

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6800991号
(P6800991)

(45) 発行日 令和2年12月16日 (2020. 12. 16)

(24) 登録日 令和2年11月27日 (2020. 11. 27)

(51) Int. Cl. F I
H 0 4 N 19/91 (2014. 01) H 0 4 N 19/91

請求項の数 8 (全 35 頁)

(21) 出願番号	特願2018-546866 (P2018-546866)	(73) 特許権者	595020643
(86) (22) 出願日	平成29年3月6日 (2017. 3. 6)		クォアルコム・インコーポレイテッド
(65) 公表番号	特表2019-511865 (P2019-511865A)		Q U A L C O M M I N C O R P O R A T E D
(43) 公表日	平成31年4月25日 (2019. 4. 25)		アメリカ合衆国、カリフォルニア州 9 2
(86) 国際出願番号	PCT/US2017/020897		1 2 1 - 1 7 1 4、サン・ディエゴ、モア
(87) 国際公開番号	W02017/155864		ハウス・ドライブ 5 7 7 5
(87) 国際公開日	平成29年9月14日 (2017. 9. 14)	(74) 代理人	100108855
審査請求日	令和2年2月10日 (2020. 2. 10)		弁理士 蔵田 昌俊
(31) 優先権主張番号	62/305, 380	(74) 代理人	100109830
(32) 優先日	平成28年3月8日 (2016. 3. 8)		弁理士 福原 淑弘
(33) 優先権主張国・地域又は機関	米国 (US)	(74) 代理人	100158805
(31) 優先権主張番号	62/415, 999		弁理士 井関 守三
(32) 優先日	平成28年11月1日 (2016. 11. 1)	(74) 代理人	100112807
(33) 優先権主張国・地域又は機関	米国 (US)		弁理士 岡田 貴志

最終頁に続く

(54) 【発明の名称】 ディスプレーストリーム圧縮のためのベクトルベースエントロピーコード化のための装置および方法

(57) 【特許請求の範囲】

【請求項 1】

複数のビデオサンプルを有するビデオ情報をコード化するための装置であって、

複数のグループに配列される複数のビデオサンプルを記憶するように構成されたメモリと、各ビデオサンプルは、ビットサイズ B を有し、前記複数のグループのそれぞれのグループは、グループサイズ K を有する、

ハードウェアプロセッサと、前記ハードウェアプロセッサは、可変長コード化 (V L C) スキームを使用して、それぞれのベクトルベースコードを形成するために前記複数のグループの各グループ中の前記複数の前記ビデオサンプルのそれぞれの長さ K のサンプルベクトルをコード化するように構成され、前記それぞれのベクトルベースコードは、少なくとも、前記ビットサイズ B または前記グループサイズ K のうちの少なくとも 1 つに対応するルックアップテーブルの複数のタイプのうちからルックアップテーブルのタイプを識別する第 1 の部分と、前記それぞれのグループ中の前記複数のビデオサンプルを表す第 2 の部分とを備える、

前記ハードウェアプロセッサは、前記複数のグループの最初のいくつかのグループ中に符号絶対値表現において前記複数のビデオサンプルをコード化するようにさらに構成され、

前記ハードウェアプロセッサは、前記複数のグループの最後のグループ中に 2 の補数表現において前記複数のビデオサンプルをコード化するようにさらに構成され、

前記ハードウェアプロセッサは、前記それぞれのベクトルベースコードを出力するよ

10

20

うにさらに構成され、

前記それぞれのベクトルベースコードを形成するために前記それぞれの長さKのサンプルベクトルをコード化するため、前記ハードウェアプロセッサは、

前記それぞれの長さKのサンプルベクトルをインデックス値に変換することと、

前記インデックス値を前記タイプのルックアップテーブルを使用してコード番号にマッピングすることと、

前記タイプのルックアップテーブルを示すように前記それぞれのベクトルベースコードの前記第1の部分をコード化することと、

前記それぞれのベクトルベースコードの前記第2の部分を形成するために前記コード番号をコード化することと、

を行うように構成される、

を備える装置。

【請求項2】

前記ハードウェアプロセッサは、

可変長コード化(VLC)スキームを使用して、それぞれのコード番号を形成するために前記それぞれのベクトルベースコードを復号することと、

前記それぞれのベクトルベースコードの前記第1の部分によって識別される前記それぞれのタイプのルックアップテーブルを使用して、前記それぞれのコード番号をそれぞれのインデックス値にマッピングすることと、

前記それぞれのインデックス値を前記それぞれのグループの前記複数のビデオサンプルに変換することと

を行うようにさらに構成される、請求項1に記載の装置。

【請求項3】

前記ハードウェアプロセッサは、前記ビットサイズBが閾値より小さいとき、前記それぞれのベクトルベースコードを供給するように構成される、請求項1に記載の装置。

【請求項4】

前記タイプのルックアップテーブルは、可能なKサイズのグループのセットのうちの前記それぞれのグループの生起確率に少なくとも部分的に基づく、請求項1に記載の装置。

【請求項5】

複数のビデオサンプルを有するビデオ情報をコード化するための方法であって、

複数のグループに配列される複数のビデオサンプルを記憶することと、各ビデオサンプルは、ビットサイズBを有し、前記複数のグループのそれぞれのグループは、グループサイズKを有する、

可変長コード化(VLC)スキームを使用して、それぞれのベクトルベースコードを形成するために前記複数のグループの各グループ中の前記複数の前記ビデオサンプルのそれぞれの長さKのサンプルベクトルをコード化することと、前記それぞれのベクトルベースコードは、少なくとも、前記ビットサイズBまたは前記グループサイズKのうちの少なくとも1つに対応するルックアップテーブルの複数のタイプの中からルックアップテーブルのタイプを識別する第1の部分と、前記それぞれのグループ中の前記複数のビデオサンプルを表す第2の部分とを備える、

ここにおいて、前記コード化することは、

前記複数のグループの最初のいくつかのグループ中に符号絶対値表現において前記複数のビデオサンプルをコード化することと、

前記複数のグループの最後のグループ中に2の補数表現において前記複数のビデオサンプルをコード化することと、を含み、

ここにおいて、前記それぞれのベクトルベースコードを形成するために前記それぞれの長さKのサンプルベクトルをコード化することは、

前記それぞれの長さKのサンプルベクトルをインデックス値に変換することと、

前記インデックス値を前記タイプのルックアップテーブルを使用してコード番号にマッピングすることと、

10

20

30

40

50

前記タイプのルックアップテーブルを示すように前記それぞれのベクトルベースコードの前記第 1 の部分をコード化することと、

前記それぞれのベクトルベースコードの前記第 2 の部分を形成するために前記コード番号をコード化することと、を備える、

を備える方法。

【請求項 6】

可変長コード化 (VLC) スキームを使用して、それぞれのコード番号を形成するために前記それぞれのベクトルベースコードを復号することと、

前記それぞれのベクトルベースコードの前記第 1 の部分によって識別される前記それぞれのタイプのルックアップテーブルを使用して、前記それぞれのコード番号をそれぞれのインデックス値にマッピングすることと、

前記それぞれのインデックス値を前記それぞれのグループの前記複数のビデオサンプルに変換することと

をさらに備える、請求項 5 に記載の方法。

【請求項 7】

前記ビットサイズ B が閾値より小さいとき、前記それぞれのベクトルベースコードを供給することをさらに備える、請求項 5 に記載の方法。

【請求項 8】

前記タイプのルックアップテーブルは、可能な K サイズのグループのセットのうちの前記それぞれのグループの生起確率に少なくとも部分的に基づく、請求項 5 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

[0001]本開示は、ビデオコード化および圧縮の分野に関し、特に、ディスプレイストリーム圧縮 (DSC) のような、ディスプレイリンクを介した送信のためのビデオ圧縮に関する。

【背景技術】

【0002】

[0002]デジタルビデオ能力は、デジタルテレビ、携帯情報端末 (PDA)、ラップトップコンピュータ、デスクトップモニタ、デジタルカメラ、デジタル記録デバイス、デジタルメディアプレーヤ、ビデオゲーミングデバイス、ビデオゲームコンソール、セルラまたは衛星無線電話、ビデオ電話会議デバイス、等を含む、広範囲のディスプレイに組み込まれることができる。ディスプレイリンクは、ディスプレイを好適なソースデバイスに接続するために使用される。ディスプレイリンクの帯域幅要件は、ディスプレイの解像度に比例し、ゆえに、高解像度ディスプレイは、広帯域幅のディスプレイリンクを必要とする。いくつかのディスプレイリンクは、高解像度ディスプレイをサポートする帯域幅を有さない。ビデオ圧縮を使用してこの帯域幅要件を低減することで、より低い帯域幅のディスプレイリンクが、高解像度ディスプレイにデジタルビデオを提供するのに使用されることができる。

【0003】

[0003]画素データに対して画像圧縮を利用しようとした。しかしながら、そのようなスキームは、時に、視覚的にロスレスでないかまたは従来のディスプレイデバイスでインプリメントするのが困難かつ高価であり得る。

【0004】

[0004]VESA (Video Electronics Standards Association) は、ディスプレイリンクビデオ圧縮のための規格として、ディスプレイストリーム圧縮 (DSC) を開発してきた。DSC のようなディスプレイリンクビデオ圧縮技法は、とりわけ、視覚的にロスレスなピクチャ品質 (例えば、圧縮がアクティブであることをユーザに悟られないような品質レベルを有するピクチャ) を提供するべきである。ディスプレイリンクビデオ圧縮技法はまた、従来のハードウェアでリアルタイムにインプリメントするのが容易かつ高価でない

10

20

30

40

50

スキームを提供するべきである。

【発明の概要】

【0005】

[0005]本開示のシステム、方法、およびデバイスは各々、いくつかの革新的な態様を有し、それらはいずれも、本明細書で開示される望ましい属性を単独で担うものではない。

【0006】

[0006]一態様では、複数のビデオサンプルを有するビデオ情報をコード化するための方法および装置が開示される。ビデオサンプルは、単一のクロックサイクル内での送信のためにグループへと区分化され、ここにおいて、サンプルは、ビット長 B に関連付けられており、グループは、グループサイズ K を有する。サンプルグループは、コード番号にマッピングされ、このマッピングを実行するために使用されるルックアップテーブルのタイプを識別する第1の部分と、このグループのサンプルを表す第2の部分とを備えるベクトルベースコードを形成するためにコード化される。ルックアップテーブルは、異なるサンプルグループの生起確率に基づいて構築され得る。加えて、異なるタイプのルックアップテーブルが、異なる B 値および K 値に対して使用され得る。

【図面の簡単な説明】

【0007】

【図1A】[0007]図1Aは、本開示で説明される態様に係る技法を利用し得る例示的なビデオ符号化および復号システムを例示するブロック図である。

【図1B】[0008]図1Bは、本開示で説明される態様に係る技法を実行し得る別の例となるビデオ符号化および復号システムを例示するブロック図である。

【図2A】[0009]図2Aは、本開示で説明される態様に係る技法をインプリメントし得るビデオエンコーダの例を例示するブロック図である。

【図2B】[0010]図2Bは、本開示で説明される態様に係る技法をインプリメントし得るビデオデコーダの例を例示するブロック図である。

【図3】[0011]図3は、量子化パラメータ (Q_P) 調整値を決定することへの例示的なアプローチを示す。

【図4】[0012]図4は、 $K = 4$ 個のサンプルを有するベクトルのための $DSU - VLC$ 構造を示す。

【図5】[0013]図5は、4個のサンプルの例示的なグループのための $DSU - VLC$ コードを例示する。

【図6】[0014]図6は、均一グループ化の例を例示し、ここで、 2×8 ブロックは、4つのグループへと区分化される。

【図7】[0015]図7は、ベクトルベース EC プロセスの例示的なブロック図を例示する。

【図8】[0016]図8は、 $K = 4$ のサンプルを含むグループをコード化するために使用され得るベクトルベース EC の構造を示し、ここで、使用される LUT のタイプは、明示的にシグナリングされる。

【図9】[0017]図9は、組み合わせられた LUT タイプ信号および VLC コードプレフィックスを備えるコードを例示する。

【図10A】[0018]図10Aは、符号絶対値表現 (sign-magnitude representation) を使用して複数のサンプルベクトルをエントロピーコード化する例を例示する。

【図10B】図10Bは、符号絶対値表現を使用して複数のサンプルベクトルをエントロピーコード化する例を例示する。

【図11】[0019]図11は、符号絶対値を使用してコード化されたサンプルベクトルをパースおよび復号するために必要とされ得る多数の (a number of) クロックサイクルを例示する例示的なグラフを例示する。

【図12】[0020]図12は、複数のサンプルベクトルのハイブリッドエントロピーコード化の例を例示する。

【図13】[0021]図13は、ハイブリッドコード化スキームを使用してコード化されたサンプルベクトルをパースおよび復号するために必要とされ得る多数のクロックサイクルを

10

20

30

40

50

例示する例示的なグラフを例示する。

【図 1 4 A】[0022]図 1 4 A は、サンプルベクトルデータを符号化 / 復号するためのプロセスのフローチャートを例示する。

【図 1 4 B】図 1 4 B は、サンプルベクトルデータを符号化 / 復号するためのプロセスのフローチャートを例示する。

【図 1 5】[0023]図 1 5 は、ハイブリッドコード化を使用してサンプルベクトルデータを符号化するための例示的なプロセスのフローチャートを例示する。

【発明の詳細な説明】

【 0 0 0 8 】

[0024]一般に、本開示は、ディスプレイストリーム圧縮 (DSC) のようなビデオ圧縮技法を改善する技法に関する。より具体的には、本開示は、ブロック値をサンプルベクトルへと区分化することで、1クロックサイクルあたりに複数のサンプル値を送信することと、サンプルベクトルの生起確率に少なくとも部分的に基づいてサンプルベクトルをコード化することとを行うためのシステムおよび方法に関する。

【 0 0 0 9 】

[0025]本明細書では特定の実施形態は、DSC規格のコンテキストで説明されているが、当業者であれば、本明細書で開示されるシステムおよび方法が、任意の適切なビデオコード化規格に適用可能であり得ることは認識するであろう。例えば、本明細書で開示される実施形態は、国際電気通信連合 (ITU) 電気通信標準化部門 (ITU-T) H.261、国際標準化機構 / 国際電気標準会議 (ISO/IEC) MPEG-1 (Moving Picture Experts Group-1) ビジュアル、ITU-T H.262 すなわち ISO/IEC MPEG-2 ビジュアル、ITU-T H.263、ISO/IEC MPEG-4 ビジュアル、(ISO/IEC MPEG-4 AVCとしても知られている) ITU-T H.264、高効率ビデオコーディング (HEVC) といった規格およびそのような規格に対する任意の拡張のうちの1つまたは複数に適用可能であり得る。本明細書で説明される技法は、固定ビットレート (CBR) バッファモデルを組み込む規格に特に適用可能であり得る。また、本開示で説明される技法は、将来開発される規格の一部になり得る。換言すると、本開示で説明される技法は、前に開発されたビデオコード化規格、現在開発中のビデオコード化規格、および今度のビデオコード化規格に適用可能であり得る。

【 0 0 1 0 】

[0026]本開示の概念は、実質上視覚的にロスレスなパフォーマンスで様々なタイプのコンテンツを符号化 / 復号することを目的としているいくつかの要素および / またはモードを含むコーデック (例えば、DSC) に統合され得るかその一部であり得る。本開示は、単一のクロックサイクル内での送信のためにグループへとビデオサンプルを区分化するためのシステムおよび方法を提供し、ここにおいて、サンプルは、ビット長 B に関連付けられており、グループは、グループサイズ K を有する。サンプルグループは、コード番号にマッピングされ得、このマッピングを実行するために使用されるルックアップテーブルのタイプを識別する第1の部分と、このグループのサンプルを表す第2の部分とを備えるベクトルベースコードを形成するためにコード化される。ルックアップテーブルは、異なるサンプルグループの生起確率に基づいて構築され得る。加えて、異なるタイプのルックアップテーブルが、異なる B 値および K 値に対して使用され得、これにより、より効率的なコード化が可能になる。

【 0 0 1 1 】

ビデオコード化規格

[0027]ビデオ画像、TV画像、静止画像、またはビデオレコードもしくはコンピュータによって生成された画像のようなデジタル画像は、水平線および垂直線に配列された画素またはサンプルを含み得る。単一の画像中の画素数は典型的に数万である。各画素は典型的に、輝度およびクロミナンス情報を含む。圧縮なしの場合、画像エンコードから画像デコードに伝達されるべき膨大な量の情報は、リアルタイムの画像送信を実現困難なものとするであろう。送信されることとなる情報の総量 (the amount) を低減するために、JPE

10

20

30

40

50

G、MPEG、およびH.263規格のような多数の異なる圧縮方法が開発されてきた。

【0012】

[0028]ビデオコード化規格には、ITU-T H.261、ISO/IEC MPEG-1 ビジュアル、ITU-T H.262 すなわち ISO/IEC MPEG-2 ビジュアル、ITU-T H.263、ISO/IEC MPEG-4 ビジュアル、(ISO/IEC MPEG-4 AVCとしても知られている) ITU-T H.264、およびHEVCと、それらの規格の拡張とが含まれる。

【0013】

[0029]加えて、ビデオコード化規格、すなわちDSC、は、VESAによって開発されてきた。DSC規格は、ディスプレイリンクを介した送信のためにビデオを圧縮することができるビデオ圧縮規格である。ディスプレイの解像度が増加するにつれ、このディスプレイを駆動するのに必要とされるビデオデータの帯域幅が相応して増加する。いくつかのディスプレイリンクは、そのような解像度について、ビデオデータのすべてをディスプレイに送信するための帯域幅を有さない可能性がある。したがって、DSC規格は、ディスプレイリンクを介した相互利用可能で視覚的にロスレスな圧縮のための圧縮規格を規定する。

10

【0014】

[0030]DSC規格は、H.264およびHEVCのような他のビデオコード化規格とは異なる。DSCは、イントラフレーム圧縮を含むがインターフレーム圧縮は含まず、つまり、ビデオデータをコード化する際に時間的情報がDSC規格によって使用されないであろうことを意味する。対照的に、他のビデオコード化規格は、それらのビデオコード化技法においてインターフレーム圧縮を用い得る。

20

【0015】

ビデオコード化システム

[0031]新規なシステム、装置、および方法の様々な態様が、添付の図面を参照して以下でより十分に説明される。しかしながら、本開示は、多くの異なる形式で具現化され得、本開示全体にわたって提示される任意の特定の構造または機能に限定されるものとして解釈されるべきではない。むしろ、これらの態様は、本開示が徹底的かつ完全となり得、本開示の範囲を当業者に十分に伝達し得るように提供される。本明細書における教示に基づき、当業者は、本開示の範囲が、本明細書で開示される新規なシステム、装置、および方法の任意の態様を、本発明の任意の他の態様から独立してインプリメントされようとそれと組み合わせられようと、カバーするように意図されることは認識するはずである。例えば、本明細書で示される任意の数の態様を使用して、装置がインプリメントされ得るか、方法が実施され得る。加えて、本開示の範囲は、本明細書で示される開示の様々な態様に加えて、または、それ以外に、他の構造、機能性、または構造と機能性を使用して実施されるこのような装置または方法をカバーするように意図されている。本明細書で開示される任意の態様が、請求項の1つまたは複数の要素によって具現化され得ることは理解されるべきである。

30

【0016】

[0032]特定の態様が本明細書で説明されるが、これらの態様の多くの変形および並替へは、本開示の範囲内である。好ましい態様のいくつかの利益および利点が述べられるが、本開示の範囲は、特定の利益、用途、または目的に限定されるように意図されるものではない。むしろ、本開示の態様は、いくつかは、図においておよび好ましい態様の以下の説明において例として例示される異なるワイヤレス技術、システム構成、ネットワーク、および送信プロトコルに広く適用可能であるように意図される。詳細な説明および図面は、限定するものというよりはむしろ、本開示の単なる例にすぎず、本開示の範囲は、添付の特許請求の範囲およびそれらの同等物によって定義されている。

40

【0017】

[0033]添付の図面は例を例示する。添付の図面において参照番号で示される要素は、以下の説明において同様の参照番号で示される要素に対応する。本開示では、順序を示す言

50

葉（例えば、「第１の」、「第２の」、「第３の」、等）で始まる名前を有する要素は、必ずしも、それら要素が特定の順序を持つことを意味するものではない。むしろ、そのような順序を示す言葉は、単に、同じまたは同様のタイプの異なる要素を指すために使用される。

【 0 0 1 8 】

[0034] 図 1 A は、本開示で説明される態様に係る技法を利用し得る例示的なビデオコード化システム 1 0 を例示するブロック図である。本明細書で説明される使用される場合、「ビデオコード」または「コード」という用語は、概して、ビデオエンコードおよびビデオデコードの両方を指す。本開示では、「ビデオコード化」または「コード化」という用語は、概して、ビデオ符号化およびビデオ復号を指し得る。ビデオエンコードおよびビデオデコードに加えて、本願で説明される態様は、トランスコード（例えば、ビットストリームを復号し、別のビットストリームを再符号化することができるデバイス）およびミドルボックス（例えば、ビットストリームを修正、変換、および／または他の方法で操作することができるデバイス）のような他の関連デバイスに拡張され得る。

10

【 0 0 1 9 】

[0035] 図 1 A に示されるように、ビデオコード化システム 1 0 は、宛先デバイス 1 4 によって後に復号されることとなる符号化済みビデオデータを生成するソースデバイス 1 2 を含む。図 1 A の例では、ソースデバイス 1 2 および宛先デバイス 1 4 は、別個のデバイスを構成する。しかしながら、図 1 B の例で示されるように、ソースデバイス 1 2 および宛先デバイス 1 4 が同じデバイス上にあり得ることまたは同じデバイスの一部であり得ることに留意されたい。

20

【 0 0 2 0 】

[0036] 図 1 A を再度参照すると、ソースデバイス 1 2 および宛先デバイス 1 4 は、それぞれ、デスクトップコンピュータ、ノートブック（例えば、ラップトップ）コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンのような電話ハンドセット、いわゆる「スマート」パッド、テレビ、カメラ、ディスプレイデバイス、デジタルメディアプレーヤ、ビデオゲーミングコンソール、車載コンピュータ、ビデオストリーミングデバイス、エンティティ内に配置されている、取り込まれている、または消費されること（consumed）ができるアイウェアおよび／またはウェアラブルなコンピュータ、デバイス、または装置のようなエンティティ（例えば、人間、動物、および／または別の被制御デバイス）によってウェアラブルな（または、それに取り外し可能に取り付けられている）デバイス、および／または同様のものを含む広範囲のデバイスのうちの任意のものを備え得る。様々な実施形態では、ソースデバイス 1 2 および宛先デバイス 1 4 は、ワイヤレス通信に対応し得る。

30

【 0 0 2 1 】

[0037] 宛先デバイス 1 4 は、復号されることとなる符号化済みビデオデータを、リンク 1 6 を介して受け取り得る。リンク 1 6 は、符号化済みビデオデータをソースデバイス 1 2 から宛先デバイス 1 4 に移動することができる任意のタイプの媒体またはデバイスを備え得る。図 1 A の例では、リンク 1 6 は、ソースデバイス 1 2 が符号化済みビデオデータを宛先デバイス 1 4 に、リアルタイムに送信することを可能にする通信媒体を備え得る。符号化済みビデオデータは、ワイヤレス通信プロトコルのような通信規格にしたがって変調され、宛先デバイス 1 4 に送信され得る。通信媒体は、無線周波数（RF）スペクトルまたは 1 つまたは複数の物理伝送線のような任意のワイヤレスまたはワイヤード通信媒体を備え得る。通信媒体は、ローカルエリアネットワーク、広域ネットワーク、またはインターネットのようなグローバルネットワークといった、パケットベースネットワークの一部を形成し得る。通信媒体は、ルータ、スイッチ、基地局、またはソースデバイス 1 2 から宛先デバイス 1 4 への通信を容易にするのに有益であり得る任意の他の機器を含み得る。

40

【 0 0 2 2 】

[0038] 図 1 A の例では、ソースデバイス 1 2 は、ビデオソース 1 8 と、ビデオエンコー

50

ダ 20 と、出力インターフェース 22 とを含む。いくつかのケースでは、出力インターフェース 22 は、変調器 / 復調器 (モデム) および / または送信機を含み得る。ソースデバイス 12 では、ビデオソース 18 は、例えば、ビデオカメラのようなビデオキャプチャデバイス、前に撮られたビデオを含むビデオアーカイブ、ビデオコンテンツプロバイダからビデオを受け取るためのビデオフィールドインターフェース、および / または、ソースビデオとしてコンピュータグラフィックデータを生成するためのコンピュータグラフィックシステムのようなソースまたはそのようなソースの組み合わせを含み得る。一例として、ビデオソース 18 がビデオカメラである場合、ソースデバイス 12 および宛先デバイス 14 は、図 1 B の例で例示されるような、いわゆる「カメラ付電話」または「ビデオ電話」を形成し得る。しかしながら、本開示で説明される技法は、概して、ビデオコード化に適用可能であり得、ワイヤレスおよび / またはワイヤードアプリケーションに適用され得る。

10

【 0023 】

[0039]撮られた、事前に撮られた (pre-captured)、またはコンピュータにより生成されたビデオは、ビデオエンコーダ 20 によって符号化され得る。符号化済みビデオデータは、ソースデバイス 12 の出力インターフェース 22 を介して、宛先デバイス 14 に直接送信され得る。符号化済みビデオデータはまた (または、代替的に)、復号および / または再生のための宛先デバイス 14 または他のデバイスによる後のアクセスのために記憶デバイス 31 上に記憶され得る。図 1 A および 1 B に例示されるビデオエンコーダ 20 は、図 2 A に例示されるビデオエンコーダ 20 または本明細書で説明される任意の他のビデオエンコーダを備え得る。

20

【 0024 】

[0040]図 1 A の例では、宛先デバイス 14 は、入力インターフェース 28、ビデオデコーダ 30、およびディスプレイデバイス 32 を含む。いくつかのケースでは、入力インターフェース 28 は、受信機および / またはモデムを含み得る。宛先デバイス 14 の入力インターフェース 28 は、リンク 16 を介しておよび / または記憶デバイス 31 から、符号化済みビデオデータを受け取り得る。リンク 16 を介して通信される、または記憶デバイス 31 上で提供される、符号化済みビデオデータは、ビデオデータを復号する際に、ビデオデコーダ 30 のようなビデオデコーダによる使用のためにビデオエンコーダ 20 によって生成される様々なシンタックス要素を含み得る。そのようなシンタックス要素は、通信媒体上で送信されるか、記憶媒体に記憶されるか、ファイルサーバに記憶される符号化済みビデオデータに含まれ得る。図 1 A および 1 B に例示されるビデオデコーダ 30 は、図 2 B に例示されるビデオデコーダ 30 または本明細書で説明される任意の他のビデオデコーダを備え得る。

30

【 0025 】

[0041]ディスプレイデバイス 32 は、宛先デバイス 14 と統合され得るか、それに外付けであり得る。いくつかの例では、宛先デバイス 14 は、統合されたディスプレイデバイスを含み得、同じく、外部のディスプレイデバイスとインターフェース接続するように構成され得る。他の例では、宛先デバイス 14 は、ディスプレイデバイスであり得る。一般に、ディスプレイデバイス 32 は、復号されたビデオデータをユーザに表示し、液晶ディスプレイ (LCD)、プラズマディスプレイ、有機発光ダイオード (OLED) ディスプレイ、または別のタイプのディスプレイデバイスのような、様々なディスプレイデバイスのうちの任意のものを備え得る。

40

【 0026 】

[0042]関連する態様では、図 1 B は、例示的なビデオコード化システム 10 ' を示し、ここにおいて、ソースデバイス 12 および宛先デバイス 14 は、デバイス 11 上にあるか、その一部である。デバイス 11 は、「スマート」フォンのような電話ハンドセットまたは同様のものであり得る。デバイス 11 は、ソースデバイス 12 および宛先デバイス 14 と動作可能に通信状態にある (オプションで存在する) プロセッサ / コントローラデバイス 13 を含み得る。図 1 B のビデオコード化システム 10 ' およびその構成要素は、他の点では (otherwise)、図 1 A のビデオコード化システム 10 およびその構成要素に類似

50

し得る。

【0027】

[0043]ビデオエンコーダ20およびビデオデコーダ30は、DSCのような、ビデオ圧縮規格にしたがって動作し得る。代替的に、ビデオエンコーダ20およびビデオデコーダ30は、別名、MPEG-4, Part 10, AVCと呼ばれるITU-T H.264規格、HEVC、またはそのような規格の拡張のような、他の専有または工業規格にしたがって動作し得る。しかしながら、本開示の技法は、いずれの特定のコード化規格にも限定されない。ビデオ圧縮規格の他の例には、MPEG-2およびITU-T H.263が含まれる。

【0028】

[0044]図1Aおよび1Bの例では示されていないが、ビデオエンコーダ20およびビデオデコーダ30は、各々、オーディオエンコーダおよびデコーダと統合され得、共通のデータストリームまたは別個のデータストリームにおけるオーディオおよびビデオの両方の符号化に対処するために、好適なMUX-DEMUXユニットまたは他のハードウェアおよびソフトウェアを含み得る。適用可能な場合、いくつかの例では、MUX-DEMUXユニットは、ITU-T H.223マルチプレクサプロトコル、またはユーザデータグラムプロトコル(UDP)のような他のプロトコルに準拠し得る。

【0029】

[0045]ビデオエンコーダ20およびビデオデコーダ30は、各々、1つまたは複数のマイクロプロセッサ、デジタルシグナルプロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、ディスクリート論理、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の組み合わせのような、様々な適切なエンコーダ回路の任意のものとしてインプリメントされ得る。本技法が部分的にソフトウェアでインプリメントされる場合、デバイスは、このソフトウェアのための命令を、適切で非一時的なコンピュータ読取可能な媒体に記憶し、1つまたは複数のプロセッサを使用してハードウェアで命令を実行して、本開示の技法を実行し得る。ビデオエンコーダ20およびビデオデコーダ30の各々は、1つまたは複数のエンコーダまたはデコーダに含まれ得、それらのどちらも、それぞれのデバイスにおける複合エンコーダ/デコーダの一部として統合され得る。

【0030】

ビデオコード化プロセス

[0046]簡単に上述したように、ビデオエンコーダ20は、ビデオデータを符号化する。ビデオデータは、1つまたは複数のピクチャを備え得る。ピクチャの各々は、ビデオの一部を形成する静止画像である。いくつかの事例では、ピクチャは、ビデオ「フレーム」と呼ばれ得る。ビデオエンコーダ20がビデオデータを符号化するとき、ビデオエンコーダ20は、ビットストリームを生成し得る。ビットストリームは、ビデオデータのコード化表現を形成するビットのシーケンスを含み得る。ビットストリームは、コード化されたピクチャと関連データとを含み得る。コード化されたピクチャは、ピクチャのコード化表現である。

【0031】

[0047]ビットストリームを生成するために、ビデオエンコーダ20は、ビデオデータ中の各ピクチャに対して符号化動作を実行し得る。ビデオエンコーダ20がピクチャに対して符号化動作を実行するとき、ビデオエンコーダ20は、一連のコード化されたピクチャと関連データとを生成し得る。関連データは、QPのようなコード化パラメータのセットを含み得る。コード化されたピクチャを生成するために、ビデオエンコーダ20は、ピクチャを、サイズが等しいビデオブロックへと区分化し得る。ビデオブロックは、2次元アレイのサンプルであり得る。コード化パラメータは、ビデオデータのすべてのブロックについてコード化オプション(例えば、コード化モード)を定義し得る。所望のレート-歪みパフォーマンス(desired rate-distortion performance)を達成するために、コード化オプションが選択され得る。

【 0 0 3 2 】

[0048]いくつかの例では、ビデオエンコーダ20は、ピクチャを、複数のスライスへと区分化し得る。スライスの各々は、画像（例えば、フレーム）中に空間的に異なる領域を含み得、それは、この画像またはフレーム中の残りの領域からの情報なしに独立して復号されることができる。各画像またはビデオフレームが単一のスライスにおいて符号化され得るか、各画像またはビデオフレームがいくつかのスライスにおいて符号化され得る。DSCでは、各スライスを符号化するために割り振られるターゲットビットは、略一定であり得る。ピクチャに対して符号化動作を実行することの一部として、ビデオエンコーダ20は、ピクチャの各スライスに対して符号化動作を実行し得る。ビデオエンコーダ20がスライスに対して符号化動作を実行するとき、ビデオエンコーダ20は、このスライスに

10

【 0 0 3 3 】

D S C ビデオエンコーダ

[0049]図2Aは、本開示で説明される態様に係る技法をインプリメントし得るビデオエンコーダ20の例を例示するブロック図である。ビデオエンコーダ20は、本開示の技法のうちのいくつかまたはすべてを実行するように構成され得る。いくつかの例では、本開示で説明される技法は、ビデオエンコーダ20の様々な構成要素の間で共有され得る。いくつかの例では、追加的または代替的に、プロセッサ（図示されない）は、本開示で説明される技法のうちのいくつかまたはすべてを実行するように構成され得る。

20

【 0 0 3 4 】

[0050]説明の目的のために、本開示は、DSCコード化のコンテキストにおいてビデオエンコーダ20を説明する。しかしながら、本開示の技法は、他のコード化規格または方法に適用可能であり得る。

【 0 0 3 5 】

[0051]図2Aの例では、ビデオエンコーダ20は、複数の機能的な構成要素を含む。ビデオエンコーダ20の機能的な構成要素は、色空間コンバータ105と、バッファ110と、平坦度検出器115と、レートコントローラ120と、予測器、量子化器、および再構築器構成要素125と、ラインバッファ130と、インデックス付き色履歴135と、エントロピーエンコーダ140と、サブストリームマルチプレクサ145と、レートバッファ150とを含む。他の例では、ビデオエンコーダ20は、より多い、より少ない、または異なる機能的な構成要素を含み得る。

30

【 0 0 3 6 】

[0052]色空間105コンバータは、入力された色空間を、コード化インプリメンテーションで使用される色空間に変換し得る。例えば、1つの例示的な実施形態では、入力ビデオデータの色空間は、赤、緑、青（RGB）色空間であり、コード化は、輝度Y、クロミナンス緑Cg、およびクロミナンスオレンジCo（Y C g C o）色空間においてインプリメントされる。色空間変換は、ビデオデータへの追加およびシフトを含む方法によって実行され得る。他の色空間にある入力ビデオデータが処理され得、他の色空間への変換が同じく実行され得ることに留意されたい。

40

【 0 0 3 7 】

[0053]関連する態様では、ビデオエンコーダ20は、バッファ110、ラインバッファ130、および/またはレートバッファ150を含み得る。例えば、バッファ110は、ビデオエンコーダ20の他の部分による使用より前、色空間変換されたビデオデータを保持し得る。別の例では、ビデオデータは、RGB色空間で記憶され得、色空間変換は、色空間変換されたデータがより多くのビットを必要とされ得るため、必要に応じて実行され得る。

【 0 0 3 8 】

[0054]レートバッファ150は、ビデオエンコーダ20においてレート制御メカニズムの一部として機能し得、これは、レートコントローラ120に関連して以下でより詳細に

50

説明されるであろう。各ブロックを符号化するのに費やされるビットは、極めて実質的にこのブロックの性質に基づいて変動することができる。レートバッファ150は、圧縮ビデオにおけるレート変動を滑らかにすることができる。いくつかの実施形態では、固定ビットレートでバッファからビットが取り出されるC B Rバッファモデルが用いられる。C B Rバッファモデルでは、ビデオエンコーダ20が過度に多くのビットをビットストリームに追加する場合、レートバッファ150がオーバーフローし得る。他方で、ビデオエンコーダ20は、レートバッファ150のアンダーフローを防ぐために、十分なビットを追加しなければならない。

【0039】

[0055]ビデオデコーダ側では、ビットが、固定ビットレートでビデオデコーダ30のレートバッファ155（以下でさらに詳細に説明される図2Bを参照）に追加され得、ビデオデコーダ30は、ブロックごとにビットの変数（variable numbers）を取り除き得る。適切な復号を確実にするために、ビデオデコーダ30のレートバッファ155は、圧縮ビットストリームの復号中、「アンダーフロー」も「オーバーフロー」もするべきではない。

【0040】

[0056]いくつかの実施形態では、バッファフルネス（B F）は、バッファ中に現在あるビット数を表す値BufferCurrentSizeと、レートバッファ150のサイズを表すBufferMaxSize、例えば、任意の時点でレートバッファ150に記憶されることが出来る最大ビット数、とに基づいて定義され得る。B Fは、次のように算出され得る：

$$B F = ((\text{BufferCurrentSize} \times 100) / \text{BufferMaxSize})$$

【0041】

[0057]B Fを算出することへの上記アプローチが単なる例示であること、および、B Fが、特定のインプリメンテーションまたはコンテキストに依存して、任意の数の異なる方法で算出され得ることが留意されたい。

【0042】

[0058]平坦度検出器115は、ビデオデータ中の複雑な（例えば、平坦でない）エリアからビデオデータ中の平坦な（例えば、単純なまたは均一の）エリアへの、および/または、その逆への変化を検出することができる。「複雑（な）」および「平坦（な）」という用語は、一般に、ビデオエンコーダ20がビデオデータのそれぞれの領域を符号化することの難しさを示すために本明細書で使われるであろう。ゆえに、複雑（な）という用語は、本明細書で使われる場合、一般に、ビデオデータの領域を、ビデオエンコーダ20が符号化し辛いものとして説明し、例えば、テキストチャ化されたビデオデータ、高空間周波数、および/または符号化し辛い他の特徴を含み得る。平坦（な）という用語は、本明細書で使われる場合、一般に、ビデオデータの領域を、ビデオエンコーダ20が符号化し易いものとして説明し、例えば、ビデオデータ中の滑らかな勾配、低空間周波数、および/または符号化し易い他の特徴を含み得る。符号化済みビデオデータ中の量子化アーティファクトを低減するために、複雑領域から平坦領域への遷移がビデオエンコーダ20によって使用され得る。具体的には、レートコントローラ120および予測器、量子化器、および再構築器構成要素125は、複雑領域から平坦領域への遷移が識別されるときに、そのような量子化アーティファクトを低減することができる。同様に、現在ブロックをコード化するのに必要とされる予想レートを低減するためにQ Pを増加するため、平坦領域から複雑領域への遷移がビデオエンコーダ20によって使用され得る。

【0043】

[0059]レートコントローラ120は、コード化パラメータ、例えばQ P、のセットを決定する。Q Pは、レートバッファ150がオーバーフローまたはアンダーフローしないことを確実にするターゲットビットレートのピクチャ品質を最大化するために、レートバッファ150のバッファフルネスとビデオデータの画像アクティビティ（例えば、複雑領域から平坦領域への遷移またはその逆）とに基づいて、レートコントローラ120によって調整され得る。レートコントローラ120はまた、最適なレート歪みパフォーマンスを達

成するために、ビデオデータの各ブロックについて特定のコード化オプション（例えば、特定のモード）を選択する。レートコントローラ120は、ビットレート制約を満たすように、例えば、実際のコード化レート全体がターゲットビットレート内に収まるように、再構築画像の歪みを最小化する。ゆえに、レートコントローラ120の1つの目的は、レート歪みパフォーマンスを最大化しつつレートに対する瞬間的および平均的な制約を満たすために、QP、コード化モード、等のコード化パラメータのセットを決定することである。

【0044】

[0060]予測器、量子化器、および再構築器構成要素125は、少なくとも、ビデオエンコーダ20の3つの符号化動作を実行し得る。予測器、量子化器、および再構築器構成要素125は、多数の異なるモードで予測を実行し得る。1つの例となる断定モードは、中央値適応予測の修正バージョンである。中央値適応予測は、ロスレスなJPEG規格（JPEG-Loss）によってインプリメントされ得る。予測器、量子化器、および再構築器構成要素125によって実行され得る中央値適応予測の修正バージョンは、3つの連続したサンプル値の並行予測を可能にし得る。別の例となる予測モードは、ブロック予測である。ブロック予測では、上の線または同じ線の左にある、前に再構築された画素からサンプルが予測される。いくつかの実施形態では、ビデオエンコーダ20およびビデオデコーダ30は両方とも、再構築された画素に対して同一探索を実行して、ブロック予測用法を決定し得るため、ブロック予測モードではいずれのビットも送られる必要はない。他の実施形態では、ビデオエンコーダ20は、探索を実行し、ビデオデコーダ30が別個の探索を実行する必要がないように、ビットストリームにおいてブロック予測ベクトルをシグナリングし得る。構成要素範囲（component range）の中間点を使用してサンプルが予測される中間点予測モードもまたインプリメントされ得る。中間点予測モードは、ワーストケースのサンプルであっても、圧縮ビデオに必要とされるビット数の制限（bounding）を可能にし得る。

【0045】

[0061]予測器、量子化器、および再構築器構成要素125もまた、量子化を実行する。例えば、量子化は、シフタを使用してインプリメントされ得る2のべき乗量子化器（power-of-2 quantizer）を介して実行され得る。2のべき乗量子化器の代わりに他の量子化技法がインプリメントされ得ることに留意されたい。予測器、量子化器、および再構築器構成要素125によって実行される量子化は、レートコントローラ120によって決定されるQPに基づき得る。最後に、予測器、量子化器、および再構築器構成要素125はまた、逆量子化残差を予測値に加えることと、その結果がサンプル値の有効範囲外にならないことを確実にすることを含む再構築を実行する。

【0046】

[0062]予測器、量子化器、および再構築器構成要素125によって実行される予測、量子化、および再構築への上述した例となるアプローチは単なる例示であること、および、他のアプローチがインプリメントされ得ることに留意されたい。予測器、量子化器、および再構築器構成要素125が、予測、量子化、および/または再構築を実行するための従属構成要素を含み得ることも留意されたい。予測、量子化、および/または再構築が、予測器、量子化器、および再構築器構成要素125の代わりに、いくつかの別個のエンコーダ構成要素によって実行され得ることにさらに留意されたい。

【0047】

[0063]ラインバッファ130は、予測器、量子化器、および再構築器構成要素125とインデックス付き色履歴135とが、バッファされたビデオデータを使用することができるように、予測器、量子化器、および再構築器構成要素125からの出力を保持する。インデックス付き色履歴135は、最近使用された画素値を記憶する。これらの最近使用された画素値は、専用シンタックスを介して、ビデオエンコーダ20によって直接参照されることができる。

【0048】

10

20

30

40

50

[0064] エントロピーエンコーダ 140 は、インデックス付き色履歴 135 と、平坦度検出器 115 によって識別される平坦度遷移とに基づいて、予測器、量子化器、および再構築器構成要素 125 から受け取った予測残差および任意の他のデータ（例えば、予測器、量子化器、および再構築器構成要素 125 によって識別されるインデックス）を符号化する。いくつかの例では、エンタロピーエンコーダ 140 は、サブストリームエンコーダにつき 1 クロックあたり 3 つのサンプルを符号化し得る。サブストリームマルチプレクサ 145 は、ヘッダーレスパケット多重化スキームに基づいて、ビットストリームを多重化し得る。これは、ビデオデコーダ 30 が、並行して 3 つのエントロピーデコーダを稼働させることを可能にし、これは、1 クロックあたり 3 つの画素の復号を容易にする。サブストリームマルチプレクサ 145 は、パケットがビデオデコーダ 30 によって効率的に復号されることができるようパケット順序を最適化し得る。エントロピーコード化への異なるアプローチがインプリメントされ得、これは、1 クロックあたり 2 のべき乗個の画素（例えば、2 画素 / クロックまたは 4 画素 / クロック）の復号を容易にし得ることに留意されたい。

10

【0049】

D S C ビデオデコーダ

[0065] 図 2 B は、本開示で説明される態様に係る技法をインプリメントし得るビデオデコーダ 30 の例を例示するブロック図である。ビデオデコーダ 30 は、本開示の技法のうちのいくつかまたはすべてを実行するように構成され得る。いくつかの例では、本開示で説明される技法は、ビデオデコーダ 30 の様々な構成要素の間で共有され得る。いくつかの例では、追加的または代替的に、プロセッサ（図示されない）は、本開示で説明される技法のうちのいくつかまたはすべてを実行するように構成され得る。

20

【0050】

[0066] 説明の目的のために、本開示は、D S C コード化のコンテキストにおいてビデオデコーダ 30 を説明する。しかしながら、本開示の技法は、他のコード化規格または方法に適用可能であり得る。

【0051】

[0067] 図 2 B の例では、ビデオデコーダ 30 は、複数の機能的な構成要素を含む。ビデオデコーダ 30 の機能的な構成要素は、レートバッファ 155 と、サブストリームマルチプレクサ 160 と、エントロピーデコーダ 165 と、レートコントローラ 170 と、予測器、量子化器、および再構築器構成要素 175 と、インデックス付き色履歴 180 と、ラインバッファ 185 と、色空間コンバータ 190 とを含む。ビデオデコーダ 30 の例示される構成要素は、図 2 A のビデオエンコーダ 20 に関連して上で説明した対応する構成要素に類似する。このように、ビデオデコーダ 30 の構成要素の各々は、上で説明したようなビデオエンコーダ 20 の対応する構成要素と同様の方式で動作し得る。

30

【0052】

Q P 算出

[0068] 1 つのアプローチでは、レートコントローラ 120 は、次の式に基づいて、ビデオデータの現在ブロックについての Q P (currQP と表される) を導出または算出し得る：

$$\text{currQP} = \text{prevQP} + \text{QpAdj} * (\text{diffBits} > 0 ? 1 : -1)$$

40

ここで、prevQP は、前のブロックに関連付けられた Q P であり、QpAdj は、diffBits の大きさ (magnitude) に基づいて算出され得る Q P オフセット値（例えば、Q P 調整値）である。diffBits は、previousBlockBits と targetBits との間の差分を表し、ここで、previousBlockBits は、前のブロックをコード化するために使用されるビット数を表し、targetBits は、現在ブロックをコード化するためのターゲットビット数を表す。previousBlockBits > targetBits であるとき、diffBits は正であり、レートコントローラ 120 は、オフセット値 QpAdj を prevQP 値に加算することで現在ブロックの Q P (currQP) を導出し得る。換言すると、Q P 値 currQP は、diffBits が正であるとき、prevQP 値から値が減少しない。previousBlockBits < targetBits であるとき、diffBits は負またはゼロであり、レートコントローラ 120 によって導出される currQP は、prevQP 値から増加しない。いくつ

50

かの実施形態では、レートコントローラ 120 は、QpAdj が diffBits の大きさの増加に伴って単調に増加するように、diffBits の関数としてオフセット値 QpAdj を算出し得ることに留意されたい。

【0053】

[0069] 図 3 は、QP 調整値 QpAdj を算出するための例示的な技法を描写するグラフ 300 を例示する。グラフ 300 は、ゼロから開始して diffBits の値がプロットされている水平軸 302 を例示する。いくつかのインプリメンテーションでは、diffBits > 0 であるとき、diffBits の値は、K 個の閾値を使用して K + 1 個の範囲へと分類され得る。図 3 では、K 個の閾値は、閾値 1、閾値 2、閾値 3 ... および閾値 K というラベルで例示され、K + 1 個の範囲は、範囲 1、範囲 2、範囲 3、... および範囲 K + 1 というラベルで例示されている。K + 1 個の範囲の各範囲は、特定の QpAdj 値に関連付けられ得る。例えば、レートコントローラ 120 は、diffBits の値の範囲インデックスが増加するに伴い、QpAdj 値を増加させ得る。いくつかの実施形態では、diffBits = 0 であるとき、レートコントローラ 120 は、J 個の閾値（図示されない）を使用して diffBits の絶対値を J + 1 個の範囲へと分類し得、ここで、特定の QpAdj 値が、J + 1 個の範囲の各々に割り当てられている。

【0054】

[0070] 他の態様では、レートコントローラ 120 は、バッファ 110 のアンダーフローおよび/またはオーバーフローを防ぐために、図 2 A に例示されたバッファ 110 のフルネスに基づいて currQP 値を調整し得る。バッファ 110 のフルネスは、算出されたバッファフルネスパラメータ BF の観点から表され得る。特に、BF が特定の閾値（例えば、 p_1 ）を超えると、レートコントローラ 120 は、currQP を、固定オフセット値（例えば、 p_1 ）だけインクリメントし得る。例えば、currQP は、次のように調整され得る： $\text{currQP} + = p_1$ 。さらに、BF が特定の閾値（例えば、 q_1 ）を下回るとき、レートコントローラ 120 は、固定値 q_1 だけ currQP をデクリメントし得る、例えば、 $\text{currQP} - = q_1$ 。特定の態様では、図 3 の diffBits に関連して例示される閾値および範囲と同様に、BF に対する複数の閾値が用いられ得る。例えば、BF の各閾値は、currQP を調整するために、対応するオフセット値に関連付けられ得る。

【0055】

デルタサイズ単位 - 可変長コード化

[0071] いくつかの実施形態では、コード（例えば、ビデオエンコーダ 20 またはビデオデコーダ 30）は、低コストで固定レートの視覚的にロスレスな圧縮を提供するためにデルタサイズ単位（DSU）コード化を使用し得る。コードは、（ブロックサイズ $P \times Q$ の）ブロックベースアプローチに基づいて設計され、多数のコード化モードを備え得る。例えば、ビデオデータの各ブロックについてのコード化モードは、変換（例えば、DCT、アダマール）、ブロック予測、DPCM、パターン、中間点予測（MPP）、および中間点予測フォールバック（MPPF）モードを含み得るがそれらに限定されるわけではない。いくつかの実施形態では、異なるタイプのコンテンツまたは画像を効率的に圧縮するために、いくつかの異なるコード化モードがエンコーダ 20 によってインプリメントされることができる。例えば、いくつかの実施形態では、エンコーダ 20 は、パターンモードを使用してテキスト画像を、変換モードを使用して自然画像を圧縮し得る。

【0056】

[0072] いくつかの実施形態では、エンコーダ 20 は、レート制御メカニズム（例えば、図 2 A に例示されるようなレートコントローラ 120）に基づいて、ビデオデータの各それぞれのブロックに対して複数のコード化モードから 1 つのコード化モードを選択する、選ぶ、または決定する。エンコーダ 20 はまた、選択されたコード化モードをビデオデータビットストリームにおいてシグナリングし得、それにより、デコーダ 30 は、このビットストリームを受信すると、ビデオデータをコード化するために使用されるコード化モードを決定することができ得る。レートコントローラ 120 は、各ブロックに対して効率的なコード化モードを、このモードのレートと歪みの両方を考慮することで、選択することを目的とし得る。レートコントローラ 120 は、（図 2 A に例示されるような）バッファ

1 1 0 または到来するブロックデータを記憶するための他のタイプのメモリに関連付けられたバッファモデルによってサポートされ得る。バッファ 1 1 0 は、バッファ 1 1 0 が一度も（例えば、バッファ中のゼロビットより少ない）アンダーフローの状態にも（例えば、バッファサイズが、設定された最大サイズを過ぎて増加している）オーバーフローの状態にもならないように、例えば、D S C のようなコーデックの要件にしたがって構成され得る。

【 0 0 5 7 】

[0073] 上で述べたように、エンコーダ 2 0 の予測器、量子化器、および再構築器構成要素 1 2 5 は、コード化されたビデオデータのブロックにおいてロスをもたらし得る量子化を実行し得、ここにおいて、ロスの総量は、ブロックの量子化パラメータ（Q P）によって制御されることができる。例えば、いくつかの実施形態では、予測器、量子化器、および再構築器構成要素 1 2 5 は、1 つまたは複数のビットプレーンをドロップすることで量子化を実行し得、ここにおいて、ドロップされるビットプレーンの数は、ブロックの Q P に関連付けられた量子化ステップサイズによって示され得る。いくつかの実施形態では、エンコーダ 2 0 が各 Q P についての量子化ステップサイズを記憶する代わりに、エンコーダ 2 0 は、Q P の関数としてスケーリングマトリックスを指定し得る。各 Q P についての量子化ステップサイズは、このスケーリングマトリックスから導出されることができ、導出される値は、必ずしも 2 の累乗である必要はなく、例えば、導出される値は、2 の累乗以外に等しい値でもあり得る。いくつかの実施形態では、エンコーダ 2 0 によるスケーリングマトリックスの使用は、予測器、量子化器、および再構築器構成要素 1 2 5 が、単にビットプレーンを取り除くことより大きい粒度で量子化を実行し、潜在的にパフォーマンスを向上させることを可能し得る。

【 0 0 5 8 】

[0074] いくつかの実施形態では、ビデオデータの所与のブロック中の単一の構成要素（例えば、特定の色成分、ルーマまたはクロマ成分、等）のすべての値がゼロである場合、エンコーダ 2 0 は、スキップモードを使用してこのブロックを効率的にコード化し得る。スキップモードコード化では、エンコーダ 2 0 は、現在ブロックがスキップモードを使用してコード化される（成分のすべての値が現在ブロックについてゼロである場合）か、スキップモードでコード化されない（ブロック中の成分の少なくとも 1 つの値が非ゼロである場合）かを示す、デコーダ 3 0 によって読み取られ得るシンタックス要素またはインジケータ（例えば、1 ビットフラグ）をコード化し得る。

【 0 0 5 9 】

[0075] いくつかの実施形態では、エンコーダ 2 0 は、プレフィックス部分およびサフィックス部分を使用して（「グループ」とも呼ばれる）長さ K のサンプルベクトル（K-length sample vector）の量子化された残差値をコード化するためにデルタサイズ単位 - 可変長コード化（D S U - V L C）を使用し得る。サンプルは、例えば、R G B 4 4 4 の場合、単一の色成分中の値、を指すことができ、各画素は、3 つのサンプルを有する。プレフィックス部分は、サフィックス部分に続く残差値のサイズ（例えば、ビットで表した長さ）（このサイズは、B ビットとして表される）を示し、サフィックス部分は、サンプルベクトル中のすべてのサンプルの実際の残差値を示す。いくつかの実施形態では、グループ中の K 個の残差値の各々は、同じ数のビットを使用して、2 の補数でコード化される。

【 0 0 6 0 】

[0076] 図 4 は、いくつかの実施形態に係る、K = 4 個のサンプルを有するサンプルベクトル 4 0 2 をコード化するための D S U - V L C 構造 4 0 4 を示す。いくつかの実施形態では、D S U - V L C 構造 4 0 4 は、図 2 A に例示されるようなエントロピーエンコーダ 1 4 0 に対応し得る。サンプルベクトル 4 0 2 は、ビデオデータの特定の色成分の残差値に各々対応する複数のサンプルを備え得る。図 4 に例示されるように、サンプルベクトル 4 0 2 は、4 つのサンプル（S 0、S 1、S 2、および S 3）を備え得る。D S U - V L C 構造 4 0 4 は、受け取ったサンプルベクトル 4 0 2 をコード化して、D S U - V L C コード 4 0 6（以降、コード 4 0 6）を形成し得る。コード 4 0 6 は、プレフィックス 4 0

8 およびサフィックス 4 1 0 を備える。サフィックス 4 1 0 は、サンプルベクトル 4 0 2 の受け取ったサンプル（例えば、S 0、S 1、等）に各々対応する複数のサフィックス部分（例えば、サフィックス 1、サフィックス 2、等）を備える。コード 4 0 6 のプレフィックス 4 0 8 およびサフィックス 4 1 0 のさらなる説明は、図 5 を参照して以下に説明される。

【 0 0 6 1 】

[0077] 図 5 は、値 [1 , - 2 , - 1 , 0] を有する 4 つのサンプルを備えるサンプルベクトル（図示されない）のためにエントロピーエンコーダ 1 4 0 によってコード化され得るコード 5 0 2 を例示する。コード 5 0 2 は、図 4 に例示するコード 4 0 6 に対応し得る。加えて、コード 5 0 2 は、（それぞれ、図 4 のプレフィックス 4 0 8 およびサフィックス 4 1 0 に対応し得る）プレフィックス 5 0 4 およびサフィックス 5 0 6 を備える。サフィックス 5 0 6 は、サンプルベクトルのサンプル値に各々対応する 4 つのサフィックス部分を備え得る。

10

【 0 0 6 2 】

[0078] 2 の補数表現（two's complement representation）を使用して、エントロピーエンコーダ 1 4 0 は、サンプルベクトルのサンプルの各々をコード化するために $B = 2$ ビットを使用し得る。コード 5 0 2 において、プレフィックス 5 0 4 は、サフィックス 5 0 6 のコード化されたサフィックス部分の各々の長さが 2 ビットであることを示す単項コード ' 0 0 1 ' を使用して表され得る。サフィックス 5 0 6 は、各々 2 ビットを使用してコード化された、実際のコード化されたサンプル値サンプルベクトルをそれぞれ表す値 [0 1 , 1 0 , 1 1 , 0 0] によって表され得る。プレフィックス 5 0 4 の復号、これは通常単一クロックにおいて行われ得る、を行うことで、デコーダ 3 0 は、サフィックス 5 0 6 の 4 つのシンボルすべてを並行して復号することができ得る。

20

【 0 0 6 3 】

グループ区分化

[0079] 図 6 は、いくつかの実施形態に係る、ビデオデータの所与の $P \times Q$ ブロックのサンプルを複数のサンプルベクトル（グループ）へと区分化する例を例示する。図 6 に例示されるように、ブロック 6 0 2 は、16 個のサンプルを備える 2×8 ブロックであり得る。ブロック 6 0 2 の各サンプルは、ブロック 6 0 2 に対応するビデオデータの特定の色成分の量子化された残差値に対応し得る。サンプルが DSU - VLC 構造 4 0 4 を使用してコード化される前に、エンコーダ 2 0 は、サンプルを複数のグループへと区分化し得る。例えば、図 6 は、4 つのサンプルを各々備える 4 つのサンプルベクトル 6 0 4（例えば、サンプルベクトル 6 0 4 A、6 0 4 B、6 0 4 C、および 6 0 4 D）へと区分化されるブロック 6 0 2 の 16 個のサンプルを例示する。DSU - VLC 構造 4 0 4 は、（例えば、図 4 に例示されるような）プレフィックスおよびサフィックスを各々有するコード（図示されない）を作り出すために、サンプルベクトル 6 0 4 A - 6 0 4 D をコード化し得る。上で説明したように、（図 3 に例示されるような）デコーダ 3 0 は、これらコードの各コードのプレフィックスおよびサフィックスを並行して復号することができ得、これは、デコーダ 3 0 が、1 クロックサイクルあたり 4 つのサンプルを復号することを可能にする。

30

40

【 0 0 6 4 】

[0080] ブロック 6 0 2 のサンプルをグループへと区分化するためにエンコーダ 2 0 を使用することで、コード化されたグループを復号するときに、1 クロックあたり複数のサンプルというスループットがデコーダ 3 0 によって達成されることができ得る。図 6 は、ブロック 6 0 2 のサンプルがサンプルベクトル 6 0 4 へと均一に区分化されていることを例示するが、エンコーダ 2 0 が、サンプルのブロックを N 個のサンプルベクトルへと均一にまたは不均一に区分化し得ることが理解される。均一のグループ化方法では、 N 個すべてのサンプルベクトル 6 0 4 は、同じ数のサンプルを有するであろう。他方で、不均一グループ化方法を使用するときは、各サンプルベクトル 6 0 4 中のサンプルの数が異なり得る。

【 0 0 6 5 】

50

[0081]いくつかの実施形態では、ブロック 602 の区分化が均一であるか不均一であるかは、このブロック 602 に関連付けられたコード化モードに基づき得る。例えば、エンコーダ 20 は、ブロック予測モードでは均一のグループ化方法を使用し得、変換モードでは不均一グループ化方法を使用する。

【0066】

ベクトルベースエントロピーコード化

[0082]図 7 は、図 2 A に例示されたエントロピーエンコーダ 140 に対応し得るコード 702 を使用してベクトルベース EC プロセスの例示的なブロック図を例示する。コード 702 は、サンプルベクトル 704 をコード化するために使用され得る。サンプルベクトル 704 は、図 4 に例示されたようなサンプルベクトル 402 および図 6 に例示されたようなサンプルベクトル 604 に類似し得る。

10

【0067】

[0083]図 7 に例示されるように、サンプルベクトル 704 は、ベクトルサイズ K (例えば、 $[S_0, S_1, \dots, S_{K-1}]$) を有する。加えて、サンプルベクトル 704 は、 B ビットという必要とされるサンプルサイズを有し得る (例えば、コード 702 は、グループ中の各サンプルを正確に表すために 1 サンプルあたり B ビットを必要とし得る)。所与の K 値および B 値について、可能な (possible) 長さ K のサンプルベクトルの合計数は $2^{B \cdot K}$ である。例えば、 $K = 2$ および $B = 1$ であるとき、1 ビットを使用して表されることが出来るサンプルを有する、可能なすべての長さ 2 のサンプルベクトルのセットは、4 つ、すなわち、(2 の補数表現を使用するとき) $[-1, 0]$ 、 $[-1, -1]$ 、 $[0, 0]$ 、 $[0, -1]$ である。図 4 - 5 に例示されたコード化スキームを使用して、コード 702 は、同じ数のビット (例えば、 $B \cdot K$ ビット、または 2 ビット) を使用して、これらのサンプルベクトルの各々に対応するサフィックスをコード化し得る。このように、 K 値および B 値の所与の組合せについて、サンプルベクトルをコード化するために使用されるビット数は、 $2^{B \cdot K}$ 個の可能なベクトルのセット中の長さ K のサンプルベクトルがコード化されているかにかかわらず、同じである。これは、セット中の各サンプルベクトルの生起確率を考慮に入れない。

20

【0068】

[0084]しかしながら、画像情報をコード化するとき、特定のサンプルベクトル 704 は、長さ K のサンプルベクトルのセット中の他のサンプルベクトルと比べて、より確からしい (probable) 傾向にあり得る。いくつかの実施形態では、 $2^{B \cdot K}$ 個のベクトルのセット中の長さ K のサンプルベクトルの分布が均一でない (例えば、セット中のすべてのベクトルが等しく確からしいとは限らない) とき、コード 702 は、生起確率に基づいて、長さ K のサンプルベクトル 704 を別々にコード化し得る。例えば、コード 702 は、より確からしくない長さ K のサンプルベクトル 704 と比べて、より少ない数のビットを使用して、より確からしい長さ K のサンプルベクトル 704 をコード化し得る。以下に説明される技法は、可能な長さ K のサンプルベクトル 704 のセットが不均一分布から引き出されるとき、長さ K のサンプルベクトル 704 をコード化するための効率的なコード化ストラテジを対象としている。

30

【0069】

[0085]いくつかの実施形態では、コード 702 は、可能な長さ K のサンプルベクトル 704 のセットが不均一分布から引き出されるとき、長さ K のサンプルベクトル 704 をより効率的にコード化するためにベクトルベースエントロピーコード化 (EC) 方法を使用し得る。いくつかの実施形態では、コード 702 は、所与の長さ K のサンプルベクトル 704 を一意的な単一値 (例えば、「インデックス」706) へと変換するために変換 712 を実行することで、グループ中の長さ K のサンプルベクトル 704 に対してベクトルベース EC を実行し得る。長さ K のサンプルベクトル 704 と「インデックス」706 との間には 1 対 1 対応が存在し得る。

40

【0070】

[0086]コード 702 は、算出されたインデックス値 706 を使用して、このインデック

50

ス値 706 を「codeNumber」708 にマッピングするマッピング 714 を実行し得る。VLC コード 716 を使用して、コード 702 は、ビデオデータビットストリームの一部としてデコーダ 165 に送信され得るコード化されたビット 710 のシーケンスを形成するために、「codeNumber」708 をコード化し得る。VLC コード 716 は、非構造化または構造化され得る。非構造化 VLC コード 716 の例には、ハフマン、算術が含まれ、構造化 VLC コード化には、指数ゴロム (EG)、ゴロム - ライス (GR)、または EG と GR の混合が含まれ得る。デコーダ 165 において、コード化されたビット 710 の単一シーケンスをパースする (典型的に単一のクロックにおいて行われることができる) と、サンプルベクトル 704 中のすべての K 個のサンプルは、再構築されることができる。したがって、ベクトルベース EC の使用は、典型的に K 個のサンプル / クロックという高いスループットを供給する。

10

【0071】

[0087]いくつかの実施形態では、コード 702 は、算出された「インデックス」値 706 を「codeNumber」708 にマッピングするために使用されることができる LUT (ルックアップテーブル、図示されない) を使用して、マッピング 714 を実行し得る。LUT は、所与のグループの長さ K のサンプルベクトル 704 の確率値 (probability value) に基づいて構築されることができる。例えば、いくつかの実施形態では、LUT は、より確からしいと考えられるサンプルベクトル 704 が、より小さい値を有するかそうでなければより少ない数のビットを使用してコード化されることができる codeNumbers 708 (例えば、より短いコード長を有する codeNumbers 708) にマッピングされるように構築され得る。例えば、より高い確率のサンプルベクトルのための codeNumbers 708 は、B K ビットより少ない数を使用してコード化され得る。いくつかの実施形態では、サンプルベクトル 704 を codeNumbers 708 にマッピングするための LUT またはデータ構造は、LUT またはデータ構造中のサンプルベクトル 704 をルックアップするときに、より確からしいと考えられるサンプルベクトル 704 がより迅速にまたはより高い優先度でアクセスされ得るように構築され得、これは、デコーダ 165 によるより正確なアクセスを可能にする。

20

【0072】

[0088]いくつかの実施形態では、エンコーダ 20 は、ルーマ成分サンプルおよびクロマ成分サンプルの両方について同じ LUT を記憶し得、他の実施形態では、異なる LUT が、ルーマ成分サンプルおよびクロマ成分サンプルをコード化するために使用され得る。加えて、デコーダ 30 は、エンコーダ 20 によって使用される LUT またはデータ構造のそれ自体の複写を記憶し得る。例えば、デコーダ 30 は、コード化されたビデオデータのビットストリームを受け取ることと、1 つまたは複数の codeNumbers 708 を復号することとに回答して、対応するサンプルベクトル 704 を決定するために、それ自体の LUT を使用し得る。

30

【0073】

[0089]いくつかの実施形態では、コード 702 は、すべてのサイズ B に対して単一の LUT を使用する。他の実施形態では、コード 702 は、異なるサイズの B に対して別個の LUT を使用し得る。例えば、ある実施形態では、コード 702 は、B = 1 であるときには第 1 の LUT 1 を、B = 2 であるときには第 2 の LUT 2 を使用し得る。コード 702 がサイズ B に基づいて特定の LUT を選択するとき、コード 702 は、使用される LUT のタイプが明示的にシグナリングされ得るように、コード化されたビット 710 のシーケンスをコード化し得る (例えば、B = 1 のときには LUT タイプ = 1 がシグナリングされ、B = 2 のときには、LUT タイプ = 2 がシグナリングされる、等)。コード化されたビット 710 において LUT のタイプをシグナリングすることで、デコーダ 165 は、ビット 710 のシーケンスを復号するときに使用されるべき対応する LUT を識別することができるであろう。いくつかの実施形態では、シグナリングは、単項コードの形であり得る。代替的に、コード化されたビット 710 において LUT タイプをシグナリングすることは、固定長のコードを使用して行われ得る。他の実施形態では、コード

40

50

化されたビット710においてLUTタイプをシグナリングするために、ハフマンまたは算術のようなVLCコードを使用し得る。

【0074】

[0090]図8は、コード化されたビット806のシーケンスを作り出すため、 $K = 4$ 個のサンプルを含むサンプルベクトル402をコード化するためにベクトルベースECを使用するコード702の例を示す。(図7に例示されたコード化されたビット710のシーケンスに対応し得る)コード化されたビット806は、LUTタイプ信号808と、ベクトルベースVLCコード810とを備え得る。LUTタイプ信号808は、コード化されたビット806を生成するためにコード702によって使用された(例えば、図7に例示されるようなコード番号708へのインデックス値706のマッピングを実行するために使用される)LUTのタイプを明示的にシグナリングする。ベクトルベースVLCコード810は、サンプルベクトル802中の4つすべてのサンプルを表す。いくつかの実施形態では、LUTタイプ信号808はまた、サンプルベクトル402のサンプルの値をコード化するのに必要とされるビット数(a number of bits)Bを示し得る。

【0075】

[0091]いくつかの実施形態では、エンコーダ20が、特定のコード化モード(例えば、変換モード)についてのサンプルベクトル402を構築するために不均一区分化方法(例えば、異なるK値)を適用するとき、別個のLUTは、各サンプルベクトル中のサンプルの数Kが異なることにより異なるサンプルベクトルサイズKに対して使用され得る。このように、異なるB値、異なるK値、または異なるKおよびB値の組み合わせに対応する異なるLUTが使用され得る。

【0076】

[0092]インデックス値706を対応するcodeNumbers 708にマッピングするためのLUTのサイズは、サイズBが増加するに伴い増加し得る。このように、いくつかの実施形態では、ベクトルベースECは、サンプルベクトル402のサイズBが特定の閾値より小さいとき、コード702によってのみ使用される。これは、エンコーダ20およびデコーダ30に、Bのより大きな値のためのより大きなサイズのLUTを記憶させないことでメモリ要件を低減するために行われ得る。例えば、Bが閾値限界(threshold limit)を超えると、コード702は、非ベクトルベースコード化技法(例えば、図4-5に例示されたDSU-VLCコード化スキーム)を使用して、サンプルベクトル402をコード化し得る。例えば、閾値が4である実施形態では、コード702は、受け取ったサンプルベクトル704がサイズ $B = 1, 2$ 、または3であるときにのみベクトルベースECを使用し、(図4-5に例示されるような)DSU-VLCコード化は、サイズBが4以上であるサンプルベクトル704に対して使用される。

【0077】

[0093]本明細書で開示される技法は、ブロック予測(BP)、変換モード、BPスキップモード、変換スキップ、MPP、またはMPPFのような、ビデオデータのブロックをコード化するために使用される任意のコード化モードに適用されることができる。ベクトルベースエントロピーコード702が1つよりも多くのコード化モードに対してベクトルベースECを適用する実施形態では、各コード化モードがそれ自体のLUTに関連付けられ得るか、共通のLUTが複数のモードに関連付けられ得る。共通のLUTが使用されるか別個のLUTが使用されるかは、メモリ要件とパフォーマンスとの間のトレードオフに基づいて決定され得る。例えば、いくつかの実施形態では、BPおよび変換は、インデックス706をcodeNumbers 708にマッピングするために、同じLUTまたはデータ構造を使用し得、他の実施形態では、BPおよび変換は、異なるLUTまたはデータ構造を使用し得る。いくつかの実施形態では、コード702は、特定のモード(例えば、BPモード)でのみベクトルベースECを使用し得、他のコード化技法は、他のモード(例えば、BPモード以外)で使用される。加えて、本明細書で開示される技法は、任意のタイプのサンプリングフォーマット、例えば、4:4:4、4:2:2、4:2:0に適用されることができる。

【 0 0 7 8 】

[0094]図 9 は、LUT タイプ信号 8 0 8 が V L C コード 8 1 0 のプレフィックスと組み合わせられている、コード化されたビット 8 0 6 のシーケンスの例を例示する。(E G、G R、または E G と G R の混合といった構造化 V L C コードのような) codeNumber 7 0 8 をコード化するためにコーダ 7 0 2 によって使用され得る V L C コード 7 1 6 のいくつかのタイプは、プレフィックス 9 0 8 およびサフィックス 9 1 0 を備える V L C コード 8 1 0 という結果になり得る。いくつかの実施形態では、V L C コード 8 1 0 のプレフィックス 9 0 8 は、単項コード、または、デコーダ(例えば、デコーダ 1 6 5)が処理するのに時間のかかる何らかの他のコードフォーマットであり得、サフィックス 9 1 0 は、固定長であり得る。処理時間を低減するために、コーダ 7 0 2 は、プレフィックス 9 0 8 と L U T タイプ信号 8 0 8 とを組み合わせ、単一の V L C コード部分 9 1 2 にし得る。

10

【 0 0 7 9 】

[0095]例えば、図 9 に例示されるように、コーダ 7 0 2 は、V L C コード部分 9 1 2 を形成するために、V L C コード化の第 2 のステージ(図示されない)を通して、コード化されたビット 8 0 6 の可変長部分(例えば、LUT タイプ信号 8 0 8 およびプレフィックス 9 0 8)を組み合わせ、コード化し得る。いくつかの実施形態では、V L C コード化の第 2 のステージは、図 2 A に例示されるように、エントロピーエンコーダ 1 4 0 によって実行され得る。LUT タイプ信号 8 0 8 および V L C コードプレフィックス 9 0 8 の両方が単項コードまたは何らかの他の処理が高価なフォーマットであるため、V L C コード化の第 2 のステージを実行することは、コード化されたビット 8 0 6 を受け取るデコーダ 1 6 5 が LUT タイプ信号 8 0 8 および V L C コードプレフィックス 9 0 8 を処理するのに必要な処理量を減らすのに役立ち得る。いくつかの実施形態では、コーダ 7 0 2 は、LUT タイプ信号 8 0 8 およびプレフィックス 9 0 8 を単一のコード番号にマッピングするために第 2 の別個の LUT(図示されない)を使用し得る。次いで、コーダ 7 0 2 は、V L C コード部分 9 1 2 を形成するために、結果として得られるコード番号を(例えば、第 2 のステージの V L C コードを使用して)コード化し得る。デコーダ 1 6 5 によって受け取られると、デコーダ 1 6 5 は、完全なコード化されたビット 8 0 6 のシーケンスを取得する目的で、完全な V L C コード 8 1 0 の LUT タイプ信号 8 0 8 およびプレフィックス 9 0 8 を決定するために V L C コード部分 9 1 2 を最初に復号し得る。いくつかの実施形態では、コード化されたビット 8 0 6 の可変長部分をコード化するために使用される第 2 のステージの V L C コードは、コード化されたビット 8 0 6 のシーケンスを形成する目的で codeNumber 7 0 8 をコード化するために使用されたのと同じタイプの V L C コード 7 1 6 であり得る。他の実施形態では、第 2 のステージの V L C コードは、V L C コード 7 1 6 とは異なり得る。

20

30

【 0 0 8 0 】

符号絶対値コード化

[0096]上で述べたように、サイズ B は、所与のグループ内のすべてのサンプル(例えば、サンプルベクトル 4 0 2、6 0 4、または 7 0 4)を正確に表すのに必要なビット数を表し得る。いくつかの実施形態では、B の値は、2 の補数表現に基づいて算出され得、ここで、 $-(2^B - 1)$ から $+(2^B - 1)$ までのサンプル値を表すのに B ビットが必要とされる。しかしながら、本明細書で開示される技法は、2 の補数レジストレーションを使用した実施形態に限定されず、他のタイプの表現を使用して算出されたサンプル値に対して適用されることができる。例えば、いくつかの実施形態では、エンコーダ 2 0 は、符号絶対値表現を使用してサンプル値をコード化し得、ここで、0 から $2^B - 1$ までの範囲にあるサンプル値を表すのに B ビットが必要とされる。

40

【 0 0 8 1 】

[0097]図 1 0 A は、符号絶対値表現を使用して、(例えば、図 6 に例示されるような)例示的なサンプルベクトル 6 0 4 のエントロピーコード化の例を示す。符号絶対値表現が特定の実施形態において使用されるとき、コーダ 7 0 2 は、サンプルベクトル 6 0 4 中のサンプルの(大きさまたは絶対部分とも呼ばれる)絶対値をコード化し、サンプルベクト

50

ル 6 0 4 のサンプルの符号値に対応する符号ビットコード化が続き得る。

【 0 0 8 2 】

[0098] コーダ 7 0 2 は、コード化されたビット 1 0 0 2 のそれぞれのグループ（例えば、グループ 1 0 0 2 A、1 0 0 2 B、1 0 0 2 C、および 1 0 0 2 D）を形成するために、サンプルベクトル 6 0 4 A 乃至 6 0 4 D の各々をコード化し得る。コード化されたビット 1 0 0 2 の各グループは、プレフィックス部分 1 0 0 4 と、サフィックス部分 1 0 0 6 と、ゼロ以上の（zero or more）符号ビット 1 0 0 8 とを備える。プレフィックス部分 1 0 0 2 は、それぞれのサンプルベクトル 6 0 4 中のサンプルの最大絶対値をシグナリングするのに必要なビット数 B を示す。サフィックス部分 1 0 0 6 は、それぞれのサンプルベクトル 6 0 4 の各サンプルの絶対値を表す。最後に、符号ビット 1 0 0 8 は、それぞれの

10

【 0 0 8 3 】

[0099] 例として、サンプルベクトル 6 0 4 A が、[1 , - 3 , - 1 , 0] の値を有する 4 つのサンプルを含むと仮定する。この例では、コード化されたビット 1 0 0 2 A のプレフィックス部分 1 0 0 4 は、B = 2 の値を示す（これは、絶対値 [1 , 3 , 1 0] から算出される）。サフィックス部分 1 0 0 6 は、サンプルベクトル 6 0 4 A の絶対サンプル値に対応する部分（例えば、0 1 , 1 1 , 0 1 , 0 0 としてコード化される）を備え得る。符号ビット 1 0 0 8 は、サンプルベクトル 6 0 4 A の非ゼロサンプルの各々についての符号値を示し得る。例えば、符号ビット 1 0 0 8 は、「 1 0 0 」としてシグナリングされ得、ここで、「 1 」は正を示し、「 0 」は負を示す。ゼロ値を有するサンプルベクトル 6 0 4 A のサンプルについての符号は、符号ビット 1 0 0 8 においてシグナリングされない。

20

【 0 0 8 4 】

[0100] いくつかの実施形態では、コーダ 7 0 2 は、サンプルベクトル 6 0 4 のサンプルの大きさまたは絶対部分に対して（例えば、図 7 - 8 に例示されるような）ベクトルベースエントロピーコード化を使用して、サンプルベクトル 6 0 4 A をコード化し得る。サンプルベクトル 6 0 4 A 中のサンプルの絶対値は、本明細書で開示されるベクトルベース技法を使用してコード化され得る。

【 0 0 8 5 】

[0101] 例えば、4 つのサンプル S 0、S 1、S 8、および S 9 を備えるサンプルベクトル 6 0 4 A の場合、ベクトルベースエントロピーコーダ 7 0 2 は、プレフィックス部分 1 0 0 4 およびサフィックス部分 1 0 0 6 を作り出すために、| S 0 |、| S 1 |、| S 8 |、| S 9 | のような、ベクトルベース E C をグループ中のサンプルの絶対値に適用し得る。プレフィックス部分 1 0 0 4 は、使用される L U T のタイプを示し得、サフィックス部分 1 0 0 6 は、サンプルベクトル 6 0 4 A のすべてのサンプルの絶対値が決定され得る codeNumber に対応し得る。

30

【 0 0 8 6 】

[0102] 他方で、コーダ 7 0 2 は、サンプル S 0、S 1、S 8、および S 9 の符号ビット 1 0 0 8 を、ベクトルベースエントロピーコード化を使用することなく、別々にシグナリングし得る。例えば、符号ビット 1 0 0 8 は、固定長のコードを使用してシグナリングされ得、ここで、非ゼロサンプルごとに 1 ビットが、このサンプルが正であるか負であるかを示すために、シグナリングされる。いくつかの実施形態では、符号ビットは、対応するサンプル値がゼロであるときシグナリングされない。

40

【 0 0 8 7 】

[0103] 図 1 0 B は、いくつかの実施形態に係る、符号絶対値表現を使用してサンプルベクトル 6 0 4 をコード化するための代替の配列を例示する。いくつかの実施形態では、コーダ 7 0 2 は、複数のサンプルベクトル 6 0 4 A 乃至 6 0 4 D の各々についてのプレフィックス部分 1 0 0 4 およびサフィックス部分 1 0 0 6 がコード化された後に、ビットストリームにおいて複数のサンプルベクトル 6 0 4 A 乃至 6 0 4 D についての符号ビット 1 0 0 8 をコード化し得る。例えば、いくつかの実施形態では、コーダ 7 0 2 は、最初に、ビットストリームにおいてすべてのサンプルベクトル 6 0 4 A - D のプレフィックス部分 1

50

004およびサフィックス部分1006をコード化し得、すべてのグループ604A-Dの符号ビット1008は、すべてのプレフィックス部分1004およびサフィックス部分1006がコード化された後に、ビットストリームにおいてコード化され得る。

【0088】

[0104]2の補数表現と比べて、符号絶対値表現がもたらす利点は、値がゼロであるシンボルについての符号情報がシグナリングされないことである。したがって、符号絶対値表現は、例えば、ブロック予測および変換モードのような、ゼロ値がより確からしいモードにおいて、優れたコード化パフォーマンスに帰着することができる。

【0089】

[0105]上で説明したように、コーダ702は、サンプルベクトル604をコード化するために符号絶対値表現を使用し得、ここにおいて、ゼロの値に対応するサンプルベクトル604のサンプルのための符号ビットは、符号ビット1008においてシグナリングされない。このように、デコーダ30のパーサロジックは、ビットストリームから、サンプルベクトル604のサンプルの各々について符号ビット1008からの符号情報を読み取るかどうかを知るために、(サフィックス部分1006としてコード化された)サンプルベクトル604のサンプルを再構築または復号する必要があるとあり得る。いくつかの実施形態では、デコーダ30のパーサは、サブストリームデマルチプレクサ160の一部としてインプリメントされ得る。

【0090】

[0106]換言すると、デコーダ30のパーサロジックは、それがサンプルベクトル604についての符号ビット1008をパースし得る前に、サンプルベクトル604の各サンプルがゼロ値を有するか非ゼロ値を有するかを知る必要がある。サンプル値が非ゼロである場合、デコーダ30は、符号ビット1008の一部としてビットストリームからサンプルについての符号情報をパースする。そうではなく、サンプル値がゼロである場合、符号ビットは、ビットストリームから読み取られない。デコーダ30の(例えば、エントロピーデコーダ165としてインプリメントされた、サンプルを復号するための)復号機能とパース機能との間のこの依存性により、デコーダ30の最大デコーダスループットは、いくつかの実施形態では低減され得る。

【0091】

[0107]いくつかの実施形態では、ビデオデータビットストリームから、コード化されたサンプルベクトルを受け取ることに応答して、パーサは、単項コードであり得る、受け取ったコード化されたサンプルベクトルのプレフィックスを、このサンプルベクトルの各サフィックス部分の長さを決定するために、復号し得る。次いで、パーサは、エントロピーデコーダ165によって復号するために、サンプルベクトルのサフィックスに対応する固定数のビットを取り出し、次いで、次のクロックサイクル中に次のサンプルベクトルへと進み得る。

【0092】

[0108]デコーダ(例えば、エントロピーデコーダ165)は、コード化されたサンプルベクトルのサフィックスに対応するビットをパーサから受け取る。次いで、デコーダ165は、codeNumber708を決定するために、受け取ったビットを復号し得る。次いで、デコーダ165は、LUTを使用して、codeNumber708をインデックス値706にマッピングし得、これは、次に、サンプルベクトルの実際のサンプル値を取得するために使用される。いくつかの実施形態では、この復号は、コード化されたサンプルベクトルをパーサがパースした後の次のクロックサイクル(例えば、パーサが後続のサンプルベクトルをパースしているのと同じクロックサイクル)において実行され得る。

【0093】

[0109]図11は、いくつかの実施形態に係る、符号絶対値を使用してコード化されたサンプルベクトル604A-Dをパースおよび復号するために必要とされ得る多数のクロックサイクルを例示する例示的なグラフ1100を例示する。いくつかの実施形態では、デコーダ30は、(プレフィックス部分1004が、サフィックス部分1006中のビット

10

20

30

40

50

数を直接識別するであろうことから)単一のクロックにおいて各サンプルベクトル604のプレフィックス部分1004およびサフィックス部分1006をパースすることができ得る。図10Aに例示されるようなサンプルベクトル604A-604Dを考慮すると、いくつかの実施形態では、デコーダ30は、4つのサンプルベクトル604A乃至604Dをパースするのに少なくとも5つのクロックサイクルを必要とし得る。

【0094】

[0110]クロックサイクル1において、デコーダ30は、サンプルベクトル604Aのプレフィックス部分およびサフィックス部分をパースする。クロックサイクル2において、デコーダ30は、サンプルベクトル604Bのプレフィックス部分およびサフィックス部分をパースする。それと同時に、デコーダ30は、サンプルベクトル604Aの絶対サンプル値を復号し、サンプルベクトル604Aについての符号ビットをパースする。クロックサイクル3において、デコーダ30は、サンプルベクトル604Bの絶対サンプル値を復号し、サンプルベクトル604Bの符号ビットだけでなく、サンプルベクトル604Cのプレフィックス部分およびサフィックス部分をパースする。クロックサイクル4において、デコーダ30は、サンプルベクトル604Cのサンプル値を復号し、サンプルベクトル604Cの符号ビットおよびサンプルベクトル604Dのプレフィックスおよびサフィックスをパースする。デコーダ30がサンプルベクトル604Dの絶対値を復号して、サンプルベクトル604Dについての対応する符号情報をパースするのに、追加のクロックサイクル(クロックサイクル5)が必要とされる。したがって、図10Aに例示されるようなこの表現は、いくつかの実施形態では、デコーダ20の達成可能な最大スループットを低減し得る。各サンプルベクトルの絶対サンプル値がベクトルエントロピーコード化を使用してコード化されるとき、同様の結果が導き出され得る。

【0095】

[0111]いくつかの実施形態では、図10Bに例示されたように、複数のサンプルベクトル604A-Dに対応する符号ビット1008は、ビットストリームにおいてグループ604A-Dのプレフィックス部分1004およびサフィックス部分1006の後に位置付けられ得る。これらのケースでは、デコーダ30は、複数のサンプルベクトルのうちの最後のサンプルベクトル(例えば、グループ604D)のプレフィックスおよびサフィックスをパースしてから1クロックサイクル後に、符号ビット1008をパースし得る。このように、複数のサンプルベクトル604A-Dのすべてのサンプルベクトルが、符号絶対値を使用してコード化されるとき、デコーダ30は、サンプルベクトル604A-Dのプレフィックスおよびサフィックスをパースした後に、符号ビット1008をパースするために余分な(extra)クロックサイクル(例えば、クロックサイクル5)を必要とし得、これは、デコーダ165の達成可能な最大スループットを低減する可能性がある。例えば、n個のグループ604をパースするために、デコーダ30は、n+1個のクロックサイクルを必要とし得る。

【0096】

[0112]図12は、(例えば、図6に例示されるような)例示的なサンプルベクトル604のハイブリッドエントロピーコード化の例を例示す。いくつかの実施形態では、デコーダ30のスループットを増やすために、コーダ702は、ハイブリッド法を使用し得る。例えば、コーダ702は、符号絶対値表現を使用して、初期数のサンプルベクトル604(例えば、サンプルベクトル604A-C)をコード化し得る。他方では、コーダ702は、2の補数表現のような異なる表現を使用して、残りのサンプルベクトル(例えば、サンプルベクトル604D)をコード化し得る。コーダ702は、(図4-5に例示されるような)DSU-VLCまたは(例えば、図7-9に例示されるような)ベクトルECに基づいて、サンプルベクトル604A-Dの各々をコード化し得る。例えば、2の補数表現のためのベクトルECは、サンプルベクトル604のシンボルが単一のcodeNumber708にマッピングされる図8に基づき得る。他方で、符号絶対値表現では、コーダ702は、サンプルベクトル604についての符号ビット1008がサンプルベクトル604の非ゼロシンボルごとに別々にシグナリングされ得ると同時に、サンプルベクトル604の

10

20

30

40

50

各シンボルの絶対値のセットを単一のcodeNumber 7 0 8 にマッピングし得る。

【 0 0 9 7 】

[0113] 4つのサンプルベクトル 6 0 4 A - D をコード化するための1つの例示的な実施形態では、符号絶対値表現が、最初の3つのサンプルベクトル 6 0 4 A - C に対して使用され、2の補数表現が最後のサンプルベクトル 6 0 4 D に対して使用される。図 1 3 は、いくつかの実施形態に係る、図 1 2 に例示されたハイブリッドコード化スキームを使用してコード化されたサンプルベクトル 6 0 4 A - D をパースおよび復号するのに必要とされ得る多数のクロックサイクルを例示する例示的なグラフ 1 3 0 0 を例示する。上で説明したようにパーサおよびシンボルデコーダ 1 6 5 をパイプラインすることで、第 4 のクロックサイクルの間に、サンプルベクトル 6 0 4 C のサンプルが復号されるのと同時に、サンプルベクトル 6 0 4 D のサンプルが、デコーダ 1 6 5 によってパースされる。サンプルベクトル 6 0 4 D のサンプルは2の補数表現に基づいてコード 7 0 2 によって符号化されるため、デコーダ 1 6 5 が、サンプルベクトル 6 0 4 D についての符号ビットをパースするためにサンプルベクトル 6 0 4 D の絶対サンプル値を復号する必要はない。したがって、ハイブリッド法を使用すると、デコーダ 3 0 が、4つのクロックサイクルにおいて4つすべてのサンプルベクトル 6 0 4 A - D をパースすることが可能であり得る。ゆえに、いくつかの実施形態では、n個のサンプルベクトルをパースするときに余分なクロックサイクルが必要となるのを回避するために、コード 7 0 2 は、符号絶対値表現を使用して最初の n - 1 個のサンプルベクトルをコード化し、それと同時に、2の補数表現のような異なる表現を使用して最後のサンプルベクトルをコード化し得る。ゆえに、デコーダ 3 0 は、n 個のクロックサイクルにおいて n 個のサンプルベクトルをパースすることができ得る。

【 0 0 9 8 】

[0114] いくつかの実施形態では、コード 7 0 2 は、非ゼロサンプル値を有するサンプルベクトル 6 0 4 の数に基づいて、2の補数表現を使用するか符号絶対値表現を使用するかを決定し得る。一例では、コード 7 0 2 は、前のサンプルベクトル 6 0 4 A 乃至 6 0 4 C の各々が少なくとも1つの非ゼロサンプル値を含む場合にのみ、最後のサンプルベクトル 6 0 4 D に対して2の補数表現を使用し得る。それ以外の場合、コード 7 0 2 は、最後のサンプルベクトル 6 0 4 D をコード化するために、符号絶対値表現を使用し得る。いくつかの実施形態では、最後のサンプルベクトル 6 0 4 D をコード化するために2の補数を使用することは、圧縮効率のかなりのロスを伴い得る。3つのサンプルベクトル 6 0 4 A 乃至 6 0 4 C の各々が少なくとも1つの非ゼロサンプル値を含むときにのみ2の補数で最後のサンプルベクトル 6 0 4 D をコード化する（ゆえに、符号ビット 1 0 0 8 のコード化を必要とする）ことによって、2の補数の使用は、所望のスループットを達成するために、必要なときにのみそれを使用することで最小限にされ得る。例えば、3つのサンプルベクトル 6 0 4 A 乃至 6 0 4 C のうちの1つでも少なくとも1つの非ゼロサンプル値を含まない場合、サンプルベクトルは、グループスキップモードを使用してコード化され得る。ゆえに、デコーダ 3 0 は、サンプルベクトル 6 0 4 A 乃至 6 0 4 C をより迅速にパースすることができ得、これにより、4つすべてのサンプルベクトルが4つのクロックサイクル内にパースされることができ得る。

【 0 0 9 9 】

[0115] 上の例では、D S U - V L C ベースエントロピーコード化は、符号絶対値表現および2の補数表現の両方に対して使用される。しかしながら、同じ技法は、両方の表現についてのベクトル E C に拡張されることができ得る。

【 0 1 0 0 】

プロセスフロー

[0116] 本開示の態様が、図 2 A のビデオエンコーダ 2 0 のようなエンコーダまたは図 8 に例示されたベクトルベースエントロピーコード 8 0 4 のようなエンコーダの一部の観点から説明されていることは留意されるべきである。上で説明したものの逆動作が、例えば、図 2 B のビデオデコーダ 3 0 によって、生成されたビットストリームを復号するために適用され得る。

【 0 1 0 1 】

[0117]例えば、図 1 4 A は、サンプルベクトルデータを符号化するためのプロセスのフローチャートを例示し、図 1 4 B は、サンプルベクトルデータに対応する受け取ったビットシーケンスを復号するためのプロセスのフローチャートを例示する。例えば、図 1 4 A のブロック 1 4 0 2 において例示されるように、エンコーダは、受け取ったサンプルベクトルをインデックス値に変換する。いくつかの実施形態では、異なるサンプルベクトルが 2 つとして同じインデックス値を有することがないように、サンプルベクトルとインデックス値との間には 1 対 1 の対応が存在する。ブロック 1 4 0 4 において、エンコーダは、LUT を使用して、インデックス値をコード番号にマッピングする。LUT は、可能なサンプルベクトルのグループのうちの所与のサンプルベクトルの生起確率に少なくとも部分的に基づいて構築され得る。いくつかの実施形態では、エンコーダは、サンプルベクトルをコード化するために使用されるコード化モード、サンプルベクトルの各サンプル値を表すのに必要なビット数、サンプルベクトル中のサンプル値の数 (a number of)、および / または同様のものに基づいて LUT を識別し得る。ブロック 1 4 0 6 において、エンコーダは、ビットストリームの一部として記憶または送信されるべき、コード化されたビットのシーケンスを形成するために、VLC スキームを使用して、コード番号をコード化し得る。

10

【 0 1 0 2 】

[0118]図 1 4 B のブロック 1 4 1 2 に例示されるように、デコーダは、コード番号を形成するために、VLC スキームを使用して、(例えば、受け取ったビットストリームから)受け取ったビットのシーケンスを復号する。ブロック 1 4 1 4 において、デコーダは、LUT を使用して、コード番号をインデックス値にマッピングする。いくつかの実施形態では、デコーダは、ビットのシーケンスの一部として(例えば、このシーケンスのプレフィックス部分として)受け取ったインジケーションに基づいて、このマッピングを実行するために使用される LUT タイプを識別する。ブロック 1 4 1 6 において、デコーダは、インデックス値をサンプルベクトルについての値へと変換する。

20

【 0 1 0 3 】

[0119]図 1 5 は、ハイブリッドコード化を使用してサンプルベクトルデータを符号化するための例示的なプロセスのフローチャートを例示する。ブロック 1 5 0 2 において、エンコーダは、複数のサンプル値を受け取り、受け取ったサンプル値を n 個のサンプルベクトルへと分ける。ブロック 1 5 0 4 において、エンコーダは、符号絶対値表現を使用して、 n 個のサンプルベクトルの第 1 の部分をコード化する。いくつかの実施形態では、サンプルベクトルの第 1 の部分は、 n 個のサンプルベクトルのうちの最初の $n - 1$ 個のサンプルベクトルを備え得る。ブロック 1 5 0 6 において、エンコーダは、2 の補数表現を使用して、 n 個のサンプルベクトルのうちの 1 つまたは複数の残りのサンプルベクトルをコード化し得る。いくつかの実施形態では、1 つまたは複数の残りのサンプルベクトルは、 n 個のサンプルベクトルのうちの最後のサンプルベクトル (the last sample vectors) を備え得る。いくつかの実施形態では、エンコーダは、サンプルベクトルの第 1 の部分の各々が、非ゼロ値を有する少なくとも 1 つのサンプルを含む場合にのみ、2 の補数表現で、1 つまたは複数の残りのサンプルベクトルをコード化する。 n 個のサンプルベクトルのうちのサンプルベクトルの第 1 の部分を符号絶対値表現を使用しておよび残りのグループを 2 の補数表現を使用してコード化することで、 n 個のサンプルベクトルのすべてのサンプル値の絶対値が復号された後に非ゼロサンプル値の符号ビットを復号することが必要となることで引き起こされ得るデコーダスループットの低減を回避しつつ、符号絶対値表現を使用してコード化することの利点の実現され得る。

30

40

【 0 1 0 4 】

他の検討事項

[0120]本明細書で開示された情報および信号は、多種多様な技術および技法のうちの任意のものを使用して表され得る。例えば、上の説明全体にわたって参照され得るデータ、命令、コマンド、情報、信号、ビット、シンボル、およびチップは、電圧、電流、電磁波

50

、磁場または磁性粒子、光場または光粒子、またはこれらの任意の組合せによって表され得る。

【 0 1 0 5 】

[0121] 本明細書で開示された実施形態と関連して説明された実例となる様々な論理ブロックおよびアルゴリズムステップは、電子ハードウェア、コンピュータソフトウェア、または両方の組合せとしてインプリメントされ得る。このハードウェアとソフトウェアとの互換性を明確に例示するために、実例となる様々な構成要素、ブロック、およびステップが、概してそれらの機能性の観点から上で説明されている。このような機能性がハードウェアとしてインプリメントされるかソフトウェアとしてインプリメントされるかは、特定の用途とシステム全体に課せられる設計制約とに依存する。当業者は、説明された機能性を特定の用途ごとに様々な方法でインプリメントし得るが、このようなインプリメンテーションの決定は、本開示の範囲からの逸脱をさせるものとして解釈されるべきでない。

10

【 0 1 0 6 】

[0122] 本明細書で説明された技法は、ハードウェア、ソフトウェア、ファームウェア、またはこれらの任意の組合せでインプリメントされ得る。そのような技法は、汎用コンピュータ、ワイヤレス通信デバイスハンドセット、または、ワイヤレス通信デバイスハンドセット、自動車、電気器具、ウェアラブルなもの、および/または他のデバイスにおけるアプリケーションを含む複数の用途を有する集積回路デバイスのような様々なデバイスの任意のものでインプリメントされ得る。デバイスまたは構成要素として説明された任意の特徴は、1つの集積論理デバイスでまとめて、または、ディスクリートではあるが相互動作可能な複数の論理デバイスとして別々にインプリメントされ得る。ソフトウェアでインプリメントされる場合、本技法は、少なくとも部分的に、命令を含むプログラムコードを備えるコンピュータ読取可能なデータ記憶媒体によって実現され得、これらの命令は、実行されると、上で説明した方法のうちの1つまたは複数を実行する。コンピュータ読取可能なデータ記憶媒体は、パッケージングマテリアルを含み得る、コンピュータプログラム製品の一部を形成し得る。コンピュータ読取可能な媒体は、同期動的ランダムアクセスメモリ (SDRAM) のようなランダムアクセスメモリ (RAM)、読取専用メモリ (ROM)、不揮発ランダムアクセスメモリ (NVRAM)、電氣的消去可能なプログラマブル読取専用メモリ (EEPROM (登録商標))、FLASHメモリ、磁気または光学データ記憶媒体、等のメモリまたはデータ記憶媒体を備え得る。本技法は、追加的にまたは代替的に、少なくとも部分的に、伝播される信号または波のような、コンピュータによってアクセス、読み取り、および/または実行されることができ、かつ、データ構造または命令の形式でプログラムコードを搬送または通信するコンピュータ読取可能な通信媒体によって実現され得る。

20

30

【 0 1 0 7 】

[0123] プログラムコードは、1つまたは複数のデジタルシグナルプロセッサ (DSP)、汎用マイクロプロセッサ、特定用途向け集積回路 (ASIC)、フィールドプログラマブル論理アレイ (FPGA)、または他の同等の集積回路またはディスクリート論理回路のような1つまたは複数のプロセッサを含み得るプロセッサによって実行され得る。そのようなプロセッサは、本開示で説明された技法のうちの任意のものを実行するように構成され得る。汎用プロセッサは、マイクロプロセッサであり得るが、代替的に、プロセッサは、任意の従来のプロセッサ、コントローラ、マイクロコントローラ、またはステートマシンであり得る。プロセッサはまた、コンピューティングデバイスの組合せ、例えば、DSPと、1つのマイクロプロセッサ、複数のマイクロプロセッサ、DSPコアに連結した1つまたは複数のマイクロプロセッサ、または任意の他のそのような構成との組合せとしてインプリメントされ得る。したがって、「プロセッサ」という用語は、本明細書で 사용되는場合、前述の構造の任意のもの、前述の構造の任意の組合せ、または本明細書で説明された技法のインプリメンテーションに適した任意の他の構造または装置を指し得る。加えて、いくつかの態様では、本明細書で説明された機能性は、符号化および復号するために構成された専用のソフトウェアまたはハードウェア内に提供され得るか、または、複

40

50

合ビデオエンコーダ - デコーダ (CODEC) に組み込まれ得る。また、本技法は、1つまたは複数の回路または論理要素で完全にインプリメントされ得る。

【0108】

[0124]本開示の技法は、ワイヤレスハンドセット、IC、またはICのセット（例えば、チップセット）を含む、幅広い種類のデバイスまたは装置でインプリメントされ得る。様々な構成要素またはユニットは、開示された技法を実行するように構成されたデバイスの機能的な態様を強調するように本開示では説明されているが、必ずしも異なるハードウェアユニットによる実現を必要とするわけではない。むしろ、上で説明したように、様々なユニットは、コーデックハードウェアユニットへと組み合わせられるか、適切なソフトウェアおよび/またはファームウェアと併せて、上で説明した1つまたは複数のプロセッサを含む、相互動作するハードウェアユニットの集合体によって提供され得る。

10

【0109】

[0125]前述のものは、様々な異なる実施形態に関連して説明されているが、1つの実施形態からの特徴または要素は、本開示の教示から逸脱することなく、他の実施形態と組み合わせられる。しかしながら、それぞれの実施形態間での特徴の組み合わせは、必ずしもそれらに限定されるわけではない。本開示の様々な実施形態が説明されている。これらのおよび他の実施形態は、以下の特許請求の範囲の範囲内である。

以下に本願発明の当初の特許請求の範囲に記載された発明を付記する。

[C1]

複数のビデオサンプルを有するビデオ情報をコード化するための装置であって、

20

1つのグループに配列される複数のビデオサンプルを記憶するように構成されたメモリと、各ビデオサンプルは、ビットサイズBを有し、前記グループは、グループサイズKを有する、

ハードウェアプロセッサと、前記プロセッサは、可変長コード化(VLC)スキームを使用して、ベクトルベースコードを形成するために前記グループ中の前記複数の前記ビデオサンプルをコード化するように構成され、前記ベクトルベースコードは、少なくとも、前記ビットサイズBまたは前記グループサイズKのうちの少なくとも1つに対応するルックアップテーブルのタイプを識別する第1の部分と、前記グループ中の前記複数のビデオサンプルを表す第2の部分とを備える、

を備える装置。

30

[C2]

前記プロセッサは、

前記グループをインデックス値に変換することと、

前記グループに関連付けられた前記グループサイズKまたは前記ビットサイズBのうちの少なくとも1つに対応する前記タイプのルックアップテーブルを使用して、前記インデックス値をコード番号にマッピングすることと、

前記VLCスキームを使用して、前記ベクトルベースコードを形成するために前記コード番号を符号化することと

を行うようにさらに構成される、C1に記載の装置。

40

[C3]

前記プロセッサは、

可変長コード化(VLC)スキームを使用して、コード番号を形成するために前記ベクトルベースコードを復号することと、

前記ベクトルベースコードの前記第1の部分によって識別される前記タイプのルックアップテーブルを使用して、前記コード番号をインデックス値にマッピングすることと、

前記インデックス値を前記グループの前記複数のビデオサンプルに変換することと

を行うようにさらに構成される、C1に記載の装置。

[C4]

前記第1の部分は、単項コードフォーマットを使用してコード化される、C1に記載の装置。

50

[C 5]

前記複数のビデオサンプルは、2の補数表現を使用して表される、C 1に記載の装置。

[C 6]

前記プロセッサが、前記ベクトルベースコードを形成するために前記グループ中の前記複数のビデオサンプルの前記絶対値をコード化するように構成されるような、符号絶対値表現を使用した前記複数のビデオサンプル、C 1に記載の装置。

[C 7]

前記プロセッサ回路は、前記ビットサイズBが閾値より小さいとき、前記ベクトルベースコードを供給するように構成される、C 1に記載の装置。

[C 8]

前記プロセッサ回路は、第2のVLCスキームを使用して、前記ベクトルベースコードの前記第1の部分と、少なくとも前記第2の部分のプレフィックス部分とを符号化するようにさらに構成される、C 1に記載の装置。

[C 9]

前記タイプのルックアップテーブルは、可能なKサイズのグループのセットのうちの前記グループの生起確率に少なくとも部分的に基づく、C 1に記載の装置。

[C 10]

ビデオ情報をコード化するための方法であって、前記情報は、サンプルのセットを備え、前記方法は、

前記サンプルのセットを1つまたは複数のサンプルベクトルへと区分化することと、前記1つまたは複数のサンプルベクトルのうちの1つのサンプルベクトルは、ビットサイズBに関連付けられたK個のサンプルを備える、

ルックアップテーブルを使用して、前記サンプルベクトルをコード番号にマッピングすることと、前記ルックアップテーブルは、可能な長さKのサンプルベクトルのセットのうちの前記サンプルベクトルの生起確率に少なくとも部分的に基づく、

前記K個のサンプルを表す可変長コード化(VLC)コードを生成するために、VLCスキームを使用して、前記コード番号をコード化することと、前記VLCコードは、少なくとも、前記ルックアップテーブルを識別する第1の部分と、前記サンプルベクトルの前記K個のサンプルを表す第2の部分とを備える、

を備える方法。

[C 11]

サンプルは、画素の色成分に対応する、C 10に記載の方法。

[C 12]

前記サンプルのセットは、ビデオ情報のブロックに関連付けられた量子化された残余に対応する、C 10に記載の方法。

[C 13]

前記VLCスキームは、ハフマン、算術、指数ゴロム(EG)、およびゴロムライス(GR)のうちの少なくとも1つを備える、C 10に記載の方法。

[C 14]

前記サンプルベクトルをコード番号にマッピングすることは、前記サンプルベクトルを、前記サンプルベクトルに対応する一意的なインデックス値に変換することと、前記ルックアップテーブルを使用して、前記インデックス値を前記コード番号にマッピングすることとを備える、C 10に記載の方法。

[C 15]

前記LUTは、前記ビットサイズBまたは前記長さKのうちの少なくとも1つに基づいて選択される、C 10に記載の方法。

[C 16]

前記第2の部分は、少なくともプレフィックス部分およびサフィックス部分を備え、前記方法は、VLCスキームを使用して、前記第1の部分および前記プレフィックス部分をコード化することをさらに備える、C 10に記載の方法。

10

20

30

40

50

[C 1 7]

前記第 1 の部分は、単項フォーマットで表される、C 1 0 に記載の方法。

[C 1 8]

前記サンプルのセットは、2 の補数表現を使用して表される、C 1 0 に記載の方法。

[C 1 9]

複数のビデオサンプルを有するビデオ情報をコード化するための装置であって、
送信のために複数のグループへと分けられる複数のビデオサンプルを記憶するように構成されたメモリと、各グループは、1 つまたは複数のビデオサンプルを備える、
ハードウェアプロセッサと、前記プロセッサは、ビデオサンプルの前記複数のグループをコード化するように構成される、
ここにおいて、前記複数のグループのうちの第 1 の複数のグループからのビデオサンプルは、符号絶対値フォーマットを使用してコード化され、
前記複数のグループのうちの 1 つまたは複数の残りのグループは、2 の補数フォーマットを使用してコード化される、
を備える装置。

10

[C 2 0]

前記複数のグループは、n 個のグループを備え、前記第 1 の複数のグループは、前記複数のグループのうちの最初の n - 1 個のグループを備える、C 1 9 に記載の装置。

[C 2 1]

前記 1 つまたは複数の残りのグループは、前記第 1 の複数のグループの各グループの少なくとも 1 つのサンプル値が非ゼロ値に対応する場合にのみ、2 の補数フォーマットを使用してコード化される、C 1 9 に記載の装置。

20

【図 1 A】

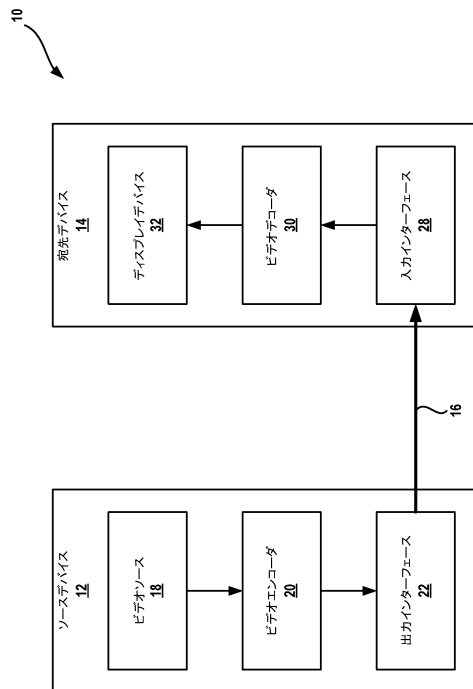


FIG. 1A

【図 1 B】

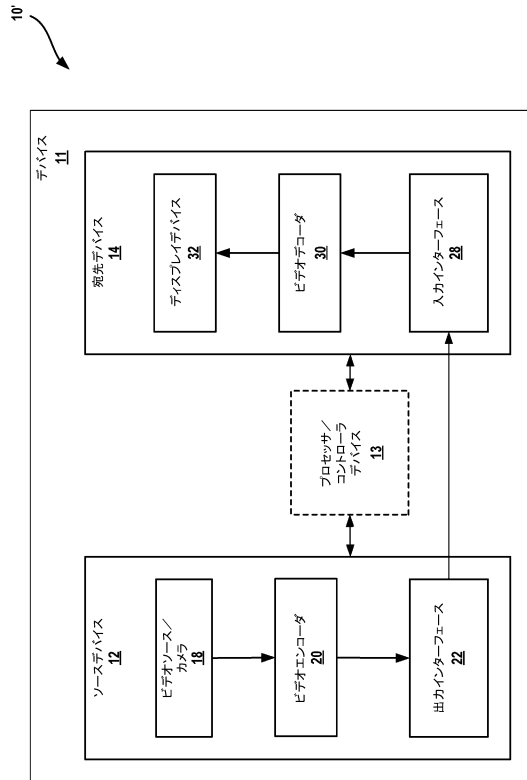


FIG. 1B

【 図 2 A 】

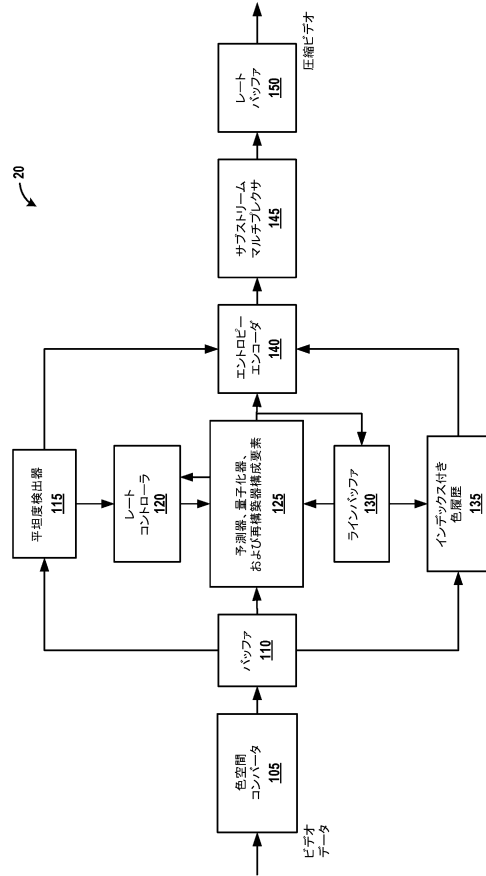


FIG. 2A

【 図 2 B 】

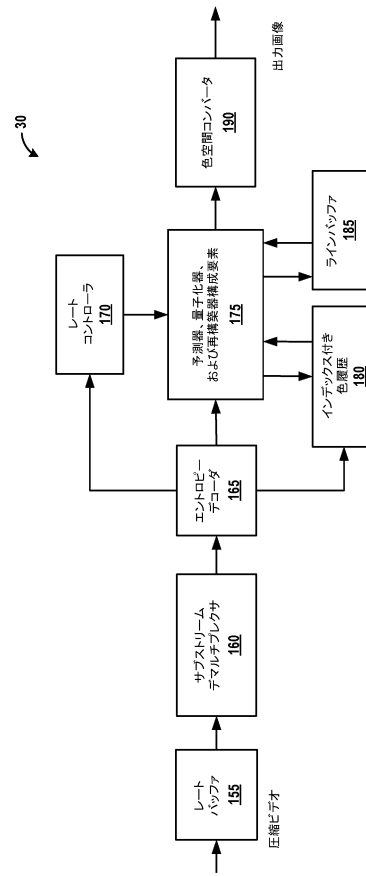


FIG. 2B

【 図 3 】

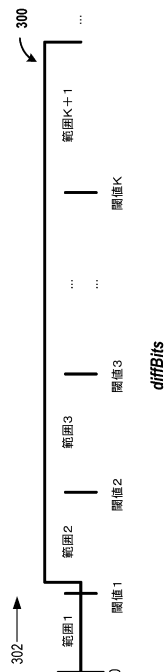


FIG. 3

【 図 4 】

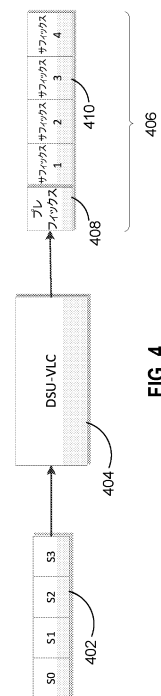


FIG. 4

【図 5】

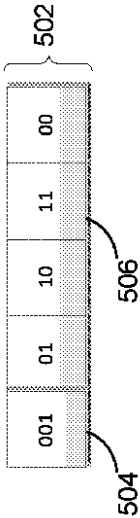


FIG. 5

【図 6】

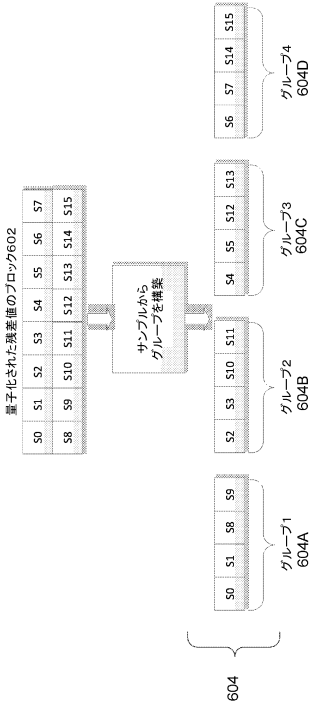


FIG. 6

【図 7】

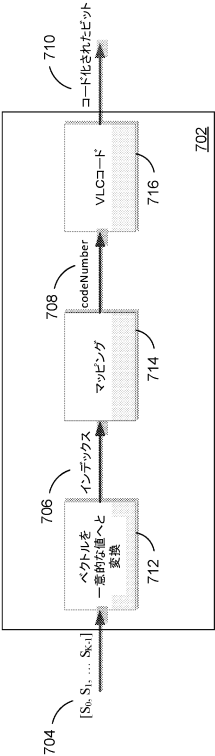


FIG. 7

【図 8】

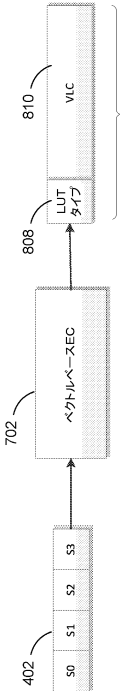


FIG. 8

【図 9】

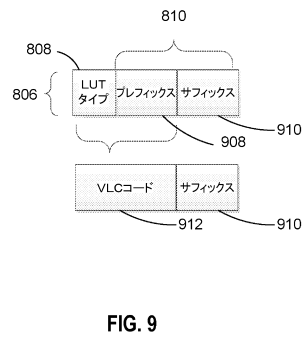


FIG. 9

【図 10 A】

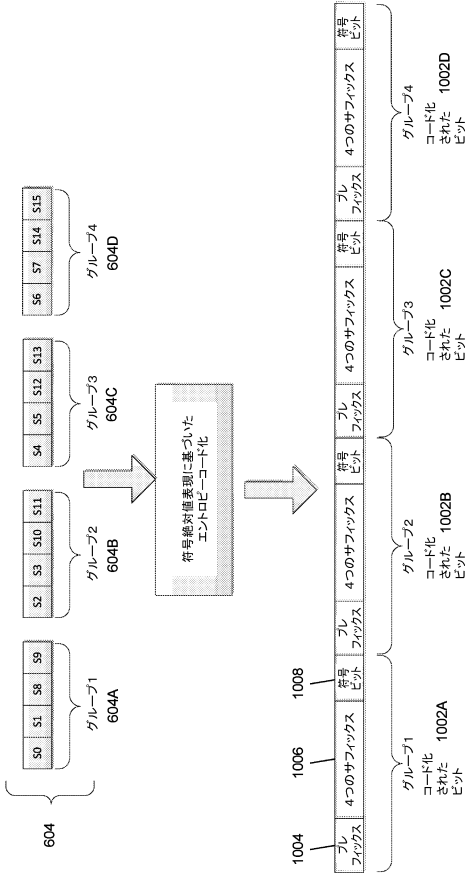


FIG. 10A

【図 10 B】

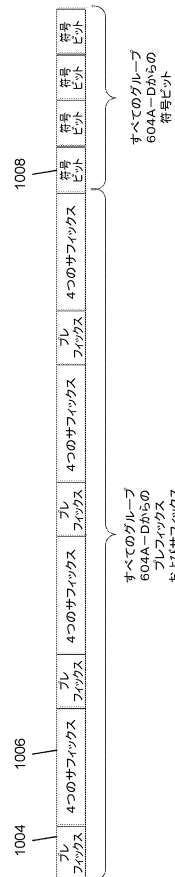


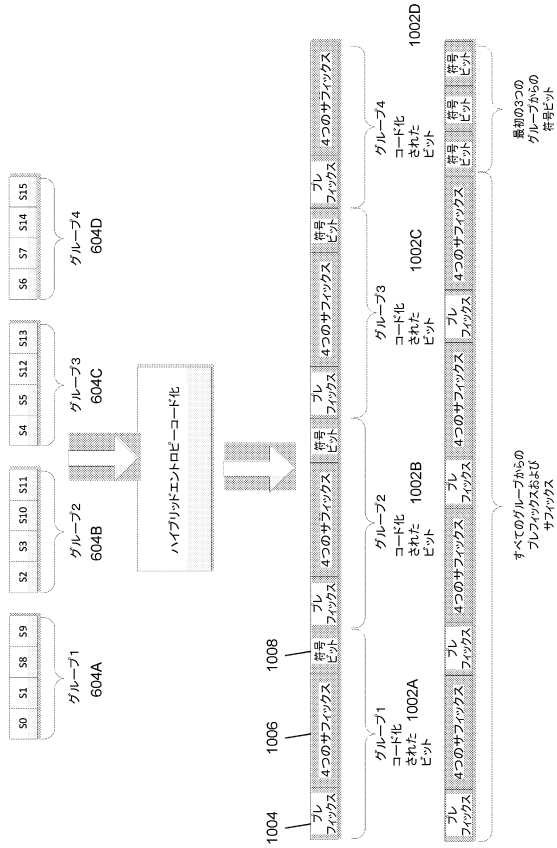
FIG. 10B

【図 11】

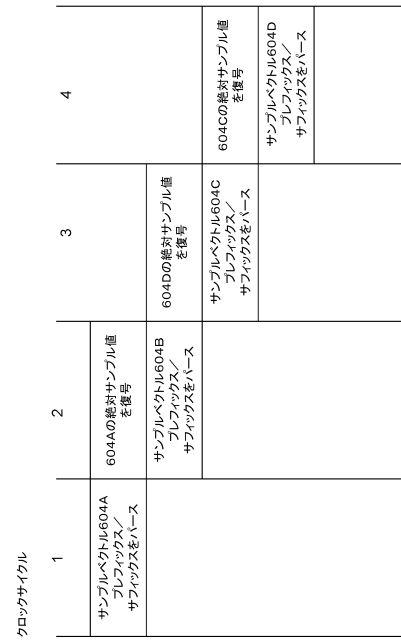
クロックサイクル		1	2	3	4	5
サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース
	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース	サンプリングレイト604A プレフィックス／ サフィックスをハース

FIG. 11

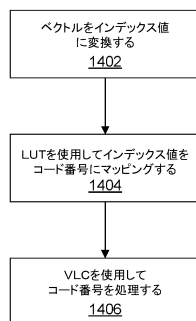
【圖 12】



【 図 1 3 】



【 図 1 4 A 】



【 ㄨ 1 4 B 】



【図 15】

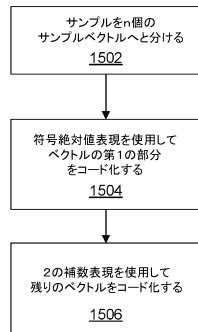


FIG. 15

フロントページの続き

- (31)優先権主張番号 15/449,755
(32)優先日 平成29年3月3日(2017.3.3)
(33)優先権主張国・地域又は機関
米国(US)

早期審査対象出願

- (74)代理人 100184332
弁理士 中丸 慶洋
(72)発明者 ティルマライ、ピジャヤラガバン
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
(72)発明者 ヤコブソン、ナタン・ハイム
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5
(72)発明者 ジョーシー、ラジャン・ラクスマン
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

審査官 岩井 健二

- (56)参考文献 国際公開第 2 0 1 5 / 1 9 1 5 3 5 (WO, A 2)
特表 2 0 1 3 - 5 0 8 8 6 7 (JP, A)
Bai-Jue Shieh, Terng-Yin Hsu and Chen-Yi Lee, A New Approach of Group-Based VLC Codec System, ISCAS 2000 - IEEE International Symposium on Circuits and Systems, IEEE, 2 0 0 0 年, pp.IV-609 - IV-612
Bai-Jue Shieh, Yew-San Lee and Chen-Yi Lee, A New Approach of Group-Based VLC Codec System with Full Table Programmability, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE, 2 0 0 1 年 2 月, VOL.11, NO.2, pp.210-221
Pei-Yin Chen, Yi-Ming Lin and Min-Yi Cho, An Efficient Design of Variable Length Decoder for MPEG-1/2/4, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE, 2 0 0 8 年 1 2 月, VOL.10, NO.7, pp.1307-1315
Frederick Walls and Alexander (Sandy) MacInnis, BDC-1: A ROBUST ALGORITHM FOR DISPLAY STREAM COMPRESSION, PCS 2013, IEEE, 2 0 1 3 年, pp.434-437

- (58)調査した分野(Int.Cl., DB名)
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8