

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2017-118547

(P2017-118547A)

(43) 公開日 平成29年6月29日(2017.6.29)

(51) Int.Cl.	F I	テーマコード (参考)
HO 4 N 19/91 (2014.01)	HO 4 N 19/91	5 C 1 5 9
HO 3 M 7/40 (2006.01)	HO 3 M 7/40	5 J 0 6 4

審査請求 有 請求項の数 30 O L (全 93 頁)

(21) 出願番号	特願2017-2986 (P2017-2986)	(71) 出願人	515089080
(22) 出願日	平成29年1月12日 (2017.1.12)		ジーイー ビデオ コンプレッション エ
(62) 分割の表示	特願2015-178257 (P2015-178257)		ルエルシー
原出願日	平成24年1月12日 (2012.1.12)		アメリカ合衆国 ニューヨーク州 1 2 2
(31) 優先権主張番号	61/432, 884		1 1 オールバニー サウスウッズ プー
(32) 優先日	平成23年1月14日 (2011.1.14)	(74) 代理人	100085497
(33) 優先権主張国	米国 (US)		弁理士 筒井 秀隆
		(72) 発明者	マルペ, デトレフ
			ドイツ連邦共和国 1 2 1 6 1 ベルリン
			ズィートヴェストコロソ 7 0
		(72) 発明者	グエン, ツング
			ドイツ連邦共和国 1 3 0 5 5 ベルリン
			ランズベルガー アレー 2 1 7

最終頁に続く

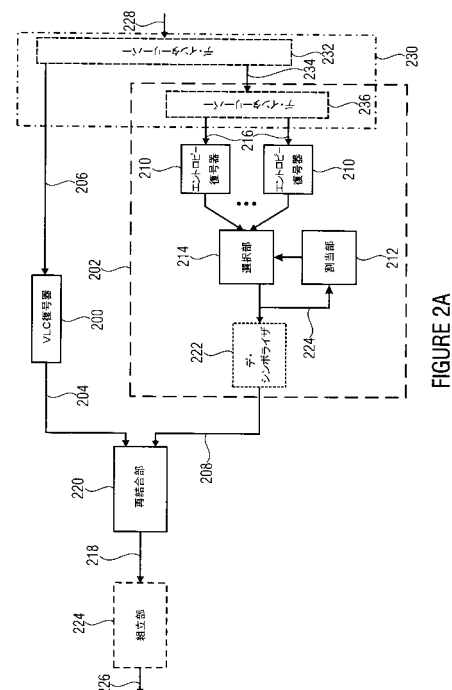
(54) 【発明の名称】 エントロピー符号化および復号化スキーム

(57) 【要約】 (修正有)

【課題】 符号化の複雑性と圧縮効率との間のより良好な妥協を達成できるエントロピー符号化の復号手段を提供する。

【解決手段】 第1復号器200がビットストリームの第1部分206を受信し、ビットストリームの第1部分206の符号語からソースシンボルの第1列のソースシンボル204を復元し、第2復号器202がビットストリームの第2部分216を受信し、ビットストリームの第2部分216の符号語からソースシンボルの第2列208を復元する。組立手段224が第1列からの全てのソースシンボルと第2列からの全てのソースシンボルとを回収し、それぞれの数のソースシンボルから各シンタックス要素を個別に組み立てることによって、ソースシンボルの第1列とソースシンボルの第2列とからパーティションの列へとサブ分割されるある値範囲を持つシンタックス要素列226を組み立てる。

【選択図】 図2 a



【特許請求の範囲】

【請求項 1】

復号化方法であって、
第 1 復号器によって、ビットストリームの第 1 部分を受信するステップと、
前記第 1 復号器によって、前記ビットストリームの第 1 部分の符号語からソースシンボルの第 1 列のソースシンボルを復元するステップと、
第 2 復号器によって、前記ビットストリームの第 2 部分を受信するステップと、
前記第 2 復号器によって、前記ビットストリームの第 2 部分の符号語からソースシンボルの第 2 列のソースシンボルを復元するステップと、
組立手段によって、前記第 1 列からのソースシンボルと前記第 2 列からのソースシンボルとを回収するステップと、
前記組立手段によって、それぞれの数のソースシンボルから各シンタックス要素を組み立てることによって、前記ソースシンボルの第 1 列と前記ソースシンボルの第 2 列とからパーティションの列へとサブ分割されるある値範囲を持つシンタックス要素列を組み立て、その組立の際に、前記シンタックス要素の少なくとも 1 つのサブグループについて、個々のシンタックス要素の値と前記パーティションの列の 1 つのパーティションとに基づいて前記ソースシンボルのそれぞれの数を決定し、前記シンタックス要素の値を得るために前記それぞれの数のソースシンボルの値を合計する、ステップと、を含み、
前記組立手段は、以前に復元されたソースシンボルに従って、前記パーティション間に 1 つ以上の境界を適応するよう構成されている、復号化方法。

【請求項 2】

前記ソースシンボルの第 2 列の中のソースシンボルは前記パーティションの列の中の 1 番目のパーティションに対応し、
前記パーティションの列は、第 1 パーティションが第 2 パーティションよりも値範囲のより高い値をカバーするように配置されている、請求項 1 に記載の復号化方法。

【請求項 3】

前記第 1 列からのソースシンボルと前記第 2 列からのソースシンボルとは異なるパーティションに対応している、請求項 1 に記載の復号化方法。

【請求項 4】

前記ソースシンボルの第 2 列の中のソースシンボルは前記パーティションの列の中の 1 番目のパーティションに対応し、
前記パーティションの列は、第 1 パーティションが第 2 パーティションよりも値範囲のより高い値をカバーするように配置されている、請求項 3 に記載の復号化方法。

【請求項 5】

前記ビットストリームの第 3 部分は少なくとも 1 つのスケールファクタを表している、請求項 1 に記載の復号化方法。

【請求項 6】

前記ソースシンボルの第 1 列の中のソースシンボルは 2 番目のパーティションに対応し、
前記第 1 復号器は前記ソースシンボルの第 1 列の中のソースシンボルを復号化するためにゴロムーライス符号を使用するよう構成されている、請求項 2 に記載の復号化方法。

【請求項 7】

再結合器により、前記ソースシンボルの第 1 列と前記ソースシンボルの第 2 列とを再結合して、ソースシンボル列を得るステップを更に含む、請求項 1 に記載の復号化方法。

【請求項 8】

前記第 1 復号器によって復元するステップは、前記ビットストリームの第 1 部分の符号語からソースシンボルの第 1 列のソースシンボルを復元することにより実行される、請求項 1 に記載の復号化方法。

【請求項 9】

請求項 1 に記載の復号化方法において、前記組立手段は、スキャン順序において先行するそれぞれの変換ブロックの絶対変換係数の少なくとも 1 つの復元された絶対変換係数レベ

ルに基づいて、又は組み立てられるべき絶対変換係数レベルのスキャン順序における位置に基づいて、又は組み立てられるべき絶対変換係数レベルの空間的に隣接する位置の変換係数の復元された絶対変換係数レベルの評価に基づいて、又は組み立てられるべき絶対変換係数レベルのスキャン順序において隣接する位置の変換係数の復元された絶対変換係数レベルの評価に基づいて、それぞれの変換ブロックの絶対変換係数の絶対変換係数レベルを組み立てしている間に、前記パーティション間に少なくとも1つの境界を適応するよう構成されている、復号化方法。

【請求項10】

前記シンタックス要素列はビデオをスケーラブルな方式で符号化している、請求項1に記載の復号化方法。

10

【請求項11】

復号化装置であって、

ビットストリームの第1部分を受信し、前記ビットストリームの第1部分の符号語からソースシンボルの第1列のソースシンボルを復元するよう構成された第1復号器と、

前記ビットストリームの第2部分を受信し、前記ビットストリームの第2部分の符号語からソースシンボルの第2列を復元するよう構成された第2復号器と、

組立手段であって、前記第1列からのソースシンボルと前記第2列からのソースシンボルとを回収し、それぞれの数のソースシンボルから各シンタックス要素を組み立てることによって、前記ソースシンボルの第1列と前記ソースシンボルの第2列とからパーティションの列へとサブ分割されるある値範囲を持つシンタックス要素列を組み立て、その組立の際に、前記シンタックス要素の少なくとも1つのサブグループについて、個々のシンタックス要素の値と前記パーティションの列の1つのパーティションとに基づいて前記ソースシンボルのそれぞれの数を決定し、前記シンタックス要素の値を得るために前記それぞれの数のソースシンボルの値を合計する、組立手段と、を含み、

20

前記組立手段は、以前に復元されたソースシンボルに従って、前記パーティション間に1つ以上の境界を適応するよう構成されている、復号化装置。

【請求項12】

前記ソースシンボルの第2列の中のソースシンボルは前記パーティションの列の中の1番目のパーティションに対応しており、

前記パーティションの列は、第1パーティションが第2パーティションよりも値範囲のより高い値をカバーするように配置されている、請求項11に記載の復号化装置。

30

【請求項13】

請求項11に記載の復号化装置において、前記第1列からのソースシンボルと前記第2列からのソースシンボルとは異なるパーティションに対応している、復号化装置。

【請求項14】

請求項13に記載の復号化装置において、前記ソースシンボルの第2列の中のソースシンボルは前記パーティションの列の中の1番目のパーティションに対応し、

前記パーティションの列は、第1パーティションが第2パーティションよりも値範囲のより高い値をカバーするように配置されている、復号化装置。

40

【請求項15】

前記パーティションの数は1、2、又は3である、請求項11に記載の復号化装置。

【請求項16】

前記ビットストリームの第3部分は少なくとも1つのスケールファクタを表している、請求項11に記載の復号化装置。

【請求項17】

前記装置は、プログラム可能な論理デバイス、プログラム可能なゲートアレイ、マイクロプロセッサ、コンピュータ又は電子回路の少なくとも一部である、請求項11に記載の復号化装置。

【請求項18】

前記第1復号器は、前記ビットストリームの第1部分の符号語からソースシンボルの第1

50

列のソースシンボルを復元することにより、前記第 1 列のソースシンボルを復元するよう構成されている、請求項 1 1 に記載の復号化装置。

【請求項 1 9】

前記組立手段は、スキャン順序において先行するそれぞれの変換ブロックの絶対変換係数の少なくとも 1 つの復元された絶対変換係数レベルに基づいて、又はスキャン順序において組み立てられるべき絶対変換係数レベルの位置に基づいて、又は組み立てられるべき絶対変換係数レベルの空間的に隣接する位置の変換係数の復元された絶対変換係数レベルの評価に基づいて、又は組み立てられるべき絶対変換係数レベルのスキャン順序において隣接する位置の変換係数の復元された絶対変換係数レベルの評価に基づいて、それぞれの変換ブロックの絶対変換係数の絶対変換係数レベルを組み立てしている間に、前記パーティション間に少なくとも 1 つの境界を適応するよう構成されている、請求項 1 1 に記載の復号化装置。

10

【請求項 2 0】

前記シンタックス要素列はビデオをスケーラブルな方式で符号化している、請求項 1 1 に記載の復号化装置。

【請求項 2 1】

コンピュータ上で実行したとき、請求項 1 に記載の復号化方法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を含む、コンピュータ読み取り可能な記憶媒体。

【請求項 2 2】

符号化装置であって、
パーティションの列にサブ分割される値範囲を持つシンタックス要素列を受信し、個々のシンタックス要素の値と前記パーティションの列の 1 つのパーティションとに基づいて、ソースシンボルの値の合計が個々のシンタックス要素の値となるようにシンタックス要素の少なくとも 1 つのサブグループを対応する数のソースシンボルに分解することにより、前記シンタックス要素列をソースシンボル列に変換する、分解手段と、
前記ソースシンボル列を受信し、前記ソースシンボル列をソースシンボルの第 1 列とソースシンボルの第 2 列へとサブ分割するサブ分割手段と、
前記第 1 列のソースシンボルを受信し、前記第 1 列のソースシンボルを符号化する第 1 符号器と、
前記第 2 列のソースシンボルを受信し、前記第 2 列のソースシンボルを符号化する第 2 符号器と、を含み、
前記分解手段は、以前に符号化されたソースシンボルに従って、前記パーティション間に 1 つ以上の境界を適応するよう構成されている、符号化装置。

20

30

【請求項 2 3】

前記ソースシンボルの第 2 列の中のソースシンボルは、前記パーティションの列の中の 1 番目のパーティションに対応しており、
前記パーティションの列は、第 1 パーティションが第 2 パーティションよりも値範囲のより高い値をカバーするように配置されている、請求項 2 2 に記載の符号化装置。

【請求項 2 4】

前記第 1 符号器又は第 2 符号器は、さらにスケールファクタを符号化するよう構成されている、請求項 2 2 に記載の符号化装置。

40

【請求項 2 5】

前記第 1 列からのソースシンボルと前記第 2 列からのソースシンボルとは異なるパーティションに対応している、請求項 2 2 に記載の符号化装置。

【請求項 2 6】

前記ソースシンボルの第 2 列の中のソースシンボルは、前記パーティションの列の中の 1 番目のパーティションに対応しており、
前記パーティションの列は、第 1 パーティションが第 2 パーティションよりも値範囲のより高い値をカバーするように配置されている、請求項 2 5 に記載の符号化装置。

50

【請求項 27】

前記ソースシンボルの第1列の中のソースシンボルは2番目のパーティションに対応し、前記第1符号器は前記ソースシンボルの第1列の中のソースシンボルをシンボル毎に符号化するためにゴロムーライズ符号を使用するよう構成されている、請求項22に記載の符号化装置。

【請求項 28】

前記装置は、プログラム可能な論理デバイス、プログラム可能なゲートアレイ、マイクロプロセッサ、コンピュータ又は電子回路の少なくとも一部である、請求項22に記載の符号化装置。

【請求項 29】

前記シンタックス要素列はビデオをスケーラブルな方式で符号化している、請求項22に記載の符号化装置。

【請求項 30】

以下のステップを含む方法によって符号化されたデータストリーム：

パーティションの列にサブ分割される値範囲を持つシンタックス要素列を受信するステップ、

個々のシンタックス要素の値と前記パーティションの列の1つのパーティションとに基づいて、ソースシンボルの値の合計が個々のシンタックス要素の値となるように前記シンタックス要素の少なくとも1つのサブグループを対応する数のソースシンボルに分解することにより、前記シンタックス要素列をソースシンボル列に変換するステップ、

前記ソースシンボル列をソースシンボルの第1列とソースシンボルの第2列へとサブ分割するステップ、

第1符号器によって前記第1列のソースシンボルを受信し、前記第1列のソースシンボルを符号化するステップ、

第2符号器によって前記第2列のソースシンボルを受信し、前記第2列のソースシンボルを符号化するステップ、

以前に符号化されたソースシンボルに従って、前記パーティション間に1つ以上の境界を適応するステップ。

【請求項 31】

前記シンタックス要素列はビデオをスケーラブルな方式で符号化している、請求項30に記載のデータストリーム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、例えばビデオ及びオーディオ圧縮のようなアプリケーションに使用可能な、エントロピー符号化および復号化に関する。

【背景技術】

【0002】

エントロピー符号化は、一般的には、ロスレスデータ圧縮の最も総称的な形態として考えられ得る。ロスレス圧縮は、元のデータ表現のために必要とされるよりも少ないビット数で、但し情報の損失が全くない状態で、離散的データを表現することを目的としている。離散的なデータは、テキスト、図形、画像、ビデオ、オーディオ、スピーチ、ファクシミリ、医療データ、気象データ、財務データの形態、または任意の他のデジタルデータの形態において提供され得る。

【0003】

エントロピー符号化においては、基礎となる離散データソースの特定の高レベルの特性が無視される場合が多い。そのため、任意のデータソースは、所与の $m - a r y$ のアルファベット値を持ち、かつ対応する（離散的）確率分布 $\{p_1, \dots, p_m\}$ によって特徴付けられるような、ソースシンボル列として与えられると考えられる。これらの概念的な設定において、シンボルあたりのビットの予想される符号語長に関して、任意のエントロピ

10

20

30

40

50

一符号化方法の下限が次式のエントロピーによって与えられる。

【数 1】

$$H = - \sum_{i=1}^m p_i \log_2 p_i . \quad (A1)$$

【0004】

ハフマン符号や算術符号は、（ある意味で）エントロピーの限界を近似できる現実的な符号の周知の例である。固定的な確率分布については、ハフマン符号は比較的容易に構築できる。ハフマン符号の最も魅力的な特性は、可変長符号（VLC）テーブルを使用することにより、その実装を効率的に実現し得るという点である。しかしながら、時間変動するソース統計、即ち変動するシンボル確率を扱う場合には、ハフマン符号とそれに対応するVLCテーブルとを適用することは、アルゴリズムの複雑さと実装に係る費用の点において、過大な要求を発生させてしまう。さらに、 $p_k > 0.5$ を持つ優勢アルファベット値がある場合には、（ランレングス符号化などの任意のアルファベット拡張子も使用しない）対応するハフマン符号は、かなりの冗長度を持つ可能性がある。ハフマン符号の他の欠点は、より高次の確率モデリングを扱う場合に、VLCテーブルの多数の集合が必要となる可能性があるという事実である。他方、算術符号化は、実質的にはVLCよりも複雑であるにも関わらず、適応型および高次の確率モデリングを扱う場合や、かなり偏りのある確率分布を扱う場合には、より一貫性があり適切な処理を提供できるという利点がある。実際に、この特徴は基本的に以下の事実から生まれるものである。すなわち、算術符号化は、少なくとも概念的には、確率推定のいかなる所与の値をも多かれ少なかれ直接的な方法で、結果として得られる符号語の一部分へとマップ（写像）するためのメカニズムを提供するという事実である。そのようなインターフェイスを備えることで、算術符号化は、一方では確率モデリング作業と確率推定作業との間の明確な分離を可能にし、他方では実際のエントロピー符号化、即ちシンボルから符号語へのマッピングを可能にしている。

【0005】

算術符号化とVLC符号化とに対する代替法は、PIPE符号化である。より正確には、PIPE符号化においては、ランダムシンボル変数の確率推定に沿って符号化処理をパイプライン化するために、ユニット区間が小さな1セットのばらばらの確率区間へと区分され（partitioned）ていく。この区分によれば、任意のアルファベット・サイズを持つ離散的なソースシンボルの入力列は、アルファベットシンボルの列へとマップされることができ、アルファベットシンボルの各々が1つの特別な確率区間へと割り当てられ、その確率区間は、次に特別仕様のエントロピー符号化処理によって符号化される。区間の各々が固定的な確率によって表現されることで、確率区間区分エントロピー（PIPE: probability interval partitioning entropy）符号化処理は、単純な可変長符号から可変長符号へのデザイン及びアプリケーションに基づいてもよい。確率モデリングは固定型または適応型のいずれであってもよい。しかし、PIPE符号化は、算術符号化よりも複雑性においてかなり低い一方で、依然としてVLC符号化よりも高い複雑性を有している。

【0006】

そこで、算術符号化およびVLC符号化の両方の長所を既に組み合わせたPIPE符号化と比較してもなお、符号化の複雑性と圧縮効率との間のより良好な妥協を達成できるようなエントロピー符号化スキームを獲得することは有益であろう。

【0007】

さらに一般的には、適度な符号化の複雑性において、より良好な圧縮効率そのものを達成できるようなエントロピー符号化スキームを獲得することは有益であろう。

【発明の概要】

【発明が解決しようとする課題】

【0008】

そこで、本発明の目的は、上述した要求を満たす、即ち符号化の複雑性と圧縮効率との間

のより良好な妥協を達成できる、エントロピー符号化の概念を提供することである。

【課題を解決するための手段】

【0009】

この目的は、請求項1乃至10に記載の復号化方法と、請求項11乃至20に記載の復号化装置と、請求項21に記載の記憶媒体と、請求項22乃至26に記載の符号化方法と、請求項27乃至30に記載の符号化装置によって達成される。

【0010】

本発明は次のような知見に基づいている。即ち、それぞれのシンタックス要素の値領域を n 個のパーティション（区分）からなる1つの列へと分解(decomposing)し、それぞれのパーティションの中に含まれるシンタックス要素の値 z の成分を別々に符号化すること、即ち少なくとも1つをVLC符号化により符号化し、少なくとも1つをPIPE符号化もしくは算術符号化又は他の任意のエントロピー符号化方法により符号化することは、使用される符号化スキームがシンタックス要素の統計に対してより良好に適応され得るという点で、適度な符号化オーバーヘッドで、圧縮効率をはるかに増大させる可能性がある、という知見である。従って、本発明の実施形態によれば、シンタックス要素は、ソースシンボル s_i のそれぞれの数 n へと分解され、ここで、 $i = 1 \dots n$ であり、ソースシンボル s_i のそれぞれの数 n は、それぞれのシンタックス要素の値領域がサブ分割されていく n 個のパーティション（140₁₋₃）からなる1列のどの部分に、それぞれのシンタックス要素の値 z が当てはまるか、に基づいており、その場合、ソースシンボル s_i のそれぞれの数の値の合計が z となり、 $n > 1$ である場合には、全ての $i = 1 \dots n-1$ について、 s_i の値が i 番目のパーティションの領域に対応するようになる。

【0011】

更に、VLC符号化と共にPIPE符号化を使用する本発明の幾つかの実施形態は、以下の事実を利用する。即ち、符号化の複雑性を一方とし圧縮効率を他方とした場合のそれらの間のより良好な妥協点は、複数の特別仕様のエントロピー符号器／復号器の間で、アルファベットシンボルがそれらの確率分布推定に基づいて分布されるような、PIPE符号化のカテゴリー化の特徴に加えて、更なるカテゴリー化ステージが提供された場合に達成される、という事実であり、その更なるステージでは、符号化されるべきソースシンボルが、VLC符号化を施される第1のサブストリームとPIPE符号化を施される第2のサブストリームとにサブ分割されており、第2のサブストリームは、PIPE符号器のシンボルアルファベット内で、即ち、アルファベットシンボルの列として表現されるという事実である。この方法によれば、適切なシンボル確率分布を有するソースシンボル、即ち本願明細書の導入部に記載した欠点がないVLC符号化によって効率的に符号化されるのに適した確率分布を有するソースシンボルは、より高い値を持つパーティションなどのようなVLC符号化されるシンボルとして分類され、他方、最も低い値を持つパーティションなどのような他のソースシンボルは、PIPE符号化されるシンボルとして扱われてPIPE符号化が施され、そのPIPE符号化の符号化複雑性は、VLC符号化よりも高度であるが、より良好な圧縮効率を示す。

【0012】

一実施形態によれば、シンタックス要素はそれぞれ整数個のソースシンボルに分解され、そのうちの1つがVLC符号化されるシンボルとして扱われ、他の1つがPIPE符号化されるシンボルとして扱われることが可能である。

【0013】

本発明の好適な態様は、添付した従属請求項の要旨となる。

【0014】

本発明の好適な実施形態は、以下に図面を参照しながら説明する。

【図面の簡単な説明】

【0015】

【図1a】一実施形態に係るエントロピー符号化装置のブロック図である。

【図1b】シンタックス要素からソースシンボルへの可能性のある分解を示す、一実施形

10

20

30

40

50

態に係る概略図である。

【図 1 c】シンタックス要素からソースシンボルへと分解する図 1 a の分解部の可能な操作モードを示す、一実施形態に従うフロー図である。

【図 2 a】一実施形態に係るエントロピー復号化装置のブロック図を示す。

【図 2 b】ソースシンボルからシンタックス要素を組み立てる図 2 a の組立部の可能な操作モードを示す、一実施形態に従うフロー図である。

【図 3】図 1 で使用され得る一実施形態に係る P I P E 符号器のブロック図である。

【図 4】図 2 で使用可能であり、一実施形態に係る図 3 の P I P E 符号器によって生成されたビットストリームを復号化するために適切な P I P E 復号器のブロック図である。

【図 5】多重化された部分的ビットストリームを持つデータパケットを示す、一実施形態に係る概略図である。

【図 6】固定サイズのセグメントを使用した代替的なセグメンテーションを持つデータパケットを示す、他の実施形態に係る概略図である。

【図 7】部分的ビットストリーム・インターリービングを使用する、一実施形態に係る P I P E 符号器のブロック図である。

【図 8】図 7 の符号器側における符号語バッファのステータスの例を概略的に示す、一実施形態に係る図である。

【図 9】部分的ビットストリーム・インターリービングを使用する、一実施形態に係る P I P E 復号器のブロック図である。

【図 1 0】符号語の単一の集合を用いた符号語インターリービングを使用する、一実施形態に係る P I P E 復号器のブロック図である。

【図 1 1】固定長のビット列のインターリービングを使用する、一実施形態に係る P I P E 符号器のブロック図である。

【図 1 2】図 1 1 の符号器側におけるグローバル・ビットバッファのステータスの例を概略的に示す、一実施形態に係る図である。

【図 1 3】固定長のビット列のインターリービングを使用する、一実施形態に係る P I P E 復号器のブロック図である。

【図 1 4】 $(0, 0.5)$ 内の均一な確率分布を仮定したとき、 $K = 4$ の区間への最適な確率区間離散化を示すグラフである。

【図 1 5】 $p = 0.38$ の L P B 確率についてのバイナリー事象のツリーと、ハフマン・アルゴリズムによって得られた関連する可変長符号と、を概略的に示す図である。

【図 1 6】テーブルエントリの最大数 L_m が与えられた場合に、最適な符号 C についての相対的なビットレートの増加率 $\rho(p, C)$ を推測し得るグラフである。

【図 1 7】 $K = 12$ の区間への理論上最適な確率区間区分と、最大数が $L_m = 65$ のテーブルエントリである場合の $V2V$ 符号を有する実際の設計と、についてのレートの増加率を示すグラフである。

【図 1 8】3 項選択ツリーを完全なバイナリー選択ツリーへと変換する例を概略的に示す図である。

【図 1 9】符号器（左側部分）と復号器（右側部分）とを含むシステムを示す、一実施形態に係るブロック図である。

【図 2 0】一実施形態に係るエントロピー符号化装置を示すブロック図である。

【図 2 1】一実施形態に係るエントロピー復号化装置を示すブロック図である。

【図 2 2】他の実施形態に係るエントロピー符号化装置を示すブロック図である。

【図 2 3】図 2 2 の符号器側におけるグローバル・ビットバッファのステータスの例を概略的に示す、一実施形態に係る図である。

【図 2 4】本発明の他の実施形態に係るエントロピー復号化装置を示すブロック図である。

【発明を実施するための形態】

【0016】

本発明の複数の実施形態を図面を参照しながら以下に説明する前に、これらの図面内にお

10

20

30

40

50

ける同一または同等の構成要素を示すために同一の参照符号が図面全体にわたって使用されている点、さらに、先の説明がその時点の図面の説明と矛盾しない限り、先の図面に關して記載されたそれら構成要素の説明が後続のいずれの図面にも適用される点について、留意されたい。

【0017】

図1aは、本発明の一実施形態に係るエントロピー符号化装置を示す。この装置は、サブ分割部100と、VLC符号器102と、PIPE符号器104とを含む。

【0018】

サブ分割部100は、ソースシンボル106の列を、ソースシンボルの第1サブ列108とソースシンボルの第2サブ列110とにサブ分割するよう構成されている。VLC符号器102は、サブ分割部100の第1出力に接続された入力を備え、さらに第1サブ列108のソースシンボルをシンボル毎に符号語へと変換し、第1ビットストリーム112を形成するよう構成されている。VLC符号器102はルックアップテーブルを含み、さらに、そのルックアップテーブル内でそれぞれの符号語をソースシンボル毎に参照するためのインデックスとして、ソースシンボルを個別に使用してもよい。VLC符号器は前記符号語を出力し、サブ列108内の後続のソースシンボルを続いて処理する結果、符号語の列が出力されて、その列内の各符号語がサブ列110内のソースシンボルのうちの正に1つと関連付けられる。符号語は異なる長さを持っていてもよく、いずれの符号語も他の符号語と一緒に接頭語を形成することがないように定義されてもよい。加えて、ルックアップテーブルは静的でよい。

【0019】

PIPE符号器104は、サブ分割部100の第2出力に接続された入力を備え、さらに、アルファベットシンボル列の形態で表現されるソースシンボルの第2サブ列110を符号化するよう構成されている。PIPE符号器104はまた、アルファベットシンボル列の先行するアルファベットシンボル内に含まれる情報に基づいて、アルファベットシンボル列の各アルファベットシンボルに対して、それぞれのアルファベットシンボルが持つ可能性のある値間における確率分布推定のための測度(measure)を割り当てるよう構成された割当部114と、複数のエントロピー符号器116であって、各符号器に送られてきたアルファベットシンボルをそれぞれの第2ビットストリーム118へと変換するよう構成されたエントロピー符号器116と、第2サブ列110の各アルファベットシンボルを複数のエントロピー符号器116のうちの選択された1つへと送るよう構成された選択部120であって、その選択がそれぞれのアルファベットシンボルに割り当てられた確率分布の推定のための上述した測度に依存する、選択部120と、を備えていてもよい。ソースシンボルとアルファベットシンボルとの間の関係は、アルファベットシンボル列内の他のアルファベットシンボルであって相互に直接続く可能性のあるものと一緒に、各アルファベットシンボルが1つのソースシンボルを表現するように、各アルファベットシンボルがサブ列110の正にその1つのソースシンボルと一意に関連付けられるような関係であってよい。

【0020】

以下により詳細に説明するように、ソースシンボルの列106は、解析可能なビットストリームのシンタックス要素の列であってもよい。解析可能なビットストリームとは、例えばスケラブル又は非スケラブルな状態でビデオ及び／又はオーディオのコンテンツを表現してもよく、その場合、シンタックス要素が例えば変換係数レベル、動きベクトル、動画参照インデックス、スケールファクタ、オーディオエンベロープ・エネルギー値又は同種のもを表現してもよい。シンタックス要素は、特に異なるタイプ又はカテゴリーのものであってもよく、その場合、異なる画像、異なるマクロブロック、異なるスペクトル成分などのような異なる部分に関する点を除き、同じタイプのシンタックス要素が例えば解析可能なビットストリーム内では同じ意味を有してもよく、他方では、例えば動きベクトルが動き予測残余を表す変換係数レベルを表現するシンタックス要素とは異なる意味を持つように、異なるタイプのシンタックス要素がビットストリーム内で異なる意味を持っ

てもよい。

【0021】

サブ分割部100は、シンタックス要素のタイプに基づいてサブ分割を実行するように構成されてもよい。つまり、サブ分割部100は、第1グループのタイプのシンタックス要素を第1サブ列108へと送り、第1グループとは異なる第2グループのタイプのシンタックス要素を第2サブ列110へと送ってもよい。サブ分割部100により実行されるサブ分割は、サブ列108内のシンタックス要素のシンボル統計がVLC符号器102によりVLC符号化されることに適するように設計されてもよい。即ち、本願明細書の導入部で記述したように、VLC符号化を使用した場合に、所定のシンボル統計に関する制限があるにも拘わらず、可能性な限り殆ど最小のエントロピーという結果をもたらす。他方、サブ分割部100は、他の全てのシンタックス要素を第2のサブ列110へと送ってもよく、その場合、VLC符号化に適さないシンボル統計を持つこれらシンタックス要素は、より複雑で、しかし圧縮率においてはより効率的なPIPE符号器104によって符号化される。

10

【0022】

後続の図においてさらに詳細に説明するケースにも当てはまるが、PIPE符号器104は、第2サブ列110の各シンタックス要素をアルファベットシンボルのそれぞれの部分的な列へと個別にマップして、アルファベットシンボルの上述した列124と一緒に形成するよう構成された、シンボライザ122を含んでもよい。換言すれば、もし例えばサブ列110のソースシンボルが既にアルファベットシンボルのそれぞれの部分的な列として表現されている場合には、シンボライザ122は存在しなくてもよい。例えばサブ列110内のソースシンボルが異なるアルファベットからなり、特にそれらアルファベットが可能なアルファベットシンボルの異なる個数を持つアルファベットである場合に、シンボライザ122は有利である。つまり、この場合には、シンボライザ122は、サブストリーム110内で到着するシンボルのアルファベットを調和させることができる。シンボライザ122は、例えばサブストリーム110内で到着するシンボルを二値化するよう構成されたバイナライザとして具体化されてもよい。

20

【0023】

上述したように、シンタックス要素は異なるタイプのものでもよい。この点は、サブ列110内のシンタックス要素についても真実である。シンボライザ122は、次に、二値化スキームなどのシンボル化のマッピングスキームであって、異なるタイプのシンタックス要素に対しては異なるスキームを使用して、サブ列110のシンタックス要素の個別のマッピングを実行するよう構成されてもよい。例えば、ユニナリー二値化スキーム、0次または1次の指数ゴロム二値化スキーム、又は切り詰められた(truncated)ユニナリー二値化スキーム、切り詰められかつ並べ替えられた0次の指数ゴロム二値化スキーム、又は非系統的な二値化スキームなど、特定の二値化スキームの例を以下の説明で示す。

30

【0024】

従って、エントロピー符号器116は二値アルファベットで作動するよう構成されてもよい。最後に、シンボライザ122は図1aに示すようにPIPE符号器104自身の一部として認識されてもよいことに留意されたい。しかし代替的に、バイナライザはPIPE符号器に対する外部要素として認識されてもよい。

40

【0025】

前段の注意書きと同様に、割当部114は、シンボライザ122と選択部120との間に直列的に接続されるよう示されているが、図3に関して後述するように、シンボライザ122の出力と選択部120の第1入力との間に接続され、割当部114の出力が選択部120の他の入力に接続されていてもよい。実際には、割当部114は各アルファベットシンボルに確率分布の推定のための上述した測度を付加する。

【0026】

図1aのエントロピー符号化装置の出力に関して言えば、当該出力は、VLC符号器102により出力された第1ビットストリーム112と、複数のエントロピー符号器116に

50

より出力された複数の第2のビットストリーム118とによって構成される。さらに後述するように、これら全てのビットストリームは並列的に伝送されてもよい。代替的に、エントロピー符号化装置の出力は、インターリーバー128を使用して1つの共通のビットストリーム126へとインターリーブされてもよい。図22～図24の実施形態がそのようなビットストリームのインターリーブの例を示す。図1にさらに示すように、PIPE符号器104自身が専用のインターリーバー130を備え、複数の第2のビットストリーム118を1つの共通のPIPE符号化されたビットストリーム132へとインターリーブしてもよい。そのようなインターリーバー130の可能性は、図5から図13に関して説明する以下の実施形態から導かれ得る。ビットストリーム132とビットストリーム112とは、並列的な構成では、図1aのエントロピー符号化装置の出力を表現してもよい。代替的に、もう1つのインターリーバー134が両方のビットストリームをインターリーブしてもよく、この場合には、インターリーバー130と134とは1つの2段階インターリーバー128の2つの段階を形成することになる。

10

【0027】

上述したように、サブ分割部100はシンタックス要素単位でサブ分割を実行してもよく、即ちサブ分割部100が操作するソースシンボルは全体的なシンタックス要素であってもよく、換言すれば、サブ分割部100はシンタックス要素からなるユニット単位で作動してもよい。

【0028】

しかしながら、代替的な実施形態によれば、図1aのエントロピー符号化装置は分解部136を備え、解析可能なビットストリーム138内のシンタックス要素を、サブ分割部100に入力されるソースシンボル列106の1つ以上のソースシンボルへと個別に分解してもよい。特に、分解部136は、各シンタックス要素をそれぞれ整数個のソースシンボルへと個別に分解することで、シンタックス要素の列138をソースシンボル列106へと変換するよう構成されてもよい。その整数の個数はシンタックス要素間で変化してもよい。特に、シンタックス要素の幾つかは分解部136によって変化なしのままとしてもよく、その一方で、他のシンタックス要素は2つ又は少なくとも2つのソースシンボルに分解されてもよい。サブ分割部100は、そのような分解されたシンタックス要素のソースシンボルの1つをソースシンボルの第1のサブ列108に送ってもよく、さらに、同じ分解されたシンタックス要素のソースシンボルの他の1つをソースシンボルの第2のサブ列110に送ってもよい。上述したように、ビットストリーム138内のシンタックス要素は異なるタイプのものでも良く、分解部136は、シンタックス要素のタイプに応じて個別の分解を実行するよう構成されてもよい。好適には、分解部136は、後に復号器側で使用される整数個のソースシンボルからそれぞれのシンタックス要素への予め決定された一意的な逆マッピングであって全てのシンタックス要素に共通するものが存在するように、シンタックス要素の個別の分解を実行する。

20

30

【0029】

例えば、分解部136は、解析可能なビットストリーム138内のシンタックス要素 z を2つのソースシンボル x と y とに分解する際に、 $z = x + y$ 、 $z = x - y$ 、 $z = x \cdot y$ 又は $z = x : y$ となるように、分解するよう構成されてもよい。この方法により、サブ分割部100は、シンタックス要素を2つの成分、即ちソースシンボルストリーム106のソースシンボルであって、そのうちの1つは x のように圧縮効率の観点からVLC符号化されることに適したものであり、他の1つは y のようにVLC符号化には適さず、従って第1サブ列108ではなく第2のサブ列110へと送られるものに分解する。分解部136によって使用されるこの分解は、全単射(bijective)である必要はない。しかし、上述したように、もしこの分解が全単射でない場合には、分解部136がその中から選ぶ可能性のある可能な分解方法によるシンタックス要素の一意的な回復を可能にするような、逆マッピングが存在しなければならない。

40

【0030】

これまで、異なるシンタックス要素の取り扱いについて異なる可能性を説明してきた。そ

50

のようなシンタックス要素またはケースが存在するか否かについては任意である。しかし、ここで以下に説明する実施形態は、以下の原理に従って分解部 136 により分解されるシンタックス要素に焦点を当てるものである。

【0031】

図 1c に示すように、分解部 136 は、解析可能なビットストリーム 138 内の所定のシンタックス要素 z を複数の段階において分解するように構成されている。2 又は 3 の段階が存在してもよい。それら段階とは、図 1b に示すように、シンタックス要素 z の値領域を 2 つ以上の隣接するサブ区間またはサブ領域へと分割するためである。シンタックス要素の値領域は、2 つの無限の終点を持ってもよく、1 つの終点だけが無限であってもよく、複数の有限の終点を持ってもよい。図 1b においては、シンタックス要素の値領域は、例示的にパーティション 140₁₋₃ にサブ分割される。図 1c に示すように、もしシンタックス要素が、第 1 パーティション 140₁ の境界 142、即ちパーティション 140₁ と 140₂ とを分けている上限以上である場合には、シンタックス要素は第 1 パーティション 140₁ の境界の限界 1 の分だけ減算され、 z' は、それが第 2 パーティション 140₂ の境界 144、即ちパーティション 140₂ と 140₃ とを分ける上限以上であるか否かを再度チェックされる。もし z' が境界 144 以上である場合には、第 2 パーティション 140₂ の境界の限界 2 の分だけ減算され、結果として z'' がもたらされる。 z が限界 1 よりも小さい 1 番目の場合には、シンタックス要素 z はそのままサブ分割部 100 に送られる。 z が限界 1 と限界 2 との間にある場合には、シンタックス要素 z はタプル(tuple) (限界 1, z') としてサブ分割部 100 に送られ、このとき $z = \text{限界 1} + z'$ である。 z が限界 2 以上の場合には、シンタックス要素 z はトリプル(triplet) (限界 1, 限界 2 - 限界 1, z'') としてサブ分割部 100 に送られ、このとき $z = \text{限界 1} + \text{限界 2} + z''$ である。第 1 の (又は唯一の) 成分、即ち z 又は限界 1 は、サブ分割部 100 によって符号化されるべき第 1 のソースシンボルを形成し、第 2 の成分、即ち z' 又は限界 2 - 限界 1 は、存在する場合にはサブ分割部 100 によって符号化されるべき第 2 のソースシンボルを形成し、第 3 の成分、即ち z'' は、存在する場合にはサブ分割部 100 によって符号化されるべき第 3 のソースシンボルを形成する。このように、図 1b と図 1c とに示す実施形態に従ってシンタックス要素は 1 ~ 3 のソースシンボルのいずれかにマップされるが、しかし、多かれ少なかれ最大数のソースシンボルへの一般化は上述の説明から容易に導くことができ、そのような代替例も以下に説明する。

【0032】

いずれにせよ、以下に記載する実施形態に従えば、これら全ての異なる成分または結果として得られるソースシンボルは符号化の選択肢の中で符号化される。それら成分のうちの少なくとも 1 つはサブ分割部によって PIPE 符号器 104 へと送られ、さらに、それらの中の少なくとも他の 1 つは VLC 符号器 102 へと送られる。

【0033】

具体的で有利な実施形態を以下にさらに詳細に説明する。

【0034】

エントロピー符号化装置の一実施形態を上述した後で、エントロピー復号化装置の一実施形態を図 2a に関連して説明する。図 2a のエントロピー復号化装置は、VLC 復号器 200 と、PIPE 復号器 202 とを含む。VLC 復号器 200 は、第 1 ビットストリーム 206 の符号語から第 1 サブ列 204 のソースシンボルを符号単位で復元するよう構成されている。第 1 ビットストリーム 206 は図 1 のビットストリーム 112 に対応し、同様に、サブ列 204 は図 1a のサブ列 108 に対応する。PIPE 復号器 202 は、アルファベットシンボル列の形態で表現されるソースシンボルの第 2 サブ列 208 を復元するよう構成されており、さらに、複数のエントロピー復号器 210 と、割当部 212 と、選択部 214 とを含む。複数のエントロピー復号器 210 は、第 2 のビットストリーム 216 のそれぞれをアルファベットシンボル列のアルファベットシンボルへと変換するよう構成されている。割当部 212 は、アルファベットシンボル列の以前の復元済みのアルファベットシンボル内に含まれる情報に基づいて、復元されるべきソースシンボルの第 2 サブ列

208を表現しているアルファベットシンボル列の各アルファベットシンボルに対して、それぞれのアルファベットシンボルが持つ可能性のある値の間の確率分布の推定のための測度を割り当てるよう構成されている。この目的のため、割当部212は、選択部214の出力と1つの入力との間に直列的に接続されてもよく、他方、選択部214の他の入力には、エントロピー復号器210の出力がそれぞれ接続されている。選択部214は、複数のエントロピー復号器210のうちの選択された1つから、アルファベットシンボル列の各アルファベットシンボルを回収するよう構成されており、その選択は、それぞれのアルファベットシンボルに割り当てられた測度に依存する。換言すれば、選択部214は割当部212と共に、アルファベットシンボル列の先行するアルファベットシンボル内に含まれる情報を調査することによって得られたエントロピー復号器210間の順序で、エントロピー復号器210によって取得されたアルファベットシンボルを回収するよう作動できる。さらに換言すれば、割当部212と選択部214とは、アルファベットシンボル毎にアルファベットシンボルの元の順序を復元できる。次のアルファベットシンボルを予想することに加え、割当部212は、それぞれのアルファベットシンボルのための確率分布の推定の上述した測度を決定することができ、それを使用することで、選択部214は、エントロピー復号器210の中から、このアルファベットシンボルの実際の値を回収させるための復号器を選択する。より正確には、かつ以下により詳細に説明するように、PIPE復号器202は、アルファベットシンボル列の形態で表現されるソースシンボルのサブ列208を、アルファベットシンボルに対して順次要求しているアルファベットシンボル要求に応じて復元するよう構成されてもよく、割当部212は、復元されるべきソースシンボルの第2サブ列208を表現するアルファベットシンボル列のアルファベットシンボルに対する各要求に対して、それぞれのアルファベットシンボルが持つ可能性のある値の間の確率分布の推定の上述した測度を割り当てるよう構成されてもよい。また、選択部214は、復元されるべきソースシンボルの第2サブ列208を表現するアルファベットシンボル列のアルファベットシンボルに対する各要求に対し、複数のエントロピー復号器210のうちの選択された1つから、アルファベットシンボル列のそれぞれのアルファベットシンボルを回収するよう構成されてもよく、その選択は、それぞれのアルファベットシンボルに対するそれぞれの要求に対して割り当てられた測度に基づいてもよい。復号化側における要求を一方とし、符号化側におけるデータフロー又は符号化を他方とした場合のそれらの間の調和については、図4を参照しながらより詳細に説明する。

【0035】

図2aのエントロピー復号化装置は、ソースシンボルの第1サブ列204とソースシンボルの第2のサブ列208と共にソースシンボルの1つの共通の列218を形成するので、任意ではあるが、第1サブ列204と第2サブ列208とを再結合してソースシンボルの共通の列218を取得するよう構成された再結合部220を含む。このソースシンボルの共通の列218は、図1aの列106の復元をもたらす。

【0036】

図1に関して上述した説明と一致して、第1及び第2のサブ列204と208とのソースシンボルは、解析可能なビットストリームのシンタックス要素であってもよい。この場合、再結合部220は、第1及び第2のサブ列204と208とを介して到着するソースシンボルを、シンタックス要素間の順序を定義するある解析規則によって規定された順序でインターリーブすることで、シンタックス要素から成る列218のこの解析可能なビットストリームを復元するよう構成されてもよい。特に、シンタックス要素は上述したように異なるタイプのものでもよく、再結合部220は、VLC復号器200から第1グループのタイプのシンタックス要素をサブ列204を介して回収または要求し、さらに、PIPE復号器202から第2グループのタイプのシンタックス要素をサブ列208を介して回収または要求するよう構成されてもよい。従って、第1グループのタイプのシンタックス要素が次の位置にあると直近で述べた解析規則が示す場合には常に、再結合部220は、サブ列204の実際のソースシンボルを共通の列218へと挿入し、その他の場合にはサブ列208から挿入する。

【0037】

また、PIPE復号器202は、選択部214の出力と再結合部220の入力との間に接続された、デ・シンボライザ（逆シンボル化部）222を含んでもよい。図1に関連する上述の説明と同様に、デ・シンボライザ222は、PIPE復号器202に対する外部要素として認識されてもよく、更に再結合部220の後方、即ち再結合部220の出力側に配置されてもよい。デ・シンボライザ222は、アルファベットシンボルの部分的な列の単位で、選択部214により出力されたアルファベットシンボル列224をソースシンボルに、即ちサブ列208のシンタックス要素に、リマップ（逆写像）するよう構成されてもよい。再結合部220と同様に、デ・シンボライザ222は、アルファベットシンボルの可能性のある部分的な列の構成について承知している。特に、デ・シンボライザ222は、選択部214から直前に受け取ったアルファベットシンボルを分析して、それら直前に受け取ったアルファベットシンボルが、それぞれのシンタックス要素のそれぞれの値に関連するアルファベットシンボルの有効な部分的な列をもたらしうか否か、又はそうでないか否か、及びどのアルファベットシンボルが次に欠損しているか、などを確認してもよい。換言すれば、デ・シンボライザ222は常に、それぞれのシンタックス要素の受領を完成させるために更なるアルファベットシンボルを選択部214から受け取る必要があるか否か、また従って、選択部214により出力されたアルファベットシンボルの各々1つがどのシンタックス要素に属しているか、について承知している。この目的で、デ・シンボライザ222は、異なるタイプのシンタックス要素については異なっている、シンボル化の（デ・）マッピングスキームを使用してもよい。同様に、割当部212は、選択部214によって任意のエントロピー復号器210から回収されるべき現在のアルファベットシンボルと、シンタックス要素のうちの各々1つと、の関連について承知しており、このアルファベットシンボルの確率分布の推定の上述した測度を、それと対応するように、即ち関連するシンタックス要素のタイプに基づいて設定してもよい。さらに、割当部212は、現在のアルファベットシンボルの同一の部分的な列に属する異なるアルファベットシンボル同士を区別し、それらのアルファベットシンボルに対して確率分布の推定の測度を異なるように設定してもよい。この点については、以下により詳細に説明する。以下に記載するように、割当部212は、コンテキストをアルファベットシンボルに割り当てるよう構成されてもよい。その割当は、シンタックス要素のタイプ及び／又は現在のシンタックス要素のアルファベットシンボルの部分的な列内の位置に基づいてもよい。選択部214によって選択された任意のエントロピー復号器210から回収されるべき現在のアルファベットシンボルに対し、割当部212がコンテキストを割り当てると直ぐに、アルファベットシンボルは、それに関連付けられた確率分布の推定の測度を本来的に持ってもよい。なぜなら、各コンテキストがそれに関連付けられたその推定の測度を有するからである。更に、コンテキストー及びその関連付けられた確率分布の推定の測度ーは、エントロピー復号器210からそれまでに回収されてきたそれぞれのコンテキストのアルファベットシンボルの実際の統計に従って適応されてもよい。この点については、以下により詳細に説明する。

【0038】

図1に関する上述の説明と同様に、シンタックス要素におけるサブ列204、208の上述したソースシンボル間の対応は、1対1の対応ではない可能性がある。むしろ、シンタックス要素は整数個のソースシンボルへと分解されており、その個数が、シンタックス要素間で変化しつつ、しかしいずれの場合にも、少なくとも1つのシンタックス要素について2以上となってもよい。上述したように、以下の実施形態は、これらの種類のシンタックス要素を扱うことに焦点を当てたものであり、しかも、他の種類のシンタックス要素は存在しない可能性もある。

【0039】

上述したシンタックス要素を扱うために、図2aのエントロピー復号化装置は、図1aの分解部136により実行された分解を元に戻すよう構成された、組立部224を含んでもよい。特に、組立部224は、それぞれ整数個のソースシンボルから各シンタックス要素

を個別に組み立てることにより、ソースシンボル列 2 1 8 から、又はもし再結合部 2 2 0 がない場合にはサブ列 2 0 4 と 2 0 8 とから、シンタックス要素の列 2 2 6 を組み立てるよう構成されてもよく、このとき、整数個のソースシンボルのうちの 1 つのソースシンボルが第 1 サブ列 2 0 4 に属し、かつ同じシンタックス要素の整数個のソースシンボルのうちの他の 1 つのソースシンボルが第 2 サブ列 2 0 8 に属していてもよい。この方法では、所定のシンタックス要素は、V L C 復号化に適する成分を P I P E 復号化経路を通して送られるべき他の成分から分離するために、符号器側において分解されていたと見ることもできる。上述の説明と同様に、シンタックス要素は異なるタイプのものであってもよく、組立部 2 2 4 は、シンタックス要素のタイプに基づいて個別の組立を実行するよう構成されてもよい。特に、組立部 2 2 4 は、論理的または数学的にそれぞれのシンタックス要素の整数個のソースシンボルを結合することにより、それぞれのシンタックス要素を取得するよう構成されてもよい。例えば、組立部 2 2 4 は、各シンタックス要素について、1 つのシンタックス要素の第 1 および第 2 のソースシンボルに対し、+、-、: 又は・を適用するよう構成されてもよい。

10

【0040】

しかし、上述したように、本明細書の以下に説明する実施形態は、分解部 1 3 6 によって図 1 b と図 1 c に示すように分解されるシンタックス要素、及びそれらシンタックス要素に関して説明された他の選択肢に焦点を当てるものである。図 2 b は、組立部 2 2 4 がどのように機能してこれらのシンタックス要素をそのソースシンボル 2 1 8 から復元するかを示している。

20

【0041】

図 2 b に示すように、組立部 2 2 4 は、入来するソースシンボル $s_1 \sim s_x$ から、そのようなシンタックス要素 z を複数の段階において組み立てるよう構成され、ここで、この実施形態において x は 1 ~ 3 の任意の数である。2 つ以上の段階が存在してもよい。図 2 b に示すように、組立部 2 2 4 は、事前に z を第 1 シンボル s_1 であると設定し、 z が第 1 の限界 1 と等しいか否かについてチェックする。もし等しくない場合には、 z が発見されたということになる。もし等しい場合には、組立部 2 2 4 は、ソースシンボルストリーム 2 1 8 の次のソースシンボル s_2 を z に加え、再度、この z が限界 2 と等しいか否かをチェックする。等しくない場合には、 z が発見されたということになる。等しい場合には、組立部 2 2 4 はソースシンボル列 2 1 8 の次のソースシンボル s_3 を z に加え、最終的な形態における z を得る。多かれ少なかれ最大数のソースシンボルへの一般化は上述の説明から容易に導くことができ、そのような選択肢も以下に説明する。

30

【0042】

いずれにせよ、以下に記載する実施形態に従えば、これら全ての異なる成分または結果として得られるソースシンボルは符号化の選択肢の中で符号化される。それらの内の少なくとも 1 つはサブ分割部によって P I P E 符号器 1 0 4 へと送られ、さらにそれらの内の少なくとも他の 1 つは V L C 符号器 1 0 2 へと送られる。

【0043】

特に有利な実施形態について、以下にさらに詳細に説明する。これらの詳細は、シンタックス要素の値領域の分割の好適な可能性と、ソースシンボルを符号化するために使用されるエントロピー V L C 及び P I P E の符号化スキームとに焦点を当てるものである。

40

【0044】

更に、図 1 に関して上述したように、図 2 a のエントロピー復号化装置は、第 1 のビットストリーム 2 0 6 と複数の第 2 のビットストリーム 2 1 6 とを、別個に又はインターリーブされた形態で、インターリーブされたビットストリーム 2 2 8 を経由して受け取るように構成されてもよい。後者の場合、図 2 a のエントロピー復号化装置は、インターリーブされたビットストリーム 2 2 8 をデ・インターリーブして、第 1 のビットストリーム 2 0 6 を一方とし、複数の第 2 のビットストリーム 2 1 6 を他方として取得するよう構成された、デ・インターリーバ 2 3 0 を含んでもよい。図 1 に関する上述の説明と同様に、デ・インターリーバ 2 3 0 は 2 つの段階にサブ分割することができる。即ち、1 番目の段

50

階は、インターリーブされたビットストリーム 228 を、ビットストリーム 206 を一方とし、第 2 のビットストリーム 216 のインターリーブされた形態 234 を他方とする、2 つの部分へとデ・インターリーブするためのデ・インターリーバー 232 であり、2 番目の段階は、前記ビットストリーム 234 をデ・インターリーブして個別のビットストリーム 216 を得るためのデ・インターリーバー 236 である。

【0045】

以上のように、図 1 a はエントロピー符号化装置を示し、図 2 a は、図 1 のエントロピー符号化装置によって取得された符号化の結果を復号化するのに適したエントロピー復号化装置を示すものである。図 1 a と図 2 a に示された要素のうちの多数に係る詳細を、更なる図面に基づいてより詳細に説明する。従って、以下の説明ではこれらの詳細について触れるが、これらの詳細が上述した実施形態の中で分離して実装し得る限り、図 1 a と図 2 a に示す実施形態にも適用されると認識されるべきである。インターリーバー 128 及びデ・インターリーバー 230 に関してのみ、幾つかの追加的な注意点がある。特に、ビットストリーム 112 と 118 のインターリーブングは、伝送するためにビットストリームが 1 つのチャンネルへと多重化されなければならない場合に、好適であり得る。この場合、ある最大復号化遅延に従うというような、満たすべき所定の条件に従うためには、VLC ビットストリーム 112 を一方とし、PIPE 符号化ビットストリーム 118 を他方として、インターリーブすることが好適となり得る。換言すれば、シンタックス要素およびソースシンボルがそれぞれ復号化側において回収可能である時間を一方とし、解析可能なビットストリーム内のそれらの位置に応じた時間における相対的なずれを他方とした場合に、それら時間同士の相対的な時間的ずれが所定の最大遅延を超えないことが必要となり得る。この問題を解決するための多数の選択肢を以下に説明する。その可能性のうちの 1 つは、エントロピー符号器 116 が、可変長符号器タイプであって、アルファベットシンボル列を符号語へとマッピングするよう構成されており、かつエントロピー復号器 210 が逆マッピングを行うことを含む。VLC ビットストリーム 112 と PIPE ビットストリーム 118 との符号語は、符号語の境界が復号器側において一意に決定可能な状態のままとなるように、これらいずれのビットストリームのどの符号語も、他のいずれのビットストリームのどの符号語の接頭語ではないように選択されてもよく、但し必須という訳ではない。いずれにせよ、インターリーバー 128 は、第 1 ビットストリーム 112 および第 2 ビットストリーム 118 内の符号語に関する符号語エントリの列を、順次リザーブしバッファするよう構成されてもよく、その順序は、選択部 120 によって複数のエントロピー符号器 116 へと送られたアルファベットシンボル列 124 のアルファベットシンボルが、それぞれのエントロピー符号器 116 においてそれぞれの符号語へとマップされるべき新たなアルファベットシンボル列の開始となる結果をもたらし、他方、第 1 のサブストリーム 108 の新たなソースシンボルが、VLC 符号器 102 によってマップされるような、それぞれの順序に基づいている。換言すれば、インターリーバー 128 は、ビットストリーム 112 の符号語を、それら符号語が VLC 符号化によって取得されたソースシンボルの順序で、つまりサブストリーム 108 内とソースシンボル 106 内におけるそれぞれの順序で、共通ビットストリーム 126 内に挿入する。エントロピー符号器 116 によって出力された符号語は、共通ビットストリーム 126 内の VLC ビットストリーム 112 の連続した符号語の間に挿入される。割当部 114 と選択部 120 とがそれぞれ実行するアルファベットシンボルの PIPE 符号化のカテゴリー化によって、エントロピー符号器 116 の各符号語が、サブ列 110 の異なるソースシンボルのアルファベットシンボルを、その中に符号化された状態で持つ。共通ビットストリーム 126 内における、PIPE 符号化されたビットストリーム 118 の符号語の、相互間の位置、およびビットストリーム 112 の VLC 符号語に対する位置は、それぞれ各符号語内における符号化された最初のアルファベットシンボルにより、即ち、時間的に最も古いものにより決定される。アルファベットシンボル列 124 内のこれら最初のアルファベットシンボルがビットストリーム 118 の符号語に符号化されていく順序が、共通ビットストリーム 126 内におけるビットストリーム 118 の符号語同士の間の順序を決定する。一方、ビットストリーム 1

10

20

30

40

50

12のVLC符号語に対しては、ビットストリーム118の符号語に符号化されたこれら最初のアルファベットシンボルが属するソースシンボルは、ビットストリーム112の連続的な符号語のうちのどの間に、任意のビットストリーム118のそれぞれの符号語が配置されるべきかを決定する。特に、連続的なVLC符号語であって、それらの間にビットストリーム118の符号語が配置されるべきVLC符号語の間には、サブ列110のソースシンボルがサブ分割されていないソースシンボル列106の元の順序に従って配置され、そのソースシンボル列106とは、ビットストリーム118のそれぞれの符号語へと符号化される最初のアルファベットシンボルが帰属する列である。インターリーバ128は、インターリーブされた符号語の共通ビットストリーム126を得るために、上述した符号語エントリに順次挿入された符号語を除外するよう構成されてもよい。既に上述したように、エントロピー符号器116は、それぞれのエントロピー符号器116のためにリザーブされている符号語エントリにそれらの符号語を順次挿入するよう構成されてもよく、また、選択部120は、第2サブ列110のソースシンボルを表すアルファベットシンボルを、ソースシンボル列106内で第1サブ列108及び第2サブ列110のソースシンボルがインターリーブされた順序を保つ順序で転送するよう構成されてもよい。

【0046】

エントロピー符号器116のうちのいくつかは選択されることがかなり稀であるため、その非常に稀にしか使用されないエントロピー符号器116内で有効な符号語を取得するために、あまりに時間が掛かり過ぎるという状況に対処する目的で、追加的な手段が提供されてもよい。そのような手段の例を、以下により詳細に説明する。この場合、特にインターリーバ128とエントロピー符号器116とは、それまで集められたそれらのアルファベットシンボルと、上述した符号語エントリにそれまで挿入された符号語とを、復号化側においてフラッシュ手順の時刻が予測又はエミュレートできるような方法で、それぞれフラッシュするよう構成されてもよい。

【0047】

復号器側においては、デ・インターリーバ230が逆の意味で作動してもよい。つまり、上述した解析スキームに従って、復号化されるべき次のソースシンボルがVLC符号化されたシンボルであるときはいつも、共通ビットストリーム228内の現在の符号語がVLC符号語として認識され、ビットストリーム206内でVLC復号器200へと送られる。反対に、サブ列208の任意のPIPE符号化されたシンボルに属する任意のアルファベットシンボルが最初のアルファベットシンボルであるときはいつも、即ち、それぞれのビットストリーム216の1つの符号語を、それぞれのエントロピー復号器210によってそれぞれのアルファベットシンボル列へと新たにマッピングすることが必要となるときはいつも、共通ビットストリーム228の現在の符号語はPIPE符号化された符号語として認識され、それぞれのエントロピー復号器210へと送られる。次の符号語の境界の検出、即ち復号器200と202とのいずれかにそれぞれ送られたばかりの符号語の終端から、入来しているインターリーブされたビットストリーム228内におけるその終端までの、次の符号語の長さ検出は延期されてもよく、さらに、復号器200と202とのいずれが上述した規則に従って次の符号語の専用の受取用復号器となるかという知識の下に、つまり、この知識に基づいて受取用復号器によって使用される符号帳が認識され、それぞれの符号語が検出可能となったときに、長さ検出が実行されてもよい。他方、もし符号帳が、復号器200と202との間でいずれが受取用復号器であるかについての先験的な知識がない状態でその符号語境界が検出できるように設計されていた場合には、符号語の分離は並列的に実行することが可能となる。いずれにせよ、インターリービングによって、ソースシンボルはエントロピー復号化された形態で、即ちそれらの正しい順序で合理的な遅延を持つソースシンボルとして、復号器において利用可能となる。

【0048】

エントロピー符号化装置およびそれぞれのエントロピー復号化装置のための上記実施形態について説明した後で、次に、上述したPIPE符号器およびPIPE復号器についてのより詳細な実施形態の説明を行う。

10

20

30

40

50

【0049】

本発明の実施形態に係るPIPE符号器を図3に示す。この符号器は、図1aのPIPE符号器として使用されてもよい。このPIPE符号器は、ソースシンボル1のストリームを2つ以上の部分的ビットストリーム12の集合へロスレスに変換する。各ソースシンボル1は、1つ以上のカテゴリー又はタイプからなる集合のうちの1つのカテゴリー又はタイプと関連付けられてもよい。一例として、カテゴリーはソースシンボルのタイプを指定することができる。ハイブリッドビデオ符号化のコンテキストにおいては、マクロブロック符号化モード、ブロック符号化モード、参照画像インデックス、動きベクトル差、分割フラグ、符号化されたブロックフラグ、量子化パラメータ、変換係数レベルなどに対し、別個のカテゴリーが関連付けられてもよい。オーディオ、スピーチ、テキスト、文書、または汎用データ符号化などの他のアプリケーション分野において、ソースシンボルのさまざまな分類が可能である。

10

【0050】

一般に、各ソースシンボルは有限または可算無限の集合の値を取ることができ、可能性のあるソースシンボル値の集合は、異なるソースシンボルカテゴリーごとに異なることができる。符号化および復号化アルゴリズムの複雑さを低減するとともに、様々なソースシンボルとソースシンボルカテゴリーのための一般的な符号化および復号化の設計を可能にするため、ソースシンボル1はバイナリー決定(binary decisions)の順序付けられた集合に変換され、その後、これらのバイナリー決定は単純なバイナリー符号化アルゴリズムによって処理される。そのため、バイナライザ2は各ソースシンボル1の値をビン3の列(またはストリング)へ全単射的にマップする。ビン3の列は順序付けられたバイナリー決定の1つの集合を表す。各ビン3またはバイナリー決定は、2つの値からなる1セットのうちの1つの値、例えば値0と1のうちの一方をとることができる。二値化スキームは、異なるソースシンボルカテゴリーについては異なってもよい。特定のソースシンボルカテゴリーについての二値化スキームは、その特定のカテゴリーのための可能なソースシンボル値の集合及び／又はソースシンボルの他の特性に基づくことができる。表1は、可算無限集合のための3つの例示的な二値化スキームを示す。これらの可算無限集合のための二値化スキームは、シンボル値の有限集合にも適用することができる。特に、シンボル値の大きな有限集合の場合は、(使用されないビン列から生じる)非効率性を無視することができるが、このような二値化スキームの普遍性は、複雑性とメモリ要件の面で利点を提供する。シンボル値の小さな有限集合については、可能性のあるシンボル値の数に対して二値化スキームを適応することが(符号化効率の点から)好ましいことが多い。表2は、8つの値から成る有限集合のための3つの例示的な二値化スキームを示す。有限集合のための二値化スキームは、ビン列の有限集合が冗長度のない符号を表す(かつ潜在的にビン列を並び換える)ように、ビン列の一部を変更することによって、可算無限集合のための普遍的二値化スキームから導出することができる。例えば表2の切り詰めユニナリー二値化スキームは、普遍的ユニナリー二値化のソースシンボル7のためのビン列を変更することによって作られた(表1参照)。表2の次数0の、切り詰められかつ並べ換えられた指数ゴロム二値化は、普遍的指数ゴロム次数0の二値化のソースシンボル7のためのビン列を変更することによって(表1参照)、及びビン列を並び換えることによって作られた(シンボル7のための切り詰められたビン列はシンボル1に割り当てられた)。シンボルの有限集合については、表2の最後のコラムに例示されるように、非系統的／非普遍的な二値化スキームを使用することも可能である。

20

30

40

【0051】

【表 1】

[表1]可算無限集合(又は大きな有限集合)のための二値化の例

シンボル値	ユーナリー二値化	次数0の指数ゴロム二値化	次数1の指数ゴロム二値化
0	1	1	10
1	01	010	11
2	001	011	0100
3	0001	0010 0	0101
4	0000 1	0010 1	0110
5	0000 01	0011 0	0111
6	0000 001	0011 1	0010 00
7	0000 0001	0001 000	0010 01
...	...	...	...

10

【 0 0 5 2 】

【表 2】

[表2]有限集合のための二値化の例

シンボル値	切り詰められたユーナリー二値化	切り詰められかつ並べ換えられた0次の指数ゴロム二値化	非系統的な二値化
0	1	1	000
1	01	000	001
2	001	010	01
3	0001	011	1000
4	0000 1	0010 0	1001
5	0000 01	0010 1	1010
6	0000 001	0011 0	1011 0
7	0000 000	0011 1	1011 1

30

【 0 0 5 3 】

バイナライザ 2 によって作成されたビン列の各ビン 3 は、パラメータ割当部 4 に順に送られる。パラメータ割当部は各ビン 3 に 1 つ以上のパラメータからなる 1 つの集合を割り当て、パラメータの関連付けられた集合を有するビン 5 を出力する。パラメータの集合は、符号器と復号器において全く同じ方法で決定される。パラメータの集合は、以下のパラメータのうちの 1 つ以上から構成されてもよい。

- ー現在のビンに関する 2 つの可能なビン値のうちの 1 つについての確率推定の測度
- ー現在のビンに関する、劣勢確率 (less probable) または優勢確率 (more probable) を持つビン値についての確率推定の測度
- ー 2 つの可能なビン値のうちのどちらが現在のビンに関して劣勢確率または優勢確率を持

50

つビン値を表すかの推定を指定する識別子

－関連付けられたソースシンボルのカテゴリ

－関連付けられたソースシンボルの重要性の測度

－関連付けられたシンボルの位置の測度（例えば時間的、空間的、または量的なデータ集合における）

－ビンまたは関連付けられたソースシンボルについてチャネル符号保護を指定する識別子

－ビンまたは関連付けられたソースシンボルについて暗号化スキームを指定する識別子

－関連付けられたシンボルについて等級を指定する識別子

－関連付けられたソースシンボルについてのビン列内のビン番号

【0054】

一実施形態においては、パラメータ割当部4は、各ビン3、5を、現在のビンに関する2つの可能なビン値のうちの1つについての確率推定の測度に関連付ける。パラメータ割当部4は、各ビン3、5を、現在のビンに関する劣勢確率または優勢確率を持つビン値についての確率推定の測度と、2つの可能なビン値のうちのどちらが現在のビンに関して劣勢確率または優勢確率を持つビン値を表すかの推定を指定する識別子と、に関連付ける。劣勢確率または優勢確率を持つビン値についての確率と、2つの可能なビン値のうちのどちらが劣勢確率または優勢確率を持つビン値を表すかを指定する識別子とは、2つの可能なビン値のうちの1つの確率についての等しい測度であることに留意されたい。

【0055】

更なる実施形態において、パラメータ割当部4は各ビン3、5を、現在のビンに関する2つの可能なビン値のうちの1つについての確率推定の測度と、1つ以上の更なるパラメータ（上記に挙げたパラメータの1つ以上であり得る）と、に関連付ける。本発明の更なる好ましい実施形態において、パラメータ割当部4は各ビン3、5を、現在のビンに関する劣勢確率または優勢確率を持つビン値についての確率推定の測度と、2つの可能なビン値のうちのどちらが現在のビンに関して劣勢確率または優勢確率を持つビン値を表すかの推定を指定する識別子と、1つ以上の更なるパラメータ（上記に挙げたパラメータの1つ以上であり得る）と、に関連付ける。

【0056】

一実施形態において、パラメータ割当部4は、既に符号化済みのシンボルからなる1つ以上の集合に基づいて、上述した確率測度（現在のビンに関する2つの可能なビン値のうちの1つについての確率推定の測度、現在のビンに関する劣勢確率または優勢確率を持つビン値についての確率推定の測度、及び2つの可能なビン値のうちのどちらが現在のビンに関して劣勢確率または優勢確率を持つビン値を表すかの推定を指定する識別子）の1つ以上を決定する。確率の測度を決定するために使用される符号化済みのシンボルは、同じシンボルカテゴリの既に符号化済みの1つ以上のシンボル、（現在のソースシンボルに関連付けられたデータ集合に対して）空間的および／または時間的に隣接した位置にある（例えばサンプルのブロックやグループなどの）データ集合に対応する同じシンボルカテゴリの1つ以上の既に符号化済みのシンボル、又は、（現在のソースシンボルに関連付けられたデータ集合に対して）空間的および／または時間的に同一および／または隣接した位置にあるデータ集合に対応する異なるシンボルカテゴリの1つ以上の既に符号化済みのシンボル、を含むことができる。

【0057】

パラメータ割当部4の出力であるパラメータの関連付けられた集合を有する各ビン5は、ビンバッファ選択部6に送られる。ビンバッファ選択部6は、入力されたビン値とこれに関連付けられたパラメータ5に基づいて、入力されたビン5の値を潜在的に変更し、出力ビン7（潜在的に変更済みの値を有する）を2つ以上のビンバッファ8のうちの1つに送る。出力ビン7が送られるビンバッファ8は、入力ビン5の値および／または関連付けられたパラメータ5の値に基づいて、決定される。

【0058】

本発明の一実施形態において、ビンバッファ選択部6はビンの値を変更しない。すなわち

10

20

30

40

50

、出力ピン7は入力ピン5と常に同一の値を有する。

【0059】

更なる実施形態においては、ピンバッファ選択部6は、入力ピン値5と、現在のピンに関する2つの可能なピン値のうちの1つについての確率推定のための関連付けられた測度と、に基づいて、出力ピン値7を決定する。一実施形態においては、出力ピン値7は、現在のピンに関する2つの可能なピン値のうちの1つについての確率の測度が、特定のしきい値未満（又はしきい値以下）の場合には、入力ピン値5と同等に設定され、他方、現在のピンに関する2つの可能なピン値のうちの1つについての確率の測度が特定のしきい値以上の（又はしきい値より大きい）場合には、出力ピン値7は変更される（即ち入力ピン値の逆に設定される）。更なる実施形態においては、出力ピン値7は、現在のピンに関する2つの可能なピン値のうちの1つについての確率の測度が特定のしきい値より大きい（又はしきい値以上の）場合には、入力ピン値5と等しい値を設定され、他方、現在のピンに関する2つの可能なピン値のうちの1つについての確率の測度が特定のしきい値以下（又はしきい値未満）の場合には、出力ピン値7は変更される（即ち入力ピン値の逆に設定される）。一実施形態において、しきい値は、両方の可能なピン値の推定される確率について、値0.5に一致する。

10

【0060】

ピンバッファ選択部6は、入力ピン値5と、2つの可能なピン値のうちのどちらが現在のピンに関して劣勢確率または優勢確率を持つピン値を表すかの推定を指定する関連付けられた識別子と、に基づいて、出力ピン値7を決定してもよい。一実施形態においては、識別子が2つの可能なピン値のうちの第1のピン値が現在のピンに関して劣勢確率（又は優勢確率）を持つピン値を表すことを指定する場合、出力ピン値7は入力ピン値5と等しい値に設定され、他方、識別子が2つの可能なピン値のうちの第2のピン値が現在のピンに関して劣勢確率（又は優勢確率）を持つピン値を表すことを指定する場合、出力ピン値7は変更される（即ち、入力ピン値の逆に設定される）。

20

【0061】

ピンバッファ選択部6は、現在のピンに関する2つの可能なピン値のうちの1つについての確率推定のために関連付けられた測度に基づいて、出力ピン7が送られるピンバッファ8を決定してもよい。一実施形態において、2つの可能なピン値のうちの1つについての確率推定の測度としての可能な値の集合は有限であり、ピンバッファ選択部6は、2つの可能なピン値のうちの1つに関する確率推定のための可能な各値に対し、正に1つのピンバッファ8を関連付けるテーブルを含み、このテーブルでは、2つの可能なピン値の1つについての確率推定の測度の異なる値が、同じピンバッファ8に対して関連付けられることができる。更なる好ましい実施形態においては、2つの可能なピン値のうちの1つについての確率推定の測度として可能な値の範囲は、幾つかの区間に区分され、ピンバッファ選択部6は、2つの可能なピン値のうちの1つについての確率推定の現在の測度のための区間インデックスを決定し、さらに、ピンバッファ選択部6は、正に1つのピンバッファ8をこの区間インデックスのための可能な各値に関連付けるテーブルを含み、ここでは、区間インデックスのための異なる値を同じピンバッファ8に対して関連付けることができる。一実施形態においては、2つの可能なピン値のうちの一方についての確率推定の逆の測度を有する入力ピン5（逆測度は確率推定 P および $1-P$ を表す測度である）が、同じピンバッファ8に供給される。更なる実施形態においては、現在のピンに関する2つの可能なピン値のうちの一方についての確率推定の測度を特定のピンバッファに関連付けることは、例えば作成された部分的ビットストリームが同様のビットレートを持つことを確実にするために、経時的に適応される。

30

40

【0062】

ピンバッファ選択部6は、現在のピンに関して劣勢確率または優勢確率を持つピン値の確率推定のために関連付けられた測度に基づいて、出力ピン7を送るべきピンバッファ8を決定する。本発明の好ましい実施形態においては、劣勢確率または優勢確率を持つピン値の確率推定のための測度としての可能な値の集合は有限であり、ピンバッファ選択部6は、

50

正に1つのピンバッファ8を、劣勢確率または優勢確率を持つピン値のための確率推定の可能な各値に関連付けるテーブルを含み、このテーブルでは、劣勢確率または優勢確率を持つピン値についての確率推定の測度のための異なる値を、同じピンバッファ8に対して関連付けることができる。更なる実施形態においては、劣勢確率または優勢確率を持つピン値についての確率推定の測度として可能な値の範囲は幾つかの区間に区分され、ピンバッファ選択部6は、劣勢確率または優勢確率を持つピン値についての確率推定のための現在の測度に関する区間インデックスを決定し、さらに、ピンバッファ選択部6は、1つのピンバッファ8をこの区間インデックスのための可能な各値に関連付けるテーブルを含み、このテーブルでは、区間インデックスのための異なる値を同じピンバッファ8に対して関連付けることができる。更なる実施形態においては、現在のピンに関して劣勢確率または優勢確率を持つピン値についての確率推定の測度を特定のピンバッファに関連付けることは、例えば作成された部分的ビットストリームが同様のビットレートを持つことを確実にするために、経時的に適応される。

10

【0063】

2つ以上のピンバッファ8の各々は、1つのピン符号器10に接続され、各ピン符号器は1つのピンバッファ8だけに連結されている。各ピン符号器10は、関連付けられたピンバッファ8からピンを読み出し、ピン列9を、ビットの列を表す符号語11に変換する。ピンバッファ8は先入れ先出しバッファを表す。即ち、(順番において)後でピンバッファ8に送られるピンは、(順番において)先にピンバッファ8に送られたピンよりも前に符号化されることはない。特定のピン符号器10の出力である符号語11は、特定の部分的ビットストリーム12に書き込まれる。全体の符号化アルゴリズムは、ソースシンボル1を2つ以上の部分的なビットストリーム12へ変換し、この場合、部分的ビットストリーム12の数はピンバッファ及びピン符号器の数と等しい。好ましい実施形態においては、ピン符号器10は、可変数のピン9を、可変数のビットからなる符号語11に変換する。上記および下記に概説されるPIPE符号化の実施形態の1つの利点は、ピンの符号化は(例えば、確率測度の異なるグループについて)並列的に実行することができる点にあり、これにより、幾つかの構成例における処理時間が短縮される。

20

【0064】

PIPE符号化の他の利点は、ピン符号器10によって行われるピン符号化が、パラメータ5の様々な集合のために特別に設計できる点にある。特に、ピン符号化と符号化は、推定された確率の異なるグループ毎に、(符号化効率および/または複雑性に関して)最適化することができる。一方で、この利点は、同様の符号化効率を有する算術符号化アルゴリズムと比較して、符号化/復号化の複雑性の低減を可能にする。他方では、同様の符号化/復号化の複雑性を有するVLC符号化アルゴリズムと比較して、符号化効率の向上を可能にする。一実施形態においては、ピン符号器10は、現在のピンに関する2つの可能なピン値5のうちの1つについての確率推定の測度の異なるグループ毎に、異なる符号化アルゴリズム(即ち、ピン列の符号語へのマッピング)を実施する。更なる実施形態においては、ピン符号器10は、現在のピンに関して劣勢確率または優勢確率を持つピン値についての確率推定の測度の異なるグループ毎に、異なる符号化アルゴリズムを実施する。更なる実施形態において、ピン符号器10は、異なるチャンネル保護コード毎に異なる符号化アルゴリズムを実施する。更なる実施形態において、ピン符号器10は、異なる暗号化スキーム毎に異なる符号化アルゴリズムを実施する。更なる実施形態において、ピン符号器10は、チャンネル保護コードと現在のピンに関する2つの可能なピン値5のうちの1つについての確率推定の測度のグループとの様々な組み合わせ毎に、異なる符号化アルゴリズムを実施する。更なる実施形態において、ピン符号器10は、チャンネル保護コードと現在のピンに関して劣勢確率または優勢確率を持つピン値5についての確率推定の測度のグループとの様々な組み合わせ毎に、異なる符号化アルゴリズムを実施する。更なる実施形態において、ピン符号器10は、暗号化スキームと現在のピンに関する2つの可能なピン値5のうちの1つについての確率推定の測度のグループとの様々な組み合わせ毎に、異なる符号化アルゴリズムを実施する。更なる好ましい実施形態において、ピン符号器10は

30

40

50

、暗号化スキームと現在のビンに関して劣勢確率または優勢確率を持つビン値5についての確率推定の測度のグループとの様々な組み合わせ毎に、異なる符号化アルゴリズムを実施する。

【0065】

ビン符号器10(又は複数のビン符号器のうちの1つ以上)は、バイナリー算術符号化エンジンを表してもよい。更なる実施形態においては、1つ以上のビン符号器がバイナリー算術符号化エンジンを表し、ここで、所与のビンバッファの代表的LPS/LPB確率 p_{LPS} から対応する符号区間幅 R_{LPS} へのマッピング、—即ち、例えば符号区間の下限を識別する、現在の区間幅 R と現在の区間オフセット L とによって定義されるバイナリー算術符号化エンジンの内部状態の区間のサブ分割—は、テーブルルックアップを使用することによって実現される。更なる好ましい実施形態においては、所与のビンバッファに関連付けられる各テーブルベースのバイナリー算術符号化エンジンについて、 K 個の代表的区間幅の値 $\{Q_0, \dots, Q_{K-1}\}$ が R_{LPS} を表すために使用され、その際、 K の選択および代表的区間幅の値 $\{Q_0, \dots, Q_{K-1}\}$ はビンバッファに依存している。 $K > 1$ を選択した場合には、ビンの算術符号化は、現在の区間幅 R を $\{0, \dots, K-1\}$ 内の値を有する量子化インデックス q にマッピングするサブステップと、 q をインデックスとして使用することでルックアップテーブルから対応する部分的区間幅の値 Q_q にアクセスすることによって区間のサブ分割を実行するサブステップと、を含む。 $K = 1$ を選択する場合には、即ち、代表的区間幅の値 Q_0 だけが与えられている場合には、この値 Q_0 は、対応するビンバッファを単一の再正規化サイクル内に入れつつ、多数のMPS/MPB値を復号化できるように、2の冪として選択される。結果として得られる各算術符号化エンジンの符号語は、個別に転送され、パケット化され、若しくは記憶されてもよく、又は、以下に説明するように伝送または記憶するためにインターリーブされてもよい。

【0066】

つまり、バイナリー算術符号化エンジン10は、そのビンバッファ8内で複数のビンを符号化するために、以下のステップを実行することができる。

1. ビンバッファから val_{LPS} 、ビンを受信するステップ：(ここで考慮対象となるそれぞれのバイナリー算術符号化エンジン10は、既に「ビン」を受け取るべく選択されていたことを想起されたい。なぜなら、 $p_state[b_in]$ などの確率分布推定が、そのバイナリー算術符号化エンジン10に関連付けられていた(換言すれば、「ビン」がそれぞれのバイナリー算術符号化エンジン10に関連付けられていた)からである。)

2. R を量子化するステップ：

$q_index = Q_{tab}[R \gg q]$ (または他の形態の量子化)

3. R_{LPS} および R を決定するステップ：

$R_{LPS} = R_{tab}[q_index]$ (ここでは、 p_state について述べている訳ではない点に留意されたい。なぜなら、 p_state は、考慮対象となるバイナリー算術符号化エンジン10に対して固定されている、即ち、 $p_state[encoder]$ であるからであり、さらに、 R_{tab} はその中に、 $p[p_state[encoder]] \cdot Q[q_index]$ について事前に計算された値を記憶しているからである。)

$R = R - R_{LPS}$ [即ち、「ビン」がMPSであるかのように、 R が予め事前更新される。]

4. 新たな部分的区間を計算するステップ：

もし(ビン = $1 - val_{MPS}$)である場合は、

$L \leftarrow L + R$

$R \leftarrow R_{LPS}$

5. L と R との再正規化、ビットを書き込むステップ：

ここで、

q_index は Q_{tab} から読み出された量子化値のインデックスを記述し、

p_state は（バイナリー算術符号化エンジン10について固定された）現在の状態を記述し、

R_{LPS} はLPSに対応する区間幅を記述し、

$valMPS$ はMPSに対応するビットの値を記述する。

【0067】

従って、バイナリー算術復号化エンジン22は、ビンバッファ20内へと出力されるビンを復号化する際、以下のステップを実行することができる。

1. ビンに対する要求を受信するステップ：（ここで考慮対象となるそれぞれのバイナリー算術復号化エンジン22は、既に「ビン」を復号すべく選択されていたことを想起されたい。なぜなら、 $p_state[b_in]$ などの確率分布推定が、そのバイナリー算術復号化エンジン22に関連付けられていた（換言すれば、「ビン」がそれぞれのバイナリー算術復号化エンジン22に関連付けられていた）からである。）

10

2. R を量子化するステップ：

$q_index = Qtab[R > q]$ （または他の形態の量子化）

3. R_{LPS} と R を決定するステップ：

$R_{LPS} = Rtab[q_index]$ （ここでは、 p_state について述べている訳ではない点に留意されたい。なぜなら、 p_state は、考慮対象となるバイナリー算術復号化エンジン22に対して固定されている、即ち、 $p_state[encoder]$ であるからであり、さらに、 $Rtab$ はその中に、 $p[p_state[encoder]] \cdot Q[q_index]$ について事前に計算された値を記憶しているからである。）

20

$R = R - R_{LPS}$ （即ち、「ビン」がMPSであるかのように、 R が予め事前更新される）

4. 部分的区間の位置に応じてビンを決定するステップ：

もし（ $V \geq R$ ）である場合、

ビン $\leftarrow 1 - valMPS$ （ビンはLPSとして復号化される：ビンバッファ選択部18は、このビン情報と $valMPS$ とを用いて実際のビン値を取得するであろう）

$V \leftarrow V - R$

$R \leftarrow R_{LPS}$

その他の場合、

ビン $\leftarrow valMPS$ （ビンはMPSとして復号化される：ビンバッファ選択部18は、このビン情報と $valMPS$ とを用いて実際のビン値を取得するであろう）

30

5. R の再正規化、1ビットを読み出して V を更新するステップ

ここで、

q_index は $Qtab$ から読み出された量子化値のインデックスを記述し、

p_state は（バイナリー算術復号化エンジン22について固定された）現在の状態を記述し、

R_{LPS} はLPSに対応する区間幅を記述し、

$valMPS$ はMPSに対応するビットの値を記述し、

V は現在の部分的区間の内部からの値を記述する。

【0068】

40

複数のビン符号器10（又は1つ以上のビン符号器）は、入力ビン列9を符号語10に直接的にマップするエントロピー符号器を表している。このようなマッピングは効率的に実施することができ、複雑な算術符号化エンジンを必要としない。（復号器で行われているような）ビン列への符号語の逆マッピングは、入力列の完全な復号化を保証するために一意的である必要があるが、ビン列9の符号語10へのマッピングは、必ずしも一意的である必要はない。即ち、1つの特定のビン列が2つ以上の符号語の列へとマップされることも可能である。一実施形態においては、入力ビン列9の符号語10へのマッピングは全単射である。更なる好ましい実施形態において、複数のビン符号器10—又は1つ以上のビン符号器—は、可変長の入力ビン列9を可変長の符号語10に直接的にマップするエントロピー符号器を表している。一実施形態において、出力された符号語は、一般的なハフマン

50

符号 (general Huffman codes) 又は基準ハフマン符号 (canonical Huffman codes) などの冗長さのない符号を表している。

【 0 0 6 9 】

冗長さのない符号にビン列を全単射的にマッピングするための 2 つの例を表 3 に示す。更なる実施形態において、出力符号語は、エラー検出とエラー回復のために適した冗長さのある符号を表している。更なる実施形態において、出力符号語は、ソースシンボルを暗号化するために適した暗号化コードを表す。

【 0 0 7 0 】

【表 3】

【表3】ビン列と符号語との間のマッピングの例

ビン列 (ビン順序は左から右)	符号語 (ビット順序は左から右)
0000 0000	1
0000 0001	0000
0000 001	0001
0000 01	0010
0000 1	0011
0001	0100
001	0101
01	0110
1	0111

ビン列 (ビン順序は左から右)	符号語 (ビット順序は左から右)
000	10
01	11
001	010
11	011
1000 0	0001
1001	0010
1010	0011
1000 1	0000 0
1011	0000 1

【 0 0 7 1 】

更なる実施形態において、複数のビン符号器 1 0 – 又は 1 つ以上のビン符号器 – は、可変長の入力ビン列 9 を固定長の符号語 1 0 に直接的にマップするエントロピー符号器を表している。更なる実施形態において、複数のビン符号器 1 0 (又は 1 つ以上のビン符号器) は、固定長の入力ビン列 9 を可変長の符号語 1 0 に直接的にマップするエントロピー符号器を表している。

【 0 0 7 2 】

一実施形態に係る P I P E 復号器を図 4 に示す。この復号器は、基本的には図 3 の符号器

の逆の操作を実行し、これによってソースシンボル 27 の（以前に符号化済みの）列が、2 つ以上の部分的ビットストリーム 24 の集合から復号化される。復号器は 2 つの異なる処理フローを含む。即ち、符号器のデータフローを複製するデータ要求のためのフローと、符号器データフローの逆を表わすデータフローとを含む。図 4 において、破線の矢印はデータ要求フローを表し、他方、実線の矢印はデータフローを表す。復号器の構成ブロックは、基本的に符号器の構成ブロックを複製しているが、逆の操作を実行する。

【0073】

ソースシンボルの復号化は、バイナライザ 14 へ送られる、新たな復号化済みソースシンボルに対する要求 13 によってトリガーされる。一実施形態において、新たな復号化済みソースシンボルに対する各要求 13 は、1 つ以上のカテゴリからなる集合の 1 つのカテゴリに関連付けられている。このソースシンボルに対する要求に関連付けられているカテゴリは、対応するソースシンボルに符号化の過程で関連付けられていたカテゴリと同一である。

10

【0074】

バイナライザ 14 は、ソースシンボルに対する要求 13 をパラメータ割当部 16 へ送られるピンに対する 1 つ以上の要求へとマップする。バイナライザ 14 によりパラメータ割当部 16 へ送られる、ピンに対する要求への最終応答として、バイナライザ 14 は、ピンバッファ選択部 18 から復号化済みのピン 26 を受信する。バイナライザ 14 は、復号化済みのピン 26 の受信された列と、要求されたソースシンボルのための特定の二値化スキームのピン列と、を比較して、もし受信された復号化済みのピン 26 の列がソースシンボルの 2 値化と合致した場合には、バイナライザは、そのピンバッファを空にして、さらに、新たな復号化済みシンボルに対する要求への最終応答としてその復号化済みのソースシンボルを出力する。もし既に受信された復号化済みのピンの列が、要求されたソースシンボルのための 2 値化スキームのピン列のいずれとも合致しない場合には、バイナライザは、復号化済みのピンの列が要求されたソースシンボルのための二値化スキームのピン列の 1 つに合致するまで、ピンに対する他の要求をパラメータ割当部へ送信する。ソースシンボルに対する各要求ごとに、復号器は、対応するソースシンボルを符号化するために使用されたものと同じ二値化スキームを使用する。二値化スキームは、異なるソースシンボルカテゴリごとに異なり得る。特定のソースシンボルカテゴリのための二値化スキームは、可能なソースシンボル値の集合および／または特定のカテゴリについてのソースシンボルの他の特性に依存し得る。

20

30

【0075】

パラメータ割当部は、ピンに対する各要求に対して 1 つ以上のパラメータの集合を割り当て、さらにそのピンに対する要求とそれに関連付けられたパラメータの集合とを、ピンバッファ選択部へ送信する。要求されたピンに対してパラメータ割当部により割り当てられたパラメータの集合は、符号化の過程において対応するピンへ割当てられたものと同一である。このパラメータの集合は、符号器の説明の中で上述したパラメータの 1 つ以上から構成されてもよい。

【0076】

パラメータ割当部 16 は、ピンに対する各要求、現在要求されているピンに関する 2 つの可能なピン値のうちの 1 つについての確率推定の測度と、に関連付けてもよい。特に、パラメータ割当部 16 は、ピンに対する各要求を、現在要求されているピンに関する劣勢確率または優勢確率を持つピン値についての確率推定の測度と、2 つの可能なピン値のうちどちらが現在要求されているピンに関する劣勢確率または優勢確率を持つピン値を表すかの推定を指定する識別子と、に関連付けてもよい。

40

【0077】

更なる実施形態において、パラメータ割当部 16 は、ピンに対する各要求 15、17 を、現在要求されているピンに関する 2 つの可能なピン値のうちの 1 つについての確率推定の測度と、1 つ以上の更なるパラメータと、に関連付ける。更なる好ましい実施形態において、パラメータ割当部 16 は、ピンに対する各要求 15、17 を、現在要求されているピ

50

ンに関する劣勢確率または優勢確率を持つビン値についての確率推定の測度と、2つの可能なビン値のうちどちらが現在要求されているビンに関する劣勢確率または優勢確率を持つビン値を表すかの推定を指定する識別子と、1つ以上の更なるパラメータ（上記に挙げたパラメータの1つ以上であり得る）と、に関連付ける。

【0078】

一実施形態において、パラメータ割当部16は、1つ以上の既に復号化済みのシンボルの集合に基づいて、上述した確率の測度（現在要求されているビンに関する2つの可能なビン値のうちの1つについての確率推定の測度と、現在要求されているビンに関する劣勢確率または優勢確率を持つビン値についての確率推定の測度と、2つの可能なビン値のうちどちらが現在要求されているビンに関する劣勢確率または優勢確率を持つビン値を表すかの推定を指定する識別子）のうちの1つ以上を決定する。ビンに対する特定の要求のための確率測度の決定は、対応するビンのための符号器における処理を複製するものである。確率測度を決定するために使用される復号化済みのシンボルは、同じシンボルカテゴリーの既に復号化済みの1つ以上のシンボル、（ソースシンボルに対する現在の要求に関連付けられたデータ集合に対して）空間的および／または時間的に隣接した位置にある（例えばサンプルのブロックやグループなどの）データ集合に対応する、同じシンボルカテゴリーの1つ以上の既に符号化済みのシンボル、又は、（ソースシンボルに対する現在の要求に関連付けられたデータ集合に対して）空間的および／または時間的に同一および／または隣接した位置にあるデータ集合に対応する、異なるシンボルカテゴリーの1つ以上の既に符号化済みのシンボルを含むことができる。

【0079】

パラメータ割当部16の出力である、関連付けられたパラメータの集合を有するビンに対する各要求17は、ビンバッファ選択部18内へ送られる。関連付けられたパラメータ17の集合に基づいて、ビンバッファ選択部18は、2つ以上のビンバッファ20のうちの1つに向かってビンに対する要求19を送信し、選択されたビンバッファ20から復号化済みのビン25を受信する。復号化済みの入力ビン25は、潜在的に変更され、復号化済みの出力ビン26（潜在的に変更済みの値を有する）は、関連付けられたパラメータの集合を有するビンに対する要求17への最終応答として、バイナライザ14へ送信される。

【0080】

ビンに対する要求が送信されるべきビンバッファ20が選択される方法は、符号器側においてビンバッファ選択部の出力ビンが送信されるべきビンバッファが選択された方法と同様である。

【0081】

一実施形態において、ビンバッファ選択部18は、ビンに対する要求19が送信されるべきビンバッファ20を、現在要求されているビンに関する2つの可能なビン値のうちの1つについての確率推定の関連付けられた測度に基づいて決定する。一実施形態において、2つの可能なビン値のうちの1つについての確率推定の測度に関する可能な値の集合は有限であり、ビンバッファ選択部18は、正に1つのビンバッファ20を、2つの可能なビン値のうちの1つについての確率推定の各可能な値と関連付ける、テーブルを含み、ここで、2つの可能なビン値のうちの1つについての確率推定の測度に関する異なる値を、同一のビンバッファ20と関連付けることができる。更なる実施形態において、2つの可能なビン値のうちの1つについての確率推定の測度に関する可能な値の範囲は、複数の区間に区分され、ビンバッファ選択部18は、2つの可能なビン値のうちの1つについての確率推定の現在の測度に係る区間インデックスを決定し、さらに、ビンバッファ選択部18は、正に1つのビンバッファ20を、区間インデックスに関する各可能な値と関連付けるテーブルを含み、ここで、区間インデックスに関する異なる値を、同じビンバッファ20と関連付けることができる。2つの可能なビン値のうちの1つについての確率推定に関し、反対の測度を有するビンに対する要求17（反対の測度とは、確率推定 P と $1-P$ とを表す測度同士である）は、同じビンバッファ20へ転送されてもよい。さらに、現在のビン要求に関する2つの可能なビン値のうちの1つについての確率推定の測度と特定のビン

バッファとの関連付けが、経時的に適応されてもよい。

【0082】

ビンバッファ選択部18は、ビンに対する要求19が送信されるべきビンバッファ20を、現在要求されているビンに関する劣勢確率または優勢確率を持つビン値についての確率推定の関連付けられた測度に基づいて決定してもよい。一実施形態においては、劣勢確率または優勢確率を持つビン値についての確率推定の測度に関する可能な値の集合は有限であり、ビンバッファ選択部18は、正に1つのビンバッファ20を劣勢確率または優勢確率を持つビン値についての確率推定の各可能な値と関連付ける、テーブルを含み、ここで、劣勢確率または優勢確率を持つビン値についての確率推定の測度に関する異なる値を同じビンバッファ20と関連付けることができる。一実施形態においては、劣勢確率または優勢確率を持つビン値についての確率推定の測度に関する可能な値の範囲が複数の区間へと区分され、ビンバッファ選択部18は、劣勢確率または優勢確率を持つビン値についての確率推定に関する現在の測度に係る区間インデックスを決定し、更に、ビンバッファ選択部18は、正に1つのビンバッファ20を区間インデックスに関する各可能な値と関連付ける、テーブルを含み、ここで、区間インデックスに関する異なる値を、同じビンバッファ20と関連付けることができる。更なる実施形態においては、現在要求されているビンに関して劣勢確率または優勢確率を持つビン値についての確率推定の測度と特定のビンバッファとの関連付けが、経時的に適応される。

10

【0083】

ビンバッファ選択部18は選択されたビンバッファ20から復号化済みのビン25を受信した後で、入力ビン25を潜在的に変更し、潜在的に変更済みの値を有する出力ビン26をバイナライザ14へと送信する。ビンバッファ選択部18の入力／出力ビンマッピングは、符号器側におけるビンバッファ選択部の入力／出力ビンマッピングの逆である。

20

【0084】

ビンバッファ選択部18は、ビンの値を変更しないように、即ち、出力ビン26が入力ビン25と同じ値を常に有するように構成されてもよい。

【0085】

ビンバッファ選択部18は、入力ビン値25と、ビンに対する要求17と関連付けられた現在要求されているビンに関する2つの可能なビン値のうちの1つについての確率推定の測度に基づいて、出力ビン値26を決定してもよい。一実施形態においては、現在のビン要求に関する2つの可能なビン値のうちの1つについての確率の測度が特定のしきい値未満（又はしきい値以下）である場合には、出力ビン値26は入力ビン値25と等しく設定されるが、他方、現在のビン要求に関する2つの可能なビン値のうちの1つについての確率の測度が特定のしきい値以上（又はしきい値より大きい）である場合には、出力ビン値26は変更される（即ち、入力ビン値の逆に設定される）。更なる実施形態において、出力ビン値26は、現在のビン要求に関する2つの可能なビン値のうちの1つについての確率の測度が特定のしきい値より大きい（又はしきい値以上の）場合には、入力ビン値25と等しく設定されるが、他方、現在のビン要求に関する2つの可能なビン値のうちの1つについての確率の測度が特定のしきい値以下（またはしきい値未満）である場合には、出力ビン値26は変更される（即ち、入力ビン値の逆に設定される）。好ましい一実施形態においては、しきい値は、両方の可能なビン値についての推定された確率に関し、0.5の値に一致している。

30

40

【0086】

更なる好ましい実施形態において、ビンバッファ選択部18は、入力ビン値25と、ビンに対する要求17と関連付けられており、2つの可能なビン値のうちのどちらが現在のビン要求に関して劣勢確率または優勢確率を持つビン値を表すかの推定を指定する識別子と、に基づいて、出力ビン値26を決定する。好ましい一実施形態においては、もし識別子が、2つの可能なビン値のうちの第1のビン値が現在のビン要求について劣勢確率（又は優勢確率）を持つビン値を表していると指定する場合には、出力ビン値26は入力ビン値25と等しく設定され、他方、もし識別子が2つの可能なビン値のうちの第2のビン値が

50

現在のピン要求について劣勢確率（又は優勢確率）を持つピン値を表していると指定する場合には、出力ピン値 26 は変更される（即ち、入力ピン値の逆に設定される）。

【0087】

上述したように、ピンバッファ選択部は、2つ以上のピンバッファ 20 のうちの 1 つへ、ピンに対する要求 19 を送信する。ピンバッファ 20 は、接続されたピン復号器 22 から復号化済みのピンの列を受け取る、先入れ先出しバッファを表す。ピンバッファ選択部 18 からピンバッファ 20 へ送られる、ピンに対する要求 19 への応答として、ピンバッファ 20 はピンバッファ 20 に最初に送られたそのコンテンツのピンを取り除き、それをピンバッファ選択部 18 へ送信する。先にピンバッファ 20 に送信されるピンは先に取り除かれ、ピンバッファ選択部 18 へ送信される。

10

【0088】

2つ以上のピンバッファ 20 の各々は、正に 1 つのピン復号器 22 に接続されており、各ピン復号器は 1 つだけのピンバッファ 20 に接続されている。各ピン復号器 22 は、分離された部分的ビットストリーム 24 からビット列を表す符号語 23 を読み出す。ピン復号器は、符号語 23 を接続されたピンバッファ 20 に送信されるピンの列 21 に変換する。全体的な復号化アルゴリズムは、2つ以上の部分的なビットストリーム 24 を複数の復号化済みソースシンボルへ変換し、ここで、部分的ビットストリームの数はピンバッファとピン復号器の数と等しく、ソースシンボルの復号化は新しいソースシンボルに対する要求によってトリガーされる。ある実施形態においては、ピン復号器 22 は、可変数のビットから成る符号語 23 を可変数のピンの列 21 へ変換する。上述した P I P E の実施形態の 1 つの利点は、2つ以上の部分的ビットストリームからのピンの復号化が（例えば確率測度の異なるグループについて）並列的に実行されることができ、これによって複数の構成にかかる処理時間を短縮できるという点にある。

20

【0089】

上述した P I P E 復号化の実施形態の他の利点は、ピン復号器 22 によって行われるピン復号化が、パラメータ 17 の異なる集合に対して特別に設計することができる点にある。特に、ピンの符号化と復号化は推定される確率の異なるグループ毎に（符号化効率及び／又は複雑性に関して）最適化することができる。一方で、このような最適化は、同様の符号化効率を有する算術符号化アルゴリズムと比較して、符号化／復号化の複雑性を低減できる。他方で、このような最適化は、同様の符号化／復号化の複雑性を有する V L C 符号化アルゴリズムと比較して、符号化効率の向上を可能にする。本発明のある実施形態において、ピン復号器 22 は、現在のピン要求に関する 2 つの可能なピン値 17 のうちの 1 つについての確率推定の測度の異なるグループ毎に、異なる復号化アルゴリズム（即ち、ピン列から符号語へのマッピング）を用いる。更なる実施形態において、ピン復号器 22 は、現在要求されているピンに関する劣勢確率又は優勢確率を持つピン値についての確率推定の測度の異なるグループ毎に、異なる復号化アルゴリズムを用いる。更なる実施形態において、ピン復号器 22 は、異なるチャンネル保護コード毎に異なる復号化アルゴリズムを用いる。更なる実施形態において、ピン復号器 22 は、異なる暗号化スキーム毎に異なる復号化アルゴリズムを用いる。更なる実施形態において、ピン復号器 22 は、チャンネル保護コードと、現在要求されているピンに関する 2 つの可能なピン値 17 のうちの 1 つについての確率推定の測度のグループと、の様々な組み合わせ毎に、異なる復号化アルゴリズムを用いる。更なる実施形態において、ピン復号器 22 は、暗号化スキームと、現在要求されているピンに関する劣勢確率又は優勢確率を持つピン値 17 についての確率推定の測度のグループと、の様々な組み合わせ毎に、異なる復号化アルゴリズムを実施する。更なる実施形態において、ピン復号器 22 は、暗号化スキームと、現在要求されているピンに関する劣勢確率又は優勢確率を持つピン値 17 についての確率推定の測度のグループと、の様々な組み合わせ毎に、異なる復号化アルゴリズムを用いる。

30

40

50

【0090】

ピン復号器22は、符号器側における対応するピン符号器の逆マッピングを行う。

【0091】

一実施形態において、複数のピン復号器22(又は1つ以上のピン復号器)は、バイナリー算術復号化エンジンを表している。

【0092】

更なる好ましい実施形態において、複数のピン復号器22(又は1つ以上のピン復号器)は、符号語23をピンの列21に直接的にマップするエントロピー復号器を表している。このようなマッピングは効率的に実施することができ、複雑な算術符号化エンジンを必要としない。ピン列への符号語のマッピングは一意である必要がある。ある実施形態において、符号語23のピンの列21へのマッピングは全単射である。更なる実施形態において、複数のピン復号器22(又は1つ以上のピン復号器)は、可変長の符号語23を可変長のピン列21へ直接的にマップするエントロピー復号器を表している。一実施形態において、入力された符号語は、一般的なハフマン符号又は基準ハフマン符号などの冗長度のない符号を表している。冗長度のない符号をピン列に全単射マッピングするための2つの例を表3に示す。更なる実施形態において、入力された符号語は、エラー検出とエラー回復のために適切な冗長度のある符号を表している。更なる実施形態において、入力された符号語は暗号化符号を表している。

10

【0093】

複数のピン復号器22(又は1つ以上のピン復号器)は、固定長の符号語23を可変長のピン列21上へと直接的にマップするエントロピー復号器を表している。代替的に、複数のピン復号器22(又は1つ以上のピン復号器)は、可変長の符号語23を固定長のピン列21へと直接的にマップするエントロピー復号器を表している。

20

【0094】

従って、図3及び図4は、ソースシンボル1の列を符号化するためのPIPE符号器と、それを復元するためのPIPE復号器とについての実施の形態を示す。つまり、図3のPIPE符号器は、図1aのPIPE符号器104として使用されてもよく、このとき、バイナライザ2はシンボライザ122として作用し、パラメータ割当部4は割当部114として作用し、ピンバッファ選択部6は選択部120として作用し、直列に接続されたピンバッファ8とピン符号器10とのペアは、エントロピー符号器116のそれぞれ1つとして作用し、そのエントロピー符号器のそれぞれが図1aに示すビットストリーム118に対応するビットストリーム12を出力する。図3と図1aとを比較すれば明らかなように、図1aの割当部114は代替的に、その入力が、シンボライザ122の出力側ではなく、入力側に接続されていてもよい。同様に、図4のPIPE復号器は、図2aのPIPE復号器202として使用されてもよく、このとき、部分的ビットストリーム24は図2のビットストリーム216に対応し、直列に接続されたバッファ20とピン復号器22とのペアは個別のエントロピー復号器210に対応し、ピンバッファ選択部18は選択部214として作用し、パラメータ割当部16は割当部212として作用し、バイナライザ14はデ・シンボライザ222として作用する。また、図2aと図4とを比較すれば明らかなように、デ・シンボライザ222と割当部212と選択部214との間の相互接続は、異なる状態で構成されてもよく、その結果、代替的な実施形態によれば、図2aの接続は図4における接続と対応するように修正されている。

30

40

【0095】

図3のPIPE符号器は、幾つかのパラメータ5を、アルファベットシンボル3の列の各アルファベットシンボルに対して割り当てるよう構成された、割当部4を含む。この割当ては、シンタックス要素1のカテゴリなど、アルファベットシンボル列の先行するアルファベットシンボル内に含まれた情報に基づいたものであり、現在のアルファベットシンボルが属する表現—例えば二値化など—であって、シンタックス要素1のシンタックス構造に従って現時点で予想される表現への割り当てであり、その予想とは、先行するシンタックス要素1とアルファベットシンボル3との履歴から推定可能なものである。更に、符

50

号器は、それぞれのエントロピー符号器へ送られたアルファベットシンボル3をそれぞれのビットストリーム12へと変換するように各々が構成されている、複数のエントロピー符号器10と、複数のエントロピー符号器10のうちの選択された1つに対して各アルファベットシンボル3を送るように構成されている選択部6であって、その選択がそれぞれのアルファベットシンボル3に割り当てられたパラメータ5の数に基づいている、選択部6とを含む。図4に示すP I P E復号器は、それぞれのビットストリーム23をアルファベットシンボル21へと変換するように各々が構成されている、複数のエントロピー復号器22と、アルファベットシンボル列の以前に復元済みのアルファベットシンボル（図4の26および27参照）内に含まれる情報に基づいて、復元されるべきアルファベットシンボル列の各アルファベットシンボル15に対し、幾つかのパラメータ17を割り当てるよう構成された割当部16と、復元されるべきアルファベットシンボルの列の各アルファベットシンボルを回収するよう構成されている選択部18であって、その選択がそれぞれのアルファベットシンボルに対して定義づけられているパラメータの数に依存する、選択部18と、を含む。割当部16は、各シンボルに割り当てられたパラメータの数が、それぞれのアルファベットシンボルが持つ可能性のあるアルファベットシンボル値の間の分布の確率推定のための測度を含むか又は測度そのものとなるように構成されていてもよい。復元されるべきアルファベットシンボル列は、二値アルファベットから作成され、割当部16は、確率分布の推定が、二値アルファベットの2つの可能なビン値の劣勢確率又は優勢確率を持つビン値の確率推定のための測度と、2つの可能なビン値のうちどちらが劣勢確率又は優勢確率を持つビン値を表すかの推定を指定する識別子と、から成るように構成されていてもよい。割当部16はさらに、復元されるべきアルファベットシンボル15の列の各アルファベットシンボルに対し、復元されるべきアルファベットシンボル列の以前の復元済みのアルファベットシンボル内に含まれる情報に基づいて、コンテキストを内部的に割り当てるよう構成されてもよく、このとき各コンテキストは、それぞれの確率分布の推定が関連付けられた状態であり、割当部16はさらに、各コンテキストについての確率分布推定を、それぞれのコンテキストが割り当てられた以前に復元済みのアルファベットシンボルのシンボル値に基づいて、実際のシンボル統計に適応するよう構成されてもよい。このコンテキストは、ビデオ又は画像符号化において、又は財務的なアプリケーションの場合には表において、シンタックス要素が属する位置の空間的關係又は近接性を考慮に入れてもよい。次いで、各アルファベットシンボルについての確率分布の推定のための測度は、例えば、それぞれのシンボルに割り当てられたコンテキストと関連付けられた確率分布推定を、確率分布の推定のための測度を得るために、複数の確率分布推定の代表値の1つに対して量子化することによって、それぞれのシンボルに割り当てられたコンテキストと関連付けられた確率分布推定に基づいて決定されてもよい。選択部は、複数のエントロピー符号器と複数の確率分布推定の代表値との間で、全射的(surjective)な関連付けが定義付けられるように構成することができる。即ち、各エントロピー符号器は、関連付けられた少なくとも1つの確率分布推定の代表値を持つが、しかし、2つ以上の確率分布推定の代表値が1つのエントロピー符号器に関連付けられてもよい。一実施形態によれば、この関連付けは全単射的でもよい。選択部18は、アルファベットシンボル列の以前に復元済みのアルファベットシンボルに基づく、予め決定された決定論的な方法で、量子化マッピングを、確率分布推定の範囲から複数の確率分布推定の代表値へと、経時的に変化させるよう構成されてもよい。つまり、選択部18は、量子化ステップのサイズ、即ち、個別の確率インデックスにマップされた確率分布の区間を変更することができ、その個別の確率インデックスは、個別のエントロピー復号器と全単射的に関連付けられていてもよい。次いで、複数のエントロピー復号器22は、量子化マッピングにおける変化に応答してアルファベットシンボルをビットストリームへと変換する方法を適応するよう構成されてもよい。例えば、各エントロピー復号器22は、それぞれの確率分布推定の量子化区間内において、所定の確率分布推定に対して最適化される、即ち最適な圧縮率を有することができ、かつ、エントロピー復号器22が最適化されるべく切り替わる際に、それぞれの確率分布推定の量子化区間内においてこの所定の確率分布推定の位置を適応させるために

10

20

30

40

50

、その符号語／シンボルの列のマッピングを変化させてもよい。選択部は、アルファベットシンボルが複数のエントロピー復号器から回収されるレートのばらつきが小さくなるように、量子化マッピングを変化させるよう構成されてもよい。バイナライザ 14 に関しては、シンタックス要素が既に二値である場合には放置されてもよいことに留意すべきである。さらに、復号器 22 のタイプによってはバッファ 20 は不要である。また、バッファは復号器内で統合されてもよい。

【0096】

これまで、図 1 a と図 2 に示す PIPE 符号器 104 と PIPE 復号器 202 とのより詳細な実施形態を、図 3 と図 4 を参照しながら説明してきた。図 3 と図 4 とに示す実施形態は、図 1 a と図 2 に示す装置の中で実装された場合には並列的なビットストリーム出力をもたらし、その出力の中では V L C 及び PIPE の部分的ビットストリームが並列的に運ばれる。以下に、PIPE の部分的ビットストリームをどのようにして V L C ビットストリームと結合させ、一緒に並列的に伝送されるようにするか、又は、両方のビットストリーム、即ち V L C ビットストリームとインターリーブされた PIPE ビットストリームに 2 回目のインターリーブを施すかについての可能性を説明する。

【0097】

有限のソースシンボル列の終了

PIPE 符号器および復号器の実施形態において、符号化と復号化は、ソースシンボルの有限集合に対して行われる。多くの場合、静止画像、ビデオシーケンスのフレームやフィールド、画像のスライス、ビデオシーケンスのフレーム又はフィールドのスライス、又は連続的なオーディオサンプルの集合などのデータの所定の量が符号化される。ソースシンボルの有限集合の場合、一般に、符号器側で作成された部分的ビットストリームを終了させる必要がある。即ち、伝送済み又は記憶済みの部分的ビットストリームから全てのソースシンボルを復号化できることが保証されなければならない。最後のビンが対応するビンバッファ 8 内へ挿入された後に、ビン符号器 10 は、完全な符号語が部分的ビットストリーム 12 に書き込まれることを保証しなければならない。もしビン符号器 10 がバイナリ算術符号化エンジンを表す場合には、算術符号語は終了されなければならない。もしビン復号器 10 が、ビン列を符号語に直接マッピングするエントロピー符号器を表す場合には、最後のビンをビンバッファに書き込んだ後にビンバッファに記憶されるビン列は、符号語に関連付けられているビン列を表さない場合がある（即ち、符号語に関連付けられている 2 つ以上のビン列の接頭語を表す場合がある）。このような場合、ビンバッファ内でそのビン列を接頭語として含むビン列に関連付けられているどのような符号語も、部分的ビットストリームに書き込まれる必要がある（ビンバッファはフラッシュされる必要がある）。これは、特定の又は任意の値を持つビンを、符号語が書き込まれるまで、ビンバッファへ挿入することにより、実行することができる。好ましい一実施形態において、ビン符号器は、（関連付けられたビン列がビンバッファ内でそのビン列を接頭語として含む必要があるという特性に加えて）最短の長さを有する符号語の 1 つを選択する。復号器側では、ビン復号器 22 は、部分的ビットストリーム内の最後の符号語に要求されるより多くのビンを復号化してもよく、これらのビンはビンバッファ選択部 18 によって要求されずに、廃棄されかつ無視される。シンボルの有限集合の復号化は、復号化済みのソースシンボルに対する要求によって制御される。即ち、もしある量のデータに対して更なるソースシンボルが要求されない場合、その復号化は終了する。

【0098】

部分的ビットストリームの伝送と多重化

PIPE 符号器によって作成される部分的ビットストリーム 12 は、別個に伝送されることが可能であり、若しくは単一のビットストリーム内へ多重化されることが可能であり、又は、部分的ビットストリームの符号語が単一のビットストリーム内へインターリーブされることが可能である。

【0099】

ある実施形態においては、ある量のデータに係る各部分的ビットストリームは 1 つのデー

10

20

30

40

50

タパケットへ書き込まれる。その量のデータとは、静止画像、ビデオシーケンスのフィールドやフレーム、静止画像のスライス、ビデオシーケンスのフィールド又はフレームのスライス、又はオーディオサンプルのフレームなどの、ソースシンボルの任意の集合であってもよい。

【0100】

他の好ましい実施形態においては、ある量のデータに係る部分的ビットストリームの2つ以上、又はある量のデータに係る全ての部分的ビットストリームが、1つのデータパケット内へ多重化される。多重化された部分的ビットストリームを含むデータパケットの構造を図5に示す。つまり、図5に示すデータパケットは、それぞれ中間的なインターリーブ済みのストリーム132と234の一部であってもよい。

10

【0101】

データパケット300は、（考慮対象となるある量のデータに係る）各部分的ビットストリームのデータに関するヘッダと1つのパーティションとから構成される。データパケットのヘッダ301は、データパケット（の残り）をビットストリームデータのセグメント302へと区分するための指示を含む。パーティション（区分）についての指示に加え、ヘッダは、更なる情報を含み得る。一実施形態においては、データパケットの区分についての指示は、ビット若しくはバイトの単位、又は多数のビット若しくは多数のバイトの単位の、データセグメントの始点の位置である。一実施形態において、データセグメントの始点の位置は、データパケットのヘッダ内に絶対値として符号化されており、データパケットの始点に対する位置か、若しくはヘッダの終点に対する位置か、又は先行するデータパケットの始点に対する位置である。更なる実施形態においては、データセグメントの始点の位置は、差分的に符号化される。即ち、データセグメントの実際の始点と、データセグメントの始点についての予測と、の間の差のみが符号化される。この予測は、データパケットの全体のサイズ、ヘッダのサイズ、データパケット内のデータセグメントの数、先行するデータセグメントの始点の位置など、既に既知であるか又は伝送された情報に基づいて導出可能である。一実施形態においては、最初のデータパケットの始点の位置は符号化されないが、データパケットのヘッダのサイズに基づいて推測される。復号器側においては、伝送されたパーティションの指示は、データセグメントの始点を導出するために使用される。次いで、データセグメントは部分的ビットストリームとして使用され、データセグメントに含まれるデータは対応するビン復号器に順に供給される。

20

30

【0102】

部分的ビットストリームをデータパケット内へ多重化するための方法には、いくつかの選択肢がある。一つの方法は、部分的ビットストリームのサイズ同士が非常に近い場合は特に必要なサイド情報を低減できる方法であり、図6に示す。データパケットのペイロード、即ちそのヘッダ311なしのデータパケット310は、予め定義された方法でセグメント312へと区分される。一例として、データパケットのペイロードは同一サイズのセグメントへと区分されることができる。次に、各セグメントは部分的ビットストリーム又は部分的ビットストリーム313の最初の部分と関連付けられる。部分的ビットストリームが関連付けられたデータセグメントよりも大きい場合には、その残りの部分314は、他のデータセグメントの終点における未使用のスペースに配置される。これは、ビットストリームの残りの部分が、（データセグメントの終点から開始する）逆の順序で挿入される方法で実行されることができ、サイド情報が削減される。部分的ビットストリームの残りの部分とデータセグメントとの関連付け、及び、2つ以上の残りの部分がデータセグメントへ追加される場合に、1つ以上の残りの部分に関する始点は、ビットストリームの内部、例えばデータパケットヘッダ内で、信号化される必要がある。

40

【0103】

可変長符号語のインターリーピング

幾つかのアプリケーションにおいては、1つのデータパケット内の（ある量のソースシンボルに係る）部分的ビットストリーム12の上記した多重化は、次のような欠点を持つ可能性がある。即ち、一方では、小さいデータパケットにとっては、パーティションについ

50

て信号化するために必要とされるサイド情報のためのビット数が、部分的ビットストリーム内の実際のデータに対して重大になりうるため、最終的に符号化効率を低下させる。他方では、多重化は、（例えばビデオ会議アプリケーションなどの）低遅延が要求されるアプリケーションには、不向きである可能性がある。上述した多重化の場合、部分的ビットストリームが完全に作成される以前に、P I P E符号器がデータパケットの伝送を開始することは不可能である。なぜなら、パーティションの始点の位置が事前に知られていないからである。さらに、一般に、P I P E復号器は、データパケットの復号化を開始し得る前に、最後のデータセグメントの始点を受信するまで待機する必要がある。ビデオ会議システムのようなアプリケーションにおいては、これらの遅延は合計されて、複数のビデオ画像を用いたシステム（特に、伝送ビットレートに近いビットレートに関して、および1つの画像を符号化／復号化するために2つの画像間の時間区間に近い要求をする符号器／復号器に関して）の追加的な全体的遅延となる可能性があり、このようなアプリケーションには致命的である。特定のアプリケーションに関するこのような欠点を克服するために、P I P E符号器は、2つ以上のビン符号器によって生成される符号語が、単一のビットストリームにインターリーブされるよう構成され得る。インターリーブ済みの符号語を有するビットストリームは、復号器に直接的に送信されることができ（小さなバッファ遅延を無視する場合は以下を参照）。P I P E復号器側においては、2つ以上のビン復号器は、復号化の順序でビットストリームから直接的に符号語を読み出す。即ち、復号化は最初に受信したビットから開始され得る。更に、部分的ビットストリームの多重化（又はインターリーブング）を信号化するために、サイド情報は必要とされない。

10

20

【0104】

符号語のインターリーブングを有するP I P E符号器の基本構造を図7に示す。ビン符号器10は、部分的ビットストリームに符号語を直接的には書き込まないが、単一の符号語バッファ29に接続されており、そこから符号語が符号化の順序でビットストリーム34へと書き込まれる。ビン符号器10は、1つ以上の新たな符号語バッファエントリに対する要求28を符号語バッファ29へと送信し、この後、符号語30を符号語バッファ29に送信し、これら符号語が、リザーブされたバッファエントリに記憶される。符号語バッファ29の（一般的には可変長の）符号語31は、符号語ライタ32によってアクセスされ、その符号語ライタ32は、対応するビット33を生成されたビットストリーム34へと書き込む。符号語バッファ29は、先入れ先出しバッファとして作動し、先にリザーブされた符号語エントリはビットストリームへ先に書き込まれる。

30

【0105】

更に一般化すれば、符号語バッファの数がビン符号器の数より少ない場合、複数の符号語バッファと部分的ビットストリーム12とが可能である。1つのビン符号器10は、符号語バッファ29内に1つ以上の符号語をリザーブし、それにより、符号語バッファ内における1つ以上の符号語のリザーブは、接続されたビンバッファ8内のある事象によってトリガーされる。ある実施形態においては、ビットストリーム34を、即ち図1aでは132と対応し、図2では134に対応するビットストリームを、P I P E復号器が瞬時に復号化できる方法で、符号語バッファ29が作動することができる。符号語がビットストリームに書き込まれる符号化の順序は、対応する符号語が符号語バッファの中にリザーブされる順序と同じである。ある実施形態においては、各ビン符号器10が1つの符号語をリザーブし、そのリザーブのトリガーは、接続されたビンバッファ内の所定の事象により実行される。他の実施形態においては、各ビン符号器10が2つ以上の符号語をリザーブし、そのリザーブのトリガーは、接続されたビンバッファ内の所定の事象により実行される。更に他の実施形態においては、特定のビン符号器によってリザーブされる符号語の量が、特定のビン符号器及び／又は（例えば関連する確率測度、既に書き込まれたビット数などの）特定のビン符号器／ビンバッファの他の特性に依存し得る場合、各ビン符号器10が異なる量の符号語をリザーブする。

40

【0106】

一実施形態においては、符号語バッファは次のように作動する。即ち、新たなビン7が特

50

定のピンバッファ 8 に送られ、ピンバッファ内に既に記憶されていたピンの数がゼロであり、かつその特定のピンバッファに接続されたピン符号器のために符号語バッファ内にリザーブされた符号語が現時点で存在しない場合には、この接続されたピン符号器 10 は符号語バッファへ要求を送信し、それにより、1 つ以上の符号語エントリが特定のピン符号器のために符号語バッファ 29 内でリザーブされる。符号語エントリは可変数のビットを有することができる。即ち、1 つのバッファエントリ内のビット数の上限しきい値は、通常、対応するピン符号器に係る最大符号語サイズによって与えられる。ピン符号器によって生成される次の単数又は複数の符号語（それら符号語のために単数又は複数の符号語エントリがリザーブされている）は、符号語バッファのリザーブされた単数又は複数のエントリ内に記憶される。もし、符号語バッファ内の特定のピン符号器のためにリザーブされた全てのバッファエントリが符号語で満たされており、かつ、その特定のピン符号器に接続されたピンバッファに対して次のピンが送信される場合には、その特定のピン符号器に関し、1 つ以上の新たな符号語が、符号語バッファ内においてリザーブされる、等が実行される。符号語バッファ 29 は、ある所定の方法での先入れ先出しバッファを表す。バッファエントリは、順にリザーブされる。対応するバッファエントリが先にリザーブされている符号語は、先にビットストリームへ書き込まれる。符号語ライタ 32 は、連続的に、又は符号語 30 が符号語バッファ 29 へと書き込まれた後に、符号語バッファ 29 の状態をチェックする。最初のバッファエントリが完全な符号語を含む場合（即ち、バッファエントリはリザーブされていないが符号語を含む場合）には、対応する符号語 31 と対応するバッファエントリは符号語バッファ 20 から取り除かれ、符号語 33 のビットはビットストリームへ書き込まれる。この処理は、最初のバッファエントリが符号語を含む（即ち、リザーブされるか又は空である）状態を抜けるまで繰り返される。復号化処理の終了時、即ち、考慮対象となる所定量のデータの全てのソースシンボルが処理された場合には、符号語バッファをフラッシュする必要がある。そのフラッシュ処理に関しては、以下の方法が第 1 のステップとして各ピンバッファ／ピン符号器に適用される。即ち、ピンバッファがピンを含む場合には、結果として得られるピン列が、符号語に関連付けられたピン列を表すようになるまで、特定の又は任意の値を有するピンが加えられる（上記したように、ピンを追加する 1 つの好ましい方法は、接頭語としてピンバッファの元のコンテンツに関するピンを含むピン列に関連付けられた、最短の長さの可能な符号語—又はそれらのうちの 1 つ—を生成するようなピン値を追加することである）。次に、符号語は対応するピン符号器のための次にリザーブされたバッファエントリへ書き込まれ、（対応する）ピンバッファは空になる。もし 2 つ以上バッファエントリが 1 つ以上のピン符号器に対してリザーブされている場合、符号語バッファは、リザーブされた符号語エントリをまだ含んでいる可能性がある。その場合、これらの符号語エントリは、対応するピン符号器のために、任意であるが、有効な符号語によって満たされる。ある好ましい実施形態においては、最短の有効な符号語、又は（複数ある場合には）最短の有効な符号語の 1 つが挿入される。最後に、符号語バッファ内の残りの全ての符号語が、ビットストリームに書き込まれる。

【0107】

符号語バッファの状態を表す 2 つの例を図 8 に示す。この例（a）において、符号語バッファは、符号語で満たされた 2 個のエントリと、リザーブされた 5 個のエントリを含む。加えて、次の空きのバッファエントリも記載されている。最初のエントリは、符号語で満たされている（即ち、ピン符号器 2 が前にリザーブされたエントリへ符号語を書き込んだばかりである）。次のステップにおいて、この符号語は符号語バッファから取り除かれ、ビットストリームへ書き込まれる。次に、ピン符号器 3 のための最初にリザーブされた符号語は、第 1 のバッファエントリである。しかし、このエントリを符号語バッファから取り除くことができない。なぜなら、このエントリはリザーブされているだけであって、このエントリにはこれまで全く符号語が書き込まれていないからである。例（b）において、符号語バッファは、符号語で満たされた 3 個のエントリとリザーブされた 4 個のエントリを含む。最初のエントリはリザーブされていると記載されており、従って、符号語

ライタは符号語をビットストリームに書き込むことはできない。3個の符号語が符号語バッファに含まれているが、符号語ライタは、ビン符号器3のための最初にリザーブされたバッファエントリに符号語が書き込まれるまで待機しなければならない。符号語は、復号器側で処理を逆転させることが可能となるように（下記参照）、それらがリザーブされた順序で書き込まれる必要があることに留意されたい。

【0108】

符号語インターリーブを有するPIPE復号器の基本的な構造を図9に示す。ビン復号器10は、個別の部分的ビットストリームから直接的に符号語を読み出さず、ビットバッファ38に接続されており、このビットバッファ38から符号語37が符号化の順序で読み出される。符号語はビットストリームから直接的に読み出されることも可能であるため、ビットバッファ38は必ずしも必要ではないことに留意されたい。ビットバッファ38をこの図に含めた理由は、主に、処理の流れの異なる態様を明確に区別するためである。インターリーブ済みの符号語を持つビットストリーム40、つまり図2ではビットストリーム234に対応するビットストリームのビット39は、ビットバッファ38内に順次挿入され、このビットバッファ38は先入れ先出しバッファを表している。もし特定のビン復号器22が1つ以上のビン列に対する要求35を受信した場合には、ビン復号器22は、ビットに対する要求36を介して、ビットバッファ38から1つ以上の符号語37を読み出す。PIPE復号器は、ソースシンボルを瞬時に復号化することができる。（上記のような）PIPE符号器は、符号語バッファを適切に作動させることにより、符号語がビン復号器によって要求されるのと同じ順序で、符号語がビットストリームへ書き込まれることを確保する必要があることに留意されたい。PIPE復号器においては、全体の復号化プロセスは、ソースシンボルに対する要求によってトリガーされる。特定のビン符号器によって符号器側でリザーブされた符号語の数としてのパラメータと、対応するビン復号器によって読み出される符号語の数としてのパラメータとは、同一でなければならない。

【0109】

更に一般化すれば、ビットバッファの数がビン復号器の数より少ない場合には、多数の符号語バッファと部分的なビットストリームが可能である。ビン復号器22は、ビットバッファ38から1つ以上の符号語をある時刻において読み出し、ここで、ビットバッファからの1つ以上の符号語のその読み出しは、接続されたビンバッファ20内の所定の事象によってトリガーされる。一実施形態においては、ビンに対する要求19が特定のビンバッファ20に送信されたときに1つ以上の符号語が読み出され、かつビンバッファはビンを含まないような方法で、復号器が作動する。しかし、符号語の読み出しをトリガーすることは、例えばビンバッファ内のビン数が予め定義されたしきい値を下回る場合など、他の事象によっても可能である。ある実施形態においては、各ビン復号器22が1つの符号語を読み出すが、この読み出しは、接続されたビンバッファ内の所定の事象によってトリガーされる。他の実施形態においては、各ビン復号器22が2つ以上の符号語を読み出すが、この読み出しは、接続されたビンバッファ内の所定の事象によってトリガーされる。更なる実施形態においては、特定のビン復号器によって読み出される符号語の量が、特定のビン復号器および／又は（例えば関連する確率測度、既に読み出されたビット数などの）特定のビン復号器／ビンバッファの他の特性に依存し得る場合に、ビン復号器22が異なる量の符号語を読み出す。

【0110】

ある実施形態においては、ビットバッファからの符号語の読み出しは以下のように操作される。もし新たなビン要求19がビンバッファ選択部18から特定のビンバッファ20へ送信され、ビンバッファ内のビンの数がゼロである場合には、接続されたビン復号器22は、ビットバッファ38に対するビット要求36を介して、ビットバッファ38から1つ以上の符号語37を読み出す。ビン復号器22は読み出された符号語37をビンの列21へ変換し、これらのビン列を接続されたビンバッファ20内に記憶する。ビンに対する要求19への最終応答として、第1の挿入されたビンはビンバッファ20から取り除かれ、ビンバッファ選択部18へ送られる。更なるビン要求に対する応答として、ビンバッファ

内に残っているピンは、ピンバッファが空になるまで除去される。追加のピン要求は、ピン復号器をトリガーし、ビットバッファから1つ以上の新たな符号語を読み出す、などの動作をさせる。ビットバッファ38は、予め定義されたサイズの先入れ先出しバッファを表し、ビットストリーム40からのビット39により連続的に満たされる。符号語が復号化プロセスによって要求されたときと同じ方法で、符号語がビットストリームに確実に書き込まれるように、符号器側の符号語バッファは、上述の方法と同様に操作されることができる。

【0111】

従って、複数のエントロピー復号器の各々は、固定長の符号語を可変長のシンボル列にマップするよう構成された可変長復号器であってもよく、符号語バッファ43の出力などの符号語エントリは、インターリーブ済みの符号語の単一のストリームを受け取るために提供されてもよい。複数のエントロピー復号器22は、符号語エントリから順次符号語を回収するよう構成されてもよく、その順序は、複数のエントロピー復号器から選択部18によって回収されるにつれて復元されるべきシンボル列のシンボルが、それぞれのエントロピー復号器において新たな符号語からマップされるべき新たなシンボル列をもたらすような順序に基づいていてもよい。

【0112】

低遅延制約のある可変長符号語のインターリーブング

ここで説明するPIPE符号化に関する符号語のインターリーブングは、何らかのパーティション情報がサイド情報として送信されることを必要とするものではない。また、符号語はビットストリーム内でインターリーブされるので、遅延も一般的に小さい。しかしながら、（例えば、符号語バッファに記憶される最大ビット数によって指定されるような）特定の遅延制約が守られることを保証するものではない。更に、符号語バッファに必要とされるバッファのサイズは、理論的には非常に大きくなる可能性がある。図8(b)に示す例を考慮した場合には、これ以上のピンがピンバッファ3へ送信されず、従って、データパケットの終点におけるフラッシュ処理が適用されるまで、ピン符号器3が符号語バッファに新たな符号語を1つも送信しないという可能性がある。その場合、ピン符号器1および2に係る全ての符号語は、それらがビットストリームへ書き込まれ得る前に、データパケットの終点まで待機しなければならない可能性がある。このような欠点は、更なるメカニズムをPIPE符号化プロセスに（及び後述するようにPIPE復号化プロセスにも）追加することにより、回避することができる。この追加メカニズムの基本的な概念は、もし遅延又は遅延の上限（以下を参照）に関連した測度が指定されたしきい値を超える場合には、最初のリザーブされたバッファエントリは、（データパケットの終点と同様のメカニズムを用いて）対応するピンバッファをフラッシュすることによって満たされる、というものである。このようなメカニズムによって、関連付けられた遅延測度が指定されたしきい値を下回るまで待機しているバッファエントリの数が削減される。復号器側においては、遅延制約を守るために、符号器側で挿入されていたピンを廃棄する必要がある。このようなピンの廃棄のためには、基本的に符号器側と同様のメカニズムを使用することができる。以下に、このような遅延制御のための2つの実施形態を説明する。

【0113】

一実施形態において、遅延（又は遅延の上限）の測度は、符号語バッファ内のアクティブなバッファエントリの数であり、ここで、アクティブなバッファエントリの数とは、リザーブされたバッファエントリの数と符号語を含むバッファエントリの数との合計である。最初のバッファエントリは常に、リザーブされたバッファエントリ又は空きのバッファエントリであることに留意されたい。なぜなら、もし最初のバッファエントリが符号語を含む場合には、この符号語はビットストリームに書き込まれるからである。例えば、（アプリケーションにより決定される）最大許容バッファ遅延がD個のビットであり、全てのピン符号器に係る最大符号語サイズがLとすると、遅延の制約を犯さずに、符号語バッファ内に含まれることができる符号語の最大数に関する下限は、 $N = D / L$ によって算出することができる。ビットにおける遅延測度Dはシステムによって必要とされる訳ではないが

、符号語の最大数 N は符号器と復号器との両方に既知である必要がある。ある実施形態においては、符号語バッファエントリの最大数 N はアプリケーションによって固定される。他の実施形態において、符号語バッファエントリの最大数 N は、ビットストリームの内部で、即ち例えばビットストリームに含まれるデータパケットのヘッダ（又はスライスヘッダ）内又はパラメータ集合内で、信号化されている。もしピン符号器 10 が、1 つ以上の新たなバッファエントリのリザーブに対する要求を符号語バッファ 29 に送信する場合には、新たな符号語バッファエントリがリザーブされる前に、以下のプロセスが実行される（即ち、多数の符号語バッファエントリが 1 つの要求によってリザーブされる場合は、多数回実行される）。もし、現時点でアクティブなバッファエントリ数に（次にリザーブされるバッファエントリを考慮して）1 を加算した数が符号語バッファエントリの最大数 N より大きい場合には、（リザーブされた）最初のバッファエントリは、現時点でアクティブなバッファエントリ数に 1 を加算した数が符号語バッファエントリの最大数 N 以下になるまで、次に記載するプロセスによってフラッシュされる。リザーブされたバッファエントリのフラッシュは、データパケットの終点におけるフラッシュに類似している。対応する最初のバッファエントリをリザーブしていたピン符号器 10 は、結果として得られるピン列が符号語と関連付けられたピン列を表すまで、接続されたピンバッファ 8 に対し、特定の又は任意のピン値を持つピンを追加することによってフラッシュされる。次に、符号語はリザーブされたバッファエントリに書き込まれ、それが最終的に（ピンバッファを空にし、前にリザーブされたバッファエントリを削除しつつ）ビットストリームへ追加される。前述したように、ピンバッファにピンを追加するための 1 つの好ましい方法は、可能性のある最短の符号語を生成するピンを追加することである。復号器側では、遅延制約を守るために追加されたピンを廃棄するために、同様のプロセスが実行される。そのため、復号器はビットバッファから読み出された符号語をカウントするカウンタ C を保有する（このカウンタはビットバッファ内で保有され得る）。カウンタ C はデータパケットの復号化の開始時に（例えばゼロを用いて）初期化され、符号語の読み出し後に 1 だけ増加される。加えて、各ピン復号器 22 はカウンタ C_x を含み、このカウンタ C_x は、最後の符号語が対応するピン復号器 22 によって読み出される前に、符号語カウンタ C の値を記憶する。即ち、特定のピン復号器 22 が新たな符号語を読み出す場合には、第 1 のステップとして、そのカウンタ C_x が C と等しくなるよう設定され、次に符号語がビットバッファから読み出される。ピンに対する要求 19 が特定のピンバッファ 20 へ送信され、全体の符号語カウンタ C と接続されたピン復号器 22 のカウンタ C_x との差（ $C - C_x$ ）が、符号語バッファエントリの最大数 N よりも大きい場合には、特定のピンバッファ 20 に現時点で記憶されている全てのピンが除去されかつ無視される。この追加的なステップ以外に、復号化は、上記のように操作される。もしピンに対する要求 19 が送信されるべきピンバッファ 20 が（全てのピンが既に除去されているため、又はピン要求が受信された後に低遅延メカニズムが第 1 のステップにおいて全てのピンを廃棄したために）空である場合には、接続されたピン復号器 22 は、ビットバッファ 38 などから 1 つ以上の新たな符号語を読み出す。

【0114】

別の実施形態において、遅延（又は遅延の上限）についての測度は、符号語バッファ内の（複数の）アクティブなバッファエントリに係る最大符号語の長さの合計であり、ここで、ある特定のバッファエントリに係る最大符号語の長さは、そのバッファエントリに関連付けられた復号化済みのピンに基づいている。説明のため、バッファエントリの最大符号語長を図 8 の例に示す。最初のバッファエントリは、常にリザーブされたバッファエントリ又は空きのバッファエントリであることに改めて留意されたい。なぜなら、もし最初のバッファエントリが符号語を含む場合には、この符号語はビットストリームに書き込まれるからである。（アプリケーションにより決定される）最大許容バッファ遅延が D 個のビットと仮定すると、この最大バッファ遅延 D は符号器と復号器との両方に知られる必要がある。好ましい実施形態においては、最大バッファ遅延 D はアプリケーションによって固定される。別の好ましい実施形態においては、最大バッファ遅延 D は、ビットストリーム

の内部で、即ち例えばビットストリームに含まれるデータパケットのヘッダ（又はスライスヘッダ）内又はパラメータ集合内で、信号化されている。最大バッファ遅延Dは、ビットもしくはバイトの単位、又は多数のビットもしくは多数のバイトの単位で信号化できる。もしピン符号器10が、符号語バッファ29に1つ以上の新たなバッファエントリのリザーブに対する要求を送信する場合には、新たな符号語バッファエントリがリザーブされる前に、以下のプロセスが実行される（即ち、多数の符号語バッファエントリが1つの要求によってリザーブされる場合は、多数回実行される）。

【0115】

現時点でアクティブな全てのバッファエントリに係る最大符号語の長さ、リザーブされるであろうバッファエントリの最大符号語の長さ、の合計が、最大バッファ遅延Dよりも大きい場合には、全てのアクティブなバッファエントリに係る最大符号語の長さとリザーブされるであろうバッファエントリに係る最大符号語の長さとの合計が、最大バッファ遅延D以下になるまで、第1の（リザーブされた）バッファエントリは、上記したプロセスによってフラッシュされる。一例として、図8（b）の例について考察する。現時点でアクティブな全てのバッファエントリの最大符号語の長さの合計は29である。ここで、最大バッファ遅延Dが32に等しいと設定されると仮定する。もし次のバッファエントリが、最大符号語長が3と等しいピン符号器2によってリザーブされる場合には、最初のバッファエントリはフラッシュされない。なぜなら、 $29 + 3$ は32より大きくないからである。しかし、もし次のバッファエントリが、最大符号語長が7と等しいピン符号器1によってリザーブされる場合には、第1のバッファエントリはフラッシュされる。なぜなら、 $29 + 7$ は32より大きいからである。リザーブされたバッファエントリのフラッシュは、（対応するピンバッファに特定又は任意の値を持つピンを追加することによって）上記のように行われる。

【0116】

復号器側においては、遅延の制約に従うために追加されたピンを廃棄するために、同様のプロセスが実行される。そのため、復号器は、ビットバッファから読み出された符号語に係る最大符号語の長さをカウントする、カウンタCを保有する（このカウンタはビットバッファ内に保有され得る）。異なるピン復号器と関連付けられた最大符号語長は、異なる場合があることに留意されたい。カウンタCは、データパケットの復号化の開始時に（例えばゼロを用いて）初期化され、符号語の読み出し後に増加される。このカウンタは、読み出された符号語の実際の長さではなく、その最大長さの分だけ増加される。即ち、もしある符号語が特定のピン復号器によって読み出され、その特定のピン復号器によって使用される符号語テーブルと関連付けられる最大符号語長がLxである場合（異なるピン復号器は異なる最大符号語長に関連付けられ得る）、カウンタCはLxだけ増加される。全体的なカウンタCに加えて、各ピン復号器22はカウンタCxを含み、このカウンタCxは、最後の符号語が対応するピン復号器22によって読み出される前に、符号語カウンタCの値を記憶する。即ち、もし特定のピン復号器22が新たな符号語を読み出す場合には、第1のステップとして、そのカウンタCxはCと等しく設定され、次に、符号語がビットバッファから読み出される。ピンに対する要求19が特定のピンバッファ20へと送信され、かつ、全体的なカウンタCと接続されたピン復号器22のカウンタCxとの差（ $C - Cx$ ）が最大バッファ遅延Dよりも大きい場合には、その特定のピンバッファ20に現時点で記憶されている全てのピンが廃棄され、無視される。この追加的なステップに加え、復号化は、上述した操作が実行される。もしピンに対する要求19が送信されるべきピンバッファ20が、（全てのピンがすでに除去されていたため、又はピン要求が受信された後に低遅延メカニズムが第1のステップにおいて全てのピンを廃棄したために）空の場合には、接続されたピン復号器22は、ビットバッファ38などから1つ以上の新たな符号語を読み出す。

【0117】

従って、複数のエントロピー復号器22と選択部18とは、復元されるべきシンボル列27の形成に関わらないように、シンボル列の接尾語を断続的に破棄するよう構成されても

よい。この断続的な破棄は、符号語エントリから複数のエントロピー復号器によって回収された幾つかの符号語であって、符号語エントリからのそれぞれのエントロピー復号器による2つの連続的な符号語回収の間に回収された符号語が、予め決定された基準を満たすという事象の発生に基づいて、実行されてもよい。次いで、複数のエントロピー符号器と符号語バッファとは、現在転送されているが未だマップされていないシンボルを接頭語として有するドント・ケアシンボルによって、現在転送されているが未だマップされていないシンボルを有効なシンボル列へと断続的に拡張し、そのように拡張されたシンボル列を符号語へとマップし、そのように得られた符号語をリザーブされた符号語エントリに入力し、更に符号語エントリをフラッシュするよう構成されてもよい。断続的な拡張、入力及びフラッシュは、幾つかのリザーブされた符号語エントリとその中に入力された符号語を有する幾つかの符号語エントリとの合計が、予め決定された基準を満たすという事象の発生に基づいて実行されてもよい。予め決定された基準は、複数の符号器／復号器ペアの最大符号語長を考慮に入れることができる。

10

【0118】

いくつかのアーキテクチャにおいて、符号語のインターリーピングに係る上述の好適な実施形態は、復号化の複雑性の点において欠点が生じる可能性がある。図9に示すように、全てのビン復号器22は、単一のビットバッファ38から符号語（一般的な場合には可変長符号語）を読み出す。符号語は正しい順序で読み出される必要があるので、符号語の読出しは並列に実行することができない。これは、特定のビン復号器は、他のビン復号器が符号語の読み出しを終了するまで待機する必要があることを意味する。更に、可変長符号語の読出しの複雑さが（部分的に並列化された）復号化プロセスの残りの部分との関係において有意である場合には、可変長符号語のこのアクセスが全体の復号化プロセスにとって障害になり得る。単一のビットバッファからのアクセスの複雑さを低減するために使用可能な上述した実施形態のいくつかの変形例があり、そのうちの幾つかを以下に説明する。好ましい一実施形態においては、（例えば冗長度のない接頭語符号を表している）符号語の単一の集合が存在し、各ビン復号器22に使用される符号語の集合は、その単一符号語の集合の部分集合である。異なるビン復号器22は、単一符号語の集合の異なる部分集合を使用できることに留意されたい。たとえビン復号器22の一部によって使用される符号語の集合が同じであった場合でも、それらとビン列との関連付けは、異なるビン復号器22については異なる。ある特定の実施形態においては、符号語の同一の集合が全てのビン復号器22のために使用される。全てのビン復号器のための（複数の）符号語の集合を含む単一符号語の集合を（複数の）部分集合として有する場合、符号語の解析はビン復号器外で行うことができることから、符号語アクセスの複雑さを低減できる。PIPE符号化プロセスは、上述のプロセスとの関係においては変更されない。修正されたPIPE復号化プロセスを図10に示す。単一の符号語リーダーは、ビットストリーム40からビット46を供給され、一般に可変長である一符号語を解析する。読み出された符号語44は、先入れ先出しバッファを表す符号語バッファ43内に挿入される。ビン復号器22は、1つ以上の符号語に対する要求41を符号語バッファ43に送信し、更に、この要求への応答として、1つ以上の符号語が符号語バッファから（順に）取り除かれ、対応するビン復号器22へと送信される。この実施形態によれば、潜在的に複雑な符号語の解析は背景処理で行うことができ、ビン復号器のために待機する必要がないことに留意されたい。ビン復号器は、既に解析された符号語へとアクセスし、潜在的に複雑な符号語の解析はもはや全体的なバッファに対する要求の一部ではない。代わりに、既に解析された符号語がビン復号器に送信されるが、符号語のインデックスだけがビン復号器に送信されるような方法で実施することも可能である。

20

30

40

【0119】

固定長ビット列のインターリーピング

PIPE復号器の複雑性を低減する更なる方法が達成される場合として、ビン復号器22が、グローバル・ビットバッファ38から可変長符号語を読み出さず、代わりに、グローバル・ビットバッファ38から常に固定長のビット列を読み出し、これら固定長のビット

50

列をローカルビットバッファへ追加する場合が挙げられる。このとき、各ビン復号器 2 2 は分離されたローカル・ビットバッファと接続されている。次いで、可変長符号語がローカル・ビットバッファから読み出される。これにより、可変長符号語の解析は並列的に実行することができ、固定長のビット列のアクセスのみが同期的な方法で実行される必要があるが、固定長のビット列のこのようなアクセスは通常は非常に迅速であり、これにより、全体的な復号化の複雑性は一部のアーキテクチャについては低減できる可能性がある。特定のローカル・ビットバッファに送信されるビンの固定数は、異なるローカル・ビットバッファごとに異なる可能性があり、ビン復号器、ビンバッファ又はビットバッファ内に発生する事象としての所定のパラメータに基づいて、経時的に変化する。しかしながら、特定のアクセスによって読み出されるビットの数は、その特定のアクセスの間に読み出される実際のビットに依存しない。この点は可変長の符号語の読み出しに対する重要な相違点である。固定長のビット列の読み出しは、ビンバッファ、ビン復号器又はローカル・ビットバッファ内における所定の事象によってトリガーされる。一例として、接続されたビットバッファ内に存在しているビットの数が予め決定されたしきい値を下回った場合に、新たな固定長のビット列の読み出しを要求することが可能であり、このとき、異なるビットバッファについては異なるしきい値を使用することが可能である。符号器では、固定長のビット列が、それらが復号器側でビットストリームから読み出されるのと同じ順序でビットストリーム内へ挿入されることを保証する必要がある。この固定長のビット列のインターリービングと、上述したものと同様の低遅延コントロールと、を組み合わせることも可能である。固定長のビット列のインターリービングのための好ましい実施形態を以下に説明する。

10

20

【0120】

図 11 は、2 つ以上のビン符号器のための固定長のビット列をインターリーブする本発明の実施形態に係る基本的な PIPE 符号器の構造を示す図である。図 7 に示された実施例とは対照的に、ビン符号器 10 は、単一の符号語バッファに接続されていない。代わりに、各ビン符号器 10 は、対応する部分的ビットストリームのためのビットを記憶する、個別のビットバッファ 48 と接続されている。全てのビットバッファ 48 は、グローバル・ビットバッファ 51 に接続されている。グローバル・ビットバッファ 51 はビットライタ 53 に接続されており、このビットライタ 53 は、グローバル・ビットバッファから符号化／復号化の順序でビット 52 を取り除き、取り除かれたビット 54 をビットストリーム 55 へ書き込む。特定のビットバッファ 48 又は接続されたビン符号器 10 もしくはビンバッファ 8 内に発生するある所定の事象に基づいて、ビットバッファ 48 はグローバル・ビットバッファ 51 へ要求 49 を送信し、これにより、幾つかのビットの数がグローバル・ビットバッファ 51 内にリザーブされる。固定長のビット列のリザーブに対する要求 49 は順に処理される。グローバル・ビットバッファ 51 は、所定方法の先入れ先出しバッファを表す。即ち、先にリザーブされたビットはビットストリームに先に書き込まれる。異なるビットバッファ 48 は、既に符号化済みのシンボルに基づいて経時的に変化可能な異なる量のビットをリザーブすることができるが、特定の要求によってリザーブされたビットの数は、要求がグローバル・ビットバッファに送信される時点で知られていることに留意されたい。

30

40

【0121】

特定の実施形態においては、ビットバッファ 48 とグローバル・ビットバッファ 51 とは以下に説明するように操作される。特定のビットバッファ 48 によってリザーブされたビットの量は N_x と表記される。このビットの数 N_x は、異なるビットバッファ 48 ごとに異なることができ、また経時的にも変化し得る。好ましい実施形態においては、特定のビットバッファ 48 によってリザーブされたビットの数 N_x は経時的に固定される。ビット 49 の固定数 N_x 個に対するリザーブは、ビットバッファ 48 内のビット数 M_x 個と、リザーブ要求のためのビット数 N_x 個と、関連付けられた最大符号語長 L_x と、に基づいてトリガーされる。各ビン符号器 10 は、異なる最大符号語長 L_x と関連付けることができることに留意されたい。もしビン 7 が特定のビンバッファ 8 へ送られ、その特定のビンバ

50

ッファ 8 が空であり、 $N \times$ 個のビットの 1 つ以下の列が、（ビン符号器を介して）その特定のビンバッファに接続されたビットバッファ 48 のためにグローバル・ビットバッファ内にリザーブされ、更に、（ビン符号器を介して）その特定のビンバッファ 8 に接続されたビットバッファ 48 のリザーブ要求によってリザーブされたビットの数 $N \times$ と、このビットバッファ 48 内に現時点で存在するビットの数 $M \times$ との差（ $N \times - M \times$ ）が、対応するビン符号器 10 と関連付けられた最大符号語長 $L \times$ より小さい場合には、接続されたビットバッファ 48 は、 $N \times$ 個のビットのリザーブに対する要求 49 をグローバル・ビットバッファ 51 へと送信する。グローバル・ビットバッファ 51 は、その特定のビットバッファ 48 のために $N \times$ 個のビットをリザーブし、次のリザーブのためにそのポインタを増加させる。 $N \times$ 個のビットがグローバル・ビットバッファにリザーブされた後、ビン 7 はビンバッファ 8 に記憶される。この単一のビンがある符号語に関連付けられたビン列を既に表している場合には、ビン符号器 10 は、ビンバッファ 8 からこのビンを取り除き、対応する符号語 47 を接続されたビットバッファ 48 へ書き込む。そうでない場合（この単一ビンが符号語に関連付けられたビン列を表していない場合）には、ビンバッファ 8 が符号語に関連付けられたビン列を含むまで、更なるビン 7 がその特定のビンバッファ 8 によって受け入れられる。この場合、接続されたビン符号器 10 は、ビンバッファ 8 からビン列 9 を取り除き、対応する符号語 47 を接続されたビットバッファ 48 に書き込む。ビットバッファ 48 内で結果として得られるビット数 $M \times$ 個がリザーブされたビットの数 $N \times$ 以上である場合には、ビットバッファ 48 に最初に書き込まれた $N \times$ 個のビットは、グローバル・ビットバッファ 51 内で以前にリザーブされたスペースへ挿入される。特定のビンバッファ 8 へ送信される次のビン 7 についても、上述と同様の処理が実行される。即ち、最初に新たな $N \times$ 個のビットがグローバル・ビットバッファ内にリザーブされる必要があるかどうか（ $N \times - M \times$ が $L \times$ より小さいかどうか）がチェックされ、次にビンがビンバッファ 8 に挿入される、等である。ビットライタは、グローバル・ビットバッファの固定長のビット列をそれらがリザーブされた順序で書き込む。もしグローバル・ビットバッファ 51 内の最初の固定長エントリが、グローバル・ビットバッファに実際に挿入された固定長のビット列を含む場合（即ち、リザーブされただけではない場合）には、ビットライタ 53 は、グローバル・ビットバッファ 51 からこのビット列 52 に係るビットを除去し、ビット 54 をビットストリームに書き込む。この処理は、グローバル・ビットバッファ内の最初の固定長エントリがリザーブされた又はフリーのエントリを表すまで、繰り返される。グローバル・ビットバッファ内の最初の固定長のエントリがリザーブされたエントリを表す場合には、ビットライタ 53 はこのエントリが実際のビットで満たされるまで待機し、その後、ビットストリーム 55 へ更なるビット 54 を書き込む。

【0122】

データパケットの終点において、ビンバッファは上記したようにフラッシュされる。更に、ビットバッファは、グローバル・ビットバッファ内の全てのリザーブされたバッファエントリが満たされ、ビットストリームに書き込まれるようになるまで、特定の又は任意の値を有するビットを追加することによってフラッシュする必要がある。

【0123】

図 12 に、グローバル・ビットバッファ 51 の可能性のある状態の 2 つの例を示す。この例（a）において、異なるビットバッファ／ビン符号器が異なる数のビットをリザーブするケースが図示されている。グローバル・ビットバッファは、実際に書き込まれた固定長のビット列を有する 3 つのエントリと、リザーブされた固定長のビット列を有する 4 つのエントリとを含む。最初の固定長エントリは、（ビットバッファ／ビン符号器 2 によって挿入されたばかりであるはずの）実際のビットを既に含む。このエントリ（即ち、対応する 8 ビット）は、除去されかつビットストリームに書き込まれることができる。次のエントリはビン符号器 3 のために 10 個のビットをリザーブするが、実際のビットはまだ挿入されていない。このエントリはビットストリームに書き込まれず、実際のビットが挿入されるまで待機する必要がある。第 2 の例（b）においては、全てのビットバッファ／ビン符号器が同一個数のビット（8 ビット）をリザーブしている。グローバル・ビットバッファ

ァは、8ビット列のための4つのリザーブと、実際に書き込まれた3つの8ビット列とを含む。最初のエントリはビン符号器3のための8ビットのリザーブを含む。任意の新たなビットがビットストリームに書き込まれ得る前に、ビットライタは、ビットバッファ／ビン符号器3がこのリザーブされたエントリに8ビットの実際の値を書き込むまで、待機する必要がある。

【0124】

図13は、固定長のビット列をインターリーブする本発明のある実施形態に係るPIPE復号器の構造を示す図である。図9に示す実施形態とは対照的に、ビン復号器22は単一のビットバッファ58には接続されていない。その代わり、各ビン復号器22は、個別のビットバッファ58に接続されており、このビットバッファ58は対応する部分的ビットストリームからビットを記憶する。全てのビットバッファ58はグローバル・ビットバッファ61に接続されている。ビットストリーム63からのビット62は、グローバル・ビットバッファ61内に挿入される。特定のビットバッファ58又は接続されたビン復号器22もしくはビンバッファ20内で、ある事象が発生すると、ビットバッファ58はグローバル・ビットバッファ61へ要求59を送信し、その要求により、固定長のビット列60がグローバル・ビットバッファ61から除去され、特定のビットバッファ58へ挿入される。固定長ビット列に対する要求59は、順に処理される。グローバル・ビットバッファ61は先入れ先出しバッファを表し、即ち、グローバル・ビットバッファへ先に挿入されたビットは先に除去される。異なるビットバッファ58は異なるビット量を要求することができ、この量は既に復号化済みのシンボルに基づいて経時的に変化し得るが、しかし、特定の要求によって要求されたビット数は、要求がグローバル・ビットバッファへ送られる時点で知られることに留意されたい。符号語はビットストリームから直接的に読み出すこともできるので、グローバル・ビットバッファ61は必ずしも必要とされないことに留意されたい。グローバル・ビットバッファ61は、主として、処理の流れの異なる態様を明確に区別ために図示されている。

【0125】

一実施形態においては、ビットバッファ58とグローバル・ビットバッファ61とは以下に説明するように操作する。特定のビットバッファ58によって要求されかつ読み出されるビットの量は Nx と表記され、この量は、符号器側において対応するビットバッファによってグローバル・ビットバッファに書き込まれるビット量と等しい。このビットの数 Nx は、異なるビットバッファ58ごとに異なることができ、また、経時的にも変化し得る。本発明のある好ましい実施形態においては、特定のビットバッファ58によって要求されかつ読み出されるビット数の Nx 個は、経時的に固定される。固定数の Nx 個のビット60の読み出しは、ビットバッファ58内のビット数 Mx 個と、関連付けられた最大符号語長 Lx とに基づいてトリガーされる。各ビン復号器22は、異なる最大符号語長 Lx と関連付けられ得ることに留意されたい。もしビンに対する要求19が特定のビンバッファ20へ送られ、その特定のビンバッファ20が空であり、(ビン復号器を介して)その特定のビンバッファ20に接続されたビットバッファ58内のビット数 Mx が、対応するビン復号器22と関連付けられた最大符号語長 Lx よりも小さい場合には、接続されたビットバッファ58は、新たな Nx 個のビットの列に対する要求59を、グローバル・ビットバッファ61へ送信する。この要求への応答として、第1の Nx 個のビットがグローバル・ビットバッファ61から除去され、この Nx 個のビットの列60は、この要求の送信元のビットバッファ58へ送信される。最終的に、この Nx 個のビットの列は、対応するビットバッファ58に追加される。次に、次の符号語57がこのビットバッファから読み出され、更に、接続されたビン復号器22は、関連付けられたビン列21を接続されたビンバッファ20内に挿入する。ビンに対する元の要求19に対する最終応答として、第1のビンがビンバッファ20から除去され、更に、この復号化済みのビン25がビンバッファ選択部18へ送られる。次のビン要求19が特定のビンバッファ20へ送られ、かつこのビンバッファが空でない場合には、次のビットがビンバッファ20から除去される。もしビンバッファが空であり、しかし接続されたビットバッファ58内のビット数 Mx が、関

10

20

30

40

50

連付けられた最大符号語長 L_x 以上である場合には、次の符号語がビットバッファから読み出され、新たなビン列が、第 1 のビットが除去されかつビンバッファ選択部へ送信されるビンバッファ内に挿入される。もしビットバッファが空であり、かつ接続されたビットバッファ 58 内のビット数 M_x が、関連付けられた最大符号語長 L_x よりも小さい場合には、 N_x 個のビットからなる次の列がグローバル・ビットバッファ 61 から読み出されかつ接続されたローカルビットバッファ 58 内へと挿入され、次の符号語がビットバッファから読み出され、新たなビン列がビンバッファ内に挿入され、更に、その列の第 1 のビンが除去されかつビンバッファ選択部へ送信される。この処理は、全てのソースシンボルが復号化されるまで繰り返しされる。

【0126】

データパケットの終点において、要求されたソースシンボルを復号化するために必要とされる以上のビン及び／又はビットが、ビンバッファ及び／又はビットバッファに挿入されてもよい。ビンバッファ内の残りのビンと、ビットバッファ内の残りのビットとは廃棄され、無視される。

【0127】

低遅延制約を有する固定長ビット列のインターリーピング

固定長のビット列のインターリーピングを有する P I P E 符号器及び復号器についての上述の実施形態は、上述した符号器バッファ遅延を制御するためのスキームと組み合わせることもできる。P I P E 符号化の概念は、上述した遅延制御を有する実施形態と同様である。もし遅延又は遅延の上限に関連した測度（以下参照）が、特定のしきい値を超えた場合には、第 1 のリザーブされたバッファエントリは、（データパケットの終点におけるものと同様のメカニズムを用いて）対応するビンバッファをフラッシュし、かつリザーブされた固定長のバッファエントリの全てのビットを満たすために追加的なビットを潜在的に書き込むことによって、満たされる。このようなメカニズムによって、関連付けられた遅延測度が特定のしきい値を下回るまで、待機するバッファエントリ数が削減される。復号器側においては、遅延制約を遵守するために符号器側で挿入されていたビン及びビットを廃棄する必要がある。このビン及びビットの廃棄については、基本的に、符号器側と同様のメカニズムを使用することができる。

【0128】

ある実施形態においては、遅延（又は遅延の上限）の測度は、グローバル・ビットバッファ内のアクティブなバッファエントリ内のビット数であり、ここで、アクティブなバッファエントリ数とは、リザーブされた固定長バッファエントリ数と、既書き込まれたビットを含む固定長のバッファエントリ数との合計である。第 1 のバッファエントリは、常にリザーブされた固定長のバッファエントリ又は空きのバッファエントリであることに留意されたい。なぜなら、もし第 1 のバッファエントリが、書き込まれたビットを含む場合には、これらのビットはビットストリームへ書き込まれるからである。（アプリケーションにより決定される）最大許容バッファ遅延を D 個のビットと仮定する。この最大バッファ遅延 D は、符号器と復号器との両方に知られている必要がある。本発明の好ましい実施形態においては、最大バッファ遅延 D はアプリケーションによって固定される。本発明の他の好ましい実施形態においては、最大バッファ遅延 D は、ビットストリームの内部で、即ち例えば、ビットストリームに含まれるデータパケットのヘッダ（又はスライスヘッダ）内又はパラメータ集合内で、信号化されている。最大バッファ遅延 D は、ビット若しくはバイトの単位、又は多数のビット若しくは多数のバイトの単位で信号化され得る。もしビン符号器 10 がグローバル・ビットバッファ 51 へ新たな固定長のビット列のリザーブに対する要求を送信する場合には、新たな固定長のバッファエントリがリザーブされる前に次のプロセスが実行される。

【0129】

もしグローバル・ビットバッファ内のアクティブなバッファエントリ内のビット数と、現在のリザーブ要求によってリザーブされるであろうビット数と、の合計が、最大バッファ遅延 D より大きい場合には、（リザーブされた）第 1 のバッファエントリは、グローバル

10

20

30

40

50

・ビットバッファ内のアクティブなバッファエントリ内のビット数と、現在のリザーブ要求によってリザーブされるであろうビットの数と、の合計が、最大バッファ遅延D以下になるまで、以下のプロセスによってフラッシュされる。リザーブされた固定長のバッファエントリのフラッシュは、データパケットの終点におけるフラッシュに類似している。即ち、対応する第1のバッファエントリをリザーブしたビットバッファ48に接続されたピン符号器10は、結果として得られるピン列が符号語と関連付けられたピン列を表すまで、接続されたピンバッファ8に対し、特定の又は任意の値を有するピンを追加することによってフラッシュされ、次に、符号語は対応するビットバッファ48へ挿入される。前述したように、ピンバッファにピンを追加するための1つの好ましい方法は、可能性のある最短の符号語を生成するピンを追加することである。接続されたビットバッファへの符号語の書き込みと、グローバル・ビットバッファへの固定長のビット列の潜在的な挿入との後で、もしビットバッファ内にビットがまだある場合（即ち、書き込まれた符号語が、リザーブされた固定長のビット列を完全には満たさなかった場合）には、全てのビットがビットバッファから除去され、かつリザーブされたバッファエントリに書き込まれるまで、特定の又は任意の値を有する更なるビットがビットバッファへ追加される。最終的に、このプロセスの最後では、完成されたバッファエントリ（グローバル・ビットバッファ内の第1の固定長エントリ）が、グローバル・ビットバッファから除去され、かつビットストリームへと書き込まれる。

【0130】

復号器側では、遅延制約を守るために追加されたピン及びビットを廃棄するために、同様のプロセスが実行される。そのため、復号器はグローバル・ビットバッファから読み出されたビットをカウントするカウンタCを保有する（このカウンタはグローバル・ビットバッファ内で保有されることができる）。カウンタCは、データパケットの復号化の開始時に（例えばゼロに）初期化され、固定長のビット列が読み出された後に増加される。もしNx個のビットからなる固定長の列がグローバル・ビットバッファ61から読み出される場合、カウンタCはNxだけ増加される。全体的なカウンタCに加え、各ビットバッファ58はカウンタCxを含み、このカウンタCxは、最後の固定長ビット列が対応するビットバッファ58内に読み込まれたときよりも前のビットカウンタCの値を記憶する。特定のビットバッファ58が新たな固定長ビット列を読み出す場合、そのカウンタCxは、第1のステップとしてCと等しく設定され、次に、固定長ビット列がグローバル・ビットバッファ61から読み出される。ピンに対する要求19が特定のピンバッファ20へ送信され、かつ、全体のカウンタCと、接続されたビットバッファ58のカウンタCxと、の差（ $C - C_x$ ）が、最大バッファ遅延Dより大きい場合には、その特定のピンバッファ20に現時点で記憶されている全てのピンと、接続されたビットバッファ58内に記憶されている全てのビットとが、破棄されかつ無視される。この追加のステップに加え、復号化は上述したように操作される。もしピンに対する要求19が送信されるべきピンバッファ20が、（全てのピンが既に除去されていたため、又はピン要求が受信された後に低遅延メカニズムが第1のステップにおいて全てのピンを破棄したため）空である場合には、接続されたピン復号器22は、接続されたビットバッファ58から新たな符号語を読み出すことを試みる。もしビットバッファ58内のビット数が最大符号語長より少ない場合には、グローバル・ビットバッファ61から新たな固定長ビット列が読み出され、符号語が読み出される等が実行される。

【0131】

図7～図13に関して上述した説明は、PIPE符号器104を一方とし、PIPE復号器202を他方とする間における、インターリーブされたビットストリームの経路を達成する可能性に関するものである。図1と図2に関連して説明したように、エントロピー符号化と復号化の装置は、2つの別個のチャンネルによって相互に接続されていてもよく、そのうちの1つはVLCビットストリーム112を運び、他の1つはインターリーブされたPIPE符号化済みビットストリームを運ぶ。しかし、VLCビットストリーム112とPIPE符号化済みビットストリーム118との両方をもインターリーブする可能性もあ

り、そのような可能性については、図 20～図 24 に関して後述する。しかしながら、その前に、P I P E 符号化についての数学的な背景と、確率区間をどのようにして最適にサブ分割し、結果として得られる個別の部分的区間を個別のエントロピー符号器 116 及びエントロピー復号器 210 にそれぞれ割り当てるとのかについての詳細と、を説明する。

【0132】

既に上述したように、P I P E 符号化においては、離散的なシンボルからなる入力列の事象スペースは、二値の確率区間の小さな集合へとマップされる。ソースシンボルについての確率モデリングは、固定型でも適応型でもよく、一方、確率区間を用いたエントロピー符号化は、固定のままであってモデリングの段階から切り離されている。確率区間の各々は、ハフマン符号の複雑さのレベルを持つ非常に単純なエントロピー符号を用いて符号化され得る。確率区間区分エントロピー (P I P E) 符号の超過率 (excess rate) は、純粋な算術符号化の場合と同様である。

【0133】

エントロピー符号化は、一般に、ロスレスデータ圧縮の最も一般的な形態と考えられ得る。ロスレス圧縮は、元のデータ表現が必要とするより少ないビット数を用いて、しかし情報を損なわずに、離散的データを表現することを目的としている。離散的データは、テキスト、図形、画像、ビデオ、オーディオ、スピーチ、ファクシミリ、医療用データ、気象データ、財務データの形態、又はデジタルデータの任意の他の形態において与えられ得る。多くの符号化アプリケーションにおいて、元のソースデータは、まず所謂符号化シンボルにマップされ、次に、これらの符号化シンボルはエントロピー符号化される。符号化シンボルへのマッピングは、量子化を含むことが可能であり、この場合、全体的な符号化スキームはロスが存在する。符号化シンボル s は、 M -ary ($M \geq 2$) のアルファベット $A = \{a_0, \dots, a_{M-1}\}$ の任意の値を取ることができる。このシンボル s を符号化するために、アルファベットは、推定された確率質量関数 (pmf) $\{p_s(a_0), \dots, p_s(a_{M-1})\}$ と関連付けられ、この pmf 内で考慮されない符号化シンボル間の全ての依存関係は無視される。これらの抽象的な設定に対し、エントロピー

【数 2】

$$H_s = -\sum_{i=0}^{M-1} p_s(a_i) \log_2 p_s(a_i) \quad (B1)$$

は、シンボル s を符号化するために、シンボルあたりのビットにおいて予想される符号語の長さに関する最大の下限であり、これはエントロピー符号化技術を用いて達成することができる。何十年もの間、ハフマン符号化と算術符号化が実用的なエントロピー符号化を支配してきた。これらは (ある意味で) エントロピーの限界を近似することが可能な実用符号の周知の例である。

【0134】

固定された確率分布については、ハフマン符号を構築することは比較的容易である。ハフマン符号の最も魅力的な特性は、それらの構成が可変長符号の (V L C) テーブルを使用することで効率的に実現できることである。しかしながら、時間に変化するソース統計を扱う場合には、即ち時間に変化するシンボル確率を扱う場合には、ハフマン符号とこれに対応する V L C テーブルとの適応は、アルゴリズムの複雑さと実装に掛かる費用との両方の観点において過大な要求を発生させてしまう。更に、 $p_s(a_i) > 0.5$ である優勢アルファベット値を有する場合には、(ランレングス符号化などの任意のアルファベット拡張子も使用しない) 対応するハフマン符号は、かなりの冗長度を持つ可能性がある。ハフマン符号の他の欠点は、より高次の確率モデリングを扱う場合に、V L C テーブルの多数の集合が必要となる可能性があるという事実である。

【0135】

一方、算術符号化は、実質的には L V C よりも複雑であるにも拘わらず、適応型および高次の確率モデリングを扱う場合やかなり偏りのある確率分布を扱う場合には、より一貫性

があり適切な処理を提供できるという利点がある。実際に、この特徴は基本的に、算術符号化が、少なくとも概念的には、確率推定のいかなる所与の値をも、多かれ少なかれ直接的な方法で、結果として得られる符号語の一部分へとマップ（写像）するためのメカニズムを提供する、という事実から生まれるものである。そのようなインターフェイスを備えることで、算術符号化は、一方では確率モデリング作業と確率推定作業との間の明確な分離を可能にし、他方では実際のエントロピー符号化、即ちシンボルから符号語へのマッピングを可能にしている。

【0136】

上述した従来型のエントロピー符号化のスキームとは異なり、P I P E符号化は確率区間のパーティションを使用する。その数学的背景について、以下により詳細に説明する。

10

【0137】

符号化シンボルの列 $\{s_0, \dots, s_{N-1}\}$ を考える。各シンボルは、アルファベット $s_i \in A_i$ から導出される。アルファベット $A = \{a^i_0, a^i_1, \dots\}$ は、それぞれが確率推定 $p_s(a^i_m)$ と関連付けられている2つ以上の文字を含む。確率推定 $p_s(a^i_m)$ は、符号器と復号器に知られており、固定型又は可変型でもよい。可変の確率が、符号器と復号器において同時に推定されると仮定する。アルファベット A_i は、シンボル列について同一であってもよく、又は、異なるシンボルタイプが異なるアルファベットと関連付けられてもよい。後者の場合には、復号器が列内の各シンボルのアルファベットを知っていると仮定される。この仮定は、実用的なソースのコーデックの記述がシンボルとそのアルファベットの順序を定めているシンタックスを含むことから、正当化される。

20

【0138】

シンボルの列 $\{s_0, \dots, s_{N-1}\}$ は、ビンとも呼ばれる二値シンボルの列に変換される。各シンボル s_i ごとに、二値化

【数3】

$$b^i = \{b^i_0 \dots\} = \gamma^i(s_i) \quad (B2)$$

は、アルファベット文字 a^i_m の、ビン b^i_m の順序付けられた集合への全単射マッピングを表している。二値化マッピング γ^i_b は、異なるシンボル s_i ごとに又はシンボルカテゴリーごとに異なってもよい。特定のシンボル s_i のための各ビン列 i は、1つ以上のビン b^i_k からなる。復号器側において、シンボル s_i は、ビン列 b^i が与えられている場合、逆マッピング $s_i = (\gamma^i_b)^{-1}(b^i)$ によって復元することができる。二値化の結果として、ソースシンボルの列 $\{s_0, \dots, s_{N-1}\}$ を表すビン列 $\{b_0, \dots, b_{B-1}\}$ が取得される。

30

【0139】

全てのビン b_j が同じ二値アルファベット $B = \{0, 1\}$ と関連付けられるが、しかし、対応するバイナリー $p m f \{p^j_0, p^j_1\}$ （但し、 $p^j_1 = p^j_0$ ）は通常は異なる。バイナリー $p m f \{p^j_0, p^j_1\}$ は、確率の低いビン（LPB）値 b^j_{LPB} と、その確率 p^j_{LPB} （但し、 $p^j_{LPB} \leq 0.5$ ）とによって記述することができる。このバイナリー確率記述 $\{b^j_{LPB}, p^j_{LPB}\}$ は、二値化マッピング γ^i_b が与えられた場合、シンボルアルファベットに対する確率推定 $p_s(a^i_m)$ から直接的に導出することができる。また、符号器側と復号器側において、 $\{b^j_{LPB}, p^j_{LPB}\}$ を同時に直接的に推定することも可能である（かつ多くの場合に好ましい）。従って、ビンは、シンタックスと、以前に符号化されたシンボル又はビンに基づく確率モデル（コンテキストとも呼ばれる）と、に対して関連付けることができる。更に、各確率モデルに関して、確率の記述 $\{b^j_{LPB}, p^j_{LPB}\}$ は、その確率モデルを用いて符号化されたビンの値に基づいて推定可能である。このようなバイナリー確率モデリングの例をH. 264のC A B A Cに関して説明する。

40

【0140】

バイナリーエントロピー関数

【数 4】

$$H(p) = -p \log_2(p) - (1-p) \log_2(1-p) \quad (B3)$$

は $p = 0.5$ 周辺で左右対称であり、同じバイナリー符号器が、 b^j_{LPB} の値とは無関係に、同一の L P B 確率 p^j_{LPB} と関連付けられた全てのビンを符号化するために使用され得る。従って、ビン列 $\{b_0, \dots, b_{B-1}\}$ は、符号化ビン列 $\{b^c_0, \dots, b^c_{B-1}\}$ へと変換される。各ビン b_j に対して、対応する全単射マッピング γ^j_b は、

$$b_j = (\gamma^j_c)^{-1}(b^c_j) = b^c_j \oplus b^j_{LPB} \quad 10$$

で特定される。ここで、

$\oplus$

は排他論理和 (exclusive or) の演算子を表す。復号器側において、符号化ビン b^c_j と対応する L P B 値 b^j_{LPB} が与えられている場合、ビン b_j は逆マッピング

【数 5】

$$b^c_j = \gamma^j_c(b_j) = b_j \oplus b^j_{LPB} \quad (B4)$$

によって復元され得る。符号化ビン $b^c_j = 0$ は、対応するビン b_j の値が L P B 値 b^j_{LPB} と等しいことを特定し、符号化ビン $b^c_j = 1$ は、対応するビン b_j の値が確率の高い値 (M P B) の値 $1 - b^j_{LPB}$ と等しいことを特定する。 20

【0 1 4 1】

符号化ビンの列 $\{b^c_0, \dots, b^c_{B-1}\}$ は、一意にソースシンボルの列 $\{s_0, \dots, s_{N-1}\}$ を表し、エントロピー符号化に使用可能な対応する確率推定は、L P B 確率 p^j_{LPB} (但し、 $p^j_{LPB} \leq 0.5$) によって完全に記述することができる。従って、符号化ビン b^c_j に関するバイナリーエントロピー符号器を設計するために、半开区間 $(0, 0.5)$ 内の確率だけを考慮すればよい。

【0 1 4 2】

実際のバイナリーエントロピー符号化については、符号化ビン列 $\{b^c_0, \dots, b^c_{B-1}\}$ が少数の確率区間 I_k へと投影される。L P B の確率区間 $(0, 0.5]$ は、K 個の区間 $I_k = (p_k, p_{k+1})$ に区分される。 30

【数 6】

$$\bigcup_{k=0}^{K-1} I_k = (0, 0.5] \quad \text{及び} \quad I_k \cap I_j = \emptyset \quad k \neq j \text{ のとき} \quad (B5)$$

【0 1 4 3】

K 個の区間の集合は、 $K - 1$ 個の区間の境界 p_k で特徴づけられ、但し $k = 1, \dots, K - 1$ である。一般性を失うことなく、 $k = 0, \dots, K$ であるときは、 $p_k < p_{k+1}$ であると仮定する。外側の区間の境界は固定され、 $p_0 = 0$ 及び $p_K = 0.5$ により与えられる。単純な非適応型のバイナリーエントロピー符号器が、各区間 I_k に設計される。関連づけられた L P B の確率 $p^j_{LPB} \in I_k$ を有する全ての符号化ビン b^c_j が区間 I_k に割り当てられ、かつ対応する固定エントロピー符号器によって符号化される。 40

【0 1 4 4】

以下の説明において、全てのビンは符号化ビン b^c_j を表し、全ての確率 p は L P B 確率 p^j_{LPB} である。

【0 1 4 5】

確率区間の離散化が符号化効率に与える影響を調べる際に、エントロピー境界を達成するある固定的な確率について最適なエントロピー符号器を設計できると仮定した。各確率区 50

間 $I_k = (p_k, p_{k+1}]$ は、代表的な確率 $p_{I_k} \in I_k$ と関連付けられており、対応する最適なエントロピー符号器は、この代表的な確率についてのエントロピー限界を達成するであろう。これらの仮定の下で、区間の代表値 p_{I_k} に関する最適なエントロピー符号器を用いて確率 p のピンを符号化するレートは、次式により与えられる。

【数 7】

$$\begin{aligned} R(p, p_{I_k}) &= p \log_2(p_{I_k}) - (1-p) \log_2(1-p_{I_k}) \\ &= H(p_{I_k}) + (p - p_{I_k}) H'(p_{I_k}) \end{aligned} \quad (B6)$$

10

ここで、 $H(p)$ はバイナリーエントロピー関数 3 を表し、

【数 8】

$$H'(p) = \log_2 \left(\frac{1-p}{p} \right) \quad (B7)$$

は、その最初の導関数である。更に、区間 $(0, 0.5)$ における確率分布が、 $f(p)$ 但し

$$\int_0^{0.5} f(p) dp = 1$$

20

により与えられることを前提とする。次に、対応する代表的な確率 $\{p_{I_k}\}$ を有する K 個の区間 $\{I_k\}$ の所与の集合に関し、各ピンについてビットで予想されるレートは次式で表される。

【数 9】

$$R = R(\{I_k\}, \{p_{I_k}\}) = \sum_{k=0}^{K-1} \left(\int_{p_k}^{p_{k+1}} R(p, p_{I_k}) f(p) dp \right) \quad (B8)$$

【0146】

30

$k = 0, \dots, K-1$ であるとき、任意の代表的な確率 p_{I_k} に関する第 1 の部分的導関数は、次式で表される。

【数 10】

$$\frac{\partial}{\partial p_{I_k}} R = \frac{p_{I_k} \int_{p_k}^{p_{k+1}} f(p) dp - \int_{p_k}^{p_{k+1}} p f(p) dp}{p_{I_k} (1 - p_{I_k}) \ln 2} \quad (B9)$$

【0147】

式

40

$$\frac{\partial}{\partial p_{I_k}} R = 0$$

は、定義 I_k のドメイン内側の代表的な確率 p_{I_k} に関し、単一の解

【数 11】

$$p_{I_k}^* = \frac{\int_{p_k}^{p_{k+1}} p f(p) dp}{\int_{p_k}^{p_{k+1}} f(p) dp} \quad (B10)$$

50

を有する。この解に関する第 2 の部分的導関数
【数 1 2】

$$\frac{\partial^2}{\partial p_{l_k}^2} R(p_{l_k}^*) = \frac{\int_{p_k}^{p_{k+1}} f(p) dp}{p_{l_k}^* (1 - p_{l_k}^*) \ln 2} \quad (B11)$$

は、
【数 1 3】

$$\int_{p_k}^{p_{k+1}} f(p) dp > 0 \quad (B12)$$

10

の場合には、常にゼロより大きい。

【0 1 4 8】

従って、条件 (B 1 2) が満たされた場合、式 (B 1 0) において与えられた値 $p_{l_k}^*$ は、区間の境界 p_k 及び p_{k+1} が与えられたとき、予想される全体のレート R を最小化させる区間 I_k に関する代表的な確率である。それ以外の場合には、区間 I_k に投影されるビン 1 つもなく、代表的確率 $p_{l_k} \in I_k$ は、全体のレート R に影響を全く与えずに、任意に選択され得る。しかし、区間 I_k がエントロピー符号化のために使用されないため、このような構成は回避すべきである。

20

【0 1 4 9】

最適な区間境界についての条件を見つけるため、 $k = 1, \dots, K - 1$ であるときに、区間境界 p_k に関して予想される全体のレート R の第 1 の導関数について検討する。もし全ての $p \in [p_{l_{k-1}}, p_{l_k}]$ について $f(p) > 0$ である場合には、次式

$$\frac{\partial}{\partial p_k} R = 0$$

は、定義 $[p_{l_{k-1}}, p_{l_k}]$ のドメイン内側の区間の境界 p_k に関して単一の解
【数 1 4】

30

$$p_k^* = \frac{H(p_{l_k}) - p_{l_k} H'(p_{l_k}) - H(p_{l_{k-1}}) + p_{l_{k-1}} H'(p_{l_{k-1}})}{H'(p_{l_{k-1}}) - H'(p_{l_k})} \quad (B13)$$

を有しており、この解に関する第 2 の部分的導関数

【数 1 5】

$$\frac{\partial^2}{\partial p_k^2} R(p_k^*) = f(p_k^*) (H'(p_{l_{k-1}}) - H'(p_{l_k})) \quad (B14)$$

40

は、ゼロより常に大きく、その結果、区間の代表値 $p_{l_{k-1}}$ 及び p_{l_k} が与えられたとき、 p_k^* は、予想される全体のレート R を最小化する区間境界 $p_k \in [p_{l_{k-1}}, p_{l_k}]$ である。もし、確率 $p \in [p_{l_{k-1}}, p_{l_k}]$ 但し $f(p) = 0$ が存在する場合には、式

$$\frac{\partial}{\partial p_k} R = 0$$

は複数の解を有するが、しかし、式 (B 1 3) で与えられる p_k^* は、更なる最適な解が存在する可能性はあるにしても、依然として最適である。

【0 1 5 0】

50

区間の個数 K と確率分布 $f(p)$ とが与えられたとき、予想される全体のレート R を最小化する、 $k = 1, \dots, K-1$ である区間境界 p_k と、 $k = 0, \dots, K-1$ である区間代表値 p_{1k} とは、 $k = 0, \dots, K-1$ については条件 (B12) を受ける式 (B10) と (B13) によって与えられる等式システムを解くことによって取得することができる。これは、以下の反復 (iterative) アルゴリズムによって達成することができる。

【0151】

アルゴリズム 1 :

- 1) 全ての $k = 0, \dots, K-1$ について条件 (B12) が守られる方法で、区間 $(0, 0.5)$ を K 個の任意の区間 $I_k = (p_k, p_{k+1})$ へと区分する。但し、 $p_0 = 0$ 、 $p_k = 0.5$ であり、全ての $k = 0, \dots, K-1$ について $p_k < p_{k+1}$ である。2) 式 (B10) に基づいて、 $k = 0, \dots, K-1$ である代表値 p_{1k} を更新する。
- 3) 式 (B13) に基づいて、 $k = 1, \dots, K-1$ である区間境界値 p_k を更新する。
- 4) 収束 (convergence) まで前の 2 つのステップを繰り返す。

【0152】

図 14 は、上記アルゴリズムを用いた最適な区間の離散化の例を示す。この例において、 $0 < p \leq 0.5$ については均一な確率分布 $f(p) = 2$ を仮定し、確率区間 $(0, 0.5]$ を $K = 4$ の区間に区分した。確率区間の離散化が、二値エントロピー関数 $H(p)$ のピース毎の線形近似 $A(p)$ をもたらしることが見て取れる。但し、全ての $p \in (0, 0.5]$ について、 $A(p) \geq H(p)$ である。

【0153】

区間の離散化が符号化効率に与える影響に関する測度として、エントロピー限界に相対して予想される全体的なレート増加率

【数16】

$$\bar{\rho} = \frac{R}{\int_0^{0.5} H(p) f(p) dp} - 1 \quad (\text{B15})$$

を使用することができる。図 14 に示す特定の例において、エントロピーの予想値

$$\bar{H} = \int_0^{0.5} H(p) f(p) dp$$

は、ピンあたり $1 / (2 \ln 2)$ ビットと等しく、レートオーバーヘッド

$$\bar{\rho}$$

は 1.01% と等しい。表 4 は、レートオーバーヘッド

$$\bar{\rho}_{uni} \text{ 及び } \bar{\rho}_{lin}$$

を表にしたものであり、これらは、選択された K 個の区間についての、均一な確率分布と、

$f(p) = 8p$ 但し $p \in (0, 0.5]$

のときの線形に増加する確率分布とをそれぞれ示す。

【0154】

10

20

30

40

【表 4】

【表4】レート・オーバーヘッド対均一な及び線形に増加する確率分布
に関する確率区間の数

K	1	2	4	8	12	16
$\bar{\rho}_{uni} [\%]$	12.47	3.67	1.01	0.27	0.12	0.07
$\bar{\rho}_{lin} [\%]$	5.68	1.77	0.50	0.14	0.06	0.04

10

【0155】

このセクションの研究では、固定された確率を有する少数の区間（例えば、8～10個の区間）へのLPBの確率区間（0, 0.5]の離散化が、符号化効率に与える影響は、極めて小さいことを示している。

【0156】

上述した確率区間に関するエントロピー符号化により、固定確率を用いた個別の符号器の使用が可能となる。

【0157】

以下においてはまず、単純な符号が固定確率に対してどのように設計され得るかを示す。これらの結果を得た上で、符号の設計とLPB確率区間（0, 0.5]のパーティションとを一緒に最適化するアルゴリズムを開発する。

20

【0158】

固定確率 $p = p_{ik}$ に関するエントロピー符号化は、算術符号化又は可変長符号化を使用して実行することができる。後者の場合には、以下の手法が単純かつ非常に効率的であるように見える。

【0159】

可変数のピンが可変長の符号語にマップされるバイナリーエントロピー符号化スキームについて考察する。一意の復号可能性を達成するために、符号語のピン列への逆マッピングは一意である必要がある。更に、エントロピー限界にできるだけ近い符号の設計を望むが故に、考慮対象を全単射マッピングに限定する。図15に示すように、このような全単射マッピングは、全てのリーフノードが符号語と関連付けられたバイナリーツリーによって表現され得る。ツリーエッジはバイナリー事象を表現する。図15の例においては、下方のエッジはLPBピン値を表し、上方のエッジはMPBピン値を表している。もしバイナリーツリーが完全なバイナリーツリーの場合には、即ち、全てのノードがリーフ(leaf)か又は2つの子系統(descendants)を持つ場合には、バイナリーツリーはピンに関する接頭語符号を表している。各リーフノードは、所与のLPBの確率 p に基づく確率と関連付けられる。ルートノードは確率 $p_{root} = 1$ を有している。他の全てのノードに関する確率は、対応する源流(ancestor)の確率に対し、LPBの末流については p を乗算し、MPBの末流については $q = 1 - p$ を乗算することによって得られる。各リーフノード L_1 は、ルートノードからリーフノードまでの、LPBエッジの数 a_1 と、MPBエッジの数 b_1 によって特徴づけられる。ある特定のLPBの確率 p について、リーフノード $L_1 = \{a_1, b_1\}$ に関する確率 p_1 は、次式と等しい。

30

40

【数17】

$$p_1 = p^{a_1} (1-p)^{b_1} \quad (B16)$$

【0160】

バイナリーツリー T は、リーフノードの数 L と、関連付けられたペア $\{a_1, b_1\}$ と、によって完全に特徴付けられ、但し $1 = 0, \dots, L-1$ である。

【0161】

50

完全なバイナリーツリー T と L P B の確率 p とが与えられた場合、リーフノードに対する符号語の最適な割当ては、ハフマン・アルゴリズムによって得ることができる。結果として得られる可変数のビットから可変長の符号語 (V 2 V) へのマッピング C は、リーフノードの数と同一である符号語の数 L と、 $l = 0, \dots, L-1$ に関するタプル $\{a_l, b_l, l_l\}$ とによって特徴付けられ、ここで、 l_l は、対応するリーフノード $L_l = \{a_l, b_l\}$ と関連付けられた符号語長を表す。符号語長 $\{l_l\}$ が与えられたとき、符号語の割り当てに関しては複数の可能性があり、かつ、符号語が一意に復号可能な接頭語符号を表している限りは、実際の符号語の割当ては重要ではないことに留意されたい。所与の符号 C と L P B の確率 p に関するビン毎のビットでの予想されるレート $R(p, C)$ は、符号語毎の予想される符号語長と予想されるビンの個数との比である。

10

【数 1 8】

$$R(p, C) = \frac{\sum_{l=0}^{L-1} p_l l_l}{\sum_{l=0}^{L-1} p_l (a_l + b_l)} = \frac{\sum_{l=0}^{L-1} p^{a_l} (1-p)^{b_l} l_l}{\sum_{l=0}^{L-1} p^{a_l} (1-p)^{b_l} (a_l + b_l)} \quad (B17)$$

【0 1 6 2】

大抵の場合、符号の設計は、符号語の最大数 L 、符号語毎のビンの最大数、又は最大符号語長などの要因によって制限されるか、又は、特定の構造の符号へと限定される（例えば、最適化された解析を可能にするために）。もし特定のアプリケーションのために利用可能な符号の集合 S_c が与えられたと仮定した場合には、特定の L P B 確率 p に関する最適な符号 $C^* \in S_c$ は、予想されるレート $R(p, C)$ を最小化することによって求め得る。

20

【数 1 9】

$$C^*(p) = \arg \min_{C \in S_c} R(p, C) \quad (B18)$$

【0 1 6 3】

他の迅速な代替例として、最小化はバイナリーツリー S_T の所与のセットに関して進めることができ、各ツリーについて、ハフマン・アルゴリズムによって得られる V 2 V コード C が 1 つだけ考慮される。一例として、リーフノードの数 L が所与の最大数 L_m 以下である全てのバイナリーツリー T を考慮することで、様々な L P B の確率 p についての V 2 V コードを設計した。図 1 6 においては、相対レートの増加率

30

$$\rho(p, C^*(p)) = R(p, C^*(p)) / H(p)$$

が、選択された最大テーブルサイズ L_m に関する L P B の確率 p にわたってプロットされる。レート増加率 $\rho(p)$ は、通常、より大きなテーブルサイズを可能にすることによって減少され得る。より大きな L P B 確率に関しては、レート増加率 $\rho(p)$ を合理的に小さく保持するために、通常は 8 ~ 16 個の符号語から成る小さなテーブルサイズ L で十分であるが、より小さい L P B 確率（例えば $p < 0.1$ ）に関しては、より大きなテーブルサイズ L が必要とされる。

40

【0 1 6 4】

前のセクションで、固定型の L P B 確率のための最適な符号とコード設計を想定して、最適な確率の離散化を検討した。しかし、一般に、限定されたテーブルサイズの実際の V 2 V コードを持つエントロピー限界を達成することはできないので、最適化されたエントロピー符号化設計を得るためには、コード設計と L P B 確率区間 $(0, 0.5]$ のパーティションとを一緒に考慮する必要がある。

【0 1 6 5】

所与の区間 $I_k = (p_k, p_{k+1})$ に関し、所与の集合 S_c の符号 C_k が、この所与の区間について予想されるレート

50

$$R = \int_{p_k}^{p_{k+1}} R(p, C_k) f(p) dp$$

を最小化する場合には、次式に示す最適な符号 C_k^* である。

【数 2 0】

$$C_k^* = \arg \min_{\forall C_k \in S_C} \int_{p_k}^{p_{k+1}} R(p, C_k) f(p) dp \quad (B19)$$

【0 1 6 6】

10

実用的な設計のために、式 (B 1 9) における積分の最小化は、まず式 (B 1 0) に基づいて、区間 I_k に関する最適な代表的確率 p_{1k}^* を決定し、次に式 (B 1 8) に基づいて、代表的確率 p_{1k}^* のための所与の集合 S_C の最適な符号 C_k^* を選択することによって、符号化効率にほとんど影響を与えずに、簡略化することができる。

【0 1 6 7】

$k = 0, \dots, K-1$ である符号 C_k の集合が与えられたとき、 $k = 1, \dots, K-1$ である最適な区間境界値 p_k は、予想される全体的なレート

【数 2 1】

$$R = R(\{p_k\}, \{C_k\}) = \sum_{k=0}^{K-1} \left(\int_{p_k}^{p_{k+1}} R(p, C_k) f(p) dp \right) \quad (B20) \quad 20$$

を最小化することにより導出され得る。

【0 1 6 8】

区間境界に関する第 1 の導関数をゼロに設定すると、 $k = 1, \dots, K-1$ である場合の式

$$\frac{\partial}{\partial p_k} R = 0$$

30

は、

【数 2 2】

$$p_k^* = p_k \quad \text{with} \quad R(p_k, C_{k-1}) = R(p_k, C_k) \quad (B21)$$

をもたらす。

【0 1 6 9】

式 (B 1 3) と同様に、 p_k^* は常に最適な解であることが分るが、しかし、確率分布 $f(p)$ によっては、更なる最適な解が存在する可能性がある。従って、所与の関連付けられた符号 C_{k-1} と C_k とをそれぞれ有する、2 つの区間 I_{k-1} と I_k との間の最適な区間境界 p_k^* は、関数 $R(p, C_{k-1})$ と $R(p, C_k)$ との交点である。 40

【0 1 7 0】

その結果、確率区間の数 K 、可能な符号の集合 S_C 、及び確率分布 $f(p)$ 但し $p \in (0, 0.5]$ が与えられた場合には、以下の対話型アルゴリズムを、確率区間のパーティションと関連付けられた符号とを一緒に導出するために使用することができる。

【0 1 7 1】

アルゴリズム 2 :

1) セクション 3 で特定されたアルゴリズム 1 を使用して、 $k = 0, \dots, K$ であるときの初期の確率区間境界値 p_k を求める。

2) 式 (B 1 0) に基づいて、 $k = 0, \dots, K-1$ であるときの確率区間 I_k に関す 50

る代表値 p_{ik} を求める。

3) 式 (B 1 8) に基づいて、 $k = 0, \dots, K-1$ であるときの区間の代表値 p_{ik} に関する符号 $C_k \in S_c$ を求める。

4) 式 (B 2 1) に基づいて、 $k = 1, \dots, K-1$ であるときの区間境界値 p_k を更新する。

5) 収束するまで前の 3 ステップを繰り返す。

【0172】

アルゴリズム 2 におけるステップ 2 及び 3 は、式 (B 1 9) に従って、 $k = 0, \dots, K$ であるときの区間境界 p_k に基づいた、 $k = 0, \dots, K-1$ であるときの符号 $C_k \in S_c$ の直接的な導出によって置き換えることができる。更に、セクション 4. 1 で上述したように、ステップ 3 における最小化は、各バイナリツリー T について、ハフマン・アルゴリズムによって得られる $V2V$ コード C_k が 1 つだけ考慮されるような、バイナリツリー S_T の所与の集合にわたって進めることもできる。

【0173】

一例として、アルゴリズム 2 を使用して、 $K = 12$ の確率区間への区分と対応する $V2V$ コードとを一緒に導出した。このとき、ステップ 3 におけるアルゴリズムの最小化は、各ツリー T に関する評価された符号 C がハフマン・アルゴリズムによって取得された、バイナリツリー S_T の所与の集合にわたる等価の最小化と置き換えた。最大数 $L_m = 65$ 個のリーフノードを有するツリー T 、つまり、最大 65 個までのテーブルエントリを有する符号 C について考察した。最大 16 個までのリーフノードを有する全てのバイナリツリー T は、最小化においてこれまでも評価されてきたが、ここでは、16 個を超えるリーフノードをもつツリーに関して、より少数のリーフノードを有するツリーに対して最良の結果をもたらす次善のサーチを採用した。

【0174】

図 17 においては、コードの設計例のためのエントロピー限界

$$\Delta R(p) = R(p) - H(p)$$

との相対関係において予想されるレートの増加が、LPB の確率 p にわたってプロットされている。比較として、(セクション 3 で展開されるように) 理論的に最適な確率区間の離散化に関する予想されるレートの増加 ΔR と、更なる制約 $p_{IK-1} = 0.5$ を有する理論的に最適な確率の離散化と、をもこの図の中にプロットした。確率区間の離散化と $V2V$ コード設計とをまとめることで、区間の境界線のシフトにつながる ($k = 1, \dots, K-1$ である区間境界 p_k が $\Delta R(p)$ 曲線の極大値によって与えられる) ことが分かる。均一な確率分布 $f(p)$ を仮定した場合、実際の $V2V$ コードを有する設計例に関するエントロピー限界との関係において相対的に予想される全体的レートの増加率は、

$$\bar{\rho} = 0.24\%$$

である。理論的に最適な確率区間の離散化と、更なる制約 $p_{IK-1} = 0.5$ を有する理論的に最適な確率の離散化のそれぞれに対応する相対的なレート増加率はそれぞれ、

$$\bar{\rho} = 0.12\%$$

と

$$\bar{\rho} = 0.13\%$$

である。

【0175】

符号語の終了は以下のように実行できる。有限のシンボル列 $\{s_0, \dots, s_{N-1}\}$ を符号化する時、 K 個のバイナリ符号器の各々は、有限の符号化ビンの列を $b^c_k = \{b^c_0, \dots, b^c_{BK-1}\}_k$ 処理する。ここで、 $k = 0, \dots, K-1$ である。また、 K 個のバイナリ符号器の各々について、符号語又は符号語の列 $c_k (b^c_k)$ が付えられた場合、列 $b^c_k = \{b^c_0, \dots, b^c_{BK-1}\}_k$ の全ての符号化ビンが復元可能であることが確認さ

れてきた。

【0176】

算術符号化を採用する場合、符号化ビンの列のための算術符号語は、符号語を与えられた場合に全ての符号化ビンが復号化され得るような方法で終了されなければならない。上記のV2Vコードについては、列 b_k の終点にあるビンは、符号語に関連付けられたビン列を表していない可能性がある。このような場合、その残りのビン列を接頭語として含む任意の符号語が書き込まれ得る。もし最短の長さを持つ対応する符号語（又はこれらの符号語の1つ）が選ばれた場合には、オーバーヘッドを最小化することができる。復号器側においては、ビットストリーム・シンタックスと二値化スキームとによって識別可能である、ビン列の終点において追加的に読み出されたビンは、廃棄される。

10

【0177】

単純なコード設計の例を以下に提供する。図示する目的で、3文字と、 $p_s(a_0) = 0.7$ 、 $p_s(a_1) = 0.18$ 、及び $p_s(a_2) = 0.12$ の固定された関連する確率とを有するソース $\{s\}$ の簡単な例について考察する。図18に示すように、対応する3項の選択ツリーは完全なバイナリツリーに変換されることができる。

【0178】

図18に示す完全なバイナリツリーへの二値化を表5に示す。3項シンボルのpmfである p_s は、2つのバイナリpmf、即ち $p_{b_0} = (0.7, 0.3)$ と $p_{b_1} = (0.6, 0.4)$ とに変換される。ビットストリーム内の各シンボル s について、ビン b_0 が存在している。ビン b_0 が0と等しい場合には、 b_1 も存在している。表2で示した二値化は、ソース s についての最適な単一文字のハフマン符号と同じであることに留意されたい。

20

【0179】

【表5】

[表5]3文字ソースの二値化。LPB確率 p_{LPB} は、第1のビンについては0.3であり、第2のビンについては0.4である。

シンボル a_i	確率 $p(a_i)$	ビン b_0	ビン b_1
a_0	0.7	1	
a_1	0.18	0	1
a_2	0.12	0	0
LPB 確率	$p_{LPB} = p(b_j = 0)$	0.3	0.4

30

【0180】

ソース s のエントロピーは、次式で与えられる。

【数23】

$$\begin{aligned}
 H(0.7, 0.18, 0.12) &= H(0.7, 0.3) + 0.3 H(0.6, 0.4) \\
 &= 1.1726 \text{ ビット/シンボル}
 \end{aligned}
 \tag{B22}$$

40

【0181】

単一文字のハフマン符号の平均符号語長は、次式で与えられる。

【数 2 4】

$$\bar{\ell}_{HC} = \sum_{i=0}^{M-1} p_i \ell_i^{HC} = 1.3 \text{ ビット/シンボル} \quad (\text{B23})$$

これは、 $\rho_{HC} = 0.1274$ ビット／シンボルの冗長度、又は 10.87% の予想されるレートオーバーヘッドに相当する。

【0 1 8 2】

固定された p_{mf} を有する特定の二値化の例については、ビン b_0 と b_1 とが、既に符号化ビンを表している。なぜなら、両ビンについて LPB の値 b_{LPB}^j が 0 と等しいからである。LPB の確率の分布 $f(s)$ は離散的であり、ここで、 $p = 0.3$ と $p = 0.4$ の場合を除いて、 $f(p) = 0$ である。その結果、最適な確率離散化は $K = 2$ の区間をもたらし、ここで、代表値は $p_{I0} = 0.3$ 及び $p_{I1} = 0.4$ である。これらの区間の間の区間境界 p_1 は、 $[0.3, 0.4)$ において任意に選択することができる。

【0 1 8 3】

ソースを符号化するために、ソースシンボルの列はビン列へと二値化される。ビン b_0 は、全てのソースシンボルについて送信される。ビン b_1 は、 $b_0 = 0$ の場合に限って送信される。ビン b_0 と b_1 は、一定の LPB 確率 $p_{I0} = 0.3$ と $p_{I1} = 0.4$ とを用いて、それぞれ個別に符号化される。

【0 1 8 4】

固定型の確率を有するバイナリーアルファベットの効率的な符号化は、単純な V2V のマッピングによって達成することができる。LPB の確率 $p_{LPB} = 0.3$ と $p_{LPB} = 0.4$ に関する小さな符号化テーブルを用いた V2V マッピングの例を、表 6 と表 7 にそれぞれ示す。 $p_{LPB} = 0.3$ に関する V2V マッピングは、 0.0069 ビット／ビン、又は 0.788% の冗長度を生じさせる。LPB の確率 $p_{LPB} = 0.4$ に関しては、冗長度は 0.0053 ビット／ビン、又は 0.548% である。

【0 1 8 5】

【表 6】

[表6] $p_{LPB} = 0.3$ の LPB 確率に関するビンツリー及び符号。
この符号の冗長度は 0.788% である。

ビンツリー	確率	符号
'11'	$0.7^2 = 0.49$	'1'
'01'	$0.7 \cdot 0.3 = 0.21$	'01'
'0'	0.3	'00'

【0 1 8 6】

【表 7】

【表7】 $p_{LPB}=0.4$ のLPB確率に関するビンツリー及び符号。
この符号の冗長度は0.548%である。

ビンツリー	確率	符号
'111'	$0.6^3 = 0.216$	'11'
'110'	$0.6^2 \cdot 0.4 = 0.144$	'001'
'10'	$0.6 \cdot 0.4 = 0.24$	'11'
'01'	$0.4 \cdot 0.6 = 0.24$	'01'
'00'	$0.4^2 = 0.16$	'000'

10

【0187】

新たな符号化方法によってもたらされる全体的な予想されるレートは、次式で示される。

【数25】

$$\bar{\ell}_{NC} = \bar{\ell}_{b_0} + 0.3 \cdot \bar{\ell}_{b_1}$$

20

$$= 1.181 \text{ bit/symbol}$$

(B24)

【0188】

全体の冗長度はエントロピー限界に対して0.73%であり、これは単一文字のハフマン符号と比較して、有意に改善された値を示している。

【0189】

同様の符号化効率の改良は、ランレングス符号を作成することによっても達成できる、と主張することもできる。上記の例において、最大2つのシンボルまでのランを考慮することによって、最大確率シンボルに対するランレングス符号を構築することができる。各々の事象の $\{a_0 a_0, a_0 a_1, a_0 a_2, a_1 a_2\}$ は、別個の符号語に関連付けられ得る。このような符号は、エントロピー限界との相対関係において冗長度1.34%をもたらす。実際に、V2Vコードは、ランレングス符号のバイナリーシンボルへの一般化としてみなされ得る（表3のV2Vコードは、ランレングス符号を効果的に表している）。固定の確率を有する単一のシンボルアルファベットの場合には、提示した手法と同様の符号化効率を、可変長符号語に対して可変数のソースシンボルをマップする符号を作成することによっても達成することができる。この提示した手法の主な利点は、固定型又は適合型の確率推定を有する任意のソースシンボル列を、固定型のLPB確率を用いて作動する少数の単純なバイナリー符号器へとマッピングする際の、その柔軟性である。

30

【0190】

次に、一意の復号可能性を達成する方法を考察する。

40

【0191】

提示したエントロピー符号化方式において、ソースシンボル列 $s = \{s_0, \dots, s_{N-1}\}$ の符号化は、以下の3つの基本的なステップから成る。

- ・シンボル二値化 $b = \{b_0, \dots, b_{B-1}\} = \gamma_b(s)$ がビン列 $b = \{b_0, \dots, b_{B-1}\}$ をもたらすステップ
- ・ビン列を符号化ビン列 $b^c = \{b^c_0, \dots, b^c_{B-1}\} = \gamma_c(b)$ へと変換するステップ
- ・確率区間の離散化とK個の固定型バイナリー符号器を用いて、符号化ビン列 $b^c = \{b^c_0, \dots, b^c_{B-1}\}$ をバイナリーエントロピー符号化するステップ

50

【0192】

もし符号化ビン列 $b^c = \{b^c_0, \dots, b^c_{B-1}\}$ が一意に復号可能であり、 γ_b と γ_c とのマッピングが可逆である場合には、シンボル列 $s = \{s_0, \dots, s_{N-1}\}$ は一意に復号可能である。

【0193】

1つ以上の符号化ビンの列 $b^c = \{b^c_0, \dots\}$ の1つ以上の符号語 $c(b^c) = \{c_0, \dots\}$ の列への符号器マッピングを、 γ_e で示すことにする。

【数26】

$$c(b^c) = \gamma_e(b^c) \quad (B25)$$

10

【0194】

符号語 $c(b^c)$ の列が付けられているとき、符号化ビン列 b^c が一意的に復号可能となるためには、符号器マッピング γ_e は、一意の符号語 $c(b^c)$ が各々の可能な符号化ビン列 b^c に割り当てられるという特性を有する必要がある。

【数27】

$$\forall b_i^c, b_j^c \quad b_i^c \neq b_j^c \rightarrow c(b_i^c) \neq c(b_j^c) \quad (B26)$$

【0195】

20

この特性は、算術符号又は接頭語の符号が使用されている場合、常に満たされる。特に、セクション4.1に説明されているV2Vコードについて満たされる。なぜなら、V2Vコードは、可変数のビンに関する接頭語の符号を表しているからである（セクション4.3において説明した符号語の終了を含む）。

【0196】

しかしながら、提示したエントロピー符号化のアプローチにおいては、符号化ビンの列 b^c は $k = 0, \dots, K-1$ である K 個のサブシーケンス b_k^c に分割され、

【数28】

$$\{b_0^c, \dots, b_{K-1}^c\} = \gamma_p(b^c) \quad (B27)$$

30

各サブシーケンス b_k^c に対して、符号語 $c_k(b_k^c)$ の列が、特定の符号器マッピング γ_e^k を使用して割り当てられる。その結果、一意的な復号の可能性の条件は、拡張されなければならない。もし符号化ビン b_k^c の各サブシーケンスが、対応する符号語 $c_k(b_k^c)$ が与えられると一意に復号可能であり、かつ区分規則 γ_p がその復号器に知られている場合には、符号化ビン b^c の列は、符号語 $c_k(b_k^c)$ の K 個の列が与えられると一意に復号可能となり、ここで、 $k = 0, \dots, K-1$ である。区分規則 γ_p は、LPB確率区間の離散化 $\{I_k\}$ と、 $j = 0, \dots, B-1$ である符号化ビン b_j^c に関連付けられたLPB確率 p^j_{LPB} と、によって付与される。従って、LPB確率区間の離散化 $\{I_k\}$ は、復号器側で知られている必要があり、 $j = 0, \dots, B-1$ である各符号化ビン b_j^c についてのLPB確率 p^j_{LPB} は、符号器側と復号器側とにおいて同一の方法で導出される必要がある。

40

【0197】

ビン列の符号化ビン列 b^c へのマッピング γ_c のために、 $j = 0, \dots, B-1$ である各単一の b_j がバイナリーマッピング

$$b_j^c = \gamma_c^j(b_j) = b_j \oplus b^j_{LPB}$$

によって変換される。復号器側では、ビン列は $j = 0, \dots, B-1$ であるバイナリーマッピング

50

【数 2 9】

$$b_j = (\gamma_c^j)^{-1}(b_j^c) = b_j^c \oplus b_{LPB}^j \quad (B28)$$

によって求めることができる。もし各ビン b_j についての L P B 値 b_{LPB}^j が符号器と復号器側とにおいて同一の方法で導出される場合には、

【数 3 0】

$$b_j^c \oplus b_{LPB}^j = b_j \oplus b_{LPB}^j \oplus b_{LPB}^j = b_j \oplus 0 = b_j \quad (B29)$$

10

であるため、これらのマッピング $(\gamma_c^j)^{-1}$ は、対応する符号器マッピング γ_c^j の逆を表しており、従って、ビン列 b の符号化ビン列 b^c への変換 γ_b は可逆である。

【0 1 9 8】

最後に、 $i = 0, \dots, N-1$ のときの各シンボル s_i がビン列 $b_j = \gamma_b^j(s_i)$ へとマップされるような二値化 $b = \gamma_b(s)$ の可逆性を調べる。もし二値化マッピング γ_b^j が、シンボル s_i に関するアルファベット A_i の各文字 a_m^i に対して異なるビン列 b_m^j を割り当てる場合には、対応するビン列 b^i が与えられると、シンボル s_i は一意に復号化することができる。しかしながら、ビン列 $b = \{b_0, \dots, b_{B-1}\}$ の、 $i = 0, \dots, N-1$ であるシンボル s_i に対応するビン列 b^i への区分が復号器に知られていないため、この条件は不十分である。十分な条件は、各シンボル s_i について、対応するアルファベット A_i の文字 a_m^i に関連付けられているビン列 b_m^j が接頭語符号を形成し、かつ $i = 0, \dots, N-1$ であるシンボル各 s_i について、二値化マッピング γ_b^j が復号器側に知られているときに与えられる。

20

【0 1 9 9】

提示したエントロピー符号化の手法において一意的な復号を可能にする条件は、次のようにまとめることができる。

- ・二値化マッピング γ_b^j が接頭語符号を表し、かつ（シンボル符号化の順序で）復号器に知られる。
- ・全てのビン b_j に関する確率モデル (b_{LPB}^j, p_{LPB}^j) が、符号器側と復号器側とにおいて同じ方法で導出される。
- ・L P B 確率区間 $(0, 0.5]$ の $k = 0, \dots, K-1$ である K 個の区間 I_k への区分が復号器に知られる。
- ・ $k = 0, \dots, K-1$ である各確率区間 I_k に関するマッピング γ_k^e が一意に復号可能な符号を表す。

30

【0 2 0 0】

以下に、全体的な符号器と復号器の設計例をより詳細に説明する。ビンに関する確率モデル $\{b_{LPB}, p_{LPB}\}$ が符号器側と復号器側とにおいて直接的に推定され、 K 個のバイナリー符号器が上述した V 2 V マッピングを使用するような、符号化スキームに焦点を当てる。各ソースシンボル s はシンボルカテゴリー c_s と関連付けられ、そのシンボルカテゴリー c_s はその値の範囲を含むシンボルのタイプを決定する。シンボルの順序と関連するシンボルカテゴリーとは、符号器側と復号器側とにおいて既知であると推定されるシンタックスによって与えられる。

40

【0 2 0 1】

P I P E 符号器と P I P E 復号器との設計例のブロック図を図 1 9 に示す。符号器側において、関連付けられたシンボルカテゴリー c_s を有するシンボル s はバイナライザへ供給され、このバイナライザは各シンボル s をビン列 $b_s = \gamma^{c_s}_b(s)$ へと変換する。

【0 2 0 2】

$\gamma^{c_s}_b$ はシンボルカテゴリー c_s に基づいて決定される。更に、バイナライザは、ビン列 b_s の各ビン b を、このビン b を符号化するために使用される確率モデルを指定する確率モデル指示 c_b と関連付ける。確率モデル指示 c_b は、シンボルカテゴリー c_s 、ビン列 b_s 内

50

部の現在のビンのビン数、及び／又は既に符号化済みのビン及びシンボルの値に基づいて、導出され得る。

【0203】

確率推定及び割当部は、値の対 $\{b_{LPB}, p_{LPB}\}$ によって特徴付けられた多数の確率モデルを保有している。この確率推定及び割当部は、ビン b とその関連付けられた確率モデル指示 c_b とをバイナライザから受信し、LPB 値 b_{LPB} と指示された確率モデルの LPB 確率 p_{LPB} とを、符号化ビン導出部と確率量子化部とにそれぞれ送信する。その後、対応する確率モデル $\{b_{LPB}, p_{LPB}\}$ は、受信されたビン b の値を用いて更新される。

【0204】

符号化ビン導出部は、バイナライザと確率推定及び割当部とから、ビン b と関連付けられた LPB 値 b_{LPB} とをそれぞれ受け取り、

$$b^c = b \oplus b_{LPB}$$

によって導出される符号化ビン b^c を確率量子化部へと送信する。この確率量子化部は、各符号化ビン b^c を K 個のバイナリー符号器の 1 つへ転送する。確率量子化部は、LPB 確率区間の量子化 $\{I_k\}$ に関する情報を含む。符号化ビン b^c に関連付けられかつ確率推定及び割当部から受信された LPB 確率 p_{LPB} は、区間境界 $\{p_k\}$ と確率区間インデックス k とに対して比較され、この確率区間インデックス k に関しては $p_{LPB} \in I_k$ が導出される。次に、符号化ビン b^c は関連するバイナリー符号器へと転送される。

【0205】

K 個のバイナリー符号器の各々はビンバッファ及びビン符号器から成る。ビンバッファは、確率量子化部から符号化ビン b^c を受け取り、これらを符号化の順序で記憶する。ビン符号器は、特定の $V2V$ マッピングを構成し、ビンバッファ内のビン列を符号語に関連付けられたビン列と比較する。ビンバッファ内のビン列がこれらのビン列の 1 つと等しい場合には、ビン符号器は、このビン列 $\{b^c\}$ をビンバッファから取り除き、関連する符号語 ($\{b^c\}$) を対応する符号語ストリームに書き込む。シンボル列に関する符号化処理の終了時に、セクション 4.3 で説明したように、関連するビンバッファが空でない全てのバイナリー符号器において、終端符号語が書き込まれる。

【0206】

結果として得られた K 個の符号語ストリームは、個別に送信され、パケット化され、若しくは記憶され、又は送信若しくは記憶を目的としてインターリーブされることもできる（セクション 6.2 参照）。

【0207】

復号器側では、ビン復号器とビンバッファから成る K 個のバイナリー復号器の各々は、1 つの符号語のストリームを受信する。ビン復号器は、符号語のストリームから符号語 ($\{b^c\}$) を読み出し、関連するビン列 $\{b^c\}$ を符号化の順序でビンバッファ内に挿入する。

【0208】

シンボル列の復号化は、基底にあるシンタックスによって駆動される。シンボル s に対する要求は、シンボルカテゴリー c_s と一緒にバイナライザへ送信される。バイナライザは、これらのシンボル要求をビンに対する要求に変換する。ビンに対する要求は、符号器内と同一の方法で導出される確率モデル指示 c_b と関連付けられ、確率推定及び割当部へと送信される。確率推定及び割当部は、符号器側におけるその等価物と同様に操作される。確率推定及び割当部は、確率モデル表示 c_b に基づいて確率モデルを識別し、その LPB 値 b_{LPB} と LPB 確率 p_{LPB} とをビン導出部と確率量子化部のそれぞれへと送信する。

【0209】

確率量子化部は、バイナリー符号器が符号器側で決定されるのと同じ方法で、LPB 確率 p_{LPB} に基づいて K 個のバイナリー復号器の 1 つを決定し、対応するビンバッファから符号化の順序で第 1 の符号化ビン b^c を取り除き、それをビン導出部へと転送する。ビン導出部は、符号化ビン b^c と関連する LPB 値 b_{LPB} とを、確率量子化部と確率推定及び割当

部とのそれぞれから受け取り、ビン値

$$b = b^c \oplus b_{LPB}$$

を求める。バイナライザから送信されたビン要求への最終応答として、ビン導出部は、復号化されたビン値 b をバイナライザと確率推定及び割当部とに送信する。

【0210】

確率推定及び割当部において、復号化されたビン b の値は、符号器側と同じ方法で、関連する値 c_b によって選択された確率モデル $\{b_{LPB}, p_{LPB}\}$ を更新するために使用される。最後に、バイナライザは、受信されたビン b をシンボル要求に関して既に受信されているビン列 b_s に加え、このビン列 b_s を二値化スキーム γ^{cs}_b によってシンボル値と関連付けられたビン列と比較する。そのビン列 b_s がこれらのビン列の1つと一致する場合には、対応する復号化されたシンボル s がシンボル要求への最終応答として出力される。それ以外の場合には、バイナライザはシンボル s が復号化されるまで更なるビン要求を送る。

【0211】

シンタックスによって駆動される更なるシンボル要求が受信されなくなった場合には、シンボル列の復号化は終了する。エントロピー復号化処理の終点において（終端符号語の結果として）、ビンバッファ内に含まれている可能性のある符号化ビン b_c は廃棄される。

【0212】

図1と図2及び図3～図13を参照しながらPIPE符号器及びPIPE復号器について幾つかの実施形態を説明し、また、図14～図19を参照しながらPIPE符号化についての一般的な数学的背景を提供した後で、次に、図20～図24を参照しながらエントロピー符号化装置及び復号化装置に関するより詳細な実施形態について説明する。以下に図22～図24に示す実施形態は、PIPE符号化されたビットストリーム同士間で互いにインターリーブするだけではなく、VLCビットストリームとPIPE符号化されたビットストリームとを全部一緒にインターリーブする。これと比較して、PIPE符号器とPIPE復号器についての実施形態であって、PIPE符号化されたビットストリームのためのインターリーブを有する実施形態、即ち図7～図13に示す実施形態では、PIPE符号化されたビットストリームの個別のインターリーブだけが提示されていた。既に上述したように、これらの実施形態であっても、もう一つのインターリーバー／デ・インターリーバーのペア134及び232をそれぞれ使用することで（図1及び図2を参照）、例えば図5及び図6に示すビットストリーム・インターリーブングなどを用いて、完全にインターリーブされたビットストリームを達成するための基礎として役割を果たすことができる。しかし、次に示す実施形態は、VLCビットストリームとPIPE符号化されたビットストリームとの両方に対して、一段階のインターリーバー128／デ・インターリーバー230をそれぞれ使用して、一度にインターリーブングを実行する。

【0213】

複雑性と符号化効率との間のより適切な妥協点を達成するために、VLC及びPIPE符号化されたシンボルがビットストリームの内部でインターリーブされる実施形態を詳細に説明する前に、インターリーブングがない基礎的な構造について図20と図21とを参照しながら説明する。

【0214】

図1aのエントロピー符号化装置の構造を図20に示す。このエントロピー符号化装置は、ソースシンボルのストリーム1a、即ちVLC符号化されたソースシンボルとPIPE符号化されたソースシンボルとの組合せであって、図1の106と図2の218とにそれぞれ対応するストリームを、2つ以上の部分的ビットストリーム12、12aからなるセットへと変換し、ここで、ビットストリーム12aは、図1及び図2のビットストリーム112及び206に対応する。

【0215】

既に上述したように、各ソースシンボル1aは、図1内のVLC符号器102に対応するVLC符号器22a内で標準的なVLC符号を使用して符号化されるのかどうか、又は、

P I P E符号化の概念を用いて符号化されるべきかどうかについて特定する、ある指示に関連づけて有している。図1及び図2に関して既に上述したように、この指示は復号化側に対して明確には送信されない可能性もある。寧ろ、その関連付けられた指示は、ソースシンボル自身のタイプ又はカテゴリーから生じる可能性もある。

【0216】

V L C符号化されたシンボル1 bは標準的なV L C符号を用いて符号化されるが、そのV L C符号は、V L C符号器2 2 aを使用して、上述のシンボルカテゴリーまたはシンボルタイプに基づいて符号化されてもよい。対応する符号語1 1 aは、独特な部分的ビットストリーム1 2 aに書き込まれる。非V L C符号シンボル1は、例えば図1及び図3に関して上述したようなP I P E符号化を使用して符号化され、その結果、多数の部分的ビットストリーム1 2が取得される。ソースシンボル1 aの幾つかは、図1 aに関して既に上述したように、二値化が2つの部分から構成される方法で既に二値化されていてもよい。これらの部分のうちの1つは、P I P Eの手法を用いて符号化され、対応する部分的ビットストリーム1 2へと書き込まれてもよい。ビン列の他の部分は、標準的なV L C符号を用いて符号化され、対応する部分的ビットストリーム1 2 aへと書き込まれてもよい。

10

【0217】

図20の実施形態に適合する基本的なエントロピー復号化装置を図21に示す。

【0218】

この復号器は、基本的に図20の符号器と逆の操作を実行するため、以前に符号化されたソースシンボル列2 7と2 7 aは、2つ以上の部分的ビットストリーム(2 4、2 4 a)の集合から復号化されている。この復号器は2つの異なる処理フローを含む。即ち、符号器のデータフローを複製する、データ要求のためのフローと、符号器のデータフローの逆を表すデータフローとである。図21においては、破線の矢印はデータ要求のフローを表し、実線の矢印はデータフローを表す。復号器の構造ブロックは、基本的に符号器の構造ブロックを複製するものであるが、但し、逆の操作を実行する。

20

【0219】

本発明の好適な実施形態においては、各シンボル要求1 3 aは、そのソースシンボルが標準的なV L C符号を使用して符号化されるか、又は、P I P E符号化の概念を用いて符号化されるかについて特定する、ある指示に関連づけられている。図20に関して既に上述したように、この指示は、ソースシンボル自身によって表現されるシンタックス要素の解析規則又はシンタックスから生じてもよい。例えば、図1及び図2に関して説明したように、異なる符号化のスキーム、即ちV L C符号化またはP I P E符号化に対しては、異なるシンタックス要素のタイプが関連付けられてもよい。同様のことが、二値化の異なる部分、又はより一般的にはシンタックス要素の他のシンボル化についても適用できる。あるシンボルがV L C符号化されている場合には、要求はV L C復号器2 2 aに送られ、V L C符号語2 3 aは独特な部分的ビットストリーム2 4 aから読み出される。対応する復号化シンボル2 7 aが出力される。あるシンボルがP I P Eを用いて符号化されている場合には、そのシンボル2 7は、例えば図4に関して上述したように部分的ビットストリーム2 4の集合から復号化される。

30

【0220】

本発明の他の好適な実施形態においては、ソースシンボルの幾つかは、2つの部分から構成されるような方法で二値化される。これらの部分のうちP I P E手法を用いて符号化された部分は、関連する部分的ビットストリーム2 4からそれに応じて復号化される。また、ビン列の他の部分は標準的なV L C符号を用いて符号化され、独特な部分的ビットストリーム2 4 aから対応する符号語2 3 aを読み出すV L C復号器2 2 aを用いて復号化される。

40

【0221】

部分的ビットストリーム(V L C符号化済み及びP I P E符号化済み)の送信と多重化
符号器によって作成される部分的ビットストリーム1 2と1 2 aは、別個に伝送されることが可能であり、若しくは単一のビットストリーム内へ多重化されることが可能であり、

50

又は、部分的ビットストリームの符号語が単一のビットストリーム内へインターリーブされることが可能である。

【0222】

本発明の好ましい実施形態においては、ある量のデータに係る各部分的ビットストリームは1つのデータパケットへ書き込まれる。そのある量のデータとは、静止画像、ビデオシーケンスのフィールドやフレーム、静止画像のスライス、ビデオシーケンスのフィールド又はフレームのスライス、又はオーディオサンプルのフレームなどの、ソースシンボルの任意の集合であってもよい。

【0223】

本発明の他の好ましい実施形態においては、ある量のデータに係る2つ以上の部分的ビットストリーム12、12a、又はある量のデータに係る全ての部分的ビットストリームが1つのデータパケット内へ多重化される。多重化された部分的ビットストリームを含むデータパケットの構造を図5に示す。

【0224】

データパケット300は、(考慮対象となるある量のデータに係る)各部分的ビットストリームのデータに関するヘッダと1つのパーティションとから構成される。データパケットのヘッダ301は、データパケット(の残り)をビットストリームデータのセグメント302へと区分するための指示を含む。区分についての指示に加え、ヘッダは更なる情報を含み得る。本発明の一実施形態においては、データパケットの区分についての指示は、ビット若しくはバイトの単位、又は多数のビット若しくは多数のバイトの単位のデータセグメントの始点位置である。一実施形態において、データセグメントの始点位置はデータパケットのヘッダ内に絶対値として符号化されており、データパケットの始点に対する位置か、若しくはヘッダの終点に対する位置か、又は先行するデータパケットの始点に対する位置である。本発明の更なる実施形態においては、データセグメントの始点位置は差分的に符号化される。即ち、データセグメントの実際の始点と、データセグメントの始点についての予測との間の差のみが符号化される。この予測は、データパケットの全体のサイズ、ヘッダのサイズ、データパケット内のデータセグメントの数、先行するデータセグメントの始点の位置など、既に既知であるか又は伝送された情報に基づいて導出可能である。本発明の一実施形態においては、第1のデータパケットの始点の位置は符号化されないが、データパケットのヘッダのサイズに基づいて推測される。復号器側においては、伝送された区分の指示はデータセグメントの始点を導出するために使用される。次いで、データセグメントは、部分的ビットストリーム12、12aとして使用され、データセグメントに含まれるデータは、対応するビン復号器及びVLC復号器に順に供給される。

【0225】

符号語のインターリービング(VLC及びPIPE符号語)

幾つかのアプリケーションにおいては、1つのデータパケット内の(ある量のソースシンボルに係る)部分的ビットストリーム12の上記した多重化は、次のような欠点を持つ可能性がある。即ち、一方では、小さいデータパケットにとっては、区分について信号化するために必要とされるサイド情報のためのビット数が、部分的ビットストリーム内の実際のデータに対して重大になりうるため、最終的に符号化効率を低下させる。他方では、(例えばビデオ会議アプリケーションなどの)低遅延が要求されるアプリケーションには、多重化は不向きである可能性がある。上述した多重化の場合、部分的ビットストリームが完全に作成される以前にPIPE符号器がデータパケットの伝送を開始することは不可能である。なぜなら、パーティションの始点位置が事前に知られていないからである。さらに、一般にPIPE復号器は、データパケットの復号化を開始し得る前に、最後のデータセグメントの始点を受信するまで待機する必要がある。ビデオ会議システムのようなアプリケーションにおいては、これらの遅延は合計されて、複数のビデオ画像を用いたシステム(特に、伝送ビットレートに近いビットレートに関して、および1つの画像を符号化/復号化するために2つの画像間の時間区間に近い要求をする符号器/復号器に関して)の追加的な全体的遅延となる可能性があり、このようなアプリケーションには致命的である

。特定のアプリケーションに関するこのような欠点を克服するために、本発明の好適な実施形態に係る符号器は、2つ以上のピン符号器とVLC符号器とによって生成される符号語が、単一のビットストリームにインターリーブされるよう構成され得る。インターリーブ済みの符号語を有するビットストリームは、復号器に直接的に送信されることができる（小さなバッファ遅延を無視する場合は以下を参照）。復号器側においては、2つ以上のピン復号器とVLC符号器とは、復号化の順序でビットストリームから直接的に符号語を読み出す。即ち、復号化は最初に受信したビットから開始され得る。更に、部分的ビットストリームの多重化（又はインターリーブング）を信号化するためにサイド情報は必要とされない。

【0226】

符号語のインターリーブングを有するPIPE符号器の基本構造を図22に示す。ピン符号器10とVLC符号器10aとは、符号語を部分的ビットストリームに直接的には書き込まないが、単一の符号語バッファ29に接続されており、そこから符号語が符号化の順序でビットストリーム34へと書き込まれる。ピン符号器10は、1つ以上の新たな符号語バッファエントリに対する要求28を符号語バッファ29へと送信し、この後、符号語30を符号語バッファ29に送信し、これら符号語がリザーブされたバッファエントリに記憶される。VLC符号器10aは、VLC符号語30aを符号語バッファ29に直接的に書き込む。符号語バッファ29の（一般的には可変長の）符号語31は符号語ライタ32によってアクセスされ、その符号語ライタ32は対応するビット33を生成されたビットストリーム34へと書き込む。符号語バッファ29は先入れ先出しバッファとして作動し、先にリザーブされた符号語エントリは先にビットストリームへ書き込まれる。

【0227】

本発明の好適な一実施形態においては、符号語バッファは次のように作動する。即ち、新たなピン7が特定のピンバッファ8に送られ、ピンバッファ内に既に記憶されていたピンの数がゼロであり、かつその特定のピンバッファに接続されたピン符号器のために符号語バッファ内にリザーブされた符号語が現時点で存在しない場合には、この接続されたピン符号器10は符号語バッファへ要求を送信し、それにより、1つ以上の符号語エントリが特定のピン符号器10のために符号語バッファ29内でリザーブされる。符号語エントリは可変数のビットを有することができる。即ち、1つのバッファエントリ内のビット数の上限しきい値は、通常、対応するピン符号器に係る最大符号語サイズによって与えられる。ピン符号器によって生成される次の単数又は複数の符号語（それら符号語のために単数又は複数の符号語エントリがリザーブされている）は、符号語バッファのリザーブされた単数又は複数のエントリ内に記憶される。もし、符号語バッファ内の特定のピン符号器のためにリザーブされた全てのバッファエントリが符号語で満たされており、かつ、その特定のピン符号器に接続されたピンバッファに対して次のピンが送信される場合には、その特定のピン符号器に関し、1つ以上の新たな符号語が符号語バッファ内においてリザーブされる、等が実行される。VLC符号器10aは、符号語バッファ29の次の空のエントリに対してVLC符号語30aを直接的に書き込む。即ち、VLC符号器にとっては符号語のリザーブと符号語の書き込みとが一度で実行される。符号語バッファ29は、ある所定の方法での先入れ先出しバッファを表す。バッファエントリは順次リザーブされる。対応するバッファエントリが先にリザーブされている符号語は、先にビットストリームへ書き込まれる。符号語ライタ32は、連続的に、又は符号語30が符号語バッファ29へと書き込まれた後に、符号語バッファ29の状態をチェックする。もし第1のバッファエントリが完全な符号語を含む場合（即ち、バッファエントリはリザーブされていないが符号語を含む場合）には、対応する符号語31と対応するバッファエントリとは符号語バッファ20から取り除かれ、符号語ビット33はビットストリームへ書き込まれる。この処理は、最初のバッファエントリが符号語を含まない（即ち、リザーブされているか又は空である）状態になるまで繰り返される。復号化処理の終了時、即ち、考慮対象となる所定量のデータの全てのソースシンボルが処理されたときには、符号語バッファをフラッシュする必要がある。そのフラッシュ処理に関しては、以下の方法が最初のステップとして各ピ

10

20

30

40

50

ンバッファ／ピン符号器に適用される。即ち、ピンバッファがピンを含む場合には、結果として得られるピン列が符号語に関連するピン列を表すようになるまで、特定の又は任意の値を有するピンが加えられる（上記したようにピンを追加する1つの好ましい方法は、接頭語としてピンバッファの元のコンテンツに関するピンを含むピン列に関連する、最短の長さの可能な符号語—又はそれらのうちの1つ—を生成するようなピン値を追加することである）。次に、符号語は対応するピン符号器のための次にリザーブされたバッファエントリへ書き込まれ、（対応する）ピンバッファは空になる。もし2つ以上のバッファエントリが1つ以上のピン符号器に対してリザーブされている場合、符号語バッファは、リザーブされた符号語エントリを依然として含んでいる可能性がある。その場合、これらの符号語エントリは、対応するピン符号器のために、任意であるが有効な符号語によって満たされる。本発明のある好ましい実施形態においては、最短の有効な符号語、又は（複数ある場合には）最短の有効な符号語の1つが挿入される。VLC符号器はいかなる終端を必要としない。最後に、符号語バッファ内の残りの全ての符号語がビットストリームに書き込まれる。

10

【0228】

符号語バッファの状態を表す2つの例を図23に示す。例(a)において、符号語バッファは、符号語で満たされた4個のエントリ（それらの内の2つはVLCエントリである）と、リザーブされた3個のエントリとを含む。加えて、次の空きのバッファエントリも記載されている。最初のエントリは符号語で満たされている（即ち、ピン符号器2が前にリザーブされたエントリへ符号語を書き込んだばかりである）。次のステップにおいて、この符号語は符号語バッファから取り除かれ、ビットストリームへ書き込まれる。次に、ピン符号器3のための最初にリザーブされた符号語は、第1のバッファエントリである。しかし、このエントリを符号語バッファから取り除くことができない。なぜなら、このエントリはリザーブされているだけであって、このエントリにはこれまで全く符号語が書き込まれていないからである。例(b)において、符号語バッファは、符号語で満たされた4個のエントリ（それらの内の1つはVLCエントリである）と、リザーブされた3個のエントリとを含む。最初のエントリはリザーブされていると記載されており、符号語ライタは符号語をビットストリームに書き込むことはできない。4個の符号語が符号語バッファに含まれているが、符号語ライタは、ピン符号器3のための最初のリザーブされたバッファエントリに符号語が書き込まれるまで、待機しなければならない。符号語は、復号器側で処理を逆転させることが可能となるように（下記参照）、それらがリザーブされた順序で書き込まれる必要があることに留意されたい。更に、VLCバッファエントリは常に完全であることに留意されたい。なぜなら、符号語のリザーブと書き込みとは一度に実行されるからである。

20

30

【0229】

符号語インターリーブを有する復号器の基本的な構造を図24に示す。ピン復号器22とVLC復号器22aとは、個別の部分的ビットストリームから直接的に符号語を読み出さず、ビットバッファ38に接続されており、このビットバッファ38から符号語37、37aが符号化の順序で読み出される。符号語はビットストリームから直接的に読み出されることも可能であるため、ビットバッファ38は必ずしも必要ではないことに留意されたい。ビットバッファ38をこの図に含めた理由は、主に処理の流れの異なる態様を明確に区別するためである。インターリーブ済みの符号語を持つビットストリーム40のビット39は、先入れ先出しバッファを表しているビットバッファ38内に順次挿入される。もし特定のピン復号器22が1つ以上のピン列に対する要求35を受信した場合には、ピン復号器22は、ビットに対する要求36を介してビットバッファ38から1つ以上の符号語37を読み出す。復号器は、ソースシンボルを瞬時に復号化することができる。同様に、もしVLC復号器22aが新たなシンボルに対する要求19aを受け取れば、VLC復号器はビットバッファ38から対応するVLC符号語37aを読み取り、復号化されたシンボル27aをリターンする。（前述の）符号器は、符号語バッファを適切に作動させることにより、符号語がピン復号器によって要求されるのと同じ順序で、符号語がビットス

40

50

トリームへ確実に書き込まれる必要があることに留意されたい。復号器においては、全体の復号化プロセスがソースシンボルに対する要求によってトリガーされる。特定のピン符号器によって符号器側でリザーブされた符号語の数としてのパラメータと、対応するピン復号器によって読み出される符号語の数としてのパラメータとは、同一でなければならない。

【0230】

低遅延制約のある可変長符号語のインターリーピング

ここで説明する符号語インターリーピングは、何らかの区分情報がサイド情報として送信されることを必要とするものではない。また、符号語はビットストリーム内でインターリーブされるので、遅延も一般的に小さい。しかしながら、（例えば、符号語バッファに記憶される最大ビット数によって指定されるような）特定の遅延制約が守られることを保証するものではない。更に、符号語バッファに必要とされるバッファのサイズは、理論的には非常に大きくなる可能性がある。図23(b)に示す例を考慮した場合には、これ以上のピンがピンバッファ3へ送信されず、従って、データパケットの終点におけるフラッシュ処理が適用されるまで、ピン符号器3が符号語バッファに新たな符号語を1つも送信しないという可能性がでてくる。その場合、ピン符号器1および2に係る全ての符号語は、それらがビットストリームへ書き込まれ得る前に、データパケットの終点まで待機しなければならない可能性がある。このような欠点は、更なるメカニズムを、符号化プロセスに（及び後述するように復号化プロセスにも）追加することにより、回避することができる。この追加メカニズムの基本的な概念は、もし遅延又は遅延の上限（以下を参照）に関連した測度が、指定されたしきい値を超える場合には、第1のリザーブされたバッファエントリは、（データパケットの終点と同様のメカニズムを用いて）対応するピンバッファをフラッシュすることによって満たされる、というものである。このようなメカニズムによって、関連付けられた遅延測度が指定されたしきい値を下回るまで待機しているバッファエントリ数が削減される。復号器側においては、遅延制約を遵守するために、符号器側で挿入されていたピンを廃棄する必要がある。このようなピンの廃棄のためには、基本的に、符号器側と同様のメカニズムを使用することができる。

【0231】

VLC及びPIPE符号化のビットストリームのインターリーピングに関する可能性を詳細に説明した後で、以下の本明細書では、既に上述したシンタックス要素が図1b、図1c及び図2bに関して説明したようにソースシンボルへと分解される点について、再度焦点を当てて説明する。明確化のために、以下の説明においては、分解されたシンタックス要素が絶対変換係数レベル（absolute transform coefficient level）であると仮定する。しかし、これは単なる一例であって、他のタイプのシンタックス要素も同様に扱われ得る。特に以下の説明では、区分（パーティション）と、ブロック・ベースで異なるエントロピー符号語及びビデオ符号器の使用と、による絶対レベルの符号化について説明する。

【0232】

例えば、ビデオシーケンスの画像は、通常ブロックへと分解される。ブロック又はブロックの色彩成分は、動き補償予測又はイントラ予測のいずれかにより予測される。ブロックは異なるサイズを有することができ、二次(quadratic)又は長方形(rectangular)のいずれかであってもよい。あるブロックの全てのサンプル又はあるブロックの色彩要素は、予測パラメータ、即ち、（既に符号化済みの画像の集合内において一つの参照画像を識別する）参照インデックス、（参照画像と現在の画像との間におけるブロック同士の動きについての測度を特定する）動きパラメータ、補間フィルタを特定するパラメータ、イントラ予測モード、その他などの同一の集合を使用して予測される。動きパラメータは、水平及び垂直の成分を持つ変位ベクトルにより、又は、6個の成分からなるアフィン動きパラメータなどの高次の動きパラメータにより、表現されることができる。また、（参照インデックスと動きパラメータなどの）予測パラメータの2つ以上の集合が単一のブロックに関連付けられることも可能である。その場合には、予測パラメータの各集合について、そのブ

ロック又はブロックの色彩要素に関する単一の中間的な予測信号が生成され、最終的な予測信号は、（複数の）中間的な予測信号の重み付き合計によって形成される。重み付けパラメータと、潜在的にはまた（重み付き合計に加算される）一定のオフセットとは、ある画像、参照画像、若しくは参照画像の集合に対して固定されるか、又はそれら是对応するブロックに関する予測パラメータの集合の中に含まれることができる。同様に、静止画像もまた複数のブロックへと分解されることが多く、更に、それらブロックは、（空間的イントラ予測方法、又はブロックのDC成分を予測する単純なイントラ予測方法であり得る）イントラ予測の方法によって予測される。極端な場合には、予測信号はゼロにもなり得る。

【0233】

元のブロック又は元のブロックの色彩要素と、対応する予測信号であって残余信号とも呼ばれる信号との間の差は、通常、変換されかつ量子化される。残余信号に対してある2次元変換が適用され、結果として得られる変換係数が量子化される。この変換符号化に関し、予測パラメータのある特定の集合が使用されたブロック又はそのブロックの色彩成分は、変換が適用される前に更に分割されていてもよい。変換ブロックは、予測のために使用されたもの以下の大きさであってよい。更に、変換ブロックが予測に使用されるブロックのうちの2つ以上を含むことも可能である。1つの静止画像又はビデオシーケンスの1つの画像の中の異なる変換ブロックは異なるサイズを有することができ、また、変換ブロックは二次又は長方形のブロックを表すことができる。

【0234】

これら予測及び残余のパラメータの全ては、シンタックス要素のストリーム138と226とをそれぞれ形成してもよい。

【0235】

結果として得られる量子化された変換係数であって変換係数レベルとも呼ばれる係数は、次に、上述した符号化スキームの任意のスキームによりエントロピー符号化を使用して伝送されてもよい。この目的で、変換係数レベルのブロックが、あるスキャンを使用して変換係数値のあるベクトル（即ち、ある順序付けられた集合）上にマップされてもよい。このとき、異なるブロックに対しては異なるスキャンが使用され得る。多くの場合、ジグザグスキャンが使用される。インターレースされたフレームの1つのフィールドのサンプルだけを含むブロック（これらのブロックは、符号化されたフィールド内のブロックか、又は符号化されたフレーム内のフィールドブロックであり得る）については、フィールドブロックのために特別に設計された異なるスキャンを使用することもよくあることである。結果として得られる順序付けられた変換係数の列を符号化するための可能な符号化スキームは、ランレベル符号化である。通常、変換係数レベルの多数はゼロであり、ゼロに等しい連続的な変換係数レベルの集合は、そのゼロに等しい連続的な変換係数レベルの数（ラン）をそれぞれのシンタックス要素により符号化することで、効率的に表現され得る。残りの（非ゼロの）変換係数については、実際のレベルがそれぞれのシンタックス要素の形態で符号化される。ランレベルの符号にはいろいろな選択肢がある。非ゼロの係数の前のランと非ゼロの変換係数のレベルとは、単一のシンタックス要素を使用して一緒に符号化され得る。多くの場合、ブロックの終点についての特別なシンタックス要素であって、最後の非ゼロの変換係数の後に送信されるものが含まれる。又は、最初に非ゼロの変換係数レベルの数が符号化され、この数に基づいてレベルとランとが符号化されることも可能である。

【0236】

H. 264/AVCにおける高度に効率的なCABACエントロピー符号化において、幾分異なる手法が用いられている。ここでは、変換係数レベルの符号化は、3つのステップに分けられている。第1のステップでは、変換ブロックごとにバイナリーシンタックス要素 `coded_block_flag` が送信され、この要素は変換ブロックが有意の変換係数レベル（即ち非ゼロである変換係数）を含むかどうかを通知する。このシンタックス要素が有意の変換係数レベルが存在していることを示す場合には、バイナリー値の有意性

10

20

30

40

50

マップが符号化され、どの変換係数レベルが非ゼロ値を有しているかを特定する。次に、逆のスキャン順序で非ゼロの変換係数レベルの値が符号化される。有意性マップは、以下のようにシンタックス要素ストリーム138へと符号化される。スキャン順序で係数ごとにバイナリーシンタックス要素 `significant_coeff_flag` が符号化され、この要素は対応する変換係数レベルがゼロと等しくないかどうかを指定する。 `significant_coeff_flag` ビンが1に等しい場合には、即ち非ゼロの変換係数レベルがこのスキャン位置に存在する場合には、更なるバイナリーシンタックス要素 `last_significant_coeff_flag` が符号化される。このピンは、現在の有意の変換係数レベルがブロック内部における最後の有意変換係数レベルであるかどうか、又は更なる有意変換係数レベルがスキャンの順序で続くかどうかを示す。 `last_significant_coeff_flag` が更なる有意変換係数が続いていることを示す場合には、そのブロックについての有意性マップを特定するために、更なるシンタックス要素が符号化されることはない。次のステップにおいて、有意の変換係数レベルの値が符号化され、それらのブロック内の位置は有意性マップによって既に決定されている。有意の変換係数レベルの値は、以下の3つのシンタックス要素を使用して、逆のスキャン順序で符号化される。バイナリーシンタックス要素 `coeff_abs_greater_one` は、有意の変換係数レベルの絶対値が1より大きいかどうかを示す。バイナリーシンタックス要素 `coeff_abs_greater_one` が絶対値は1より大きいと示す場合には、さらなるシンタックス要素 `coeff_abs_level_minus_two` が送信され、これは変換係数レベルの絶対値マイナス2を指定する。この種のシンタックス要素の処理は、以下に記載する実施形態において図1b、図1c及び図2bに従って実行される。最後に、変換係数値の正負符号を指定するバイナリーシンタックス要素 `coeff_sign_flag` が、有意の変換係数レベルごとに符号化される。ここでも、有意性マップに関連しているシンタックス要素はスキャンの順序で符号化される一方で、変換係数レベルの実際の値に関連しているシンタックス要素は逆のスキャン順序で符号化され、より適切なコンテキストモデルの使用を可能にすることに留意されたい。また、H. 265/HEVCの最初のテストモデルのように、適応型のスキャンパターンが有意性マップのために使用されることも可能である。H. 265/HEVCの最初のテストモデルでは、 4×4 よりも大きな変換ブロックについては、絶対変換係数レベルの符号化のために他の概念が使用される。 4×4 よりも大きな変換ブロックの場合には、その大きな変換ブロックは 4×4 ブロックへと区分され、それら 4×4 ブロックはスキャンの順序で符号化される一方で、 4×4 ブロックの変換係数の符号化については逆のスキャン順序が使用される。

【0237】

H. 264/AVCのCABACエントロピー符号化においては、変換係数レベルに関する全てのシンタックス要素はバイナリー確率モデリングを用いて符号化される。非バイナリーシンタックス要素である例えば `coeff_abs_level_minus_two` は、まず二値化される。即ち、バイナリー決定 (`bins`) の列上にマップされ、これらのピンは順次符号化される。二値のシンタックス要素 `significant_coeff_flag` と、`last_significant_coeff_flag` と、`coeff_abs_greater_one` と、`coeff_sign_flag` とは、直接的に符号化される。符号化された各ピン (二値のシンタックス要素を含む) は、コンテキストと関連付けられる。コンテキストは、符号化されたピンの等級ごとの確率モデルを表わしている。2つの可能なピン値のうちの1つに係る確率に関連する測度は、対応するコンテキストを用いて既に符号化されているピンの値に基づいて、各コンテキストごとに推定される。変換符号化に関連する複数のピンに関し、符号化に使用されるコンテキストは、既に伝送されたシンタックス要素に基づいて又はブロック内部の位置に基づいて選択される。

【0238】

有意性マップを符号化した後で、ブロックは逆のスキャン順序で処理される。上述したよ

うに、H. 265/H.265/HEVCの最初のテストモデルでは別の概念が使用される。4×4よりも大きな変換ブロックは4×4のブロックへと区分され、結果として得られる4×4のブロックがスキャンの順序で符号化される一方で、その4×4のブロックの係数は逆のスキャン順序で符号化される。以下の説明は、H. 265/H.265/HEVCの最初のテストモデルとH. 264/H.264/AVCとにおける全ての4×4のブロックと、H. 264/H.264/AVCにおける8×8のブロックとについて有効であり、この説明は、シンタックス要素のストリーム138と226との構造についてそれぞれ適用可能である。

【0239】

もしスキャン位置が有意であれば、即ち係数がゼロでない場合には、二値のシンタックス要素 `coeff_abs_greater_one` がストリーム138の中で伝送される。最初に（ブロック内部において）、対応するコンテキストモデルの集合の第2のコンテキストモデルが、そのシンタックス要素 `coeff_abs_greater_one` のために選択される。もしブロック内部の任意のシンタックス要素 `coeff_abs_greater_one` の符号化された値が1と等しい場合には（即ち絶対係数が2よりも大きい場合には）、そのコンテキストモデリングは、集合のうちの第1のコンテキストモデルへとスイッチバックし、このコンテキストモデルをブロックの終点まで使用する。その他の場合（ブロック内部の `coeff_abs_greater_one` の全ての符号化された値がゼロであり、対応する絶対係数レベルが1に等しい場合）には、考慮対象であるブロックの逆のスキャン順序で既に処理されたゼロに等しいシンタックス要素 `coeff_abs_greater_one` の数に基づいて、コンテキストモデルが選択される。シンタックス要素 `coeff_abs_greater_one` に関するコンテキストモデルの選択は、以下の式によって要約され得る。ここで、現在のコンテキストモデル・インデックス C_{t+1} は、前のコンテキストモデル・インデックス C_t と、以前に符号化され次式において bin_t で表されるシンタックス要素 `coeff_abs_greater_one` の値と、に基づいて選択される。ブロック内の最初のシンタックス要素 `coeff_abs_greater_one` に関しては、コンテキストモデル・インデックスは $C_t = 1$ と等しく設定される。

$$C_{t+1}(C_t, bin_t) = \begin{cases} 0 & bin_t = 1 \\ \min(C_t + 1, 4) & bin_t = 0 \end{cases}$$

【0240】

同じスキャン位置に関する `coeff_abs_greater_one` のシンタックス要素が1と等しい場合には、絶対変換係数レベルを符号化するための2番目のシンタックス要素である `coeff_abs_level_minus_two` だけが符号化される。非二値のシンタックス要素 `coeff_abs_level_minus_two` はビン列へと二値化され、この二値化の第1のビンに関し、コンテキストモデル・インデックスは以下に説明するように選択される。二値化の残りのビンは固定型のコンテキストで符号化される。二値化の第1のビンに関するコンテキストは次のように選択される。最初のシンタックス要素 `coeff_abs_level_minus_two` については、シンタックス要素 `coeff_abs_level_minus_two` の第1のビンに関するコンテキストモデルの集合のうちの第1のコンテキストモデルが選択され、対応するコンテキストモデル・インデックスは $C_t = 0$ に等しく設定される。シンタックス要素 `coeff_abs_level_minus_two` の更なる第1のビンについては、コンテキストモデリングは集合内の次のコンテキストモデルに切り替わり、ここで、集合内のコンテキストモデルの数は5に制限されている。コンテキストモデルの選択は次式によって表すことができ、ここで、現在のコンテキストモデル・インデックス C_{t+1} は前のコンテキストモデル・インデックス C_t に基づいて選択される。

$$C_{t+1}(C_t) = \min(C_t + 1, 4)$$

【0241】

上述のように、ブロック内の第1のシンタックス要素 `coeff_abs_level_minus_two` に関し、コンテキストモデル・インデックスは $C_t = 0$ と等しく設定される。コンテキストモデルの異なる集合が、シンタックス要素 `coeff_abs_greater_one` と、`coeff_abs_level_minus_two` と、に対して定義され得ることに留意されたい。更に、H. 265/HEVCの最初のテストモデルに関しては、 4×4 よりも大きな変換ブロックは 4×4 のブロックへと区分され得ることに留意されたい。区分された 4×4 のブロックはスキャンの順序で符号化されてもよく、区分された各 4×4 のブロックに関し、コンテキストの集合が先行する 4×4 のブロック内の2以上の係数の数に基づいて導出されてもよい。 4×4 よりも大きい変換ブロックの最初の 4×4 のブロックと、元の 4×4 変換ブロックとについて、別々のコンテキストの集合が使用されてもよい。

10

【0242】

つまり、以下の説明において、`coeff_abs_greater_one` と `coeff_abs_level_minus_two` とが以下の実施形態に従ってその中に分解されるようなソースシンボルのうちの任意のものに対して、コンテキストベースの符号化が使用される場合はいつも、このコンテキストの導出が、例えば割当部114及び212と、VLC符号器／復号器102及び200によって使用されてもよい。

20

【0243】

現状技術との比較におけるCABAC又はPIPEによって処理されるビンの数の観点と演算量の観点とから見た複雑性を低減する目的で、又は、符号化効率を高めるためにも、以下に記載する実施形態は、画像及びビデオの符号器及び復号器における異なる区分 $140_1 \sim 140_3$ に対して異なる可変長符号を使用して、絶対レベルを符号化するための手法を説明するものである。しかし、以下に記載する実施形態は、画像及びビデオの符号器に関する全ての種類の絶対レベル、例えば動きベクトルの差、又は適応型ループフィルタの係数などに対して適用可能である。変換係数レベルの符号化は以下に記載のように実行される一方で、有意性マップの符号化は、H. 265/HEVCの最初のテストモデル又は上述の説明にある符号化のままであってもよく、又は、有意性マップの符号化は、H. 264/AVCその他における場合のように実行されることもできる。

30

【0244】

上述のように、絶対変換レベルの符号化は複数の区分 140_{1-3} 内において実行される。符号化スキームは、図1bにおいて3つの区分 140_{1-3} を用いてこれまで例示的に説明してきた。スキームの境界142と144とは可変であり、その結果、可変の区分サイズをもたらす。符号化のスキームは以下のように実行される。

【0245】

絶対変換係数レベル (z) が限界1よりも小さい場合には、第1の成分又はソースシンボル、即ちその絶対変換係数レベルを符号化するために、第1のエントロピー符号が使用される。もし絶対変換係数レベルが第1の区分 140_1 の境界の限界1以上である場合には、第1の区分 (140_1) の境界の限界1 (142) がその絶対変換係数レベルから減算され、結果として得られる値 z' が第2のエントロピー符号を用いて符号化される。もし残りの絶対変換係数レベル z' が、第2の区分 140_2 の境界の限界2－限界1以上である場合には、第2の区分 140_2 の境界の限界2－限界1は、絶対変換係数レベル z' から再度減算され、結果として得られる値は第3のエントロピー符号を用いて符号化される。一般的に言えば、区分の境界に到達したときに、絶対変換係数レベルから対応する区分を減算した結果の値を符号化するために、その境界の次の区分のためのエントロピー符号が使用される。

40

50

【0246】

エントロピー符号は、ランレングス符号若しくはルックアップテーブル（例えばハフマン符号など）のように単純な可変符号であってもよく、又は、C A B A C若しくはP I P Eのような確率モデルを使用するより複雑なエントロピー符号であってもよい。区分の数と区分の境界とは、可変であっても又は実際のシンタックス要素に基づくものであってもよい。図1bに示す区分の概念は以下の利点を有する。以下においては、絶対変換係数を一例として使用するが、それらは任意の他のシンタックス要素と置き換えられてもよいことを理解されたい。一例として、絶対変換係数レベルの確率分布はほぼ幾何分布であってもよい。そのため、幾何分布に最適化されたエントロピー符号が絶対変換係数レベルを符号化するために使用されてもよい。しかし、そのようなモデルは、たとえコンテキストモデリングと確率モデルの選択が使用されたとしても、常に局所的に最適とは限らない。例えばある変換ブロックに関し、もしその変換ブロックが同一量の非常に低位及び中間の範囲の変換係数レベルを含む場合には、局所的な絶対変換係数レベルは、全く幾何分布でない分布に従ってしまう一方で、画像又はビデオの特定量のブロックに関しては、多数の原則によって、そのモデルが（所定の正確さをもって）正しいままであることも起こり得る。そのような場合には、幾何分布モデルは適切なモデルではない。また、大きな値を有する絶対変換係数レベルを考慮する場合には、これらの大きな値の分布は均一であることが多い。区分の概念は、異なる絶対変換係数レベルについては異なる確率モデルを許可する。低い絶対値については、より高効率のためにより複雑なエントロピー符号を適用することが可能であり、他方、大きな絶対値レベルについては、複雑性を低減するために、より複雑でないエントロピー符号を使用することが可能である。

10

20

【0247】

上述したように、様々な区分に対して適切なエントロピー符号が使用される。本発明の好適な実施形態においては、3つのタイプのエントロピー符号が使用される。第1のエントロピー符号はP I P Eを使用する。しかし、他の実施形態によれば、C A B A Cのようなエントロピー符号化方法、又は任意の他の算術復号器を代替的に使用することもできる。つまり、第1のシンボル s_1 （図2bを参照）は、P I P E符号化の経路を介して符号化されてもよい。サブ分割部100及び再結合器220も適切に作動する。

【0248】

第2のタイプは、ゴロム符号と、サブセット（例えばゴロムーライス符号）を含む幾つかの切り詰められた変形例とが使用されてもよい。つまり、第2のシンボル s_2 （図2bを参照）は、V L C符号器／復号器102／200内でそのようなV L C符号を介して符号化されてもよい。サブ分割部100及び再結合器220も適切に作動する。

30

【0249】

指数ゴロム符号は、第3のタイプとして使用される。つまり、第3のシンボル s_3 （図2bを参照）は、V L C符号器／復号器102／200内でそのようなV L C符号を介して符号化されてもよい。サブ分割部100及び再結合器220も適切に作動する。様々なV L C符号、及び、V L CとP I P E又はV L Cと算術符号との組合せも可能である。

【0250】

第1の符号は、より複雑ではあるがより良好な圧縮性能をもたらす一方で、第2のエントロピー符号は、複雑性と性能との間の合理的な妥協点を表している。最後のエントロピー符号は例えば指数ゴロム符号であるが、複雑性は非常に低い。以下に異なる区分の符号化について説明する。

40

【0251】

もし区分140₁など、即ちシンボル s_1 などが、P I P E符号器104(C A B A C又は任意の他の算術符号器を代替的に使用し得るが、ここではこれ以上説明しない)のように確率モデルを用いたエントロピー符号器を使用してエントロピー符号化される場合には、サブ分割部100はそれをP I P E符号器104へと送る。まず、非二値の値を持つ絶対変換係数レベルがある二値化方法を用いてシンボライザ122内で二値化されてもよい。この二値化は、非二値の値を持つ絶対変換係数レベルを二値のビン列へとマップする。ビ

50

ン列の各ピンは、割当部 1 1 4 により選択されたコンテキストと一緒に符号化される。コンテキストモデリングは、第 1 のピンについて実行され、ピン列の後続のピンについては、H. 2 6 4 / A V C における `coeff__abs__level__minus__two` として固定されてもよく、又は、ピン列の各ピンについて異なるコンテキストモデリングが使用されてもよい。二値化は、ゴロム符号又は指数ゴロム符号のような可変長符号、又は他の可変長符号であってもよいことに留意すべきである。

【0 2 5 2】

次に、区分 1 4 0₂ 又はシンボル s_2 などの区分が、ゴロム符号を用いて、ここでは V L C 符号器 1 0 2 内で符号化され、それぞれ V L C 復号器内で復号化されてもよい。ゴロム符号は、幾何分布されたソースのために設計された、エントロピー符号の集合である。もしゴロム符号の次数がゼロであれば、そのゴロム符号はユニナリー符号としても知られている。ユニナリー符号は、H. 2 6 4 / A V C における `coeff__abs__level__minus__two` の二値化に関連している。ゴロム符号は以下のように構築される。特定のゴロムパラメータ k について、値 n が整数除算を用いてゴロムパラメータ k によって除算され、剰余 r が計算される。

$$p = \left\lfloor \frac{n}{k} \right\rfloor$$

$$r = n - pk$$

【0 2 5 3】

上述した式により特定されたパラメータを導出した後で、値 n は 2 つの部分を用いて符号化され得る。第 1 の部分は接頭部とも呼ばれ、ユニナリー符号である。結果として得られる値 $p + 1$ は、1 の数と終了のゼロとを特定するか、又はその逆を特定する。剰余の値は、剰余部とも呼ばれかつ r により示されるものであり、切り詰められた二値符号により表される。特定の実施形態によれば、ソースシンボル s_2 のようなソースシンボルを符号化するためには、ゴロム符号のサブセットであるゴロムーライス符号が使用される。また、区分 1 4 0₂ のように境界を含む区分 1 4 0_{1..3} に関してそのようなエントロピー符号を使用する場合には、それぞれのソースシンボル（ソースシンボル s_2 など）のアルファベットの制限され、ゴロムーライス符号は、符号化効率が改善され得るように、修正されることができる。ゴロムーライス符号のパラメータは固定型でも可変型でもよい。パラメータが可変型である場合には、パラメータはコンテキストモデリングの段階の一部として推定されてもよい。例えばソースシンボル s_2 が V L C 符号器 1 0 2 に入る場合には、V L C 符号器 1 0 2 はゴロムーライス符号のパラメータを s_2 のコンテキストから決定してもよい。ゴロムーライス符号は、2 の冪に対するパラメータを有するゴロム符号である。従って、ゴロムーライス符号は 2 による除算と乗算とに基づいており、また、ゴロムーライス符号は、シフトと追加の操作を有する二値のアーキテクチャにおいて効率的に実現され得る。従って、ゴロムーライスパラメータとゴロムパラメータとの間の関係は、 $k_{\text{GOLOMB}} = 2^{k_{\text{RICE}}}$ である。ゴロムーライス符号の場合には、剰余部はその剰余の値の正に二値表現である。ゼロのゴロムーライスパラメータについては、結果として得られる符号はユニナリー符号と同一であり、剰余部を有していない。1 と等しいパラメータについては、剰余部は同じユニナリー接頭部を共有する 2 つの入力シンボルを有する 1 つのピンから構成される。以下に、選択されたゴロムーライスパラメータのための幾つかの例示的なテーブルを示す。

【0 2 5 4】

10

20

30

40

【表 8】

Value	Prefix	Rem.	Prefix	Rem.	Prefix	Rem.	Prefix	Rem.
	k=0		k=1		k=2		k=3	
0	0		0	0	0	00	0	000
1	10		0	1	0	01	0	001
2	110		10	0	0	10	0	010
3	1110		10	1	0	11	0	011
4	11110		110	0	10	00	0	100
5	111110		110	1	10	01	0	101
6	1111110		1110	0	10	10	0	110
7	11111110		1110	1	10	11	0	111
8	111111110		11110	0	110	00	10	000
9	1111111110		11110	1	110	01	10	001
10	11111111110		111110	0	110	10	10	010
11	111111111110		111110	1	110	11	10	011
12	1111111111110		1111110	0	1110	00	10	100
13	11111111111110		1111110	1	1110	01	10	101

【0 2 5 5】

例えば区分 $1\ 4\ 0_2$ の場合の限界 2－限界 1 などのような区分の範囲と、ゴロム－ライス符号のパラメータとが与えられた場合、トランケーション（切り詰め）は以下のように実行され得る。ゴロム－ライス・パラメータは、剰余部を表すために必要とされるビンの数を記述し、パラメータの 2 の冪に対する値は、同一の接頭部を用いて表され得る値の数を記述する。これらの値は接頭部のグループを形成する。例えば、パラメータ 0 については、1 つの接頭部だけが 1 つの特定の値を表し得る一方で、パラメータ 3 については、8 個の入力値が同一の接頭部を共有し、従ってパラメータ 3 に関しては 1 つの接頭部のグループが 8 個の値を含む。ある制限されたソースアルファベットと所与のゴロム－ライス符号に関し、接頭部の最後のピンは、最後の接頭部グループ内の値のために残しておかれてもよく、その結果、固定長を有する接頭部が生じてよい。

【0 2 5 6】

例えば、範囲が 9 で、ゴロム－ライス・パラメータが 2 であるとする。この例の場合には、同じ接頭部を用いて表され得る値の数は 4 である。最大値は 9 であり、この数はまた、境界を超えて、次の区分の次のエントロピー符号が使用されなければならないということを示している。この例の場合には、0～3 の値は接頭部 0 を有し、4～7 の値は接頭部 1 0 を有し、8～9 の値は接頭部 1 1 0 を有する。8～9 の値が最後の接頭部グループを形成したので、それらのトレーリング・ゼロ（trailing zero）は無視され、値 8～9 は 1 1 によって表されることができる。換言すれば、仮に、ソースシンボル s_2 が V L C 符号器 1 0 2 に入り、ここで s_2 の可能な値の数が 9（＝限界 2－限界 1＋1）であり、そのソースシンボルに関するゴロム－ライス・パラメータが 2 であったとする。その場合、それぞれのゴロム－ライス符号が V L C 符号器 1 0 2 によってそのソースシンボルのために出力され、そのゴロム－ライス符号は上述した接頭部を有することになる。符号語の剰余部については、切り詰められた符号は V L C 符号器 1 0 2 によって以下のように導出されてもよい。通常は、ゴロム－ライス符号のパラメータは剰余部のビンの数を示す。切り詰められた場合には、剰余の全てのピンが符号化される必要はない。切り詰められた場合に関しては、固定された接頭部を有する全ての値（例えば接頭部のピンは無視されている）がカウントされる。符号は切り詰められているので、カウントされる値は常に接頭部のための値の最大数以下であることに留意されたい。もし剰余部の切り詰めが可能である場合には、最後の接頭部グループに関する切り詰められた剰余部の導出は、以下のステップにより実行され得る。第 1 に、カウントされる数以下の 2 の冪に対する最大数 l が導出される。次に、第 2 のステップでは、カウントされる数よりも大きい 2 の冪に対する最小数 h が導出される。第 1 の値 l は、接頭部グループ内の値の数を記述し、剰余の h 個のピンを

有している。これらの値のすべての剰余は0で始まり、その後には、剰余グループ内の値の数に限定された剰余の二値の表現が続く。最後の接頭部グループであって、この時点では新たな接頭部グループとして扱われるグループの剰余の値に関しては、同様の処理が実行されるが、剰余が1で始まる第1の剰余グループを形成した結果的な値は例外である。このような処理は剰余が導出されるまで実行される。一例として、範囲が14でパラメータが3と仮定する。第1の接頭部グループは値0～7を含み、第2の選択部グループは値8～13を含む。第2の選択部グループは6個の値を含む。パラメータは、 $l = 2$ 及び $h = 3$ である。従って、選択部グループの始めの4個の値は、3個のビン（トレーリング・ゼロと4個の値を判別するための二値表現と）を有する剰余によって表される。最後の2つの値に関しては、同様の処理が再度実行される。パラメータは、 $l = 1$ 及び $h = 2$ である。最後の2つの値の剰余は10と11として表され得る。この方法を説明する他の例は、4個のゴロム・ライス・パラメータと10個の範囲である。この例に関しては、パラメータは、 $l = 3$ 及び $h = 4$ である。これらのパラメータと共に、始めの8個の値についての切り詰められた剰余部分は4個のビンによって表される。残りの2つの値は、前の例と同じ剰余部分を有する。もし前の例に関する範囲が9であれば、第2のランについてのパラメータは $l = 0$ 及び $h = 1$ である。残っているただ1つの値の剰余部分は1である。

【0257】

エントロピー符号の第3のタイプは指数ゴロム符号であってもよい。それらの符号は、ソースシンボル s_3 など、（例えばパラメータ0を用いた）均等な確率分布のために使用されてもよい。つまり、それらの符号化はVLC符号器／復号器のペアに任されてもよい。上述したように、大きな絶対変換係数レベルが均一的に分布される場合が多い。さらに詳細には、ゼロ次の指数ゴロム符号が最後の区分140₃を符号化するために使用されてもよい。その始点、従って前の区分140₂との境界144は可変でもよい。境界144の位置は、VLC符号器102／復号器200により、符号化／復号化されたソースシンボル106、108及び／若しくは110、又はシンタックス要素138（又は、218、204及び／若しくは208、又はシンタックス要素226）に基づいて制御されてもよい。

【0258】

ある好適な実施形態においては、図1bに示すように、区分の数は3であり、境界142と144とは可変であってもよい。第1の区分140₁に関しては、上述のようなPIPE符号化が使用されてもよい。しかしながら、代替的にCABACが使用されてもよい。その場合には、PIPE符号器／復号器のペアはバイナリー算術符号器／復号器のペアで置き換えられてもよい。切り詰められたゴロム・ライス符号が第2の区分140₂に関して使用され、ゼロ次の指数ゴロム符号が最後の区分140₃に関して使用されてもよい。

【0259】

他の好適な実施形態においては、区分140₁₋₃の数は3であり、第1の境界142は固定されている一方で、第2の境界144は可変である。第1の区分140₁に関しては、CABAC又はPIPEが使用されてもよい。切り詰められたゴロム・ライス符号が第2の区分140₂に関して使用され、ゼロ次の指数ゴロム符号が最後の区分140₃に関して使用されてもよい。

【0260】

他の好適な実施形態においては、区分の数は2に等しい。第1の区分140₁はCABAC又はPIPEを使用してもよい。第2の区分140₂はゴロム・ライス符号を使用してもよい。

【0261】

他の好適な実施形態においては、区分の数は3に等しく、両方の境界142と144とは可変である。例えば、第1の区分140₁に関してはCABAC又はPIPEが使用され、第2の区分140₂は切り詰められたゴロム・ライス符号を使用し、第3の区分140₃はゼロ次の指数ゴロム符号を使用してもよい。

【0262】

10

20

30

40

50

ある好適な実施形態においては、適応型の確率モデルを用いたC A B A C又はP I P Eを使用する第1の区分1 4 0₁の境界1 4 2は2である。この好適な実施形態においては、第1のピンに係るコンテキストモデリングは、c o e f f _ a b s _ g r e a t e r _ o n eについて上述したように実行されてもよく、第2のピンに係るコンテキストモデリングは、同じく上述したように、H. 2 6 4 / A V Cにおいてc o e f f _ a b s _ l e v e l _ m i n u s _ t w oについて記述されているように実行されてもよい。後者のコンテキスト決定は、割当部1 1 4及び2 1 2によってそれぞれ決定されてもよい。他の好適な実施形態においては、（例えばP I P E又はC A B A Cの）確率モデリングを用いたエントロピー符号化を使用する第1の区分1 4 0₁の境界1 4 2は2である。この好適な実施形態に関しては、第1及び第2の両方のピンのためのコンテキストモデリングは、上述したように、H. 2 6 4 / A V Cにおいてc o e f f _ a b s _ g r e a t e r _ o n eについて記述されているように実行されてもよい。第2のピンに関しては、c o e f f _ a b s _ g r e a t e r _ o n eについて記述されているようなコンテキストセットの評価が別々に実行されてもよい。

10

【0 2 6 3】

ある好適な実施形態においては、（例えばC A B A C又はP I P Eの）確率モデリングを用いたエントロピー符号化を使用する第1の区分1 4 0₁の境界1 4 2は、1と等しくてもよい。各ソースシンボルのピン列の唯一のピン（又はアルファベットシンボル）に関し、コンテキストモデリングは、上述したように、H. 2 6 4 / A V Cにおいてc o e f f _ a b s _ g r e a t e r _ o n eについて記述されているように実行されてもよい。

20

【0 2 6 4】

ある好適な実施形態においては、（例えばC A B A C又はP I P Eの）確率モデリングを用いたエントロピー符号化を使用する第1の区分1 4 0₁の境界1 4 2は、3であってもよい。各ソースシンボルのピン列の第1及び第2のピンのコンテキストモデリングは、H. 2 6 4 / A V Cにおけるc o e f f _ a b s _ g r e a t e r _ o n eのように実行されてもよい。第3のピンに関するコンテキストモデリングは、H. 2 6 4 / A V Cにおけるc o e f f _ a b s _ l e v e l _ m i n u s _ t w oのように実行されてもよい。第2のピンに関しては、c o e f f _ a b s _ g r e a t e r _ o n eについて記述されているようなコンテキストセットの評価が別々に実行されてもよい。

【0 2 6 5】

ある好適な実施形態においては、切り詰められたゴロムーライス符号の1セットが第2の区分1 4 0₂のエントロピー符号として使用されてもよい。エントロピー符号パラメータに基づいて第3の区分1 4 0₃の始点を特定する第2の区分の境界1 4 4は、可変であってもよい。また、この好適な実施形態においては、ゴロムーライス・パラメータは3に制限され、パラメータの選択は、H. 2 6 4 / A V Cにおけるc o e f f _ a b s _ l e v e l _ m i n u s _ t w oに関するコンテキストモデリングのように実行されてもよい。限界1～限界2の範囲は可変であってもよく、ゴロムーライス・パラメータに依存してもよい。もしそのパラメータがゼロである場合には、その範囲は8になる。パラメータ1に対しては、範囲は10である。パラメータ2の場合には、範囲は12であり、パラメータ3の場合には、範囲は16と等しくなる。この好適な実施形態では、変換係数のブロックの始点において、ゴロムーライス・パラメータはゼロに設定される。第1の境界以上であるブロック内の各符号化済みの変換係数レベルに対し、対応するゴロムーライス符号が使用される。レベルを符号化（復号化）した後で、次の評価が実行され、第1の境界以上である次のレベルを符号化（復号化）するためのゴロムーライス・パラメータが更新される。この形態の適応を用いることによって、ゴロムーライス・パラメータは減少され得ない点に留意されたい。

30

40

【0 2 6 6】

パラメータの適応規則は次式のように要約することができる。ここで、 k_{t+1} は次のレベル値の符号化に使用されるべきゴロムーライス・パラメータを示し、 $value_t$ は対応するゴロムーライス・パラメータ k_t を有する以前に符号化された値を示す。

50

【数 3 1】

$$k_{i+1} = \begin{cases} 0 & \text{value}_i \in [0,1] \wedge k_i < 1 \\ 1 & \text{value}_i \in [2,3] \wedge k_i < 2 \\ 2 & \text{value}_i \in [4,5] \wedge k_i < 3 \\ 3 & \text{value}_i > 5 \wedge k_i < 4 \\ k_i & \text{otherwise} \end{cases} \quad (\text{QQ})$$

【0 2 6 7】

10

ある好適な実施形態においては、切り詰められたゴロムーライス符号のセットが第2の区分140₂のエントロピー符号として使用されてもよい。エントロピー符号パラメータに基づいて第3の区分140₃の始点を特定する第2の区分140₂の境界144は、可変であってもよい。また、この好適な実施形態においては、ゴロムーライス・パラメータは3により制限され、パラメータの選択はH. 264/AVCにおけるcoeff__abs__level__minus__twoに関するコンテキストモデリングのように実行されてもよい。範囲は可変であってもよく、ゴロムーライス・パラメータに依存してもよい。パラメータが0である場合には、その範囲は8になる。パラメータ1に対しては、範囲は10である。パラメータ2の場合には、範囲は12であり、パラメータ3の場合には、範囲は16と等しくなる。この好適な実施形態では、ブロックの始点においてゴロムーライス・パラメータはゼロに設定される。このゴロムーライス・パラメータの適応は、上述の等式(QQ)に説明したように実行される。この形態の適応を用いることによって、パラメータは減少され得ない点に留意されたい。

20

【0 2 6 8】

他の好適な実施形態においては、切り詰められたゴロムーライス符号のセットが第2の区分140₂のエントロピー符号として使用されてもよい。エントロピー符号パラメータに基づいて第3の区分140₃の始点を特定する第2の区分140₂の境界144は、固定であってもよい。また、この好適な実施形態においては、ゴロムーライス・パラメータは3により制限され、パラメータの選択はH. 264/AVCにおけるcoeff__abs__level__minus__twoに関するコンテキストモデリングのように実行されてもよい。第2の区分140₂の範囲は14に固定されてもよい。この好適な実施形態においては、ゴロムーライス・パラメータはブロックの始点においてゼロに設定されてもよい。このゴロムーライス・パラメータの適応は、上述の等式(QQ)に説明したように実行される。この形態の適応を用いることによって、パラメータは減少され得ない点に留意されたい。

30

【0 2 6 9】

更に他の好適な実施形態においては、切り詰められたゴロムーライス符号のセットが第2の区分140₂のエントロピー符号として使用されてもよい。エントロピー符号パラメータに基づいて第3の区分140₃の始点を特定する第2の区分140₂の境界144は、可変であってもよい。また、この好適な実施形態においては、ゴロムーライス・パラメータは3により制限され、パラメータの選択は、H. 264/AVCにおけるcoeff__abs__level__minus__twoに関するコンテキストモデリングのように実行されてもよい。範囲は可変であってもよく、ゴロムーライス・パラメータに依存してもよい。もしそのパラメータが0である場合には、その範囲は8になる。パラメータ1に対しては、範囲は10である。パラメータ2の場合には、範囲は12であり、パラメータ3の場合には、範囲は16と等しくなる。この好適な実施形態では、ブロックの始点において、ゴロムーライス・パラメータはゼロに設定されてもよい。このゴロムーライス・パラメータの適応は、上述の等式(QQ)に説明したように実行される。パラメータはこの適応形態を使用することで減少できる訳ではないことに留意されたい。更に、例えば0から3への直接的な切り換えが可能であることにも留意されたい。この好適な実施形態においては、ゴ

40

50

ロムーライス符号の接頭部が確率モデルを使用するエントロピー符号を用いて符号化される。コンテキストモデリングはH. 264/AVCにおけるcoeff__abs__level__minus__twoについてと同様に実行され得る。

【0270】

他の好適な実施形態では、固定のゴロムーライス・パラメータが全ての変換係数レベルを現在の変換ブロック内で符号化するために使用されてもよい。この実施形態では、先行するブロックの最適なパラメータが計算され、現在の変換ブロックのために使用されてもよい。この実施形態に関しては、範囲は14で固定されてもよい。

【0271】

他の好適な実施形態では、固定のゴロムーライス・パラメータが全ての変換係数レベルを現在の変換ブロック内で符号化するために使用されてもよい。この実施形態では、先行するブロックの最適なパラメータが計算され、現在の変換ブロックのために使用されてもよい。この実施形態に関しては、範囲は上述のように可変でよい。

【0272】

他の好適な実施形態では、現在のスキャンインデックスの既に符号化（又は復号化）された隣接部が、前の境界よりも大きな絶対変換係数レベルを含むかどうかが評価される。この好適な実施形態では、最適なパラメータは局所的に因果関係のあるテンプレート内で隣接部を使用して導出されてもよい。

【0273】

従って、上述した（複数の）実施形態は、以下の符号化装置を特に記述する。即ち、シンタックス要素の少なくとも1つのサブグループをそれぞれn個のソースシンボル s_i ($i = 1 \dots n$) に個々に分解することによって、シンタックス要素列138をソースシンボル列106に変換するよう構成された分解手段136であって、ソースシンボルのそれぞれの数nは、それぞれのシンタックス要素の値範囲がサブ分割されていくn個のパーティション140_{1..3}の列のどのパーティションにそれぞれのシンタックス要素の値zが当てはまるかに依存しており、ソースシンボル s_i のそれぞれの数の値の合計がzであり、 $n > 1$ の場合に全ての $i = 1 \dots n-1$ について s_i の値がi番目のパーティションの範囲に対応している、分解手段136と、ソースシンボル列106をソースシンボルの第1サブ列108とソースシンボルの第2サブ列110へとサブ分割するよう構成されたサブ分割手段100であって、 $\{1 \dots n\}$ の第1サブセットの要素であるxを持つすべてのソースシンボル s_x が第1サブ列108の中に含まれ、第1サブセットとは共通元を持たない $\{1 \dots n\}$ の第2サブセットの要素であるyを持つすべてのソースシンボル s_y が第2サブ列110の中に含まれている、サブ分割手段100と、第1サブ列108のソースシンボルをシンボル毎に符号化するように構成されたVLC符号器102と、ソースシンボルの第2サブ列110を符号化するよう構成されたPIPE符号器又は算術符号器104と、を備えたエントロピー符号化装置である。

【0274】

シンタックス要素のサブグループの値zは絶対値でもよい。第2サブセットは $\{1\}$ であってもよく、 $p > q$ であるすべての $p, q \in \{1 \dots n\}$ について、p番目のパーティションがq番目のパーティションよりも前記値範囲のより高い値をカバーするように、n個のパーティションからなる前記列が配置されていてもよい。nは3でもよい。第1サブセットは $\{2, 3\}$ であってもよく、このときVLC符号器(102)は、ソースシンボル s_2 をシンボル毎に符号化するためにゴロムーライス符号を使用し、ソースシンボル s_3 をシンボル毎に符号化するために指数ゴロム符号を使用するよう構成されていてもよい。さらに一般的には、第1サブセットの要素は2であり、VLC符号器(102)は、ソースシンボル s_2 をシンボル毎に符号化するためにゴロムーライス符号を使用し、以前に符号化されたシンボルにしたがって、ゴロムーライス・パラメータ、即ちゴロムーライス符号のkを適応させるよう構成されていてもよい。分解手段は、以前に符号化されたソースシンボルにしたがって、パーティション間に1つ以上の限界を適用するよう構成されてもよい。両方の適用は組み合わせられていてもよい。つまり、第2のパーティションを限定す

10

20

30

40

50

る限界同士の位置は、ゴロムーライス符号の長さ、即ちその符号語の数が第2のパーティションの幅の長さに対応する（又は長さを規定する）ように、互いに間隔をおいて配置されるよう適応されてもよい。第1と第2のパーティションを分離する限界は、他のコンテキストに基づく他の方法に従って定義により定められるか適応されるなど、別のやり方で定義されてもよく、この場合には、 k を適応することにより、ゴロムーライス符号の長さと第2のパーティションの幅とをそれぞれ介して、第2と第3のパーティションを分離する限界の位置を定義し得る。第2のパーティションがゴロムーライス符号の長さに対応するように k を適応することを組み合わせることで、符号効率は最適に使用される。シンタックス要素の統計に k を適応することにより、第3のパーティションができるだけ多くをカバーできるように、第2のパーティションの幅を適応させることが可能になる。これは、指数ゴロム符号など複雑性の低い符号が第3のパーティションに使用される可能性があるため、全体の符号化の複雑性を低減させるためである。更に、第1のパーティションの長さは、最も低い3つのレベルなど、 $j \in \{1, 2, 3\}$ の可能なシンタックス要素値に限定されてもよい。考慮対象となるシンタックス要素は、上段で例示した予測残余を表す変換係数レベルの場合のように、差分的に符号化されるか、又は予測残余を表現してもよい。第1のソースシンボル s_1 は、切り詰めされたユニナリ符号を使用してシンボル化／逆シンボル化されてもよく、ここで、結果として得られる j 個のビンは、部分的に又はそれらの全てがコンテキスト適応型で又は上述していない方法で符号化されてもよい。

【0275】

シンタックス要素のサブグループは、あるピクチャの変換ブロックの絶対変換係数の絶対変換係数レベルを包含してもよく、各変換ブロックの絶対変換係数レベルは、各変換ブロックの絶対変換係数を通過して進むスキャン経路に従ってシンタックス要素の列（138）の内部に配置されており、ここで、分解手段は、各変換ブロックの絶対変換係数の絶対変換係数レベルを分解する間に、パーティション同士の間に1つ以上の限界を適応するよう構成されており、その適応は、スキャン順序において先行する各変換ブロックの絶対変換係数の既に符号化済みの絶対変換係数レベルに基づくか、スキャン順序において現在分解されるべき絶対変換係数レベルの位置に基づくか、又は、現在分解されるべき絶対変換係数レベルの位置に対して空間的又はスキャン順序において隣接している変換係数の既に復元された絶対変換係数レベルの評価に基づいてもよい。

【0276】

さらに、上述した実施形態は、以下の復号化装置を特に記述する。即ち、第1ビットストリーム206の符号語からソースシンボルの第1サブ列204のソースシンボルを符号語毎に復元するよう構成されたVLC復号器200と、ソースシンボルの第2サブ列208を復元するよう構成されたPIPE復号器又は算術復号器202と、それぞれの数のソースシンボルから各シンタックス要素を個別に組み立てることによって、ソースシンボルの第1サブ列204とソースシンボルの第2サブ列208とからシンタックス要素列226を組み立てるよう構成された組立手段224と、を含むエントロピー復号化装置である。組立手段224は、シンタックス要素の少なくとも1つのサブグループについて、それぞれのシンタックス要素の値範囲がサブ分割されていく n 個のパーティション 140_{1-3} からなる列のうちのどのパーティションにそれぞれのシンタックス要素の値 z が当てはまるかに依存して、 s_i の値が i 番目のパーティションの範囲に対応している間ではシンタックス要素 z の値を得るために1から n までソースシンボル s_i のそれぞれの数の値を合計することによって、 $i = 1 \dots n$ のソースシンボル s_i のそれぞれの数 n を決定するよう構成される。組立手段224は、第1サブ列204から $\{1 \dots n\}$ の第1サブセットの要素である x を持つ全てのソースシンボル s_x と、第2サブ列208から第1サブセットとは共通元を持たない $\{1 \dots n\}$ の第2サブセットの要素である y を持つ全てのソースシンボル s_y とを回収するよう構成されている。シンタックス要素のサブグループの値 z は絶対値であってもよい。第2サブセットは、 $p > q$ であるすべての $p, q \in \{1 \dots n\}$ について、 p 番目のパーティションが q 番目のパーティションよりも前記値範囲のより高い値をカバーするように、 n 個のパーティションの列を持つ $\{1\}$ であっても

よい。 n は3であってもよい。第1サブセットは $\{2, 3\}$ であり、VLC復号器200は、ソースシンボル s_2 を符号語毎に復元するためにゴロム・ライス符号を使用し、ソースシンボル s_3 を符号語毎に復元するために指数ゴロム符号を使用するよう構成されてもよい。さらに一般的には、第1のサブセットの要素は2であり、VLC復号器200は、前記ソースシンボル s_2 を符号語毎に復号化するためにゴロム・ライス符号を使用し、以前に復元されたソースシンボルにしたがって、前記ゴロム・ライス符号のゴロム・ライスパラメータを適用するよう構成されてもよい。このエントロピー復号化装置は、ソースシンボルの第1サブ列(204)とソースシンボルの第2サブ列とを再結合し、ソースシンボル列(218)を得るよう構成された、再結合部220をさらにそなえてもよい。シンタックス要素は異なるタイプからなり、組立手段はシンタックス要素のタイプに基づいて個別の組立を実行するよう構成されてもよい。シンタックス要素のサブグループは、ピクチャの変換ブロックの絶対変換係数の絶対変換係数レベルを包含しており、それぞれの変換ブロックの絶対変換係数レベルは、それぞれの変換ブロックの絶対変換係数を通過して進むスキャン経路にしたがってシンタックス要素列(138)内に配置されており、このとき組立部は、スキャン順序において先行するそれぞれの変換ブロックの絶対変換係数の既に復元された絶対変換係数レベルに基づいて、又はスキャン順序において現時点で組み立てられるべき絶対変換係数レベルの位置に基づいて、又は現時点で組み立てられるべき絶対変換係数レベルの空間的又はスキャン順序的に隣接する位置の変換係数の既に復元された絶対変換係数レベルの評価に基づいて、それぞれの変換ブロックの絶対変換係数の絶対変換係数レベルを組み立てている間に、前記パーティション間に1つ又はそれ以上の限界を適用するよう構成されてもよい。

10

20

【0277】

PIPE符号化とVLC符号化とを結合し、図1bに示す分解部を使用する実施形態に関し、以下の記載によりその幾つかの態様について繰り返し説明する。

【0278】

シンボル列からビットストリームへのマッピングと逆マッピングについてこれまで説明した。各シンボルはそれ自身と共に、関連するパラメータを運ぶものであり、それらは符号器と復号器とに対して同時に知られている。エントロピーコーデックは多数の先入れ先出し(FIFO)バッファを含み、それらバッファの各々が、シンボルに関連付けられたパラメータのサブセットに割り当てられている。あるシンボルの所与の(複数の)パラメータに関し、符号器はそのシンボルを対応するFIFOバッファへと割り当てる。符号器の割り当て規則は復号器側でも知られているため、復号器は、符号器がシンボルを割り当てたFIFOバッファから読み出しを実行する。

30

【0279】

幾つかのシンタックス要素は標準的な可変長符号を使用して符号化され、特定のバッファに書き込まれる。他のシンタックス要素は、確率区間区分エントロピー(PIPE)符号化の概念を用いて符号化される。この場合、シンボルは最初に二値化され、結果として得られるピンは関連する確率推定に基づいて分類される。確率推定は、符号器と復号器とにおいて同時に実行され得るような測度から与えられるか又は導出されてもよい。特定のFIFOバッファは、選択された確率のサブセット内に当てはまる推定された確率の値を有するシンボルを含み、その結果、エントロピー符号化が改善され得る。PIPEの概念とVLCとの組み合わせによって達成されるこの改善は、複雑性を低減する一方で、なお高い符号化効率を提供するということである。標準的なVLC符号が適合するシンボルは、単純で複雑性の低いVLCの手法により符号化され、他方、VLC符号を用いて符号化することでビットレートがかなり増大するであろう他のシンボルは、より洗練されたPIPEの概念を用いて符号化される。

40

【0280】

従って、エントロピー符号化の複雑性をさらに低減するために、シンボルは2つのカテゴリーへと分離されてきた。第1のカテゴリーのシンボルは、VLC符号を用いて適切に表現されることができ、より複雑なPIPE符号化は必要としない。他方、第2のカテゴリ

50

一のシンボルは、V L C 符号を用いて効率的に表現されることは不可能であり、これらのシンボルを P I P E 符号化することで、必要なビットレートがかなり低減された。

【0281】

上述の実施形態は1つのV L C 符号器及び復号器だけを用いて説明してきたが、しかしこの概念は、2つ以上のV L C 符号器及び復号器が使用されてP I P E 符号器と一緒にインターリーブされる場合にも一般化できることは明白であることに留意されたい。

【0282】

これまで装置を説明する文脈で幾つかの態様を示してきたが、これらの態様は対応する方法の説明でもあることは明らかであり、そのブロック又は装置が方法ステップ又は方法ステップの特徴に対応することは明らかである。同様に、方法ステップを説明する文脈で示した態様もまた、対応する装置の対応するブロックもしくは項目又は特徴を表している。方法ステップの幾つか又は全ては、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、又は電子回路等のハードウェアにより（を使用して）実行されても良い。幾つかの実施形態においては、最も重要な方法ステップの内の1つ又は複数のステップはそのような装置によって実行されても良い。

【0283】

本発明の符号化済み／圧縮済みの信号は、デジタル記憶媒体に記憶されても良く、インターネットのような無線伝送媒体や有線伝送媒体などの伝送媒体によって伝送されても良い。

【0284】

所定の構成要件にも依るが、本発明の実施形態は、ハードウェア又はソフトウェアにおいて構成可能である。この構成は、その中に格納される電子的に読み取り可能な制御信号を有し、本発明の各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（又は協働可能な）、デジタル記憶媒体、例えばフレキシブルディスク、DVD、ブルーレイ、CD、ROM、PROM、EPROM、EEPROM、フラッシュメモリなどを使用して実行することができる。従って、そのデジタル記憶媒体はコンピュータ読み取り可能であっても良い。

【0285】

本発明に従う幾つかの実施形態は、上述した方法の1つを実行するようプログラム可能なコンピュータシステムと協働可能で、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んでも良い。

【0286】

一般的に、本発明の実施例は、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構成することができ、このプログラムコードは当該コンピュータプログラム製品がコンピュータ上で作動するときに、本発明の方法の一つを実行するよう作動する。そのプログラムコードは例えば機械読み取り可能なキャリアに記憶されても良い。

【0287】

本発明の他の実施形態は、上述した方法の1つを実行するための、機械読み取り可能なキャリアに記憶されたコンピュータプログラムを含む。

【0288】

換言すれば、本発明の方法のある実施形態は、そのコンピュータプログラムがコンピュータ上で作動するときに、上述した方法の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。

【0289】

本発明の他の実施形態は、上述した方法の1つを実行するために記録されたコンピュータプログラムを含む、データキャリア（又はデジタル記憶媒体又はコンピュータ読み取り可能な媒体）である。

【0290】

本発明の他の実施形態は、上述した方法の1つを実行するためのコンピュータプログラムを表現するデータストリーム又は信号列である。そのデータストリーム又は信号列は、例

10

20

30

40

50

例えばインターネットを介するデータ通信接続を介して伝送されるように構成されても良い。

【0291】

他の実施形態は、上述した方法の1つを実行するように構成又は適用された、例えばコンピュータ又はプログラム可能な論理デバイスのような処理手段を含む。

【0292】

他の実施形態は、上述した方法の1つを実行するためのコンピュータプログラムがインストールされたコンピュータを含む。

【0293】

幾つかの実施形態においては、（例えば書換え可能ゲートアレイのような）プログラム可能な論理デバイスは、上述した方法の幾つか又は全ての機能を実行するために使用されても良い。幾つかの実施形態では、書換え可能ゲートアレイは、上述した方法の1つを実行するためにマイクロプロセッサと協働しても良い。一般的に、そのような方法は、好適には任意のハードウェア装置によって実行される。

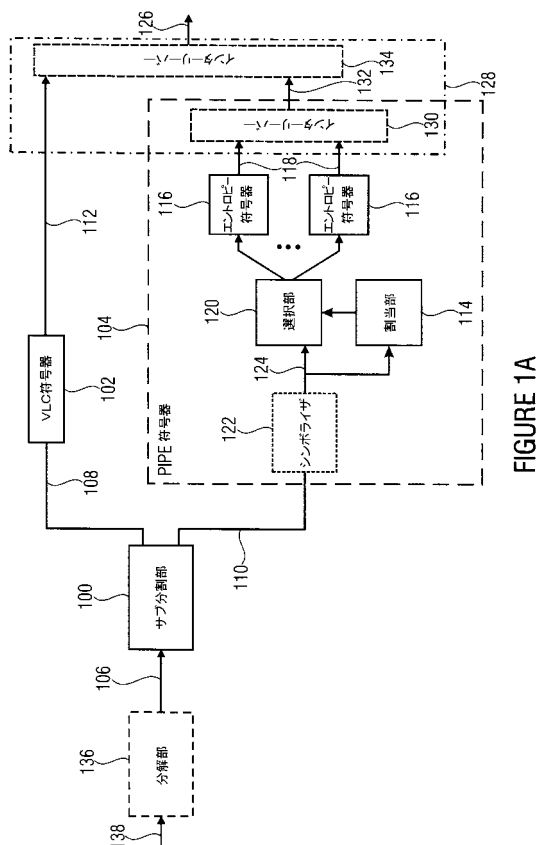
【0294】

上述した実施形態は、本発明の原理を単に例示的に示したにすぎない。本明細書に記載した構成及び詳細について修正及び変更が可能であることは、当業者にとって明らかである。従って、本発明は、本明細書に実施形態の説明及び解説の目的で提示した具体的詳細によって限定されるものではなく、添付した特許請求の範囲によってのみ限定されるべきである。

10

20

【図1a】



【図1b】

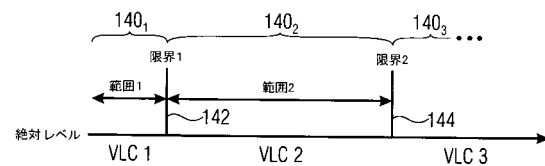


FIGURE 1B

【図 4】

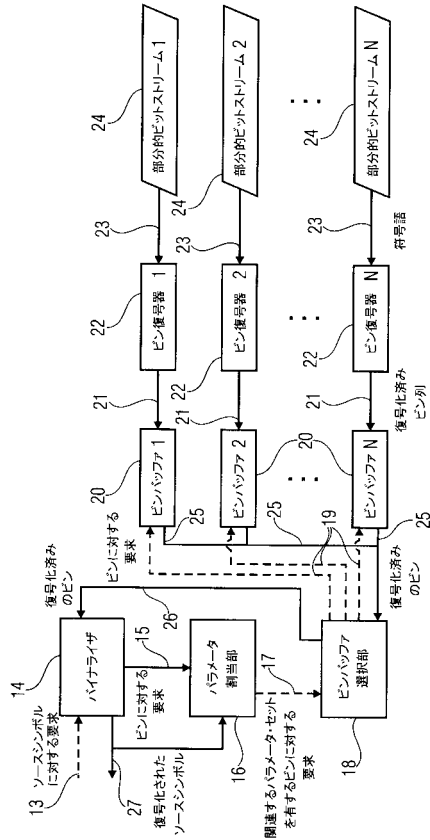


FIGURE 4

【図 5】

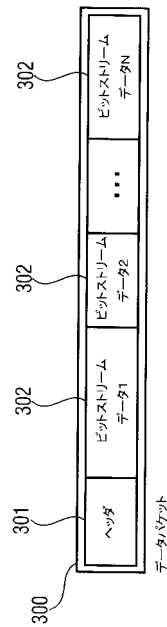


FIGURE 5

【图 6】

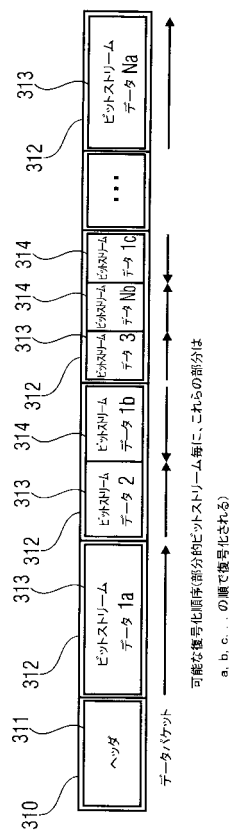


FIGURE 6

【図 7】

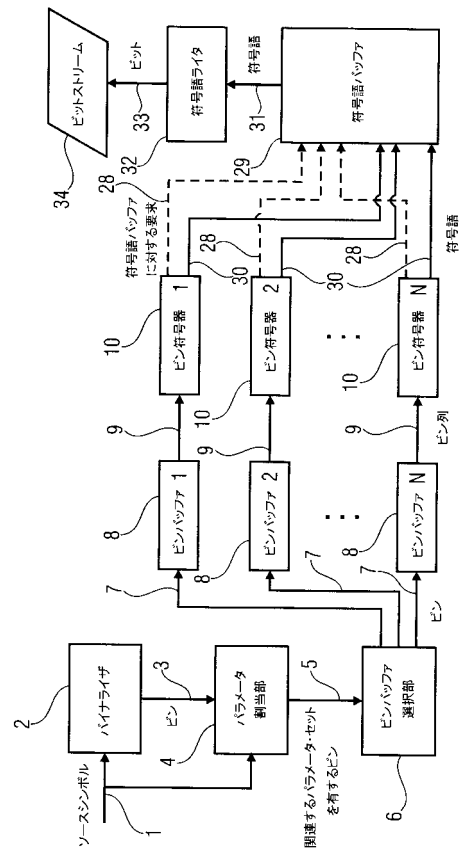


FIGURE 7

【 図 8 】

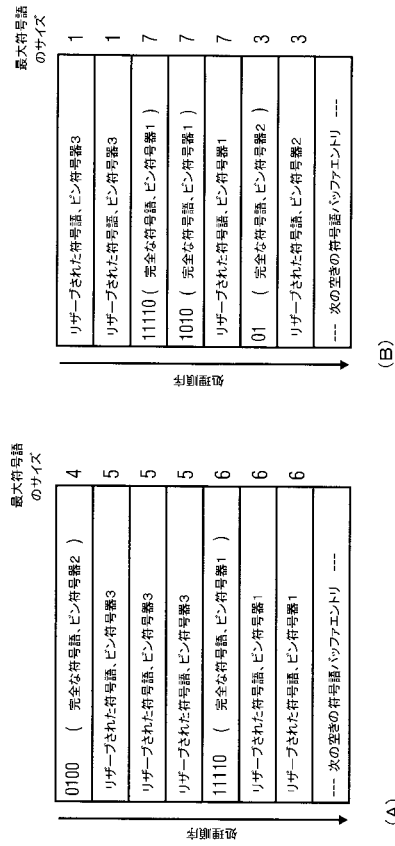


FIGURE 8

【 図 9 】

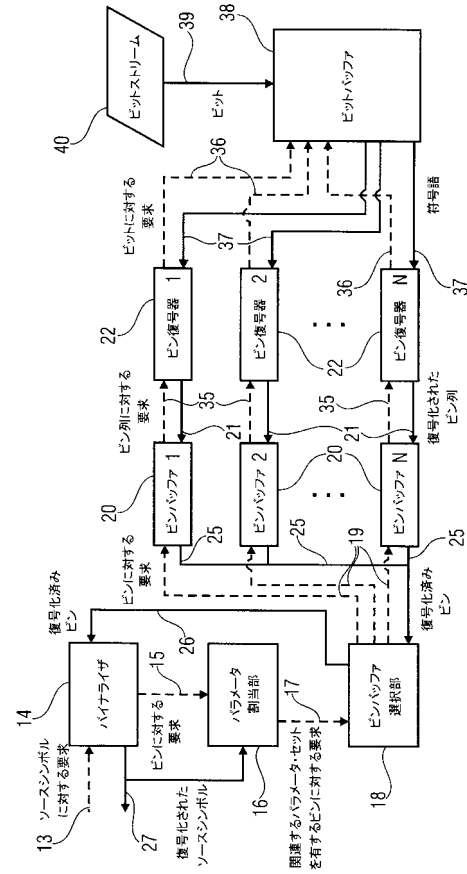


FIGURE 9

【 図 1 0 】

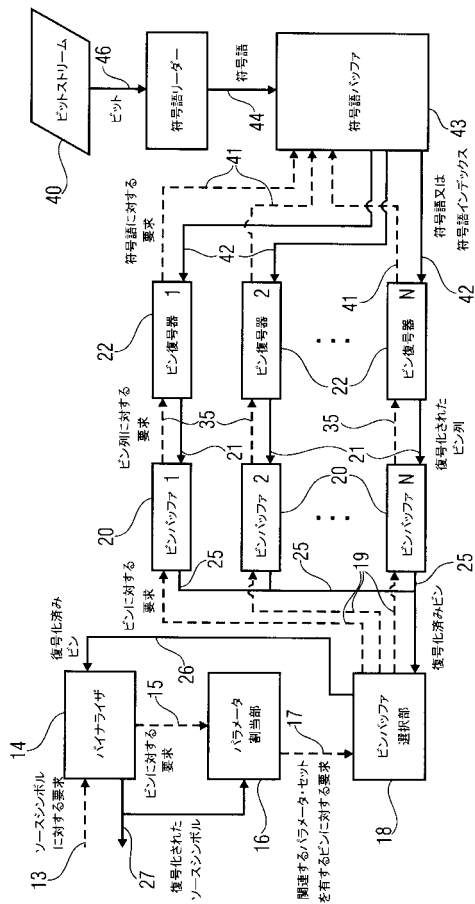


FIGURE 10

【 図 1 1 】

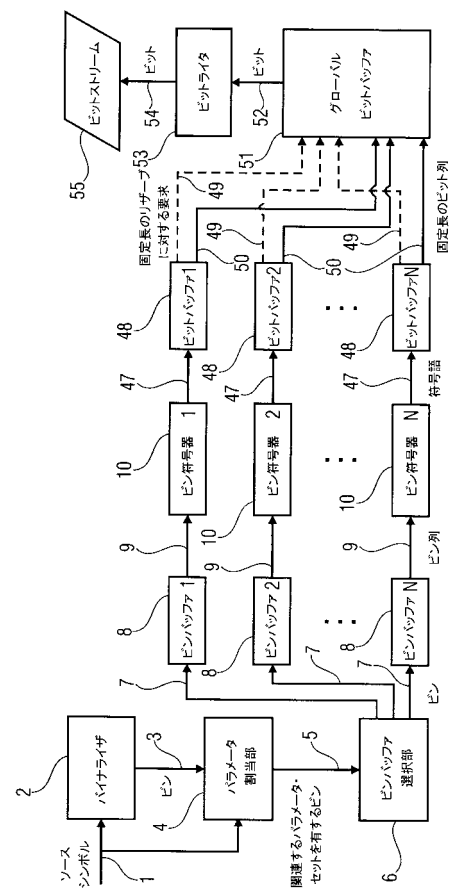


FIGURE 11

【図 1 2】

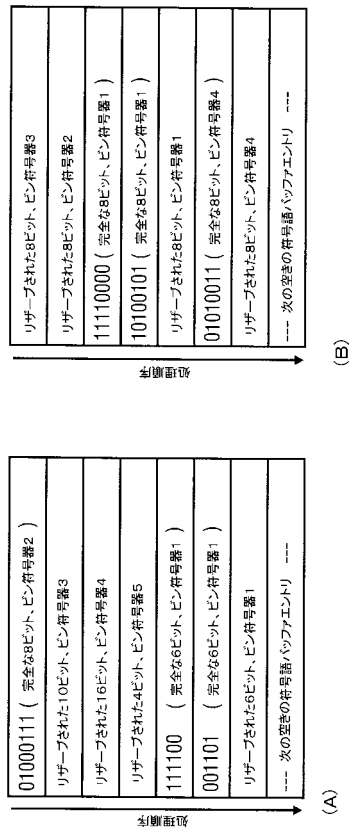


FIGURE 12

【図 1 3】

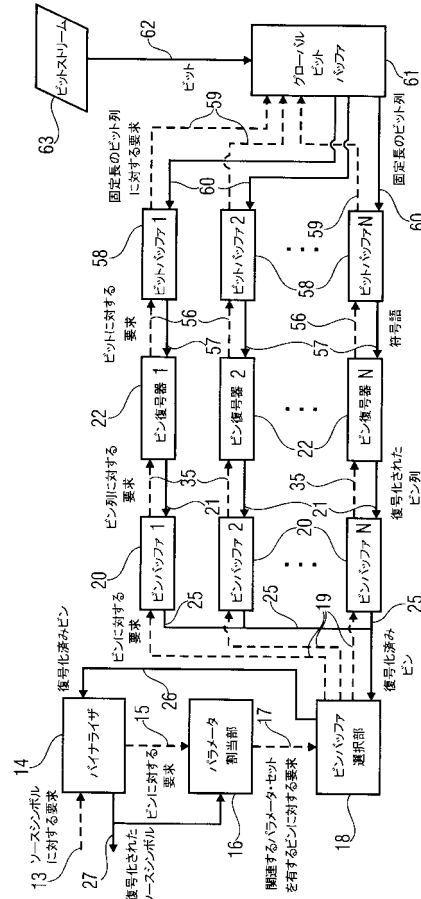


FIGURE 13

【図 1 4】

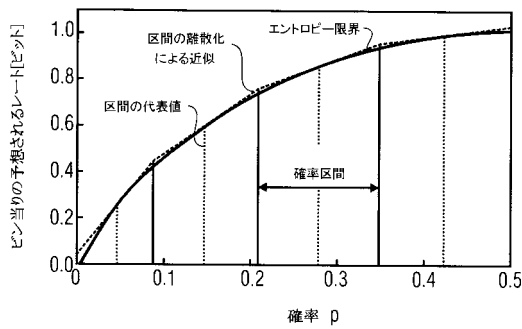


FIGURE 14

【図 1 6】

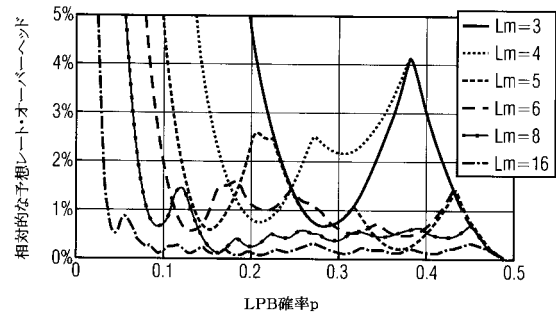


FIGURE 16

【図 1 5】

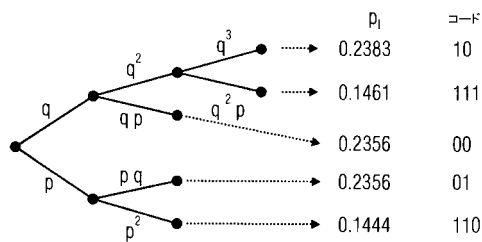


FIGURE 15

【図 1 7】

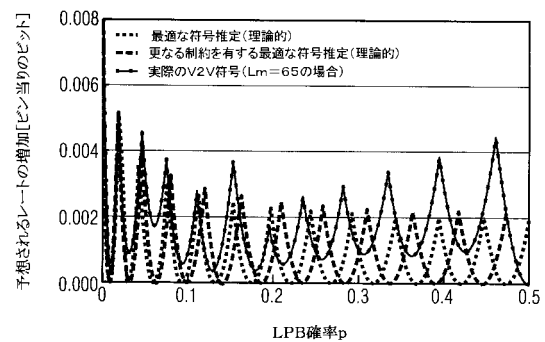


FIGURE 17

【 図 1 8 】

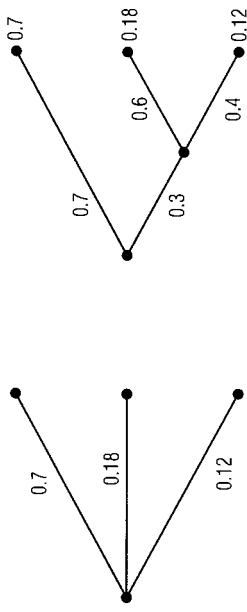


FIGURE 18

【 図 1 9 】

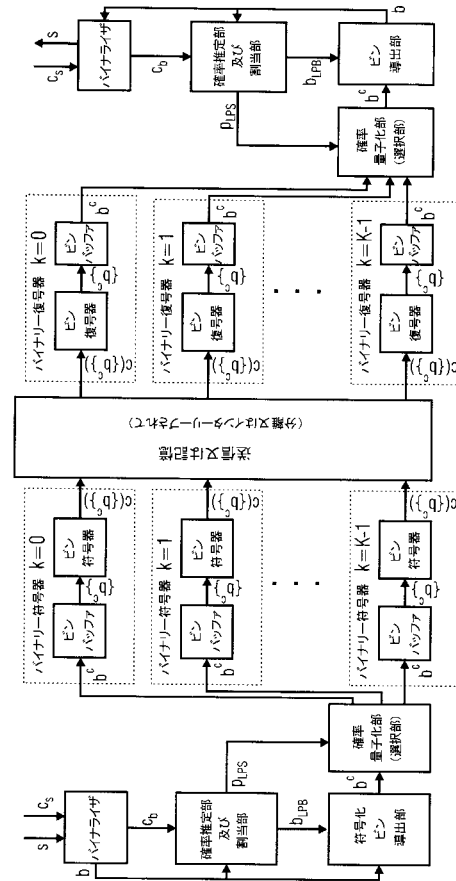


FIGURE 19

【图 20】

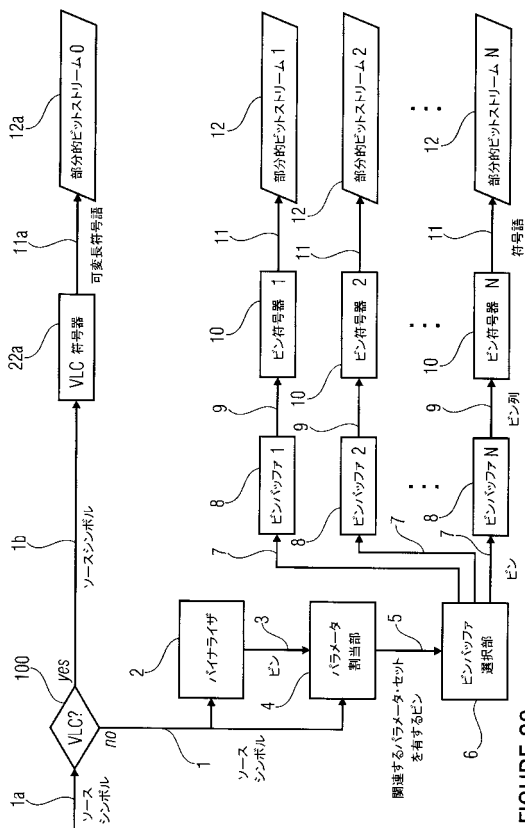


FIGURE 20

【 図 2 1 】

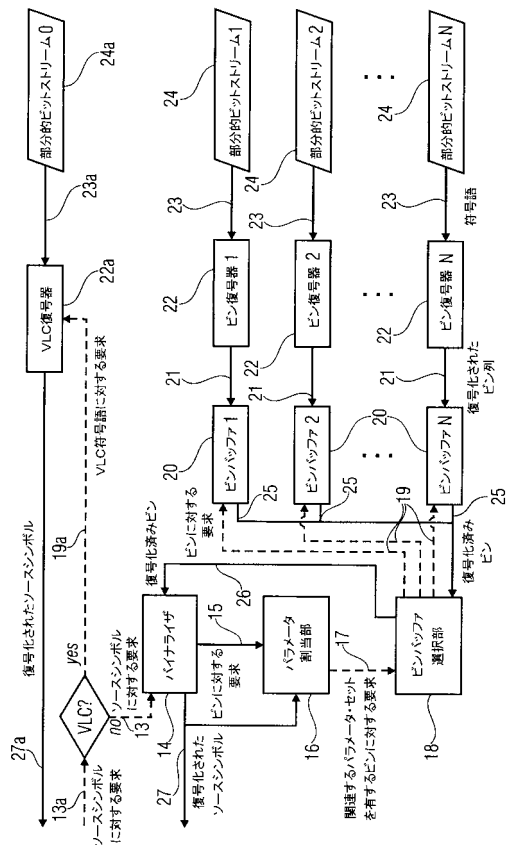


FIGURE 21

【図 2 2】

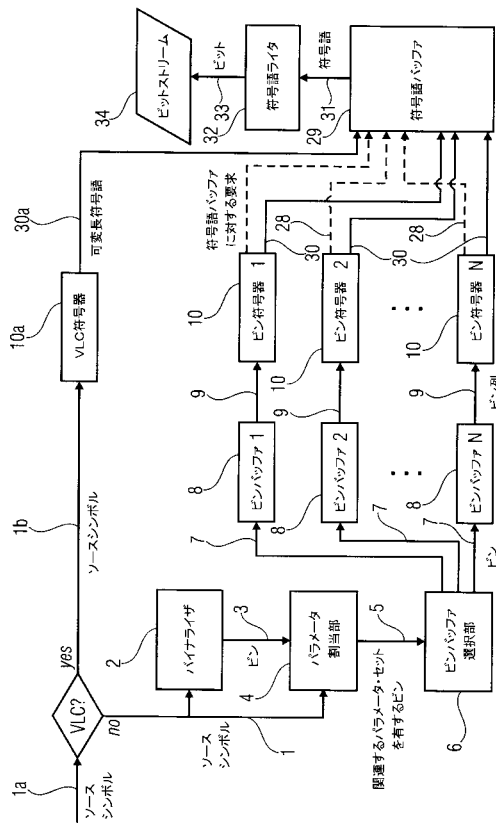


FIGURE 22

【図 2 4】

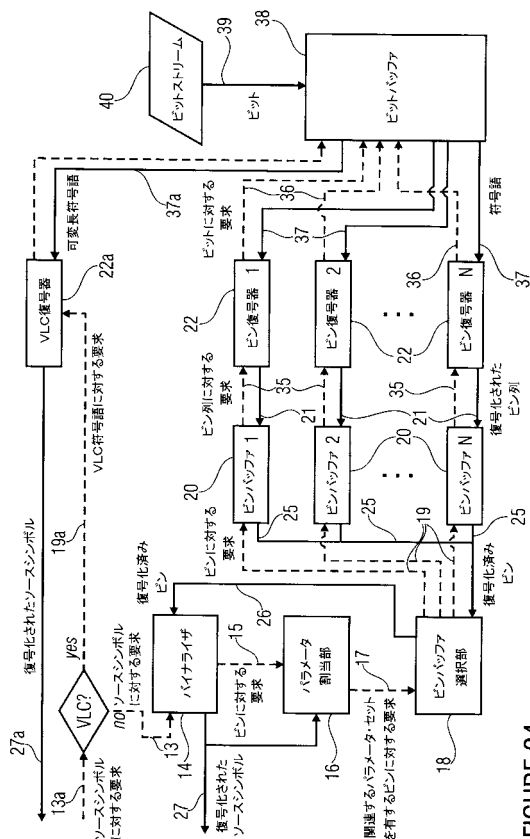


FIGURE 24

【図 2 3】

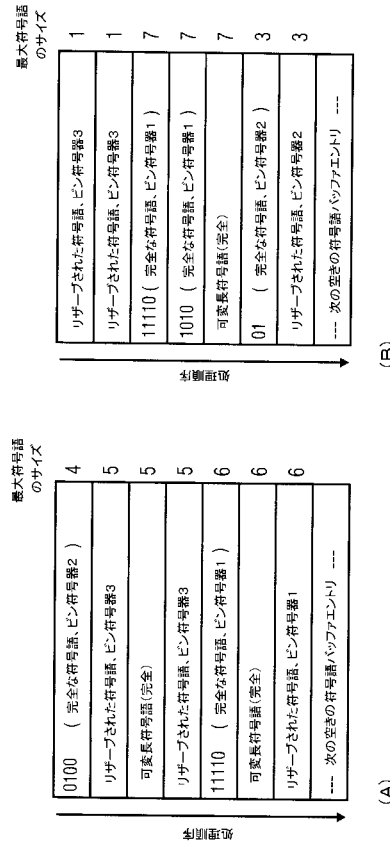


FIGURE 23

【手続補正書】

【提出日】平成29年1月16日(2017.1.16)

【手続補正 1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

復号化方法であって、

第 1 復号器によって、ビットストリームの第 1 部分を受信するステップと、
前記第 1 復号器によって、前記ビットストリームの第 1 部分の符号語からソースシンボルの第 1 列のソースシンボルを復元するステップと、
第 2 復号器によって、前記ビットストリームの第 2 部分を受信するステップと、
前記第 2 復号器によって、前記ビットストリームの第 2 部分の符号語からソースシンボルの第 2 列のソースシンボルを復元するステップと、
組立手段によって、前記第 1 列からのソースシンボルと前記第 2 列からのソースシンボルとを回収するステップと、
前記組立手段によって、それぞれの数のソースシンボルから各シンタックス要素を組み立てることによって、前記ソースシンボルの第 1 列と前記ソースシンボルの第 2 列とからパーティションの列へとサブ分割されるある値範囲を持つシンタックス要素列を組み立て、その組立の際に、前記シンタックス要素の少なくとも 1 つのサブグループについて、個々のシンタックス要素の値と前記パーティションの列の 1 つのパーティションとに基づいて前記ソースシンボルのそれぞれの数を決定し、前記シンタックス要素の値を得るために前記それぞれの数のソースシンボルの値を合計する、ステップと、を含み、
前記組立手段は、以前に復元されたソースシンボルに従って、前記パーティション間に 1 つ以上の境界を適応するよう構成されている、復号化方法。

【請求項 2】

前記第 1 列からのソースシンボルと前記第 2 列からのソースシンボルとは異なるパーティションに対応している、請求項 1 に記載の復号化方法。

【請求項 3】

前記ソースシンボルの第 2 列の中のソースシンボルは前記パーティションの列の中の 1 番目のパーティションに対応し、
前記パーティションの列は、第 1 パーティションが第 2 パーティションよりも値範囲のより高い値をカバーするように配置されている、請求項 1 又は 2 に記載の復号化方法。

【請求項 4】

前記ビットストリームの第 3 部分は少なくとも 1 つのスケールファクタを表している、請求項 1 乃至 3 のいずれか 1 項 に記載の復号化方法。

【請求項 5】

前記ソースシンボルの第 1 列の中のソースシンボルは 2 番目のパーティションに対応し、
前記第 1 復号器は前記ソースシンボルの第 1 列の中のソースシンボルを復号化するためにゴロム・ライス符号を使用するよう構成されている、請求項 3 に記載の復号化方法。

【請求項 6】

再結合器により、前記ソースシンボルの第 1 列と前記ソースシンボルの第 2 列とを再結合して、ソースシンボル列を得るステップを更に含む、請求項 1 乃至 5 のいずれか 1 項 に記載の復号化方法。

【請求項 7】

前記第 1 復号器によって復元するステップは、前記ビットストリームの第 1 部分の符号語からソースシンボルの第 1 列のソースシンボルを復元することにより実行される、請求項 1 乃至 6 のいずれか 1 項 に記載の復号化方法。

【請求項 8】

前記組立手段は、スキャン順序において先行するそれぞれの変換ブロックの絶対変換係数の少なくとも 1 つの復元された絶対変換係数レベルに基づいて、又は組み立てられるべき絶対変換係数レベルのスキャン順序における位置に基づいて、又は組み立てられるべき絶対変換係数レベルの空間的に隣接する位置の変換係数の復元された絶対変換係数レベルの評価に基づいて、又は組み立てられるべき絶対変換係数レベルのスキャン順序において隣接する位置の変換係数の復元された絶対変換係数レベルの評価に基づいて、それぞれの変換ブロックの絶対変換係数の絶対変換係数レベルを組み立てしている間に、前記パーティション間に少なくとも 1 つの境界を適応するよう構成されている、請求項 1 乃至 7 のいずれか 1 項に記載の復号化方法。

【請求項 9】

前記シンタックス要素列はビデオをスケーラブルな方式で符号化している、請求項 1 乃至 8 のいずれか 1 項に記載の復号化方法。

【請求項 10】

コンピュータ上で実行したとき、請求項 1 乃至 9 のいずれか 1 項に記載の復号化方法を実行するための、プログラムコードを有するコンピュータプログラム。

【請求項 11】

復号化装置であって、
ビットストリームの第 1 部分を受信し、前記ビットストリームの第 1 部分の符号語からソースシンボルの第 1 列のソースシンボルを復元するよう構成された第 1 復号器と、
前記ビットストリームの第 2 部分を受信し、前記ビットストリームの第 2 部分の符号語からソースシンボルの第 2 列を復元するよう構成された第 2 復号器と、
組立手段であって、前記第 1 列からのソースシンボルと前記第 2 列からのソースシンボルとを回収し、それぞれの数のソースシンボルから各シンタックス要素を組み立てることによって、前記ソースシンボルの第 1 列と前記ソースシンボルの第 2 列とからパーティションの列へとサブ分割されるある値範囲を持つシンタックス要素列を組み立て、その組立の際に、前記シンタックス要素の少なくとも 1 つのサブグループについて、個々のシンタックス要素の値と前記パーティションの列の 1 つのパーティションとに基づいて前記ソースシンボルのそれぞれの数を決定し、前記シンタックス要素の値を得るために前記それぞれの数のソースシンボルの値を合計する、組立手段と、を含み、
前記組立手段は、以前に復元されたソースシンボルに従って、前記パーティション間に 1 つ以上の境界を適応するよう構成されている、復号化装置。

【請求項 12】

前記第 1 列からのソースシンボルと前記第 2 列からのソースシンボルとは異なるパーティションに対応している、請求項 11 に記載の復号化装置。

【請求項 13】

前記ソースシンボルの第 2 列の中のソースシンボルは前記パーティションの列の中の 1 番目のパーティションに対応し、
前記パーティションの列は、第 1 パーティションが第 2 パーティションよりも値範囲のより高い値をカバーするように配置されている、請求項 11 又は 12 に記載の復号化装置。

【請求項 14】

前記パーティションの数は 1、2、又は 3 である、請求項 11 乃至 13 のいずれか 1 項に記載の復号化装置。

【請求項 15】

前記ビットストリームの第 3 部分は少なくとも 1 つのスケールファクタを表している、請求項 11 乃至 14 のいずれか 1 項に記載の復号化装置。

【請求項 16】

前記装置は、プログラム可能な論理デバイス、プログラム可能なゲートアレイ、マイクロプロセッサ、コンピュータ又は電子回路の少なくとも一部である、請求項 11 乃至 15 のいずれか 1 項に記載の復号化装置。

【請求項 17】

前記第 1 復号器は、前記ビットストリームの第 1 部分の符号語からソースシンボルの第 1 列のソースシンボルを復元することにより、前記第 1 列のソースシンボルを復元するよう構成されている、請求項 11 乃至 16 のいずれか 1 項に記載の復号化装置。

【請求項 18】

前記組立手段は、スキャン順序において先行するそれぞれの変換ブロックの絶対変換係数の少なくとも 1 つの復元された絶対変換係数レベルに基づいて、又はスキャン順序において組み立てられるべき絶対変換係数レベルの位置に基づいて、又は組み立てられるべき絶対変換係数レベルの空間的に隣接する位置の変換係数の復元された絶対変換係数レベルの評価に基づいて、又は組み立てられるべき絶対変換係数レベルのスキャン順序において隣接する位置の変換係数の復元された絶対変換係数レベルの評価に基づいて、それぞれの変換ブロックの絶対変換係数の絶対変換係数レベルを組み立てしている間に、前記パーティション間に少なくとも 1 つの境界を適応するよう構成されている、請求項 11 乃至 17 のいずれか 1 項に記載の復号化装置。

【請求項 19】

前記シンタックス要素列はビデオをスケーラブルな方式で符号化している、請求項 11 乃至 18 のいずれか 1 項に記載の復号化装置。

【請求項 20】

符号化装置であって、

パーティションの列にサブ分割される値範囲を持つシンタックス要素列を受信し、個々のシンタックス要素の値と前記パーティションの列の 1 つのパーティションとに基づいて、ソースシンボルの値の合計が個々のシンタックス要素の値となるようにシンタックス要素の少なくとも 1 つのサブグループを対応する数のソースシンボルに分解することにより、前記シンタックス要素列をソースシンボル列に変換する、分解手段と、

前記ソースシンボル列を受信し、前記ソースシンボル列をソースシンボルの第 1 列とソースシンボルの第 2 列へとサブ分割するサブ分割手段と、

前記第 1 列のソースシンボルを受信し、前記第 1 列のソースシンボルを符号化する第 1 符号器と、

前記第 2 列のソースシンボルを受信し、前記第 2 列のソースシンボルを符号化する第 2 符号器と、を含み、

前記分解手段は、以前に符号化されたソースシンボルに従って、前記パーティション間に 1 つ以上の境界を適応するよう構成されている、符号化装置。

【請求項 21】

前記第 1 符号器又は第 2 符号器は、さらにスケールファクタを符号化するよう構成されている、請求項 20 に記載の符号化装置。

【請求項 22】

前記第 1 列からのソースシンボルと前記第 2 列からのソースシンボルとは異なるパーティションに対応している、請求項 20 又は 21 に記載の符号化装置。

【請求項 23】

前記ソースシンボルの第 2 列の中のソースシンボルは、前記パーティションの列の中の 1 番目のパーティションに対応しており、

前記パーティションの列は、第 1 パーティションが第 2 パーティションよりも値範囲のより高い値をカバーするように配置されている、請求項 20 乃至 22 のいずれか 1 項に記載の符号化装置。

【請求項 24】

前記ソースシンボルの第 1 列の中のソースシンボルは 2 番目のパーティションに対応し、前記第 1 符号器は前記ソースシンボルの第 1 列の中のソースシンボルをシンボル毎に符号化するためにゴロムーライス符号を使用するよう構成されている、請求項 20 に記載の符号化装置。

【請求項 25】

前記装置は、プログラム可能な論理デバイス、プログラム可能なゲートアレイ、マイクロプロセッサ、コンピュータ又は電子回路の少なくとも一部である、請求項 20 乃至 24 のいずれか 1 項に記載の符号化装置。

【請求項 26】

前記シンタックス要素列はビデオをスケーラブルな方式で符号化している、請求項 20 乃至 25 のいずれか 1 項に記載の符号化装置。

【請求項 27】

以下のステップを含む、符号化方法：

パーティションの列にサブ分割される値範囲を持つシンタックス要素列を受信するステップ、

個々のシンタックス要素の値と前記パーティションの列の 1 つのパーティションとに基づいて、ソースシンボルの値の合計が個々のシンタックス要素の値となるように前記シンタックス要素の少なくとも 1 つのサブグループを対応する数のソースシンボルに分解することにより、前記シンタックス要素列をソースシンボル列に変換するステップ、

前記ソースシンボル列をソースシンボルの第 1 列とソースシンボルの第 2 列へとサブ分割するステップ、

第 1 符号器によって前記第 1 列のソースシンボルを受信し、前記第 1 列のソースシンボルを符号化するステップ、

第 2 符号器によって前記第 2 列のソースシンボルを受信し、前記第 2 列のソースシンボルを符号化するステップ、

以前に符号化されたソースシンボルに従って、前記パーティション間に 1 つ以上の境界を適応するステップ。

【請求項 28】

コンピュータ上で実行したとき、請求項 27 に記載の符号化方法を実行するための、プログラムコードを有するコンピュータプログラム。

【請求項 29】

以下のステップを含む方法によって符号化されたデータストリーム：

パーティションの列にサブ分割される値範囲を持つシンタックス要素列を受信するステップ、

個々のシンタックス要素の値と前記パーティションの列の 1 つのパーティションとに基づいて、ソースシンボルの値の合計が個々のシンタックス要素の値となるように前記シンタックス要素の少なくとも 1 つのサブグループを対応する数のソースシンボルに分解することにより、前記シンタックス要素列をソースシンボル列に変換するステップ、

前記ソースシンボル列をソースシンボルの第 1 列とソースシンボルの第 2 列へとサブ分割するステップ、

第 1 符号器によって前記第 1 列のソースシンボルを受信し、前記第 1 列のソースシンボルを符号化するステップ、

第 2 符号器によって前記第 2 列のソースシンボルを受信し、前記第 2 列のソースシンボルを符号化するステップ、

以前に符号化されたソースシンボルに従って、前記パーティション間に 1 つ以上の境界を適応するステップ。

【請求項 30】

前記シンタックス要素列はビデオをスケーラブルな方式で符号化している、請求項 29 に記載のデータストリーム。

フロントページの続き

(72)発明者 シュヴァルツ, ハイコ

ドイツ連邦共和国 1 3 1 8 7 ベルリン ノイエ シェーンホルツァー ストラーセ 1 2

(72)発明者 ヴィーガンツ, トーマス

ドイツ連邦共和国 1 4 1 9 5 ベルリン オットー-アッペル-ストラーセ 5 2

Fターム(参考) 5C159 KK13 ME01 ME11 ME17 RB02 TA59 UA02 UA05

5J064 AA02 BA09 BA10 BB05 BC14