(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6195997号
(P6195997)
(45)発行日　平成29年9月13日(2017.9.13)　(24)登録日　平成29年8月25日(2017.8.25)

(51)Int.Cl.　F I
H03M 7/42 (2006.01)　H03M 7/42

請求項の数 26　(全 35 頁)

(21)出願番号　特願2016-549759 (P2016-549759)
(86)(22)出願日　平成27年2月20日 (2015.2.20)
(65)公表番号　特表2017-507590 (P2017-507590A)
(43)公表日　平成29年3月16日 (2017.3.16)
(86)国際出願番号　PCT/EP2015/025008
(87)国際公開番号　WO2015/124324
(87)国際公開日　平成27年8月27日 (2015.8.27)
審査請求日　平成28年9月21日 (2016.9.21)
(31)優先権主張番号　1403039.9
(32)優先日　平成26年2月20日 (2014.2.20)
(33)優先権主張国　英国 (GB)

早期審査対象出願

(73)特許権者　513156386
グルロジック　マイクロシステムズ　オーワイ
Ｇｕｒｕｌｏｇｉｃ　Ｍｉｃｒｏｓｙｓｔｅｍｓ　Ｏｙ
フィンランド共和国　２０１００　トゥルク　リンナンカツ　３４
Ｌｉｎｎａｎｋａｔｕ　３４　２０１００　Ｔｕｒｋｕ　ＦＩＮＬＡＮＤ
(74)代理人　100127188
弁理士　川守田　光紀
(72)発明者　カレヴォ　オッシ
フィンランド共和国　ＦＩＮ－３７８００　アカー　ケトゥンハンタ　１

最終頁に続く

(54)【発明の名称】シンボル圧縮を伴うデータのソース符号化・復号方法及び装置

(57)【特許請求の範囲】
【請求項１】
入力データ（Ｄ１）をエンコーダ（５０）で符号化して対応する符号化データ（Ｅ２）を生成する方法であって、前記方法は：
（ａ）前記入力データ（Ｄ１）に存在するシンボルを解析し、該入力データ（Ｄ１）を１つ又は複数のデータチャンクに分割及び／又は変換することと；
（ｂ）前記１つ又は複数のデータチャンクに存在する前記シンボルに対して、１つ又は複数の符号テーブル、１つ又は複数の頻度テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルのうち少なくとも１つを、前記シンボルの出現の関数として生成することと；
（ｃ）各データチャンクにおけるシンボルを、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つにおけるエントリに関連付ける１つ又は複数のインデクスセットを計算することと；
（ｄ）各データチャンクにおけるシンボルを関連付ける前記１つ又は複数のインデクスセットの情報を、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つと共に、前記符号化データ（Ｅ２）に含めることと；
（ｅ）前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの少なく

とも１つを用いて前記シンボルを圧縮し、前記符号化データ（Ｅ２）に含めることと；
を含む、方法。
【請求項２】
前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの少なくとも１つの少なくとも１つは定義済みである、請求項１に記載の方法。
【請求項３】
前記符号化データ（Ｅ２）は、前記圧縮シンボルと一緒に、前記各データチャンクにおけるシンボルを関連付ける前記１つ又は複数のインデクスセットの情報を含み、該情報は、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つを伴い、前記方法は前記符号化データ（Ｅ２）を送出することを含む、請求項１又は２に記載の方法。
【請求項４】
入力データ（Ｄ１）を符号化して対応する符号化データ（Ｅ２）を生成するエンコーダ（５０）であって、前記エンコーダ（５０）は、
（ａ）前記入力データ（Ｄ１）を１つ又は複数のデータチャンクに分割及び／又は変換するために、該入力データ（Ｄ１）に存在するシンボルを解析する解析器と；
（ｂ）前記１つ又は複数のデータチャンクに存在する前記シンボルに対して、１つ又は複数の符号テーブル、１つ又は複数の頻度テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルのうち少なくとも１つを、前記シンボルの出現の関数として生成する生成器と；
（ｃ）各データチャンクにおけるシンボルを、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つにおけるエントリに関連付ける１つ又は複数のインデクスセットを計算する計算エンジンと；
（ｄ）各データチャンクにおけるシンボルを関連付ける前記１つ又は複数のインデクスセットの情報を、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つと共に、前記符号化データ（Ｅ２）に含めるデータアセンブラと；
（ｅ）前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの少なくとも１つを用いて前記シンボルを圧縮し、前記符号化データ（Ｅ２）に含める圧縮器と；
を備える、エンコーダ（５０）。
【請求項５】
前記符号化データ（Ｅ２）は、前記圧縮シンボルと一緒に、前記各データチャンクにおけるシンボルを関連付ける前記１つ又は複数のインデクスセットの情報を含み、該情報は、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つを伴い、前記符号化データ（Ｅ２）を送出するように動作可能である、請求項４に記載のエンコーダ（５０）。
【請求項６】
前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの少なくとも１つの少なくとも１つが後で再利用できるように保存可能な仕方で、該１つ又は複数のテーブルの少なくとも１つを送出するように動作可能である、請求項４に記載のエンコーダ（５０）。
【請求項７】
前記入力データ（Ｄ１）を複数のデータチャンクに分割し、前記複数のデータチャンクを実質的に同時に処理する並列プロセッサアーキテクチャを用いるように動作可能である

、請求項４から６の何れかに記載のエンコーダ（５０）。
【請求項８】
前記インデクスは、Ｒ、Ｇ、Ｂの画素値又はＹ、Ｕ、Ｖの画素値を含む１つ又は複数のＲＧＢ画素又はＹＵＶ画素から算出され、
前記データチャンクが前記符号化データ（Ｅ２）に含まれる場合、前記データチャンクに対して達成可能なデータ圧縮率の関数として、前記データチャンクを前記符号化データ（Ｅ２）に集める際に前記データチャンクの非符号化又は符号化を動的に切り換えるように動作可能である、請求項４に記載のエンコーダ（５０）。
【請求項９】
装置の処理手段によって実行されると、前記装置に、請求項１から３の何れかに記載の方法を遂行させるように構成されたコンピュータ可読命令を含む、コンピュータプログラム。
【請求項１０】
請求項４から８の何れかに記載のエンコーダ（５０）によって生成された符号化データ（Ｅ２）をデコーダ（６０）で復号して、対応する復号データ（Ｄ３）を生成する方法であって：
（ｉ）前記符号化データ（Ｅ２）を受け取り、１つ又は複数の頻度テーブル、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルのうちの少なくとも１つと、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの前記少なくとも１つを示す情報と共に、１つ又は複数のインデクスセットの情報を前記符号化データから抽出することと；
（ｉｉ）前記１つ又は複数のインデクスセットから、１つ又は複数のデータチャンクの圧縮シンボルに関連する、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブル、の少なくとも１つを計算することと；
（ｉｉｉ）前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つからの情報を用いて、前記圧縮シンボルから、前記１つ又は複数のデータチャンクを再生成することと；
（ｉｖ）前記復号データ（Ｄ３）を生成するために、前記１つ又は複数のデータチャンクを結合及び／又は変換することと；
を含む、方法。
【請求項１１】
前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの少なくとも１つの少なくとも１つは定義済みである、請求項１０に記載の方法。
【請求項１２】
対応するトランスコードデータ（Ｄ４）を生成するために前記復号データ（Ｄ３）をトランスコードすること、及び／又は、前記符号化データ（Ｅ２）から対応するトランスコードデータ（Ｄ４）を生成することを含む、請求項１０に記載の方法。
【請求項１３】
複数のデータソースから前記符号化データ（Ｅ２）を受け取り、前記符号化データ（Ｅ２）を再生成するために、前記ソースから提供されたデータを結合することを含む、請求項１０から１２の何れかに記載の方法。
【請求項１４】
請求項４から８の何れかに記載のエンコーダ（５０）によって生成された符号化データ（Ｅ２）をデコーダ（６０）で復号して、対応する復号データ（Ｄ３）を生成するデコーダ（６０）であって、前記デコーダ（６０）は：
（ｉ）前記符号化データ（Ｅ２）を受け取り、１つ又は複数の頻度テーブル、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブ

ルのうちの少なくとも１つと、、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの前記少なくとも１つを示す情報と共に、１つ又は複数のインデクスセットの情報を前記符号化データから抽出することと；
（ｉｉ）前記１つ又は複数のインデクスセットから、１つ又は複数のデータチャンクの圧縮シンボルに関連する、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブル、の少なくとも１つを計算することと；
（ｉｉｉ）前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つからの情報を用いて、前記圧縮シンボルから、前記１つ又は複数のデータチャンクを再生成することと；
（ｉｖ）前記復号データ（Ｄ３）を生成するために、前記１つ又は複数のデータチャンクを結合及び／又は変換すること
を実行するように動作可能である、デコーダ（６０）。
【請求項１５】
対応するトランスコードデータ（Ｄ４）を生成するために前記復号データ（Ｄ３）をトランスコードするトランスコーダ（７０）、及び／又は、前記符号化データ（Ｅ２）から対応するトランスコードデータ（Ｄ４）を生成することを更に備える、請求項１４に記載のデコーダ（６０）。
【請求項１６】
前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの少なくとも１つの少なくとも１つが後で再利用できるように保存可能な仕方で、該１つ又は複数のテーブルの少なくとも１つを受け取るように動作可能である、請求項１４に記載のデコーダ（６０）。
【請求項１７】
前記復号データ（Ｄ３）を生成するために、前記１つ又は複数のデータチャンクのうち複数を実質的に同時に処理する並列プロセッサアーキテクチャを用いることによって、該複数のデータチャンクを結合するように動作可能である、請求項１４から１６の何れかに記載のデコーダ（６０）。
【請求項１８】
一緒に結合された複数のデータ値に基づいて、前記１つ又は複数のインデクスセットを生成するように動作可能であり、
前記１つ又は複数のデータチャンクが前記符号化データ（Ｅ２）に含まれる場合、該データチャンクに対して達成可能なデータ伸長率の関数として、該１つ又は複数のデータチャンクを該符号化データ（Ｅ２）に生成する際に該データチャンクの非符号化又は符号化を動的に切り換えるように動作可能である、請求項１４又は１７に記載あのデコーダ（６０）。
【請求項１９】
前記符号化データ（Ｅ２）に含まれる前記１つ又は複数の符号テーブルを伸張するように動作可能である、請求項１４から１８の何れかに記載のデコーダ（６０）。
【請求項２０】
次の送信データを復号するために、前記１つ又は複数の符号テーブルを前記デコーダ（６０）で使用可能にする仕方で該１つ又は複数の符号テーブルを受け取るように動作可能である、請求項１４から１９の何れかに記載のデコーダ（６０）。
【請求項２１】
前記１つ又は複数の符号テーブルがアクセスされ易い場所を示す１つ又は複数の識別コードを、１つ又は複数のデータベース及び／又は１つ又は複数のプロキシデータベースを介して、前記符号化データ（Ｅ２）から抽出するように動作可能である、請求項１４から２０の何れかに記載のデコーダ（６０）。

【請求項２２】
複数のデータソースから前記符号化データ（Ｅ２）を受け取り、前記符号化データ（Ｅ２）を再生成するために、前記ソースから提供されたデータを結合するように動作可能である、請求項１４に記載のデコーダ（６０）。
【請求項２３】
入力データ（Ｄ１）を符号化して対応する符号化データ（Ｅ２）を生成する、請求項４に記載のエンコーダ（５０）を少なくとも一つと；
前記符号化データ（Ｅ２）を復号して対応する復号データ（Ｄ３）を生成する、請求項１４に記載のデコーダ（６０）を少なくとも一つ
を備える、コーデック（１００）。
【請求項２４】
前記少なくとも１つのエンコーダ（５０）及び前記少なくとも１つのデコーダ（６０）は、互いに空間的に離隔され、データ通信ネットワークを介して互いに接続されている、請求項２３に記載のコーデック（１００）。
【請求項２５】
前記データ通信ネットワークは、ピアツーピアネットワーク方式で構成される、請求項２４に記載のコーデック（１００）。
【請求項２６】
前記エンコーダ（５０）及び前記デコーダ（６０）は、該エンコーダ及びデコーダを通じたデータ処理に関して対称である、請求項２４に記載のコーデック（１００）。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、入力データの符号化方法に関し、それに対応する符号化データを生成するものである。また本開示は、前述の符号化データの復号方法に関し、それに対応する復号出力データを生成するものでもある。さらにまた、本開示は、前述の方法を実装するように動作可能なエンコーダとデコーダにも関連する。さらに加えて、本開示は、非一時的コンピュータ可読記憶媒体を含むコンピュータプログラム製品であって、非一時的コンピュータ可読記憶媒体はコンピュータ可読命令を含み、コンピュータ可読命令は、前述の方法を実行する処理ハードウェアを備える計算デバイスによって実行可能である、コンピュータプログラム製品にも関連する。
【背景】
【０００２】
概要としては、図１に示すように、入力データＤ１を符号化してそれに対応する符号化出力データＥ２を生成する既知の符号化方法は、入力データＤ１に対する１つ又は複数の変換Ｔの適用を伴い、それに対応する変換済み符号化出力データＥ２を生成する。ここで、変換済み符号化出力データＥ２は、それに関連付けられる符号テーブルデータＣの情報を有し、その情報は、採用される１つ又は複数の変換Ｔを規定する１つ又は複数の符号テーブルを指し示す。符号化変換データＥ２と符号テーブルデータＣ情報をまとめて符号化出力データＥ２と呼ぶ。符号化出力データＥ２は通常、データキャリア、データ通信ネットワークの何れか又は両方を介して１つ又は複数のデコーダに伝送される。デコーダは、１つ又は複数の逆変換T^{-1}を適用し、符号化出力データＥ２を復号してそれに対応する復号データＤ３を生成するように動作可能である。通常、符号化出力データＥ２は、例えば符号化出力データＥ２を伝送するときに通信ネットワークの容量負荷を減らすために、入力データＤ１よりも圧縮されていることが望ましい。また、符号化出力データＥ２が実質的に可逆方式で圧縮され、復号データＤ３が入力データＤ１に含まれる情報を正確に再現したものであることも望まれる。入力データＤ１に対して符号化出力データＥ２で実現可能なデータ圧縮は、符号テーブルデータＣ情報が符号化変換データＥ２と比べてかなりの大きさである場合、即ち、符号テーブルデータＣ情報が符号化変換データＥ２において大幅なデータオーバヘッドに相当する場合は非効率になる可能性がある。

【０００３】
入力データＤ１を符号化して符号化出力データＥ２を生成する既知の方法が幾つか存在する。例えば、既知のハフマン符号化（Huffman encoding）又は他の可変長符号化（VLC encoding）方法は、様々な種類のデータの圧縮で度々採用される。また、算術符号化（Arithmetic coding）やレンジ符号化（Range coding）も入力データ圧縮によく採用されてきているが、次のような状況ではかなり非効率である：
（ｉ）入力データＤ１に関する頻度テーブルが、入力データＤ１を符号化してそれに対応する符号化出力データＥ２を生成するように動作可能なエンコーダには未知であり、その符号化出力データＥ２を復号するように動作可能なデコーダに対しても未知である場合；
（ｉｉ）入力データ量が比較的小さい場合。例えば、入力データＤ１が小さいデータセグメント又はデータチャンクで伝送され、データセグメント又はデータチャンクの各々がそれぞれに対応する頻度テーブルを伴っている場合。
【０００４】
前述のように、こうした非効率性は、多量のデータ空間を使う１つ又は複数の頻度テーブルを送出することに起因する。例えば、頻度テーブルがデコーダにローカルな形で格納されていて、可能な頻度テーブルのリストから比較的少数の識別パラメータを用いて選択することが不可能な場合に起こりうる。また、そうしたリストから適切な頻度テーブルを探すことは、適切な符号テーブルを探すことよりも確実ではないこともある。符号化対象入力データＤ１が局所的に変化することも度々あり、例えば、通信ネットワークに対して空間ローカルのデータ規格に適合させるために、入力データがその通信ネットワークで伝送中に変換されることもある。
【０００５】
シンボルから派生した符号化データコンテンツの伝送に関連して、符号テーブル又は頻度テーブルの送出に利用可能な既知の方法も存在する。こうした既知の方法の多くは、シンボルに関するハフマンツリー又は頻度の直接送出を利用する。しかし、こうした既知の方法は十分満足のいくものではない。こうした方法では、エンコーダからそれに対応するデコーダに送出すべき情報が相当量必要となるためである。また、符号テーブルのシンボル長を送出する方法も知られており、例えば既知のインテルＩＰＰライブラリは、本願出願時においては非推奨であるが、いわゆる「HuffLenCodeTablePacK」によって符号テーブルを圧縮し、「HuffLenCodeTableUnpack」によって復号して元に戻す方法が採用されている。しかし、この方法でも十分ではなく、符号化処理中でもデータサイズが増大することがある。この方法は更に、256種のシンボルが在り、0から255の全シンボルがゼロでないコードワード長を持つ必要がある。ハフマン符号化技術等で生成されるプレフィックスコードに対して現在利用可能な送出機構の中では、符号テーブルを送出する方法が最も効率がよいことは明白である。エンコーダからそれに対応するデコーダにハフマンツリーが送出される場合、エンコーダから生成されたコードシンボルは、エンコーダとデコーダにおいて常に同じである。頻度テーブルのみが送出される場合、ハフマンツリーが必要であると仮定すると、頻度テーブルから実際のハフマンツリーを生成するアルゴリズムは、デコーダが適切にシンボルを復号できるように、エンコーダとデコーダで同一のアルゴリズムが使用されなくてはならない。符号テーブルのシンボル長が送出される場合、シンボル長から頻度テーブルに変換する方法もまた、デコーダが適切にシンボルを復号できるように、エンコーダとデコーダで同一の方法が必要である。一方、算術符号化とレンジ符号化では、エンコーダからデコーダにシンボル長を伝送することは出現頻度の送出方法として実用的ではない。これらの符号化方法は、単にコードシンボル長の伝送によって有効に機能するよりも、より正確な頻度テーブルをサポートするために設計されているためである。コードシンボル長自体は算術符号化でもレンジ符号化でも使用することはできるが、後続データに対して後からテーブル適応更新が実行されなければ、これらの符号化方法にはハフマン符号化等と比べてあまり有益ではない。しかし、ハフマン符号化とは対照的にレンジ符号化又は算術符号化では、出現確率を示す情報の送出でより最適化された符号化結果が得られる。シンボルの出現確率は、シンボルの出現頻度をシンボル出現頻度の和、即ち

シンボル数で割って算出することができる。こうした確率の送出は、スケール確率値を用いることで有益に行われる。スケール確率値は、元のシンボル出現確率値に特定の整数を掛け、近似整数値に丸めて算出することができる。この特定の整数として２の累乗、即ち2^nとするのが有益である。ここで、ｎは整数である。こうしたスケール確率の和は整数であり、乗数の値と同じになるように均される。別の方法で非ゼロスケール確率値を割り当てることができないシンボルに対して、エスケープコードシンボルの生成が有用である。このことは、エスケープシンボルを必要するこれらのシンボルが、選択された乗数値で存在しうるものよりも確率が低いことを意味する。また、二つの別々の機構を用いてエスケープコードを使わずにスケール確率を計算することもできる。乗数値を大きくして新しい確率値を算出することもできる。利用可能なシンボルの確率値を０から１に更新することもできる。この確率値更新では、この確率値の上昇が他のシンボルの確率値を下げることによって補償されなくてはならない。これは、確率の和を乗数値と完全に一致させるように行われる。この処理では確率値は最大限に最適化されないが、エスケープシンボルは不要であり、場合によっては最適な符号化方法となりうる。シンボル長又は確率値は、可変長符号化を採用する方法において使用可能な頻度テーブルの概算を規定する。ここで、可変長符号化には、例えばハフマン符号化やレンジ符号化、算術符号化、その他の可変長符号化が含まれる。当然ながら、スケール確率テーブルは、必要であればシンボル出現頻度の概算として直接使用されてもよく、使用前にはシンボル長が最初にシンボル出現頻度の概算として変換されなくてはならない。データ符号化とテーブル送出におけるシンボル長から頻度テーブルへの変換はこの後で詳述する。

【０００６】

多数の公知かつ実用的なデータ符号化方法には、最適化された符号テーブルが全く利用されていない。つまり、こうした方法は、データを符号化して符号化データを生成するのに固定の符号テーブルを利用し、その後の符号化データの復号でも固定符号テーブルを使用している。ただし、送出されたシンボルに基づく適応方法で符号テーブルが更新されることもある。ある既知の方法では、エンコーダでデータを符号化し、それに対応してデコーダで符号化データを復号するために、別々の符号テーブルあるいは頻度テーブルの組が利用されることもある。この場合、選択された符号テーブルや確率テーブル、頻度テーブルを規定するインデクスがエンコーダからデコーダへの情報として送出される。画像の輝度／色差チャネル、インター／イントラブロック、異種データに対して別々のテーブルを用いる方法も存在するが、こうした別々のテーブルは非効率な仕方で伝送される。例えば、次のインターネットウェブサイト（ウィキペディア）を参照されたい：http://en.wikipedia.org/wiki/Huffman_coding。ハフマンベースの方法を用いる場合、伸張時にはハフマンツリーを再構成しなくてはならない。文字の出現頻度が比較的予測可能であるような最も単純な場合では、ツリーは再構成し易く、各圧縮サイクルで統計的に調節することも可能である。したがって、圧縮率の少なくとも一部だけ犠牲にすることで、毎回ツリーを再利用することができる。あるいは、ハフマンツリーの情報がアプリオリに、即ち予め送られていなくてはならない。

【０００７】

圧縮データ出力列に符号化されるシンボルに関連する出現頻度数を先頭に追加する単純なアプローチには、圧縮データの量を実際に少なくとも数キロバイト（kB）まで増加させる欠点があり、そうした単純なアプローチが実際に用いられることは殆ど無い。データがカノニカル符号化（canonical encoding）で圧縮される場合、圧縮モデルは丁度$B2^B$ビットの情報で正確に再構成することができる。ここで、Ｂは１シンボル当りのビット数で、例えば８ビットであれば２ｋＢを必要とする。

【０００８】

もう一つの方法は、単純に圧縮出力列に対してビット毎にハフマンツリーを先頭に追加するというものである。例えば、値0が親ノード、値1が葉ノードを表わすと仮定すると、葉ノードが現れたら必ず、ツリー構築ルーチンは、この特定の葉ノードの文字値を決定するため単純に次の8ビットを読み取る。こうした処理は最後の葉ノードに到達するまで再

帰的に継続し、最終葉ノードに到達すると、デコーダ等でハフマンツリーが忠実に再構成される。こうした方法を用いることで生じるデータオーバヘッドは、8ビットのアルファベットを仮定した場合、概ね2から320バイトの範囲になる。

【０００９】

次に、既知のデータ符号化方法及びそれに対応する符号化データの復号方法を更に明らかにするために、ハフマン復号の概要を説明する。当然ながら、ハフマン復号以外にも、例えばレンジ復号や算術復号等、他の方法が利用されてもよい。データファイルの圧縮を開始する前に、エンコーダの圧縮器は、圧縮を実行するときに用いるコードを決定しなくてはならない。

【００１０】

ハフマン復号を採用する場合、エンコーダは、符号化出力データを生成するために、シンボルを含む所定の対応するデータファイルを圧縮し始める前に、この所定データを表現するために使用すべきコードを決定しなくてはならない。このコードは、所定のデータファイルにあるシンボルの確率、即ち出現頻度に基づいていると都合がよい。しかし、シンボルの出現頻度、確率、シンボル長は、例えばサイド情報、即ち補足情報として符号化出力データに記録されなくてはならない。こうして、任意のハフマンデコーダが符号化出力データを復号してそれに対応する復号データを生成することができるようになる。シンボルの出現頻度又はシンボル長は、整数であると都合がよい。また確率であれば、スケールした整数として表現可能である。補足情報に含まれるこうした整数は通常、符号化出力データに対して僅か数百バイト程度を追加するだけである。また、符号化出力データに可変長コード自体を書き込むことも可能であるが、コードが互いに異なるサイズを持つことになるため、状況によっては厄介になる可能性もある。あるいは、符号化出力データにハフマンツリーを書き込むことも可能である。しかしこの場合、所定のデータにおけるシンボルの出現頻度を単に伝送するときよりも多くのデータを伝送しなくてはならない。

【００１１】

デコーダは動作時に、デコーダが受け取った復号対象の符号化圧縮ファイルの先頭が何であるかという情報の提供を受けなくてはならない。デコーダは、符号化圧縮ファイルの先頭等、ファイルから抽出されたデータからハフマンツリーの文字を構築するように動作可能である。デコーダでハフマンツリーが構成された後、デコーダは、そのハフマンツリーを復号ツリーとして使用してファイルの残り部分を復号することができる。デコーダは、次のステップを含む比較的単純な復号アルゴリズムを採用する：

（ａ）ハフマンツリーの根から開始し、ハフマンツリーを用いて復号されるべき符号化出力データの最初のビットを読み取る；

（ｂ）最初のビットが「1」であればハフマンツリーの上位エッジを辿り、最初のビットが「0」であればハフマンツリーの下位エッジを辿る；

（ｃ）符号化出力データの２番目のビットを読み取り、ハフマンツリーの「葉」に向かってステップ（ｂ）と同様の仕方で２番目のビットを使用する。ハフマンツリーの「葉」に到達すると、元の非圧縮シンボルが見つけられることになり、これは通常、関連するASCIIコードであり、このコードがデコーダから出力される；

（ｄ）符号化出力データの復号が完了するまで、ステップ（ｂ）と（ｃ）が繰り返される。

【００１２】

符号化列のデータサイズがデータ列の生成に用いられる符号テーブルのそれと比べて大きい場合には、現在既知のハフマン符号化を採用するのが有益である。また、符号テーブルがエンコーダとそれに対応するデコーダの両方で事前に規定される場合でも、現在のこうしたハフマン符号化を採用するのが有益である。したがって、前述したハフマン符号化及び復号方法等、データを符号化及び復号する既知のアプローチに関してこれまでに記述した制限に対処するために、代替となる符号化方法が必要とされている。

【摘要】

【００１３】

本発明は、データ（Ｄ１）を符号化してそれに対応する符号化データ（Ｅ２）を生成する改良方法を提供しようとするものである。

【００１４】

また本発明は、データを符号化する前述の改良方法を用いるように動作可能である改良型エンコーダを提供しようとするものである。

【００１５】

本発明は、符号化データ（Ｅ２）を復号してそれに対応する復号データ（Ｄ３）を生成する改良方法を提供しようとするものでもある。

【００１６】

本発明は、前述の符号化データ（Ｅ２）を復号してそれに対応する復号データ（Ｄ３）を生成する改良型デコーダを提供しようとするものでもある。

【００１７】

第１の態様によれば、入力データ（Ｄ１）をエンコーダで符号化してそれに対応する符号化データ（Ｅ２）を生成する方法が提供される。本方法は：

（ａ）前記入力データ（Ｄ１）に存在するシンボルを解析し、該入力データ（Ｄ１）を１つ又は複数のデータチャンクに分割及び／又は変換することと；

（ｂ）前記１つ又は複数のデータチャンクに存在する前記シンボルに対して、１つ又は複数の符号テーブル、１つ又は複数の頻度テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルのうち少なくとも１つを、前記シンボルの出現の関数として生成することと；

（ｃ）各データチャンクにおけるシンボルを、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つにおけるエントリに関連付ける１つ又は複数のインデクスセットを計算することと；

（ｄ）各データチャンクにおけるシンボルを関連付ける前記１つ又は複数のインデクスセットの情報を、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つと共に、前記符号化データ（Ｅ２）に含めることと；

（ｅ）前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの少なくとも１つを用いて前記シンボルを圧縮し、前記符号化データ（Ｅ２）に含めることと；

を含む。

【００１８】

本発明は入力データ（Ｄ１）を１つ又は複数のデータチャンクに小分けにする、即ち分割することと、入力データ（Ｄ１）にあるシンボルを圧縮することの何れか又は両方を伴う。それによって本発明は、入力データ（Ｄ１）が、例えばインデクスとそれが参照する１つ又は複数の関連するテーブルを用いて、各データチャンク又は圧縮シンボルに最適な仕方で効率よく符号化されようになるという優位性を持つ。

【００１９】

本方法は、１つ又は複数のテーブルの少なくとも１つが後で再利用できるように保存可能な仕方で、１つ又は複数のテーブルの少なくとも１つを送出することを含んでもよい。

【００２０】

前述のこうした小分け、即ち分割には、入力データ（Ｄ１）のサブ分割が含まれてもよい。

【００２１】

前述（ａ）では通常、分割が行われるが、データを新たなデータチャンクに分割せず、利用可能なデータチャンクを最適な符号テーブルで圧縮のみを行うこともできる。また、元のデータを分割する代わりに、例えば、最大効率で圧縮可能な１つ又は複数のデータチャンクが得られる１つ又は複数の変換によって、新たなデータが生成されてもよい。本開示に従って実装されたエンコーダは、様々なデータチャンクの生成に使用することができ

る。したがって当然のことながら、本方法は、別々の時間スロットで行われるチャンキングでデータを符号化するように動作可能なビデオコーデックやオーディオコーデックに適している。フレーム又はセクション毎にデータチャンクは異なり、本開示に従う１つ又は複数の方法を実装するエンコーダを用いて、こうしたデータを更に１つ又は複数のデータチャンクに分割することもできる。こうしたデータチャンクは全て、同一フレームやその前のフレーム等で既に送出されている任意のテーブルを再利用することができる。

【００２２】

本発明の実施形態により、符号テーブル又は頻度テーブルの効率的な送出が可能になる。これにより、テーブルの送出、保存の何れか又は両方に必要なデータ通信・データ保存のオーバヘッドを減らすことができる。また、個々のデータチャンクに最適化された符号テーブルを利用することで、より小さいデータチャンクの符号化も可能となる。こうして、更に高い圧縮効率の実現が可能となり、データ保存容量、伝送帯域幅、エネルギー消費を削減することができる。

【００２３】

データの様々な部分の出現頻度は、通常互いに異なるものであり、それに関連するデータエントロピーもそれぞれ異なる。このため、データを複数の部分、即ちデータチャンクに分割することが有益となる。こうした部分に対して、その部分に関するデータの本質、データの種類、データの内容の何れか又は全てに応じて、別々の符号テーブルを使用するのが有益である。ここでデータの「本質」とは、そのデータの１つ又は複数の特性、パラメータの何れか又は両方を意味する。本発明は、所定の大きなデータファイルをより効率的に小さな部分、即ちデータチャンクに分割可能な方法であって、そうしたデータチャンクに対する符号テーブル又は頻度テーブルの送出も最適化されるという関連した利益ももたらされる方法を提供する。大規模データファイルの分割により、データエントロピーの修正も伴うことから、実質的な利益がもたらされ、伝送対象である符号化データの量を大幅に削減することができる。１つ又は複数のデータチャンクにおけるデータ値も、前述のようにして分割することができる。こうしたデータ値の小分け、即ち分割の実装は、例えばＭＳＢ（最上位ビット）とＬＳＢ（最下位ビット）を互いに分離することによって可能となる。データ値がデータ値チャンクに分割される数が２を超えることもある。

【００２４】

前述した本発明の方法は、復号データ（Ｄ３）を生成するステップ（ｄ）において、１つ又は複数のデータ圧縮アルゴリズムを適用することを含んでもよい。本方法では更に、１つ又は複数のデータ圧縮アルゴリズムは、ハフマン符号化、ＶＬＣ、エントロピー符号化、算術符号化、レンジ符号化の少なくとも１つを含む。ただし、これらに限定されない。

【００２５】

本方法は、入力データ（Ｄ１）を複数のデータチャンクに分割することと、その複数のデータチャンクを実質的に同時に処理する並列プロセッサアーキテクチャを用いることを含んでもよい。

【００２６】

本方法は、一緒に結合された複数のデータ値に基づいて、１つ又は複数のインデクスセットを生成することを含んでもよい。本方法では更に、このインデクスは、Ｒ、Ｇ、Ｂの画素値又はＹ、Ｕ、Ｖの画素値を含む１つ又は複数のＲＧＢ画素又はＹＵＶ画素から算出されてもよい。

【００２７】

本方法は更に、データチャンクが符号化データ（Ｅ２）に含まれる場合、そのデータチャンクに対して達成可能なデータ圧縮率に応じて、データチャンクを符号化データ（Ｅ２）に集める際にデータチャンクの非符号化又は符号化を動的に切り換えることを含んでもよい。

【００２８】

本方法は、シンボルが「符号テーブル変更」又は「データ終了」に関連するかを示す少

なくとも１ビットの終了ビットを符号化データ（Ｅ２）に含めることを含んでもよい。
【００２９】
本方法は、所定のデータチャンクに存在する１つ又は複数のシンボルを参照するために必要十分なインデクスから、実質的にその所定のデータチャンクを生成することを含んでもよい。
【００３０】
送出された符号テーブルは、例えばハフマン符号化を用いて圧縮されてもよく、こうした符号テーブル圧縮方法が、それ特有の関連する１つ又は複数の符号テーブルと共に提供されてもよい。
【００３１】
第２の態様によれば、入力データ（Ｄ１）を符号化してそれに対応する符号化データ（Ｅ２）を生成するエンコーダが提供される。本エンコーダは：
（ａ）前記入力データ（Ｄ１）を１つ又は複数のデータチャンクに分割及び／又は変換するために、該入力データ（Ｄ１）に存在するシンボルを解析する解析器と；
（ｂ）前記１つ又は複数のデータチャンクに存在する前記シンボルに対して、１つ又は複数の符号テーブル、１つ又は複数の頻度テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルのうち少なくとも１つを、前記シンボルの出現の関数として生成する生成器と；
（ｃ）各データチャンクにおけるシンボルを、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つにおけるエントリに関連付ける１つ又は複数のインデクスセットを計算する計算エンジンと；
（ｄ）各データチャンクにおけるシンボルを関連付ける前記１つ又は複数のインデクスセットの情報を、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つと共に、前記符号化データ（Ｅ２）に含めるデータアセンブラと；
（ｅ）前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの少なくとも１つを用いて前記シンボルを圧縮し、前記符号化データ（Ｅ２）に含める圧縮器と；
を備える。
【００３２】
こうした小分け、即ち分割には、入力データ（Ｄ１）のサブ分割が含まれてもよい。
【００３３】
前述（ａ）では通常、分割が行われるが、データを新たなデータチャンクに分割せず、利用可能なデータチャンクを最適な符号テーブルで圧縮のみを行うこともできる。また、元のデータを分割する代わりに、例えば、最大効率で圧縮可能な１つ又は複数のデータチャンクが得られる１つ又は複数の変換によって、新たなデータが生成されてもよい。本開示に従って実装されたエンコーダは、様々なデータチャンクの生成に使用することができる。したがって当然のことながら、本方法は、別々の時間スロットで行われるチャンキングでデータを符号化するビデオコーデックやオーディオコーデックに適している。フレーム又はセクション毎にデータチャンクは異なり、本開示に従う方法を実装するエンコーダを用いて、こうしたデータを更に１つ又は複数のデータチャンクに分割することもできる。こうしたデータチャンクは全て、同一フレームやその前のフレーム等で既に送出されている任意のテーブルを再利用することができる。
【００３４】
本エンコーダは、符号化データ（Ｅ２）を生成するために、データアセンブラにおいて１つ又は複数のデータ圧縮アルゴリズムを適用するように動作可能でもよい。本エンコーダでは更に、１つ又は複数のデータ圧縮アルゴリズムは、ハフマン符号化、ＶＬＣ、エントロピー符号化、算術符号化、レンジ符号化の少なくとも１つを含んでもよい。
【００３５】

本エンコーダは、入力データ（Ｄ１）を複数のデータチャンクに分割し、その複数のデータチャンクを実質的に同時に処理する並列プロセッサアーキテクチャを用いるように動作可能でもよい。

【００３６】

本エンコーダにおいて生成器は、一緒に結合された複数のデータ値に基づいて、１つ又は複数のインデクスセットを生成するように動作可能でもよい。本エンコーダでは更に、このインデクスは、Ｒ、Ｇ、Ｂの画素値又はＹ、Ｕ、Ｖの画素値を含む１つ又は複数のＲＧＢ画素又はＹＵＶ画素から算出されてもよい。本エンコーダは更に、データチャンクが符号化データ（Ｅ２）に含まれる場合、そのデータチャンクに対して達成可能なデータ圧縮率に応じて、データチャンクを符号化データ（Ｅ２）に集める際にデータチャンクの非符号化又は符号化を動的に切り換えるように動作可能でもよい。

【００３７】

本エンコーダは、シンボルが「符号テーブル変更」又は「データ終了」に関連するかを示す少なくとも１ビットの終了ビットを符号化データ（Ｅ２）に含めることを含むように動作可能でもよい。

【００３８】

本エンコーダにおいて生成器は、所定のデータチャンクに存在する１つ又は複数のシンボルを参照するために必要十分なインデクスから、実質的にその所定のデータチャンクを生成するように動作可能でもよい。

【００３９】

送出された符号テーブルは、例えばハフマン符号化を用いて圧縮されてもよく、こうした圧縮方法が、それ特有の１つ又は複数の関連する符号テーブルを必要とする場合もある。

【００４０】

送出された符号テーブルは、同一フレームやその前のフレーム等で再利用されてもよい。すなわち、データチャンクの符号化は、このデータフレーム又はその前のデータフレームにおける他のデータチャンクに対して、それ以前に送出された任意の符号テーブルを再利用することができる。

【００４１】

本エンコーダを実装する場合、テーブル送出のための最適な実装が、例えば符号化データにおいて有益に採用されてもよい。あるいは、例えば１つ又は複数のデータベース、１つ又は複数のプロキシデータベースのような、テーブルがアクセス可能な場所を示す１つ又は複数の識別コードを符号化データに含めることによって行われてもよい。

【００４２】

より効率的なデータの符号化、符号化データの送出、符号化データの復号を提供するために、前述のような送出・参照テーブルが、例えばそのテーブルのインデクスが送出されている場合、後で使用する等の用途で保存されてもよい。こうしたアプローチは、本開示に従って、例えばエンコーダからそれに対応するデコーダに伝送すべきデータの量を削減することができる。

【００４３】

第３の態様によれば、非一時的コンピュータ可読記憶媒体を含むコンピュータプログラム製品であって、非一時的コンピュータ可読記憶媒体はコンピュータ可読命令を含み、コンピュータ可読命令は、第１の態様に従う方法を実行する処理ハードウェアを備える計算デバイスによって実行可能である、コンピュータプログラム製品が提供される。

【００４４】

第４の態様によれば、第２の態様に従うエンコーダによって生成された符号化データ（Ｅ２）を復号する方法が提供される。このエンコーダによって生成された符号化データ（Ｅ２）を復号し、それに対応する復号データ（Ｄ３）を生成する、デコーダで行われる前記方法は：

（ｉ）前記符号化データ（Ｅ２）を受け取り、１つ又は複数の頻度テーブル、１つ又は複

数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルのうちの少なくとも１つと、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの前記少なくとも１つを示す情報と共に、１つ又は複数のインデクスセットの情報を前記符号化データから抽出することと；
（ｉｉ）前記１つ又は複数のインデクスセットから、１つ又は複数のデータチャンクの圧縮シンボルに関連する、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブル、の少なくとも１つを計算することと；
（ｉｉｉ）前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つからの情報を用いて、前記圧縮シンボルから、前記１つ又は複数のデータチャンクを再生成することと；
（ｉｖ）前記復号データ（Ｄ３）を生成するために、前記１つ又は複数のデータチャンクを結合及び／又は変換すること
を含む。

【００４５】
本方法は、対応するトランスコードデータ（Ｄ４）を生成するために復号データ（Ｄ３）をトランスコードすること、及び／又は、符号化データ（Ｅ２）からそれに対応するトランスコードデータ（Ｄ４）を生成することを含んでもよい。

【００４６】
本方法は、１つ又は複数のテーブルの少なくとも１つが後で再利用できるように保存可能な仕方で、１つ又は複数のテーブルの少なくとも１つを受け取ることを含んでもよい。

【００４７】
本方法は、復号データ（Ｄ３）を生成するステップ（ｉｖ）において、１つ又は複数のデータ伸張アルゴリズムを適用することを含んでもよい。本方法では更に、１つ又は複数のデータ伸張アルゴリズムは、ハフマン復号、ＶＬＣ復号、エントロピー復号、算術復号、レンジ復号の少なくとも１つを含んでもよい。

【００４８】
本方法は、復号データ（Ｄ３）を生成するために、１つ又は複数のデータチャンクのうち複数を実質的に同時に処理する並列プロセッサアーキテクチャを用いることによって、その複数のデータチャンクを結合することを含んでもよい。

【００４９】
本方法は、一緒に結合された複数のデータ値に基づいて、１つ又は複数のインデクスセットを生成することを含んでもよい。本方法では更に、このインデクスは、Ｒ、Ｇ、Ｂの画素値又はＹ、Ｕ、Ｖの画素値を含む１つ又は複数のＲＧＢ画素から算出されてもよい。本方法は更に、データチャンクが符号化データ（Ｅ２）に含まれる場合、そのデータチャンクに対して達成可能なデータ伸長率に応じて、符号化データ（Ｅ２）における１つ又は複数のデータチャンクの非符号化生成又は符号化生成を動的に切り換えることを含んでもよい。

【００５０】
本方法では、デコーダは、シンボルが符号化データ（Ｅ２）から、「符号テーブル変更」又は「データ終了」に関連するかを示す終了ビットを少なくとも１つ抽出するよう動作可能でもよい。

【００５１】
本方法は、所定のデータチャンクに存在する１つ又は複数のシンボルを参照するために必要十分なインデクスから、実質的にその所定のデータチャンクを生成することを含んでもよい。

【００５２】
本方法は、符号化データ（Ｅ２）に含まれる１つ又は複数の符号テーブルを伸張することを含んでもよい。本方法は更に、ハフマン復号を用いて１つ又は複数の符号テーブルを

伸張することを含んでもよい。本方法では更に、１つ又は複数の符号テーブルの伸張は、１つ又は複数の補助符号テーブルを用いてもよい。

【００５３】

本方法は、次の送信データを復号するために、１つ又は複数の符号テーブルをデコーダで使用可能にする仕方で１つ又は複数の符号テーブルを受け取ることを含んでもよい。

【００５４】

本方法は、１つ又は複数の符号テーブルがアクセスされ易い場所を示す１つ又は複数の識別コードを、１つ又は複数のデータベースと１つ又は複数のプロキシデータベースの何れか又は両方を介して、符号化データ（Ｅ２）から抽出することを含んでもよい。

【００５５】

本方法は、次の種類のデータ：録音音声信号、録画ビデオ信号、静止画像、テキストデータ、感震データ、センサ信号データ、アナログ－デジタル変換（ADC）データ、生体信号データ、カレンダーデータ、経済データ、数学的データ、二進数データの１つ又は複数を復号することを含んでもよい。

【００５６】

本方法は、複数のデータソースから符号化データ（Ｅ２）を受け取り、符号化データ（Ｅ２）を再生成するために、ソースから提供されたデータを結合することを含んでもよい。

【００５７】

第５の態様によれば、非一時的コンピュータ可読記憶媒体を含むコンピュータプログラム製品であって、非一時的コンピュータ可読記憶媒体はコンピュータ可読命令を含み、コンピュータ可読命令は、第４の態様に従う方法を実行する処理ハードウェアを備える計算デバイスによって実行可能である、コンピュータプログラム製品が提供される。

【００５８】

第６の態様によれば、第２の態様に従うエンコーダによって生成された符号化データ（Ｅ２）を復号するデコーダが提供される。このエンコーダによって生成された符号化データ（Ｅ２）を復号し、それに対応する復号データ（Ｄ３）を生成する前記デコーダは：

（ｉ）前記符号化データ（Ｅ２）を受け取り、１つ又は複数の頻度テーブル、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルのうちの少なくとも１つと、、前記１つ又は複数の頻度テーブル、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の符号確率テーブルのうちの前記少なくとも１つを示す情報と共に、１つ又は複数のインデクスセットの情報を前記符号化データから抽出することと；

（ｉｉ）前記１つ又は複数のインデクスセットから、１つ又は複数のデータチャンクの圧縮シンボルに関連する、前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブル、の少なくとも１つを計算することと；

（ｉｉｉ）前記１つ又は複数の符号テーブル、前記１つ又は複数のコードワード長テーブル、前記１つ又は複数の確率テーブルのうち少なくとも１つからの情報を用いて、前記圧縮シンボルから、前記１つ又は複数のデータチャンクを再生成することと；

（ｉｖ）前記復号データ（Ｄ３）を生成するために、前記１つ又は複数のデータチャンクを結合及び／又は変換すること

を実行するように動作可能である。

【００５９】

本デコーダは更に、対応するトランスコードデータ（Ｄ４）を生成するために復号データ（Ｄ３）をトランスコードするトランスコーダ、及び／又は、符号化データ（Ｅ２）からそれに対応するトランスコードデータ（Ｄ４）を生成することを含んでもよい。

【００６０】

本デコーダは、１つ又は複数のテーブルの少なくとも１つが後で再利用できるように保存可能な仕方で、１つ又は複数のテーブルの少なくとも１つを受け取るように動作可能で

もよい。

【００６１】

本デコーダは、復号データ（Ｄ３）を生成する（ｉｖ）において、１つ又は複数のデータ伸張アルゴリズムを適用するように動作可能でもよい。本デコーダでは更に、１つ又は複数のデータ伸張アルゴリズムは、ハフマン復号、ＶＬＣ復号、エントロピー復号、算術復号、レンジ復号の少なくとも１つを含んでもよい。

【００６２】

本デコーダは、復号データ（Ｄ３）を生成するために、１つ又は複数のデータチャンクのうち複数を実質的に同時に処理する並列プロセッサアーキテクチャを用いることによって、その複数のデータチャンクを結合するように動作可能でもよい。

【００６３】

本デコーダは、一緒に結合された複数のデータ値に基づいて、１つ又は複数のインデクスセットを生成するように動作可能でもよい。本デコーダでは更に、このインデクスは、Ｒ、Ｇ、Ｂの画素値又はＹ、Ｕ、Ｖの画素値を含む１つ又は複数のＲＧＢ画素から算出されてもよい。本デコーダは更に、データチャンクが符号化データ（Ｅ２）に含まれる場合、そのデータチャンクに対して達成可能なデータ伸長率に応じて、符号化データ（Ｅ２）における１つ又は複数のデータチャンクの非符号化生成又は符号化生成を動的に切り換えるように動作可能でもよい。

【００６４】

本デコーダは、シンボルが符号化データ（Ｅ２）から、「符号テーブル変更」又は「データ終了」に関連するかを示す終了ビットを少なくとも１つ抽出するよう動作可能でもよい。

【００６５】

本デコーダは、所定のデータチャンクに存在する１つ又は複数のシンボルを参照するために必要十分なインデクスから、実質的にその所定のデータチャンクを生成するように動作可能でもよい。

【００６６】

本デコーダは、符号化データ（Ｅ２）に含まれる１つ又は複数の符号テーブルを伸張するように動作可能でもよい。本デコーダは更に、ハフマン復号を用いて１つ又は複数の符号テーブルを伸張するように動作可能でもよい。本デコーダでは更に、１つ又は複数の符号テーブルの伸張は、１つ又は複数の補助符号テーブルを用いてもよい。

【００６７】

本デコーダは、次の送信データを復号するために、１つ又は複数の符号テーブルをデコーダ（６０）で使用可能にする仕方で１つ又は複数の符号テーブルを受け取るように動作可能でもよい。

【００６８】

本デコーダは、１つ又は複数の符号テーブルがアクセスされ易い場所を示す１つ又は複数の識別コードを、１つ又は複数のデータベースと１つ又は複数のプロキシデータベースの何れか又は両方を介して、符号化データ（Ｅ２）から抽出するように動作可能でもよい。

【００６９】

デコーダは、次の種類のデータ：録音音声信号、録画ビデオ信号、静止画像、テキストデータ、感震データ、センサ信号データ、アナログーデジタル変換（ADC）データ、生体信号データ、カレンダーデータ、経済データ、数学的データ、二進数データの１つ又は複数を復号するように動作可能でもよい。

【００７０】

本デコーダは、複数のデータソースから符号化データ（Ｅ２）を受け取り、符号化データ（Ｅ２）を再生成するために、ソースから提供されたデータを結合するように動作可能でもよい。

【００７１】

第７の態様によれば、入力データ（Ｄ１）を符号化してそれに対応する符号化データ（Ｅ２）を生成する、第２の態様に従うエンコーダを少なくとも１つと、その符号化データ（Ｅ２）を復号してそれに対応する復号データ（Ｄ３）を生成する、少なくとも１つのデコーダを備える、コーデックが提供される。

【００７２】

本コーデックは、少なくとも１つのエンコーダ及び少なくとも１つのデコーダが互いに空間的に離隔され、データ通信ネットワークを介して互いに接続されるように実装されてもよい。本コーデックは更に、データ通信ネットワークがピアツーピアネットワーク方式で構成されるように実装されてもよい。本コーデックは、そのエンコーダ及びデコーダがそれらを通じたデータ処理に関して対称であるように実装されてもよい。換言すれば、エンコーダで実行される処理関数は、デコーダにおいてそれに対応する逆関数として、デコーダでは逆の順序で実行されるものが実装される。

【００７３】

本発明の構成は、添付の特許請求の範囲に定義される本発明の範囲を逸脱しない限りにおいて、様々に組合せ可能であることを理解されたい。

【図面の簡単な説明】

【００７４】

以下、本開示の実施形態を、一例として次の図面を参照しながら説明する。

【図１】データを符号化及び復号する既知のエンコーダ及び既知のデコーダを描いた図である。

【図２】本開示の実施形態に従うデータ符号化方法を示す図である。

【図３Ａ】本開示に従うエンコーダ及びデコーダ、合わせてコーデックと呼ぶものの実施形態を示す図である。

【図３Ｂ】本開示に従うエンコーダ及びデコーダ、合わせてコーデックと呼ぶものの別の実施形態であって、デコーダの復号データＤ３がトランスコードされてトランスコードデータＤ４を生成する実施形態を示す図である。　添付図面において下線の引かれた番号は、その番号が位置するアイテムやその番号が隣接するアイテムを表わすために使用される。番号に下線が無く矢印を伴って書かれている場合、その番号は矢印が示す汎用アイテムを特定するのに使用される。

【実施形態の説明】

【００７５】

本開示は概括すると、例えばエンコーダ、デコーダ、コーデック、関連する動作方法に関する。また、本開示の実施形態は、既知の方法と比べ符号テーブルや頻度テーブル、コードワード長テーブル、出現確率テーブルの送出を改善する動作も可能にする。さらに、本開示の実施形態は、１つ又は複数のハフマンツリーの送出において送出に用いるビット数を少なくする仕方で送出することができる。それによって、特に、１つ又は複数のテーブルを付随する符号化データの量が比較的少ないときに、データ符号化中のデータ圧縮率を高めることができる。符号テーブルや頻度テーブル、コードワード長テーブル、確率テーブルは、多種多様なエントロピー符号化法に必要である。こうした符号化法には、例えばハフマン符号化、算術符号化、レンジ符号化のような可変長符号化（VLC）があるが、これらに限定されない。送信機等のエンコーダと受信機等のデコーダの両方において、以下で記述される方法を採用することが有益である。

【００７６】

以降で開示する実施形態は、保存及び伝送されるデータの量が時代の経過に伴って急速に増えている状況に関連する。こうしたデータの保存及び伝送は、相当量の記憶容量や伝送帯域幅、そしてエネルギーを消費する。こうした状況におけるデータの殆どは、録音音声信号、録画ビデオ信号、静止画像、テキストデータ、感震データ、センサ信号データ、アナログ－デジタル変換（ADC）データ、生体信号データ、カレンダーデータ、経済データ、数学的データ、二進数データ等であるが、これらに限定されない。本開示の実施形態は、前述の全ての種類のデータと、それ以外の種類のデータに対して符号化データ量を減

らすように動作可能である。それによって、符号テーブルや頻度テーブル、コードワード長テーブル、確率テーブルの効率的送出を可能にし、データのエントロピー、即ちデータサイズを効率的に減らせる、より小さいデータチャンクの使用を可能にする。

【００７７】

また、データチャンクは小さい程、並列処理で効率的に処理され速く結果を出力することができる。こうした並列化は現在のマイクロプロセッサのアーキテクチャでは一般的であり、特に、データプロセッサアレイ、RISC（縮小命令セットコンピュータ）プロセッサの高速構成等、将来のマイクロプロセッサ構成においても共通である。

【００７８】

所定の符号化方法に対して、対応する符号テーブルが含む情報は、ビット等で表わされるコードワード長、コードワードを再現するコード、コードワードのインデクスを示す。符号テーブルは、コードワード長から生成することもできる。コードワードのインデクスは、コードワードで符号化される元のシンボルに対応する値を表わす。同様に、頻度テーブルには、シンボルの出現頻度とそのシンボルのインデクスが含まれる。シンボルのインデクスは、そのインデクスで符号化される元のシンボルの値を表わす。頻度テーブルは確率テーブルに変換することができ、この確率テーブルを頻度テーブルの概算として使用することもできる。頻度テーブルとコードワード長とは相互に変換可能である。

【００７９】

前述のテーブルの何れかが送出される場合、そうした送出において非常に重要なパラメータの一つはテーブルの最大インデクスである。テーブルの最大インデクスは、送出されたテーブルにおいて、同じく入力データにおいて利用可能な相異なるシンボルが何種類あるか、あるいは最大何種類ありうるかを表わしている。例として、次のデータが与えられたとする：

4，3，0，1，0，4，3，4。

【００８０】

この場合、実際の最大インデクスは4、最小インデクスは0であり、このデータに存在する相異なるシンボルは最大で5種類（最大インデクス－最小インデクス＋１＝4-0+1）ということになる。実際にデータに存在する異なるシンボルは4種類（0，1，3，4）しかないため、テーブルは、可能な相異なるシンボルの数である「4」ではなく、最大インデクス、即ち利用可能な相異なるシンボルの数としての「3」を用いて送出されてもよい。テーブルの最大インデクス値として値3が用いられる場合、各テーブルインデクスに対してどのシンボルが用いられるのかという情報を送出するために、他の機構が必要になることもある。

【００８１】

シンボルが順番に並んでいる場合、実際の最大インデクス（4）と可用性ビットが送出されてもよい。この実施例の場合、11011が送出される。こうして送出される情報は、テーブルインデクス0が、それをインデクスとしてマークされるシンボル0に等しくなるように対応付けされ、関連するシンボルの対は（0，0）になる。同様に、残りのインデクスとシンボルの対は(1，1)、(2，3)、(3，4)である。こうしたインデクス・シンボル対を、別のテーブルインデクスとして用いられるインデクスを直接定義するために使用し、テーブルの最大インデクスである「3」を送出することもできる。これは、例えば送出されるテーブルの中のシンボルがその出現頻度に基づいて格納される場合、非常に価値のある方法である。

【００８２】

例えば前述の例では、インデクス・シンボル対は、例えば(0，4)、(1，0)、(2，3)、(3，1)である。場合によっては、使用されるインデクス・シンボル対が予め定義され、別の時点で、この使用されるインデクス・シンボル対のテーブルのインデクスが送出される。また別の異なる状況では、このインデクス・シンボル対が符号化データ（Ｅ２）と共に送出される。さらに別の異なる状況では、デコーダ６０は、既知の場所から使用されるインデクス・シンボル対を読み出すように動作可能である。また別の異なる状況では、デコ

ーダ６０は、符号化データ（Ｅ２）と共に送出された情報に対応する場所から、使用されるインデクス・シンボル対を読み出すように動作可能である。

【００８３】

状況によっては、テーブルが対称性を活用して送出されると見積もれるという有利な点もある。対称性を利用することによって、使用されている符号テーブルを、必要となるテーブル全体を送出するときよりも小さいデータサイズで送出することができる。これは、符号テーブルがコード長、出現確率、出現頻度の何れかに基づいているか否かに拘らず可能である。また、エンコーダ５０及びデコーダ６０の両方で対称なテーブルを用いる場合、それに含まれる要素、即ちシンボルが少ないほど高速で生成することができる。対称性を利用することでテーブルの部分最適化が可能になるが、送出されるテーブルのデータサイズの増加節約分は、とりわけ送信対象データの量がかなり少ない場合、全体での損失と相殺することになる。

【００８４】

前述の最後の実施例では、テーブルの送出に対称性が利用されてもよい。これは、シンボル値「0」は「1」よりも確率が高く、それに対応して、シンボル値「4」は「3」よりも確率が高いためである。また、シンボル値「0」と「4」の確率は互いに近く、それに対応して、シンボル値「1」と「3」の確率も互いに近い。しかし、値「2」はこのデータには全く出現しない。それ故、データが検査される向きとは無関係に最小の確率値になる。

【００８５】

対称性が利用される場合、符号テーブルは常に、対称的に対応する要素の出現数の和に基づいて生成することができる。この場合、シンボル値「0」と「4」の出現数は合わせて5であり、それに対応して、シンボル値「1」と「3」の出現数は合わせて3である。要素「2」はこのデータには全く出現しないので、それ自身のシンボルを与える必要は無い。ただし状況によっては、要素「2」にも同様にシンボルが生成されてもよい。この場合、対称性を利用するなら、このシンボルが右手系テーブルと左手系テーブルの両方に含まれることになる。

【００８６】

こうして、この範囲に対する出現数が5，3，(0)であるようにして、レンジ符号テーブルが生成されてもよい。使用されているレンジテーブルはシンボル0から4に対して5，3，0，3，5であり、最適な出現頻度ベースのレンジテーブルでは当然ながら3，2，0，１，2である。前述の対称性に基づく最適な実装では、出現頻度として2つの値を送信、即ち送出する必要があるが、一方で、レンジテーブルを用いる場合では出現頻度として４つの値の送出が必要になる。

【００８７】

対称性に基づく同様の着想が、ハフマン符号化等の他の符号化法と共に用いられてもよい。この場合、対称性に基づくテーブルは、例えば、左手系の値がコード「0」を受け取り、右手系の値がコード「1」を受け取るテーブルになる。したがって、ハフマンコードワードは例えば01や00となり、10や11は存在しない。あるいは、値「2」が将来存在できる選択肢も保持したい場合、コードワードは01，001，000/100，101，11となる。この実装において原理的には、送信／送出するのに２つのコード長（1と1）だけでもよく、あるいは３つのコード長（1、2、2）でもよい。テーブルが対称性を利用していることが分かっている場合には、使用されているテーブルはデコーダ６０で完全に復元することができる。こうした利点は、テーブルが長くなる程、より一層明確になってくる。

【００８８】

当然ながら、テーブルが対称性を利用しているか否かに関する情報が事前に分かっていてもよく、そうでなければ、最適テーブル又は定義済みテーブルが使用中であるかに関する情報や、前のテーブルから動的に生成されたテーブルであるかに関する情報が、同様の仕方で送信／送出されてもよい。テーブルが対称性を利用しているか否かに関する情報の送出は、使用されているテーブルのインデクスをデコーダ６０に送信することによって実行される。

【００８９】
例えば：
（ｉ）テーブルインデクス「0」は、そのテーブル全体が送信／送出されることを意味する；
（ｉｉ）テーブルインデクス「1」は、そのテーブルが対称的でその半分のみが送出されることを意味する；
（ｉｉｉ）テーブルインデクス2から63までは、定義済みテーブルが使用されていることを意味する；
（ｉｖ）テーブルインデクス64から127までは、以前に送出・格納された動的テーブルが使用されていることを意味する。
【００９０】
当然ながら、対称テーブルが定義済みテーブルや動的格納済みテーブルとして利用されてもよい。
【００９１】
また、例えばODelta符号化を用いて又は用いずに、様々な符号化方法が使用されてもよい。ODelta符号化は、データの値0・1の列への差動符号化と、計数ラップアラウンド（counting wraparound）の採用を含んでいる。さらに、こうした様々な符号化方法を利用する場合、使用されているテーブルのテーブルインデクスと一緒に利用することが有益であり、符号化前にODelta法がデータに適用されたか否かを示したり、それに対応してデコーダ６０では復号後にその逆変換を行わなくてはならないか否かを示したりしてもよい。
【００９２】
このような場合、例えばテーブルインデクスは、ODelta法が使用されていたことを示す値128_{dec}が追加される以外はこれまでと同じである。 この追加が行われなかったならば、ODelta法は符号化前のデータに適用されなかったことになる。当然ながら、テーブルインデクス値に他の値を追加することもできる。しかし基本的なことであるが、符号化データに付随するテーブルインデクスは、使用されているテーブルの種類、そのテーブルインデクスと共に伝送／送出される追加データの種類を表わしている。
【００９３】
次に図２と図３Ａを参照する。図には、入力データＤ１を符号化してそれに対応する符号化出力データＥ２を生成する方法のステップが示され、この方法のステップは概括して参照番号１０で示されている。この方法は、次の１つ又は複数のステップを採用してもよいが、これらに限定されない：１つ又は複数の頻度テーブル２５を生成するステップ２０；１つ又は複数の符号テーブル３５を生成するステップ３０；最適符号化法を選択するために入力データＤ１を解析するステップ４０。こうした入力データＤ１を符号化する方法を実装する場合、入力データＤ１にある１つ又は複数のシンボルをそれに対応するインデクスに変換する利用可能な機構が存在しなくてはならない。例えば、複数のインデクスの中のある特定のインデクスは、例えば画素配列画像における画素値に等しい。インデクスが画素配列画像にある画素値から最小画素値を引いた差に等しくてもよい。こうした状況では、本方法は符号化出力データＥ２で最小画素値を送出する必要がある。すなわち、最小画素値は、方法１０又はその逆の方法を用いてエンコーダ５０からそれに対応するデコーダ６０に何らかの形で送出されなくてはならない。これが送出されないと、デコーダ６０は所定のシンボルブロックからそのインデクスを用いてそれに対応する元の値に復号することができなくなる。インデクスは複数の情報から生成されてもよい。例えば１つ又は複数の離散コサイン変換（ＤＣＴ）を介する場合、ＡＣ係数の絶対値を含む個別のＡＣ係数、ＡＣ係数の符号、非ゼロＡＣ／ＤＣ係数とその前の非ゼロＡＣ／ＤＣ係数との間にあるゼロＡＣ係数の実行、現在のＡＣ係数及び最後の非ゼロＡＣ係数に関する情報を表わす標示フラグ等から生成することができる。インデクスは、統合される複数の画素値に基づく生成の影響も受けやすい。こうした画素値には、例えば８ビットのＲ、Ｇ、Ｂの各画素値を含む２４ビットＲＧＢ画素や、２つの５ビットＹ画素値を含む１０ビット値がある。
【００９４】

次に、図３Ｂを参照する。デコーダ６０において対応するトランスコードデータＤ４を生成するためにデータＤ３をトランスコードする動作が、トランスコーダ７０を介して実装される。これは、例えば次のようなマルチキャストを行う状況において実装される：
（ｉ）複数のデバイスであって、各デバイスは符号化データＥ２を受け取るデコーダ６０をホストする；
（ｉｉ）複数のデバイスのうち少なくとも一部は互いに異なり、ディスプレイスクリーンレイアウト、解像度、アスペクト比、ディスプレイスクリーンドライバのバッファ容量等、関連する相異なる出力フォーマットを有する。

【００９５】

データＤ３の変換符号化は、デバイスのハードウェア層及び対応ソフトウェア層の何れか又は両方によって互換性のある仕方で処理可能な、対応するデータＤ４を生成するために必要とされる。ここでデバイスは、例えばスマートフォンや専門の科学装置、テレビ受像機、ハイファイ装置、ビデオ会議装置等の計算ハードウェアに基づいてもよい。トランスコーダ７０は、ソフトウェア、ＡＳＩＣ等の専用データ処理ハードウェアの何れか又は両方で実装される。

【００９６】

後述するように、方法１０を実装する場合、１つ又は複数のシンボル値を１つ又は複数の対応するインデクスに変換するステップが必ず含まれなくてはならない。このステップ又はその逆ステップは、デコーダ６０に伝送されるか、あるいはエンコーダ５０とデコーダ６０の両方に予め設定されていなくてはならない。こうしたやり取りを実現する最も単純なアプローチは、所定のシンボル値とそれに対応するインデクス値との間の直接的な関係を用いることである。例えば、インデクス値がそれに対応する画素値に等しくてもよく、あるいは、S＝符号フラグ、V＝１０ビット係数値、R＝６ビット非ゼロ実行値、L＝最終フラグとして、例えば次のように表現される数に等しくてもよい：
ＳＶＶＶＶＶＶＶＶＶＶＲＲＲＲＲＲＬ。

【００９７】

幾つかの理由により、こうした直接関係を常に利用できるとは限らない。例えば、直接関係が用いられるときには、次の状況の何れかがありうる：
（ａ）データやインデクス、出現頻度、出現確率、シンボル長の符号化又は復号は不可能又は非効率である；
（ｂ）相異なるインデクスの数が膨大である；
（ｃ）全てのシンボルが利用可能な出現頻度を持つわけではなく、このため、アルゴリズムによっては、こうしたシンボルに対するコードを生成することができない。

【００９８】

こうした（ａ）から（ｃ）の問題の一部は、エスケープコードを用いて、又は全シンボルに対する頻度情報を生成する論理を用いて少なくとも部分的に解決することができる。シンボルをインデクスに変換する他のアプローチを用いるのが有益であることも非常に多い。特定のアプローチは、利用可能なシンボルに用いられるインデクスを特定するルックアップテーブル（ＬＵＴ）の確保を必ず伴う。ここで、エスケープコードは、符号テーブルのサイズを小さくするのに非常に有益である。このＬＵＴはエンコーダ５０とデコーダ６０で利用可能でなくてはならない。あるいは、エンコーダ５０からデコーダ６０へ、又はその逆で送出されなくてはならない。高圧縮を実現するためにより適した符号化法が要求される場合、利用可能なＬＵＴのインデクスに基づいて選択可能な複数のテーブルを利用することが有益である。しかしこれは、場合によっては実用的ではない。それは、出現頻度又はコードワード長の組合せは膨大で、データメモリに全ての相異なるテーブルを格納する意味は無く、こうしたＬＵＴの送出でも、エンコーダ５０とデコーダ６０の間で多量のデータをやり取りしなくてはならないからである。したがって、本開示に従う方法１０は、シンボルからインデクスへの適切な変換を１つ又は複数用いることによって、出現頻度やコードワード長、出現確率をエンコーダ５０からデコーダ６０へ伝送する効率を向上させることができる。符号化法が、頻度情報として提供されるより正確な情報を利用で

きない場合、頻度の代わりにコードワード長を用いるのが常に有益である。例えば、ＶＬＣ符号化法は頻度情報を利用できないが、算術符号化、レンジ符号化であれば、頻度や確率の情報を利用することができる。

【００９９】

次に、本開示の実施形態の一例を詳述する。ここでは符号化にコードワード長が用いられているが、それに対応する実施形態において頻度情報や確率情報を用いるものも実現可能である。

【０１００】

符号化データＥ２として図２を参照する。符号化データは、例えば符号化データ列であって、値0から19までの20種類のシンボルを格納しうるデータ値を含む。現データ列では、このうち最小値が2、最大値が19の8種類のデータ値のみが実際に利用可能である。後述するように、こうした最小値と最大値は、テーブル送出の省力化を実現するために別々に送出されてもよい。それに対応する頻度は合計が148で、コードワード長は最小長が1、最大長が6、シンボルのインデクスは、例えば以下の表１に基づいている。これらのシンボルを圧縮しない場合、ビット列の搬送には148*5ビット＝740ビットが必要だと決められる。

【表１】

ビット列符号化の例

値	頻度	コードワード長	頻度１(ビット)	頻度２	頻度３(ビット)	インデクス１	インデクス２
2	7	4	4 (4)	84	128 (4)	2	0
4	2	6	1 (6)	24	32 (6)	4	1
7	81	1	32 (1)	972	1024 (1)	7	2
9	1	6	1 (6)	12	16 (7)	9	3
12	35	2	16 (2)	420	512 (2)	12	4
13	9	4	4 (4)	108	128 (4)	13	5
14	5	5	2 (5)	60	64 (5)	14	6
19	8	4	4 (4)	96	128 (4)	19	7

【０１０１】

表１に示されるような符号化形態では、適切な頻度テーブル又は符号テーブルが利用可能でない場合、例えば、エンコーダ５０とデコーダ６０で参照用インデクスが予め定義されるか特定可能であり、必要なコードワード、即ちコードとコード長をエンコーダ５０からデコーダ６０へ送信するのに利用可能な方法も複数存在する。

【０１０２】

第１の例示的方法は、データの出現頻度を修正してそれに対応する符号テーブルを生成する。ここで、最も確率の低い（最低確率）シンボル、即ち最も長いコードワードには追加ビットが割り当てられ、データの中で利用できない全てのシンボルには、符号化に影響を及ぼさないような長いコードワード長が割り当てられる。こうして、この方法により、エンコーダ５０とデコーダ６０は相互に類似する頻度テーブルとハフマンツリーを生成することができる。ここでは、12種類のシンボルが無いため、こうした出現頻度の修正は、例えば元の出現頻度に12を掛けておき、実際の出現頻度値を持たない、即ち頻度値がゼロである全てのシンボルに対する頻度値を1に設定することによって実装することができる。利用可能な頻度値を持つ全シンボルに対する修正後頻度値は、例えば、表１の頻度２という欄の記載のようになる。これらの新たな頻度に基づいて、全20種類のシンボルに対するコードワード長は次のように生成される：

11，11，4，11，6，11，11，1，11，7，11，11，2，4，5，10，10，10，10，4。

【０１０３】

このような符号テーブルを用いると、エンコーダ５０からデコーダ６０に符号化シンボ

ルを送出するのに必要なビット数は（7*4 + 2*6 + 81*1 + 1 *7 + 35*2 + 9*4 + 5*5 + 8*4 =）291ビットである。正しいデータ符号化とそれに次ぐデータ復号のためには、エンコーダ５０とデコーダ６０は同じ出現頻度を使用しなくてはならない。このような出現頻度は、前述の新しいコードワード長に基づいて生成することができ、出現頻度を持つシンボルに対する結果は、「頻度３」という欄の記載のようになる。この結果は$2^{(maxbitlen - bitlen)}$で表わされる。ここで、"maxbitlen"は「最大ビット長（maximum bit length）」の略であり、"bitten"は「ビット長（bit length）」の略である。他のシンボルには頻度として2（bitlen = 10）又は1（bitlen = 11）が割り当てられる。

【０１０４】

この第１の方法により、全ての可能なシンボル、即ち前述の例では20種のシンボルを含む符号テーブルを生成することができる。これは、他の同種のデータに対して同じ符号テーブルを使用する場合に有益である。この種のワード長は、追加情報を用いずに任意の圧縮法で圧縮することができる。そのため、可能性のあるあらゆる状況において、符号テーブルをエンコーダ５０からデコーダ６０へ容易に送出することができる。例えば、この符号テーブルは非圧縮では1コードワード長当り4ビットを要し、全コードワード長では、20コードワード長 * 4ビット／コードワード長＝80ビットとなる。

【０１０５】

この第１の例示的方法では、全ての最低確率シンボルに対して1ビットだけ使用するため、最適ハフマンコードと比べると非効率である。さらに、コードワード長、即ち符号テーブルを圧縮してエンコーダ５０からデコーダ６０へ送出するためのビット数も必要である。全ての可能なシンボル数である数20も送出されてもよく、あるいは、デコーダ６０が既知でもよい。

【０１０６】

次に、本発明の別の実施形態を示す第２の例示的方法について述べる。第２の方法は、利用可能なシンボル、即ち、出現頻度値が0よりも大きいシンボルに対してのみコードワード長を生成する。（表１を参照して）インデクス２は、ハフマン符号テーブルの生成に用いられるインデクスであり、インデクス１はエンコーダ５０からデコーダ６０へ送出されなくてはならないインデクスであって、シンボル値そのものである。このように生成されたコードワード長は、（表１を参照すると）「コードワード長（CWLen）」という欄の記載のようになる。このような符号テーブルを用いると、符号化シンボルを送出するのに必要なビット数は（7*4 + 2*6 + 81*1 + 1 *6 + 35*2 + 9*4 + 5*5 + 8*4 =）290ビットである。こうしたコードワード長に基づいて、エンコーダ５０とデコーダ６０は、表１の「頻度１」という欄の記載のような頻度を生成するように動作可能である。

【０１０７】

この種の符号テーブルを送出する第１の方法は、コードワード長とそのコードワードのインデクスを次の数のペアとして送出する：
(2, 4), (4, 6), (7, 1), (9, 6), (12, 2), (13, 4), (14, 5), (19, 4)。

【０１０８】

ここで、ペアは括弧を用いて表記される。

【０１０９】

こうした送出方法はインデクスに５ビット、コードワード長に３ビットを必要とするため、ペアには８ビットを要し、全ペアでは8 * 8ビット＝64ビットとなる。

【０１１０】

こうしたインデクスはデルタ符号化を行うことができ、ペアは次のようになる：
(2, 4), (2, 6), (3, 1), (2, 6), (3, 2), (1 , 4), (1 , 5), (5, 4)。

【０１１１】

こうして当然のことながら、インデクスに必要なビット数は3ビットのみとなり、コードワード長の３ビットと合わせてペアに必要なビット数は6ビット、符号テーブルの搬送に必要なビット数は8 * 6ビット＝48ビットとなる。

【０１１２】

こうしたインデクスとコードワード長の値が分離されていることによって、統合された８ビット又は６ビットの値と比べて改善された圧縮を実現し易いデータ列を持てるようになり、有益である。ここで、データ列は：
2，4，7，9，12，13，14，19
及び
4，6，1，6，2，4，5，4
であり、合計8＊5ビット＋8＊3ビット＝64ビットである。
第１列のインデクスはデルタ符号化され、
2，2，3，2，3，1，1，5
及び
4，6，1，6，2，4，5，4
であり、合計8＊3ビット＋8＊3ビット＝48ビットである。
これは、可能性のあるシンボルの種類が多いが、データには相異なるシンボルの種類が少ししか含まれていない場合には最もよい、即ち最適な送出方法である。

【０１１３】

こうした前述のデータ列は全て、動作時にエンコーダ５０からデコーダ６０へ圧縮して送出することができる。本開示のこの方法は最適ハフマン符号化と比べて非効率な点は無いが、この方法でも未だ、インデクスとコードワード長に関する情報を含むこうしたデータ列を送出するのに相当なビット数を消費する。利用可能なシンボル数である値８も送出されなくてはならない。さもなければ、復号対象ペア値の数をデコーダ６０が符号テーブルに確定できないためである。この場合、可能な全シンボル数である数20は、エンコーダ５０とデコーダ６０の間で送出不要である。

【０１１４】

前述の方法によっては、特に、エンコーダ５０で入力データＤ１に対して望ましい符号化を行ってそれに対応する符号化データＥ２を生成し、デコーダ６０でこの逆の動作を実現するために、組み合わせて利用しやすいものである。また、こうした解決方法の全てが、例えば第２の方法と同様の仕方で、利用可能なシンボルにのみハフマン符号を生成してもよい。さらに、こうしたデータ長から生成される頻度テーブルが（表１の）頻度１という欄の記載のようでもよい。

【０１１５】

第１の方法と比べると、第２の方法は、コードワード長が利用可能でない場合にはそれを0に設定することができる。しかしそれでも、こうした情報をエンコーダ５０からデコーダ６０へ送出しなくてはならない。これは、実際のコードワード長は決して0にはならないという事実から生じうることである。これにより、次のようなワード長のデータ列：
0，0，4，0，6，0，0，1，0，6，0，0，2，4，5，0，0，0，0，4
では、どのコードワード長についても元々3ビットしか必要でないので、合計すると20＊3ビット＝60ビットとなる。

【０１１６】

この種のワード長は、追加情報を用いずに任意の圧縮法で圧縮することができる。そのため、可能性のあるあらゆる状況において、符号テーブルをエンコーダ５０からデコーダ６０へ容易に送出することができる。また、このデータは値が0である数を頻繁に含むため、例えばＶＬＣやＲＬＥを用いて容易に圧縮できる。可能な全シンボル数である数20がエンコーダ５０からデコーダ６０へ送出されてもよく、デコーダ６０が既知であってもよい。

【０１１７】

別の実施形態では、どのデータが利用可能なシンボルでどのデータがそうでないかを特定するビットを使用する。２つのデータ列が生成される場合、第１のデータ列が含むビットと第２のデータ列が含むコードワード長は、それぞれ次のようになる：
0，0，1，0，1，0，0，1，0，1，0，0，1，1，1，0，0，0，0，1
及び

4，6，1，6，2，4，5，4。
この場合、各シンボルに対して1ビット、利用可能なコードワード長に対して3ビットを必要とするため、合計では20 * 1ビット＋8 * 3ビット＝44ビットとなる。これは通常、最もよい、即ち最適な送出方法である。

【０１１８】

こうしたデータ列は更に、ＥＭ（entropy modifier）ＶＬＣを利用して圧縮可能である。前述のように、可能な全シンボル数である数20がエンコーダ５０かとコーダ６０の間で送出されてもよく、既知としてデコーダ６０に与えられてもよい。また、利用可能なシンボル数である数8は、値が1であるビットから計算できるため、エンコーダ５０とコーダ６０の間で送出される必要はない。

【０１１９】

本開示に従う前述した全ての方法について、エンコーダ５０とデコーダ６０は、前述の例では20であった、使用されうる相異なるシンボルの数や、同じく前述の例では8であった、利用可能な相異なるシンボルの数に関する情報を処理しなくてはならない。こうした値、即ち相異なるシンボルの数がデコーダ６０で利用できない場合は、デコーダ６０に送出されなくてはならない。あるいは、利用可能なデータ値が含まれる範囲を特定する少量の追加情報を送信することによって、送出されるテーブルに含まれるデータの一部を節約してもよい。例えば、データ値が含まれる範囲を特定する値2（最小値）と19（最大値）を送出することも可能である。この実施例では、こうした送出で節約できるビット数よりも多くのビット数を使うことも度々あるが、例えば、8ビット画素が60から200までの値しか含まない場合は、エンコーダ５０からデコーダ６０へ伝送されるビット数を大幅に節約できる。こうした範囲の送出によって、最小値未満又は最大値を超える、使用されない全てのビット又は値をエンコーダ５０からデコーダ６０へ送出する必要がなくなる。また当然ながら、こうした範囲がエンコーダ５０からデコーダ６０へ送出される場合、インデクス値がデルタ符号化を用いて又は用いずに送信されなかった状況においては、最初と最後のインデクス値の送出は不要である。さらにこれは、最後の実施例における最初と最後の１ビットについても同様に適用できる。最小値と最大値の送出は、他の方法を用いる場合でも同様に利用することができる。このような方法としては、例えば英国特許出願第1303661.1号（出願日2013年3月1日、出願人はGurulogic Microsystems Oy）に開示されたODelta符号化があり、以下、参照によって本願に包含されるものとする。最小値と最大値を送出する最大の利点は、エントロピーを修正してエントロピー符号化を実装する全ての方法が同一情報を使用し、その情報が一度だけ送出される場合に発揮される。

【０１２０】

前述の方法は選択的に使用されてもよい。例えば、所定のデータチャンクであって、例えばエンコーダ５０からデコーダ６０へ伝送すべきデータ全体から分割されたデータチャンクの中で符号化対象シンボルの数に応じて、選択的に使用されてもよい。したがって、前述において明らかにされた方法は、利用可能な相異なるシンボルの数、それらの内、実際に使用されるシンボルの数、最小確率のシンボルの頻度、利用可能なシンボルのインデクスが可能なシンボルを通じて分配される仕方に応じて選択される。

【０１２１】

表１に示されたシンボルに関するスケール確率は、その頻度値に基づいて計算することもできる。シンボル数は例えば148である。この実施例におけるスケール確率は、２つの相異なる確率乗数、即ち256と32を用いて算出され、有益である。最初のシンボルについては、確率乗数256を用いると、スケール確率はRound(256 * 7/148)＝12として算出される。ここで、「Round」は整数値に切り上げる関数である。乗数256で算出される全てのスケール確率は次のようになる：
12，3，140，2，61，16，9，14。
スケール確率値の合計が257と256を超えているが、これは1ずつ減らしていくと有益である。こうした減算は、実際の符号化への影響を最小限にするよう実装されるのが有益である。例えばこの場合、最小値である値2を値1に減らしたり、最小の切り上げ値である値9

を値8に減らしたりしてもよい。これにより、レンジ符号化又は算術符号化に対して、乗数256でスケールした確率値は次のようになる：
12，3，140，1，61，16，9，14（合計256＝確率乗数）。
（乗数256による）スケール確率の送出は次のように行われる：
0，0，1，0，1，0，0，1，0，1，0，0，1，1，1，0，0，0，0，1
及び
12，3，140，1，61，16，9，14。
確率乗数の値が32の場合、スケール確率値は次のようになる：
2，0，18，0，8，2，1，2。
そして、合計値が等しくなるようにすると、次のようになる：
1，0，18，0，8，2，1，2。

【０１２２】

なお、当然のことながら、スケール確率値が0と算出されることもある。これは、こうした値が例えばエスケープシンボルを用いて送出されなくてはならないことを意味する。エスケープシンボルに対するスケール確率も算出される必要があるが、その値は1未満でなくてもよい。この場合、Round（32 *（2 + 1）/ 148）= 1であるから、値1が割り当てられる。したがって、このエスケープシンボル（"escape"）も他のシンボル群に追加されなくてはならず、新しいシンボルセットは次のようになる："escape"，2，7，12，13，14，及び19。こうした新しいシンボル群には、0から6の範囲でインデクスを割り当てるのが有益である。レンジ符号化又は算術符号化に関して、新しいシンボル群に対するスケール確率は、エスケープシンボルの確率が１つ又は複数の他のシンボルよりも小さい場合、次のようになる：
1 及び 1，18，8，2，1，1 （合計32＝確率乗数）。
（乗数32とエスケープシンボルを用いる）スケール確率の送出は次のように行われる：
0，0，1，0，0，0，0，1，0，0，0，0，1，1，1，0，0，0，0，1
及び
1，1，18，8，2，1，1。
当然のことながら、最初のシンボルはエスケープシンボルであり、ビットは他のシンボル値を特定している。

【０１２３】

前述のように、エスケープシンボルが定義されると、このテーブルは後でより使い易くすることができる。それは、このデータに現れなかったシンボルについても、それらが後でデータに現れるならば、エスケープシンボルを用いて送出することができるからである。

【０１２４】

レンジ符号化でエスケープシンボルを使用することは、別の特許出願において開示されている。この出願は、英国特許出願第1403038.1号、出願日2014年2月20日、出願人Gurulogic Microsystems Oy、発明の名称「エンコーダ、デコーダ及びその方法（Encoder，decoder and method）」であり、以下、参照によって本願に包含されるものとする。

【０１２５】

前述の表１に関する説明のように、データＤ１がエンコーダ５０で符号化され、次いでデコーダ６０で復号される場合、送出されるテーブルに代わって定義済みテーブルやインデクスで示されるテーブルを使用することもできる。これは、使用されるコード、頻度、確率、コード長の各種テーブルがエンコーダ５０とデコーダ６０には既知であること、あるいは、こうしたテーブルが別の限定的なテーブル群から選択されて、エンコーダ５０が選択したものをデコーダ６０へ送出することを意味する。定義済みテーブルは、デコーダ６０でローカルに利用可能でもよい。

【０１２６】

送出されたパラメータであって、例えばテーブルのインデクス、テーブルの最大インデクスの何れか又は両方に基づいて、そのテーブルを予め格納することもできる。あるいは

テーブルが、初期化機能に実装されたアルゴリズムや、予め格納済みのテーブルにあるアルゴリズムで生成されてもよい。こうしたテーブルの作成あるいは生成は、例えばテーブルのインデクス、テーブルの最大インデクス、テーブルの最小インデクスの何れか又は全てといった送出パラメータに基づいてもよい。予め格納済みのテーブルの代わりに、データＤ１を符号化して符号化データＥ２を生成する前に送出されたテーブルを使用してもよい。

【０１２７】

例えば、ＶＬＣ符号化が用いられる場合、コード長は通常、相異なるテーブルインデクスに対して事前に格納され、こうしたコード長の値に基づき、テーブル全体又はテーブル値の一部のみを用いて符号テーブルを生成することができる。この使用される部分は、送出されたテーブルパラメータやテーブルインデクスに基づいて定義されてもよい。同様に、レンジ符号化が用いられる場合、出現確率テーブルは通常、テーブルの形態とテーブル長に基づいて生成される。テーブル形態は、例えばテーブルインデクスを用いて送出されたということを指し、テーブル長は、例えばテーブルの最大インデクスが送出されたことを指す。

【０１２８】

次に、本開示の別の例示的方法について述べる。この方法は、個別のエスケープシンボルを用いなくても効率的なコード送出を可能にし、かつ、現在の所定のデータチャンクと、シンボル頻度が僅かに異なる将来のデータチャンクに関しても、全シンボルの効率的符号化を可能にする。この別の方法は、全シンボル、即ち利用可能な値を持つシンボルだけでなく利用可能な値を持たないシンボルを含めた全てにスケール確率として少なくとも１つの値を割り当てるように実装されてもよい。スケール確率値が1ならば可用性ビットは0に等しく、他のスケール確率値ならば可用性ビットは1に等しい。スケール確率値は、可用性ビットが1に等しいシンボルのみについて送出される必要がある。この方法の詳細は、Gurulogic Microsystems Oyが出願した英国特許出願第1403038.1号に記載されており、以下、参照によって本願に包含されるものとする。ただし、前述の実施例におけるテーブル送出は次のようになる：

0，0，0，0，0，0，0，1，0，0，0，0，1，0，0，0，0，0，0，0

及び

10，4（合計18*1 + 10 + 4 =32＝確率乗数）。

【０１２９】

このように、この実施例は、データのエントロピーを符号化するハフマン符号化で実現されるものよりもむしろ、レンジ符号化にかなり近い性能を生み出すような有益な状況を示している。こうした解決方法は、符号確率テーブルを非常に効率的に送出することができる。他の種類のデータや他の種類の確率乗数値を用いても、この解決方法は、あらゆる状況において採用されうる最良、即ち最適な符号化方法であることが明白である。こうした理由で、符号テーブルの送出について、より詳細に述べている。

【０１３０】

エンコーダ５０とデコーダ６０は、静的に又は動的に更新可能な仕方で全てのテーブルを格納することができる。こうしたやり方はエンコーダ５０とデコーダ６０の両方で利用されるべきものである。テーブルが格納されれば、そのテーブルは、それに対応する格納済みテーブルを一意に特定するインデクスを用いて、エンコーダ５０からデコーダ６０へ送信されるデータにおいて有益に特定される。こうしたテーブルのインデキシングは、エンコーダ５０からデコーダ６０へ符号テーブルを送出するのに必要な送信データと比べてオーバヘッドデータを大幅に節約できる可能性がある。

【０１３１】

予め格納済みのテーブルを用いる例は、例えば次の実施例で理解することができる。送出テーブルについて、前述の実施例で示した最後の確率テーブル、即ち、1，1，1，1，1，1，1，10，1，1，1，1，4，1，1，1，1，1，1，1は、再利用目的で追加されるように選択される。この目的のため、テーブルには例えば「17」というインデクス値が割り当てら

れる。そして、符号化対象である新しいデータチャンクが必要であり、それは、表２に示されるようなシンボル値と頻度を有する。

【表２】

テーブル再利用のための第2のシンボル値及び頻度の例

値	頻度
1	3
5	1
7	68
8	1
10	4
12	32
14	3
18	1

【０１３２】

レンジ符号化における全ての利用可能な確率テーブルは、表２から評価することができる。その結果、レンジ符号化を用いてこの新しいデータをエンコーダ５０からデコーダ６０へ送出するために使用されるべき最良の確率テーブルとして、テーブル17の選択が最も確からしい。少なくとも、理想的なエントロピー符号化の結果に関してテーブル17が生成する追加データよりも、新しい確率テーブルの送出の方が多くのデータを必要とすることは容易に理解できる。したがって、新しいより最適化された確率テーブルの送出を求める代わりに、テーブル17又は他の確率テーブルを再利用することができる。テーブルが再利用される場合、そのインデクスである17がエンコーダ５０からデコーダ６０へ送出され、次いでレンジ符号化値がエンコーダ５０からデコーダ６０へ送出される。テーブルを再利用できない場合、始めに新しいテーブル送出を規定する値として例えば0や次の利用可能なテーブルのインデクスがエンコーダ５０からデコーダ６０へ送出され、次にテーブルがエンコーダ５０からデコーダ６０へ送出される。またその次に、レンジ符号化値をエンコーダ５０からデコーダ６０へ送出することもできる。符号化シンボルは通常、レンジ符号化データ等の前に送出されなくてはならないが、それによって、デコーダ６０でデータを適切に復号することができる。

【０１３３】

既に使用中の符号テーブルを修正し、その修正箇所のみを送信して、新しい符号テーブルを得るようにしてもよい。さらに、送出／受取後に送出済み符号テーブルが適応的に使用されてもよい。

【０１３４】

エンコーダ５０とデコーダ６０は、同種のデータであって、現在欠落しているシンボルがあるかもしれないデータにおいて、後で使用されうる他のシンボルの符号化も可能にする類似の完全テーブルを作成するように動作可能でもよい。こうしたテーブルは、新しい符号テーブルインデクスを持って格納されてもよい。テーブルは、不完全及び完全という両方の種類を格納することができる。また、元のテーブルのみを格納してもよい。次のときに完全な特性が必要である場合は、エンコーダ５０からデコーダ６０への伝送において別々にインデキシングされてもよい。完全テーブルを格納する解決方法の方が好ましいのは、決定を単純化して、必要とされるテーブルが充足しているか否かを示す追加標示の送出を不要にするからである。テーブルの充足は、全ての値を含むテーブルが、例えば4から1までの降順になっている頻度で充足されるように実装されてもよい。それによって、次のシンボルには比較的短いシンボルが割り当てられ、最後のシンボルには比較的長いシンボルが割り当てられる。こうしたアプローチを採用することによって、シンボルの順序は、将来のデータ列における利用可能なシンボルの順序に対応する。

【０１３５】
前述の実施例において、コードワード長の送出とスケール確率の送出は有益に利用される。ただし、同様の技術をハフマン符号化や算術符号化、レンジ符号化で必要な頻度値の送出のために用いることもできる。採用すべき最良の符号化方法は、符号テーブルやコードワード長、確率テーブル、頻度テーブルの送出に必要なオーバヘッド情報に符号化データが追加される場合、最小限のビットを使用する。これによって、例えば、データの小部分、即ちデータチャンクに含まれるデータの性質、種類の何れか又は両方に対して具体的に最適化された符号化方法を用いて、このデータの小部分をエンコーダ５０からデコーダ６０へ送ることができる。こうした理由で、頻度値が少なくとも限られた範囲で量子化される場合に、算術符号化とレンジ符号化による最良の結果が得られる。それによって、頻度テーブルは、ほぼ正確な値を示しつつも厳密なテーブルよりも明らかに少ないビット数で送出可能であり、例えばエントロピー符号化を用いる場合、シンボルの符号化に関する最適性が僅かに減るだけで済む。さらに、スケール確率テーブルの送出によって、非常に効率がよくほぼ最適なレンジ符号化及び算術符号化の実装が可能になる。

【０１３６】
エンコーダ５０からデコーダ６０へ伝送すべきデータが少量しかない場合、通常、実質的に如何なる符号化形態も用いずに通信する方がよい。しかし、データ量が増加する場合、ほぼ正確なコードワード長でハフマン符号化を用いるのが有益である。エンコーダ５０からデコーダ６０へ伝送すべきデータ量は増加しているため、使用する符号テーブルが正確であるほど、それに応じて得られる利益も大きくなる。また、算術符号化又はレンジ符号化が行われるデータが相当量ある場合、採用するエントロピー符号化の効率がよい程、最良の符号化結果が得られる。それによって、符号化の際、符号テーブルの送出よりも頻度又は確率の送出の方が、必要なビット数の点で有益となる。確率テーブルや頻度テーブル、コードワード長、符号テーブルのインデクスの送出は常に似通っていて、所定のインデクスが使用される場合、利用可能な最良テーブルを有する方法によって、最良の圧縮性能を実現することができる。使用される符号化方法の選択がデータやデータ量に基づいて定まらなければ、その選択もエンコーダ５０からデコーダ６０へ送出されなくてはならない。

【０１３７】
エンコーダ５０とデコーダ６０を一体にしてコーデック１００が形成される。実践的な状況では、１つのエンコーダ５０と１つ又は複数のデコーダ６０があってもよい。例えば、エンコーダ５０が符号化データＥ２を生成し、それが、例えば無線、光ファイバ等の通信ネットワークを介して広範囲に亘る多数のデコーダ６０にブロードキャスト、即ち「マルチキャスト」される。また、例えばピアツーピアデータ通信ネットワークでは、このピアツーピアネットワーク内で接続される複数のエンコーダ５０からデコーダ６０に与えられる符号化データＥ２が、複数の符号化データチャンクで別々に与えられ、デコーダ６０で一緒に集められるという状況も生じる。こうした構成は、符号化データＥ２の特定部分がより局在的にデコーダ６０に与えられ、こうしたピアツーピアネットワークの実装に利用される長距離データ通信ネットワークのデータ負荷を軽減でき、有益である。エンコーダ５０とデコーダ６０は、専用のデジタルハードウェアに容易に実装することができる。あるいは、非一時的データ記憶媒体に保存された１つ又は複数のソフトウェア製品を実行可能なコンピュータハードウェアに実装されてもよく、これらの組合せでもよい。エンコーダ５０とデコーダ６０は、以下に限定されないが、音声録音・再生装置、ビデオ録画・再生装置、パーソナルコンピュータ、スマートフォン、デジタルカメラ、ビデオカメラ、テレビ受像機、インターネット端末、科学装置、監視・セキュリティシステム、地上監視機能用に構成された衛星、地震感知システムで利用可能である。

【０１３８】
本開示の実施形態は、例えば符号テーブルや頻度テーブル、確率テーブル、コードワード長等、テーブルのより効率的な送出を可能にする。それによって、優れた形でデータをより小さいデータチャンクに分割でき、例えば、そうしたデータチャンクを最適な仕方で

個別に符号化することができる。さらに、テーブルはエントロピー符号化法を用いて符号化されてもよい。こうした小さいデータチャンクにはそれに対応する符号テーブルや頻度テーブル、確率テーブル、コードワード長が必要になることもある。これは、こうした様々なテーブルが利用可能である場合には、相異なるデータチャンクに対して所定のテーブルのインデクスのみを送出するだけでよいため有益である。そうでないと、新しいテーブルをデコーダ６０に送出しなくてはならなくなる。所定のテーブルが送出される場合、そのテーブル固有の参照インデクスで後から利用できるように、例えばデコーダ６０のデータメモリにテーブルを格納するのが有益である。

【０１３９】

本開示の例示的実施形態では、全てのデータブロックは個別のデータブロックとして送出される。データブロックは、類似データ群に属するデータブロックの数を示す補助情報を伴って送出される。しかし、こうしたデータブロックのやり取りは通常、採用される符号化方法、シンボル数、使用される符号テーブルや頻度テーブル又は確率テーブルに関する識別情報を全データブロックに対して送出する必要があるため、極めて非効率である。加えて、類似データ群に属するデータブロックの数も、エンコーダ５０からデコーダ６０へ送出されなくてはならない。

【０１４０】

符号テーブルには、それぞれに対応する意味を有する１つ又は複数の追加シンボルの挿入が有益に行われてもよい。通常、大きい符号テーブルでは、「エスケープ（escape）」シンボルが特有のコードワードを持つことが有益である。さらに、ＪＰＥＧのＤＣＴ係数に用いられる符号テーブルには、「係数終了（end of coefficients）」シンボルに対する特有のコードワードが利用可能であることが有益である。これは、こうした方法がデコーダ６０には既知であって、例えば、「符号テーブル変更（change of code table）」や「データ終了（end of data）」、また必要であれば「エスケープ」に使用可能な新しい符号シンボルを追加することによって、その方法が極めて効率のよい仕方で利用できることを意味する。こうした追加シンボルは、それが使用される度にその頻度が1となるように生成することができる。利用可能なテーブルが使用される場合、最長コードワードを有するシンボルの１つでコードを分割するように、対応する識別情報がシンボルとして追加される。例えば、現在のデータチャンクの後に新しいデータチャンクがあってデータが残っている場合、エンコーダ５０は、「データ終了」シンボルではなく「符号テーブル変更」シンボルを使うのが有益である。この「符号テーブル変更」シンボルが送出されると、その後に新しい符号テーブルのインデクスが送出される。この新しい符号テーブルを規定するインデクス値は、例えば、利用可能なテーブルが存在しないときも使用され、利用可能なテーブルが既に存在する場合は1からテーブル数までの値が用いられる。この新しい符号テーブルに対するインデクスは、そのデータに対して利用可能又は適切な全てのテーブルを表わすのに必要なビット数を使って表わされる値を有してもよい。新しい符号テーブルのインデクスとして値0が送出された場合、エンコーダ５０からデコーダ６０に提供されるデータ列に次のシンボルが符号化される前に、符号テーブルの送出が必要とされる。それ以外の場合は、新しい符号テーブルを特定するインデクスの後、この新しい符号テーブルで新しいシンボルを即座に符号化することができる。最後のデータチャンクが符号化されて最後のデータ値がデコーダ６０へ送出されると、エンコーダ５０は、「データ終了」シンボルを送出する。この場合、「データ終了」シンボルのみが有効で、「符号テーブル変更」シンボルは使用されない。「データ終了」シンボルが送出されると、それ以降データの送出を継続する必要はない。この「データ終了」シンボルを用いることで、各データチャンクに対してデータ値の数の送出は不要になる。さらに、相異なるデータチャンクに対しては使用される符号テーブルや頻度テーブル、確率テーブル、コードワード長のみが変更されるため、符号化方法をデコーダ６０へ送出する必要も無い。したがって、エンコーダ５０からデコーダ６０へ送信されるオーバヘッドデータの総量は、データの符号化とそれに続く復号の最中に符号テーブルが変更される場合には相当小さくなる。シンボルが「符号テーブル変更」や「データ終了」に関連する場合、1ビットの終了ビットの検

出が必要である。あるいは、頻度1の２つのシンボルを符号テーブルに生成することも可能である。

【０１４１】

場合によっては、データ値の量や符号化データの量を送信したり、データ、データ量、使用される符号化方法、デコーダ６０及びエンコーダ５０の実装に応じて「データ終了」シンボルを使用したりすることも有益である。さらに、エンコーダ５０とデコーダ６０の何れか又は両方でデータを処理するときに並列化を採用することも有益である。これは即ち、符号化データが送出され、デコーダ６０が別々のプロセッサ、処理、スレッドに対してデータを容易に分割できることである。復号対象データ値の量に関する情報の送出に最適なアプローチには、通常様々なものが存在するが、こうした場合においても、どのアプローチを選択したかについては送出不要である。ただし、最適な選択が複数利用可能である場合、エンコーダ５０は、特定の方法を選択し、その最適な選択に対応する決定をデコーダ６０へ送出する。

【０１４２】

前述から当然のことながら、例えば図３Ａに示すように、データＤ１とデータＤ３が実質的に互いに類似するものである場合、デコーダ６０は、エンコーダ５０で実行される符号化機能の逆変換を実質的に実装している。しかし

例えば符号化データＥ２を複数の相異なるデバイスにマルチキャストする等といった種々の状況では、データＤ３をトランスコードしてそれに対応するトランスコードデータＤ４を生成するトランスコーダ７０の利用を必要とする。トランスコードデータＤ４は、図３Ｂに示すように、デコーダ６０と関連するトランスコーダ７０をホストする所定のデバイスに対応している。デコーダ６０とトランスコーダ７０の両方とも、コンピュータハードウェアを用いて実装されてもよく、あるいは、トランスコーダ７０が、ハードウェアドングルのような専用のトランコードハードウェアに実装されてもよい。本開示の実施形態は、データに対する可逆又は非可逆の符号化及び復号を提供するように構成されることに関する。デコーダ６０は、例えば、データ（Ｄ１）のレンダリングに必要なものとは異なるディスプレイデバイスにデータを提供するエンコードを実行可能でもよい。このような場合、コーデック１００を通じて処理されるデータは、元のフォーマットに復号することはできない。その代わり、符号化データＥ２は、例えば、他のフォーマットに直接変換され、スクリーン等にレンダリングされたり、ファイルに保存されたりする。こうしたトランスコードの実施例の一つは、データＤ１が元々ＹＵＶフォーマットで、圧縮されてレシーバに送信される。レシーバはこのデータをブロック毎に復元し、色変換を行い、完全解像度を持つＹＵＶ画像を再構成せず、スクリーンへのレンダリングに適したＲＧＢ画像にスケールする。

【０１４３】

デコーダ６０は、エンコーダ５０が生成した符号化データ（Ｅ２）を復号し、それに対応する復号データ（Ｄ３）を生成する方法の利用が可能である。本方法は次のステップを含む：

（ｉ）符号化データ（Ｅ２）を受け取り、１つ又は複数の頻度テーブル、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルの何れか又は全て、及び／又は、これら１つ又は複数のテーブルを示す情報と共に、１つ又は複数のインデクスセットを符号化データから抽出すること；

（ｉｉ）１つ又は複数のインデクスセットから、１つ又は複数のデータチャンクにおいて対応するシンボル、及び／又は、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルの何れか又は全てにおいてエントリの圧縮シンボルを計算すること；

（ｉｉｉ）前記シンボルから、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルの何れか又は全てからの情報を用いて、１つ又は複数のデータチャンクを再生成すること；

（ｉｖ）復号データ（Ｄ３）を生成するために、１つ又は複数のデータチャンクを結合及

び／又は変換する。

【０１４４】

本方法は、１つ又は複数のテーブルの少なくとも１つが後で再利用できるように保存可能な仕方で、１つ又は複数のテーブルの少なくとも１つを受け取ることを含んでもよい。

【０１４５】

本方法は、復号データ（Ｄ３）を生成するステップ（ｉｖ）において、１つ又は複数のデータ伸張アルゴリズムを適用することを含んでもよい。本方法では更に、１つ又は複数のデータ伸張アルゴリズムは、ハフマン復号、ＶＬＣ復号、エントロピー復号、算術復号、レンジ復号の少なくとも１つを含んでもよい。

【０１４６】

本方法は、復号データ（Ｄ３）を生成するために、１つ又は複数のデータチャンクのうち複数を実質的に同時に処理する並列プロセッサアーキテクチャを用いることによって、その複数のデータチャンクを結合することを含んでもよい。

【０１４７】

本方法は、一緒に結合された複数のデータ値に基づいて、１つ又は複数のインデクスセットを生成することを含んでもよい。本方法では更に、このインデクスは、Ｒ、Ｇ、Ｂの画素値又はＹ、Ｕ、Ｖの画素値を含む１つ又は複数のＲＧＢ画素から算出されてもよい。本方法は更に、データチャンクが符号化データ（Ｅ２）に含まれる場合、そのデータチャンクに対して達成可能なデータ伸長率に応じて、符号化データ（Ｅ２）における１つ又は複数のデータチャンクの非符号化生成又は符号化生成を動的に切り換えることを含んでもよい。

【０１４８】

本方法では、デコーダ６０は、シンボルが符号化データ（Ｅ２）から、「符号テーブル変更」又は「データ終了」に関連するかを示す終了ビットを少なくとも１つ抽出するよう動作可能でもよい。

【０１４９】

本方法は、所定のデータチャンクに存在する１つ又は複数のシンボルを参照するために必要十分なインデクスから、実質的にその所定のデータチャンクを生成することを含んでもよい。

【０１５０】

本方法は、符号化データ（Ｅ２）に含まれる１つ又は複数の符号テーブルを伸張することを含んでもよい。本方法は更に、ハフマン復号を用いて１つ又は複数の符号テーブルを伸張することを含んでもよい。本方法では更に、１つ又は複数の符号テーブルの伸張は、１つ又は複数の補助符号テーブルを用いてもよい。

【０１５１】

本方法は、次の送信データを復号するために、１つ又は複数の符号テーブルをデコーダ（６０）で使用可能にする仕方で１つ又は複数の符号テーブルを受け取ることを含んでもよい。

【０１５２】

本方法は、１つ又は複数の符号テーブルがアクセスされ易い場所を示す１つ又は複数の識別コードを、１つ又は複数のデータベースと１つ又は複数のプロキシデータベースの何れか又は両方を介して、符号化データ（Ｅ２）から抽出することを含んでもよい。

【０１５３】

本方法は、次の種類のデータ：録音音声信号、録画ビデオ信号、静止画像、テキストデータ、感震データ、センサ信号データ、アナログ－デジタル変換（ADC）データ、生体信号データ、カレンダーデータ、経済データ、数学的データ、二進数データの１つ又は複数を復号することを含んでもよい。

【０１５４】

本方法は、複数のデータソースから符号化データ（Ｅ２）を受け取り、符号化データ（Ｅ２）を再生成するために、ソースから提供されたデータを結合することを含んでもよい

。

【０１５５】

デコーダ６０は、復号データ（Ｄ３）を生成するために符号化データ（Ｅ２）を復号する前述の方法を実装するように動作可能であり、エンコーダ５０が生成した符号化データ（Ｅ２）を復号するデコーダ６０が提供される。デコーダ６０は、

（ｉ）符号化データ（Ｅ２）を受け取り、１つ又は複数の頻度テーブル、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルの何れか又は全て、及び／又は、これら１つ又は複数のテーブルを示す情報と共に、１つ又は複数のインデクスセットを符号化データから抽出すること；

（ｉｉ）１つ又は複数のインデクスセットから、１つ又は複数のデータチャンクにおいて対応するシンボル、及び／又は、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルの何れか又は全てにおいてエントリの圧縮シンボルを計算すること；

（ｉｉｉ）前記シンボルから、１つ又は複数の符号テーブル、１つ又は複数のコードワード長テーブル、１つ又は複数の確率テーブルの何れか又は全てからの情報を用いて、１つ又は複数のデータチャンクを再生成すること；

（ｉｖ）復号データ（Ｄ３）を生成するために、１つ又は複数のデータチャンクを結合及び／又は変換すること

を実行するように動作可能である。

【０１５６】

デコーダ６０は更に、対応するトランスコードデータ（Ｄ４）を生成するために復号データ（Ｄ３）をトランスコードするトランスコーダ７０、及び／又は、符号化データ（Ｅ２）からそれに対応するトランスコードデータ（Ｄ４）を生成することを含んでもよい。

【０１５７】

デコーダ６０は、１つ又は複数のテーブルの少なくとも１つが後で再利用できるように保存可能な仕方で、１つ又は複数のテーブルの少なくとも１つを受け取るように動作可能でもよい。

【０１５８】

デコーダ６０は、復号データ（Ｄ３）を生成するステップ（ｉｖ）において、１つ又は複数のデータ伸張アルゴリズムを適用するように動作可能でもよい。デコーダ６０では更に、１つ又は複数のデータ伸張アルゴリズムは、ハフマン復号、ＶＬＣ復号、エントロピー復号、算術復号、レンジ復号の少なくとも１つを含んでもよい。

【０１５９】

デコーダ６０は、復号データ（Ｄ３）を生成するために、１つ又は複数のデータチャンクのうち複数を実質的に同時に処理する並列プロセッサアーキテクチャを用いることによって、その複数のデータチャンクを結合するように動作可能でもよい。

【０１６０】

デコーダ６０は、一緒に結合された複数のデータ値に基づいて、１つ又は複数のインデクスセットを生成するように動作可能でもよい。デコーダ６０では更に、このインデクスは、Ｒ、Ｇ、Ｂの画素値又はＹ、Ｕ、Ｖの画素値を含む１つ又は複数のＲＧＢ画素から算出されてもよい。デコーダ６０は更に、データチャンクが符号化データ（Ｅ２）に含まれる場合、そのデータチャンクに対して達成可能なデータ伸長率に応じて、符号化データ（Ｅ２）における１つ又は複数のデータチャンクの非符号化生成又は符号化生成を動的に切り換えるように動作可能でもよい。

【０１６１】

デコーダ６０は、シンボルが符号化データ（Ｅ２）から、「符号テーブル変更」又は「データ終了」に関連するかを示す終了ビットを少なくとも１つ抽出するよう動作可能でもよい。

【０１６２】

デコーダ６０は、所定のデータチャンクに存在する１つ又は複数のシンボルを参照する

ために必要十分なインデクスから、実質的にその所定のデータチャンクを生成するように動作可能でもよい。

【０１６３】

デコーダ６０は、符号化データ（Ｅ２）に含まれる１つ又は複数の符号テーブルを伸張するように動作可能でもよい。デコーダ６０は更に、ハフマン復号を用いて１つ又は複数の符号テーブルを伸張するように動作可能でもよい。デコーダ６０では更に、１つ又は複数の符号テーブルの伸張は、１つ又は複数の補助符号テーブルを用いてもよい。

【０１６４】

デコーダ６０は、次の送信データを復号するために、１つ又は複数の符号テーブルをデコーダ（６０）で使用可能にする仕方で１つ又は複数の符号テーブルを受け取るように動作可能でもよい。

【０１６５】

デコーダ６０は、１つ又は複数の符号テーブルがアクセスされ易い場所を示す１つ又は複数の識別コードを、１つ又は複数のデータベースと１つ又は複数のプロキシデータベースの何れか又は両方を介して、符号化データ（Ｅ２）から抽出するように動作可能でもよい。

【０１６６】

デコーダ６０は、次の種類のデータ：録音音声信号、録画ビデオ信号、静止画像、テキストデータ、感震データ、センサ信号データ、アナログ－デジタル変換（ADC）データ、生体信号データ、カレンダーデータ、経済データ、数学的データ、二進数データの１つ又は複数を復号するように動作可能でもよい。

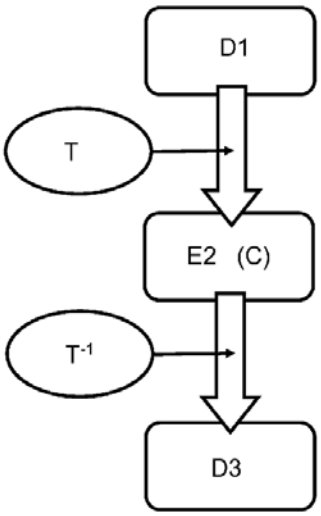
【０１６７】

デコーダ６０は、複数のデータソースから符号化データ（Ｅ２）を受け取り、符号化データ（Ｅ２）を再生成するために、ソースから提供されたデータを結合するように動作可能でもよい。例えば、エンコーダ５０からデコーダ６０へ符号化データ（Ｅ２）を送るピアツーピアデータ通信ネットワークに複数のソースが含まれている。

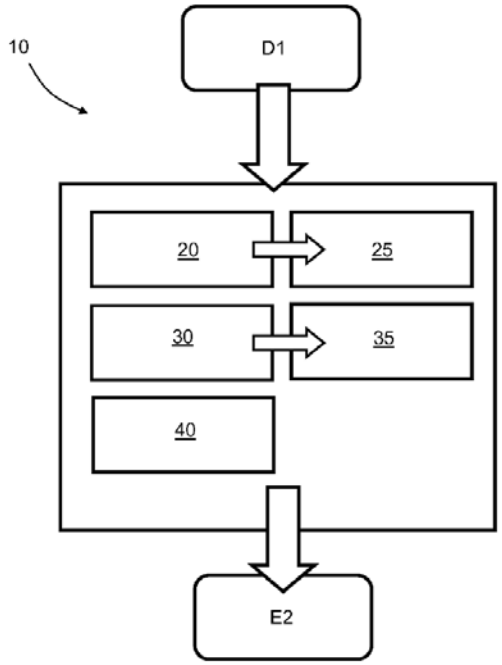
【０１６８】

前述した本発明の実施形態への修正は、添付の特許請求の範囲に定義される発明の範囲を逸脱しない限りにおいて可能である。本発明の記述と特許請求の範囲で用いられる「含む」，「備える」，「包含する」，「構成される」，「有する」，「存在する」等の表現は、包括的構成であると解釈されることを意図しており、明示的に記載されていないアイテムや部品，構成要素も含まれうることを意図している。単数表現もまた、複数に関連するものと解釈されるべきものである。添付の特許請求の範囲における括弧内の数字は、請求項の理解を助けることを意図したものであり、これらの請求項によって定義される発明の範囲を限定するように解釈されるべきではない。

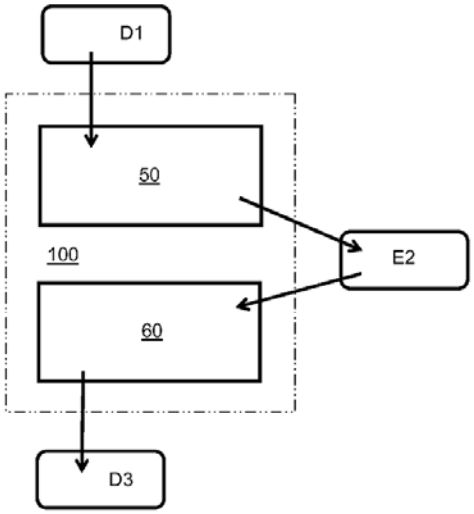
【図１】


D1
T
E2 (C)
T-1
D3

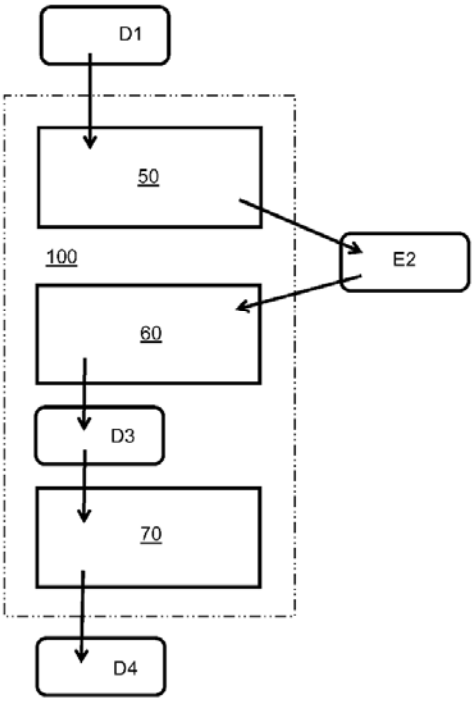
【図２】


10
D1
20
25
30
35
40
E2

【図３Ａ】


D1
50
100
E2
60
D3

【図３Ｂ】


D1
50
100
E2
60
D3
70
D4

フロントページの続き

(72)発明者 カルッカイネン トゥオマス
フィンランド共和国 ＦＩＮ－２０３２０ トゥルク ラウタランカツ ２ Ｂ１７
(72)発明者 フフタニエミ アルトゥル
フィンランド共和国 ＦＩＮ－２０１００ トゥルク リンナンカツ ３４

審査官 北村 智彦

(56)参考文献 特表２００２－５１５２０１（ＪＰ，Ａ）
特開２００１－１１１８４３（ＪＰ，Ａ）
特開平０５－２４１７７４（ＪＰ，Ａ）
特開平１０－０５１７７１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０３Ｍ ３／００－１１／００
ＩＥＥＥ Ｘｐｌｏｒｅ