(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2006-191534
(P2006-191534A)
(43)公開日　平成18年7月20日(2006.7.20)

(51)Int.Cl.			F I			テーマコード（参考）
H04N	7/167	(2006.01)	H04N	7/167	Z	5C059
H04N	7/26	(2006.01)	H04N	7/13	Z	5C164
H04L	9/32	(2006.01)	H04L	9/00	675B	5J104
G09C	1/00	(2006.01)	G09C	1/00	640D	

審査請求 未請求　請求項の数 35　ＯＬ　外国語出願　（全 19 頁）

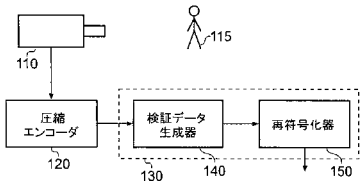
(21)出願番号　特願2005-318868 (P2005-318868)
(22)出願日　平成17年11月1日 (2005.11.1)
(31)優先権主張番号　0424205.3
(32)優先日　平成16年11月1日 (2004.11.1)
(33)優先権主張国　英国 (GB)

(71)出願人　593081408
ソニー・ユナイテッド・キングダム・リミテッド
Ｓｏｎｙ　Ｕｎｉｔｅｄ　Ｋｉｎｇｄｏｍ　Ｌｉｍｉｔｅｄ
イギリス国　サリー，ウェーブリッジ，ブルックランズ，ザ　ハイツ（番地なし）
(74)代理人　100067736
弁理士　小池　晃
(74)代理人　100086335
弁理士　田村　榮一
(74)代理人　100096677
弁理士　伊賀　誠司

最終頁に続く

(54)【発明の名称】データ処理装置及びデータ処理方法

(57)【要約】　（修正有）
【課題】保護されたメディア信号を表す保護された圧縮符号化データを生成するデータ符号化処理装置を提供する。
【解決手段】供給されたビデオ信号を圧縮符号化して圧縮された符号化データを生成しこれにより通信されるデータ量を低減する圧縮エンコーダ１２０と、検証データを生成して圧縮ビデオ信号に埋め込み圧縮ビデオ信号及び検証データを受信装置に送信する検証器１３０とを備える。検証器１３０は、圧縮されたビデオデータから検証データを生成する検証データ生成器１４０と、圧縮されたビデオデータ及び検証データを含む保護された圧縮ビデオデータを生成して受信装置に供給する再符号化器１５０とを備える。
【選択図】図１



【特許請求の範囲】
【請求項１】
保護されたメディア信号を表す保護された圧縮符号化データを生成するデータ符号化処理装置において、
メディア信号を表す圧縮符号化データが供給され、所定の基準に基づいて、該圧縮符号化データ内のデータビットを選択する際に、選択したデータビットが変更された後に該圧縮符号化データを復号することによって再生されるメディア信号に対する該選択したデータビットの知覚性が比較的低くなるように選択するデータ解析器と、
上記圧縮符号化データ内の選択したデータビットを所定の値に設定することによって正規化された圧縮符号化データを生成し、該正規化された圧縮符号化データの安全なハッシュ値を生成し、該正規化された圧縮符号化データの上記選択したデータビットを安全なハッシュ値のデータビットに置換することによって保護された圧縮符号化データを生成する情報検証エンジンとを備えるデータ符号化処理装置。
【請求項２】
上記安全なハッシュ値は、上記正規化された圧縮符号化データ及び秘密鍵の両方について安全なハッシュ値を算出することによって生成されたメッセージ認証コードであることを特徴とする請求項１記載のデータ符号化処理装置。
【請求項３】
上記安全なハッシュ値は、上記正規化された圧縮符号化データについてハッシュ値を算出し、秘密鍵を用いて、該算出したハッシュ値を暗号化することによって生成されることを特徴とする請求項１記載のデータ符号化処理装置。
【請求項４】
上記圧縮符号化データは、１つ以上のデータのブロックを含み、該各ブロックは、１つ以上のコードを含み、該各コードは、１つ以上の連続するデータビットを含み、上記データ解析器は、所定の基準に基づいて候補コードを選択し、上記知覚性が低いデータビットは、該候補コードから選択されることを特徴とする請求項１乃至３いずれか１項記載のデータ符号化処理装置。
【請求項５】
上記候補コードは、所定の長さ条件を満たすことを特徴とする請求項４記載のデータ符号化処理装置。
【請求項６】
上記候補コードは、データブロック内における所定の位置に従うことを特徴とする請求項４又は５記載のデータ符号化処理装置。
【請求項７】
上記候補コードは、所定の量子化条件を満たすことを特徴とする請求項４乃至６いずれか１項記載のデータ符号化処理装置。
【請求項８】
上記データ解析器は、所定数の選択したデータビットが特定されると、該選択したデータビットの特定を中止することを特徴とする請求項１乃至７いずれか１項記載のデータ符号化処理装置。
【請求項９】
上記データ解析器は、変更された場合に、上記圧縮符号化データを復号することによって再生されるメディア信号に対する知覚性がより低い新たに選択するデータビットが発見されると、先の選択したデータビットを破棄することを特徴とする請求項１乃至７いずれか１項記載のデータ符号化処理装置。
【請求項１０】
メディア信号が供給され、該メディア信号から圧縮符号化データを生成し、上記データ解析器に圧縮符号化データを供給する圧縮エンコーダを更に備える請求項１乃至９いずれか１項記載のデータ符号化処理装置。
【請求項１１】

上記データ解析器は、上記ビットストリームの部分から知覚性が低いデータビットを選択し、上記情報検証エンジンは、該ビットストリームの部分から選択したデータビットを所定の値に設定することによって正規化されたビットストリームの部分を生成し、該正規化されたビットストリームの部分から安全なハッシュ値を生成し、該ビットストリームの部分の選択したデータビットを該安全なハッシュ値のデータビットに置換することによって保護された圧縮符号化データを生成することを特徴とする請求項１乃至１０いずれか１項記載のデータ符号化処理装置。

【請求項１２】

上記ビットストリームの部分は、グループオブピクチャ又はグループオブピクチャ内のフレームであることを特徴とする請求項１１記載のデータ符号化処理装置。

【請求項１３】

保護されたメディア信号を表す保護された圧縮符号化データであって、所定の基準に基づいて、変更された場合に該メディア信号に対する知覚的効果が比較的低い圧縮符号化データ内の選択データビットを特定し、該選択データビットを所定の値に設定して、正規化された圧縮符号化データを生成し、該正規化された圧縮符号化データの安全なハッシュ値を算出し、該正規化された圧縮符号化データの選択データビットを安全なハッシュ値のデータビットに置換することによって算出した検証データを含む保護された圧縮符号化データを復号して、該保護されたメディア信号を検証するデータ復号処理装置において、

上記保護された圧縮符号化データが供給され、上記所定の基準に基づいて、該保護された圧縮符号化データ内の選択データビットを特定するデータ解析器と、

上記選択データビットから検証データを抽出し、該検証された圧縮符号化データ内の選択データビットを所定の値に設定することによって、検証され、正規化された圧縮符号化データを生成し、該検証され、正規化された圧縮符号化データのハッシュ値を算出し、上記抽出された検証データと上記算出したハッシュ値とを比較し、該検証データ及び該算出したハッシュ値が一致する場合、上記保護された圧縮符号化データの妥当性を認証する認証エンジンとを備えるデータ復号処理装置。

【請求項１４】

上記検証された圧縮符号化データ内の安全なハッシュ値は、上記正規化された圧縮符号化データ及び秘密鍵の両方のハッシュ値を算出することによって生成されたメッセージ認証コードであり、上記認証エンジンによって算出されたハッシュ値は、上記正規化された圧縮符号化データ及び秘密鍵の両方のハッシュ値を算出することによって生成されたメッセージ認証コードであることを特徴とする請求項１３記載のデータ復号処理装置。

【請求項１５】

上記検証された圧縮符号化データの安全なハッシュ値は、上記正規化された圧縮符号化データのハッシュ値を算出し、秘密鍵を用いて該算出したハッシュ値を暗号化することによって生成されており、上記認証エンジンは、該秘密鍵に対応する公開鍵を用いて上記検証データを復号し、該検証データ内のハッシュ値を再生し、該再生したハッシュ値を算出したハッシュ値と比較することを特徴とする請求項１３記載のデータ復号処理装置。

【請求項１６】

上記圧縮符号化データは、１つ以上のデータのブロックを含み、該各ブロックは、１つ以上のコードを含み、該各コードは、１つ以上の連続するデータビットを含み、上記データ解析器は、所定の基準に基づいて候補コードを選択し、上記知覚的効果が低いデータビットは、該候補コードから選択されることを特徴とする請求項１３乃至１５いずれか１項記載のデータ復号処理装置。

【請求項１７】

上記候補コードは、所定の長さ条件を満たすことを特徴とする請求項１６記載のデータ復号処理装置。

【請求項１８】

上記候補コードは、データブロック内における所定の位置に従うことを特徴とする請求項１６又は１７記載のデータ復号処理装置。

【請求項１９】
上記候補コードは、所定の量子化条件を満たすことを特徴とする請求項１６乃至１８いずれか１項記載のデータ復号処理装置。
【請求項２０】
上記データ解析器は、所定数の選択データビットが特定されると、該選択データビットの特定を中止することを特徴とする請求項１３乃至１９いずれか１項記載のデータ復号処理装置。
【請求項２１】
上記データ解析器は、変更された場合に上記圧縮符号化データを復号することによって再生されるメディア信号に対する知覚的効果がより低い新たに選択するデータビットが発見されると、先の選択データビットを破棄することを特徴とする請求項１３乃至１９いずれか１項記載のデータ復号処理装置。
【請求項２２】
上記検証データは、上記圧縮符号化データの部分において、上記所定の基準に基づいて、選択データビットを特定することによって判定され、該圧縮符号化データの部分における該選択データビットは、安全なハッシュ値の算出のために正規化され、安全なハッシュ値のデータビットによって置換され、
上記認証エンジンは、上記圧縮符号化データの部分の選択データビットから上記検証データを抽出し、上記検証された圧縮符号化データの部分内の選択データビットを所定の値に設定することによって検証され、正規化された圧縮符号化データを生成し、該検証され、正規化された圧縮符号化データの部分のハッシュ値を算出し、該算出したハッシュと上記抽出された検証データとを比較し、該検証データ及び該算出したハッシュ値が一致する場合、上記保護された圧縮符号化データの妥当性を認証する請求項１３乃至２１いずれか１項記載のデータ復号処理装置。
【請求項２３】
上記圧縮符号化データの部分は、グループオブピクチャ又はグループオブピクチャ内のフレームであることを特徴とする請求項２２記載のデータ復号処理装置。
【請求項２４】
保護されたメディア信号を表す保護された圧縮符号化データを生成するデータ符号化処理方法において、
メディア信号を表す圧縮符号化データを受け取るステップと、
所定の基準に基づいて、上記圧縮符号化データにおいて、選択データビットが変更された場合に該圧縮符号化データを復号することによって再生されるメディア信号に対する知覚性が比較的低くなるように選択した該選択データビットを特定するステップと、
上記圧縮符号化データ内の選択データビットを所定の値に設定することによって正規化された圧縮符号化データを生成するステップと、
上記正規化された圧縮符号化データの安全なハッシュ値を算出するステップと、
上記正規化された圧縮符号化データの上記選択データビットを安全なハッシュ値のデータビットに置換することによって保護された圧縮符号化データを生成するステップとを有するデータ符号化処理方法。
【請求項２５】
上記安全なハッシュ値は、上記正規化された圧縮符号化データ及び秘密鍵の両方について安全なハッシュ値を算出することによって生成されたメッセージ認証コードであることを特徴とする請求項２４記載のデータ符号化処理方法。
【請求項２６】
上記安全なハッシュ値は、上記正規化された圧縮符号化データについてハッシュ値を算出し、秘密鍵を用いて、該算出したハッシュ値を暗号化することによって生成されることを特徴とする請求項２４記載のデータ符号化処理方法。
【請求項２７】
メディア信号を表す保護された圧縮符号化データであって、所定の基準に基づいて、変

更された場合に該メディア信号に対する知覚的効果が比較的低くなるように選択された圧縮符号化データ内の選択データビットを特定し、該選択データビットを所定の値に設定して正規化された圧縮符号化データを生成し、該正規化された圧縮符号化データの安全なハッシュ値を算出し、該正規化された圧縮符号化データの選択データビットを安全なハッシュ値のデータビットに置換することによって算出した検証データを含む保護された圧縮符号化データを復号して、該メディア信号を検証するデータ復号処理方法において、

上記保護された圧縮符号化データを受け取るステップと、

上記所定の基準に基づいて、上記保護された圧縮符号化データ内の選択データビットを特定するステップと、

上記選択データビットから検証データを抽出するステップと、

上記検証された圧縮符号化データ内の選択データビットを所定の値に設定することによって検証され、正規化された圧縮符号化データを生成するステップと、

上記検証され、正規化された圧縮符号化データのハッシュ値を算出するステップと、

上記抽出された検証データと上記算出したハッシュ値とを比較し、該検証データ及び該算出したハッシュ値が一致する場合、上記保護された圧縮符号化データの妥当性を認証するステップとを有するデータ復号処理方法。

【請求項２８】

上記検証された圧縮符号化データ内の安全なハッシュ値は、上記正規化された圧縮符号化データ及び秘密鍵の両方のハッシュ値を算出することによって生成されたメッセージ認証コードであり、上記算出したハッシュ値は、上記正規化された圧縮符号化データ及び秘密鍵の両方のハッシュ値を算出することによって生成されたメッセージ認証コードであることを特徴とする請求項２７記載のデータ復号処理方法。

【請求項２９】

上記検証された圧縮符号化データの安全なハッシュ値は、上記正規化された圧縮符号化データのハッシュ値を算出し、秘密鍵を用いて該算出したハッシュ値を暗号化することによって生成されており、上記検証データを、該秘密鍵に対応する公開鍵を用いて復号し、該検証データ内のハッシュ値を再生し、該再生したハッシュ値を算出したハッシュ値と比較するステップを有する請求項２７記載のデータ復号処理方法。

【請求項３０】

データプロセッサにロードされて、該データプロセッサに請求項２４乃至２９いずれか１項記載の方法を実行させるプログラムコードを含むコンピュータソフトウェア。

【請求項３１】

データプロセッサにロードされて、該データプロセッサに請求項２４乃至２９いずれか１項記載の方法を実行させるプログラムコードを提供するデータ提供媒体。

【請求項３２】

当該データ提供媒体は、記録媒体であることを特徴とする請求項３１記載のデータ提供媒体。

【請求項３３】

メディア信号を表す保護された圧縮符号化データであって、所定の基準に基づいて、変更された場合に該メディア信号に対する知覚的効果が比較的低くなるように選択された圧縮符号化データ内の選択データビットを特定し、該選択データビットを所定の値に設定して正規化された圧縮符号化データを生成し、該正規化された圧縮符号化データの安全なハッシュ値を算出し、該正規化された圧縮符号化データの選択データビットを安全なハッシュ値のデータビットに置換することによって算出した検証データを含む保護された圧縮符号化データを表すデータ信号。

【請求項３４】

請求項３３記載のデータ信号を搬送するデータ担体。

【請求項３５】

請求項２４乃至２６いずれか１項記載の方法によって生成された保護された圧縮符号化データを搬送するデータ担体。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、圧縮データを検証するデータ処理装置及びデータ処理方法に関する。本発明の実施形態は、データ符号化処理装置及びデータ符号化処理方法、検証されたメディア信号を表す検証された圧縮符号化データを生成するデータ復号処理装置及びデータ復号処理方法を提供する。
【背景技術】
【０００２】
デジタル圧縮データを普及させる際、圧縮データを権限なく変更することを防止することが望ましい。この望ましくない変更を防ぐためのコピー保護及び認証技術として、公開鍵／秘密鍵暗号化及びハッシュにより生成されたデジタル署名を用いる手法がある。論文「ＭＰＥＧ－２圧縮ビデオの実時間ラベル付け（"Real-time Labelling of MPEG-2 Compressed Video" <4> by Gerrit C. Langelaar, Reginald L. Lagendijk and Jan Biemond of Delft University of Technology, The Netherlands）」には、デジタルビデオデータに対し、実時間でラベル付けを行うことによって変更を防ぐ技術の具体例が開示されている。
【発明の開示】
【課題を解決するための手段】
【０００３】
本発明の第１の側面であるデータ符号化処理装置は、保護されたメディア信号を表す保護された圧縮符号化データを生成する。データ符号化処理装置は、メディア信号を表す圧縮符号化データが供給され、圧縮符号化データ内のデータビットを選択する際に、選択したデータビットが変更された後に圧縮符号化データを復号することによって再生されるメディア信号に対する該選択したデータビットの知覚性が比較的低くなるように選択するデータ解析器を備える。このデータビットは、所定の基準に基づいて選択される。更に、データ符号化処理装置は、圧縮符号化データ内の選択したデータビットを所定の値に設定することによって正規化された圧縮符号化データを生成し、正規化された圧縮符号化データの安全なハッシュ値を生成する情報検証エンジンを備える。そして、情報検証エンジンは、正規化された圧縮符号化データの選択したデータビットを安全なハッシュ値のデータビットに置換することによって保護された圧縮符号化データを生成する。
【０００４】
ハッシュ値は、ハッシュアルゴリズムを用いて入力ビットストリームから生成され、元のメッセージを表す「メッセージダイジェスト」と呼ばれるディジットの一意的な固定長ストリング（通常１６０ビット）を含む。ハッシュ値をリバースエンジニアリングして元のメッセージを判定することはできず、２つの異なる入力からは、通常、同じメッセージダイジェストが生成されることはない。ハッシュ値は、これらの特性のために、ビットストリームを一意的に特定する目的に特に有用であり、したがって、信頼性、特に、ハッシュ値が生成された後にビットストリームが変更されているか否かを判定するために用いることができる。
【０００５】
認証を目的として、ハッシュ等の検証データをビットストリームに加えるには、ビットストリームに検証データを挿入してもよく、ビットストリームの始め又は終わりに検証データを設けてもよい。この場合、ビットストリームからは如何なるデータも失われないが、ビットストリームの長さが長くなってしまう。これに代えて、ビットストリームの既存のデータビットを検証データの検証ビットに置換することもできる。この場合、ビットストリームを同じ長さのままにすることができるが、ビットストリームから元のデータの一部が失われる。
【０００６】
ビットストリームを変更する前と後にビットストリームのハッシュを生成した場合、そ

れぞれ異なるハッシュ値が生成される。したがって、選択したビットをハッシュ値に置換することによってビットストリームを表すハッシュ値をビットストリームに埋め込めば、ビットストリームは変更され、後に変更されたビットストリームのハッシュ値を算出すると、ビットストリーム自体が搬送するハッシュ値とは異なる値となる。

【０００７】

本発明では、ハッシュデータのビットに置換されるデータビットが、ハッシュを生成する度に同じ値を有するべきであり、この値が異なれば、如何なるハッシュベースの認証ルーチンも、ビットストリームに変更が加えられており、ビットストリームは認証されたものではないと判定してしまうことに注目した。そこで、本発明では、選択したビットの全てが、所定の既知の値に設定された圧縮された情報信号の正規化されたバージョンを生成する。そして、この圧縮された情報信号の正規化されたバージョンからハッシュを生成する。このようにハッシュを生成し、保存することにより、圧縮符号化データ内のどのビットがハッシュ値を保存するために選択したビットであるかを判定し、保存されたハッシュ値を抽出し、選択したビットを、先に保存すべきハッシュ値を生成する際に用いられた所定の値に設定することによって、後に圧縮符号化データを認証できる。そして、ビットをそれらの所定の値に設定して、圧縮符号化データのハッシュを算出する。そして、新たに生成されたハッシュ値を抽出されたハッシュ値と比較することによって、圧縮符号化データに変更が加えられているか否かを判定することができる。この構成により、圧縮ビットストリームへのハッシュデータの埋込は、ハッシュ生成に影響せず、したがって、認証処理の妨げとはならない。

【０００８】

安全性を高め、ハッシュ値が置換されることを防止するために、ハッシュ値は、暗号化される。暗号化には、多くの異なる形式を用いることができる。例えば、ハッシュ値を生成した後に、ハッシュ値を圧縮符号化データに埋め込んで検証された圧縮された情報信号を生成する前に、秘密鍵を用いてハッシュ値を暗号化することにより、ハッシュ値にデジタル署名してもよい。この場合、ハッシュを復号するためには、対応する公開鍵が必要である。これに代えて、ハッシュ値は、単に圧縮ビットストリームだけではなく、秘密鍵にも基づいて、メッセージ認証コードの形式で生成してもよい。この場合、復号装置側では、圧縮ビットストリームが有効であるか否か、及び復号するべきであるか否かを判定するために、秘密鍵が必要である。秘密鍵／公開鍵暗号化を用いる前者の手法では、２段階の検証処理を行い、詳しくは、デジタル署名を検証した後、ハッシュ値自体を検証する。ハッシュ値を暗号化するために用いられた秘密鍵に対応する公開鍵を有していれば、如何なる機器でもデータを検証することができる。なお、ハッシュにデジタル署名すると、ビットストリームに挿入するべき検証データが比較的大きくなる。例えば、ハッシュ値自体の長さは、１６０ビットであっても、対応するデジタル署名の長さは、約１０２４ビットとなることがある。一方、メッセージ認証コードを用いる手法では、圧縮ビットストリームに変更が加えられているか否かを判定するためには、単一の検証ステップを行えばよいが、この場合、メッセージ認証コードを生成した際に用いた秘密鍵が必要となり、したがって、圧縮ビットストリームの発信者は、如何なる変更も行われていないことを実際に確かめる必要がある。なお、メッセージ認証コードは、サイン付きハッシュ値より遙かに小さく、例えば、約１２８ビット程度である。

【０００９】

本発明の第２の側面に基づくデータ復号処理装置は、保護されたメディア信号を表す保護された圧縮符号化データを復号することによって、保護されたメディア信号を検証する。このデータ復号処理装置は、所定の基準に基づいて、変更された場合にメディア信号に対する知覚的効果が比較的低い圧縮符号化データ内の選択データビットを特定し、選択データビットを所定の値に設定して正規化された圧縮符号化データを生成し、正規化された圧縮符号化データの安全なハッシュ値を算出し、正規化された圧縮符号化データの選択データビットを安全なハッシュ値のデータビットに置換することによって算出した検証データを含む保護された圧縮符号化データが供給される。データ復号処理装置は、保護された

圧縮符号化データが供給され、所定の基準に基づいて、保護された圧縮符号化データ内の選択データビットを特定するデータ解析器を備える。更に、データ復号処理装置は、検証された圧縮符号化データ内の選択データビットを所定の値に設定することによって検証され、正規化された圧縮符号化データを生成し、検証され、正規化された圧縮符号化データのハッシュ値を算出する認証エンジンを備える。そして、データ復号処理装置は、抽出された検証データと算出したハッシュ値とを比較し、検証データ及び算出したハッシュ値が一致する場合、保護された圧縮符号化データの妥当性を認証する。

【００１０】

本発明の更なる各側面及び特徴は、添付の特許請求の範囲に定義されており、データ符号化処理方法、データ復号処理方法、コンピュータソフトウェア及びデータ提供媒体を含む。

【発明を実施するための最良の形態】

【００１１】

ビデオ情報の記録及び符号化を行う例示的な装置の構成を図１に示す。この装置は、シーン１１５に対応するビデオ信号を生成するビデオカメラ１１０と、ビデオカメラ１１０からビデオ信号が供給され、ビデオ信号を圧縮符号化し、圧縮されたビデオデータを生成し、これにより、通信されるデータ量を低減する圧縮エンコーダ１２０と、検証データを生成し、圧縮ビデオ信号に埋め込み、圧縮ビデオ信号及び検証データを受信装置（図７を参照して後述する）に送信する検証器１３０とを備える。検証器１３０は、圧縮されたビデオデータから検証データを生成する検証データ生成器１４０と、圧縮されたビデオデータ及び検証データを含む保護された圧縮ビデオデータを生成し、保護された圧縮ビデオデータを受信装置に供給する再符号化器１５０とを備える。

【００１２】

図２は、図１に示す検証器１３０に対応する、検証データを圧縮ビットストリームに埋め込むための例示的な装置を示している。図２に示す装置は、例えば、図１の圧縮エンコーダ１２０によって生成された圧縮ビットストリーム等のコードを含む圧縮ビットストリーム２０５が供給される解析器２１０を備える。解析器２１０は、検証データに置換することによって変更される、ビットストリーム内の候補ビットを特定する。具体的には、解析器２１０は、ビットストリーム内で、コードにおける１つ以上の重要性が低いビット（least important bit）を検証データに置換することによって変更するための候補コードを特定する。これらの重要性が低いビットを候補ビットと呼び、候補コード内のこれらの候補ビットは、候補コードを復号した際に影響が最も小さいビットである。幾つかの圧縮技術でこれらの重要性が低いビットは、コードの最下位ビット（least significant bit：ＬＳＢ）であってもよいが、他の圧縮技術においては、重要性が低いビットは、候補コード内の他の位置に存在するビットであってもよい。解析器２１０は、特定された候補コードに対応する選択データを出力する。

【００１３】

選択される候補コードは、変更してもビットストリームに含まれている情報に対する影響が小さいコードであり、好ましくは、変更してもビットストリームに含まれている情報に対する影響が最も小さいコードである。圧縮ビットストリームは、解析器２１０によって生成された候補コード選択データと共に情報検証エンジン２２０に渡され、情報検証エンジン２２０は、検証データを生成し、ビットストリームに埋め込む。具体的には、情報検証エンジンは、正規化器２３０を備え、正規化器２３０は、解析器２１０から圧縮ビットストリーム及び候補コード選択データが供給され、各候補コードの１つ以上の候補ビットを所定の値に設定することによって正規化されたビットストリームを生成する。なお、正規化されたビットストリーム内の全ての候補ビットは、例えば、０又は１等の単一の値に設定してもよく、或いは、例えば、一連の候補ビットに亘って、０と１を交互に並べる等、０と１の所定のパターンに基づく値に設定してもよい。

【００１４】

正規化されたビットストリームは、ハッシュ算出器２４０に渡され、ハッシュ算出器２

４０は、正規化されたビットストリームからハッシュを生成する。そして、算出されたハッシュは、暗号化モジュール２６０に渡され、暗号化モジュール２６０は、供給された秘密鍵２５０を用いてハッシュを暗号化し、サイン付きハッシュ（signed hash）を生成する。そして、正規化器２３０からの正規化されたビットストリーム及び暗号化モジュール２６０からのサイン付きハッシュの両方は、埋込器２７０に渡され、埋込器２７０は、正規化されたビットストリーム内の正規化された候補ビットをサイン付きハッシュの各ビットに置換して保護された圧縮ビットストリーム２８０を生成する。保護された圧縮ビットストリーム２８０内にサイン付きハッシュを埋め込むことにより、サイン付きハッシュが埋め込まれた後に、保護された圧縮ビットストリーム２８０が全く変更されていないことを検証できるようになる。すなわち、ハッシュを暗号化してサイン付きハッシュを生成することにより、保護された圧縮ビットストリーム内のハッシュ値を改竄するあらゆる試みを検出でき、システムを安全にすることができるが、このために、多くのデータをビットストリームに埋め込む必要がある。また、他のハッシュ処理を行い、ビットストリームにデータを埋め込むこともできる。例えば、２つ以上の連続したデータブロックの連続したハッシュ（running hashes）を生成し、データのブロックがビットストリームから抜き取られ又はビットストリームの他の領域に移動することを防ぐことができる。更に、ハッシュは、データの連続したブロックに関して生成された連続したハッシュであってもよい。

【００１５】

図３は、図１を参照して説明した検証器１３０に対応する、検証データを圧縮ビットストリームに埋め込むための装置の代替例を示している。図３に示す装置は、圧縮ビットストリーム３０５が供給され、圧縮ビットストリーム３０５に基づいて候補選択データを生成する解析器３１０を備える。解析器３１０は、図２の解析器２１０に対応し、同様の処理を行うため、ここでは説明を省略する。解析器３１０は、ビットストリームを解析し、対応する候補コード選択データを情報検証エンジン３２０に渡し、情報検証エンジン３２０は、検証データを生成し、ビットストリームに埋め込む。具体的には、情報検証エンジン３２０は、解析器３１０から圧縮ビットストリーム及び候補コード選択データが供給され、各候補コードの１つ以上の候補ビットを所定の値に設定することによって正規化されたビットストリームを生成する正規化器３３０を備える。正規化器３３０は、図２を参照して説明した正規化器２３０に対応し、同様の処理を行うため、ここでは説明を省略する。

【００１６】

正規化されたビットストリームは、供給された秘密鍵３５０を用いて、正規化されたビットストリームからメッセージ認証コード（message authentication code：以下、ＭＡＣという。）を生成するメッセージ認証コード算出器３４０に渡される。ＭＡＣは、上述した標準のハッシュとは異なり、入力ビットストリームのみではなく、秘密鍵にも基づいて生成される鍵付きハッシュの一種である。したがって、この手法では、メッセージ認証コードの生成時に、メッセージ認証コードが既に暗号化されているので、メッセージ認証コードを暗号化する暗号化モジュールは不要である。そして、正規化器３３０からの正規化されたビットストリーム及びメッセージ認証コード算出器３４０からのＭＡＣの両方は、埋込器３６０に渡され、埋込器３６０は、正規化されたビットストリーム内の正規化された候補ビットをＭＡＣの各ビットに置換し、保護された圧縮ビットストリーム３８０を生成する。図２のサイン付きハッシュと同様に、保護された圧縮ビットストリーム３８０内に埋め込まれたメッセージ認証コードにより、メッセージ認証コードが埋め込まれた後に、保護された圧縮ビットストリーム３８０が全く変更されていないことを検証できるようになる。

【００１７】

図４に示す例示的なＭＰＥＧ－２ビットストリームフラグメントは、４つの可変長コード（variable length code：ＶＬＣ）４１０、４２０、４３０、４４０と、１つの固定長コード（fixed length code：ＦＬＣ）４５０と、エンドオブブロック（end of block：ＥＯＢ）コード４６０とを有する。ここに例示するビットストリームフラグメントは、Ｍ

ＰＥＧ－２ビットストリームフラグメントであるが、ビデオデータ、オーディオデータ又は他のデータのこの他のデータフォーマットを用いてもよい。可変長コードは、このコードが表す情報に応じて、様々な異なる長さを有する。例えば、ＭＰＥＧ－２における可変長コードは、２ビットから１６ビットまでの範囲の長さを有する。一方、ＭＰＥＧ－２における固定長コードは、例えば、２４ビット等、常に長さが固定されている。固定長コードの２４ビットのうち、最初の６ビットは、エスケープコードを指定し、次の６ビットは、固定長コードのランレングスを指定し、最後の１２ビットは、コード値を指定する。

【００１８】

ビットストリーム内の既存のビットを検証ビットに置換することによって検証データをビットストリームに挿入する場合、この処理によるビットストリームの変化は、ビットストリームによって表される情報コンテンツへの影響が比較的小さくなることを確実にする必要がある。これは、ビットストリームによって表される情報における大きな振幅の変化に対応するビットストリーム内のコードを特定し、特定されたコードの最下位ビット（ＬＳＢ）を検証ビットに置換することによって実現される。ＭＰＥＧ－２圧縮ビットストリームの場合、ビットストリームは、連続したブロックを含み、各ブロックは、コードワードのシーケンスを含み、画像フレームの部分を表す。ビットストリーム内の各ブロックは、エンドオブブロック（ＥＯＢ）コードで終わり、エンドオブブロックコードに続く全てのコードは、後続するブロックに関連する。

【００１９】

なお、最も重要でないビット（非符号ビット）を異なる値に切り換えても、幾つかの可変長コード及び殆どの固定長コードの基本的意味は、維持される。具体的には、固定長コードの場合、コードの最下位ビットを、同じランレングスに対応し、コードが複数個以上のビットを含むと仮定して、殆ど同じ信号レベルを有する同じ長さ（ビット数）の他のコードに変更する。可変長コードの場合、ある所定の可変長コードについてのみ、この原理を適用する。

【００２０】

どのコードが置換のために適切な候補であるかは、多くの因子によって判定する。例えば、コードが長いほど、置換のための良い候補となる傾向があり、固定長コード（ＦＬＣ）は、それらのコード値部分の長さ（この具体例では、１２ビット）のために、特に良い候補であり、２つ以上の最下位ビットを検証データビットによって置換しても導入される歪みは許容範囲内に収まる。例えば、圧縮符号化データが、２ビットから１６ビットまでの範囲で長さが変化する可変長コード及び常に１２ビットの長さのコード値部分を含む固定長コードを含んでいる場合、データ解析器は、長さが１０ビットを超える可変長コードのみ又はブロック内の任意の固定長コードから候補コードを選択してもよい。なお、エンドオブブロック（ＥＯＢ）コードは、変更するべきでない。

【００２１】

候補コードを選択する際、圧縮アルゴリズムによって適用された何らかの適応型の量子化、特に各コードに適用された量子化の度合いを考慮する必要がある。コードが画像の部分の離散コサイン変換（Discrete Cosine Transform：ＤＣＴ）係数を表すＭＰＥＧ－２ビットストリーム内のコードの場合、最下位ビットを変更することにより、ＤＣＴ係数を逆量子化したときの振幅における変化が許容できる範囲内にあると判定された場合に限り、そのコードに検証データを挿入することができる。したがって、検証データの挿入には、粗く量子化されたＤＣＴ係数より細かく量子化されたＤＣＴ係数が適する。ＭＰＥＧ－２では、ビットストリーム内のＤＣＴ係数に適用される量子化レベルを決定する量子化設定は、通常、各データブロックのヘッダ部分に示されており、この情報は、ビットストリームを完全に復号しなくても抽出することができる。コードに適用された量子化のレベルが高いほど、コードのビットの１つを変更した場合にコードの逆量子化されたバージョンにおける影響が大きくなるため、候補コードを選択する場合、所定の閾値を下回る量子化レベルによって量子化されたコードを優先する。したがって、所定の閾値を下回る量子化のレベルで量子化されたコードの最下位ビットを置換することにより、画像の視覚的な変

化を小さく抑えることができる。

【００２２】

ＭＰＥＧ－２においては、多くの場合、ブロックの情報コンテンツの知覚性は、ブロックのＤＣ係数に関連するブロックの最初のコードから、ブロック内の知覚可能性が最小のゼロではないＡＣ係数に関連するブロックの最後のコードに向かって減少する。この特徴は、ＭＰＥＧ－２で用いられる量子化マトリックスの結果であり、また、多くの場合、典型的なビデオ画像は、主に各ブロックの始めの方に置かれる低周波数係数によって表現されるためである。したがって、候補コードを選択する場合、各ブロックの最後の方のコードを優先して選択することが望ましい。

【００２３】

より良い結果を得るために、これらの手法の幾つか又は全てを組み合わせて用いてもよい。例えば、コード長が最短であること、適用された量子化レベルが最大であること、及び現在のブロック内におけるコードの位置等に基づき、所定のコードがある基準を満たすか否かを判定する階層的な選択ロジックを用いてコードを選択してもよい。これらの基準は、個別に定義してもよく、例えば、用いられる閾値を固定し、候補コードとして選択するためには、コードがこれらの１つ以上を満たす必要があるとしてもよく、又は全てを満たさなければならないとしてもよい。これに代えて、これらの基準を相関的に検討してもよい。基準を相関的に検討することが有利である場合とは、例えば、１つの基準を大きく上回っているが他の基準を僅かに満たさないコードを候補コードとして選択することが望ましい場合等である。具体例として、量子化のレベルが低い第１のコードは、１１ビット以上のビット長を有する場合に候補コードとして選択し、量子化レベルが第１のコードより高い第２のコードは、１４ビット以上のビット長を有する場合のみ候補コードとして選択するようにしてもよい。

【００２４】

階層的なロジックを用いて候補コードを決定するのではなく、各基準の可能な値又は範囲をルックアップテーブルにし、このルックアップテーブルによって、コードを候補コードとして選択するための様々な評価基準の値の各組合せを指定してもよい。これに代えて、例えば、コード長さ（ビット数）、適用された量子化レベル、及びこれに関連するブロック内のコードの位置等のパラメータを変数とする１つ以上の数式表現を定義し、コードを候補コードとして選択するべきであるか否かを評価してもよい。

【００２５】

候補コードを探し出す処理は、十分な数の候補コードが見つかった場合に終了してもよい。これにより、選択処理を高速に行うことができる。或いは、候補コードを探し出す処理は、全体の画像フレームのデータに亘って実行し、先に見つけた候補を、新たに見つけたより良い候補、すなわち、変更による知覚的歪みがより小さくて済むコードに置き換えてもよく、これにより、圧縮符号化データの情報コンテンツの歪みをより小さく抑えることができる。

【００２６】

なお、圧縮符号化データは、如何なる種類の情報を表していてもよく、多くの様々な符号化法及びアルゴリズムを用いて符号化されたデータであってもよいが、好ましくは、圧縮符号化データは、ＭＰＥＧ－２符号化されたビデオデータである。

【００２７】

図５に示すように、上述した候補コードの選択法を適用することにより、ビットストリーム内のあるコードに、候補コードとしてのマークを付すことができる。これらのコードの１つ以上の下位ビットには、検証データを構成する検証ビットを保存することができる。図５に示す具体例では、可変長コード４２０’、４４０’及び固定長コード４５０’が候補コードとして選択され、可変長コード４１０’、４３０’及びエンドオブブロックコード４６０’は、候補コードとして選択されておらず、したがって、変更されない。

【００２８】

ビットストリームに挿入される検証データは、圧縮ビットストリームのハッシュを算出

することによって生成されたハッシュデータを含む。ハッシュデータは、例えば、ＳＨＡ－１アルゴリズム（安全ハッシュ規格（Secure Hash Standard：ＳＨＳ）（National Institute of Standards and Technology (NIST). FIPS Publication 180: Secure Hash Standard (SHS). May 1993 & Updated May 1994）に定義されている。）等の数式を用いて生成される。ＳＨＡ－１ハッシュアルゴリズムは、任意の長さの一連のビットを、元のメッセージを表す「メッセージダイジェスト」と呼ばれる固有の固定長ストリング（通常１６０ビット）に変換する。したがって、ビットストリームを変更する前及び後にビットストリームからハッシュを生成すると、それぞれ異なるハッシュ値が算出される。このコンテキストにおいては、ビットストリームからハッシュを生成した後にビットストリームにハッシュデータを埋め込むことによってビットストリームを変更すると、後に、変更されたビットストリームから生成したハッシュは、元のビットストリームから生成されたハッシュとは異なるものになり、したがって、２つのハッシュを比較することによって、矛盾が生じるため、ハッシュの値が不確実になる。

【００２９】

この問題は、図６に示すように、ビットストリームのハッシュを生成する前に、候補ビットを所定の値、例えば０に設定することによって解決される。図６に示すように、コード４１０”、４３０”、４６０”は、候補コードではないので、これらのコード内には、如何なる候補ビットも存在せず、候補可変長コード４２０”のＬＳＢ６２２は、０に設定され、候補可変長コード４４０”のＬＳＢ６４２は、０に設定され、候補固定長コード４５０”の２つのＬＳＢ６５２、６５４は、０に設定されている。そしてハッシュが生成された後に、ハッシュデータは、検証ビットとして候補ビットに挿入できる。後にビットストリームの認証を確認する場合、ビットストリームから検証データを抽出し、候補ビットを同じ所定の値にリセットした後、ビットストリームのハッシュを生成することができる。ビットストリームに対して、如何なる更なる変更も加えられていなければ、新たに生成されたハッシュは、ビットストリームに埋め込まれていたハッシュと同じになり、したがって、ビットストリームが正規のものであることを確認することができる。

【００３０】

保護されたビデオ情報、例えば、図１を参照して説明した装置によって生成された保護された圧縮ビデオデータが供給され、復号する例示的な装置の構成を図７に示す。この装置は、保護されたビデオ情報を受信する受信機７１０と、保護されたビデオ情報内の検証データを用いて、受信したビデオ情報が変更されているか否かを判定する認証器７２０と、ビデオ情報が変更されていないことを認証器７２０が確認した場合のみ、保護されたビデオ情報を伸張する伸張器７３０とを備える。伸張されたビデオ情報は、表示し、又は他の装置に供給することができる。

【００３１】

保護された圧縮ビットストリームに保存されている検証データ、特にサイン付きハッシュデータを検出するために動作可能な検出器の構成例を図８に示す。例えば、図２の情報検証エンジン２２０によって生成された保護されたビットストリーム２８０等の保護された圧縮ビットストリーム８０５は、解析器８１０に供給される。解析器８１０は、図２の解析器２１０と同じ基準を用いて、認証ビットストリーム内で検証ビットが存在している候補コードを判定する。したがって、ここでは、図２の埋込装置と同じ候補ビットを特定する必要がある。そして、候補選択データが生成される。また、検出器は、保護されたビットストリーム内に存在している検証データを検出し、これを新たに生成された検証データと比較し、ビットストリームに何らかの変更が加えられているか否か、したがって、ビットストリームが正規のものであるか否かを判定する認証エンジン８２０を備える。認証エンジン８２０は、解析器８１０によって、保護された圧縮ビットストリームから判定された候補ビットから、検証データを抽出する検証データ抽出器８３０を備える。そして、例えば、図２の暗号化モジュール２６０によって生成された、サイン付きハッシュを含む抽出された検証データは、復号モジュール８５０に渡され、復号モジュール８５０は、秘密鍵、例えば、図２の埋込装置においてハッシュを符号化するために用いられた秘密鍵２

５０に対応する公開鍵８４０を用いて、この抽出された検証データを復号する。

【００３２】

また、認証エンジン８２０は、正規化器８６０を備え、この正規化器８６０は、解析器８１０から保護された圧縮ビットストリーム及び候補ビット選択データが供給され、受信ビットストリーム内の候補ビットの値を、図２の埋込装置の正規化器２３０で用いた値と同じ所定の値に設定し、これにより正規化されたビットストリームを生成する。正規化されたビットストリームは、ハッシュ算出器８７０に渡され、ハッシュ算出器８７０は、正規化されたビットストリームからハッシュを生成する。この具体例では、解析器８１０及び正規化器８６０は、埋込処理において適用された基準と同じ基準を直接適用することによって保護されたビットストリームの候補コードを判定するが、候補コードの位置を特定する識別コードを保護されたビットストリームによって提供してもよい。この場合、解析器は、ビットストリームから識別コードを読み出すことによって、所定の基準を満たす候補ビットを判定する。

【００３３】

復号モジュール８５０からの復号されたハッシュ及びハッシュ算出器８７０からの算出されたハッシュは、比較モジュール８８０に渡され、比較モジュール８８０は、２つのハッシュを比較し、２つのハッシュが同じであると判定された場合、圧縮ビットストリームを正規のものであると判断する。圧縮ビットストリームの信頼性を示す信号は、認証信号８９０として出力される。認証信号８９０は、圧縮ビットストリームに対する又はこれに関する更なる処理をトリガし又は防止するために用いることができる。

【００３４】

保護された圧縮ビットストリームに保存されている検証データ、特にメッセージ認証コードを検出することができる検出器の変形例を図９に示す。保護された圧縮ビットストリーム９０５、例えば、図３の情報検証エンジン３２０によって生成された保護されたビットストリーム３８０は、解析器９１０に供給される。解析器９１０は、図３の解析器３１０と同じ基準を用いて、認証ビットストリーム内で検証ビットが存在している候補コードを判定する。したがって、ここでは、図３の埋込装置と同じ候補ビットを特定する必要がある。そして、候補選択データが生成される。また、検出器は、保護されたビットストリーム内に存在している検証データを検出し、これを新たに生成された検証データと比較し、ビットストリームに何らかの変更が加えられているか否か、したがって、ビットストリームが正規のものであるか否かを判定する認証エンジン９２０を備える。認証エンジン９２０は、解析器９１０によって、保護された圧縮ビットストリームから判定された候補ビットから、検証データを抽出する検証データ抽出器９３０を備える。この具体例では、検証データは、メッセージ認証コード、例えば、図３のメッセージ認証コード算出器３４０によって生成されたメッセージ認証コードを含む。この具体例には、図８を用いて上述した候補コードの決定の変形例を適用してもよい。

【００３５】

また、認証エンジン９２０は、正規化器９４０を備え、この正規化器９４０は、解析器９１０から保護された圧縮ビットストリーム及び候補ビット選択データが供給され、受信ビットストリーム内の候補ビットの値を、図２の埋込装置の正規化器２３０で用いた値と同じ所定の値に設定し、これにより正規化されたビットストリームを生成する。正規化されたビットストリームは、メッセージ認証コード算出器９６０に渡され、メッセージ認証コード算出器９６０は、図３に示す装置で用いられた秘密鍵３５０と同じ秘密鍵９５０を用いて、正規化されたビットストリームからメッセージ認証コードを生成する。

【００３６】

そして、検証データ抽出器９３０によって抽出されたメッセージ認証コード及びメッセージ認証コード算出器９６０によって算出されたメッセージ認証コードは、比較モジュール９７０において比較される。比較モジュール９７０は、２つのメッセージ認証コードが一致した場合、保護された圧縮ビットストリームは、正規のものであると判定する。圧縮ビットストリームの信頼性を示す信号は、認証信号９８０として出力される。認証信号９

８０は、圧縮ビットストリームに対する又はこれに関する更なる処理をトリガし又は防止するために用いることができる。

【００３７】

ビデオ信号のビットストリーム内において、全てのブロックに検証データを挿入する必要はないが、保護すべきピクチャ又はグループオブピクチャには、検証データを挿入することが望ましい。ＭＰＥＧ－２ビットストリーム内で、特にＰフレーム及びＢフレームに関して、ビデオデータのそのフレームから選択される候補コードの数が不十分である場合もある。このような場合、候補コードの選択の基準を緩和し、これにより、許容できる画像の歪みのレベルを高めてもよく、或いは、単にそのフレームを保護しないようにしてもよい。したがって、幾つかの実施形態においては、例えば、ＧＯＰ全体等、複数のフレームに亘って正規化されたビットストリームのハッシュ値を算出してもよい。このような場合、幾つかの候補コードが検証エンジンによって特定され、ハッシュ値の生成のために正規化されたＧＯＰのＩフレームのみに安全なハッシュ値（secure hash value）を加えることができる。なお、安全なハッシュ値は、如何なる量の圧縮データについて生成してもよく、例えば、ＧＯＰの１つの領域又はフレーム、ＧＯＰ又はその他の部分等、圧縮符号化されたビットストリームの一部のみについて生成してもよい。このような実施形態では、データ復号処理装置の認証エンジンは、圧縮符号化データストリームの適切な部分から安全なハッシュを再生できる。

【００３８】

図１０Ａ及び図１０Ｂは、ＭＰＥＧ－２可変長コードスキームにおいて、コードの変化のレベルを許容範囲内に抑えながら変更できる可能な候補可変長コードを示すテーブルを示している。これらのテーブルは、いずれもＭＰＥＧ－２で用いられるが、図１０Ａのテーブルは、非イントラブロックＡＣ係数、及びある符号化条件の下で、イントラブロックＡＣ係数を符号化するために用いられる可変長コードを示している。一方、図１０Ｂのテーブルは、他の符号化条件の下で、イントラブロックＡＣ係数を符号化するために用いられる可変長コードを示している。

【００３９】

図１０Ａ及び図１０Ｂの符号化テーブルにおいて、可変長コードは、可変長コードによって表された値が正の値であるか負の値であるかを表す符号ビット「ｓ」で終わっている。例えば、正の値を０の符号ビットで表し、負の値を１の符号ビットで表してもよく、これを逆にしてもよい。ＭＰＥＧ－２では、コード値のレベルへの影響が最も小さいビットは、通常、符号ビットに先行するビットであり、多くの場合、このビット及びこのビットの直前の１又は複数のビットを候補ビットとみなすことができる。勿論、符号ビット自体を変更すると、コードの基本的な意味が変わってしまう。図１０Ａの符号化テーブル内の潜在的候補コードは、この符号化テーブルの領域１０１０、１０２０、１０３０、１０４０、１０５０内のコードである。図１０Ｂの符号化テーブル内の潜在的候補コードは、この符号化テーブルの領域１０６０、１０７０、１０８０、１０９０、１１００内のコードである。

【００４０】

図１０Ａ及び図１０Ｂの両方において、潜在的候補コードを含む領域の全ては、ランレングスが０又は１のいずれかであるコードに関連する。これらの領域の１つにおける各潜在的候補コードは、同じ領域内の、同じランレングスを有し、同じビット数からなり、１つ以上の最下位（非符号）ビットのみが違う他の少なくとも１つの可変長コードと対にすることができる。なお、ランレングスが２～３１の領域については、同じビット数から構成されるコードの対がなく、したがって、これらのコードのどれかを変更すると、規則に違反するコードが生成されるか、異なるランレングスに関連するコードが生成されてしまう。また、図１０Ａ及び図１０Ｂでは、エンドオブブロックコード及びエスケープコードが特定されており、これらはいずれも変更するべきではない。

【００４１】

これらの符号化テーブルは、適切な候補コードを作ることができる可変長コードを示し

ているが、これらのテーブルにおいてマークされた領域内のマークされたコードのある対が、他のコードの対より適切である場合がある。更に、コードの適切な評価は、特に、コードに適用された量子化の度合い等の他の因子に依存する。選択処理では、これらの因子を用いて、他の潜在的候補コードと相対的に潜在的候補コードを格付けし、格付けされたリストに基づいて、候補コードの最終的な選択を行ってもよい。このような方式では、量子化を考慮に入れない場合に所定の順序で格付けされる２つのコードの順位が、量子化を考慮に入れた場合に逆転することもあり得る。

【００４２】

殆どの固定長コードは、可変長コードとは異なり、通常、コードの１つ以上の最下位ビットを変更することによって、コードの基本的意味を変えないで妥当に変更できる。変更できない例外的なコードとしては、変更されると、規則に違反する値が生成されてしまう固定長コードがある。このようなコードの具体例としては、０のみが連続し、最後のビットが１で終わる固定長コードがある。この固定長コードの最下位ビットを１から０に変更することによってこの固定長コードを変更すると、０だけからなるコードが生成され、このようなコードは、ＭＰＥＧ－２コード体系内では、規則に違反するコードである。但し、殆どの場合、固定長コードの最下位ビットを所定の値に設定しても、例えば、値が１３から１４になる等、値は、最大で１しか変化しない。更に、固定長コードの２つの最下位ビットを所定の値に設定しても、例えば、値が１３から１６になる等、値は、最大で３しか変化しない。非常に大きい値を有する候補固定長コードについては、コードに適用された量子化のレベルに応じて、その固定長コードからより多数の候補ビットを選択してもよい。

【００４３】

本発明の更なる様々な側面及び特徴は、請求の範囲において定義されている。この請求の範囲から逸脱することなく、上述した実施の形態を様々に変更することができる。

【図面の簡単な説明】

【００４４】

【図１】ビデオ情報の記録及び符号化のための例示的な装置の構成を示す図である。

【図２】図１の装置と共に好適に用いられる検証データ埋込装置の構成を示す図である。

【図３】図１の装置と共に好適に用いられる検証データ埋込装置の変形例を示す図である。

【図４】圧縮ビットストリームの小さいセクションを示す図である。

【図５】検証ビットを保存するために選択した、図１のビットストリーム内の候補コードを示す図である。

【図６】図５のビットストリームを正規化するための準備を示す図である。

【図７】例えば、図１の装置によって生成されたビデオ情報を受信し、検証されたビデオ信号を復号する例示的な装置の構成を示す図である。

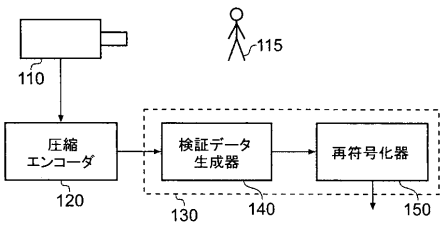
【図８】図７の装置と共に好適に用いられる認証装置の構成を示す図である。

【図９】図７の装置と共に好適に用いられる認証装置の変形例を示す図である。

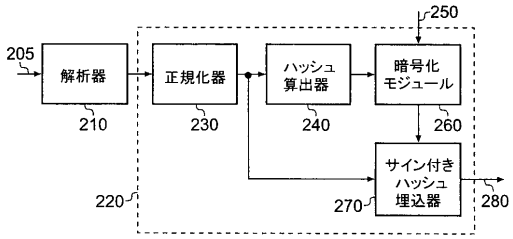
【図１０Ａ】検証ビットの保存に適する例示的な候補コードのテーブルを示す図である。

【図１０Ｂ】検証ビットの保存に適する例示的な候補コードのテーブルを示す図である。

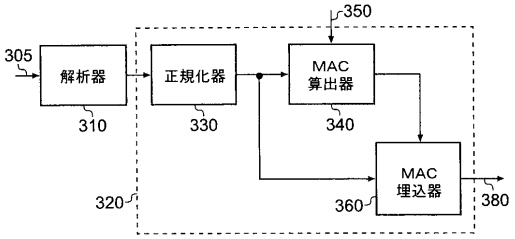
【図１】



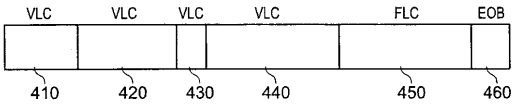
【図２】



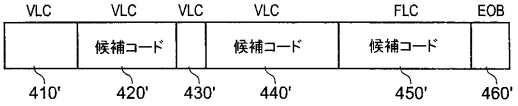
【図３】



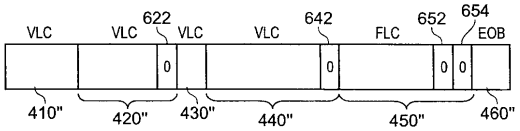
【図４】



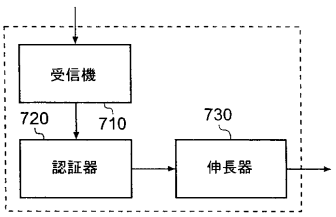
【図５】



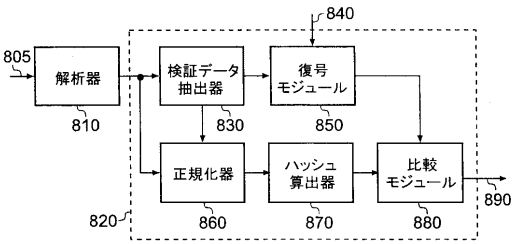
【図６】



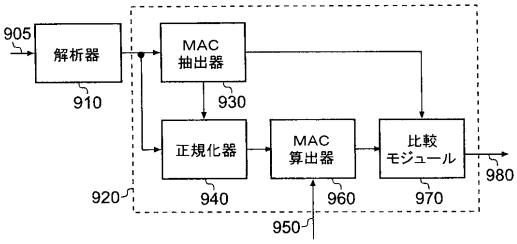
【図７】



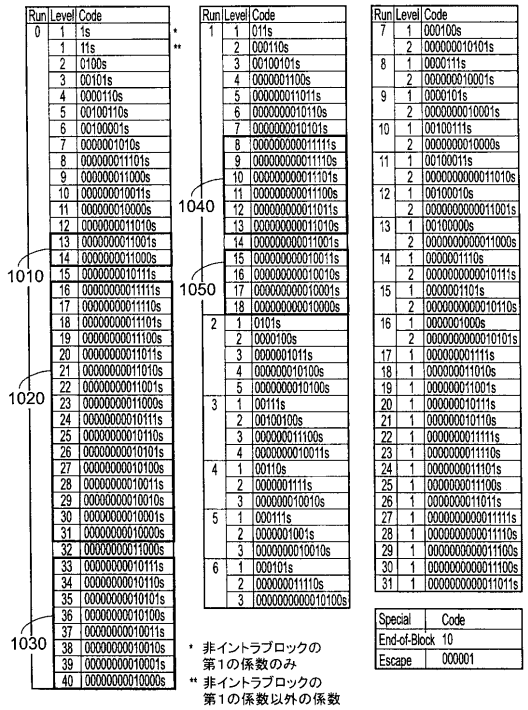
【図８】



【図９】



【図１０Ａ】



Run	Level	Code	
0	1	1s	*
	1	11s	**
	2	0100s	
	3	00101s	
	4	0000110s	
	5	00100110s	
	6	00100001s	
	7	0000001010s	
	8	0000000111101s	
	9	0000000011000s	
	10	0000000010011s	
	11	0000000010000s	
	12	0000000011010s	
	13	0000000011001s	
	14	0000000011000s	
	15	0000000010111s	
	16	0000000001111s	
	17	0000000011110s	
	18	0000000011101s	
	19	0000000011100s	
	20	0000000011011s	
	21	0000000011010s	
	22	0000000011001s	
	23	0000000011000s	
	24	0000000010111s	
	25	0000000010110s	
	26	0000000010101s	
	27	0000000010100s	
	28	0000000010011s	
	29	0000000010010s	
	30	0000000010001s	
	31	0000000010000s	
	32	0000000011000s	
	33	0000000010111s	
	34	0000000010110s	
	35	0000000010101s	
	36	0000000010100s	
	37	0000000010011s	
	38	0000000010010s	
	39	0000000010001s	
	40	0000000010000s	

Run	Level	Code
1	1	011s
	2	000110s
	3	00100101s
	4	0000001100s
	5	0000000111011s
	6	0000000010110s
	7	0000000010101s
	8	0000000000111111s
	9	0000000000111110s
	10	0000000000111101s
	11	0000000000111100s
	12	0000000000111011s
	13	0000000000111010s
	14	0000000000111001s
	15	0000000000100011s
	16	0000000000100010s
	17	0000000000100001s
	18	0000000000100000s
2	1	0101s
	2	0000100s
	3	0000001011s
	4	0000000100s
	5	0000000010100s
3	1	00111s
	2	00100100s
	3	0000001100s
	4	0000000010011s
4	1	00110s
	2	0000001111s
	3	0000001010s
5	1	000111s
	2	0000001001s
	3	0000000100100s
6	1	000101s
	2	0000001111 10s
	3	0000000000010100s

Run	Level	Code
7	1	000100s
	2	0000000101101s
8	1	0000111s
	2	0000000100001s
9	1	0000101s
	2	0000000010001s
10	1	0010011 1s
	2	0000000010000s
11	1	0010001 1s
	2	0000000000011010s
12	1	0010001 0s
	2	0000000000011001s
13	1	0010000s
	2	0000000000011000s
14	1	0000001110s
	2	0000000000010111s
15	1	0000001101s
	2	0000000000010110s
16	1	0000001000s
	2	0000000000010101s
17	1	0000000111 1s
18	1	0000000110 10s
19	1	0000000110 01s
20	1	0000000101 11s
21	1	0000000101 10s
22	1	0000000111 111s
23	1	0000000111 110s
24	1	0000000111 101s
25	1	0000000011 100s
26	1	0000000111 011s
27	1	0000000000011111s
28	1	0000000000011110s
29	1	0000000000011100s
30	1	0000000000011100s
31	1	0000000000011011s

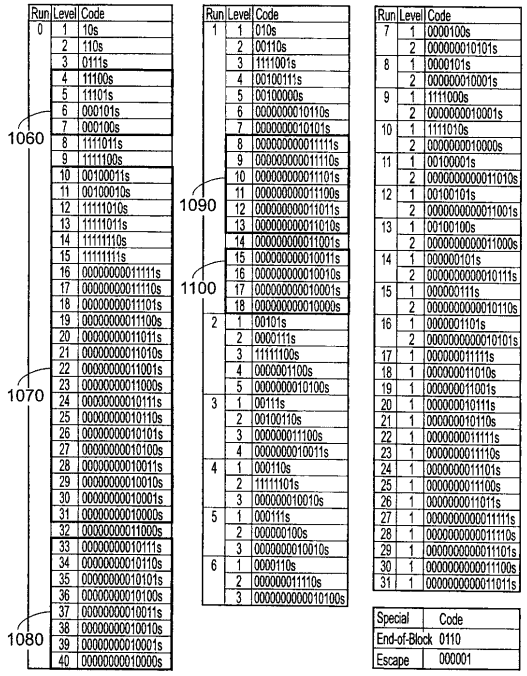
Special	Code
End-of-Block	10
Escape	000001

* 非イントラブロックの第１の係数のみ

** 非イントラブロックの第１の係数以外の係数

符号ビットは、正の場合０、負の場合１

【図１０Ｂ】



Run	Level	Code
0	1	10s
	2	110s
	3	0111s
	4	11100s
	5	11101s
	6	000101s
	7	000100s
	8	1111011s
	9	1111100s
	10	00100011s
	11	00100010s
	12	11111010s
	13	11111011s
	14	11111110s
	15	11111111s
	16	0000000011111s
	17	0000000011110s
	18	0000000011101s
	19	0000000011100s
	20	0000000011011s
	21	0000000011010s
	22	0000000011001s
	23	0000000011000s
	24	0000000010111s
	25	0000000010110s
	26	0000000010101s
	27	0000000010100s
	28	0000000010011s
	29	0000000010010s
	30	0000000010001s
	31	0000000010000s
	32	0000000011000s
	33	0000000010111s
	34	0000000010110s
	35	0000000010101s
	36	0000000010100s
	37	0000000010011s
	38	0000000010010s
	39	0000000010001s
	40	0000000010000s

Run	Level	Code
1	1	010s
	2	00110s
	3	1111001s
	4	00100111s
	5	00100000s
	6	0000000010110s
	7	0000000010101s
	8	0000000000111111s
	9	0000000000111110s
	10	0000000000111101s
	11	0000000000111100s
	12	0000000000111011s
	13	0000000000111010s
	14	0000000000111001s
	15	0000000000100011s
	16	0000000000100010s
	17	0000000000100001s
	18	0000000000100000s
2	1	00101s
	2	0000111s
	3	11111100s
	4	0000001100s
	5	0000000010100s
3	1	00111s
	2	00100110s
	3	0000000111 00s
	4	0000000010011s
4	1	000110s
	2	11111101s
	3	0000000010010s
5	1	000111s
	2	0000000100s
	3	0000000010010s
6	1	0000110s
	2	0000000111 10s
	3	0000000000010100s

Run	Level	Code
7	1	0000100s
	2	0000000101101s
8	1	0000101s
	2	0000000100001s
9	1	1111000s
	2	0000000010001s
10	1	1111010s
	2	0000000010000s
11	1	00100001s
	2	0000000000011010s
12	1	00100101s
	2	0000000000011001s
13	1	00100100s
	2	0000000000011000s
14	1	0000001 01s
	2	0000000000010111s
15	1	0000001 11s
	2	0000000000010110s
16	1	0000001101s
	2	0000000000010101s
17	1	0000000111 11s
18	1	0000000110 10s
19	1	0000000110 01s
20	1	0000000101 11s
21	1	0000000101 10s
22	1	0000000111 111s
23	1	0000000111 110s
24	1	0000000111 101s
25	1	0000000011 100s
26	1	0000000111 011s
27	1	0000000000011111s
28	1	0000000000011110s
29	1	0000000000011101s
30	1	0000000000011100s
31	1	0000000000011011s

Special	Code
End-of-Block	0110
Escape	000001

符号ビットは、正の場合０、負の場合１

フロントページの続き

(72)発明者　テイラー　アンドリュー　ロバート
イギリス国　ＫＴ１３　ＯＸＷ　サリー、　ウエィブリッジ　ブルックランズ、　ザ　ハイツ　（番地無し）　ソニー　ユナイテッド　キングダム　リミテッド内

Ｆターム(参考)　5C059 KK01 KK43 MA00 MA23 MC11 ME01 RB08 RB09 RC35 SS20
TB03 TB04
5C164 MB31S PA04 PA09 SA31S SB03P UA21S UB03P
5J104 AA01 AA09 AA12 AA32 JA21 LA03 LA06 NA02 NA12 NA27
NA37

【外国語明細書】
2006191534000001.pdf
2006191534000002.pdf
2006191534000003.pdf
2006191534000004.pdf