

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号  
特許第7590397号  
(P7590397)

(45)発行日 令和6年11月26日(2024.11.26)

(24)登録日 令和6年11月18日(2024.11.18)

(51)国際特許分類

H 03 M 7/40 (2006.01)  
H 04 N 19/91 (2014.01)

F I

H 03 M 7/40  
H 04 N 19/91

7/40

19/91

## 請求項の数 31 外国語出願 (全136頁)

(21)出願番号	特願2022-187949(P2022-187949)
(22)出願日	令和4年11月25日(2022.11.25)
(62)分割の表示	特願2021-522144(P2021-522144) の分割 原出願日 令和1年7月5日(2019.7.5)
(65)公開番号	特開2023-21214(P2023-21214A)
(43)公開日	令和5年2月10日(2023.2.10)
審査請求日	令和4年12月23日(2022.12.23)
(31)優先権主張番号	18182308.9
(32)優先日	平成30年7月6日(2018.7.6)
(33)優先権主張国・地域又は機関	歐州特許庁(EP)
(31)優先権主張番号	18196400.8
(32)優先日	平成30年9月24日(2018.9.24)
(33)優先権主張国・地域又は機関	最終頁に続く

(73)特許権者	500341779 フランホーファー - ゲゼルシャフト・ ツール・フェルデルング・デル・アング ヴァンテン・フォルシュング・AIN トライガーネル・フェライン ドイツ連邦共和国, 80686 ミュン ヘン, ハンザシュトラーセ 27ツェ 100134119 弁理士 奥町 哲行
(74)代理人	キルヒホッファー・ハイナー ドイツ連邦共和国, 10587 ベルリ ン, アインシティンウファー 37 フ ラウンホーファー - インスティテュート ・フル・ナハリヒテンテヒニック, ハイ ンリッヒ - ヘルツ - インスティテュート 最終頁に続く
(72)発明者	最終頁に続く

(54)【発明の名称】 算術エンコーダ、算術デコーダ、ビデオエンコーダ、ビデオデコーダ、符号化方法、復号方法、およびコンピュータプログラム

## (57)【特許請求の範囲】

## 【請求項 1】

シンボル値を有する複数のシンボル (24, 520, 1622, 1722) を復号するための算術デコーダ (50, 500, 1620, 1720) であって、前記算術デコーダは、1つまたは複数の状態変数値 ( $s_1^k, s_2^k, 642, 644, 710, 810, 910, 1010a, 1010b, 1082a, 1082b, 1110, 1210, 1310, 1410, 1510, 1632, 1642, 1732, 1742$ ) を決定するように構成され、これは、以前に符号化された複数のシンボル値の統計を表し、および

前記算術デコーダは、前記1つまたは複数の状態変数値 ( $s_i^k$ ) に基づいて復号される1つまたは複数のシンボル値の算術復号の間隔サイズ情報 ( $p^k, R * p^k$ ) を導出するように構成され、これは複数の以前に復号されたシンボル値の統計を表し、

前記算術デコーダは、復号されるシンボルに依存して、ルックアップテーブル (A) を使用して、第1の状態変数値 ( $s_1^k$ ) を更新するように構成され、

前記算術デコーダは、以前の状態変数値を、復号されるシンボルが第1の値をとる場合に、前記以前の状態変数値が正である場合と比較して、前記以前の状態変数値が負である場合に、比較的大きい値だけ増加させるように構成され、

前記算術デコーダは、以前の状態変数値を、復号されるシンボルが、前記第1の値とは異なる第2の値をとる場合に、前記以前の状態変数値が負である場合と比較して、前記以前の状態変数値が正である場合に、比較的大きい値だけ減少させるように構成される、算

術デコーダ。

【請求項 2】

前記算術デコーダは、復号されるシンボルに依存して、前記ルックアップテーブル (A) を使用して、第 2 の状態変数値 ( $s^k_2$ ) を更新するように構成される、請求項 1 に記載の算術デコーダ。

【請求項 3】

前記算術デコーダは、異なる適合時定数を使用して、前記第 1 の状態変数値および第 2 の状態変数値を更新するように構成されている、請求項 1 または請求項 2 に記載の算術デコーダ。

【請求項 4】

前記算術デコーダは、復号されるシンボルが第 1 の値をとるか、前記第 1 の値とは異なる第 2 の値をとるかに応じて、前記ルックアップテーブルを使用して決定された値によって以前の状態変数値を選択的に増減するように構成される、請求項 1 または請求項 2 または請求項 3 に記載の算術デコーダ。

【請求項 5】

前記算術デコーダは、復号されるシンボルが第 1 の値をとる場合、所定のオフセット値 ( $z$ ) と以前に計算された第 1 の状態変数値 ( $s^k_1$ )

)、またはそのスケーリングおよび / または丸められたバージョン ( $|s^k_1 \cdot m^k_1|$ )

) の合計に依存して、前記第 1 の状態変数値を更新するときに評価される前記ルックアップテーブルのエントリのインデックスを決定するように構成され、

前記算術デコーダは、復号されるシンボルが第 2 の値をとる場合、所定のオフセット値 ( $z$ ) と以前に計算された第 1 の状態変数値の逆バージョン ( $-s^k_1$ )

)、またはそのスケーリングおよび / または丸められたバージョン ( $|-s^k_1 \cdot m^k_1|$ )

) の合計に依存して、前記第 1 の状態変数値を更新するときに評価される前記ルックアップテーブルのエントリのインデックスを決定するように構成される、請求項 1 から 4 のいずれか一項に記載の算術デコーダ。

【請求項 6】

前記算術デコーダは、復号されるシンボルが第 1 の値をとる場合、所定のオフセット値 ( $z$ ) と以前に計算された第 2 の状態変数値 ( $s^k_2$ )

)、またはそのスケーリングおよび / または丸められたバージョン ( $|s^k_2 \cdot m^k_2|$ )

) の合計に依存して、第 2 の状態変数値を更新するときに評価される前記ルックアップテーブルのエントリのインデックスを決定するように構成され、

前記算術デコーダは、復号されるシンボルが第 2 の値をとる場合、所定のオフセット値 ( $z$ ) と以前に計算された第 2 の状態変数値の逆バージョン ( $-s^k_2$ )

10

20

30

40

50

)、またはそのスケーリングおよび / または丸められたバージョン (  $|-s_2^k \cdot m_2^k|$  ) の合計に依存して、前記第 2 の状態変数値を更新するときに評価される前記ルックアップテーブルのエントリのインデックスを決定するように構成される、請求項 1 から 5 のいずれか一項に記載の算術デコーダ。

【請求項 7】

前記算術デコーダは、前記第 1 の状態変数値を更新するときに第 1 のスケーリング値を使用して前記ルックアップテーブルの評価によって返される値をスケーリングするように構成され、

前記算術デコーダは、第 2 の状態変数値を更新するときに第 2 のスケーリング値を使用して前記ルックアップテーブルの評価によって返される値をスケーリングするように構成され、

前記第 1 のスケーリング値は前記第 2 のスケーリング値とは異なる、請求項 1 から 6 のいずれか一項に記載の算術デコーダ。

【請求項 8】

前記算術デコーダは、  

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If decoded symbol is 1.} \\ s_i^k - [A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If decoded symbol is 0.} \end{cases}$$

$A$  はルックアップテーブルであり、

$z$  は所定のオフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値である

ことに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

を決定するように構成される、請求項 1 から 7 のいずれか一項に記載の算術デコーダ。

【請求項 9】

前記算術デコーダは、  

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If decoded symbol is 1.} \\ s_i^k - [A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If decoded symbol is 0.} \end{cases}$$

$A$  はルックアップテーブルであり、

$z$  は所定のオフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値である

ことに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

を決定するように構成される、請求項 1 から 2 のいずれか一項に記載の算術デコーダ。

【請求項 1 0】

前記算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If decoded symbol is 1.} \\ s_i^k + \lfloor -A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If decoded symbol is 0.} \end{cases}$$

$A$  はルックアップテーブルであり、

$z$  は所定のオフセット値であり、

$m_i^k$

10

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値である

ことに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

を決定するように構成される、請求項 1 から 2 のいずれか一項に記載の算術デコーダ。

【請求項 1 1】

20

前記算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If decoded symbol is 1.} \\ s_i^k + \lfloor -A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If decoded symbol is 0.} \end{cases}$$

$A$  はルックアップテーブルであり、

$z$  は所定のオフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

30

は 1 つまたは複数の重み付け値である

ことに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

を決定するように構成される、請求項 1 から 2 のいずれか一項に記載の算術デコーダ。

【請求項 1 2】

$A$  のエントリは、ルックアップテーブルインデックスの増加とともに単調に減少する、  
請求項 8 から 11 のいずれか一項に記載の算術デコーダ。

40

【請求項 1 3】

$A$  は

$$A[z + \hat{s}] = offset + \sum_{i=1}^{\hat{N}} \max(0, \hat{b}_i - \hat{s}) \ll \hat{a}_i$$

または、ゼロ設定または大きさの減少によって、引数の 1 つまたは複数の極値についてのみそこから逸脱して、前記更新された

$s_i^k$

50

が所定の値の範囲を残すのを回避し、  
 $offset$

、  
 $\hat{N}$

、  
 $\hat{b}_i$

、 および  
 $\hat{a}_i$

10

は所定のパラメータである、請求項 8 から 12 のいずれか一項に記載の算術デコーダ。

【請求項 14】

前記ルックアップテーブルの最後のエントリはゼロに等しい、請求項 1 から 13 のいずれか一項に記載の算術デコーダ。

【請求項 15】

$A = \{ 157, 143, 129, 115, 101, 87, 73, 59, 45, 35, 29, 23, 17, 13, 9, 5,$

4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0 \}

20

である、請求項 8 から 14 のいずれか一項に記載の算術デコーダ。

【請求項 16】

前記算術デコーダは、前記更新された状態変数値にクリッピング操作を適用して、前記更新されクリップされた状態変数値を所定の値の範囲内に保つように構成される、請求項 1 から 15 のいずれか一項に記載の算術デコーダ。

【請求項 17】

前記算術デコーダは、

前記更新された状態変数値に、

$$s_i^k = \min(\max(s_i^k, h_i^k), g_i^k)$$

30

、 に従ってクリッピング操作を適用するように構成され、

$g_i^k$

は、

$s_i^k$

の最大許容値であり、

$h_i^k$

40

は、

$s_i^k$

の最小許容値である、請求項 16 に記載の算術デコーダ。

【請求項 18】

前記算術デコーダは、異なるコンテキストに異なるスケーリング値を適用するように構成されており、前記異なるコンテキストは、異なる統計特性を有する、請求項 1 から 17 のいずれか一項に記載の算術デコーダ。

【請求項 19】

シンボル値を有する複数のシンボル ( 24 ' ' 、 410 ) を符号化するための算術エンコ

50

ーダ ( 3 4 ; 4 0 0 ) であって、

前記算術エンコーダは、1つまたは複数の状態変数値 (  $s_1^k, s_2^k ; 642, 644$  ;  $710, 810, 910, 1010a, 1010b, 1082a, 1082b ; 111$   $0, 1210, 1310, 1410, 1510, 1632, 1642, 1732, 174$  2 ) を決定するように構成され、これは、以前に符号化された複数のシンボル値の統計を表し、および

前記算術エンコーダは、前記1つまたは複数の状態変数値 (  $s_i^k$  ) に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ情報 (  $p^k, R * p^k ; 4$   $32, 534, 620, 720, 820, 820, 1020, 1084, 1120, 13$   $20, 1420, 1520, 1662, 1762$  ) を導出するように構成され、これは複数の以前に符号化されたシンボル値の統計を表し、

前記算術エンコーダは、符号化されるシンボルに依存して、ルックアップテーブル ( A ) を使用して、第1の状態変数値 (  $s^k_1$  ) を更新するように構成され、

前記算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k - \lfloor A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

A はルックアップテーブルであり、

z は所定のオフセット値であり、

$m_i^k$

は1つまたは複数の重み付け値であり、

$n_i^k$

は1つまたは複数の重み付け値である

ことに従って、1つまたは複数の更新された状態変数値

$s_i^k$

を決定するように構成される、算術エンコーダ。

【請求項 20】

シンボル値を有する複数のシンボル ( 2 4 ' ' , 4 1 0 ) を符号化するための算術エンコーダ ( 3 4 ; 4 0 0 ) であって、

前記算術エンコーダは、1つまたは複数の状態変数値 (  $s_1^k, s_2^k ; 642, 644$  ;  $710, 810, 910, 1010a, 1010b, 1082a, 1082b ; 111$   $0, 1210, 1310, 1410, 1510, 1632, 1642, 1732, 174$  2 ) を決定するように構成され、これは、以前に符号化された複数のシンボル値の統計を表し、および

前記算術エンコーダは、前記1つまたは複数の状態変数値 (  $s_i^k$  ) に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ情報 (  $p^k, R * p^k ; 4$   $32, 534, 620, 720, 820, 820, 1020, 1084, 1120, 13$   $20, 1420, 1520, 1662, 1762$  ) を導出するように構成され、これは複数の以前に符号化されたシンボル値の統計を表し、

前記算術エンコーダは、符号化されるシンボルに依存して、ルックアップテーブル ( A ) を使用して、第1の状態変数値 (  $s^k_1$  ) を更新するように構成され、

前記算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k - \lfloor A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

10

20

30

40

50

A はルックアップテーブルであり、

z は所定のオフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値である

ことに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

10

を決定するように構成される、算術エンコーダ。

【請求項 2 1】

シンボル値を有する複数のシンボル (24'、410) を符号化するための算術エンコーダ (34; 400) であって、

前記算術エンコーダは、1 つまたは複数の状態変数値 ( $s_1^k, s_2^k; 642, 644; 710; 810; 910; 1010a, 1010b; 1082a, 1082b; 1110; 1210; 1310; 1410; 1510; 1632, 1642; 1732, 1742$ ) を決定するように構成され、これは、以前に符号化された複数のシンボル値の統計を表し、および

前記算術エンコーダは、前記 1 つまたは複数の状態変数値 ( $s_i^k$ ) に基づいて符号化される 1 つまたは複数のシンボル値の算術符号化の間隔サイズ情報 ( $p^k, R * p^k; 432; 534; 620; 720; 820; 820; 1020; 1084; 1120; 1320; 1420; 1520; 1662; 1762$ ) を導出するように構成され、これは複数の以前に符号化されたシンボル値の統計を表し、

前記算術エンコーダは、符号化されるシンボルに依存して、ルックアップテーブル (A) を使用して、第 1 の状態変数値 ( $s^k_1$ ) を更新するように構成され、

前記算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

30

A はルックアップテーブルであり、

z は所定のオフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値である

ことに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

40

を決定するように構成される、算術エンコーダ。

【請求項 2 2】

シンボル値を有する複数のシンボル (24'、410) を符号化するための算術エンコーダ (34; 400) であって、

前記算術エンコーダは、1 つまたは複数の状態変数値 ( $s_1^k, s_2^k; 642, 644; 710; 810; 910; 1010a, 1010b; 1082a, 1082b; 1110; 1210; 1310; 1410; 1510; 1632, 1642; 1732, 1742$ ) を決定するように構成され、これは、以前に符号化された複数のシンボル値の統計を表し、および

前記算術エンコーダは、前記 1 つまたは複数の状態変数値 ( $s_i^k$ ) に基づいて符号化される 1 つまたは複数のシンボル値の算術符号化の間隔サイズ情報 ( $p^k, R * p^k; 432; 534; 620; 720; 820; 820; 1020; 1084; 1120; 1320; 1420; 1520; 1662; 1762$ ) を導出するように構成され、これは複数の以前に符号化されたシンボル値の統計を表し、

前記算術エンコーダは、符号化されるシンボルに依存して、ルックアップテーブル (A) を使用して、第 1 の状態変数値 ( $s^k_1$ ) を更新するように構成され、

前記算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

50

2) を決定するように構成され、これは、以前に符号化された複数のシンボル値の統計を表し、および

前記算術エンコーダは、前記1つまたは複数の状態変数値 ( $s_i^k$ ) に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ情報 ( $p^k, R * p^k$ ; 4 3 2; 5 3 4; 6 2 0; 7 2 0; 8 2 0; 8 2 0; 1 0 2 0; 1 0 8 4; 1 1 2 0; 1 3 2 0; 1 4 2 0; 1 5 2 0; 1 6 6 2; 1 7 6 2) を導出するように構成され、これは複数の以前に符号化されたシンボル値の統計を表し、

前記算術エンコーダは、符号化されるシンボルに依存して、ルックアップテーブル (A) を使用して、第1の状態変数値 ( $s_1^k$ ) を更新するように構成され、

前記算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

A はルックアップテーブルであり、

z は所定のオフセット値であり、

$m_i^k$

は1つまたは複数の重み付け値であり、

$n_i^k$

は1つまたは複数の重み付け値である

ことに従って、1つまたは複数の更新された状態変数値

$s_i^k$

を決定するように構成される、算術エンコーダ。

【請求項23】

$A = \{ 157, 143, 129, 115, 101, 87, 73, 59, 45, 35, 29, 23, 17, 13, 9, 5,$

$4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0 \}$

である、請求項19から22のいずれか一項に記載の算術エンコーダ。

【請求項24】

ビデオデコーダ (20; 1600; 1700) であって、

前記ビデオデコーダは、複数のビデオフレーム (12') を復号するように構成され、

前記ビデオデコーダは、請求項1に記載の、バイナリシーケンスの符号化表現 (14; 510; 1610; 1710) に基づいて復号されたバイナリシーケンス (24', 520; 1622; 1722) を提供するための算術デコーダ (50; 500; 1620; 1720) を備え、

前記バイナリシーケンスは、前記複数のビデオフレームのビデオコンテンツを表す、ビデオデコーダ。

【請求項25】

ビデオエンコーダ (10) であって、

前記ビデオエンコーダは、複数のビデオフレーム (12) を符号化するように構成され、前記ビデオエンコーダは、請求項18に記載の、前記複数のビデオフレームのビデオコンテンツを表すバイナリ値 (410) のシーケンスに基づいて符号化されたバイナリシーケンス (420) を提供するための算術エンコーダ (34; 400) を備える、ビデオエンコーダ。

【請求項26】

シンボル値を有する複数のシンボルを復号するための方法であって、

10

20

30

40

50

前記方法は、1つまたは複数の状態変数値( $s_1^k, s_2^k$ )を決定することを含み、これは、以前に復号された複数のシンボル値の統計を表し、および

前記方法は、前記1つまたは複数の状態変数値( $s_i^k$ )に基づいて復号される1つまたは複数のシンボル値の算術復号の間隔サイズ情報( $p^k, R * p^k$ )を導出することを含み、これは複数の以前に復号されたシンボル値の統計を表し、

前記方法は、復号されるシンボルに依存して、ルックアップテーブル(A)を使用して、第1の状態変数値( $s^k_1$ )を更新することを含み、

前記方法は、以前の状態変数値を、復号されるシンボルが第1の値をとる場合に、前記以前の状態変数値が正である場合と比較して、前記以前の状態変数値が負である場合に、比較的大きい値だけ増加させることを含み、

前記方法は、以前の状態変数値を、復号されるシンボルが、前記第1の値とは異なる第2の値をとる場合に、前記以前の状態変数値が負である場合と比較して、前記以前の状態変数値が正である場合に、比較的大きい値だけ減少させることを含む、方法。

#### 【請求項27】

シンボル値を有する複数のシンボルを符号化するための方法であって、

前記方法は、1つまたは複数の状態変数値( $s_1^k, s_2^k$ )を決定することを含み、これは、以前に符号化された複数のシンボル値の統計を表し、および

前記方法は、前記1つまたは複数の状態変数値( $s_i^k$ )に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ情報( $p^k, R * p^k$ )を導出することを含み、これは複数の以前に符号化されたシンボル値の統計を表し、

前記方法は、符号化されるシンボルに依存して、ルックアップテーブル(A)を使用して、第1の状態変数値( $s^k_1$ )を更新することを含み、

前記方法は、

$$s_i^k = \begin{cases} s_i^k + \left| A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \right|, & \text{If symbol to be encoded is 1.} \\ s_i^k - \left| A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \right|, & \text{If symbol to be encoded is 0.} \end{cases}$$

$A$ はルックアップテーブルであり、

$z$ は所定のオフセット値であり、

$m_i^k$

は1つまたは複数の重み付け値であり、

$n_i^k$

は1つまたは複数の重み付け値である

ことに従って、1つまたは複数の更新された状態変数値

$s_i^k$

を決定することを含む、方法。

#### 【請求項28】

シンボル値を有する複数のシンボルを符号化するための方法であって、

前記方法は、1つまたは複数の状態変数値( $s_1^k, s_2^k$ )を決定することを含み、これは、以前に符号化された複数のシンボル値の統計を表し、および

前記方法は、前記1つまたは複数の状態変数値( $s_i^k$ )に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ情報( $p^k, R * p^k$ )を導出することを含み、これは複数の以前に符号化されたシンボル値の統計を表し、

前記方法は、符号化されるシンボルに依存して、ルックアップテーブル(A)を使用して、第1の状態変数値( $s^k_1$ )を更新することを含み、

前記方法は、

10

20

30

40

50

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k - \lfloor A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

$A$  はルックアップテーブルであり、  
 $z$  は所定のオフセット値であり、  
 $m_i^k$

は 1 つまたは複数の重み付け値であり、  
 $n_i^k$

は 1 つまたは複数の重み付け値である  
 ことに従って、1 つまたは複数の更新された状態変数値  
 $s_i^k$

を決定することを含む、方法。

【請求項 2 9】

シンボル値を有する複数のシンボルを符号化するための方法であって、  
 前記方法は、1 つまたは複数の状態変数値 ( $s_1^k$ 、 $s_2^k$ ) を決定することを含み、これは、以前に符号化された複数のシンボル値の統計を表し、および  
 前記方法は、前記 1 つまたは複数の状態変数値 ( $s_1^k$ ) に基づいて符号化される 1 つまたは複数のシンボル値の算術符号化の間隔サイズ情報 ( $p^k$ 、 $R * p^k$ ) を導出することを含み、これは複数の以前に符号化されたシンボル値の統計を表し、

前記方法は、符号化されるシンボルに依存して、ルックアップテーブル ( $A$ ) を使用して、第 1 の状態変数値 ( $s_1^k$ ) を更新することを含み、

前記方法は、  
 $s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$

$A$  はルックアップテーブルであり、  
 $z$  は所定のオフセット値であり、  
 $m_i^k$

は 1 つまたは複数の重み付け値であり、  
 $n_i^k$

は 1 つまたは複数の重み付け値である  
 ことに従って、1 つまたは複数の更新された状態変数値  
 $s_i^k$

を決定することを含む、方法。

【請求項 3 0】

シンボル値を有する複数のシンボルを符号化するための方法であって、  
 前記方法は、1 つまたは複数の状態変数値 ( $s_1^k$ 、 $s_2^k$ ) を決定することを含み、これは、以前に符号化された複数のシンボル値の統計を表し、および  
 前記方法は、前記 1 つまたは複数の状態変数値 ( $s_1^k$ ) に基づいて符号化される 1 つまたは複数のシンボル値の算術符号化の間隔サイズ情報 ( $p^k$ 、 $R * p^k$ ) を導出することを含み、これは複数の以前に符号化されたシンボル値の統計を表し、

前記方法は、符号化されるシンボルに依存して、ルックアップテーブル ( $A$ ) を使用し

て、第1の状態変数値( $s^k_1$ )を更新することを含み、

前記方法は、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

$A$ はルックアップテーブルであり、

$z$ は所定のオフセット値であり、

$m_i^k$

10

は1つまたは複数の重み付け値であり、

$n_i^k$

は1つまたは複数の重み付け値である

ことに従って、1つまたは複数の更新された状態変数値

$s_i^k$

を決定することを含む、方法。

【請求項31】

コンピュータに方法を実行させるためのコンピュータプログラムであって、前記方法として、請求項26から30のいずれか一項に記載の方法を実行させるコンピュータプログラム。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明による実施形態は、算術エンコーダを作成する。

本発明によるさらなる実施形態は、算術デコーダを作成する。

本発明によるさらなる実施形態は、ビデオエンコーダを作成する。

本発明によるさらなる実施形態は、ビデオデコーダを作成する。

本発明によるさらなる実施形態は、複数のシンボルを符号化するための方法および複数のシンボルを復号するための方法を作成する。

30

本発明によるさらなる実施形態は、対応するコンピュータプログラムを作成する。

一般的に言えば、本発明による実施形態は、有限状態機械を使用したコンテキストモデル更新方法を作成する。

【背景技術】

【0002】

算術符号化および復号は、オーディオおよびビデオコンテンツの符号化および復号、ならびに画像、ニューラルネットワーク係数などの他のタイプの情報の符号化においても貴重な手段であることが証明されている。本発明の実施形態は、これらすべての用途に使用することができる。例えば、ビデオまたはオーディオコンテンツ(または他のタイプのコンテンツ)を表すバイナリシーケンス内のバイナリ値(例えば、シンボル)の既知の発生確率を利用して、符号化の効率を高めることができる。特に、算術符号化は、「0」と「1」の様々な確率を効率的に処理でき、微調整して確率の変化に適合できる。

40

【0003】

ただし、算術符号化と復号で最適なコーディング効率を実現するには、実際の発生頻度をよく反映した「0」と「1」の確率に関する適切な情報を入手することが重要である。

【0004】

「0」と「1」の確率に適合するために(または一般に、符号化されるシンボルの確率に適合するために)、概念は通常、値の合計(現在の)範囲内の間隔の境界を調整するために使用され、間隔の細分割を取得する(例えば、値の全範囲が、異なるバイナリ値また

50

はバイナリ値群に関連付けられた間隔に細分割されるようとする)。

【0005】

言い換えると、様々なシンボル(「0」や「1」など)の確率に関する情報を使用して、シンボルに関連付けられた間隔の幅を表す間隔サイズ情報(または同等に間隔サイズ値)を導出する(合計間隔幅は、例えば、間隔の再正規化のために、符号化または復号プロセスに応じて時間とともに変化する可能性がある)。

【発明の概要】

【発明が解決しようとする課題】

【0006】

したがって、間隔の細分割(例えば、コーディング間隔の合計の細分割)のソース統計値(例えば、状態変数値)および/または範囲値(間隔サイズ値など)を決定するための概念が必要であり、これは、計算効率と責任の間の適切なトレードオフを提供する。

10

【課題を解決するための手段】

【0007】

本発明による実施形態は、シンボル値(例えば、バイナリ値)を有する複数のシンボルを符号化するための算術エンコーダを作成し、算術エンコーダは、間隔サイズ情報( $p^k$ 、 $R * p^k$ )を、1つまたは複数のシンボル値(例えば、バイナリ値)の算術符号化が複数の状態変数値( $s_i^k$ )(例えば、インデックス $k$ によって表示される所与のコンテキストモードに関連付けられるもの)に基づいて符号化されるように導出するように構成され、これは、異なる適合時定数で以前に符号化された複数のシンボル値(例えば、バイナリ値0と1のシーケンス)の統計を表し、算術エンコーダは、ルックアップテーブル(LUT1)を使用して、第1の状態変数値( $s^k_1$ )、またはそのスケーリングおよび/または丸められたバージョン( $[s^k_1 * a^k_1]$ )をマッピングし、ルックアップテーブル(LUT1)を使用して、第2の状態変数値( $s^k_2$ )、またはそのスケーリングおよび/または丸められたバージョン( $[s^k_2 * a^k_2]$ )をマッピングするように構成され、符号化される1つまたは複数のシンボルの算術符号化のための間隔サイズを説明する間隔サイズ情報(例えば、 $p_k$ または $R * p_k$ )を取得するようとする。

20

【0008】

本発明によるこの実施形態は、ルックアップテーブルベースのマッピングが異なる適合時定数に関連付けられた状態変数値に適用される場合、間隔サイズ情報を特に良好な信頼性で取得できるという考えに基づいている。言い換えると、異なる時間スケールで統計(例えばシンボル確率)を記述する2つの状態変数値に同じルックアップテーブルを適用することにより、間隔サイズ情報を効率的な方法(1つのルックアップテーブルのみが必要なため)でありながらも、信頼性が高い(間隔サイズ情報の決定では、様々な時間スケールの統計が考慮されるため)様式で取得できる。ルックアップテーブルを使用した状態変数値のマッピングは、状態変数値に基づいて間隔サイズ情報を導出する際の重要な即時のステップと見なすことができる。任意選択で、1つまたは複数の追加のマッピングおよび/またはマッピング結果の結合は、間隔サイズ情報の導出における第1および第2の状態変数値のルックアップテーブルベースのマッピングに従うことができる。任意選択で、確率値は、第1の状態変数値および第2の状態変数値のルックアップテーブルベースのマッピングを使用して中間量として取得され得る。さらに、第1の状態変数値のルックアップテーブルベースのマッピングの結果から、および第2の状態変数値のルックアップテーブルベースのマッピングの結果から、間隔サイズ情報を導出するための異なる概念が可能である。

30

【0009】

結論として、本発明によるこの実施形態は、(少なくとも)2つのルックアップテーブルベースのマッピングを使用する第1の状態変数値および第2の状態変数値に基づく間隔サイズ情報の導出を提供する。したがって、異なる適合時定数に関連付けられている状態変数値は、個別にマッピングできるが、同じマッピングルール(ルックアップテーブルで定義)を使用して、間隔サイズ情報を決定するためのリソースの要求を適度に少なく保つ

40

50

が、異なる適合時定数または統計計算時定数で取得された複数の以前に処理された（例えば、符号化または復号された）シンボルの統計（または統計情報）の考慮（例えば、重み付けの考慮）を依然として可能にする。

#### 【0010】

好ましい実施形態では、算術エンコーダは、ルックアップテーブルを使用して、第1の状態変数値、またはそのスケーリングおよび/または丸められたバージョン（ $[s^k_1 * a^k_1]$ ）を第1の確率値（ $p^k_1$ ）にマッピングするように構成され、算術エンコーダは、ルックアップテーブルを使用して、第2の状態変数値、またはそのスケーリングおよび/または丸められたバージョン（ $[s^k_2 * a^k_2]$ ）を第2の確率値（ $p^k_2$ ）にマッピングするように構成され、算術エンコーダは、第1の確率値および第2の確率値を使用して（例えば、加重和または加重平均を使用して）結合確率値（ $p^k$ ）を取得するように構成される。10

#### 【0011】

このような概念を使用すると、シンボル確率（例えば、「1」シンボルの確率または「0」シンボルの確率）を表す結合確率値は、第1の状態変数値と第2の状態変数値に基づいて容易に導出できる。例えば、第1の敏捷性を有する以前に処理された（例えば、符号化または復号された）シンボルに続く第1の状態変数値は、第1の確率値にマッピングすることができ、また第2の敏捷性を有する以前に処理された（例えば、符号化または復号された）シンボルに続く第2の状態変数値は、第2の確率値に効率的にマッピングできる。したがって、異なる時間スケールで発生する以前に処理されたシンボルの傾向を考慮することができ、それでも非常に効率的な方法で結合確率値を導出することが可能である。第1および第2の状態変数値により、異なる適合時定数で以前に処理されたシンボルの傾向を追跡でき、状態変数値を「部分的な」確率値（第1の確率値および第2の確率値）にマッピングすることは、結合確率値に寄与し、上記の概念を使用して非常にリソース効率の高い方法で実行できる。20

#### 【0012】

好ましい実施形態では、算術エンコーダは、符号化されるシンボルが第1の値（例えば「1」）をとる場合に状態変数値を第1の方向に変更する（例えばより正になる）ように構成され、また符号化されるシンボルが第1の値とは異なる第2の値（例えば「0」）をとる場合（例えば、状態変数値が正と負の値をとることができるように）に状態変数値を第2の方向に（例えば、より負になるもの）変更するように構成され、算術エンコーダは、それぞれの状態変数値の絶対値（ $s^k_i > 0$  の場合は  $s^k_i$ 、それ以外の場合は  $-s^k_i$ ）に応じて（例えば、状態変数値の絶対値のスケーリングおよび丸められたバージョンに応じて）評価されるルックアップテーブルのエントリを決定するように構成される。30

#### 【0013】

例えば、正と負の値をとることができる状態変数値を使用することによって（例えば、以前に処理されたシンボルの履歴に応じて、また例えば、以前に処理された反対のシンボルに対して対称的に）、それぞれの状態変数値の絶対値に依存するルックアップテーブルのエントリを選択することにより、概念の効率をさらに向上させることができる。例えば、状態変数値の可能な値（または量子化された値）ごとにルックアップテーブルの専用エントリを用意する必要は、もはやなくなった。むしろ、正の状態変数値と対応する負の状態変数値の両方（つまり、絶対値は等しいが符号が反対の「逆」状態変数値）に対してルックアップテーブルのエントリを二重に使用することができる。その結果、ルックアップテーブルのエントリの数を少なく保つことができ、以前に処理された反対のシンボルに関する間隔サイズ情報の決定の「対称性」を利用することができる。40

#### 【0014】

好ましい実施形態では、算術エンコーダは、第1の状態変数値が第1の符号（例えば、正の符号）をとる場合に、第1の確率値（ $p^k_1$ ）をルックアップテーブルによって提供される値（例えば、

$UT1[[s^k_i * a^k_i]]$

10

20

30

40

50

) に設定するように構成され、算術エンコーダは、第 1 の状態変数値が第 2 の符号 ( 例えば、負の符号 ) をとる場合、第 1 の確率値 (  $p^k_1$  ) を、所定の値 ( 例えば、 1 ) からルックアップテーブルによって提供される値 ( 例えば、

$$1 - LUT1[s_i^k \cdot a_i^k]$$

) を減算することによって得られる値に設定するように構成される。

#### 【 0 0 1 5 】

このようなメカニズムを使用すると、ルックアップテーブルのエントリの数を少なく保つことができ ( 例えば、ルックアップテーブルのエントリは、それぞれの状態変数値の絶対値に基づいてのみ選択されるため ) 、一方でそれぞれの状態変数値の反対の符号の「相補的な」確率値を取得することが依然として可能である。したがって、確率値を決定するために、高度なリソース効率と低い計算の複雑さが達成される。

好ましい実施形態では、算術エンコーダは、

$$p_i^k = \begin{cases} LUT1[s_i^k \cdot a_i^k], & \text{if } s_i^k \geq 0 \\ 1 - LUT1[-s_i^k \cdot a_i^k], & \text{else.} \end{cases}$$

LUT1 は確率値を含むルックアップテーブルであり、

[ . ] はフロア演算子であり、

$s^k_i$  は  $i$  番目の状態変数値であり、

$a^k_i$  は、  $i$  番目の状態変数値に関連付けられた重み付け値である ( 例えば、  $i$  番目の状態変数値の数の範囲をルックアップテーブルのいくつかのエントリに適合させる重み付け値 ) ことにより、 2 つ以上の確率値  $p^k_i$  を決定するように構成される。

このような計算は計算効率が高く、リソース需要を適度に小さく保つことがわかっている。

#### 【 0 0 1 6 】

好ましい実施形態では、算術エンコーダは、

$$p_i^k = \begin{cases} LUT1[s_i^k \cdot a_i^k], & \text{if } |s_i^k \cdot a_i^k| \geq 0 \\ 1 - LUT1[-|s_i^k \cdot a_i^k|], & \text{else} \end{cases}$$

LUT1 は確率値を含むルックアップテーブルであり、

[ . ] はフロア演算子であり、

$s^k_i$  は  $i$  番目の状態変数値であり、

$a^k_i$  は、  $i$  番目の状態変数値に関連付けられた重み付け値である ( 例えば、  $i$  番目の状態変数値の数の範囲をルックアップテーブルのいくつかのエントリに適合させる重み付け値 ) ことにより、 2 つ以上の確率値  $p^k_i$  を決定するように構成される。

このような計算は、実際の数の表現に応じて、一部の状況においては有利であることがわかっている。特に、フロア演算子は、状態変数値の符号に関係なく、同じオペランドに適用される。特に、フロア演算子のオペランドの符号を削除する必要がないため、幾つかの計算の複雑さが軽減される。むしろ、否定は、通常は整数値であるフロア演算子の結果にのみ適用される。したがって、否定演算子を適用する複雑さは特に少程度である。言い換えれば、本明細書で説明される概念はまた、特に低度の複雑さをもたらす。

#### 【 0 0 1 7 】

好ましい実施形態では、算術エンコーダは、

$$p_k = \sum_{i=1}^N p_i^k \cdot b_i^k$$

式中  $N$  は考慮される確率値の数で ( 考慮される状態変数値の数に等しい場合がある ) 、  $b^k_i$  は、重み付け値 ( 例えば、結合確率値に対する個々の状態変数値の影響を制御する重み付け係数 ) である (  $b^k_i$  は、好ましくは、  $v \circ n$  2 の整数値の効力であり、 2 つの異なる  $b^k_i$  間の比は好ましくは、 2 の整数値の効力である )

10

20

30

40

50

ことに従って、複数の確率値  $p^k_i$  に基づいて結合確率値  $p^k$  を取得するように構成される。

異なる状態変数値に基づいて取得された確率値に異なる重み付けを適用することにより、短期統計と長期統計の結合確率値への異なる影響を考慮することができ、特に意味のある結合確率値を取得できる。

#### 【0018】

好ましい実施形態では、算術エンコーダは、2次元ルックアップテーブルを使用して、第1の状態変数値、またはそのスケーリングおよび/または丸められたバージョン ( $[s^k_1 * a^k_1]$ ) を第1のサブインターバル幅値 ( $R * p^k_1$ ) にマッピングするように構成され、そのエントリは、第1の状態変数値に依存して（例えば、確率指数  $i$  を使用するなどして、第1のルックアップテーブルエントリ座標を決定するために）、およびシンボルの符号化前の算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、 $R$ 、または  $R$  から導出される指数  $j$ ）に依存して（例えば、第2のルックアップテーブルエントリ座標を決定するために）アドレス指定され、算術エンコーダは、2次元ルックアップテーブルを使用して、第2の状態変数値、またはそのスケーリングおよび/または丸められたバージョン ( $[s^k_2 * a^k_2]$ ) を第2のサブインターバル幅値 ( $R * p^k_2$ ) にマッピングするように構成され、そのエントリは、第2の状態変数値に依存して（例えば、第1のルックアップテーブルエントリ座標を決定するために）、およびシンボルの符号化前の算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、 $R$ ）に依存してアドレス指定され（例えば、第2のルックアップテーブルエントリ座標を決定するために）、算術エンコーダは、第1のサブインターバル幅値および第2のサブインターバル幅値を使用して（例えば、加重和または加重平均を使用して）、結合されたサブインターバル幅値を取得するように構成される。

10

20

30

#### 【0019】

複数の異なる確率値と複数の異なる間隔サイズ値との乗算を反映するこのような2次元ルックアップテーブルを使用することにより、乗算の演算を省けるため、計算の複雑さを軽減することができる。例えば、2次元ルックアップテーブルのエントリを指定する2つのインデックス（行インデックスと列インデックス）の1つは、それぞれの状態変数値（またはそのスケーリングおよび/または丸められたバージョン）によって定義され、第2のインデックスは、現在の（合計）コーディング間隔サイズによって決定される。したがって、第1のインデックスおよび第2のインデックスに基づいて、2次元ルックアップテーブルの要素（エントリ）を一意に識別することができ、識別された要素は、通常、それぞれの状態変数値に関連付けられた確率値と、第2のテーブルインデックスに関連付けられたコーディング間隔サイズとの積を反映する。その結果、場合によっては読み取り専用メモリである可能性があるメモリを2次元ルックアップテーブルに費やすことにより、乗算演算の実行を省ける。これは、計算リソースの観点からも、またエネルギー消費の観点からも有利な場合がある。

#### 【0020】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブルは、そのエントリが第1および第2の状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび/または丸められたバージョン ( $[s^k * a^k]$ ;

$|s_k| \cdot a_k$

40

) を含む第1の1次元ベクトル（1次元ルックアップテーブルを形成する）、およびそのエントリがコーディング間隔サイズ情報の量子化レベルを含む第2の1次元ベクトル ( $Q_{T_2}(R)$ )

）の間の2項積として表すことができる。

このような3次元ルックアップテーブルを使用することにより、確率値とコーディング間隔サイズの異なるペア間の乗算演算をテーブルにより反映させることができる。したが

50

つて、2次元ルックアップテーブルの適切な要素を選択することにより、乗算演算の実行を省くことができる。さらに、このようなアプローチを使用すると、2次元ルックアップテーブルのエントリを非常に簡単な方法で取得できる。

#### 【0021】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブル (R a n g T a b L P S) の要素が、ベースルックアップテーブル (Base T a b L P S) に基づいて定義され、2次元ルックアップテーブルの要素の第1のグループ (またはブロック、例えば、「上半分」) は、ベースルックアップテーブルの要素と同一であるか、ベースルックアップテーブルの要素の丸められたバージョンであり、2次元ルックアップテーブルの要素の第2のグループ (またはブロック、例えば「下半分」) は、ベースルックアップテーブルの要素のスケーリングおよび丸められたバージョンである。

このようなアプローチを使用することにより、2次元ルックアップテーブルの要素の概ね指數関数的である増加または減衰を取得できる。例えば、2次元ルックアップテーブルの要素を定義して、2次元ルックアップテーブルの要素の第2のグループが、実質的に (例えば、丸めによって引き起こされる偏差を除いて) 2次元ルックアップテーブルの要素の第1のグループのスケーリングされたバージョンになるようにすることによって、非常に均一な2次元ルックアップテーブルを取得できる。また、このようなアプローチを使用して、2次元ルックアップテーブルの要素を簡単に取得できることに留意されたい。

#### 【0022】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブルの要素の第2のグループは、ベースルックアップテーブルの要素の右シフトバージョンである。

このようなアプローチを使用することにより、右シフト操作を非常に簡単に実行できるため、2次元ルックアップテーブルの要素を特に効率的な方法で取得できる。また、右シフト操作は適切なスケーリングを引き起こし、非常に効率的な方法で丸め操作を実行することもできる。

#### 【0023】

算術エンコーダの好ましい実施形態では、確率指数 (Q p<sub>2</sub> (p L P S) ; i) は、2次元ルックアップテーブルの要素の第1のグループの要素または2次元ルックアップテーブルの要素の第2のグループの要素のどちらを評価するかを決定し、確率指数 (例えば、確率値、例えば p L P S を定量化することによって取得される) の第1の範囲 (例えば、0 ~  $\mu$  - 1) は、要素の第1のグループの要素に関連付けられ、確率指数 (例えば、確率値、例えば p L P S を、例えば、量子化関数 Q p<sub>2</sub> (.) を使用して定量化することによって取得される) の第2の範囲 (例えば、 $\mu$  以上) は、要素の第2のグループの要素に関連付けられている。そのような概念を使用することにより、要素の第1のグループの要素を使用すべきか、要素の第2のグループの要素を使用すべきかの違いが、確率指数に応じてできるようになり得、例えば、それぞれの状態変数値に基づき得る。例えば、確率指数 (例えば、最も可能性の低いシンボルの確率によって、または整数の指數値によって定義され得る) は、例えば、マッピングを使用して、それぞれの状態変数値に基づいて導出され得る。例えば、第1の状態変数値のマッピングを使用して得られた第1の確率値、または第2の状態変数値のマッピングを使用して得られた第2の確率値を使用して、2次元ルックアップテーブルのどの要素を評価すべきかを決定することができる (特に、2次元ルックアップテーブルの要素の第1のグループの要素を評価するか、2次元ルックアップテーブルの要素の第2のグループの要素を評価するかを決定する)。

また、この概念を使用すると、ベースルックアップテーブルに基づいて「オンザフライ」で2次元ルックアップテーブルのそれぞれの要素を決定することができ、その要素の数は、確率指数とコーディング間隔サイズ指數によって対処されるテーブル要素の数よりも少なくなる。

#### 【0024】

算術エンコーダの好ましい実施形態では、確率指数 (i) と第1のサイズ値 (例えば、 $\mu$ 、サイズ値は例えば、第1の方向のベースルックアップテーブルの拡張を説明する) お

10

20

30

40

50

および間隔サイズ指標（例えば量子化演算  $Q \ r \ 2 \ ( \ . \ )$ 、例えば  $j$  を使用するなどして間隔サイズ情報  $R$  に基づいて取得できる）との間の除算の除算残差（ $i \% \mu$ ）は、2次元ルックアップテーブルの要素を取得するためにベースルックアップテーブルのどの要素を使用するかを決定する。

このような概念を使用することにより、ベースルックアップテーブルの第1の方向への拡張が可能な確率指標値の数よりも小さい場合でも、ベースルックアップテーブルの適切な要素を選択できる。確率指標と第1のサイズ値の間の除算の分割残差を評価することにより、ベースルックアップテーブルの第1の方向への拡張を表すことができ、ベースルックアップテーブルの要素を2つ以上の異なる確率指標値に再利用できる（例えば、第1のサイズ値の分異なる場合がある）。その結果、ベースルックアップテーブルのエントリは、第1のサイズ値の分が異なる2つの確率指標値に対して、例えば異なるスケーリングを使用して2回使用できる。したがって、2次元ルックアップテーブルは通常、確率指標値の進化を記述し、確率指標値の第2の範囲の進化は、確率指標値の第1の範囲の進化と比較した場合に、スケーリングされたバージョン（例えば、丸め効果を受け入れる）であることが利用できる。  
10

#### 【0025】

好ましい実施形態では、算術エンコーダは、

`RangeTabLPS[i][j] = Scal(BaseTabLPS[i % μ][j], [i / μ]);`

`BaseTabLPS` は、次元  $μ \times$  のベースルックアップテーブルであり、 $i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は、間隔サイズ情報に関連付けられたテーブルインデックスであり（例えば、現在のコーディング間隔サイズを記述する）、% は除算残差演算であり、/ は除算演算であり、`Scal(x, y)` はスケーリング関数であり（例えば、 $Scal(x, y) = [x * a^{-b} y]$  として定義され、[.] はフロア演算であり、 $a$  は、好ましくは2以上の定数であり、または  $b$  は、好ましくは1以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右側にシフトするかどうか、および何ビットシフトするかを決定する、

ことに従って2次元ルックアップテーブル（`RangeTabLPS`）の要素を取得するように構成される。

そのようなアプローチを使用することによって、スケーリング関数またはスケーリング操作の結果によって定義され得る間隔サイズ情報は、特にメモリ効率の良い方法で取得され得る。例えば、「`BaseTabLPS`」ルックアップテーブルは、その第1の次元  $μ$  が、確率情報に関連付けられたテーブルインデックス  $i$  の値の範囲よりも通常小さいため、またその第2の次元  $\cdot$  が間隔サイズ情報によって記述される可能な異なる間隔サイズに等しい可能性があるため、特に小さい場合がある。さらに、 $i$  と  $μ$  の商のフロア演算によって定義される異なるスケーリング係数の数が比較的小さいため、スケーリング関数は特に効率的な方法で実装することができる。例えば、 $i$  と  $μ$  の間の除算の範囲が0と2より小さい最大値との間である場合、2つの異なるスケーリング操作のみを実行できる。例えば、2つ、3つ、または4つの異なるスケーリングの任意の選択のみが存在する可能性があり（ $i$  と  $μ$  の間に応じて）、これらのスケーリングの任意の選択は、少数の所定の値のみを使用した乗算を使用して、または単なるシフト演算さえも使用して、効率的に実装できる。  
30  
40

#### 【0026】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブル（`RangeTabLPS`）の要素は、確率テーブル（`probTabLPS`）に基づいて定義され、確率テーブルは、複数の確率値のセット（例えば、指数  $i$  によって表される）および所与のコーディング間隔サイズの間隔サイズを記述し、複数の確率値のセットにない確率値および/または所与の（参照）コーディング間隔サイズとは異なるコーディング間隔サイズの2次元ルックアップテーブルの要素は、スケーリングを使用して確率テーブルから導出される。  
50

そのようなアプローチを使用して、異なる間隔サイズが、関連する確率値間の差異、および／または関連するコーディング間隔サイズ間の差異に依存するスケーリングによって互いに関連していることが多いことを利用することができる。別の言い方をすれば、2次元ルックアップテーブルが現在考慮されている確率値および／または現在考慮されているコーディング間隔サイズに適合する要素を含まない場合でも、適切な間隔サイズを依然として取得でき、2次元ルックアップテーブルの別の要素は相応してスケーリングされる（例えば、現在考慮されている確率値に依存して、および／または現在考慮されているコーディング間隔サイズに依存して）。

#### 【0027】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブルの要素が、コーディング間隔サイズ( $R$ )に応じて、確率テーブルの選択された要素( $probTabLPS[i \% \mu]$ )の（乗法の）第1のスケーリングを使用し、現在の確率値（指数 $i$ により指定）に関連付けられた要素が確率値のセットに含まれるかどうかに応じて（例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて）、第1のスケーリングの結果の第2のスケーリングを使用して取得される。

したがって、2次元ルックアップテーブルの要素は、確率テーブルの適切なエントリを使用して、また適切なスケーリングを使用して「オンザフライ」で取得でき、評価される「確率テーブル」は、通常、2次元のルックアップテーブルよりも大幅に小さくなる。言い換えると、2次元ルックアップテーブルの要素をアドレス指定する2つのインデックスに基づいて、確率テーブルの適切な要素が選択され、スケーリングされる。ただし、多くの場合、このような概念により、メモリの要求と計算の複雑さの間のトレードオフが改善される。

#### 【0028】

算術エンコーダの好ましい実施形態では、確率指数（例えば $i$ 、例えば、現在の確率値を表す）、および第1のサイズ値（例えば $\mu$ 、例えばサイズ値が確率テーブルの拡張について説明する）の間の除算の除算残差( $i \% \mu$ )は、確率テーブルのどの要素が第1のスケーリングでスケーリングされるかを決定する、および／または確率指数( $i$ )と第1のサイズ値との間の除算の整数除算結果( $[i / \mu]$ )は、第2のスケーリングで使用されるスケーリング係数( $2 - [i / \mu]$ )を決定するおよび／またはコーディング間隔サイズは、第1のスケーリングの乗法スケーリング係数( $Qr2(R)$ )を決定する。

このような概念を使用すると、確率指数とコーディング間隔サイズ指数の2次元グリッドの（比較的小さい）部分の間隔サイズのみを表す小さな確率テーブルを使用でき、メモリースペースを省ける。次に、適切な間隔サイズ情報は、確率テーブルの要素の上記の選択によって、および確率テーブルの選択された要素の上記の2倍のスケーリングによっても取得することができる。

#### 【0029】

好ましい実施形態では、算術エンコーダは、

$$RangeTabLPS[i][j] = Scal([probTabLPS[i \% \mu] \cdot Qr2(R)], [i / \mu])$$

$i$ は、確率情報に関連付けられたテーブルインデックスであり、 $j$ は間隔サイズ情報に関連付けられたテーブルインデックスであり、%は除算残差演算であり、／は除算演算であり、 $probTabLPS$  [ ]は確率テーブルであり、 $\mu$ は確率テーブルの要素の数であり（ $I$ の値の範囲は通常 $\mu$ よりも大きくなる）、 $R$ は間隔サイズ（または現在のコーディング間隔サイズ）であり、 $Qr2(R)$ は $R$ に依存するスケーリング係数であり、 $Scal(x, y)$ はスケーリング関数であり（例えば、 $Scal(x, y) = [x * a^{-b} y]$ として定義され、[.]はフロア演算であり、 $a$ は、好ましくは2以上の定数であり、 $b$ は、好ましくは1以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$ は、 $x$ の右シフトをするかどうか、および何ビットシフトするかを決定する）、

ことに従って2次元ルックアップテーブルの要素

10

20

30

40

50

*RangeTabLPS[i][j]*

を取得するように構成される。

間隔サイズ情報を表している可能性があるか、間隔サイズ情報に等しい可能性がある2次元ルックアップテーブルの適切な要素を取得するためにこのような概念を使用することにより、メモリの要件と計算の複雑さの間の非常に良いトレードオフを得ることができる。テーブル「probTabLPS」は、例えば、1次元テーブルであり得、該テーブルの要素の数は、テーブルインデックス  $i$  のいくつかの異なる可能な値よりも小さくてもよい。ただし、確率テーブル  $probTabLPS$  の選択された要素と、間隔サイズ  $R$  に依存するスケーリング係数  $Qr_2(R)$  の積をスケーリングすることにより、良好な精度に到達でき、発生する可能性のある丸め誤差を適度に小さく保つことができる。さらに、スケーリングは「フロア」演算子によって取得される整数値について演算するため、効率的なスケーリングの概念を使用でき、これは、例えば、整数の乗算、整数の除算、またはビットシフト演算によって定義できる。したがって、計算の負荷は非常に小さい。

### 【0030】

好ましい実施形態では、算術エンコーダは、

$RangeTabLPS[i][j] = Scal([probTabLPS[i \% \mu] \cdot Qr_2(R)], [i / \mu])$

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は間隔サイズ情報に関連付けられたテーブルインデックスであり、 $\%$  は除算残差演算であり、 $/$  は除算演算であり、 $probTabLPS$  [ ] は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり ( $i$  の値の範囲は通常  $\mu$  よりも大きくなる)、 $R$  は間隔サイズ (または現在のコーディング間隔サイズ) であり、 $Qr_2(R)$  は  $R$  に依存するスケーリング係数であり、 $Scal(x, y)$  はスケーリング関数であり (例えば、 $Scal(x, y) = [x * a^{-b} y]$  として定義され、[.] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って 2 次元ルックアップテーブルの要素

$RangeTabLPS[i][j]$

10

20

を取得するように構成される。

また、間隔サイズ情報を取得するためのこのアプローチは特に効率的であり、間隔サイズ情報 (第 1 のスケーリングと第 2 のスケーリングの結果として取得される) に対して質の高い結果をもたらすことがわかっている。

### 【0031】

算術エンコーダの好ましい実施形態では、確率指数 (例えば  $i$ 、例えば、現在の確率値を表す) および第 1 のサイズ値 (例えば  $\mu$ 、例えばサイズ値が確率テーブルの拡張について説明する) の間の除算の除算残差 ( $i \% \mu$ ) は、確率テーブルのどの要素が第 1 のスケーリングでスケーリングされるかを決定する、および / または確率指数 ( $i$ ) と第 1 のサイズ値との間の除算の整数除算結果 ( $[i / \mu]$ ) は、第 1 のスケーリングで使用されるスケーリング係数 ( $2^{-[i / \mu]}$ ) を決定する、および / またはコーディング間隔サイズ ( $R$ ) は、第 2 のスケーリングの乗法スケーリング係数 ( $Qr_2(R)$ ) を決定する。

上記の分割残差に依存して確率テーブルの要素 (1 次元の確率テーブルである可能性がある) を選択することにより、間隔サイズが、スケーリングを除いて、確率指数の異なる範囲で実質的に類似するという事実が利用され得る。したがって、確率テーブルは、確率指数の単一の範囲の値のみを反映し、確率指数の他の範囲の間隔サイズ値は、第 1 のスケーリングを使用して取得される。第 2 のスケーリングは、確率テーブルによって表される値、または第 1 のスケーリングを使用してそれに基づいて得られる値を、コーディング間隔サイズに適合させ、それによって適切な間隔サイズ情報を取得するようとする。

40

50

その結果、メモリ消費、計算の複雑さ、および精度の間の良好なトレードオフが得られ、例えば、第1のサイズ値が適切に選択された場合、除算残差と整数除算の結果は、計算上非常に簡単な方法で取得できる（例えば、2つの効力になる）。

#### 【0032】

好ましい実施形態では、算術エンコーダは、

$RangeTabLPS[i][j] = [Scal(probTabLPS[i \% \mu], [i/\mu]) \cdot Qr_2(R)]$

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は間隔サイズ情報に関連付けられたテーブルインデックスであり、 $\%$  は除算残差演算であり、 $/$  は除算演算であり（例えば、整数の結果を提供する）、probTabLPS [ ] は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり（ $I$  の値の範囲は通常  $\mu$  よりも大きくなる）、 $R$  は間隔サイズであり、 $Qr_2(R)$  は  $R$  に依存するスケーリング係数であり、 $Scal(x, y)$  はスケーリング関数であり（例えば、 $Scal(x, y) = [x * a - b y]$  として定義され、[.] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って 2 次元ルックアップテーブルの要素

$RangeTabLPS[i][j]$

10

を取得するように構成される。

この計算ルールは、前述の概念を非常に効率的な方法で実装する。

#### 【0033】

算術エンコーダの好ましい実施形態では、2 次元ルックアップテーブルは、そのエントリが第1および第2の状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$  ;  $[[s_k] \cdot a_k]$ )

) を含む第1の1次元ベクトル（1次元ルックアップテーブルを形成する）、およびそのエントリがコーディング間隔サイズ情報の量子化レベルを含む第2の1次元ベクトル ( $Qr_2(R)$ )

30

) の間の2項積として、表すことができる。

このような2次元ルックアップテーブルは、状態変数値に基づいて、およびコーディング間隔サイズ情報に基づいて、間隔サイズ情報を効率的に導出するために十分に使用可能である。2次元ルックアップテーブル内のルックアップは、状態変数値の確率値へのマッピング、および取得された確率値とコーディング間隔サイズの乗算に対応する。したがって、2次元ルックアップテーブルの選択されたエントリに等しい可能性がある間隔サイズ情報を取得することは非常に容易に可能であり、2次元ルックアップテーブルのそれぞれの要素は、それぞれの状態変数値に応じて、またコーディング間隔サイズ情報の独立性で、選択され得る（それぞれの状態変数値が2次元ルックアップテーブルの要素の第1のインデックスを決定し得、コーディング間隔サイズ情報が第2のインデックスを決定し得る）。

#### 【0034】

好ましい実施形態では、算術エンコーダは、第1および第2の状態変数値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ ) からそれぞれ第1および第2のサブインターバル幅値 ( $R * p^k$ ) を計算するよう構成され、この計算は、そのエントリが第1および第2の状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k_2 * a^k_2]$ ) を含む1次元ルックアップテーブル (LUT4) を使用して、第1および第2の状態変数値 ( $s_k$ )、ま

40

50

たはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ ) を第 1 および第 2 の確率値にマッピングすること、およびシンボルを量子化レベルに符号化する前に、算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報 (例えば、 $R$ ) を量子化することにより、これを行い、算術エンコーダは、さらに、第 1 および第 2 の確率値と、一方で量子化レベルの間の積を決定する (事前計算された積のルックアップまたは乗算によって) し、第 1 のサブインターバル幅値と第 2 のサブインターバル幅値を使用して (例えば、加重和または加重平均を使用して) 結合されたサブインターバル幅値を取得するように構成される。

このようなアプローチを使用することはまた、特定の状況下で非常に効率的であることがわかっている。第 1 の状態値のみに基づいて (ただし、第 2 の状態変数値を考慮せずに) 第 1 のサブインターバル幅の値を導出し、第 2 の状態変数値のみに基づいて (ただし、第 1 の状態変数値を考慮せずに) 第 2 のサブインターバル幅の値を計算することにより、異なる適合時定数を使用して取得される状態変数値の実質的に別個の処理は、処理の大半を通じて維持される。最終段階でのみ、第 1 のサブインターバル幅の値と第 2 のサブインターバル幅の値を結合して、結合したサブインターバル幅の値を取得し、これは高い精度をもたらし、状況によっては、第 1 の状態変数値と第 2 の状態変数値が、あまりにも早い段階で組み合わされる場合に発生する劣化を回避する。

#### 【 0 0 3 5 】

好ましい実施形態では、算術エンコーダは、コーディング間隔サイズ情報に論理右シフトを適用することによってコーディング間隔サイズ情報の量子化を実行するように構成される。

論理右シフトは最小限の計算リソースしか必要としないため、この概念は特に実装が簡単である。

好ましい実施形態では、算術エンコーダは、コーディング間隔サイズ情報

$R$

の量子化を

$$Q_{T_2}(R) = ([R \cdot 2^{-u}] + v) \cdot 2^{-w}$$

により実行するように構成され、式中、

$u$

$v$

および

$w$

はパラメータである。

このような量子化も非常に簡単に実装できることがわかった。特に、パラメータ

$u, v$

$w$

が整数の値 (または 1 より大きい整数の値) になるように選択されている場合、計算の労力は非常に低くなる。

#### 【 0 0 3 6 】

算術エンコーダの好ましい実施形態では、1 次元ルックアップテーブルのエントリは、第 1 および第 2 の状態変数値、またはそのスケーリングおよび / または丸められたバージ

$w$

10

20

30

40

50

ヨン ( $[s^k * a^k]$ ) の増加で単調に減少する。

1次元ルックアップテーブルの単調に減少するエントリを使用すると、間隔サイズ情報に良好な結果がもたらされることが示されている。

#### 【0037】

算術エンコーダの好ましい実施形態では、第1および第2の状態変数値の値ドメインの異なる値間隔、またはそのスケーリングおよび/または丸められたバージョン ( $[s^k * a^k]$ ) は、等しいサイズになる。

値の間隔のこのような等しいサイズの使用により、単純な量子化が可能になる。さらに、値の間隔を同じサイズにすることで、ルックアップテーブルの要素を適度な労力でオンザフライで決定できる。

10

#### 【0038】

算術エンコーダの好ましい実施形態では、第1および第2の状態変数値の値ドメインの異なる値間隔、またはそのスケーリングおよび/または丸められたバージョン ( $[s^k * a^k]$ ) は、等しいサイズになる。

1次元ルックアップテーブルのエントリのこのような単調な減少を使用すると、指数関数的減衰を高い精度で表すことができるようになる。

#### 【0039】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための算術エンコーダを作成し、算術エンコーダは、間隔サイズ情報 ( $p^k$ 、 $R * p^k$ ) を、1つまたは複数のシンボル値（例えば、バイナリ値）の算術符号化が複数の状態変数値 ( $s_i^k$ )（例えば、インデックス  $k$  によって表示される所与のコンテキストモードに関連付けられるもの）に基づいて符号化されるように導出するように構成され、これは、異なる適合時定数で以前に符号化された複数のシンボル値（例えば、バイナリ値0と1のシーケンス）の統計を表し、算術エンコーダは、結合状態変数値 ( $s_k$ )（例えば、状態変数値の加重和であり得る）、複数の個々の状態変数値 ( $s_i^k$ ) の基礎となるものを導出するように構成され、算術エンコーダは、結合状態変数値 ( $s_k$ )、またはそのスケーリングおよび/または丸められたバージョン ( $[s^k * a^k]$ ) を、ルックアップテーブルを使用してマッピングして、符号化される1つまたは複数のシンボルの算術符号化の間隔サイズを説明する間隔サイズ情報（例えば、 $p_k$  または  $R * p_k$ ）を取得するように構成される。

20

#### 【0040】

本発明によるこの実施形態は、ルックアップテーブルを使用してマッピングが実行される前に結合状態変数値が決定される場合、間隔サイズ情報の決定において高い効率が得られるという考えに基づく。したがって、2つ（またはそれ以上）の状態変数値を使用して個別のルックアップテーブルルックアップを実行する必要がもはやなくなった。むしろ、間隔サイズ情報を決定するには、単一のテーブルルックアップで十分な場合がある。特に、ルックアップテーブルルックアップを実行する前の第1の状態変数値と第2の状態変数値の結合は、多くの状況下で間隔サイズ情報の質を著しく低下させないことがわかった。

30

#### 【0041】

好ましい実施形態では、算術エンコーダは、結合状態変数値を取得するために、状態変数値の加重和を決定するように構成される。

40

状態変数値の加重和の計算は、結合状態変数値を決定するための効率的な方法であり、また2つの状態変数値の異なる関連性を考慮するのにも適していることがわかつており、それは状態変数値を導出するときに使用される様々な適合時定数により引き起こされる。

#### 【0042】

好ましい実施形態では、算術エンコーダは、状態変数値 (

$s_i^k$

) および関連する重み付け値 (

$d_i^k$ )。

50

結合状態変数値 ( $s_k$ ) を得るため) の積を丸めることによって得られる丸められた値 ( $|s_i^k \cdot d_i^k|$ )

) の合計を決定するように構成される。

合計を実行する前にスケーリングされた値の丸めを実行すると、特に意味のある結果が得られることがわかっている。状態変数値の1つの無視できる寄与は、丸めによって排除され、結合状態変数値に影響を与えない。したがって、信頼性の高い結果を得ることができ、結合状態変数値は通常整数の値をとり、これは、ルックアップテーブルの要素を選択するためのインデックスとして機能するのに非常に適している。

#### 【0043】

好ましい実施形態では、算術エンコーダは、

$$s_k = \sum_{i=1}^N |s_i^k \cdot d_i^k|$$

$s^k_2$  は状態変数値であり、Nは考慮される状態変数値の数であり、

[.]はフロア演算子であり、 $d^k_i$ は状態変数値に関連付けられた重み付け値(例えば、結合状態変数値への個々の状態変数値の影響を制御する重み付け係数)( $d^k_i$ は好ましくはフォン2の整数値の効力であり、2つの異なる $d^k_i$ 間の比率は好ましくは2の整数値の効力である)[2つの異なる $d^k_i$ 間の比率は、好ましくは8以上]であることに従って、結合状態変数値  $s_k$  を決定するように構成される。

結合状態変数値を導出するためのこの概念は、前に説明したように、非常に意味のある結合状態変数値をもたらす。

#### 【0044】

好ましい実施形態では、算術エンコーダは、符号化されるシンボルが第1の値(例えば「1」)をとる場合に状態変数値を第1の方向に変更する(例えばより正になる)ように構成され、また符号化されるシンボルが第1の値とは異なる第2の値(例えば「0」)をとる場合(例えば、状態変数値が正と負の値をとることができるように)に状態変数値を第2の方向に(例えば、より負になるもの)変更するように構成され、算術エンコーダは、結合状態変数値の絶対値( $s_k^i > 0$ の場合は $s_k$ 、それ以外の場合は $-s_k$ )に応じて(例えば、結合状態変数値の絶対値のスケーリングおよび丸められたバージョンに応じて)評価されるルックアップテーブルのエントリを決定するように構成される。

状態変数値の決定のためのこの概念(例えば、第1の状態変数値の決定および第2の状態変数値のため)は、第1の状態変数値および第2の状態変数値の別個のマッピングが使用される場合と同じ利点をもたらす。

#### 【0045】

好ましい実施形態では、算術エンコーダは、結合状態変数値が第1の符号(例えば、正の符号)をとる場合に、確率値( $p^k$ )をルックアップテーブルによって提供される値(例えば、

$$LUT2[[-s_k \cdot a_k]]$$

)に設定するように構成され、算術エンコーダは、状態変数値が第2の符号(例えば、負の符号)をとる場合、確率値( $p^k$ )を、所定の値(例えば、1)からルックアップテーブルによって提供される値(例えば、

$$LUT2[[-s_k \cdot a_k]]$$

)を減算することによって得られる値に設定するように構成される。

ルックアップテーブルのサイズを小さくできるため、結合状態変数値を確率値にマッピングすることに対するこの概念は効率的である。特に、ルックアップテーブルの同じ要素が、所与の正の結合状態変数値および所与の(正の)結合状態変数値の負のバージョンに関連付けられるため、ルックアップテーブルの要素の数を減らすことができる。言い換えると、所与の概念では、結合状態変数値の絶対値が、確率値の提供のためにルックアップ

10

20

30

40

50

テーブルのどの要素が評価されるかを決定する。

ただし、結合状態変数値の符号は、確率値をルックアップテーブルによって提供される値、またはルックアップテーブルによって提供される値を符号に応じて所定の値から減算することによって得られる値に設定することにより、適切かつ効率的な方法で依然として考慮される。したがって、計算の複雑さが低い程度の結合状態値に基づいて、意味のある確率値を取得することができる。

#### 【0046】

好ましい実施形態では、算術エンコーダは、

$$p_k = \begin{cases} LUT2[[s_k \cdot a_k]], & \text{if } s_k \geq 0 \\ 1 - LUT2[[-s_k \cdot a_k]], & \text{else.} \end{cases}$$

10

LUT2は確率値を含むルックアップテーブルであり、[.]はフロア演算子であり、 $s^k$ は結合変数値であり、 $a^k$ は、結合状態変数値に関連付けられた重み付け値である（例えば、i番目の状態変数値の数の範囲をルックアップテーブルのいくつかのエントリに適合させる重み付け値）ことにより、結合確率値 $p_k$ を決定するように構成される。

結合状態変数値 $s_k$ に基づいて結合確率値を決定するためのこの概念は、計算上非常に効率的な方法で前に概説された概念を実装する。

#### 【0047】

好ましい実施形態では、算術エンコーダは、

$$p_k = \begin{cases} LUT2[[s_k \cdot a_k]], & \text{if } |s_k \cdot a_k| \geq 0 \\ 1 - LUT2[[-|s_k \cdot a_k|]], & \text{else} \end{cases}$$

20

LUT2は確率値を含むルックアップテーブルであり、[.]はフロア演算子であり、 $s^k$ は結合変数値であり、 $a^k$ は、結合状態変数値に関連付けられた重み付け値である（例えば、i番目の状態変数値の数の範囲をルックアップテーブルのいくつかのエントリに適合させる重み付け値）ことにより、結合確率値 $p_k$ を決定するように構成される。

この概念では、スケーリングされた結合状態変数値 $s_k$ の絶対値の計算はない。むしろ、スケーリングされた（「重み付けされた」）結合状態変数値に適用される「フロア」の演算のみがあり、状況によっては、絶対値の形成よりも少ない労力で実装できる。否定は整数値にのみ適用される。整数値は、重み付けされた結合状態変数値の下方の丸め（フロア演算子）によって取得される。ただし、整数値の否定は、通常、小数値または浮動小数点表現の値の否定ほど複雑ではない。したがって、ここで説明する概念は、状況によっては複雑さを軽減するのに役立ち得る。

30

#### 【0048】

好ましい実施形態では、算術エンコーダは、結合状態変数値、またはそのスケーリングおよび/または丸められたバージョン $([s^k * a^k])$ を、2次元ルックアップテーブルを使用してサブインターバル幅値 $(R * p^k)$ にマッピングするように構成され、そのエントリは、結合状態変数値に依存して（例えば、第1のルックアップテーブルエントリの座標を決定するために）、およびシンボルの符号化前の算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、R）に依存して（例えば、第2のルックアップテーブルエントリの座標を決定するために）アドレス指定される。

この概念を使用して、結合状態変数値の結合確率値へのマッピング、および結合確率値とコーディング間隔サイズとの乗算を、単一のルックアップテーブルルックアップ演算に結合することができる。したがって、2次元のルックアップテーブルが必要だが、乗算の演算は省ける。2次元ルックアップテーブルのエントリは事前に計算されている場合があり、これにより、実行時の結合の負荷が非常に低く抑えられる。むしろ、第1のテーブルインデックスは、結合状態変数値、またはそのスケーリングおよび/または丸められたバージョンに基づいて決定され得、第2のテーブルインデックスは、コーディング間隔サイズ情報に基づいて決定され得る（例えば、丸めまたは量子化を使用）。第1のステップ指数および第2の状態指数は、2次元ルックアップテーブルの要素を一意に指定することができ、2次元ルックアップテーブルの指定された要素は、サブ間隔幅値として（または間隔サイズ情報として）使用することができる。したがって、非常に効率的な概念が得られ

40

50

、ルックアップテーブルに十分なメモリが利用可能な場合の計算の複雑さが軽減される。

#### 【0049】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブルは、そのエントリが結合状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび／または丸められたバージョン ( $[s^k * a^k]$  ;

$|s_k| \cdot a_k$

) を含む第1の1次元ベクトル (

$LUT4[...]$

10

、1次元ルックアップテーブルを形成する)、およびそのエントリがコーディング間隔サイズ情報の量子化レベルを含む第2の1次元ベクトル (

$Q_{T_2}(R)$

) の間の2項積として、表すことができる。

このような2次元ルックアップテーブルは、非常に良い結果をもたらす。特に、この2次元ルックアップテーブルのすべての要素は、結合状態変数値に関連付けられたそれぞれの確率値とコーディング間隔サイズの乗算を表す。したがって、本明細書で説明する2次元ルックアップテーブルは、乗算の必要性を排除し、これは、非常にリソース効率が高いと見なすことができる。

20

#### 【0050】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブル (R a n g T a b L P S) の要素が、ベースルックアップテーブル (Base Tab L P S) に基づいて定義され、2次元ルックアップテーブルの要素の第1のグループ (またはブロック、例えば、「上半分」) は、ベースルックアップテーブルの要素と同一であるか、ベースルックアップテーブルの要素の丸められたバージョンであり、2次元ルックアップテーブルの要素の第2のグループ (またはブロック、例えば「下半分」) は、ベースルックアップテーブルの要素のスケーリングおよび丸められたバージョンである。

ベースルックアップテーブルに基づいて2次元ルックアップテーブルの要素を定義することにより、2次元ルックアップテーブルを非常に簡単な方法で生成できる。さらに、2次元ルックアップテーブルの要素の第2のグループは、ベースルックアップテーブルの要素のスケーリングおよび丸められたバージョンであるため (2次元ルックアップテーブルの要素の第1のグループの要素は、ベースルックアップテーブルの要素またはベースルックアップテーブルの要素の丸められたバージョンと同一)、2次元ルックアップテーブルの行または列の要素の指数関数的な展開がよく反映される。2次元ルックアップテーブルの要素の第2のブロックが実質的に (丸め効果を除く) 2次元ルックアップテーブルの要素の第1のブロックのスケーリングされたバージョンであるという概念を使用することにより、結合状態変数値を間隔サイズ情報にマッピングすることに対する適切な特性を反映することができる。

30

#### 【0051】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブルの要素の第2のグループはベースルックアップテーブルの要素の右シフトバージョンである。

これにより、2次元ルックアップテーブルのエントリ (要素) を簡単に生成できる。

ベースルックアップテーブルの右シフト要素は、スケーリングと丸め操作を結合した非常に効率的な概念であることがわかっている。

40

#### 【0052】

算術エンコーダの好ましい実施形態では、確率指数 (Q p<sub>2</sub> (p L P S) ; i) は、2次元ルックアップテーブルの要素の第1のグループの要素または2次元ルックアップテーブルの要素の第2のグループの要素のどちらを評価するかを決定し、確率指数 (例えば、確率値、例えば p L P S を定量化することによって取得される) の第1の範囲 (例えば、0 ~

50

$\mu - 1$  ) は、要素の第 1 のグループの要素に関連付けられ、確率指數 ( 例えは、確率値、例えは  $p_{LPS}$  を、例えは、量子化関数  $Qp2(\cdot)$  を使用して定量化することによって取得される ) の第 2 の範囲 ( 例えは、 $\mu$  以上 ) は、要素の第 2 のグループの要素に関連付けられている。

( 中間量として確率値を使用せずに ) 結合状態変数値から直接導出され得る、または結合状態変数値に基づく結合確率値を使用して導出され得る確率指數を使用することによって、2 次元ルックアップテーブルの要素の選択は非常に効率的に行うことができる。また、確率指數は、要素の第 1 のグループの要素と要素の第 2 のグループの要素の使用を切り替えるために使用され、これにより、2 次元ルックアップテーブルの要素をオンザフライで効率的に決定できる。

10

#### 【 0 0 5 3 】

算術エンコーダの好ましい実施形態では、確率指數 (  $i$  ) と第 1 のサイズ値 ( 例えは、 $\mu$  、サイズ値は例えは、第 1 の方向のベースルックアップテーブルの拡張を説明する ) および間隔サイズ指數 ( 例えは量子化演算  $Qr2(\cdot)$  、例えは  $j$  を使用するなどして間隔サイズ情報  $R$  に基づいて取得できる ) との間の除算の除算残差 (  $i \% \mu$  ) は、2 次元ルックアップテーブルの要素を取得するためにベースルックアップテーブルのどの要素を使用するかを決定する。

#### 【 0 0 5 4 】

好ましい実施形態では、算術エンコーダは、

$\text{RangeTabLPS}[i][j] = \text{Scal}(\text{BaseTabLPS}[i \% \mu][j], [i / \mu])$

20

$\text{BaseTabLPS}$  は、次元  $\mu \times$  のベースルックアップテーブルであり、 $i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は、間隔サイズ情報に関連付けられたテーブルインデックスであり ( 例えは、現在のコーディング間隔サイズを記述する ) 、 $\%$  は除算残差演算であり、 $/$  は除算演算であり、 $\text{Scal}(x, y)$  はスケーリング関数であり ( 例えは、 $\text{Scal}(x, y) = [x * a - b y]$  として定義され、 $[.]$  はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、または  $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右側にシフトするかどうか、および何ビットシフトするかを決定する、

ことに従って 2 次元ルックアップテーブル (  $\text{RangeTabLPS}$  ) の要素を取得するように構成される。

30

該除算残差を評価して、ベースルックアップテーブルのどの要素を使用して 2 次元ルックアップテーブルの要素を取得するかを決定することにより、2 次元ルックアップテーブルの要素のオンザフライの決定を、非常に高い効率で実行することができる。特に、2 次元ルックアップテーブルが、ベースルックアップテーブルの同じ要素に基づく 2 つ以上の要素グループで構成されているという事実は、ベースルックアップテーブルのどの要素を 2 次元ルックアップテーブルの要素を取得するために使用するかを決定するための分割残差の使用によって反映され得る。言い換えれば、2 次元ルックアップテーブルの要素とベースルックアップテーブルの要素との間の周期的関係は、除算残差がまた確率指數の増加に関して周期的であるため、該除算残差の考慮によってよく反映される。

40

#### 【 0 0 5 5 】

算術エンコーダの好ましい実施形態では、2 次元ルックアップテーブル (  $\text{RangeTabLPS}$  ) の要素は、確率テーブル (  $\text{probTabLPS}$  ) に基づいて定義され、確率テーブルは、複数の確率値のセット ( 例えは、指数  $i$  によって表される ) および所与のコーディング間隔サイズの間隔サイズを記述し、複数の確率値のセットにない確率値および / または所与の ( 参照 ) コーディング間隔サイズとは異なるコーディング間隔サイズの 2 次元ルックアップテーブルの要素は、スケーリングを使用して確率テーブルから導出される。

この概念は、状態変数値の個別のマッピングの場合について説明した対応する概念と同じ考慮事項に基づいている。

50

## 【0056】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブル (R a n g T a b L P S) の要素は、確率テーブル (p r o b T a b L P S) に基づいて定義され、確率テーブルは、複数の確率値のセット（例えば、指数  $i$  によって表される）および所与のコーディング間隔サイズの間隔サイズを記述し、複数の確率値のセットにない確率値および／または所与の（参照）コーディング間隔サイズとは異なるコーディング間隔サイズの2次元ルックアップテーブルの要素は、スケーリングを使用して確率テーブルから導出される。

2次元ルックアップテーブルの要素のそのようなオンザフライの決定は、特に高い資源効率を含むことがわかっている。さらにまた、第1の状態変数値および第2の状態変数値の別個のマッピングの文脈内で使用される対応するアルゴリズムに関してなされた上記のコメントが適用されることに留意されたい。2次元ルックアップテーブルの要素の決定の基礎として、典型的には2次元ルックアップテーブルよりも小さい（例えば、より少ない要素を含む）確率テーブルを使用することによって、非常に高い効率を得ることができる。例えば、確率テーブルは、重要な範囲にわたる様々な確率値と様々な（量子化された）コーディング間隔サイズのマッピングを表す場合があり、したがって、単純なスケーリングを超える乗算を回避するのに役立つ場合がある（「単純な」スケーリングでは、例えば、シフト演算を使用して実装する場合がある）。

複数の確率値のセットに含まれない1つまたは複数の確率値、および所与のコーディング間隔サイズとは異なる1つまたは複数のコーディング間隔サイズの2次元ルックアップテーブルの要素は、確率テーブルから、スケーリングを使用して導出される。したがって、例えば、1つの行または1つの列のみを含み得る非常に小さい確率テーブルを有することで十分である。したがって、確率テーブルの要素の数は、異なる可能性のある確率値の数よりも少ない場合さえもあり得る（つまり、2次元ルックアップテーブルの異なる可能な確率指数の数よりも少ない場合がある）。さらに、スケーリングは確率値に依存する、および／またはコーディング間隔サイズに依存することに留意されたい。スケーリングは、例えば、確率テーブルのサイズが適切に選択されている場合、ビットシフト演算などの非常に単純なメカニズムを使用して実行できる。ビットシフト演算に基づくこのような「単純な」スケーリング演算は、任意の可変オペランド（例えば、2の効力とは異なる）を使用した「通常の」乗算と比較すると、必要な計算リソースが大幅に少なくなる。

## 【0057】

算術エンコーダの好ましい実施形態では、2次元ルックアップテーブルの要素が、コーディング間隔サイズ (R) に応じて、確率テーブルの選択された要素 (p r o b T a b L P S [i % μ]) の（乗法の）第1のスケーリングを使用し、現在の確率値（指数  $i$  により指定）に関連付けられた要素が確率値のセットに含まれるかどうかに応じて（例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて）、第1のスケーリングの結果の第2のスケーリングを使用して取得される。

2次元ルックアップテーブルの要素のそのような計算は特に効率的であることがわかっている。さらに、第1の状態変数値および第2の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

## 【0058】

算術エンコーダの好ましい実施形態では、確率指数（例えば  $i$ 、例えば、現在の確率値を表す）、および第1のサイズ値（例えば  $μ$ 、例えばサイズ値が確率テーブルの拡張について説明する）の間の除算の除算残差 ( $i \% μ$ ) は、確率テーブルのどの要素が第1のスケーリングでスケーリングされるかを決定する、および／または確率指数 ( $i$ ) と第1のサイズ値との間の除算の整数除算結果 ( $[i / μ]$ ) は、第2のスケーリングで使用されるスケーリング係数 ( $2^{-[i / μ]}$ ) を決定するおよび／またはコーディング間隔サイズは、第1のスケーリングの乗法スケーリング係数 (Q r 2 (R)) を決定する。

この機能に関しては、第1の状態変数値および第2の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

10

20

30

40

50

## 【0059】

好ましい実施形態では、算術エンコーダは、

$RangeTabLPS[i][j] = Scal([probTabLPS[i \% \mu] \cdot Qr_2(R)], [i/\mu])$

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は間隔サイズ情報に関連付けられたテーブルインデックスであり、 $\%$  は除算残差演算であり、 $/$  は除算演算であり、 $probTabLPS[]$  は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり（ $I$  の値の範囲は通常  $\mu$  よりも大きくなる）、 $R$  は間隔サイズ（または現在のコーディング間隔サイズ）であり、 $Qr_2(R)$  は  $R$  に依存するスケーリング係数であり、 $Scal(x, y)$  はスケーリング関数であり（例えば、 $Scal(x, y) = [x * a^{-b} y]$  として定義され、 $[.]$  はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って

## 2 次元ルックアップテーブルの要素

$RangeTabLPS[i][j]$

を取得するように構成される。

この機能に関しては、第 1 の状態変数値および第 2 の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

## 【0060】

算術エンコーダの好ましい実施形態では、2 次元ルックアップテーブルの要素が、確率テーブルの選択された要素（ $probTabLPS[i \% \mu]$ ）の（乗法の）第 1 のスケーリングを使用し、現在の確率値（指数  $i$  により指定）に関連付けられた要素が確率値のセットに含まれるかどうかに応じて（例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて）、また、コーディング間隔サイズ（ $R$ ）に応じて、第 1 のスケーリングの結果の第 2 のスケーリングを使用して取得される。

この機能に関しては、第 1 の状態変数値および第 2 の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

## 【0061】

算術エンコーダの好ましい実施形態では、確率指数（例えば  $i$ 、例えば、現在の確率値を表す）、および第 1 のサイズ値（例えば  $\mu$ 、例えばサイズ値が確率テーブルの拡張について説明する）の間の除算の除算残差（ $[i \% \mu]$ ）は、確率テーブルのどの要素が第 1 のスケーリングでスケーリングされるかを決定する、および / または確率指数（ $i$ ）と第 1 のサイズ値との間の除算の整数除算結果（ $[i / \mu]$ ）は、第 1 のスケーリングで使用されるスケーリング係数（ $2^{-[i / \mu]}$ ）を決定する、および / またはコーディング間隔サイズ（ $R$ ）は、第 2 のスケーリングの乗法スケーリング係数（ $Qr_2(R)$ ）を決定する。

この機能に関しては、第 1 の状態変数値と第 2 の状態変数値との別個のマッピングの文脈で提供される対応する機能の上記の議論も参照され、結合状態変数値または結合確率値は、それぞれの個々の状態変数値またはそれぞれの個々の確率値の代わりに使用される。

## 【0062】

好ましい実施形態では、算術エンコーダは、

$RangeTabLPS[i][j] = Scal(probTabLPS[i \% \mu], [i/\mu]) \cdot Qr_2(R)$

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は間隔サイズ情報に関連付けられたテーブルインデックスであり、 $\%$  は除算残差演算であり、 $/$  は除算演算であり（例えば、整数の結果を提供する）、 $probTabLPS[]$  は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり（ $I$  の値の範囲は通常  $\mu$  よりも大きくなる）、 $R$  は間隔サイズであり、 $Qr_2(R)$  は  $R$  に依存するスケーリング係数であり、 $Scal$

10

20

30

40

50

$(x, y)$  はスケーリング関数であり（例えば、 $\text{Scal}(x, y) = [x * a - b y]$  として定義され、 $[.]$  はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って、2 次元ルックアップテーブルの要素

$\text{RangeTabLPS}[i][j]$

を取得するように構成される。

この機能に関しては、第 1 の状態変数値と第 2 の状態変数値との別個のマッピングの文脈で提供される対応する機能の上記の議論も参照され、結合状態変数値または結合確率値は、個々の状態変数値の代わりになり、結合確率値が個々の確率値の代わりになる。

#### 【0063】

好ましい実施形態では、算術エンコーダは、結合状態変数値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ ) からサブインターバル幅値 ( $R * p^k$ ) を計算するよう構成され、この計算は、そのエントリが結合状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ );

$[[s_k], a_k]$

10

) を含む 1 次元ルックアップテーブル (LUT4) を使用して、結合状態変数値 ( $s_k$ ) 、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ );

$[[s_k], a_k]$

20

) を結合確率値にマッピングすること、およびシンボルを量子化レベルに符号化する前に、算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、 $R$ ）を量子化することにより、これを行い、算術エンコーダは、さらに、結合確率値と量子化レベルの間の積を計算する（事前に計算された積のルックアップまたは乗算によって）ように構成される。

この機能に関しては、第 1 の状態変数値と第 2 の状態変数値との別個のマッピングの文脈で提供される対応する機能の上記の議論も参照され、ここで結合状態変数値または結合確率値は、個々の状態変数値の代わりになり、ここで結合確率値が個々の確率値の代わりになる。

30

#### 【0064】

好ましい実施形態では、算術エンコーダは、コーディング間隔サイズ情報に論理右シフトを適用することによってコーディング間隔サイズ情報の量子化を実行するように構成される。

この機能に関しては、第 1 の状態変数値および第 2 の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

#### 【0065】

好ましい実施形態では、算術エンコーダは、コーディング間隔サイズ情報

$R$

40

の量子化を

$$Q_{T_2}(R) = ([R \cdot 2^{-u}] + v) \cdot 2^{-w}$$

により実行するように構成され、式中、

$u$

50

*v*

および

*w*

はパラメータである。

この機能に関しては、第1の状態変数値および第2の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

#### 【0066】

算術エンコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、結合状態変数値、またはそのスケーリングおよび／または丸められたバージョン（[ $s^k * a^k$ ]）の増加で単調に減少する。

この機能に関しては、第1の状態変数値および第2の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

#### 【0067】

算術エンコーダの好ましい実施形態では、結合状態変数値の値ドメインの異なる値間隔、またはそのスケーリングおよび／または丸められたバージョン（[ $s^k * a^k$ ]）は、等しいサイズである。

この機能に関しては、第1の状態変数値および第2の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

#### 【0068】

算術エンコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、結合状態変数値、またはそのスケーリングおよび／または丸められたバージョン（[ $s^k * a^k$ ]）の増加で減少率を伴って単調に減少する。

この機能に関しては、第1の状態変数値および第2の状態変数値の別個のマッピングの文脈で提供される、対応する機能の上記の議論も参照される。

#### 【0069】

算術エンコーダの好ましい実施形態では、ルックアップテーブルは、指数関数的減衰（例えば、0.5からの低下）を（例えば、+/-10%または+/-20%の許容範囲内で）定義する。

指数関数的減衰は、状態変数値に基づいて間隔サイズ情報を導出するための適切な関係をよく反映していることがわかっている。また、指数関数的減衰は、ルックアップテーブルを使用して非常に効率的に表すことができ、ルックアップテーブルの要素のオンザフライでの決定でさえ、わずかな労力で可能である。

#### 【0070】

好ましい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 1.} \\ s_i^k - [A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k] & \text{If symbol to be encoded is 0.} \end{cases}$$

*z* は所定の（一定の）オフセット値であり、

 $m_i^k$ 

は1つまたは複数の重み付け値であり、

 $n_i^k$ 

は1つ以上の重み付け値であり、*A* は

$$A[z + \hat{s}] = offset + \sum_{i=1}^{\hat{N}} \max(0, \hat{b}_i - \hat{s}) << \hat{a}_i$$

であるか、ゼロ設定または大きさの縮小によって、引数の1つ以上の極値に対してのみ、それから逸脱して、更新された

10

20

30

40

50

$s_i^k$ 

を所定の値の範囲のままにすることを回避する（例えば、

 $s_i^k$ 

は

 $s_i^k \cdot m_i^k$ 

より大きな値ドメインを持つ；つまり、

 $s_i^k$ 

10

は

 $s_i^k \cdot m_i^k$ 

に準量子化される；

 $s_i^k \cdot m_i^k$ 

の極値の場合、

 $s_i^k$ 

20

は上記の式に従って未修正の  $A$  によって変更され、その値ドメインを離れることができる：これを回避するために、これらの極値に対応するエントリが削減またはゼロにされる場合があることを考慮されたい）、 $offset$  $\hat{N}$  $\hat{b}_i$ 

30

、および

 $\hat{a}_i$ 

は所定のパラメータである（例は上記に記載されている）

ことに従って複数の可変状態値

 $s_i^k$ 

を更新するように構成される。

状態変数のそのような更新は、高い計算効率で行うことができ、マッピングテーブルを事前に計算することができるがわかっている。例えば、状態変数値

 $s_i^k$ 

40

が所定の範囲（例えば、所定の最小値と所定の最大値との間）を維持することを達成することができる。さらに、マッピングテーブル  $A$  の適切な選択を使用して、状態変数値が、以前に処理された（例えば、符号化または復号された）シンボルの統計の記述にうまく適合されることが達成され得る。また、それぞれの状態変数値の適合時定数は、重み付け値  $m$  および  $n$  を適切に選択することによって適合できることに留意されたい。したがって、ここで説明するアルゴリズムは、状態変数値 $s_i^k$ 

50

の更新に効率的に使用できる。

【 0 0 7 1 】

好ましい実施形態では、算術エンコーダは、テーブルルックアップまたは計算によって  
 $A[z + \lfloor s_i^k \cdot m_i^k \rfloor]$

を導出するように構成される。

したがって、更新された状態変数値を導出するためには、様々な概念が可能である。

【 0 0 7 2 】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための算術エンコーダであって、算術エンコーダは、1つまたは複数の状態変数値（ $s_1^k, s_2^k$ ）を決定するように構成され、これは、以前に符号化された複数のシンボル値（例えば、バイナリ値0および1のシーケンス）の統計（例えば、複数の状態変数値の場合、異なる適合時定数の統計）を表し、および

算術エンコーダは、1つまたは複数の状態変数値（ $s_i^k$ ）（例えば、インデックスkにより示される、所与のコンテキストモードに関連付けられている）に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ情報（ $p^k, R * p^k$ ）を導出するように構成され、これは複数の以前に符号化されたシンボル値（例えば、バイナリ値0および1のシーケンス）の統計（例えば、複数の状態変数値の場合、異なる適合時定数の統計）を表し、算術エンコーダは、符号化されるシンボルに依存して、ルックアップテーブル（A）を使用して（例えば、シンボルを符号化した後）、第1の状態変数値（ $s_1^k$ ）を更新するように構成される。

【 0 0 7 3 】

本発明によるこの実施形態は、算術符号化（例えば、符号化または復号）のための間隔サイズ情報の導出に使用される状態変数値の更新が、ルックアップテーブルを使用しながら特に良好な結果を伴って実行され得るという発見に基づいており、その理由はルックアップテーブルを使用することが、符号化または復号される信号の特性に特によく適合した状態変数値の更新を可能にするためである。例えば、ルックアップテーブルを使用して、「古い」状態変数値と更新された状態変数値の間の微調整された関係を簡単に表すことができるが、大規模な計算（例えば、三角関数または指数関数または対数関数などの評価）を実行する必要はない。したがって、ルックアップテーブルを使用した状態変数値の更新の実装は、計算の複雑さを適度に小さく保つのに役立つ。理想的には、「古い」状態変数値に基づいて更新された状態変数値を取得するために、ルックアップテーブルルックアップの他に、乗算（または単純な乗算、例えばビットシフト演算）、丸め演算、および加算のみが使用される。例えば、現在処理されているシンボル（例えば、符号化されるシンボルまたは復号されるシンボル）は、更新された状態変数値を取得するために、例えばルックアップテーブルベースのマッピングルールのどの部分を評価するかを決定する。

結論として、ルックアップテーブルを使用して更新された状態変数値を提供すると、高い柔軟性と低い計算の複雑さの両方が提供されることがわかっている。

【 0 0 7 4 】

好ましい実施形態では、算術エンコーダは、符号化されるシンボルに依存して、ルックアップテーブル（A）を使用して（例えば、符号化されるシンボルを符号化した後）第2の状態変数値（ $s_2^k$ ）を更新するように構成される。

第1の状態変数値を更新するために使用されるのと同じルックアップテーブルを使用して、第2の状態変数値を更新することが有利であることがわかっている。例えば、第1の状態変数値および第2の状態変数値の適合時定数に関して考えられる差異は、例えば、1つまたは複数のスケーリング係数を使用して考慮することができ、これは、例えば、ルックアップテーブルの要素の選択、および/またはルックアップテーブルの選択された要素のスケーリングに適用され得る。結論として、2つ以上の状態変数値の導出に单一のルックアップテーブルのみが使用されている場合でも、2つ以上の状態変数値を適合させて、処理されたシンボル値の（つまり、以前に符号化されたシンボル値または復号されたシン

10

20

30

40

50

ボル値の ) 異なる統計特性を表すことができる。

【 0 0 7 5 】

好ましい実施形態では、算術エンコーダは、異なる適合時定数を使用して、第 1 の状態変数値および第 2 の状態変数値を更新するように構成される。

異なる適合時定数を使用して第 1 の状態変数値と第 2 の状態変数値を更新することにより、以前に処理されたシンボルの異なる統計特性を状態変数値に反映させることができる。異なる適合時定数で処理されたシンボルの統計を表す状態変数値の可用性は、シンボルの算術コーディング（符号化 / 復号）の間隔サイズの正確な調整に非常に役立つことがわかっている。また、状態変数値のルックアップテーブルベースの更新により、非常に高い信頼性と低い計算の複雑さが実現されることがわかっている。

10

【 0 0 7 6 】

好ましい実施形態では、算術エンコーダは、符号化されるシンボルが第 1 の値をとるか、第 1 の値とは異なる第 2 の値をとるかに応じて、ルックアップテーブルを使用して決定された値によって以前の状態変数値を選択的に増減するように構成される。

このようなアプローチを使用することにより、状態変数値を再帰的に適合させることができ、処理されたシンボル（例えば、符号化されるシンボル、または以前に符号化されたシンボル、または以前に復号されたシンボル）が、状態変数値の変化の方向（増加または減少）を決定する。他方、適合のサイズ（すなわち、増加または減少のサイズ）は、選択されたルックアップテーブルエントリによって決定され、スケーリングが適用され得る。その結果、状態変数値を更新するための効率的なメカニズムが存在し、これは、高度な柔軟性を実現し、それでも非常にリソース効率が高くなる。

20

【 0 0 7 7 】

好ましい実施形態では、算術エンコーダは、以前の状態変数値を、符号化されるシンボルが第 1 の値をとる場合に、以前の状態変数値が正である場合と比較して、以前の状態変数値が負である場合に、比較的大きい値だけ増加させるように構成され、算術エンコーダは、以前の状態変数値を、符号化されるシンボルが、第 1 の値とは異なる第 2 の値をとる場合に、以前の状態変数値が負である場合と比較して、以前の状態変数値が正である場合に、比較的大きい値だけ減少させるように構成される（それは、例えば、ルックアップテーブルを適切に選択することで到達する）。

このようなアプローチを使用すると、状態変数値が指数関数的に最大の正の値に向かって発展し、指数関数的に最小値に向かって発展するように到達できる。（正の）最大値および（負の）最小値に向かうこの近似は、ほぼ漸近的になる可能性がある。言い換えると、現在の状態変数値が（正の）最大値から離れるほど、（増加する）ステップが（正の）最大値に向かって大きくなり、現在の状態変数値が（負の）最大値から離れるほど、（負の）最小値に向かう（減少）ステップが大きくなる。したがって、指数漸近的挙動は、この概念を使用して近似できる。ただし、このような概念は、状態変数値の更新に非常に適していることがわかっている。特に、このようなアプローチは、状態変数値を決定するための「無限インパルス応答」アプローチに非常に適していることがわかっている。

30

【 0 0 7 8 】

好ましい実施形態では、算術エンコーダは、符号化されるシンボルが第 1 の値をとる場合、所定（例えば、固定の）のオフセット値（ $z$ ）と以前に計算された第 1 の状態変数値（ $s_1^k$ ）

40

）、またはそのスケーリングおよび / または丸められたバージョン（ $\lfloor s_1^k \cdot m_1^k \rfloor$ ）

）の合計に依存して、第 1 の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するように構成され、算術エンコーダは、符号化されるシンボルが第 2 の値をとる場合、所定（例えば、固定の）のオフセット値（ $z$ ）と以前に計算された第 1 の状態変数値の逆（- 1 を乗算する）バージョン（

50

$-s_1^k$ 

) 、またはそのスケーリングおよび / または丸められたバージョン (  $\lfloor -s_1^k \cdot m_1^k \rfloor$  )

) ( 例えは、以前に計算された第 1 の状態変数値の逆バージョン ) の合計に依存して、第 1 の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するように構成される。

このようなアプローチを使用すると、現在処理されているシンボルを考慮して、適度な労力でルックアップテーブルの適切なエントリを選択できる。以前に計算された状態変数値と処理された ( 符号化または復号された ) シンボルの両方がルックアップテーブルのエントリの選択を決定し、その結果、以前に計算された状態変数値と比較したときに更新された状態変数値がどれだけ増加または減少するかによって決定され得る。オフセット値は、例えは、オフセット値とスケーリングされた ( 処理されたシンボルによっては逆にされる可能性がある ) 以前計算された状態変数値の合計が、有効なルックアップテーブルインデックスになることを保証し得る。これは、有効なルックアップテーブルインデックスが通常負ではないためである。このような概念を使用すると、適切なルックアップテーブルインデックスを選択し、更新された状態変数値を効率的に提供することが簡単に可能になる。

#### 【 0 0 7 9 】

好ましい実施形態では、算術エンコーダは、符号化されるシンボルが第 1 の値をとる場合、所定 ( 例えは、固定の ) のオフセット値 (  $z$  ) と以前に計算された第 2 の状態変数値 (  $s_2^k$  )

) 、またはそのスケーリングおよび / または丸められたバージョン (  $\lfloor s_2^k \cdot m_2^k \rfloor$  )

) の合計に依存して、第 2 の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するように構成され、算術エンコーダは、符号化されるシンボルが第 2 の値をとる場合、所定 ( 例えは、固定の ) のオフセット値 (  $z$  ) と以前に計算された第 2 の状態変数値の逆 ( -1 を乗算する ) バージョン (  $-s_2^k$  )

) 、またはそのスケーリングおよび / または丸められたバージョン (  $\lfloor -s_2^k \cdot m_2^k \rfloor$  )

) ( 例えは、以前に計算された第 2 の状態変数値の逆バージョン ) の合計に依存して、第 2 の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するように構成される。

第 2 の状態変数値の更新に関するこの概念は、第 1 の状態変数値の更新に関する概念と実質的に同一であり、例えは、同じルックアップテーブルを評価してメモリリソースを省くことができ、例えは、第 1 の状態変数値の更新と比較した場合の異なるスケーリング値を使用して、それにより、修正された状態変数値の更新特性を取得することができる。例えは、第 1 の状態変数値および状態変数値についての異なる適合時定数は、更新された第 1 の状態変数値および更新された第 2 の状態変数値の決定のために異なるスケーリング値を使用することによって取得され得る。

#### 【 0 0 8 0 】

好ましい実施形態では、算術エンコーダは、第 1 の状態変数値の更新時に評価されるルックアップテーブルのエントリのインデックスを決定するときに、第 1 のスケーリング値 (  $m^k_1$  ) を適用して、以前に計算された第 1 の状態変数値 (  $s^k_1$  ) をスケーリングする

10

20

30

40

50

ように構成され、算術エンコーダは、第2の状態変数値の更新時に評価されるルックアップテーブルのエントリのインデックスを決定するときに、第2のスケーリング値( $m^k_2$ )を適用して、以前に計算された第2の状態変数値( $s^k_2$ )をスケーリングするように構成され、第1のスケーリング値が第2のスケーリング値とは異なる(また、第1および第2のスケーリング値は好ましくは2の整数効力であり、第1のスケーリング値と第2のスケーリング値との間の比は、2の場合、好ましくは整数効力であり、第1のスケーリング値および第2のスケーリング値は、好ましくは少なくとも8倍異なる)。

第1の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するときと、第2の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するときに異なるスケーリング値を使用することによって、異なる適合時定数が効率的に実装でき、基本的な状態変数値の更新アルゴリズムと、使用されるルックアップテーブルとは同一であり得、唯一の重要な違いはスケーリング値の選択であり得る。これにより、非常にリソースを省く実装が可能になる。

#### 【0081】

好ましい実施形態では、算術エンコーダは、第1の状態変数値を更新するときに第1のスケーリング値(例えば、 $n^k_1$ )を使用してルックアップテーブルの評価によって返される値をスケーリングするように構成され、算術エンコーダは、第2の状態変数値を更新するときに第2のスケーリング値(例えば、 $n^k_2$ )を使用してルックアップテーブルの評価によって返される値をスケーリングするように構成され、第1のスケーリング値は第2のスケーリング値とは異なる。

(例えば、選択されたルックアップテーブルエントリの)ルックアップテーブルの評価によって返される値の説明された異なるスケーリングは、第1の状態変数値および第2の状態変数値を更新するときに、異なる適合時定数の効率的な実装を可能にする。さらに、このようなスケーリングにより、第1の状態変数値の更新と第2の状態変数値の更新に同じルックアップテーブルを使用できるようになるため、メモリリソースの節約に役立つ。

好ましい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{if symbol to be encoded is 1.} \\ s_i^k - [A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{if symbol to be encoded is 0.} \end{cases}$$

$A$ はルックアップテーブルであり(例えば、整数の値を含む)、 $z$ は所定の(一定の)オフセット値であり、

$$m_i^k$$

は1つまたは複数の重み付け値であり、

$$n_i^k$$

は1つまたは複数の重み付け値であることに従って、1つまたは複数の更新された状態変数値

$$s_i^k$$

を決定するように構成される。

状態変数値のこのような更新メカニズムは、高い計算効率で実装でき、信頼できる結果を提供することがわかっている。

#### 【0082】

好ましい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{if symbol to be encoded is 1.} \\ s_i^k - [A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{if symbol to be encoded is 0.} \end{cases}$$

$A$ はルックアップテーブルであり(例えば、整数の値を含む)、 $z$ は所定の(一定の)オフセット値であり、

10

20

30

40

50

$m_i^k$ 

は 1 つまたは複数の重み付け値であり、

 $n_i^k$ 

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

 $s_i^k$ 

を決定するように構成される。

10

状態変数値を更新するためのそのようなメカニズムはまた、状況によっては非常に有利である可能性があることがわかっている。特に、このようなアプローチを使用して、一部の実装では計算効率が悪い可能性がある浮動小数点値を反転する必要はない。したがって、本概念は、状況によっては非常に優れたリソース効率をもたらす可能性がある。

**【 0 0 8 3 】**

好みらしい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 0.} \end{cases}$$

A はルックアップテーブルであり（例えば、整数の値を含む）、z は所定の（一定の）オフセット値であり、

 $m_i^k$ 

は 1 つまたは複数の重み付け値であり、

 $n_i^k$ 

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

 $s_i^k$ 

30

を決定するように構成される。

この概念はまた、一部の実装環境で特に高い計算効率と優れた精度をもたらすことがわかっている。

**【 0 0 8 4 】**

好みらしい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 0.} \end{cases}$$

A はルックアップテーブルであり（例えば、整数の値を含む）、z は所定の（一定の）オフセット値であり、

 $m_i^k$ 

は 1 つまたは複数の重み付け値であり、

 $n_i^k$ 

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

 $s_i^k$ 

40

を決定するように構成される。

50

この概念はまた、状況によっては計算効率と信頼性の面で利点をもたらすことがわかっている。

【 0 0 8 5 】

算術エンコーダの好ましい実施形態では、Aのエントリは、ルックアップテーブルインデックスの増加とともに単調に減少する。

このようなアプローチを使用すると、状態変数値の最大値または最小値への近似が単調および／または連続的および／または漸近的であることに到達する可能性がある。例えば、それぞれの最大値または最小値から遠く離れている状態変数は、それぞれの最大値または最小値に向かって比較的速く修正される一方で、それぞれの最大値または最小値に近い状態変数値は、該最大値または最小値に向かって比較的ゆっくりと修正されることに到達することができる。したがって、ルックアップテーブルのエントリの上記の選択により、最大値または最小値のスムーズな近似が可能になり、それはもう1つの状態変数値に基づいて間隔サイズ情報を導出するのに非常に役立つことがわかっている。

【 0 0 8 6 】

好ましい実施形態では、算術エンコーダは、Aは

$$A[z + \hat{s}] = offset + \sum_{i=1}^{\hat{N}} \max(0, \hat{b}_i - \hat{s}) \ll \hat{a}_i$$

であるか、ゼロ設定または大きさの縮小によって、引数の1つ以上の極値に対してのみ、それから逸脱して、更新された

$$s_i^k$$

10

を所定の値の範囲のままにすることを回避する（例えば、

$$s_i^k$$

20

は

$$s_i^k \cdot m_i^k$$

より大きな値ドメインを持つ；つまり、

$$s_i^k$$

30

は

$$s_i^k \cdot m_i^k$$

に準量子化される；

$$s_i^k \cdot m_i^k$$

の極値の場合、

$$s_i^k$$

は上記の式に従って未修正のAによって変更され、その値ドメインを離れることができる：これを回避するために、これらの極値に対応するエントリが削減またはゼロにされる場合があることを考慮されたい）、

$$offset$$

40

、

$$\hat{N}$$

、

$$\hat{b}_i$$

50

、および  
 $\hat{a}_i$

は所定のパラメータである（例は上記に記載されている）。

ルックアップテーブル のそのような選択は、該ルックアップテーブル A を使用して更新される状態変数値の特に有利な挙動をもたらすことがわかっている。

【 0 0 8 7 】

算術エンコーダの好ましい実施形態では、ルックアップテーブルの最後のエントリ（これは、第 1 の状態変数値が最大許容値まで及ぶ所定の値の範囲に達したとき、または第 1 の状態変数値が所定のしきい値を超えたときにアドレス指定される）はゼロに等しい。

ルックアップテーブルの最後のエントリ（0 に等しい）を使用することにより、更新された状態変数値が最大値および / または最小値を超えることを簡単に回避できる。

好ましい実施形態では、算術エンコーダは、更新された状態変数値にクリッピング操作を適用して、更新されクリップされた状態変数値を所定の値の範囲内に保つように構成される。

このようなメカニズムを使用すると、状態変数値が最小値と最大値の間の所定の範囲を超えることを容易に防ぐことができる。したがって、状態変数値が「合理的な」値をとることを保証することができる。

【 0 0 8 8 】

好ましい実施形態では、算術エンコーダは、

$$s_i^k = \min(\max(s_i^k, h_i^k), g_i^k)$$

によってクリッピング操作を

更新された状態変数値に適用するように構成され、

$$g_i^k$$

は

$$s_i^k$$

の最大許容値であり、

$$h_i^k$$

は

$$s_i^k$$

の最小許容値である。

このようなクリッピング操作は効率的な方法で実装でき、無効な状態変数値を回避できることがわかっている。

【 0 0 8 9 】

好ましい実施形態では、算術エンコーダは、異なるコンテキストモデルに異なるスケーリング値を適用するように構成される（例えば、スケーリング値の少なくとも 1 つが 2 つの異なるコンテキストモデル間で異なるようにする）。

異なるコンテキストモデルに対して異なるスケーリング値を使用して、異なるコンテキストモデルの異なる統計特性（異なるタイプの情報および / またはタイプのビットストリーム構文要素に関連付けられている可能性がある）を考慮することができる。異なるコンテキストモデルに異なるスケーリング値を使用することにより、状態変数値の更新手順を、基礎となるアルゴリズムを根本的に変更することなく、異なるコンテキストモデルに簡単に適合させることができる。したがって、適切なスケーリング値を非常に効率的な方法で取得できる。

【 0 0 9 0 】

10

20

30

40

50

好ましい実施形態では、算術エンコーダは、上記の実施形態の1つで定義された間隔サイズ情報を取得するように構成される。

状態変数値の更新の概念は、間隔サイズ情報の導出のための上記の概念と結合して十分に使用できることがわかっている。

#### 【0091】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための算術エンコーダを作成するものであり、算術エンコーダは、1つまたは複数の状態変数値（ $s_i^k$ ）に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ値（ $R_{LPS}$ ）（それは例えば、インデックス  $k$  で示される所与のコンテキストモードに関連付けられている）を導出するように構成され、これは、以前に符号化された複数のシンボル値（例えば、バイナリ値0および1のシーケンス）の統計を表し（例えば、異なる適合時定数を使用）、算術エンコーダは、ベースルックアップテーブル（Base Table）を使用して間隔サイズ値（ $R_{LPS}$ ）を決定するように構成され（その次元は、確率指數の観点から、可能な確率指數  $i$  の数よりも小さい）、算術エンコーダは、1つまたは複数の状態変数値（例えば、 $i = Qp (p_{LPS})$  のように）を基にして得られる確率指數（ $i$ ）が第1の範囲（例えば、 $\mu$ 未満）内にある場合、決定された間隔サイズ値がベースルックアップテーブルの要素と同一であるか、またはベースルックアップテーブルの要素の丸めのバージョンであり、確率指數が第2の範囲（例えば、 $\mu$ 以上）内にある場合は、ベースルックアップテーブルの要素のスケーリングと丸めを使用して、決定された間隔サイズ値が取得されるように、間隔サイズ値（ $R_{LPS}$ ）が決定されるように構成され、算術エンコーダは、間隔サイズ値（ $R_{LPS}$ ）を使用して1つまたは複数のシンボルの算術符号化を実行するように構成される。

10

20

30

#### 【0092】

本発明によるこの実施形態は、算術エンコーダにおける間隔サイズ値の決定が「ベースルックアップテーブル」を使用して実行できるという考えに基づいており、その次元は、該ベースルックアップテーブルの要素を再利用することによる所与の現在のコーディング間隔サイズに関連するいくつかの異なる間隔サイズ値よりも小さく、1回はスケーリングなしで、もう1回はスケーリングがありである。したがって、異なる範囲の状態変数値に関連付けられている算術符号化（符号化 / 復号）の間隔サイズ値は、スケーリングによって実質的に異なる（例えば、いくつかの丸め効果を除いて）ことが利用可能である。したがって、比較的小さな「ベースルックアップテーブル」を使用することができ、そのエントリは、異なる確率指數に対して複数回使用される（確率指數は、それぞれの状態変数値に基づいて導出され得る）。

結論として、ここで説明する概念により、1つまたは複数の状態変数値に基づいて間隔サイズ値を非常に効率的に決定できる。

#### 【0093】

好ましい実施形態では、算術エンコーダは、間隔サイズ値を決定するように構成され、決定された間隔サイズ値（ $R_{LPS}$ ）は、確率指數が第2の範囲内にある場合、ベースルックアップテーブルの要素の右シフトバージョンである。

この概念は、右シフト演算が計算上非常に効率的であり、また、確率指數が第2の範囲内にある場合、信頼できる間隔サイズ値を提供するという考えに基づいている（一方、好ましくは、確率指數が第1の範囲内にある場合は、ベースルックアップテーブルの要素にシフト演算が適用されない）。その結果、第1の範囲内と第2の範囲内の「対応する」確率指數に提供される間隔サイズ値は、主にビットシフトによって異なる（可能な丸めを除く）。これは通常、確率指數値の範囲全体、または完全な「第1の範囲」（通常は3つ以上の異なる値で構成される）にも当てはまる。さらに、右シフト演算は通常、2の効力による除算に対応することに留意されたい。

40

#### 【0094】

好ましい実施形態では、確率指數（ $Qp2 (p_{LPS})$ ）は、ルックアップテーブルの要素が間隔サイズ値（ $R_{LPS}$ ）として提供されるかどうか、またはルックアップテーブルの

50

要素が間隔サイズ値 (R<sub>LPS</sub>) を得るようにスケーリングおよび丸められるかどうかを決定する。

確率指数（または、より具体的には、確率指数が第1の範囲内にあるか、第2の範囲内にあるかという質問）が、ルックアップテーブル（ベースルックアップテーブル）の要素に基づいて、スケーリング（および任意選択で丸め）を適用して間隔サイズ値を取得するかどうか決定するため、アルゴリズムは非常に単純に保つことができる。例えば、確率指数が第1の範囲内にあるか、第2の範囲内にあるかを確認することは、確率指数を所定の値で除算するか、確率指数を1つ以上のしきい値と比較することで簡単に実行できる。したがって、確率指数に基づいて、スケーリング（および任意選択の丸め）を実行するかどうかを簡単に決定できる。したがって、間隔サイズ値の導出の概念は非常に効率的である。

#### 【0095】

算術エンコーダの好ましい実施形態では、確率指数 (i) と第1のサイズ値（例えば、 $\mu$ 、サイズ値は例えば、第1の方向のベースルックアップテーブルの拡張を説明する）および間隔サイズ指数（例えば量子化演算  $Q_{r2}(\cdot)$  を使用するなどして間隔サイズ情報または合計間隔サイズ情報  $R$  に基づいて取得できる）との間の除算の除算残差 ( $i \% \mu$ ) は、間隔サイズ値を取得するためにベースルックアップテーブルのどの要素を使用するかを決定する。

分割残差と区間サイズ指数に依存してベースルックアップテーブルのエントリを選択することにより、2次元ベースルックアップテーブルを簡単に評価でき、2次元ベースルックアップテーブルの要素には、間隔サイズ値（間隔サイズ指数で表すことができる）を使用して乗算を組み込むことができる。したがって、2次元のベースルックアップテーブルを有することにより、間隔サイズ値を使用した乗算を省くことができる（分割残差を第1のテーブルインデックスとして使用でき、間隔サイズ指数を第2のテーブルインデックスとして使用できる）。さらに、第1のテーブルインデックスとしての除算残差の使用は、ベースルックアップテーブルの要素が確率指数の増加に伴って周期的に選択されるという事実によく適合している（確率指数の後続の範囲がベースルックアップテーブルの共通範囲を選択するため）。結論として、上記の実装は、ベースルックアップテーブルの要素への非常に簡単なアクセスを可能にし、ベースルックアップテーブルの2次元の性質により、間隔サイズ値との乗算を回避するのに役立つ。

#### 【0096】

好ましい実施形態では、算術エンコーダは、

$$R_{XPS} = \text{Scal}(\text{BaseTabLPS}[i \% \mu][j], [i / \mu])$$

BaseTabLPS は、次元  $\mu \times$  のベースルックアップテーブルであり、i は、確率情報に関連付けられたテーブルインデックスであり、j は、間隔サイズ情報に関連付けられたテーブルインデックスであり（例えば、合計間隔サイズ情報 R）、% は除算残差演算であり、/ は除算演算であり、Scal(x, y) はスケーリング関数であり（例えば、 $\text{Scal}(x, y) = [x * a - b * y]$  として定義され、[.] はフロア演算であり、a は、好ましくは 2 以上の定数であり、または b は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、y は、x の右側にシフトするかどうか、および何ビットシフトするかを決定する、

ことに従って、間隔サイズ値  $R_{XPS}$ （例えば、 $R_{LPS}$ ）を取得するように構成される。

間隔サイズ情報を構成する間隔サイズ値を決定するためのそのような概念は、計算効率が高く、比較的小さなベースルックアップテーブルの使用を可能にすることがわかっている。特に、乗算演算を回避できる。さらに、分割残差演算および除算演算は、例えば、次元  $\mu$  が 2 の効力である場合、非常に計算効率の高い方法で実施することもできる。したがって、間隔サイズ値の導出について説明した概念により、計算上非常に効率的な実装が可能になる。

#### 【0097】

好ましい実施形態では、算術エンコーダは、1つまたは複数の状態変数値 ( $s_i^k$ ) (

10

20

30

40

50

これはインデックス  $k$  によって示される所与のコンテキストモードに関連付けられる) に基づいて符号化される 1 つまたは複数のシンボル値の算術符号化の間隔サイズ値 (  $R_{LPS}$  ) を導出するように構成され、これは、以前に符号化された複数のシンボル値の統計を表し ( 例えば、バイナリ値 0 と 1 のシーケンス ) ( 例えば、異なる適合時定数で ) ; 算術エンコーダは、1 つまたは複数の状態変数値から導出された ( 現在の ) 確率値に基づいて、また ( 現在の ) コーディング間隔サイズ (  $R$  ) に基づいて、確率テーブル (  $Prob\_TabLPS$  ) ( 確率指数という観点で次元が可能な確率指数  $i$  の数値より小さい ) を使用して間隔サイズ値 (  $R_{LPS}$  ) を決定するように構成され、確率テーブルは、複数の確率値のセット ( 例えば、0 と  $\mu - 1$  の間の確率指数に対して ) および ( 単一の ) 所与の ( 参照 ) コーディング間隔サイズ ( 間隔サイズ値 ) を記述し、算術エンコーダは、現在の確率値が複数の確率値のセットにない ( 例えば、 $\mu$  以上の現在の確率値と関連付けられた確率指数である ) 場合、および / または現在のコーディング間隔サイズ (  $R$  ) が、所与 ( 参照 ) のコーディング間隔サイズとは異なる場合、間隔サイズ値 (  $R_{LPS}$  ) を取得するよう所与 ( 参照 ) の確率テーブル (  $Prob\_TabLPS$  ) の要素をスケーリングするように構成され、また、算術エンコーダは、間隔サイズ値 (  $R_{LPS}$  ) を使用して 1 つまたは複数のシンボルの算術符号化を実行するように構成される。

この概念は、確率値 ( または確率指数 ) の異なる ( 重複しない ) 範囲に関連付けられた間隔サイズ値が、スケーリング操作によって実質的に ( 可能な丸め効果を除いて ) 関連付けられているという考えに基づいている。ルックアップテーブル ( 確率テーブル ) のサイズが適切に選択されている場合、スケーリング演算は、例えば、ビットシフト演算を使用して、計算効率の高い方法で実装され得ることにも留意されたい。その結果、小さな計算労力で算術コーディング ( 符号化または復号 ) に使用され、メモリを省く小さなサイズのルックアップテーブルのみを使用する間隔サイズ値を導出することが可能である。

#### 【 0098 】

好ましい実施形態では、算術エンコーダは、( 現在の ) コーディング間隔サイズ (  $R$  ) に応じて、確率テーブルの選択された要素 (  $probTabLPS[i \% \mu]$  ) の ( 乗法の ) 第 1 のスケーリングを使用し、現在の確率値 ( 指数  $i$  により指定 ) に関連付けられた要素が複数の確率値のセットに含まれるかどうかに応じて ( 例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて ) 、第 1 のスケーリングの結果の第 2 のスケーリングを使用して間隔サイズ値を取得されるように構成される。

2 段階の乗算またはスケーリングを使用して、間隔サイズ情報を取得することにより、小さな確率テーブルを使用できる。例えば、確率テーブルは、単一のコーディング間隔サイズおよび所与の比較的狭い範囲の確率値 ( 「複数の確率値のセット」 によって表され得る ) のみを「直接」カバーし得る。したがって、他のいずれかのコーディング間隔サイズ、および確率テーブルによって「直接」カバーされる複数の確率値のセットに含まれない任意の確率値について、意味のある信頼できる間隔サイズ値が取得されるように、スケーリングが実行される。

#### 【 0099 】

算術エンコーダの好ましい実施形態では、確率指数 ( 例えば  $i$  、例えば、現在の確率値を表す ) および第 1 のサイズ値 ( 例えば  $\mu$  、例えばサイズ値が確率テーブルの拡張について説明する ) の間の除算の除算残差 (  $i \% \mu$  ) は、確率テーブルのどの要素が第 1 のスケーリングでスケーリングされるかを決定する、および / または確率指数 (  $i$  ) および第 1 のサイズ値間の除算の整数除算結果 (  $[i / \mu]$  ) は、第 2 のスケーリングで使用されるスケーリング係数 (  $2 - [i / \mu]$  ) を決定する、および / またはコーディング間隔サイズ (  $R$  ) は、第 1 のスケーリングの乗法スケーリング係数 (  $Qr2(R)$  ) を決定する。

除算残差を使用して確率のどの要素がスケーリングされるかを決定することは、確率テーブルのエントリが確率指数の増加とともに定期的に再利用される ( 例えば、周期的に ) という事実を活用するのに役立つ。除算残差の使用は、この事実を表している。また、除算残差は、状況によっては、特に除算が 2 の効力で行われる場合、非常に高い計算効率で

10

20

30

40

50

計算できる。

さらに、整数除算の結果に基づいてスケーリング係数を決定することにより、スケーリング係数を確率指數値の異なる（隣接する）範囲に簡単に割り当てることができる。さらに、整数除算の結果は、状況によっては、特に除算が2の効力で行われる場合、計算上非常に効率的な方法で計算できる。

さらに、コーディング間隔サイズに依存して乗法スケーリング係数を決定することは、間隔サイズ値がコーディング間隔サイズに比例するという事実を反映している。したがって、間隔サイズ値を高い効率かつ精度で得ることができる。

#### 【0100】

好ましい実施形態では、算術エンコーダは、

$$R_{XPS} = Scal([probTabLPS[i \% \mu] \cdot Qr_2(R)], [i/\mu])$$

10

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は間隔サイズ情報に関連付けられたテーブルインデックスであり、 $\%$  は除算残差演算であり、 $/$  は除算演算であり、`probTabLPS` [ ] は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり ( $i$  の値の範囲は通常  $\mu$  よりも大きくなる)、 $R$  は間隔サイズ（または現在のコーディング間隔サイズ）であり、 $Qr_2(R)$  は  $R$  に依存するスケーリング係数であり、`Scal(x, y)` はスケーリング関数であり（例えば、 $Scal(x, y) = [x * a - b^y]$  として定義され、[.] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

20

ことに従って間隔サイズ値  $R_{XPS}$ （例えば、 $R_{LPS}$ ）を取得するように構成される。

間隔サイズ値を決定するためのそのようなアルゴリズムは、計算上効率的であり、良質の結果を提供することがわかっている。確率テーブルは比較的小さくすることができ、スケーリング関数は、例えば、1つまたは複数のビットシフト演算を使用して、計算効率の良い方法で実装することができる。

#### 【0101】

好ましい実施形態では、算術エンコーダは、現在の確率値（指數  $i$  により指定）に関連付けられた要素が確率値に含まれるかどうかに応じて（例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて）、確率テーブルの選択された要素（`probTabLPS[i \% \mu]`）の（乗法の）第1のスケーリングを使用し、また、コーディング間隔サイズ（ $R$ ）に応じて、第1のスケーリングの結果の第2のスケーリングを使用して、間隔サイズ値を取得されるように構成される。

30

この概念では、上記の概念と比較すると、第1のスケーリングと第2のスケーリングの処理順序が逆になっている。ただし、基本的な考慮事項は同じままである。

#### 【0102】

好ましい実施形態では、確率指數（例えば  $i$ 、例えば、現在の確率値を表す）、および第1のサイズ値（例えば  $\mu$ 、例えばサイズ値が確率テーブルの拡張について説明する）の間の除算の除算残差（ $i \% \mu$ ）は、確率テーブルのどの要素が第1のスケーリングでスケーリングされるかを決定する、および / または確率指數（ $i$ ）および第1のサイズ値間の除算の整数除算結果（ $[i / \mu]$ ）は、第1のスケーリングで使用されるスケーリング係数（例えば、 $2 - [i / \mu]$  または  $a - b[i / \mu]$ ）を決定する、および / または（現在の）コーディング間隔サイズ（ $R$ ）は、第2のスケーリングの乗法スケーリング係数（ $Qr_2(R)$ ）を決定する。

40

この概念では、上記の概念と比較すると、第1のスケーリングと第2のスケーリングの順序が逆になっている。ただし、基本的な考慮事項は変更されていない。

#### 【0103】

好ましい実施形態では、算術エンコーダは、

$$R_{XPS} = Scal(probTabLPS[i \% \mu], [i/\mu]) \cdot Qr_2(R)$$

50

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は（現在の）間隔サイズ情報に関連付けられたテーブルインデックスであり、 $\%$  は除算残差演算であり、 $/$  は除算演算であり（例えば、整数の結果を提供する）、 $\text{probTableLPS} [ ]$  は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり（ $i$  の値の範囲は通常  $\mu$  よりも大きくなる）、 $R$  は間隔サイズ（または現在のコーディング間隔サイズ）であり、 $Qr_2(R)$  は  $R$  に依存するスケーリング係数であり、 $\text{Scal}(x, y)$  はスケーリング関数であり（例えば、 $\text{Scal}(x, y) = [x * a^{-b} y]$  として定義され、 $[.]$  はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する。 10

ことに従って間隔サイズ値  $R_{XPS}$ （例えば、 $R_{LPS}$ ）を取得するように構成される。

この概念では、上記の実装と比較すると、第 1 のスケーリングおよび第 2 のスケーリングのスケーリング順序が逆になっている。ただし、基本的な基礎となる考え方は変わりない。

#### 【0104】

以下では、算術復号に関連するいくつかの実施形態について説明する。ただし、算術復号に関連する概念、考慮事項、およびこれらの概念の基礎となる詳細は、算術符号化の概念の基礎となる概念、考慮事項、および詳細と実質的に同じである。したがって、上記の説明も同様に当てはまる。ただし、符号化されるシンボル値は、復号されるシンボル値または以前に復号されたシンボルに対応し、以前に符号化されたシンボル値は、以前に復号されたシンボル値に対応する。さらに、符号化特徴と復号特徴との間の対応は、当業者には明らかであり、請求項の文言の比較からも明らかである。 20

#### 【0105】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを復号するための算術デコーダを作成するものであり、算術デコーダは、1つまたは複数の状態変数値（ $s_{i^k}$ ）に基づいて復号される1つまたは複数のシンボル値の算術復号の間隔サイズ値（ $R_{LPS}$ ）（それは例えば、インデックス  $k$  で示される所与のコンテキストモードに関連付けられている）を導出するように構成され、これは、以前に復号された複数のシンボル値（例えば、バイナリ値 0 および 1 のシーケンス）の統計を表し（例えば、異なる適合時定数を使用）、算術デコーダは、ベースルックアップテーブル（ $\text{BaseTableLPS}$ ）を使用して間隔サイズ値（ $R_{LPS}$ ）を決定するように構成され（その次元は、確率指數の観点から、可能な確率指數  $i$  の数よりも小さい）、算術デコーダは、1つまたは複数の状態変数値（例えば、 $i = Qp(p_{LPS})$  のように）を基にして得られる確率指數（ $i$ ）が第 1 の範囲（例えば、 $\mu$  未満）内にある場合、決定された間隔サイズ値がベースルックアップテーブルの要素と同一であるか、またはベースルックアップテーブルの要素の丸めのバージョンであり、確率指數が第 2 の範囲（例えば、 $\mu$  以上）内にある場合は、ベースルックアップテーブルの要素のスケーリングと丸めを使用して、決定された間隔サイズ値が取得されるように、間隔サイズ値（ $R_{LPS}$ ）が決定されるように構成され、算術デコーダは、間隔サイズ値（ $R_{LPS}$ ）を使用して1つまたは複数のシンボルの算術復号を実行するように構成される。 30

#### 【0106】

好ましい実施形態では、算術デコーダは、間隔サイズ値を決定するように構成され、決定された間隔サイズ値（ $R_{LPS}$ ）は、確率指數が第 2 の範囲内にある場合、ベースルックアップテーブルの要素の右シフトバージョンである。 40

#### 【0107】

算術デコーダの好ましい実施形態では、確率指數（ $Qp2(p_{LPS})$ ）は、ルックアップテーブルの要素が間隔サイズ値（ $R_{LPS}$ ）として提供されるかどうか、またはルックアップテーブルの要素が間隔サイズ値（ $R_{LPS}$ ）を得るようスケーリングおよび丸められるかどうかを決定する。

#### 【0108】

10

20

30

40

50

算術デコーダの好ましい実施形態では、確率指数 ( $i$ ) と第 1 のサイズ値 (例えば、 $\mu$ 、サイズ値は例えば、第 1 の方向のベースルックアップテーブルの拡張を説明する) および間隔サイズ指数 (例えば量子化演算  $Q_{r2}(\cdot)$  を使用するなどして間隔サイズ情報または合計間隔サイズ情報  $R$  に基づいて取得できる)との間の除算の除算残差 ( $i \% \mu$ ) は、間隔サイズ値を取得するためにベースルックアップテーブルのどの要素を使用するかを決定する。

#### 【0109】

好ましい実施形態では、算術デコーダは、

$$R_{XPS} = \text{Scal}(\text{BaseTabLPS}[i \% \mu][j], [i / \mu])$$

$\text{BaseTabLPS}$  は、次元  $\mu \times$  のベースルックアップテーブルであり、 $i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は、間隔サイズ情報に関連付けられたテーブルインデックスであり (例えば、合計間隔サイズ情報  $R$ )、% は除算残差演算であり、/ は除算演算であり、 $\text{Scal}(x, y)$  はスケーリング関数であり (例えば、 $\text{Scal}(x, y) = [x * a - b y]$  として定義され、[.] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、または  $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右側にシフトするかどうか、および何ビットシフトするかを決定する。

ことに従って、間隔サイズ値  $R_{XPS}$  (例えば、 $R_{LPS}$ ) を取得するように構成される。

#### 【0110】

本発明による実施形態は、シンボル値 (例えば、バイナリ値) を有する複数のシンボルを復号するための算術デコーダを作成するものであり、算術デコーダは、1 つまたは複数の状態変数値 ( $s_i^k$ ) (これは、例えば、インデックス  $k$  によって示される所与のコンテキストモードに関連付けられる) に基づいて復号される 1 つまたは複数のシンボル値の算術復号の間隔サイズ値 ( $R_{LPS}$ ) を導出するように構成され、これは、以前に復号された複数のシンボル値の統計を表し (例えば、バイナリ値 0 と 1 のシーケンス) (例えば、異なる適合時定数で) ; 算術デコーダは、1 つまたは複数の状態変数値から導出された (現在の) 確率値に基づいて、また (現在の) コーディング間隔サイズ ( $R$ ) に基づいて、確率テーブル ( $\text{ProbTabLPS}$ ) (確率指数という観点で次元が可能な確率指数  $i$  の数値より小さい) を使用して間隔サイズ値 ( $R_{LPS}$ ) を決定するように構成され、確率テーブルは、複数の確率値のセット (例えば、0 と  $\mu - 1$  の間の確率指数に対して) および (单一の) 所与の (参照) コーディング間隔サイズの間隔サイズ (間隔サイズ値) を記述し、算術デコーダは、現在の確率値が複数の確率値のセットにない (例えば、 $\mu$  以上の現在の確率値と関連付けられた確率指数である) 場合、および / または現在のコーディング間隔サイズ ( $R$ ) が、所与 (参照) のコーディング間隔サイズとは異なる場合、間隔サイズ値 ( $R_{LPS}$ ) を取得するよう確率テーブル ( $\text{ProbTabLPS}$ ) の要素 (例えば、現在の確率値に依存して選択された要素) をスケーリングするように構成され、また、算術デコーダは、間隔サイズ値 ( $R_{LPS}$ ) を使用して 1 つまたは複数のシンボルの算術復号化を実行するように構成される。

#### 【0111】

好ましい実施形態では、算術デコーダは、(現在の) コーディング間隔サイズ ( $R$ ) に応じて、確率テーブルの選択された要素 ( $\text{ProbTabLPS}[i \% \mu]$ ) の (乗法の) 第 1 のスケーリングを使用し、現在の確率値 (指数  $i$  により指定) に関連付けられた要素が複数の確率値のセットに含まれるかどうかに応じて (例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて) 、第 1 のスケーリングの結果の第 2 のスケーリングを使用して間隔サイズ値を取得されるように構成される。

#### 【0112】

算術デコーダの好ましい実施形態では、確率指数 (例えば  $i$ 、例えば、現在の確率値を表す)、および第 1 のサイズ値 (例えば  $\mu$ 、例えばサイズ値が確率テーブルの拡張について説明する) の間の除算の除算残差 ( $i \% \mu$ ) は、確率テーブルのどの要素が第 1 のスケーリングの結果の第 2 のスケーリングを使用して間隔サイズ値を取得されるように構成される。

10

20

30

40

50

ーリングでスケーリングされるかを決定する、および / または確率指數 (  $i$  ) および第 1 のサイズ値間の除算の整数除算結果 (  $[i / \mu]$  ) は、第 2 のスケーリングで使用されるスケーリング係数 (  $2 - [i / \mu]$  ) を決定する、および / またはコーディング間隔サイズ (  $R$  ) は、第 1 のスケーリングの乗法スケーリング係数 (  $Qr_2(R)$  ) を決定する。

#### 【0113】

好ましい実施形態では、算術デコーダは、

$$R_{XPS} = Scal([probTabLPS[i \% \mu] \cdot Qr_2(R)], [i / \mu])$$

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は間隔サイズ情報に関連付けられたテーブルインデックスであり、% は除算残差演算であり、/ は除算演算であり、probTabLPS [ ] は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり ( $i$  の値の範囲は通常  $\mu$  よりも大きくなる)、 $R$  は間隔サイズ ( または現在のコーディング間隔サイズ ) であり、 $Qr_2(R)$  は  $R$  に依存するスケーリング係数であり、Scal (  $x, y$  ) はスケーリング関数であり ( 例えば、 $Scal(x, y) = [x * a - b y]$  ) として定義され、[ . ] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って間隔サイズ値  $R_{XPS}$  ( 例えば、 $R_{LPS}$  ) を取得するように構成される。

#### 【0114】

好ましい実施形態では、算術デコーダは、現在の確率値 ( 指数  $i$  により指定 ) に関連付けられた要素が確率値に含まれるかどうかに応じて ( 例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて )、確率テーブルの選択された要素 ( probTabLPS [  $i \% \mu$  ] ) の ( 乗法の ) 第 1 のスケーリングを使用し、また、コーディング間隔サイズ (  $R$  ) に応じて、第 1 のスケーリングの結果の第 2 のスケーリングを使用して、間隔サイズ値を取得されるように構成される。

#### 【0115】

算術デコーダの好ましい実施形態では、確率指數 ( 例えば  $i$  、例えば、現在の確率値を表す )、および第 1 のサイズ値 ( 例えば  $\mu$  、例えばサイズ値が確率テーブルの拡張について説明する ) との間の除算の除算残差 (  $i \% \mu$  ) は、確率テーブルのどの要素が第 1 のスケーリングでスケーリングされるかを決定する、および / または確率指數 (  $i$  ) および第 1 のサイズ値間の除算の整数除算結果 (  $[i / \mu]$  ) は、第 1 のスケーリングで使用されるスケーリング係数 ( 例えば、 $2 - [i / \mu]$  または  $a - b [i / \mu]$  ) を決定する、および / または ( 現在の ) コーディング間隔サイズ (  $R$  ) は、第 2 のスケーリングの乗法スケーリング係数 (  $Qr_2(R)$  ) を決定する。

#### 【0116】

好ましい実施形態では、算術デコーダは、

$$R_{XPS} = Scal(probTabLPS[i \% \mu], [i / \mu]) \cdot Qr_2(R)$$

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は ( 現在の ) 間隔サイズ情報に関連付けられたテーブルインデックスであり、% は除算残差演算であり、/ は除算演算であり ( 例えば、整数の結果を提供する )、probTabLPS [ ] は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり ( $i$  の値の範囲は通常  $\mu$  よりも大きくなる)、 $R$  は間隔サイズ ( または現在のコーディング間隔サイズ ) であり、 $Qr_2(R)$  は  $R$  に依存するスケーリング係数であり、Scal (  $x, y$  ) はスケーリング関数であり ( 例えば、 $Scal(x, y) = [x * a - b y]$  ) として定義され、[ . ] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って間隔サイズ値  $R_{XPS}$  ( 例えば、 $R_{LPS}$  ) を取得するように構成される。

10

20

30

40

50

## 【0117】

以下では、算術符号化に関連するいくつかのさらなる実施形態について説明する。

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための算術エンコーダを作成するものであり、算術エンコーダは、以前に符号化された複数のシンボル値の統計を表す1つの状態変数値（ $s_k$ ）を決定するように構成され、算術エンコーダは、結合状態変数値、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ）から、符号化されるシンボル値の算術符号化のためのサブインターバル幅値（ $R_{LPS}$ ）

10

）を計算するよう構成され、この計算は、そのエントリが結合状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ）；

$[s_k \cdot a_k]$

）を含む1次元ルックアップテーブル（

$probTabLPS[Qp_2(\dots)]$

）を使用して、1つの状態変数値（ $s_k$ ）、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ）；

$[s_k \cdot a_k]$

20

）を確率値にマッピングすること、および量子化レベル（

$Qp_2(R)$

）に符号化されるシンボル値の算術符号化の前に、算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、 $R$ ）を量子化することにより、これを行い、算術エンコーダは、さらに、確率値と、量子化レベルの間の積を決定する（事前に計算された積のルックアップまたは乗算によって）ように構成され、算術エンコーダは、符号化されるシンボル値に依存して状態変数値の更新を実行するよう構成される。

30

## 【0118】

この実施形態は、非常に単純な一次元ルックアップテーブルを、結合状態変数値に基づくサブインターバル幅値の決定に使用できるという発見に基づいています。コーディング間隔サイズは、コーディング間隔サイズ情報の量子化、および確率値と量子化値（または量子化レベル）の間の積の決定によって考慮される。したがって、信頼できる結果は適度な努力で得られる。

## 【0119】

好ましい実施形態では、算術エンコーダは、1つの状態変数値（ $s_k$ ）として、結合状態変数値（例えば、状態変数値の加重和であり得る）、複数の状態変数値（ $s_{i^k}$ ）（例えば、バイナリ値0および1のシーケンス）（例えば、複数の状態変数値の場合、異なる適合時定数の統計）の基礎となるものを導出するように構成され、これは、異なる適合時定数の複数の以前に符号化されたシンボル値（例えば、バイナリ値0および1のシーケンス）の統計を表す。

40

結合状態変数値を1つの状態変数値として使用すると、特に良好な結果が得られることがわかっている。様々な適合時定数を考慮すると、短時間の統計と長時間の統計の両方を考慮することができるため、サブインターバル幅の値の信頼性が特に高くなる。

## 【0120】

好ましい実施形態では、算術エンコーダは、結合状態変数値を取得するために、状態変数値の加重和を決定するように構成される。

結合状態変数値のそのような計算は、計算の労力を適度に少なく保ちながら、結合状態

50

変数値に対する短期統計と長期統計の異なる影響の考慮を可能にする。

【0121】

好ましい実施形態では、算術エンコーダは、状態変数値 ( $s_i^k$ )

) および関連する重み付け値 ( $d_i^k$ )。

結合状態変数値 ( $s_k$ ) を得るため) の積を丸めることによって得られる丸められた値 ( $[s_i^k \cdot d_i^k]$ )

10

) の合計を決定するように構成される。

合計の前に丸め演算を適用すると、計算の労力が軽減され、状態変数値と関連する重み付け値の非常に小さな積の影響も排除される。したがって、信頼性が向上する。

好ましい実施形態では、算術エンコーダは、

$$s_k = \sum_{i=1}^N [s_i^k \cdot d_i^k]$$

$s_k$  は状態変数値であり、 $N$  は考慮される状態変数値の数であり、 $[ \cdot ]$  はフロア演算子であり、 $d_i^k$  は状態変数値に関連付けられた重み付け値 (例えば、結合状態変数値への個々の状態変数値の影響を制御する重み付け係数) ( $d_i^k$  は好ましくはファン 2 の整数値の効力であり、2 つの異なる  $d_i^k$  間の比率は好ましくは 2 の整数値の効力である) [2 つの異なる  $d_i^k$  間の比率は、好ましくは 8 以上] であることに従って、結合状態変数値  $s_k$  を決定するように構成される。

結合状態変数値のそのような導出は特に有利であることがわかっている。結合状態変数値を決定するための対応する概念の上記の説明の言及もなされている。

【0122】

好ましい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + [-s_i^k \cdot m_i^k]] \cdot n_i^k], & \text{If symbol to be encoded is 1.} \\ s_i^k - [A[z + [-s_i^k \cdot m_i^k]] \cdot n_i^k], & \text{If symbol to be encoded is 0.} \end{cases}$$

20

30

$z$  は所定の (一定の) オフセット値であり、  
 $m_i^k$

は 1 つまたは複数の重み付け値であり、  
 $n_i^k$

は 1 つ以上の重み付け値であり、 $A$  は

$$A[z + \hat{s}] = offset + \sum_{i=1}^N \max(0, \hat{b}_i - \hat{s}) \ll \hat{a}_i$$

40

であるか、ゼロ設定または大きさの縮小によって、引数の 1 つ以上の極値に対してのみ、それから逸脱して、更新された

$s_i^k$

を所定の値の範囲のままにすることを回避する (例えば、  
 $s_i^k$

は

$s_i^k \cdot m_i^k$

50

より大きな値ドメインを持つ；つまり、

$s_i^k$

は

$s_i^k \cdot m_i^k$

に準量子化される；

$s_i^k \cdot m_i^k$

の極値の場合、

$s_i^k$

は上記の式に従って未修正の  $A$  によって変更され、その値ドメインを離れることができる：これを回避するために、これらの極値に対応するエントリが削減またはゼロにされる場合があることを考慮されたい）、

$offset$

$\hat{N}$

10

$\hat{b}_i$

、および

$\hat{a}_i$

は所定のパラメータである（例は上記に記載されている）

ことに従って、状態変数値の更新を行う際、複数の状態変数値

$s_i^k$

を更新するように構成される。

状態変数値のそのような更新は特に有利であることがわかっている。状態変数の更新について、この概念に関する上記の説明も参照される。

【0 1 2 3】

好ましい実施形態では、算術エンコーダは、テーブルルックアップまたは計算によって  $A[z + \lfloor s_i^k \cdot m_i^k \rfloor]$

を導出するように構成される。

この概念に関しては、上記の説明が参照される。

【0 1 2 4】

好ましい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 1.} \\ s_i^k - \lfloor A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 0.} \end{cases}$$

$A$  はルックアップテーブルであり（例えば、整数の値を含む）、 $z$  は所定の（一定の）オフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

20

30

40

50

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

を決定するように構成される。

1 つまたは複数の状態変数値を更新するためのこの概念の利点については、上記の説明が参照される。

【 0 1 2 5 】

好ましい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

10

A はルックアップテーブルであり（例えば、整数の値を含む）、z は所定の（一定の）オフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

20

を決定するように構成される。

1 つまたは複数の状態変数値を更新するためのこの概念の利点について、上記の説明が参照される。

【 0 1 2 6 】

好ましい実施形態では、算術エンコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

30

A はルックアップテーブルであり（例えば、整数の値を含む）、z は所定の（一定の）オフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

40

を決定するように構成される。

1 つまたは複数の状態変数値を更新するためのこの概念の利点については、上記の説明が参照される。

【 0 1 2 7 】

好ましい実施形態では、算術エンコーダは、コーディング間隔サイズ情報に論理右シフトを適用することによってコーディング間隔サイズ情報の量子化を実行するように構成される。

コーディング間隔サイズ情報の論理的な右シフトは、計算上非常に効率的である。

50

好ましい実施形態では、算術エンコーダは、コーディング間隔サイズ情報

<sub>R</sub>

の量子化を

$$Q_{T_2}(R) = (\lfloor R \cdot 2^{-u} \rfloor + v) \cdot 2^{-w}$$

により実行するように構成され、式中、

<sub>u</sub>

<sub>v</sub>

および

<sub>w</sub>

はパラメータである。

コーディング間隔サイズ情報のこの量子化の利点に関しては、上記の論述が参照される。

算術エンコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、1つの状態変数値、またはそのスケーリングおよび／または丸められたバージョン ( $[s^k * a^k]$ ) の増加で単調に減少する。

1次元ルックアップテーブルのこの構造の利点については、上記の説明が参照される。

算術エンコーダの好ましい実施形態では、1つの状態変数値の値ドメインの異なる値間隔、またはそのスケーリングおよび／または丸められたバージョン ( $[s^k * a^k]$ ) は、等しいサイズである。

この概念の利点については、上記の説明が参照される。

#### 【0128】

算術エンコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、1つの状態変数値、またはそのスケーリングおよび／または丸められたバージョン ( $[s^k * a^k]$ ) の増加で減少率を伴って単調に減少する。

この概念の利点については、上記の説明が参照される。

以下では、算術符号化についての上記の概念に対応する算術復号の概念について説明する。したがって、同じ説明が適用され、上記と同じ詳細を任意選択で使用できる。ただし、算術エンコーダは算術デコーダに対応することに留意されたい。さらに、以前に符号化されたシンボル値は、通常、以前に復号されたシンボル値に対応し、符号化されるシンボル値は、通常、以前に復号されたシンボル値（または復号されるシンボル値）に対応し得る。ただし、特徴の対応に関しては、関連する（または対応する）概念を定義する対応するクレームの比較も参照される。

#### 【0129】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを復号するための算術デコーダを作成し、算術デコーダは、間隔サイズ情報 ( $p^k, R * p^k$ ) を、1つまたは複数のシンボル値（例えば、バイナリ値）の算術復号が複数の状態変数値 ( $s_i^k$ )（例えば、インデックス  $k$  によって表示される所与のコンテキストモードに関連付けられるもの）に基づいて復号されるように導出するように構成され、これは、異なる適合時定数で以前に復号された複数のシンボル値（例えば、バイナリ値0と1のシーケンス）の統計（例えば、復号される1つまたは複数のシンボルが特定のシンボル値を含む確率の推定）を表し、算術デコーダは、ルックアップテーブル (LUT1) を使用して、第1の状態変数値 ( $s^k_1$ )、またはそのスケーリングおよび／または丸められたバージョン ( $[s^k_1 * a^k_1]$ ) をマッピングし、ルックアップテーブル (LUT1) を使用して、第2の状態変数値 ( $s^k_2$ )、またはそのスケーリングおよび／または丸められたバージョン ( $[s^k_2 * a^k_2]$ ) をマッピングするように構成され、復号される1つまた

10

20

30

40

50

は複数のシンボルの算術復号のための間隔サイズを説明する間隔サイズ情報（例えば、 $p_k$  または  $R * p_k$ ）を取得するようとする。

【0130】

好ましい実施形態では、算術デコーダは、ルックアップテーブルを使用して、第1の状態変数値、またはそのスケーリングおよび／または丸められたバージョン ( $[s^k_1 * a^k_1]$ ) を第1の確率値 ( $p_k^1$ ) にマッピングするように構成され、算術デコーダは、ルックアップテーブルを使用して、第2の状態変数値、またはそのスケーリングおよび／または丸められたバージョン ( $[s^k_2 * a^k_2]$ ) を第2の確率値 ( $p_k^2$ ) にマッピングするように構成され、算術デコーダは、第1の確率値および第2の確率値を使用して（例えば、加重和または加重平均を使用して）結合確率値 ( $p_k$ ) を取得するように構成される。

好ましい実施形態では、算術デコーダは、復号されるシンボルが第1の値（例えば「1」）をとる場合に状態変数値を第1の方向に変更する（例えばより正になる）ように構成され、また復号されるシンボルが第1の値とは異なる第2の値（例えば「0」）をとる場合（例えば、状態変数値が正と負の値をとることができるように）に状態変数値を第2の方向に（例えば、より負になるもの）変更するように構成され、算術デコーダは、それぞれの状態変数値の絶対値 ( $s_k^i > 0$  の場合は  $s_k^i$ 、それ以外の場合は  $-s_k^i$ ) に応じて（例えば、状態変数値の絶対値のスケーリングおよび丸められたバージョンに応じて）評価されるルックアップテーブルのエントリを決定するように構成される。

【0131】

好ましい実施形態では、算術デコーダは、第1の状態変数値が第1の符号（例えば、正の符号）をとる場合に、第1の確率値 ( $p_k^1$ ) をルックアップテーブルによって提供される値（例えば、

$$LUT1[[s_i^k \cdot a_i^k]]$$

）に設定するように構成され、算術デコーダは、第1の状態変数値が第2の符号（例えば、負の符号）をとる場合、第1の確率値 ( $p_k^1$ ) を、所定の値（例えば、1）からルックアップテーブルによって提供される値（例えば、

$$1 - LUT1[[s_i^k \cdot a_i^k]]$$

）を減算することによって得られる値に設定するように構成される。

【0132】

好ましい実施形態では、算術デコーダは、

$$p_i^k = \begin{cases} LUT1[[s_i^k \cdot a_i^k]], & \text{if } s_i^k \geq 0. \\ 1 - LUT1[[s_i^k \cdot a_i^k]], & \text{else.} \end{cases}$$

LUT1は確率値を含むルックアップテーブルであり、[.]はフロア演算子であり、 $s^k_i$  は  $i$  番目の状態変数値であり、 $a^k_i$  は、 $i$  番目の状態変数値に関連付けられた重み付け値である（例えば、 $i$  番目の状態変数値の数の範囲をルックアップテーブルのいくつかのエントリに適合させる重み付け値）ことにより、2つ以上の確率値  $p^k_i$  を決定するよう構成される。

【0133】

好ましい実施形態では、算術デコーダは、

$$p_i^k = \begin{cases} LUT1[[s_i^k \cdot a_i^k]], & \text{if } |s_i^k \cdot a_i^k| \geq 0. \\ 1 - LUT1[[-s_i^k \cdot a_i^k]], & \text{else} \end{cases}$$

LUT1は確率値を含むルックアップテーブルであり、[.]はフロア演算子であり、 $s^k_i$  は  $i$  番目の状態変数値であり、 $a^k_i$  は、 $i$  番目の状態変数値に関連付けられた重み付け値である（例えば、 $i$  番目の状態変数値の数の範囲をルックアップテーブルのいくつかのエントリに適合させる重み付け値）ことにより、2つ以上の確率値  $p^k_i$  を決定するよう構成される。

10

20

30

40

50

## 【0134】

好ましい実施形態では、算術デコーダは、

$$p_k = \sum_{i=1}^N p_i^k \cdot b_i^k$$

式中  $N$  は考慮される確率値の数で（考慮される状態変数値の数に等しい場合がある）、 $b^k_i$  は、重み付け値（例えば、結合確率値に対する個々の状態変数値の影響を制御する重み付け係数）である [ $b^k_i$  は、好ましくは、 $v \circ n - 2$  の整数値の効力であり、2つの異なる  $b^k_i$  間の比は好ましくは、2の整数値の効力である]

ことに従って、複数の確率値  $p^k_i$  に基づいて結合確率値  $p^k$  を取得するように構成される。 10

## 【0135】

好ましい実施形態では、算術デコーダは、2次元ルックアップテーブルを使用して、第1の状態変数値、またはそのスケーリングおよび/または丸められたバージョン ( $[s^k_1 * a^k_1]$ ) を第1のサブインターバル幅値 ( $R * p^k_1$ ) にマッピングするように構成され、そのエントリは、第1の状態変数値に依存して（例えば、第1のルックアップテーブルエントリ座標を決定するために）、およびシンボルの復号前の算術復号のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、 $R$ ）に依存して（例えば、第2のルックアップテーブルエントリ座標を決定するために）アドレス指定され、算術デコーダは、2次元ルックアップテーブルを使用して、第2の状態変数値、またはそのスケーリングおよび/または丸められたバージョン ( $[s^k_1 * a^k_1]$ ) を第2のサブインターバル幅値 ( $R * p^k_2$ ) にマッピングするように構成され、そのエントリは、第2の状態変数値に依存して（例えば、第1のルックアップテーブルエントリ座標を決定するために）、およびシンボルの復号前の算術復号のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、 $R$ ）に依存してアドレス指定され（例えば、第2のルックアップテーブルエントリ座標を決定するために）、算術デコーダは、第1のサブインターバル幅値および第2のサブインターバル幅値を使用して（例えば、加重和または加重平均を使用して）、結合されたサブインターバル幅値を取得するように構成される。 20

## 【0136】

算術デコーダの好ましい実施形態では、2次元ルックアップテーブルは、そのエントリが第1および第2の状態変数値の値ドメインの異なる値の確率値またはそのスケーリングおよび/または丸められたバージョン ( $[s^k * a^k]$ ;  
 $|s_k| \cdot a_k$  30

) を含む第1の1次元ベクトル（1次元ルックアップテーブルを形成する）、およびそのエントリがコーディング間隔サイズ情報の量子化レベルを含む第2の1次元ベクトル ( $Q_{T_2}(R)$ )

）の間の二項積として表すことができる。

## 【0137】

算術デコーダの好ましい実施形態では、2次元ルックアップテーブル ( $R a n g T a b L P S$ ) の要素が、ベースルックアップテーブル ( $B a s e T a b L P S$ ) に基づいて定義され、2次元ルックアップテーブルの要素の第1のグループ（またはブロック、例えば、「上半分」）は、ベースルックアップテーブルの要素と同一であるか、ベースルックアップテーブルの要素の丸められたバージョンであり、2次元ルックアップテーブルの要素の第2のグループ（またはブロック、例えば「下半分」）は、ベースルックアップテーブルの要素のスケーリングおよび丸められたバージョンである。 40

## 【0138】

算術デコーダの好ましい実施形態では、2次元ルックアップテーブルの要素の第2のグループはベースルックアップテーブルの要素の右シフトバージョンである。 50

## 【0139】

算術デコーダの好ましい実施形態では、確率指数 ( $Q p_2(p_{LPS}) ; i$ ) は、2次元ルックアップテーブルの要素の第1のグループの要素または2次元ルックアップテーブルの要素の第2のグループの要素のどちらを評価するかを決定し、確率指数（例えば、確率値、例えば  $p_{LPS}$  を定量化することによって取得される）の第1の範囲（例えば、 $0 \sim \mu - 1$ ）は、要素の第1のグループの要素に関連付けられ、確率指数（例えば、確率値、例えば  $p_{LPS}$  を、例えば、量子化関数  $Q p_2(\cdot)$  を使用して定量化することによって取得される）の第2の範囲（例えば、 $\mu$  以上）は、要素の第2のグループの要素に関連付けられている。

## 【0140】

10

算術デコーダの好ましい実施形態では、確率指数 ( $i$ ) と第1のサイズ値（例えば、 $\mu$ 、サイズ値は例えば、第1の方向のベースルックアップテーブルの拡張を説明する）および間隔サイズ指数（例えば量子化演算  $Q r_2(\cdot)$ 、例えば  $j$  を使用するなどして間隔サイズ情報  $R$  に基づいて取得できる）との間の除算の除算残差 ( $i \% \mu$ ) は、2次元ルックアップテーブルの要素を取得するためにベースルックアップテーブルのどの要素を使用するかを決定する。

## 【0141】

好ましい実施形態では、算術デコーダは、

$R a n g e T a b L P S [i] [j] = S c a l ( B a s e T a b L P S [i \% \mu] [j], [i / \mu] )$

20

$B a s e T a b L P S$  は、次元  $\mu \times$  のベースルックアップテーブルであり、 $i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は、間隔サイズ情報に関連付けられたテーブルインデックスであり（例えば、現在のコーディング間隔サイズを記述する）、% は除算残差演算であり、/ は除算演算であり、 $S c a l (x, y)$  はスケーリング関数であり（例えば、 $S c a l (x, y) = [x * a - b y]$  として定義され、[.] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、または  $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右側にシフトするかどうか、および何ビットシフトするかを決定する、

ことに従って 2 次元ルックアップテーブル ( $R a n g e T a b L P S$ ) の要素を取得するように構成される。

30

## 【0142】

算術デコーダの好ましい実施形態では、2次元ルックアップテーブル ( $R a n g e T a b L P S$ ) の要素は、確率テーブル ( $p r o b T a b L P S$ ) に基づいて定義され、確率テーブルは、複数の確率値のセット（例えば、指数  $i$  によって表される）および所与のコーディング間隔サイズの間隔サイズを記述し、複数の確率値のセットにない確率値および/または所与の（参照）コーディング間隔サイズとは異なるコーディング間隔サイズの 2 次元ルックアップテーブルの要素は、スケーリングを使用して確率テーブルから導出される。

## 【0143】

算術デコーダの好ましい実施形態では、2次元ルックアップテーブルの要素が、コーディング間隔サイズ ( $R$ ) に応じて、確率テーブルの選択された要素 ( $p r o b T a b L P S [i \% \mu]$ ) の（乗法の）第1のスケーリングを使用し、現在の確率値（指数  $i$  により指定）に関連付けられた要素が確率値のセットに含まれるかどうかに応じて（例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて）、第1のスケーリングの結果の第2のスケーリングを使用して取得される。

40

## 【0144】

算術デコーダの好ましい実施形態では、確率指数（例えば  $i$ 、例えば、現在の確率値を表す）、および第1のサイズ値（例えば  $\mu$ 、例えばサイズ値が確率テーブルの拡張について説明する）との間の除算の除算残差 ( $i \% \mu$ ) は、確率テーブルのどの要素が第1のスケーリングでスケーリングされるかを決定する、および/または確率指数 ( $i$ ) および第1のサイズ値間の除算の整数除算結果 ( $[i / \mu]$ ) は、第2のスケーリングで使用されるス

50

ケーリング係数 ( $2^{-[i/\mu]}$ ) を決定する、および / またはコーディング間隔サイズは、第 1 のスケーリングの乗法スケーリング係数 ( $Qr_2(R)$ ) を決定する。

【 0 1 4 5 】

好ましい実施形態では、算術デコーダは、

$RangeTabLPS[i][j] = Scal([probTabLPS[i \% \mu] \cdot Qr_2(R)], [i/\mu])$

i は、確率情報に関連付けられたテーブルインデックスであり、j は間隔サイズ情報に関連付けられたテーブルインデックスであり、% は除算残差演算であり、/ は除算演算であり、probTabLPS [ ] は確率テーブルであり、μ は確率テーブルの要素の数であり (I の値の範囲は通常 μ よりも大きくなる)、R は間隔サイズ (または現在のコーディング間隔サイズ) であり、Qr2(R) は R に依存するスケーリング係数であり、Scal(x, y) はスケーリング関数であり (例えば、Scal(x, y) = [x \* a - b y] として定義され、[.] はフロア演算であり、a は、好ましくは 2 以上の定数であり、b は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、y は、x の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って 2 次元ルックアップテーブルの要素

$RangeTabLPS[i][j]$

を取得するように構成される。

【 0 1 4 6 】

算術デコーダの好ましい実施形態では、現在の確率値 (指数 i により指定) に関連付けられた要素が確率値に含まれるかどうかに応じて (例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて)、確率テーブルの選択された要素 (probTabLPS [ i \% μ ]) の (乗法の) 第 1 のスケーリングを使用し、また、コーディング間隔サイズ (R) に応じて、第 1 のスケーリングの結果の第 2 のスケーリングを使用して、2 次元ルックアップテーブルの要素を取得するように構成される。

【 0 1 4 7 】

算術デコーダの好ましい実施形態では、確率指数 (例えば i、例えば、現在の確率値を表す)、および第 1 のサイズ値 (例えば μ、例えばサイズ値が確率テーブルの拡張について説明する) の間の除算の除算残差 ( $[i \% \mu]$ ) は、確率テーブルのどの要素が第 1 のスケーリングでスケーリングされるかを決定する、および / または確率指数 (i) と第 1 のサイズ値との間の除算の整数除算結果 ( $[i/\mu]$ ) は、第 1 のスケーリングで使用されるスケーリング係数 ( $2^{-[i/\mu]}$ ) を決定する、および / またはコーディング間隔サイズ (R) は、第 2 のスケーリングの乗法スケーリング係数 ( $Qr_2(R)$ ) を決定する。

【 0 1 4 8 】

好ましい実施形態では、算術デコーダは、

$RangeTabLPS[i][j] = Scal(probTabLPS[i \% \mu], [i/\mu]) \cdot Qr_2(R)$

i は、確率情報に関連付けられたテーブルインデックスであり、j は間隔サイズ情報に関連付けられたテーブルインデックスであり、% は除算残差演算であり、/ は除算演算であり (例えば、整数の結果を提供する)、probTabLPS [ ] は確率テーブルであり、μ は確率テーブルの要素の数であり (I の値の範囲は通常 μ よりも大きくなる)、R は間隔サイズであり、Qr2(R) は R に依存するスケーリング係数であり、Scal(x, y) はスケーリング関数であり (例えば、Scal(x, y) = [x \* a - b y] として定義され、[.] はフロア演算であり、a は、好ましくは 2 以上の定数であり、b は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、y は、x の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って、2 次元ルックアップテーブルの要素

10

20

30

40

50

*RangeTabLPS[i][j]*

を取得するように構成される。

**【 0 1 4 9 】**

好ましい実施形態では、算術デコーダは、第1および第2の状態変数値、またはそのスケーリングおよび／または丸められたバージョン（[s<sup>k</sup> \* a<sup>k</sup>]）からそれぞれ第1および第2のサブインターバル幅値（R \* p<sup>k</sup>）を計算するよう構成され、第1および第2の状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび／または丸められたバージョン（[s<sup>k</sup><sub>2</sub> \* a<sup>k</sup><sub>2</sub>]）を含む1次元ルックアップテーブル（LUT4）エントリを第1および第2の確率値へ使用して、第1および第2の状態変数値（s<sub>k</sub>）<sup>10</sup>、またはそのスケーリングおよび／または丸められたバージョン（[s<sup>k</sup> \* a<sup>k</sup>]）にマッピングすること、およびシンボルを量子化レベルに符号化する前に、算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、R）を量子化すること、第1および第2の確率値と、一方で量子化レベルの間の積を決定する（事前計算された積のルックアップまたは乗算によって）こと、および第1のサブインターバル幅値と第2のサブインターバル幅値を使用して（例えば、加重和または加重平均を使用して）結合されたサブインターバル幅値を取得することによる。

好ましい実施形態では、算術デコーダは、コーディング間隔サイズ情報に論理右シフトを適用することによってコーディング間隔サイズ情報の量子化を実行するよう構成される。<sup>20</sup>

**【 0 1 5 0 】**

好ましい実施形態では、算術デコーダは、コーディング間隔サイズ情報

*R*

の量子化を

$$Q\eta_2(R) = ([R \cdot 2^{-u}] + v) \cdot 2^{-w}$$

により実行するよう構成され、式中、<sup>30</sup>

*u*

*v*

および

*w*

はパラメータである。

**【 0 1 5 1 】**

算術デコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、第1および第2の状態変数値、またはそのスケーリングおよび／または丸められたバージョン（[s<sup>k</sup> \* a<sup>k</sup>]）の増加で単調に減少する。<sup>40</sup>

**【 0 1 5 2 】**

算術デコーダの好ましい実施形態では、第1および第2の状態変数値の値ドメインの異なる値間隔、またはそのスケーリングおよび／または丸められたバージョン（[s<sup>k</sup> \* a<sup>k</sup>]）は、等しいサイズになる。

**【 0 1 5 3 】**

算術デコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、第1および第2の状態変数値、またはそのスケーリングおよび／または丸められたバージョン（[s<sup>k</sup> \* a<sup>k</sup>]）の増加で減少率を伴って単調に減少する。

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボル

10

20

30

40

50

を復号するための算術デコーダを作成し、算術デコーダは、間隔サイズ情報 ( $p^k$ 、  $R * p^k$ ) を、1つまたは複数のシンボル値（例えば、バイナリ値）の算術復号が複数の状態変数値 ( $s_i^k$ )（例えば、インデックス  $k$  によって表示される所与のコンテキストモードに関連付けられるもの）に基づいて復号されるように導出するように構成され、これは、異なる適合時定数で以前に復号された複数のシンボル値（例えば、バイナリ値 0 と 1 のシーケンス）（例えば、復号される 1 つまたは複数のシンボルが特定のシンボル値を含む確率の推定）の統計を表し、算術デコーダは、結合状態変数値 ( $s_k$ )（例えば、状態変数値の加重和であり得る）、複数の個々の状態変数値 ( $s_i^k$ ) の基礎となるものを導出するように構成され、算術デコーダは、結合状態変数値 ( $s_k$ )、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k_2 * a^k_2]$ ) を、ルックアップテーブルを使用してマッピングして、復号される 1 つまたは複数のシンボルの算術復号の間隔サイズを説明する間隔サイズ情報（例えば、 $p_k$  または  $R * p_k$ ）を取得するように構成される。

好ましい実施形態では、算術デコーダは、結合状態変数値を取得するために、状態変数値の加重和を決定するように構成される。

#### 【0154】

好ましい実施形態では、算術デコーダは、状態変数値 ( $s_i^k$ ) および関連する重み付け値 ( $d_i^k$ )。

）および関連する重み付け値 ( $d_i^k$ ) ）および関連する重み付け値 ( $d_i^k$ )。

結合状態変数値 ( $s_k$ ) を得るため）の積を丸めることによって得られる丸められた値 ( $[s_i^k * d_i^k]$ )

）の合計を決定するように構成される。

#### 【0155】

好ましい実施形態では、算術デコーダは、

$$s_k = \sum_{i=1}^N [s_i^k * d_i^k]$$

$s^k_2$  は状態変数値であり、 $N$  は考慮される状態変数値の数であり、 $[.]$  はフロア演算子であり、 $d^k_i$  は状態変数値に関連付けられた重み付け値（例えば、結合状態変数値への個々の状態変数値の影響を制御する重み付け係数）（ $d^k_i$  は好ましくはファン 2 の整数値の効力であり、2つの異なる  $d^k_i$  間の比率は好ましくは 2 の整数値の効力である）[2つの異なる  $d^k_i$  間の比率は、好ましくは 8 以上] であることに従って、結合状態変数値  $s_k$  を決定するように構成される。

#### 【0156】

好ましい実施形態では、算術デコーダは、復号されるシンボルが第 1 の値（例えば「1」）をとる場合に状態変数値を第 1 の方向に変更する（例えばより正になる）ように構成され、また復号されるシンボルが第 1 の値とは異なる第 2 の値（例えば「0」）をとる場合（例えば、状態変数値が正と負の値をとることができるように）に状態変数値を第 2 の方向に（例えば、より負になるもの）変更するように構成され、算術デコーダは、結合状態変数値の絶対値 ( $s_k^1 > 0$  の場合は  $s_k$ 、それ以外の場合は  $-s_k$ ) に応じて（例えば、結合状態変数値の絶対値のスケーリングおよび丸められたバージョンに応じて）評価されるルックアップテーブルのエントリを決定するように構成される。

#### 【0157】

好ましい実施形態では、算術デコーダは、結合状態変数値が第 1 の符号（例えば、正の符号）をとる場合に、確率値 ( $p^k$ ) をルックアップテーブルによって提供される値（例えば、

$$LUT2[[s_k * a_k]]$$

10

20

30

40

50

) に設定するように構成され、算術デコーダは、状態変数値が第 2 の符号 ( 例えば、負の符号 ) をとる場合、確率値 (  $p^k$  ) を、所定の値 ( 例えば、 1 ) からルックアップテーブルによって提供される値 ( 例えば、

$LUT2[[-s_k \cdot a_k]]$

) を減算することによって得られる値に設定するように構成される。

【 0 1 5 8 】

好ましい実施形態では、算術デコーダは、

$$p_k = \begin{cases} LUT2[[-s_k \cdot a_k]], & \text{if } s_k \geq 0, \\ 1 - LUT2[[-s_k \cdot a_k]], & \text{else.} \end{cases}$$

10

LUT2 は確率値を含むルックアップテーブルであり、 [ . ] はフロア演算子であり、  $s^k$  は結合変数値であり、  $a^k$  は、結合状態変数値に関連付けられた重み付け値である ( 例えば、  $i$  番目の状態変数値の数の範囲をルックアップテーブルのいくつかのエントリに適合させる重み付け値 ) ことにより、結合確率値  $p_k$  を決定するように構成される。

【 0 1 5 9 】

好ましい実施形態では、算術デコーダは、

$$p_k = \begin{cases} LUT2[[-s_k \cdot a_k]], & \text{if } |s_k \cdot a_k| \geq 0, \\ 1 - LUT2[[-s_k \cdot a_k]], & \text{else} \end{cases}$$

LUT2 は確率値を含むルックアップテーブルであり、 [ . ] はフロア演算子であり、  $s^k$  は結合変数値であり、  $a^k$  は、結合状態変数値に関連付けられた重み付け値である ( 例えば、  $i$  番目の状態変数値の数の範囲をルックアップテーブルのいくつかのエントリに適合させる重み付け値 ) ことにより、結合確率値  $p_k$  を決定するように構成される。

20

【 0 1 6 0 】

好ましい実施形態では、算術デコーダは、結合状態変数値、またはそのスケーリングおよび / または丸められたバージョン (  $[s^k * a^k]$  ) を、 2 次元ルックアップテーブルを使用してサブインターバル幅値 (  $R * p^k$  ) にマッピングするように構成され、そのエントリは、結合状態変数値に依存して ( 例えば、第 1 のルックアップテーブルエントリの座標を決定するために ) 、およびシンボルの復号前の算術復号のコーディング間隔のサイズを記述するコーディング間隔サイズ情報 ( 例えば、  $R$  ) に依存して ( 例えば、第 2 のルックアップテーブルエントリの座標を決定するために ) アドレス指定される。

30

【 0 1 6 1 】

算術デコーダの好ましい実施形態では、 2 次元ルックアップテーブルは、そのエントリが結合状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび / または丸められたバージョン (  $[s^k * a^k]$  ;

$[-s_k \cdot a_k]$

) を含む第 1 の 1 次元ベクトル (

$LUT4[...]$

、 1 次元ルックアップテーブルを形成する ) 、およびそのエントリがコーディング間隔サイズ情報の量子化レベルを含む第 2 の 1 次元ベクトル (

40

$Q_{T_2}(R)$

) の間の 2 項積として、表すことができる。

【 0 1 6 2 】

算術デコーダの好ましい実施形態では、 2 次元ルックアップテーブル ( R a n g T a b L P S ) の要素が、ベースルックアップテーブル ( B a s e T a b L P S ) に基づいて定義され、 2 次元ルックアップテーブルの要素の第 1 のグループ ( またはブロック、例えば、「上半分」 ) は、ベースルックアップテーブルの要素と同一であるか、ベースルックアップテーブルの要素の丸められたバージョンであり、 2 次元ルックアップテーブルの要

50

素の第 2 のグループ（またはブロック、例えば「下半分」）は、ベースルックアップテーブルの要素のスケーリングおよび丸められたバージョンである。

#### 【 0 1 6 3 】

算術デコーダの好ましい実施形態では、2次元ルックアップテーブルの要素の第 2 のグループはベースルックアップテーブルの要素の右シフトバージョンである。

#### 【 0 1 6 4 】

算術デコーダの好ましい実施形態では、確率指数 ( $Q p_2(p_{LPS}; i)$ ) は、2次元ルックアップテーブルの要素の第 1 のグループの要素または2次元ルックアップテーブルの要素の第 2 のグループの要素のどちらを評価するかを決定し、確率指数（例えば、確率値、例えば  $p_{LPS}$  を定量化することによって取得される）の第 1 の範囲（例えば、 $0 \sim \mu - 1$ ）は、要素の第 1 のグループの要素に関連付けられ、確率指数（例えば、確率値、例えば  $p_{LPS}$  を、例えば、量子化関数  $Q p_2(\cdot)$  を使用して定量化することによって取得される）の第 2 の範囲（例えば、 $\mu$  以上）は、要素の第 2 のグループの要素に関連付けられている。10

#### 【 0 1 6 5 】

算術デコーダの好ましい実施形態では、確率指数 ( $i$ ) と第 1 のサイズ値（例えば、 $\mu$ 、サイズ値は例えば、第 1 の方向のベースルックアップテーブルの拡張を説明する）および間隔サイズ指数（例えば量子化演算  $Q r_2(\cdot)$ 、例えば  $j$  を使用するなどして間隔サイズ情報  $R$  に基づいて取得できる）との間の除算の除算残差 ( $i \% \mu$ ) は、2次元ルックアップテーブルの要素を取得するためにベースルックアップテーブルのどの要素を使用するかを決定する。20

#### 【 0 1 6 6 】

好ましい実施形態では、算術デコーダは、

$$RangeTablePS[i][j] = Scale(BaseTablePS[i \% \mu][j], [i / \mu])$$

$BaseTablePS$  は、次元  $\mu \times$  のベースルックアップテーブルであり、 $i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は、間隔サイズ情報に関連付けられたテーブルインデックスであり（例えば、現在のコーディング間隔サイズを記述する）、% は除算残差演算であり、/ は除算演算であり、 $Scale(x, y)$  はスケーリング関数であり（例えば、 $Scale(x, y) = [x * a - b y]$  として定義され、[.] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、または  $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右側にシフトするかどうか、および何ビットシフトするかを決定する、30

ことに従って、2次元ルックアップテーブル ( $RangeTablePS$ ) の要素を取得するように構成される。

#### 【 0 1 6 7 】

算術デコーダの好ましい実施形態では、2次元ルックアップテーブル ( $RangeTablePS$ ) の要素は、確率テーブル ( $probTablePS$ ) に基づいて定義され、確率テーブルは、複数の確率値のセット（例えば、指数  $i$  によって表される）および所与のコーディング間隔サイズの間隔サイズを記述し、複数の確率値のセットにない確率値および/または所与の（参照）コーディング間隔サイズとは異なるコーディング間隔サイズの2次元ルックアップテーブルの要素は、スケーリングを使用して確率テーブルから導出される。40

#### 【 0 1 6 8 】

算術デコーダの好ましい実施形態では、2次元ルックアップテーブルの要素が、コーディング間隔サイズ ( $R$ ) に応じて、確率テーブルの選択された要素 ( $probTablePS[i \% \mu]$ ) の（乗法の）第 1 のスケーリングを使用し、現在の確率値（指数  $i$  により指定）に関連付けられた要素が確率値のセットに含まれるかどうかに応じて（例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて）、第 1 のスケーリングの結果の第 2 のスケーリングを使用して取得される。

#### 【 0 1 6 9 】

10

20

30

40

50

算術デコーダの好ましい実施形態では、確率指数（例えば  $i$ 、例えば、現在の確率値を表す）、および第 1 のサイズ値（例えば  $\mu$ 、例えばサイズ値が確率テーブルの拡張について説明する）の間の除算の除算残差（ $i \% \mu$ ）は、確率テーブルのどの要素が第 1 のスケーリングでスケーリングされるかを決定する、および / または確率指数（ $i$ ）および第 1 のサイズ値間の除算の整数除算結果（ $[i / \mu]$ ）は、第 2 のスケーリングで使用されるスケーリング係数（ $2^{-[i / \mu]}$ ）を決定する、および / またはコーディング間隔サイズは、第 1 のスケーリングの乗法スケーリング係数（ $Q_{r2}(R)$ ）を決定する。

#### 【0170】

好ましい実施形態では、算術デコーダは、

$$RangeTabLPS[i][j] = Scal([probTabLPS[i \% \mu] \cdot Qr_2(R)], [i / \mu])$$

10

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は間隔サイズ情報に関連付けられたテーブルインデックスであり、% は除算残差演算であり、/ は除算演算であり、probTabLPS [ ] は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり（ $I$  の値の範囲は通常  $\mu$  よりも大きくなる）、 $R$  は間隔サイズ（または現在のコーディング間隔サイズ）であり、 $Q_{r2}(R)$  は  $R$  に依存するスケーリング係数であり、Scal（ $x, y$ ）はスケーリング関数であり（例えば、 $Scal(x, y) = [x * a^b y]$ ）として定義され、[ . ] はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

20

ことに従って

2 次元ルックアップテーブルの要素

$$RangeTabLPS[i][j]$$

を取得するように構成される。

#### 【0171】

算術デコーダの好ましい実施形態では、現在の確率値（指数  $i$  により指定）に関連付けられた要素が確率値に含まれるかどうかに応じて（例えば、現在の確率値は、確率テーブルでカバーされている確率値の範囲内にあるかどうかに応じて）、確率テーブルの選択された要素（probTabLPS [  $i \% \mu$  ]）の（乗法の）第 1 のスケーリングを使用し、また、コーディング間隔サイズ（ $R$ ）に応じて、第 1 のスケーリングの結果の第 2 のスケーリングを使用して、2 次元ルックアップテーブルの要素を取得するように構成される。

30

#### 【0172】

算術デコーダの好ましい実施形態では、確率指数（例えば  $i$ 、例えば、現在の確率値を表す）、および第 1 のサイズ値（例えば  $\mu$ 、例えばサイズ値が確率テーブルの拡張について説明する）の間の除算の除算残差（ $i \% \mu$ ）は、確率テーブルのどの要素が第 1 のスケーリングでスケーリングされるかを決定する、および / または確率指数（ $i$ ）および第 1 のサイズ値間の除算の整数除算結果（ $[i / \mu]$ ）は、第 1 のスケーリングで使用されるスケーリング係数（ $2^{-[i / \mu]}$ ）を決定する、および / またはコーディング間隔サイズ（ $R$ ）は、第 2 のスケーリングの乗法スケーリング係数（ $Q_{r2}(R)$ ）を決定する。

40

#### 【0173】

好ましい実施形態では、算術デコーダは、

$$RangeTabLPS[i][j] = [Scal(probTabLPS[i \% \mu], [i / \mu]) \cdot Qr_2(R)]$$

$i$  は、確率情報に関連付けられたテーブルインデックスであり、 $j$  は間隔サイズ情報に関連付けられたテーブルインデックスであり、% は除算残差演算であり、/ は除算演算であり（例えば、整数の結果を提供する）、probTabLPS [ ] は確率テーブルであり、 $\mu$  は確率テーブルの要素の数であり（ $I$  の値の範囲は通常  $\mu$  よりも大きくなる）、 $R$  は間隔サイズであり、 $Q_{r2}(R)$  は  $R$  に依存するスケーリング係数であり、Scal

50

$(x, y)$  はスケーリング関数であり（例えば、 $\text{S c a l}(x, y) = [x * a - b y]$ ）として定義され、 $[.]$  はフロア演算であり、 $a$  は、好ましくは 2 以上の定数であり、 $b$  は、好ましくは 1 以上の定数であり、スケーリング関数は、好ましくは、右シフトビットシフト演算を使用して実装され、 $y$  は、 $x$  の右シフトをするかどうか、および何ビットシフトするかを決定する、

ことに従って、2 次元ルックアップテーブルの要素

$\text{RangeTabLPS}[i][j]$

を取得するように構成される。

【0174】

好ましい実施形態では、算術デコーダは、結合状態変数値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ ) からサブインターバル幅値 ( $R * p^k$ ) を計算するよう構成され、このことは、結合状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]; [s_k \cdot a_k]$ )

) を含む 1 次元ルックアップテーブル (LUT4) エントリを使用して、結合状態変数値 ( $s_k$ )、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]; [s_k \cdot a_k]$ )

) を結合確率値にマッピングすること、およびシンボルを量子化レベルに符号化する前に、算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、 $R$ ）を量子化すること、結合確率値と量子化レベルの間の積を決定することによる。

【0175】

好ましい実施形態では、算術デコーダは、コーディング間隔サイズ情報に論理右シフトを適用することによってコーディング間隔サイズ情報の量子化を実行するよう構成される。

【0176】

好ましい実施形態では、算術デコーダは、コーディング間隔サイズ情報

$R$

30

の量子化を

$Qr_2(R) = ([R \cdot 2^{-u}] + v) \cdot 2^{-w}$

により実行するように構成され、式中、

$u$

$v$

および

$w$

はパラメータである。

【0177】

算術デコーダの好ましい実施形態では、1 次元ルックアップテーブルのエントリは、結合状態変数値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ ) の増加で単調に減少する。

【0178】

算術デコーダの好ましい実施形態では、結合状態変数値の値ドメインの異なる値間隔、

40

50

またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ ) は、等しいサイズである。

算術デコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、結合状態変数値、またはそのスケーリングおよび / または丸められたバージョン ( $[s^k * a^k]$ ) の増加で減少率を伴って単調に減少する。

【0179】

算術デコーダの好ましい実施形態では、ルックアップテーブルは、指数関数的減衰（例えば、0.5からの低下）を（例えば、+ / - 10% または + / - 20% の許容範囲内で）定義する。

【0180】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 1.} \\ s_i^k - [A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 0.} \end{cases}$$

$z$  は所定の（一定の）オフセット値であり、  
 $m_i^k$

は 1 つまたは複数の重み付け値であり、  
 $n_i^k$

は 1 つ以上の重み付け値であり、 $A$  は

$$A[z + \hat{s}] = offset + \sum_{i=1}^N \max(0, \hat{b}_i - \hat{s}) \ll \hat{a}_i$$

であるか、ゼロ設定または大きさの縮小によって、引数の 1 つ以上の極値に対してのみ、それから逸脱して、更新された

$s_i^k$

を所定の値の範囲のままにすることを回避する（例えば、  
 $s_i^k$

は

$s_i^k \cdot m_i^k$

より大きな値ドメインを持つ；つまり、  
 $s_i^k$

は

$s_i^k \cdot m_i^k$

に準量子化される；  
 $s_i^k \cdot m_i^k$

の極値の場合、  
 $s_i^k$

は上記の式に従って未修正の  $A$  によって変更され、その値ドメインを離れることができる：これを回避するために、これらの極値に対応するエントリが削減またはゼロにされる場合があることを考慮されたい）、  
 $offset$

10

20

30

40

50

、  
 $\hat{N}$

、  
 $\hat{b}_i$

、 および  
 $\hat{a}_i$

は所定のパラメータである（例は上記に記載されている）

10

ことに従って複数の可変状態値

$s_i^k$

を更新するように構成される。。

**【 0 1 8 1 】**

好ましい実施形態では、算術デコーダは、テーブルルックアップまたは計算によって  
 $A[z + \lfloor s_i^k \cdot m_i^k \rfloor]$

を導出するように構成される。

**【 0 1 8 2 】**

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための算術デコーダを作成するものであり、算術デコーダは、1つまたは複数の状態変数値（ $s_1^k, s_2^k$ ）を決定するように構成され、これは、以前に復号された複数のシンボル値（例えば、バイナリ値0および1のシーケンス）の統計（例えば、復号される1つまたは複数のシンボルが特定のシンボル値を含む確率の推定値）（例えば、複数の状態変数値が決定される場合、異なる適合時定数の統計）を表し、また、算術デコーダは、1つまたは複数の状態変数値（ $s_i^k$ ）（例えば、インデックスkにより示される、所与のコンテキストモードに関連付けられている）に基づいて復号される1つまたは複数のシンボル値の算術復号の間隔サイズ情報（ $p^k, R * p^k$ ）を導出するように構成され、これは複数の以前に符号化されたシンボル値（例えば、バイナリ値0および1のシーケンス）の統計（例えば、複数の状態変数値が決定される場合、異なる適合時定数の統計）を表し、算術デコーダは、復号されるシンボルに依存して、ルックアップテーブル（A）を使用して（例えば、復号されるシンボルを復号した後）、第1の状態変数値（ $s_1^k$ ）を更新するように構成される。

20

**【 0 1 8 3 】**

好ましい実施形態では、算術デコーダは、復号されたるシンボルに依存し、ルックアップテーブル（A）を使用して（例えば、復号されるシンボルを復号した後）、第2の状態変数値（ $s_2^k$ ）を更新するように構成される。

**【 0 1 8 4 】**

好ましい実施形態では、算術デコーダは、異なる適合時定数を使用して、第1の状態変数値および第2の状態変数値を更新するように構成される。

40

**【 0 1 8 5 】**

好ましい実施形態では、算術デコーダは、復号されるシンボルが第1の値をとるか、第1の値とは異なる第2の値をとるかに応じて、ルックアップテーブルを使用して決定された値によって以前の状態変数値を選択的に増減するように構成される。

**【 0 1 8 6 】**

好ましい実施形態では、算術デコーダは、以前の状態変数値を、復号されるシンボルが第1の値をとる場合に、以前の状態変数値が正である場合と比較して、以前の状態変数値が負である場合に、比較的大きい値だけ増加させるように構成され、算術デコーダは、以前の状態変数値を、復号されるシンボルが、第1の値とは異なる第2の値をとる場合に、

50

以前の状態変数値が負である場合と比較して、以前の状態変数値が正である場合に、比較的大きい値だけ減少させるように構成される（それは、例えば、ルックアップテーブルを適切に選択することで到達する）。

【0187】

好ましい実施形態では、算術デコーダは、復号されるシンボルが第1の値をとる場合、所定（例えば、固定の）のオフセット値（ $z$ ）と以前に計算された第1の状態変数値（ $s_1^k$ ）

）、またはそのスケーリングおよび／または丸められたバージョン（ $\lfloor s_1^k \cdot m_1^k \rfloor$ ）

10

）の合計に依存して、第1の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するように構成され、算術デコーダは、復号されるシンボルが第2の値をとる場合、所定（例えば、固定の）のオフセット値（ $z$ ）と以前に計算された第1の状態変数値の逆（-1を乗算する）バージョン（ $-s_1^k$ ）

）、またはそのスケーリングおよび／または丸められたバージョン（ $\lfloor -s_1^k \cdot m_1^k \rfloor$ ）

20

）（例えば、以前に計算された第1の状態変数値の逆バージョン）の合計に依存して、第1の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するように構成される。

【0188】

好ましい実施形態では、算術デコーダは、復号されるシンボルが第1の値をとる場合、所定（例えば、固定の）のオフセット値（ $z$ ）と以前に計算された第2の状態変数値（ $s_2^k$ ）

）、またはそのスケーリングおよび／または丸められたバージョン（ $\lfloor s_2^k \cdot m_2^k \rfloor$ ）

30

）の合計に依存して、第2の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するように構成され、算術デコーダは、復号されるシンボルが第2の値をとる場合、所定（例えば、固定の）のオフセット値（ $z$ ）と以前に計算された第2の状態変数値の逆（-1を乗算する）バージョン（ $-s_2^k$ ）

）、またはそのスケーリングおよび／または丸められたバージョン（ $\lfloor -s_2^k \cdot m_2^k \rfloor$ ）

40

）（例えば、以前に計算された第2の状態変数値の逆バージョン）の合計に依存して、第2の状態変数値を更新するときに評価されるルックアップテーブルのエントリのインデックスを決定するように構成される。

【0189】

好ましい実施形態では、算術デコーダは、第1の状態変数値の更新時に評価されるルックアップテーブルのエントリのインデックスを決定するときに、第1のスケーリング値（ $m^k_1$ ）を適用して、以前に計算された第1の状態変数値（ $s^k_1$ ）をスケーリングするように構成され、算術デコーダは、第2の状態変数値の更新時に評価されるルックアップテーブルのエントリのインデックスを決定するときに、第2のスケーリング値（ $m^k_2$ ）を適用して、以前に計算された第2の状態変数値（ $s^k_2$ ）をスケーリングするように構成

50

され、第1のスケーリング値が第2のスケーリング値とは異なる（また、第1および第2のスケーリング値は好ましくは2の整数効力であり、第1のスケーリング値と第2のスケーリング値との間の比は、2の場合、好ましくは整数効力であり、第1のスケーリング値および第2のスケーリング値は、好ましくは少なくとも8倍異なる）。

#### 【0190】

好ましい実施形態では、算術デコーダは、第1の状態変数値を更新するときに第1のスケーリング値（例えば、 $n^k_1$ ）を使用してルックアップテーブルの評価によって返される値をスケーリングするように構成され、

算術デコーダは、第2の状態変数値を更新するときに第2のスケーリング値（例えば、 $n^k_2$ ）を使用してルックアップテーブルの評価によって返される値をスケーリングするように構成され、第1のスケーリング値は第2のスケーリング値とは異なる。

#### 【0191】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If decoded symbol is 1.} \\ s_i^k - [A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If decoded symbol is 0.} \end{cases}$$

$A$ はルックアップテーブルであり（例えば、整数の値を含む）、 $z$ は所定の（一定の）オフセット値であり、

$$m_i^k$$

10

は1つまたは複数の重み付け値であり、

$$n_i^k$$

20

は1つまたは複数の重み付け値であることに従って、1つまたは複数の更新された状態変数値

$$s_i^k$$

を決定するように構成される。

#### 【0192】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 1.} \\ s_i^k - [A[z - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 0.} \end{cases}$$

30

$A$ はルックアップテーブルであり（例えば、整数の値を含む）、 $z$ は所定の（一定の）オフセット値であり、

$$m_i^k$$

は1つまたは複数の重み付け値であり、

$$n_i^k$$

40

は1つまたは複数の重み付け値であることに従って、1つまたは複数の更新された状態変数値

$$s_i^k$$

を決定するように構成される。

#### 【0193】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 1.} \\ s_i^k + [-A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 0.} \end{cases}$$

50

A はルックアップテーブルであり（例えば、整数の値を含む）、z は所定の（一定の）オフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

10

を決定するように構成される。

【0 1 9 4】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

A はルックアップテーブルであり（例えば、整数の値を含む）、z は所定の（一定の）オフセット値であり、

$m_i^k$

20

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

50

を決定するように構成される。

【0 1 9 5】

算術デコーダの好ましい実施形態では、A のエントリは、ルックアップテーブルインデックスの増加とともに単調に減少する。

【0 1 9 6】

好ましい実施形態では、算術デコーダは、A は

$$A[z + \hat{s}] = offset + \sum_{i=1}^N \max(0, \hat{b}_i - \hat{s}) << \hat{a}_i$$

であるか、ゼロ設定または大きさの縮小によって、引数の 1 つ以上の極値に対してのみ、それから逸脱して、更新された

$s_i^k$

40

を所定の値の範囲のままにすることを回避する（例えば、

$s_i^k$

は

$s_i^k \cdot m_i^k$

より大きな値ドメインを持つ；つまり、

$s_i^k$

50

は

$s_i^k \cdot m_i^k$ 

に準量子化される；

 $s_i^k \cdot m_i^k$ 

の極値の場合、

 $s_i^k$ は上記の式に従って未修正の A によって変更され、その値ドメインを離れることができる  
：これを回避するために、これらの極値に対応するエントリが削減またはゼロにされる場  
合があることを考慮されたい）、 $offset$ 

10

 $\hat{N}$  $\hat{b}_i$ 

、および

 $\hat{a}_i$ 

20

は所定のパラメータである（例は上記に記載されている）。

## 【0197】

算術デコーダの好ましい実施形態では、ルックアップテーブルの最後のエントリ（これは、第 1 の状態変数値が最大許容値まで及ぶ所定の値の範囲に達したとき、または第 1 の状態変数値が所定のしきい値を超えたときにアドレス指定される）はゼロに等しい。

## 【0198】

好ましい実施形態では、算術デコーダは、更新された状態変数値にクリッピング操作を適用して、更新されクリップされた状態変数値を所定の値の範囲内に保つように構成される。

## 【0199】

好ましい実施形態では、算術デコーダは、更新された状態変数値に、

 $s_i^k = \min(\max(s_i^k, h_i^k), g_i^k)$ 

30

、に従ってクリッピング操作を適用するように構成され、

 $g_i^k$ 

は、

 $s_i^k$ 

40

の最大許容値であり、

 $h_i^k$ 

は、

 $s_i^k$ 

の最小許容値である。

## 【0200】

好ましい実施形態では、算術デコーダは、異なるコンテキストモデルに異なるスケーリ

50

ング値を適用するように構成される（例えば、スケーリング値の少なくとも 1 つが 2 つの異なるコンテキストモデル間で異なるようにする）。

【 0 2 0 1 】

好ましい実施形態では、算術デコーダは、上記の実施形態の 1 つで定義された間隔サイズ情報を取得するように構成される。

【 0 2 0 2 】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを復号するための算術デコーダを作成するものであり、算術デコーダは、以前に復号された複数のシンボル値の統計を表す 1 つの状態変数値（ $s_k$ ）を決定するように構成され、算術デコーダは、結合状態変数値、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ）から、復号されるシンボル値の算術復号のためのサブインターバル幅値（ $R_{LPS}$ ）

）を計算するよう構成され、この計算は、そのエントリが結合状態変数値の値ドメインの異なる値間隔の確率値、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ；  
[[ $s_k$ ] ·  $a_k$ ]

）を含む 1 次元ルックアップテーブル（  
probTabLPS[ $Qp_2(\dots)$ ]

）を使用して、1 つの状態変数値（ $s_k$ ）、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ；  
[[ $s_k$ ] ·  $a_k$ ]

）を結合確率値にマッピングすること、および量子化レベル（  
 $Qr_2(R)$

）に符号化されるシンボル値の算術復号の前に、算術符号化のコーディング間隔のサイズを記述するコーディング間隔サイズ情報（例えば、 $R$ ）を量子化することにより、これを行い、算術デコーダは、さらに、確率値と、量子化レベルの間の積を決定する（事前に計算された積のルックアップまたは乗算によって）ように構成され、算術デコーダは、（実際に復号される）復号されるシンボルに依存して状態変数値の更新を実行するように構成される。

【 0 2 0 3 】

好ましい実施形態では、算術デコーダは、1 つの状態変数値（ $s_k$ ）として、結合状態変数値（例えば、状態変数値の加重和であり得る）、複数の状態変数値（ $s_{i^k}$ ）（例えば、バイナリ値 0 および 1 のシーケンス）（例えば、複数の状態変数値の場合、異なる適合時定数の統計）の基礎となるものを導出するように構成され、これは、異なる適合時定数の複数の以前に復号されたシンボル値（例えば、バイナリ値 0 および 1 のシーケンス）の統計を表す。

【 0 2 0 4 】

好ましい実施形態では、算術デコーダは、結合状態変数値を取得するために、状態変数値の加重和を決定するように構成される。

【 0 2 0 5 】

好ましい実施形態では、算術デコーダは、状態変数値（ $s_i^k$ ）および関連する重み付け値（ $w_i^k$ ）

）および関連する重み付け値（ $w_i^k$ ）

10

20

30

40

50

$d_i^k)$

結合状態変数値 ( $s_k$ ) を得るため) の積を丸めることによって得られる丸められた値 ( $\lfloor s_i^k \cdot d_i^k \rfloor$ )

) の合計を決定するように構成される。

【0206】

好ましい実施形態では、算術デコーダは、

$$s_k = \sum_{i=1}^N \lfloor s_i^k \cdot d_i^k \rfloor$$

10

$s_k$  は状態変数値であり、 $N$  は考慮される状態変数値の数であり、 $\lfloor \cdot \rfloor$  はフロア演算子であり、 $d_i^k$  は状態変数値に関連付けられた重み付け値 (例えば、結合状態変数値への個々の状態変数値の影響を制御する重み付け係数) ( $d_i^k$  は好ましくはファン 2 の整数値の効力であり、2 つの異なる  $d_i^k$  間の比率は好ましくは 2 の整数値の効力である) [2 つの異なる  $d_i^k$  間の比率は、好ましくは 8 以上) であることに従って、結合状態変数値  $s_k$  を決定するよう構成される。

【0207】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 1.} \\ s_i^k - [A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 0.} \end{cases}$$

20

$z$  は所定の (一定の) オフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つ以上の重み付け値であり、 $A$  は

$$A[z + \hat{s}] = offset + \sum_{i=1}^N \max(0, \hat{b}_i - \hat{s}) \ll \hat{a}_i$$

30

であるか、ゼロ設定または大きさの縮小によって、引数の 1 つ以上の極値に対してのみ、それから逸脱して、更新された

$s_i^k$

を所定の値の範囲のままにすることを回避する (例えば、

$s_i^k$

は

$s_i^k \cdot m_i^k$

40

より大きな値ドメインを持つ ; つまり、

$s_i^k$

は

$s_i^k \cdot m_i^k$

に準量子化される ;

$s_i^k \cdot m_i^k$

50

の極値の場合、

$s_i^k$

は上記の式に従って未修正の  $A$  によって変更され、その値ドメインを離れることができる  
：これを回避するために、これらの極値に対応するエントリが削減またはゼロにされる場合があることを考慮されたい）、

$offset$

$\hat{N}$

10

$\hat{b}_i$

、および

$\hat{a}_i$

は所定のパラメータである（例は上記に記載されている）

ことに従って、状態変数値の更新を行う際、複数の状態変数値

$s_i^k$

20

を更新するように構成される。

【0208】

好ましい実施形態では、算術デコーダは、テーブルルックアップまたは計算によって  
 $A[z + \lfloor s_i^k \cdot m_i^k \rfloor]$

を導出するように構成される。

【0209】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k - \lfloor A[z - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

30

$A$  はルックアップテーブルであり（例えば、整数の値を含む）、 $z$  は所定の（一定の）  
オフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変  
数値

$s_i^k$

40

を決定するように構成される。

【0210】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

$A$  はルックアップテーブルであり（例えば、整数の値を含む）、 $z$  は所定の（一定の）

50

オフセット値であり、

$m_i^k$

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

10

を決定するように構成される。

【 0 2 1 1 】

好ましい実施形態では、算術デコーダは、

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 0.} \end{cases}$$

$A$  はルックアップテーブルであり（例えば、整数の値を含む）、 $z$  は所定の（一定の）オフセット値であり、

$m_i^k$

20

は 1 つまたは複数の重み付け値であり、

$n_i^k$

は 1 つまたは複数の重み付け値であることに従って、1 つまたは複数の更新された状態変数値

$s_i^k$

を決定するように構成される。

【 0 2 1 2 】

好ましい実施形態では、算術デコーダは、コーディング間隔サイズ情報に論理右シフトを適用することによってコーディング間隔サイズ情報の量子化を実行するように構成される。

30

【 0 2 1 3 】

好ましい実施形態では、算術デコーダは、コーディング間隔サイズ情報

$R$

の量子化を

$$Q_{12}(R) = (\lfloor R \cdot 2^{-u} \rfloor + v) \cdot 2^{-w}$$

により実行するように構成され、式中、

$u$

40

$v$

$w$

はパラメータである。

【 0 2 1 4 】

50

算術デコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、1つの状態変数値、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ）の増加で単調に減少する。

【0215】

算術デコーダの好ましい実施形態では、1つの状態変数値の値ドメインの異なる値間隔、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ）は、等しいサイズである。

【0216】

算術デコーダの好ましい実施形態では、1次元ルックアップテーブルのエントリは、1つの状態変数値、またはそのスケーリングおよび／または丸められたバージョン（ $[s^k * a^k]$ ）の増加で減少率を伴って単調に減少する。

10

【0217】

本発明による実施形態は、ビデオエンコーダを作成するものであり、ビデオエンコーダは、複数のビデオフレームを符号化するように構成され、ビデオエンコーダは、上記の実施形態のいずれかに記載の、ビデオコンテンツを表すバイナリ値のシーケンスに基づいて符号化されたバイナリシーケンスを提供するための算術エンコーダを備える。

【0218】

本明細書で論じられる算術エンコーダは、ビデオエンコーダ内部での使用に非常に適していることに留意されたい。この場合、符号化されるシンボルおよび／または以前に符号化されたシンボルは、ビデオビットストリームのシンボルであり得る。例えば、符号化されるシンボルおよび／または以前に符号化されたシンボルは、ビデオコンテンツを表すサイド情報または制御情報のビットおよび／または符号化された変換係数のビットを表すことができる。言い換えれば、符号化されるシンボルおよび／または以前に符号化されたシンボルは、ビデオコンテンツを表すためにビットストリームに含まれる情報のいずれかを表すことができる。ただし、状態変数値は、様々な「コンテキスト」、つまり様々なタイプの情報に対して個別に決定される場合があることに留意されたい。例えば、特定のタイプの情報（例えば、特定のタイプのサイド情報）に関連付けられたビットのみが、特定の結合状態変数値を取得するために使用される特定の状態変数値、または特定の状態変数値のセットに寄与することができる。したがって、間隔サイズ情報はまた、異なるコンテキストに対して、すなわち、異なるタイプの情報（例えば、サイド情報）に関連付けられたシンボル値の符号化のために、個別に導出され得る。

20

【0219】

本発明による実施形態は、ビデオデコーダを作成するものであって、ビデオエンコーダは、複数のビデオフレームを復合するように構成され、ビデオデコーダは、上記の実施形態のいずれかに記載の、バイナリシーケンスの符号化表現（211）に基づいて復号されたバイナリシーケンス（例えば、復号されたシンボル値に基づく）を提供するための算術デコーダ（120；220）を備える。

30

ビデオデコーダは、ビデオエンコーダと同じ考慮事項に基づいている。したがって、上記の説明もまた、当てはまるものであり、符号化されたシンボルまたは符号化されるシンボルは、復号されたシンボルに対応する。

40

【0220】

さらに、本発明によるさらなる実施形態は、それぞれの方法およびコンピュータプログラムを作成することに留意されたい。

【0221】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための方法を作成し、方法は、間隔サイズ情報（ $p^k, R * p^k$ ）を、1つまたは複数のシンボル値（例えば、バイナリ値）の算術符号化が複数の状態変数値（ $s_i^k$ ）（例えば、インデックス  $k$  によって表示される所与のコンテキストモードに関連付けられるもの）に基づいて符号化されるように導出することを含み、これは、異なる適合時定数で以前に符号化された複数のシンボル値（例えば、バイナリ値0と1のシーケンス）の

50

統計を表し、方法は、ルックアップテーブル（LUT1）を使用して、第1の状態変数値（ $s^{k_1}$ ）、またはそのスケーリングおよび／または丸められたバージョン（ $[s^{k_1} * a^{k_1}]$ ）をマッピングし、ルックアップテーブル（LUT1）を使用して、第2の状態変数値（ $s^{k_2}$ ）、またはそのスケーリングおよび／または丸められたバージョン（ $[s^{k_2} * a^{k_2}]$ ）をマッピングすることを含み、符号化される1つまたは複数のシンボルの算術符号化のための間隔サイズを説明する間隔サイズ情報（例えば、 $p_k$ または $R * p_k$ ）を取得するようにする。

#### 【0222】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための方法を作成し、方法は、間隔サイズ情報（ $p^k$ 、 $R * p^k$ ）を、1つまたは複数のシンボル値（例えば、バイナリ値）の算術符号化が複数の状態変数値（ $s_i^k$ ）（例えば、インデックス $k$ によって表示される所与のコンテキストモードに関連付けられるもの）に基づいて符号化されるように導出することを含み、これは、異なる適合時定数で以前に符号化された複数のシンボル値（例えば、バイナリ値0と1のシーケンス）の統計を表し、方法は、結合状態変数値（ $s_k$ ）（例えば、状態変数値の加重和であり得る）、複数の個々の状態変数値（ $s_i^k$ ）の基礎となるものを導出するように構成され、算術デコーダは、結合状態変数値（ $s_k$ ）、またはそのスケーリングおよび／または丸められたバージョン（ $[s^{k_2} * a^{k_2}]$ ）を、ルックアップテーブルを使用してマッピングして、符号化される1つまたは複数のシンボルの算術符号化の間隔サイズを説明する間隔サイズ情報（例えば、 $p_k$ または $R * p_k$ ）を取得することを含む。

#### 【0223】

本発明による実施形態は、シンボル値（二進値）を有する複数のシンボルを符号化するための方法を作成し、この方法は、1つまたは複数の状態変数値（ $s_1^k$ 、 $s_2^k$ ）を決定することを含み、これは、以前に符号化された複数のシンボル値（例えば、バイナリ値0および1のシーケンス）の統計（例えば、複数の状態変数値の場合、異なる適合時定数の統計）を表し、また、方法は、1つまたは複数の状態変数値（ $s_i^k$ ）（例えば、インデックス $k$ により示される、所与のコンテキストモードに関連付けられている）に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ情報（ $p^k$ 、 $R * p^k$ ）を導出することを含み、これは複数の以前に符号化されたシンボル値（例えば、バイナリ値0および1のシーケンス）の統計（例えば、複数の状態変数値の場合、異なる適合時定数の統計）を表し、方法は、符号化されるシンボルに依存して、ルックアップテーブル（A）を使用して（例えば、シンボルを符号化した後）、第1の状態変数値（ $s^{k_1}$ ）を更新することを含む。

#### 【0224】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを復号するための方法を作成し、方法は、間隔サイズ情報（ $p^k$ 、 $R * p^k$ ）を、1つまたは複数のシンボル値（例えば、バイナリ値）の算術復号が複数の状態変数値（ $s_i^k$ ）（例えば、インデックス $k$ によって表示される所与のコンテキストモードに関連付けられるもの）に基づいて復号されるように導出することを含み、これは、異なる適合時定数で以前に復号された複数のシンボル値（例えば、バイナリ値0と1のシーケンス）の統計（例えば、復号される1つまたは複数のシンボルが特定のシンボル値を含む確率の推定）を表し、方法は、ルックアップテーブル（LUT1）を使用して、第1の状態変数値（ $s^{k_1}$ ）、またはそのスケーリングおよび／または丸められたバージョン（ $[s^{k_1} * a^{k_1}]$ ）をマッピングし、ルックアップテーブル（LUT1）を使用して、第2の状態変数値（ $s^{k_2}$ ）、またはそのスケーリングおよび／または丸められたバージョン（ $[s^{k_2} * a^{k_2}]$ ）をマッピングすることを含み、復号される1つまたは複数のシンボルの算術復号のための間隔サイズを説明する間隔サイズ情報（例えば、 $p_k$ または $R * p_k$ ）を取得するようになる。

#### 【0225】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボル

10

20

30

40

50

を復号するための方法を作成し、方法は、間隔サイズ情報 ( $p^k$ 、 $R * p^k$ ) を、1つまたは複数のシンボル値（例えば、バイナリ値）の算術復号が複数の状態変数値 ( $s_i^k$ )（例えば、インデックス  $k$  によって表示される所与のコンテキストモードに関連付けられるもの）に基づいて復号されるように導出することを含み、これは、異なる適合時定数で以前に復号された複数のシンボル値（例えば、バイナリ値0と1のシーケンス）（例えば、復号される1つまたは複数のシンボルが特定のシンボル値を含む確率の推定）の統計を表し、方法は、結合状態変数値 ( $s_k$ )（例えば、状態変数値の加重和であり得る）、複数の個々の状態変数値 ( $s_i^k$ ) の基礎となるものを導出することを含み、方法は、結合状態変数値 ( $s_k$ )、またはそのスケーリングおよび/または丸められたバージョン ( $[s_k^2 * a_k^2]$ ) を、ルックアップテーブルを使用してマッピングして、復号される1つまたは複数のシンボルの算術復号の間隔サイズを説明する間隔サイズ情報（例えば、 $p_k$ または  $R * p_k$ ）を取得することを含む。

#### 【0226】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための方法を作成するものであり、方法は、1つまたは複数の状態変数値 ( $s_1^k$ 、 $s_2^k$ ) を決定することを含み、これは、以前に復号された複数のシンボル値（例えば、バイナリ値0および1のシーケンス）の統計（例えば、復号される1つまたは複数のシンボルが特定のシンボル値を含む確率の推定値）（例えば、複数の状態変数値が決定される場合、異なる適合時定数の統計）を表し、また、方法は、1つまたは複数の状態変数値 ( $s_i^k$ )（例えば、インデックス  $k$  により示される、所与のコンテキストモードに関連付けられている）に基づいて復号される1つまたは複数のシンボル値の算術復号の間隔サイズ情報 ( $p^k$ 、 $R * p^k$ ) を導出することを含み、これは複数の以前に符号化されたシンボル値（例えば、バイナリ値0および1のシーケンス）の統計（例えば、複数の状態変数値が決定される場合、異なる適合時定数の統計）を表し、算術デコーダは、復号されるシンボルに依存して、ルックアップテーブル ( $A$ ) を使用して、第1の状態変数値 ( $s_1^k$ ) を更新することを含む。

#### 【0227】

本発明による実施形態は、上記の実施形態のうちの1つによるエンコーダおよびデコーダによって実行される方法を作成する。

#### 【0228】

本発明による実施形態は、コンピュータプログラムがコンピュータ上で実行されるときに、上記の実施形態のうちの1つによる方法を実行するためのコンピュータプログラムを作成する。

#### 【0229】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための方法を作成するものであり、方法は、1つまたは複数の状態変数値 ( $s_i^k$ ) に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ値 ( $R_{LPS}$ )（それは例えば、インデックス  $k$  で示される所与のコンテキストモードに関連付けられている）を導出することを含み、これは、以前に符号化された複数のシンボル値（例えば、バイナリ値0および1のシーケンス）の統計を表し（例えば、異なる適合時定数を使用）、方法は、ベースルックアップテーブル ( $BaseTablePS$ ) を使用して間隔サイズ値 ( $R_{LPS}$ ) を決定することを含み（その次元は、確率指數の観点から、可能な確率指數  $i$  の数よりも小さい）、方法は、1つまたは複数の状態変数値（例えば、 $i = Q_p (p_{LPS})$  のように）を基にして得られる確率指數 ( $i$ ) が第1の範囲（例えば、 $\mu$ 未満）内にある場合、決定された間隔サイズ値がベースルックアップテーブルの要素と同一であるか、またはベースルックアップテーブルの要素の丸めのバージョンであり、確率指數が第2の範囲（例えば、 $\mu$ 以上）内にある場合は、ベースルックアップテーブルの要素のスケーリングと丸めを使用して、決定された間隔サイズ値が取得されるように、間隔サイズ値 ( $R_{LPS}$ ) を決定することを含み、方法は、間隔サイズ値 ( $R_{LPS}$ ) を使用して1つまたは複数のシンボルの算術符号化を実行することを含む。

10

20

30

40

50

## 【 0 2 3 0 】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを符号化するための方法を作成するものであり、方法は、1つまたは複数の状態変数値（ $s_i^k$ ）（これはインデックス  $k$  によって示される所与のコンテキストモードに関連付けられる）に基づいて符号化される1つまたは複数のシンボル値の算術符号化の間隔サイズ値（ $R_{LPS}$ ）を導出することを含み、これは、以前に符号化された複数のシンボル値の統計を表し（例えば、バイナリ値0と1のシーケンス）（例えば、異なる適合時定数で）；算術デコーダは、1つまたは複数の状態変数値から導出された（現在の）確率値に基づいて、また（現在の）コーディング間隔サイズ（ $R$ ）に基づいて、確率テーブル（ $ProbTable$ ）（確率指數という観点で次元が可能な確率指數  $i$  の数値より小さい）を使用して間隔サイズ値（ $R_{LPS}$ ）を決定することを含み、確率テーブルは、複数の確率値のセット（例えば、0と $\mu - 1$ の間の確率指數に対して）および（单一の）所与の（参照）コーディング間隔サイズの間隔サイズ（間隔サイズ値）を記述し、方法は、現在の確率値が複数の確率値のセットにない（例えば、 $\mu$ 以上の現在の確率値と関連付けられた確率指數である）場合、および/または現在のコーディング間隔サイズ（ $R$ ）が、所与（参照）のコーディング間隔サイズとは異なる場合、間隔サイズ値（ $R_{LPS}$ ）を取得するよう所与（参照）の確率テーブル（ $ProbTable$ ）の要素をスケーリングすることを含み、また、方法は、間隔サイズ値（ $R_{LPS}$ ）を使用して1つまたは複数のシンボルの算術符号化を実行することを含む。

## 【 0 2 3 1 】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを復号するための方法を作成するものであり、方法は、1つまたは複数の状態変数値（ $s_i^k$ ）に基づいて復号される1つまたは複数のシンボル値の算術復号の間隔サイズ値（ $R_{LPS}$ ）（それは例えば、インデックス  $k$  で示される所与のコンテキストモードに関連付けられている）を導出することを含み、これは、以前に復号された複数のシンボル値（例えば、バイナリ値0および1のシーケンス）の統計を表し（例えば、異なる適合時定数を使用）、方法は、ベースルックアップテーブル（ $BaseTable$ ）を使用して間隔サイズ値（ $R_{LPS}$ ）を決定することを含み（その次元は、確率指數の観点から、可能な確率指數  $i$  の数よりも小さい）、方法は、1つまたは複数の状態変数値（例えば、 $i = Qp$ （ $p_{LPS}$ ）のように）を基にして得られる確率指數（ $i$ ）が第1の範囲（例えば、 $\mu$ 未満）内にある場合、決定された間隔サイズ値がベースルックアップテーブルの要素と同一であるか、またはベースルックアップテーブルの要素の丸めのバージョンであり、確率指數が第2の範囲（例えば、 $\mu$ 以上）内にある場合は、ベースルックアップテーブルの要素のスケーリングと丸めを使用して、決定された間隔サイズ値が取得されるように、間隔サイズ値（ $R_{LPS}$ ）を決定することを含み、方法は、間隔サイズ値（ $R_{LPS}$ ）を使用して1つまたは複数のシンボルの算術復号を実行するように構成される。

## 【 0 2 3 2 】

本発明による実施形態は、シンボル値（例えば、バイナリ値）を有する複数のシンボルを復号するための方法を作成するものであり、方法は、1つまたは複数の状態変数値（ $s_i^k$ ）（これはインデックス  $k$  によって示される所与のコンテキストモードに関連付けられる）に基づいて復号される1つまたは複数のシンボル値の算術復号の間隔サイズ値（ $R_{LPS}$ ）を導出することを含み、これは、以前に復号された複数のシンボル値の統計を表し（例えば、バイナリ値0と1のシーケンス）（例えば、異なる適合時定数で）；算術デコーダは、1つまたは複数の状態変数値から導出された（現在の）確率値に基づいて、また（現在の）コーディング間隔サイズ（ $R$ ）に基づいて、確率テーブル（ $ProbTable$ ）（確率指數という観点で次元が可能な確率指數  $i$  の数値より小さい）を使用して間隔サイズ値（ $R_{LPS}$ ）を決定することを含み、確率テーブルは、複数の確率値のセット（例えば、0と $\mu - 1$ の間の確率指數に対して）および（单一の）所与の（参照）コーディング間隔サイズの間隔サイズ（間隔サイズ値）を記述し、方法は、現在の確率値が複数の確率値のセットにない（例えば、 $\mu$ 以上の現在の確率値と関連付けられた確率指數であ

10

20

30

40

50

る)場合、およびノまたは現在のコーディング間隔サイズ( $R$ )が、所与(参照)のコーディング間隔サイズとは異なる場合、間隔サイズ値( $R_{LPS}$ )を取得するよう所与(参照)の確率テーブル( $Prob\_TabLPS$ )の要素をスケーリングすることを含み、また、方法は、間隔サイズ値( $R_{LPS}$ )を使用して1つまたは複数のシンボルの算術復号を実行することを含む。

【0233】

本発明による実施形態は、コンピュータプログラムがコンピュータ上で実行されるときに、上記の実施形態のうちの1つによる方法を実行するためのコンピュータプログラムを作成する。

【0234】

上記の方法は、上記の装置と同じ考慮事項に基づいている。しかし、この方法は、装置に関しても、本明細書に記載の特徴、機能、および詳細のいずれかによって任意選択で補足できることに留意されたい。これらの方法は、任意選択で、該特徴、機能、および詳細によって、個別および組み合わせの両方で、補足することができる。同じことがコンピュータプログラムにも当てはまる。

【0235】

続いて、本発明による実施形態を、以下の付属の図面を参照して説明する。

【図面の簡単な説明】

【0236】

【図1】本発明の実施形態による、データストリーム内の画像を予測的にコーディングするための装置のブロック概略図を示す。

20

【図2】本発明の実施形態による、デコーダのブロック概略図を示す。

【図3】再構成された信号と、予測残差および予測信号の結合との間の関係の概略図を示す。

【図4】本発明の実施形態による、算術エンコーダのブロック概略図を示す。

【図5】本発明の実施形態による、算術デコーダのブロック概略図を示す。

【図6】本発明の実施形態による、間隔サイズ情報を決定するための概念の概略図を示す。

【図7】発明の実施形態による、間隔サイズ情報を決定するための概念の概略図を示す。

【図8】発明の実施形態による、間隔サイズ情報を決定するための概念の概略図を示す。

【図9】発明の実施形態による、間隔サイズ情報を決定するための概念の概略図を示す。

【図10a】本発明の実施形態による、間隔サイズ情報を決定するための概念の概略図を示す。

30

【図10b】本発明の実施形態による、間隔サイズ情報を決定するための概念の概略図を示す。

【図11】発明の実施形態による、間隔サイズ情報を決定するための概念の概略図を示す。

【図12】発明の実施形態による、1つまたは複数の更新された状態変数を決定するための概念の概略図を示す。

【図13】発明の実施形態による、間隔サイズ値を決定するための概念の概略図を示す。

【図14】発明の実施形態による、間隔サイズ値を決定するための概念の概略図を示す。

【図14】発明の実施形態による、間隔サイズ値を決定するための概念の概略図を示す。

【図15】発明の実施形態による、間隔サイズ値を決定するための概念の概略図を示す。

【図16】発明の実施形態による、ビデオデコーダの概略図を示す。

【図17】発明の実施形態による、ビデオデコーダの概略図を示す。

【発明を実施するための形態】

【0237】

1. 図1によるエンコーダ

以下の図の説明は、コーディングフレームワークの例を形成するために、ビデオの画像をコーディングするためのブロックベースの予測コーデックのエンコーダ(図1)とデコーダ(図2)の説明から始まり、これに本発明の実施形態が組み込まれ得る。それぞれのエンコーダとデコーダは、図1から図3に関して説明されている。以下、本発明の概念の

40

50

実施形態の説明は、そのような概念をそれぞれ図1および図2のエンコーダおよびデコーダに組み込むことができる方法に関する説明とともに提示されるが、後続の図4以降で説明される実施形態はまた、図1および図2のエンコーダおよびデコーダの基礎となるコーディングフレームワークに従って動作しないエンコーダおよびデコーダを形成するために使用され得る。

【0238】

本発明による実施形態は、図1および図2に関して説明したように、ビデオエンコーダおよびビデオデコーダを含み得る。また、本明細書に開示される概念のいずれも、例えば、図1および図2を参照して説明されるように、エントロピーコーダ34またはエントロピーデコーダ50で使用され得る。

10

【0239】

図1は、例示的に変換ベースの残差コーディングを使用して、画像12をデータストリーム14に予測的にコーディングするための装置を示す。装置またはエンコーダは、参照符号10を使用して示されている。図2は、対応するデコーダ20、すなわち、変換ベースの残差復号も使用してデータストリーム14から画像12'を予測的に復号するように構成される装置20を示し、アポストロフィは、デコーダ20によって再構成された画像12'が、予測残差信号の量子化によって導入されたコーディング損失に関して、装置10によって元々符号化されている画像12から逸脱していることを示すために使用されている。図1および図2は、変換ベースの予測残差コーディングを例示的に使用するが、本出願の実施形態は、この種の予測残差コーディングに限定されない。これは、以下に概説されるように、図1および図2に関して説明される他の詳細にも当てはまる。

20

【0240】

エンコーダ10は、予測残差信号を空間からスペクトルへの変換にかけ、こうして得られた予測残差信号をデータストリーム14に符号化するように構成される。同様に、デコーダ20は、データストリーム14からの予測残差信号を復号し、こうして得られた予測残差信号を、スペクトルから空間への変換にかけるように構成される。

【0241】

内部的には、エンコーダ10は、元の信号からの、すなわち画像12からの予測信号26の偏差を測定するために予測残差24を生成する予測残差信号形成器22を備えることができる。予測残差信号形成器22は、例えば、元の信号から、すなわち画像12から予測信号を減算する減算器であってもよい。次に、エンコーダ10は、予測残差信号24を空間対スペクトル変換にかけてスペクトル領域予測残差信号24'を取得する変換器28をさらに含み、これは次いで、同様にエンコーダ10に含まれる量子化器32によって量子化される。このように量子化された予測残差信号24'は、ビットストリーム14にコード化される。この目的のために、エンコーダ10は、任意選択として、データストリーム14に変換および量子化され場合に予測残差信号をエントロピー符号化するエントロピーコーダ34を備えることができる。予測信号26は、データストリーム14に符号化され、このストリームから復号可能な予測残差信号24'に基づいて、エンコーダ10の予測ステージ36によって生成される。この目的のために、予測ステージ36は、図1に示されるように、量子化損失を除いて信号24'に対応するスペクトル領域予測残差信号24'''を得るように予測残差信号24'を逆量子化する逆量子化器38を内部で含み得、その後に逆変換器40が続き、これは後者の予測残差信号24'''を逆変換、すなわちスペクトルから空間への変換にかけ、量子化損失を除いて元の予測残差信号24に対応する予測残差信号24'''''を取得する。次に、予測ステージ36のコンバイナ42は、加算などにより、予測信号26と予測残差信号24'''とを再結合して、再構成された信号46、すなわち元の信号12の再構成を得る。再構成された信号46は、信号12'に対応し得る。次に、予測ステージ36の予測モジュール44は、例えば、空間予測、すなわち画像内予測、および/または時間予測、すなわち画像間予測を使用することによって、信号46に基づいて予測信号26を生成する。

30

【0242】

50

## 2. 図 2 によるデコーダ

同様に、図 2 に示されるように、デコーダ 20 は、予測ステージ 36 に対応し、それに対応する方法で相互接続されるコンポーネントから内部に構成されてもよい。特に、デコーダ 20 のエントロピー・デコーダ 50 は、データストリームからの量子化されたスペクトル領域予測残差信号 24' をエントロピー復号することができ、それにおいて、予測ステージ 36 のモジュールに関して上述したように相互接続および協働される逆量子化器 52、逆変換器 54、コンバイナ 56 および予測モジュール 58 は、予測残差信号 24' に基づいて再構成された信号を回復し、その結果、図 2 に示すように、コンバイナ 56 の出力が再構成された信号、すなわち画像 12' をもたらすようにする。

## 【0243】

10

上記で具体的に説明されていないが、エンコーダ 10 が、例えば、いくつかのレートおよび歪みに関連する基準、つまりコーディングコストを最適化する方法などの、いくつかの最適化スキームに従って、例えば予測モード、動きパラメータなどを含むいくつかのコーディングパラメータを設定できることは容易に明らかである。例えば、エンコーダ 10 およびデコーダ 20 ならびに対応するモジュール 44、58 は、それぞれ、イントラコーディングモードおよびインターフォーマンスモードなどの異なる予測モードをサポートすることができる。エンコーダとデコーダがこれらの予測モードタイプを切り替える粒度は、それぞれ画像 12 と 12' のコーディングセグメントまたはコーディングブロックへの細分に対応する。これらのコーディングセグメントの単位で、例えば、画像は、イントラコーディングされるブロックとインターフォーマンスされるブロックに細分され得る。以下でより詳細に概説するように、イントラコード化されたブロックは、それぞれのブロックの空間的な、すでにコード化 / 復号された近傍に基づいて予測される。いくつかのイントラコーディングモードが存在でき、方向または角度のイントラコーディングモードを含むそれぞれのイントラコーディングされたセグメントに対して選択され、これに従って、それぞれのイントラコーディングされたセグメントへ、それぞれの方向性イントラコーディングモードに特異的な特定の方向に沿って、近傍のサンプル値を外挿することにより、それぞれのセグメントが満たされる。イントラコーディングモードは、例えば、1つまたは複数のさらなるモードを含むこともでき、例えば、それぞれのイントラコーディングされたブロックの予測がそれぞれのイントラコーディングされたセグメント内のすべてのサンプルに DC 値を割り当てる DC コーディングモード、および / または、それぞれのブロックの予測が、隣接するサンプルに基づいて、2 次元の線形関数によって定義される平面の駆動の傾きとオフセットを備える、それぞれのイントラコーディングブロックのサンプル位置にわたる 2 次元線形関数によって記述されるサンプル値の空間分布であると近似または決定される平面イントラコーディングモードが挙げられる。それと比較して、インターフォーマンスされたブロックは、例えば、時間的に予測され得る。インターフォーマンスされたブロックの場合、モーションベクトルはデータストリーム内で通知され得て、モーションベクトルは、以前にコーディング / 復号された画像がサンプリングされる、画像 12 が属するビデオの以前にコーディングされた画像の部分の空間変位を示し、それぞれのインターフォーマンスブロックの予測信号を取得するようにする。これは、量子化されたスペクトル領域予測残差信号 24' を表すエントロピー・コード化変換係数レベルなど、データストリーム 14 に含まれる残差信号コーディングに加えて、データストリーム 14 は、様々なブロックへのコーディングモード、インターフォーマンスされたセグメントのモーションパラメータなどの一部のブロックの予測パラメータ、および画像 12 および 12' の各セグメントへの細分を制御およびシグナリングするためのパラメータなどの任意選択の追加パラメータを割り当てるためのコーディングモードパラメータを符号化した可能性があるということを意味する。デコーダ 20 は、これらのパラメータを使用して、エンコーダが行ったのと同じ方法で画像を細分割し、同じ予測モードをセグメントに割り当て、同じ予測を実行して同じ予測信号を生成する。

## 【0244】

20

## 3. 図 3 による機能

30

40

50

図3は、再構成された信号、すなわち再構成された画像12'、一方でデータストリーム14で信号が送られる予測残差信号24'、'、'と結合と、他方で予測信号26との間の関係を示している。すでに上で示したように、結合は追加である場合がある。予測信号26は、図3において、ハッチングを使用して例示的に示されるイントラコーディングブロック、およびハッチングなしで例示的に示されるインターフォーマンスブロックへの画像領域の細分として示される。細分は、画像領域を正方形ブロックまたは非正方形ブロックの行と列へと通常の細分をすること、または四分木細分割など画像12をツリールートブロックから様々なサイズの複数のリーフブロックへとマルチツリー細分することなど、任意の細分にすることができ、それらの混合物が図3に示され、画像領域は最初にツリールートブロックの行および列に細分され、次に再帰的マルチツリー細分に従って1つ以上に細分化する。

10

#### 【0245】

再び、データストリーム14は、いくつかのサポートされたイントラコーディングモードのうちの1つをそれぞれのイントラコーディングブロック80に割り当てるイントラコーディングブロック80のためにそこにコーディングイントラコーディングモードを有し得る。インターフォーマンスブロック82の場合、データストリーム14は、その中にコード化された1つまたは複数の運動パラメータを有することができる。一般的に言えば、インターフォーマンスブロック82は、時間的にコード化されることに限定されない。あるいは、インターフォーマンスブロック82は、現在の画像12自体を超えて以前にコーディングされた部分から予測される任意のブロック、例えば、画像12が属するビデオの以前にコーディングされた画像、または別のビューの画像、またはエンコーダとデコーダがそれぞれスケーラブルなエンコーダとデコーダである場合には、階層的に下位のレイヤであり得る。

20

#### 【0246】

図3の予測残差信号24'、'、'はまた、ブロック84への画像領域の細分として示されている。これらのブロックは、コーディングブロック80および82と区別するために、変換ブロックと呼ばれる場合がある。事実上、図3は、エンコーダ10およびデコーダ20が、それぞれ、画像12および画像12'の2つの異なる細分割をブロックに、すなわち、1つはそれぞれコーディングブロック80および82に細分割し、別の細分割を変換ブロック84に使用できることを示している。両方の細分割は同じであり得、すなわち、各コーディングブロック80および82は、同時に変換ブロック84を形成し得るが、図3は、例えば、変換ブロック84への細分割が、コーディングブロック80、82への細分割の拡張を形成する場合を示し、ブロック80と82の2つのブロック間の任意の境界が2つのブロック84間の境界に重なるようにし、または代替的に言えば、各ブロック80、82は、変換ブロック84の1つと一致するか、または変換ブロック84のクラスターと一致するようにしている。しかし、細分割はまた、変換ブロック84がブロック80、82の間のブロックの境界を交互に横切ることができるように、互いに独立して決定または選択され得る。したがって、変換ブロック84への細分割に関する限り、ブロック80、82への細分割に関して提起されたものと同様のステートメントが当てはまる、すなわち、ブロック84は、画像領域のブロックへの通常の細分の結果、画像領域の再帰的なマルチツリー細分、またはそれらの組み合わせ、またはその他の種類のブロックの結果である可能性がある（行と列への配置の有無にかかわらず）。余談だが、ブロック80、82、84は2次、長方形、その他の形状に限定されないことに留意されたい。

30

#### 【0247】

図3はさらに、予測信号26と予測残差信号24'、'、'の結合が直接再構成された信号12'をもたらすことを示している。しかし、代替実施形態によれば、2つ以上の予測信号26を予測残差信号24'、'、'と結合して、結果として画像12'を得ることに留意されたい。

40

#### 【0248】

図3において、変換ブロック84は、以下の意味を有するものとする。変換器28およ

50

び逆変換器 54 は、これらの変換ブロック 84 のユニットでそれらの変換を実行する。例えば、多くのコーデックは、すべての変換ブロック 84 にある種の DST または DCT を使用する。いくつかのコーデックは、変換をスキップすることを可能にするので、いくつかの変換ブロック 84 について、予測残差信号は、空間領域で直接コード化される。しかし、以下に説明する実施形態によれば、エンコーダ 10 およびデコーダ 20 は、それらがいくつかの変換をサポートするように構成される。例えば、エンコーダ 10 およびデコーダ 20 によってサポートされる変換は、以下を含むことができる。

- DCT - II ( または DCT - III ) 、 DCT は離散コサイン変換の略
- DST - IV 、 DST は Discrete Sine Transform の略
- DCT - IV
- DST - VII
- アイデンティティ変換 ( IT )

#### 【 0249 】

当然、変換器 28 はこれらの変換のすべての順方向変換バージョンをサポートするが、デコーダ 20 または逆変換器 54 はその対応する逆方向または逆バージョンをサポートする。

- 逆 DCT - II ( または逆 DCT - III )
- 逆 DST - IV
- 逆 DCT - IV
- 逆 DST - VII
- アイデンティティ変換 ( IT )

#### 【 0250 】

以下の説明は、エンコーダ 10 およびデコーダ 20 によってサポートされ得る変換についてのさらなる詳細を提供する。いずれの場合でも、サポートされる変換のセットは、1 つのスペクトルから空間への変換や空間からスペクトルへの変換など、1 つの変換のみを含む場合があることに留意されたい。

#### 【 0251 】

すでに上で概説したように、図 1 から図 3 は、本出願によるエンコーダおよびデコーダの特定の例を形成するために、以下でさらに説明する本発明の概念を実施できる例として提示されている。これまでのところ、図 1 および図 2 のエンコーダおよびデコーダは、それぞれ、本明細書で以下に説明するエンコーダおよびデコーダの可能な実装を表すことができる。ただし、図 1 と図 2 は単なる例である。しかし、本出願の実施形態によるエンコーダは、画像 12 のブロックベースの符号化を実行することができ、以下でより詳細に概説される概念を使用し、例えばビデオエンコーダはないが、依然画像エンコーダである点、相互予測をサポートしていない点、またはブロック 80 への細分割が図 3 に例示されているのとは異なる方法で実行されるという点で、図 1 のエンコーダとは異なる。同様に、本出願の実施形態によるデコーダは、データストリーム 14 からの画像 12' のブロックベースの復号を実行することができ、以下でより詳細に概説される概念を使用しているが、例えばビデオデコーダはないが、依然画像デコーダである点、相互予測をサポートしていない点、またはブロックへの画像 12' の細分割が図 3 に記載されているのとは異なる点、および / または変換ドメインではなく、空間ドメインのデータストリーム 14 から予測残差を導出する点、などで、図 2 のデコーダ 20 とは異なり得る。

#### 【 0252 】

##### 4. 図 4 による算術エンコーダ

図 4 は、本発明の一実施形態による、算術エンコーダのブロック概略図を示す。

図 4 による算術エンコーダ 400 は、例えば、ビデオエンコーダで使用することができる。しかし、任意選択で、算術エンコーダ 400 は、オーディオエンコーダ、画像エンコーダ、ニューラルネットワークの係数を符号化するためのエンコーダなどでも使用することができる。

#### 【 0253 】

10

20

30

40

50

算術エンコーダ 400 は、符号化されるシンボル 410 を受信するように構成され、符号化されるシンボル 410 は、シンボル値によって表すことができる。さらに、エンコーダ 400 は、典型的には、コンテキスト情報 412 も受信し、これは、例えば、どのタイプの情報が、符号化されるシンボル 410 によって表されるかを記述し得る。例えば、コンテキスト情報 412 は、例えば、符号化されるシンボル 410 がどのタイプのサイド情報を記述するか、またはシンボル 410 がどのタイプの変換係数を符号化するかを記述するコンテキストインデックス k によって表すことができる。

#### 【0254】

さらに、算術エンコーダ 400 は、符号化されるシンボル 410 を表す、または符号化されるシンボル 410 のシーケンスを表すビットストリーム 420 を提供するように構成される。

10

#### 【0255】

算術エンコーダ 400 は、算術符号化コアまたは算術エンコーダコア 430 を備え、これは、符号化されるシンボル 410 を受け取り、それに基づいて、ビットストリーム 420 を提供する。算術符号化コアまたは算術エンコーダコア 430 は、通常、間隔サイズ情報を受信し、これは、例えば、シンボル（例えば、最も可能性の低いシンボル）がマップされるサブインターバル（全コーディング間隔のうち）のサイズを表すことができる。さらに、算術符号化コアまたは算術エンコーダコアはまた、現在のコーディング間隔サイズ（例えば、コーディング間隔の合計サイズ）を記述するコーディング間隔サイズ情報 434 を提供する。コーディング間隔サイズ 434 は、符号化されるシンボル 410 に応じて（または、より正確には、符号化されるシンボル 410 のシーケンスに応じて）、時間とともに変化し得ることに留意されたい。

20

#### 【0256】

コーディング間隔サイズは、例えば、算術符号化の過程で実行される再スケーリング操作のために変化する可能性がある。例えば、算術符号化コアは、例えば、High Efficiency Video Coding (HEVC) の規格 (H.265) に記載されている機能を実行することができる。

#### 【0257】

算術エンコーダ 400 はまた、コーディング間隔サイズ決定またはコーディング間隔サイズ決定器 440 を備える。コーディング間隔サイズ決定 440 は、符号化されるシンボル 410、または少なくとも 1 つまたは複数の以前に符号化されたシンボルを受信し、好ましくは（必ずしもではないが）コンテキスト情報 412 も受信する。さらに、間隔サイズ決定 440 は、コーディング間隔サイズ情報 434 を受信する。間隔サイズ決定 440 は、コーディング間隔サイズ情報に基づいて、符号化されるシンボル 410（または少なくとも 1 つまたは複数の以前に符号化されたシンボル）、および任意選択で、算術符号化コア 430 によって使用される間隔サイズ情報 432 のコンテキスト情報 412 を提供する。

30

#### 【0258】

図 4 による概念の機能に関して、間隔サイズ決定 440 は、間隔サイズ情報 432 を更新するのに役立つことに留意されたい（継続的に、例えば、符号化される新しいシンボルごとに）。この更新では、以前に符号化されたシンボルの統計が考慮される。さらに、算術符号化 430 によって提供されるコーディング間隔サイズ情報 434 も考慮される。なぜなら、間隔サイズ情報 432 は、好ましくは、コーディング間隔サイズ情報 434 との適切な関係で提供されるべきものであるからである。この文脈において、コーディング間隔サイズ情報 434 は、例えば、（合計）コーディング間隔の現在のサイズを表すことができる（これは、時々発生するコーディング間隔の再正規化によって引き起こされ得る）が、一方で間隔サイズ情報 432 は、例えば、特定のシンボル（例えば、最も可能性の低いシンボル）に関連付けられた全体の（合計）コーディング間隔内の部分のサイズを記述し得る。したがって、間隔サイズ情報 434 は、コーディング間隔の合計サイズを表しながら、間隔サイズ情報 434 は、特定のシンボルに関連付けられているコーディング間隔

40

50

の一部のサイズを表しているため、間隔サイズ情報は、通常、コーディング間隔サイズ情報 434 よりも小さい。その結果、間隔サイズ情報 432 は、通常、コーディング間隔サイズ情報 434 に合わせてスケーリングする（スケーリングは、コーディングが非常に非効率的である状況を回避するために、任意選択でいくらか非線形の振る舞いを含み得る）。

#### 【0259】

さらに結論として、適切な間隔サイズ情報 432 を決定することにより、コーディング間隔の合計サイズ（コーディング間隔サイズ情報 434 によって表される）および以前に符号化されたシンボル（ならびにコンテキスト）の統計を考慮して、算術符号化は効率的な方法で実行され、以前に符号化されたシンボルの統計を考慮することは、コーディング効率を改善するのに役立つ。

10

#### 【0260】

しかし、図 4 による算術エンコーダ 400 は、本明細書に開示されるような信号エンコーダのいずれか（例えば、ビデオエンコーダ）で使用できることに留意されたい。さらに、図 4 による算術エンコーダ 400 は、本明細書に記載の特徴、機能、および詳細のいずれかによって任意選択で補足できることに留意されたい。特に、間隔サイズ決定 440 は、本明細書に開示された概念のいずれかを、個別に、および組み合わせて使用することができる。

#### 【0261】

図 4 による算術エンコーダ 400 を結論付けることは、任意選択で、個別におよび組み合わせて、本明細書に開示される特徴、機能および詳細のいずれかによって補足され得る。

20

#### 【0262】

##### 5. 図 5 による算術デコーダ

図 5 は、本発明の実施形態による、算術デコーダ 500 のブロック概略図を示す。算術デコーダ 500 は、ビットストリーム 510（ビットストリーム 420 に対応し得る）を受信し、それに基づいて、復号されたシンボル 520（または復号されたシンボル 520 のシーケンス）を提供するように構成される。典型的には、復号されたシンボル 520 は、符号化されるシンボル 410 に対応し得る。典型的には、算術エンコーダ 400 および算術デコーダ 500 は、一緒にされると、ロスレス符号化および復号を実行し得、その結果、算術エンコーダ 400 によって符号化されたシンボル 410 は、算術デコーダ 500 によって復号されたときに「完全な」再構成を可能にするビットストリーム 420 を提供し、復号されたシンボル 520 が符号化されたシンボル 410 に対応するようになる。

30

#### 【0263】

算術デコーダ 500 は、ビットストリームを受信し、それに基づいて、復号されたシンボル 520 を提供する、算術復号コアまたは算術デコーダコア 530 を備える。算術復号コア 530 は、典型的には、コーディング間隔サイズ情報 434 に実質的に対応し得るコーディング間隔サイズ情報 532 を提供し、間隔サイズ情報 432 に対応し得る間隔サイズ情報 534 を受信する。算術デコーダ 500 はまた、コーディング間隔サイズ情報 532 に基づいて、また、1つまたは複数の復号されたシンボル 520 に基づいて、算術復号コア 530 によって使用される、間隔サイズ情報 534 を提供するように構成された、コーディング間隔サイズ決定またはコーディング間隔サイズ決定器 540 を備える。さらに、コーディング間隔サイズ決定 540 は、任意選択で、コンテキスト情報 550 を使用することができ、これは、現在考慮されている復号されたシンボル 520 によって表される情報のタイプを記述し得る。したがって、コーディング間隔サイズ決定 540 は、複数の異なる「コンテキスト」、すなわち、複数の異なるタイプの復号された情報（または復号される異なるタイプの情報）の間隔サイズ情報 534 を決定することができる。

40

#### 【0264】

算術復号コア 530 は、例えば、全コーディング間隔（そのサイズはコーディング間隔サイズ情報によって記述される）のうち、ビットストリーム 510 によって表される値がどのサブ間隔にあるかを決定し、その結果、どのシンボルがビットストリームで表されているか決定することができる。総コーディング間隔のサブ間隔のサイズは、例えば、間隔

50

サイズ情報 534 によって記述される。例えば、間隔サイズ情報 534 は、特定のシンボル（例えば、最も可能性の低いシンボル）に関連付けられた合計コーディング間隔のサブインターバルのサイズを記述し得る。さらに、再正規化のために、コーディング間隔サイズ（例えば、総コーディング間隔のサイズ）が時々（または復号されたシンボルごとにさえ）変化する可能性があることに留意されたい。

#### 【0265】

結論として、算術復号 500 は、ビットストリーム 510 に基づいて復号されたシンボル 520 を提供するために使用され、以前に復号されたシンボルの履歴（またはより正確には、以前に復号されたシンボルの統計）は、復号されるシンボルに関連付けられた間隔サイズを（動的に）調整するために使用される（調整された間隔サイズは、間隔サイズ情報 534 によって記述される）。さらに、間隔サイズ決定 540 は、例えば、間隔サイズ決定 440 と実質的に同一であり得ることに留意されたい。また、間隔サイズ決定 540 は、本明細書に開示される機能のいずれかを含み得ることに留意されたい。したがって、間隔サイズを決定するための概念のいずれかを、間隔サイズ決定 540 で使用することができる。

#### 【0266】

本明細書に記載の算術デコーダ 500 は、本明細書に記載のデコーダ（例えば、オーディオデコーダまたはビデオデコーダ）のいずれかで使用できることに留意されたい。しかし、算術デコーダ 500 はまた、画像またはニューラルネットワークの係数の復号のために使用され得る。

#### 【0267】

一般的に言えば、本明細書に記載の算術デコーダ 500 は、任意選択で、個別におよび組み合わせて、本明細書に開示される特徴、機能、および詳細のいずれかによって補足され得る。

#### 【0268】

##### 6. 図 6 による間隔サイズ情報の決定の概念

図 6 は、間隔サイズ情報を決定するための概念の概略図を示し、これは、例えば、図 4 による算術エンコーダ 400 または図 5 による算術デコーダ 500 で使用することができる。間隔サイズ決定の形で、または間隔サイズ決定器の形で実装され得る概念 600 は、例えば、間隔サイズ決定または間隔サイズ決定器 440 を実現するために使用され得、および / または間隔サイズ決定または間隔サイズ決定器 540 を実現するために使用される。

#### 【0269】

間隔サイズ決定 600 は、例えば、1つまたは複数のシンボル値 610 を受け取ることができ、これは、符号化されるシンボル 410 に対応すしても、以前に符号化されたシンボルに対応しても、または1つまたは複数の以前に復号されたシンボル 520 に対応してもよい。さらに、間隔サイズ決定 600 は、任意選択で、コンテキスト情報 612 を受信することができ、これは、例えば、コンテキストモデルインデックス  $k$  の形をとることができる。さらに、間隔サイズ決定 600 は、間隔サイズ情報 620 を提供することができ、これは、例えば、間隔サイズ情報 432 に対応することができ、または間隔サイズ情報 534 に対応することができる。

#### 【0270】

間隔サイズ決定は、シンボル値 610 を考慮し、任意選択でコンテキスト情報 612 を考慮して、1つまたは複数の以前に決定された状態変数 644 に基づいて、1つまたは複数の更新された状態変数 642 を提供する状態変数更新 640 を含む。状態変数更新 640 はまた、1つまたは複数の他のパラメータを考慮することができ、これらは、例えば、静的であり得るか、または、例えば、コンテキスト情報に依存してそれぞれのコンテキストに適合され得る。例えば、状態変数更新 640 は、1つまたは複数の重み付け係数、すなわち、 $m_i^k$

10

20

30

40

50

および / または

$n_i^k$

を考慮することができる。状態変数更新 640 は、任意選択で、「オフセット」、例えば、 $z$ 、およびルックアップテーブル、例えば、 $A$ も考慮することができる。さらに、状態変数更新 640 は、任意選択で、状態変数の初期化を実行する必要がある情報信号に応答して、初期値に初期化され得る。状態変数の初期化の場合、以前に決定された状態変数値 644 は無視され得、「更新された」状態変数 642 は、例えば、事前に決定され得る初期値に設定され得る。

**【 0 2 7 1 】**

さらに、間隔サイズ決定 600 は、更新された状態変数（または状態変数值）620 に基づいて間隔サイズ情報 620 を決定する間隔サイズ決定コアを備える。さらに、間隔サイズ決定コア 650 は、例えば、コーディング間隔サイズ情報 652 を考慮することができ、これは、例えば、コーディング間隔サイズ情報 432 またはコーディング間隔サイズ情報 532 に対応し得る。

**【 0 2 7 2 】**

したがって、間隔サイズ決定コア 650 は、コーディング間隔の合計サイズを説明するコーディング間隔サイズ情報 652 に基づいて、また以前に符号化された、または以前に復号されたシンボルの統計に関する情報に基づいて、特定のシンボルに関連付けられたサブインターバルのサイズを説明する間隔サイズ情報を提供することができ、それは 1 つまたは複数の更新された状態変数 642 によって表される。

**【 0 2 7 3 】**

しかし、間隔サイズ決定 600 は、本明細書で説明される算術エンコーダおよび算術デコーダで使用され得ることに留意されたい。さらに、間隔サイズ決定 600 は、本明細書に開示される特徴、機能、および詳細のいずれかによって補足され得る。特に、状態変数の更新は、本明細書に開示された概念のいずれかを使用することができる。さらに、間隔サイズ決定コアはまた、本明細書に開示される概念のいずれかを使用することができる。

**【 0 2 7 4 】**

状態変数の更新について本明細書で説明されている概念のいずれかを、本明細書で開示されている間隔サイズ決定コアの概念のいずれかと任意選択に結合することができることに留意されたい。本明細書に開示される特徴、機能、および詳細のいずれかは、任意選択で、個別に、および組み合わせて、概念 600 に導入することができる。1 つまたは複数の更新された状態変数 642 に基づく間隔サイズ情報 620 の決定に関して開示された特徴、機能、および詳細のいずれも、状態変数の更新に関して説明する特徴、機能、および詳細のいずれかと独立して使用できることに留意されたい。

**【 0 2 7 5 】**

7. 図 7 による間隔サイズ決定の概念

図 7 は、間隔サイズ決定概念 700 の概略図を示す。特に、図 7 は、「間隔サイズ決定コア」の機能を利用する概念を示している。言い換えれば、図 7 による概念 700 は、更新された状態変数に基づいて、またコーディング間隔サイズ情報に基づいて、間隔サイズ情報を提供するのに適している。しかし、図 7 による概念 700 は、間隔サイズ決定または間隔サイズ決定器で実装され得る。

**【 0 2 7 6 】**

「間隔サイズ決定コア」と見なすことができる間隔サイズ決定 700 は、更新された状態変数 710 を受け取り、これは、例えば、更新された状態変数 642 に対応し得る。さらに、間隔サイズ決定 700 は、コーディング間隔サイズ情報 712 を受信し、これは、例えば、コーディング間隔サイズ情報 652 に対応し得る。任意選択として、間隔サイズ決定 700 の機能は、コンテキストに応じて調整され得るので、間隔サイズ決定 700 はまた、コンテキスト情報を使用する。例えば、コンテキスト情報は、コンテキストモデルインデックス  $k$  の形をとることができる。したがって、間隔サイズ決定 700 で使用され

10

20

30

40

50

る特定の機能またはパラメータは、コンテキストモデルインデックス  $k$  によって表されるように、実際のコンテキストに依存して適合され得る。

【0277】

間隔サイズ決定 700 は、間隔サイズ情報 720 を提供し、これは、間隔サイズ情報 620 に対応し得る。

したがって、間隔サイズ決定 700 は、例えば、図 6 に記載された間隔サイズ決定コア 650 の代わりになり得る。

【0278】

以下では、間隔サイズ決定コア 700 のいくつかの詳細について説明する。

間隔サイズ決定コア 700 は、第 1 の処理パス 730 および第 2 の処理パス 750 を含み、これらは代替案と見なされ得ることに留意されたい。第 1 の処理パス 730 は、(任意選択の)スケーリング / 丸め 732 を含み、第 1 の更新された状態変数(例えば、 $s_1^k$ )

がスケーリングおよび / または丸められる。例えば、第 1 の更新された状態変数(値)のスケーリングにスケーリング係数

$a_1^k$

を使用できる。例えば、スケーリングは、スケーリング係数との乗算であり得る。例えば、丸めは、隣の整数の値へ切り捨て、スケーリングの結果以下にする場合がある。同様に、第 1 の処理パス 730 は、第 2 の(任意選択の)スケーリング / 丸め 734 を含み得、これは、例えば、それぞれのスケーリング係数(例えば、 $a_2^k$ )

を有する第 2 の更新された状態変数(例えば、 $s_2^k$ )

のスケーリングを含み得、それはまた、スケーリングの結果以下へと、隣り合う整数の値へ切り捨てる場合もある。したがって、第 1 のスケーリングされ丸められた状態変数 733 および第 2 のスケーリングされ丸められた状態変数 735 を取得することができる。さらに、第 1 の処理パス 730 は、第 1 のスケーリングおよび / または丸められた更新状態変数 733 を受け取り、さらに量子化されたコーディング間隔サイズ情報 762 を受け取る、第 1 のルックアップテーブルベースのマッピング 736 を含み得る。例えば、第 1 のルックアップテーブルベースのマッピング 736 は、2 次元ルックアップテーブルを使用して、第 1 の間隔サイズ寄与 737 を取得することができる。同様に、第 1 の信号処理パス 730 は、第 2 のルックアップテーブルベースのマッピング 738 を含み得、これは、第 2 のスケーリングおよび / または丸められた更新状態変数 735 および量子化されたコーディング間隔サイズ情報 762 を受け取れ、それに基づいて、第 2 の間隔サイズの寄与 739 を提供する。

【0279】

例えば、第 1 のルックアップテーブルベースのマッピング 736 は、2 次元ルックアップテーブルを使用することができ、第 1 のルックアップテーブルインデックスは、スケーリングおよび / または丸められた第 1 の更新された状態変数 733 によって決定され得、第 2 のルックアップテーブルインデックスは、量子化されたコーディング間隔サイズ情報 762 によって決定され得る。例えば、第 1 のスケーリングおよび / または丸められた更新状態変数 733 は、特定の範囲内(例えば、0 から最大値、または 1 から最大値、または第 1 のテーブルインデックスとして許容可能な範囲内)の整数の値をとることができる。同様に、量子化された間隔サイズ情報 762 は、第 2 のテーブルインデックスとして機能することができる整数値の形式をとることができる。例えば、量子化されたコーディング間隔サイズ情報 762 は、0 から最大値の間、または 1 から最大値の間の範囲(または

10

20

30

40

50

第 2 のテーブルインデックスとして機能することができる任意の値の範囲内 ) の整数値の形式をとることができる。言い換えれば、第 1 のスケーリングおよび / または丸められた更新状態変数 7 3 3 および量子化されたコーディング間隔サイズ情報 7 6 2 の両方が、ルックアップテーブルベースのマッピング 7 3 6 で使用されるルックアップテーブルの要素を選択するためのテーブルインデックスとして使用される。したがって、使用されるルックアップテーブルのエントリは、第 1 の間隔サイズ寄与 7 3 7 として提供される。

#### 【 0 2 8 0 】

しかし、第 2 のルックアップテーブルベースのマッピング 7 3 8 は、同じ方法で実行することができ、スケーリングおよび / または丸められた第 2 の更新された状態変数 7 3 5 および量子化されたコーディング間隔サイズ情報 7 6 2 が、第 2 のルックアップテーブルベースのマッピング 7 3 8 で使用されるルックアップテーブルのエントリの選択のため、2 つのインデックス ( 例えば、 $i$  および  $j$  ) として使用される。したがって、第 2 の間隔サイズ寄与 7 3 9 が得られる。

#### 【 0 2 8 1 】

第 1 の間隔サイズ寄与 7 3 7 および第 2 の間隔サイズ寄与 7 3 9 は、結合 7 4 0 で結合され得、それにより、結合間隔サイズ寄与 7 4 2 が得られる。結合間隔サイズ寄与 7 4 2 は、間隔サイズ情報 7 2 0 として直接機能し得るか、または間隔サイズ情報 7 2 0 は、後処理 ( 例えば、固定スケーリング、または丸めなど ) を使用して結合間隔サイズ寄与 7 4 2 から導出され得る。

#### 【 0 2 8 2 】

結論として、第 1 の処理パス 7 3 0 を使用する場合、間隔サイズ情報 7 2 0 は、2 つ以上のルックアップテーブルベースのマッピング 7 3 6 、 7 3 8 の結果を結合することによって取得することができ、使用されるルックアップテーブルのそれぞれのエントリは、それぞれの更新された状態変数に基づいて、またコーディング間隔サイズ情報に基づいて、決定される。したがって、間隔サイズ情報 7 2 0 は、非常に効率的な方法で取得することができる。

#### 【 0 2 8 3 】

しかし、あるいは、第 2 の信号処理パス 7 5 0 に示されるような処理を実行することができる。第 2 の信号処理パスによって実行される処理は、精度がわずかに低下するという代償を払って、かなり計算効率が高くなる。例えば、第 2 の信号処理パス 7 5 0 は、第 1 の更新された状態変数および第 2 の更新された状態変数 ( および任意選択で追加の更新された状態変数 ) が結合され、それによって結合された更新された状態変数 7 5 1 a を取得する結合 7 5 1 を含む。第 2 の処理パス 7 5 0 はまた、任意選択のスケーリング / 丸め 7 5 2 を含み、これは、例えば、スケーリング / 丸め 7 3 2 、 7 3 4 に対応し得る。したがって、スケーリング / 丸め 7 5 0 は、ルックアップテーブルベースのマッピング 7 5 6 内の要素を選択するために使用される、スケーリングおよび / または丸められた結合された更新された状態変数 7 3 3 を提供する。ルックアップテーブルベースのマッピング 7 5 6 は、好ましくは 2 次元マッピングであり、ルックアップテーブルの第 1 のインデックスは、結合された更新された状態変数 7 5 1 a またはそのスケーリングおよび / または丸められたバージョン 7 5 3 によって決定され、2 次元ルックアップテーブルの第 2 のインデックスは、量子化されたコーディング間隔サイズ情報 7 6 2 によって決定される。したがって、ルックアップテーブルベースのマッピング 7 5 6 は、間隔サイズ寄与 7 5 7 を提供し、これは、間隔サイズ情報 7 5 2 として使用され得るか、または任意の後処理 ( 例えば、スケーリング ) によって間隔サイズ情報 7 2 0 から導出され得る。

#### 【 0 2 8 4 】

結論として、間隔サイズ決定コア 7 0 0 は、更新された状態変数 7 1 0 に基づいて、コーディング間隔サイズ情報 7 1 2 を考慮して、間隔サイズ情報 7 2 0 を取得するように構成される。ルックアップテーブルベースのマッピングは、間隔サイズ情報 7 2 0 を導出するために使用される 2 つ以上の間隔サイズ寄与 7 3 7 、 7 3 9 を決定するために、第 1 の信号処理パス 7 2 0 で使用される。ルックアップテーブルベースのマッピング 7 5 6 は、

10

20

30

40

50

代替の第 2 の信号処理バス 750 で使用されて、間隔サイズ寄与 757 または間隔サイズ情報 720 を決定する。更新された状態変数とコーディング間隔サイズ情報 712 の両方を考慮してルックアップテーブルのインデックスを決定するルックアップテーブルベースのマッピングを使用することによって、特に効率的な計算を達成することができ、乗算の演算を省くことが可能になる。

#### 【0285】

図 7 による間隔サイズ決定コア 700 は、本明細書に記載の算術エンコーダおよび算術デコーダのいずれかで使用できることに留意されたい。さらに、間隔サイズ決定コア 700 は、任意選択で、本明細書に開示される特徴、機能、および詳細のいずれかによって、個別におよび組み合わせて、補足することができる。

10

#### 【0286】

##### 8. 図 8 による間隔サイズ決定コア

図 8 は、本発明の実施形態による、間隔サイズ決定コア 800 のプロック概略図を示す。図 8 による間隔サイズ決定コア 800 は、本明細書に開示されるオーディオエンコーダおよびオーディオデコーダのいずれかにおいて使用され得る。

#### 【0287】

間隔サイズ決定コア 800 は、更新された状態変数 810 と、コーディング間隔サイズ情報 812 も受け取る。更新された状態変数 810 は、例えば、更新された状態変数 710 または更新された状態変数 642 に対応し得る。コーディング間隔サイズ情報 812 は、例えば、コーディング間隔サイズ情報 712 またはコーディング間隔サイズ情報 652 またはコーディング間隔サイズ情報 532 またはコーディング間隔サイズ情報 434 に対応し得る。さらに、間隔サイズ決定コア 800 は、間隔サイズ情報 820 を提供し、これは、間隔サイズ情報 720 または間隔サイズ情報 620 または間隔サイズ情報 534 または間隔サイズ情報 432 に対応し得る。

20

#### 【0288】

間隔サイズ決定コア 800 は、例えば、確率値 832 の決定 830 を含み得る。確率値 832 は、例えば、1つまたは複数の更新された状態変数 810 に基づいて取得され得、例えば、符号化されるシンボル（例えば、「0」または「1」）の確率を記述し得る。間隔サイズ決定コア 830 は、可能性の低いシンボル（または可能性の最も低いシンボル）の確率の決定 840 をさらに含み、これは、可能性の低いシンボルまたは最も可能性の低いシンボルの確率を説明する確率値 842 を提供し得る。間隔サイズ決定コア 800 はまた、量子化 850 を含み、これは、確率値 842 を量子化して、量子化された確率値 852 を取得する。さらに、間隔サイズ決定コア 800 はまた、量子化されたコーディング間隔サイズ情報 862 を得るために、コーディング間隔サイズ情報を量子化する量子化 860 を備える。間隔サイズ決定コア 800 はまた、量子化確率値 852 および量子化コーディング間隔サイズ情報 862 を受け取り、量子化確率値 852 および量子化コーディング間隔サイズ情報 862 の両方を間隔サイズ情報 820 にマッピングするマッピング 870 を備える。

30

#### 【0289】

以下の間に、間隔サイズ決定コア 800 の機能に関するいくつかの詳細が説明される。

40

確率値 832 の決定 830 は、例えば、第 1 の信号処理バス 880 または第 2 の信号処理バス 890 を含み得る。信号処理バス 880、890 が代替案として考慮され得ることに留意されたい。第 1 の信号処理バス 880 は、例えば、ルックアップテーブルを使用して、第 1 の更新された状態変数 810 を第 1 の確率値 883 にマッピングする第 1 のマッピング 882 を備える。第 1 の信号処理バス 880 はまた、第 2 のマッピング 884 を含み、これは、例えば、ルックアップテーブルを使用して、第 2 の更新された状態変数 810 を第 2 の確率値 885 にマッピングする。第 1 の信号処理バス 880 はまた、例えば、異なるスケーリングが第 1 の確率値 883 および第 2 の確率値 885（および、任意選択で、量子化を使用することができる）に適用され得る線形の結合を使用するなどして、第 1 の確率値 883 および第 2 の確率値 885 を結合するように構成され得る結合 886 を

50

含む。したがって、結合 8 8 6 は、確率値 8 3 2 を提供する。

【 0 2 9 0 】

あるいは、第 2 の信号処理パス 8 9 0 を使用することができる。第 2 の信号処理パス 8 9 0 は、結合 8 9 2 を含み、これは、2 つ以上の更新された状態変数 8 1 0 を受け取り、それに基づいて、結合された更新された状態変数 8 9 3 を（例えば、線形結合を使用して）提供する。さらに、第 2 の信号処理パス 8 9 0 は、例えば、ルックアップテーブルを使用して、結合された更新された状態変数 8 9 3 を確率値 8 3 2 にマッピングするマッピング 8 9 4 を備える。

【 0 2 9 1 】

したがって、第 1 の信号処理パス 8 8 0 を使用するか、または第 2 の信号処理パス 8 9 0 を使用して、確率値 8 3 2 を取得することが可能であり、これは、2 つの異なる代替案と見なすことができる。第 1 の信号処理パスは、2 つのマッピング 8 8 2 、8 8 4 があり、精度もわずかに向上しているため、複雑さがわずかに増加する。対照的に、第 2 の信号処理パス 8 9 0 は、単一のマッピングのみを含み、したがって、わずかに精度が低下するという犠牲を払い、わずかに複雑さが減少する。

【 0 2 9 2 】

可能性の低いシンボルの確率の決定 8 5 0 において、可能性の低いシンボルの確率は、例えば、確率値 8 3 2 および確率値の補数（1 から確率値を引いたもの）からより小さな値を選択することによって決定され得る。したがって、可能性の低いシンボルの確率を表す確率値 8 4 2 が得られる。この確率値 8 4 2 は、例えば、量子化関数

$Q_p$

10

20

（ . ）または

$Q_{p_2}$

（ . ）を使用して、量子化 8 5 0 で量子化される。したがって、量子化された確率値または確率指数  $i_{8 5 2}$  が取得され、マッピング 8 7 0 で使用される。

【 0 2 9 3 】

コーディング間隔サイズ情報 8 1 2 の量子化 8 6 0 は、例えば、量子化されたコーディング間隔サイズ値 8 6 2 または量子化されたコーディング間隔サイズ指数  $j$  を提供することができる。しかし、場合によっては、両方が量子化されたコーディング間隔サイズ値  $Q_r$  (  $R$  ) であり、コーディング間隔サイズ指数  $j$  が取得され得、これは、例えば、両方ともマッピング 8 6 0 で使用され得る。マッピング 8 6 0 は、例えば、本明細書に記載のルックアップテーブル RangeTableLPS に基づく、および / または本明細書に記載のルックアップテーブル BaseTableLPS に基づくことができる、および / または本明細書に記載のルックアップテーブル ProbTableLPS を使用することができるマッピングメカニズムを使用することができる。任意選択で、本明細書で説明されるスケーリング関数 Scale ( . ) もまた、マッピング 8 7 0 で使用され得る。

【 0 2 9 4 】

言い換えれば、量子化された確率値または確率指数 8 5 2 に基づいて、および量子化されたコーディング間隔サイズ 8 6 2 および / またはコーディング間隔サイズ指数  $j$  に基づいて、間隔サイズ情報 8 2 0 を導出するために使用することができるいくつかの異なるマッピングがあり得る。

【 0 2 9 5 】

追加の注意として、量子化 8 5 0 は、例えば、確率値 8 4 2 ( あるいは、決定 8 4 0 が省略される場合、確率値 8 3 2 ) を整数値にマッピングすることができることに留意されたい。あるいは、量子化 8 5 0 は、値 8 4 2 ( あるいは、決定 8 4 0 が任意選択で省略される場合、値 8 3 2 ) を、確率値 8 3 2 または確率値 8 4 2 よりも低い数値分解能を有する値にマッピングすることができる。しかし、量子化された確率値 8 5 2 は、確率指数 ( 例えば、  $i$  ) の形式、すなわち整数の表現の形式をとることが好ましく、それはルックア

30

40

50

ップテーブル（例えば、RangeTablePS）のエントリを指定するためのテーブルインデックスとして使用できる。

【0296】

量子化860は、異なる結果をもたらす可能性がある。コーディング間隔サイズ情報812は、例えば、コーディング間隔サイズ情報812よりも低い解像度を含む量子化されたコーディング間隔サイズ情報862に量子化され得る。言い換えれば、コーディング間隔サイズ情報812の値の間隔は、量子化されたコーディング間隔サイズ情報862の個々の量子化された値にマッピングされ得、量子化は、例えば、線形または非線形であり得る。あるいは、またはさらに、コーディング間隔サイズ情報は、コーディング間隔サイズインデックスj、すなわち、ルックアップテーブルのエントリを選択するためのテーブルインデックスとして直接使用することができる整数値の連続範囲にマッピングされ得る。しかし、場合によっては、異なる値に量子化された量子化されたコーディング間隔サイズ情報と、異なる指数j（すなわち、後続の整数値）に量子化された量子化されたコーディング間隔サイズ情報の両方が使用され得ることに留意されたい。

上記のように、マッピング870は、本明細書で説明される異なる概念を使用して作成することができる。

【0297】

結論として、間隔サイズ決定コア800は、本明細書に記載の算術エンコーダおよび算術デコーダのいずれかで使用することができる。さらに、間隔サイズ決定コア800は、任意選択で、本明細書に開示される特徴、機能、および詳細のいずれかによって、個別におよび組み合わせて、補足することができる。

【0298】

9. 図9による間隔サイズ決定コア

図9は、本明細書に開示される算術エンコーダおよび算術デコーダのいずれかで使用可能な間隔サイズ決定コア900のブロック概略図を示す。

【0299】

間隔サイズ決定コア900は、状態変数値910を受け取り、間隔サイズ情報920を提供するように構成される。一般的に言えば、間隔サイズ決定コア900は、複数の状態変数値910に基づいて、符号化される1つまたは複数のシンボル値の算術符号化（または復号される1つまたは複数のシンボル値の算術復号）の間隔サイズ情報920を導出するように構成され、これは、異なる適合時定数で以前に符号化された複数のシンボル値の統計を表す。間隔サイズ決定コア900は、任意選択の第1のスケーリングおよび/または丸め930を含み、第1の状態変数（更新された状態変数であり得る）がスケーリングおよび/または丸められる。間隔サイズ決定コア900はまた、任意選択の第2のスケーリングおよび/または丸め934を含み、第2の状態変数がスケーリングおよび/または丸められる。したがって、任意選択の第1のスケーリングおよび/または丸めは、第1のスケーリングおよび/または丸められた状態変数932を提供し、任意選択の第2のスケーリングおよび/または丸め934は、第2のスケーリングおよび/または丸められた状態変数934を提供する。

【0300】

間隔サイズ決定コア900はまた、ルックアップテーブルを使用する第1のマッピング940を備える。第1のマッピング940は、ルックアップテーブル（例えば、ルックアップテーブルLUT1、または複数の異なる値Rに対するR・LUT1を定義する2次元ルックアップテーブル）を使用して、第1の状態変数、または第1の状態変数のスケーリングおよび/または丸められたバージョン932をマッピングする。したがって、第1の確率値942は、第1のマッピング940によって得られる。間隔サイズ決定コア900はまた、第2のマッピング950を備え、これはルックアップテーブル（例えば、ルックアップテーブルLUT1、または複数の異なる値Rに対するR・LUT1を定義する2次元ルックアップテーブルを使用）を使用して、第2の状態変数、またはそのスケーリングおよび/または丸められたバージョン934をマッピングする。したがって、第2の確率

10

20

30

40

50

値 9 5 2 は、第 2 の状態変数値に基づいて、またはそのスケーリングおよび／または丸められたバージョン 9 3 4 に基づいて、第 2 のマッピング 9 5 0 によって得られる。

#### 【 0 3 0 1 】

最後に、間隔サイズ情報 9 2 0 は、第 1 の確率値 9 4 2 に基づいて、および第 2 の確率値 9 5 2 に基づいて取得される。

しかし、第 1 の確率値 9 4 2 および第 2 の確率値 9 5 2 の決定は、第 1 のマッピング 9 4 0 および第 2 のマッピング 9 5 0 を使用して取得できる量の例としてのみ考慮されるべきであることに留意されたい。対照的に、例えば、間隔サイズ情報 9 2 0 への寄与のように、第 1 のマッピング 9 4 0 および第 2 のマッピング 9 5 0 を使用して他の量を導出することも可能であろう。

10

#### 【 0 3 0 2 】

しかし、図 9 に記載されている間隔サイズ決定コア 9 0 0 は、本明細書に開示されている算術エンコーダまたは算術デコーダのいずれかで使用できることに留意されたい。さらに、間隔サイズ決定コア 9 0 0 は、任意選択で、個別におよび組み合わせて、本明細書に記載の特徴、機能、および詳細のいずれかによって補足され得ることに留意されたい。

#### 【 0 3 0 3 】

##### 1 0 . 図 1 0 a および 1 0 b による間隔サイズの決定

図 1 0 a は、例えば、間隔サイズ決定コアで使用することができる、間隔サイズ情報を決定するための概念の概略図を示している。

#### 【 0 3 0 4 】

見てわかるように、図 1 0 a の概念は、状態変数値 9 1 0 に対応することができ、状態変数値を更新することができる複数の状態変数値 1 0 1 0 a 、 1 0 1 0 b を受け取る。さらに、概念 1 0 0 0 は、間隔サイズ情報 1 0 2 0 を提供し、これは、間隔サイズ情報 9 2 0 に対応し得る。概念 1 0 0 0 は、第 1 の状態変数値 1 0 1 0 a ( またはそのスケーリングおよび／または丸められたバージョン ) を第 1 の確率値 9 4 2 に対応し得る第 1 の確率値 1 0 4 2 にマッピングする第 1 のマッピング 1 0 4 0 を備える。さらに、概念 1 0 0 0 は、第 2 の状態変数値 1 0 1 0 b ( またはそのスケーリングおよび／または丸められたバージョン ) を第 2 の確率値 1 0 5 2 にマッピングする第 2 のマッピング 1 0 5 0 を含む。さらに、概念 1 0 0 0 は、第 1 の確率値 1 0 4 2 と第 2 の確率値 1 0 5 2 との結合を含み、この結合は、例えば、式 ( 7 ) を使用して実行され得る。したがって、結合 1 0 6 0 は、結合確率値 1 0 6 2 を提供する。さらに、概念 1 0 0 0 は、結合確率値 1 0 6 2 の乗算 1 0 7 0 を含み、乗算は、コーディング間隔サイズ値 R で行うことができる。したがって、間隔サイズ情報 1 0 2 0 は、結合確率値 1 0 6 2 a およびコーディング間隔サイズ値 R 間の乗算 1 0 7 0 に基づいて得られる。間隔サイズ情報 1 0 2 0 は、例えば、可能性の低いシンボルまたは可能性の最も低いシンボルに関連付けられているコーディング間隔のサイズ ( または、あるいは、より可能性の高いシンボルまたは最も可能性の高いシンボルに関連付けられたコーディング間隔のサイズ ) を記述し得る。任意選択で、概念 1 0 0 0 は、可能性の低いシンボルまたは可能性の低いシンボルの確率を導出するためのメカニズムを含み得 ( 例えば、結合 1 0 6 0 と乗算 1 0 7 0 の間 ) 、または乗算 1 0 7 0 の結果に基づいて、より可能性の低いシンボルまたは最も可能性の低いシンボルに関連する間隔サイズ情報を導出するためのメカニズムを含み得る。

20

したがって、図 1 0 a による概念 1 0 0 0 は、図 9 に関して説明したような機能を実装することができる。

30

#### 【 0 3 0 5 】

概念 1 0 0 0 は、本明細書に開示される算術エンコーダまたは算術デコーダのいずれかで使用できることに留意されたい。

また、概念 1 0 0 0 は、任意選択で、個別におよび組み合わせて、本明細書に開示される特徴、機能、および詳細のいずれかによって補足され得る。

#### 【 0 3 0 6 】

図 1 0 b は、第 1 の状態変数値 1 0 8 2 a および第 2 の状態変数値 1 0 8 2 b に基づい

40

50

て間隔サイズ情報 1084 を提供するための概念 1080 のブロック概略図を示す。概念 1080 は、第 1 の状態変数値 1082a、またはそのスケーリングおよび / または量子化されたバージョンの第 1 の間隔サイズ寄与 1086a に対するマッピング 1085a を含む。さらに、概念 1080 は、第 2 の状態変数値 1082b、またはそのスケーリングおよび / または量子化されたバージョンを第 2 の間隔サイズ寄与 1086b にマッピングする、第 2 のマッピング 1085b を含む。

#### 【0307】

第 1 のマッピング 1085a および第 2 のマッピング 1085b は、例えば、2 次元ルックアップテーブルを使用して、該マッピングを実行することができる。2 次元ルックアップテーブルは、例えば、コーディング間隔サイズ情報 R の量子化された値と、R の複数の異なる値に対する一次元ルックアップテーブル（例えば、LUT1）との間の積を定義することができる。言い換えれば、マッピング 1085a で使用される 2 次元ルックアップテーブルの各行または各列は、本明細書で説明されるルックアップテーブル LUT1 と、該行または列に関連するそれぞれのコーディング間隔サイズ値との間の積を定義することができる。同じことが、第 2 のマッピング 1085b で使用される 2 次元ルックアップテーブルにも当てはまり得、該ルックアップテーブルは、第 1 のマッピング 1085a で使用されるルックアップテーブルと同一であり得るか、または該ルックアップテーブルとは異なり得る。さらに、コーディング間隔サイズ情報は、第 1 のマッピング 1085a および第 2 のマッピング 1085b で使用されて、2 次元ルックアップテーブルの適切な要素を選択することができることに留意されたい（例えば、第 1 のルックアップテーブルインデックスは、それぞれの状態変数値に基づくことができ、第 2 のルックアップテーブルインデックスは、コーディング間隔サイズ値（またはその量子化バージョン Qr (R) に基づくことができる）。

10

#### 【0308】

したがって、間隔サイズ情報 1084 は、非常に低い計算の複雑さで取得することができる。さらに、マッピング 1085a、1085b と、間隔サイズ寄与 1086a、1086b が結合された結合 1088 との間に適切な処理を挿入して、可能性の低いシンボルまたは可能性の最も低いシンボルに関連する間隔の間隔サイズへの寄与を決定することに留意されたい。あるいは、しかし、後処理は、間隔サイズ寄与 1086a、1086b の結合 1088 の結果に基づいて、最も可能性の低いシンボルまたはより可能性の低いシンボルの間隔サイズを決定するために、結合 1088 の後に任意選択で挿入され得る。

20

#### 【0309】

結論として、図 10b による概念 1080 を使用して、状態変数値 1082a、1082b に基づいて、またコーディング間隔サイズ情報またはコーディング間隔サイズ値に依存して、間隔サイズ情報 1086 を導出することができる。

30

さらに、概念 1080 は、本明細書に開示される算術エンコーダまたは算術デコーダのいずれかで使用できることに留意されたい。

さらに、図 10b の概念 1080 は、本明細書に開示される特徴、機能、および詳細のいずれかによって、個別におよび組み合わせて、任意に補足することができることに留意されたい。

40

#### 【0310】

##### 11. 図 11 による間隔サイズ決定コア

図 11 は、本発明の実施形態による、間隔サイズ決定コア 1100 のブロック概略図を示す。

#### 【0311】

間隔サイズ決定コアは、複数の（通常は更新される）状態変数値 1110 を受け取り、それに基づいて、間隔サイズ情報 1120 を提供する。一般的に言えば、間隔サイズ決定コア 1100 は、複数の状態変数値 1110 に基づいて、符号化される 1 つまたは複数のシンボルの算術符号化（または復号される 1 つまたは複数のシンボル値の算術復号）の間隔サイズ情報を導出するように構成され、これは、異なる適合時定数で以前に符号化され

50

た（または、以前に復号された）複数のシンボル値の統計を表す。間隔サイズ決定コア 1100 は、複数の状態変数値 1110 に基づいて結合状態変数値 1132 を導出するよう構成される結合 / コンバイナ 1130 を備える。任意選択で、間隔サイズ決定コア 1100 は、スケーリングおよび / または丸められた結合状態変数値 1142 を得るために、結合状態変数値 1132 をスケーリングおよび / または丸める、スケーリングおよび / または丸め 1140 を備える。さらに、間隔サイズ決定コア 1100 は、ルックアップテーブルに基づくマッピング 1150 を含み、これは、ルックアップテーブルを使用して、結合状態変数値 1132、またはそのスケーリングおよび / または丸められたバージョン 1142 をマッピングし、算術符号化 / 復号のため、間隔サイズ（例えば、重要度の低いシンボルまたは重要度の高いシンボルなどの特定のシンボルに関連付けられた間隔のサイズ）を説明する間隔サイズ情報を取得するように構成される。しかし、任意選択で、後処理 1160 を使用して、ルックアップテーブルベースのマッピング 1150 の結果 1152 に基づいて間隔サイズ情報 1120 を導出することができる。

### 【0312】

間隔サイズ決定コア 1100 は、本明細書に開示される算術エンコーダおよび算術デコーダのいずれかで使用できることに留意されたい。さらに、間隔サイズ決定コア 11 は、本明細書に開示される特徴、機能、および詳細のいずれかによって任意選択で補足され得ることに留意されたい。

### 【0313】

#### 12. 図 12 による状態変数の更新

図 12 は、本発明の実施形態による、状態変数更新のプロック概略図を示す。状態変数更新 1200（これは、状態変数アップデータと見なされてもよい）は、シンボル値 1210 を受信するように構成され、これは、符号化されるシンボルのシンボル値、または以前に復号されたシンボルのシンボル値（または以前に符号化されたシンボルのシンボル値）であり得る。さらに、状態変数更新 1200 は、1 つまたは複数の以前に決定された状態変数値 1212 を受け取り、1 つまたは複数の更新された状態変数値 1220 を提供するように構成され得る。特に、状態変数更新 1200 は、シンボル値に依存して（例えば、符号化されるシンボルまたは以前に符号化されたシンボルまたは以前に復号されたシンボルを表す）、ルックアップテーブルを使用して、第 1 の状態変数値を更新するように構成され得る。言い換えれば、状態変数更新 1210 は、ルックアップテーブルベースの状態変数更新またはルックアップテーブルベースの状態変数アップデータ 1230 を含み得る。したがって、更新された状態変数値は、対応する以前に決定された状態変数値に基づいて、符号化されるシンボルまたは以前に符号化されたシンボルまたは以前に復号されたシンボルに基づくシンボル値を考慮して提供され得る。

### 【0314】

したがって、状態変数更新 1200 は、1 つまたは複数の更新された状態変数値を提供することができる。複数の更新された状態変数値が決定される場合、ルックアップテーブルベースの状態変数更新は、各状態変数値に対して個別に実行され得る（ただし、同じメカニズムおよび / または同じルックアップテーブルは、おそらく異なるスケーリングパラメータおよび / または量子化機能とともに使用される可能性がある）。

しかし、本明細書で説明される状態変数更新 1200 は、本明細書で開示される算術エンコーダおよび算術デコーダのいずれかで使用することができる。しかし、状態変数更新 1200 は、任意選択で、本明細書に開示される特徴、機能、および詳細のいずれかによって、個別におよび組み合わせて取られて補足され得ることに留意されたい。

### 【0315】

#### 13. 図 13 による間隔サイズ決定コア

図 13 は、本発明の実施形態による、間隔サイズ決定コア 1300 のプロック概略図を示す。

### 【0316】

間隔サイズ決定コアは、1 つまたは複数の状態変数値 1310 を受け取り、これは、例

えば、更新された状態変数値であり得る。さらに、間隔サイズ決定コアは、1つまたは複数の状態変数値1310に基づいて、間隔サイズ値1320を提供する。

間隔サイズ決定コアは、ルックアップテーブル評価メカニズム1330を含み、これは、1つまたは複数の状態変数値1310に基づくことができ、間隔サイズ値1320を提供する確率指数1332を受け取る。

#### 【0317】

例えば、確率指数1332は、確率指数導出1340（これは、スケーリングおよび／または丸めおよび／または量子化を含み得る）を使用して、1つまたは複数の（更新された）状態変数値から導出され得る。

例えば、1つまたは複数の更新された状態変数値1310は、本明細書に開示されるように、算術エンコーダまたは算術デコーダで決定され得、複数の以前に符号化されたシンボル値の統計を表すことができる。

#### 【0318】

一般的に言えば、間隔サイズ決定コア1300は、ベースルックアップテーブル1350を使用して間隔サイズ値1320を決定するように構成される。ルックアップテーブル評価メカニズムは、間隔サイズ値1320を決定するように構成され得、1つまたは複数の状態変数値に基づいて取得される確率指数1332が第1の範囲内にある場合、決定された間隔サイズ値がベースルックアップテーブル1350の要素またはベースルックアップテーブル1350の要素の丸められたバージョンと同一であるようにし、確率指数1332が第2の範囲内にある場合、ベースルックアップテーブル1350の要素のスケーリング1360（および任意選択で丸め）を使用して、決定された間隔サイズ値が取得されるようにする。したがって、間隔サイズ値1320を使用して、1つまたは複数のシンボルの算術符号化または算術復号を実行することができる。

#### 【0319】

言い換えれば、例えば、図8に示されるように、1つまたは複数の（更新された）状態変数値から導出され得る確率指数1332（第1の信号処理バス880または第2の信号処理バス890、任意選択の決定840および量子化850との組み合わせ）は、ベースルックアップテーブルのエントリが「そのまま」使用されるか、またはスケーリング1360で使用されるかを決定するために使用され得る（確率指数1332によって決定され得る）。

#### 【0320】

言い換えれば、確率指数1332は、参照符号1370に概略的に示されているように、ベースルックアップテーブル1350のどの要素が選択されるか、およびベースルックアップテーブル1350のこの選択されたエントリのスケーリングが、参照符号1380に象徴的に示されるように、実行されるかどうかの両方を決定することができる。例えば、ベースルックアップテーブル1370のエントリの選択は、確率指数1332の1つまたは複数の最下位ビットによって決定され得、スケーリング1360が実行されるかどうかの決定は、確率指数1332の1つまたは複数の最上位ビットに基づいてなされ得る。しかし、ビットのグループを評価するのではなく、ベースルックアップテーブル1350のエントリを選択することは、所定の値を有する確率指数1332の除算の除算残差によって決定され得、スケーリング1360を実施するか否かを判断することは、複数の範囲のうち、現在考えられている確率指数値がどの範囲にあるかを判断することにより、なされ得る。

#### 【0321】

さらに、任意選択で、例えば、ベースルックアップテーブルが、2次元テーブルである場合、ベースルックアップテーブル1350のエントリの選択において、コーディング間隔サイズ情報（例えば、R）が考慮され得ることに留意されたい。別の任意選択の代替案では、コーディング間隔サイズ情報（例えば、R）を使用して、間隔サイズ値1320を取得するために、コーディング間隔サイズ情報に依存する追加のスケーリングを実行する必要があるかどうかを決定することができる（例えば、ベースルックアップテーブルの選

10

20

30

40

50

択されたエントリに適用される)。

【0322】

しかし、ここで説明する間隔サイズ決定コア1300は、間隔サイズ値を導出するため、本明細書で開示される算術エンコーダおよび算術デコーダのいずれかで使用できることに留意されたい。

さらに、間隔サイズ決定コア1300は、任意選択で、本明細書に開示される特徴、機能、および詳細のいずれかによって補足され得ることに留意されたい。

【0323】

14. 間隔サイズ決定コア

図14は、本発明の実施形態による、間隔サイズ決定コア1400のブロック概略図を示す。

10

【0324】

間隔サイズ決定コア1400は、1つまたは複数の状態変数値1410を受け取り、これは、例えば、更新された状態変数値であり得る。さらに、間隔サイズ決定コア1400は、1つまたは複数の状態変数値1410に基づいて、間隔サイズ値1420を提供する。

間隔サイズ決定コア1400は、ルックアップテーブル評価メカニズム1430を含み、これは、1つまたは複数の状態変数値1410に基づくことができ、間隔サイズ情報または間隔サイズ値1420を提供する確率指数1432を受け取る。

【0325】

例えば、確率指数1432は、1つまたは複数(確率指数導出1440を使用して更新された状態変数値)から導出され得る。

20

例えば、1つまたは複数の更新された状態変数値1410は、本明細書に開示されるように、算術エンコーダまたは算術デコーダで決定され得、複数の以前に符号化されたシンボル値および/または複数の以前に復号されたシンボル値の統計を表すことができる。

【0326】

一般的に言えば、間隔サイズ決定コア1400は、1つまたは複数の状態変数値1410から導出された確率値(例えば、確率指数1432)に基づいて、およびコーディング間隔サイズに基づいて(例えば、コーディング間隔サイズ情報1412によって記述される)、確率テーブル(例えば、ProbTabLPS)を使用して間隔サイズ値1420を決定するように構成される。例えば、確率テーブル1450は、複数の異なる確率値(または確率指数1432)のセットおよび所与のコーディング間隔サイズ(例えば、単一の所与の参照コーディング間隔サイズ)の間隔サイズを説明する。図14に見られるように、確率指数1432は、例えば、確率テーブルのどの要素を間隔サイズ値1420の提供の基礎として使用すべきか選択するために使用され得る。別の言い方をすれば、確率指数1432は、さらなる処理のための確率テーブルの要素の選択を決定する。さらに、確率指数1432が決定され、その値によって、確率テーブルの選択されたエントリがスケーリング1460でスケーリングされる。一般的に言えば、間隔サイズ決定コアは、現在の確率値が複数の確率値のセットでない場合、および/または現在のコーディング間隔サイズが所与の(参照)コーディング間隔サイズと異なる場合、確率テーブルの要素またはエントリ(例えば、現在の確率値または確率指数1432に依存して選択された要素)をスケーリングするように構成され得る。言い換えれば、間隔サイズ値1420は、スケーリング1460を使用して確率テーブル1450の選択されたエントリから導出され得、スケーリングは、確率値または確率指数1432およびコーディング間隔サイズ情報1412の両方に依存し得る。

30

【0327】

例えば、確率テーブル1450のエントリは、単一のコーディング間隔サイズと、所与の範囲の確率値または確率指数(この範囲は、通常、3つ以上の異なる確率値または確率指数をカバーし得、確率テーブルのエントリの数は、計算操作を容易にするために、好ましくは2の効力であり得る)に関連付けられ得る。スケーリング1460は、例えば、コーディング間隔サイズ1412が、確率テーブル1450のエントリが関連付けられてい

40

50

る参照コーディング間隔サイズと異なる場合、確率テーブル 1460 の選択されたエントリをスケーリングすることができる。言い換えれば、実際のコーディング間隔サイズ 1412 が参照コーディング間隔サイズ（例えば、確率テーブル 1450 のエントリが関連付けられるコーディング間隔サイズ）と異なる場合、スケーリング 1460 は、この偏差を考慮し得る（少なくとも、参照コーディング間隔サイズと実際のコーディング間隔サイズ 1410 との間の偏差は、コーディング間隔サイズを量子化するために使用される量子化間隔のサイズを超える）。さらに、スケーリング 1460 はまた、確率値または確率指数 1432 が、確率テーブル 1450 のエントリが関連付けられている確率値または確率指数の範囲外にあるかどうかを考慮し得る。例えば、確率テーブル 1450 のエントリが確率指数の第 1 の範囲に関連付けられているが、実際の確率指数 1432 が、確率指数の第 1 の範囲と重複しない指数の第 2 の範囲内にある場合、スケーリング 1460 はこの発見を考慮して追加のスケーリングを適用することができる（実際のコーディング間隔サイズの参照コーディング間隔サイズからの偏差に基づくスケーリングに加えて）。

#### 【0328】

結論として、間隔サイズ決定コア 1400 は、確率値または確率指数 1432 が複数の確率値のセット内にある場合、および、実際のコーディング間隔サイズ 1412 が参照コーディング間隔サイズに等しい（または参照コーディング間隔サイズに等しい値に量子化される）場合、確率テーブル 1450 の選択されたエントリを間隔サイズ値 1420 として提供することができ、そのための確率テーブルエントリが提供される。他方、確率値または確率指数 1432 が、確率テーブルエントリが提供される「複数の確率値のセット」の外側にある場合、および / または実際のコーディング間隔サイズ 1412 が参照コーディング間隔サイズから逸脱する場合（例えば、実際のコーディング間隔サイズが異なる値に量子化される分だけ）、スケーリング 1460 は、間隔サイズ値 1420 を得るために、確率テーブル 1450 の選択されたエントリをスケーリングする。

#### 【0329】

したがって、間隔サイズ値 1420 は、比較的小さいルックアップテーブル 1450 を使用して取得することができ、これは、例えば、エントリの数がより少ない（例えば、少なくとも 2 倍）1 次元確率テーブルであり得、次に、いくつかの異なる可能な確率値または確率指数値であり得る。

しかし、間隔サイズ決定コア 1400 は、任意選択で、本明細書に記載の特徴、機能、および詳細のいずれかによって補足され得ることに留意されたい。さらに、間隔サイズ決定コア 1400 は、本明細書に開示される算術エンコーダまたは算術デコーダのいずれか、および本明細書に開示されるビデオエンコーダまたはビデオデコーダのいずれかにおいて任意選択で使用できることに留意されたい。

#### 【0330】

##### 15. 図 15 による間隔サイズ決定コア

図 15 は、本発明の実施形態による、間隔サイズ決定コア 1500 のブロック概略図を示す。

#### 【0331】

間隔サイズ決定コア 1500 は、1 つまたは複数の状態変数値を受信するように構成され、これは、例えば、更新された状態変数値であり得る。例えば、1 つまたは複数の状態変数値 1510 は、結合状態変数値を含み得る。さらに、間隔サイズ決定コア 1500 は、間隔サイズ値 1520 を提供するように構成され、これは、例えば、サブインターバル幅値、例えば、R<sub>LPS</sub> であり得る。

#### 【0332】

間隔サイズ決定コア 1500 は、任意選択のマッピングおよび / または丸めを含み、これは、1 つの状態変数値を決定するために使用され、これは、複数の以前に処理された（例えば、以前に符号化または以前に復号された）シンボル値の統計を表す。一般的に言えば、算術エンコーダは、結合状態変数値（例えば、1 つの更新された結合状態変数値 1510 から）から、またはそのスケーリングおよび / または丸められたバージョンから、符

号化または復号されるシンボル値の算術符号化または算術復号のためのサブインターバル幅値（例えば、間隔サイズ値 1520）を計算するよう構成される。間隔サイズ決定コア 1500 は、例えば、1 つの状態変数値（例えば、結合状態変数値）、またはそのスケーリングおよび／または丸められたバージョンを確率値にマッピングするために使用される 1 次元ルックアップテーブル 1550 を備える。例えば、1 次元ルックアップテーブル 1550 のエントリは、結合状態変数値の値ドメインの異なる値間隔の確率値を含む。さらに、間隔サイズ決定コアは、量子化レベルに、算術符号化または算術復号のコーディング間隔のサイズを記述するコーディング間隔サイズ情報 1512 を量子化するように構成される量子化 1560 を備える（例えば、符号化されるシンボル値の算術符号化の前に、または復号されるシンボル値の算術復号の前に）。したがって、量子化されたコーディング間隔サイズ情報 1562 が提供される。さらに、スケーリング／乗算 1570 があり、これは、確率値と量子化レベル（より正確には、量子化されたコーディング間隔サイズ情報 1562）との間の積を決定する（例えば、事前計算された積のルックアップを使用する、または乗算を使用する）。例えば、確率値は、状態変数値またはそのスケーリングおよび／または丸められたバージョンに依存して、1 次元ルックアップテーブル 1550 のエントリを選択することによって提供される。次に、この確率値 1552 は、（量子化された）コーディング間隔サイズ情報 1562 に依存して（すなわち、「量子化レベル」に依存して）スケーリングされ得、スケーリングは、確率値と量子化レベルの間の積の決定に対応し得る。したがって、間隔サイズ値 1520 は、効率的な方法で提供され、1 次元ルックアップテーブルを使用することで十分である。

### 【0333】

しかし、ここで説明される間隔サイズ決定コア 1500 は、任意選択で、個別におよび組み合わせて、本明細書に開示される特徴、機能、および詳細のいずれかによって補足され得ることに留意されたい。さらに、間隔サイズ決定コア 1500 は、本明細書に記載の算術エンコーダおよび算術デコーダのいずれか、また本明細書に記載のビデオエンコーダおよびビデオデコーダのいずれかにおいても使用できることに留意されたい。

### 【0334】

#### 16. 図 16 によるビデオデコーダ

図 16 は、本発明の一実施形態による、ビデオデコーダ 1600 のブロック概略図を示す。

### 【0335】

ビデオデコーダ 1600 は、符号化されたビデオ情報を受信し、それに基づいて、復号されたビデオ情報（または復号されたビデオコンテンツ）を提供するように構成される。

符号化されたビデオ情報 1610（これは、ビデオビットストリームと見なされ得る）は、例えば、スライスタイプの情報を含み得、また、バイナリシーケンスの符号化された表現を含み得る。任意選択で、符号化されたビデオ情報 1610 は、追加の情報を含み得るが、これは、本発明にとって必須ではない。

### 【0336】

一般的に言えば、ビデオデコーダは、複数のビデオフレーム（例えば、ビデオフレームのシーケンス）を復号するように構成され、ビデオデコーダは、例えば、1 つまたは複数のスライスのセットに細分されるビデオフレームを復号するように構成され得る（好ましくは、複数のスライスに）。ビデオデコーダは、例えば、符号化されたビデオ情報 1610 に含まれ、以前のフレームのビデオコンテンツに基づく現在のフレームのビデオコンテンツの予測がない、独立した符号化モードを使用するか、または以前のフレームのピクセルの 1 つのブロックに基づくピクセルのブロックの予測がある単一予測モードを使用するか、または 1 つまたは複数の以前のフレームのピクセルの 2 つ以上のブロックに基づくピクセルのブロックの予測が存在する二重予測モードを使用して、スライスが符号化されているかどうかを示すスライスタイプ情報を評価するように構成され得、スライスの復号のための動作モードを選択するようにする（例えば、「ビデオ再構成」ブロック 1680 によって実行され得る）。

10

20

30

40

50

## 【0337】

ビデオデコーダ 1600 は、符号化されたビデオ情報 1610 に含まれる、バイナリシーケンスの符号化された表現に基づいて（「ビデオ再構成」ブロックによって使用するための）復号されたバイナリシーケンス 1622 を提供するように構成される算術デコーダ 1620 を備える。算術デコーダは、好ましくは、第 1 のソース静的値決定 1630 および第 2 のソース統計値決定 1640 を含む。したがって、算術デコーダ 1620 は、第 1 のウィンドウサイズを使用して（または第 1 の時定数を使用して）第 1 のソース統計値 1632（例えば、第 1 の状態変数値）を決定し（例えば、本明細書に記載の状態変数更新が使用され得る）、第 2 のウィンドウサイズを使用して（または第 2 の時定数を使用して）第 2 のソース統計値 1642（例えば、第 2 の状態変数値）を決定する（例えば、本明細書に記載の状態変数更新を使用することができる）ように構成される。算術デコーダはまた、任意選択でコンバイナ 1650 を備える。したがって、算術デコーダは、第 1 のソース静的値および第 2 のソース統計値に基づいて、結合ソース統計値 1652（例えば、結合状態変数値）を決定するように構成され得る。

10

## 【0338】

さらに、算術デコーダ 1620 は、好ましくは、範囲値決定 1660 を備える（これは、例えば、本明細書に記載の間隔サイズ決定コア、または本明細書に記載の間隔サイズ決定のいずれかを含むことができる）。したがって、算術デコーダは、バイナリシーケンス（符号化されたビデオ情報 1610）の符号化された表現のマッピングに使用される間隔細分割のための 1 つまたは複数の範囲値（または本明細書で説明される間隔サイズ情報）を、結合されたソース統計値 1652 に基づいて（または第 1 および第 2 のソース統計値に基づいて）、復号されたバイナリシーケンス 1622（ビデオ再構成ブロック 1680 によって使用される）に決定するように構成され得る。

20

## 【0339】

好ましくは、算術デコーダ 1620 はまた、範囲値決定 1660 から 1 つまたは複数の範囲値 1662 を受け取り、符号化されたビデオ情報 1610 に含まれる符号化されたバイナリシーケンスから復号されたバイナリシーケンス 1622 を導出する範囲値を使用する算術復号コア 1670（例えば、ブロックまたはユニットであり得る）を備える。

さらに、ビデオデコーダは、例えば、復号されたバイナリシーケンス 1622 を受信し、復号されたバイナリシーケンス 1622 に基づいて復号されたビデオコンテンツ 1612 を提供する（おそらくスライスタイルの情報などの追加の制御情報を考慮して）ビデオ再構成ブロック（またはユニット）1680 を含み得る。

30

## 【0340】

結論として、算術デコーダ 1600 は、符号化されたビデオ情報 1610 を受信し、バイナリシーケンスの符号化された表現の算術復号を実行して、復号されたバイナリシーケンス 1622 を導出する。算術復号は、復号されたバイナリシーケンス 1622 におけるバイナリ値の確率に関する知識を利用する。復号されたバイナリシーケンス 1622 内のバイナリ値の確率（または推定確率）に関するこの知識は、間隔の細分割を定義する範囲値 1662 に依存することによって、算術復号コア 1670 によって考慮される。簡単に言えば、算術復号コアは、範囲値 1662 を使用して、異なる間隔を定義することができる（例えば、0 から 1 の間、または整数値の範囲にわたって）。算術復号コアは、例えば、バイナリシーケンスの符号化された表現を、範囲値を使用して定義された間隔の 1 つにある数値の表現として解釈できる。バイナリシーケンスの符号化された表現によって表される数がどの間隔にあるかを認識することによって、算術復号コア 1670 は、バイナリシーケンスの符号化された表現を使用してどのビットまたはどのビットシーケンスが符号化されたかを結論付けることができる。

40

## 【0341】

しかし、算術復号コア 1670 のこの説明は、非常に簡潔で一般的な説明としてのみ考慮されるべきであることに留意されたい。算術復号コアに関する詳細は、例えば、規格 H . 264 および H . 265 で確認できる。しかし、（算術復号コアの動作のための）異な

50

る概念も文献に見ることができ、算術復号コアの詳細は、本発明にとって特に関連性がない。

#### 【0342】

しかし、（高いピットレート効率を可能にする）適切な範囲値を得るために、算術デコーダ1620（または、一般的に言えば、ビデオデコーダ）は、異なるウィンドウサイズを使用して2つのソース統計値1632、1642を決定する（「ウィンドウサイズ」は、復号されたバイナリシーケンス1622の複数の復号されたバイナリ値にわたる平滑化の程度を定義する）。さらに、算術復号コア1670に提供される範囲値の信頼性を高めるために、第1のソース統計値1632および第2のソース統計値1642は、いくつかの実施形態では、結合されたソース統計値1652に組み合わされる。

10

#### 【0343】

したがって、算術復号コア1670によって使用される範囲値は、ピット値の実際の確率（例えば、復号されたバイナリシーケンス1622内のピット値「0」および「1」のもの）によく適合しているので、ビデオデコーダ1600は、高効率を提供すると言うことができる。

#### 【0344】

補足として、ビデオデコーダ1600も変更可能であることに留意されたい。代替の実装形態では、第2のソース統計値決定1640は、固定値（復号されたバイナリシーケンスから独立していてもよいが、1つまたは複数のパラメータに依存し得る）の提供によって置き換えることができる。この場合、算術デコーダは、結合されたソース統計値1652を取得するために、第1のソース統計値1632を固定の非ゼロ値と結合するように任意選択で構成される。そのような単純化は、場合によっては良い結果をもたらし、例えば、結合されたソース統計値の不適切に強い変動を回避する可能性があることがわかっている。言い換えれば、結合されたソース統計値の決定に固定寄与を導入することによって、結合されたソース統計値がこの固定値から大きく逸脱することがもはやできなくなることを達成することができる。したがって、復号されたバイナリシーケンス1622内に同一のピット値のより長いシーケンスが偶然に存在する場合、復号されたバイナリシーケンスの統計に対するいくつかの「後知恵」を使用して、コーディング効率の大幅な低下を回避することができる。

20

#### 【0345】

補足として、算術デコーダ（および算術デコーダの個々のブロック）の機能は、一般に、ビデオデコーダ全体の機能と見なすこともできることに留意されたい。言い換えれば、算術デコーダの機能として本明細書に記載されている機能は、ビデオデコーダの他のブロックによっても実行することができる。

30

さらに、図1によるビデオデコーダ1600は、個別におよび組み合わせて、本明細書に記載の特徴、機能、および詳細のいずれかによって補足され得ることに留意されたい。

#### 【0346】

##### 17. 図17によるビデオデコーダ

図17は、本発明の実施形態による、ビデオデコーダ1700のブロック概略図を示す。ビデオデコーダ1700は、符号化されたビデオ情報1710（例えば、ビデオピットストリーム）を受信し、それに基づいて、復号されたビデオコンテンツ1712（例えば、ビデオフレームのシーケンス）を提供するように構成される。符号化されたビデオ情報1710は、例えば、本明細書で説明されるように、スライスタイル情報を持み得る。符号化されたビデオ情報1710は、構成情報をさらに含み得、これはまた、制御情報と見なすことができる。さらに、符号化されたビデオ情報1710は、バイナリシーケンスの符号化された表現を含み得る。

40

#### 【0347】

図17には、ビデオデコーダ1700の2つの主要なブロック、すなわち、算術デコーダ1720およびビデオ再構成1780が示されている。ただし、ビデオデコーダの機能の分布は、固定ブロック構造に拘束されるのではなく、広範囲にわたって変更される可能

50

性があることに留意されたい。また、ビデオデコーダの実際の実装は、当業者によく知られている追加のブロックおよび / または機能を有する可能性があることに留意されたい。

#### 【 0 3 4 8 】

算術デコーダ 1 7 2 0 は、バイナリシーケンスの符号化された表現 1 7 1 1 を受信する。しかし、算術デコーダ（または算術デコーダの外側にあり得る制御ブロック）は、任意選択で、スライスタイプ情報および構成情報（または制御情報）を受信することができる。特に、算術デコーダ 1 7 2 0 は、任意選択でスライスタイプ情報および構成情報または制御情報の一部またはすべてを考慮に入れて、バイナリシーケンスの符号化表現 1 7 1 1 に基づいて、復号されたバイナリシーケンス 1 7 2 2 をビデオ再構成 1 7 8 0 に提供する。

#### 【 0 3 4 9 】

以下では、算術デコーダ 1 7 2 0 の機能についてより詳細に説明する。算術デコーダは、算術復号コア 1 7 7 0 を含み、これは、バイナリシーケンスの符号化された表現 1 7 1 1 を受信し、復号されたバイナリシーケンス 1 7 2 2 を提供する。算術復号コアは、復号されたバイナリシーケンス 1 7 2 2 のどのビット値が、バイナリシーケンスの符号化された表現 1 7 1 1 によって表されるかを決定する。この目的のために、算術復号コア 1 7 7 0 は、バイナリシーケンスの符号化された表現 1 7 1 1 によって表されるどの数値の範囲の間隔にあるかを確認する。複数の（少なくとも 2 つの）間隔のうち、バイナリシーケンスの符号化された表現 1 7 1 1 によって表される数が存在する間隔の決定に応じて、復号されたバイナリシーケンス 1 7 2 2 の特定のビット値、またはビット値のグループ、またはシンボル、が認識される。

#### 【 0 3 5 0 】

復号されたバイナリシーケンス 1 7 2 2 を導出する目的で、算術復号コアは、間隔に関する情報を受信し、これは通常、ビット値の確率にある程度対応する。この場合、算術復号コア 1 7 7 0 は、間隔細分割に使用される「範囲値」または「間隔サイズ情報」 1 7 6 2（すなわち、算術復号コア 1 7 7 0 によって使用される数範囲の間隔を定義するのに役立つ範囲値 1 7 6 2）を受け取る。特に、算術復号コア 1 7 7 0 は、例えば、規格 H . 2 6 4 によるビデオエンコーダ / デコーダまたは規格 H . 2 6 5 によるビデオエンコーダ / デコーダで使用される算術復号コアと類似または同一であり得ることに留意されたい。しかし、算術復号コア 1 7 7 0 を実現するための異なるアプローチも使用できることに留意されたい。

#### 【 0 3 5 1 】

上記の議論を考慮すると、算術復号コア 1 7 7 0 の間隔細分割を定義する範囲値または間隔サイズ情報 1 7 6 2 を提供することが算術デコーダ 1 7 2 0 の重要な機能であることが明らかになる。一般的に言えば、算術デコーダ 1 7 2 0 は、復号されたバイナリシーケンス 1 7 2 2 の以前に復号されたバイナリ値からこれらの範囲値 1 7 6 2 を導出し、任意選択で、例えば、初期化値、「ウィンドウサイズ」、「ウィンドウサイズ適合」などのパラメータを定義するいくつかの制御情報を考慮に入れる。

#### 【 0 3 5 2 】

算術デコーダ 1 7 0 0 では、2 つのソース統計値決定ブロック（またはユニット） 1 7 3 0 、 1 7 4 0 が使用される（これは、例えば、本明細書で説明される状態変数の更新に対応し得る）。例えば、第 1 のソース統計値決定ブロック 1 7 3 0 は、復号されたバイナリシーケンス 1 7 2 2 ( $\times_t$  でも示される) の 1 つまたは複数の以前に復号されたバイナリ値（またはシンボル値）を受け取り、それに基づいて、第 1 のソース統計値 1 7 3 2（これは、本明細書で説明される第 1 の状態変数値に対応し得る）を提供する。第 1 のソース統計値決定ブロックは、例えば、ソース統計値 1 7 3 2 を表すために使用されるビット数を定義する定数または変数  $B_{ITSA}$ 、およびソース統計値決定ブロック 1 7 3 0 が使用するために「ウィンドウサイズ」を定義する定数または変数  $n_a$ 、および / または本明細書に記載されている他の任意のパラメータなどのようないくつかの構成情報を受け取ることができる。例えば、第 1 のソース統計値決定ブロック 1 7 3 0 は、第 1 のソース統計値 1 7 3 2 を再帰的に決定することができ、ウィンドウサイズ  $n_a$  またはパラメータ  $n_i^k$

10

20

30

40

50

は、第1のソース統計値または第1の状態変数値1732の決定において、復号されたバイナリシーケンス1722の最後に復号されたバイナリ値の重み付けを決定する。

#### 【0353】

第1のソース統計値決定ブロック1730の機能は、例えば、いくつかの「無限インパルス応答」特性を導入する再帰的アルゴリズムが使用されるという事実を除いて、特定のウィンドウサイズを有するスライディングする平均の形成に類似している。したがって、第1のソース統計値1732は、スライディングウィンドウ合計操作またはスライディングウィンドウ平均化操作の結果を正確に表すのではなく、結果が非常に類似しているため、「仮想スライディングウィンドウ」操作と見なす必要がある。

#### 【0354】

さらに、第2のソース静的値決定ブロック1740は、第1のソース統計値決定ブロック1730と比較した場合、同様の操作を実行する。しかし、第2のソース統計値決定ブロック1740は、通常、異なるパラメータ（例えば、異なるウィンドウ長 $n_b$ および/または異なるビット数パラメータBIT $S_b$ または異なるパラメータ $n_i^k$ ）を使用し、その結果、第2のソース統計値または第2の状態変数値1742を提供し、これは通常、第1のソース統計値または第1の状態変数値1732とは異なる。例えば、ソース統計値1732、1742の1つは、短期（または短めの期間）平均ソース統計値であり得、ソース統計値1732、1742の1つは、長期（または長めの期間）の平均ソース統計値であり得る。

#### 【0355】

ソース統計値決定ブロック1730、1740は、例えば、本明細書の状態変数更新について説明したような機能を実行することができることに留意されたい。さらに、いくつかの実施形態では、ソース統計値決定ブロック1730、1740において異なる計算規則を使用することもできることに留意されたい。

#### 【0356】

算術デコーダ1720は、任意選択で、第1のソース統計値1732および第2のソース統計値1742を受信するように構成された、結合されたソース統計値決定ブロック（またはユニット）250をさらに備える。ソース統計値結合ブロック1750は、それに基づいて、結合されたソース統計値1752を提供する。例えば、ソース統計値結合ブロック1750は、第1のソース統計値1732および第2のソース統計値1742の合計または平均を形成し、それによって、結合されたソース統計値1752を取得することができる。

しかし、ソース統計値結合ブロック1750は、結合されたソース統計値1752を導出するときに、第1のソース統計値1732および第2のソース統計値1742に異なる重みを適用することもでき、異なる重みは、スライス内または異なるスライス間でさえ変化し得る。

#### 【0357】

例えば、ソース統計値結合ブロック1750は、結合751または結合892の機能を実行することができる。ただし、この機能のバリエーションも可能である。

例えば、1つの（代替の）実施形態では、ソース統計値結合ブロック1750は、第1の統計値のうちの1つのみを固定値と結合し、それによって、結合されたソース統計値または結合状態変数値1752を取得する。そのような概念は、結合されたソース統計値1752が、復号されたバイナリシーケンス1722内のバイナリ値の期待される確率から大きく逸脱することを回避するために有利である可能性がある。

#### 【0358】

算術デコーダ1720は、結合されたソース統計値または結合状態変数値1752に基づいて、または代わりに、2つ以上の状態変数値に基づいて、間隔細分割（算術復号コア1770に提供される）の範囲値1762を導出するように構成される。

この処理ステップは、例えば、「範囲値の決定」と見なすことができる。例えば、範囲値の決定は、結合されたソース統計値1752を受け取り、それに基づいて、確率値また

10

20

30

40

50

は状態指數値を提供する、任意選択の値処理 1766 を含み得る。値処理 1766 は、例えば、結合されたソース統計値 1752 の値の範囲を、0 から 1 の間の範囲、0 から 0.5 の間の範囲、または整数の指數値にマッピングすることができる。

#### 【0359】

任意選択で、値処理 1766 は、情報 1767 を提供することができ、これは、次の復号された値（例えば、復号されたバイナリシーケンス 1722 の）が「1」の値をとるか、または「0」の値をとる可能性が高いかどうかを示すバイナリ情報であり得る。任意選択で、算術デコーダ（または範囲値決定）は、マッピングテーブル 1769 を含み得る。マッピングテーブル 1769 は、例えば、テーブルエントリを指定する指數値（例えば、pStateIdx）を受け取ることができる。したがって、マッピングテーブル 1769 は、指數値（例えば、pStateIdx）によって指定された該テーブルエントリに対応する 1 つまたは複数の範囲値 1762 を提供することができる。したがって、「状態指數値」（例えば、pStateIdx）を導出し、状態指數値に基づいてマッピングテーブルを評価することによって、間隔細分割のための 1 つまたは複数の範囲値が、結合されたソース静的値 1752 に基づいて提供され得る。

#### 【0360】

マッピングテーブル 1769 は、例えば、規格 H.264 または規格 H.265 で説明されているマッピングテーブルと同じ構造を有することができる。しかし、任意選択で、マッピングテーブルのコンテンツは、ビデオデコーダの特定の詳細に適合させることができる。特に、マッピングテーブルのエントリは、特定のビデオデコーダで期待される統計プロパティに適合させることができる。

#### 【0361】

しかし、1 つまたは複数の状態変数値に基づく、または結合状態変数値に基づく間隔サイズ情報の導出は、本明細書に開示される概念のいずれかに従って実行され得ることに留意されたい。例えば、間隔サイズ決定を使用するか、間隔サイズ決定コアを使用するか、または図 4 から図 15 のいずれかを参照して説明された状態変数更新を使用することが挙げられる。

また、本明細書で言及される算術復号コアは、本明細書で開示される算術復号コアのいずれかに対応し得ることに留意されたい。

#### 【0362】

算術デコーダ（または、一般的に言えば、ビデオデコーダ）はまた、任意選択で、制御ブロック 1790 を含み、これは制御情報または構成情報を受信でき、それに基づいて範囲値の提供に使用されるパラメータ（および場合によっては他のパラメータも、例えば算術復号コア 1770 によって使用される追加のパラメータ）を調整できる。例えば、制御ブロック 1790 は、符号化されたビデオ情報 1710 に含まれ得る、「cabac\_in\_it\_flag」、「ws\_flag」、および「ctu\_ws\_flag」という 1 つまたは複数のスライスタイプ情報を受信することができる。

#### 【0363】

さらに、制御 1790 は、例えば、制御情報に応じて、ウィンドウサイズパラメータ  $n_a$ 、 $n_b$  または  $n_i^k$  および / またはビットサイズパラメータ  $BITS_a$ 、 $BITS_b$  を調整することができる。特に、制御ブロック 1790 はまた、現在のコンテキストモデルを考慮し得る。この問題に関して、復号される復号されたバイナリシーケンス 1722 の各ビット（またはビットのグループ）について、どのコンテキストモデルを使用するかを決定できることに留意されたい。例えば、どのコンテキストモデルを使用するかの決定は、どのタイプの情報（復号パラメータ、変換係数など）がそれぞれのビット（またはビットのグループ）によって表されるかという事実に基づくことができる。例えば、制御ブロック 1790 は、復号されたバイナリシーケンス 1722 の構文を認識するように構成され得、それにより、どの構文要素（または構文要素のどの部分、例えば、最上位ビットまたは最下位ビットなど）が次に復号されるべきかを認識する。したがって、異なるコンテキストモデル間の選択を行うことができる。また、ウィンドウサイズパラメータおよび / ま

10

20

30

40

50

たはビットサイズパラメータおよび／または他の任意のパラメータは、コンテキストモデルに応じて選択され得ることに留意されたい。さらに、ソース統計値 1732、1742 または結合されたソース統計値 1752 は、異なるソース統計値または結合されたソース統計（または状態変数値、または結合状態変数値）値が、様々なコンテキストモデルで使用できるように、特定のコンテキストモデルに関連付けられ得ることにも留意されたい。例えば、特定のコンテキストモデルに関連付けられたソース統計値は、それぞれのコンテキストモデルを使用して復号された復号されたバイナリシーケンス 1722 の復号されたバイナリ値に基づいて選択的に提供され得る。言い換えると、パラメータ  $n_a$ 、 $n_b$ 、 $n_i^k$ 、 $BITS_a$ 、 $BITS_b$  などに関する個別の独立した処理、および個別の（場合によっては独立した）決定を、異なるコンテキストモデルに対して行うことができる。

10

#### 【0364】

制御ユニット 1790 の機能に関して、制御ユニットは、例えば、以下に説明するメカニズムに従って、パラメータ  $n_a$ 、 $n_b$ 、 $n_i^k$ 、 $BITS_a$ 、 $BITS_b$  を決定することができることに留意されたい。例えば、ウィンドウサイズパラメータ  $n_a$ 、 $n_b$ 、 $n_i^k$  は、スライスタイプ情報に依存して、および／または  $c a b a c \_i n i t \_f l a g$  に依存して、および／または  $w s \_f l a g$  に依存して、および／または  $c t u \_w s \_f l a g$  に依存して選択できる。また、ビットサイズパラメータ  $BITS_a$ 、 $BITS_b$  は、いくつかの実施形態におけるいくつかの構成情報に依存して選択され得る。しかし、他の実施形態では、ビットサイズパラメータは固定され得る。パラメータの調整のメカニズムについては、例えば、以下の説明を参照されたい。

20

#### 【0365】

ここでビデオ再構成ブロック 1780 を参照すると、ビデオ再構成ブロック 1780 は、通常、復号されたバイナリシーケンス 1722 と、構成情報の少なくともいくつかの要素も受信することに留意されたい。例えば、ビデオ再構成 1780 は、復号されたバイナリシーケンス 1722 に基づいて、整数パラメータおよび／または浮動小数点パラメータおよび／または画像データを再構成することができる。例えば、復号されたバイナリシーケンスの特定のビットまたは部分をいかに整数パラメータ、浮動小数点パラメータ、または画像データ（例えば、変換係数など）にマッピングするべきかを定義するマッピングルールが存在する場合がある。したがって、ビデオ再構成ブロック 1780 は、復号されたバイナリシーケンス 1722 からビデオフレームの再構成に使用される情報を再構成する。次に、ビデオ再構成ブロックは、再構成された情報（復号されたバイナリシーケンス 1722 から導出された）に基づいて画像情報を生成することができる。

30

#### 【0366】

例えば、ビデオ再構成 1780 は、規格 H.264 または規格 H.265 に記載されているものと同じ機能を備え得る。しかし、復号されたバイナリシーケンス（および、場合によっては、追加の構成情報または制御情報）に基づいて、復号されたビデオコンテンツを提供するように適合された他のアプローチも、ビデオ再構成に使用され得る。したがって、ビデオ再構成 1780 は、復号されたビデオコンテンツ 1712 を提供し、これは、ビデオフレームのシーケンスとしての形態をとることができる。

40

#### 【0367】

結論として、本発明の実施形態によるビデオデコーダの概要が提供された。しかし、機能ブロックには異なる実装があることに留意されたい（例えば、ソース統計値決定ブロック 1730、1740、ソース統計値結合ブロック 1750、値処理ブロック 1766、マッピングテーブル 1769 および算術復号コア 1770 用）。また、ビデオ再構成ブロック 1780 および制御ブロック 1790 に対して異なる実装が可能である。

#### 【0368】

しかし、本明細書に記載の機能ブロックは、本出願内にその全体が開示されている特徴、機能、および詳細のいずれかによって補足され得ることにも留意されたい。また、本出願内で開示される特徴、機能、および詳細は、個別に導入することも、組み合わせて使用することもでき、それによってビデオデコーダ 1700 の機能を拡張できることに留意さ

50

れたい。

【0369】

18. さらなる実施形態

以下では、本発明によるさらなる実施形態について説明する。

他の特徴、機能および詳細の中で、有限状態機械を使用するコンテキストモデル更新方法が説明され、これは、個別に、または本明細書に開示されるすべての特徴、機能および詳細と結合して使用することができる。

【0370】

序論

以下では、異なる本発明の実施形態および態様が、例えば、「状態変数からの確率推定値の導出」および「状態変数の更新」および「好ましい実施形態」の章で説明される。

しかし、他のいずれかの章に記載されている特徴、機能、および詳細はまた、任意選択で、本発明による実施形態に導入することができる。

【0371】

また、上記の章で説明された実施形態は、個別に使用することができ、また、別の章の特徴、機能、および詳細のいずれかによって補足することができる。

また、本明細書に記載の個々の態様は、個別にまたは組み合わせて使用できることに留意されたい。したがって、詳細は、該態様の別の1つに詳細を追加することなく、該個々の態様のそれぞれに追加することができる。

【0372】

特に、実施形態はまた特許請求の範囲に記載されている。請求項に記載の実施形態は、任意選択で、個別におよび組み合わせて、本明細書に記載の特徴、機能、および詳細のいずれかによって補足することができる。

【0373】

本開示は、明示的または暗黙的に、ビデオエンコーダ（入力ビデオ信号の符号化された表現を提供するための装置）およびビデオデコーダ（ビデオ信号の符号化された表現に基づいて、ビデオ信号の復号された表現を提供するための装置）で使用可能な機能を説明することにも留意されたい。したがって、本明細書に記載の特徴のいずれも、ビデオエンコーダのコンテキストおよびビデオデコーダのコンテキストで使用することができる。

【0374】

さらに、方法に関連して本明細書に開示される特徴および機能は、（そのような機能を実行するように構成された）装置においても使用され得る。さらに、装置に関して本明細書に開示される任意の特徴および機能性もまた、対応する方法で使用することができる。言い換えれば、本明細書に開示される方法は、装置に関して説明された特徴および機能のいずれかによって補足され得る。

【0375】

また、本明細書で説明する特徴および機能のいずれも、「実装の代替」のセクションで説明するように、ハードウェアまたはソフトウェアで、あるいはハードウェアとソフトウェアの組み合わせを使用して実装することができる。

【0376】

緒言

コンテキストモデルの更新は、コーディングエンジンを基礎となるソース統計に適合させる可能性を提供することにより、効率的なバイナリ算術エントロピーコーダの重要な機能となる。

【0377】

本発明では、各コンテキストモデルは、例えば、シンボルのシーケンスの各シンボルに対して、エントロピー符号化または復号のための確率推定を提供する独立した確率推定ステージを備えている。このコンテキストモデルの更新方法の基本的な概念は、現在の確率推定を共同で表すコンテキストモデルごとに1つ以上の状態変数を維持することである。シンボルシーケンスの統計に適合するために、コンテキストモデルの状態変数は頻繁に更

10

20

30

40

50

新される（例えば、状態変数の更新）。例えば、各シンボルの符号化後である。

【0378】

本発明は、他の態様の中でもとりわけ、2つの重要な態様を含む。第1の態様は、状態変数から確率推定（または間隔サイズ情報）を導出する方法に対処するものである。第2の態様は、シンボルの発生後に状態変数を更新する方法に対処するものである。態様は、任意選択で結合して使用できる。

【0379】

本発明は、任意選択で、バイナリ算術コーディングエンジンと結合することができるが、符号化または復号されるバイナリシンボルの個々の確率推定を必要とする他のエントロピー符号化エンジンとも結合することができる。例えば、本発明はまた、PIPEコーディング概念と結合することができる。

【0380】

実装の詳細

例えば、

$M \geq 1$

を伴う1つ以上のコンテキストモデル

$c_1, \dots, c_M$

を使用するビデオ圧縮アルゴリズムのようなエントロピーエンコーディングまたはデコーディングアプリケーションについて考察してみる。符号化または復号されるバイナリシンボルのほとんどは、コンテキストモデルの1つに明確に関連付けられている。このようなバイナリシンボルを符号化または復号するために、関連するコンテキストモデル

$c_k$

は、エントロピーコーディングステージに、「1」であるバイナリシンボルの確率推定値

$p_k$

を提供する。良好な圧縮効率を実現するには、バイナリシンボルを符号化するエントロピ

ーによって生成されるビットストリームの部分を、「1」に対しては

$-\log_2 p_k$

に、「0」に対しては

$-\log_2(1 - p_k)$

に近づける必要がある。

【0381】

コンテキストモデル

$c_k$

ごとに、1つ以上の状態変数（または状態変数値）

$s_1^k, \dots, s_N^k$

が

$N \geq 1$

で維持される。各状態変数

$s_i^k$

は、例えば、符号付き整数値として実装され、確率値

10

20

30

40

50

$P(s_i^k, i, k) = p_i^k$

を表す。コンテキストモデル  
 $c_k$

の確率推定値

$p_k$

は、例えば、コンテキストモデルのすべての状態変数の確率値

$p_i^k$

10

の加重和として定義する必要がある。

【0382】

状態変数は、次のプロパティを有することが好ましいが、必ずしもやうするものではない：

1.

$s_i^k = 0$

の場合、

$p_i^k = 0.5$

20

。

2.

$s_i^k$

のより大きい値はより大きい

$p_i^k$

に対応する。

3.  $P(-s_i^k, i, k) = 1 - P(s_i^k, i, k)$

30

。

したがって、負の状態変数は

$p_i^k < 0.5$

に対応する可能性がある。一般に、各コンテキストモデルの状態変数ごとに異なる関数  
 $P(\cdot)$

を指定することができる。

【0383】

40

総論 1

本発明は、多くの状況でビットシフト演算の非常に効率的な使用を可能にする。式  
 $[x \cdot y]$

は、  
 $x$

が整数で

$y$

50

が  $\chi$  の累乗の場合、単純なビットシフト演算で実装できる。これは、

が  $\chi$  の場合、2つの補数表現が

に使用されている場合にも機能する。

### 【0384】

#### 総論 2

例えば確率値、重み付け係数、ルックアップテーブル要素などの非整数を格納するいずれかの変数は、予告なしに固定小数点または浮動小数点演算で実装される場合がある。これは、本発明の特定の実装ごとに異なる可能性がある丸め操作を導入する。

状態変数を確率値に関連付けるための例示的な構成

状態変数を確率値に関連付ける、つまり

$P(\cdot)$

を実装するための多くの便利な方法がある。例えば、HEVCの確率推定量に匹敵する状態表現は、次の式

$$P(\chi, i, k) = \begin{cases} 0.5 \cdot \alpha \cdot \chi \beta_i^k, & \text{if } \chi \geq 0, \\ 1 - 0.5 \cdot \alpha \cdot \chi \beta_i^k, & \text{else.} \end{cases}$$

10

20

は重み付け係数である

で実現できる。

$\beta_i^k$

は重み付け係数である。

$\alpha$

は

$0 < \alpha < 1$

30

を伴うパラメータである。

HEVCに匹敵する構成を実現するには、

$\alpha \approx 0.949217$

および

$\beta_i^k = 1$

をすべての  $i$  および

$k$

40

に対してを設定する。

### 【0385】

この例示的な構成は、状態変数をどのように定義できるかについての何らかの洞察を与えるものとする。以下に示すように、一般的には直接使用されないため、

$P(\cdot)$

を定義する必要はない。代わりに、それはしばしば本発明の個々の部分の実際の実施から生じる。

### 【0386】

50

### 状態変数の初期化

コンテキストモデルを使用して第 1 のシンボルを符号化または復号する前に、すべての状態変数は、圧縮アプリケーションに最適化できる正常な値で任意選択で初期化される。

#### 状態変数からの確率推定値の導出

シンボルの符号化または復号の場合、確率推定値はコンテキストモデルの状態変数から導出される。以下に、2 つの代替アプローチを示す。アプローチ 1 は、アプローチ 2 よりも正確な結果をもたらすが、計算の複雑さも高次になる。

【 0 3 8 7 】

#### アプローチ 1

このアプローチは 2 つのステップからなる。まず、コンテキストモデルの各状態変数  $s_i^k$

10

が確率値

$p_i^k$

に変換される。次に、確率推定値

$p_k$

は、確率値の加重和

$p_i^k$

20

として導出される。

【 0 3 8 8 】

#### ステップ 1 :

##### ルックアップテーブル

$LUT1$

は、例えば式 (1) に従って、状態変数

$s_i^k$

30

を対応する確率値

$p_i^k$

に変換するために使用される。

$$p_i^k = \begin{cases} LUT1[s_i^k \cdot a_i^k], & \text{if } s_i^k \geq 0. \\ 1 - LUT1[-s_i^k \cdot a_i^k], & \text{else.} \end{cases}$$

( 1 )

$LUT1$

40

は、確率値を含むルックアップテーブルである。

$a_i^k$

は、

$s_i^k$

を

$LUT1$

のサイズに適合する重み付け係数である。

50

このステップは、例えば、ブロック 882 および 884 によって実行され得る。

【0389】

ステップ 2 :

確率推定値

$p_k$

は、例えば次のように、確率値

$p_i^k$

から導出される。

$p_k = \sum_{i=1}^N p_i^k \cdot b_i^k$

( 7 )

$b_i^k$

は、個々の状態変数の影響を制御する重み付け係数である。

このステップは、例えば、ブロック 886 によって実行され得る。

【0390】

アプローチ 2

状態変数から確率推定値を導出するための代替アプローチを以下に示す。結果の精度が  
低くなり、計算の複雑さが軽減される。最初に、状態変数の加重和

$s_k$

が、例えば次のように導出される。

$s_k = \sum_{i=1}^N [s_i^k \cdot d_i^k]$

( 10 )

$d_i^k$

は、各状態変数の影響を制御する重み付け係数である。

このステップは、例えば、ブロック 892 によって実行され得る。

次に、確率推定値

$p_k$

は、状態変数の加重和

$s_k$

から導出される。例えば、次のようになる。

$$p_k = \begin{cases} LUT2[[s_k \cdot a_k]], & \text{if } s_k \geq 0, \\ 1 - LUT2[[-s_k \cdot a_k]], & \text{else.} \end{cases}$$

( 2 )

$LUT2$

は、確率推定を含むルックアップテーブルである。

$a_k$

は、

$s_k$

を

10

20

30

40

50

*LUT2*

のサイズに適合する重み付け係数である。

このステップは、例えば、ブロック 8 9 4 によって実行され得る。

【0 3 9 1】

アプローチ 1 および 2 への任意選択の変更

ソフトウェア実装の場合、可能な限り多くの操作をルックアップテーブルにシフトできること、スループットが向上する可能性がある。式 (1) を次のように置き換えると、アプローチ 1 ではこれが簡単になる。

$$p_i^k = \begin{cases} LUT1[\lfloor s_i^k \cdot a_i^k \rfloor], & \text{if } \lfloor s_i^k \cdot a_i^k \rfloor \geq 0, \\ 1 - LUT1[-\lfloor s_i^k \cdot a_i^k \rfloor], & \text{else} \end{cases}$$

10

アプローチ 2 の場合、式 (2) は次のように置き換えられる。

$$p_k = \begin{cases} LUT2[\lfloor s_k \cdot a_k \rfloor], & \text{if } \lfloor s_k \cdot a_k \rfloor \geq 0, \\ 1 - LUT2[-\lfloor s_k \cdot a_k \rfloor], & \text{else} \end{cases}$$

どちらの方程式でも、この変更により、ケースの区別をルックアップテーブルに統合でき、これにより、要素数は以前のほんの 2 倍になる必要がある。ただし、高スループットを実現するには、演算の数を減らすと役立つ場合があるが、ルックアップテーブルのサイズを大きくすると関連性が低くなる場合がある。

さらに通知することなく、アプローチ 1 および 2 の任意の修正を、本発明のアプローチ 1 および 2 のすべての用途に任意選択で適用することができる。

20

【0 3 9 2】

C A B A C (M コーダー) のバイナリ算術コーディングエンジンとの結合 (任意選択)

C A B A C の M コーダーは、2 D ルックアップテーブルを使用して、コーディング間隔の間隔細分割に必要な乗算を置き換える。

*R*

を、バイナリシンボルの算術符号化前のコーディング間隔とする。例えば、

*R*

30

は間隔

[256, 510]

内にあるものとする。符号化されるシンボルは、バイナリ算術符号化または復号の確率推定  $p_k$

を導出するために使用されるコンテキストモデル

$c_k$

に関連付けられている。2 つのサブインターバル

$R \cdot p_k$

40

と

$R \cdot (1 - p_k)$

の幅をそれぞれ導出する必要がある。例えば、C A B A C の M コーダー (例として機能する) は、2 D ルックアップテーブルを使用することでこれらの乗算を回避する。確率推定値  $p_k$

は、シンボルが 1 である推定確率であることを思い返されたい。ただし、M コーダーは、

50

シンボル 0 または 1 の両方の確率が低い確率を使用して、そのサブインターバル幅  
 $R_{LPS}$

を導出する。次に、より可能性の高いシンボル  
 $R_{MPS}$

のサブインターバル幅は、

$R$

から

$R_{LPS}$

10

減算することで簡単に導き出すことができる。

【 0 3 9 3 】

$R_{LPS}$

は M コーダーの実装に応じて、事前定義された最小許容値を下回らない場合があることに留意されたい。例えば、この最小許容値は 2、3、または 4 になり得る。テキストの残りの部分では、

$R_{LPS}$

20

が M コーダーに渡される前に、少なくとも事前定義された最小許容値と同じ大きさになるまで、（あるとしても）1 ずつ繰り返し増加すると想定される。これはクリッピング操作に対応し、個別の計算ステップとして実装できる。以下に示すいくつかのアプローチ（ルックアップテーブル値の調整など）に直接組み込むことができる場合がある。

【 0 3 9 4 】

可能性の低いシンボルの確率

$p_{LPS}$

は

$p_{LPS} = \min(p_k, 1 - p_k)$

30

のように得られる。

例えば、ルックアップテーブル  
 $RangeTabLPS$

に、 $8 \times 3 2$  または  $8 \times 1 6$  の要素を含めることができ、

$R$

の許容値は 8 セルに量子化され、

$p_{LPS}$

40

は 3 2 セルまたは 1 6 セルに量子化される。次に、

$R_{LPS} = RangeTabLPS[Qp(p_{LPS})][Qr(R)]$

$Qp(\cdot)$

$Qr(\cdot)$

50

は、それぞれ

$p_{LPS}$

と

$R$

の量子化関数である。

【 0 3 9 5 】

代替案 1 :

あるいは、ルックアップテーブル  $probTabLPS$  は 3 2 または 1 6 の要素を持つ  
ことができ、 $R$  の許容値は 8 または 1 6 セルに量子化され、

$p_{LPS}$

は 3 2 または 1 6 セルに量子化される。次に、

$R_{LPS} = [probTabLPS[Qp_2(p_{LPS})] * Qr_2(R)]$

( 1 1 )

$Qp_2(\cdot)$

と

$Qr_2(\cdot)$

は、それぞれ

$p_{LPS}$

と

$R$

の量子化関数である。

通常、両方の量子化関数を簡単に実装できることが望ましい。

$RangeTabLPS$

30

の要素が適切に選択されている限り、任意の量子化関数を使用できる。

【 0 3 9 6 】

これは、本発明のために利用することができる。

ルックアップテーブル

$LUT1$

または

$LUT2$

40

を使用し、コンテキストモデルの確率推定値を導出する必要があったことを思い返された  
い。これらのルックアップテーブルは、

$RangeTabLPS$

に（または、一般的に言えば、M コーダーのルックアップテーブルとは異なる 2 次元ルッ  
クアップテーブルに）組み込むことができる。

【 0 3 9 7 】

アプローチ 1 および 2 で確率推定値を計算する代わりに、サブインターバル幅は、例え  
ば次のように、

50

$s_i^k$ 

の特定の値に対して直接計算される：

アプローチ 1 の場合、

 $R$ 

が式 (1) と式 (7) の両側で乗算され、

$$R \cdot p_k = \sum_{i=1}^N R \cdot p_i^k \cdot b_i^k$$

10

$$R \cdot p_i^k = \begin{cases} R \cdot LUT1[[s_i^k \cdot a_i^k]], & \text{if } s_i^k \geq 0, \\ R - R \cdot LUT1[[-s_i^k \cdot a_i^k]], & \text{else.} \end{cases}$$

になる。

【0398】

次に、

 $R \cdot LUT1$ 

は、必要な数の異なる量子化値

 $Qr(R)$ 

20

の事前計算された積

 $R \cdot LUT1$ 

を含む 2D ルックアップテーブルに置き換えられる。（実施形態の実際の実装の例）

結果として得られる 2D ルックアップテーブルは、例えば、ブロック 736 および 738 で使用することができる。

【0399】

アプローチ 2 の場合、

 $R$ 

30

は式 (2) の両側で乗算され、次のようになる。

$$R \cdot p_k = \begin{cases} R \cdot LUT2[[s_k \cdot a_k]], & \text{if } s_k \geq 0, \\ R - R \cdot LUT2[[-s_k \cdot a_k]], & \text{else.} \end{cases}$$

(8)

次に、

 $R \cdot LUT2$ 

は、所望の数の異なる量子化値

 $Qr(R)$ 

40

について事前に計算された積

 $R \cdot LUT2$ 

を含む 2D ルックアップテーブルに置き換えられる（実施形態の実際の実装の例）。

 $LUT3$ 

がこのような 2D ルックアップテーブルであると、式 (8) は次のようになる。

$$R \cdot p_k = \begin{cases} LUT3[[s_k \cdot a_k]][Qr(R)], & \text{if } s_k \geq 0, \\ R - LUT3[[-s_k \cdot a_k]][Qr(R)], & \text{else.} \end{cases}$$

50

( 9 )

式 ( 9 ) からわかるように、例えば、

LUT3

を 1 つの単一のルックアップテーブルにおいて  
RangeTabLPS

と

LUT2

10

に組み込む。

【 0 4 0 0 】

LUT3、または式 ( 9 ) は、例えば、ブロック 756 で使用することができる。  
あるいは、LUT3 を LUT4 ( 1D ) に置き換え、代替案 1 として提示されたスキームに従って乗算演算を行うこともできる

$$R \cdot p_k = \begin{cases} \lfloor LUT4[s_k \cdot a_k] \rfloor \cdot Qr_2(R), & \text{if } s_k \geq 0. \\ R - \lfloor LUT4[-s_k \cdot a_k] \rfloor \cdot Qr_2(R), & \text{else.} \end{cases}$$

関数

Qr2(R)

20

は、例えば

$$Qr_2(R) = (\lfloor R \cdot 2^{-u} \rfloor + v) \cdot 2^{-w}$$

のように定義できる。

u

v

および

w

30

は圧縮アプリケーションによって異なる。

例えば、

u = 5

v

v = 1

、および

w = 3

40

c

コンテキストモデル

ck

に関連付けられたバイナリシンボルの算術符号化は、次いでサブインターバル幅  
R · pk

を導出するために、式 ( 10 ) に従って状態変数

50

$s_k$ 

の加重和を導出し、式(9)を実行することだけを必要とする。

同じ議論がアプローチ1にも当てはまる。

### 【0401】

以下に、

*RangeTabLPS*

を効率的に実装する（または間隔サイズ情報を導出する）ためのさらなる（任意選択の代替）方法を示す。ルックアップテーブル全体を格納する代わりに、要素は必要に応じてオンザフライで計算される。この目的のために、新しいルックアップテーブル

*BaseTabLPS*

を定義する必要がある。これには、簡単な操作を使用して

*RangeTabLPS*

の各要素を導出するために使用できる値が含まれている。

### 【0402】

一般的に言えば、例えば、*RangeTabLPS*[i][j]は、ブロック870で評価され得、*RangeTabLPS*のエントリは、例えば、*BaseTabLPS*および対応するスケーリング関数 *Scal*(x, y) (*BaseTabLPS*を使用した実施形態)を使用するか、または*ProbTabLPS*と対応するスケーリング関数 *Scal*(x, y) (*ProbTabLPS*を使用した実施形態)を使用することで取得できる。

### 【0403】

好みの実施形態（例えば、*BaseTabLPS*を使用する実施形態）では、

*BaseTabLPS*

は、サイズ

$\mu \cdot \lambda$

であり、

$\mu' \cdot \lambda$

サイズ

*RangeTabLPS*

は以下のように

*BaseTabLPS*

から生成されなければならない：

*RangeTabLPS*[i][j] = *Scal*(*BaseTabLPS*[i %  $\mu$ ][j],  $i / \mu$ )

*Scal*(x, y)

はスケーリング関数である。例えば、

$\mu = 16$

、

$\lambda = 8$

10

20

30

40

50

$\mu' = 32$

$\text{Scal}(x, y) = \lfloor x \cdot 2^{-3y} \rfloor$

、 および

$\text{BaseTabLPS}[16][8] =$

{

{ 1 2 5、 1 4 0、 1 5 6、 1 7 1、 1 8 7、 2 0 3、 2 1 8、 2 3 4、 }、  
 { 1 0 8、 1 2 1、 1 3 5、 1 4 8、 1 6 2、 1 7 5、 1 8 9、 2 0 2、 }、  
 { 9 4、 1 0 5、 1 1 7、 1 2 9、 1 4 1、 1 5 2、 1 6 4、 1 7 6、 }、  
 { 8 1、 9 1、 1 0 1、 1 1 1、 1 2 1、 1 3 1、 1 4 1、 1 5 1、 }、  
 { 7 1、 7 9、 8 8、 9 7、 1 0 6、 1 1 5、 1 2 4、 1 3 3、 }、  
 { 6 1、 6 8、 7 6、 8 3、 9 1、 9 9、 1 0 6、 1 1 4、 }、  
 { 5 3、 5 9、 6 6、 7 2、 7 9、 8 6、 9 2、 9 9、 }、  
 { 4 6、 5 1、 5 7、 6 3、 6 9、 7 4、 8 0、 8 6、 }、  
 { 4 0、 4 5、 5 0、 5 5、 6 0、 6 5、 7 0、 7 5、 }、  
 { 3 5、 3 9、 4 3、 4 8、 5 2、 5 6、 6 1、 6 5、 }、  
 { 3 1、 3 4、 3 8、 4 2、 4 6、 5 0、 5 4、 5 8、 }、  
 { 2 7、 3 0、 3 3、 3 7、 4 0、 4 3、 4 7、 5 0、 }、  
 { 2 4、 2 7、 3 0、 3 3、 3 6、 3 9、 4 2、 4 5、 }、  
 { 2 1、 2 3、 2 6、 2 8、 3 1、 3 4、 3 6、 3 9、 }、  
 { 1 8、 2 0、 2 2、 2 4、 2 7、 2 9、 3 1、 3 3、 }、  
 { 1 6、 1 8、 2 0、 2 2、 2 4、 2 6、 2 8、 3 0、 }、  
 }。 (例)

【0 4 0 4】

言い換えると、例えば、

$\text{RangeTabLPS}$

10

20

30

の上半分は、フロア操作が各要素に適用される

$\text{BaseTabLPS}$

の値からなる（または構成される）。例えば、

$\text{RangeTabLPS}$

の下半分は、各要素が 8（または別の数値）で除算された

$\text{BaseTabLPS}$

40

の値からなり（または構成され）、その後、フロア操作が続き、これは、例えば、3だけに右シフトに対応する。

このアプローチは、

$\text{RangeTabLPS}$

よりも小さい値

$\text{BaseTabLPS}$

を格納するだけによく、例えば、

$\text{RangeTabLPS}$

50

の行の値が行の増加に伴って指数関数的に減衰するという事実を利用する。指数関数的減衰はパラメータ<sup>a</sup>

に依存する。

【 0 4 0 5 】

別の例（例えば、BaseTabLPSを使用する別の実施形態）では、

$\mu = 4$

、  
 $\lambda = 8$

、  
 $\mu' = 32$

、  
 $Scal(x, y) = \lfloor x \cdot 2^{-y} \rfloor$

、 および

*BaseTabLPS*[4][8] =

10

{  
 { 1 2 1、 1 3 6、 1 5 1、 1 6 6、 1 8 1、 1 9 6、 2 1 1、 2 2 6、 }、  
 { 1 0 0、 1 1 2、 1 2 5、 1 3 7、 1 5 0、 1 6 2、 1 7 5、 1 8 7、 }、  
 { 8 3、 9 3、 1 0 3、 1 1 4、 1 2 4、 1 3 4、 1 4 5、 1 5 5、 }、  
 { 7 0、 7 8、 8 7、 9 6、 1 0 5、 1 1 3、 1 2 2、 1 3 1、 }  
 }。（例）

この構成は、

<sup>a</sup>

20

が

*RangeTabLPS*

30

の行の値が4行ごとに半分になるように選択されている場合に、使用できる。

*BaseTabLPS*

の値は、例えば、

*RangeTabLPS*

に関連付けられたから簡単に抽出できることに留意されたい。

【 0 4 0 6 】

関与するアプローチ

*BaseTabLPS*

40

は、（任意選択で）式（11）および関連する好ましい実施形態で定義される代替案1と結合することができる。代替案1は、1つの定数からの

*RangeTabLPS*

の行と計算演算（各行）を導出するためのアプローチと見なすことができる。例えば、

*BaseTabLPS*

50

は、（丸めを無視して）

*RangeTabLPS*

のスケーリングされた行が含まれているため、代替案1を適用して

*BaseTabLPS*

を導出できる。

【0 4 0 7】

好みの実施形態（例えば、別の実施形態、例えば、ProbTabLPSを使用する実施形態）では、サイズ

$\mu$

10

の

*probTabLPS*

が定義される。サイズ

$\mu \cdot \lambda$

の

*BaseTabLPS*

20

は、例えば、

*probTabLPS*

から次のように生成される。

$j = \lfloor R \cdot 2^{-5} \rfloor - 8$

*BaseTabLPS*[ $\vartheta$ ][ $j$ ] = [*probTabLPS*[ $\vartheta$ ] ·  $Qr_2(R)$ ]

【0 4 0 8】

30

例えば、

$\mu = 16$

$\lambda = 8$

$\mu' = 32$

$Qr_2(R) = \lfloor R \cdot 2^{-5} \rfloor \cdot 2^{-3}$

40

$R \in [256..510]$

$Scal(x, y) = \lfloor x \cdot 2^{-3y} \rfloor$

および

*probTabLPS*[16] =

50

```

{
  1 2 5,
  1 0 8,
  9 4,
  8 1,
  7 1,
  6 1,
  5 3,
  4 6,
  4 0,
  3 5,
  3 1,
  2 7,
  2 4,
  2 1,
  1 8,
  1 6,
}。 (例)
【0 4 0 9】
次に、
RangeTabLPS

```

20

は前述のアプローチに従って

BaseTabLPS

から計算できる。

RangeTabLPS

の各値は、例えば、

BaseTabLPS

30

の関連する値を計算してからスケーリング関数

Scal(x,y)

を適用することにより、オンザフライで計算できることに留意されたい：

RangeTabLPS[i][ $R \cdot 2^{-5}$ ] - 8] = Scal([probTabLPS[i % μ] · Qr<sub>2</sub>(R)], [i/μ])

このアプローチでは、

BaseTabLPS

40

より、したがって

RangeTabLPS

よりも小さい

probTabLPS

を格納することだけを要する。

【0 4 1 0】

同じ議論に続いて、例えば、

BaseTabLPS

50

の 1 つの列は、

*BaseTabLPS*

および

*RangeTabLPS*

のその他すべての列を単に無視することによって、

*RangeTabLPS*

の 1 つの列のみを導出するためには、

*RangeTabLPS*

10

の導出列などを使用して、式 ( 1 1 ) に従って代替 1 を使用することにより、

*RangeTabLPS*

の残りの列を導出させることができる。

【 0 4 1 1 】

好みの実施形態では、サイズ

$\mu$

20

の

*probTabLPS*

が定義される。

$\mu' \cdot \lambda$

サイズの

*RangeTabLPS*

は、次のように

*probTabLPS*

30

から生成される：

*RangeTabLPS*[ $i$ ][ $[R \cdot 2^{-3}] - 8$ ] = [*Scal*(*probTabLPS*[ $i \% \mu$ ], [ $i / \mu$ ])  $\cdot Qv_2(R)$ ]

*Scal*( $x, y$ ) = [ $x \cdot 2^{-3}y$ ].

を留意されたい（例）

状態変数の更新（発明の態様）

シンボルの符号化または復号の後、シンボルシーケンスの統計的動作を追跡するために、コンテキストモデルの 1 つまたは複数の状態変数を更新することができる。これは、例えば、ブロック 6 4 0 で行うことができる。

【 0 4 1 2 】

更新は、例えば、次のように実行される：

$$s_i^k = \begin{cases} s_i^k + [A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 1.} \\ s_i^k - [A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k], & \text{If symbol to be encoded is 0.} \end{cases}$$

40

( 3 )

*A*

50

は整数値を格納するルックアップテーブルである。

$m_i^k$

と

$n_i^k$

は、更新の「敏捷性」を制御する重み付け係数である。

$z$

はルックアップテーブル

$A$

10

が負でない値でのみアクセスされることを保証するオフセットである。

【0 4 1 3】

以下で説明するように、ルックアップテーブル

$A$

の値は、

$s_i^k$

20

が特定の所与の間隔にとどまるように選択できる。

あるいは、例えば、以下のように、式(3)の各実行後に実行されるクリッピング操作を導入することが可能である：

$$s_i^k = \min(\max(s_i^k, h_i^k), g_i^k)$$

( 6 )

$g_i^k$

は

$s_i^k$

30

の最大許容値である

$h_i^k$

は

$s_i^k$

の最小許容値である

【0 4 1 4】

圧縮アプリケーションに関してルックアップテーブルAを最適化するために、ルックアップテーブルAの値で近似される想定される関数を計算操作に置き換えることもできる。

例えば、

$$A[z + \hat{s}] = offset + \sum_{i=1}^N \max(0, \hat{b}_i - \hat{s}) \ll \hat{a}_i$$

( 7 )

$\hat{N}$

40

、

$\hat{b}_i$

50

、および  
 $\hat{a}_i$

は、確率更新の動作を制御するパラメータである。

【 0 4 1 5 】

ルックアップテーブル A を作成するこの方法には、上記のようなクリッピング操作が含まれていないことに留意されたい。ルックアップテーブル A は、式 7 の

$\hat{a}_i, \hat{b}_i, offset$

の様々な結合の圧縮効率を評価し、続いて目的の値

$z + \delta$

で

$A[z + \delta]$

をサンプリングすることによって最適化できる。

【 0 4 1 6 】

特定のエンコーダまたはデコーダアプリケーションは、計算操作を使用するか、同等のテーブル（計算操作によって導出された値を含む）を使用するかを選択できる。

ビデオ圧縮コンテキストのルックアップテーブル A を最適化するために、次の範囲の可能な構成が使用される：

$\hat{N} = 3$

$offset = \{2^4, 2^5 \dots 2^{11}\}$

$\hat{a}_1 = \{2, 3 \dots 7\}$

$\hat{a}_2 = \{3, 4 \dots 7\}$

$\hat{a}_3 = \{5, 6 \dots 8\}$

$\hat{b}_1 = \{-0, -1, -2, -3, -4, -5, -6, -7\}$

$\hat{b}_2 = \{-9, -11, -13, -15, -17, -19, -21, -23\}$

$\hat{b}_3 = \{-24, -26, -28, -30, -32, -34, -36, -38\}$

【 0 4 1 7 】

式 ( 3 ) の任意選択の変更

最適化されたソフトウェア実装でより高いスループットを達成するために、2つの任意選択の変更を式 ( 3 ) に適用できる。

第 1 の任意選択の変更では、式 ( 3 ) は次のように置き換えられる：

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 1.} \\ s_i^k - \lfloor A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 0.} \end{cases}$$

式 ( 3 a )

第 2 の任意選択の変更では、式 ( 3 ) は次のように置き換えられる：

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z + \lfloor -s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{if symbol to be encoded is 0.} \end{cases}$$

10

20

30

40

50

## 式 ( 3 b )

第 1 および第 2 の任意選択の変更の結合の場合、式 ( 3 ) は次のように置き換えられる：

$$s_i^k = \begin{cases} s_i^k + \lfloor A[z + \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 1.} \\ s_i^k + \lfloor -A[z - 1 - \lfloor s_i^k \cdot m_i^k \rfloor] \cdot n_i^k \rfloor, & \text{If symbol to be encoded is 0.} \end{cases}$$

## 式 ( 3 c )

## 【 0 4 1 8 】

第 1 および第 2 の任意選択の変更の結合を採用することにより、ケースの区別をルックアップテーブル

*A*

10

に統合する事が可能になり、ルックアップテーブルは、以前の 2 倍の要素数しか必要としない。ただし、高スループットを実現するには、演算の数を減らすと役立つ場合があるが、ルックアップテーブルのサイズを大きくすると関連性が低くなる場合がある。

さらに通知することなく、2 つの任意選択の変更（個別にまたは組み合わせて）を、本発明の式 3 のすべての適用例に適用することができる。

## 【 0 4 1 9 】

好ましい実施形態（詳細は任意選択である）

好ましい実施形態では、以下の構成が使用される：

コンテキストモデル

*M*

20

の数は、圧縮アプリケーションによって異なる。

アプローチ 2 は、状態変数から確率推定値を導出するために使用される。

## 【 0 4 2 0 】

式 ( 3 ) は、状態変数の更新に使用される。

式 ( 6 ) による任意選択のクリッピングは使用されない。

*N* = 2*d*<sub>1</sub><sup>k</sup> = 16

30

*d*<sub>2</sub><sup>k</sup> = 1

*LUT2* = {0.5000, 0.4087, 0.3568, 0.3116, 0.2721, 0.2375, 0.2074, 0.1811,

0.1581, 0.1381, 0.1206, 0.1053, 0.0919, 0.0803, 0.0701, 0.0612,

0.0534, 0.0466, 0.0407, 0.0356, 0.0310, 0.0271, 0.0237, 0.0207,

0.0180, 0.0158, 0.0138, 0.0120, 0.0105, 0.0092, 0.0080, 0.0070}

40

*a*<sub>k</sub> = 2<sup>-7</sup>

*A* = {165, 146, 128, 110, 94, 79, 66, 54, 43, 34, 26, 20, 15, 11, 8, 5,

4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0}

*m*<sub>1</sub><sup>k</sup> = 2<sup>-3</sup>

*m*<sub>2</sub><sup>k</sup> = 2<sup>-7</sup>

50

$n_1^k = 2^{-1}$

$n_2^k = 1$

$z = 16$

【 0 4 2 1 】

シンボルを符号化する前に、

$s_1^k$

10

は間隔

[-127,127]

の値で初期化され、

$s_2^k$

は間隔

[-2047,2047]

の値で初期化される。

20

この構成の場合、式(3)は次のように記述できる：

$$s_2^k = \begin{cases} s_1^k + A[16 + (s_1^k >> 3)] >> 1, & \text{if symbol to be encoded is 1.} \\ s_1^k - A[16 + ((-s_1^k) >> 3)] >> 1, & \text{if symbol to be encoded is 0.} \end{cases}$$

( 4 )

$$s_2^k = \begin{cases} s_2^k + A[16 + (s_2^k >> 7)], & \text{if symbol to be encoded is 1.} \\ s_2^k - A[16 + ((-s_2^k) >> 7)], & \text{if symbol to be encoded is 0.} \end{cases}$$

( 5 )

$s_1^k$

30

が間隔

[-127,127]

内にある場合、式(4)は、符号化されるシンボルの値に関係なく、同じ間隔内にとどまる

$s_1^k$

に対する更新された値を生成する。

$s_2^k$

40

が間隔

[-2047,2047]

内にある場合、式(5)は、符号化されるシンボルの値に関係なく、同じ間隔内にとどまる

$s_2^k$

に対する更新された値を生成する。

【 0 4 2 2 】

したがって、

$s_1^k$

50

は例えば、8ビットの符号付き整数値で実装でき、

$s_2^k$

は12ビットの符号付き整数で実装できる。

どちらの式も、1を符号化した後に状態変数を（最大値まで）増加させ、0を符号化した後に状態変数を（最小値まで）減少させる。ただし、式（4）は、

$s_2^k$

に関して式（5）の場合よりも大幅に

$s_1^k$

10

を増減する傾向がある。言い換えると、

$s_1^k$

は以前に符号化されたシンボルに速く適合し、一方で

$s_2^k$

はゆっくりと適合する。

【0 4 2 3】

別の好ましい実施形態では、前の好ましい実施形態と同じ構成が使用されるが、第1および第2の任意の修正が式（3）に適用されて、以下が得られる：

20

$$s_2^k = \begin{cases} s_1^k + (A[16 + (s_1^k >> 3)] >> 1), & \text{if symbol to be encoded is 1.} \\ s_1^k + ((-A[15 - (s_1^k >> 3)]) >> 1), & \text{if symbol to be encoded is 0.} \end{cases}$$

$$s_2^k = \begin{cases} s_2^k + A[16 + (s_2^k >> 7)], & \text{if symbol to be encoded is 1.} \\ s_2^k - A[15 - (s_2^k >> 7)], & \text{if symbol to be encoded is 0.} \end{cases}$$

ルックアップテーブル

$LUT2$

30

と

$A$

は通常、特定の圧縮アプリケーションに最適化されていることに留意されたい。

ルックアップテーブルAが最適化されると、次の構成が式（7）で使用される：

$\hat{N} = 3$

$offset = 2^{10}$

$\hat{a}_1 = 8$

40

$\hat{a}_2 = 7$

$\hat{a}_3 = 9$

$\hat{b}_1 = -3$

$\hat{b}_2 = -15$

$\hat{b}_3 = -28$

50





$m_1^k = 2^{-3}$

$n_1^k = 1$

$z = 16$

シンボルを符号化する前に、

$s_1^k$

は任意選択で間隔  
[-127, 127]

10

からの値で初期化する必要がある。

【0 4 3 0】

別の好ましい実施形態では、アプローチ 2 が使用され、パラメータ  
 $m_i^k$

$n_i^k$

20

、および  
 $a_k$

は、異なるコンテキストモデルに対して異なる確率更新動作を得るために、異なるコンテキストモデルに対して異なるように設定される。

【0 4 3 1】

別の好ましい実施形態では、アプローチ 1 が使用され、パラメータ  
 $m_i^k$

$n_i^k$

30

、および  
 $a_i^k$

は、異なるコンテキストモデルに対して異なる確率更新動作を得るために、異なるコンテキストモデルに対して異なるように設定される。別の好ましい実施形態では、アプローチ 1 またはアプローチ 2 が使用される場合、 $N = 1$  およびパラメータ  
 $m_i^k$

40

$n_i^k$

、および  
 $a_i^k$

は、異なるコンテキストモデルに対して異なる確率更新動作を得るために、異なるコンテキストモデルに対して異なるように設定され、

【0 4 3 2】

ルックアップテーブル A が最適化されると、次の構成が式 (7) で使用される：

50

$\hat{N} = 3$  $offset = 2^{10}$  $\hat{a}_1 = 8$  $\hat{a}_2 = 7$  $\hat{a}_3 = 10$ 

10

 $\hat{b}_1 = -3$  $\hat{b}_2 = -15$  $\hat{b}_3 = -30$ **【 0 4 3 3 】****その結果、**

$A = \{225, 203, 181, 159, 137, 115, 93, 71, 49, 35, 29, 23, 17, 13, 9, 5, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0\}.$

20

A はすでにクリップされているため、最後の値は 4 ではなく 0 に等しいことに留意されたい。

**【 0 4 3 4 】**

別の好ましい実施形態では、アプローチ 1 またはアプローチ 2 が使用される場合、 $N = 2$  およびパラメータ

 $m_i^k$  $n_i^k$ 

30

**、 および** $a_i^k$ 

は、異なるコンテキストモデルに対して異なる確率更新動作を得るために、異なるコンテキストモデルに対して異なるように設定される。

ルックアップテーブル A が最適化されると、次の構成が式 (7) で使用される。

 $\hat{N} = 3$  $offset = 2^{10}$ 

40

 $\hat{a}_1 = 8$  $\hat{a}_2 = 7$  $\hat{a}_3 = 9$  $\hat{b}_1 = -3$  $\hat{b}_2 = -15$ 

50

$\hat{b}_3 = -26$

### 【 0 4 3 5 】

その結果、

$A = \{ 165, 151, 137, 123, 109, 95, 81, 67, 53, 39, 29, 23, 17, 13, 9, 5, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0 \}.$

あるいは、

$\hat{b}_3 = -28$

はまた、

$A = \{ 157, 143, 129, 115, 101, 87, 73, 59, 45, 35, 29, 23, 17, 13, 9, 5, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0 \}.$

10

に至るこの実施形態における所望の構成である。Aがすでにクリップされており、したがって、最後の値は4ではなく0に等しいことに留意されたい。

### 【 0 4 3 6 】

19. 実装の代替

いくつかの態様は、装置の文脈で説明されているが、これらの態様が、対応する方法の説明も表すことは明らかであり、ブロックまたはデバイスは、方法ステップまたは方法ステップの特徴に対応する。同様に、方法ステップの文脈で説明された態様は、対応する装置の対応するブロックまたはアイテムまたは特徴の説明も表す。方法ステップの一部またはすべては、例えば、マイクロプロセッサ、プログラム可能なコンピュータ、または電子回路などのハードウェア装置によって（または使用して）実行することができる。いくつかの実施形態では、最も重要な方法ステップの1つまたは複数は、そのような装置によって実行され得る。

20

### 【 0 4 3 7 】

特定の実装要件に応じて、本発明の実施形態は、ハードウェアまたはソフトウェアで実装することができる。実装は、フロッピーディスク、DVD、ブルーレイ、CD、ROM、PROM、EPROM、EEPROM、フラッシュメモリなどのデジタル記憶媒体を使用して実行でき、電子的に読み取り可能な制御信号が格納されて、それぞれの方法が実行されるように、プログラム可能なコンピュータシステムと協働する（または協働することができる）。したがって、デジタル記憶媒体は、コンピュータ可読であり得る。

30

### 【 0 4 3 8 】

本発明によるいくつかの実施形態は、本明細書に記載の方法の1つが実行されるように、プログラム可能なコンピュータシステムと協働することができる電子的に読み取り可能な制御信号を有するデータキャリアを含む。

一般に、本発明の実施形態は、プログラムコードを有するコンピュータプログラム製品として実装することができ、プログラムコードは、コンピュータプログラム製品がコンピュータ上で実行されるときに方法の1つを実行するように動作する。プログラムコードは、例えば、機械可読キャリアに格納されてもよい。

40

### 【 0 4 3 9 】

他の実施形態は、機械可読キャリアに記憶された、本明細書に記載された方法の1つを実行するためのコンピュータプログラムを含む。

言い換えれば、本発明の方法の実施形態は、したがって、コンピュータプログラムがコンピュータ上で実行されるときに、本明細書で説明される方法の1つを実行するためのプログラムコードを有するコンピュータプログラムである。

### 【 0 4 4 0 】

したがって、本発明の方法のさらなる実施形態は、本明細書に記載の方法の1つを実行するためのコンピュータプログラムを記録し含むデータキャリア（またはデジタル記憶媒体、またはコンピュータ可読媒体）である。データキャリア、デジタル記憶媒体、または記録された媒体は、通常、有形および/または非過渡的である。

50

したがって、本発明の方法のさらなる実施形態は、本明細書で説明される方法の1つを実行するためのコンピュータプログラムを表すデータストリームまたは信号のシーケンスである。データストリームまたは信号のシーケンスは、例えば、インターネットなどのデータ通信接続を介して転送されるように構成されてもよい。

#### 【0441】

さらなる実施形態は、本明細書に記載された方法の1つを実行するように構成または適合された処理手段、例えば、コンピュータまたはプログラム可能な論理デバイスを含む。

さらなる実施形態は、本明細書に記載される方法の1つを実行するためのコンピュータプログラムがインストールされているコンピュータを含む。

#### 【0442】

本発明によるさらなる実施形態は、本明細書に記載の方法の1つを実行するためのコンピュータプログラムを（例えば、電子的または光学的に）受信機に転送するように構成された装置またはシステムを含む。受信機は、例えば、コンピュータ、モバイルデバイス、メモリデバイスなどであり得る。装置またはシステムは、例えば、コンピュータプログラムを受信機に転送するためのファイルサーバを含み得る。

#### 【0443】

いくつかの実施形態では、プログラマブルロジックデバイス（例えば、フィールドプログラマブルゲートアレイ）を使用して、本明細書に記載されている方法の機能の一部またはすべてを実行することができる。いくつかの実施形態では、フィールドプログラマブルゲートアレイは、本明細書で説明される方法の1つを実行するために、マイクロプロセッサと協働し得る。一般に、方法は、好ましくは、任意のハードウェア装置によって実行される。

#### 【0444】

本明細書で説明する装置は、ハードウェア装置を使用して、またはコンピュータを使用して、またはハードウェア装置とコンピュータの組み合わせを使用して実装することができる。

本明細書で説明される装置、または本明細書で説明される装置の任意のコンポーネントは、少なくとも部分的にハードウェアおよび／またはソフトウェアで実装され得る。

#### 【0445】

本明細書で説明する方法は、ハードウェア装置を使用して、またはコンピュータを使用して、またはハードウェア装置とコンピュータの組み合わせを使用して実行することができる。

本明細書で説明される方法、または本明細書で説明される装置の任意のコンポーネントは、ハードウェアおよび／またはソフトウェアによって少なくとも部分的に実行され得る。

#### 【0446】

本明細書に記載の実施形態は、本発明の原理を単に例示するものである。本明細書に記載の配置および細部の修正および変形は、当業者には明らかであることが理解される。したがって、本明細書の実施形態の記載および説明として提示される特定の細部によってではなく、間近の特許請求の範囲によってのみ制限されることが意図されている。

算術エンコーダ、算術デコーダ、ビデオエンコーダ、ビデオデコーダ、符号化方法、復号方法、およびコンピュータプログラム

10

20

30

40

50

【図面】

【図 1】

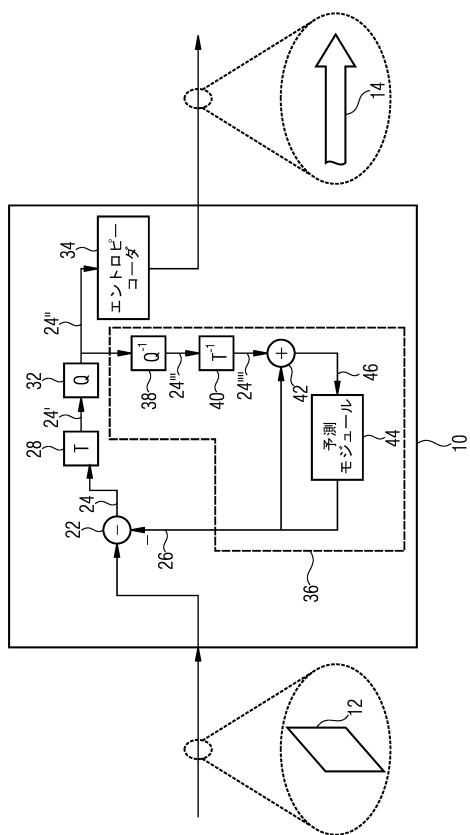


Fig. 1

【図 2】

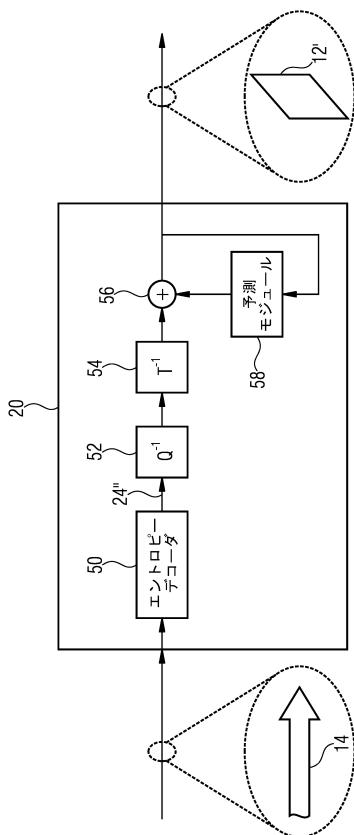


Fig. 2

10

20

30

40

【図 3】

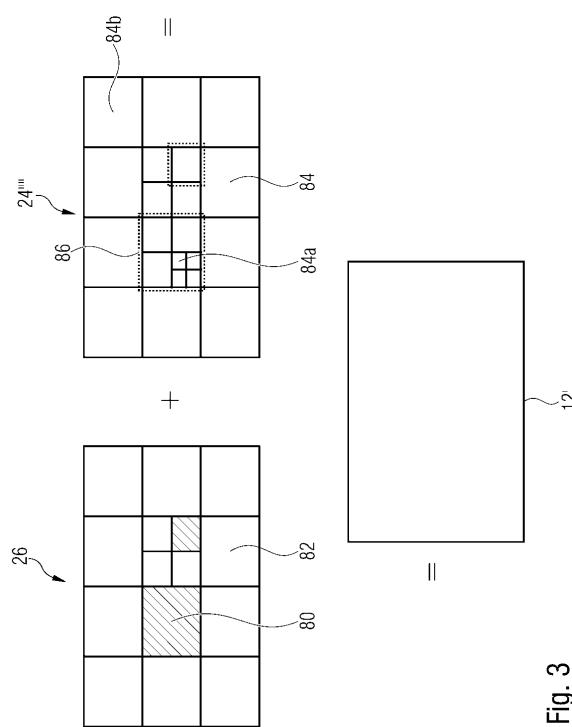


Fig. 3

【図 4】

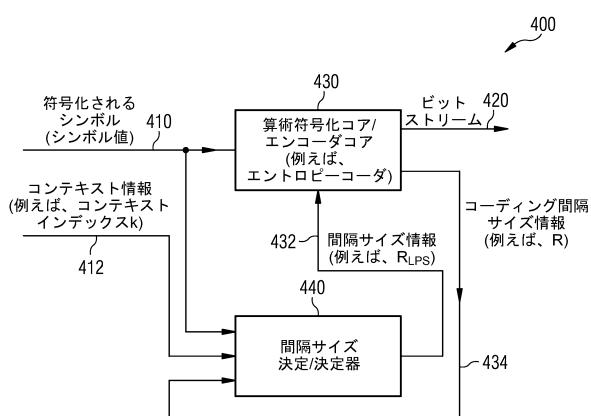


Fig. 4

50

【 図 5 】

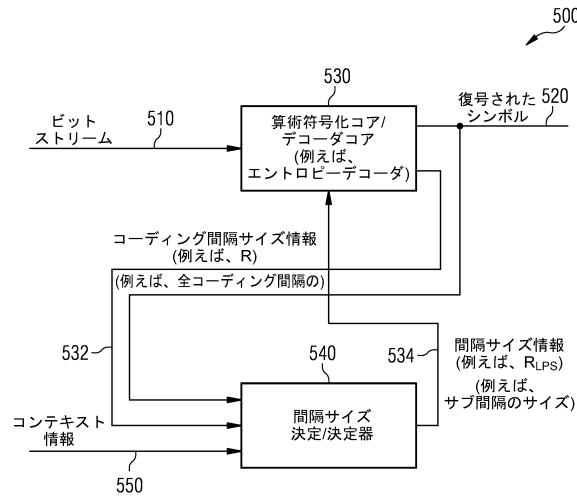


Fig. 5

【 図 6 】

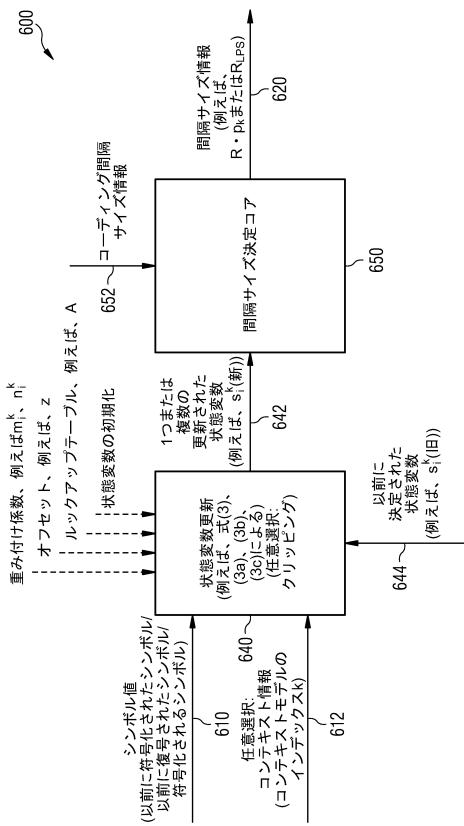


Fig. 6

【 図 7 】

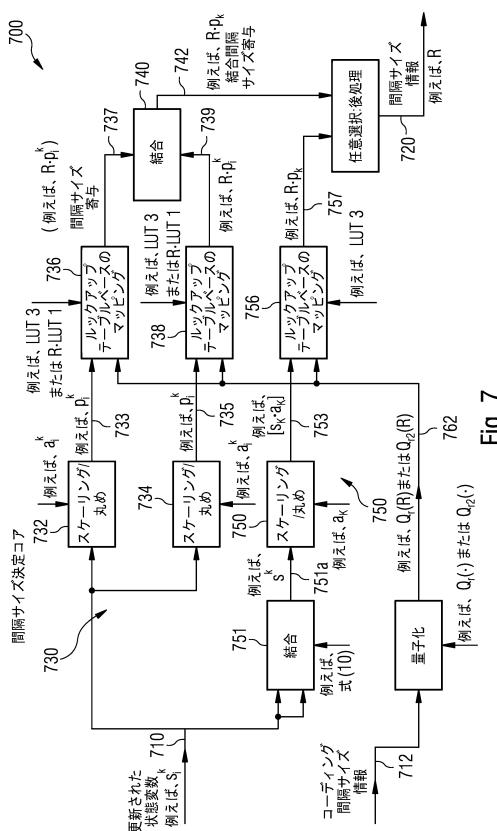
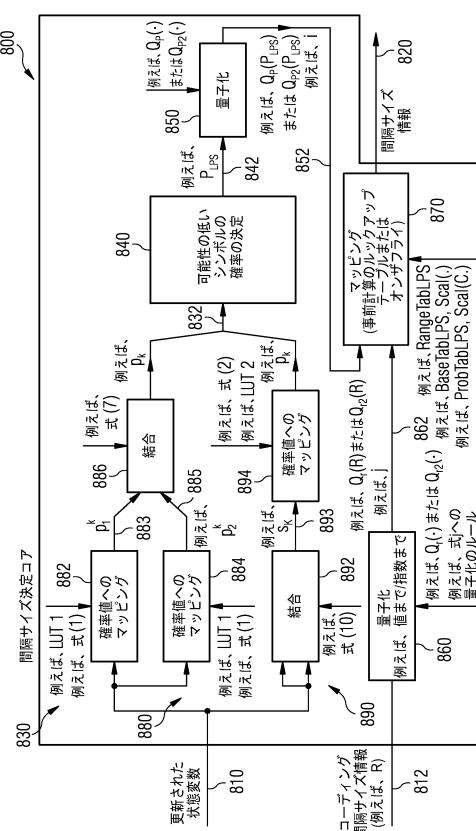


Fig. 7

【 図 8 】



8  
ig.  
15

【図 9】

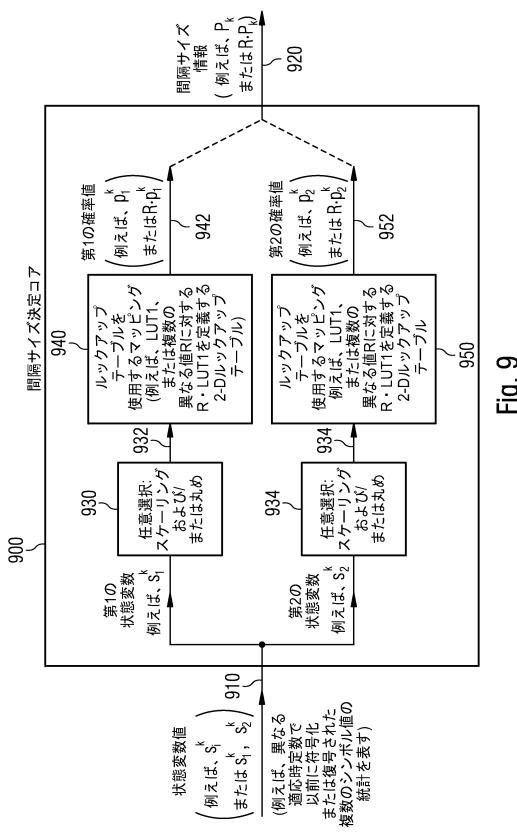


Fig. 9

【図 10 a】

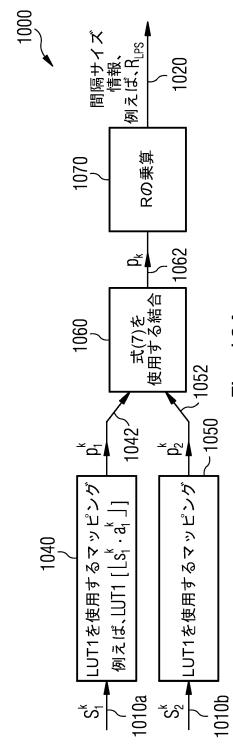


Fig. 10A

【図 10 b】

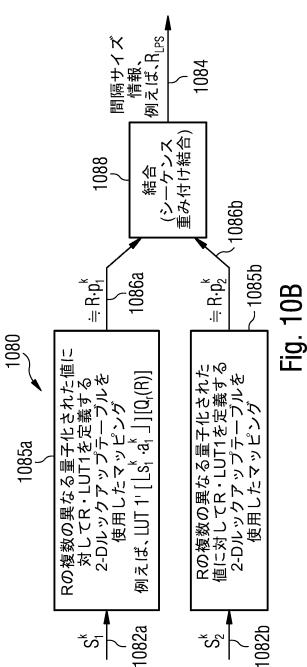


Fig. 10B

【図 11】

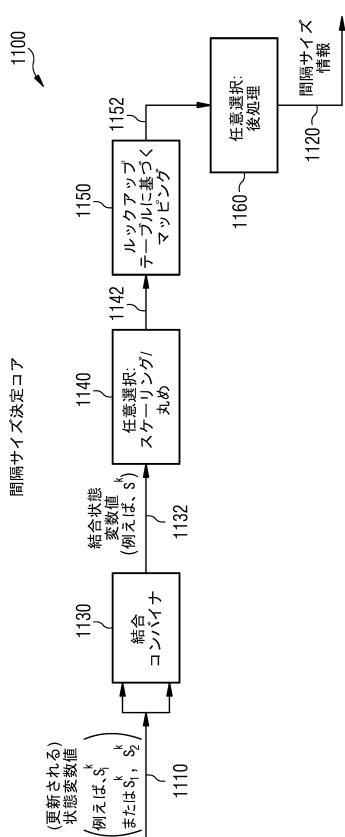


Fig. 11

## 【図 1 2】

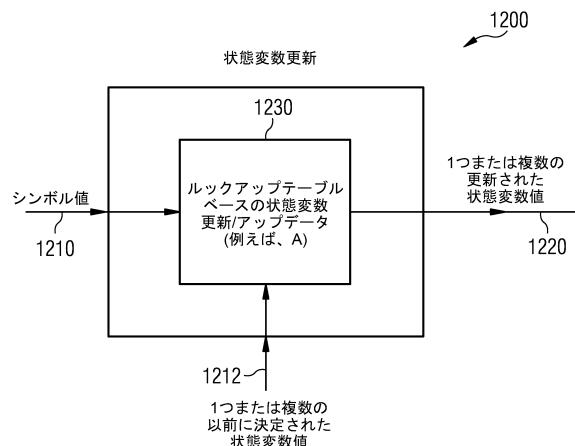


Fig. 12

### 【図13】

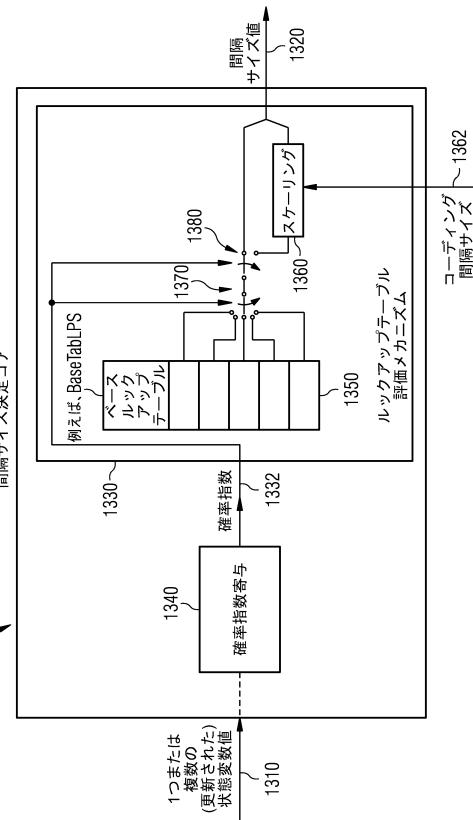


Fig. 13

10

20

### 【図14】

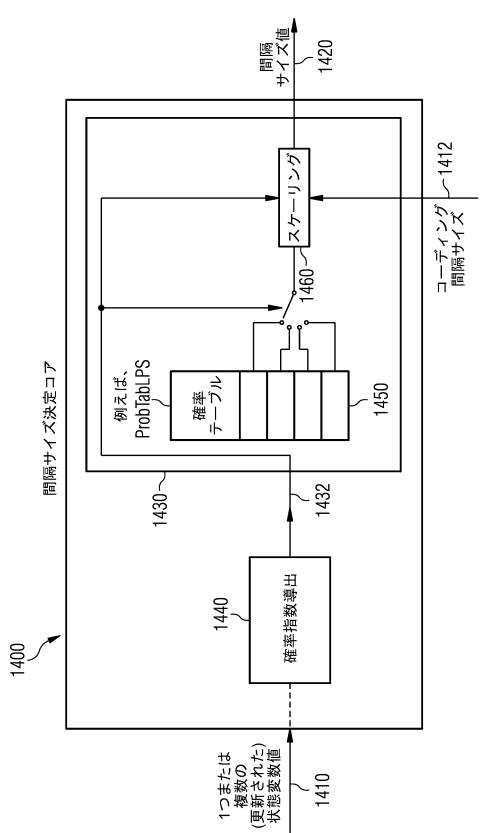


Fig. 14

【図15】

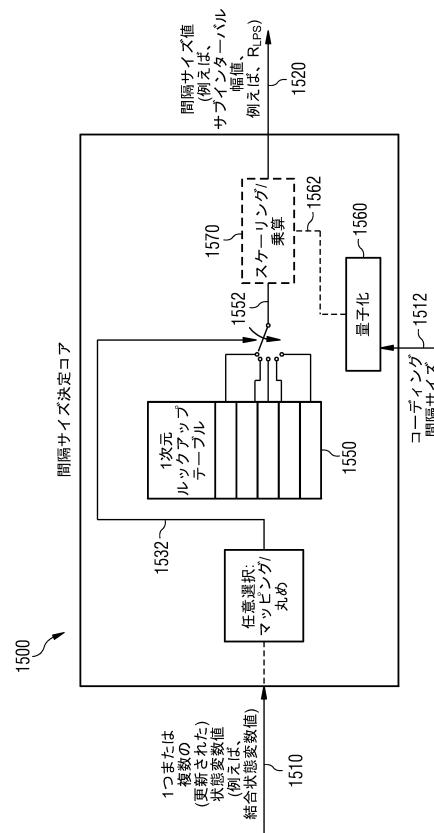


Fig. 15

30

40

50

【図16】

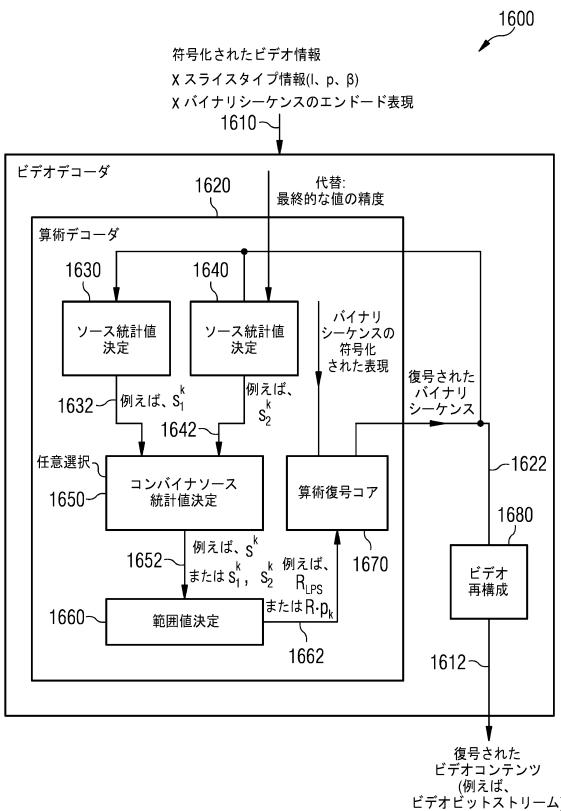


Fig. 16

【図17】

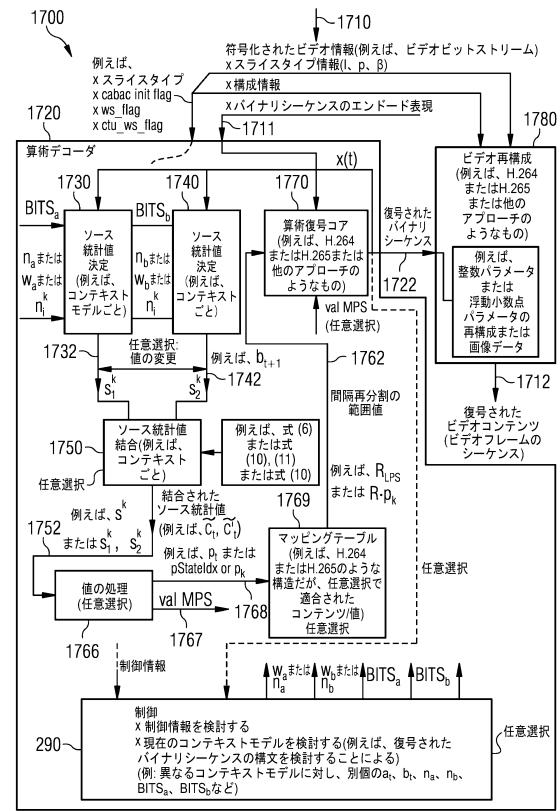


Fig. 17

## フロントページの続き

欧州特許庁(EP)  
(31)優先権主張番号 18248294.3  
(32)優先日 平成30年12月28日(2018.12.28)  
(33)優先権主張国・地域又は機関  
　　欧州特許庁(EP)  
　　, ハーハーイー内  
(72)発明者 ステゲマン・ヤン  
　　ドイツ連邦共和国, 79219 シュタウフェン, イム ベヒレフェルト 8  
(72)発明者 マルペ・デトレフ  
　　ドイツ連邦共和国, 10587 ベルリン, アインシュタインウファー 37 フラウンホーファー  
　　- インスティテュート・フル・ナハリヒテンテヒニック, ハインリッヒ-ヘルツ-インスティテュ  
　　ート, ハーハーイー内  
(72)発明者 ハーゼ・パウル  
　　ドイツ連邦共和国, 10587 ベルリン, アインシュタインウファー 37 フラウンホーファー  
　　- インスティテュート・フル・ナハリヒテンテヒニック, ハインリッヒ-ヘルツ-インスティテュ  
　　ート, ハーハーイー内  
(72)発明者 マトラッジ・シュテファン  
　　ドイツ連邦共和国, 10587 ベルリン, アインシュタインウファー 37 フラウンホーファー  
　　- インスティテュート・フル・ナハリヒテンテヒニック, ハインリッヒ-ヘルツ-インスティテュ  
　　ート, ハーハーイー内  
(72)発明者 バルトニック・クリスチャン  
　　ドイツ連邦共和国, 10587 ベルリン, アインシュタインウファー 37 フラウンホーファー  
　　- インスティテュート・フル・ナハリヒテンテヒニック, ハインリッヒ-ヘルツ-インスティテュ  
　　ート, ハーハーイー内  
(72)発明者 シュヴァルツ・ハイコ  
　　ドイツ連邦共和国, 10587 ベルリン, アインシュタインウファー 37 フラウンホーファー  
　　- インスティテュート・フル・ナハリヒテンテヒニック, ハインリッヒ-ヘルツ-インスティテュ  
　　ート, ハーハーイー内  
(72)発明者 ヴィーガンド・トマス  
　　ドイツ連邦共和国, 10587 ベルリン, アインシュタインウファー 37 フラウンホーファー  
　　- インスティテュート・フル・ナハリヒテンテヒニック, ハインリッヒ-ヘルツ-インスティテュ  
　　ート, ハーハーイー内  
審査官 川口 貴裕  
(56)参考文献  
　　特表 2014-534736 (JP, A)  
　　特開 2005-217517 (JP, A)  
　　米国特許出願公開第 2013/0107951 (US, A1)  
　　Detlev Marpe et al., Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard, IEEE Transactions on Circuits and Systems for Video Technology, 2003年, Vol. 13, No. 7, pp. 620-636  
　　HENDRIK EECKHAUT et al., Tuning the M-coder to improve Dirac's Entropy Coding[online], WSEAS Transactions on Information Science and Applications, 2005年, インターネット URL:<a href="https://www.researchgate.net/profile/Jan-Campenhout/publication/228673154\_Tuning\_the\_M-coder\_to\_improve\_Dirac's\_Entropy\_Coding/links/02e7e532dc675b63b800000/Tuning-the-M-coder-to-improve-Diracs-Entropy-Coding.pdf?\_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9ulicGFnZSI6InB1YmxpY2F0aW9uln19Evgeny Belyaev et al., An Efficient Adaptive Binary Range Coder and Its VLSI Architecture, IEEE Transactions on Circuits and Systems for Video Technology, 2014年, Vol. 25, No. 8, pp. 1435-1446<br/>(58)調査した分野 (Int.Cl., D B名)

H 0 3 M 3 / 0 0 - 9 / 0 0  
H 0 4 N 1 9 / 9 1  
I E E E X p l o r e