

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7216351号
(P7216351)

(45)発行日 令和5年2月1日(2023.2.1)

(24)登録日 令和5年1月24日(2023.1.24)

(51)国際特許分類 F I
H 0 4 N 19/70 (2014.01) H 0 4 N 19/70
H 0 4 N 19/597(2014.01) H 0 4 N 19/597

請求項の数 12 (全23頁)

(21)出願番号	特願2021-535985(P2021-535985)	(73)特許権者	000002185 ソニーグループ株式会社 東京都港区港南1丁目7番1号
(86)(22)出願日	令和1年9月18日(2019.9.18)	(74)代理人	100092093 弁理士 辻居 幸一
(65)公表番号	特表2022-517060(P2022-517060 A)	(74)代理人	100109070 弁理士 須田 洋之
(43)公表日	令和4年3月4日(2022.3.4)	(74)代理人	100067013 弁理士 大塚 文昭
(86)国際出願番号	PCT/IB2019/057835	(74)代理人	100109335 弁理士 上杉 浩
(87)国際公開番号	WO2020/128650	(74)代理人	100120525 弁理士 近藤 直樹
(87)国際公開日	令和2年6月25日(2020.6.25)	(74)代理人	100151987 弁理士 谷口 信行
審査請求日	令和3年6月18日(2021.6.18)		
(31)優先権主張番号	62/781,968		
(32)優先日	平成30年12月19日(2018.12.19)		
(33)優先権主張国・地域又は機関	米国(US)		
(31)優先権主張番号	62/787,637		
(32)優先日	平成31年1月2日(2019.1.2)		
	最終頁に続く		最終頁に続く

(54)【発明の名称】 点群符号化構造

(57)【特許請求の範囲】

【請求項1】

装置の非一時的メモリにプログラムされた方法であって、
点群符号化構造を実装するステップと、
前記点群符号化構造を使用して点群メタデータを符号化するステップと、
を含み、
前記点群メタデータを符号化するステップは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタデータ情報とを使用することを含む双方向予測を利用し、
双方向予測は、前記過去のパッチ及びメタデータ情報のための第1の構造、並びに前記将来のパッチ及びメタデータ情報のための第2の構造という、参照パッチリストのための2つの別の構造を使用して実装され、
前記点群メタデータを符号化するステップは、過去及び将来のパッチを参照できるようにフレームを並べ替えることを含み、
前記点群の層のバッファリストを利用して、前記点群の点が互いに重なり合って投影されることを可能にし、ここで、前記点群の層は、第1の層が第1の点であり、第2の層が最後の点であり、これらの間の点は符号化されない2つの層に限定されるものであるステップをさらに含むことを特徴とする方法。

【請求項2】

前記点群符号化構造を実装するステップは、予測のために他のどのフレームにも依存しないキー点群フレームを決定するステップを含む、

請求項 1 に記載の方法。

【請求項 3】

前記第 1 の構造及び前記第 2 の構造は、点群パッチバッファに記憶される、
請求項 1 に記載の方法。

【請求項 4】

V - P C C のブロックを相関させるシグナリング法が実装されるようにされたものである、

請求項 1 に記載の方法。

【請求項 5】

点群符号化構造を実装し、

前記点群符号化構造を使用して点群メタデータを符号化し、

前記点群の層のバッファリストを利用して、前記点群の点が互いに重なり合って投影されることを可能にする、

ためのアプリケーションを記憶する非一時的メモリと、

前記メモリに結合されて前記アプリケーションを処理するように構成されたプロセッサと、

を備え、

前記点群メタデータを符号化することは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタデータ情報とを使用することを含む双方向予測を利用し、

双方向予測は、前記過去のパッチ及びメタデータ情報のための第 1 の構造、並びに前記将来のパッチ及びメタデータ情報のための第 2 の構造という、参照パッチリストのための 2 つの別の構造を使用して実装され、

前記点群メタデータを符号化することは、過去及び将来のパッチを参照できるようにフレームを並べ替えることを含み、

前記点群の層は、第 1 の層が第 1 の点であり、第 2 の層が最後の点であり、これらの間の点は符号化されない 2 つの層に限定されるものであることを特徴とする装置。

【請求項 6】

前記点群符号化構造を実装することは、予測のために他のいずれのフレームにも依存しないキーポイントフレームを決定することを含む、

請求項 5 に記載の装置。

【請求項 7】

前記第 1 の構造及び前記第 2 の構造は、点群パッチバッファに記憶される、
請求項 5 に記載の装置。

【請求項 8】

V - P C C のブロックを相関させるシグナリング法が実装されるようにされたものである、

請求項 5 に記載の装置。

【請求項 9】

点群符号化構造を実装する手段と、

前記点群符号化構造を使用して点群メタデータを符号化するエンコーダと、
を備え、

前記点群メタデータを符号化することは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタデータ情報とを使用することを含む双方向予測を利用し、

双方向予測は、前記過去のパッチ及びメタデータ情報のための第 1 の構造、並びに前記将来のパッチ及びメタデータ情報のための第 2 の構造という、参照パッチリストのための 2 つの別の構造を使用して実装され、

前記メタデータ情報は、時間的相関情報を含み、参照パッチリストが、柔軟な予測を実装するための所定の参照補助情報セットから生成され、

前記点群メタデータを符号化することは、過去及び将来のパッチを参照できるようにフレームを並べ替えることを含み、

10

20

30

40

50

前記エンコーダは、前記点群の層のバッファリストを利用して、前記点群の点が互いに重なり合って投影されることを可能にするようにさらに構成され、ここで、前記点群の層は、第1の層が第1の点であり、第2の層が最後の点であり、これらの間の点は符号化されない2つの層に限定されるものであることを特徴とするシステム。

【請求項10】

前記点群符号化構造を実装することは、予測のために他のいずれのフレームにも依存しないキーフレームを決定することを含む、請求項9に記載のシステム。

【請求項11】

前記第1の構造及び前記第2の構造は、点群パッチバッファに記憶される、請求項9に記載のシステム。

10

【請求項12】

V-PCCのブロックを相関させるシグナリング法が実装されるようにされたものである、請求項9に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

〔関連出願との相互参照〕

本出願は、2019年3月20日に出願された「点群符号化構造 (POINT CLOUD CODING STRUCTURE)」という名称の米国仮特許出願第62/821,139号、2019年1月11日に出願された「点群符号化構造」という名称の米国仮特許出願第62/791,328号、2019年1月2日に出願された「点群符号化構造」という名称の米国仮特許出願第62/787,637号、2018年12月19日に出願された「点群符号化構造」という名称の米国仮特許出願第62/781,968号の米国特許法第119条に基づく優先権を主張するものであり、これらの文献はその全体が全ての目的で引用により本明細書に組み入れられる。

20

【0002】

本発明は、3次元グラフィックスに関する。具体的には、本発明は、3次元グラフィックスの符号化に関する。

30

【背景技術】

【0003】

点群 (point clouds) は、3Dスキャナ、LIDARセンサによって取り込まれた、又は仮想現実/拡張現実 (VR/AR) などの大衆向け用途で使用される3Dデータの伝送フォーマット候補とみなされてきた。点群は、3D空間内の点の集合である。通常、各点は、空間位置 (X、Y、Z) 以外に、色 (R、G、B)、又は (LIDAR画像などにおける) 反射率及び時間的タイムスタンプなどの関連する属性を有する。装置は、標的3Dオブジェクトの高品質表現を取得するために、約数千個又は数百万個もの点の点群を取り込む。さらに、VR/AR用途で使用される動的3Dシーンでは、しばしば全ての単一フレームが固有の高密度な点群を有し、結果として毎秒数百万個もの点群が伝送されるようになる。このような大量データの伝送を実行できるように、しばしば圧縮が適用される。

40

【0004】

2017年に、MPEGは点群圧縮のための提案募集 (CFP) を行った。MPEGは、複数の提案を評価した後に、(八分木及び同様の符号化法に基づく) 3Dネイティブな符号化技術、又は3Dから2Dへの投影後における従来のビデオ符号化という2つの異なる技術を点群圧縮のために検討している。MPEGは、動的3Dシーンについては、パッチサーフェスマデリング (patch surface modeling) に基づくテストモデルソフトウェア (TMC2)、3Dから2D画像へのパッチ投影、及びHEVCなどのビデオエンコーダを用いた2D画像の符号化を使用する。この方法は、ネイティブな

50

3D符号化よりも効率的であることが証明されており、許容できる品質で優位性のあるビットレートを達成することができる。

【0005】

TMC2は、点群の符号化時に、2Dキャンバス画像内のパッチ位置及び境界ボックスサイズなどの、パッチ投影に関連する補助情報を符号化する。補助情報の時間符号化では、現在の点群からのパッチと復号直後の点群からのパッチとの間のパッチマッチングが予測に使用される。この手順は直近のものに限定され、シーケンス内の全てのフレームについてデルタ符号化(delta coding)を実行することを含む。

【発明の概要】

【課題を解決するための手段】

10

【0006】

点群符号化構造が、点群パッチの予測に使用される全ての参照が現在の点群フレームに制限される「キー」点群フレームを定める。参照パッチのリスト及びその点群フレームからのそれぞれの境界ボックスを別のフレーム内のパッチ予測に使用するために記憶する点群パッチバッファについて説明する。点群フレームの符号化順が提示順と異なることができる場合、参照パッチのリストは過去からのパッチ及び将来からのパッチを含むことができ、双方向予測が使用される。点群の層にも同様の参照バッファリストの概念を適用することができる。上位レベル情報を含むブロックのIDへの参照をペイロード内でシグナリングすることによってV-PPCのブロックを関連させるシグナリング法についても説明する。

20

【0007】

1つの態様では、装置の非一時メモリにプログラムされた方法が、点群符号化構造を実装するステップと、点群符号化構造を使用して点群メタデータを符号化するステップとを含む。点群符号化構造を実装するステップは、予測のために他のどのフレームにも依存しないキー点群フレームを決定するステップを含む。点群メタデータを符号化するステップは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタデータ情報とを使用することを含む双方向予測を利用する。双方向予測は、過去のパッチ及びメタデータ情報のための第1の構造、並びに将来のパッチ及びメタデータ情報のための第2の構造という2つの別の構造を実装する。第1の構造及び第2の構造は、点群パッチバッファに記憶される。方法は、点群の層のバッファリストを利用して、点群の点が互いに重なり合って投影されることを可能にするステップをさらに含む。方法は、V-PPCのブロックを関連させるシグナリング法を実装するステップをさらに含む。

30

【0008】

別の態様では、装置が、点群符号化構造を実装し、点群符号化構造を使用して点群メタデータを符号化するためのアプリケーションを記憶する非一時的メモリと、メモリに結合されてアプリケーションを処理するように構成されたプロセッサとを含む。点群符号化構造を実装することは、予測のために他のいずれのフレームにも依存しないキー点群フレームを決定することを含む。点群メタデータを符号化することは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタデータ情報とを使用することを含む双方向予測を利用する。双方向予測は、過去のパッチ及びメタデータ情報のための第1の構造、並びに将来のパッチ及びメタデータ情報のための第2の構造という2つの別の構造を実装する。第1の構造及び第2の構造は、点群パッチバッファに記憶される。方法は、点群の層のバッファリストを利用して、点群の点が互いに重なり合って投影されることを可能にすることをさらに含む。装置は、V-PPCのブロックを関連させるシグナリング法を実装することをさらに含む。

40

【0009】

別の態様では、システムが、点群符号化構造と、点群符号化構造を使用して点群メタデータを符号化するエンコーダとを含む。点群符号化構造を実装することは、予測のために他のいずれのフレームにも依存しないキー点群フレームを決定することを含む。点群メタデータを符号化することは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタ

50

データ情報とを使用することを含む双方向予測を利用する。双方向予測は、過去のパッチ及びメタデータ情報のための第1の構造、並びに将来のパッチ及びメタデータ情報のための第2の構造という2つの別の構造を実装する。第1の構造及び第2の構造は、点群パッチバッファに記憶される。システムは、点群の層のバッファリストを利用して、点群の点が互いに重なり合っって投影されることを可能にすることをさらに含む。システムは、V - P C Cのブロックを相関させるシグナリング法を実装することをさらに含む。

【図面の簡単な説明】

【0010】

【図1】いくつかの実施形態による、「キー」点群フレームのための補助情報バッファの図である。

10

【図2】いくつかの実施形態による、時間的予測を利用する補助情報バッファの図である。

【図3】いくつかの実施形態による、双方向予測を利用する補助情報バッファの図である。

【図4】いくつかの実施形態による、層の階層的グループ化の図である。

【図5】いくつかの実施形態によるV - P C Cシグナリングの図である。

【図6】いくつかの実施形態による、論理チェーン情報を含むV - P C Cシグナリングの例示的な符号化の図である。

【図7】いくつかの実施形態による、論理チェーン情報を含むV - P C Cシグナリングの例示的な符号化の図である。

【図8】いくつかの実施形態による、論理チェーン情報を含むV - P C Cシグナリングの例示的な符号化の図である。

20

【図9】いくつかの実施形態による、点群符号化構造を実装する方法のフローチャートである。

【図10】いくつかの実施形態による、点群符号化構造を実装するように構成された例示的なコンピュータ装置のブロック図である。

【発明を実施するための形態】

【0011】

本明細書では、動的点群の時間的圧縮のための新規符号化構造について説明する。この符号化構造は、点群パッチの予測に使用される全ての参照が現在の点群フレームに制限される「キー」点群フレームの概念を定める。さらに、参照パッチのリスト及びその点群フレームからのそれぞれの境界ボックスを別のフレーム内のパッチの予測に使用するために記憶する点群パッチバッファについても説明する。点群フレームの符号化順が提示順と異なることができる場合、参照パッチのリストは過去からのパッチ及び将来からのパッチを含むことができ、双方向予測が使用される。点群の層にも同様の参照バッファリストの概念を適用することができる。上位レベル情報を含むブロックのIDへの参照をペイロード内でシグナリングすることによってV - P C Cのブロックを相関させるシグナリング法についても説明する。

30

【0012】

メタデータ情報（パッチ及びそれぞれの境界ボックス）を含むバッファが、前のフレームからのデータを記憶する。次に、エンコーダが、現在のフレームを符号化するために、現在のフレームと前のフレームから記憶されたパッチとの間のパッチマッチング（p a t c h m a t c h i n g）を実行する。さらに、これらのパッチを、将来的に配置されることも又はされないこともある2又は3以上前のフレームからの2又は3以上のパッチとマッチングすることができる。双方向予測を可能にするために、参照パッチリストを、一方が過去のフレームを示して他方が将来のフレームを示す2つの別のリストに記憶することができる。「キー」点群フレームが定められる場合、過去の点群フレームを使用した予測は認められず、使用される全ての参照は現在のフレームに由来すべきである。さらに、V - P C Cユニットのシグナリング構造は、上位レベルの要素への参照を含むものとする。例えば、属性パッチパラメータセット（A t t r i b u t e P a t c h P a r a m e t e r S e t : A P P S）のための情報を含むブロックは、属性パラメータセット（A t t r i b u t e P a r a m e t e r S e t : A P S）の情報を含むブロックを参照し

40

50

、これはIDを使用して行われる。

【0013】

HEVCなどのビデオエンコーダにおいて使用される復号ピクチャバッファ (Decoded Picture Buffer: DPB) と同様の、現在の情報を予測するための過去の情報のリストについて説明する。この技術の新規性は、画像を符号化するためだけに参照リストを使用するのではなく、メタデータ情報 (パッチ) を符号化するためにも使用する点である。さらに、本明細書では、上位レベルの情報を含むブロックへの参照をペイロード内で示すシグナリング法についても説明する。

【0014】

本明細書で説明する点群符号化構造は、構造をより効率的なものにするために点群符号化標準に追加された新たな要素に関連する情報を含む。含まれる情報は、点群圧縮ビットストリーム (point cloud compressed bitstreams) の関係性、並びに符号化及び復号方法を提供する。

10

【0015】

点群符号化標準では、点群を曲面パッチ (surface patches) にセグメント化して曲面パッチを投影する。この投影により、1又は2以上の2Dビデオエンコーダを使用して符号化できる2D画像が生成される。2D画像の生成法及び3D画像からの2D画像の再構築法、又はこれらの逆 (例えば、3Dの点から2Dビデオへの移行方法などのマッピング情報) を示す補助情報 (又はメタデータ) が送信される。例えば、点群は100個のパッチに分割され、すなわち点群からパッチへの移行方法又は3Dパッチから2Dパッチへの移行方法を示すメタデータが存在する。メタデータは、フレーム間の時間的相関などの特性を含む。このメタデータ (例えば、時間的相関情報) は符号化される。例えば、点群の1つのフレームは100個のパッチに分割され、1つのパッチは、顔上の全ての点が単一のパッチにグループ化された顔パッチである。次のフレームはわずかにシフトした点群を含み、今回は点群が110個のパッチに分割され、2つのパッチが顔を表す。フレーム毎のパッチのリストを他のフレームからのパッチのリストと相関させることができる。例えば、補助情報はパッチの位置情報 (例えば、位置10) などの3D情報を含み、第2のフレーム内のパッチの位置情報 (例えば、位置11) は異なる/変化する。新たな位置を符号化する必要はなく、古いパッチと比べた新たなパッチの関係が符号化される (例えば、この例では第2のフレームからのパッチが位置 $p+1$ である)。従って、これらのフレーム間のパッチ間の関係/相関を記憶することができる。この例では現在のフレーム及び前のフレームについて説明しているが、あらゆるフレームを使用及び相関させることができる。

20

30

【0016】

パッチが5フレーム前のフレームの前のフレーム内のパッチに時間的に相関するように、補助情報のためのバッファリング構造を生成することができる。バッファリング構造は、パッチ及びフレームの全ての依存性を示す。その後、キーフレームに類似する前のフレームに依存しない補助情報などの異なるタイプの補助情報を生成することができる。過去からのフレームに依存するパッチは時間的予測を含む。フレームが順に符号化されるのではなくパッチが過去及び将来のパッチを参照できるようにフレームを並べ替える双方向予測を実装することができ、これによって高い柔軟性が可能になる。

40

【0017】

図1に、いくつかの実施形態による、「キー」点群フレームのための補助情報バッファの図を示す。「キー」点群フレームでは、各パッチの補助情報が同じフレーム内の前のパッチからのものである。さらなるフレームからの補助情報は利用されない。

【0018】

図2に、いくつかの実施形態による、時間的予測を利用する補助情報バッファの図を示す。時間的予測では、パッチの補助情報が前のフレーム (例えば、Frame_{N-1} . . . Frame₀) からのパッチに由来することができる。

【0019】

50

図3に、いくつかの実施形態による、双方向予測を利用する補助情報バッファの図を示す。双方向予測では、パッチの補助情報が前のフレーム（例えば、Frame_{N-1}...Frame₀）又は将来のフレーム（例えば、Frame_{N+1}）からのパッチに由来することができる。

【0020】

補助情報のデルタ性/差動性を考えると、現在のTMC2マッチング手法の限界は、参照として前の補助情報データユニットしか使用されない点である。現在の補助情報データユニットのより柔軟な予測スキームは、一連の予め定められた参照補助情報セット（reference auxiliary information set: RAIS）から参照リストを作成することである。図3に、参照補助情報セット及びLDのリストの例を示す。

10

【0021】

図3の例は、参照セット及びリストの概念を使用する単純な実装を示す。しかしながら、この実装は、ランダムアクセスのための双予測の使用、及びL0に加えたL1型参照リストの導入により、ジオメトリビデオコーデック（geometry video codec）のGOP構成に従って一連の予め定められた参照補助情報セットを生成するための柔軟性を提供する。また、現在のスキームを、現在のフレーム内の一致するパッチが異なる参照フレームインデックスを参照できるようにさらに拡張して一般化することもできる。

【0022】

ビデオ層バッファでは、3D面から2D面に点を投影した時に、互いに重なり合って投影される点があくつか存在する。これらの点を保持するために、第1の層が第1の点であり、第2の層が第2の点であり、以下同様であるようなビデオ層を実装することができる。いくつかの実装では、第1の層が最初の点であり、第2の層が最後の点である2つの層のみを利用し、これらの間の点は符号化されない。いくつかの実装では、さらなる層（例えば、16個の層）を利用することもできる。

20

【0023】

具体的に言えば、現在のV-PCC仕様では最大16個の層が存在できるのに対し、TMC2-SWの実装では層数がたった2つに制限される。また、フラグ「absolute_d1_flag」を使用して、単一のインターリーブ層0/層1入力（absolute_d1_flag=1）、又は2つの入力（absolute_d1_flag=0）：層0及びジオメトリビデオコーデックへの残差（layer_1-layer_0）を生成する。2つよりも多くの層を取り扱う場合には、予測順のシグナリングがより複雑になる。従って、参照補助情報セット及びリストの概念を複数の層に拡張することで、両ケースに対応する統一された方法が得られる。

30

【0024】

2つよりも多くの層（3D面からのスライス）の利用を可能にするためにFIFOバッファを実装することができる。FIFOバッファは、どの層が以前に符号化されたかを示すことができる。これらの層は、階層的な予測スキームを生成するために階層的にグループ化することもできる。例えば、層を階層グループに組み合わせて、これらの階層グループを予測に使用することができる。図4に、いくつかの実施形態による層の階層的グループ化の図を示す。

40

【0025】

単一の/2つの層を作成する別の方法は、現在のTMC2実装に従って2層スキーム又は単一層スキームのいずれかを生成する、層の階層的グループ化である。このようなグループ化の情報は、固定/事前定義し、又はデコーダに送信することができる。

【0026】

図5に、いくつかの実施形態によるV-PCCシグナリングの図を示す。V-PCCビットストリームはV-PCCユニットを含み、各V-PCCユニットは、V-PCCユニットヘッダ及びV-PCCユニットペイロードを含む。V-PCCユニットペイロードは、シーケンスパラメータセット、占有パラメータセット、ジオメトリパッチパラメータセ

50

ット、占有ビデオデータユニット、フレームパラメータセット、ジオメトリパラメータセット、属性パッチパラメータセット、ジオメトリビデオデータユニット、属性パラメータセット、補助情報データユニット、又は属性ビデオデータユニットのうちのいずれかとすることができる。

【0027】

各V-PCCUユニットの論理チェーンを生成するために、各V-PCCUユニットのV-PCCUビットストリームと共にシグナリングを含めることができる。例えば、ジオメトリパッチパラメータセットは、シーケンスパラメータセットに関連することができる。シグナリングは、V-PCCUユニットを関連付ける情報を含むことができる。

【0028】

図6～図8に、いくつかの実施形態による、論理チェーン情報を含むV-PCCUシグナリングの例示的な符号化を示す。シーケンスパラメータセット(sps)は、シーケンス全体にとって有効なパラメータを定める。シーケンスパラメータセットのIDはsps__sps__idである。属性パラメータセット(aps)は、特定の属性のパラメータを定める。属性パラメータセットは、ペイロード内のaps__aps__id及びsps__sps__idによって示されるシーケンスパラメータセットに関連する。パラメータ又は条件のライブラリを生成した後に、シグナリング/IDを使用してライブラリからの選択を行うことができる。図8に示すように、フレームパラメータセットユニットは、イントラフレームの指示、デコーダバッファをリセットするためのメタデータ情報、デコーダへの通知、パッチ配向などの新たなデフォルト値、などの情報を送信するために使用することができる。

【0029】

spsペイロードはグローバルスコープ(global scope)を有しているのので、属性タイプ、対応する(単複の)次元(この情報はテーブルから非明示的に導出することができる)及び各属性のインスタンス数をシグナリングすることができる。

【0030】

現在のTMC2 SW内のビデオデータユニットはフレームグループ(Group of Frame: GoF)固有のものであり、CD構文仕様(CD syntax specification)には従わない。

【0031】

現在の草案には層0及び層1をインターリーブする順序が指定されていないので、デフォルト順を指定することができる。この順序が固定されていて変更できない場合には、エンコーダに制約を課すことができる。ヘッダにlayer__index値を導入することによって、より柔軟な順序で層を符号化することが可能になる。

【0032】

「パラメータセットid」を追加することで、各パッチのメタデータ情報をシグナリングする必要なく同様の/共通する機能(例えばROIなどのスケール、オフセット)を有するパッチのグループを表すためのよりコンパクトな方法、誤り耐性、及びHEVC/AVCにおいてパラメータセットIDの使用が十分に確立される協調をサポートすることができる。TMC13も同様の概念を使用する。

【0033】

GoFの概念は、TMC2では導入されているが現在の草案ではサポートされていないので、HEVC「IDR」又は「CRA」ピクチャに若干類似する、フレーム内のランダムアクセスポイント(クローズド又はオープンGOP)を識別できる機構が依然として必要である。ある実装は、フレームパラメータセットユニットを使用して、1)IRAP(例えばIDR、CRAなどのイントラランダムアクセスポイント)の指示、2)デコーダバッファをリセットするためのメタデータ情報、及び3)デフォルト値のリセット、などの他の情報を伝えることを含む。CRA(オープンGOP)の場合には必ずしも出力順と復号順が同じではないので、その使用は補助情報入力の並べ替えが可能であるかどうかにかかわらず依存する。

10

20

30

40

50

【 0 0 3 4 】

以下は例示的なコードである。

```

sequence_parameter_set( ) {
    profile_idc                u(7)
    tier_flag                   u(1)
    level_idc                   u(8)
    frame_width                 u(16)
    frame_height                u(16)
    additional_points_patch_enabled_flag    u(1)
    if ( additional_points_patch_enabled_flag ) {
        additional_points_separate_video_enabled_flag    u(1)
    }
    auxiliary_information_delta_coding_enabled_flag    u(1)
    auxiliary_information_delta_coding_enabled_flag    u(1)
    layer_count_minus1        u(4)
    layer_ref_enabled_flag    u(1)
    num_layer_ref_sets        u(4)
    for( i = 0; i < num_layer_ref_sets; i++)
        rlayer_ref_sets(i)
    attribute_count            u(16)
    geometry_metadata_enabled_flag    u(1)
    if ( geometry_metadata_enabled_flag ) {
        geometry_smoothing_metadata_enabled_flag    u(1)
        geometry_scale_metadata_enabled_flag    u(1)
        geometry_offset_metadata_enabled_flag    u(1)
    }
frame_parameter_set ( ) {
    byte_alignment ( )
}

```

【 0 0 3 5 】

以下は例示的なコードである。

```

auxiliary_information_data_unit( ) {
    patch_count_minus1        u(32)
    if(auxiliary_information_orientation_enabled_flag) {
        auxiliary_information_patch_orientation_present_flag    u(1)
    }
    patch_2d_shift_u_bit_count_minus1    u(8)
    patch_2d_shift_v_bit_count_minus1    u(8)
    patch_3d_shift_tangent_axis_bit_count_minus1    u(8)
    patch_3d_shift_bitangent_axis_bit_count_minus1    u(8)
    patch_3d_shift_normal_axis_bit_count_minus1    u(8)
    patch_lod_bit_count_minus1    u(8)
    if( auxiliary_information_delta_coding_enabled_flag &&
pc_frame_type != IDR || != CRA ){
        use_bit_count_for_unmatched_patch_enabled_flag    u(1)
        if( bit_count_for_unmatched_patch_enabled_flag ) {
            inherit_patch_2d_shift_u_bit_count_for_
unmatched_patch_flag    u(1)
        }
        if( inherit_patch_2d_shift_u_bit-count_for_
unmatched_patch_flag ){
            unmatched_patch_2d_shift_u_bit_    u(8)

```

```

count_minus1
    }
    inherit_patch_2d_shift_v_bit_count_for_      u(1)
unmatched_patch_flag

    if( inherit_patch_2d_shift_v_bit_count_for_
unmatched_patch_flag ){
        unmatched_patch_2d_shift_v_bit_      u(8)
count_minus1
    }
    inherit_patch_3d_shift_tangent_axis_bit_      u(1)
count_for_unmatched_patch_flag
    if( inherit_patch_3d_shift_tangent_axis_
bit_count_for_unmatched_patch_flag ){
        unmatched_patch_3d_shift_tangent_      u(8)
axis_bit_count_minus1
    }
    inherit_patch_3d_shift_bitangent_axis_bit_      u(1)
count_for_unmatched_patch_flag
    if( inherit_patch_3d_shift_bitangent_axis_bit_      20
count_for_unmatched_patch_flag ){
        unmatched_patch_3d_shift_bitangent_axis_ u(8)
bit_count_minus1
    }
    inherit_patch_3d_shift_normal_axis_bit_      u(1)
count_for_unmatched_patch_flag
    if( inherit_patch_3d_shift_normal_axis_bit_
count_for_unmatched_patch_flag ){
        unmatched_patch_3d_shift_normal_axis_  u(8)
bit_count_minus1
    }
}
for( p = 0; p < matched_patch_count; p++ ) {
    delta_patch_index[ p ]          ae(v)
    delta_patch_2d_shift_u[ p ]      se(v)
    delta_patch_2d_shift_v[ p ]      se(v)
    delta_patch_3d_shift_tangent_axis[ p ]      se(v)
    delta_patch_3d_shift_bitangent_axis[ p ]      se(v)
    delta_patch_3d_shift_normal_axis[ p ]      se(v)
    if( geometry_absolute_coding_enabled_flag )      40
        patch_projection_mode[ p ]          ae(v)
    delta_patch_2d_size_u[ p ]      se(v)
    delta_patch_2d_size_v[ p ]      se(v)
}
for( p = matched_patch_count; p < patch_count_minus1; p++ ) {
    patch_2d_shift_u[ p ]          ae(v)
    patch_2d_shift_v[ p ]          ae(v)
    if(auxiliary_information_flexible_orientation
present_flag)
        patch_orientation_index[ p ]      ae(v)      50

```

```

        patch_3d_shift_tangent_axis[ p ]          ae(v)
        patch_3d_shift_bitangent_axis[ p ]        ae(v)
        patch_3d_shift_normal_axis[ p ]          ae(v)
        patch_lod[ p ]                            ae(v)
        if( geometry_absolute_coding_enabled_flag )
            patch_projection_mode[ p ]            ae(v)
        delta_patch_2d_size_u[ p ]                se(v)
        delta_patch_2d_size_v[ p ]                se(v)
        normal_axis[ p ]                           ae(v)
    }
}
byte_alignment( )
}

```

【0036】

委員会草案では、「sps_attribute_count」のセマンティックが以下のように指定される。

sps_attribute_countは、点群に関連する属性の数を示す。
sps_attribute_countは、0～65535の範囲内とする。sps_attribute_countのセマンティックは、点群に関連する各属性の属性インスタンスの総数を示すように見える。その後、以下の構文テーブルは、「aps_attribute_codec_id」「aps_attribute_dimension_minus_1」などの「attribute_parameter_set()」構文要素の一部（或いは全て又はそのグループ）が変化しないままであるにもかかわらず、同じ属性タイプについて「attribute_parameter_set()」構造が属性インスタンス毎に繰り返し呼び出されることを示す。

【0037】

シーケンスパラメータセット構文

```

sps_attribute_count          u(16)
for (i = 0; i < sps_attribute_count; i++)
    attribute_parameter_set (i)

```

以下に示すように、属性インスタンスではなく属性タイプの数を参照する別の定義が存在することもできる。この後者の解釈では、sps_attribute_count値の範囲は0～15となる。従って、以下の構文テーブルの変更を実装することができる。

【0038】

V-PCCユニットヘッダ構文

```

vpcc_unit_header          記述子
    . . . . .
    if (vpcc_unit_type == VPCC_AVD) {
        vpcc_attribute_type_index          u(4)
        vpcc_attribute_instance_index      u(7)
        . . . . .
    }

```

【0039】

シーケンスパラメータセット構文

```

sps_attribute_type_count          u(4)

for (i = 0; i < sps_attribute_type_count; i++)
    attribute_parameter_set (i)

```

【0040】

属性パラメータセット構文

10

20

30

40

50

```

attribute_parameter_set (attributeTypeIndex) {                               記述子
    aps_attribute_dimension_minus1[attributeTypeIndex]                       u(8)
    attributeDimension = aps_attribute_dimension_minus1
[attributeTypeIndex] + 1
    aps_attribute_type_id[attributeTypeIndex]                                u(4)
    aps_attribute_codec_id[attributeTypeIndex]                               u(8)
    aps_attribute_instance_count[attributeTypeIndex]                         u(7)
    attrInstCnt = aps_attribute_instance_count [attributeTypeIndex]
    if(attrInstCnt > 1)
        aps_attribute_group_present_flag[attributeTypeIndex]                u(1)                               10
    if(aps_attribute_group_present_flag [attributeTypeIndex])
        aps_attribute_group_count_minus1[attributeTypeIndex]                u(7)
    attrGrpCnt = aps_attribute_group_count_minus1[attributeTypeIndex]
+ 1
    for (i = 0; i < attrGrpCnt; i++) {
        for (j = 0; attrGrpCnt > 1 && j < attrInstCnt; j++)
            aps_attribute_group_instance_map [attributeTypeIndex][i][j]    u(1)
            aps_attribute_params_enabled_flag[attributeTypeIndex][i]        u(1)
            if(aps_attribute_params_enabled_flag[attributeTypeIndex][i])
                attribute_sequence_params(i, attributeDimension)              20
            aps_attribute_patch_params_enabled_flag[attributeTypeIndex][i] u(1)

            if(aps_attribute_patch_params_enabled_flag
[attributeTypeIndex][i]) {
                aps_attribute_patch_scale_params_enabled_flag                u(1)
[attributeTypeIndex][i]
                aps_attribute_patch_offset_params_enabled_flag                u(1)
[attributeTypeIndex][i]
            }
        }
    }
}

```

【0041】

v p c c _ a t t r i b u t e _ t y p e _ i n d e x は、点群に関連する属性タイプへのインデックスを示す。s p s _ a t t r i b u t e _ t y p e _ i n d e x は、0 ~ 15 の範囲内とする。

【0042】

v p c c _ a t t r i b u t e _ i n s t a n c e _ i n d e x は、属性ビデオデータユニット (Attribute Video Data unit) で伝えられる属性データのインスタンスインデックスを示す。v p c c _ a t t r i b u t e _ i n d e x の値は、0 ~ 127 の範囲内とする。

【0043】

s p s _ a t t r i b u t e _ t y p e _ c o u n t は、点群に関連する属性タイプの数を示す。s p s _ a t t r i b u t e _ t y p e _ c o u n t は、0 ~ 15 の範囲内とする。

【0044】

a p s _ a t t r i b u t e _ i n s t a n c e _ c o u n t [i] は、点群に関連する属性タイプ i の属性インスタンスの数を示す。a p s _ a t t r i b u t e _ i n s t a n c e _ c o u n t は、0 ~ 127 の範囲内とする。

【0045】

a p s _ a t t r i b u t e _ g r o u p _ p r e s e n t _ f l a g [i] が 1 に等

50

しい場合には、属性タイプ i に関連する属性インスタンスがグループ化されることを示す。 $aps_attribute_group_flag[i]$ が 0 に等しい場合には、属性タイプ i に関連する属性インスタンスがグループ化されないことを示す。 $aps_attribute_group_present_flag[i]$ が存在しない場合には、その値が 0 に等しいと推測される。

【0046】

$aps_attribute_group_count_minus1[i] + 1$ は、属性タイプ i に関連する属性インスタンスグループの数を示す。 $aps_attribute_group_count$ は、1 ~ 127 の範囲内とする。 $aps_attribute_group_count_minus1[i]$ が存在しない場合には、その値が 0

10

【0047】

$aps_attribute_group_instance_map[i][j][k]$ は、属性タイプ i に関連する属性インスタンスグループ j に属性インスタンス k が属するかどうかを示す。 $aps_attribute_group_instance_map[i][j][k]$ は、0 ~ 1 の範囲内とする。 $aps_attribute_group_instance_map[i][j][k]$ が存在しない場合には、その値が 1 に等しいと推測される。 $aps_attribute_group_instance_map[i][j][k]$ が存在しない場合には、全ての属性インスタンスが単一グループにマッピングされると推測される。

20

【0048】

上記の変更は、 $vpcc_unit_header()$ における「 $vpcc_attribute_index$ 」と「 $vpcc_attribute_type$ 」との間の対応を確立し、同じ属性タイプの全てのインスタンスが同様の属性パラメータセットを共有するようにするか、それともグループとするかに関する柔軟性をもたらし、属性タイプに基づいて $attribute_parameter_set()$ 構文構造をアクティブにする。

【0049】

ジオメトリ層と属性層との間の緊密な境界

委員会草案では、ジオメトリ層の数と属性層の数とが同様である。これにより、必ずしも属性がジオメトリと同じ数の層を有していない事例に対処する柔軟性が制限される。単純な例としては、属性が 3 よりも大きな次元を有する事例が挙げられる。第 2 の例としては、属性層を単一層のジオメトリにインターリーブできる場合と複数層のジオメトリにインターリーブできる場合、又はこの逆を挙げることができる。

30

【0050】

このような柔軟性をサポートするために、構文テーブルに以下の変更を行う。

【0051】

属性パラメータセット構文

```

attribute_parameter_set( attributeIndex {
    aps_attribute_type_id[attributeIndex]          記述子 u(4)
    aps_attribute_dimension_minus1[ attributeIndex ] u(8)
    aps_attribute_codec_id[ attributeIndex ]       u(8)
    aps_attribute_layer_count_present_flag[ attributeIndex ] u(1)
    if( asp_attribute_layer_count_present_flag[ attributeIndex ] ) {
        asp_attribute_layer_abs_delta_count_minus1[ attributeIndex ] u(2)
    }
    asp_attribute_layer_count_sign_flag [ attributeIndex ] u(1)
}
aps_attribute_patch_params_enabled_flag[ attributeIndex ] u(1)
if(aps_attribute_patch_params_enabled_flag[ attributeIndex ] ) {
    aps_attribute_patch_scale_params_enabled_flag[ attributeIndex ] u(1)
}
attributeIndex ]

```

40

50

```

)
    aps_attribute_patch_offset_params_enabled_flag[      u(1)
attributeIndex ]
    }
}

```

【0052】

aps_attribute_layer_count_present_flag[i]が1に等しい場合には、属性層の数がインデックスiを有する属性の「sps_layer_count_minus1+1」と同じではないことを示す。aps_attribute_layer_count_present_flag[i]が0に等しい場合には、属性層の数がインデックスiを有する属性の「sps_layer_count_minus1+1」と同じであることを示す。

10

【0053】

asp_attribute_layer_abs_delta_count_minus1[i]+1は、インデックスi及びsps_layer_countを有する属性の属性層数間の絶対差を指定する。asp_attribute_layer_abs_delta_count_minus1[i]の値は、0～3の範囲内とする。存在しない場合、asp_attribute_layer_abs_delta_count_minus1[i]の値は1に等しいと推測される。

【0054】

asp_attribute_layer_count_sign_flag[i]が1に等しい場合には、asp_attribute_layer_abs_delta_count_minus1[i]+1が0よりも大きい値を有することを指定する。asp_attribute_layer_count_sign_flag[i]が0に等しい場合には、asp_attribute_layer_abs_delta_count_minus1[i]+1が0未満の値を有することを指定する。存在しない場合、asp_attribute_layer_count_sign_flag[i]の値は0に等しいと推測される。

20

【0055】

ここに属性復号プロセスの変更が反映される。

30

sps_attribute_countが0に等しい場合には、属性ビデオフレームが復号されず、最終的な再構築された点群に属性情報が関連付けられない。

$$\text{attributeLayerCount} = (\text{sps_layer_count_minus1} + 1) + \text{asp_attribute_layer_count_sign_flag}[\text{vpcc_attribute_index}] * (\text{asp_attribute_layer_abs_delta_count_minus1}[\text{vpcc_attribute_index}] + 1)$$

【0056】

それ以外の場合（sps_attribute_countが0に等しくない場合）には、以下が適用される。

【0057】

attributeLayerCountが0に等しい場合には、関連する異なるvpcc_attribute_indexと、aps_attribute_codec_id[vpcc_attribute_index]によって指定された関連するコーデックとをそれぞれが入力として伴うsps_attribute_count数のビデオ復号プロセスが呼び出される。このプロセスの出力は、復号され表示/出力順に並べられた属性ビデオフレームAttrFrame[attrIdx][layerIdx][orderIdx][compIdx][y][x]、並びにその関連するビット深度AttrBitDepth[attrIdx][layerIdx][orderIdx]、幅AttrWidth[attrIdx][layerIdx][orderIdx]、及び高さAttrHeight[attrIdx][layerIdx][orderIdx]情報であり、ここでのattrIdxは属性インデックスに対応して0～sps_

40

50

`attribute_count - 1`の範囲内であり、`layerIdx`は層のインデックスに対応して0に等しく、`orderIdx`は復号された属性フレームの表示順インデックスであり、`compIdx`は属性コンポーネントインデックスに対応して0 ~ `aps_attribute_dimension_minus1[attrIdx] - 1`の範囲内であり、`y`は0 ~ `AttrHeight[attrIdx][layerIdx][orderIdx] - 1`の範囲内であり、`x`は復号されたフレーム内の列インデックスであって0 ~ `AttrWidth[attrIdx][layerIdx][orderIdx] - 1`の範囲内である。

【0058】

それ以外の場合 (`sps_attribute_count`が0に等しくない場合) には以下が適用される。

`sps_multiple_layer_streams_present_flag`が0に等しい場合には、それぞれ関連する異なる`vpcc_attribute_index`と、`aps_attribute_codec_id[vpcc_attribute_index]`によって指定された関連するコーデックとを入力として伴う`sps_attribute_count`数のビデオ復号プロセスが呼び出される。このプロセスの出力は、復号され表示/出力順に並べられた中間属性ビデオフレーム`AttrFrame[attrIdx][tempOrderIdx][compIdx][y][x]`、並びにその関連するビット深度`tempAttrBitdepth[attrIdx][tempOrderIdx]`、幅`tempAttrWidth[attrIdx][tempOrderIdx]`、及び高さ`tempAttrHeight[attrIdx][tempOrderIdx]`情報であり、ここでの`attrIdx`は属性インデックスに対応して0 ~ `sps_attribute_count - 1`の範囲内であり、`tempOrderIdx`は復号された全属性フレームの表示順インデックスであり、`compIdx`は属性コンポーネントインデックスに対応して0 ~ `aps_attribute_dimension_minus1[attrIdx - 1]`の範囲内であり、`y`は0 ~ `tempAttrHeight[attrIdx][tempOrderIdx] - 1`の範囲内であり、`x`は復号されたフレーム内の列インデックスであって0 ~ `tempAttrWidth[attrIdx][tempOrderIdx] - 1`の範囲内である。インデックス`attrIdx`を有する属性の復号された属性ビデオフレームは、各層の表示/出力順`orderIdx`で以下のように導出される。

```
for(i=0;i =sps_layer_count_minus1;i++){
    mappedIdx=orderIdx*attributeLayerCount)+i
    AttrBitdepth[attrIdx][i][orderIdx]=tempAttrBitdepth[attrIdx][mappedIdx]
    AttrWidth[attrIdx][i][orderIdx]=tempAttrWidth[attrIdx][mappedIdx]
    AttrHeight[attrIdx][i][orderIdx]=tempAttrHeight[attrIdx][mappedIdx]
    AttrFrame[attrIdx][i][orderIdx]=tempAttrFrame[mappedIdx]
}
```

【0059】

最後に、解釈を明確かつ容易にするために、両方に「`vpcc_layer_index`」ではなく「`vpcc_geometry_layer_index`」及び「`vpcc_attribute_layer_index`」を使用する。

【0060】

ジオメトリ及び属性のための単一及び複数ビットストリーム

`sps_multiple_layer_streams_present_flag`が0に等しい場合には、全てのジオメトリ又は属性層がそれぞれ単一のジオメトリ又は属性ビデオストリームに配置されることを示す。`sps_multiple_layer_streams_present_flag`が1に等しい場合には、全てのジオメトリ又は属性層が別のビデオストリームに配置されることを示す。`sps_multiple_layer_streams_present_flag`が存在しない場合には、その値

が 0 に等しいと推測される。

【 0 0 6 1 】

上記のセマンティックに基づけば、その意図は、単一層 / 複数層ストリームをジオメトリ層と属性層との間で緊密に境界することである。将来的な拡張の柔軟性を可能にするために、1つの選択枝は、`sps__multiple__layer__streams__present__flag` をジオメトリ及び属性について別々に定めることである（例えば、`sps__multiple__geometry__layer__streams__present__flag = sps__multiple__attribute__layer__streams__present__flag` をデフォルトとする `sps__multiple__geometry__layer__streams__present__flag` 及び `sps__multiple__attribute__layer__streams__present__flag`）。

10

【 0 0 6 2 】

ジオメトリシーケンスパラメータ構文

`geometry__parameter__set()` / `attribute__parameter__set()` 対 `geometry__sequence__params()` / `attribute__sequence__params()` の順序付け。

現在の構文では、上記の構文構造の呼び出し順は以下の通りである。

`sequence__parameter__set -> geometry / attribute__sequence__params`

20

又は

`sequence__parameter__set -> geometry / attribute__parameter__set`

【 0 0 6 3 】

図 9 に、いくつかの実施形態による、点群符号化構造を実装する方法のフローチャートを示す。ステップ 900 において、点群符号化構造を展開 / 実装する。点群符号化構造の展開及び実装は、「キー」点群フレームを決定することを含む。キー点群フレームは、予測のために他のどのフレームにも依存しないフレームである。

30

【 0 0 6 4 】

ステップ 902 において、双方向予測を符号化に使用する。双方向予測は、パッチ情報及び / 又はメタデータ情報などの過去及び将来の情報を記憶する 2 つの独立構造（例えばリスト）を使用して実装され、予測の実行時には過去の情報及び / 又は将来の情報を使用して予測を行うことができる。これらの構造は、点群パッチバッファに記憶することができる。点群の点を互いに重なり合って投影できるように、点群の層にはバッファリストが使用される。また、V-PPC のブロックを相関させるシグナリング法も実装される。いくつかの実施形態では、これよりも少ない又はさらなるステップが実装される。例えば、復号ステップも実装される。いくつかの実施形態では、ステップの順序が変更される。

【 0 0 6 5 】

40

図 10 に、いくつかの実施形態による、点群符号化構造を実装するように構成された例示的なコンピュータ装置のブロック図を示す。コンピュータ装置 1000 は、3D コンテンツを含む画像及びビデオなどの情報の取得、記憶、計算、処理、通信及び / 又は表示のために使用することができる。コンピュータ装置 1000 は、点群符号化構造の態様のいずれかを実装することができる。一般に、コンピュータ装置 1000 を実装するのに適したハードウェア構造は、ネットワークインターフェイス 1002、メモリ 1004、プロセッサ 1006、I/O 装置 1008、バス 1010 及び記憶装置 1012 を含む。プロセッサの選択は、十分な速度の好適なプロセッサを選択する限り重要ではない。メモリ 1004 は、当業で周知のいずれかの従来のコンピュータメモリとすることができる。記憶装置 1012 は、ハードドライブ、CDROM、CDRW、DVD、DVDRW、高精細

50

ディスクドライブ、ウルトラHDドライブ、フラッシュメモリカード、又はその他のいずれかの記憶装置を含むことができる。コンピュータ装置1000は、1又は2以上のネットワークインターフェイス1002を含むことができる。ネットワークインターフェイスの例としては、イーサネット又は他のタイプのLANに接続されたネットワークカードが挙げられる。(単複の)I/O装置1008は、キーボード、マウス、モニタ、画面、プリンタ、モデム、タッチ画面、ボタンインターフェイス及びその他の装置のうちの1つ又は2つ以上を含むことができる。記憶装置1012及びメモリ1004には、点群符号化構造を実装するために使用される(単複の点群符号化構造アプリケーション1030が記憶されて、アプリケーションが通常処理されるように処理される可能性が高い。コンピュータ装置1000には、図10に示すものよりも多くの又は少ないコンポーネントを含めることもできる。いくつかの実施形態では、点群符号化構造ハードウェア1020が含まれる。図10のコンピュータ装置1000は、点群符号化構造のためのアプリケーション1030及びハードウェア1020を含むが、点群符号化構造は、ハードウェア、ファームウェア、ソフトウェア、又はこれらのあらゆる組み合わせでコンピュータ装置上に実装することもできる。例えば、いくつかの実施形態では、点群符号化構造アプリケーション1030がメモリにプログラムされ、プロセッサを使用して実行される。別の例として、いくつかの実施形態では、点群符号化構造ハードウェア1020が、点群符号化構造を実行するように特別に設計されたゲートを含むプログラムされたハードウェアロジックである。

10

【0066】

20

いくつかの実施形態では、(単複の)点群符号化構造アプリケーション1030が、複数のアプリケーション及び/又はモジュールを含む。いくつかの実施形態では、モジュールが1又は2以上のサブモジュールも含む。いくつかの実施形態では、これよりも少ない又はさらなるモジュールを含めることもできる。

【0067】

いくつかの実施形態では、点群符号化構造ハードウェア1020が、レンズ、イメージセンサ及び/又は他のいずれかのカメラコンポーネントなどのカメラコンポーネントを含む。

【0068】

好適なコンピュータ装置の例としては、パーソナルコンピュータ、ラップトップコンピュータ、コンピュータワークステーション、サーバ、メインフレームコンピュータ、ハンドヘルドコンピュータ、携帯情報端末、セルラ/携帯電話機、スマート家電、ゲーム機、デジタルカメラ、デジタルカムコーダ、カメラ付き電話機、スマートホン、ポータブル音楽プレーヤ、タブレットコンピュータ、モバイル装置、ビデオプレーヤ、ビデオディスクライタ/プレーヤ(DVDライタ/プレーヤ、高精細ディスクライタ/プレーヤ、超高精細ディスクライタ/プレーヤなど)、テレビ、家庭用エンターテイメントシステム、拡張現実装置、仮想現実装置、スマートジュエリ(例えば、スマートウォッチ)、車両(例えば、自動走行車両)、又はその他のいずれかの好適なコンピュータ装置が挙げられる。

30

【0069】

本明細書で説明した点群符号化構造を利用するには、装置が3Dコンテンツを取得又は受信し、3Dコンテンツの正しい効率的な表示を可能にするのに最適な方法でコンテンツを処理及び/又は送信する。点群符号化構造は、ユーザの支援を伴って、又はユーザの関与を伴わずに自動的に実行することができる。

40

【0070】

動作中、点群符号化構造は、3Dコンテンツをより効率的に符号化する。点群符号化構造は、点群パッチの予測に使用される全ての参照が現在の点群フレームに制限される「キー」点群フレームを定める。点群パッチバッファは参照パッチのリストを含み、点群フレームからのそれぞれの境界ボックスが別のフレーム内のパッチの予測に使用されるように記憶される。点群フレームの符号化順が提示順と異なることができる場合、参照パッチのリストは過去からのパッチ及び将来からのパッチを含むことができ、双方向予測が使用さ

50

れる。点群の層にも同様の参照バッファリストの概念を適用することができる。上位レベル情報を含むブロックのIDへの参照をペイロード内でシグナリングすることによってV-PC Cのブロックを相関させるシグナリング法についても説明する。これらの実装は符号化の効率を高める。

【0071】

点群符号化構造のいくつかの実施形態

1. 装置の非一時的メモリにプログラムされた方法が、

点群符号化構造を実装するステップと、

前記点群符号化構造を使用して点群メタデータを符号化するステップと、

を含む。

10

【0072】

2. 前記点群符号化構造を実装するステップは、予測のために他のどのフレームにも依存しないキー点群フレームを決定するステップを含む、条項1に記載の方法。

【0073】

3. 前記点群メタデータを符号化するステップは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタデータ情報とを使用することを含む双方向予測を利用する、条項1に記載の方法。

【0074】

4. 双方向予測は、前記過去のパッチ及びメタデータ情報のための第1の構造、並びに前記将来のパッチ及びメタデータ情報のための第2の構造という2つの別の構造を実装する、条項3に記載の方法。

20

【0075】

5. 前記第1の構造及び前記第2の構造は、点群パッチバッファに記憶される、条項1に記載の方法。

【0076】

6. 前記点群の層のバッファリストを利用して、点群の点が互いに重なり合って投影されることを可能にするステップをさらに含む、条項1に記載の方法。

【0077】

7. V-PC Cのブロックを相関させるシグナリング法を実装するステップをさらに含む、条項1に記載の方法。

30

【0078】

8.

点群符号化構造を実装し、

前記点群符号化構造を使用して点群メタデータを符号化する、

ためのアプリケーションを記憶する非一時的メモリと、

前記メモリに結合されて前記アプリケーションを処理するように構成されたプロセッサと、

を備えた装置。

【0079】

9. 前記点群符号化構造を実装することは、予測のために他のいずれのフレームにも依存しないキー点群フレームを決定することを含む、条項8に記載の装置。

40

【0080】

10. 前記点群メタデータを符号化することは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタデータ情報とを使用することを含む双方向予測を利用する、条項8に記載の装置。

【0081】

11. 双方向予測は、前記過去のパッチ及びメタデータ情報のための第1の構造、並びに前記将来のパッチ及びメタデータ情報のための第2の構造という2つの別の構造を実装する、条項10に記載の装置。

【0082】

50

12. 前記第1の構造及び前記第2の構造は、点群パッチバッファに記憶される、条項8に記載の装置。

【0083】

13. 前記点群の層のバッファリストを利用して、点群の点が互いに重なり合って投影されることを可能にすることをさらに含む、条項8に記載の装置。

【0084】

14. V-PPCのブロックを相関させるシグナリング法を実装することをさらに含む、条項8に記載の装置。

【0085】

15.

点群符号化構造と、

前記点群符号化構造を使用して点群メタデータを符号化するエンコーダと、
を備えたシステム。

【0086】

16. 前記点群符号化構造を実装することは、予測のために他のいずれのフレームにも依存しないキーフレームを決定することを含む、条項15に記載のシステム。

【0087】

17. 前記点群メタデータを符号化することは、過去のパッチ及びメタデータ情報と、将来のパッチ及びメタデータ情報とを使用することを含む双方向予測を利用する、条項15に記載のシステム。

【0088】

18. 双方向予測は、前記過去のパッチ及びメタデータ情報のための第1の構造、並びに前記将来のパッチ及びメタデータ情報のための第2の構造という2つの別の構造を実装する、条項17に記載のシステム。

【0089】

19. 前記第1の構造及び前記第2の構造は、点群パッチバッファに記憶される、条項15に記載のシステム。

【0090】

20. 前記点群の層のバッファリストを利用して、点群の点が互いに重なり合って投影されることを可能にすることをさらに含む、条項15に記載のシステム。

【0091】

21. V-PPCのブロックを相関させるシグナリング法を実装することをさらに含む、条項15に記載のシステム。

【0092】

本発明の構成及び動作の原理を容易に理解できるように、詳細を含む特定の実施形態に関して本発明を説明した。本明細書におけるこのような特定の実施形態及びこれらの実施形態の詳細についての言及は、本明細書に添付する特許請求の範囲を限定することを意図したものではない。当業者には、特許請求の範囲によって定められる本発明の趣旨及び範囲から逸脱することなく、例示のために選択した実施形態において他の様々な修正を行えることが容易に明らかになるであろう。

10

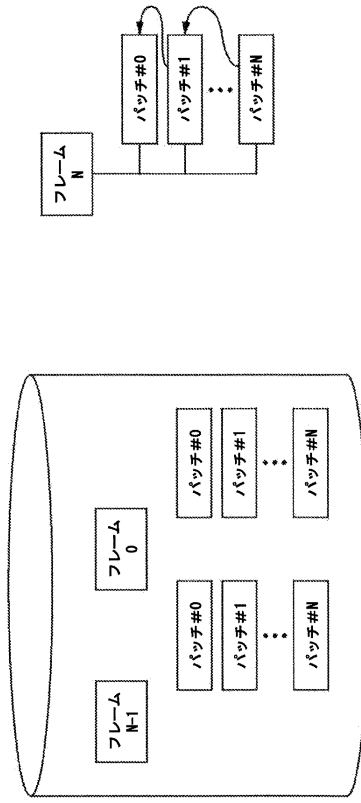
20

30

40

50

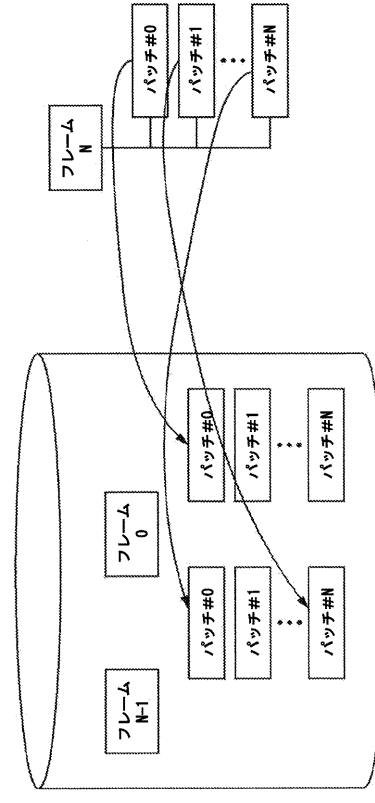
【図面】
【図 1】



バッチ×境界ボックス情報u0、v0、v1、sizeU1、sizeV1

Fig. 1

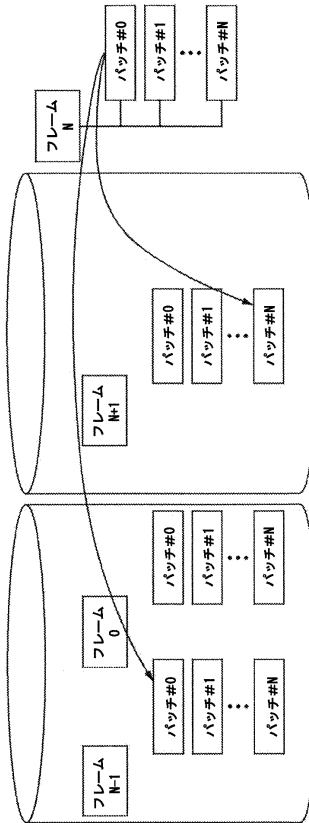
【図 2】



バッチ×境界ボックス情報u0、v0、v1、sizeU1、sizeV1

Fig. 2

【図 3】



バッチ×境界ボックス情報u0、v0、v1、sizeU1、sizeV1

Fig. 3

【図 4】

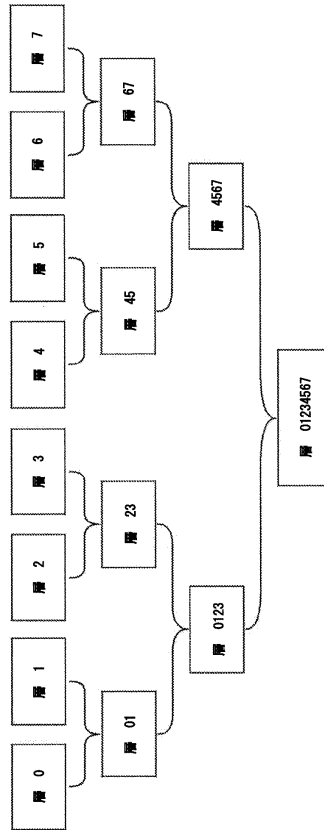
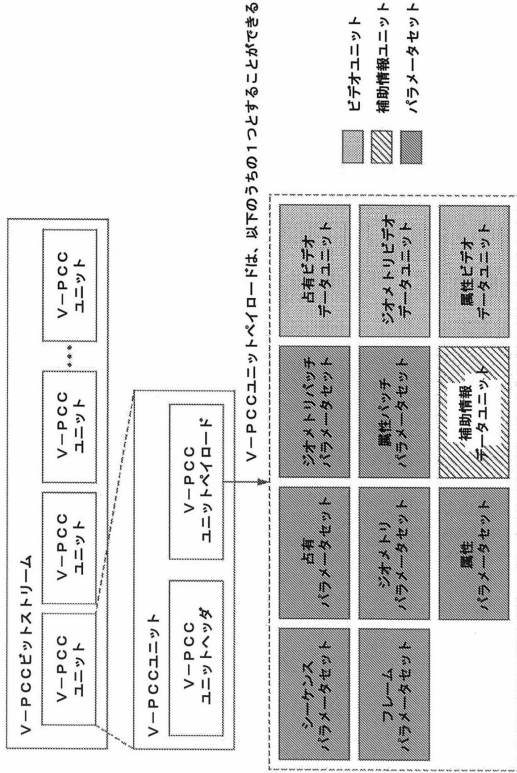


Fig. 4

【 5 】



【 6 】

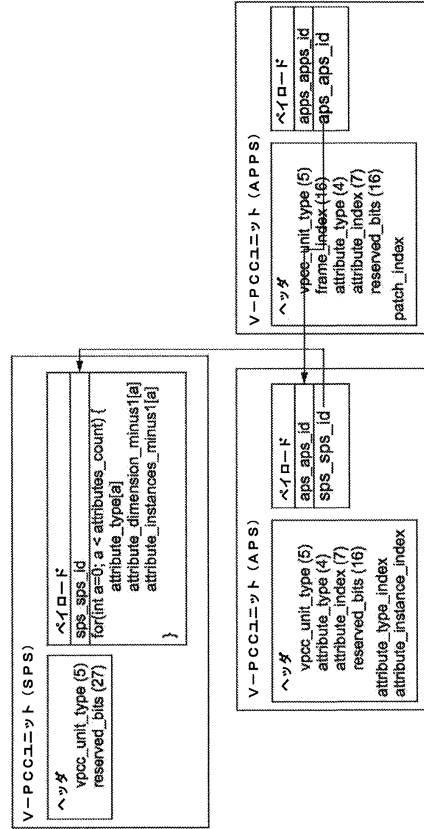


Fig. 5

Fig. 6

【 7 】

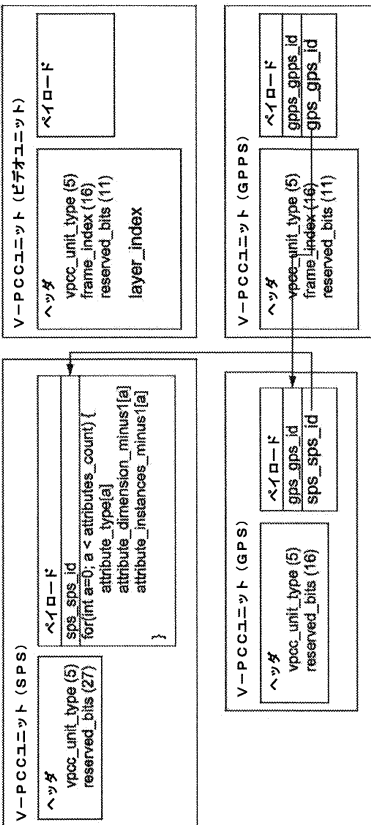


Fig. 7

【 8 】

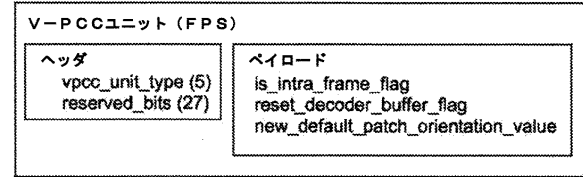


Fig. 8

【 図 9 】

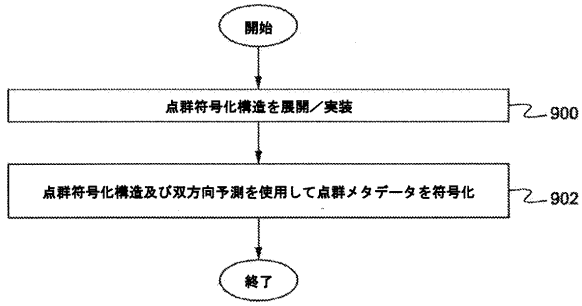


Fig. 9

【 図 10 】

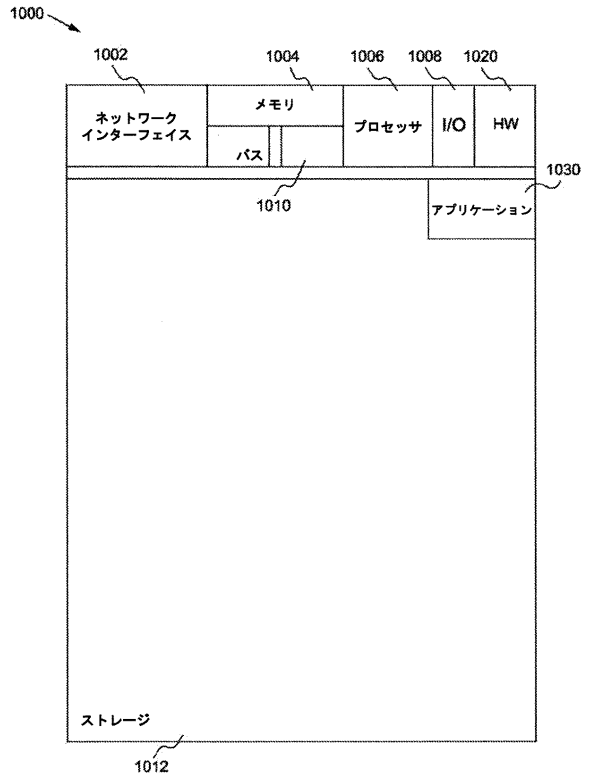


Fig. 10

10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 62/791,328

(32)優先日 平成31年1月11日(2019.1.11)

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 62/821,139

(32)優先日 平成31年3月20日(2019.3.20)

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 16/408,124

(32)優先日 令和1年5月9日(2019.5.9)

(33)優先権主張国・地域又は機関

米国(US)

(72)発明者 グラジオッシ ダニーロ

アメリカ合衆国 カリフォルニア州 9 5 1 2 5 サンノゼ カーサ ベルデ ストリート 3 7 0 0
アパートメント 3 5 0 3

(72)発明者 中神 央二

東京都港区港南1 - 7 - 1

(72)発明者 タバタバイ アリ

アメリカ合衆国 カリフォルニア州 9 5 0 1 4 クパチーノ イースト エステーツ ドライヴ 1 0
2 6 5

審査官 岩井 健二

(56)参考文献 特開2020 - 017946 (JP, A)

特開2018 - 198421 (JP, A)

国際公開第2020 / 003037 (WO, A1)

国際公開第2019 / 142164 (WO, A1)

国際公開第2019 / 055963 (WO, A1)

国際公開第2018 / 130491 (WO, A1)

国際公開第2017 / 104115 (WO, A1)

米国特許出願公開第2018 / 0026870 (US, A1)

Vida Fakour Sevom, et al., Geometry-Guided 3D Data Interpolation for Projection-Based Dynamic Point Cloud Coding, 2018 7th European Workshop on Visual Information Processing (EUVIP), IEEE, 2018年11月26日, pp.1-6

Dong Liu, Tong Shao, Houqiang Li, and Feng Wu, Three-Dimensional Point-Cloud Plus Patches: Towards Model-Based Image Coding in the Cloud, 2015 IEEE International Conference on Multimedia Big Data, IEEE, 2015年, pp.395-400

Rufael Mekuria, Kees Blom, and Pablo Cesar, Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE, 2017年04月, VOL. 27, NO. 4, pp.828-842

(58)調査した分野 (Int.Cl., DB名)

H 0 4 N 1 9 / 0 0 - 1 9 / 9 8