

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5687341号
(P5687341)

(45) 発行日 平成27年3月18日(2015.3.18)

(24) 登録日 平成27年1月30日(2015.1.30)

(51) Int.Cl.

F I

HO 4 N	19/11	(2014.01)	HO 4 N	19/11
HO 4 N	19/136	(2014.01)	HO 4 N	19/136
HO 4 N	19/159	(2014.01)	HO 4 N	19/159
HO 4 N	19/176	(2014.01)	HO 4 N	19/176
HO 4 N	19/196	(2014.01)	HO 4 N	19/196

請求項の数 14 (全 29 頁) 最終頁に続く

(21) 出願番号	特願2013-519828 (P2013-519828)
(86) (22) 出願日	平成23年7月14日(2011.7.14)
(65) 公表番号	特表2013-534797 (P2013-534797A)
(43) 公表日	平成25年9月5日(2013.9.5)
(86) 国際出願番号	PCT/US2011/044014
(87) 国際公開番号	W02012/009540
(87) 国際公開日	平成24年1月19日(2012.1.19)
審査請求日	平成25年12月25日(2013.12.25)
(31) 優先権主張番号	61/364,322
(32) 優先日	平成22年7月14日(2010.7.14)
(33) 優先権主張国	米国 (US)
(31) 優先権主張番号	61/388,541
(32) 優先日	平成22年9月30日(2010.9.30)
(33) 優先権主張国	米国 (US)

(73) 特許権者	392026693
	株式会社 N T T ドコモ
	東京都千代田区永田町二丁目 1 1 番 1 号
(74) 代理人	100088155
	弁理士 長谷川 芳樹
(74) 代理人	100113435
	弁理士 黒木 義樹
(74) 代理人	100121980
	弁理士 沖山 隆
(74) 代理人	100128107
	弁理士 深石 賢治

早期審査対象出願

最終頁に続く

(54) 【発明の名称】 ビデオ符号化のための簡素化されたイントラ予測

(57) 【特許請求の範囲】

【請求項 1】

ビデオエンコーダのプロセッサによって実行されるビデオ符号化方法であって、
対象ブロックの直上に配置された水平境界ピクセルの配列と対象ブロックの直左に配置された垂直境界ピクセルの配列の一方から少なくともいくつかのピクセルを取り出すステップと、

取り出されたピクセルを他方の境界ピクセルに当該他方の境界ピクセルの配列方向に沿って追加して当該他方の境界ピクセルの配列を当該他方の境界ピクセルの配列方向に沿って拡張するステップと、

前記他方の境界ピクセルの配列方向に沿って拡張された他方の境界ピクセルの配列だけを参照して、イントラ予測を実行するステップと、

を、前記プロセッサによって実行する、ビデオ符号化方法。

【請求項 2】

前記取り出すステップが、垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理、を含み、

前記拡張するステップが、前記取り出されたピクセルを前記水平境界ピクセルに追加して前記水平境界ピクセルの前記配列を拡張する処理、を含む、

請求項 1 に記載のビデオ符号化方法。

【請求項 3】

前記垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理が、

10

20

【数 1】

$$\left[\frac{size \times col}{angle} \right]$$

によって表される垂直ピクセル識別子を使用して前記垂直境界ピクセルの中から前記少なくともいくつかのピクセルを識別することであって、ここで、*size*が、予測される対象ブロックのサイズを表し、*angle*が予測方向を表し、*col*が、-1から*angle*まで1ずつデクリメントされるカウンタである、当該識別すること、を含み、

前記取り出されたピクセルを前記水平境界ピクセルに追加する処理が、

10

前記垂直ピクセル識別子によって識別されたピクセルを、水平ピクセル識別子 [*col*] によって識別される位置で前記水平境界ピクセルに追加すること、を含む、

請求項 2 に記載のビデオ符号化方法。

【請求項 4】

前記垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理が、

InvAngle を

【数 2】

$$\frac{N \times size}{angle}$$

20

から計算することであって、ここで、*size*が、予測される対象ブロックのサイズを表し、*angle*が予測方向を表し、*N*が2の整数乗である、当該計算することと、

[*col* × *InvAngle* > > log₂ *N*] によって表される垂直ピクセル識別子を使用して、前記垂直境界ピクセルの中から前記少なくともいくつかのピクセルを識別することであって、ここで、*col*が、-1から*angle*まで1ずつデクリメントされるカウンタである、当該識別することと、

を含み、

前記取り出されたピクセルを前記水平境界ピクセルに追加する処理が、

30

前記垂直ピクセル識別子によって識別されたピクセルを、水平ピクセル識別子 [*col*] によって識別される位置で前記水平境界ピクセルに追加すること、を含む、

請求項 2 に記載のビデオ符号化方法。

【請求項 5】

前記垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理が、

angle の値に関連付けられた *InvAngle* の値を列挙するルックアップテーブルから *InvAngle* を獲得することであって、ここで、*angle* が予測方向を表す、当該獲得することと、

[*col* × *InvAngle*] を用いた関数によって表される垂直ピクセル識別子を使用して、前記垂直境界ピクセルの中から前記少なくともいくつかのピクセルを識別することであって、ここで、*col*が、-1から*angle*まで1ずつデクリメントされるカウンタである、当該識別することと、

40

を含み、

前記取り出されたピクセルを前記水平境界ピクセルに追加する処理が、

前記垂直ピクセル識別子によって識別されたピクセルを、水平ピクセル識別子 [*col*] によって識別される位置で前記水平境界ピクセルに追加すること、を含む、

請求項 2 に記載のビデオ符号化方法。

【請求項 6】

前記垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理が、

垂直ピクセル識別子 [*row*] を使用して、前記垂直境界ピクセルの中からピクセルを

50

識別することであって、ここで、*row*が、0から*size*まで1ずつインクリメントされるカウンタであり、*size*が、予測される対象ブロックのサイズを表す、当該識別すること、を含み、

前記取り出されたピクセルを前記水平境界ピクセルに追加する処理が、

前記垂直ピクセル識別子によって識別されたピクセルを、 $[int + 1]$ によって表される水平ピクセル識別子によって識別される位置で前記水平境界ピクセルに追加することであって、ここで、*int*が、予測方向と交差するピクセルの位置の整数表現である、当該追加すること、を含む、

請求項2に記載のビデオ符号化方法。

【請求項7】

ビデオデコーダのプロセッサによって実行されるビデオ復号方法であって、

対象ブロックの直上に配置された水平境界ピクセルの配列と対象ブロックの直左に配置された垂直境界ピクセルの配列の一方から少なくともいくつかのピクセルを取り出すステップと、

取り出されたピクセルを他方の境界ピクセルに当該他方の境界ピクセルの配列方向に沿って追加して当該他方の境界ピクセルの配列を当該他方の境界ピクセルの配列方向に沿って拡張するステップと、

前記他方の境界ピクセルの配列方向に沿って拡張された他方の境界ピクセルの配列だけを参照して、イントラ予測を実行するステップと、

を、前記プロセッサによって実行する、ビデオ復号方法。

【請求項8】

前記取り出すステップが、垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理、を含む、

前記拡張するステップが、前記取り出されたピクセルを前記水平境界ピクセルに追加して前記水平境界ピクセルの前記配列を拡張する処理、を含む、

請求項7に記載のビデオ復号方法。

【請求項9】

前記垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理が、

【数3】

$$\left[\frac{size \times col}{angle} \right]$$

によって表される垂直ピクセル識別子を使用して前記垂直境界ピクセルの中から前記少なくともいくつかのピクセルを識別することであって、ここで、*size*が、予測される対象ブロックのサイズを表し、*angle*が予測方向を表し、*col*が、-1から*angle*まで1ずつデクリメントされるカウンタである、当該識別すること、を含み、

前記取り出されたピクセルを前記水平境界ピクセルに追加する処理が、

前記垂直ピクセル識別子によって識別されたピクセルを、水平ピクセル識別子 $[col]$ によって識別される位置で前記水平境界ピクセルに追加すること、を含む、

請求項8に記載のビデオ復号方法。

【請求項10】

前記垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理が、

*InvAngle*を

【数4】

$$\frac{N \times size}{angle}$$

10

20

30

40

50

から計算することであって、ここで、 $size$ が、予測される対象ブロックのサイズを表し、 $angle$ が予測方向を表し、 N が2の整数乗である、当該計算することと、

$[col \times InvAngle > \log_2 N]$ によって表される垂直ピクセル識別子を使用して、前記垂直境界ピクセルの中から前記少なくともいくつかのピクセルを識別することであって、ここで、 col が、 -1 から $angle$ まで1ずつデクリメントされるカウンタである、当該識別することと、

を含み、

前記取り出されたピクセルを前記水平境界ピクセルに追加する処理が、

前記垂直ピクセル識別子によって識別されたピクセルを、水平ピクセル識別子 $[col]$ によって識別される位置で前記水平境界ピクセルに追加すること、を含む、

10

請求項8に記載のビデオ復号方法。

【請求項11】

前記垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理が、

$angle$ の値に関連付けられた $InvAngle$ の値を列挙するルックアップテーブルから $InvAngle$ を獲得することであって、ここで、 $angle$ が予測方向を表す、当該獲得することと、

$[col \times InvAngle]$ を用いた関数によって表される垂直ピクセル識別子を使用して、前記垂直境界ピクセルの中から前記少なくともいくつかのピクセルを識別することであって、ここで、 col が、 -1 から $angle$ まで1ずつデクリメントされるカウンタである、当該識別することと、

20

$[col \times InvAngle]$ によって表される垂直ピクセル識別子を使用して、前記垂直境界ピクセルの中から前記少なくともいくつかのピクセルを識別することであって、ここで、 col が、 -1 から $angle$ まで1ずつデクリメントされるカウンタである、当該識別することと、

を含み、

前記取り出されたピクセルを前記水平境界ピクセルに追加する処理が、

前記垂直ピクセル識別子によって識別されたピクセルを、水平ピクセル識別子 $[col]$ によって識別される位置で前記水平境界ピクセルに追加すること、を含む、

請求項8に記載のビデオ復号方法。

【請求項12】

30

前記垂直境界ピクセルから少なくともいくつかのピクセルを取り出す処理が、

垂直ピクセル識別子 $[row]$ を使用して、前記垂直境界ピクセルの中からピクセルを識別することであって、ここで、 row が、 0 から $size$ まで1ずつインクリメントされるカウンタであり、 $size$ が、予測される対象ブロックのサイズを表す、当該識別すること、を含む、

前記取り出されたピクセルを前記水平境界ピクセルに追加する処理が、

前記垂直ピクセル識別子によって識別されたピクセルを、 $[int + 1]$ によって表される水平ピクセル識別子によって識別される位置で前記水平境界ピクセルに追加することであって、ここで、 int が、予測方向と交差するピクセルの位置の整数表現である、当該追加すること、を含む、

40

請求項8に記載のビデオ復号方法。

【請求項13】

コンピュータシステムのプロセッサとメモリとを備えるビデオエンコーダであって、

前記メモリは、

前記プロセッサによって実行可能なプログラムであって、

対象ブロックの直上に配置された水平境界ピクセルの配列と対象ブロックの直左に配置された垂直境界ピクセルの配列の一方から少なくともいくつかのピクセルを取り出すステップと、

取り出されたピクセルを他方の境界ピクセルに当該他方の境界ピクセルの配列方向に沿って追加して当該他方の境界ピクセルの配列を当該他方の境界ピクセルの配列方向に沿っ

50

て拡張するステップと、

前記他方の境界ピクセルの配列方向に沿って拡張された他方の境界ピクセルの配列だけを参照して、イントラ予測を実行するステップと、

を実行させるための当該プログラム、

を記憶し、

前記プロセッサは、

前記メモリから前記プログラムを読み出し、

読み出された前記プログラムを実行する、ビデオエンコーダ。

【請求項 14】

コンピュータシステムのプロセッサとメモリとを備えるビデオデコーダであって、

前記メモリは、

前記プロセッサによって実行可能なプログラムであって、

対象ブロックの直上に配置された水平境界ピクセルの配列と対象ブロックの直左に配置された垂直境界ピクセルの配列の一方から少なくともいくつかのピクセルを取り出すステップと、

取り出されたピクセルを他方の境界ピクセルに当該他方の境界ピクセルの配列方向に沿って追加して当該他方の境界ピクセルの配列を当該他方の境界ピクセルの配列方向に沿って拡張するステップと、

前記他方の境界ピクセルの配列方向に沿って拡張された他方の境界ピクセルの配列だけを参照して、イントラ予測を実行するステップと、

を実行させるための当該プログラム、

を記憶し、

前記プロセッサは、

前記メモリから前記プログラムを読み出し、

読み出された前記プログラムを実行する、ビデオデコーダ。

【発明の詳細な説明】

【技術分野】

【0001】

[0001]本発明は、ビデオ符号化に関し、特に、同じビデオフレームに属する先に符号化され復号されたピクセルを使用してサンプルブロックを予測する、フレーム内予測に関する。

【背景技術】

【0002】

[0002]圧縮されていない状態では、デジタルビデオシーケンス（例えば一連のフレーム）内のフレームの各々のデジタルビデオデータは膨大な量になる。コンピュータネットワークを介して非圧縮デジタルビデオを送信することは、帯域幅制限のため、ほとんどのアプリケーションで実現可能ではない。加えて、非圧縮デジタルビデオは、大量の記憶空間を必要とする。デジタルビデオデータは、記憶要件を引き下げるために、また帯域幅要件を引き下げるために、通常は何らかの方式で符号化される。

【0003】

[0003]デジタルビデオを符号化するための1つの技法は、フレーム間予測、即ちインター予測である。インター予測は、異なるフレーム間で時間をおいて出現する類似のパターンを利用してビデオデータを圧縮する。一般に、時間的に隣接するビデオフレームは、実質的に類似するピクセルブロックを含む。インター予測を利用した符号化プロセスにおいては、1つのフレーム内のピクセルのブロックの動きを、別のフレーム内の類似ピクセルのブロックに関連付ける動きベクトルを定義する。したがって、システムは、ピクセルのブロックを2回符号化する必要はなく、代わりに、ピクセルブロックを1回符号化し、定義された動きベクトルを利用して、他のピクセルブロックを予測することができる。

【0004】

[0004]デジタルビデオを符号化するための別の技法は、フレーム内予測、即ちイントラ予測と呼ばれる。イントラ予測は、他のフレーム内のピクセルを参照することなく、フレーム又はフレームの一部を符号化する。イントラ予測は、フレーム内において距離を置いて存在する類似のピクセルブロックを利用する。一般に、空間的に隣接するピクセルブロックは類似の属性を有するので、隣接するブロック間の空間的相関を参照することによって、符号化プロセスの効率が改善される。隣接ブロックで使用される予測モードに基づいて対象ブロックを予測するのに、この相関を利用することができる。

【発明の概要】

【0005】

[0005]本発明は、ビデオ符号化の効率を改善する独特なイントラ予測プロセスを提供する。H.264/AVCは、予測される対象ブロックの直上に配置された水平境界内の参照ピクセルと、対象ブロックの直左に配置された垂直境界内の参照ピクセルとを使用する。本発明では、水平境界ピクセルの配列又は垂直境界ピクセルの配列から少なくともいくつかのピクセルが取り出される。その後、取り出されたピクセルを他方の境界ピクセルに追加して、他方の境界ピクセルの配列を拡張する。イントラ予測は、境界ピクセルの拡張された配列だけに基づいて実行される。本発明の一実施形態では、垂直境界ピクセルから少なくともいくつかのピクセルが取り出され、水平境界ピクセルに追加されて、水平境界ピクセルの配列を拡張する。

【0006】

[0006]本発明は、参照ピクセルの取り出しを水平境界から行うのか、それとも垂直境界から行うのか選択する判定プロセスを不要にすることができる。本発明は、予測方向と交差する垂直境界の位置を計算する反復プロセス（除法演算を含んでいる）も不要にすることができる。これらのプロセスを不要にすることで、単一命令複数データ（SIMD: Single-Instruction Multiple Data）アーキテクチャ上でイントラ予測プロセスを実施することが可能になり、それによって、ビデオ符号化の計算効率が改善する。

【0007】

[0007]本発明による一実施形態では、

【数1】

$$\left[\frac{size \times col}{angle} \right]$$

によって垂直ピクセルが指定される（垂直ピクセル識別子）。この識別子を利用して垂直境界ピクセルの中から少なくともいくつかのピクセルが取り出される。ここで、sizeは、予測される対象ブロックのサイズを表し、angleは、予測方向を表し、colは、-1からangleまで1ずつデクリメントされるカウンタである。取り出されたピクセルは、水平ピクセル識別子[col]によって識別される位置で水平ピクセルに追加される。

【0008】

[0008]別の実施形態では、垂直境界ピクセルから少なくともいくつかのピクセルを取り出すとき、InvAngleを

【数2】

$$\frac{N \times size}{angle}$$

から計算する。ここで、Nは、2の整数乗である。その後、 $[col \times InvAngle > \log_2 N]$ によって計算される垂直ピクセル識別子を使用して、垂直境界ピクセル

の中から少なくともいくつかのピクセルが取り出される。取り出されたピクセルは、水平ピクセル識別子 [c o l] によって識別される位置で水平ピクセルに追加される。

【 0 0 0 9 】

[0009]別の実施形態では、 a n g l e の値に関連付けられた I n v A n g l e の値を列挙するルックアップテーブルから、 I n v A n g l e が獲得される。

【 0 0 1 0 】

[0010]別の実施形態では、垂直ピクセル識別子 [r o w] を使用して、垂直境界ピクセルの中からピクセルが識別される。ここで、 r o w は、 0 から s i z e まで 1 ずつインクリメントされるカウンタである。取り出されたピクセルは、水平ピクセル識別子 [i n t + 1] によって識別される位置で水平ピクセルに追加される。ここで、 i n t は、予測方向と交差するピクセルの位置を整数値で表現する。

10

【 0 0 1 1 】

[0011]本発明はまた、水平境界ピクセルの配列又は垂直境界ピクセルの配列から少なくともいくつかのピクセルが取り出されるイントラ予測操作を実行する、エンコーダ及びデコーダを提供する。その後、取り出されたピクセルが、他方の境界ピクセルに追加されて、他方の境界ピクセルの配列を拡張する。イントラ予測は、拡張された境界ピクセルの配列だけに基づいて実行される。

【図面の簡単な説明】

【 0 0 1 2 】

【図 1】本発明を実施できる例示的なハードウェアアーキテクチャを示すブロック図である。

20

【図 2】本発明を適用できるビデオエンコーダの概観を示すブロック図である。

【図 3】本発明を適用できるビデオデコーダの概観を示すブロック図である。

【図 4】本発明の一実施形態によるエンコーダの機能モジュールを示すブロック図である。

【図 5】本発明の実施形態のイントラ予測モジュールによって実行されるイントラ予測プロセスを示すフローチャートである。

【図 6】本発明の一実施形態によるデコーダの機能モジュールを示すブロック図である。

【図 7】 H . 2 6 4 / A V C でサポートされる I n t r a _ 4 x 4 予測モードを説明する予測方向を示す図である。

30

【図 8】文書番号 J C T - V C A 1 1 9 において提案される予測方向を示す図である。

【図 9】図 7 に示される予測方向の 1 つに沿って予測ブロックを生成する、 J C T - V C A 1 1 9 において提案される、プロセスを示すフローチャートである。

【図 10】本発明の一実施形態に従って実行される簡素化されたイントラ予測のプロセスを示すフローチャートである。

【図 11A】予測ブロックと水平及び垂直境界ピクセルの配列とを示す概略図である。

【図 11B】垂直境界ピクセルを用いて拡張された水平境界ピクセルの配列を示す概略図である。

【図 12】本発明の一実施形態に従って実行される水平境界ピクセルの配列を拡張するプロセスを示すフローチャートである。

40

【図 13】水平境界ピクセルの配列を拡張する別の実施形態を示すフローチャートである。

【図 14】本発明の別の実施形態に従って実行される簡素化されたイントラ予測のプロセスを示すフローチャートである。

【発明を実施するための形態】

【 0 0 1 3 】

[0027]図 1 は、本発明を実施できるコンピュータ 1 0 0 の例示的なハードウェアアーキテクチャを示している。図 1 に示されるハードウェアアーキテクチャは、本発明の実施形態を実施するビデオエンコーダとビデオデコーダの両方に共通して使用することができる。コンピュータ 1 0 0 は、ローカルインタフェース 1 0 7 を介して通信可能に結合される

50

、プロセッサ１０１、メモリ１０２、記憶デバイス１０５、及び、１つ又は複数の入出力（Ｉ／Ｏ）デバイス１０６（又は周辺装置）を含む。ローカルインタフェース１０７は、例えば、当該技術分野で知られているような、１つ若しくは複数のバス接続、又は、他の有線若しくは無線の接続とすることができるが、これらに限定されない。

【００１４】

[0028]プロセッサ１０１は、ソフトウェア、特にメモリ１０２に記憶されたソフトウェアを実行するためのハードウェアデバイスである。プロセッサ１０１は、任意のカスタムメイド若しくは市販のプロセッサ、中央処理装置（ＣＰＵ）、コンピュータ１００に関連するいくつかのプロセッサ間の補助プロセッサ、（マイクロチップ若しくはチップセットの形態を取る）半導体ベースのマイクロプロセッサ、又は一般にソフトウェア命令を実行するための任意のデバイスとすることができる。

10

【００１５】

[0029]メモリ１０２は、揮発性メモリ要素（例えば、ランダムアクセスメモリ（ＤＲＡＭ、ＳＲＡＭ、ＳＤＲＡＭなどのＲＡＭ））と不揮発性メモリ要素（例えば、ＲＯＭ、ハードドライブ、テープ、ＣＤＲＯＭなど）のいずれか一方又は両方の組み合わせを含むことができる、コンピュータ可読媒体を含む。さらに、メモリ１０２は、電子、磁気、光、及び／又は他のタイプの記憶媒体を含むことができる。コンピュータ可読媒体は、命令実行システム、装置、又はデバイスが使用するプログラム、又はそれらに関連したプログラムを記憶、伝達、伝搬、又は輸送できる任意の手段とすることができる。メモリ１０２は分散アーキテクチャを有することができ、その場合、様々なコンポーネントは互いに遠く離れて配置されるが、プロセッサ１０１によってアクセスできる。

20

【００１６】

[0030]メモリ１０２内のソフトウェア１０３は、１つ又は複数の別個のプログラムを含むことができ、その各々は、以下で説明されるような、コンピュータ１００の論理機能を実施するための実行可能命令の順序付けられたリストを含む。図１の例では、メモリ１０２内のソフトウェア１０３は、本発明によるコンピュータ１００のビデオ符号化又はビデオ復号機能を具現化する。加えて、必ずしも必要ではないが、メモリ１０２は、オペレーティングシステム（Ｏ／Ｓ）１０４を含むことが可能である。基本的に、オペレーティングシステム１０４は、コンピュータプログラムの実行を制御し、スケジューリング、入出力制御、ファイル及びデータ管理、メモリ管理、並びに、通信制御及び関連サービスを提供する。

30

【００１７】

[0031]コンピュータ１００の記憶デバイス１０５は、１つのタイプ又は多くの異なるタイプの記憶デバイスとすることができるが、固定型記憶デバイス又はポータブル記憶デバイスを含む。一例として、記憶デバイス１０５は、磁気テープ、ディスク、フラッシュメモリ、揮発性メモリ、又は異なる記憶デバイスとすることができる。加えて、記憶デバイス１０５は、セキュアデジタルメモリカード又は他の任意の着脱可能記憶デバイス１０５とすることができる。

【００１８】

[0032]Ｉ／Ｏデバイス１０６は、例えば、タッチスクリーン、キーボード、マウス、スキャナ、マイクロフォン、又は他の入力デバイスなどの、しかしそれらに限定されない、入力デバイスを含むことができる。さらに、Ｉ／Ｏデバイス１０６は、例えば、ディスプレイ、又は他の出力デバイスなどの、しかしそれらに限定されない、出力デバイスを含むこともできる。さらに、Ｉ／Ｏデバイス１０６は、例えば、（別のデバイス、システム、若しくはネットワークにアクセスするための）変復調器（モデム）、無線周波数（ＲＦ）、ワイヤレス、若しくは他のトランシーバ、電話インタフェース、ブリッジ、ルータ、又は入力と出力の両方として機能する他のデバイスなどの、しかしそれらに限定されない、入力と出力の両方を介して通信するデバイスを含むことができる。

40

【００１９】

[0033]当業者によってよく知られているように、ビデオ圧縮は、ビデオシーケンス内の

50

冗長情報を除去することによって達成される。多くの異なるビデオ符号化規格が存在し、それらの例として、MPEG-1、MPEG-2、MPEG-4、H.261、H.263、及びH.264/AVCがある。本発明は特定のビデオ符号化規格いすれにも限定的に適用されるものではないが、本発明は、参照により本明細書に組み込まれるH.264/AVC規格の例を使用し、以下に説明する。H.264/AVCは、最新のビデオ符号化規格であり、MPEG-1、MPEG-2、H.261、及びH.263などのこれまでの符号化規格と比べて、著しい性能改善を達成する。

【0020】

[0034] H.264/AVCでは、ビデオの各フレーム又はピクチャは、いくつかのスライスに分解することができる。次に、スライスは、マクロブロックと呼ばれる 16×16 ピクセルのブロックに分割され、次に、マクロブロックは、 8×16 、 16×8 、 8×8 、 4×8 、 8×4 、最小では 4×4 ピクセルのブロックにさらに分割することができる。H.264/AVCによってサポートされるスライスには5つのタイプがある。Iスライスでは、すべてのマクロブロックが、イントラ予測を使用して符号化される。Pスライスでは、マクロブロックは、イントラ予測又はインター予測を使用して符号化することができる。Pスライスは、マクロブロック当たりただ1つの動き補償予測(MCP: motion compensated prediction)信号を使用することを可能にする。Bスライスでは、マクロブロックは、イントラ予測又はインター予測を使用して符号化することができる。予測当たり2つのMCP信号を使用することができる。SPスライスは、Pスライスを異なるビデオストリーム間で効率的に切り換えることを可能にする。SIスライスは、ランダムアクセス又はエラー回復を行うためのもので、SPスライスに完全に対応しているが、イントラ予測だけを使用する。

【0021】

[0035] 図2は、本発明を適用できるビデオエンコーダの概観を示している。図示されるブロックは、メモリ102内のソフトウェア103を実行するプロセッサ101によって実現される機能モジュールを表す。ビデオフレームのピクチャ200が、ビデオエンコーダ201に供給される。ビデオエンコーダは、マクロブロック200Aを単位として、ピクチャ200を処理する。各マクロブロックは、ピクチャ200のいくつかのピクセルを含む。各マクロブロック上で、変換係数への変換が実行され、それに続いて、変換係数レベルへの量子化が実行される。さらに、イントラ予測又はインター予測を使用して、符号化ステップをピクセルデータに対して直接的に実行せず、ピクセルデータの差分に対して実行して、予測ピクセル値を得る。それによって、小さな値がより容易に圧縮される。

【0022】

[0036] エンコーダ201は、スライス毎に多くのシンタックス要素を生成し、シンタックス要素は、それぞれのスライスのマクロブロックの符号化バージョンを生成する。変換係数レベル、又はスキップされる変換係数レベルを示す有意性マップ(significance map)など、変換係数の符号化に関連する、シンタックス要素内のすべての残差データ要素は、残差データシンタックス要素と呼ばれる。これらの残差データシンタックス要素に加えて、エンコーダ201によって生成されるシンタックス要素は、各マクロブロックをどのように符号化するかに関する情報と、どのように復号しなければならないかに関する情報をそれぞれ含んだ制御情報シンタックス要素を含む。言い換えると、シンタックス要素は、2つのカテゴリに分けることができる。第1のカテゴリである制御情報シンタックス要素は、例えば、マクロブロックタイプ、サブマクロブロックタイプ、及び空間タイプと時間タイプ両方の予測モードについての情報、並びにスライスベース及びマクロブロックベースの制御情報に関する要素を含む。第2のカテゴリでは、量子化変換係数のブロック内のすべての有意係数の位置と、量子化ステップに対応するレベルを単位として表される有意係数の値とを示す有意性マップなど、すべての残差データ要素が、組み合わせられ、残差データシンタックス要素になる。

【0023】

[0037] エンコーダ201は、スライス毎にシンタックス要素を符号化し、算術符号を生

10

20

30

40

50

成する、エントロピコードを備える。スライスについての算術符号を生成する場合、エントロピコードは、ビデオ信号ビットストリーム内のシンタックス要素のデータ値の間の統計的依存性を利用する。エンコーダ 201 は、ピクチャ 200 のスライスについての符号化ビデオ信号を、図 3 に示されるビデオデコーダ 301 に出力する。

【0024】

[0038] 図 3 は、本発明を適用できるビデオデコーダの概観を示している。同様に、図示されるブロックは、メモリ 102 内のソフトウェア 103 を実行するプロセッサ 101 によって実現される機能モジュールを表す。ビデオデコーダ 301 は、符号化ビデオ信号を受信し、最初に信号をエントロピ復号して、シンタックス要素に戻す。デコーダ 301 は、マクロブロックを順々に、その後、スライスを順々に処理して、ピクチャ 300 内のピクセルからなるピクチャサンプル 300A を再構成するために、シンタックス要素を使用する。

10

【0025】

[0039] 図 4 は、ビデオエンコーダ 201 の機能モジュールを示している。これらの機能モジュールは、メモリ 102 内のソフトウェア 103 を実行するプロセッサ 101 によって実現される。入力ビデオピクチャは、クロミナンス（「クローマ」）及びルミナンス（「ルーマ」）など（例えば、色相、彩度、及び色値など、他の成分も可能である）、元の色の成分を表すサンプル点によって定義される自然な（非圧縮の）ビデオイメージのフレーム又はフィールドである。入力ビデオピクチャは、ピクチャ色のルーマ成分の 16×16 ピクセルから成る正方形のピクチャ領域を各々が表す、マクロブロック 400 に分割される。入力ビデオピクチャは、ピクチャ色の 2 つのクローマ成分の各々についての 8×8 ピクセルを各々が表す、マクロブロックにも区分される。一般的なエンコーダ操作では、入力されるマクロブロックは、インター予測又はイントラ予測を使用して、時間的又は空間的に予測することができる。しかし、本願発明の説明の目的のため、マクロブロック 400 は、すべて I スライスタイプのマクロブロックであり、イントラ予測だけが行われると仮定する。

20

【0026】

[0040] イントラ予測は、イントラ予測モジュール 401 において達成され、その動作が、以下で詳細に説明される。イントラ予測モジュール 401 は、先に符号化及び再構成が行われ、フレームメモリ 403 に記憶された、近隣ブロックの水平及び垂直境界ピクセルから、予測ブロック 402 を生成する。対象ブロック 400 と予測ブロック 402 の間の差である予測ブロック 402 の残差 404 は、ビデオ符号化分野の当業者に知られた方法及び技法を使用して、変換 / 量子化モジュール 405 において変換され、スケーリングされ、量子化される。その後、量子化変換係数 406 は、エントロピ符号化モジュール 407 においてエントロピ符号化され、符号化ビデオ信号 408 として（イントラ予測に関する他の情報と一緒に）送信される。

30

【0027】

[0041] ビデオエンコーダ 201 は、対象ブロックにおいてイントラ予測を実行するために、復号の機能性を有する。復号の機能性として、逆量子化 / 逆変換モジュール 409 を備え、逆量子化 / 逆変換モジュール 409 は、量子化変換係数 406 に対して逆量子化及び逆変換を実行して、復号予測残差 410 を生成し、復号予測残差 410 は、予測ブロック 402 に加算される。復号予測残差 410 と予測ブロック 402 の合算が、再構成ブロック 411 であり、再構成ブロック 411 は、フレームメモリ 403 に記憶され、次の対象ブロック 400 を符号化するための予測ブロック 402 を生成するために、イントラ予測モジュール 401 によってフレームメモリ 403 から読み出され、使用される。

40

【0028】

[0042] 図 5 は、イントラ予測モジュール 401 によって実行されるプロセスを示すフローチャートである。H.264 / AVC 規格によれば、イントラ予測は、先に符号化され及びその後復号された近隣ブロックの境界ピクセル（「参照ピクセル」）を補間して、複数の予測モードの下で対象ブロック 400 の各ピクセルを予測することを含む。予測モー

50

ドは、正の整数番号 0、1、2、...によって識別され、各々は、対象ブロック 400 内の特定のピクセルを予測するための異なる命令又はアルゴリズムに関連付けられる。イントラ予測モジュール 401 は、それぞれの予測モードの下でイントラ予測を実行し、異なる予測ブロックを生成する。フルサーチ（「FS」）アルゴリズムの下では、生成された予測ブロックの各々が、対象ブロック 400 と比較されて、予測残差 404 を最小化する、又はより小さな予測残差 404 を生成する最適予測モードを予測モードの中から見つける。最適予測モードを識別する情報は、圧縮され、他の制御情報シンタックス要素とともにデコーダ 301 に送信される。

【0029】

[0043]各予測モードは、予測のおおよその方向、すなわち、水平上方、垂直、及び対角線左下方の語によって呼ばれる。予測方向は、図 7 に示されるような矢印を備えた図を用いて表現される角度方向によって、図式的に記述することができる。このタイプの図では、各矢印は、予測方向又は予測モードを表していると見なすことができる。予測モードに対応する角度は、対象ピクセルを予測するために使用される参照ピクセルの加重平均位置から対象ピクセル位置に向かう方向とおおよその関係がある。予測モードは、どの予測方向とも関連せず、したがって、図に他の予測モードのように図式的に記述できない、DC 予測モードを含む。DC 予測モードでは、予測ブロック 402 の各ピクセルが参照ピクセルの平均値に一律に設定されるように、予測ブロック 402 が生成される。

【0030】

[0044]図 5 に戻ると、ステップ 501 において、予測モードが初期化される。その後、ステップ 502 において、予測モードが DC 予測を示しているかどうか判定される。DC 予測を示している場合、フローはステップ 503 に進み、ステップ 503 において、DC 予測ブロック 402 が、参照ピクセルの平均値を用いて生成される。予測モードが DC 予測を示していない場合、ステップ 504 において、予測モードに関連付けられた命令又はアルゴリズムに従って、予測ブロック 402 が生成されるが、そのプロセスは、以下で詳細に説明される。ステップ 503 又はステップ 504 の後、フローはステップ 505 に進み、予測モードのすべてについて予測ブロックが生成されたかどうか判定される。予測モードのすべての下でイントラ予測が実行された場合、フローはステップ 506 に進む。すべてにおいて実行されていない場合、ステップ 507 において、予測モードがインクリメントされ、フローはステップ 502 に戻る。ステップ 506 において、生成された予測ブロックの各々が対象ブロック 400 と比較されて、予測残差 404 を最小化する最適予測モードが決定される。

【0031】

[0045]図 6 は、ビデオデコーダ 301 の機能モジュールを示している。これらの機能モジュールは、メモリ 102 内のソフトウェア 103 を実行するプロセッサ 101 によって実現される。最初に、エンコーダ 201 からの符号化ビデオ信号は、エントロピデコーダ 600 によって受信され、エントロピ復号が施されて量子化変換係数 601 に戻される。量子化変換係数 601 は、逆量子化 / 逆変換モジュール 602 によって逆量子化及び変換されて、予測残差 603 が生成される。エンコーダ 201 によって選択された予測モードが、イントラ予測モジュール 604 に通知される。選択された予測モードに従って、イントラ予測モジュール 604 は、先に復号され、フレームメモリ 606 に記憶された近隣ブロックの境界ピクセルを使用して、図 5 のステップ 502、503、504 で実行されたプロセスと類似したイントラ予測プロセスを実行して、予測ブロック 605 を生成する。予測ブロック 605 は、予測残差 603 に加算されて、復号ビデオ信号のブロック 607 が復号される。復号されたブロック 607 は、次のブロックの予測で使用するために、フレームメモリ 606 に記憶される。

【0032】

[0046]予測モード（DC 予測モードを除く）の 1 つの下で予測ブロックを生成するために、イントラ予測モジュール 401 及び 604 によって実行されるステップ 504 のプロセスについての以下に詳細に説明する。H.264 / AVC は、Intra_4x4 予測

10

20

30

40

50

、 $\text{Intra_}8 \times 8$ 予測、及び $\text{Intra_}16 \times 16$ 予測をサポートする。 $\text{Intra_}4 \times 4$ 予測は、通常、ピクチャ内に重要な細部が存在する場合に使用される。 $\text{Intra_}4 \times 4$ 予測は、1つのマクロブロック内の16個の 4×4 ルーマブロックを個々に予測する。 $\text{Intra_}4 \times 4$ 予測は、1つのDC予測モードを含む、9つの予測モードの下で実行される。 $\text{Intra_}4 \times 4$ 予測がそれに沿って実行される空間予測方向が、図7に示されている。 $\text{Intra_}8 \times 8$ 予測は、1つのDC予測モードを含む、9つの予測モードの下で実行される。 $\text{Intra_}16 \times 16$ 予測は、1つのDC予測モードを含む、4つの予測モードの下で実行される。

【0033】

[0047]最近の研究により、予測方向の数を増加させた、又は予測モードの数を増加させれば、一般に、ビデオ符号化における圧縮効率が改善することが判っている。例えば、ともに参照により本明細書に組み込まれる、ビデオ符号化共同研究部会(JCT-VC: Joint Collaborative Team on Video Coding)に提出された文書番号JCT-VC A119(「Angular Intra Prediction」)及びJCT-VC A124(「Arbitrary Direction Intra」)を参照されたい。予測方向の数の増加することは、利用可能な予測方向の角度間隔の数が増加することを意味する。したがって、これにより予測ブロック候補の数も増加する。予測ブロック候補の数を増加させることにより、単純に、符号化される対象ブロックとほぼ同じ予測ブロックを得る可能性を高めることができる。図8は、文書番号JCT-VC A119において提案される予測方向を示す図である。図8では、参照ピクセルは、17個の水平ピクセルと17個の垂直ピクセルとから成り、左上隅のピクセルは、水平境界と垂直境界の両方に共通である。したがって、 8×8 ブロックにおいて予測ピクセルを生成するために、33個の異なる予測方向が利用可能である。JCT-VC A124は、予測されるブロックのサイズに応じて予測方向の数が調整される、任意方向イントラ予測を提案している。

【0034】

[0048]図9は、図8に示される予測方向の1つに沿って予測ブロックを生成する、JCT-VC A119において提案される、プロセスを示すフローチャートである。説明を容易にするため、プロセスについての以下の説明では、いくつかのアルゴリズムは簡略化されて説明されている。また、プロセスは、主に垂直の予測方向に沿ったイントラ予測に限定して説明されている。主に水平の予測方向に沿ったイントラ予測は、JCT-VC A119によって提供されるソフトウェアにおいて示されるように、図9に示されるプロセスを用いて対称的に実施することができる。図8は予測される 8×8 ブロックを示しているが、図9に示されるプロセスは、異なる構成に配置された様々な数のピクセルに適用されるように拡張可能である。例えば、予測されるブロックは、 4×4 配列のピクセルを含むことができる。予測ブロックは、 8×8 配列のピクセル、 16×16 配列のピクセル、又はより大きな配列のピクセルを含むこともできる。正方形配列と長方形配列の両方を含む他のピクセル構成も、予測ブロックを構成することができる。

【0035】

[0049]図9のステップ900において、対象ブロックのそれぞれ直上及び直左に存在する、水平境界及び垂直境界における参照ピクセルが、先に符号化及びその後復号され、図4に示されるメモリ403などのフレームメモリに記憶された近隣ブロックから読み取られる。水平境界からのピクセルは、「refH」と呼ばれるメモリ領域に記憶される。垂直境界からのピクセルは、「refV」と呼ばれる別のメモリ領域に記憶される。図8に戻ると、参照ピクセルは、 8×8 ブロックの左上隅のピクセル位置に原点を有する座標系における座標によって識別される。したがって、水平境界ピクセルは、 $p[x, y]$ によって表される座標を有し、 $x = 0, 1, \dots, 16$ 、 $y = 0$ である。垂直境界ピクセルは、 $p[x, y]$ によって表される座標を有し、 $x = 0$ 、 $y = 0, -1, -2, \dots, -16$ である。

【0036】

[0050]メモリ領域 $refH$ に記憶される水平境界ピクセルは、論理アドレス (x)、 $x = 0, 1, \dots, 16$ によって識別され、同様に、メモリ領域 $refV$ に記憶される垂直境界ピクセルは、論理アドレス (y)、 $y = 0, -1, -2, \dots, -16$ によって識別され、各ピクセルは、それが読み取られた座標の数値を有するアドレスに記憶されると仮定する。したがって、図8に水平ピクセル及び垂直ピクセルが図式的に表されているように、メモリ領域 $refH$ 及び $refV$ は、直線的に互いに直交して延びていると考えることができ、各々は、 $2 \times size + 1$ の長さを有する。ここで、「 $size$ 」は、対象ブロックのサイズを表すパラメータである。 $size$ は、4、8、16...など、2の整数乗に等しい値を有すると仮定される。 $H.264/AVC$ のセクション8.3.2.2.1で説明されているようなローパスフィルタを、 $refH$ 及び $refV$ 内のピクセルに任意選択的に適用することができる。

10

【0037】

[0051]ステップ901において、「 row 」と呼ばれるカウンタが、ゼロ(「0」)に設定される。カウンタ row は、0から $size$ までの値を取り、予測ブロック内における予測ピクセルの行位置を表す。ステップ902において、「 pos 」と呼ばれるパラメータが、 $angle \times (row + 1)$ によって計算される。 $angle$ は、固定小数点表現の分数を有するパラメータである。そのため、 $angle$ は、整数部と小数部から形成され、小数部は、固定数の2進数から成る。 $angle$ は、図8に示される予測方向の1つを表す。例えば、「 $angle = -size$ 」は、図8における座標 [$x = 0, y = 0$] を通過する予測方向を識別する。正の値を有する $angle$ は、水平境界だけと交差する予測方向を識別し、一方、負の値を有する $angle$ は、水平境界と垂直境界の両方と交差する予測方向を識別する。 $angle$ は、使用するのが望ましい予測方向の数によって決定される範囲内で変化する。 $JCT-VC A124$ で提案されたように、使用される予測方向の数は、予測されるブロックのサイズに従って決定することができる。以下の説明では、 $angle$ は、「 $-size$ 」から「 $size$ 」までの範囲内で変化する分数値を取ると仮定される。 $angle$ の範囲限界は他の値を用いても定義できることに留意されたい。

20

【0038】

[0052] $angle$ 同様、パラメータ pos も、整数部と小数部から成り、 pos の小数部は、固定された数の2進数から成る。ここで上述したように $angle$ の範囲限界は $size$ の値に設定されている。これによれば、固定された数は $angle$ の範囲限界の底が2の対数に等しい。即ち、 $\log_2 size$ によって表すことができる。 pos は、水平境界と $angle$ によって表される予測方向とが交差する位置を識別する。ステップ902に戻ると、演算「 $pos > \log_2 size$ 」は、 pos の整数部を識別し、 pos の整数部は、パラメータ「 int 」に記憶され、演算「 $pos \& (size - 1)$ 」は、 pos の小数部を識別し、 pos の小数部は、パラメータ「 $frac$ 」に記憶される。演算子「 $>$ 」は、2進数の算術右シフトを意味する。演算子「 $\&$ 」は、ビット毎の「 and 」演算を求める。

30

【0039】

[0053]ステップ903において、 $angle$ がゼロ(「0」)以上の値を有するかどうか判定される。 $angle$ がゼロ以上の値を有する場合、フローはステップ904に進む。さもなければ、フローはステップ913に進む。ゼロ以上の $angle$ は、水平境界内に配置された参照ピクセル、即ち $refH$ に記憶された参照ピクセルだけを用いて予測ブロック内の予測ピクセルを導出することができることを意味する。他方、ゼロよりも小さい $angle$ は、予測ブロック内の予測ピクセルを導出するために、垂直境界内に配置された参照ピクセル即ち $refV$ に記憶された参照ピクセルが必要とされることを意味する。

40

【0040】

[0054]ステップ904において、 $frac$ がゼロでないかが判定される。 $frac$ がゼロでない場合、フローはステップ905に進む。 $frac$ がゼロである場合、フロ

50

ーはステップ906に進む。ゼロに等しい $frac$ は、予測ブロック内の予測ピクセルを、水平境界内の参照ピクセルから直接的にコピーできることを示唆する。非ゼロ $frac$ は、予測方向が非整数位置で水平境界と交差し、予測ブロック内の予測ピクセルを導出するために、2つ以上の参照ピクセルの補間が必要とされることを示唆する。

【0041】

[0055]ステップ905において、「 col 」と呼ばれるカウンタが、ゼロ(「0」)に設定される。カウンタ col は、 $refH$ 内の参照ピクセルをアドレッシングするために使用される。ステップ907において、「 $int + col + 1$ 」と「 $int + col + 2$ 」によって識別される2つの参照ピクセルが、 $refH$ から取り出される。これら2つの参照ピクセルは、予測ピクセル v を導出するために、加重平均され、即ち $frac$ を用いて補間される。具体的には、「 $int + col + 1$ 」によって識別される $refH$ 内の参照ピクセルには、「 $size - frac$ 」が乗算され、パラメータ a に記憶される。「 $int + col + 2$ 」によって識別される $refH$ 内の参照ピクセルには、「 $frac$ 」が乗算され、パラメータ b に記憶される。その後、パラメータ a 及び b は加算され、 $size$ 、すなわち $(size - frac) + frac$ によって除算される。 $size$ による除算は、 $\log_2 size$ 回の右シフトで置き換えることができる。導出された予測ピクセル v は、特定の予測方向の下での対象ブロックのための予測ブロックを表す、「 $pred$ 」と呼ばれるメモリ領域の配列に記憶される。 $pred$ 内の各メモリ領域は、パラメータ row 及び col によって識別される。その後、 col は、ステップ908において、1だけインクリメントされ、ステップ909において、 $size$ と比較される。 col が $size$ よりも小さい限り、ステップ907及びステップ908が繰り返される。 col が $size$ と等しくなったとき、フローはステップ920に進む。

【0042】

[0056]ステップ904において、 $frac$ がゼロであると判定された場合、ステップ906において、カウンタ col が、ゼロに設定される。ステップ910において、予測ピクセル v が、 $refH(int + col + 1)$ から直接的にコピーされ、その後、 $pred$ 内の対応するメモリ領域に記憶される。その後、 col は、ステップ911において、1だけインクリメントされ、ステップ912において、 $size$ と比較される。 col が $size$ よりも小さい限り、ステップ910及びステップ911が繰り返される。 col が $size$ と等しくなったとき、フローはステップ920に進む。

【0043】

[0057]ステップ903に戻ると、ゼロよりも小さい $angle$ は、予測ブロック内の予測ピクセルを導出するために、 $refV$ からの参照ピクセルを必要とする。ステップ913において、カウンタ col が、ゼロに設定される。その後、ステップ914において、「 $int + col + 1$ 」がゼロよりも低いかが判定され、ゼロ以上の「 $int + col + 1$ 」は、予測ブロック内の予測ピクセルを導出するために、 $refH$ に記憶された参照ピクセルだけにまだ依存できることを意味し、フローはステップ915に進む。ステップ915で実行されるプロセスは、ステップ907のプロセスと同様であり、その説明はここでは繰り返さない。その後、 col は、ステップ916において、1だけインクリメントされ、ステップ917において、 $size$ と比較される。 col が $size$ よりも小さい限り、ステップ914、ステップ915、及びステップ916が繰り返される。 col が $size$ と等しくなったとき、フローはステップ920に進む。

【0044】

[0058]ステップ914において、「 $int + col + 1$ 」がゼロよりも小さいと判定された場合、予測ブロック内の予測ピクセルを導出するために、 $refV$ に記憶された参照ピクセルが必要とされる。ステップ918において、最初に、垂直境界と予測方向が交差する位置が決定される。ステップ918において、位置は $pos2$ によって表される。ステップ902では、 pos 、すなわち、水平境界と予測方向が交差する位置は、「 $angle \times (row + 1)$ 」によって決定されることに留意されたい。 $angle$ が水平差と垂直差の比を表すことを考慮して、垂直境界と予測方向が交差する位置を決定するために

、「 $angle \times (row + 1)$ 」の代わりに、「 $angle^{-1} \times (col + 1)$ 」が計算される。上で仮定したように、 $angle$ は、 $-size$ から $size$ までの範囲内にある（ $-size \leq angle \leq size$ ）。したがって、 $angle$ と $size$ の間の比は、

【数3】

$$\alpha = \frac{angle}{size} \quad (-1 \leq \alpha \leq 1)$$

10

によって定義される。その場合、 $angle^{-1}$ は、

【数4】

$$angle^{-1} = \frac{size}{\alpha} \text{ 又は } \frac{size^2}{angle}$$

によって定義される。そのため、ステップ918において、 $pos2$ は、以下のように、 $size$ の2乗に $col + 1$ を乗算し、その後、 $angle$ の絶対値で除算することで決定される。

20

【数5】

$$pos2 = \frac{size^2 \times (col + 1)}{|angle|}$$

【0045】

[0059] pos 同様、 $pos2$ は、整数部と小数部から形成される、固定小数点表現の分数を有する。小数部は、 $\log_2 size$ によって決定される数の2進数から成る。 $pos2$ の整数部は、パラメータ $int2$ に記憶され、 $pos2$ の小数部は、パラメータ $frac2$ に記憶される。ステップ919において、「 $int2 + row + 1$ 」と「 $int2 + row + 2$ 」によって識別される2つの参照ピクセルが、 $refV$ から取り出される。これら2つの参照ピクセルは、予測ピクセル v を導出するために、加重平均され、即ち $frac2$ を用いて補間される。具体的には、 $refV(int2 + row + 1)$ からの参照ピクセルには、「 $size - frac2$ 」が乗算され、パラメータ a に記憶される。 $refV(int2 + row + 2)$ からの参照ピクセルには、「 $frac2$ 」が乗算され、パラメータ b に記憶される。その後、パラメータ a 及び b は加算され、 $size$ によって除算され、又は $\log_2 size$ だけ右シフトされる。導出された予測ピクセル v は、 $pred$ の対応するメモリ領域に記憶される。ステップ917において、 col が $size$ に等しくなるまで、ステップ914、ステップ918、ステップ919、及びステップ916が繰り返される。

30

40

【0046】

[0060] ステップ920において、 row が1だけインクリメントされる。ステップ921において、 row が $size$ よりも小さいかが判定される。 row が $size$ よりも小さい限り、予測ブロック内の予測ピクセルを導出するために、ステップ902からのステップが繰り返される。ステップ921において、 row が $size$ と等しくなったときに、フローは終了する。

【0047】

[0061] 上述したように、予測ブロック候補の数を増加させると、符号化効率が改善する

50

。一方で、予測ブロック候補の数を増加させると、計算作業負荷も増加する。したがって、予測ブロック候補の数を増加させ、それによって、符号化効率を改善するために、予測ブロック候補を生成するプロセスを検討して、さらにプロセスの効率を改善する必要がある。図9に示されるプロセスを検討すると、2つの計算ボトルネックがあることが判る。第1の計算ボトルネックは、ループ内で繰り返される、ステップ914の比較及び分岐操作である。第2の計算ボトルネックは、やはりループ内で繰り返される、ステップ918の除算操作である。

【0048】

[0062]昨今は、単一命令複数データ(SIMD)を使用して効率的なコンピューティングが可能である。SIMDでは、複数の処理要素を備えるコンピュータが、複数のデータに対して同じ操作を同時に実行することが可能である。しかし、SIMDアーキテクチャは、通常、ループ内における除算及び計算/分岐の実施をサポートしてない。したがって、ステップ907及びステップ910から開始するループは、SIMDを用いて実施するのに十分なロバスト性を有するけれども、ループ内にはステップ914及びステップ918が含まれるので、図9に示されるプロセスを実施するために、SIMDアーキテクチャを使用することができない。したがって、本発明の目的は、図9に示されるプロセスから計算上のボトルネックを取り除き、典型的なSIMDアーキテクチャが、図8に示される予測方向のすべてに沿って、並列処理を実施することを可能にする、簡素化されたイントラ予測を提供することである。

【0049】

[0063]図10は、図5のステップ504におけるプロセスを実施する際に、図9のプロセスを置き換えるように設計された、本発明の一実施形態による簡素化されたイントラ予測のプロセスを示すフローチャートである。図10では、ステップ900、901、902、904、905、906、907、908、909、910、911、912、920、921など、図9で実行されるのと同じプロセスステップは、図9で使用されるのと同じステップ番号によって識別される。これら共通のステップについての説明は、ここでは繰り返さない。ステップ1000及びステップ1001が、図10のプロセスに特有のステップである。図9に示されるプロセスとの比較から明らかなように、図10のプロセスは、ステップ903の比較ステップと、angleがゼロよりも小さいときに実行される、ステップ903から左に分岐したステップのすべてとを除去しており、それによって、ステップ914及びステップ918の計算上のボトルネックを除去している。

【0050】

[0064]追加されたステップ1000及びステップ1001において、angleが-1以上であるかどうか判定される。angleが-1以上である場合、予測ブロック内の予測ピクセルを生成するには、水平境界内に配置された参照ピクセルがあれば十分であり、垂直境界内の参照ピクセルは必要とされない。他方、angleが-1よりも小さい場合、予測ブロック内の予測ピクセルを生成するために、垂直境界内の参照ピクセルが必要とされる。ステップ1001では、refVに記憶されたピクセルの少なくともいくつかを使用して、refHに記憶された参照ピクセルが負の方向に拡張される。図11A及び図11Bは、ステップ1001で実行されるrefHの拡張を示す概略図である。図11Aでは、refHに記憶される参照ピクセル1102は、対象ブロック1101の上に配置された水平境界からのものである。refVに記憶される参照ピクセル1103は、対象ブロック1101の左に配置された垂直境界からのものである。図11Bに示されるように、図10のステップ1001の後、refV内の参照ピクセルのいくつかは、refHにコピーされ、refHは、負の方向に拡張された拡張部1104を有する。

【0051】

[0065]図12は、ステップ1001で実行されるプロセスの詳細を示すフローチャートである。ステップ1201において、カウンタcolが、-1に設定される。colは、refHの拡張部のアドレスを識別するために使用される。ステップ1202において、refHの拡張部にコピーされるrefV内の参照ピクセルは、

【数 6】

$$\frac{size \times col}{angle}$$

によって識別される。上の式における除算は整数除算であり、式の結果は整数である。この式は、図 9 に示されるステップ 918 のプロセスと同様に機能する。ステップ 918 において、*p o s 2* の整数値は、

【数 7】

$$\frac{(size^2 \times (col + 1))}{angle} \gg \log 2_size$$

によって計算される。*l o g 2 _ s i z e* 回の右シフトは *s i z e* による除算と等価である。

【0052】

[0066] ステップ 1203 において、*c o l* が 1 だけデクリメントされる。その後、ステップ 1204 において、*c o l* が *a n g l e* と等しいかどうか判定される。*c o l* が *a n g l e* と等しくない場合、フローはステップ 1202 に戻る。*c o l* が *a n g l e* と等しくなるまで、ステップ 1202 及びステップ 1203 が繰り返される。したがって、参照ピクセルは、昇順、又は垂直境界の最上部から最下部に向かって *r e f V* から読み取られ、やはり降順、又は水平境界の右から左に向かって *r e f H* にコピーされる。また、*r e f V* 内の必ずしもすべてのピクセルが、*r e f H* にコピーされるわけではない。最上部から予測方向の交差部までの範囲内に配置された参照ピクセルだけが、*r e f V* から *r e f H* にコピーされる。

【0053】

[0067] 図 10 に戻ると、ステップ 902 から開始するプロセスは、図 9 からコピーしたものであり、図 9 のステップ 903 の比較ステップから右に分岐する予測ピクセルを生成するためのステップを含む。しかし、予測ピクセルを生成するための図 10 のステップは、拡張された *r e f H* (図 11B の部分 1102 と部分 1104 の総和) を使用し、一方、図 9 の対応するステップは、元の *r e f H* (図 11A の部分 1102) を使用することに留意されたい。*r e f H* は負の方向に拡張されているので、図 9 においてステップ 903 から左に分岐する、*r e f V* に記憶された参照ピクセルを使用するように特に設計された別個のイントラ予測操作は、*a n g l e* の符号に係わらず必要とされない。

【0054】

[0068] 図 13 は、*r e f V* の参照ピクセルを使用して、*r e f H* を拡張するためのプロセスの別の実施形態を示している。図 11A、図 11B 及び図 12 に示されるプロセスは、図 9 に示されるステップ 914 及びステップ 918 のボトルネックステップを除去し、したがって、イントラ予測プロセスの効率を改善することが期待される。図 13 に示されるプロセスは、参照ピクセルを *r e f V* から *r e f H* にコピーするためのループから、図 12 のステップ 1202 で実行される除法演算を除去する。ループから除法演算を除去することによって、図 13 に示されるプロセスは、イントラ予測プロセスの効率をさらに改善することが期待される。

【0055】

[0069] 図 13 に示されるプロセスは、図 12 のステップ 1202 をステップ 1301 及び 1302 で置き換える。ステップ 1302 は、参照ピクセルを *r e f V* から *r e f H* にコピーするためのループの内側にあり、一方、ステップ 1301 は、ループの外側にある

10

20

30

40

50

。ステップ1301は、「InvAngle」と呼ばれる新しいパラメータを導入する。
InvAngleは、
【数8】

$$256 \times \frac{size}{angle}$$

によって定義される。256による乗算は、8回の左シフトと等価であり、「size / angle」の演算から得られるすべてのビットが、refVの参照ピクセルを識別する計算に係わることを保証する。ステップ1302では、refHの拡張部にコピーされるrefVの参照ピクセルのアドレスが、
【数9】

$$col \times InvAngle \gg 8$$

によって識別される。「col × InvAngle」の結果は、ステップ1301で実行された左シフト演算を取り消すために、8だけ右シフトされる。ステップ1302における右シフト演算は、「col × InvAngle」の結果の端数を切り捨てるように機能することに留意されたい。最も近い整数に丸めるために、右シフト演算を実行する前に、丸めオフセットである128を「col × InvAngle」の結果に加算することができる。数「256」は一例にすぎず、ステップ1301は、その数が「size / angle」の演算の結果であるすべてのビットを保存するのに十分な大きさである限り、別のオフセット数、好ましくは2の整数乗を採用できることに留意されたい。例えば、ステップ1301において、数は、256の代わりに、64とすることができ、ステップ1302における右シフトの回数は、8回の代わりに、6回である。64が採用された場合、丸めオフセットは32としなければならない。

【0056】

[0070]ステップ1301で実行される計算は、計算作業負荷をさらに低減させるために、検索操作で置き換えることができる。言い換えると、angleの値に関連付けられたInvAngleの値を記憶するルックアップテーブルが準備される。以下に提供される表1は、ステップ1301における検索のための例示的なテーブルである。

【表1】

表1

angle	1	2	3	4	5	6	7	8
InvAngle	2048	1024	683	512	410	341	293	256

上の表では、sizeは8であり、angleは1から8までの整数値を取ると仮定される。しかし、sizeは8に限定されず、4及び16などの別の値を取ることもし得る。また、angleは、上で定義されたように、固定小数点表現の分数とすることができる。

【0057】

[0071]参照ピクセルが、図12のステップ1202又は図13のステップ1302において、refVからrefHにコピーされる場合、参照ピクセルは、予測ブロック内に存在し得るエイリアシングを低減させるために、ローパスフィルタを通過することができる。ローパスフィルタの強度は、angleの値に応じて様々とする事ができる。例えば、angleが-sizeに等しい場合、弱いローパスフィルタリングを適用することが

10

20

30

40

50

でき、 $angle$ が -2 に等しい場合、強いローパスフィルタリングを適用することができる。

【0058】

[0072]上で説明したように、 $refV$ の参照ピクセルの必ずしもすべてが $refH$ にコピーされるわけではない。 $refV$ の参照ピクセルの必ずしもすべてがコピーされるわけではないので、ピクセルがコピーされるときに、いくつかの情報が失われる。情報の喪失を緩和するため、 $refH$ 及び $refV$ が、先に符号化及び再構成されたブロックからのピクセルばかりでなく、2つの隣接するピクセルを補間することによって生成される、2つの隣接する再構成ピクセルの間の1つのピクセルも含むように、 $refH$ 及び $refV$ の参照ピクセルの解像度を2倍にすることができる。補間ピクセルを生成するために、2つの隣接するピクセルを単純に平均することができる。補間プロセスは、参照ピクセルが図9のステップ900において読み取られたときに実行することができる。 $refH$ 及び $refV$ においてピクセルの解像度が2倍にされた場合、図9のステップ907、910、915、919、及び図10のステップ1001において実行される、 $refH$ 及び $refV$ に記憶される参照ピクセルのアドレスの識別をスケーリングする必要がある。例えば、ステップ907、910、915において実行される「 $int + col + 1$ 」は、「 $int + 2 \times col + 2$ 」に変更する必要がある。ステップ907、910、915において実行される「 $int + col + 2$ 」は、「 $int + 2 \times col + 3$ 」に変更する必要がある。ステップ919において実行される「 $int2 + row + 1$ 」及び「 $int2 + row + 2$ 」は、それぞれ「 $int2 + 2 \times row + 2$ 」及び「 $int2 + 2 \times row + 3$ 」に変更する必要がある。

【0059】

[0073]別の実施形態では、図12のステップ1202のプロセスは、コピープロセスをさらに簡略化するために、「 $refH[col] \quad refV[-col]$ 」に単純に変更することができる。予測の精度を低下させるものの、この実施形態は、イントラ予測操作を最も簡素化することができる。

【0060】

[0074]図11Bは、 $refH$ に追加される拡張部1104を示している。拡張部1104は、 $refV$ からの参照ピクセルを用いて形成する必要はない。拡張部1104は、拡張部1104の位置と空間的に対応する、先に復号されたブロックの領域に属するピクセルを用いて形成することができる。図11Bでは、負の方向に拡張されるので、拡張 $refH$ (部分1102及び部分1104)は、 $(-size + 1)$ から $(2 \times size)$ までの範囲にわたる。拡張 $refH$ の範囲は、拡張 $refH$ 内の参照ピクセルをアドレッシングするときに、適切なオフセットを追加することによって、0から $(3 \times size - 1)$ までの範囲に再スケーリングすることができる。 $refV$ の範囲を再スケーリングする場合も同じことが言える。

【0061】

[0075]別の実施形態では、 $angle$ の範囲限界は、自由に選択することができる。上述の実施形態では、 $angle$ は $-size$ から $size$ までの範囲内($-size \quad angle \quad size$)の値を取ると仮定される。言い換えると、上述の実施形態では、 $angle$ の範囲限界は、対象ブロックの $size$ を用いて定義される。 $\log_2 _rangeLimit$ が正の整数であり、式「 $rangeLimit = 1 < \log_2 _rangeLimit$ 」が成り立つように、範囲限界を2の整数乗を用いて定義することがやはり好ましいが、 $angle$ の範囲限界は、対象ブロックの $size$ とは独立に定義することに留意されたい。 $rangeLimit$ のための適切な大きさの数を選択することによって、多数の予測方向を確立し、十分に広い角度間隔を取る $angle$ の値によって表すことができる。

【0062】

[0076] $angle$ の範囲限界が、対象ブロックの $size$ とは独立に定義される場合、ステップ909、912、917、921を除き、図9及び図10に現れる $size$ は、

`rangelim`で置き換える必要があり、`log2_size`は、`log2_rangelimit`で置き換える必要がある。図10のステップ1000において実行される比較「`angle - 1`」も、「`angle * size / rangelim - 1`」又は「`angle * size - rangelim`」で置き換える必要がある。さらに、図12及び図13のステップ1202及びステップ1301に現れる`size`は、`rangelim`で置き換える必要があり、ステップ1204において実行される比較「`col = angle?`」は、「`col = angle * size / rangelim?`」で置き換える必要がある。

【0063】

[0077] `angle`の範囲限界として`rangelim`が導入される場合、（上で提供された）表1は、以下のように変更することができる。

表2

<code>angle*</code>	2	5	9	13	17	21	26	32
<code>InvAngle</code>	4096	1638	910	630	482	390	315	256

表2では、`rangelim`は、32に設定される。`angle*`は、「`rangelim * tan (x * angle / 8)`」の整数近似に等しく、ここで、`angle` = 1、2、3、4、5、6、7、8である。`InvAngle`は、「`256 * rangelim / angle*`」に等しい。表2の値はすべて、切り上げによって導出される整数である。切り上げを行う代わりに、数値の端数を切り捨てることもできる。以下で提供される表3では、`InvAngle`は、「`32 * rangelim / angle*`」に等しい。「256」の代わりに「32」が使用されるので、予測の精度は、表2の精度よりも必然的に低い。

表3

<code>angle*</code>	2	5	9	13	17	21	26	32
<code>InvAngle</code>	512	204	113	78	60	48	39	32

【0064】

[0078] 図14は、図10に示されるプロセスをさらに簡略化する別の実施形態を示すフローチャートである。参照ピクセルを`refV`から`refH`にコピーする図10に示されるプロセスは、フローがメイン予測ループに入る前に実行されるが、図14に示されるコピープロセスは、メイン予測ループの内側で実行される。また、図14に示されるプロセスでは、変数`InvAngle`が除去されている。図14に示されるステップ900、902、921は、図10の対応するステップからのものである。

【0065】

[0079] ステップ1401において、カウンタ`lastInt`が、-1に初期化される。`lastInt`は、`refH`に追加された最終ピクセルのインデックスを表す。ステップ902において、`pos`が、`angle * (row + 1)`によって計算される。上で説明したように、`pos`は、境界と`angle`によって表される予測方向との交差の位置を識別する。図9の文脈では、ステップ902は、水平境界と`angle`によって表される予測方向との交差の位置を識別する`pos`を生成する。さらに、ステップ902では、`pos`の整数部が`int`に記憶され、`pos`の小数部が「`frac`」に記憶される。ステップ1402において、`int`が`lastInt`よりも小さいかどうか判定される。`int`が`lastInt`よりも小さい場合、`row`によって識別される`refV`の参照ピクセルが、`refH`の「`int + 1`」によって識別されるアドレスにコピーされる。ステップ1404は、図9及び図10に示されるステップ904、905、906、907、908

10

20

30

40

50

、909、910、911、912から成り、それらの説明はここでは繰り返さない。ステップ1405において、`int`が`lastInt`にコピーされる。`int`を`lastInt`にコピーする操作は、ステップ1405の代わりに、ステップ1403において実行することができる。

【0066】

[0080]ステップ1403におけるコピー操作は、ステップ1202及びステップ1302においてコピーされるのと同じピクセルのコピーをもたらし、これらのステップでは、切り捨てが使用される。ステップ1403は、ステップ902において計算された小数部`frac`が、`rangeLimit + (angle >> 1)`によって定義されるオフセットよりも大きいという条件が成り立つ場合に、ステップ1403において「`row`」の代わりに「`row + 1`」を使用することによって、最も近い整数に丸めるように変更することができる。`angle`は`-ve`であり、`frac`は`+ve`であることに留意されたい。「`row + 1`」の使用は、切り上げをもたらす。`row`を条件付きで1だけインクリメントするため、32ビット算術では、「`offset - frac`」の右シフトは、`frac`が`offset`よりも大きい場合は、`-1`をもたらし、それ以外の場合は、`0`をもたらすことを前提して、ステップ1403において実行されるプロセスは、`refH[int + 1]` `refV[row - ((offset - frac) >> 31)]`に変更される。したがって、アドレス識別子「`row - ((offset - frac) >> 31)`」は、`frac`が`offset`よりも大きい場合は、「`row + 1`」になり、それ以外の場合は、「`row`」になる。`offset`が`rangeLimit`になるように設定される場合、`offset - frac`は常に正であり、したがって、丸めは行われない。

【0067】

[0081]図14に示されるプロセスを実施する、C++プログラミング言語で開発したソースコードが、以下に示されている。ソースコードは、<http://hevc.kw.bbc.co.uk/svn/jctvc.al24/tags/0.7>から入手可能なJCT-VCによって開発されたTMuC 0.7ソフトウェアの部分であるTComPrediction.cppファイル内に見出される、TComPrediction::xPredIntraAng関数から変更したものである。

【数10】

```
// Function for deriving the simplified angular intra predictions
```

10

20

30

【数 1 1】

```

Void TComPrediction: :xPredIntraAng( Int* pSrc, Int iSrcStride, Pel*& rpDst,
Int iDstStride, UInt iWidth, UInt iHeight, UInt uiDirMode, Bool bAbove, Bool
bLeft ){

```

```

    Int k,l;

    Int deltaInt, deltaFract, refMainIndex;

    Int intraPredAngle      = 0;

    Int absAng              = 0;

    Int signAng              = 0;
    10

    Int blkSize              = iWidth;

    Bool modeDC              = false;

    Bool modeVer             = false;

    Bool modeHor             = false;

    Pel* pDst                = rpDst;

    // Map the mode index to main prediction direction and angle

    if (uiDirMode == 0)
    20
        modeDC = true;

    else if (uiDirMode < 18)
        modeVer = true;

    else
        modeHor = true;

    intraPredAngle = modeVer ? uiDirMode - 9 : modeHor ? uiDirMode - 25 : 0;

    absAng          = abs(intrapredAngle);
    30

    signAng         = intraPredAngle < 0 ? -1 : 1;

    // Set bitshifts and scale the angle parameter to size2

    Int iAngTable[9] = { 0, 2, 5, 9, 13, 17, 21, 26, 32};

    absAng = iAngTable[absAng];

    intraPredAngle = signAng * absAng;

    // Do the DC prediction
    40

    if (modeDC){

        Pel dcval = predIntraGetPredValDC(pSrc, iSrcStride, iWidth, iHeight,
bAbove, bLeft);

```

【数 1 2】

```

    for (k=0;k<blkSize;k++){
        for (l=0;l<blkSize;l++){
            pDst(k*iDstStride+l) = dcval;
        }
    }
}

// Do angular predictions
else {
    Pel tmp;
    Int *pSrcTL = pSrc - iSrcStride - 1;
    Int iStepMain = (modeVer) ? 1 : iSrcStride;
    if (intraPredAngle == 0){
        for (k=0;k<blkSize;k++){
            for (l=0;l<blkSize;l++){
                pDst (k*iDstStride+l) = pSrcTL[(l+1) * iStepMain];
            }
        }
    }
    else {
        Int iStepSide = (modeVer) ? iSrcStride 1;
        int lastDeltaInt = -1;

        Int iOffset = 32 + (intraPredAngle >> 1); // enables rounding to
nearest side reference
        // Int iOffset = 32; // no rounding.

        Pel ref [2*MAX_CU_SIZE];

        Pel* refMain = ref + ((intraPredAngle < 0) ? blkSize : 0);

        if (intraPredAngle > 0){
            for (k = 0; k < 2*blkSize; k++)
                refMain[k] = pSrcTL[(k+1) * iStepMain];
        }
        else {
            for (k = -1; k < blkSize; k++) // the rest are copied later in step
1403, as and when required

```

10

20

30

40

【数 1 3】

```

        refMain[k] = pSrcTL[(k+1) * iStepMain];
    }

    for (k = 0; k < blkSize; k++){
        Int deltaPos = (k+1) * intraPredAngle;

        deltaInt      = deltaPos >> 5;

        deltaFract = deltaPos & (32 - 1);
    10

        if (deltaInt < lastDeltaInt) { // step 1402

            lastDeltaInt = deltaInt;

            refMain[deltaInt] = pSrcTL[(k-((iOffset-deltaFract)>>31))*iStepSide];
        // step 1403

        }

        // step 1404

        if (deltaFract){
    20

            // Do linear filtering

            for (l=0;l<blkSize;l++){

                refMainIndex = l+deltaInt;

                pDst[k*iDstStride+l] = (Pel) ( ((32-deltaFract) *
refMain[refMainIndex] + deltaFract * refMain[refMainIndex+1] + 16) >> 5 );

            }

        }

        else {
    30

            // Just copy the integer samples

            for (l=0;l<blkSize;l++) {

                pDst[k*iDstStride+l] = refMain[l+deltaInt];

            }

        }

    }

    // Flip the block if this is the horizontal mode
    40

    if (modeHor){

        for (k=0;k<blkSize-1;k++){

            for (l=k+1;l<blkSize;l++){

```


【数 1 4】

```

    tmp = pDst[k*iDstStride+1];

    pDst(k*iDstStride+1) = pDst(1*iDstStride+k);

    pDst(1*iDstStride+k) = tmp;

  }

}

}

}

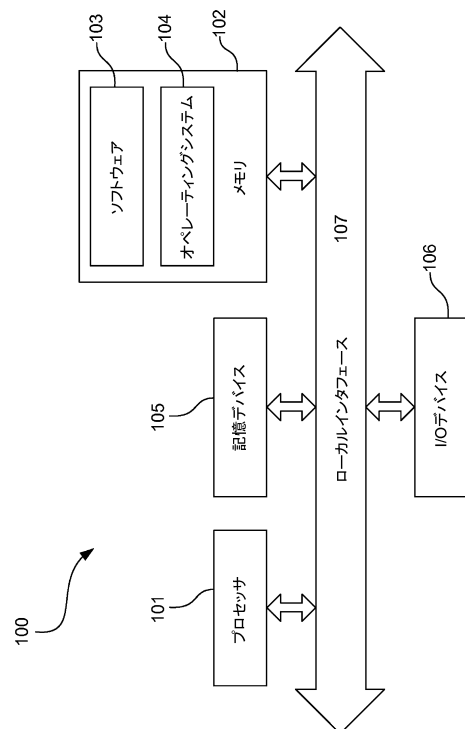
```

10

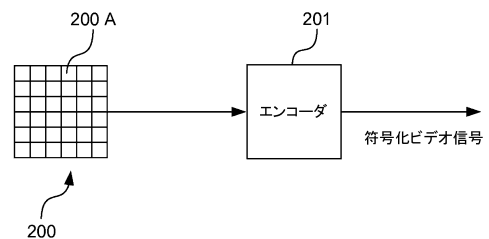
【 0 0 6 8 】

[0082] 上述の説明を読んだ後では、本発明の多くの代替形態及び変更形態が、当業者に明白になることは疑いないが、例によって示され、説明された特定の実施形態はいずれも、限定的なものと思なされることは決して意図していない。したがって、様々な実施形態の細部についての言及は、特許請求の範囲を限定することを意図しておらず、特許請求の範囲それ自体が、本発明に必須と思なされる特徴をもっぱら列挙する。

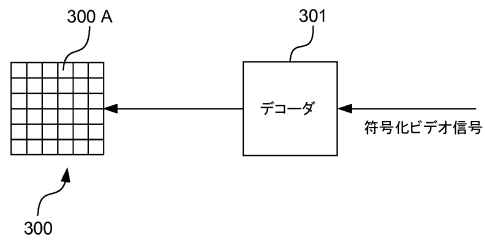
【図 1】



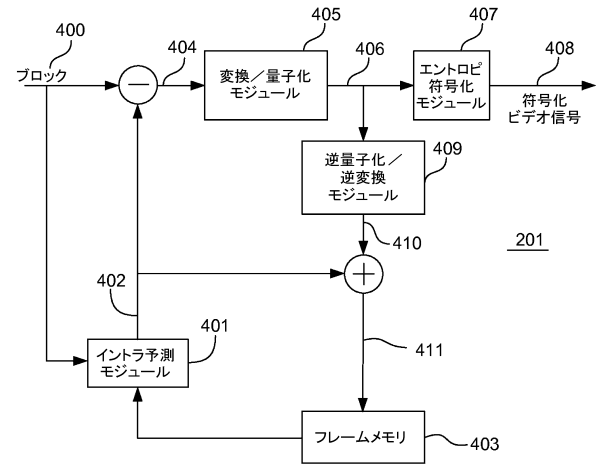
【図 2】



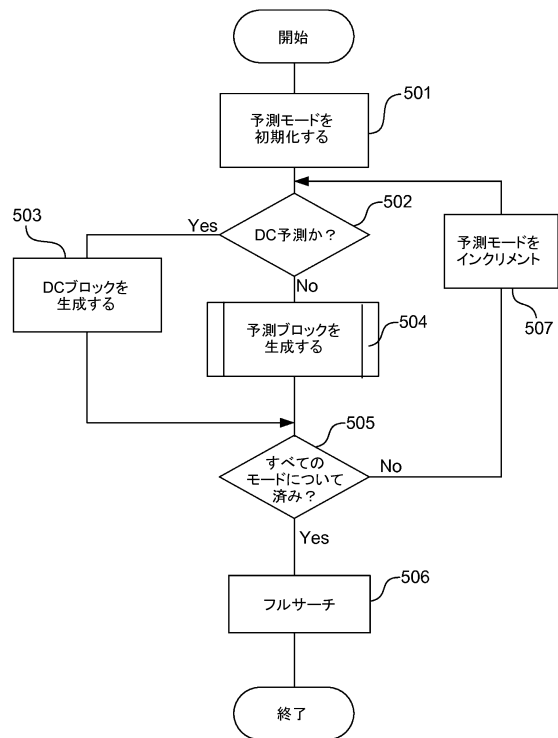
【図 3】



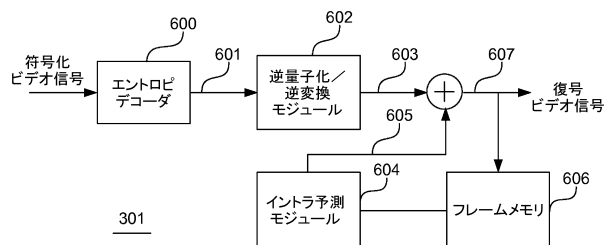
【図 4】



【図 5】



【図 6】



【図 7】

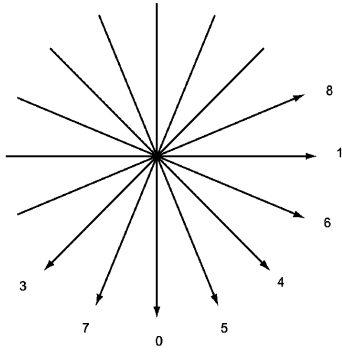


Fig. 7

【図 8】

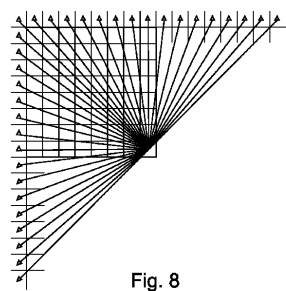
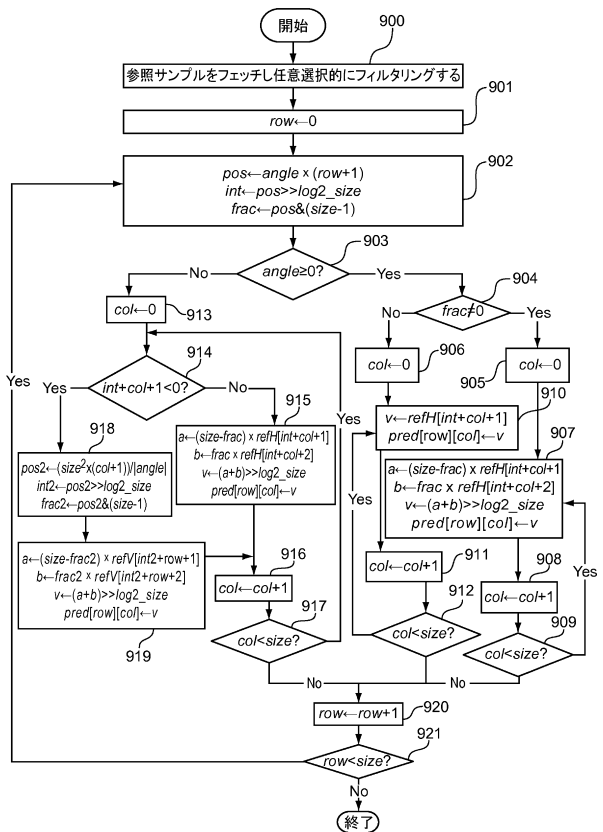
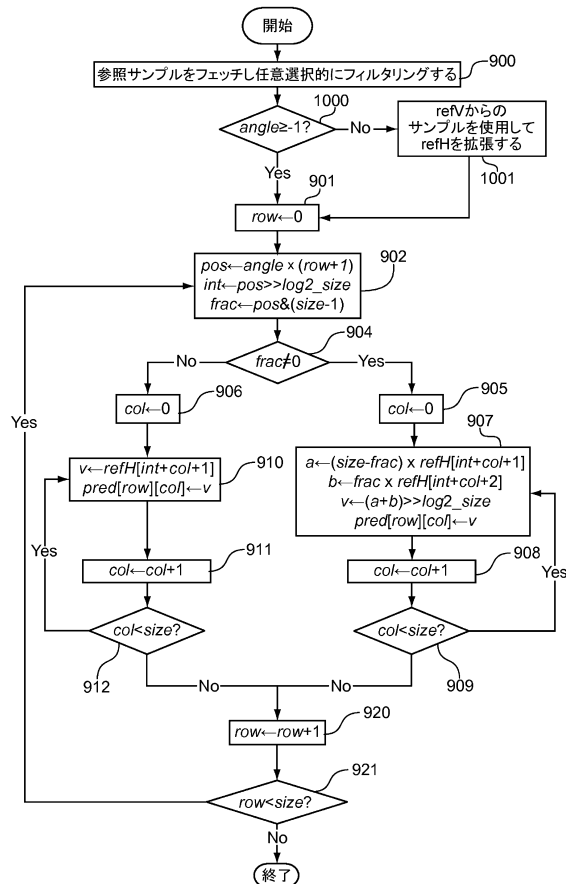


Fig. 8

【図 9】



【図 10】



【図 11 A】

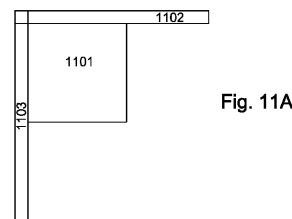


Fig. 11A

【図 11 B】

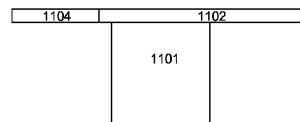
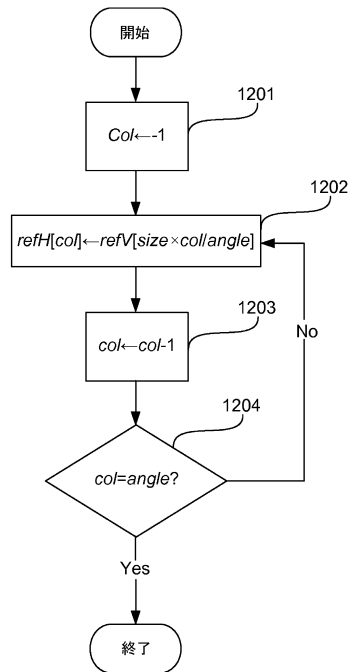
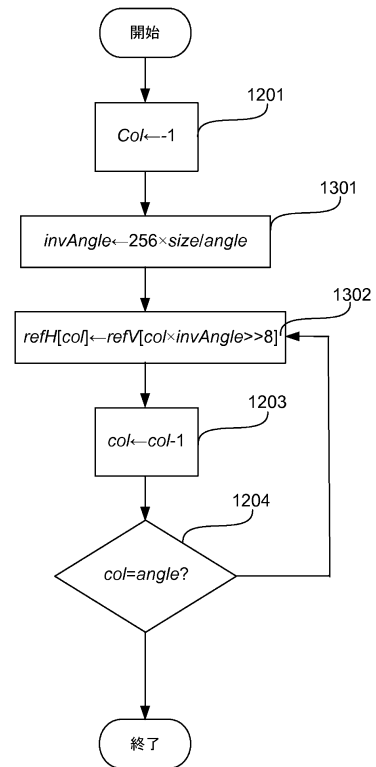


Fig. 11B

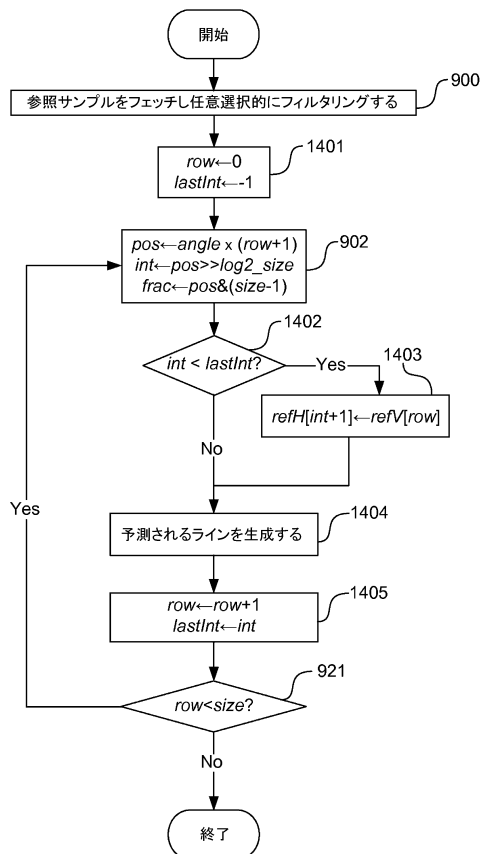
【図 12】



【図 13】



【図 14】



フロントページの続き

(51)Int.Cl. F I
H 0 4 N 19/593 (2014.01) H 0 4 N 19/593

前置審査

(72)発明者 ボッセン, フランク, ジャン
アメリカ合衆国, カリフォルニア州, パロ アルト, ヒルビュー アベニュー 3240,
ケア オブ ドコモ コミュニケーション ラボラトリーズ ユーエスエー インコーポレイテ
ッド

(72)発明者 タン, ショウ, ケン
シンガポール, シンガポール 808379, ジャラン シンダー 24

審査官 梅本 達雄

(56)参考文献 国際公開第2011/021839(WO, A1)
韓国公開特許第10-2011-0018189(KR, A)
特開2009-177722(JP, A)
特開2007-214641(JP, A)
米国特許出願公開第2008/0247462(US, A1)
特表2013-502808(JP, A)
Kemal Ugur, Kenneth R. Andersson, Arild Fuldseth, Appendix to Description of video cod
ing technology proposal by Tandberg Nokia Ericsson, Joint Collaborative Team on Video
Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 1st Meeting: Dresden, DE
, 2010年 4月23日, [JCTVC-A119], pp.1,39-41
TK Tan, Madhukar Budagavi, Jani Lainema, Tool Experiment 5: Simplification of Unified
Intra Prediction, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3
and ISO/IEC JTC1/SC29/WG11 2nd Meeting: Geneva, CH, 2010年 7月28日, [JCTVC-B30
5_r1], pp.1-3
Frank Bossen, TK Tan, Junya Takiue, Simplified angular intra prediction, Joint Collab
orative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 2nd
Meeting: Geneva, CH, 2010年 7月28日, [JCTVC-B093], pp.1-4
Madhukar Budagavi, Angular intra prediction and ADI simplification, Joint Collaborativ
e Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 2nd Meeti
ng: Geneva, CH, 2010年 7月28日, [JCTVC-B118], pp.1-4

(58)調査した分野(Int.Cl., DB名)
H 0 4 N 19/00 - 19/98