置の値と、少なくとも1つの置換値とを使用して、前記少なくとも1つの分数サンプル位置の値を取得する、実施形態25に記載の方法。

【0204】

実施形態27．テンプレート整合または双方向整合で使用する置換値が、対応する評価済みの整数サンプル位置の値を、フィルタの軸線に対してミラーリングすることによって決定される、実施形態26に記載の方法。

【0205】

実施形態28．テンプレート整合または双方向整合で使用する置換値が、アクセスされた整数サンプル位置に最も近い値と等しくなるように決定される、実施形態26または27に記載の方法。

【0206】

実施形態29．別の事前定義されたウィンドウが、探索空間、初期動きベクトル位置に配置された予測ブロックのサイズを有するブロック、および初期動きベクトル位置のうちの少なくとも1つに対する、その垂直方向サイズおよび／または水平方向サイズで定義される、実施形態25から28のいずれか1つに記載の方法。

【0207】

実施形態30．前記別の事前定義されたウィンドウが、前記事前定義されたウィンドウと同じである、実施形態25から29のいずれか1つに記載の方法。

【0208】

実施形態31．前記精密化が双方向整合によって決定され、取得する前記ステップが、異なる参照ピクチャ内の位置を指し示し、初期動きベクトルのペアを形成する、少なくとも2つの初期動きベクトルを取得する、実施形態17から30のいずれか1つに記載の方法。

【0209】

実施形態32．前記精密化がテンプレート整合によって決定され、動きベクトルを決定する前記ステップが、

予測ブロックのテンプレートを取得するステップ（1330；2130）と、前記探索空間における前記テンプレートを用いたテンプレート整合によって、初期動きベクトルの精密化を決定するステップ（1340；2140）とをさらに含む、実施形態17から31のいずれか1つに記載の方法。

【0210】

実施形態33．プロセッサで実行されると、プロセッサに実施形態17から32のいずれか1つに記載の方法を実行させる命令を記憶したコンピュータ可読記憶媒体。

【符号の説明】

【0211】

- 100 エンコーダ
- 101 ピクチャブロック
- 102 入力
- 104 残差計算
- 105 残差ブロック
- 106 変換ユニット
- 107 変換係数
- 108 量子化ユニット
- 109 量子化係数
- 110 逆量子化ユニット
- 111 逆量子化係数

10

20

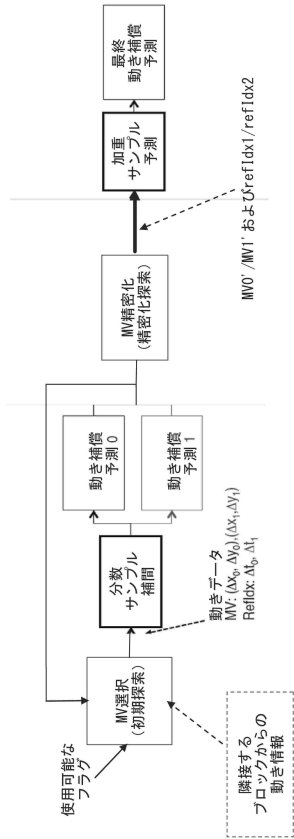
30

40

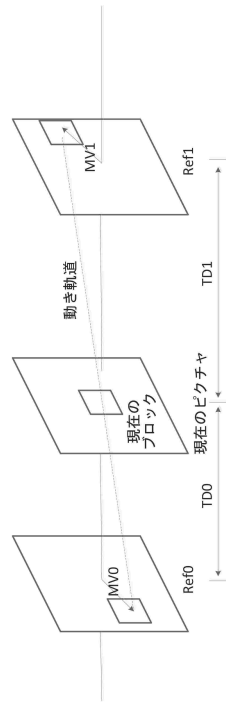
50

112	逆変換ユニット	
113	逆変換されたブロック	
114	再構築	
115	再構築されたブロック	
116	バッファ	
117	参照サンプル	
120	ループフィルタリングユニット	
121	フィルタリングされたブロック	
130	デコード済みピクチャバッファ	
131	デコードされたピクチャ	10
142	インター推定ユニット	
144	インター予測ユニット	
152	イントラ推定ユニット	
154	イントラ予測ユニット	
160	モード選択ユニット	
165	予測ブロック	
170	エントロピーエンコードユニット (エントロピー符号化ユニット)	
171	エンコードされたピクチャデータ (エンコードされたビットストリーム)	
172	出力	
200	デコーダ	20
202	入力	
204	エントロピーデコードユニット	
209	量子化された残差変換係数	
210	逆量子化ユニット	
211	逆量子化変換係数	
212	逆変換ユニット	
213	逆変換されたブロック	
214	再構築	
215	再構築されたブロック	
216	バッファ	30
217	参照サンプル	
220	ループフィルタリングユニット	
221	フィルタリングされたブロック	
230	デコード済みピクチャバッファ	
231	デコードされたピクチャ	
232	出力	
244	インター予測ユニット	
254	イントラ予測ユニット	
260	モード選択	
265	予測ブロック	40
600	処理回路	
610	テンプレート決定ユニット	
620	動きベクトル精密化ユニット	
640	内部メモリ	
650	外部メモリ	
810	中心点	
1010	精密化されていない動きベクトルによって指し示される位置	
1020	分数位置	
1030	別のサンプル位置	
1120	別の事前定義されたウィンドウ	50

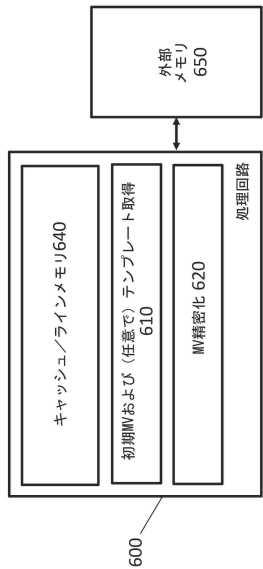
【図 5】



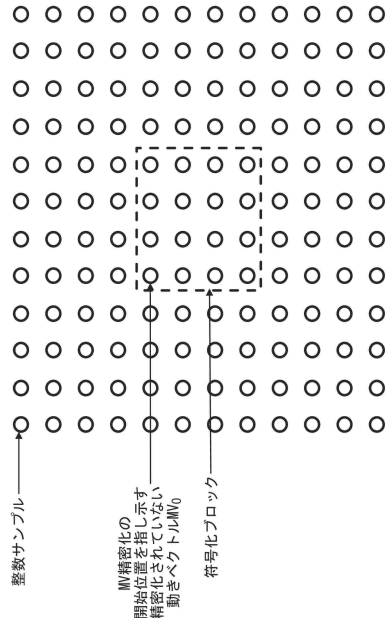
【図 6】



【図 7】



【図 8】



10

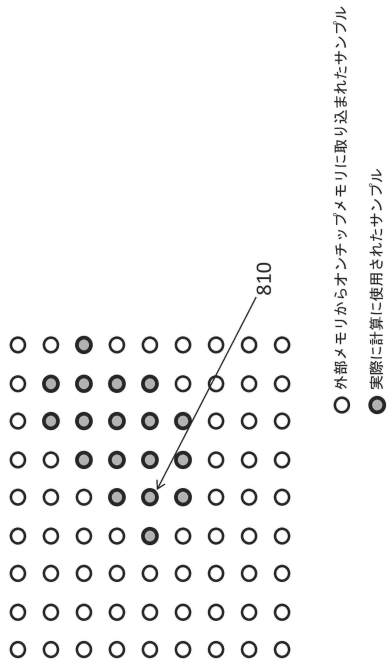
20

30

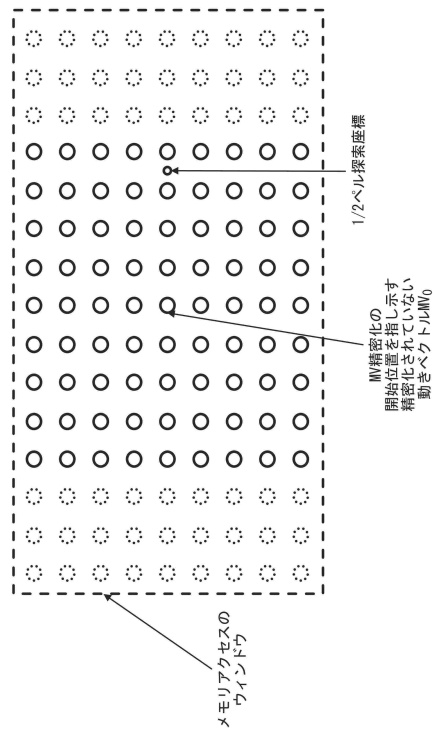
40

50

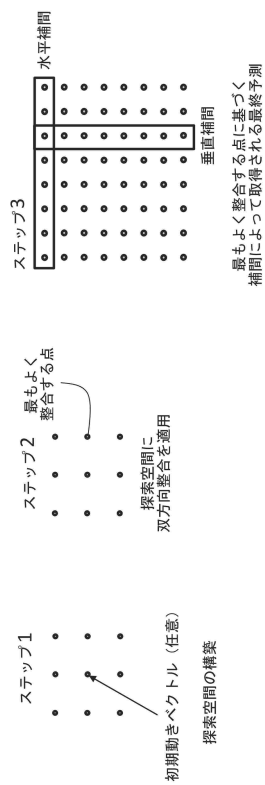
【図 9】



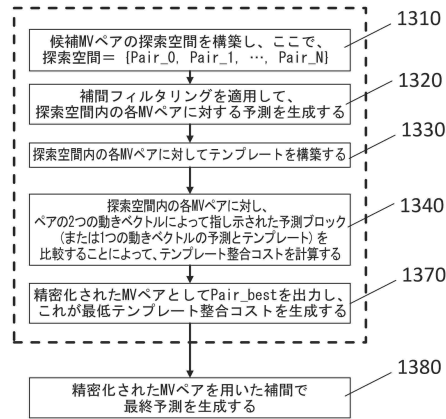
【図 10】



【図 11】



【図 12】



10

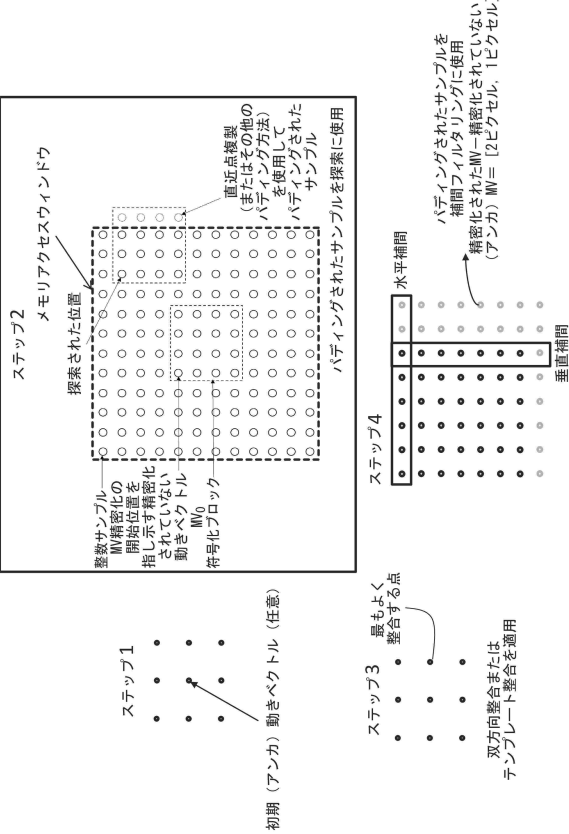
20

30

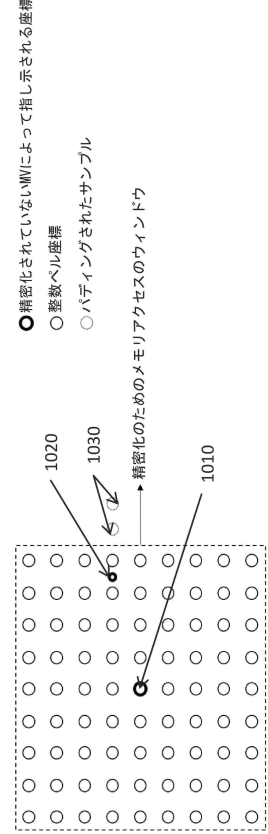
40

50

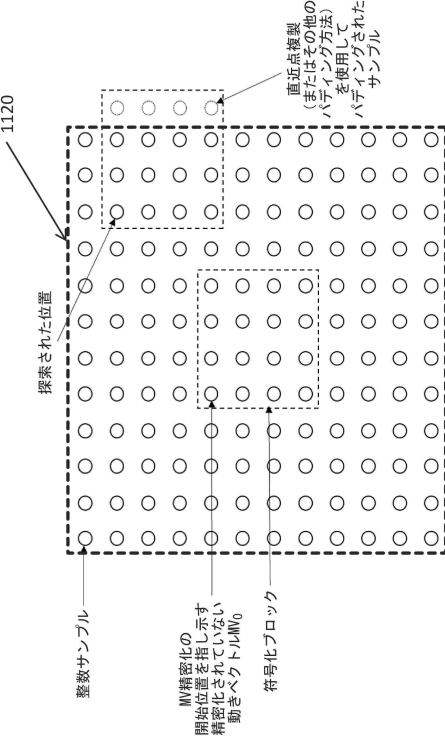
【図 13】



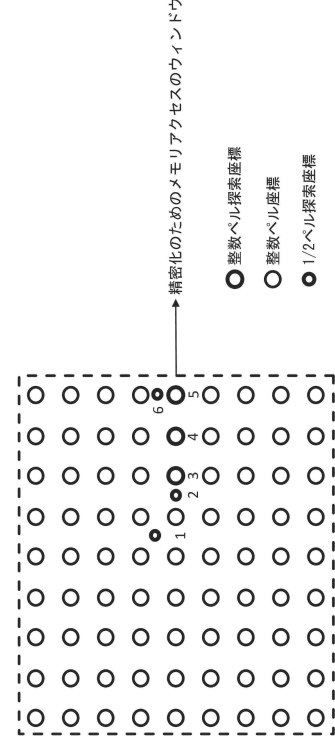
【図 14】



【図 15】



【図 16】



10

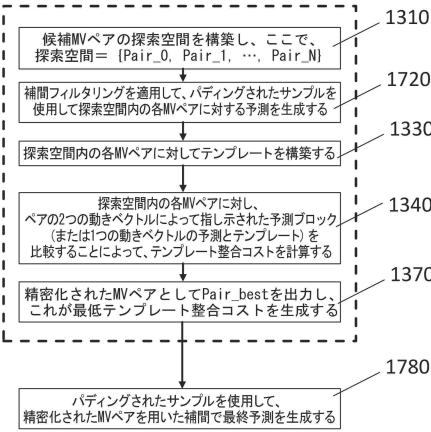
20

30

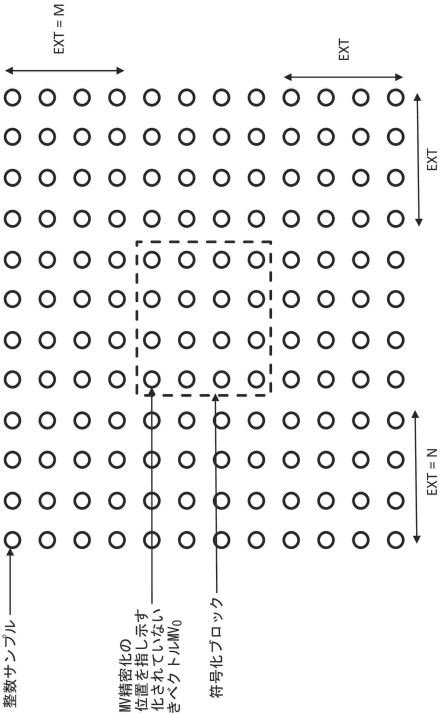
40

50

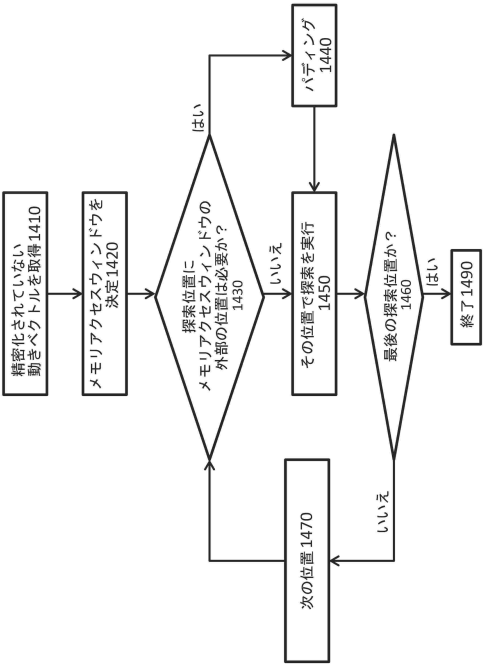
【図 17】



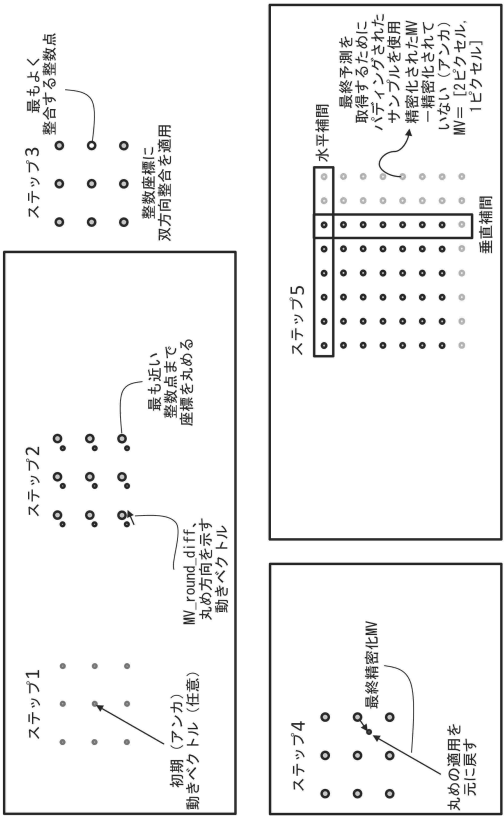
【図 18】



【図 19】



【図 20】



10

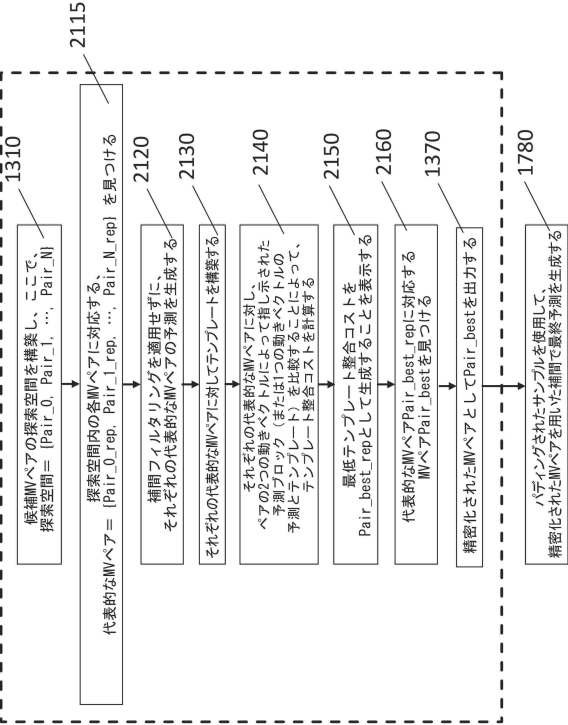
20

30

40

50

【図 21】



10

20

30

40

50

フロントページの続き

- 米国(US)
クノロジーズ・デュッセルドルフ・ゲーエムベーハー
- (72)発明者 イヴァン・クラスノフ
ドイツ・80992・ミュンヘン・リースシュトラッセ・25・ホアウェイ・テクノロジーズ・デュッセルドルフ・ゲーエムベーハー
- (72)発明者 ジジエ・ジャオ
ドイツ・80992・ミュンヘン・リースシュトラッセ・25・ホアウェイ・テクノロジーズ・デュッセルドルフ・ゲーエムベーハー
- (72)発明者 ジエンレ・チェン
アメリカ合衆国・カリフォルニア・95050・サンタ・クララ・セントラル・エクスプレスウェイ・2330
- 審査官 鉢呂 健
- (56)参考文献 CHEN, Jianle et al., Algorithm Description of Joint Exploration Test Model 7 (JEM 7), Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 7th Meeting: Torino, IT, 13-21 July 2017, [JVET-G1001-v1], JVET-G1001 (version 1), ITU-T, 2017年08月19日, <URL:http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/7_Torino/wg11/JVET-G1001-v1.zip>: JVET-G1001-v1.docx: pp.23-27
CHEN, Huanbang et al., Description of SDR, HDR and 360 ° video coding technology proposal by Huawei, GoPro, HiSilicon, and, buJoint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 10th Meeting: San Diego, US, 10-20 Apr. 2018, [JVET-J0025_v2], JVET-J0025 (version 4), ITU-T, 2018年04月14日, <URL:http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/10_San%20Diego/wg11/JVET-J0025-v4.zip>: JVET-J0025_v3.docx: pp.66-69
ESENLIK, Semih, et al., CE9: Report on the results of tests CE9.2.15 and CE9.2.16, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, 3-12 Oct. 2018, [JVET-L0163], JVET-L0163 (version 1), ITU-T, 2018年09月24日, <URL:http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/12_Macao/wg11/JVET-L0163-v1.zip>: JVET-L0163.DOCX: pp.1-6
- (58)調査した分野 (Int.Cl., DB名)
H04N 19/00-19/98
IEEE Explore