

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7195251号
(P7195251)

(45)発行日 令和4年12月23日(2022.12.23)

(24)登録日 令和4年12月15日(2022.12.15)

(51)国際特許分類

F I

H 0 4 N	19/13 (2014.01)	H 0 4 N	19/13
H 0 4 N	19/176 (2014.01)	H 0 4 N	19/176
H 0 4 N	19/136 (2014.01)	H 0 4 N	19/136

請求項の数 25 (全35頁)

(21)出願番号	特願2019-517817(P2019-517817)
(86)(22)出願日	平成29年8月31日(2017.8.31)
(65)公表番号	特表2019-531026(P2019-531026)
A)	
(43)公表日	令和1年10月24日(2019.10.24)
(86)国際出願番号	PCT/EP2017/071847
(87)国際公開番号	WO2018/065154
(87)国際公開日	平成30年4月12日(2018.4.12)
審査請求日	令和2年8月18日(2020.8.18)
(31)優先権主張番号	16306307.6
(32)優先日	平成28年10月5日(2016.10.5)
(33)優先権主張国・地域又は機関	欧州特許庁(EP)

(73)特許権者	319002876 インターデジタル マディソン パテント ホールディングス、エスアーエス フランス国, 75017 パリ, ル デュ コロネル モル 3
(74)代理人	100079108 弁理士 稲葉 良幸
(74)代理人	100109346 弁理士 大貫 敏史
(74)代理人	100117189 弁理士 江口 昭彦
(74)代理人	100134120 弁理士 内藤 和彦
(74)代理人	100108213 弁理士 阿部 豊隆

最終頁に続く

(54)【発明の名称】 ピクチャデータに関する構文要素を表すバイナリシンボルのシーケンスのコンテキスト
適応型二進算術コード化のための方法及びデバイス

(57)【特許請求の範囲】**【請求項1】**

ピクチャブロックのピクチャデータに関する構文要素を表すバイナリシンボルをコンテキスト適応型二進算術符号化するための方法であって、

- 前記ピクチャブロックのサイズが2の累乗である際は、前記ピクチャブロックの前記サイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術符号化することと、

- 前記ピクチャブロックの前記サイズが2の累乗ではない際は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術符号化することであって、前記ブロックサイズが2の累乗に等しい、算術符号化することとを含む、方法。

【請求項2】

前記ブロックサイズ又は前記ピクチャブロックの前記サイズが、前記ピクチャブロックの幅又は高さのいずれかである、請求項1に記載の方法。

【請求項3】

コンテキストが、

- 2の累乗に等しい異なるピクチャブロックサイズに従って少なくとも1つのコンテキストセットを得ることであって、各コンテキストセットが、特定の2の累乗に等しいピクチャブロックサイズに対して得られる、得ることと、

- 前記ピクチャブロックの前記サイズが 2 の累乗かどうかチェックすることと、
- 前記ピクチャブロックの前記サイズが 2 の累乗である場合は、前記ピクチャブロックの前記サイズに従って、前記得られた少なくとも 1 つのコンテキストセットの中からコンテキストセットを選択することと、
- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、前記得られた少なくとも 1 つのコンテキストセットの中からコンテキストセットを選択することであって、前記ブロックサイズが 2 の累乗に等しい、選択することと、
- バイナリシンボルのシーケンスのうちの前記バイナリシンボルのインデックスに従って、前記選択されたコンテキストセットの中からコンテキストを決定することと
によって、前記ピクチャブロックの前記サイズのすぐ上の又は下の前記ピクチャブロックの前記サイズ又は前記ブロックサイズのいずれかに従って決定される、請求項 1 又は 2 に記載の方法。

【請求項 4】

前記ピクチャブロックが走査順番に従って走査される際、ピクチャデータに関する前記構文要素が、前記ピクチャブロックに属する最終有意係数の空間座標のうちの 1 つを表し、前記方法が、

- 前記最終有意係数の前記空間座標のうちの 1 つが属する少なくとも 1 つの区間インデックスのセットの中から、整数値の区間を識別する区間インデックスを選択するステップと、

 - 前記選択された区間インデックスを二進化することによって前記バイナリシンボルのシーケンスを得るステップと
をさらに含み、

- 前記バイナリシンボルのシーケンスの各ビンが符号化される、請求項 3 に記載の方法。

【請求項 5】

前記最終有意係数の前記空間座標のうちの 1 つが前記区間インデックス及びオフセット値によって表されるロスレスであるように、前記選択された区間内の前記オフセット値を得ることをさらに含む、請求項 4 に記載の方法。

【請求項 6】

前記ピクチャブロックが、少なくとも 2 つのピクチャサブブロックに分割され、ピクチャデータに関する前記構文要素が、前記少なくとも 2 つのピクチャサブブロックのうちの 1 つの係数に関する有意係数フラグを表し、コンテキストが、

- 前記ピクチャブロックの前記サイズが 2 の累乗かどうかチェックすることと、
- 前記ピクチャブロックの前記サイズが 2 の累乗である場合は、前記ピクチャブロックの前記サイズに従って、ベースコンテキストインデックスを決定することと、
- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、ベースコンテキストインデックスを決定することであって、前記ブロックサイズが 2 の累乗に等しい、決定することと

によって、前記ピクチャブロックの前記サイズのすぐ上の又は下の前記ピクチャブロックの前記サイズ又は前記ブロックサイズのいずれかに従って決定され、

- 前記コンテキストのインデックスが、前記ベースコンテキストインデックスを、近隣の係数の有意係数フラグから得られたコンテキスト増分に加えることによって決定される、請求項 1 に記載の方法。

【請求項 7】

前記ピクチャブロック又はピクチャサブブロックの前記サイズが、前記ピクチャブロック又はピクチャサブブロックの幅及び / 又は高さであり、前記方法が、2 の累乗と整数値との積に等しいその高さ及び / 又はその幅を有するピクチャブロック又はピクチャサブブロックを得ることであって、前記積が 2 の累乗ではない、得ることをさらに含む、請求項 6 に記載の方法。

10

20

30

40

50

【請求項 8】

ピクチャサブブロックを得ることが、親ブロックを、2の累乗と前記整数値との積に等しいその高さ及び／又はその幅を有する2つのピクチャサブブロックに分割することを含む、請求項7に記載の方法。

【請求項 9】

ピクチャブロックのピクチャデータに関連する構文要素を表すバイナリシンボルをコンテキスト適応型二進算術復号するための方法であって、

前記ピクチャブロックのサイズが2の累乗である際は、前記ピクチャブロックの前記サイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術復号することと、

- 前記ピクチャブロックの前記サイズが2の累乗ではない際は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術復号することであって、前記ブロックサイズが2の累乗に等しい、算術復号することとを含む、方法。

【請求項 10】

前記ブロックサイズ又は前記ピクチャブロックの前記サイズが、前記ピクチャブロックの幅又は高さのいずれかである、請求項9に記載の方法。

【請求項 11】

コンテキストが、

- 2の累乗に等しい異なるピクチャブロックサイズに従って少なくとも1つのコンテキストセットを得ることであって、各コンテキストセットが、特定の2の累乗に等しいピクチャブロックサイズに対して得られる、得ることと、

- 前記ピクチャブロックの前記サイズが2の累乗かどうかチェックすることと、

- 前記ピクチャブロックの前記サイズが2の累乗である場合は、前記ピクチャブロックの前記サイズに従って、前記得られた少なくとも1つのコンテキストセットの中からコンテキストセットを選択することと、

- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、前記得られた少なくとも1つのコンテキストセットの中からコンテキストセットを選択することであって、前記ブロックサイズが2の累乗に等しい、選択することと、

- バイナリシンボルのシーケンスのうちの前記バイナリシンボルのインデックスに従つて、前記選択されたコンテキストセットの中からコンテキストを決定することと

によって、前記ピクチャブロックの前記サイズのすぐ上の又は下の前記ピクチャブロックの前記サイズ又は前記ブロックサイズのいずれかに従って決定される、請求項9又は10に記載の方法。

【請求項 12】

前記ピクチャブロックが走査順番に従って走査される際、ピクチャデータに関連する前記構文要素が、前記ピクチャブロックに属する最終有意係数の空間座標のうちの1つを表し、前記方法が、

- 前記最終有意係数の前記空間座標のうちの1つが属する少なくとも1つの区間インデックスのセットの中から、整数値の区間を識別する区間インデックスを選択するステップと、

- 前記選択された区間インデックスを二進化することによって前記バイナリシンボルのシーケンスを得るステップとをさらに含み、

- 前記バイナリシンボルのシーケンスの各ビンが復号される、請求項11に記載の方法。

【請求項 13】

前記ピクチャブロックが、少なくとも2つのピクチャサブブロックに分割され、ピクチャデータに関連する前記構文要素が、前記少なくとも2つのピクチャサブブロックのうちの

10

20

30

40

50

1つの係数に関する有意係数フラグを表し、コンテキストが、

- 前記ピクチャブロックの前記サイズが2の累乗かどうかチェックすることと、
- 前記ピクチャブロックの前記サイズが2の累乗である場合は、前記ピクチャブロックの前記サイズに従って、ベースコンテキストインデックスを決定することと、
- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、ベースコンテキストインデックスを決定することであって、前記ブロックサイズが2の累乗に等しい、決定することと

によって、前記ピクチャブロックの前記サイズのすぐ上の又は下の前記ピクチャブロックの前記サイズ又は前記ブロックサイズのいずれかに従って決定され、

- 前記コンテキストのインデックスが、前記ベースコンテキストインデックスを、近隣の係数の有意係数フラグから得られたコンテキスト増分に加えることによって決定される
- 請求項9に記載の方法。

【請求項14】

ピクチャブロックのピクチャデータに関連する構文要素を表すバイナリシンボルをコンテキスト適応型二進算術符号化するためのデバイスであって、

- 前記ピクチャブロックのサイズが2の累乗である際は、前記ピクチャブロックの前記サイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術符号化することと、

- 前記ピクチャブロックの前記サイズが2の累乗ではない際は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術符号化することであって、前記ブロックサイズが2の累乗に等しい、算術符号化することと

を行うように構成された少なくとも1つのプロセッサを含む、デバイス。

【請求項15】

前記ブロックサイズ又は前記ピクチャブロックの前記サイズが、前記ピクチャブロックの幅又は高さのいずれかである、請求項14に記載のデバイス。

【請求項16】

コンテキストが、

- 2の累乗に等しい異なるピクチャブロックサイズに従って少なくとも1つのコンテキストセットを得ることであって、各コンテキストセットが、特定の2の累乗に等しいピクチャブロックサイズに対して得られる、得ることと、

- 前記ピクチャブロックの前記サイズが2の累乗かどうかチェックすることと、
- 前記ピクチャブロックの前記サイズが2の累乗である場合は、前記ピクチャブロックの前記サイズに従って、前記得られた少なくとも1つのコンテキストセットの中からコンテキストセットを選択することと、

- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、前記得られた少なくとも1つのコンテキストセットの中からコンテキストセットを選択することであって、前記ブロックサイズが2の累乗に等しい、選択することと、

- バイナリシンボルのシーケンスのうちの前記バイナリシンボルのインデックスに従って、前記選択されたコンテキストセットの中からコンテキストを決定することと
によって、前記ピクチャブロックの前記サイズのすぐ上の又は下の前記ピクチャブロックの前記サイズ又は前記ブロックサイズのいずれかに従って決定される、請求項14又は15に記載のデバイス。

【請求項17】

前記ピクチャブロックが走査順番に従って走査される際、ピクチャデータに関連する前記構文要素が、前記ピクチャブロックに属する最終有意係数の空間座標のうちの1つを表し、前記少なくとも1つのプロセッサが、

- 前記最終有意係数の前記空間座標のうちの1つが属する少なくとも1つの区間インデックスのセットの中から、整数値の区間を識別する区間インデックスを選択することと、

10

20

30

40

50

- 前記選択された区間インデックスを二進化することによって前記バイナリシンボルのシーケンスを得ることと
をさらに行うように構成され、

- 前記バイナリシンボルのシーケンスの各ビンが符号化される、請求項 1_6 に記載のデバイス。

【請求項 18】

前記ピクチャブロックが、少なくとも 2 つのピクチャサブブロックに分割され、ピクチャデータに関する前記構文要素が、前記少なくとも 2 つのピクチャサブブロックのうちの 1 つの係数に関する有意係数フラグを表し、コンテキストが、

- 前記ピクチャブロックの前記サイズが 2 の累乗かどうかチェックすることと、 10

- 前記ピクチャブロックの前記サイズが 2 の累乗である場合は、前記ピクチャブロックの前記サイズに従って、ベースコンテキストインデックスを決定することと、

- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、ベースコンテキストインデックスを決定することであって、前記ブロックサイズが 2 の累乗に等しい、決定することと

によって、前記ピクチャブロックの前記サイズのすぐ上の又は下の前記ピクチャブロックの前記サイズ又は前記ブロックサイズのいずれかに従って決定され、

- 前記コンテキストのインデックスが、前記ベースコンテキストインデックスを、近隣の係数の有意係数フラグから得られたコンテキスト増分に加えることによって決定される、請求項 1_4 に記載のデバイス。 20

【請求項 19】

ピクチャブロックのピクチャデータに関する構文要素を表すバイナリシンボルをコンテキスト適応型二進算術符号化するための方法を 1 つ又は複数のプロセッサに実行させるための命令を含む非一時的なコンピュータ可読媒体であって、前記方法が、

- 前記ピクチャブロックのサイズが 2 の累乗である際は、前記ピクチャブロックの前記サイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術符号化することと、

- 前記ピクチャブロックの前記サイズが 2 の累乗ではない際は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術符号化することであって、前記ブロックサイズが 2 の累乗に等しい、算術符号化することと
を含む、非一時的なコンピュータ可読媒体。 30

【請求項 20】

ピクチャブロックのピクチャデータに関する構文要素を表すバイナリシンボルをコンテキスト適応型二進算術復号するための方法を 1 つ又は複数のプロセッサに実行させるための命令を含む非一時的なコンピュータ可読媒体であって、前記方法が、

- 前記ピクチャブロックのサイズが 2 の累乗である際は、前記ピクチャブロックの前記サイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術復号することと、

- 前記ピクチャブロックの前記サイズが 2 の累乗ではない際は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術復号することであって、前記ブロックサイズが 2 の累乗に等しい、算術復号することと

を含む、非一時的なコンピュータ可読媒体。 40

【請求項 21】

ピクチャブロックのピクチャデータに関する構文要素を表すバイナリシンボルをコンテキスト適応型二進算術復号するためのデバイスであって、

- 前記ピクチャブロックのサイズが 2 の累乗である際は、前記ピクチャブロックの前記サイズに従って決定されたコンテキストの関数として前記バイナリシンボルを算術復号することと、 50

- 前記ピクチャブロックの前記サイズが 2 の累乗ではない際は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って決定されたコンテキストの閲数として前記バイナリシンボルを算術復号することであって、前記ブロックサイズが 2 の累乗に等しい、算術復号することと
を行うように構成された少なくとも 1 つのプロセッサを含む、デバイス。

【請求項 2 2】

前記ブロックサイズ又は前記ピクチャブロックの前記サイズが、前記ピクチャブロックの幅又は高さのいずれかである、請求項 2 1 に記載のデバイス。

【請求項 2 3】

コンテキストが、

- 2 の累乗に等しい異なるピクチャブロックサイズに従って少なくとも 1 つのコンテキストセットを得ることであって、各コンテキストセットが、特定の 2 の累乗に等しいピクチャブロックサイズに対して得られる、得ることと、
- 前記ピクチャブロックの前記サイズが 2 の累乗かどうかチェックすることと、
- 前記ピクチャブロックの前記サイズが 2 の累乗である場合は、前記ピクチャブロックの前記サイズに従って、前記得られた少なくとも 1 つのコンテキストセットの中からコンテキストセットを選択することと、
- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、前記得られた少なくとも 1 つのコンテキストセットの中からコンテキストセットを選択することであって、前記ブロックサイズが 2 の累乗に等しい、選択することと、

- バイナリシンボルのシーケンスのうちの前記バイナリシンボルのインデックスに従つて、前記選択されたコンテキストセットの中からコンテキストを決定することと
によって、前記ピクチャブロックの前記サイズのすぐ上の又は下の前記ピクチャブロックの前記サイズ又は前記ブロックサイズのいずれかに従って決定される、請求項 2 1 又は 2 2 に記載のデバイス。

【請求項 2 4】

前記ピクチャブロックが走査順番に従って走査される際、ピクチャデータに関連する前記構文要素が、前記ピクチャブロックに属する最終有意係数の空間座標のうちの 1 つを表し、前記少なくとも 1 つのプロセッサが、

- 前記最終有意係数の前記空間座標のうちの 1 つが属する少なくとも 1 つの区間インデックスのセットの中から、整数値の区間を識別する区間インデックスを選択することと、
- 前記選択された区間インデックスを二進化することによって前記バイナリシンボルのシーケンスを得ることと

をさらに行うように構成され、

- 前記バイナリシンボルのシーケンスの各ビンが復号される、請求項 2 3 に記載のデバイス。

【請求項 2 5】

前記ピクチャブロックが、少なくとも 2 つのピクチャサブブロックに分割され、ピクチャデータに関連する前記構文要素が、前記少なくとも 2 つのピクチャサブブロックのうちの 1 つの係数に関する有意係数フラグを表し、コンテキストが、

- 前記ピクチャブロックの前記サイズが 2 の累乗かどうかチェックすることと、
- 前記ピクチャブロックの前記サイズが 2 の累乗である場合は、前記ピクチャブロックの前記サイズに従って、ベースコンテキストインデックスを決定することと、
- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、ベースコンテキストインデックスを決定することであって、前記ブロックサイズが 2 の累乗に等しい、決定することと

によって、前記ピクチャブロックの前記サイズのすぐ上の又は下の前記ピクチャブロックの前記サイズ又は前記ブロックサイズのいずれかに従って決定され、

- 前記コンテキストのインデックスが、前記ベースコンテキストインデックスを、近隣

10

20

30

40

50

の係数の有意係数フラグから得られたコンテキスト増分に加えることによって決定される
請求項 2 1 に記載のデバイス。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

1. 分野

本原理は、概して、ピクチャ / 映像符号化に関する。具体的には、本原理の技術分野は、上記ピクチャデータの復号に必要な構文要素を表すバイナリシンボルのシーケンスのコンテキスト適応型二進算術コード化に関する。

【背景技術】

【0 0 0 2】

2. 背景

本項目は、以下で説明及び / 又は特許請求される本原理の様々な態様に関連し得る当技術分野の様々な態様を読者に紹介することを意図する。この論考は、本原理の様々な態様をより理解し易くするために背景情報を読者に提供する際に役立つと考えられている。それに従って、これらの説明は、これを踏まえて読み進めるべきであり、先行技術を容認するものとして読み進めるべきではないことを理解すべきである。

【0 0 0 3】

以下では、ピクチャデータは、例えば、ピクチャの画素値に関するすべての情報や、ピクチャを視覚化及び / 又は復号するためのディスプレイ及び / 又は他の任意のデバイスによって使用することができるすべての情報を指定する特定の映像フォーマットの 1 つ又はいくつかのサンプルアレイ（画素値）を含む。静止ピクチャ（又は映像のピクチャ）は、第 1 のサンプルアレイの形状の少なくとも 1 つの成分（通常、ルマ（又は輝度）成分）を含み、場合により、少なくとも 1 つの他のサンプルアレイの形状の少なくとも 1 つの他の成分（通常、色成分）を含む。或いは、同等に、同じ情報は、従来の 3 色 R G B 表現など、1 組のカラーサンプルアレイによって表すこともできる。

【0 0 0 4】

画素値は、C 値のベクトルによって表すことができるピクチャデータであり、C は、成分の数である。ベクトルの各値は、ビット数で表され、ビット数は、画素値の最大ダイナミックレンジを定義する。

【0 0 0 5】

一般的に言えば、ピクチャブロックは、ピクチャデータの要素又はピクチャデータに関する構文要素のセットを表す。例えば、ピクチャブロックは、ピクチャに属する画素のセットである。

【0 0 0 6】

映像コード化デバイスは、空間及び時間冗長性を利用することによってピクチャデータを圧縮することを試みることができる。例えば、映像エンコーダは、近隣のブロック（以前にコード化済みのブロック）に関してピクチャブロックをコード化することによって、空間冗長性を利用することができる。同様に、映像エンコーダは、以前にコード化済みのピクチャのデータに関してピクチャブロックをコード化することによって、時間冗長性を利用することができる。具体的には、映像エンコーダは、空間的な近隣のブロックのデータから又は以前にコード化済みのピクチャのデータから現在のピクチャブロックを予測することができる。次いで、映像エンコーダは、ピクチャブロックに対する実際の画素値とピクチャブロックに対する予測画素値との差として、ピクチャブロックに対する残差を計算することができる。それに従って、ピクチャブロックに対する残差は、画素（又は空間）領域におけるそれぞれの画素の差の値を含み得る。

【0 0 0 7】

次いで、映像エンコーダは、周波数領域における比較的小さな数値の変換済み係数への画素値のエネルギーの圧縮を行うための変換を残差の値に適用することができる。次いで、映像エンコーダは、変換済み係数を量子化し、量子化された変換済み係数を走査し、量

10

20

30

40

50

子化された変換済み係数の二次元行列を、量子化された変換済み係数を含む一次元ベクトルに変換することができる。次いで、映像エンコーダは、走査済み係数をエントロピー符号化するためにエントロピーコード化プロセスを適用することができる。例示的なエントロピーコード化プロセスは、例えば、コンテキスト適応型可変長コード化（C A V L C）、コンテキスト適応型二進算術コード化（C A B A C）、構文ベースコンテキスト適応型二進算術コード化（S B A C）、確率区間区分エントロピー（P I P E）コード化又は他のエントロピー符号化方法を含み得る。

【 0 0 0 8 】

映像デコーダは、映像エンコーダによって実行される符号化技法とは一般的には逆の技法を実行することができる。一般的には逆であるが、映像デコーダは、いくつかの例では、映像エンコーダによって実行されるものと同様の技法を実行することができる。10

【 0 0 0 9 】

また、映像エンコーダは、ピクチャデータの復号において映像デコーダによって使用するために、符号化済みのピクチャデータと関連付けられた構文要素をエントロピー符号化することができる。次いで、映像デコーダは、受信されたビットストリームに含まれる構文要素又は他のデータ（映像エンコーダに関して説明されたデータを含む）に依存することができる。

【 0 0 1 0 】

構文要素のいくつかの例によれば、「最終有意係数座標」は、ピクチャブロックと関連付けられた走査順番によるピクチャブロックにおける最終非ゼロ係数の空間位置をデコーダに提供する構文要素である。次いで、ピクチャブロックにおける最終有意係数のx座標をデコーダに提供するために、「最終有意係数x座標」構文要素を符号化し、ピクチャブロックにおける最終有意係数のy座標をデコーダに提供するために、「最終有意係数y座標」構文要素を符号化することができる。「有意係数フラグ」は、ピクチャブロックにおける量子化された変換済み係数がゼロに等しいか否かを示す別の構文要素である。20

【 0 0 1 1 】

量子化された変換済み係数のレベルをエントロピーコード化する場合、映像コーダは、量子化された変換済み係数の各々のレベルをエントロピーコード化することができる。例えば、映像コーダは、非ゼロの量子化された変換済み係数の各々の絶対値を二進形式に変換することができる。このように、非ゼロの変換済み係数の各々は、例えば、1つ又は複数のビット又は「ビン」を含む単項コードを使用して、「二進化」することができる。また、量子化された変換済み係数の符号に対するビットも符号化することができる。30

【 0 0 1 2 】

それに加えて、映像デコーダがピクチャデータを復号できるようにするために、多くの他の構文要素を含めることができる。例えば、H . 2 6 4 / M P E G - 4 A V C (“Advanced video coding for generic audiovisual Services”, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Recommendation ITU-T H.264, Telecommunication Standardization Sector of ITU, February 2014) 及び高効率映像コード化（H E V C）規格（ITU-T H.265 TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU (10/2014), SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services - Coding of moving video, High efficiency video coding, Recommendation ITU-T H.265）で説明されるように、構文要素は、動きベクトル残差、変換単位コード化フラグ、コード化グループコード化フラグ、量子化された変換済み係数の大きさ（1より大きい及び2より大きい）フラグ、S A Oデータなどを表し得る。40

【 0 0 1 3 】

映像コーダは、ピクチャブロックに対する量子化された変換済み係数又は構文要素（情報）に対応するかどうかにかかわらず、各バイナリシンボルに対する確率推定を使用して、ピクチャデータのブロックに対する各バイナリシンボル（又はビン）をエントロピーコード化することができる。確率推定は、所定の二進値（例えば、「0」又は「1」）を有50

するバイナリシンボルの尤度を示し得る。確率推定は、「コンテキスト」とも呼ばれる確率モデル内に含めることができる。次いで、映像コーダは、コード化予定のバイナリシンボルに対するコンテキスト（確率モデルを含む）を決定することができる。

【0014】

構文要素を表すバイナリシンボルのシーケンスのうちのバイナリシンボルに対するコンテキストは、以前にコード化済みの近隣の構文要素の関連バイナリシンボルの値を含み得る。コンテキストは、コンテキストセットに属する場合があり、その場合、コンテキストは、上記コンテキストセット内の上記コンテキストを識別するコンテキストインデックスから得られる。

【0015】

一例として、ピクチャブロックに属する量子化された変換済み係数に関する有意係数フラグに対するコンテキストは、上記ピクチャブロックのタイプ（例えば、ピクチャブロックサイズ、ルム又はクロマ要素）及びピクチャブロックと関連付けられた走査順番によるピクチャブロック内の上記量子化された変換済み係数の空間位置を含む。

【0016】

別の例として、ピクチャブロックに対する二進化された残差変換係数の絶対値のバイナリシンボルに対するコンテキストは、絶対値を表す単項コードワード内のバイナリシンボルの空間位置及びピクチャブロックの以前にコード化済みの係数の値を含む。

【0017】

他の例では、コード化済みブロックパターン（「C B P」）を表すバイナリシンボルのシーケンスのうちのバイナリシンボルに対するコンテキストは、以前にコード化済みの近隣の構文要素（例えば、現在の構文要素の上側及び左側のもの）の関連バイナリシンボルの値を含む。

【0018】

本原理は、上記の例に限定されず、例えば、H.264 / MPEG-4 AVC 及び高効率映像コード化（HEVC）規格で定義されるものなどの任意の構文要素及びコンテキストまで拡張される。

【0019】

いかなる事例においても、各コンテキストに対して異なる確率モデルが定義される。

【0020】

バイナリシンボルをエントロピーコード化した後、最新の確率推定を反映させるために、コード化済みのバイナリシンボルの値に従って確率モデルを更新することができる。

【0021】

ピクチャブロックと関連付けられたバイナリシンボルは、1つ又は複数のコード化「パス」においてコード化することができる。例えば、第1のパスの間、映像コーダは、有意係数フラグ及び最終有意係数座標をエントロピーコード化することができる。第2のパスの間、映像コーダは、量子化された変換済み係数レベルの第1のビンをエントロピーコード化することができる。映像コーダは、ピクチャブロックの量子化された変換済み係数と関連付けられたすべての情報がコード化されるまで、コード化パスの実行を続けることができる。

【0022】

いくつかの例では、映像コーダは、コンテキスト適応型及び非コンテキスト適応型コード化の組合せを使用して、ピクチャブロックのビンをコード化することができる。例えば、1つ又は複数のパスに対し、映像コーダは、通常の算術コード化プロセスを迂回するか又は省略するためにバイパスモードを使用することができる。そのような例では、バイパスコード化済みのビンをコード化するために、固定等確率モデルを使用することができる。バイパスコード化済みのビンは、コンテキスト又は確率更新を含まない。

【0023】

コンテキスト適応型コード化を実行する際、いくつかの例では、複数のフィードバックループに起因する比較的高い連続依存性が存在し得る。

10

20

30

40

50

【 0 0 2 4 】

例えば、バイナリシンボルをコード化するための特定の確率モデルを示すコンテキストは、以前にコード化済みのバイナリシンボル（例えば、以前にコード化済みの構文要素の関連バイナリシンボル）の値による影響を受ける場合がある。

【 0 0 2 5 】

それに加えて、バイナリシンボルをコード化するために使用される確率モデルは、以前にコード化済みのバイナリシンボルの値による影響を受ける場合もある。すなわち、確率モデルは、有限状態機械の状態として維持することができる。特定の状態の各々は、特有の確率値に相当し得る。確率モデルの更新に相当する次の状態は、現在のバイナリシンボル（例えば、現在コード化されているビン）の値に依存し得る。

10

【 0 0 2 6 】

それに加えて、上記で記述されるように、特定の確率モデルは、有限状態機械の状態として維持することができ、特定の状態の各々は、特有の確率値に相当し得る。確率モデルの更新に相当する次の状態は、現在のバイナリシンボルの値に依存し得る。状態更新と呼ぶことができるそのようなフィードバックループは、演算上のボトルネックを提示し得る。例えば、フィードバックループにおける依存性が原因で、スループットの増大が難しい場合がある。すなわち、あるバイナリシンボルは、以前のバイナリシンボルからの確率モデルが更新されるまで処理されない場合がある。いくつかの例では、同じコンテキストを高い頻度で連続的に（バイナリシンボルごとに）呼び出すことができる。

【 0 0 2 7 】

コンテキスト適応型二進算術コード化（C A B A C）は、H . 2 6 4 / M P E G - 4 及びH E V Cで使用されるエントロピー符号化の形態である。

20

【 0 0 2 8 】

C A B A Cは、ロスレス圧縮技法であるが、C A B A Cが使用される映像コード化規格は、典型的には、ロッシャー圧縮用途のためのものである。

【発明の概要】**【 0 0 2 9 】**

図1は、C A B A Cで構文要素を符号化するための一般的なプロック図を示す。

【 0 0 3 0 】

基本的には、C A B A Cは、3つの基本的なステップを含む。

30

【 0 0 3 1 】

ステップ100では、バイナライザは、所定の非二進値構文要素をバイナリシンボルのシーケンス（ビnstリング）にマッピングする。二進値構文要素が与えられている際は、この初期のステップは、図1に示されるように、迂回される。

【 0 0 3 2 】

ビnstリングの現在のビンの各々に対し又は二進値構文要素の各々に対し、コード化モードに応じて、1つ又は複数の後続のステップが続く場合がある。

【 0 0 3 3 】

いわゆる通常のコード化モードでは、コンテキストモデル化ステップ110に続いて、通常のコード化ステップ120が行われる。

40

【 0 0 3 4 】

コンテキストモデル化ステップ110では、コンテキストモデルは、コード化予定のビン（通常、いくつかのコンテキストモデルのセットの中からコンテキストモデルを識別するコンテキストインデックス）に従ってコンテキストモデルを選択し、現在のビンがコード化されている時に、同期的にエンコーダ側とデコーダ側の両方で利用可能なコンテキスト情報（以前に符号化済みの構文要素又はコード化済みのバイナリシンボルなど）を考慮して、上記選択されたコンテキストモデルからコンテキストを決定する。次いで、決定されたコンテキストは、現在のビンが特有の値（例えば、「0」）を有する条件付き確率を得るために使用される。

【 0 0 3 5 】

50

次いで、通常のコード化ステップ 120 では、決定されたコンテキスト（条件付き確率）及び現在のビンの値 s （図 1 のビン値）は、上記決定されたコンテキストを考慮して現在のビンを算術コード化するエンジンに送られる。こうして、現在のビンのコード化済みの値が得られ、コンテキスト情報を更新するために使用される。これが、図 1 のコンテキストモデル更新フィードバックループである。コンテキスト情報を更新することは、ビンがコード化されている限り、各ビンに対してコンテキストを更新し、従って、ビンストリングの各ビンに対するコンテキストモデル化を徐々に改善できるようにすることを伴う。従って、CABAC エンコーダは、通常の符号化済みのビンの各々の統計的挙動を徐々に学習する。

【0036】

通常の符号化済みのビット（通常のコード化ステップ 120 の出力）は、出力ビットストリームに追加される。

【0037】

或いは、いわゆるバイパスコード化モードは、図 1 のスイッチの右下のプランチによって示されるように、明示的に割り当てられた確率モデルを使用することなく、簡略化されたコード化エンジンによって符号化（及び復号）プロセス全体を加速させるようにビンをコード化するために選ばれる。バイパスモードは、単に、出力ビットストリームにビットを書き込み、そのように得られたビット値を現在のビンに割り当てるものである。このコード化モードは、单刀直入であり、従って、速いという利点を有する。このコード化モードは、典型的には効率的であり、従って、均一な統計的分布を有する（すなわち、「1」又は「0」に等しくなる確率が等しい）ビンに使用される。

【0038】

図 2 は、バイナリシンボルのシーケンスがピクチャブロックに関する有意係数フラグを表す際の CABAC コード化の例を示す。

【0039】

本原理は構文要素を表すバイナリシンボルのいかなるシーケンスにも適用されるため、これは、非限定的な例である。

【0040】

図 2 の例では、有意係数フラグ（フラグは、ピクチャブロックの量子化された変換済み係数の各々と関連付けることができる）のブロックを表すピクチャブロック（ここでは、バイナリシンボルの）が表されている。量子化された変換済み係数が 0 に等しい場合は、このフラグは 0 に等しく、そうでない場合は、1 に等しい）。このピクチャブロックは、特有の走査順番に従って走査される。例えば、点線によって示される走査順番は、ピクチャブロックの右下隅から始まり、ピクチャブロックの左上隅へと進む。バイナリシンボル s をコード化すると想定すると、上記走査順番に従って、バイナリシンボル s に対して以前にいくつかの候補コンテキスト C_i が定義され、各コンテキスト C_i は、候補コンテキストセットの中からコンテキスト C_i を識別した $ContVal_i$ というコンテキストインデックスと関連付けられる。

【0041】

例えば、バイナリシンボル s に対して、2 つの以前にコード化済みの構文要素 $\times 1$ 及び $\times 2$ （有意係数フラグ）から 4 つの候補コンテキスト C_i ($i = 1, \dots, 4$) が定義される。各コンテキスト C_i は、 $\times 1$ 及び $\times 2$ の二進値に従って定義され、 $ContVal_i$ という特有のコンテキストインデックスは、コンテキストの各々に割り当てられる。

【0042】

例として、図 3 に示されるように、コンテキストインデックスは、バイナリシンボル s の特有の二進値を表す $valMPS$ （いわゆる、優勢確率シンボル（MPS））というビットと、確率 p を推測する対象となる確率 p' （又は状態）を表す $pStatIdx$ というビット（例えば、7 ビット）とを含む。

【0043】

例によれば、コンテキストモデル化ステップ 110 では、 $\times 1$ 及び $\times 2$ の値に従って、

10

20

30

40

50

4つの候補コンテキストからコンテキスト C_s が選択され、 $ContVals$ という関連コンテキストインデックスがバイナリシンボル s に割り当てられる。

【0044】

通常のコード化段階120では、 $ContVals$ というインデックスを有する選択されたコンテキストのコンテキストモデルの更新は、図4に示されるプロセスに従って、バイナリシンボル s がMPSに等しいか否かに応じて行われる。

【0045】

$transIdxMPS$ （バイナリシンボル s がMPSである場合）及び $transIdxLPS$ （バイナリシンボル s がMPSではない場合、すなわち、劣勢確率シンボル（LPS）である場合）の2つの表を通じてエボリューションが行われる。これらの表は、図5において $pState$ Idx というエントリに対して提供されている。10

【0046】

この例に従い、バイナリシンボル s がMPSである確率 p_{MPS} は、0から127まで、8ビットで線形量子化される。それは、

$$p_{MPS} = (p' + 64) / 127 = (pStatIdx + 64) / 127$$

によって、選択されたコンテキストインデックスから推測され、シンボル s が1である確率 p は、MPSの値に応じて p_{MPS} から明白に推測される。

MPS = 1 の場合 $p = p_{MPS}$

MPS = 0 の場合 $p = 1 - p_{MPS}$

【0047】

コンテキスト適応型コード化は、符号化済みのピクチャデータに関する異なる構文要素の統計に動的に従えるようにする強力なツールである。20

【0048】

従って、ピクチャデータ及び／又はピクチャデータに関する構文要素のエントロピーコード化のためのCABACの使用により、それらの統計をモデル化するために、HEVC/H265の多くのコンテキスト（最大で数百）が広範囲にわたって使用されるようになった。

【0049】

コンテキストインデックスを選択することは、多くの物事に依存し、従って、構文要素の統計を捕らえるために莫大な数のコンテキストを使用することはよくあることであり、従って、CABACの性能が最適化される。しかし、バイナリシンボルの1つ又は複数のシーケンスのすべてのバイナリシンボルを符号化するため、コンテキストはメモリに格納されるものであるため、コンテキストの数の増加は、メモリ及び処理の複雑性の観点から要件を増加する。それが、構文要素を符号化するためにCABACを実装する映像デコーダの効率的なハードウェア実現可能性を保証するためにコンテキストの数の制限が重要である理由である。30

【0050】

従って、課題の1つは、コンテキストの数をそれほど増加することなく、CABACによって提供される圧縮能力を増大することで利益を得ることである。

【0051】

図6は、図1の方法に従ってコード化されたビットを含む入力ビットストリームに与えられた構文要素のCABAC復号を描写する一般的なブロック図を示す。入力ビットストリームは、HEVC仕様に準拠し得る。40

【0052】

復号の何れの時点においても、どの構文要素を次に復号すべきかをデコーダは知っていることが想定される。このことは、標準化されたビットストリーム構文及び復号プロセスにおいて完全に指定されている。その上、復号予定の現在の構文要素がどのように二進化されるか（すなわち、ビンストリングとして表され、各ビンは、「1」又は「0」に等しい）及び、各ビンを符号化するために使用されたコード化モードは何か（通常のコード化又はバイパスモード）も想定される。50

【 0 0 5 3 】

これら情報は、考慮される映像圧縮規格 H . 2 6 4 又は H E V C において完全に指定されている。

【 0 0 5 4 】

従って、ステップ 6 0 0 では、パーサは、入力ビットストリームからコード化済みのビットを復号する（構文解析する）ことによって、一連のビットを得る。

【 0 0 5 5 】

復号予定の現在のコード化済みのビンの各々に対し、上記現在のコード化済みのビンをコード化するために使用されたコード化モードに応じて、1つ又は複数の後続のステップが続く場合がある。

10

【 0 0 5 6 】

現在のコード化済みのビンをコード化するためにバイパスコード化モードが使用された際は、バイパス復号ステップ 6 1 0 では、現在のコード化済みのビンの値は、現在のコード化済みのビンの復号済みの値である。

【 0 0 5 7 】

現在のコード化済みのビンをコード化するために通常のコード化モードが使用された際は、コンテキストモデル化ステップ 1 1 0 の後に通常の復号ステップ 6 2 0 が続く。

20

【 0 0 5 8 】

図 1 及び 6 のコンテキストモデル化ステップは、厳密には同じであり、図 6 の通常の復号ステップ 6 2 0 は、図 1 の通常のコード化ステップ 1 1 0 とかなり似ている。

【 0 0 5 9 】

通常の復号ステップ 6 2 0 では、現在のコード化済みのビンの復号済みの値は、コンテキストモデル化ステップ 1 1 0 によって与えられたコンテキストから得られる。上記コンテキスト（条件付き確率）及び現在のビットの値は、上記コンテキストを考慮して現在のビンを算術復号するエンジンに送られる。こうして、現在のビンの復号済みの値が得られ、コンテキスト情報を更新するために使用される。これが、図 6 のコンテキストモデル更新フィードバックループである。コンテキスト情報を更新することは、ビンが復号されている限り、各ビンに対してコンテキストを更新し、従って、各ビンに対するコンテキストモデル化を徐々に改善できるようにすることを伴う。従って、C A B A C デコーダは、現在のビンのオリジナルの値に等しい現在のコード化済みのビンの復号済みの値を徐々に得る（ロスレス圧縮）。

30

【 0 0 6 0 】

現在のビンの復号済みの値（バイパス復号ステップ 6 1 0 又は通常の復号ステップ 6 2 0 の出力）が得られる際、現在のビンの復号済みの値は、二進値構文要素又は非二進値構文要素に関するものである。これは、考慮される映像圧縮規格 H . 2 6 4 又は H E V C によって指定することができる。次に、現在のビンの復号済みの値が二進値構文要素に関する場合は、C A B A C 復号は終了し、現在のビンの復号済みの値が非二進値構文要素に関する場合は、デコーダは、ビンの復号済みの値（ステップ 6 2 0 又は 6 1 0 の出力ビンストリング）から構文要素をどのように構築するかを知っている。

40

【 0 0 6 1 】

次いで、ステップ 6 3 0 では、変換器は、二進化（ステップ 1 0 0 ）とは逆のプロセスを適用することによって、上記出力ビンストリングを非二進値構文要素に変換する。

【 0 0 6 2 】

映像圧縮では、ピクチャ全体を单一のピクチャブロックとして符号化することは最適ではなく、コード化木の各葉を符号化する前にピクチャ全体をサブブロックに再帰的に分割することでコード化効率が増大することはよく知られている。

【 0 0 6 3 】

例えば、H E V C 規格では、符号化予定のピクチャは、典型的には正方形の形状（6 4 × 6 4 、 1 2 8 × 1 2 8 又は 2 5 6 × 2 5 6 画素）を有するコード化木単位（C T U ）に分割される。各 C T U は、圧縮領域においてコード化木によって表される。コード化木は

50

、 C T U の再帰的分割を説明する。コード化木の各葉（コード化単位）は、四分木分割が使用される場合は、正方形の形状を有するが、葉のいくつかは、四分木プラス二分木（Q T B T : Quad-tree Plus Binary Tree）分解が使用される際は、長方形の形状も有し得る。より正確に言えば、C T U の Q T B T 分解は、二段階で構成される（最初に、C T U が四分木形式で分割され、次いで、各四分木葉を二進形式でさらに分割することができる）。このことは、図 7 の右側に示されており、実線は四分木分解相を表し、破線は四分木葉に空間的に埋め込まれた二進分解を表す。コード化単位の分割は、エンコーダ側でレート歪み最適化手順を通じて決定され、レート歪み最適化手順は、最小レート歪みコストで C T U の Q T B T 表現を決定するものである。

【 0 0 6 4 】

10

H E V C 規格では、コード化単位のサイズ（すなわち、幅及び高さ）は常に 2 の累乗であり、典型的には 4 ~ 1 2 8 である。

【 0 0 6 5 】

本発明人は、コード化木がピクチャの空間構造とよりうまく整合し、それにより、より均質的なコンテンツがコード化単位に提供されるため、Q T B T 分解における非対称分割モードを可能にすることにより、コード化効率が増大されることを見出した。

【 0 0 6 6 】

非対称分割モードは、整数値 m の倍数に等しいサイズ（幅及び / 又は高さ）のコード化単位を、 $m \cdot 2^n$ に等しいサイズ（幅及び / 又は高さ）の 2 つのサブコード化単位に分割するものであり、 n は、整数値であり、 m は、 $m \cdot 2^n$ が 2 の累乗に等しくならないよう選ばれた整数値である。

20

【 0 0 6 7 】

図 8 は、 m が 3 に等しい際の 4 つの非対称分割モードの例を示す。それらの例は、単なる例示のために与えられており、5、7、...などの他の m の値まで拡張することができる。 m の倍数に等しいサイズ（幅 w 及び / 又は高さ h ）のコード化単位の符号化において非対称分割モードを導入することにより、上記非対称分割モードのうちの 1 つを通じて上記コード化単位を分割することができる。例えば、図 8 の非対称分割モード H O R _ U P （水平上側）に従って 3 の倍数に等しい幅 w 及び高さ h のコード化単位を分割することにより、それぞれの長方形のサイズ

【 数 1 】

30

$$\left(w, \frac{h}{4} \right) \text{ 及び } \left(w, \frac{3h}{4} \right)$$

の 2 つのサブコード化単位がもたらされることになる。

【 0 0 6 8 】

非対称分割モードが使用される際は、いくつかのコード化単位は、 $m \cdot 2^n$ に等しいその幅及び / 又はその高さを有する。次いで、2 の累乗に等しいサイズのコード化単位の符号化用に最初に設計された通常の H E V C ツールを、2 の累乗に等しくないコード化単位のこれらの新しいサイズに適応させる必要がある。

40

【 0 0 6 9 】

本原理によって解決される問題の 1 つは、コード化設計の最小複雑性増大と共に良い圧縮効率（レート歪み性能）を提供するように、非対称コード化単位に含まれる量子化された変換済み係数をどのように効率的にエントロピー符号化 / 復号するかである。言い換えれば、課題の 1 つは、H E V C エントロピー符号化（C A B A C）によって使用されるコンテキストの数を増加することなく、非対称分割モードを使用することによって提供される圧縮能力の増大から利益を得ることである。

【 0 0 7 0 】

3 . 概要

以下は、本原理のいくつかの態様の基本的な理解を提供するために、本原理の簡単な概

50

要を提示する。この概要は、本原理の広範な要約ではない。本原理の重要な又は必須の要素を特定することは意図しない。

【0071】

以下の概要是、単に、以下で提供されるより詳細な説明の前置きとして、簡略化された形態で、本原理のいくつかの態様を提示する。

【0072】

本原理は、ピクチャブロックに属するピクチャデータに関連する構文要素を表すバイナリシンボルのシーケンスをコンテキスト適応型二進算術コード化／復号するための方法を用いて、先行技術の欠点の少なくとも1つを改善することを目指す。上記方法は、バイナリシンボルのシーケンスのうちの各バイナリシンボルに対し、

- ピクチャブロックのサイズ及びコード化予定のバイナリシンボルに従ってコンテキストを決定することと、

- 上記決定されたコンテキストを考慮して上記バイナリシンボルを算術コード化／復号することと

を含み、上記コンテキストを決定することが、

- 上記ピクチャブロックのサイズが2の累乗である際は、上記ピクチャブロックのサイズに従ってコンテキストを決定することと、

- 上記ピクチャブロックのサイズが2の累乗ではない際は、上記ピクチャブロックのサイズのすぐ上の又は下のブロックサイズに従ってコンテキストを決定することであって、上記ブロックサイズが2の累乗に等しい、決定することと
を含むことを特徴とする。

【0073】

それらの態様の他の態様によれば、本原理は、上記の方法のステップを実行するためのデバイスと、このプログラムがコンピュータ上で実行される際に上記の方法のステップを実行するためのプログラムコード命令を含むコンピュータプログラム製品と、上記プログラムがコンピューティングデバイス上で実行される際に上記の方法のステップを実行するためのプログラムコードの命令を保持する非一時的な記憶媒体とに関する。

【0074】

本原理の具体的な本質並びに本原理の他の目的、利点、特徴及び使用は、添付の図面と併せて取り入れられる例の以下の説明から明らかになるであろう。

【0075】

4. 図面の簡単な説明

図面では、本原理の例を示す。

【図面の簡単な説明】

【0076】

【図1】C A B A Cを使用して構文要素を符号化するための一般的なブロック図を示す。

【図2】バイナリシンボルのシーケンスが映像データのブロックに関する有意性マップを表す際のコンテキスト選択段階の例を示す。

【図3】コンテキストインデックスの例を示す。

【図4】コンテキストインデックスを更新するための一般的なブロック図を示す。

【図5】コンテキストインデックス更新に通常使用される周知の表を示す。

【図6】本原理の例による、バイナリシンボルのシーケンスのうちのバイナリシンボルsをコンテキスト適応型二進算術復号するための方法を示す。

【図7】バイナリシンボルsが、例えば、映像データのブロックに関する有意性マップの有意性フラグを表す際の、図6の方法の実施形態の例を示す。

【図8】mが3に等しい際の4つの非対称分割モードの例を示す。

【図9】本原理の例による、ピクチャブロックのサイズによるバイナリシンボルに対するいくつかのコンテキストのセットの中からのコンテキストの選択（ステップ110）を示すブロック図を示す。

【図10】8×8 T Bの走査順番の例を示す。

10

20

30

40

50

【図11】「最終有意係数座標」構文要素の二進化のために使用される表を示す。

【図12】異なるTBサイズに従ってコンテキストインデックスのセットを与える表の例を示す。

【図13】異なるTBサイズに従ってコンテキストインデックスのセットを与える表の例を示す。

【図14】本原理の実施形態による、ctxIncrementというコンテキスト増分を演算する例を示す。

【図15】本原理の例による、デバイスのアーキテクチャの例を示す。

【発明を実施するための形態】

【0077】

同様の又は同じ要素は、同じ参照番号で参照される。

10

【0078】

5. 本原理の例の説明

本原理は、本原理の例が示される添付の図を参照して、以下でより完全に説明する。しかし、本原理は、多くの代替の形態で具体化することができ、本明細書に記載される例に限定されるものと解釈すべきではない。それに従って、本原理は、様々な変更形態及び代替の形態が可能であるが、その具体的な例は、図面では例として示され、本明細書で詳細に説明する。しかし、本原理を開示される特定の形態に限定するという意図は全くなく、それどころか、本開示は、請求項によって定義されるような本原理の精神及び範囲内に収まるすべての変更形態、均等形態及び代替の形態を包含することを理解すべきである。

20

【0079】

本明細書で使用される専門用語は、特定の例について説明するためだけのものであり、本原理を制限することを意図しない。本明細書で使用される場合は、「a」、「an」及び「the」の単数形は、文脈で明確に示されない限り、複数形も含むことを意図する。「含む（「comprises」、「comprising」、「includes」及び／又は「including」）」という用語は、この明細書で使用される際は、記述される特徴、整数、ステップ、動作、要素及び／又はコンポーネントの存在を指定するが、1つ又は複数の他の特徴、整数、ステップ、動作、要素、コンポーネント及び／又はそれらのグループの存在又は追加を除外しないことがさらに理解されよう。その上、ある要素が別の要素に「応答する」又は「接続される」ものとして言及されている際は、その要素は、他の要素に直接応答する又は接続することも、介在要素を存在させることもできる。対照的に、ある要素が他の要素に「直接応答する」又は「直接接続される」ものとして言及されている際は、介在要素は存在しない。本明細書で使用される場合は、「及び／又は」という用語は、関連リストアイテムのうちの1つ又は複数のありとあらゆる組合せを含み、「／」と短縮することができる。

30

【0080】

「第1の」、「第2の」などの用語は、本明細書において様々な要素を説明するために使用することができるが、これらの要素は、これらの用語によって制限されるべきではないことが理解されよう。これらの用語は、ある要素を別の要素と区別するためだけに使用される。例えば、本原理の教示から逸脱することなく、第1の要素は、第2の要素と呼ぶことができ、同様に、第2の要素は、第1の要素と呼ぶことができる。

40

【0081】

図のいくつかは、通信の主要な方向を示すために通信経路上に矢印を含むが、通信は、描寫される矢印とは反対の方向でも起こり得ることを理解されたい。

【0082】

いくつかの例は、ブロック図及び動作上のフローチャートに関して説明され、各ブロックは、指定された論理機能を実装するための1つ又は複数の実行可能命令を含む、回路素子、モジュール又はコードの一部分を表す。また、他の実装形態では、ブロックで記述される機能は、記述される順番以外でも起こり得ることにも留意すべきである。例えば、実際には、関与する機能性に応じて、連続して示される2つのブロックを実質的に同時に実行することも、場合により、ブロックを逆の順番で実行することもできる。

50

【 0 0 8 3 】

本明細書における「例によれば」又は「例では」への言及は、例と関係して説明された特定の特徴、構造又は特性を本原理の少なくとも1つの実装形態に含めることができるこ¹⁰とを意味する。本明細書の様々な場所における「例によれば」又は「例では」という記載の出現は、必ずしもすべてが同じ例を指すとは限らず、別個の又は代替の例が必ずしも他の例の相互排他的なものであるとも限らない。

【 0 0 8 4 】

請求項で見られる参照番号は、単なる例示であり、請求項の範囲への制限効果はないものとする。

【 0 0 8 5 】

明示的に説明されてはいないが、本例及び変形形態は、いかなる組合せ又は副組合せでも採用することができる。

【 0 0 8 6 】

本原理は、コンテキストセットの中からコンテキストを選択するための戦略が符号化予定のピクチャブロックのサイズに依存する C A B A C の修正バージョンを開示する。

【 0 0 8 7 】

図9は、本原理の例による、ピクチャブロックのサイズによるバイナリシンボルに対するいくつかのコンテキストのセットの中からのコンテキストの選択（ステップ110）を示すブロック図を示す。

【 0 0 8 8 】

ステップ900では、モジュールは、ピクチャブロックのサイズ（幅及び高さ）が2の累乗であるか否かをチェックする。

【 0 0 8 9 】

「はい」の場合、すなわち、上記ピクチャブロックのサイズが2の累乗である際は、ステップ910において、セレクタS L 1は、上記ピクチャブロックのサイズに従ってコンテキストを決定する。

【 0 0 9 0 】

そうでない場合は、ステップ920において、セレクタS L 2は、上記ピクチャブロックのサイズのすぐ上の又は下のブロックサイズに従ってコンテキストを決定し、上記ブロックサイズは、2の累乗である。

【 0 0 9 1 】

これらの本原理は、不均一な統計的挙動を有するピクチャデータに関するいかなる種類のピクチャ要素又は構文要素も符号化するために適用することができる。

【 0 0 9 2 】

以下では、これらの本原理の実施形態は、H E V C 規格 (ITU-T H.265 TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU (10/2014), SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services - Coding of moving video, High efficiency video coding, Recommendation ITU-T H.265) と関連して説明するが、ピクチャデータ又は構文要素を符号化するためのエントロピーコード化を必要とする他のいかなる映像符号化／復号スキームまでも拡張することができる。

【 0 0 9 3 】

H E V C 規格及び関連付けられた加入調査モデル (“Algorithm Description of Joint Exploration Test Model 3”, Document JVET-C1001_v3, Joint Video Exploration Team of ISO/IEC JTC1/SC29/WG11, 3rd meeting, 26 May - 1 June 2015, Geneva, CH) では、変換ブロック (T B) と呼ばれるピクチャブロックに属する量子化された変換済み係数は、通常、C A B A C を使用してエントロピーコード化される。T B は、例えば、上記で説明されるような、得られたコード化木のコード化単位（葉）であり得るピクチャブロックである。

【 0 0 9 4 】

10

20

30

40

50

最初に、変換ブロック T B は、コード化グループ (C G) と呼ばれる量子化された変換済み係数のサブブロックに分割される。エントロピーコード化 / 復号は、いくつかの走査パスで構成され、いくつかの走査パスは、いくつかの可能な走査パターンの中から選択された走査パターンに従って T B を走査する。

【 0 0 9 5 】

例として、図 10 に示されるように、 8×8 T B は、4つの 4×4 コード化グループ (C G) に分割される。インター予測コード化単位の場合は、対角方向走査パターンが常に使用され、 4×4 及び 8×8 イントラ予測コード化単位の場合は、選択される走査パターンは、その C U に対してアクティブなイントラ予測モードに依存する。

【 0 0 9 6 】

次いで、ある T B にわたる走査パスは、3つの走査順番 (対角方向、水平方向、垂直方向) のうちの 1 つに従って各 C G を順次処理することを含み、各 C G 内の 16 の量子化された変換済み係数は、考慮される走査順番に従って走査される。走査パスは、T B 内の最終有意 (非ゼロ) 係数から始まり、C G の左上隅に位置する量子化された変換済み係数 (通常、C G のエネルギーのほとんどを保持する) に至るまで、すべての量子化された変換済み係数を処理する。

【 0 0 9 7 】

従って、「最終有意係数 × 座標」及び「最終有意係数 y 座標」と呼ばれる 2 つの構文要素は、上記 T B と関連付けられた走査順番による T B の最終の非ゼロの量子化された変換済み係数の空間位置 (x 及び y 座標) をデコーダに提供するために、各 T B に対する H E VC 準拠エンコーダの出力ビットストリームに追加される。

【 0 0 9 8 】

量子化された変換済み係数のエントロピーコード化は、最大で 5 つの走査パスを含み得、走査パスはそれぞれ、以下の構文要素の各々のコード化専用である。

- significant-coeff-flag : これは、量子化された変換済み係数がゼロに等しいか否かを示すバイナリフラグである
- coeff-abs-level-greater1-flag : これは、量子化された変換済み係数レベルの絶対値が 1 より大きいか否かを示すバイナリフラグである
- coeff-abs-level-greater2-flag : これは、係数レベルの絶対値が 2 より大きいか否かを示す二進の量子化された変換済み係数である
- coeff-sign-flag : これは、有意 (非ゼロ) の量子化された変換済み係数の符号を示すバイナリフラグである
- coeff-abs-level-remaining : この構文要素は、量子化された変換済み係数レベルの絶対値の残存値を表す (量子化された変換済み係数値が以前のパスにおいてコード化されたものより大きい場合)

【 0 0 9 9 】

上記の 4 つの第 1 のパスのサブセットを復号することによって係数の絶対値が分かった時点で、その絶対値に関して、残存パスにおけるその係数に対するさらなる構文要素のコード化は行われなくなる。符号コード化についても同じことが言え、coeff-sign-flag は、非ゼロ係数に対してのみ送信される。

【 0 1 0 0 】

すべての走査パスは、次の C G に行く前に、その C G のすべての量子化された変換済み係数を再構成することができるまで、所定の C G に対してコード化される。

【 0 1 0 1 】

ビットストリームからの T B の全体的な復号は、デコーダに以下の構文要素を提供する上記ビットストリームの構文解析を含む。

- 1 . 「最終有意座標」構文要素
- 2 . T B の最終有意係数を含む C G から T B の左上の C G までの各連続 C G に対し、
 - a . H E V C 規格では coded_sub_block_flag と呼ばれる C G 有意性フラグ
 - b . 考慮される C G の量子化された変換済み係数の各々に対する有意係数フラグ。こ

10

20

30

40

50

れは、HEVC 規格では sig_coeff_flag という構文要素に相当する

c . 考慮される CG の非ゼロの量子化された変換済み係数に対し、

i . coeff_abs_level_greater1_flag : このフラグは、現在の係数の絶対値が 1 より高いか否かを示す。そうでない場合は、絶対値、1 に等しい場合

i i . coeff_abs_level_greater2_flag : このフラグは、現在の係数の絶対値が 1 より高いか否かを示す。そうでない場合は、絶対値、1 に等しい場合

i i i . coeff_sign_flag : これは、非ゼロ係数の符号を示す

i v . coeff_bas_level_remaining : これは、絶対値が 2 より高い係数の絶対値を符号化する

【0102】

10

本原理の以下の実施形態は、「最終有意係数座標」構文要素のコード化 / 復号及び「有意係数フラグ」構文要素のコード化 / 復号のみに焦点を置くが、その理由は、関連付けられた加入調査モデル（“Algorithm Description of Joint Exploration Test Model 3”, Document JVET-C1001_v3, Joint Video Exploration Team of ISO/IEC JTC1/SC 29/WG11, 3rd meeting, 26 May - 1 June 2015, Geneva, CH）において、それらのコード化 / 復号に使用されるコンテキストが TB のサイズに依存するためである。

【0103】

本原理の第 1 の実施形態によれば、図 9 の方法は、コード化グループ CG に関する「最終有意係数座標」構文要素のコンテキストベース算術コード化 / 復号に対して適用される。

【0104】

20

「最終有意係数 × 座標」構文要素は、TB に属する CG に関するものであり、以下によって表されるロスレスである。

- (N + 1) 区間のセットの中から、コード化予定の × 座標が含まれる整数値の区間を識別する区間インデックス（プレフィックス）(N=2 × log₂(T)-1 であり、T は、CG における変換済み係数を得るために使用される変換のサイズ（すなわち、正方形ブロックの事例では、幅又は高さ）である）、及び、

- 上記区間インデックスによって伝えられた区間内の最終有意係数の × 座標を決定するオフセット（サフィックス）

【0105】

30

方法は、上記区間インデックスによって識別された値の区間に × 座標値が属するように上記区間インデックスを得て、「最終有意係数 × 座標」値が上記プレフィックス及びサフィックスによって表されるロスレスであるように、識別された区間内の上記オフセットを得る。

【0106】

サフィックスは、図 11 に示されるように、通常、例えば、b ビット (b=max(0,[prefi x/2]-1)) にわたる固定長コードを使用することによってバイパスモードでコード化される二進値構文要素である。コード化済みサフィックスは、出力ビットストリームに追加される。

【0107】

40

区間インデックスを表すプレフィックスは、本原理の例（ステップ 110、120）に従ってコンテキストベース算術コード化された非二進値構文要素である。次いで、プレフィックスは、出力ビットストリームに追加される。

【0108】

「最終有意係数 × 座標」（「最終有意係数 y 座標」）構文要素を符号化する場合、最初に、ステップ 100（図 1）において、バイナライザは、T = 32 の際、図 11 によって示されるように、区間インデックスを切り捨て単項コードにマッピングする。1 つの座標に対する切り捨て単項コード（コンテキストベースコード化済みのビンの数もある）の最大長は、T = 4、8、16、32 に対してそれぞれ、3、5、7、9 である。実際に、例えば、T が 4 に等しい場合は、最終有意係数 × 座標は、0、1、2 又は 3 に等しいものであり得る。これらの値はそれぞれ、切り捨て単項コード「0」、「10」、「110」

50

、「11110」によって識別される。従って、Tが4に等しい場合は、最大で3つのピンが必要である。その上、Tが8に等しい場合は、最大切り捨てコードワードは、「111110」である。最終有意係数×座標は最大でも7であるため、デコーダは、「1」に等しい5つの連続ピンが復号された時点で、プレフィックスコードが「111110」であると推論することができる。これが、変換サイズが8である事例において、プレフィックスに対して、最大でも5つのピンを必要とする理由である。

【0109】

次に、ステップ900(図9)において、TBサイズが2の累乗であるかどうかをチェックする。少なくとも1つのコンテキストセットは、2の累乗に等しい異なるTBサイズに従って得られ、各コンテキストセットは、特定の2の累乗に等しいTBサイズに対して得られる。

10

【0110】

「はい」の場合は、ステップ910において、セレクタSL1は、TBサイズに従って、切り捨て単項コードの各ピンに対して、上記得られた少なくとも1つのコンテキストセットの中からコンテキストセットを選択する。

【0111】

「いいえ」の場合は、ステップ920において、選択SL2 SL1は、TBサイズのすぐ上の又は下のブロックサイズに従って、切り捨て単項コードの各ピンに対して、上記得られた少なくとも1つのコンテキストセットの中からコンテキストセットを選択し、上記ブロックサイズは、2の累乗に等しい。

20

【0112】

最後に、「最終有意係数×座標」構文要素を表す切り捨て単項コードの各ピンは、切り捨て単項コードにおけるピンのインデックスに従って、上記選択されたコンテキストセットの中から決定されたコンテキストを考慮して算術コード化される(ステップ120)。

【0113】

図12は、異なるTBサイズに従ってコンテキストインデックスのセットを与える表の例を示す。図12の表の「ピンインデックス」は、切り捨て単項コードにおけるコード化予定のピンのインデックスを識別することに留意されたい。

【0114】

この例によれば、TBサイズが2の累乗である(すなわち、T=2、4、8、16又は32)際は、コンテキストセットは、Tに従って得られる。例えば、T=8の際、コンテキストインデックスのセット(3、3、3、4、4、5)が選択される。最終的に決定されるコンテキストインデックスは、切り捨て単項コードにおける現在のピンのインデックスに依存する。例えば、切り捨て単項コードにおけるランク2のピンが現在のピンである場合は、コンテキストインデックス4が考慮される。

30

【0115】

TBサイズが2の累乗ではない(すなわち、T=3、6、12又は24)際は、コンテキストセットは、TBサイズのすぐ下のブロックサイズに従って選択される。

【0116】

例えば、TBサイズが12に等しい際は、8に等しいTBサイズに関するコンテキストインデックスのセットが選択される。決定されるコンテキストインデックスは、切り捨て単項コードの現在のピンのインデックスに依存する。例えば、切り捨て単項コードのランク2のピンが現在のピンである場合は、コンテキストインデックス4が考慮される。

40

【0117】

図13は、異なるTBサイズに従ってコンテキストインデックスのセットを与える表の例を示す。図13の表の「ピンインデックス」は、切り捨て単項コードにおけるコード化予定のピンのインデックスを識別することに留意されたい。

【0118】

この例によれば、TBサイズが2の累乗である(すなわち、T=2、4、8、16又は32)際は、コンテキストセットは、TBサイズに従って選択される。例えば、T=8の

50

際、コンテキストインデックスのセット（3、3、3、4、4、5）が選択される。決定されるコンテキストインデックスは、切り捨て単項コードにおける現在のビンのインデックスに依存する。例えば、切り捨て単項コードのランク2のビンが現在のビンである場合は、コンテキストインデックス4が考慮される。

【0119】

TBサイズが2の累乗ではない（すなわち、T = 3、6、12又は24）際は、コンテキストセットは、TBサイズのすぐ上のブロックサイズに従って選択される。

【0120】

例えば、TBサイズが12に等しい際は、16に等しいTBサイズに関するコンテキストインデックスのセットが選択される。決定されるコンテキストインデックスは、切り捨て単項コードの現在のビンのインデックスに依存する。例えば、切り捨て単項コードのランク2のビンが現在のビンである場合は、コンテキストインデックス7が考慮される。

10

【0121】

「最終有意係数×座標」に関するプレフィックスをコード化する場合、本原理は、 $m \cdot 2^n$ （2の累乗とは異なる）に等しい現在のTBサイズのすぐ上の又は下の、2の累乗であるTBサイズと関連付けられたコンテキストを再利用する。

【0122】

数学的に言えば、 $3 \cdot 2^n$ ($m = 3$) に等しいTBサイズ（幅又は高さ）の場合、TBサイズ 2^{n+1} （図12）又は 2^{n+2} （図13）に関するコンテキストセットが使用される。

20

【0123】

従って、本原理は、2の累乗に等しくないサイズのピクチャブロックに対して新しいコンテキストを定義するよりむしろ、2の累乗に等しいサイズのピクチャブロックに対して既に使用されたコンテキストを再利用する（HEVC規格で又はHEVC規格と関連付けられた加入調査モデルで定義されるように）。これにより、格納する及び取り扱うコンテキストの数の増加が避けられ、従って、2の累乗に等しくないサイズの変換ブロック（TB）の符号化／復号の複雑性の著しい増大が避けられる。

【0124】

実施形態によれば、「最終有意係数×座標」構文要素に関するプレフィックスを符号化する切り捨て単項コードのビンに対するコンテキストは、以下の通り、TBサイズに従って決定することができる（ステップ910又は920）。上記コンテキストは、少なくとも1つの以前に決定されたコンテキストセットの中から識別されたコンテキストセット自身の中からcurrCtxIdxというコンテキストインデックスによって識別される。

30

【0125】

最初に、TBの幅からプレフィックスをコード化するために、「最終有意係数×座標」に対して、baseCtxIdxというベースコンテキストインデックスが演算される。

【0126】

このベースコンテキストインデックスは、上記少なくとも1つの以前に決定されたコンテキストセットの中からコンテキストセットを識別する。

40

【0127】

ベースインデックスは、ピクチャのクロマ成分に対しては、組織的にゼロに設定することができ、ルマ成分に対しては、TBの幅wの関数g_uiLastPrefixCtxとして計算することができる。

【数2】

$$\text{baseCtxIdx} = \text{Chroma?}0: g_uiLastPrefixCtx[\log_2 \left(w \% m == 0 ? w * \frac{4}{m} : w \right)]$$

【0128】

次に、ctxIdx_shiftXというコンテキストシフト値を以下の通り計算することができる。

50

【数3】

$$CtxIdx_{shiftX} = is\ chroma? Clip3(0,2,(\gg 3)): \left(\left(log2\left(\frac{w}{m} == 0? w * \frac{4}{m}:w\right) + 1 \right) \gg 2 \right)$$

【0 1 2 9】

或いは、

【数4】

$$baseCtxIdx = Chroma? 0: g_uiLastPrefixCtx[\log2\left(w \% m == 0? w * \frac{2}{m}:w\right)]$$

$$CtxIdx_{shiftX} = is\ chroma? Clip3(0,2,(\gg 3)): \left(\left(log2\left(\frac{w}{m} == 0? w * \frac{2}{m}:w\right) + 1 \right) \gg 2 \right)$$

10

である。

【0 1 3 0】

次に、プレフィックスのビンに対して、`baseCtxIdx`というベースコンテキストインデックスの関数として、`currCtxIdx`というコンテンツインデックスが決定され、切り捨て単項コードにおけるコード化予定の現在のビンの`prefixBinIdx`というインデックスに従って、`ctxIdx_shiftX`というコンテキストシフト値が決定される。

`currCtxIdx=baseCtxIdx+(prefixBinIdx ctxIdx_shiftX)` (1)

20

プレフィックスの第1のビンに対するコンテキストインデックスは、`baseCtxIdx`というベースコンテキストインデックスに等しいものでありますことに留意されたい。

【0 1 3 1】

図6の実施形態によれば、「最終有意係数×座標」（「最終有意係数 y 座標」）構文要素は、以下の通り、入力ビットストリームから復号される。

【0 1 3 2】

`currCtxIdx`というコンテキストインデックスは、以下の通り、方程式(1)によって演算される。ステップ110において`baseCtxIdx`というベースコンテキストインデックスが得られた時点で、デコーダは、入力ビットストリームから一連の復号済みのビンを生成する。復号予定の一連のビンの各ビンに対し、エンコーダ側(方程式1)と厳密に同じ方法で得られたベースコンテキストインデックスとコンテキスト増分値(`prefixBinIdx ctxIdxShiftX`)との総和として、`currCtxIdx`というコンテキストインデックスが演算される。そのように得られた`currCtxIdx`というコンテキストインデックスは、コンテキストモデル化ステップ110によって提供されるコンテキストモデルのインデックスに相当する。次いで、決定された識別子を有するコンテキストによって、復号予定の一連のビンの現在のビン値(1又は0)が算術復号される(図6のステップ620)。

30

【0 1 3 3】

次いで、復号予定の一連のビンの「0」ビン値が復号されるか又は考慮されるTBサイズに対して「1」ビンの最大数に達した時点で、これは、一連の復号済みのビンが図11の切り捨て単項二進化プレフィックスに相当することを意味する。次いで、デコーダは、バイパス復号予定のサフィックスビンの数bを知っており、それにより、上記復号済みプレフィックスによって伝えられた区間内の最終有意係数の×座標を決定するサフィックスが提供される。

40

【0 1 3 4】

最後に、以下の通り、復号済みプレフィックス及びサフィックス値に基づいて、最終有意係数×座標(以下の方程式では、lastと記述される)が再構成される。

`prefix > 3 の場合 last=2^b(2+(prefix%2))+suffix`

それ以外の場合 `last=prefix`

【0 1 3 5】

「最終有意係数 y 座標」構文要素に対するコンテキストインデックスは、同様の方法で

50

、 T B サイズに従ってコンテキストセットの中から選択することができる。

【 0 1 3 6 】

本原理の第 2 の実施形態によれば、図 9 の方法は、T B に属するコード化グループ C G に関する「有意係数フラグ」構文要素のコンテキストベース算術コード化 / 復号に対して適用される。

【 0 1 3 7 】

「有意係数フラグ」は、C G の量子化された変換済み係数に関する二進値構文要素である。

【 0 1 3 8 】

第 1 に、以下の通り、ステップ 1 1 0 において、ctxOffsetというベースコンテキストオフセットが計算される。

```
ctxOffset=isChroma?0:indexOnTbSize ( 2 )
```

式中、indexOnTbSizeは、以下の通り演算された区間インデックスである。

```
indexOnTbSize=uiLog2BlockSize 2?0:f(uiLog2BlockSize) ( 3 )
```

式中、uiLog2BlockSizeは、T B のサイズ（エリア）に従って演算された値である。

【 0 1 3 9 】

第 2 に、最終係数から最低周波数係数に向けて C G の量子化された変換済み係数の各々が考慮される。量子化された変換済み係数の各々に対し、ctxIncrementというコンテキスト増分を、indexOnTbSizeという上記区間インデックスによって伝えられた整数値の区間内の最終有意係数の x 座標を決定するctxOffsetというベースコンテキストオフセットに加えることによって、その有意性をコード化するためのcurrCtxIdxというコンテキストインデックスが演算される。

【 0 1 4 0 】

最後に、コンテキストインデックスと関連付けられたコンテキストを考慮して「有意係数フラグ」が算術コード化される（ステップ 1 2 0）。

```
currCtxIdx=ctxOffset+ctxIncrement ( 4 )
```

【 0 1 4 1 】

実施形態によれば、ctxIncrementというコンテキスト増分は、図 1 4 に示されるように、以下の通り演算される。

【 0 1 4 2 】

ctxIncrementというコンテキスト増分は、ローカルテンプレートによってカバーされた近隣の以前にコード化された量子化された変換済み係数の「有意係数フラグ」値の値に依存する。より具体的には、ctxIncrementというコンテキスト増分は、近隣の係数の有意フラグの総和に基づいて演算される。図 1 4 は、そのようなテンプレートの例を示す。この例によれば、現在の量子化された変換済み係数（白い円によって表されている）と関連付けられたコンテキスト増分は、以下の通り計算される。

```
ctxIncrement=sigCoeffFlag(x+1,y)+sigCoeffFlag(x+2,y)+sigCoeffFlag(x,y+1)+sigCoeffFlag(x,y+2)+sigCoeffFlag(x+1,y+1) ( 5 )
```

【 0 1 4 3 】

式中、x 及び y は、考慮される T B における現在の係数の座標であり、sigCoeffFlag(x,y) は、位置 (x , y) における量子化された変換済み係数が非ゼロである場合は、1 に等しく、そうでない場合は、0 に等しい。x 0 、 x 3 、 x 1 、 x 4 、 x 2 はそれぞれ、(x + 1 , y) 、 (x + 2 , y) 、 (x , y + 1) 、 (x , y + 2) 、 (x + 1 , y + 1) のそれぞれの座標を有する近隣の係数であることに留意されたい。

【 0 1 4 4 】

異なる周波数における量子化された変換済み係数の特性を捕らえるため、T B は、最大で 3 つの領域に分割することができ、分割方法は、図 1 4 に示されるように、T B サイズにかかわらず固定される。コンテキストインデックスは、各領域に割り当てられる。

【 0 1 4 5 】

本原理の第 2 の実施形態によれば、ステップ 9 0 0 （図 9 ）において、T B の幅 w 及び

10

20

30

40

50

/ 又は高さ h が 2 の累乗であるかどうかをチェックする。

【0 1 4 6】

「はい」の場合は、ステップ 9 1 0において、セレクタ S L 1 は、TB サイズに従って決定された indexOnTbSize というベースコンテキストインデックス自体に従ってコンテキストを決定する。

【0 1 4 7】

「いいえ」の場合は、ステップ 9 2 0において、セレクタ S L 2 は、TB サイズのすぐ上の又は下のブロックサイズに従って決定された indexOnTbSize というベースコンテキストインデックス自体に従ってコンテキストを決定し、上記ブロックサイズは、2 の累乗に等しい。

10

【0 1 4 8】

実施形態によれば、indexOnTbSize というベースコンテキストインデックスは、

【数 5】

$$\begin{aligned} uiLog2BlkSize &= (\log_2(w \% m == 0 ? w * \frac{2^{\lfloor \log_2(m) \rfloor}}{m} : w) + \log_2(h \% m = \\ &= 0 ? h * \frac{2^{\lfloor \log_2(m) \rfloor}}{m} : h)) >> 1 \end{aligned}$$

によって得られ、式中、w 及び h はそれぞれ、TB の幅及び高さであり、f(.) は、数学関数である。

20

【0 1 4 9】

変形形態によれば、

【数 6】

$$\begin{aligned} uiLog2BlkSize &= (\log_2(w \% m == 0 ? w * \frac{2^{\lfloor \log_2(m) \rfloor + 1}}{m} : w) + \log_2(h \% m = \\ &= 0 ? h * \frac{2^{\lfloor \log_2(m) \rfloor + 1}}{m} : h)) >> 1 \end{aligned}$$

である。

30

【0 1 5 0】

indexOnTbSize というベースコンテンツインデックスは、特定の TB エリアに関するコンテキストの特定のセットを識別する。これは、異なるエリアを有する TB の「有意係数フラグ」をコード化するために、コンテキストの分離されたセットを採用できることを意味する。

【0 1 5 1】

ctxOffset というコンテキストオフセットは、ルマ成分の場合は、上記ベースコンテキストインデックスに等しくてもよく、クロマ成分の場合は、0 に等しい。

【0 1 5 2】

図 6 の実施形態によれば、「最終有意フラグ」構文要素は、以下の通り、入力ストリームから復号される。

40

【0 1 5 3】

最初に、少なくとも「有意係数フラグ」と関連付けられたコンテキストインデックスは、エンコーダのコンテキストモデル化ステップと同一のコンテキストモデル化ステップ 1 1 0) から得られる。

【0 1 5 4】

従って、コンテキストインデックスは、ctxOffset というベースコンテキストオフセットと ctxIndexIncrement というコンテキスト増分との総和として計算され、各々は、エンコーダ側と全く同じ方法で、方程式 (4) 及び (5) に従って計算される。

【0 1 5 5】

50

次に、上記計算されたコンテキストインデックス及び入力ビットストリームのビットを有するコンテキストモデルを考慮すると、算術復号ステップ 620 は、現在復号されている C G の現在の係数と関連付けられた「有意係数フラグ」を表すビン値を復号することができる。

【 0156 】

図 1 ~ 14 では、モジュールは、区別可能な物理ユニットと関係があつてもなくともよい機能ユニットである。例えば、これらのモジュール又はそれらのいくつかは、特有のコンポーネント又は回路にまとめるこども、ソフトウェアの機能性に寄与することもできる。それに反して、いくつかのモジュールは、別個の物理的実体から潜在的に構成することができる。本原理に適合する装置は、純粹なハードウェアを使用して、例えば、A S I C 10 、F P G A 若しくはV L S I (それぞれ、 特定用途向け集積回路 、 フィールドプログラマブルゲートアレイ 、 超大規模集積)などの専用ハードウェアを使用して、或いは、デバイスに埋め込まれたいくつかの統合電子コンポーネントから、又は、ハードウェアコンポーネント及びソフトウェアコンポーネントの混合物から実装される。

【 0157 】

図 15 は、図 1 ~ 14 に関連して説明される方法を実装するように構成することができるデバイス 1500 の例示的なアーキテクチャを表す。

【 0158 】

デバイス 1500 は、データ及びアドレスバス 1501 によってつながっている以下の要素、すなわち、

- 例えば D S P (又はデジタル信号プロセッサ) である、マイクロプロセッサ 1502 (又は C P U) 、
- R O M (又は読み取り専用メモリ) 1503 、
- R A M (又はランダムアクセスメモリ) 1504 、
- アプリケーションから送信用のデータを受信するための I / O インタフェース 1505 、及び、
- バッテリ 1506

を含む。

【 0159 】

例によれば、バッテリ 1506 は、デバイスの外部のものである。言及されるメモリの各々では、本明細書で使用される「レジスタ」という用語は、小容量のエリア (数ビット) 又は非常に大きなエリア (例えば、全プログラム又は大量の受信若しくは復号データ) に対応し得る。R O M 1503 は、プログラム及びパラメータを少なくとも含む。R O M 1503 は、本原理による技法を実行するためのアルゴリズム及び命令を格納することができる。電源を入れると、C P U 1502 は、R A M にプログラムをアップロードし、対応する命令を実行する。

【 0160 】

R A M 1504 は、プログラム (C P U 1502 によって実行され、デバイス 1500 の電源を入れた後にアップロードされる) をレジスタに含み、入力データをレジスタに含み、方法の異なる状態の中間データをレジスタに含み、方法の実行のために使用される他の変数をレジスタに含む。

【 0161 】

本明細書で説明される実装形態は、例えば、方法若しくはプロセス、装置、ソフトウェアプログラム、データストリーム、又は、信号で実装することができる。実装の単一の形態の文脈においてのみ論じられる (例えば、方法又はデバイスとしてのみ論じられる) 場合であつても、論じられる特徴の実装形態は、他の形態 (例えば、プログラム) でも実装することができる。装置は、例えば、適切なハードウェア、ソフトウェア及びファームウェアで実装することができる。方法は、例えば、コンピュータ、マイクロプロセッサ、集積回路又はプログラマブル論理デバイスを含む、例えば、一般に処理デバイスを指す装置 (例えば、プロセッサなど) で実装することができる。また、プロセッサは、例えば、コ

10

20

30

40

50

ンピュータ、携帯電話、ポータブル／携帯情報端末（「PDA」）、及び、エンドユーザ間の情報の通信を容易にする他のデバイスなどの通信デバイスも含む。

【0162】

符号化又はエンコーダの例によれば、映像データのブロックBc、構文要素又はバイナリシンボルのシーケンスは、供給源から得られる。例えば、供給源は、

- ローカルメモリ（1503又は1504）（例えば、ビデオメモリ又はRAM（若しくはランダムアクセスメモリ）、フラッシュメモリ、ROM（若しくは読み取り専用メモリ）、ハードディスク）、

- 記憶装置インターフェース（1505）（例えば、大容量記憶装置、RAM、フラッシュメモリ、ROM、光ディスク又は磁気サポートとのインターフェース）、

- 通信インターフェース（1505）（例えば、有線インターフェース（例えば、バスインターフェース、広域ネットワークインターフェース、ローカルエリアネットワークインターフェース）又は無線インターフェース（IEEE802.11インターフェース若しくはBluetooth（登録商標）インターフェースなど））、及び、

- ピクチャ捕捉回路（例えば、センサ、例えば、CCD（若しくは電荷結合素子）又はCMOS（若しくは相補型金属酸化膜半導体）など）

を含むセットに属する。

【0163】

復号又はデコーダの例によれば、復号済みデータは、送り先に送信される。具体的には、送り先は、

- ローカルメモリ（1503又は1504）（例えば、ビデオメモリ又はRAM、フラッシュメモリ、ハードディスク）、

- 記憶装置インターフェース（1505）（例えば、大容量記憶装置、RAM、フラッシュメモリ、ROM、光ディスク又は磁気サポートとのインターフェース）、

- 通信インターフェース（1505）（例えば、有線インターフェース（例えば、バスインターフェース（例えば、USB（若しくはユニバーサルシリアルバス））、広域ネットワークインターフェース、ローカルエリアネットワークインターフェース、HDMI（登録商標）（高解像度マルチメディアインターフェース）インターフェース）又は無線インターフェース（IEEE802.11インターフェース、WiFi（登録商標）若しくはBluetooth（登録商標）インターフェースなど））、

- ディスプレイ

を含むセットに属する。

【0164】

符号化又はエンコーダの例によれば、ビットストリームは、送り先に送信される。例として、ビットストリームは、ローカル又はリモートメモリ（例えば、ビデオメモリ（1504）又はRAM（1504）、ハードディスク（1503））に格納される。変形形態では、ビットストリームは、記憶装置インターフェース（1505）（例えば、大容量記憶装置、フラッシュメモリ、ROM、光ディスク又は磁気サポートとのインターフェース）に送られる、及び／又は、通信インターフェース（1505）（例えば、ポイントツーポイントリンク、通信バス、ポイントツーマルチポイントリンク又は放送ネットワークとのインターフェース）上で送信される。

【0165】

復号又はデコーダの例によれば、ビットストリームは、供給源から得られる。例示的には、ビットストリームは、ローカルメモリ（例えば、ビデオメモリ（1504）、RAM（1504）、ROM（1503）、フラッシュメモリ（1503）又はハードディスク（1503））から読み取られる。変形形態では、ビットストリームは、記憶装置インターフェース（1505）（例えば、大容量記憶装置、RAM、ROM、フラッシュメモリ、光ディスク又は磁気サポートとのインターフェース）から受信される、及び／又は、通信インターフェース（1505）（例えば、ポイントツーポイントリンク、バス、ポイントツーマルチポイントリンク又は放送ネットワークとのインターフェース）から受信される。

10

20

30

40

50

【0166】

例によれば、デバイス1500は、図1～14に関連して説明される方法を実装するよう構成され、

- モバイルデバイス、
- 通信デバイス、
- ゲームデバイス、
- タブレット（又はタブレットコンピュータ）、
- ラップトップ、
- 静止ピクチャカメラ、
- ビデオカメラ、
- 符号化チップ、
- 静止ピクチャサーバ、
- 映像サーバ（例えば、放送サーバ、ビデオオンデマンドサーバ又はウェブサーバ）
- セットトップボックス、
- T V セット、
- ディスプレイ、及び、
- 復号チップ

10

を含むセットに属する。

【0167】

本明細書で説明される様々なプロセス及び特徴の実装形態は、様々な異なる機器又はアプリケーションで具体化することができる。そのような機器の例は、エンコーダ、デコーダ、デコーダからの出力を処理するポストプロセッサ、エンコーダに入力を提供するプリプロセッサ、ビデオコーダ、ビデオデコーダ、ビデオコードック、ウェブサーバ、セットトップボックス、ラップトップ、パーソナルコンピュータ、携帯電話、P D A、及び、ピクチャ若しくは映像を処理するための他の任意のデバイス、又は、他の通信デバイスを含む。明確であるべきだが、機器は、モバイルであり得、移動車両にインストールすることさえも可能である。

20

【0168】

それに加えて、方法は、プロセッサによって実行されている命令によって実装することができ、そのような命令（及び／又は実装形態によって生成されたデータ値）は、コンピュータ可読記憶媒体上に格納することができる。コンピュータ可読記憶媒体は、コンピュータ可読プログラム製品の形態を取ることができ、コンピュータ可読プログラム製品は、1つ又は複数のコンピュータ可読媒体において具体化され、その上で具体化されたコンピュータ可読プログラムコードを有し、コンピュータ可読プログラムコードは、コンピュータによって実行可能である。コンピュータ可読記憶媒体は、本明細書で使用される場合は、情報をその中に格納するための固有の能力及びそこからの情報の回収を提供するための固有の能力が与えられた非一時的な記憶媒体と見なされる。コンピュータ可読記憶媒体は、例え、これらに限定されないが、電子、磁気、光、電磁、赤外線若しくは半導体システム、装置若しくはデバイス、又は、前述の適切な任意の組合せであり得る。当業者であれば容易に把握できるように、以下、すなわち、ポータブルコンピュータディスクケット、ハードディスク、読み取り専用メモリ（R O M）、消去型プログラム可能読み取り専用メモリ（E P R O M若しくはフラッシュメモリ）、ポータブルコンパクトディスク読み取り専用メモリ（C D - R O M）、光学記憶装置、磁気記憶装置、又は、前述の適切な任意の組合せは、本原理が当てはまるコンピュータ可読記憶媒体のより具体的な例を提供する一方で、単なる例示であり、網羅的なリストではないことを把握されたい。

30

【0169】

命令は、プロセッサ可読媒体上で有形に具体化されたアプリケーションプログラムを形成することができる。

40

【0170】

命令は、例え、ハードウェア、ファームウェア、ソフトウェア又は組合せにおけるも

50

のであり得る。命令は、例えば、オペレーティングシステム、別個のアプリケーション又はその2つの組合せで見つけることができる。従って、プロセッサは、例えば、プロセスを実行するように構成されたデバイスと、プロセスを実行するための命令を有するプロセッサ可読媒体（記憶装置など）を含むデバイスとの両方として特徴付けることができる。さらに、プロセッサ可読媒体は、命令に加えて又は命令の代わりに、実装形態によって生成されたデータ値を格納することができる。

【0171】

当業者には明らかであるように、実装形態は、例えば、格納又は送信することができる、情報を伝えるようにフォーマットされた様々な信号を生成することができる。情報は、例えば、方法を実行するための命令、又は、説明される実装形態のうちの1つによって生成されたデータを含み得る。例えば、信号は、本原理の説明される例の構文を書き込む又は読み取るための規則をデータとして伝えるように、或いは、本原理の説明される例によって書き込まれた実際の構文・値をデータとして伝えるように、フォーマットすることができる。そのような信号は、例えば、電磁波として（例えば、スペクトルの高周波部分を使用して）又はベースバンド信号としてフォーマットすることができる。フォーマットすることは、例えば、データストリームを符号化すること、及び、符号化されたデータストリームで搬送波を変調することを含み得る。信号が伝える情報は、例えば、アナログ又はデジタル情報であり得る。信号は、知られているように、様々な異なる有線又は無線リンク上で送信することができる。信号は、プロセッサ可読媒体上に格納することができる。

【0172】

多くの実装形態について説明してきた。それにもかかわらず、様々な変更を行えることが理解されよう。例えば、他の実装形態を生成するために、異なる実装形態の要素を組み合わせることも、補足することも、変更することも、除去することもできる。それに加えて、当業者は、開示されるものの代わりに、他の構造及びプロセスを代用することができ、結果として得られる実装形態は、開示される実装形態と少なくとも実質的に同じ結果を達成するために、少なくとも実質的に同じ方法で、少なくとも実質的に同じ機能を実行することを理解するであろう。それに従って、これらの及び他の実装形態は、この出願によつて企図される。

（付記1）

ピクチャブロックに属するピクチャデータに関連する構文要素を表すバイナリシンボルのシーケンスをコンテキスト適応型二進算術コード化／復号するための方法であつて、前記バイナリシンボルのシーケンスのうちの各バイナリシンボルに対し、

- 前記ピクチャブロックのサイズ及びコード化予定の前記バイナリシンボルに従つてコンテキストを決定すること（110）と、

- 前記決定されたコンテキストを考慮して前記バイナリシンボルを算術コード化／復号すること（120）と

を含む方法において、前記コンテキストを決定すること（110）が、

- 前記ピクチャブロックの前記サイズが2の累乗である際は、前記ピクチャブロックの前記サイズに従つてコンテキストを決定すること（910）と、

- 前記ピクチャブロックの前記サイズが2の累乗ではない際は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従つてコンテキストを決定すること（920）であつて、前記ブロックサイズが2の累乗に等しい、決定すること（920）とを含むことを特徴とする、方法。

（付記2）

ピクチャブロックに属するピクチャデータに関連する構文要素を表すバイナリシンボルのシーケンスをコンテキスト適応型二進算術コード化／復号するためのデバイスであつて、前記バイナリシンボルのシーケンスのうちの各バイナリシンボルに対し、

- 前記ピクチャブロックのサイズ及びコード化予定の前記バイナリシンボルに従つてコンテキストを決定するための手段と、

- 前記決定されたコンテキストを考慮して前記バイナリシンボルを算術コード化／復号

10

20

30

40

50

するための手段とを含むデバイスにおいて、前記コンテキストを決定することが、

- 前記ピクチャブロックの前記サイズが 2 の累乗である際は、前記ピクチャブロックの前記サイズに従ってコンテキストを決定することと、
- 前記ピクチャブロックの前記サイズが 2 の累乗ではない際は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロックサイズに従ってコンテキストを決定することであって、前記ブロックサイズが 2 の累乗に等しい、決定することとを含むことを特徴とする、デバイス。

(付記 3)ブロックサイズ又は前記ピクチャブロックの前記サイズが、前記ブロックの幅又は高さのいずれかである、付記 1 に記載の方法又は付記 2 に記載のデバイス。(付記 4)前記ピクチャブロックの前記サイズに従ってコンテキストを決定すること(110)が、2 の累乗に等しい異なるピクチャブロックサイズに従って少なくとも 1 つのコンテキストセットを得ることであって、各コンテキストセットが、特定の 2 の累乗に等しいピクチャブロックサイズに対して得られる、得ることと、

- 画像ブロックの前記サイズが 2 の累乗かどうかチェックすることと、
- 前記ピクチャブロックの前記サイズが 2 の累乗である場合は、前記ピクチャブロックの前記サイズに従って、前記得られた少なくとも 1 つのコンテキストセットの中からコンテキストセットを選択することと、

そうでない場合は、前記画像ブロックの前記サイズのすぐ上の又は下のブロックサイズに従って、前記得られた少なくとも 1 つのコンテキストセットの中からコンテキストセットを選択することであって、前記ブロックサイズが 2 の累乗に等しい、選択することと、- 前記バイナリシンボルのシーケンスのうちの前記バイナリシンボルのインデックスに従って、前記選択されたコンテキストセットの中からコンテキストを決定することとを含む、付記 1 若しくは 3 に記載の方法又は付記 2 若しくは 3 に記載のデバイス。(付記 5)前記ピクチャブロックが走査順番に従って走査される際、ピクチャデータに関連する前記構文要素が、前記ピクチャブロックに属する最終有意係数の空間座標のうちの 1 つを表し、前記方法又は前記デバイスが、

- 前記最終有意係数の前記空間座標のうちの 1 つが属する少なくとも 1 つの区間インデックスのセットの中から、整数値の区間を識別する区間インデックスを選択するためのステップ又は手段、

- 前記選択された区間インデックスを二進化することによって前記バイナリシンボルのシーケンスを得るためにステップ又は手段(100)のそれぞれをさらに含み、- 前記バイナリシンボルのシーケンスの各ビンがコード化される(110、910、920)、付記 4 に記載の方法又はデバイス。(付記 6)前記方法が、前記最終有意係数の前記空間座標のうちの 1 つが前記区間インデックス及び前記オフセット値によって表されるロスレスであるように、前記選択された区間内のオフセット値を得ることをさらに含む、付記 5 に記載の方法又はデバイス。(付記 7)前記ピクチャブロックが、少なくとも 2 つのピクチャサブブロックに分割され、ピクチャデータに関連する前記構文要素が、前記少なくとも 2 つのピクチャサブブロックのうちの 1 つの係数に関する有意係数フラグを表し、前記コンテキストを決定すること(110)が、

- 画像ブロックの前記サイズが 2 の累乗かどうかチェックすることと、
- 前記ピクチャブロックの前記サイズが 2 の累乗である場合は、前記ピクチャブロックの前記サイズに従って、ベースコンテキストインデックスを決定することと、
- そうでない場合は、前記ピクチャブロックの前記サイズのすぐ上の又は下のブロック

10

20

30

40

50

エリアサイズに従って、ベースコンテキストインデックスを決定することであって、前記
ブロックエリアサイズが2の累乗に等しい、決定することと
を含み、

- 前記コンテキストインデックスが、前記ベースコンテキストインデックスを、近隣の
係数の有意変換係数フラグから得られたコンテキスト増分に加えることによって決定され
る、付記1若しくは3に記載の方法又は付記2若しくは3に記載のデバイス。

(付記8)

前記ピクチャブロック又はサブブロックの前記サイズが、前記ピクチャブロック又はサ
ブブロックの幅及び/又は高さであり、前記方法が、2の累乗による整数値の積に等しい
その高さ及び/又はその幅を有するピクチャブロック又はサブブロックを得ることであつ
て、前記積が2の累乗ではない、得ることをさらに含む、付記1、3~7の何れか一項に
記載の方法又は付記2~7の何れか一項に記載のデバイス。

(付記9)

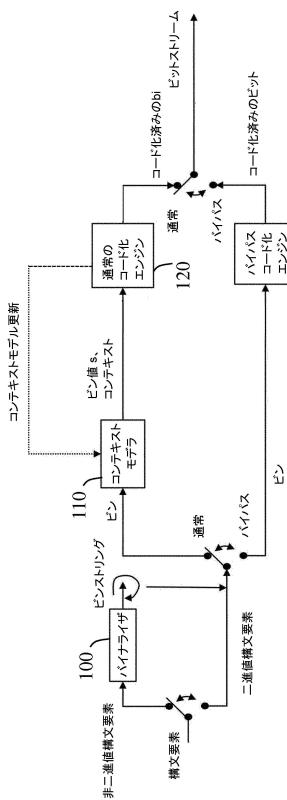
画像ブロックを得ることが、親ブロックを、2の累乗による前記整数値の積に等しいそ
の高さ及び/又はその幅を有する2つのサブブロックに分割することを含む、付記7に記
載の方法又はデバイス。

(付記10)

付記1、2~9の何れか一項に記載の方法の命令に従って符号化されたピクチャデータ
に関連する構文要素を表すバイナリシンボルのシーケンスを有する非一時的な記憶媒体。

【図面】

【図1】



【図2】

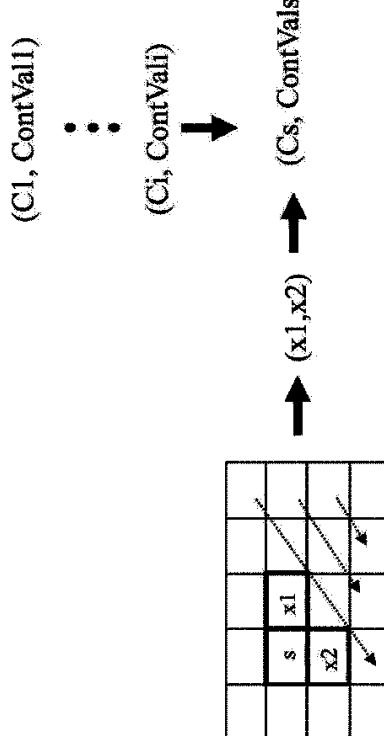


Fig. 2

10

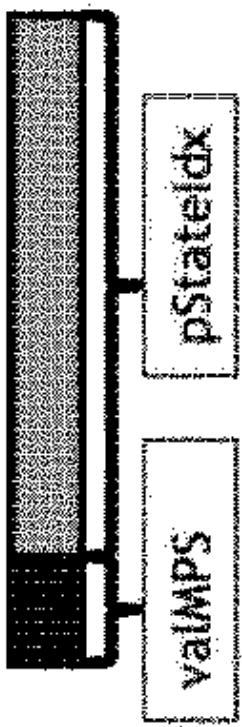
20

30

40

50

【図3】



【 四 4 】

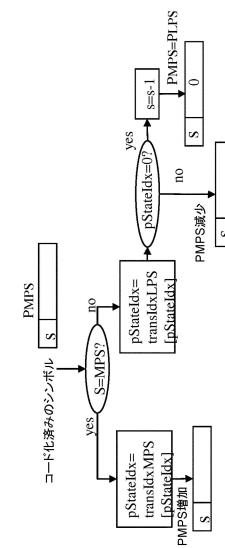


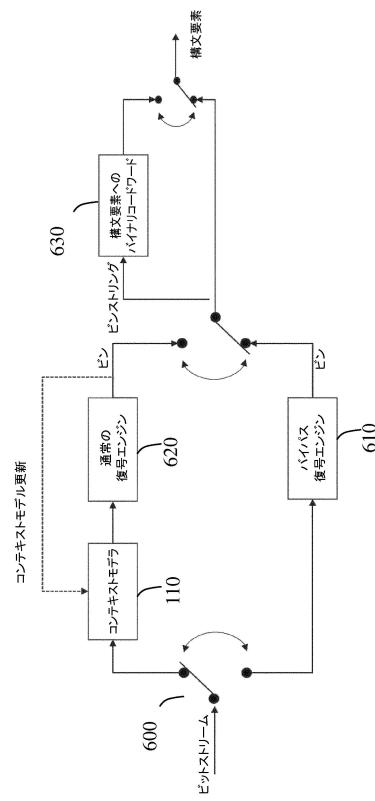
Fig. 3

〔 5 〕

Relationship	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
translates	0	0	1	2	2	4	4	5	6	7	8	9	9	11	11	12
translates	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
translates	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
translates	33	33	35	35	36	36	38	38	39	39	39	39	39	39	39	39
translates	37	38	39	26	21	22	23	24	25	26	27	28	29	30	31	32
translates	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
translates	24	25	26	26	27	27	28	29	29	30	30	31	32	32	33	33
translates	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
translates	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
translates	33	33	34	34	35	35	36	36	36	36	37	37	37	38	38	38
translates	49	50	51	52	53	54	55	56	57	58	59	60	61	62	62	63

5

〔 6 〕



【図 7】

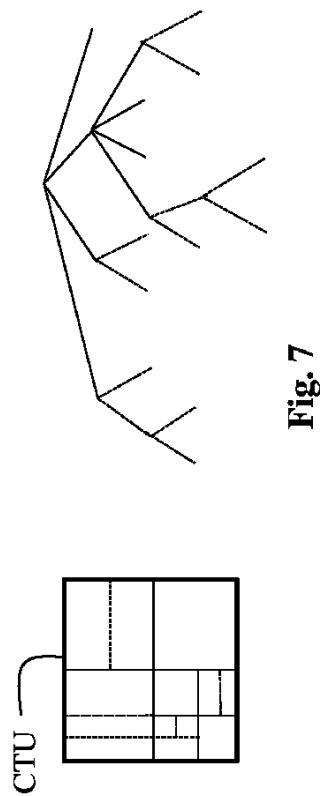


Fig. 7

【図 8】

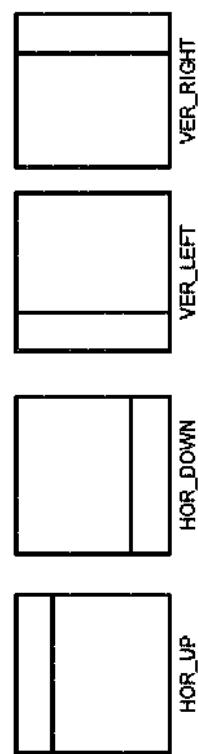
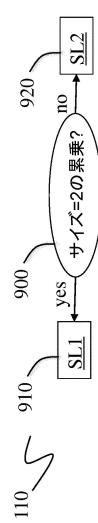
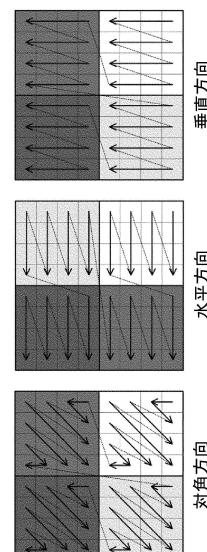


Fig. 8

【図 9】



【図 10】



10

20

30

40

50

【図 1 1】

ビンディング	0	1	2	3	4	5	6	7	8
T	3	0	1	2					
4	0	1	2						
6	3	4	4						
8	3	4	4	5					
12	6	6	7	7	8	8			
16	6	6	7	7	8	8			
24	10	10	11	11	12	12	13	13	9
32	10	10	11	11	12	12	13	13	14
T	3	15	16						
4	15	16	17						
6	15	16	16	16					
8	15	16	16	16	17				
12	15	15	15	15	16	16			
16	15	15	15	15	16	16	16		
24	15	15	15	15	15	16	16	16	
32	15	15	15	15	15	16	16	16	17

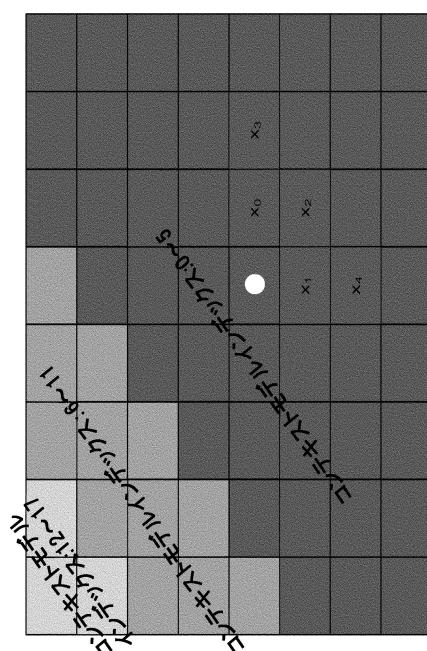
【図 1 2】

位置	プレブックス 切り捨てる箇所 (通常のモード)	サブブックス 固定長 (ハイバスモード)	サブブックス範囲
0	0	—	—
1	10	—	—
2	110	—	—
3	1110	—	—
4.5	11110	X	0から1
6-7	111110	XX	0から3
8-11	1111110	XXX	0から7
12-15	11111110	XXXX	0から7
16-23	111111110	XXXXX	0から7
24-31	111111111	XXXXX	0から7

【図 1 3】

ビンディング	0	1	2	3	4	5	6	7	8
T	3	0	1	2					
4	0	1	2						
6	3	4	4	5					
8	3	4	4	5					
12	6	6	7	7	8	8	9		
16	6	6	7	7	8	8	9		
24	6	6	7	7	8	8	9		
32	10	10	11	11	12	12	13	13	14

【図 1 4】



【図 15】

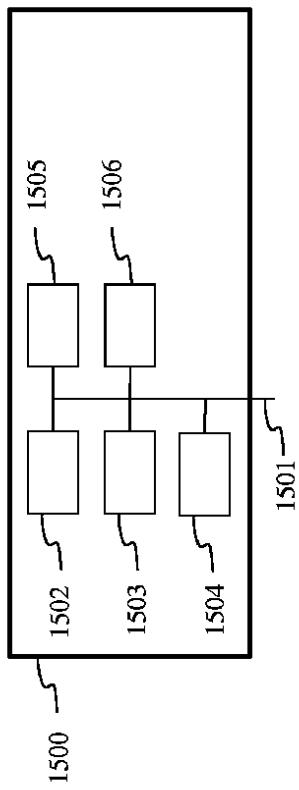


Fig. 15

10

20

30

40

50

フロントページの続き

(72)発明者 ルリアネック , フアブリス
フランス国 , 35576 セゾン - セビニエ , セーエス 17616 , アベニュー デ シャン ブラン 975 , テクニカラー内

(72)発明者 ラス , ガガン
フランス国 , 35576 セゾン - セビニエ , セーエス 17616 , アベニュー デ シャン ブラン 975 , テクニカラー内

(72)発明者 アーバン , フアブリス
フランス国 , 35576 セゾン - セビニエ , セーエス 17616 , アベニュー デ シャン ブラン 975 , テクニカラー内

審査官 鉢呂 健

(56)参考文献 特表2013-509788 (JP, A)
国際公開第2012/169184 (WO, A1)
SOLE, Joel et al. , Transform Coefficient Coding in HEVC , IEEE Transactions on Circuits and Systems for Video Technology , IEEE , 2012年10月05日 , Vol. 22, Issue. 12 , pp. 1765-1777 , URL: <https://ieeexplore.ieee.org/document/6324418> , DOI: 10.1109/TCSVT.2012.2223055
SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video , Recommendation ITU-T H.265 (04/2015) High efficiency video coding , ITU-T , 2015年07月09日 , pp. 104-107, 206-209 , <URL: <https://www.itu.int/rec/dologin.asp?lang=e&id=T-REC-H.265-201504-S!PDF-E&type=items>>

(58)調査した分野 (Int.Cl. , DB名)
H 04 N 19 / 00 - 19 / 98