

【公報種別】特許法第17条の2の規定による補正の掲載
 【部門区分】第7部門第3区分
 【発行日】平成18年1月5日(2006.1.5)

【公表番号】特表2001-521348(P2001-521348A)
 【公表日】平成13年11月6日(2001.11.6)
 【出願番号】特願2000-517560(P2000-517560)
 【国際特許分類】

H 0 3 M 13/27 (2006.01)

H 0 3 M 13/01 (2006.01)

H 0 4 N 7/26 (2006.01)

【F I】

H 0 3 M 13/27

H 0 3 M 13/01

H 0 4 N 7/13 A

【手続補正書】

【提出日】平成17年10月24日(2005.10.24)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【書類名】明細書

【発明の名称】 伝送時のデータ欠落によるエラーを強力に回復する原始コーディング方法

【特許請求の範囲】

【請求項1】 複数の信号成分(S E)からなる信号に対する強力エラー回復を実現する原始コーディング方法において、

上記各信号成分が、可変長データを有する複数のビットによって表されるように該各信号成分を符号化するステップと、

上記符号化された信号成分を複数の信号成分のセットに分割するステップと、

上記各信号成分のセットについて、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成するステップと、

上記信号成分のセットの信号成分を記述するパラメータが複数のビットストリームに亘って分散するように、上記各ビットストリームの複数のビットを該複数のビットストリームに亘って分散するステップとを有する原始コーディング方法。

【請求項2】 上記符号化された信号成分のビットを上記複数のビットストリームに亘って分散するステップの前に、バッファ内に符号化された信号成分を分散するステップを有することを特徴とする請求項1記載の原始コーディング方法。

【請求項3】 上記信号は、ビデオ信号であることを特徴とする請求項1記載の原始コーディング方法。

【請求項4】 上記複数のビットストリームから各ビットストリームのビットを再分散するステップと、

上記各ビットストリームの上記複数の符号化された信号成分を再割当てするステップと、

上記複数の符号化された信号成分から上記信号を再構築するステップとをさらに有する請求項1記載の原始コーディング方法。

【請求項5】 上記各符号化された信号成分を表すビットを、分散するステップの前の位置に戻すように、上記複数のビットストリームから各ビットストリームのビットを再分散

するステップと、

上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てするステップとをさらに有する請求項 1 記載の原始コーディング方法。

【請求項 6】 複数の信号成分 (SE) からなる信号に対する強力エラー回復を実現する原始コーディングを実行するプロセッサを備えるデジタル信号処理システムにおいて、

上記プロセッサは、

上記各信号成分が、可変長データを有する複数のビットによって表されるように該各信号成分を符号化するステップと、

上記符号化された信号成分を複数の信号成分のセットに分割するステップと、

上記各信号成分のセットについて、該信号成分のセットの各符号化された信号成分を表す上記複数のビットからなるビットストリームを形成するステップと、

上記信号成分のセットの信号成分を記述するパラメータが複数のビットストリームに亘って分散するように、上記各ビットストリームの複数のビットを該複数のビットストリームに亘って分散するステップとを実行することを特徴とするデジタル信号処理システム。

【請求項 7】 上記プロセッサは、さらに、上記符号化された信号成分のビットを上記複数のビットストリームに亘って分散する前に、バッファ内に符号化された信号成分を分散することを特徴とする請求項 6 記載のデジタル信号処理システム。

【請求項 8】 上記信号は、ビデオ信号であることを特徴とする請求項 6 記載のデジタル信号処理システム。

【請求項 9】 上記プロセッサは、さらに、

上記複数のビットストリームから各ビットストリームのビットを再分散し、

上記各ビットストリームの上記複数の符号化された信号成分を再割当てし、

上記複数の符号化された信号成分から上記信号を再構築することを特徴とする請求項 6 記載のデジタル信号処理システム。

【請求項 10】 上記プロセッサは、さらに、

上記各信号成分を表すビットを、分散するステップの前の位置に戻すように、上記複数のビットストリームから各ビットストリームのビットを再分散し、

上記複数のビットストリームの符号化された信号成分を上記元の信号成分のセットに再割当てすることを特徴とする請求項 6 記載のデジタル信号処理システム。

【請求項 11】 プロセッサにより実行可能な命令を格納するコンピュータにより読取可能な記録媒体であって、上記プロセッサに、

上記各信号成分が、可変長データを有する複数のビットによって表されるように該各信号成分を符号化するステップと、

上記符号化された信号成分を複数の信号成分のセットに分割するステップと、

上記各信号成分のセットについて、該信号成分のセットの各符号化された信号成分を表す上記複数のビットからなるビットストリームを形成するステップと、

上記信号成分のセットの信号成分を記述するパラメータが複数のビットストリームに亘って分散するように、上記各ビットストリームの複数のビットを該複数のビットストリームに亘って分散するステップとを実行させて、複数の信号成分 (SE) からなる信号に対する強力エラー回復を実現する原始コーディングを実行させる命令が格納されていることを特徴とするコンピュータにより読取可能な記録媒体。

【請求項 12】 上記プロセッサに、上記符号化された信号成分のビットを上記複数のビットストリームに亘って分散する前に、バッファ内に符号化された信号成分を分散するステップとを実行させる命令が格納されていることを特徴とする請求項 11 記載のコンピュータにより読取可能な記録媒体。

【請求項 13】 上記プロセッサに、

上記複数のビットストリームから各ビットストリームのビットを再分散するステップと、

上記各ビットストリームの上記信号成分のセットの符号化された信号成分を再割当てするステップと、

上記複数の信号成分のセットから信号を集めるステップとを実行させる命令がさらに格納されていることを特徴とする請求項 1 1 記載のコンピュータにより読取可能な記録媒体。

【請求項 1 4】 各信号成分 (S E) が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を復号する復号方法において、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、

上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散するステップと、

上記各ビットストリームから上記複数の符号化された信号成分を再割当てするステップと、

上記複数の符号化された信号成分から上記信号を再構築するステップとを有する復号方法。

【請求項 1 5】 各信号成分 (S E) が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を再生する再生方法において、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、

上記各信号成分を表すビットを、分散する前の位置に戻すように、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散するステップと、

上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てするステップとを有する再生方法。

【請求項 1 6】 各信号成分 (S E) が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を復号するように構成された信号処理システムにおいて、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、

上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散し、

上記各ビットストリームから上記複数の符号化された信号成分を再割当てし、

上記複数の符号化された信号成分から上記信号を再構築するように構成された信号処理システム。

【請求項 1 7】 各信号成分 (S E) が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を再生するように構成された信号処理システムにおいて、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、

上記各信号成分を表すビットを、分散する前の位置に戻すように、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散し、

上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てするように構成された信号処理システム。

【請求項 1 8】 プロセッサにより実行可能な命令を格納するコンピュータにより読取可能な記録媒体であって、上記プロセッサに、各信号成分 (S E) が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を復号させ、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに

分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、

上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散するステップと、

上記各ビットストリームから上記複数の符号化された信号成分を再割当てするステップと、

上記複数の符号化された信号成分から上記信号を再構築するステップとを実行させる命令が格納されていることを特徴とするコンピュータにより読取可能な記録媒体。

【請求項 19】 プロセッサにより実行可能な命令を格納するコンピュータにより読取可能な記録媒体であって、上記プロセッサに、各信号成分 (SE) が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を再生させ、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、

上記各信号成分を表すビットを、分散する前の位置に戻すように、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散するステップと、

上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てするステップとを実行させる命令が格納されていることを特徴とするコンピュータにより読取可能な記録媒体。

【請求項 20】 複数の信号成分 (SE) からなる信号を原始コーディングして、強力エラー回復を実現する原始コーディング装置において、

上記各信号成分が、可変長データを有する複数のビットによって表されるように該各信号成分を符号化する符号化手段と、

上記符号化された信号成分を複数の信号成分のセットに分割する分割手段と、

上記各符号化された信号成分のセットについて、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成するビットストリーム形成手段と、

上記信号成分のセットの信号成分を記述するパラメータが複数のビットストリームに亘って分散するように、上記各ビットストリームの複数のビットを該複数のビットストリームに亘って分散する分散手段とを備える原始コーディング装置。

【請求項 21】 各信号成分 (SE) が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を復号する復号装置において、上記信号は、各符号化された信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、

上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散する再分散手段と、

上記各ビットストリームから上記複数の符号化された信号成分を再割当てする再割当て手段と、

上記複数の符号化された信号成分から上記信号を再構築する再構築手段とを備える復号装置。

【請求項 22】 各信号成分 (SE) が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を再生する再生装置において、上記信号は、各符号化された信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、

上記各信号成分を表すビットを、分散する前の位置に戻すように、上記原始コーディン

グされた信号の複数のビットストリームから各ビットストリームのビットを再分散する再分散手段と、

上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てする再割当て手段とを備える再生装置。

【発明の詳細な説明】

【0001】

発明の背景

1. 関連出願

本出願は、1997年10月23日出願の米国特許出願番号08/956,632号「伝送損失におけるエラーを強力に回復する画像 - ブロックマッピング (Image-to-Block Mapping to Provide for Robust Error Recovery During Transmission Losses)」、1997年10月23日出願の米国特許出願番号08/957,555号「伝送損失におけるエラーを強力に回復する原始コーディング (Source Coding to Provide for Robust Error Recovery During Transmission Losses)」、1997年10月23日出願の米国特許出願番号08/956,870号「伝送損失におけるエラーを強力に回復するマルチプルブロック回復方法 (Multiple Block Recovery Method to Provide for Robust Error Recovery During Transmission Losses)」の一部継続出願として1998年1月2日に出願された米国特許出願番号09/002,547号「伝送損失におけるエラーを強力に回復する画像 - ブロックマッピング (Image-to-Block Mapping to Provide for Robust Error Recovery During Transmission Losses)」、1998年1月2日に出願された米国特許出願番号09/002,470号「伝送損失におけるエラーを強力に回復する原始コーディング (Source Coding to Provide for Robust Error Recovery During Transmission Losses)」、1998年1月2日に出願された米国特許出願番号09/002,553号「伝送損失におけるエラーを強力に回復するマルチプルブロック回復方法 (Multiple Block Recovery Method to Provide for Robust Error Recovery During Transmission Losses)」の一部継続出願として1998年1月30日に出願された米国特許出願番号09/016,083号「伝送損失におけるエラーを強力に回復する原始コーディング (Source Coding to Provide for Robust Error Recovery During Transmission Losses)」の継続出願である。1998年1月30日に出願された米国特許出願番号09/016,083号、1998年1月2日に出願された米国特許出願番号09/002,547号、米国特許出願番号09/002,470号、米国特許出願番号09/002,553号、1997年10月23日出願の米国特許出願番号08/956,632号、米国特許出願番号08/957,555号、米国特許出願番号08/956,870号は、参照により本願に組み込まれるものとする。

【0002】

2. 発明の分野

本発明は、信号伝送中のデータ欠落に起因するエラーを強力に回復する技術に関する。

【0003】

3. 背景技術

信号伝送中に発生するランダムエラーにより失われたデータを再構築するための様々な手法が提案されている。しかしながら、これらの手法は、連続したデータパケットの欠落を回復することはできない。連続したデータパケットの欠落は、当分野では、バーストエラーと呼ばれる。バーストエラーにより、再生信号は、大きく劣化する。さらに、通信の高速化を目的とする圧縮技術は、バーストエラーにより起こる信号の劣化を増長させ、これにより再生信号の劣化はさらに深刻なものとなる。バーストエラーによる伝送信号及び/又は記録信号への影響の例は、圧縮技術が重要な役割を果たす高精細度テレビジョン (high definition television: 以下、HDTVという。) や、移動通信装置等において確認される。

【0004】

HDTVの出現により、テレビジョンシステムにおいて、ナショナルテレビジョンシス

テムコミッティ (National Television Systems Committee: N T S C) が提案する従来の標準規格より高い解像度が実現されるようになった。現在提案されている H D T V 信号は、主にデジタル信号である。カラーテレビジョン信号がデジタル形式に変換される場合、通常、輝度信号及び色差信号は、8ビットにデジタル化される。カラーテレビジョン信号をデジタル伝送するためには、216メガビット毎秒の基準データ伝送速度が必要である。この伝送速度は、H D T V の場合はさらに高く、通常、1200メガビット毎秒の伝送速度が必要となる。このような高い伝送速度は、現在の無線規格によりサポートされている帯域幅をはるかに上回る。したがって、効率的な圧縮技術が求められている。

【0005】

圧縮技術は、移動通信の分野においても重要な役割を果たす。通常、移動通信の用途における遠隔の端末間でパケットデータが送受信される。移動通信の伝送チャンネル数は限られており、パケット送信の前にパケットを効率的に圧縮する技術が必要である。高い伝送速度を達成するために、多くの圧縮技術が提案されている。

【0006】

画像圧縮のための適応ダイナミックレンジコーディング (Adaptive Dynamic Range Coding: 以下、A D R C という。) 及び、離散コサイン変換 (discrete cosine transform: 以下、D C T という。) 技術が知られている。これらの技術は、画像内の局所的な相関を利用して高圧縮比を達成するものである。しかしながら、符号化された信号を復号する際にエラーが顕著となるため、効率的な圧縮アルゴリズムは、複合的なエラーの増殖の原因ともなる。このようなエラーの増大により、再生映像は大幅に劣化する。

【0007】

発明の開示

本発明は、信号を原始コーディングする方法を提供する。特に、本発明は、複数の信号成分からなる信号を処理する。各信号成分は、ビットストリームとして符号化される。与えられたビットストリームは、異なる複数のビットストリームに分散される。これにより、セグメント成分のパラメータ記述成分が異なる複数のビットストリームに分散される。分散処理により、エラーは複数レベルに分散される。したがって、分散処理の逆の処理がデコーダによりなされると、伝送時のバーストエラーは局所的な欠落として分散される。

【0008】

本発明は、複数レベルのシャッフリング処理を提供する。信号は、複数のレベルを有するものと定義される。各レベルは、複数のフレーム、複数の画素、複数のビットから構成される。一具体例において、各レベル内及びレベル間でシャッフリングが行われ、これにより欠落が発生した画像領域における画像の再構築が容易となる。

【0009】

すなわち、本発明に係る原始コーディング方法は、複数の信号成分 (S E) からなる信号に対する強力エラー回復を実現する原始コーディング方法において、

上記各信号成分が、可変長データを有する複数のビットによって表されるように該各信号成分を符号化するステップと、上記符号化された信号成分を複数の信号成分のセットに分割するステップと、上記各信号成分のセットについて、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成するステップと、上記信号成分のセットの信号成分を記述するパラメータが複数のビットストリームに亘って分散するように、上記各ビットストリームの複数のビットを該複数のビットストリームに亘って分散するステップとを有する。

【0010】

そして、この原始コーディング方法は、上記符号化された信号成分のビットを上記複数のビットストリームに亘って分散するステップの前に、バッファ内に符号化された信号成分を分散するステップを有する。

【0011】

ここで、上記信号は、ビデオ信号である。

【0012】

また、上記複数のビットストリームから各ビットストリームのビットを再分散するステップと、上記各ビットストリームの上記複数の符号化された信号成分を再割当てするステップと、上記複数の符号化された信号成分から上記信号を再構築するステップとをさらに有する。

【0013】

さらに、この原始コーディング方法は、上記各符号化された信号成分を表すビットを、分散するステップの前の位置に戻すように、上記複数のビットストリームから各ビットストリームのビットを再分散するステップと、上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てするステップとをさらに有する。

【0014】

次に、本発明に係るデジタル信号処理システムは、複数の信号成分（SE）からなる信号に対する強力エラー回復を実現する原始コーディングを実行するプロセッサを備えるデジタル信号処理システムにおいて、上記プロセッサは、上記各信号成分が、可変長データを有する複数のビットによって表されるように該各信号成分を符号化するステップと、上記符号化された信号成分を複数の信号成分のセットに分割するステップと、上記各信号成分のセットについて、該信号成分のセットの各符号化された信号成分を表す上記複数のビットからなるビットストリームを形成するステップと、上記信号成分のセットの信号成分を記述するパラメータが複数のビットストリームに亘って分散するように、上記各ビットストリームの複数のビットを該複数のビットストリームに亘って分散するステップとを実行するものである。

【0015】

次に、本発明に係る記録媒体は、プロセッサにより実行可能な命令を格納するコンピュータにより読取可能な記録媒体であって、上記プロセッサに、上記各信号成分が、可変長データを有する複数のビットによって表されるように該各信号成分を符号化するステップと、上記符号化された信号成分を複数の信号成分のセットに分割するステップと、上記各信号成分のセットについて、該信号成分のセットの各符号化された信号成分を表す上記複数のビットからなるビットストリームを形成するステップと、上記信号成分のセットの信号成分を記述するパラメータが複数のビットストリームに亘って分散するように、上記各ビットストリームの複数のビットを該複数のビットストリームに亘って分散するステップとを実行させて、複数の信号成分（SE）からなる信号に対する強力エラー回復を実現する原始コーディングを実行させる命令が格納されてコンピュータにより読取可能なものである。

【0016】

次に、本発明の復号装置は、各信号成分（SE）が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を復号する復号方法において、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散するステップと、上記各ビットストリームから上記複数の符号化された信号成分を再割当てするステップと、上記複数の符号化された信号成分から上記信号を再構築するステップとを有する。

【0017】

一方、本発明に係る再生方法は、各信号成分（SE）が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を再生する再生方法において、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、上記各信号成分を表すビットを、分散する前の位置に戻すように、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再

分散するステップと、上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てするステップとを有する。

【0018】

そして、本発明に係る信号処理システムは、各信号成分（SE）が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を復号するように構成された信号処理システムにおいて、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散し、上記各ビットストリームから上記複数の符号化された信号成分を再割当てし、上記複数の符号化された信号成分から上記信号を再構築するように構成されている。

【0019】

また、本発明に係る信号処理システムは、各信号成分（SE）が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を再生するように構成された信号処理システムにおいて、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、上記各信号成分を表すビットを、分散する前の位置に戻すように、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散し、上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てするように構成される。

【0020】

次に、本発明に係る記録媒体は、プロセッサにより実行可能な命令を格納するコンピュータにより読取可能な記録媒体であって、上記プロセッサに、各信号成分（SE）が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を復号させ、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散するステップと、上記各ビットストリームから上記複数の符号化された信号成分を再割当てするステップと、上記複数の符号化された信号成分から上記信号を再構築するステップとを実行させる命令が格納されてコンピュータにより読取可能なものである。

【0021】

また、本発明に係る記録媒体は、プロセッサにより実行可能な命令を格納するコンピュータにより読取可能な記録媒体であって、上記プロセッサに、各信号成分（SE）が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を再生させ、上記信号は、各信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、上記各信号成分を表すビットを、分散する前の位置に戻すように、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散するステップと、上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てするステップとを実行させる命令が格納されてコンピュータにより読取可能なものである。

【0022】

次に、本発明に係る原始コーディング装置は、複数の信号成分（SE）からなる信号を原始コーディングして、強力エラー回復を実現する原始コーディング装置において、上記各信号成分が、可変長データを有する複数のビットによって表されるように該各信号成分

を符号化する符号化手段と、上記符号化された信号成分を複数の信号成分のセットに分割する分割手段と、上記各符号化された信号成分のセットについて、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成するビットストリーム形成手段と、上記信号成分のセットの信号成分を記述するパラメータが複数のビットストリームに亘って分散するように、上記各ビットストリームの複数のビットを該複数のビットストリームに亘って分散する分散手段とを備える。

【0023】

次に、本発明の復号装置は、各信号成分（SE）が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を復号する復号装置において、上記信号は、各符号化された信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散する再分散手段と、上記各ビットストリームから上記複数の符号化された信号成分を再割当てする再割当て手段と、上記複数の符号化された信号成分から上記信号を再構築する再構築手段とを備える。

【0024】

一方、本発明の再生方法は、各信号成分（SE）が可変長データを有する複数のビットとして符号化された複数の信号成分から生成された原始コーディングされた信号を再生する再生装置において、上記信号は、各符号化された信号成分のセットが、該信号成分のセットの各信号成分を表す上記複数のビットからなるビットストリームを形成する複数の信号成分のセットに分割され、各ビットストリームのビットは複数のビットストリームに亘って分散されており、上記各信号成分を表すビットを、分散する前の位置に戻すように、上記原始コーディングされた信号の複数のビットストリームから各ビットストリームのビットを再分散する再分散手段と、上記複数のビットストリームの符号化された信号成分を上記信号成分のセットに再割当てする再割当て手段とを備える。

【0025】

発明の詳細な説明

本発明は、強力なエラー回復を実現するコーディング及び信号ストリームの処理の手法を提供する。以下の記述においては、説明を目的として、本発明の理解を容易にするために多くの詳細事項について述べる。しかしながら、当業者にとって、これら特定の詳細事項が本発明の実施に必ずしも必要でないことは明らかである。また、本発明を具体的かつ明瞭に説明するために、ブロック図においては、周知の電氣的構造及び回路を例示する。

【0026】

本発明を適用した信号処理方法及び信号構造について、映像信号の処理を例に説明する。しかしながら、ここに説明する方法及び装置は、音声信号や他のデジタルデータビットストリーム等、複数の信号成分からなる種々の信号に適用することができる。さらに、ここに説明する処理例は、データ圧縮のための適応ダイナミックレンジコーディング（Adaptive Dynamic Range Coding：以下、ADRCという。）処理を用いている。しかしながら、他の様々なコーディング方式及びアルゴリズムを用いてもよい。ADRCの詳細については、1991年9月4日～6日にイタリアのトリノで開催されたHDTV及びその未来に関する第4回国際研究会（Fourth International Workshop on HDTV and Beyond）において、近藤、藤森、中屋らにより発表された「将来のHDTVデジタルVTRのための適応ダイナミックレンジコーディング方式（Adaptive Dynamic Range Coding Scheme for Future HDTV Digital VTR）」に記載されている。

【0027】

この論文では、異なる種類のADRCが紹介されている。これらは、以下の式により実現される。

【0028】

【数1】

ノンエッジマッチングADRC

$$DR=MAX-MIN+1$$

$$q=\left[\frac{(x-MIN+0.5)\cdot 2^Q}{DR}\right]$$

$$\tilde{x}=\left[\frac{(q+0.5)\cdot DR}{2^Q}+MIN\right]$$

【 0 0 2 9 】

【 数 2 】

エッジマッチングADRC

$$DR=MAX-MIN$$

$$q=\left[\frac{(x-MIN)\cdot (2^Q-1)}{DR}+0.5\right]$$

$$\tilde{x}=\left[\frac{q\cdot DR}{2^Q-1}+MIN+0.5\right]$$

【 0 0 3 0 】

【 数 3 】

マルチステージADRC

$$DR = MAX - MIN + 1$$

$$q = \left\lceil \frac{(x - MIN + 0.5) \cdot 2^q}{DR} \right\rceil$$

$$\tilde{x} = \left\lfloor \frac{(q + 0.5) \cdot DR}{2^q} + MIN \right\rfloor$$

($q = 2^q - 1$ の場合、MAXは、 \tilde{x} の平均値であり、

$q = 0$ の場合、MINは、 \tilde{x} の平均値である。)

$$DR' = MAX' - MIN'$$

$$q' = \left\lceil \frac{(x - MIN') \cdot (2^{q'} - 1)}{DR'} + 0.5 \right\rceil$$

$$\tilde{x}' = \left\lfloor \frac{q' \cdot DR'}{2^{q'} - 1} + MIN' + 0.5 \right\rfloor$$

MAXは、ブロックの最大レベルを表し、MINは、ブロックの最小レベルを表し、 x は、各サンプルの信号レベルを表し、 Q は、量子化ビットのビット数を表し、 q は、量子化符号（符号化されたデータ）を表し、 \tilde{x} は、各サンプルの復号レベルを表し、角括弧「 $\lceil \cdot \rceil$ 」は、角括弧内の値の切り捨て処理を表す。

【 0 0 3 1 】

信号の符号化、伝送、続いて行われる復号処理を包括的にFIG. 1に示す。信号100は、エンコーダ110に入力されるデータストリームである。エンコーダ110は、ADRC圧縮アルゴリズムに基づいて、伝送媒体135を介して伝送されるパケット1...Nを生成する。デコーダ120は、これらパケット1...Nを受信し、信号130を生成する。信号130は、信号100を再構築する信号である。

【 0 0 3 2 】

エンコーダ110とデコーダ120は、様々な形式で実現されて、以下に説明する機能を実行することができる。ある具体例においては、エンコーダ110及び/又はデコーダ120は、通常、中央演算処理装置、メモリ、1又は複数の入出力装置、コプロセッサ等を備える汎用又は専用のコンピュータシステムにより実行される、媒体に格納されたソフトウェアプログラムとして実現される。これに代えて、エンコーダ110及び/又はデコーダ120を以下に説明する機能を実行する論理回路として実現してもよい。さらに、エンコーダ110及び/又はデコーダ120は、ハードウェア、ソフトウェア又はファームウェアの組み合わせとして実現してもよい。

【 0 0 3 3 】

この具体例においては、信号100は、それぞれがビデオシステムにおいてインターレースされた画像を示す情報を含む一連のビデオフレームからなるカラービデオ画像を示す信号である。各フレームは、2つのフィールドからなり、一方のフィールドは画像の偶数ラインに相当し、他方のフィールドは、画像の奇数ラインに相当する。データは画素値を含み、画素値は、画像内の対応する位置の色成分を表す。例えば、この具体例において、色成分は、輝度信号Yと、色差信号U、Vからなる。なお、本発明による処理がインターレース処理されたビデオ信号以外の信号にも適用できることは明らかである。さらに、本発明はYUV色空間に限定されるものではなく、この他の色空間を示す画像にも適用できることは明らかである。

【 0 0 3 4 】

FIG. 1において、エンコーダ110は、Y信号、U信号、V信号を分離し、それぞれの信号グループをADRCアルゴリズムに基づいて別々に処理する。以下では、説明を簡潔に行うために、Y信号の処理について述べるが、符号化処理は、U信号及びV信号に対しても同様に行われる。

【0035】

この具体例において、エンコーダ110は、信号100の、この明細書ではフレーム対と呼ばれる2つの連続するフレームに亘るY信号を3つの3次元(3D)ブロックにグループ化する。さらに、一具体例では、3Dブロックは、所定のフレーム対に亘る同一の局所領域における2つの2次元ブロックをグループ化して生成される。ここで、2次元(2D)ブロックは、フレーム又はフィールド内の局所画素をグループ化して生成される。ここに記述する処理は、他のブロック構造に対しても適用できる。信号のグループ化については、画像-ブロックマッピングの説明において詳述する。

【0036】

この具体例においては、任意の3Dブロックに対し、エンコーダ110は、3Dブロックを形成する2つの2Dブロック間について、画素値に変化があるか否かを算出する。値に実質的な変化が認められた場合は、モーションフラグ(Motion Flag)が設定される。当分野で周知のとおり、モーションフラグを用いることにより、エンコーダ110は、各フレーム対において画像の特定領域の画像が同じ場合、量子化符号の数を減らすことができる。エンコーダ110は、3Dブロック内における最大画素強度値(MAX)及び最小画素強度値(MIN)を検出する。エンコーダ110は、これら値MAX、MINを用いて、与えられたデータの3Dブロックのダイナミックレンジ値(DR)を算出する。変形例では、ノン・エッジ・マッチングADRCの場合、 $DR=MAX-MIN+1$ である。また、エッジ・マッチングADRCの場合、 $DR=MAX-MIN$ である。他の具体例においては、エンコーダ110は、ビデオフレームのシーケンスを示すフレームのストリーム信号をフレーム毎に符号化する。さらに、他の変形例においては、エンコーダ110は、ビデオフィールドのシーケンスを示すフィールドのストリーム信号をフィールド毎に符号化する。この場合、モーションフラグは使用されず、2Dブロックを用いて値MIN、MAX、DRを算出する。

【0037】

この具体例においては、エンコーダ110は、算出されたDRと閾値テーブル(図示せず)を参照し、DRに対応するブロック内の画素を符号化するために用いる量子化ビット(Qビット)のビット数を決定する。画素を符号化することにより、量子化符号(Qコード)が得られる。Qコードは、記録又は伝送を目的として使用される圧縮画像データに相当する。したがって、任意の3Dブロック内の全ての画素は、同じQビットにより符号化され、これにより3D符号化ブロックが形成される。3D符号化ブロックにおけるQコード、MIN、モーションフラグ、及びDRをまとめて、3D-ADRCブロックと呼ぶ。これに代えて、2Dブロックを符号化する場合、与えられた2DブロックにおけるQコード、MIN及びDRにより2D-ADRCブロックが構成される。

【0038】

様々な閾値テーブルを実現することができる。一具体例においては、閾値テーブルは、一列のDR閾値からなる。Qビットは、閾値テーブルの列において隣り合う2つのDR値の範囲を符号化するために用いる量子化ビット数に対応する。変形例では、複数の列を有し、所望の伝送速度に応じて列を選択する。閾値テーブルの各列は、閾値インデックスにより識別される。閾値の選択については、後述する部分バッファリングの説明で詳述する。ADRC符号化及びバッファリングのさらなる詳細については、本発明の譲受人が譲受した米国特許番号第4,722,003号「高効率コーディング装置(High Efficiency Coding Apparatus)」及び同じ名称の米国特許番号第4,845,560号「高効率コーディング装置(High Efficiency Coding Apparatus)」に開示されている。

【0039】

以下では、Qコードを可変長データ(variable length data: VLデータ)とも呼ぶ。

また、DR、MIN、モーションフラグをブロック属性と呼ぶ。ブロック属性は、閾値インデックスとともに固定長データ (fixed length data: FLデータ) を構成する。さらに、上述の説明の観点からみれば、ブロック属性は、複数の成分を有する信号要素のその成分に関するパラメータを記述するものである。変形例においては、FLデータは、Qビット符号を含む。これにより、復号処理においてQビット情報をDRから導き出す必要がなくなる。したがって、DR情報が欠落又は破損された場合でもQビット符号に基づいてQビット情報を求めることができる。さらにまた、Qビット符号が欠落又は破損された場合には、DRに基づいてQビット情報を導き出すこともできる。したがって、DR及びQビットを回復する必要性を低下させることができる。

【0040】

Qビット符号を新たに設けることにより、各ADRCブロックについて伝送すべきビット数が増加するという問題がある。しかしながら、一具体例においては、グループ化されたADRCブロックの複数のQコードは、例えば加算関数や連結関数に基づいて連結される。たとえば、3つのADRCブロックがグループ化され、各ADRCブロックのQビット値がそれぞれ3、4、4である場合、FLデータに符号化される合計値は11である。このように合計値を表すために必要なビット数は、それぞれの値を個別に表すために必要なビット数に比べて少なく、後述するように、破損されていないグループのQビット値を用いて、Qビット回復処理を行うことなく、Qビット値を求めることができる。

【0041】

さらに他の変形例も可能である。例えば、モーションフラグデータを符号化してもよい。この場合、Qビット及びモーションフラグデータにタグが付され、このタグを用いて符号テーブルを参照する。コーディングの手法及び機能はアプリケーションにより異なる。

【0042】

フレーム、ブロック属性、VLデータは、ビデオ信号内の様々な成分を記述している。これら成分の境界、位置、量は、ビデオ信号の伝送及び圧縮属性に応じて異なる。本実施の形態においては、これら成分は、ビデオ信号のビットストリーム内で変更及びシャッフリングされ、これにより伝送欠落に対する強力なエラー回復が実現される。

【0043】

以下の説明では、ビデオ信号のADRC符号化及びシャッフリングに基づいて、1/6連続パケット伝送欠落の許容について例示的に記述する。すなわち、以下に示す成分の定義及び分割の手法は、一具体例にすぎない。したがって、他の形態も可能である。データセットは、ビデオデータの区分又は他の種類のデータ信号を含む。したがって、一具体例においては、フレームセットは、1又は複数の連続するフレームを含むデータセットの1種である。セグメントは、1/6に分割されたQコードを格納する記憶容量を有し、フレームセットには、ブロック属性が含まれる。データのシャッフリングは、セグメント及び/又はバッファ内の成分を相互に交換することにより行われる。続いて、セグメントに格納されたデータを用いて、伝送のためのデータパケットを生成する。したがって、以下に説明するように、伝送中にあるセグメントが欠落した場合、そのセグメントから生成される全てのパケットがなくなる。同様に、伝送中にセグメントの一部が欠落した場合、そのセグメントから生成される対応する数のパケットが伝送中になくなる。

【0044】

以下では、ADRC符号化されたデータにおける1/6連続パケット欠落について説明するが、ここに説明する方法及び装置は、様々な符号化/復号方式に結びつけられた。

【0045】

FIG. 2は、ポイントトゥポイント (point-to-point) 接続又はネットワークを介してデータを伝送するために用いられるパケット構造200の具体例を示す。パケット構造200は、エンコーダ110により生成され、伝送媒体135を介して伝送される。一具体例においては、パケット構造200は、5バイトのヘッダ情報と、8ビットのDRと、8ビットのMINと、モーションフラグと、5ビットの閾値インデックスと、354ビットのQコードとを有する。ここに示すパケット構造200は、パケット構造の一例であり

、ネットワークの非同期伝送モード (asynchronous transfer mode : A T M) による伝送に適應するように構成されたものである。なお、本発明は、このようなパケット構造に限定されるものではなく、様々なネットワークにおける種々のパケット構造に適用できるものである。

【 0 0 4 6 】

上述のように、伝送媒体 (例えば、媒体) 1 3 5 は、エラーフリーの伝送媒体とみなすことはできず、したがってパケットは欠落又は破損するおそれがある。上述のとおり、従来よりこのような欠落や破損を検出する手法が提案されているが、通常、実質的な画像の劣化は避けられなかった。本発明は、このような欠落又は破損に対する強力なエラー回復の手法を提供する。以下の説明においては、複数の連続するパケットの欠落であるバースト欠落が最も発生する可能性の高いエラーであると仮定するが、ランダムパケット欠落も発生する可能性がある。

【 0 0 4 7 】

1 又は複数の連続するパケットのデータを確實且つ強力に回復するために、本発明に基づく装置及び方法は、複数レベルのシャッフリングを行う。特に、伝送パケットに含まれる F L データ及び V L データは、ある画像の空間的及び時間的に離間した位置にあるデータを含んでいる。データのシャッフリングにより、いかなるバーストエラーも分散され、これによりエラー回復が容易となる。後述するように、シャッフリングによりブロック属性と Q ビット値を回復することができる。

【 0 0 4 8 】

データ符号化 / 復号

F I G . 3 は、エンコーダ 1 1 0 により実行される符号化処理の一具体例を説明するフローチャートである。また、F I G . 3 は、画像の劣化を防止し、強力なエラー回復を容易にするためのシャッフリング処理の概要を示す図でもある。

【 0 0 4 9 】

F I G . 3 におけるステップ 1 において、表示成分とも呼ばれる入力フレームセットに対し、伝送量を削減するための間引きを行う。Y 信号については、水平方向に元の幅から 3 / 4 となるよう間引きし、U 信号及び Y 信号については、それぞれ元の高さ及び元の幅から 1 / 2 となるよう間引きする。これにより、各フレーム対が 3 9 6 0 Y ブロック、6 6 0 U ブロック、6 6 0 V ブロックを有する 3 : 1 : 0 のビデオフォーマットが形成される。上述のように、ここでは Y 信号の処理に関する説明を行うが、この処理は U 信号及び Y 信号についても適用される。ステップ 2 において、2 つの Y フレーム画像を 3 D ブロックにマッピングする。ステップ 3 において、3 D ブロックをシャッフリングする。ステップ 4 において、A D R C バッファリング及び符号化を行う。ステップ 5 において、符号化された Y ブロック、U ブロック、V ブロックをバッファ内でシャッフリングする。

【 0 0 5 0 】

ステップ 6 において、符号化された 3 D ブロックのグループ用の V L データと、これに対応するブロック属性をシャッフリングする。ステップ 7 において、F L データを異なるセグメントに亘ってシャッフリングする。ステップ 8 において、ポストアンプルを充足し、すなわちバッファ端の可変領域が所定のビットストリームにより充足される。ステップ 9 において、V L データを異なるセグメントに亘ってシャッフリングする。

【 0 0 5 1 】

以下では、符号化処理の前及び後の画素データの処理を例に本発明に基づくシャッフリングの手法を説明する。変形例として、ハードウェアにより独立したデータ値のシャッフリング及びデシャッフリングを行ってもよい。特に、ハードウェアは、ブロック値のアドレスを異なるアドレスにマッピングすることによりシャッフリング / デシャッフリング処理を実現する。しかしながら、シャッフリング処理は、データ処理の後に行う必要があるため、アドレスマッピングは、データに従属する値に対しては行うことができない。一方、後述するグループ内 V L データシャッフリングは、データに従属する値に対しても行うことができる。さらに、本発明を説明する一例として、以下に示すシャッフリングは、離

散的なデータセットに対して行われる。しかしながら、変形例においては、信号は、ビットから画素及びフレームまでの複数のデータレベルに基づいて定義される。シャッフリング処理は、信号内で定義される各レベルにおいて、及び信号内の異なるレベル間でも行うことができる。

【 0 0 5 2 】

F I G . 4 は、デコーダ 1 2 0 により実行される復号処理の例を説明するフローチャートである。好ましくは、変換処理及びデシャッフリング処理は、F I G . 3 に示す処理と逆の処理である。F I G . 4 は、さらに Q ビット、モーションフラグ、D R、M I N 及び画素データの異なる組み合わせにおいてエラー回復を実現する革新的な処理を示している。Q ビット回復、モーションフラグ回復、D R 回復、M I N 回復、画素回復の異なる具体例において異なる組み合わせを用いるエラー回復処理については後に詳細に説明する。

【 0 0 5 3 】

画像 - ブロックマッピング

本発明において、1つのフレームは、通常 5 2 8 0 個の 2 D ブロックを含み、各 2 D ブロックは、6 4 画素からなる。第 1 のフレームに基づく 2 D ブロックとそれに続くフレームの 2 D ブロックが結合されて 3 D ブロックが形成されるため、フレーム対は 5 2 8 0 個の 3 D ブロックから構成される。

【 0 0 5 4 】

画像 - ブロックマッピングは、フレーム又はフレームセットのデータをそれぞれ 2 D ブロック又は 3 D ブロックに分割する目的で行われる。さらに、画像 - ブロックマッピングは、相補的パターン及び / 又は連鎖的 (interlocking) パターンを用いてフレーム内の画素を分割し、これにより伝送欠落に対する強力なエラー回復が実現される。しかしながら、与えられた D R 値が過大になってしまう可能性を低減するために、各 2 D ブロックは、局所的な範囲の画素から構成する。

【 0 0 5 5 】

F I G . 5 は、例示的に示す画像の 1 6 画素セクションに対する画像 - ブロック変換処理の具体例を示す図である。画像 5 0 0 は、単一のフレームの局所的な領域を形成する 1 6 個の画素からなる。画像 5 0 0 内の各画素は、強度値により示される。この例では、画像 5 0 0 の最上列左端の画素の強度は 1 0 0 であり、最下列右端の画素の強度は 1 0 である。

【 0 0 5 6 】

この具体例においては、画像 5 0 0 内の異なる領域における画素を用いて、2 D ブロック 5 1 0、5 2 0、5 3 0、5 4 0 を生成する。2 D ブロック 5 1 0、5 2 0、5 3 0、5 4 0 は、符号化され、(後述するように) シャッフリングされ、伝送される。この伝送処理の後、2 D ブロック 5 1 0、5 2 0、5 3 0、5 4 0 は再結合され、画像 5 5 0 が形成される。画像 5 5 0 は、画像 5 0 0 を再構築したものである。

【 0 0 5 7 】

発生する可能性のある伝送欠落に関わらず、画像 5 0 0 を正確に再生するために、F I G . 5 に示す具体例では、連鎖的相補ブロック構造を用いる。特に、2 D ブロック 5 1 0、5 2 0、5 3 0、5 4 0 を生成するための画素を選択することにより、画像 5 5 0 を再構築する際に、相補的及び / 又は連鎖的パターンは、ブロックの再結合に用いられる。したがって、特定のブロック属性が伝送中に欠落しても、再構築された画像 5 5 0 においてエラーは隣接しない。例えば、F I G . 5 において、2 D ブロック 5 4 0 の D R が伝送中に欠落したとする。ここで、デコーダは隣り合うブロックの隣り合う複数の画素を利用し、2 D ブロック 5 4 0 の失われた D R を回復することができる。また、後述するように、相補的パターン及びシフティングを用いることにより、隣り合う画素が増加し、好ましくは他のブロックに基づく隣り合う画素の数を最大化し、これにより D R 及び M I N の回復の可能性を大幅に向上させることができる。

【 0 0 5 8 】

F I G . 5 a は、画像 - ブロック変換処理の一具体例において、2 D ブロックを形成す

るために用いられるシャッフリングパターンの例を示す図である。ここでは、画素を交互に抜き取ることにより1つの画像を2つのサブ画像、すなわちサブ画像560及びサブ画像570に分割する。サブ画像560内の長方形の区切りは、2Dブロックの境界線を示す。説明のため、2Dブロックに0, 2, 4, 7, 9, 11, 12, 14, 16, 19, 21, 23といった番号を付す。タイル565は、サブ画像560内の2Dブロックのための画素分布を示す。

【0059】

サブ画像570においては、2Dブロックの割り当ては水平方向に8画素分、及び垂直方向に4画素分シフトされている。これにより、再構築処理においてサブ画像560とサブ画像570とが結合されると、割り当てられた2Dブロックが包み込まれ、重なり合う。2Dブロックは、1, 3, 5, 6, 8, 10, 13, 15, 17, 18, 20, 22と番号付けされている。タイル575は、サブ画像570内の2Dブロックの画素分布を示す。タイル575は、タイル565に対する相補構造を有している。したがって、特定のブロック属性が伝送中に欠落した場合、失われた2Dブロックのブロック属性を回復することができる隣り合う画素が存在することとなる。さらに、同様なブロック属性のセットを有する重なり合う2Dブロックが存在する。したがって、画像の再構築の過程において、デコーダは、隣り合う2Dブロックから複数の隣り合う画素を取得し、これにより失われたブロック属性を回復することができる。

【0060】

FIG. 6は、他の相補的及び連鎖的2Dブロックの構造例を示す図である。これら以外の構造を用いてもよい。FIG. 5と同様に、FIG. 6に示す2Dブロック構造によっても、任意の2Dブロックに伝送欠落が生じても、周囲の2Dブロックの存在が保証される。なお、パターン610a、610b、610dでは、画素を後続する2Dブロックにマッピングする際、水平シフト及び/又は垂直シフトを行う。水平シフトとは、新たな2Dブロック境界が始まる前に、タイル構造を水平方向に所定画素数分シフトすることを意味する。垂直シフトとは、新たな2Dブロック境界が始まる前に、タイル構造を垂直方向に所定画素数分シフトすることを意味する。アプリケーションに応じて、水平シフトのみを行ってもよく、垂直シフトのみを行ってもよく、水平シフト及び垂直シフトを組み合わせて行ってもよい。

【0061】

パターン610aは、画像-ブロック変換に用いる渦巻状パターンを示す。渦巻状パターンでは、画像-ブロック変換処理において後続する2Dブロックを生成する際に水平シフトを行う。パターン610b及びパターン610dは、相補パターンを示し、画像-ブロックマッピング処理において、後続する2Dブロックを生成するために、水平シフト及び垂直シフトにより画素が選択される。さらに、パターン610b及びパターン610dは、2つの2Dブロック間の画素選択において交互にオフセットを設けた例である。パターン610cは、画像-ブロック変換処理のための2Dブロックを構成する画素を不規則にサンプリングした例を示す。すなわち、画素が2Dブロックに対して一度だけマッピングされるならば、画像-ブロックマッピングにはいかなるマッピング構造を用いてもよい。

【0062】

FIG. 5、FIG. 5a、FIG. 6は、2Dブロックを生成するための画像-ブロックマッピングを示すものである。この処理は、3Dブロックに対しても適用できることは明らかである。上述のように、3Dブロックの生成は、2Dブロックと同様の境界定義を用いるが、3Dブロックの生成では、境界分割は、3Dブロックを構成する後続するフレームに亘って拡張される。すなわち、3Dブロックは、第1のフレーム内の2Dブロックを定義するために用いる画素と、後続するフレーム内の2Dブロックの画素とを結合して生成される。一具体例においては、第1のフレームにおける2Dブロックの画素と、後続するフレームにおける2Dブロックの画素とは、同一の位置から抽出される。

【0063】

フレームセット内ブロックシャッフリング

任意の画像の画素値は、局所的な領域において緊密な関係を有する。一方、同一画像内の離れた領域にある画素は、大きく異なる値を有している可能性がある。したがって、任意の画像のある部分において空間的に隣接する2Dブロック又は3Dブロックを連続して符号化した場合、DR及びMINは近い値を有し、画像の離れた部分のDR及びMINは大きく異なる値を有するものとなる。したがって、画像において空間的に隣接する2Dブロック又は3Dブロックを符号化して得られたデータを連続的にバッファに格納した場合、バッファ空間の使用率に不均衡が生じる。フレームセット内のブロックシャッフリングは、ADC符号化に先立って行われ、画像-ブロックマッピング処理により生成された2Dブロック又は3Dブロックのシャッフリングを行う。シャッフリング処理により、後段のADC符号化において、バッファを均等に使用することが保証される。

【0064】

FIG. 7a~FIG. 7dは、3D-Yブロックのシャッフリングの具体例を示すものである。FIG. 7a~FIG. 7d内の3D-Yブロックは、Y信号のみを有するフレーム対に上述の画像-ブロックマッピング処理を施して得られるものである。3D-Yブロックは、シャッフリングされ、これにより符号化されたフレーム対を格納するバッファには、そのフレーム対の異なる部分に対応する3D-Yブロックが含まれるようになる。これによりADC符号化処理におけるDR分布の均一化が促される。各バッファのDR分布が均一化されることによりバッファを均等に利用することができる。

【0065】

FIG. 7a~FIG. 7dは、物理的に離れた3Dブロックのシャッフリングを示す図である。これにより連続するパケットに伝送欠落が生じても、欠落するブロック属性は、画像内の特定の領域に集中することなく、画像全体に分散される。

【0066】

ブロックシャッフリングは、小規模、中規模及び大規模のバースト的パケット欠落のいずれが生じても、欠落するブロック属性が広範囲に分散されるように設計されている。この実施の形態においては、小規模なバースト欠落とは、数個のパケットの欠落を意味し、中規模のバースト欠落とは、1バッファ分の欠落を意味し、大規模な欠落とは1セグメント分の欠落を意味するものとする。ブロックシャッフリングにおいては、画像内の比較的離れた部分から、3つの隣り合うブロックを選択する。したがって、次のグループ内VLデータシャッフリング(後に詳細に説明する)において、各グループは互いに異なる静的特性を有する3Dブロックから構成される。ブロック属性の欠落を分散させることにより、破損した3Dブロックの周囲には破損していない3Dブロックが残り、これら破損していない3Dブロックにより欠落したデータを回復することができるため、強力なエラー回復を実現することができる。

【0067】

FIG. 7aは、水平方向に66個の3D-Yブロックを有し、垂直方向に60個の3D-Yブロックを有するフレーム対を示す。これら3D-Yブロックは、セグメント0~5に割り当てられている。図に示すように、3D-Yブロックの割り当ては、2x3の枠を設け、各枠内の各3D-Yブロックが各セグメントに対応するように割り当てる。ここで、さらなるシャッフリングを行わず、最初の880個のパケットにバーストエラーが生じたとすると、セグメント0に対応するブロック属性が失われる。しかしながら、後述するように、FLデータシャッフリングを行うことにより、ブロック属性の欠落を分散させることができる。

【0068】

FIG. 7bは、セグメント0に割り当てられる番号0が付された3D-Yブロックの走査順序を示す。FIG. 7aにおいて「0」で示される各3D-Yブロックには、0, 1, 2, 3, ..., 659という番号が付され、これらはセグメント0に入力されるストリーム内の位置を示すものである。セグメントの割り当てを行うためのこのようなブロック番号を用いて、残りの3D-Yブロックもセグメント1~5に割り当てられる。この

ようにして、フレーム対内のデータは、複数のセグメントにシャッフリングされる。

【0069】

FIG. 7cは、1セグメントを構成する660個の3D-Yブロックを示す図である。0~65の番号が付された3D-Yブロックは、バッファ0に入力される。同様に、番号が付された3D-Yブロックに隣接する3D-Yブロックは、バッファ1に入力される。このような処理を繰り返してバッファ2~9が充足される。このような処理により、伝送中のバッファの欠落により失われる複数の3D-Yブロックは、画像内の異なる部分に分散される。

【0070】

FIG. 7dは、セグメント0に対応する3D-Yブロックのバッファにおける最終的な配列順序を示す図である。3D-Yブロック0, 1, 2は、バッファ内の最初の3つの部分を占める。この処理は、バッファの残りの部分において繰り返される。これにより、伝送中に3つの3D-Yブロックが欠落した場合、失われる3D-Yブロックは、画像内で離れた位置に対応するものとなる。

【0071】

FIG. 7a~7dは、フレームセット内の3D-Yブロックの分散の一具体例を示すものである。この他に、3D-Uブロック及び3D-Vブロックを分散することもできる。3D-Uブロックは、U信号のみを含むフレームセットに上述の画像-ブロックマッピング処理を適用して生成される。同様に、3D-Vブロックは、V信号のみを含むフレームセットに上述の画像-ブロックマッピング処理を適用して生成される。そして、3D-Uブロック及び3D-Vブロックの両方に対して、上述の3D-Yブロックの分散と同様の処理を行う。なお、上述のように、3D-Uブロックと3D-Yブロックの数は、それぞれ3D-Yブロックの数の1/6である。

【0072】

FIG. 7a~7dは、Y信号に対するフレーム内ブロックシャッフリングの具体例を示し、この具体例では、伝送中の1/6のパケット欠落が許容され、さらにバッファの使用率の均等化が実現されている。セグメント、バッファ、ADRCブロックの割り当てを変更することにより1/nバーストエラーに対応でき、またバッファ使用率を変更できることは当業者にとって明らかである。

【0073】

部分バッファリング

FIG. 3に示すステップ4において、ADRC符号化及びバッファリング処理が行われる。符号化技術により、画像-ブロックマッピング処理により生成された2Dブロック又は3Dブロックは、2D-ADRCブロック又は3D-ADRCブロックに符号化される。3D-ADRCブロックは、それぞれQコード、MIN、モーションフラグ、DRを含む。2D-ADRCブロックは、Qコード、MIN、DRを含む。すなわち、2D-ADRCブロックでは、単一のフレーム又は単一のフィールドに対する符号化が行われるため、2D-ADRCブロックは、モーションフラグを含まない。

【0074】

従来より多くのバッファリングの手法が知られている(例えば、近藤らによる米国特許番号第4,845,560号「高効率コーディング装置(High Efficiency Coding Apparatus)」及び米国特許番号第4,722,003号「高効率コーディング装置(High Efficiency Coding Apparatus)」を参照)。これらの記載内容は本出願に組み込まれるものとする。

【0075】

以下に説明する部分バッファリングは、ADRC符号化において使用する符号化ビットを決定するための革新的な手法を提供するものである。詳しくは、部分バッファリングは、閾値テーブルから閾値を選択する手法を提供するものであり、閾値テーブルは、遠隔の端末間の伝送速度を一定に保つとともに、エラーが増えることを制限するために設計されている。変形例においては、閾値テーブルはさらに、バッファが最大限活用されるよう設

計される。一具体例においては、バッファは、任意のフレームセットに対応する符号化されたデータを16分の1に分割して格納するメモリである。閾値は、上述した画像-ブロックマッピング処理により生成された2Dブロック又は3Dブロック内の画素を符号化するために用いるQビットのビット数を決定するために用いられる。

【0076】

閾値テーブルは、閾値の列を備え、この列は閾値セットとも呼ばれ、閾値テーブルの各列には閾値インデックスが付されている。一具体例においては、閾値テーブルは、閾値テーブルの上位列に位置する閾値セットが、より多くのQコードビットを生成するように配列されている。これにより、利用可能な所定のビット数を有する任意のバッファに対して、エンコーダ110は、この所定のビット数より小さいビット数を生成する閾値に到達するまで、閾値テーブルを下位方向に順次検索する。バッファ内の画素データの符号化には、適切な閾値が用いられる。

【0077】

一具体例においては、30Mbps程度の伝送速度が望まれる。この所望の伝送速度により、全ての任意のバッファにおいてVLデータの格納に31,152ビットが利用可能となる。これに応じて、各バッファについての累積的DR分布が算出され、閾値テーブルから閾値セットが選択され、これに基づいて3Dブロックあるいは2Dブロック内の画素をVLデータとして符号化する。

【0078】

FIG. 8は、バッファ0における選択された閾値及びDR分布の具体例を示すグラフである。FIG. 8における垂直方向の軸は、累積DR分布を示す。例えば、値bは、DRがL3以上の3Dブロック又は2Dブロックの数に等しい。水平方向の軸は、DR値のとり得る値を示す。ある具体例では、DR値は0~255の値をとる。閾値L4, L3, L2, L1は、バッファの符号化を決定する閾値セットを表す。

【0079】

一具体例において、バッファ0に格納されている全てのブロックは、閾値L4, L3, L2, L1を用いて符号化される。すなわち、DR値がL4よりおおきなブロックは、4ビットを用いて符号化された画素値を有することとなる。同様に、DR値がL3とL4の間にあるブロックに属する画素は、3ビットを用いて符号化される。また、DR値がL2とL3の間にあるブロックに属する画素は、2ビットを用いて符号化される。DR値がL1とL2の間にあるブロックに属する画素は1ビットを用いて符号化される。そして、DR値がL1より小さいブロックは、0ビットを用いて符号化される。L4, L3, L2, L1は、バッファ0内の全てのブロックを符号化するために用いるビットの総数が、31,152ビットを超えることなく、可能な限り31,152ビットに近い数となるように選択される。

【0080】

FIG. 8aは、部分バッファリングの一具体例を示す図である。フレーム800は、符号化されてバッファ0~59に格納される。伝送エラーがデータの回復の妨げとなる場合、欠落したデータに対するエラー回復処理が行われるまで、フレーム800の復号処理は中断される。ここで、部分バッファリングは、バッファ内のエラーの伝播を制限するため、残りのバッファについて復号処理を行うことができる。例えば、伝送エラーによりバッファ0内のブロック80のQビット及びモーションフラグの回復が妨げられているとする。部分バッファリングは、バッファ0内にこのエラーが伝播することを制限する。バッファ長は固定であるため、バッファ0の最後とバッファ1の先頭は既知であり、このためエラーの伝播バッファ0内に制限される。これにより、デコーダ120は、バッファ1内のブロックに対する処理を遅延することなく開始することができる。さらに、異なるバッファの符号化に対して異なる閾値セットを用いることにより、エンコーダ110は、任意のバッファ内に含まれるQコードビットの数を最大化/制御することができ、したがってより高い圧縮比が実現できる。さらにまた、部分バッファリング処理では、バッファ0~59は、固定長を有するため、一定の伝送速度を実現できる。

【0081】

一具体例においては、バッファの可変領域は、限られた数の閾値セットが存在するために、Qコードビットのみにより充足されるわけではない。すなわち、固定長を有するバッファの残りのビットは、ポストアンプルと呼ばれる所定のビットストリームパターンにより充足される。後述するように、ポストアンプルは、バッファの最後より前にあるVLデータの最後を示しているため、ポストアンプルにより双方向のデータ回復が実現できる。

【0082】

バッファ内YUVブロックシャッフリング

Y信号、U信号、V信号は、それぞれ固有の静的特性を有している。Qビット及びモーションフラグの回復処理（後述する）を改善するために、Y信号、U信号及びV信号は、バッファ内で多重化される。したがって、伝送欠落は、特定の信号に対して実質的な影響を与えない。

【0083】

FIG. 9は、それぞれY信号、U信号、V信号から導き出されるY-ADRCブロック、U-ADRCブロック、V-ADRCブロックに対するシャッフリング処理であるバッファ内YUVブロックシャッフリング処理を説明する図である。バッファ900は、フレームセット内ブロックシャッフリング後のADRCブロックの割当を示す。バッファ900は、66個のY-ADRCブロックと、これに続く11個のU-ADRCブロックと、これに続く11個のV-ADRCブロックを含む。バッファ910は、バッファ内YUVブロックシャッフリングを行った後のYUV-ADRCブロック配列構造を示す。図に示すように、3つのY-ADRCブロックに1つのU-ADRCブロック又は1つのV-ADRCブロックが後続する。バッファ内YUVブロックシャッフリングによりバッファ内の隣り合うブロックのビットストリームの類似の度合いが低減される。バッファ内ブロックシャッフリングは、この形態に限られることなく、初期の画像フォーマットに応じて、異なる信号、すなわち異なるYUV比又は異なる色空間を有する信号に対して変形して実行されるようにしてもよい。

【0084】

グループ内VLデータシャッフリング

グループ内VLデータシャッフリングは3つのステップを有する。この3つのステップは、Qコードの連結と、Qコードの再割当と、連結されたQコードのランダム化である。FIG. 10は、グループ内VLデータシャッフリングの手順を示す図であり、バッファ内に格納されたQコードには、3つの処理が順次施される。この変形例としては、処理手順のサブセットをグループ内VLデータシャッフリングに適用してもよい。各処理手順は、それぞれ独立して伝送中のデータに対するエラー回復に寄与する。したがって、各処理ステップを個別に説明する。エラー回復に関する詳細は、データ回復の説明の項目で述べる。

【0085】

1. Qコード連結

Qコード連結により、複数のADRCブロックが互いに復号されることが保証される。グループ復号によれば、後述するようにデータ回復工程において隣り合うブロックから追加的な情報を取得することが可能なため、エラー回復を容易に行うことができる。一具体例においては、Qコード連結は、バッファ内に格納された3つのADRCブロックからなる各グループに対して個別に適用される。変形例として、グループ内に他のバッファからのADRCを含めることもできる。ADRCブロックに亘るQコードの連結は、1つの連結ADRCタイルの生成として記述される。FIG. 11及びFIG. 11aは、連結ADRCタイルの生成を例示する図である。

【0086】

FIG. 11は、2ADRCブロックから連結ADRCタイルを生成する例を示す図である。具体的には、この連結は、2D-ADRCブロック0, 1, 2に含まれる各Qコード(q0~q63)に対して行われ、この結果64個のQコードを含む連結ADRCタイ

ルAが生成される。例えば、2D-ADRCブロック0の第1のQコード $q_{0,0}$ （量子化された値の0番目）は、2D-ADRCブロック1の第1のQコード $q_{0,1}$ に連結される。この2つの連結されたQコードは、続いて、2D-ADRCブロック2の第1のQコード $q_{0,2}$ に連結され、これにより連結ADRCタイルAのQ0が生成される。この処理は、Q63が生成されるまで繰り返される。これに代えて、連結ADRCタイルA内の Q_i の生成を以下の式に基づいて行ってもよい。

【0087】

$$Q_i = [q_{i,0}, q_{i,1}, q_{i,2}] \quad i = 0, 1, 2, \dots, 63$$

さらに、連結ADRCタイルA内の各 Q_i には、単一の Q_i を生成するために連結されたビット総数であるNビットを示す値が関連付けられる。

【0088】

FIG. 11aは、動ブロックを含むフレーム対から生成された連結ADRCタイルの具体例を示す図である。動ブロックとは、モーションフラグが設定された3D-ADRCブロックである。モーションフラグは、上述した画像-ブロックマッピング処理により生成された2つの2Dブロック構造内の所定数の画素が第1のフレームと第2のフレームとの間で変化している場合に設定される。この変形例として、第1のフレームと、これに続くフレームとの間の各画素の変化の最大値が所定の値を超えた場合にモーションフラグを設定するようにしてもよい。一方、非動ブロック（すなわち静ブロック）は、モーションフラグが設定されていない3D-ADRCブロックである。第1のフレームと、これに続く第2のフレームの2つの2Dブロック間で所定数の画素が変化していない場合、モーションフラグの設定は行われぬ。変形例として、第1のフレームと、これに続く第2のフレームとの間の各画素の変化の値が所定の値を超えない場合には、モーションフラグの設定を行わないものとしてもよい。

【0089】

動ブロックは、第1のフレームにおける符号化された2Dブロックと、後続するフレームにおける符号化された2Dブロックとに基づくQコードを含む。単一の符号化された2Dブロックに対応するQコードの集合をADRCタイルと呼ぶ。すなわち、動ブロックは2つのADRCタイルを生成する。一方、静ブロックは、動きがないため、動ブロックの半数のQコードのみしか含まず、したがって、1つのADRCタイルしか生成しない。この具体例においては、静ブロックのQコードは、第1のフレームの2Dブロックと、後続するフレームの対応する2Dブロックとの間の対応する画素の画素値の平均を求めることにより生成される。平均化された各画素値は、符号化されて、これにより単一のADRCタイルを形成するQコードの集合が生成される。このようにして、動ブロック1110は、ADRCタイル0,1を生成し、静ブロック1120は、ADRCタイル2を生成し、動ブロック1130は、ADRCタイル3,4を生成する。

【0090】

FIG. 11aに示すADRCの連結処理では、ADRCタイル0~4を連結して連結ADRCタイルBを生成している。具体的には、ADRCタイル0~4内に含まれる各Qコード（ $q_{0,0} \sim q_{63,4}$ ）が連結され、連結ADRCタイルB内の64個のQコードが生成される。これに代えて、連結ADRCタイルB内の各Qコード、すなわち Q_i を以下の数式に基づいて生成してもよい。

【0091】

$$Q_i = [q_{i,0}, q_{i,1}, q_{i,2}, q_{i,3}, q_{i,4}] \quad i = 0, 1, 2, \dots, 63$$

【0092】

2. Qコード再割当

Qコードの再割当により、伝送欠落に起因するビットエラーが空間的に離れた画素に分散されることが保証される。特に、Qコードの再割当処理において、Qコードは、再分散され、再分散されたQコードのビットは、シャッフルされる。したがって、Qコードの再割当により、破損された画素の周囲に破損されていない画素が残るため、エラーの回復が

容易となる。画素の破損はA D R Cブロック全体に亘って分散するため、D R及びM I Nの回復が容易となる。D R及びM I Nの回復については、データ回復の説明で詳述する。

【0093】

F I G . 1 2 に 1 / 6 パーストエラー欠落による画素の劣化の具体例を示す。具体的には、2 D - A D R C ブロック 1 2 1 0、1 2 2 0、1 2 3 0 は、それぞれ 3 ビットを用いて符号化された 6 4 個の画素を含む。すなわち、2 D - A D R C ブロックの各画素 P 0 ~ P 6 3 は、3 ビットにより表される。2 D - A D R C ブロック 1 2 1 0 は、6 ビット毎の第 1 ビットが欠落した場合におけるビット欠落パターンを影付きの長方形として示す図である。同様に A D R C ブロック 1 2 2 0 及び A D R C ブロック 1 2 3 0 は、それぞれ 6 ビット毎に第 2 及び第 4 ビットが欠落した場合を示す。F I G . 1 2 から、Q コードの再割当を行わなければ、1 / 6 パーストエラー欠落により 2 D - A D R C ブロック 1 2 1 0、1 2 2 0、1 2 3 0 において、それぞれ半数の画素に劣化が生じることがわかる。

【0094】

一具体例において、Q コードの再割当はバッファに格納されている各連結 A D R C タイル対して別個に行われ、これによりデシャッフリングにおいて、ビットエラーが空間的に分散された画素に位置されることが保証される。変形例として、バッファ内に格納された各 A D R C ブロックに対して Q コードの再割当を行ってもよい。

【0095】

F I G . 1 2 a は、連結 A D R C タイルに基づくシャッフリングされた Q コードビットのビットストリームを生成する Q コード再割当の具体例を示す。テーブル 1 2 2 及びテーブル 1 3 2 は、Q コードの再分配を示す。ビットストリーム 1 3 0 及び 1 4 0 は、Q コードビットのシャッフリングを示す。

【0096】

テーブル 1 2 2 は、連結 A D R C タイル A の連結された Q コードを示す。Q 0 は、最初の連結 Q コードであり、Q 6 3 は、最後の連結 Q コードである。テーブル 1 3 2 は、Q コードの再分配を示す。この具体例においては、第 1 の組である区画 0 には、Q 0、Q 6、Q 1 2、Q 1 8、Q 2 4、Q 3 0、Q 3 6、Q 4 2、Q 4 8、Q 5 4、Q 6 0 が含まれる。テーブル 1 3 2 内において後続する区画 1 には、1 1 個の連結 Q コードが含まれている。同様にして区画 2 ~ 5 にも 1 1 個の連結 Q コードが含まれる。テーブル 1 3 2 においては、区画間の境界を垂直方向の線分で示している。このように連結 Q コードを空間的に分離された 6 つの区画に割り当てることにより、1 / 6 パーストエラー欠落により生じるビット欠落パターンが連続する画素グループに分散されることが保証される。

【0097】

F I G . 1 2 b は、再割り当てされた Q コードにおける 1 / 6 パーストエラーにより生じるビット欠落のパターンを示す図である。具体的には、2 D - A D R C ブロック 1 2 1 5、1 2 2 5、1 2 3 5 は、それぞれ 3 ビットを用いて符号化されている 6 4 個の画素を有している。すなわち、各 2 D - A D R C ブロックの内の各画素 P 0 ~ P 6 3 は、3 ビットにより表される。2 D - A D R C ブロック 1 2 1 5、1 2 2 5、1 2 3 5 において、影付きの長方形で表されるビット欠落パターンは、連続する画素に亘って位置している。すなわち、与えられたセグメント欠落に対して、各 2 D - A D R C ブロック 1 2 1 5、1 2 2 5、1 2 3 5 内のそれぞれ 1 1 個の連続する画素のみが劣化している。変形例においては、区画への Q コードの割当処理において、異なる動ブロックからの Q コードを割り当てるようにしてもよく、これにより空間的及び時間的に分離された Q コードを 6 つのセグメントに割り当ててもよい。これにより、1 / 6 パーストエラー欠落に対して破損されていない空間的及び時間的画素が追加され、より強力なエラー回復処理が提供される。

【0098】

F I G . 1 2 a において、テーブル 1 3 2 内の再割当された Q コードは、生成されるビットストリームに亘ってシャッフリングされ、ビットストリーム内で隣接するビットが異なる区画からのビットであるようにされる。テーブル 1 3 2 内の全ての区画内の Q コードは、ビットストリーム 1 3 0 内で連結される。ビットストリーム 1 3 0 内で隣り合う任意

のビットの区画は、生成されるビットストリーム140において6ビット位置毎に分散される。これにより、ビットストリーム140内のビット番号0～5は、各区画における第1のQコードからの第1ビットを含む。同様に、ビットストリーム140のビット番号6～11は、各区画の第1のQコードからの第2のビットを含む。このような処理は、全てのQコードビットに対して繰り返される。これにより、1/6バーストエラー欠落による画素欠落が空間的に分離されることが保証される。

【0099】

FIG. 12cは、再割当された（すなわち再分配されシャッフリングされた）Qコードの1/6バーストエラー欠落により生成されたビット欠落パターンを示す図である。特に、2D-ADRCブロック1217, 1227, 1237は、それぞれ3ビットを用いて符号化された64個の画素を含む。すなわち、各2D-ADRCブロックの各画素P0～P63は、3ビットで表される。2D-ADRCブロック1217, 1227, 1237において、影付き長方形で示されているビット欠落パターンは、空間的に分離された画素に分散されており、したがって、画素エラー回復を容易に行うことができる。

【0100】

3. Qコードビットのランダム化

Qコードビットは、マスキングキーを用いてランダム化される。マスキングキーは、欠落及び破損されたデータの回復処理においてデコードの処理を補助するものである。具体的には、符号化処理において、KEYとして示されるキーを用いてQコードのビットストリームをマスキングする。これに対応して、デコードは、このKEYの正確な値を知っている必要があり、これによりQコードのビットストリームのマスクを外すことができる。

【0101】

一具体例において、KEYは、3つのADRCブロックに対するQコード再割当により生成されたQコードのビットストリームをマスキングするために用いられる。上述のように、ADRCブロックはFLデータ及びQコードを有している。マスキングキーの各キー要素(d_i)は、FLデータの値と、対応するADRCブロックに関連する量子化ビット数(q_i)との組み合わせにより生成される。具体例においては、モーションフラグとQビットを用いてKEYを定義する。すなわち、この具体例においては、キー要素は、以下の数式に基づいて生成される。

【0102】

$$d_i = 5 \cdot m_i + q_i \quad \text{ここで、} i = 0, 1, 2 \quad q_i = 0, 1, 2, 3, 4$$

変数 m_i は、モーションフラグに等しい。すなわち、対応するADRCブロックが動ブロックである場合、 m_i は0となり、対応するADRCブロックが静ブロックである場合、 m_i は1となる。さらに、変数 q_i は対応するADRCブロックを符号化するために用いる量子化ビットを表す。すなわち、 q_i の値は、4ビットADRC符号化方式においては0, 1, 2, 3, 4のいずれかである。一具体例において、3つのADRCブロックからなるグループに用いるKEYは、以下に示す数式に基づいて得られるキー要素(d_i)により定義される。

【0103】

$$KEY = d_0 + 10 \cdot d_1 + 100 \cdot d_2$$

これにより、モーションフラグ又はQビットデータの回復処理において、マスキングキーの生成に用いられた値に基づいて、使用された可能性のあるキーの再生が可能となる。再生されたキーの値は、受信されたQコードのビットストリームのアンマスキングに使用され、これにより復号候補が生成される。キーの値の再生及び特定の復号候補の選択については、データ回復の説明において詳述する。

【0104】

変形例として、マスキングキーは、様々な要素から生成できる。これにより、伝送媒体を介して要素に関する情報を伝送することなく、デコードに要素に関連する特定の情報を提供することができる。一具体例においては、任意のADRCブロックに対応するDR値又はMIN値を用いてマスキングキーを生成し、このマスキングキーによりそのADRC

ブロックを示すビットストリームをマスキングする。

【0105】

FIG. 10 ~ 12には、伝送中に生じる1/6パケットデータ欠落までを許容するグループ内VLデータシャッフリングの例を示しているが、当業者にとって、区画の総数、ビットの間隔を変更することにより、1/nバーストエラー欠落に対応するよう設計を変更できることは明らかである。

【0106】

セグメント間FLデータシャッフリング

セグメント間FLデータシャッフリングは、異なるセグメント間でブロック属性を再割当するものである。ブロック属性を再割当することによりデータ欠落を分散させることができる。具体的には、あるセグメントのFLデータが伝送中に欠落しても、失われたDR値、MIN値及びモーションフラグ値が同一のブロックに属さないようにする。FIG. 13及びFIG. 14は、セグメント間FLデータシャッフリングの具体例を示す図である。

【0107】

FIG. 13にセグメント0 ~ 5のコンテンツを示す。一具体例において、各セグメントは、880個のDRと、880個のMINと、880個のモーションフラグと、660個のYブロック、110個のUブロック、110個のVブロックに対応するVLデータとを有する。図に示すように、MINシャッフリング1300のセグメント0のMIN値は、セグメント2に移動され、セグメント2のMIN値は、セグメント4に移動され、セグメント4のMIN値は、セグメント0に移動される。さらにセグメント1のMIN値はセグメント3に移動され、セグメント3のMIN値はセグメント5に移動され、セグメント5のMIN値は、セグメント1に移動される。

【0108】

FIG. 13aは、モーションフラグのシャッフリングを示す。図に示すように、モーションフラグシャッフリング1305のセグメント0のモーションフラグ値はセグメント4に移動され、セグメント2のモーションフラグ値はセグメント0に移動され、セグメント4のモーションフラグ値はセグメント2に移動される。さらに、セグメント1のモーションフラグはセグメント5に移動され、セグメント3のモーションフラグはセグメント1に移動され、セグメント5のモーションフラグはセグメント3に移動される。欠落パターン1310は、伝送中にセグメント0が欠落した結果のFLデータ欠落を示す。

【0109】

特定のブロック属性に対して、FIG. 13及びFIG. 13aに示すように、セグメント間の特定のブロック属性の全ての瞬時値のシャッフリングを行う。例えば、FIG. 13においては、セグメント2の880個のMIN値は、セグメント0の880個のMIN値に一括して置き換えられる。同様に、FIG. 13aにおいては、セグメント4の880個のモーションフラグ値は、セグメント0のモーションフラグ値に一括して置き換えられる。このようにブロック属性を一括してシャッフリングすることにより、連続するパケットの伝送欠落に対して、欠落は、特定のブロックの特定のブロック属性に偏らない。一具体例においては、1つのブロックグループは3つのADRCブロックを含む。

【0110】

FIG. 14は、DR値、MIN値、モーションフラグ値に対する3つのモジュールシャッフリングを示す図である。3つモジュールシャッフリングとは、3つの異なるセグメントにおける3つのグループ(すなわちブロックグループ)に亘ってシャッフリングパターンを共有することを意味する。シャッフリングパターンは、3つの異なるセグメント内の全てのブロックグループに対して繰り返される。なお、異なるブロック属性に対しては異なるシャッフリングパターンを用いる。すなわち、3つのモジュールシャッフリングにより、3つの全てのセグメントに亘ってブロック属性が分散される。具体的には、任意のブロックグループについて、3つのモジュールシャッフリングは、伝送中のセグメントの欠落に対して、特定のブロック属性の瞬時値が1つだけしか失われなことを保証する。したがっ

て、後述するデータ回復処理において、ブロック内の失われたデータ回復するための復号候補の数を低減することができる。

【0111】

DRモジュールシャッフル1410に示すように、1セグメントには880個のDR値が格納されている。DR値には、そのDR値が由来するブロックに応じた番号が付されている。3つのモジュールシャッフリングにおいては、3つのセグメントのFLデータのコンテンツがシャッフリングされる。カウント値0～2は、モジュールシャッフリングのために識別される3つのセグメント内の各DR値を識別するために付された値である。すなわち、ブロック番号0, 3, 6, 9・・・に属するDR値は、カウント0に属する。同様に、ブロック番号1, 4, 7, 10・・・に属するDR値は、カウント1に属し、ブロック番号2, 5, 8, 11に属するDR値は、カウント2に属する。これにより、所定のカウント値に対して、そのカウント値に関連付けられたDR値は、セグメント0, 2, 4に亘ってシャッフリングされる。同様に、同一のカウント値に関連付けられたDR値は、セグメント1, 3, 5に亘ってシャッフリングされる。

【0112】

DRモジュールシャッフリング1410において、カウント値0に属するDR値はシャッフリングされない。カウント値1に属するDR値は、シャッフリングされる。具体的には、セグメントA内のカウント値1に属するDR値は、セグメントBに移動され、セグメントB内のカウント値1に属するDR値は、セグメントCに移動され、セグメントC内のカウント値1に属するDR値は、セグメントAに移動される。

【0113】

カウント値2に属するDR値もまたシャッフリングされる。具体的には、セグメントA内のカウント値2に属するDR値は、セグメントCに移動され、セグメントB内のカウント値2に属するDR値は、セグメントAに移動され、セグメントC内のカウント値2に属するDR値は、セグメントBに移動される。

【0114】

MINモジュールシャッフリング1420は、MIN値用の3つのブロック属性のモジュールシャッフリング処理の具体例を示す。1つのセグメントは、880個のMIN値を含んでいる。MINモジュールシャッフリング1420においては、DRモジュールシャッフリング1410においてカウント値1及びカウント値2に対して用いたシャッフリングパターンをカウント値0及びカウント値1にシフトして用いる。詳しくは、DRモジュールシャッフリング1410においてカウント値1に対して用いたシャッフリングパターンをカウント値0に適用する。また、DRモジュールシャッフリング1410においてカウント値2に対して用いたシャッフリングパターンをカウント値1に適用し、カウント値2に属するMIN値については、シャッフリングを行わない。

【0115】

モーションフラグモジュールシャッフリング1430は、モーションフラグ値用の3つのブロック属性のモジュールシャッフリング処理の具体例を示す。1セグメントは880個のモーションフラグを含んでいる。モーションフラグモジュールシャッフリング1430においては、DRモジュールシャッフリング1410においてカウント値1及びカウント値2に対して用いたシャッフリングパターンをそれぞれカウント値2及びカウント値0にシフトして用いる。具体的には、DRモジュールシャッフリング1410においてカウント値2に対して用いたシャッフリングパターンをカウント値0に適用する。また、DRモジュールシャッフリング1410においてカウント値1に対して用いたシャッフリングパターンをカウント値2に適用し、カウント値1に属するモーションフラグ値については、シャッフリングを行わない。

【0116】

FIG. 14aは、モジュールシャッフリング1410, 1420, 1430により得られたモジュールシャッフリングの結果を示す図である。モジュールシャッフリング結果1416は、各ブロック属性移動先がセグメント0に属することを示す。この例では、セグメント

0は、FIG. 14に示すセグメントAに対応する。この移動先は、FIG. 14に示すモジュールシャッフル1410, 1420, 1430により決定される。FIG. 14aは、セグメント0が伝送中に欠落した場合のブロック属性の欠落分布を示す。具体的には、欠落パターン1415は、モジュールシャッフル1410, 1420, 1430によりシャッフリングされた受信データをデシャッフリング処理した後に、6つのセグメントに亘るDR値、モーションフラグ値、MIN値の欠落を示す。FIG. 14aに示すように、ブロック属性の欠落は、セグメント0, 2, 4に亘って周期的に分散されており、セグメント1, 3, 5にはブロック属性欠落は生じていない。さらに、空間欠落パターン1417は、伝送中にセグメント0が欠落して失われたFLデータのデシャッフリング後の空間的分布を示す。空間欠落パターン1417は、後に受信データに対して施されるデシャッフリング処理後のDR値、モーションフラグ値、MIN値の欠落を示す。空間欠落パターン1417においては、破損されたブロックは破損されていないブロックに取り囲まれており、破損されたブロック属性は、破損されていないブロック属性に基づいて回復することができる。

【0117】

FIG. 14及びFIG. 14aは、3つのモジュールシャッフリングパターン及び伝送中にセグメントが失われた後のブロック属性の欠落分布を示す。変形例として、カウント値又はセグメント数を変更することによりブロック属性の欠落分布を変更してもよい。FIG. 14bにモジュールシャッフリング結果1421と欠落パターン1420とを示す。同様に、FIG. 14cにモジュールシャッフリング結果1426と欠落パターン1425とを示す。欠落パターン1420及び欠落パターン1425は、上述した3つのセグメントではなく、6つのセグメントに亘るブロック属性欠落の分布を示す。

【0118】

シャッフリング処理を行うための、ブロック属性を組み合わせて分散させる手法については、様々な変形例が可能である。

【0119】

セグメント間VLデータシャッフリング

セグメント間VLデータシャッフリング処理では、所定数のセグメント、例えば6つのセグメントが並べ替えられ、これにより1/6パケット伝送欠落までの欠落に対し、空間的に分離された周期的なVLデータ欠落が保証される。FIG. 15及びFIG. 16は、セグメント間VLデータシャッフリング処理の一具体例を示す。

【0120】

この具体例においては、30Mbpsに近い伝送速度が望ましい。したがって、望ましい伝送速度では、60のバッファのそれぞれにおけるVLデータとして31, 152ビットが利用可能となる。残りの領域には、バッファ内に含まれる88個のブロック用のFLデータに使用される。FIG. 15は、30Mbpsに近い伝送速度用のフレームセット内のVLデータバッファ配列構造を示す図である。上述のとおり、部分バッファリングにより各バッファの使用可能なVLデータの使用効率が最適化されており、VLデータの未使用領域は、ポストアンブルにより充足されている。

【0121】

FIG. 16は、空間的に分散された周期的なVLデータ欠落を保証するシャッフリング処理の具体例を示す図である。第1列は、FIG. 15に示す60個のバッファから1, 869, 120ビットの連結ストリームに並べ替えられたVLデータを示す図である。第2列は、6ビット毎にビットを抽出してまとめた新たなビットストリームを示す図である。これにより、デコーダが後に逆の処理を行う際に、伝送中に起こった1/6までのバースト欠落は、2つの破損されたビット間に少なくとも5つの未破損ビットがある周期的な欠落パターンに変換される。

【0122】

第3列は、ストリーム2のビットを10ビット毎に抽出してグループ化して生成された新たなビットストリーム、すなわちストリーム3を示す。この場合も、グループの境界は

、セグメントにおけるビット番号により規定される。ストリーム 2 を 10 ビット毎にグループ化することにより、1 / 60 のデータ欠落が生じた場合、2 つの破損されたビット間に少なくとも 59 個の未破損ビットが存在することが保証される。これにより、88 個の連続するパケットのデータが欠落した場合に、空間的に分離された周期的な V L データ欠落が実現される。

【 0 1 2 3 】

第 4 列は、ストリーム 3 のビットを 11 ビット毎に抽出して生成されたストリーム 4 を示す。この場合も、グループの境界は、セグメントにおけるビット番号により規定される。ストリーム 3 を 11 ビット毎にグループ化することにより、1 / 660 のデータ欠落が生じた場合、2 つの破損されたビット間に少なくとも 659 個の未破損ビットが存在することが保証され、8 個の連続するパケットの伝送欠落が生じた場合、空間的に分離された周期的な V L データ欠落を実現する。

【 0 1 2 4 】

ストリーム 4 内の 31, 152 ビットからなる各グループは、連続的にバッファ 0 ~ 59 に再格納される。すなわち、最初のビットグループは、バッファ 0 に格納され、最後のビットグループは、バッファ 59 に格納される。

【 0 1 2 5 】

FIG. 16 に示すグループ化の手法を変更することにより、1 / n までの伝送欠落を許容する空間的に分離された周期的な V L データ欠落が保証されることは、当業者にとって明らかである。

【 0 1 2 6 】

伝送

上述したシャッフリング処理により、F L データ及び V L データが相互に混合されたバッファが形成される。一具体例においては、パケット構造 200 に従って各バッファからパケットを生成し、各パケットを伝送媒体 135 を介して送信する。

【 0 1 2 7 】

データ回復

上述のとおり、ビットストリームを符号化する革新的な本手法により、データパケットの欠落により生じるエラーに対する強力なエラー回復を実現することができる。復号処理の概要については、FIG. 4 に示すとおりである。

【 0 1 2 8 】

FIG. 4 に示すように、パケットとして受信されたデータは、複数レベルのデシャッフリング処理、すなわちステップ 425, 430, 435, 440 において処理され、ここでは、パケットとして受信されたデータの異なるデータ又は部分がデシャッフリングされ、データが復元される。そして、ステップ 445 において、当分野で知られた技術（1991年9月4日～6日にイタリアのトリノで開催された HDTV 及びその未来に関する第 4 回国際研究会（Fourth International Workshop on HDTV and Beyond）において、近藤、藤森、中屋らにより発表された「将来の HDTV デジタル VTR のための適応ダイナミックレンジコーディング方式（Adaptive Dynamic Range Coding Scheme for Future HDTV Digital VTR）」参照）を用いてデータを ADR C 復号する。

【 0 1 2 9 】

ステップ 445, 450 において、フレームセット内ブロックデシャッフリング及びブロック - 画像マッピングが順次実行される。ステップ 425, 430, 435, 440, 445, 450, 455 は、データを符号化するために実行される上述の処理の逆の処理であり、ここでは、詳細には説明しない。なお、一具体例においては、ステップ 425, 430, 440 により示されるデシャッフリングは、データに対して独立である。例えば、実行されるデシャッフリング処理は、予め定められているか、あるいはアドレスマッピング又はテーブルルックアップにより特定される。デシャッフリングステップ 425, 430, 440 は、データコンテンツに対して独立しているので、例えばパケット欠落に起因するデータ欠落によってデシャッフリング処理ができなくなることはない。同様に、ス

トップ450, 455は、データに対して独立である。詳しくは、グループ内VLデータデシャッフリング処理により、ブロックグループ用の量子化コードが決定される。したがって、ステップ435では、パケットが欠落している場合、影響のあるグループを処理することはできない。

【0130】

デシャッフリング処理、復号処理、マッピング処理（ステップ425, 430, 435, 440, 445, 450, 455）の後、回復処理が実行され、これにより欠落パケット内に存在していたQビット及びモーションフラグ値が回復される。Qビットの欠落は、通常DR欠落（パケット欠落に起因する）に起因する。Qビット又はモーションフラグ値が未知の場合、画素のQコードビットをデータビットストリームから判定することができない。Qビット又はモーションフラグ値が不正確に判定された場合、バッファ内のデータにおいて後続するブロックの開始点が誤って判断されるため、このエラーは伝播性を有している。

【0131】

FIG. 17は、Qビット及びモーションフラグ値を回復するための処理の概要を示す図である。ここでは、特定の具体例により複数のデータブロックを用いてQビットとモーションフラグ値を回復する処理を説明する。しかしながら、ブロックの数は、この具体例に例示するものに制限されるわけではなく、1又は複数のブロックを用いることができる。FIG. 17に示すように、ステップ1705において、ビットストリーム内に検出されたエラーに基づいて、ステップ1710において、特定のパラメータに基づいて、検査された3つのブロック用の復号候補が生成される。ステップ1715において、各復号候補が正しい復号候補である可能性（likelihood）に基づいて各復号候補に得点を付ける。そして、最も得点の高い復号候補を選択する。選択された復号候補は、Qビット及びモーションフラグ値を修正し、これにより検査されたブロックの画素が復号される。

【0132】

FIG. 4に示す復号処理において、一旦最良の復号候補が選択されると、ステップ465において、パケット欠落に起因して失われたDR値又はMIN値が回復される。ここでは、例えば隣り合うブロックの最小自乗又は平均を用いた手法など、当業者に既知の回復処理を用いてDR及びMINを回復する。この処理の詳細については、例えば、1991年9月4日～6日にイタリアのトリノで開催されたHDTV及びその未来に関する第4回国際研究会（Fourth International Workshop on HDTV and Beyond）において、近藤、藤森、中屋らにより発表された「将来のHDTVデジタルVTRのための適応ダイナミックレンジコーディング方式（Adaptive Dynamic Range Coding Scheme for Future HDTV Digital VTR）」に記載されている。この具体例においては、革新的な画像-ブロックマッピング処理及びこれに基づくデータ構造により隣り合うブロックを増加させ、これにより追加的データを提供し、より正確なDR及びMIN回復を容易に行うことができる。詳しくは、一具体例において、DR及びMINは以下の式に基づいて回復される。

【0133】

【数4】

$$DR' = \frac{m \cdot \sum_i (y_i - MIN) \cdot q_i}{\sum_i q_i^2}$$

【0134】

ここで、DR'は、回復されたDRに対応し、qiは、ADRCブロック内のi番目の値であり、qi ∈ {0, 1, …, 2q - 1}であり、エッジマッチングADRCの場合m = 2Q - 1、非エッジマッチングADRCの場合m = 2Qであり、yiは、隣接するブロックの復号された値であり、Qは、Qビット値である。

【0135】

【数 5】

$$\text{MIN}' = \frac{\sum_i \left[y_i \cdot \frac{\text{DR}}{m} \cdot q_i \right]}{N}$$

【0136】

ここで、MIN' は、回復されたMINに対応し、Nは、総和に用いた項の数を表す（例えば、iが0～31であればNは32である）。他の具体例において、同一のブロックのDR及びMINが同時に破損されている場合、DR及びMINは以下の式に基づいて回復される。

【0137】

【数 6】

$$\text{DR}' = \frac{m \cdot \left[N \cdot \sum_i q_i \cdot y_i \cdot \sum_i q_i \cdot \sum_i y_i \right]}{N \cdot \sum_i q_i^2 \cdot \left[\sum_i q_i \right]}$$

$$\text{MIN}' = \frac{\sum_i \left[y_i \cdot \frac{\text{DR}'}{m} \cdot q_i \right]}{N}$$

【0138】

ステップ470において、Qビット及びモーションフラグ回復を実行する以前に復号されていないデータをADRC復号し、ステップ475において、パケット欠落又はランダムエラーに起因して生じたエラーを含む全ての画素データを回復する。そして、ステップ485において、3:1:0-4:2:2逆変換を行い、画像を表示のための望ましいフォーマットに変換する。

【0139】

FIG. 18は、本発明が提供する復号処理におけるQビット及びモーションフラグ回復処理の特定の具体例を示す図である。この特定の具体例において、処理に入力される情報は、隣接ブロック情報と、処理すべき3つのブロックのブロック属性及び画素データである。さらに、失われたデータの位置を示すエラーフラグも入力される。エラーフラグは、当業者に知られた様々な手法で生成することができるため、詳細には説明しない。エラーフラグは、どのビットが破損された又は欠落したパケットにより伝送されてきたかを示す。

【0140】

ステップ1805において、復号候補を生成する。復号候補は様々な手法で生成できる。例えば、処理負担がかなり大きくなってしまいが、復号候補に可能な復号の全てを含ませてもよい。これに代えて、事前に特定されたパラメータに基づいて、復号候補を生成し、評価すべき復号候補の数を減らしてもよい。

【0141】

この具体例においては、復号候補は、上述したグループ内VLデータシャッフリング処理においてビットストリームをランダム化するために用いられた可能性のあるキー値に基づいて判定される。さらに、復号候補は、復号すべき状態で残っているビットのビット長及び残りのブロック数に関する情報により制限される。例えば、後述するように、最後のブロックの処理においては、ブロックの復号長は通常既知である。

【0142】

この具体例の説明を続ける。FIG. 19は、この具体例における可能な組合わせを表す、ここで x は、未知の値（パケット欠落等に起因して失われた値）を示す。さらに具体例を挙げて説明する。 m_i は、 i 番目のブロックのモーションフラグであり、 q_i は、 i 番目のブロックの量子化ビットの数であり、 n_i は、 i 番目のブロックに対して可能な復号候補の数であり、 d_i は、グループ内VLデータシャッフリングの説明で述べた i 番目のブロックのキー要素の値である。 i 番目のブロックは、各グループに内で定義される。この具体例では、各グループ内のブロック数は3である。この3ブロックグループに対するキーは、 $d_0 + 10 \cdot d_1 + 100 \cdot d_2$ として生成される。第1のブロックにおいてモーションフラグが未知であり、量子化ビットのビット数が2であり、 m_0 が x であり、 q_0 が2であるとする。上述の式に基づいて、キー要素 d_i は、 $d_i = 5 \cdot m_i + q_i$ となり、 d_0 として可能なデジットの集合は $\{2, 7\}$ となる。したがって、可能な値(n_0)は、2となる。第2のブロックのモーションフラグ値が1であり、量子化ビット数が1であり、 d_i の値が $d_i = 5 \cdot 1 + 1 = 6$ であり、 $n_i = 1$ であるとする。また、第3のブロックのモーションフラグ値が1であり、量子化ビット数が未知であるとする。これにより、デジット d_2 は $\{6, 7, 8, 9\}$ の集合からなり、 $n_2 = 4$ となる。したがって、このグループの復号候補の可能数 M は、 $2 \times 1 \times 4 = 8$ となり、復号候補を生成するキーのバリエーションは、 $662, 667, 762, 767, 862, 867, 962, 967$ となる。この処理は、好ましくは、データ欠落の影響を受けた各グループに対して実行される。

【0143】

FIG. 17の説明に戻る。ステップ1715において、データがキーデータに基づいて復号されると、生成された復号候補は、復号されたデータが正しい可能性に基づいて得点付けされる。ステップ1720において、最高の得点を得た復号候補を選択する。

【0144】

復号候補の得点付けには、様々な手法を用いることができる。例えば、特定の復号候補に基づくブロック内の画素が画像内の他の画素にどれくらい合致しているかを分析して復号候補を得点付けすることができる。好ましくは、得点は、例えば自乗誤差や相関などのエラーを示す判定基準に基づいて決定する。例えば、相関については、隣接する画素との相関度が高いほど、仮定が正しい確率が高い。したがって、相関度が高ければ復号候補が正しいことが予測され、相関性がなければ復号候補が正しくないことが予測される。

【0145】

FIG. 18に示すように、4つの異なる判定基準を分析して最良の復号候補を選択する。しかしながら、復号候補を選択するために分析する判定基準の数は、1乃至3個であってもよく、5個以上であってもよい。

【0146】

FIG. 18に示す具体例では、4つの副得点付け判定基準を設け、これらに対する得点を合計して最終得点を算出する。具体的には、ステップ1815において、自乗誤差を求め、ステップ1820において、水平相関を判定し、ステップ1825において垂直相関を判定し、ステップ1830において、時間的動きを測定する（それぞれ、 M 個の復号候補と N 個のブロックと2データフレーム/ブロックに対応する $M \times 2N$ 行列として示される）。ここでは、水平相関及び垂直相関を用いるが、例えば対角相関等の様々な相関値を測定することができる。ステップ1835, 1840, 1845, 1850において、各判定基準の信頼度を測定し、測定値を正規化する。ステップ1855, 1860, 1865, 1870においては、それぞれ異なる判定基準の確率関数を生成する。これら確率関数は、例えば確率値を掛け合わせるにより結合され、例えばFIG. 18のステップ1875においては、尤度関数等による得点付けを行う。そして各復号候補の得点を他の全ての復号候補の得点と比較し、最も正しいと考えられる復号候補を選択する。

【0147】

各復号候補を評価し、それぞれの候補に得点を付けるために様々な手法を用いることができる。例えば、信頼度測定は、判定基準を正規化する一手法である。さらに、以下に説

明するもの以外にも、様々な信頼度測定の手法がある。同様に、各判定基準に基づく確率値を掛け合わせて全体の尤度関数を求める手法は、検査された様々な判定基準を結合する手法の単なる例示に過ぎない。

【0148】

通常、候補となる可能性が低い復号候補が得る得点は低く、一方、候補として有力な復号候補が得る得点が極めて高くなるため、符号化処理により最良の復号候補の判定が容易となる。特に、上述したグループ内データシャッフリング処理におけるQコードのランダム化処理によってこの判定を容易に行うことができる。

【0149】

FIG. 20a、20b、20c、20dは、FIG. 18に示すステップ1815、1820、1825、1830において特定の復号候補に得点を付け、及び総得点を算出するための異なる測定法の例を示す。FIG. 20aは、復号候補により復号された画素 x_i を復号された隣接する画素 y_i, j と比較して評価する自乗誤差による測定を説明する図である。添え字「 i, j 」は、 i に隣接するアドレスに対応する。なお、画像内の輪郭の境界に存在するために生じる急峻な変化（スパイク：spike）の影響を取り除くために、値が大きな項のいくつかを除外することが望ましい。好ましくは、 $(x_i - y_i, j)^2$ を除外し、発生する可能性のあるスパイクを回避するとよい。FIG. 20bは、時間的動きの判定基準を測定する手法を説明する図である。これは、ブロックが動ブロックであるか、動ブロックであると仮定される場合にのみ適用可能である。時間的動きの判定基準では、ブロック間の相違は小さいものであると仮定する。したがって、復号候補が悪いほど、ブロック間の相違が大きくなる。空間相関では、実際の画像が密接な関係を保ちながら徐々に変化するため、復号候補が正しい可能性が高いほど、相関度が高くなると仮定する。FIG. 20cに示す水平相関測定及びFIG. 20dに示す垂直相関測定は、この仮定を利用する。

【0150】

FIG. 18に示すステップ1835、1840、1845、1850における信頼度測定は、先のステップ（ステップ1815、1820、1825、1830）において判定された判定基準の正規化を行う処理である。一具体例において、例えば、自乗誤差が $[0, 1]$ 間の値をとり、エラーが等しい場合、信頼度は0となり、エラーが0である場合信頼度は1となる。正規化のためには他の測定法及び手法を用いてもよい。

【0151】

同様に、空間相関に関する信頼度は、以下の式により算出される。

【0152】

$$\text{maximum}(Y, 0) - \text{maximum}(X, 0)$$

ここで、 Y は、最良の相関値であり、 X は、現在の復号候補による相関値である。時間的動きの信頼度の測定は、以下の式に基づいて行われる。

【0153】

$$\text{conf} = (a - b) / (a + b)$$

ここで、 $a = \max(X, M_TH)$ であり、 $b = \max(Y, M_TH)$ であり、さらにここで、 M_TH は候補ブロックの動き閾値であり、 Y は最良の測定値、すなわち最小の時間的動きを表し、 X は現在の候補の時間的動きの測定値を表す。

【0154】

FIG. 18に示すステップ1855、1860、1865、1870において、それぞれ異なる判定基準に関する確率関数を生成する。確率を測定する手法は様々なものが存在する。例えば、得点を信頼度の測定に用いることができる。信頼度の測定値が所定の値、例えば0.8より大きければ、基礎得点から10を減じ、0.5～0.8の間であれば基礎得点から5を減じる。FIG. 21は、自乗誤差測定判定基準用の確率関数を生成するために用いる具体例である。この表は、経験的に求められた任意のデータを格納し、信頼度と、自乗誤差測定値と、既知復号候補が含まれている。具体的には、この表は、破損されていないデータを用い、DR値が破損又は欠落したと仮定して作成することができる。

。これにより、適切な復号及び不適切な復号に対するキー及び信頼度測定値が求められる。この表は、適切な復号と不適切な復号との確率比を反映したものである。この表を用いて、特定の自乗誤差の値（列）及び信頼度の値（行）に対する確率が判定される。例えば、信頼度0において、様々な自乗誤差が存在し、候補が正しい確率は40%～50%であることが判る。信頼度が0ではないが、低い値である場合、この確率は、急速に低下する。相関及び時間的動きに関しても、対応する経験的に求められた判定基準測定値及び信頼度測定値に基づいて、同様の表を作成することができる。

【0155】

生成された確率を用いて、この具体例における、上述の「得点」を求めることができる。候補復号の得点付けには、他の手法を用いてもよい。ステップ1875において、異なる確率を結合して尤度関数 $L_i = j \cdot P_{i,j}$ を求める。ここで、 j は確率関数 $P_{i,j}$ の乗算関数であり、 $P_{i,j}$ は候補 i ブロック j の確率関数である。候補としては、関数 L_i が最大となるものを選択する。

【0156】

FIG. 18に示す処理においては、欠落したパケットにより伝送されたブロック属性を回復する必要がある場合がある。したがって、ステップ1810において、DR値及びMIN値を必要に応じて回復する。この回復処理には、初期値、平均、自乗誤差関数を用いた手法や、例えば、1993年9月20～22日にオーストラリアのメルボルンで開催されたIEEEビジュアル信号処理及び通信学会で発表された近藤、藤森、中屋らによる「将来のHDTVデジタルVTRのための適応ダイナミックレンジコーディング (Adaptive Dynamic Range Coding Scheme for Future HDTV Digital VTR)」及び近藤、藤森、中屋、内田らによる「新たなデジタルVCRの秘匿法 (A New Concealment Method for Digital VCRs)」に開示されたより高度な手法を用いることができる。回復された値は、上述した復号候補の生成に利用される。

【0157】

あるいは、DR値及びMIN値は、Qビット判定処理において判定される。この手法をFIG. 22に示す。具体的には、上述のとおり、この具体例においては、モーションフラグ及び量子化ビット数を用いて符号化処理を行い、さらにこれらにより、回復処理において復号候補の可能な数を減少させている。上述のとおり、他の情報を用いてもよい。すなわち、DR値及び/又はMIN値を符号化にも用いてもよい。これに代えて、DRの一部のビット（例えば、DRにおいて重要度が低い2つのビット）を符号化に用いてもよい。DRデータを符号化しても、変数を追加するにつれて可能な復号候補の数は増加する。FIG. 22の処理では、 $K \cdot M$ 復号候補が生成される。ここで、 K は、未知のデータの候補値の数であり、例えば、DR1、DR2、DR3（DR1、DR2、DR3は、グループ内のブロックのDR値を示す）の合計の2ビットが符号化された場合、 $K = 4$ である。DR及びMINは、例えばDR1、DR2、DR3の合計の2ビット等の補助情報を用いて回復される。この処理により、復号候補の最大数を調べるためのオーバーヘッドを代償として、候補選択処理の精度を向上させることができる。

【0158】

なお、通常、復号された隣接するブロックの数が多いほど、Qビット及びモーションフラグの回復処理が向上する。さらに、いくつかの具体例において、この処理は、バッファ内の連続するブロックに対して適応される。FLデータの全て又は一部が利用可能であれば、復号候補の数を減らすことができ、ブロックのFLデータ全てが与えられれば、可能な復号候補は1つとなる。しかしながら、Qビット及びモーションフラグの回復処理は比較的時間を要するものであるため、回避した方が望ましい。さらに、Qビット及びモーションフラグの回復処理を行うためには、可能な限り多くの情報を用いることが望ましい。一具体例においては、Qビット/モーションフラグ情報が失われているブロックに到達するまで、バッファの先頭から順次ブロックを処理する。このような処理を順方向Qビット及びモーションフラグ回復処理と呼ぶ。他の具体例においては、バッファの最後を参照してバッファ内の最終ブロックを判定し、この最終ブロックからQビット/モーションフラ

グ情報が失われているブロックに到達するまで、バッファの最後から先頭の方向に順次データを回復する。この処理を逆方向Qビット及びモーションフラグ回復処理と呼ぶ。

【0159】

上述したように、ブロックは、VLデータを含むため、ブロック長は可変である。したがって、バッファ内で後続するブロックの位置を正確に検出するために、VLデータを形成するビットの総数を判定する必要がある。復号処理において、バッファ内の未使用のビット位置には、所定の、望ましくは認識が容易なパターンを有するポストアンブルを設ける。復号処理においては、ポストアンブルは、ブロックとバッファの最後との間に存在することとなる。容易に認識できるパターンを用いているため、復号装置は、このパターンを確認することにより、ポストアンブルの開始位置を検出でき、したがって、バッファ内のブロックの最後を検出できる。この情報は、2つの処理において使用される。ブロックが破損されたQビット/モーションフラグデータを含み、ブロックの開始位置が既知である場合（例えば、先行するブロックが無事に復号されている場合）、直前のブロックの最後からポストアンブルの先頭までの差分がブロック長に対応する。この情報は、ブロックのQビット及び/又はモーションフラグの算出に使用できる。ポストアンブルの開始位置は、直前のブロックからバッファの先頭方向におけるQビット及びモーションフラグ回復にも使用できる。

【0160】

FIG. 23は、ポストアンブルを用いた双方向Qビット及びモーションフラグ回復処理を説明する図である。FIG. 23においては、バッファ2300は、FLデータのブロックからなるN個のグループ用のFLデータ2303を含む。各グループは、複数の（例えば、3個の）ブロックから構成される。この具体例においては、1番目及び2番目のグループ2305, 2310が復号され、3番目のグループは、DR/モーションフラグデータが破損されているために、直ちには復号することができない。この時点で、破損されたデータを回復するためにQビット/モーションフラグ回復処理が必要となる。ここで、順方向へのグループの処理を継続せず、ポストアンブルパターン220を検出することにより、バッファの最後を読み出す。すなわち、ポストアンブルの先頭を判定することにより、最後のブロックグループが判定される。DR/モーションフラグデータはVLデータの長さを暗示しているため、最終ブロックのVLデータの先頭と、したがって直前のブロック最後とが判定される。これにより、破損データを含むブロック240に到達するまで、ブロック225, 230, 235等のブロックが順次処理される。続いて、破損されたブロック215, 240及び妨害された複数のブロック250が回復される。ここでは、望ましくは上述のQビット/モーションフラグ回復処理を用いる。

【0161】

なお、双方向処理の順序は、順方向から逆方向の順序に限定されるものではなく、処理は、いずれの方向から開始してもよく、両方向から開始してもよい。さらに、いくつかの具体例においては、複数の処理を平行して実行し、処理効率を高めることもできる。さらに、破損されていない妨害されたブロックについては、上述のQビット/モーションフラグ回復処理を行うことなく、Qビット/モーションフラグ情報に直接アクセスしてこれらのブロックを回復してもよい。

【0162】

上述のように、最良の復号候補を実際の復号処理として選択するための得点付けの手法は、様々である。変形例においては、各復号候補を用いた画像の平滑度を評価する。一具体例においては、ラプラス測定を実行する。ラプラス測定では、例えば画像の曲率等の画像表面の2次特性を利用する。線形画像表面、すなわち平滑な表面に対しては、このラプラス測定の結果は0に近い値となる。

【0163】

この処理をFIG. 24a、24b、24cを用いて説明する。FIG. 24aは、ラプラスカーネルの一具体例を示す。他のラプラスカーネルを用いてもよい。ラプラスカーネルLは、3×3の領域で示される。画像領域の平滑度を測定するために、画像のサブ領

域をカーネルにより畳み込み、畳み込まれた値の平均を求める。領域及びサブ領域のサイズ（したがってカーネルのサイズ）は、アプリケーションに応じて変更できる。

【 0 1 6 4 】

処理の一例を F I G . 2 4 c に示す。この具体例では、 3×3 のカーネル及びサブ領域サイズ及び 8×8 の領域サイズと、インデックス i, j により識別される独立要素とを用いる。ステップ 2 4 6 0 において、候補復号値 $x[i][j]$ を正規化する。例えば、値は以下の式に基づいて正規化される。

【 0 1 6 5 】

【数 7】

$$\tilde{x}[i][j] = \frac{x[i][j]}{\sqrt{\sum_{i,j} (x[i][j] - X_{mean})^2}}, \quad 0 \leq i, j < 8$$

$$\text{ここで、 } X_{mean} = \frac{\sum_{i,j} x[i][j]}{64}, \quad 0 \leq i, j < 8$$

【 0 1 6 6 】

ステップ 2 4 6 5 では、正規化された値を用い、以下の式に基づいて、平滑度を示すブロックラプラス値 L_x を求める。

【 0 1 6 7 】

【数 8】

$$l[i][j] = \sum_{m=-l}^l \sum_{n=-l}^l L[m][n] \cdot x'[i+m][j+n], \quad 0 \leq i, j < 8$$

$$L_x = \frac{\sum_{i,j} |l[i][j]|}{64}$$

【 0 1 6 8 】

ブロックラプラス値が 0 に近いほど、画像の部分の平滑度が高い。したがって、このブロックラプラス値に基づいて得点を算出することができ、ラプラス値が最も小さい復号候補を正しいと判定することができる。

【 0 1 6 9 】

ラプラス評価法は、符号化された候補値 $q[i][j]$ を用いて実現することもできる。基本的な処理手順は、F I G . 2 4 c に示す手順と同様である。この具体例では、 3×3 のカーネル及びサブ領域サイズ及び 8×8 の領域サイズと、インデックス i, j により識別される独立要素とを用いる。例えば、値は以下の式により正規化される。

【 0 1 7 0 】

【数 9】

$$q'[i][j] = \frac{q[i][j]}{\sqrt{\sum_{i,j} (q[i][j] - Q_{mean})^2}}, \quad 0 \leq i, j < 8$$

$$\text{ここで、} Q_{mean} = \frac{\sum_{i,j} q[i][j]}{64}$$

【 0 1 7 1 】

ステップ 2 4 6 5 では、正規化された値を用いて、以下の式に基づいて、平滑度を示すブロックラプラス値 L_q を求める。

【 0 1 7 2 】

【数 1 0】

$$r[i][j] = \sum_{m=-1}^1 \sum_{n=-1}^1 L[m][n] \cdot q'[i+m][j+n], \quad 1 \leq i, j < 7$$

$$L_q = \frac{\sum_{i,j} |r[i][j]|}{36}$$

【 0 1 7 3 】

ブロックラプラス値が 0 に近いほど、画像の部分の平滑度が高い。したがって、このブロックラプラス値に基づいて得点を算出することができ、ラプラス値が最も小さい復号を正しいと判断することができる。

【 0 1 7 4 】

さらに他の変形例も可能である。変形例として、平滑度測定のために、より高次元の画像特性を用いることもできる。この場合、高次カーネルを用いることができる。例えば、4 次カーネルを用いて、4 次ブロックラプラス測定を行うことができる。このような 4 次カーネルは、2 次ラプラス演算を縦続的に実行することにより実現できる。

【 0 1 7 5 】

さらに、画像が所定の値より大きな変化又は動きを有しているか否かに基づいて評価処理を変更してもよい。画像の部分が所定の値より大きな動きを示している場合、フレームではなくフィールドに対する測定を行うことが望ましい。この点について、FIG. 25 を用いて説明する。FIG. 25 は、平滑度測定を用いた処理を示す図である。しかしながら、この処理は様々な種類の測定法を用いて実現することができる。

【 0 1 7 6 】

画像領域のフレーム 2 5 0 5 は、フィールド 0 及びフィールド 1 から構成される。ステップ 2 5 1 0 において動きが検出されない場合、ステップ 2 5 1 5 において、各フレーム内のブロックについてブロックラプラス値を求めることにより平滑度を測定する。所定の値より大きな動きが検出された場合、ステップ 2 5 2 0 及びステップ 2 5 2 5 においてそれぞれのフィールドに対してブロックラプラス測定を行い、得られた 2 つの測定値をステップ 2 5 3 0 において結合、例えば平均値を求め、平滑度測定値を算出する。動きの検出及び測定には様々な手法を用いることができる。一具体例においては、フィールド間の変化を評価し、得られた値が所定の値を超えた場合に動きを検出する。

【 0 1 7 7 】

値の回復（通常、欠落又は破損された値の置換）のために動き検出、フレーム情報及びフィールド情報を使用する手法は、値を回復する必要があるあらゆる処理の過程に適用で

きる。例えば、値を回復するための動き検出、フレーム情報及びフィールド情報を使用する手法は、DR/MIN回復処理、画素回復処理、Qビット及びモーションフラグ回復処理等に適用することができる。このように、検出された動きレベルに基づいて、回復処理は、フィールド単位又はフレーム単位で既存の情報を利用する。さらに、この処理は、特定の方向（例えば、水平方向又は垂直方向）での相関レベルに基づいて選択される重み付けの値を用いるアプリケーションと組み合わせて用いることもできる。

【0178】

Qビット及びモーションフラグ回復処理の他の具体例では、復号候補は、ブロック内及びブロック間測定に基づいて評価される。以下の説明では、用語「ブロック」は、フレーム又はフィールドの部分の意味するものとする。ブロック内測定は、復号候補により復号された画像の部分の例えば平滑度を評価する。ブロック間測定は、復号候補が隣接する画像部分にどれほど合致するかを測定する。FIG. 26a及びFIG. 26bは、ブロック間評価とブロック内評価とを組み合わせた手法を説明する図である。具体的には、FIG. 26aは、ブロック間測定及びブロック内測定の結果が良好であるために許容可能な復号候補を示し、一方、FIG. 26bでは、ブロック内測定の結果は極めて良好であるが、ブロック間測定の結果が良好ではない場合を示している。

【0179】

ブロック内測定の具体例では、上述した平滑化測定を行う。ブロック間測定の具体例では、上述の自乗誤差測定を行う。ブロック間測定の変形例では、ADRCブロック候補における適合する境界画素の比及び境界画素の総数を使用する。

【0180】

ADRC符号化された 8×8 ブロックに対するブロック間評価及びブロック内評価について、FIG. 26c、26d、26eを用いて説明する。FIG. 26dは、 q 個の値からなる符号化された値2650のデータの画像の一部（ブロック）を示し、このブロックから候補復号値 x が生成される。隣接する復号されたデータ2655は、 y 個の値からなる。上述のとおり、FIG. 26cのフローチャートに示すステップ2605において、ブロック内測定が実行され、これにより測定値、例えばブロックラプラス値 LX が算出される。ステップ2610において、ブロック間測定値 SX が算出され、これにより隣接するブロック間の適合性が測定される。ステップ2615において、結合測定値 MX が算出される。結合測定値は、復号候補を選択するために用いる情報を提供する。

【0181】

この具体例において、 SX は、復号候補による各境界画素の有効範囲内に存在する隣接データの数として算出される（FIG. 26e参照）。FIG. 26eは、一具体例の有効範囲を示すグラフであり、ここでは観察された量子化値 q_i のそれぞれの有効範囲が示されている。すなわち、 $LQ < DR < UQ$ となり、ここで、 LQ 、 UQ は、それぞれ量子化ビット $= Q$ に対応するDRの上位及び下位の境界を示している。 SX については、 $SX = SX / \text{境界画素数}$ として正規化することが望ましい。

【0182】

この具体例では、結合測定値 MX は、 $MX = SX + (1 - LX)$ という式に基づいて算出する。これに代えて、以下の式を用いて結合測定値の重み付けを行ってもよい。 $MX = w \cdot SX + (1 - w) \cdot (1 - LX)$ ここで、 w は重みであり、通常経験的に決定された重み付けの値を有する。

【0183】

欠落又は破損されたDR値及びMIN値を判定するための他の具体例も考えられる。例えば、上述の式を変形して用い、より高精度なDR値及びMIN値の回復を実現することができる。他の具体例では、メジアン法を用いてもよい。メジアン法の一具体例では、MINの値を全ての MIN_i の値のメジアンとして以下のようにして求める。

【0184】

$MIN_i = y_i - q_i \cdot s$ ここで、 q_i は、符号化された画素値を示し、 y_i は、 q_i に隣接する復号された画素を示す。エッジマッチングADRCにおいて、 $s = DR / (2$

Q - 1) である。ノンエッジマッチング A D R C においては、 $s = D R / 2 Q$ である。ここで、Q は、画素毎の量子化ビット数 (Q ビット値) を示す。

【 0 1 8 5 】

使用される値は、時間的近似値又は空間的近似値であってもよい。y_i の値としては、隣接するフレーム / フィールド又は同じフィールドの隣接する画素の復号値を用いてもよい。y_i の値としては、隣接するフレーム / フィールド又は同じフィールド内の同じ位置の画素の復号値を用いてもよい。

【 0 1 8 6 】

さらに、D R 及び / 又は M I N 回復処理をクリップ処理と組み合わせて用い、回復精度を向上させるとともに、回復処理中のデータのオーバーフローを防止することもできる。クリッピング処理は、回復されたデータの値を所定の範囲内に制限する処理である。すなわち、所定の範囲外の値は、この範囲の最も近い境界値にクリップされる。一具体例においては、クリップ処理により値を [L Q , U Q] の範囲内に制限する。ここで、L Q , U Q は、量子化ビット数 = Q で表される画素値の範囲の下限及び上限を示す。量子化ビット及びさらに M I N + D R を制限する制限値 Num を設ける。Num は最大画素値であり、この具体例においては Num は 2 5 5 である。この具体例において、適応可能であれば、 $U Q + 1 = L Q + 1$ とする。

【 0 1 8 7 】

判定基準を単一の式に結合することにより、D R のための拘束されない回復値 (val') と、最終的にクリップされた回復値 (val) とが以下の式から得られる。

【 0 1 8 8 】

$$val = \max(\min(val', \min(UQ, 255 - MIN)), LQ)$$

ここで、min 及び max は、それぞれ最小関数及び最大関数を示す。

【 0 1 8 9 】

他の具体例においては、境界画素 y_i を用いて回復された D R 及び / 又は M I N をフィルタリングして相関値が最も高いものだけを用いるようにする。これにより D R 及び M I N の回復の精度が向上する。これら境界画素は、使用されていない判定基準には適合しない。一具体例においては、境界画素 y_i は、 $L Q - D R < U Q$ を満たす D R 値が存在し、元の画素 y_i が q_i として符号化されている場合、D R 算出に有効であるとみなされる。すなわち、以下の式が満たされるとき、画素は有効である。

【 0 1 9 0 】

【 数 1 1 】

$$\frac{(y_i - MIN)m}{\max(q_i - 0.5, 0)} \geq L_Q$$

$$\frac{(y_i - MIN)m}{\min(q_i - 0.5, m)} < U_Q$$

【 0 1 9 1 】

ここで、m は、最大量子化レベル = 2 Q - 1 を表す。次に、D R 回復値 (val') は、以下の式に基づいて算出される。

【 0 1 9 2 】

【 数 1 2 】

$$val = \frac{m \cdot \sum_i (y_i - MIN) q_i}{\sum_i q_i^2}$$

【0193】

続いて、この値は有効範囲内にクリップされる。この処理により、DR回復値は、閾値テーブルにより定義されている有効範囲内に納められる。これにより、DRが閾値テーブル境界近傍に存在する点の精度が低下する。

【0194】

ここで、量子化雑音の影響により静ADRCブロックのDRは、フレーム間で若干変化することがある。このような変化がADRC符号化境界を横断し、DRが複数の連続フレームにおいて回復された場合、有効画素選択とともに回復されたDRの回復値は、各横断点で過大になる傾向があり、この結果、表示画面上で顕著なちらつきが生じる。このような問題を抑制する手法として、一具体例では、有効画素選択処理を变形し、上限及び下限の境界による制約を緩和する。これにより隣接する有効領域に侵入する境界画素を許容する。境界に外接する点を含むことにより、回復値が上限又は下限に近い値となる傾向を強めることができる。緩和された境界L'QとU'Qは、緩和定数rにより算出される。一具体例においては、rの値を0.5に設定する。他の値を用いてもよい。

【0195】

$$L'Q = rLQ - 1 + (1 - r)LQ$$

$$U'Q = (1 - r)UQ + rUQ + 1$$

以上、DR及びMINが破損又は欠落した場合にDR及びMINを回復するための様々な手法を説明した。算出され、回復された対応するデータの時間的及び/又は空間的相関を調べ、重み付けを行うことにより、さらに回復の精度を高めることができる。具体的には、特定の方向又は時間軸において高度な相関が存在する、例えば水平相関が存在する場合、画像特性はその高度な相関を有する方向において平滑に連続し、したがって、相関値の高い回復値の使用に最良の評価を与える。これにより、境界データは対応する方向（例えば、垂直、水平、フィールド間）に含まれ、相関測定値に基づいて重み付けされ、最終的な回復値が得られる。

【0196】

この処理の一具体例をFIG. 7aを参照に説明する。ステップ2710において、DR又はMINの回復値を一方向について生成し、ステップ2715において、他の方向の回復値を生成する。例えば、この処理を空間的に適用する場合、水平方向の境界に沿う境界画素を用いて第1の回復値hestを生成し、垂直方向の境界に沿う境界画素を用いて第2の回復値vestを生成する。これに代えて、処理を時間的に適用する場合、隣接するフィールド間の境界画素を用いて第1の回復値を生成し、隣接するフレーム間の境界画素を用いて第2の回復値を生成する。

【0197】

ステップ2720において、各方向における相関レベルを示す相関演算に基づいて回復値の重み付けを行う。続いてステップ2725において、重み付けされた第1及び第2の回復値を結合して結合回復値を生成する。なお、この処理は、2方向の回復値に対する重み付けのみに限定されるわけではない。すなわち、重み付けされ、結合される回復値の数は、アプリケーションに応じて変更できることは明らかである。特定の方向における相関レベルを示す相関値を算出するために様々な手法を用いることができる。さらに、様々な判定基準を用いて相関レベルに関する重み付け要素を選択することができる。通常、一方の相関値が他方の相関値よりかなり大きい場合、結合された回復値は、主に対応する回復値に基づくものである必要がある。一具体例において、結合された回復値は以下の式により求められる。

【 0 1 9 8 】

【 数 1 3 】

$$val' = \begin{cases} \alpha hest + (1-\alpha) vest & : hc \geq vc \\ (1-\alpha) hest + \alpha vest & : hc < vc \end{cases}$$

【 0 1 9 9 】

ここで、hcは水平相関を表し、vcは垂直相関を表し、hestは左右の境界情報のみに基づくDR回復値を表し、vestは上下の境界情報のみに基づく回復値を表し、 α は重みを表す。重みの値は様々な方法で決定できる。FIG. 27bは、水平相関と垂直相関との差分の関数として重みの値を決定する一具体例を説明する図である。具体的には α は、以下の式を満たすように選択される。

【 0 2 0 0 】

【 数 1 4 】

$$\alpha(|hc-vc|) = \begin{cases} 0.5 + 0.25 \cdot e^{-8(0.35-|hc-vc|)} & : |hc-vc| < 0.35 \\ 1 - 0.25 \cdot e^{-8(|hc-vc|-0.35)} & : |hc-vc| \geq 0.35 \end{cases}$$

【 0 2 0 1 】

上述のように、適応相関処理は、DR及びMINのいずれの回復にも用いることができる。しかしながら、望ましくは、MIN+DRが255になるようにMIN回復処理においてクリッピングを行う。したがって、関数 $val = \max(\min(val', 255 - MIN), 0)$ を用いることができる。さらに、上述のとおり、時間的相関を評価し、これにより回復値の重み付けを行うこともできる。また、時間的相関と空間的相関を組み合わせる利用してもよい。例えば、フィールド間のある回復値を時間的回復値として生成する。他方の回復値を1フィールド内の空間的回復値として生成する。最終的な回復値は、時間的相関及び空間的相関を組み合わせる算出される。相関の組み合わせを動きの量に置換してもよい。他の変形例も可能である。この手法をオーディオデータに対して適用してもよい。

【 0 2 0 2 】

一変形例として、最小自乗法に単純な修正を加えた手法を用いることができる。この具体例により、回復されたDR値に起因するちらつきを低減することができる。以下の説明においては、QVは、画像部分又は点 q_i の集合を有し、DRが回復されようとしているADRCブロックを表し、Yは、QV内の垂直方向又は水平方向に隣接する点から得られた復号値のリストを表し、 y_i は、 q_i の垂直方向又は水平方向の隣接点を表す。各点 q_i は、最大4個の復号された隣接点を有するので、1画素又は点について (q_i, y_i) の組み合わせは最大4つとなる。DR(DRuls)の無制約最小自乗評価値は、以下のとおりである。

【 0 2 0 3 】

【 数 1 5 】

$$(DR)_{uls} = \frac{2^Q \cdot \sum_i (y_i - MIN) \cdot (0.5 + q_i)}{\sum_i (0.5 + q_i)^2}$$

【 0 2 0 4 】

ここで、Qは量子化ビットの数であり、MINはブロック属性として伝送された最小値を表す。上述の式は、ノンエッジマッチングADRCに対応するものであり、エッジマッチ

ング A D R C の場合は、 $2Q$ を $2Q - 1$ に置き換え、 $(0.5 + q_i)$ を q_i に置き換える。

【0205】

好ましくは、無制約自乗評価値をクリップし、閾値テーブル及び等式 $MIN + DR = 255$ に整合させる。この条件は符号化処理中に行わなければならない（通常、ノンエッジマッチング A D R C の場合、D R 値として許容される範囲は $1 \sim 256$ である）。すなわち、最小自乗評価値は、以下の式に基づいてクリップされる。

【0206】

$$(DR)_{isc} = \max(\min(UB, DR_{uls}), LB)$$

ここで、U B は上限を表し、L B は加減を表し、min及びmaxは、それぞれ最小関数及び最大関数を表す。

【0207】

変形例として、D R 評価値に最も適合する画素を選択することにより評価を高めて D R の評価値を算出することもできる。例えば、画像内の変化の少ない領域の画素に対する D R 評価値は、変化が多い領域の画素に対する D R 評価値より適合性が高くなる。具体的には、画像の輪郭が鋭ければ、評価の精度が低くなる。以下、D R 評価値を算出するために用いる画素を選択する簡易な計算手法の具体例を説明する。

【0208】

この具体例では、最小自乗評価値 (DR_{lse})。例えば (DR_{uls}) 又は (DR_{isc}) を算出する。この評価値を用いて、符号化値 Q V のリストを復号候補値 X に変換する。ここで、 x_i は q_i に由来する X の構成要素を表す。 x_i 値は D R の第 1 の評価値を用いて生成された、回復された復号値である。 x_i 値は、以下の式により定義される。

【0209】

【数16】

$$\text{エッジマッチング ADRC: } x_i = \text{MIN} + \left\{ 0.5 + \frac{q_i \cdot DR_{lse}}{2^Q - 1} \right\}$$

$$\text{ノンエッジマッチング ADRC: } x_i = \text{MIN} + \left\{ \frac{(q_i + 0.5) \cdot DR_{lse}}{2^Q} \right\}$$

【0210】

DR_{lse} が真の D R の正当な評価値であると仮定すると、いかなる場合も x_i は y_i に比較的近似し、したがって変化の少ない領域と判断され、望ましい適合性が得られる。新たな X 及び Y リストは、 x_i と y_i が近似しており、最小自乗評価値が再計算されて評価値が更新される。

【0211】

「近似」しているか否かを決定するために、様々な判定基準を設けることができる。一具体例においては、エラー関数の A D R C 符号化を用いる。この手法は、計算が低コストであるという利点を有する。処理においては、点 $e_i = |y_i - x_i|$ から構成されるリスト E を定義する。 e_{min} 及び e_{max} をそれぞれリストの最大値及び最小値として定義すると、 $eDR = e_{max} - e_{min}$ となる。したがって、符号化されたエラー値は、以下のように定義される。

【0212】

$$g_i = (e_i - e_{min})n_l / eDR$$

ここで、 n_l は上述した A D R C 処理と同様の手法により e_i を再量子化する際の量子化レベルの数を表す。

【0213】

g_i がいずれかの閾値以下であるという条件に合致するもののみを選択することにより、新たなリスト X、Y が生成される。これら新たなリストが十分長ければ、これらのリストを用いて精密な最小自乗評価値 D R r l s を生成できる。最小自乗評価値を精密化する

ため必要となる g_i の閾値及び合致数は、好ましくは経験的に求められる。例えば、 n_1 を 10 とした $8 \times 8 \times 2$ の水平サブサンプルブロックに対する具体例では、 $g_i = 0$ に対応する合致数のみを用い、評価値は、新たなリストが少なくとも 30 個以上の合致数を含む場合にのみ精密化される。

【0214】

変形例として、潜在的 DR 値をクリッピングし、DR 評価値を再計算することにより DR 評価値の精度を高めることができる。詳しくは、一具体例において、リスト D は、構成要素 d_i から構成され、 d_i は x_i を y_i に一致させる DR 値を含む。より具体的には、以下の式のとおりとなる。

【0215】

$$d_i = 2Q(y_i - \text{MIN}) / (0.5 + q_i)$$

以下に示すように、各 d_i をクリッピングすることにより精度が高められる。

【0216】

$$d_i' = \max(\min(UB, d_i), LB)$$

ここで、DR $c_1 s$ は d_i' の平均となるように算出される。重み付け平均において、クリップ処理 (DR $c_1 s$) を他の DR 評価、例えば DR $l s e$ と組み合わせて最終の DR 値を求めるようにしてもよい。例えば、重み平均 DR $e s t$ を以下の式に基づいて算出する。

【0217】

$$DR\ e\ s\ t = w_1 (DR\ c_1\ s) + w_2 (DR\ l\ s\ e)$$

重み w_1 及び w_2 は、好ましくは、結果として得られた評価値及び特定の重みから生成された画像に基づいて経験的に決定される。一具体例においては、 $w_1 = 0.22513$ とし、 $w_2 = 0.80739$ とする。

【0218】

本発明を好ましい実施の形態に基づいて説明した。上述の説明から、多数の代替、修正、変形及び用法が実現できることは当業者にとって明らかである。

【図面の簡単な説明】

【図1】

FIG. 1 は、信号の符号化、伝送、復号の処理を包括的に示す図である。

【図2】

FIG. 2 は、パケット構造の具体例を示す図である。

【図3】

FIG. 3 は、本発明に基づく符号化処理の一例を説明するフローチャートである。

【図4】

FIG. 4 は、本発明に基づく復号処理の一例を説明するフローチャートである。

【図5】

FIG. 5 は、本発明に基づく画像 - ブロックマッピングの一例を示す図である。

【図6】

FIG. 5 a は、画像 - ブロックマッピングにおいて用いられるシャッフリングパターンの一例を示す図である。

【図7】

FIG. 6 は、相補的及び連鎖的なブロック構造の例を示す図である。

【図8】

FIG. 7 a 及び FIG. 7 b は、フレームセット内の Y ブロックのシャッフリングパターンの具体例を説明する図である。

【図9】

FIG. 7 c は、フレームセット内の Y ブロックのシャッフリングパターンの具体例を説明する図である。

【図10】

FIG. 7 d は、フレームセット内の Y ブロックのシャッフリングパターンの具体例を説

明する図である。

【図 1 1】

F I G . 8 は、バッファ 0 内の累積的 D R 分布の具体例を示す図である。

【図 1 2】

F I G . 8 a は、本発明に基づく部分バッファリング処理の具体例を示す図である。

【図 1 3】

F I G . 9 は、本発明に基づくバッファ内 Y U V ブロックシャッフリング処理の具体例を示す図である。

【図 1 4】

F I G . 1 0 は、本発明に基づくグループ内 V L データシャッフリング処理の具体例を示す図である。

【図 1 5】

F I G . 1 1 は、本発明に基づく 3 ブロックグループ内における Q コードの結合の具体例を示す図である。

【図 1 6】

F I G . 1 1 a は、本発明に基づく動ブロックを含むフレーム対における Q コードの結合の具体例を示す図である。

【図 1 7】

F I G . 1 2 は、1 / 6 パーストエラー欠落に起因する画素データエラーの具体例を示す図である。

【図 1 8】

F I G . 1 2 a は、本発明に基づく Q コードのシャッフリング及び Q コードビットの分散の具体例を示す図である。

【図 1 9】

F I G . 1 2 b は、Q コードを再分散した場合における 1 / 6 パーストエラーに起因する画素データエラーの具体例を示す図である。

【図 2 0】

F I G . 1 2 c は、Q コードを再割当した場合における 1 / 6 パーストエラーに起因する画素データエラーの具体例を示す図である。

【図 2 1】

F I G . 1 3 は、本発明に基づく M I N シャッフリングの具体例を示す図である。

【図 2 2】

F I G . 1 3 a は、モーションフラグシャッフリング及び 1 フレーム対内における固定長データ欠落を示す図である。

【図 2 3】

F I G . 1 4 は、モジュールシャッフリングの具体例を示す図である。

【図 2 4】

F I G . 1 4 a は、モジュールシャッフリングの結果及びモジュールシャッフリングに関連する固定長データ欠落を示す図である。

【図 2 5】

F I G . 1 4 b は、変形されたモジュールシャッフリングの結果及びこのモジュールシャッフリングに関連する固定長データ欠落を示す図である。

【図 2 6】

F I G . 1 4 c は、変形されたモジュールシャッフリングの結果及びこのモジュールシャッフリングに関連する固定長データ欠落を示す図である。

【図 2 7】

F I G . 1 5 は、フレームセット内の可変長データバッファリングの具体例を示す図である。

【図 2 8】

F I G . 1 6 は、本発明に基づくセグメント間 V L データシャッフリングを説明する図で

ある。

【図29】

FIG. 17は、本発明に基づくデータ回復処理の具体例を示すフローチャートである。

【図30】

FIG. 18は、本発明に基づくQビット及びモーションフラグ回復処理の具体例を示すフローチャートである。

【図31】

FIG. 19は、復号候補の具体例を示す図である。

【図32】

FIG. 20a及びFIG. 20bは、FIG. 18に示すQビット及びモーションフラグ回復処理において用いられる測定を説明する図である。

【図33】

FIG. 20c及びFIG. 20dは、FIG. 18に示すQビット及びモーションフラグ回復処理において用いられる測定を説明する図である。

【図34】

FIG. 21は、Qビット及びモーションフラグ回復処理において自乗誤差確率関数を求めるために用いられる表を示す図である。

【図35】

FIG. 22は、本発明に基づくQビット、モーションフラグ及び補助情報回復処理の具体例を示すフローチャートである。

【図36】

FIG. 23は、双方向Qビット及びモーションフラグ回復処理の具体例におけるポストアンプルの使用を説明する図である。

【図37】

FIG. 24a及びFIG. 24bは、復号候補を評価する変形例を説明する図である。

【図38】

FIG. 24cは、復号候補を評価する変形例を説明する図である。

【図39】

FIG. 25は、本発明に基づく平滑度測定の具体例を示す図である。

【図40】

FIG. 26a及びFIG. 26bは、復号候補の評価法の他の具体例を説明する図である。

【図41】

FIG. 26cは、復号候補の評価法の他の具体例を説明する図である。

【図42】

FIG. 26d及びFIG. 26eは、復号候補の評価法の他の具体例を説明する図である。

【図43】

FIG. 27aは、復号候補の評価処理の他の具体例を示す図である。

【図44】

FIG. 27bは、重み付けに用いる重みを決定するための具体例を示す図である。