

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号  
特許第7573555号  
(P7573555)

(45)発行日 令和6年10月25日(2024.10.25)

(24)登録日 令和6年10月17日(2024.10.17)

(51)国際特許分類	F I
G 0 6 T 9/00 (2006.01)	G 0 6 T 9/00
H 0 4 N 19/597 (2014.01)	H 0 4 N 19/597

請求項の数 4 (全37頁)

(21)出願番号	特願2022-1928(P2022-1928)	(73)特許権者	000208891 K D D I 株式会社 東京都新宿区西新宿二丁目 3 番 2 号
(22)出願日	令和4年1月7日(2022.1.7)	(74)代理人	110001564 フェリシテ弁理士法人
(65)公開番号	特開2023-101340(P2023-101340 A)	(72)発明者	海野 恭平 埼玉県ふじみ野市大原二丁目 1 番 1 5 号 株式会社 K D D I 総合研究所内
(43)公開日	令和5年7月20日(2023.7.20)	(72)発明者	河村 圭 埼玉県ふじみ野市大原二丁目 1 番 1 5 号 株式会社 K D D I 総合研究所内
審査請求日	令和6年1月26日(2024.1.26) (出願人による申告)令和元年度、総務省、「多様な用途、環境下での高精細映像の活用」に資する次世代映像伝送・通信技術の研究開発」委託事業、産業技術力強化法第17条の適用を受ける特許出願	審査官	岩井 健二

最終頁に続く

(54)【発明の名称】 点群復号装置、点群復号方法及びプログラム

## (57)【特許請求の範囲】

## 【請求項1】

点群復号装置であって、複数の投影面候補の中から、ノードの各辺上に存在する複数の頂点を前記投影面候補の各々に投影した際に前記頂点で定義される多角形の面積が最大となる投影面を、投影面として選択するように構成されている近似表面合成部を備え、前記近似表面合成部は、所定の原点と、前記投影面候補に投影した前記頂点のうち1つである第1頂点と、前記第1頂点と隣り合う第2頂点と、の三点で構成される三角形の面積を、隣り合う全ての頂点のペアに対して算出することで、前記多角形の面積を算出するように構成されていることを特徴とする点群復号装置。

## 【請求項2】

前記近似表面合成部は、前記投影面候補に投影した前記頂点を反時計回り或いは時計回りに並べ替え、前記並べ替えた順に連続する2つの頂点を前記第1頂点及び前記第2頂点とするように構成されている請求項1に記載の点群復号装置。

## 【請求項3】

点群復号方法であって、複数の投影面候補の中から、ノードの各辺上に存在する複数の頂点を前記投影面候補の各々に投影した際に前記頂点で定義される多角形の面積が最大となる投影面を、投影面として選択する工程を備え、前記工程において、所定の原点と、前記投影面候補に投影した前記頂点のうち1つであ

10

20

る第1頂点と、前記第1頂点と隣り合う第2頂点と、の三点で構成される三角形の面積を、隣り合う全ての頂点のペアに対して算出することで、前記多角形の面積を算出することを特徴とする点群復号方法。

【請求項4】

コンピュータを、点群復号装置として機能させるプログラムであって、

前記点群復号装置は、複数の投影面候補の中から、ノードの各辺上に存在する複数の頂点を前記投影面候補の各々に投影した際に前記頂点で定義される多角形の面積が最大となる投影面を、投影面として選択するように構成されている近似表面合成部を備え、

前記近似表面合成部は、所定の原点と、前記投影面候補に投影した前記頂点のうち1つである第1頂点と、前記第1頂点と隣り合う第2頂点と、の三点で構成される三角形の面積を、隣り合う全ての頂点のペアに対して算出することで、前記多角形の面積を算出するように構成されていることを特徴とするプログラム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、点群復号装置、点群復号方法及びプログラムに関する。

【背景技術】

【0002】

非特許文献1では、予め定めた1種類のノードサイズにおいてTrisoupと呼ばれる方法で復号点群を再構成する技術が開示されている。

20

【先行技術文献】

【非特許文献】

【0003】

【文献】G-PCC Future Enhancement、ISO/IEC JTC1/SC29/WG11 N19328

【発明の概要】

【発明が解決しようとする課題】

【0004】

しかしながら、非特許文献1の方法では、ノードサイズが1種類のみ（固定ノードサイズ）であり、符号化効率の観点で改善の余地があるという問題点があった。

30

【0005】

そこで、本発明は、上述の課題に鑑みてなされたものであり、復号点群の主観画質を損なうことなく、符号化効率を高めることができる点群復号装置、点群復号方法及びプログラムを提供することを目的とする。

【課題を解決するための手段】

【0006】

本発明の第1の特徴は、点群復号装置であって、複数のノードサイズ毎に頂点の復号処理を実行し、単一のノードサイズで前記頂点に基づいた点の再構成処理を実行するように構成されている近似表面合成部を備えることを要旨とする。

【0007】

本発明の第2の特徴は、点群復号方法であって、複数のノードサイズ毎に頂点の復号処理を実行する工程と、単一のノードサイズで前記頂点に基づいた点の再構成処理を実行する工程とを備えることを要旨とする。

40

【0008】

本発明の第3の特徴は、コンピュータを、点群復号装置として機能させるプログラムであって、前記点群復号装置は、複数のノードサイズ毎に頂点の復号処理を実行し、単一のノードサイズで前記頂点に基づいた点の再構成処理を実行するように構成されている近似表面合成部を備えることを要旨とする。

【発明の効果】

【0009】

50

本発明によれば、復号点群の主観画質を損なうことなく、符号化効率を高めることができる点群復号装置、点群復号方法及びプログラムを提供することができる。

【図面の簡単な説明】

【0010】

【図1】図1は、一実施形態に係る点群処理システム10の構成の一例を示す図である。

【図2】図2は、一実施形態に係る点群復号装置200の機能ブロックの一例を示す図である。

【図3】図3は、一実施形態に係る点群復号装置200の幾何情報復号部2010で受信する符号化データ(ビットストリーム)の構成の一例を示す図である。

【図4】図4は、GPS2011のシンタックス構成の一例を示す図である。

10

【図5】図5は、GSH2012のシンタックス構成の一例を示す図である。

【図6】図6は、一実施形態に係る点群復号装置200のツリー合成部2020における処理の一例を示すフローチャートである

【図7】図7は、一実施形態に係る点群復号装置200のツリー合成部2020によるノード毎のoccupancy codeの復号処理の一例を示すフローチャートである。

【図8】図8は、一実施形態に係る点群復号装置200のツリー合成部2020によるインター予測を用いて4種類のコンテキスト(no pred/pred0/pred1/predL)からいずれか1つを選択する処理の一例を示すフローチャートである。

【図9】図9は、一実施形態に係る点群復号装置200の近似表面合成部2030の処理の一例を示すフローチャートである。

20

【図10】図10は、一実施形態に係る点群復号装置200の近似表面合成部2030によるTrisoupの頂点位置の復号処理の一例を示すフローチャートである。

【図11】図11は、図10におけるステップS1006の処理の一例を説明するための図である。

【図12】図12は、図10におけるステップS1006の処理の一例を説明するための図である。

【図13】図13は、図10におけるステップS1006の処理の別の一例を説明するための図である。

【図14】図14は、図10におけるステップS1006の処理の別の一例を説明するための図である。

30

【図15】図15は、図9におけるステップS902の処理の一例を説明するための図である。

【図16】図16は、図9におけるステップS902の処理の一例を説明するための図である。

【図17】図17は、図9におけるステップS902の処理の別の一例を説明するための図である。

【図18】図18は、図9におけるステップS902の処理の別の一例を説明するための図である。

【図19】図19は、図9におけるステップS902の処理の別の一例を説明するための図である。

40

【図20】図20は、図9におけるステップS902の処理の別の一例を説明するための図である。

【図21】図21は、一実施形態に係る点群符号化装置100の機能ブロックの一例を示す図である。

【図22】図22は、図9におけるステップS902の処理の一例を説明するための図である。

【発明を実施するための形態】

【0011】

以下、本発明の実施の形態について、図面を参照しながら説明する。なお、以下の実施形態における構成要素は、適宜、既存の構成要素等との置き換えが可能であり、また、他

50

の既存の構成要素との組み合わせを含む様々なバリエーションが可能である。したがって、以下の実施形態の記載をもって、特許請求の範囲に記載された発明の内容を限定するものではない。

【0012】

(第1実施形態)

以下、図1～図2を参照して、本発明の第1実施形態に係る点群処理システム10について説明する。図1は、本実施形態に係る実施形態に係る点群処理システム10を示す図である。

【0013】

図1に示すように、点群処理システム10は、点群符号化装置100及び点群復号装置200を有する。 10

【0014】

点群符号化装置100は、入力点群信号を符号化することによって符号化データ(ビットストリーム)を生成するように構成されている。点群復号装置200は、ビットストリームを復号することによって出力点群信号を生成するように構成されている。

【0015】

なお、入力点群信号及び出力点群信号は、点群内の各点の位置情報と属性情報とから構成される。属性情報は、例えば、各点の色情報や反射率である。

【0016】

ここで、かかるビットストリームは、点群符号化装置100から点群復号装置200に対して伝送路を介して送信されてもよい。また、ビットストリームは、記憶媒体に格納された上で、点群符号化装置100から点群復号装置200に提供されてもよい。 20

【0017】

(点群復号装置200)

以下、図2を参照して、本実施形態に係る点群復号装置200について説明する。図2は、本実施形態に係る点群復号装置200の機能ブロックの一例について示す図である。

【0018】

図2に示すように、点群復号装置200は、幾何情報復号部2010と、ツリー合成部2020と、近似表面合成部2030と、幾何情報再構成部2040と、逆座標変換部2050と、属性情報復号部2060と、逆量子化部2070と、RAHT部2080と、LoD算出部2090と、逆リフティング部2100と、逆色変換部2110と、フレームバッファ2120とを有する。 30

【0019】

幾何情報復号部2010は、点群符号化装置100から出力されるビットストリームのうち、幾何情報に関するビットストリーム(幾何情報ビットストリーム)を入力とし、シンタックスを復号するように構成されている。

【0020】

復号処理は、例えば、コンテキスト適応二値算術復号処理である。ここで、例えば、シンタックスは、位置情報の復号処理を制御するための制御データ(フラグやパラメータ)を含む。 40

【0021】

ツリー合成部2020は、幾何情報復号部2010によって復号された制御データ及び後述するツリー内のどのノードに点群が存在するかを示すoccupancy codeを入力として、復号対象空間内のどの領域に点が存在するかというツリー情報を生成するように構成されている。

【0022】

なお、occupancy codeの復号処理をツリー合成部2020内部で行うよう構成されていてもよい。

【0023】

本処理は、復号対象空間を直方体で区切り、occupancy codeを参照して 50

各直方体内に点が存在するかを判断し、点が存在する直方体を複数の直方体に分割し、occupancy codeを参照するという処理を再帰的に繰り返すことで、ツリー情報を生成することができる。

【0024】

ここで、かかるoccupancy codeの復号に際して、後述するインター予測を用いてもよい。

【0025】

本実施形態では、上述の直方体を常に立方体として8分木分割を再帰的に行う「Octree」と呼ばれる手法、及び、8分木分割に加え、4分木分割及び2分木分割を行う「QtBt」と呼ばれる手法を使用することができる。QtBtを使用するか否かは、制御データとして点群符号化装置100側から伝送される。

10

【0026】

或いは、制御データによってPredictive codingを使用するように指定された場合、ツリー合成部2020は、点群符号化装置100において決定した任意のツリー構成に基づいて各点の座標を復号するように構成されている。

【0027】

近似表面合成部2030は、ツリー合成部2020によって生成されたツリー情報を用いて近似表面情報を生成し、かかる近似表面情報に基づいて点群を復号するように構成されている。

【0028】

近似表面情報は、例えば、物体の3次元点群データを復号する際等において、点群が物体表面に密に分布しているような場合に、個々の点群を復号するのではなく、点群の存在領域を小さな平面で近似して表現したものである。

20

【0029】

具体的には、近似表面合成部2030は、例えば、「Trisoup」と呼ばれる手法で、近似表面情報を生成し、点群を復号することができる。「Trisoup」の具体的な処理例については後述する。また、Lidar等で取得した疎な点群を復号する場合は、本処理を省略することができる。

【0030】

幾何情報再構成部2040は、ツリー合成部2020によって生成されたツリー情報及び近似表面合成部2030によって生成された近似表面情報を元に、復号対象の点群データの各点の幾何情報(復号処理が仮定している座標系における位置情報)を再構成するように構成されている。

30

【0031】

逆座標変換部2050は、幾何情報再構成部2040によって再構成された幾何情報を入力として、復号処理が仮定している座標系から、出力点群信号の座標系に変換を行い、位置情報を入力するように構成されている。

【0032】

フレームバッファ2120は、幾何情報再構成部2040によって再構成された幾何情報を入力として、参照フレームとして保存するように構成されている。保存した参照フレームは、ツリー合成部2020において時間的に異なるフレームのインター予測を行う場合に、フレームバッファ2120から読み出されて参照フレームとして使用される。

40

【0033】

ここで、各フレームに対してどの時刻の参照フレームを用いるかどうかは、例えば、点群符号化装置100からビットストリームとして伝送されてくる制御データに基づいて決定されてもよい。

【0034】

属性情報復号部2060は、点群符号化装置100から出力されるビットストリームのうち、属性情報に関するビットストリーム(属性情報ビットストリーム)を入力とし、シントックスを復号するように構成されている。

50

## 【0035】

復号処理は、例えば、コンテキスト適応二値算術復号処理である。ここで、例えば、シンタックスは、属性情報の復号処理を制御するための制御データ（フラグ及びパラメータ）を含む。

## 【0036】

また、属性情報復号部2060は、復号したシンタックスから、量子化済み残差情報を復号するように構成されている。

## 【0037】

逆量子化部2070は、属性情報復号部2060によって復号された量子化済み残差情報と、属性情報復号部2060によって復号された制御データの一つである量子化パラメータとを元に、逆量子化処理を行い、逆量子化済み残差情報を生成するように構成されている。

10

## 【0038】

逆量子化済み残差情報は、復号対象の点群の特徴に応じて、RAHT部2080及びLoD算出部2090のいずれかに出力される。いずれに出力されるかは、属性情報復号部2060によって復号される制御データによって指定される。

## 【0039】

RAHT部2080は、逆量子化部2070によって生成された逆量子化済み残差情報及び幾何情報再構成部2040によって生成された幾何情報を入力とし、RAHT (Region Adaptive Hierarchical Transform) と呼ばれるHaar変換（復号処理においては、逆Haar変換）の一種を用いて、各点の属性情報を復号するように構成されている。RAHTの具体的な処理としては、例えば、文献1 ([PCC] An exploratory model for inter geometry-based PCC, ISO/IEC JTC1/SC29/WG11 m44754) に記載の方法を用いることができる。

20

## 【0040】

LoD算出部2090は、幾何情報再構成部2040によって生成された幾何情報を入力とし、LoD (Level of Detail) を生成するように構成されている。

## 【0041】

LoDは、ある点の属性情報から、他のある点の属性情報を予測し、予測残差を符号化或いは復号するといった予測符号化を実現するための参照関係（参照する点及び参照される点）を定義するための情報である。

30

## 【0042】

言い換えると、LoDは、幾何情報に含まれる各点を複数のレベルに分類し、下位のレベルに属する点については上位のレベルに属する点の属性情報を用いて属性を符号化或いは復号するといった階層構造を定義した情報である。

## 【0043】

LoDの具体的な決定方法としては、例えば、上述の文献1に記載の方法を用いてもよい。

## 【0044】

逆リフティング部2100は、LoD算出部2090によって生成されたLoD及び逆量子化部2070によって生成された逆量子化済み残差情報を用いて、LoDで規定した階層構造に基づいて各点の属性情報を復号するように構成されている。逆リフティングの具体的な処理としては、例えば、上述の文献1に記載の方法を用いることができる。

40

## 【0045】

逆色変換部2110は、復号対象の属性情報が色情報であり且つ点群符号化装置100側で色変換が行われていた場合に、RAHT部2080又は逆リフティング部2100から出力される属性情報に逆色変換処理を行うように構成されている。かかる逆色変換処理の実行の有無については、属性情報復号部2060によって復号された制御データによって決定される。

50

## 【 0 0 4 6 】

点群復号装置 2 0 0 は、以上の処理により、点群内の各点の属性情報を復号して出力するように構成されている。

## 【 0 0 4 7 】

( 幾何情報復号部 2 0 1 0 )

以下、図 3 ~ 図 5 を用いて幾何情報復号部 2 0 1 0 で復号される制御データについて説明する。

## 【 0 0 4 8 】

図 3 は、幾何情報復号部 2 0 1 0 で受信する符号化データ ( ビットストリーム ) の構成の一例である。

## 【 0 0 4 9 】

第 1 に、ビットストリームは、GPS 2 0 1 1 を含んでいてもよい。GPS 2 0 1 1 は、ジオメトリパラメータセットとも呼ばれ、幾何情報の復号に関する制御データの集合である。具体例については後述する。各 GPS 2 0 1 1 は、複数の GPS 2 0 1 1 が存在する場合に個々を識別するための GPS id 情報を少なくとも含む。

## 【 0 0 5 0 】

第 2 に、ビットストリームは、GSH 2 0 1 2 A / 2 0 1 2 B を含んでいてもよい。GSH 2 0 1 2 A / 2 0 1 2 B は、ジオメトリスライスヘッダ或いはジオメトリデータユニットヘッダとも呼ばれ、後述するスライスに対応する制御データの集合である。以降では、スライスという呼称を用いて説明するが、スライスをデータユニットと読み替えることもできる。具体例については後述する。GSH 2 0 1 2 A / 2 0 1 2 B は、各 GSH 2 0 1 2 A / 2 0 1 2 B に対応する GPS 2 0 1 1 を指定するための GPS id 情報を少なくとも含む。

## 【 0 0 5 1 】

第 3 に、ビットストリームは、GSH 2 0 1 2 A / 2 0 1 2 B の次に、スライスデータ 2 0 1 3 A / 2 0 1 3 B を含んでいてもよい。スライスデータ 2 0 1 3 A / 2 0 1 3 B には、幾何情報を符号化したデータが含まれている。スライスデータ 2 0 1 3 A / 2 0 1 3 B の一例としては、後述する occupancy code が挙げられる。

## 【 0 0 5 2 】

以上のように、ビットストリームは、各スライスデータ 2 0 1 3 A / 2 0 1 3 B に、1 つずつ GSH 2 0 1 2 A / 2 0 1 2 B 及び GPS 2 0 1 1 が対応する構成となる。

## 【 0 0 5 3 】

上述のように、GSH 2 0 1 2 A / 2 0 1 2 B にて、どの GPS 2 0 1 1 を参照するかを GPS id 情報で指定するため、複数のスライスデータ 2 0 1 3 A / 2 0 1 3 B に対して共通の GPS 2 0 1 1 を用いることができる。

## 【 0 0 5 4 】

言い換えると、GPS 2 0 1 1 は、スライスごとに必ずしも伝送する必要がない。例えば、図 3 のように、GSH 2 0 1 2 B 及びスライスデータ 2 0 1 3 B の直前では、GPS 2 0 1 1 を符号化しないようなビットストリームの構成とすることもできる。

## 【 0 0 5 5 】

なお、図 3 の構成は、あくまで一例である。各スライスデータ 2 0 1 3 A / 2 0 1 3 B に、GSH 2 0 1 2 A / 2 0 1 2 B 及び GPS 2 0 1 1 が対応する構成となっていれば、ビットストリームの構成要素として、上述以外の要素が追加されてもよい。

## 【 0 0 5 6 】

例えば、図 3 に示すように、ビットストリームは、シーケンスパラメータセット ( SPS ) 2 0 0 1 を含んでいてもよい。また、同様に、伝送に際して、図 3 と異なる構成に整形されてもよい。更に、後述する属性情報復号部 2 0 6 0 で復号されるビットストリームと合成して単一のビットストリームとして伝送されてもよい。

## 【 0 0 5 7 】

図 4 は、GPS 2 0 1 1 のシンタックス構成の一例である。

10

20

30

40

50

## 【0058】

なお、以下で説明するシンタックス名は、あくまで一例である。以下で説明したシンタックスの機能が同様であれば、シンタックス名は異なっても差し支えない。

## 【0059】

GPS 2011は、各GPS 2011を識別するためのGPS id情報(gps\_geom\_parameter\_set\_id)を含んでもよい。

## 【0060】

なお、図4のDescriptor欄は、各シンタックスが、どのように符号化されているかを意味している。u(v)は、符号無し0次指数ゴロム符号であることを意味し、u(1)は、1ビットのフラグであることを意味する。

10

## 【0061】

GPS 2011は、ツリー合成部2020でインター予測を行うか否かを制御するフラグ(interprediction\_enabled\_flag)を含んでもよい。

## 【0062】

例えば、interprediction\_enabled\_flagの値が「0」の時はインター予測を行わないと定義し、interprediction\_enabled\_flagの値が「1」の時はインター予測を行うと定義してもよい。

## 【0063】

なお、interprediction\_enabled\_flagは、GPS 2011ではなくSPS 2001に含んでいてもよい。

20

## 【0064】

GPS 2011は、近似表面合成部2030でTrisoupを使用するか否かを制御するフラグ(trisoup\_enabled\_flag)を含んでもよい。

## 【0065】

例えば、trisoup\_enabled\_flagの値が「0」の時はTrisoupを使用しないと定義し、trisoup\_enabled\_flagの値が「1」の時はTrisoupを使用すると定義してもよい。

## 【0066】

幾何情報復号部2020は、Trisoupを使用するとき、すなわち、trisoup\_enabled\_flagの値が「1」の時、追加で以下のシンタックスを復号するように構成されていてもよい。

30

## 【0067】

なお、trisoup\_enabled\_flagは、GPS 2011ではなくSPS 2001に含まれていてもよい。

## 【0068】

GPS 2011は、複数レベルでのTrisoupを許可するか否かを制御するフラグ(trisoup\_multilevel\_enabled\_flag、第1フラグ)を含んでもよい。

## 【0069】

例えば、trisoup\_multilevel\_enabled\_flagの値が「0」の時は複数レベルでのTrisoupを許可しない、すなわち、単一のレベルでのTrisoupを行うと定義し、trisoup\_multilevel\_enabled\_flagの値が「1」の時は複数レベルでのTrisoupを許可すると定義してもよい。

40

## 【0070】

なお、当該シンタックスがGPS 2011に含まれない場合、当該シンタックスの値を単一のレベルでのTrisoupを行う場合の値、すなわち、「0」とみなしてよい。

## 【0071】

なお、trisoup\_multilevel\_enabled\_flagは、GPS 2011ではなくSPS 2001に含まれるよう定義してもよい。この場合、SPS 2001にtrisoup\_multilevel\_enabled\_flagが含まれていない

50

場合、当該シンタックスの値を単一のレベルでの `Trisoup` を行う場合の値、すなわち、「0」とみなしてよい。

【0072】

図5は、`GSH2012` のシンタックス構成の一例である。なお、前述の通り、`GSH` は、`GDUH` (`Geometry Data Unit Header`) と呼ばれる。

【0073】

幾何情報復号部2020は、複数レベルでの `Trisoup` を許可するとき、すなわち、`trisoup_multilevel_enabled_flag` の値が「1」の時、追加で以下のシンタックスを復号するように構成されていてもよい。

【0074】

`GSH2012` は、複数レベルでの `Trisoup` を許可するとき、`Trisoup` ノードサイズの最大値を規定するシンタックス (`log2_trisoup_max_node_size_minus2`) を含んでもよい。

【0075】

当該シンタックスは、実際の `Trisoup` ノードサイズの最大値について、2を底とする対数に変換した値として表現されていてもよい。更に、当該シンタックスは、実際の `Trisoup` ノードサイズの最大値について、2を底とする対数に変換した後に2を引いた後の値として表現されていてもよい。

【0076】

`GSH2012` は、複数レベルでの `Trisoup` を許可するとき、`Trisoup` ノードサイズの最小値を規定するシンタックス (`log2_trisoup_min_node_size_minus2`) を含んでもよい。

【0077】

当該シンタックスは、実際の `Trisoup` ノードサイズの最小値について、2を底とする対数に変換した値として表現されていてもよい。更に、当該シンタックスは、実際の `Trisoup` ノードサイズの最小値について、2を底とする対数に変換した後に2を引いた後の値として表現されていてもよい。

【0078】

また、当該シンタックスの値は、必ず0以上且つ `log2_trisoup_max_node_size_minus2` 以下となるように制約されていてもよい。

【0079】

また、この時、図5に示すように `trisoup_depth` を  $\text{trisoup\_depth} = \log_2 \text{trisoup\_max\_node\_size\_minus2} - \log_2 \text{trisoup\_min\_node\_size\_minus2} + 1$  と定義してもよい。

【0080】

幾何情報復号部2020は、複数レベルでの `Trisoup` を許可しないとき、すなわち、`trisoup_multilevel_enabled_flag` の値が「0」の時、追加で以下のシンタックスを復号するように構成されていてもよい。

【0081】

`GSH2012` は、複数レベルでの `Trisoup` を許可しないとき、かつ、`Trisoup` を使用するとき、`Trisoup` ノードサイズを規定するシンタックス (`log2_trisoup_node_size_minus2`) を含んでもよい。

【0082】

当該シンタックスは、実際の `Trisoup` ノードサイズについて、2を底とする対数に変換した値として表現されていてもよい。更に、当該シンタックスは、実際の `Trisoup` ノードサイズについて、2を底とする対数に変換した後に2を引いた後の値として表現されていてもよい。

【0083】

また、この時、図5に示すように `trisoup_depth` を `trisoup_depth = 1` と定義してもよい。

10

20

30

40

50

## 【0084】

GSH2012は、Trisoupを使用するとき、復号点のサンプリング間隔を制御するシンタックス(`trisoup_sampling_value_minus1`)を含んでもよい。当該シンタックスの具体的な定義は、例えば、上述の文献1に記載の定義と同様とすることができる。

## 【0085】

GSH2012は、Trisoupを使用するとき、かつ、複数レベルでのTrisoupを許可するとき、階層 $i$  ( $i = 0, \dots, trisoup\_depth - 1$ )ごとにユニークセグメントが対象階層に存在するか否かを示すフラグ(`unique_segments_exist_flag[i]`)を含んでもよい。例えば、`unique_segments_exist_flag[i]`の値が「1」の場合は、階層 $i$ に少なくとも1つ以上のユニークセグメントが存在することを意味する。また、`unique_segments_exist_flag[i]`の値が「0」の場合は、階層 $i$ にユニークセグメントが1つも存在しないことを意味する。

10

## 【0086】

GSH2012は、階層 $i$  ( $i = 0, \dots, trisoup\_depth - 1$ )ごとにユニークセグメントが対象階層に存在する場合、すなわち、`unique_segments_exist_flag[i]`の値が「1」の場合、追加で対象階層のユニークセグメント数を示すシンタックスのビット数を示すシンタックス(`num_unique_segments_bits_minus1[i]`)及び対象階層のユニークセグメント数を示すシンタックス(`num_unique_segments_minus1[i]`)を含んでもよい。

20

## 【0087】

ここで、`num_unique_segments_bits_minus1[i]`及び`num_unique_segments_minus1[i]`の両方について、それぞれ本来の値から「1」を引いた値をシンタックスの値として符号化してもよい。

## 【0088】

(ツリー合成部2020)

以下、図6～図8を用いてツリー合成部2020の処理について説明する。図6は、ツリー合成部2020における処理の一例を示すフローチャートである。なお、以下では「Octree」を使用してツリーを合成する場合の例について説明する。

30

## 【0089】

ステップS601において、ツリー合成部2020は、全てのDepthの処理が完了したかどうかを確認する。なお、Depth数は、点群符号化装置100から点群復号装置200に伝送するビットストリーム内に制御データとして含まれていてもよい。

## 【0090】

ツリー合成部2020は、対象Depthのノードサイズを算出する。「Octree」の場合、最初のDepthのノードサイズは、「2のDepth数乗」と定義されてもよい。すなわち、Depth数を $N$ とする場合、最初のDepthのノードサイズは、2の $N$ 乗と定義されてもよい。

40

## 【0091】

また、2番目以降のDepthでのノードサイズは、 $N$ の数を1つずつ減じていくことで定義されてもよい。すなわち、2番目のDepthのノードサイズは、「2の $(N - 1)$ 乗」として定義され、3番目のDepthのノードサイズは、「2の $(N - 2)$ 乗」として定義され、・・・と定義されてもよい。

## 【0092】

或いは、ノードサイズは、常に2のべき乗で定義されるため、単純に指数部分( $N$ 、 $N - 1$ 、 $N - 2$ 、等)の値をノードサイズと考えてもよい。以降の説明では、ノードサイズとは、指数部分の値を指すこととする。

## 【0093】

50

全てのDepthの処理が完了した場合は、ツリー合成部2020は、ステップS609へ進み、全てのDepthの処理が完了していない場合は、ツリー合成部2020は、ステップS602へ進む。

【0094】

言い換えると、対象Depthがn番目のDepthの時、 $(N - n) = 0$ の場合は、ツリー合成部2020は、ステップS609へ進み、 $(N - n) > 0$ の場合は、ツリー合成部2020は、ステップS602へ進む。

【0095】

ここで、Trisoupを使用するか否かを制御するフラグ(trisoup\_enabled\_flag)がTrisoupを使用することを示している場合、すなわち、trisoup\_enabled\_flagの値が「1」の時は、ツリー合成部2020は、処理するDepth数を、Trisoupノードサイズの最小値を規定するシンタックス( $\log_2\_trisoup\_min\_node\_size\_minus2$ )又はTrisoupノードサイズを規定するシンタックス( $\log_2\_trisoup\_node\_size\_minus2$ )の値に基づいて変化させてもよい。かかる場合、例えば、以下のように定義してもよい。

【0096】

処理Depth数 = 全Depth数 - (最小の)Trisoupノードサイズ

ここで、最小のTrisoupノードサイズについては、例えば、( $\log_2\_trisoup\_min\_node\_size\_minus2 + 2$ )で定義できる。同様に、Trisoupノードサイズについては、( $\log_2\_trisoup\_node\_size\_minus2 + 2$ )で定義できる。

【0097】

この場合、全ての処理Depth数の処理が完了した場合は、ツリー合成部2020は、ステップS609へ進み、そうでない場合は、ツリー合成部2020は、ステップS602へ進む。

【0098】

言い換えると、(処理Depth数 - n) = 0の場合は、ツリー合成部2020は、ステップS609へ進み、(処理Depth数 - n) > 0の場合は、ツリー合成部2020は、ステップS602へ進む。

【0099】

また、ツリー合成部2020は、ステップS609へ進む際のノードサイズ(N - 処理Depth数)を持つ全てのノードには、Trisoupが適用されると判定してもよい。

【0100】

ステップS602において、ツリー合成部2020は、対象Depthにて後述するTrisoup\_applied\_flagを復号する必要があるか否かを判定する。

【0101】

例えば、「複数レベルでのTrisoupが許可(trisoup\_multilevel\_enabled\_flagの値が「1」)」で、且つ、「対象Depthのノードサイズ(N - n)が最大のTrisoupノードサイズ以下」の場合に、ツリー合成部2020は、「Trisoup\_applied\_flagの復号が必要である」と判定してもよい。

【0102】

また、ツリー合成部2020は、上述の条件が満たされない場合、「Trisoup\_applied\_flagの復号が必要ない」と判定してもよい。

【0103】

ここで、最大のTrisoupノードサイズについては、例えば、( $\log_2\_trisoup\_max\_node\_size\_minus3 + 2$ )で定義できる。

【0104】

上述の判定が完了したら、ツリー合成部2020は、ステップS703へ移る。

## 【0105】

ステップS603において、ツリー合成部2020は、対象Depthに含まれる全てのノードの処理が完了したかどうかを判定する。

## 【0106】

対象Depthの全てのノードの処理が完了したと判定した場合、ツリー合成部2020は、ステップS601へ移り、次のDepthの処理を行う。

## 【0107】

一方、対象Depthの全てのノードの処理が完了していない場合、ツリー合成部2020は、ステップS604へ移る。

## 【0108】

ステップS604において、ツリー合成部2020は、ステップS602で判定したTrisoup\_applied\_flagの復号の要否について確認する。

## 【0109】

Trisoup\_applied\_flagの復号が必要であると判定されている場合は、ツリー合成部2020は、ステップS605へ進み、Trisoup\_applied\_flagの復号が必要でないと判定されている場合は、ツリー合成部2020は、ステップS608へ移る。

## 【0110】

ステップS605において、ツリー合成部2020は、Trisoup\_applied\_flagを復号する。

## 【0111】

Trisoup\_applied\_flagは、対象ノードにTrisoupを適用するか否かを示す1ビットのフラグ(第2フラグ)である。例えば、かかるフラグの値が「1」の時に対象ノードにTrisoupを適用すると定義し、かかるフラグの値が「0」の時に対象ノードにTrisoupを適用しないと定義してもよい。

## 【0112】

ツリー合成部2020は、Trisoup\_applied\_flagを復号した後、ステップS606へ移る。

## 【0113】

ステップS606において、ツリー合成部2020は、ステップS605で復号したTrisoup\_applied\_flagの値を確認する。

## 【0114】

対象ノードにTrisoupを適用する場合、すなわち、Trisoup\_applied\_flagの値が「1」の場合は、ツリー合成部2020は、ステップS607へ移る。

## 【0115】

対象ノードにTrisoupを適用しない場合、すなわち、Trisoup\_applied\_flagの値が「0」の場合は、ツリー合成部2020は、ステップS608へ移る。

## 【0116】

ステップS607において、ツリー合成部2020は、対象ノードを、Trisoupを適用するノード、すなわち、Trisoupノードとして記憶する。かかる対象ノードに対しては、これ以上、「Octree」によるノードの分割を適用しないこととする。その後、ツリー合成部2020は、ステップS603に進み、次のノードの処理へ移る。

## 【0117】

ステップS608において、ツリー合成部2020は、occupancy codeと呼ばれる情報を復号する。

## 【0118】

occupancy codeは、「Octree」の場合は、対象ノードをx、y、z軸方向にそれぞれ半分に分割して、8つのノード(子ノードと呼ぶ)に分割した際に、それぞれの子ノード内に復号対象の点が含まれているか否かを示す情報である。

10

20

30

40

50

## 【0119】

例えば、occpancy codeは、各子ノードに対して1ビットの情報を割り当て、かかる1ビットの情報が「1」の場合は、かかる子ノード内に復号対象の点が含まれると定義され、かかる1ビットの情報が「0」の場合は、かかる子ノード内に復号対象の点が含まれないと定義されてもよい。

## 【0120】

ツリー合成部2020は、かかるoccpancy codeを復号する際に、各子ノードに復号対象の点が存在する確率を予め推定し、その確率に基づいて各子ノードに対応するビットをエントロピー復号してもよい。

## 【0121】

同様に、点群符号化装置100においては、エントロピー符号化してもよい。更に、かかる確率の推定に、インター予測を用いてもよい。インター予測の具体的な方法としては、例えば、上述の文献1に記載の方法を適用できる。更に、インター予測を行う際の参照点群に、アップサンプルした点群を用いてもよい。

## 【0122】

図7～8を用いて、インター予測を用いる場合のoccpancy codeの復号処理の具体例について説明する。

## 【0123】

図7は、ノード毎のoccpancy codeの復号処理の具体例を示すフローチャートである。言い換えると、図7は、図6のステップS608の処理の具体例を示すフローチャートである。

## 【0124】

図7に示すように、ステップS701において、ツリー合成部2020は、コンテキストの選択を行う。コンテキストは、後段の占有情報復号の際のエントロピー復号において使用する確率分布に対応している。

## 【0125】

図8は、インター予測を用いて4種類のコンテキスト(no pred/pred0/pred1/predL)からいずれか1つを選択する場合のフローチャートの一例である。

## 【0126】

以下、図8を用いてコンテキスト選択の一例を説明する。なお、図8では、コンテキストが4種類の例について説明するが、コンテキストの数は、4種類でなくともよい。

## 【0127】

図8に示すように、ステップS801において、ツリー合成部2020は、インター予測の精度を判定する。

## 【0128】

インター予測の精度については、例えば、対象ノードの親ノードにおける子ノード(Octreeの場合、8つの子ノード)について、選択したコンテキスト及び実際の点の有無によって判断できる。

## 【0129】

例えば、pred0のコンテキストを選択した子ノードについて、実際に点が存在しなかった場合を正解とし、実際に点が存在した場合を不正解とする。一方、pred1又はpredLを選択した子ノードについては、実際に点が存在した場合を正解とし、実際に点が存在しなかった場合を不正解とする。

## 【0130】

ここで、親ノードに属する子ノードについて、正解となった子ノードの数が予め定めた閾値より大きい場合、インター予測の精度は「良い」と判断できる。一方、親ノードに属する子ノードについて、正解となった子ノードの数が予め定めた閾値以下の場合、インター予測の精度は「悪い」と判断できる。

## 【0131】

インター予測の精度が「良い」と判断された場合、ツリー合成部2020は、ステップ

10

20

30

40

50

S 8 0 2へ進む。インター予測の精度が「悪い」と判断された場合、ツリー合成部 2 0 2 0は、ステップS 8 0 5へ進み、対象ノードに属する全ての子ノードについてコンテキストを「no pred」とし、処理を終了する。

【0132】

ステップS 8 0 2において、ツリー合成部 2 0 2 0は、動き補償した参照フレームにおいて、対象ノードと同じ位置の領域に少なくとも1点の参照点が存在するか否かを確認する。

【0133】

対象ノードと同じ位置の領域に少なくとも1点の参照点が存在する場合、ツリー合成部 2 0 2 0は、ステップS 8 0 3へ進む。一方、対象ノードと同じ位置の領域に参照点が1点もない場合、ツリー合成部 2 0 2 0は、ステップS 8 0 5へ進み、対象ノードに属する全ての子ノードについてコンテキストを「no pred」とし、処理を終了する。

10

【0134】

ステップS 8 0 3以降では、ツリー合成部 2 0 2 0は、対象ノードに属する子ノード毎に判定を行う。

【0135】

ステップS 8 0 3において、ツリー合成部 2 0 2 0は、動き補償した参照フレームにおいて、対象子ノードと同じ位置の領域に少なくとも1点の参照点が存在するか否かを確認する。

【0136】

対象子ノードと同じ位置の領域に少なくとも1点の参照点が存在する場合、ツリー合成部 2 0 2 0は、ステップS 8 0 4へ進む。一方、対象子ノードと同じ位置の領域に参照点が1点もない場合、ツリー合成部 2 0 2 0は、ステップS 8 0 6へ進み、対象子ノードのコンテキストを「pred 0」とする。

20

【0137】

対象ノードの全ての子ノードについてコンテキスト選択が完了していない場合、ツリー合成部 2 0 2 0は、ステップS 8 0 3へ戻り、次の子ノードについて同様の処理を行う。

【0138】

対象ノードの全ての子ノードについてコンテキスト選択が完了した場合、ツリー合成部 2 0 2 0は、処理を終了する。

30

【0139】

ステップS 8 0 4において、ツリー合成部 2 0 2 0は、動き補償した参照フレームにおいて、対象子ノードと同じ位置の領域に予め定めた閾値以上の数の参照点が存在するか否かを確認する。

【0140】

対象子ノードと同じ位置の領域に予め定めた閾値以上の数の参照点が存在する場合、ツリー合成部 2 0 2 0は、ステップS 8 0 8へ進み、対象子ノードのコンテキストを「pred L」とする。

【0141】

一方、対象子ノードと同じ位置の領域に予め定めた閾値未満の数の参照点しか存在しない場合、ツリー合成部 2 0 2 0は、ステップS 8 0 7へ進み、対象子ノードのコンテキストを「pred 1」とする。

40

【0142】

ステップS 8 0 7又はS 8 0 8の処理の後、対象ノードの全ての子ノードについてコンテキスト選択が完了していない場合、ツリー合成部 2 0 2 0は、ステップS 8 0 4へ戻り、次の子ノードについて同様の処理を行う。

【0143】

対象ノードの全ての子ノードについてコンテキスト選択が完了した場合、ツリー合成部 2 0 2 0は、処理を終了する。

【0144】

50

以上のようにコンテキストを選択した後、ツリー合成部 2020 は、ステップ S702 へ進む。

【0145】

ステップ S702 において、ツリー合成部 2020 は、ステップ S701 で選択したコンテキストに基づいて、対象ノードの各子ノードの占有情報、すなわち、occupancy code を復号する。

【0146】

ここで、コンテキストは、それぞれ独立した確率分布に対応している。後述するステップ S703 のコンテキスト更新によって、コンテキストごとに復号対象の点が存在する確率が学習される。

【0147】

なお、各コンテキストにおける確率分布の初期値は、例えば、点が存在する確率 50% (= 存在しない確率も 50%) と定義することができる。

【0148】

ステップ S702 において、ツリー合成部 2020 は、コンテキストに対応する確率分布を基にエントロピー復号を行い、各子ノードに点が存在するか(「1」であるか)否か(「0」であるか)を復号する。occupancy code の復号完了後、ツリー合成部 2020 は、ステップ S703 へ進む。

【0149】

ステップ S703 において、ツリー合成部 2020 は、コンテキストを更新する。

【0150】

例えば、ツリー合成部 2020 は、ステップ S703 において、各コンテキストにおいて、占有情報を復号した結果点が存在した場合、各コンテキストに紐づく確率分布を、点が存在する確率が高くなるように更新する。

【0151】

一方、ツリー合成部 2020 は、ステップ S703 において、各コンテキストにおいて、占有情報を復号した結果点が存在しなかった場合、各コンテキストに紐づく確率分布を、点が存在する確率が低くなるように更新する。

【0152】

コンテキストの更新が完了したら、ツリー合成部 2020 は、ステップ S704 へ進み、処理を終了する。

【0153】

ツリー合成部 2020 は、occupancy code を復号した後、ステップ S603 へ進み、次のノードの処理へ移る。

【0154】

(近似表面合成部 2030)

近似表面合成部 2030 は、図 6 で説明したようにツリー合成部 2020 にて Trisoup ノードと判定された各ノードについて、復号処理を行うように構成されている。

【0155】

以下、図 9 ~ 図 20 を用いて近似表面合成部 2030 の処理の例について説明する。

【0156】

図 9 は、近似表面合成部 2030 の処理の一例を示すフローチャートである。

【0157】

図 9 に示すように、ステップ S901 において、近似表面合成部 2030 は、ノード毎の頂点位置を復号する。

【0158】

ここで、近似表面合成部 2030 は、複数レベルでの Trisoup を許可するとき、すなわち、trisoup\_multilevel\_enabled\_flag の値が「1」の時は、最小 Trisoup ノードサイズにおけるノード毎に頂点位置を復号する。

【0159】

10

20

30

40

50

一方、近似表面合成部 2030 は、複数レベルでの `Trisoup` を許可しないとき、すなわち、`trisoup_multilevel_enabled_flag` の値が「0」の時は、`Trisoup` ノードサイズにおけるノード毎に頂点位置を復号する。具体的な処理については後述する。

【0160】

頂点位置の復号が完了した後、近似表面合成部 2030 は、ステップ S902 へ進む。

【0161】

ステップ S902 において、近似表面合成部 2030 は、頂点位置を復号したノード毎（最小 `Trisoup` ノードサイズ、又は、`Trisoup` ノードサイズにおけるノード毎）に、投影面を決定する。

10

【0162】

例えば、`x` 軸、`y` 軸、`z` 軸の 3 つの軸があるとき、どれか一つの軸を縮退した平面を投影面と呼ぶ。

【0163】

ステップ S902 では、近似表面合成部 2030 は、上述の軸のうちどの軸を縮退するか、すなわち、`x-y` 平面、`x-z` 平面、`y-z` 平面のうちどの平面を投影面とするかを決定する。具体的な処理については後述する。

【0164】

投影面を決定した後、近似表面合成部 2030 は、ステップ S903 へ進む。

【0165】

ステップ S903 において、近似表面合成部 2030 は、投影面に投影した頂点を、例えば反時計回りの順になるようにソートし、前記の順序に従って `index` を付与する。

20

【0166】

`index` を付与した後、近似表面合成部 2030 は、ステップ S904 へ進む。

【0167】

ステップ S904 において、近似表面合成部 2030 は、上述の `index` 及び対象ノードに存在する頂点数に基づいて、三角形を生成する。

【0168】

例えば、近似表面合成部 2030 は、予め頂点数毎にどの `index` から三角形を生成するかを定義した表を作成しておき、それを参照することで三角形を生成できる。具体的な表の例については、例えば、上述の文献 1 に記載の表を用いることができる。

30

【0169】

三角形を生成した後、近似表面合成部 2030 は、ステップ S905 へ進む。

【0170】

ステップ S905 において、近似表面合成部 2030 は、ステップ S904 で生成した三角形を基に、点を生成する。具体的な方法は、例えば、上述の文献 1 に記載の方法を用いることができる。

【0171】

全てのノードについて点の生成が完了した後、近似表面合成部 2030 は、ステップ S906 へ進み、処理を終了する。

40

【0172】

次に、ステップ S901 の頂点位置の復号について、処理の具体例を説明する。図 10 は、`Trisoup` の頂点位置の復号処理の一例を示すフローチャートである。

【0173】

図 10 に示すように、ステップ S1001 において、近似表面合成部 2030 は、全 `Trisoup` 階層の処理が完了したかどうかを判定する。ここで、全 `Trisoup` 階層の数は、以下の通り定義できる。

【0174】

複数レベルでの `Trisoup` を許可するとき、すなわち、`trisoup_multilevel_enabled_flag` の値が「1」の時は、全 `Trisoup` 階層数は、

50

(最大 `Trisoup` ノードサイズ - 最小 `Trisoup` ノードサイズ + 1) で定義できる。

【0175】

すなわち、かかる場合、全 `Trisoup` 階層数は、( $\log_2\_trisoup\_max\_node\_size\_minus\ 2 - \log_2\_trisoup\_min\_node\_size\_minus\ 2 + 1$ ) で定義できる。

【0176】

複数レベルでの `Trisoup` を許可しないとき、すなわち、`trisoup\_multilevel\_enabled\_flag` の値が「0」の時は、全 `Trisoup` 階層数は、1 である。

【0177】

全ての `Trisoup` 階層の処理が完了した場合は、近似表面合成部 2030 は、ステップ S1007 へ進む。全ての `Trisoup` 階層の処理が完了していない場合は、近似表面合成部 2030 は、ステップ S1002 へ移る。

【0178】

ステップ S1002 において、近似表面合成部 2030 は、対象 `Trisoup` 階層に属するユニークセグメント数を確認する。

【0179】

ユニークセグメント数が「0」の場合、すなわち、対象 `Trisoup` 階層には `Trisoup` ノードが1つも含まれない場合は、近似表面合成部 2030 は、ステップ S1001 へ進み、次の `Trisoup` 階層の処理へ移る。

【0180】

ユニークセグメント数が「0」より大きい場合は、近似表面合成部 2030 は、ステップ S1003 へ移る。

【0181】

ステップ S1003 において、近似表面合成部 2030 は、各ユニークセグメントに対して `Trisoup` 処理に用いる頂点が含まれるか否かを復号する。

【0182】

なお、各ユニークセグメントに対して存在できる頂点は、1点のみに制限してもよい。この場合、頂点が存在するユニークセグメント数が頂点数と等しいと解釈できる。

【0183】

近似表面合成部 2030 は、対象 `Trisoup` 階層の全てのユニークセグメントについて頂点の有無を復号した後、ステップ S1004 へ移る。

【0184】

ステップ S1004 において、近似表面合成部 2030 は、ステップ S1003 において頂点が存在すると判定された各ユニークセグメントに対して、各ユニークセグメント上のどこに頂点が存在するのかという位置情報を復号する。

【0185】

位置情報は、例えば、対象 `Trisoup` 階層における `Trisoup` ノードサイズが  $L$  ( $2$  の  $L$  乗) の場合、 $L$  ビットの等長符号で符号化されていてもよい。

【0186】

近似表面合成部 2030 は、対象 `Trisoup` 階層において、頂点が存在する全てのユニークセグメントに対して頂点位置を復号した後、ステップ S1006 へ進む。

【0187】

ステップ S1006 において、近似表面合成部 2030 は、対象 `Trisoup` 階層におけるノードサイズが最小 `Trisoup` ノードサイズの場合、何も処理をせず S1001 へ進む。

【0188】

同様に、近似表面合成部 2030 は、複数レベルでの `Trisoup` が許可されていない場合、何も処理をせず、ステップ S1001 へ進む。

10

20

30

40

50

## 【0189】

そうでない場合、すなわち、複数レベルでの *Trisoup* が許可されており、かつ、対象 *Trisoup* 階層におけるノードサイズが最小 *Trisoup* ノードサイズより大きい場合は、近似表面合成部 2030 は、ステップ S1004 で復号した対象 *Trisoup* 階層に対応するノードサイズにおける頂点位置に基づいて、最小 *Trisoup* ノードサイズにおける頂点位置を生成する。具体的な処理例については後述する。

## 【0190】

最小ノードサイズにおける頂点位置生成が完了した後、近似表面合成部 2030 は、ステップ S1001 へ進む。

## 【0191】

ステップ S1007 において、近似表面合成部 2030 は、頂点位置の統合を行う。

## 【0192】

例えば、近似表面合成部 2030 は、ステップ S1006 において、大きなノードサイズでの頂点位置から、最小 *Trisoup* ノードサイズにおける頂点位置を生成した場合、ノードの各辺に頂点が 2 点以上生成される可能性がある。

## 【0193】

もし、頂点が 2 点以上生成されている辺があった場合、近似表面合成部 2030 は、これを 1 点に統合して、辺ごとに頂点が 1 点ずつになるようにする。

## 【0194】

例えば、近似表面合成部 2030 は、辺上に存在する頂点の座標の平均値をとることで、複数の点を 1 点に統合することができる。

## 【0195】

別の例としては、頂点の数が奇数の場合、近似表面合成部 2030 は、座標の中央値を持つ点を選択することで頂点を 1 つに絞ることができる。

## 【0196】

また、頂点の数が偶数の場合は、近似表面合成部 2030 は、座標の中央値を持つ 2 点を抽出した後、かかる 2 点の座標値の平均をとることで頂点を 1 つに絞ることができる。

## 【0197】

全ての辺についてそれぞれ頂点を 1 つずつに統合した後、近似表面合成部 2030 は、ステップ S1008 へ進み、処理を終了する。

## 【0198】

次に、前記ステップ S1006 の処理の具体例について説明する。

## 【0199】

以下、図 11 及び図 12 を用いて、上述のステップ S1006 の処理の一例を説明する。

## 【0200】

図 12 に、ステップ S1004 で復号した対象 *Trisoup* 階層に対応するノードサイズを  $N$  (ノードの一辺の長さが  $2$  の  $N$  乗) として、最小の *Trisoup* ノードサイズを  $N - 1$  とした場合の例を示す。

## 【0201】

ステップ S1006 の処理の目的は、図 12-1 に示す対象 *Trisoup* 階層に対応するノードサイズ (図 12 では、 $N$ ) における頂点 (図 12 では、4 点) から、図 12-3 に示す最小の *Trisoup* ノードサイズ (図 12 では、 $N - 1$ ) における頂点 (図 12 では、9 点) を生成することである。

## 【0202】

図 11 は、上述のステップ S1006 の処理の一例を示すフローチャートである。

## 【0203】

図 11 に示すように、近似表面合成部 2030 は、ステップ S1004 で復号した対象 *Trisoup* 階層に対応するノードサイズにおける頂点位置を入力として、ステップ S1101 ~ S1104 の処理を行う。

## 【0204】

10

20

30

40

50

ステップ S 1 1 0 1 ~ S 1 1 0 4 の処理については、図 9 におけるステップ S 9 0 2 ~ S 9 0 5 と同様な方法で実現できる。

【 0 2 0 5 】

ここで、図 1 2 - 1 は、ステップ S 1 1 0 3 の処理結果の例（すなわち、ステップ S 1 1 0 4 の入力データの例）を示し、図 1 2 - 2 は、ステップ S 1 1 0 4 の処理結果の例を示す。

【 0 2 0 6 】

図 1 2 - 1 に示すように、近似表面合成部 2 0 3 0 は、ステップ S 1 1 0 4 では、図 1 2 - 1 に示す三角形の面と整数座標位置との交点の全てに点を生成する。

【 0 2 0 7 】

近似表面合成部 2 0 3 0 は、図 1 2 - 1 に示す三角形を基に、図 1 2 - 2 に示すように点への変換を行った後、ステップ S 1 1 0 5 へ進む。

【 0 2 0 8 】

ステップ S 1 1 0 5 において、近似表面合成部 2 0 3 0 は、ステップ S 1 1 0 4 で生成した点から、最小の `Trisoup` ノードサイズにおける頂点位置を決定する。

【 0 2 0 9 】

具体的には、近似表面合成部 2 0 3 0 は、例えば、ステップ S 1 1 0 4 で生成した点の中から、最小の `Trisoup` ノードサイズに対応する各ノードの辺に隣接する点をそれぞれ抽出し、抽出した点の座標の平均値を求めることで、各辺に対する頂点位置を決定できる。

【 0 2 1 0 】

例えば、近似表面合成部 2 0 3 0 は、辺の位置が  $x$  軸方向に沿っている場合、例えば  $(x, y, z) = (a \sim a + 2^{(N-1)}, b, c)$  を満たす位置に辺があった場合、例えば  $(a \sim a + 2^{(N-1)}, b - 1 \sim b, c - 1 \sim c)$  の領域に存在する点を抽出し、それらの点の  $x$  軸座標の平均値  $e$  を算出することで、 $(e, b, c)$  を頂点位置として決定できる。

【 0 2 1 1 】

近似表面合成部 2 0 3 0 は、辺が  $y$  軸方向及び  $z$  軸方向に沿っている場合も、上記と同様に計算できる。

【 0 2 1 2 】

なお、近似表面合成部 2 0 3 0 は、辺に隣接する領域に点が存在しない場合は、かかる辺上には頂点を生成しない。

【 0 2 1 3 】

以上のようにして、近似表面合成部 2 0 3 0 は、例えば、図 1 2 - 3 に示すように、最小 `Trisoup` ノードサイズにおける頂点位置を生成することができる。

【 0 2 1 4 】

次に、図 1 3 及び図 1 4 を用いて、ステップ S 1 1 0 6 の処理の別の具体例について説明する。

【 0 2 1 5 】

図 1 3 は、前記ステップ S 1 0 0 6 の処理の一例を示すフローチャートである。

【 0 2 1 6 】

図 1 3 に示すように、近似表面合成部 2 0 3 0 は、ステップ S 1 0 0 4 で復号した対象 `Trisoup` 階層に対応するノードサイズにおける頂点位置を入力として、ステップ S 1 3 0 1 ~ S 1 3 0 3 の処理を行う。

【 0 2 1 7 】

ステップ S 1 3 0 1 ~ S 1 3 0 3 の処理については、図 9 におけるステップ S 9 0 2 ~ S 9 0 4 と同様な方法で実現できる。

【 0 2 1 8 】

図 1 4 - 1 は、ステップ S 1 3 0 3 の処理結果の例を示している。図 1 4 - 1 から分かる通り、ステップ S 1 3 0 3 の処理結果及び、ステップ S 1 1 0 3 の処理結果は同様である。

10

20

30

40

50

## 【0219】

近似表面合成部2030は、図14-1に示すように、三角形を生成した後、ステップS1304へ進む。

## 【0220】

ステップS1304において、近似表面合成部2030は、ステップS1303で生成した各三角形の面と最小Trisoupノードサイズに対応するノードの各辺との交点に点を生成する。

## 【0221】

図14-2は、ステップS1304の処理結果の一例を示している。近似表面合成部2030は、各辺上に点を生成した後、ステップS1305へ進む。

## 【0222】

ステップS1305において、近似表面合成部2030は、ステップS1304において最小Trisoupノードサイズに対応する各辺上に生成された点を統合する。

## 【0223】

ステップS1304の処理は、各三角形面と辺との交点に点を生成するため、例えば、ある1つの辺に対して2つ以上の三角面が交差していた場合、同一辺上に2つ以上の頂点が生成される可能性がある。

## 【0224】

その場合、近似表面合成部2030は、同一の辺上に存在する点を統合して、辺ごとに頂点を一点とする様に処理してもよい。

## 【0225】

具体的には、例えば、辺上に存在する頂点の座標の平均値をとることで、複数の点を1点に統合することができる。

## 【0226】

別の例としては、頂点の数が奇数の場合、近似表面合成部2030は、座標の中央値を持つ点を選択することで頂点を1つに絞ることができる。

## 【0227】

また、頂点の数が偶数の場合は、近似表面合成部2030は、座標の中央値を持つ2点を抽出した後、かかる2点の座標値の平均をとることで頂点を1つに絞ることができる。

## 【0228】

近似表面合成部2030は、上述のように各辺上の点を統合した後、ステップS1306へ進み、処理を終了する。

## 【0229】

以上のように、本発明における近似表面合成部2030は、複数レベルでのTrisoupを許容する場合、図10のステップS1004に示すように、頂点位置についてはTrisoupノードサイズ毎に復号し、ステップS1005に示すように、上述の頂点位置から最小Trisoupノードサイズにおける頂点位置を生成し、ステップS905に示すように、最小Trisoupノードサイズにおいて点を生成するように構成されていてもよい。

## 【0230】

言い換えると、近似表面合成部2030は、複数のノードサイズ毎に頂点の復号処理を実行し、単一のノードサイズで頂点に基づく点の再構成処理を実行するように構成されていてもよい。

## 【0231】

また、以上のように、近似表面合成部2030は、制御データから復号した最大ノードサイズ及び最小ノードサイズに基づいて、最大と最小との間の各ノードサイズにおいて頂点の復号処理を実行するように構成されており、かかるノードサイズが最小ノードサイズではない場合、復号した頂点に基づいて、最小ノードサイズにおける頂点位置を生成するように構成されており、単一のノードサイズを最小ノードサイズとするように構成されており、最小ノードサイズにおける頂点位置に基づいて、点の再構成処理を行うように構成

10

20

30

40

50

されていてもよい。

このような構成とすることで、頂点位置の復号単位を可変とすることで点群の幾何的な特性に応じて頂点に係る情報の符号量を削減しつつ、点の再構成については単一且つ小さなノードサイズで実行することで、復号点群の主観品質を高めることができる。

【0232】

また、以上のように、近似表面合成部2030は、ノードサイズが最小ノードサイズではない場合、復号した頂点に基づいて点を生成し、生成した点のうち、かかるノードサイズのノードを最小ノードサイズで分割した際の各辺の近傍に存在する点の座標値に基づいて、最小ノードサイズにおける頂点位置を生成するように構成されていてもよい。

【0233】

このように、近似表面合成部2030における点の再構成処理の一部を流用して最小ノードサイズにおける頂点位置を生成することで、処理回路及び処理関数が流用できるため設計コストを下げるができる。

【0234】

また、以上のように、近似表面合成部2030は、ノードサイズが最小ノードサイズではない場合、復号した頂点に基づいて点を生成する際、かかるノードサイズのノードを最小ノードサイズで分割した際の各辺上にのみ点を生成するように構成されていてもよい。このように、生成する点数を最小限とすることで、当該処理に必要なメモリ量や、処理量を削減することができる。

【0235】

また、以上のように、近似表面合成部2030は、ノードサイズが最小ノードサイズではない場合、復号した頂点に基づいて点を生成する際、かかるノードサイズのノードを最小ノードサイズで分割した際の各辺上に複数の点が存在した場合、それらの点を統合し各辺上に頂点が1点ずつとなるように最小ノードサイズにおける頂点位置を生成するように構成されていてもよい。このように、各辺上に頂点数を1点ずつとなるように制限することで、点の再構成処理を簡素化することができる。

【0236】

また、以上のように、近似表面合成部2030は、ある辺上に、かかる辺に隣接する異なるノードサイズのノードから生成された複数の頂点が存在する場合、それらの点を統合し、各辺上に頂点が1点ずつとなるように頂点位置を生成するように構成されていてもよい。このような構成とすることで、点の再構成処理を簡素化することができる。

【0237】

また、以上のように、近似表面合成部2030は、ある辺上に、かかる辺に隣接する異なるノードサイズのノードから生成された複数の頂点が存在する場合、かかる隣接するノードのうち、最も小さなノードサイズを持つノードによって生成された頂点を、かかる辺の頂点位置とするように構成されていてもよい。このように、小さなノードサイズにおける頂点位置を使用するようにすることで、より局所的な特徴を再現した点群の再構成が可能となり、主観品質が向上する。

【0238】

次に、図9におけるステップS902の処理の具体例について説明する。

【0239】

以下、図15及び図16を用いて、図9におけるステップS902の処理の一例を説明する。

図15は、ステップS902の処理の一例を示すフローチャートである。

【0240】

図15に示すように、ステップS1501において、近似表面合成部2030は、頂点を各投影面に投影した際の頂点で形成される多角形の面積をそれぞれ算出する。

【0241】

図16に、ステップS1501の処理の具体例を示す。

【0242】

10

20

30

40

50

例えば、図 1 6 - 1 に示すように、対象ノードについて A ~ E の五つの頂点がある場合を考える。

【 0 2 4 3 】

図 1 6 - 2 ~ 図 1 6 - 4 は、図 1 6 - 1 に示す頂点を、それぞれの投影面に投影した際の図である。

【 0 2 4 4 】

ここで、近似表面合成部 2 0 3 0 は、各投影面における多角形の面積（図 1 6 - 2 ~ 図 1 6 - 4 において、それぞれ網掛けになっている部分の面積）を算出する。

【 0 2 4 5 】

具体的には、近似表面合成部 2 0 3 0 は、各投影面における正方形の中心を原点 O とし、原点 O、原点 O と隣り合う 2 つの頂点（例えば、図 1 6 - 2 における点 E 及び点 D）の計 3 点で構成される三角形の面積 S は、以下の式で算出できる。

【 0 2 4 6 】

$$S = | E \times D | / 2$$

ここで、E 及び D は、原点 O に対する点 E 及び点 D の三次元座標を示すベクトルを意味し、 $\times$  記号は、ベクトルの外積を計算する演算子を意味し、 $| \cdot |$  は、ベクトルの L 2 ノルムを意味している。

【 0 2 4 7 】

近似表面合成部 2 0 3 0 は、上述の事項を応用して、ステップ S 9 0 3 と同様な方法で、各投影面における頂点を、例えば、反時計回りの順になるようにソートし、上述と同様な手法で、全ての隣り合う頂点同士と原点とで作られる三角形の面積を求めたうえで合計することで、図 1 6 - 2 ~ 図 1 6 - 4 の網掛け部分の面積をそれぞれ算出することができる。

【 0 2 4 8 】

以上では、原点 O を各投影面における正方形の中心と定義した場合について説明したが、原点 O は、各投影面の平面上であれば、他の位置に定義しても、上述と同様な方法で多角形の面積を算出することができる。

【 0 2 4 9 】

例えば、近似表面合成部 2 0 3 0 は、原点 O について、投影面における正方形の辺上に位置してもよい。また、例えば、近似表面合成部 2 0 3 0 は、各投影面に投影した頂点のうち的一点を、原点 O と定義してもよい。例えば、近似表面合成部 2 0 3 0 は、各投影面に投影した頂点を反時計回りの順になるようにソートした後、先頭の頂点を原点 O として用いてもよい。

【 0 2 5 0 】

近似表面合成部 2 0 3 0 は、以上のように各投影面における多角形の面積をそれぞれ算出した後、ステップ S 1 5 0 2 へ進む。

【 0 2 5 1 】

ステップ S 1 5 0 2 において、近似表面合成部 2 0 3 0 は、ステップ S 1 5 0 1 で求めた多角形の面積が最も大きい投影面を、投影面として決定する。

【 0 2 5 2 】

近似表面合成部 2 0 3 0 は、投影面を決定した後、ステップ S 1 5 0 3 へ進み、処理を終了する。

【 0 2 5 3 】

なお、近似表面合成部 2 0 3 0 は、図 1 5 に示す手法で投影面を決定する場合、既にステップ S 1 5 0 1 において頂点をソートしているため、ステップ S 9 0 3 の処理について省略することができる。

【 0 2 5 4 】

以上のように、近似表面合成部 2 0 3 0 は、ノードの各辺上に存在する複数の頂点を複数の投影面候補の各々に投影した際に、かかる頂点で定義される多角形の面積が最大となる投影面を、上述の投影面として選択するように構成されてもよい。

【 0 2 5 5 】

10

20

30

40

50

このような構成とすることで、投影面上での2次元的な広がり最も大きな投影面を選択できる。これにより、適切でない投影面を選択することで復号点群の主観画質が劣化することを防止することができる。

【0256】

以上のように、近似表面合成部2030は、所定の原点と、上述の投影面候補に投影した頂点のうち1つ(第1頂点)と、かかる第1頂点と隣り合う第2頂点と、の三点で構成される三角形の面積を、隣り合う全ての頂点のペアに対して算出することで、上述の多角形(三角形)の面積を算出するように構成されてもよい。

【0257】

このように、多角形を小領域に分割して面積を求めることで、それぞれの小領域の面積算出処理を並列に実行できるため、処理速度を向上させることができる。

10

【0258】

また、以上のように、近似表面合成部2030は、上述の原点から第1頂点を指す第1ベクトルと、上述の原点から第2頂点を指す第2ベクトルを定義し、かかる第1ベクトルと第2ベクトルとの外積を用いて、上述の三角形の面積を算出するように構成されてもよい。このように、ベクトルの外積を用いて演算することで、他の処理で外積処理を行う場合は、処理回路又は処理関数を共通化することで設計を簡素化できる。

【0259】

また、以上のように、近似表面合成部2030は、上述の投影面候補に投影した頂点を反時計回り或いは時計回りに並べ替え、並べ替えた順に連続する2つの頂点を第1頂点及び第2頂点としてもよい。このように、後段のステップS903と同様な方法でソートを行うことで、処理を共通化して処理量の増加を防ぐことができる。

20

【0260】

また、以上のように、近似表面合成部2030は、上述の投影面候補に投影した頂点のうち1つ(第3頂点)を、所定の原点とするように構成されてもよい。このような構成とすることで、所定の原点を頂点以外の位置に設定した場合と比較して、多角形の面積を算出するために計算が必要な三角形の個数が1つ少なくなるため、演算量の増加を防ぐことができる。

【0261】

次に、図17~図20を用いて、ステップS902の処理の別の例について説明する。

30

【0262】

図17は、ステップS902の処理の一例を示すフローチャートである。

【0263】

図17に示すように、ステップS1701において、近似表面合成部2030は、頂点座標のx軸、y軸、z軸方向それぞれの最大値と最小値の差分を算出する。

【0264】

例えば、図19-1に示すように、対象ノードにA~Eの計5点の頂点があった場合、近似表面合成部2030は、図19-2に示す $x_{max} - x_{min}$ (x軸方向の座標値の最大値と最小値の差分)、 $y_{max} - y_{min}$ (y軸方向の座標値の最大値と最小値の差分)及び図19-3に示す $z_{max} - z_{min}$ (z軸方向の座標値の最大値と最小値の差分)をそれぞれ算出し、ステップS1702へ進む。

40

【0265】

ステップS1702において、近似表面合成部2030は、ステップS1701で算出した「最大値と最小値の差分」のうちの差分が最小となる軸方向がいくつあるかを確認する。

【0266】

例えば、 $z_{max} - z_{min} < x_{max} - x_{min} < y_{max} - y_{min}$ という関係となっていた場合(或いは、 $z_{max} - z_{min} < x_{max} - x_{min} = y_{max} - y_{min}$ という関係となっていた場合)、「最大値と最小値の差分」の最小値を持つ軸方向は1つのみとなる。この場合は、近似表面合成部2030は、ステップS1703へ進む。

50

## 【0267】

一方、例えば、 $z_{max} - z_{min} = x_{max} - x_{min} < y_{max} - y_{min}$ という関係となっていた場合、「最大値と最小値の差分」の最小値を持つ軸方向は2つとなる。この場合は、近似表面合成部2030は、ステップS1704へ進む。

## 【0268】

ステップS1703は、差分の最小値を持つ軸方向が一つのみのケースに対応する処理であり、この場合は、近似表面合成部2030は、差分が最小となる軸を縮退することで投影面を決定する。

## 【0269】

例えば、上述の例のように、 $z_{max} - z_{min} < x_{max} - x_{min} < y_{max} - y_{min}$ となっていた場合、z軸を縮退して、x-y平面を投影面として決定する。

10

## 【0270】

近似表面合成部2030は、投影面を決定した後、ステップS1706へ進んで処理を終了する。

## 【0271】

ステップS1704において、近似表面合成部2030は、各軸方向の辺上の頂点数をカウントする。

## 【0272】

例えば、図20-1に示すように、対象ノードにA～Eの計5点の頂点があった場合、x軸方向の辺上に存在する頂点は、図20-2に示す点Aのみであり、y軸方向の辺上に存在する頂点は、図20-3に示す通り0点であり、z軸方向の辺上に存在する頂点は、図20-4に示す通りB～Eの計4点である。

20

## 【0273】

なお、この処理は、x、y、z軸のうち全ての軸を対象にしてもよいし、ステップS1701で求めた「最大値と最小値の差分」の最小値を持つ軸方向のみを対象としてもよい。

## 【0274】

例えば、 $z_{max} - z_{min} = x_{max} - x_{min} < y_{max} - y_{min}$ という関係となっていた場合、本ステップの処理対象をz軸及びx軸のみとしてもよい。

## 【0275】

近似表面合成部2030は、上述のように頂点数をカウントした後、ステップS1705へ進む。

30

## 【0276】

ステップS1705において、近似表面合成部2030は、ステップS1704で算出した各軸方向の頂点数が最も大きい軸を縮退することで、投影面を決定する。

## 【0277】

例えば、図20の例では、z軸方向の辺上に存在する頂点が最も多いため、近似表面合成部2030は、z軸を縮退して、x-y平面を投影面として決定する。

## 【0278】

近似表面合成部2030は、投影面決定後、ステップS1706へ進んで、処理を終了する。

40

## 【0279】

以上のように、近似表面合成部2030は、ノードの各辺を三次元座標のどの座標軸と平行かどうかで分類し、各座標軸に分類された辺上の頂点数をカウントし、カウントした値(頂点数)を用いて、複数の投影面候補の中から、上述の投影面を決定するように構成されていてもよい。

## 【0280】

また、以上のように、近似表面合成部2030は、上述の頂点の数が最も大きい軸を縮退することで定義される面を、上述の投影面と決定するように構成されていてもよい。

## 【0281】

このような構成とすることで、図15及び図16を用いて説明した手法よりも少ない演

50

算量で、頂点位置の二次元的な広がりを考慮した投影面の決定が可能となる。

【0282】

また、以上のように、近似表面合成部2030は、上述の三次元座標の座標軸ごとに、各頂点の座標値の最大値と最小値との差分値を算出し、かかる座標軸のうち差分値の最小値を持つ軸が2つ以上ある場合、各座標軸に分類された辺上の頂点数が最も大きい軸を縮退することで定義される面を、上述の投影面とするように構成されていてもよい。

【0283】

また、以上のように、近似表面合成部2030は、各座標軸に分類された辺上の頂点数が最も大きい軸が2つ以上ある場合、三次元座標の座標軸ごとに、各頂点の座標値の最大値と最小値との差分値を算出し、座標軸のうち差分値が最小となる軸を縮退することで定義される面を、上述の投影面とするように構成されていてもよい。

10

【0284】

なお、図17では、近似表面合成部2030は、先にステップS1701及びS1702において「最大値と最小値の差分」を評価した後、必要に応じてステップS1704及びS1705において各軸方向の辺上の頂点数を評価した。この順番は、例えば、図18に示すように、逆にすることもできる。

【0285】

例えば、図18の例に示す通り、近似表面合成部2030は、ステップS1801及びステップS1802において各軸方向の辺上の頂点数を評価した後、必要に応じてステップS1804及びステップS1805において「最大値と最小値の差分」を評価してもよい。

20

【0286】

図18の例では、ステップS1801については、ステップS1704と同様な処理で実現できる。

【0287】

ステップS1802において、近似表面合成部2030は、x軸、y軸、z軸方向の中で最大の頂点数を持つ軸が1つのみである場合は、ステップS1803へ進み、頂点数が最大となる軸を縮退することで投影面を決定する。

【0288】

そうでない場合（頂点数の最大値を持つ軸が2つ以上ある場合）は、近似表面合成部2030は、ステップS1804へ進む。

30

【0289】

ステップS1804については、ステップS1701と同様な処理で実現できる。

【0290】

ここで、ステップS1804の処理対象は、x軸、y軸、z軸の全てとしてもよいし、ステップS1802で頂点数が最大となった軸方向のみを処理対象としてもよい。ステップS1805において、近似表面合成部2030は、ステップS1804の処理対象となった軸の中で、差分値が最も小さい軸を縮退することで、投影面を決定する。

【0291】

次に、図22を用いて、ステップS902の処理の別の例について説明する。図22は、ステップS902の処理の一例を示すフローチャートである。

40

【0292】

図22に示すように、ステップS2201において、近似表面合成部2030は、各投影面に投影した際の各頂点の座標に対して、 $2 \times 2$ の直交変換を行う。

【0293】

以下では、直交変換としてアダマール変換を用いる場合を例に説明する。例えば、座標値として $(P_x, P_y, P_z)$ を持つ頂点Pをx-y平面に投影した際の座標は、 $(P_x, P_y)$ となる。これに対して $2 \times 2$ のアダマール変換を行って得られる直交変換後の座標 $(H_a, H_b)$ は、以下の式によって算出できる。

【0294】

50

【数 1】

$$\begin{pmatrix} H_a \\ H_b \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \end{pmatrix}$$

ここで、 $1/\sqrt{2}$  は、ノルムを正規化するための係数であるが、後述のように、本処理では、各頂点の直交変換後の座標値をその大小関係を比較するために用いるため、正規化係数  $1/\sqrt{2}$  は省略してもよい。すなわち、近似表面合成部 2030 は、以下の式によって、 $(H_a, H_b)$  を算出してもよい。

【0295】

【数 2】

$$\begin{pmatrix} H_a \\ H_b \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \end{pmatrix}$$

上述のように、近似表面合成部 2030 は、直交変換としてアダマール変換を使用し、正規化係数を省略した場合、座標値  $P_x$  と  $P_y$  の加減算のみで直交変換後の座標  $(H_a, H_b)$  を算出できる。

【0296】

このように、まず、近似表面合成部 2030 は、 $x - y$  平面に投影した際の各頂点  $(1 \sim n)$  座標に直交変換を適用して  $n$  組の座標  $(H_{a1}, H_{b1}), \dots, (H_{an}, H_{bn})$  を得た後、直交変換後の各座標軸（ここでは  $a$  軸と  $b$  軸）における座標値の最大値  $H_{amax}$ 、 $H_{bmax}$  と最小値  $H_{amin}$ 、 $H_{bmin}$  をそれぞれ検出する。

【0297】

次に、各軸における最大値と最小値との差分値  $H_{ad} = H_{amax} - H_{amin}$ 、 $H_{bd} = H_{bmax} - H_{bmin}$  を算出する。

【0298】

最後に、近似表面合成部 2030 は、上述の  $H_{ad}$  及び  $H_{bd}$  を用いて、 $x - y$  平面に投影した場合の頂点座標の広がり的大小を表す変数  $A_{xy}$  を算出する。

【0299】

近似表面合成部 2030 は、 $A_{xy}$  について、例えば、 $A_{xy} = H_{ad} + H_{bd}$  のように、直交変換後の座標値の各軸における最大値と最小値の差分を足し合わせることで算出してもよい。

【0300】

また、近似表面合成部 2030 は、 $A_{xy}$  について、例えば、 $A_{xy} = H_{ad} \times H_{bd}$  のように、直交変換後の座標値の各軸における最大値と最小値の差分を掛け合わせることで算出してもよい。

【0301】

上述と同様に、近似表面合成部 2030 は、他の投影面候補である  $y - z$  平面に対応する  $A_{yz}$  及び  $x - z$  平面に対応する  $A_{xz}$  を算出した後、ステップ S2202 へ進む。

【0302】

ステップ S2202 において、近似表面合成部 2030 は、ステップ S2201 で算出した  $A_{xy}$ 、 $A_{yz}$ 、 $A_{xz}$  を基に投影面を決定する。

【0303】

例えば、近似表面合成部 2030 は、 $A_{xy}$ 、 $A_{yz}$ 、 $A_{xz}$  のうち、値が最も大きいものに対応する投影面に決定することができる。具体的には、近似表面合成部 2030 は、例えば、 $A_{yz} > A_{xz} > A_{xy}$  の場合、投影面を  $y - z$  平面と決定することができる。

10

20

30

40

50

## 【0304】

上述のように、近似表面合成部2030は、投影面を決定した後、ステップS2203へ移り、処理を終了する。

## 【0305】

以上のように、近似表面合成部2030は、各投影面に投影した際の頂点座標値に直交変換を適用して得られる直交変換後の座標値を用いて、投影面を決定するように構成されていてもよい。

## 【0306】

また、以上のように、近似表面合成部2030は、直交変換後の各座標軸において座標値の最大値と最小値との差分値を計算し、かかる差分値に基づいて投影面を決定するように構成されていてもよい。

10

## 【0307】

また、以上のように、近似表面合成部2030は、各投影面候補において、上述の差分値を直交変換後の全ての座標軸について算出し、それらの和或いはそれらの積によって投影面を決定するように構成されていてもよい。

## 【0308】

このように直交変換によって座標軸を回転した後の座標値を用いることで、頂点の二次元的な広がりを評価しやすくなる。

## 【0309】

また、以上のように、近似表面合成部2030は、直交変換としてアダマール変換を使用するように構成されていてもよい。

20

## 【0310】

また、以上のように、近似表面合成部2030は、直交変換時の正規化係数を省略するように構成されていてもよい。

## 【0311】

このような構成とすることで、直交変換前の座標値の加減算のみで直交変換後の座標値を得ることができ、計算量の増加を防ぐことができる。

## 【0312】

## (点群符号化装置100)

以下、図21を参照して、本実施形態に係る点群符号化装置100について説明する。図21は、本実施形態に係る点群符号化装置100の機能ブロックの一例について示す図である。

30

## 【0313】

図21に示すように、点群符号化装置100は、座標変換部1010と、幾何情報量子化部1020と、ツリー解析部1030と、近似表面解析部1040と、幾何情報符号化部1050と、幾何情報再構成部1060と、色変換部1070と、属性転移部1080と、RAHT部1090と、LOD算出部1100と、リフティング部1110と、属性情報量子化部1120と、属性情報符号化部1130と、フレームバッファ1140とを有する。

## 【0314】

座標変換部1010は、入力点群の3次元座標系から、任意の異なる座標系への変換処理を行うよう構成されている。座標変換は、例えば、入力点群を回転することにより、入力点群のx、y、z座標を任意のs、t、u座標に変換してもよい。また、変換のバリエーションの1つとして、入力点群の座標系をそのまま使用してもよい。

40

## 【0315】

幾何情報量子化部1020は、座標変換後の入力点群の位置情報の量子化及び座標が重複する点の除去を行うように構成されている。なお、量子化ステップサイズが1の場合は、入力点群の位置情報と量子化後の位置情報とが一致する。すなわち、量子化ステップサイズが1の場合は、量子化を行わない場合と等価になる。

## 【0316】

50

ツリー解析部 1030 は、量子化後の点群の位置情報を入力として、後述のツリー構造に基づいて、符号化対象空間のどのノードに点が存在するかについて示す `occupancy code` を生成するように構成されている。

【0317】

ツリー解析部 1030 は、本処理において、符号化対象空間を再帰的に直方体で区切ることにより、ツリー構造を生成するように構成されている。

【0318】

ここで、ある直方体内に点が存在する場合、かかる直方体を複数の直方体に分割する処理を、直方体が所定のサイズになるまで再帰的に実行することでツリー構造を生成することができる。なお、かかる各直方体をノードと呼ぶ。また、ノードを分割して生成される各直方体を子ノードと呼び、子ノード内に点が含まれるか否かについて 0 又は 1 で表現したものが `occupancy code` である。

【0319】

以上のように、ツリー解析部 1030 は、所定のサイズになるまでノードを再帰的に分割しながら、`occupancy code` を生成するように構成されている。

【0320】

本実施形態では、上述の直方体を常に立方体として 8 分木分割を再帰的に行う「Octree」と呼ばれる手法、及び、8 分木分割に加え、4 分木分割及び 2 分木分割を行う「QtBt」と呼ばれる手法を使用することができる。

【0321】

ここで、「QtBt」を使用するか否かについては、制御データとして点群復号装置 200 に伝送される。

【0322】

或いは、任意のツリー構成を用いる `Predictive coding` を使用するように指定されてもよい。かかる場合、ツリー解析部 1030 が、ツリー構造を決定し、決定されたツリー構造は、制御データとして点群復号装置 200 へ伝送される。

【0323】

例えば、ツリー構造の制御データは、図 6 で説明した手順で復号できるよう構成されていてもよい。

【0324】

近似表面解析部 1040 は、ツリー解析部 1030 によって生成されたツリー情報を用いて、近似表面情報を生成するように構成されている。

【0325】

近似表面情報は、例えば、物体の 3 次元点群データを復号する際等において、点群が物体表面に密に分布しているような場合に、個々の点群を復号するのではなく、点群の存在領域を小さな平面で近似して表現したものである。

【0326】

具体的には、近似表面解析部 1040 は、例えば、「Trisoup」と呼ばれる手法で、近似表面情報を生成するように構成されていてもよい。また、Lidar 等で取得した疎な点群を復号する場合は、本処理を省略することができる。

【0327】

幾何情報符号化部 1050 は、ツリー解析部 1030 によって生成された `occupancy code` 及び近似表面解析部 1040 によって生成された近似表面情報等のシンタックスを符号化してビットストリーム（幾何情報ビットストリーム）を生成するように構成されている。ここで、ビットストリームには、例えば、図 4 及び図 5 で説明したシンタックスが含まれていてもよい。

【0328】

符号化処理は、例えば、コンテキスト適応二値算術符号化処理である。ここで、例えば、シンタックスは、位置情報の復号処理を制御するための制御データ（フラグやパラメータ）を含む。

10

20

30

40

50

## 【 0 3 2 9 】

幾何情報再構成部 1 0 6 0 は、ツリー解析部 1 0 3 0 によって生成されたツリー情報及び近似表面解析部 1 0 4 0 によって生成された近似表面情報に基づいて、符号化対象の点群データの各点の幾何情報（符号化処理が仮定している座標系、すなわち、座標変換部 1 0 1 0 における座標変換後の位置情報）を再構成するように構成されている。

## 【 0 3 3 0 】

フレームバッファ 1 1 4 0 は、幾何情報再構成部 1 0 6 0 によって再構成された幾何情報を入力とし、参照フレームとして保存するように構成されている。

## 【 0 3 3 1 】

保存された参照フレームは、ツリー解析部 1 0 3 0 において時間的に異なるフレームのインター予測を行う場合に、フレームバッファ 1 1 4 0 から読み出されて参照フレームとして使用される。

10

## 【 0 3 3 2 】

ここで、各フレームに対してどの時刻の参照フレームを用いるかが、例えば、符号化効率を表すコスト関数の値に基づいて決定され、使用する参照フレームの情報が制御データとして点群復号装置 2 0 0 へ伝送されてもよい。

## 【 0 3 3 3 】

色変換部 1 0 7 0 は、入力の属性情報が色情報であった場合に、色変換を行うように構成されている。色変換は、必ずしも実行する必要は無く、色変換処理の実行の有無については、制御データの一部として符号化され、点群復号装置 2 0 0 へ伝送される。

20

## 【 0 3 3 4 】

属性転移部 1 0 8 0 は、入力点群の位置情報、幾何情報再構成部 1 0 6 0 における再構成後の点群の位置情報及び色変換部 1 0 7 0 での色変化後の属性情報に基づいて、属性情報の歪みが最小となるように属性値を補正するように構成されている。具体的な補正方法は、例えば、文献 2 (Text of ISO/IEC 23090-9 DIS Geometry-based PCC, ISO/IEC JTC1/SC29/WG11 N19088) に記載の方法を適用できる。

## 【 0 3 3 5 】

RAHT部 1 0 9 0 は、属性転移部 1 0 8 0 による転移後の属性情報及び幾何情報再構成部 1 0 6 0 によって生成された幾何情報を入力とし、RAHT (Region Adaptive Hierarchical Transform) と呼ばれる Haar 変換の一種を用いて、各点の残差情報を生成するように構成されている。RAHT の具体的な処理としては、例えば、上述の文献 2 に記載の方法を用いることができる。

30

## 【 0 3 3 6 】

L o D 算出部 1 1 0 0 は、幾何情報再構成部 1 0 6 0 によって生成された幾何情報を入力とし、L o D (Level of Detail) を生成するように構成されている。

## 【 0 3 3 7 】

L o D は、ある点の属性情報から、他のある点の属性情報を予測し、予測残差を符号化或いは復号するといった予測符号化を実現するための参照関係（参照する点及び参照される点）を定義するための情報である。

40

## 【 0 3 3 8 】

言い換えると、L o D は、幾何情報に含まれる各点を複数のレベルに分類し、下位のレベルに属する点については上位のレベルに属する点の属性情報を用いて属性を符号化或いは復号するといった階層構造を定義した情報である。

## 【 0 3 3 9 】

L o D の具体的な決定方法としては、例えば、上述の文献 2 に記載の方法を用いてもよい。

## 【 0 3 4 0 】

リフティング部 1 1 1 0 は、L o D 算出部 1 1 0 0 によって生成された L o D 及び属性転移部 1 0 8 0 での属性転移後の属性情報を用いて、リフティング処理により残差情報を

50

生成するように構成されている。

【0341】

リフティングの具体的な処理としては、例えば、上述の文献2に記載の方法を用いてもよい。

【0342】

属性情報量子化部1120は、RAHT部1090又はリフティング部1110から出力される残差情報を量子化するように構成されている。ここで、量子化ステップサイズが1の場合は、量子化を行わない場合と等価である。

【0343】

属性情報符号化部1130は、属性情報量子化部1120から出力される量子化後の残差情報等をシンタックスとして符号化処理を行い、属性情報に関するビットストリーム（属性情報ビットストリーム）を生成するように構成されている。

10

【0344】

符号化処理は、例えば、コンテキスト適応二値算術符号化処理である。ここで、例えば、シンタックスは、属性情報の復号処理を制御するための制御データ（フラグ及びパラメータ）を含む。

【0345】

点群符号化装置100は、以上の処理により、点群内の各点の位置情報及び属性情報を入力として符号化処理を行い、幾何情報ビットストリーム及び属性情報ビットストリームを出力するように構成されている。

20

【0346】

また、上述の点群符号化装置100及び点群復号装置200は、コンピュータに各機能（各工程）を実行させるプログラムであって実現されていてもよい。

【0347】

なお、上記の各実施形態では、本発明を点群符号化装置100及び点群復号装置200への適用を例にして説明したが、本発明は、かかる例のみに限定されるものではなく、点群符号化装置100及び点群復号装置200の各機能を備えた点群符号化/復号システムにも同様に適用できる。

【産業上の利用可能性】

【0348】

なお、本実施形態によれば、例えば、動画通信において総合的なサービス品質の向上を実現できることから、国連が主導する持続可能な開発目標（SDGs）の目標9「レジリエントなインフラを整備し、持続可能な産業化を推進するとともに、イノベーションの拡大を図る」に貢献することが可能となる。

30

【符号の説明】

【0349】

10...点群処理システム

100...点群符号化装置

1010...座標変換部

1020...幾何情報量子化部

1030...ツリー解析部

1040...近似表面解析部

1050...幾何情報符号化部

1060...幾何情報再構成部

1070...色変換部

1080...属性転移部

1090...RAHT部

1100...LOD算出部

1110...リフティング部

1120...属性情報量子化部

40

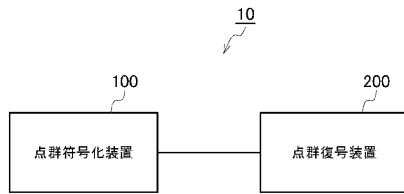
50

- 1 1 3 0 ... 属性情報符号化部
- 1 1 4 0 ... フレームバッファ
- 2 0 0 ... 点群復号装置
- 2 0 1 0 ... 幾何情報復号部
- 2 0 2 0 ... ツリー合成部
- 2 0 3 0 ... 近似表面合成部
- 2 0 4 0 ... 幾何情報再構成部
- 2 0 5 0 ... 逆座標変換部
- 2 0 6 0 ... 属性情報復号部
- 2 0 7 0 ... 逆量子化部
- 2 0 8 0 ... R A H T 部
- 2 0 9 0 ... L o D 算出部
- 2 1 0 0 ... 逆リフティング部
- 2 1 1 0 ... 逆色変換部
- 2 1 2 0 ... フレームバッファ

10

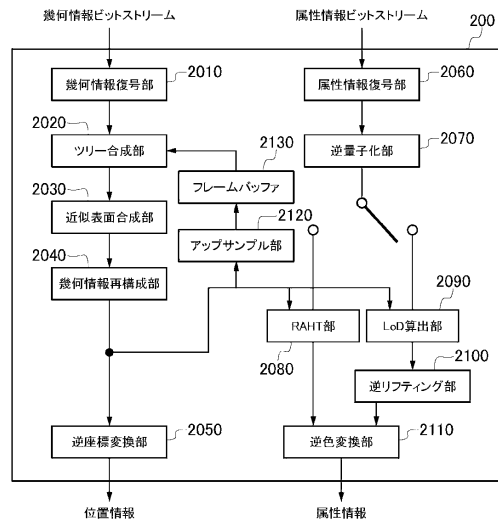
【 図 面 】

【 図 1 】



【 図 2 】

20



30

40

50

【 図 3 】



【 図 4 】

	Descriptor
geometry_parameter_set() {	ue(v)
gps_geom_parameter_set_id	u(1)
...	u(1)
interprediction_enabled_flag	u(1)
trisoup_enabled_flag	u(1)
if (trisoup_enabled_flag) {	
trisoup_multilevel_enabled_flag	u(1)
}	
...	
}	

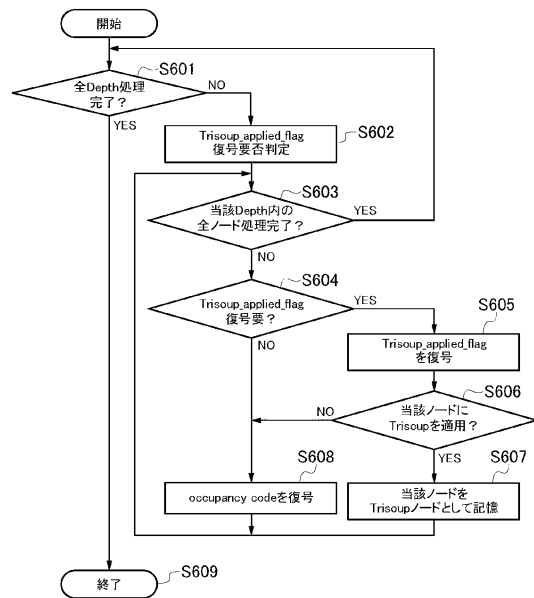
10

20

【 図 5 】

	Descriptor
geometry_data_unit_header() {	ue(v)
gps_geometry_parameter_set_id	u(1)
...	
if (trisoup_enabled_flag) {	
if (trisoup_multilevel_enabled_flag) {	
log2_trisoup_max_node_size_minus2	ue(v)
log2_trisoup_min_node_size_minus2	ue(v)
} else {	
log2_trisoup_max_node_size_minus2 -	
log2_trisoup_min_node_size_minus2 + 1	
}	
log2_trisoup_node_size_minus2	ue(v)
trisoup_depth = 1	
}	
trisoup_sampling_value_minus1	ue(v)
for (i = 0; i < trisoup_depth; i++) {	
if (trisoup_multilevel_enabled_flag)	
unique_segments_exist_flag[i]	u(1)
} else {	
num_unique_segments_bits_minus1[i]	ue(v)
num_unique_segments_minus1[i]	u(v)
}	
}	
...	
}	

【 図 6 】

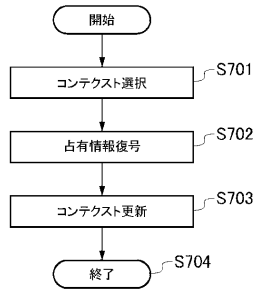


30

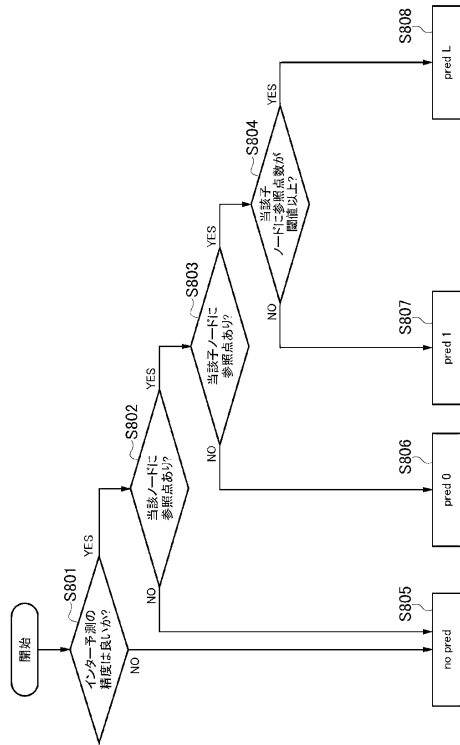
40

50

【図 7】



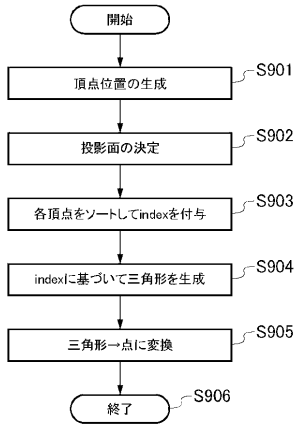
【図 8】



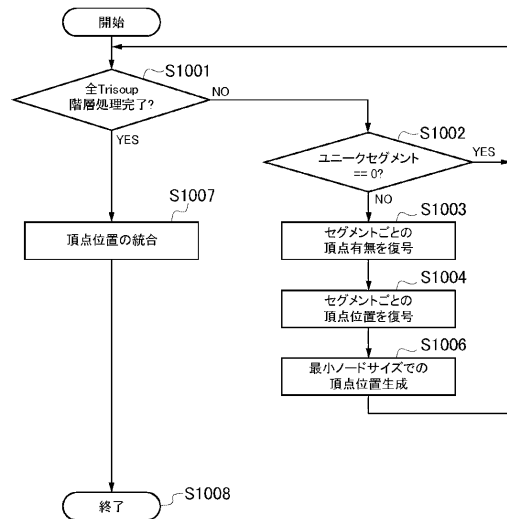
10

20

【図 9】



【図 10】

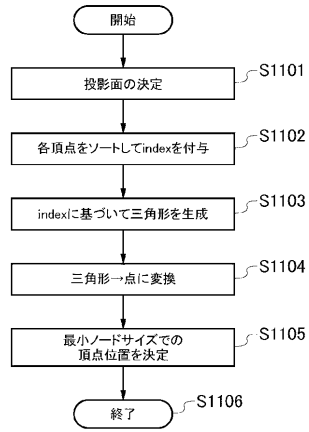


30

40

50

【 図 1 1 】



【 図 1 2 】

図12-1

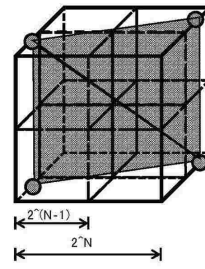


図12-2

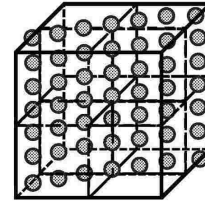
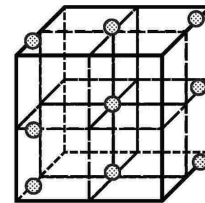


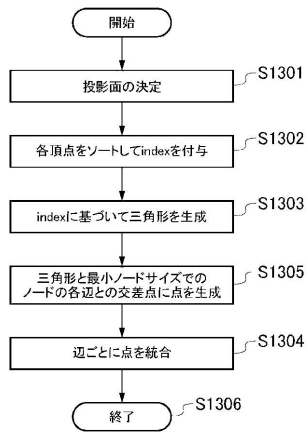
図12-3



10

20

【 図 1 3 】



【 図 1 4 】

図14-1

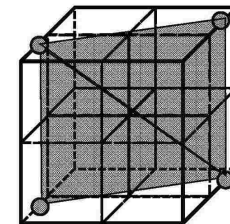
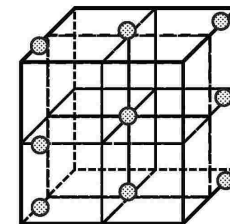


図14-2

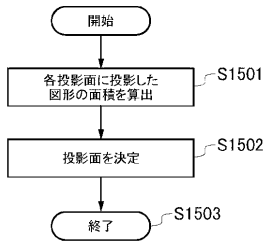


30

40

50

【 図 1 5 】



【 図 1 6 】

図16-1

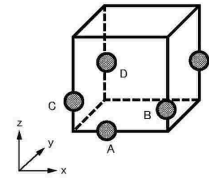


図16-2



図16-3

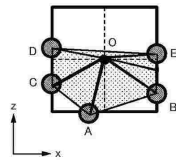
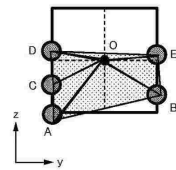


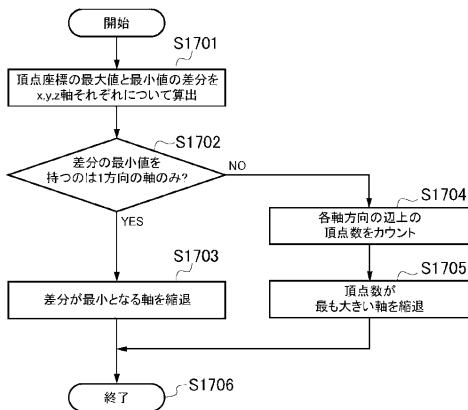
図16-4



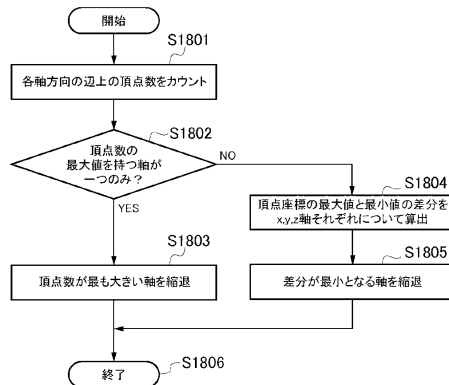
10

20

【 図 1 7 】



【 図 1 8 】



30

40

50

【図 19】

図19-1

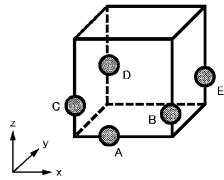


図19-2

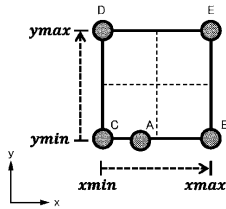
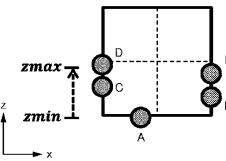


図19-3



【図 20】

図20-1

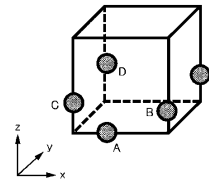


図20-2

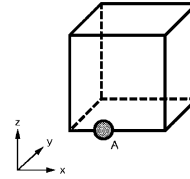


図20-3

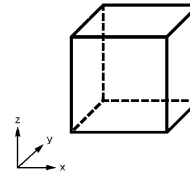
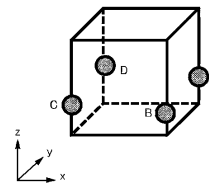


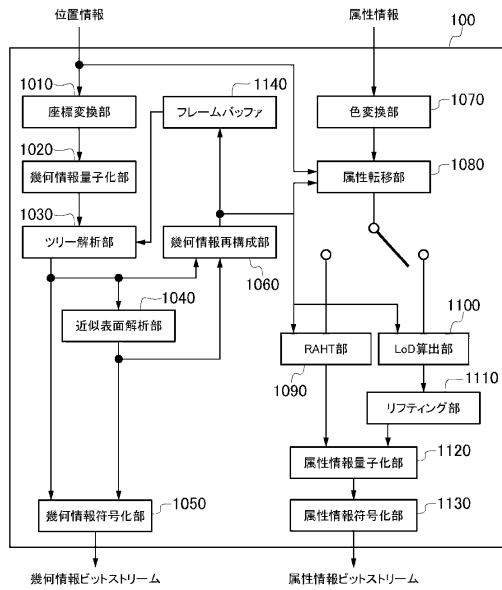
図20-4



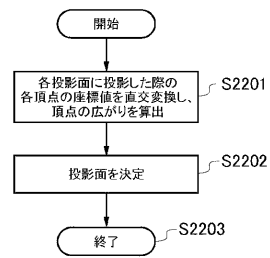
10

20

【図 21】



【図 22】



30

40

50

## フロントページの続き

- (56)参考文献 国際公開第2021/261142 (WO, A1)  
国際公開第2021/256486 (WO, A1)  
Antoine Dricot, and Joao Ascenso, Adaptive Multi-level Triangle Soup for Geometry-based Point Cloud Coding, 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), IEEE, 2019年09月, pp.1-6  
Chaofei Wang, et al., Point-Voting based Point Cloud Geometry Compression, 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP), IEEE, 2021年10月, pp.1-5  
Khaled Mammou, Philip A. Chou, David Flynn, Maja Krivokuca, Ohji Nakagami and Toshiyasu Sugio, G-PCC codec description v2, CODING OF MOVING PICTURES AND AUDIO, INTERNATIONAL ORGANISATION FOR STANDARDISATION, 2019年01月, pp.2-3,17-19
- (58)調査した分野 (Int.Cl., DB名)  
G06T 9/00 - 9/40  
H04N 19/00 - 19/98