

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)公開番号
特開2024-124401
(P2024-124401A)

(43)公開日 令和6年9月12日(2024.9.12)

(51)国際特許分類

G 10 L 19/00 (2013.01)

H 04 S 7/00 (2006.01)

G 10 L 19/008 (2013.01)

F I

G 10 L

H 04 S

G 10 L

19/00

7/00

19/008

3 3 0 B

3 2 0

1 0 0

テーマコード(参考)

5 D 1 6 2

審査請求 有 請求項の数 17 O L 外国語出願 (全89頁)

(21)出願番号	特願2024-67881(P2024-67881)	(71)出願人	590000248 コーニンクレッカ フィリップス エヌ ヴェ Koninklijke Philips N.V. オランダ国 5656 アーヘー アイン ドーフェン ハイテック キャンパス 52 High Tech Campus 52, 5656 AG Eindhoven, N etherlands 110001690 弁理士法人M&Sパートナーズ
(22)出願日	令和6年4月19日(2024.4.19)		
(62)分割の表示	特願2024-523595(P2024-523595)の分割 原出願日 令和4年10月20日(2022.10.20)		
(31)優先権主張番号	21204639.5		
(32)優先日	令和3年10月26日(2021.10.26)		
(33)優先権主張国・地域又は機関	歐州特許庁(EP)		
		(74)代理人	
		(72)発明者	

(54)【発明の名称】 環境内の音声を表すビットストリーム

(57)【要約】 (修正有)

【課題】音声環境を表すビットストリームを生成する装置及びビットストリームを処理する装置を提供する。

【解決手段】視聴覚分配システムにおいて、符号化装置は、環境内の音源を表す複数の音声要素の音声データのメタデータを生成するメタデータ生成器203を含む。メタデータは、環境の音響環境データを含む。音響環境データは、環境内の音源の音の伝播影響を与える特性を記述し、その少なくとも一部は、環境内の複数のリスニングポーズに適用可能である。前記特性は、静的特性及び動的特性の両方を含む。符号化装置はまた、メタデータを含むビットストリームを生成するビットストリーム生成器205を含む。ビットストリームは多くの場合、環境内の音源の音声要素を表す音声データをさらに含む。復号装置は、ビットストリームを受信する受信機と、音響環境データ及び音声要素の音声データに基づいて音声環境の音声をレンダリングするレンダラーと、を含む。

【選択図】図2

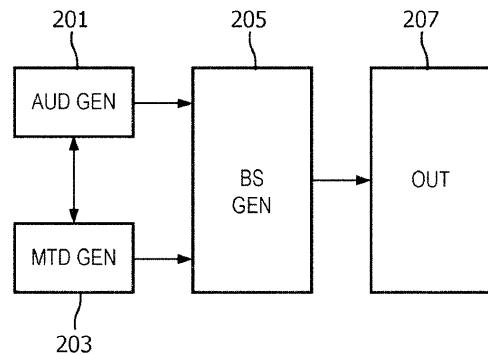


図2

【特許請求の範囲】**【請求項 1】**

ビットストリームを生成するための装置であって、前記装置は、

環境内の音源を表す複数の音声要素の音声データのメタデータを生成するメタデータ生成器であって、前記メタデータは前記環境の音響環境データを含み、前記音響環境データは、前記環境内の前記音源の音の伝播に影響を及ぼす特性を記述し、前記音響環境データの少なくとも一部は、前記環境内の複数のリスニングポーズに適用可能であり、前記特性は静的特性および動的特性の両方を含む、メタデータ生成器と、

前記メタデータを含む前記ビットストリームを生成するビットストリーム生成器とを備え、

前記音響環境データは、向き特性を表すための向き表現フォーマットを記述するデータグループと、前記向き表現フォーマットを使用して、前記特性のうちの向き特性を記述するデータを含む少なくとも1つのデータグループとを含み、前記データグループは、

所定のデフォルト向き表現を示すデータと、

所定の角度のセットを示すデータと、

量子化グリッド上で角度を示すデータとのうちの少なくとも1つを含む、装置。

【請求項 2】

前記音響環境データは、周波数グリッドを記述するデータグループと、前記周波数グリッドを使用して、前記特性のうちの周波数依存特性を記述するデータをそれぞれが含む複数のデータグループとを含み、前記ビットストリームは、前記周波数グリッドを記述する前記データグループを前記ビットストリームが含むか否かを示すインジケータを含み、前記データグループは、前記周波数グリッドを記述するデータのフォーマットのインジケーションを含み、前記データグループは、

所定のデフォルトグリッドを示すデータと、

前記周波数グリッドの少なくともいくつかの部分範囲の開始周波数および周波数範囲を示すデータと、

個別の周波数を示すデータとのうちの少なくとも1つを含む、請求項1に記載の装置。

【請求項 3】

前記データグループは、

所定の角度のセットを示すデータと、

量子化グリッド上で角度を示すデータとのうちの少なくとも1つを含む、請求項1または2に記載の装置。

【請求項 4】

前記メタデータ生成器は、前記向き表現フォーマットを、前記所定のデフォルト向き表現、前記所定の角度のセット、および前記量子化グリッド上の角度のうちの1つとして選択する、請求項1から3のいずれか一項に記載の装置。

【請求項 5】

前記所定の角度のセットの各所定の角度がインデックスによって表され、前記向き特性が、所定の角度のインデックスのインジケーションによって表される、装置。

【請求項 6】

前記データグループは、所定の角度のセットを示すデータを含む、請求項1から5のいずれか一項に記載の装置。

【請求項 7】

前記データグループは、量子化グリッド上で角度を示すデータを含む、請求項1から6のいずれか一項に記載の装置。

【請求項 8】

前記音響環境データは、向き特性を表すためのさらなる向き表現フォーマットを記述するさらなるデータグループと、前記さらなる向き表現フォーマットを使用して、前記特性のうちの向き特性を記述するデータを含む少なくとも1つのさらなるデータグループとを含み、前記さらなるデータグループは、所定の角度のセットを示すデータを含む、請求項

7 に記載の装置。

【請求項 9】

前記音響環境データは、音の伝播に影響を及ぼす前記特性のうちの第1の特性の値を表す第1のビットのための第1のデータフィールドと、前記第1の特性の前記値を表す第2のビットのための拡張データを前記音響環境データが含むか否かを示す第2のデータフィールドとを含む、請求項1から8のいずれか一項に記載の装置。

【請求項 10】

前記メタデータ生成器は、前記環境が空間的に制約された環境であることを示すグローバルインジケータを含むように前記音響環境データを生成し、前記メタデータ生成器は、前記環境が空間的に制約されていることを示す前記グローバルインジケータについてのデータ値のための所定の制限付きフォーマットに従うように、前記音響環境データのデータ値を制限する、請求項1から9のいずれか一項に記載の装置。

【請求項 11】

前記音響環境データは複数の連続したデータセット内に配置され、各データセットは、時間間にわたるデータを含み、前記連続したデータセットのうちの第1のデータセットは、音に影響を及ぼす前記特性のうちの少なくとも1つの特性の第1の特性値と、前記第1のデータセットによって表される時間間隔内の時間を示す、前記第1の特性値の時間インジケーションとを含む、請求項1から10のいずれか一項に記載の装置。

【請求項 12】

前記音響環境データは複数の連続したデータセット内に配置され、各データセットは、時間間にわたるデータを含み、前記ビットストリーム生成器は、音の伝播に影響を及ぼす前記特性のうちの第1の特性の特性値が、第1のデータセットによって表される時間間隔内のデフォルト時間のために提供されているかを判定し、そうである場合は、時間インジケーションなしで前記第1のデータセット内に第1の特性値を含め、そうでない場合は、前記第1の特性値のための時間インジケーションとともに前記第1のデータセット内に前記第1の特性値を含める、請求項1から11のいずれか一項に記載の装置。

【請求項 13】

第1の音声要素の前記音響環境データは、音の伝播に影響を及ぼす前記特性のうちの第1の特性の第1の特性値についての第1の適用可能性領域および第2の適用可能性領域のインジケーションを含み、前記第1の適用可能性領域は、前記第1の特性値が適用される前記第1の音声要素の位置の領域を示し、前記第2の適用可能性領域は、前記第1の特性値が適用されるリスニング位置の領域を示す、請求項1から12のいずれか一項に記載の装置。

【請求項 14】

レンダリング音声を生成するための装置であって、前記装置は、
環境内の音源を表す複数の音声要素の音声データを受信する第1の受信機と、
前記環境内の音源を表す前記複数の音声要素の前記音声データのメタデータを含むビットストリームを受信する第2の受信機であって、前記メタデータは前記環境の音響環境データを含み、前記音響環境データは、前記環境内の前記音源の音の伝播に影響を及ぼす特性を記述し、前記音響環境データの少なくとも一部は、前記環境内の複数のリスニングポーズに適用可能であり、前記特性は静的特性および動的特性の両方を含む、第2の受信機と、

前記音声データおよび前記音響環境データに応答して前記環境の出力音声データを生成するレンダラーとを備え、

前記音響環境データは、向き特性を表すための向き表現フォーマットを記述するデータグループと、前記向き表現フォーマットを使用して、前記特性のうちの向き特性を記述するデータを含む少なくとも1つのデータグループとを含み、前記データグループは、

所定のデフォルト向き表現を示すデータと、

所定の角度のセットを示すデータと、

量子化グリッド上で角度を示すデータとのうちの少なくとも1つを含む、装置。

10

20

30

40

50

【請求項 1 5】

環境内の音源を表す複数の音声要素の音声データのメタデータを含むビットストリームであって、前記メタデータは前記環境の音響環境データを含み、前記音響環境データは、前記環境内の前記音源の音の伝播に影響を及ぼす特性を記述し、前記音響環境データの少なくとも一部は、前記環境内の複数のリスニングポーズに適用可能であり、前記特性は静的特性および動的特性の両方を含み、

前記音響環境データは、向き特性を表すための向き表現フォーマットを記述するデータグループと、前記向き表現フォーマットを使用して、前記特性のうちの向き特性を記述するデータを含む少なくとも1つのデータグループとを含み、前記データグループは、

所定のデフォルト向き表現を示すデータと、

所定の角度のセットを示すデータと、

量子化グリッド上で角度を示すデータとのうちの少なくとも1つを含む、ビットストリーム。

【請求項 1 6】

ビットストリームを生成する方法であって、前記方法は、

環境内の音源を表す複数の音声要素の音声データのメタデータを生成するステップであって、前記メタデータは前記環境の音響環境データを含み、前記音響環境データは、前記環境内の前記音源の音の伝播に影響を及ぼす特性を記述し、前記音響環境データの少なくとも一部は、前記環境内の複数のリスニングポーズに適用可能であり、前記特性は静的特性および動的特性の両方を含む、生成するステップと、

前記メタデータを含む前記ビットストリームを生成するステップとを含み、

前記音響環境データは、向き特性を表すための向き表現フォーマットを記述するデータグループと、前記向き表現フォーマットを使用して、前記特性のうちの向き特性を記述するデータを含む少なくとも1つのデータグループとを含み、前記データグループは、

所定のデフォルト向き表現を示すデータと、

所定の角度のセットを示すデータと、

量子化グリッド上で角度を示すデータとのうちの少なくとも1つを含む、方法。

【請求項 1 7】

レンダリング音声を生成する方法であって、前記方法は、

環境内の音源を表す複数の音声要素の音声データを受信するステップと、

前記環境内の音源を表す前記複数の音声要素の前記音声データのメタデータを含むビットストリームを受信するステップであって、前記メタデータは前記環境の音響環境データを含み、前記音響環境データは、前記環境内の前記音源の音の伝播に影響を及ぼす特性を記述し、前記音響環境データの少なくとも一部は、前記環境内の複数のリスニングポーズに適用可能であり、前記特性は静的特性および動的特性の両方を含む、受信するステップと、

前記音声データおよび前記音響環境データに応答して前記環境の出力音声データを生成するステップとを含み、

前記音響環境データは、向き特性を表すための向き表現フォーマットを記述するデータグループと、前記向き表現フォーマットを使用して、前記特性のうちの向き特性を記述するデータを含む少なくとも1つのデータグループとを含み、前記データグループは、

所定のデフォルト向き表現を示すデータと、

所定の角度のセットを示すデータと、

量子化グリッド上で角度を示すデータとのうちの少なくとも1つを含む、方法。

【発明の詳細な説明】**【技術分野】****【0 0 0 1】**

本発明は、音声環境を表すビットストリーム、ならびにそのようなビットストリームを生成するための装置およびそのようなビットストリームを処理するための装置に関し、特に、限定はされないが、例えば仮想現実アプリケーション用の仮想音声環境を表すビット

10

20

30

40

50

ストリームに関する。

【背景技術】

【0002】

近年、視聴覚コンテンツに基づく体験の多様性および範囲が大きく増大しており、そのようなコンテンツを利用および消費する新しいサービスや方法が継続的に開発および導入されている。特に、関与性や没入感がより高い体験をユーザに提供するために、多くの空間的インタラクティブサービス、アプリケーション、および体験が開発されている。

【0003】

このようなアプリケーションの例としては、仮想現実（VR）アプリケーション、拡張現実（AR）アプリケーション、および急速に主流になりつつある複合現実（MR）アプリケーションがあり、多くのソリューションが消費者市場を標的としている。また、様々な標準化団体によって様々な規格が開発されている。そのような標準化活動は、例えばストリーミング、ブロードキャスト、およびレンダリングなど、VR/AR/MRシステムの様々な側面のための規格を積極的に開発している。

【0004】

VRアプリケーションは、異なる世界／環境／シーンにいるユーザに対応するユーザ体験を提供する傾向がある一方、AR（複合現実MRを含む）アプリケーションは、追加情報または仮想オブジェクトもしくは情報が追加された現在の環境にいるユーザに対応するユーザ体験を提供する傾向がある。したがって、VRアプリケーションは完全に没入型の人工世界／シーンを提供する傾向がある一方、ARアプリケーションは、ユーザが物理的に存在する現実のシーンに重ねられた部分的に人工の世界／シーンを提供する傾向がある。しかし、これらの用語はしばしば同義で使用され、大きく重複している。以下では、仮想現実／VRという用語は仮想現実および拡張／複合現実の両方を表すために使用される。

【0005】

例えばVR体験を提供するためにユーザ側の適応を可能にする柔軟な表現を提供できるように、環境、特に音声環境を表す視聴覚データ、特に音声データを通信することは、非常に困難なタスクである。通信されるデータは、（仮想）リスニング位置の変化および環境自体の変化を反映する動的な体験をレンダリングするためにローカル使用できるように、環境を表すことがほしい。

【0006】

このような環境を表すデータを効率的に通信するための有利なアプローチを得るために、多くの研究が行われてきた。適切なデータストリームおよびフォーマットに関する様々な提案がなされており、そのほとんどが、個々の音源が別々に提示され、音源の位置などの様々な特性を記述するメタデータに関連づけられている、個別化されたモデルを含む。また、残響や減衰などを記述するデータなど、音声環境を記述する一般的データが提供される可能性がある。

【0007】

しかし、そのような情報の効率的な（例えば、低減されたデータレート）通信を提供するビットストリームフォーマットを定義することは非常に困難であり、有利なアプローチを実現するには、多くの問題、特性、およびトレードオフを慎重に検討し、これらのバランスを取る必要がある。MPEG (Moving Picture Experts Group) は、VRおよび同様の体験に適したビットストリーム用のMPEG-Iと呼ばれる規格を開発するための標準化アプローチを開始した。

【0008】

したがって、VRおよびARなどの没入型アプリケーションおよびサービスにおいて音声をサポートするための改善されたアプローチおよびデータフォーマット／ビットストリームは有利であろう。特に、動作の向上、柔軟性の増加、複雑さの軽減、実装の容易化、音声体験の向上、複雑さの軽減、計算負荷の軽減、音声品質の向上、データレートの低下、トレードオフの向上、ならびに／またはパフォーマンスおよび／もしくは動作の向上を

10

20

30

40

50

可能にするアプローチ / ビットストリーム / フォーマットは有利であろう。

【発明の概要】

【0009】

したがって、本発明は、上記欠点の1つ以上を単独で、または任意の組み合わせで好適に緩和、低減、または排除することを目的とする。

【0010】

本発明の態様および任意選択の特徴によれば、ビットストリームを生成するための装置が提供され、装置は、環境内の音源を表す複数の音声要素の音声データのメタデータを生成するメタデータ生成器(203)であって、メタデータは環境の音響環境データを含み、音響環境データは、環境内の音源の音の伝播に影響を及ぼす特性を記述し、音響環境データの少なくとも一部は、環境内の複数のリスニングポーズに適用可能であり、特性は静的特性および動的特性の両方を含む、メタデータ生成器と、メタデータを含むビットストリームを生成するビットストリーム生成器とを含む。

【0011】

このアプローチは、没入型で、柔軟で、かつ変化に富んだ視聴覚アプリケーションを含む多くのアプリケーション(例えば、多くのVRおよびARアプリケーション)のパフォーマンスおよび動作を改善することができる。このアプローチは、多くの場合において異なる要望間のトレードオフ、例えば、環境の正確、完全、かつ/または動的なデータを提供するという要望と、低いデータレートのビットストリームを提供するという要望との間のトレードオフを改善することができる。このアプローチは、多くの場合において高度な柔軟性を提供し、レンダリング側の適合およびカスタマイズを容易にし、改善し、さらには可能にする可能性がある。このアプローチは、環境の音声のレンダリングを容易化および/または改善する可能性があり、具体的には、変化する環境および/または変化するリスニング位置の動的レンダリングを容易化および/または改善する可能性がある。

【0012】

このアプローチは、具体的には、動的アプリケーションに特に適した、注意深く適合された視聴覚ビットストリームを提供する可能性があり、ビットストリームは、注意深く選択され、かつ少なくとも部分的に最適化された、音声環境のデータ表現を提供する。このデータ表現は、音源および環境の音響特性の両方を含み、場合によってはさらに環境内の個別のオブジェクトを含む。

【0013】

一部の実施形態では、装置はさらに、環境内の音源を表す複数の音声要素の音声データを生成するように構成された音声データ生成器を含み得、ビットストリーム生成器は、ビットストリーム内に音声データを含めるように構成され得る。

【0014】

静的特性は、(少なくとも特性値が対象とする時間間隔では)時不变の特性であり得る。動的特性は、(少なくとも特性値が対象とする時間間隔では)時変の特性であり得る。

【0015】

多くの実施形態では、少なくとも1つの特性は位置および向きに依存する。多くの実施形態では、少なくとも1つの特性の音響環境データは、位置および/または向き依存性を示す。位置および/または向き依存性は、音源の向きおよび/もしくは位置、ならびに/またはリスナーポーズの向きおよび/または位置への依存性であり得る。

【0016】

一部の実施形態では、ビットストリーム内の音響環境データは、複数の一連のデータグループに分割され、少なくとも一部のグループは、それぞれ、音の伝播に影響を及ぼす特性の特性値を提供する。

【0017】

本発明の任意選択の特徴によれば、音響環境データは、音の伝播に影響を及ぼす特性のうちの少なくとも1つの特性の特性値の表現の少なくとも一部のデータフォーマットを記述するデータグループと、表現を使用して少なくとも1つの特性値を記述するデータをそ

10

20

30

40

50

れぞれが含む複数のデータグループとを含む。

【0018】

一部の実施形態では、音響環境データは、音の伝播に影響を及ぼす特性のうちの特性のデータフォーマットを記述するデータグループと、データフォーマットに従ってその特性の特性値を記述するデータをそれぞれが含む複数のデータグループとを含む。

【0019】

データグループは1つ以上のデータ値であり得、具体的にはビットのセットであり得る。データグループはデータのセットであり得、具体的にはビットストリームの1つ以上のデータフィールドであり得る。

【0020】

一部の実施形態では、音響環境データは、周波数グリッドを記述するデータグループと、周波数グリッドを使用して特性のうちの周波数依存特性を記述するデータをそれぞれが含む複数のデータグループとを含む。

【0021】

周波数グリッドは、例えば周波数部分範囲の中心周波数を定義することによって、周波数範囲を周波数部分範囲に細分化したものであってもよい。

【0022】

一部の実施形態では、ビットストリームは、ビットストリームが、周波数グリッドを記述するデータグループを含むか否かを示すインジケータを含む。

【0023】

一部の実施形態では、データグループは、周波数グリッドを記述するデータのフォーマットのインジケーションを含む。

【0024】

一部の実施形態では、音響環境データは、周波数グリッドを記述するデータグループと、周波数グリッドを使用して、特性のうちの周波数依存特性を記述するデータをそれぞれが含む複数のデータグループとを含み、ビットストリームは、周波数グリッドを記述するデータグループをビットストリームが含むか否かを示すインジケータを含み、データグループは、周波数グリッドを記述するデータのフォーマットのインジケーションを含む。

【0025】

一部の実施形態では、データグループは、所定のデフォルトグリッドを示すデータと、周波数グリッドの少なくともいくつかの部分範囲の開始周波数および周波数範囲を示すデータと、個別の周波数を示すデータとの少なくとも1つを含む。

【0026】

データグループのデータによって示される少なくとも1つの部分範囲または個別の周波数は、オクターブバンドの分数と位置合わせされる。

【0027】

本発明の任意選択の特徴によれば、音響環境データは、周波数グリッドを記述するデータグループと、周波数グリッドを使用して、特性のうちの周波数依存特性を記述するデータをそれぞれが含む複数のデータグループとを含み、ビットストリームは、周波数グリッドを記述するデータグループをビットストリームが含むか否かを示すインジケータを含み、データグループは、周波数グリッドを記述するデータのフォーマットのインジケーションを含み、データグループは、所定のデフォルトグリッドを示すデータと、周波数グリッドの少なくともいくつかの部分範囲の開始周波数および周波数範囲を示すデータと、個別の周波数を示すデータとの少なくとも1つを含む。

【0028】

一部の実施形態では、音響環境データは、向き特性を表すための向き表現フォーマットを記述するデータグループと、向き表現フォーマットを使用して、特性のうちの向き特性を記述するデータを含む少なくとも1つのデータグループとを含む。

【0029】

ビットストリームは、向き表現フォーマットを記述するデータグループをビットストリーム

10

20

30

40

50

ームが含むか否かを示すインジケータを含む可能性がある。

【0030】

一部の実施形態では、データグループは、所定のデフォルト向き表現を示すデータと、所定の角度のセットを示すデータと、量子化グリッド上で角度を示すデータと、のうちの少なくとも1つを含む。

【0031】

本発明の任意選択の特徴によれば、音響環境データは、向き特性を表すための向き表現フォーマットを記述するデータグループと、向き表現フォーマットを使用して、特性のうちの向き特性を記述するデータを含む少なくとも1つのデータグループとを含み、データグループは、所定のデフォルト向き表現を示すデータと、所定の角度のセットを示すデータと、量子化グリッド上で角度を示すデータとのうちの少なくとも1つを含む。

10

【0032】

本発明の任意選択の特徴によれば、音響環境データは、音の伝播に影響を及ぼす特性のうちの第1の特性の値を表す第1のビットのための第1のデータフィールドと、第1の特性の値を表す第2のビットのための拡張データを音響環境データが含むか否かを示す第2のデータフィールドとを含む。

【0033】

本発明の任意選択の特徴によれば、第2のビットは、第1の特性が取り得る値の範囲を拡張する。

【0034】

第2のビットは、第1の特性の値を表す、データワードの第1のビットよりも上位のビットであってもよい。

20

【0035】

本発明の任意選択の特徴によれば、第2のビットは、第1の特性が取り得る値の分解能を高める。

【0036】

第2のビットは、第1の特性の値を表す、データワードの第1のビットよりも下位のビットであってもよい。第2のビットは、第1の特性が取り得る値の精度を拡張し得る。

30

【0037】

一部の実施形態では、第1の特性は、時間的特性、空間的特性、量、ゲイン特性、容量特性、周波数特性、インデックス特性、および識別特性を含む群から選択される特性である。

【0038】

本発明の任意選択の特徴によれば、メタデータ生成器(203)は、環境が空間的に制約された環境であることを示すグローバルインジケータを含むように音響環境データを生成し、メタデータ生成器は、環境が空間的に制約されていることを示すグローバルインジケータについてのデータ値のための所定の制限付きフォーマットに従うように、音響環境データの(少なくとも1つの)データ値を制限する。

【0039】

空間的に制限された環境は、閾値未満の空間的広がりを有する環境であり得る。所定の制限付きフォーマットは、環境が空間的に制限された環境でない場合にビットストリームに使用されるデータフォーマットよりも少ないビットを使用する、音の伝播に影響する特性の少なくともいくつかの値のデータフォーマットであり得る。

40

【0040】

グローバルインジケータは任意選択のインジケータであり得る。一部の実施形態では、グローバルインジケータは、環境が空間的に制約された環境であるか、または空間的に制約がより少ない環境(または空間的な制約がない環境)であるかを示し得る。

【0041】

一部の実施形態では、音響環境データは、少なくとも第1の音声要素のアニメーションインジケーションを含み、アニメーションインジケーションは、第1の音声要素の少なく

50

とも 1 つの特性が時間間隔中に変化するか否かを示す。

【 0 0 4 2 】

本発明の任意選択の特徴によれば、音響環境データは、少なくとも第 1 の音声要素のアニメーションインジケーションを含み、アニメーションインジケーションは、第 1 の音声要素の少なくとも 1 つの特性が時間間隔中に変化するかを示し、第 1 の音声要素が少なくとも 1 つの変化する特性を有することを示すアニメーションインジケーションのために、音響環境データは、少なくとも 1 つの変化する特性の変動を記述するデータを含む。

【 0 0 4 3 】

一部の実施形態では、アニメーションインジケーションが、第 1 の音声要素の少なくとも 1 つの特性が時間間隔中に変化することを示す場合、音響環境データはさらに、少なくとも 2 つの特性のそれぞれのためのさらなるアニメーションインジケーションを含み、さらなるアニメーションインジケーションは、対応する特性がアニメーション化されているか否かを示し、アニメーションインジケーションが、時間間隔中に変化する第 1 の音声要素の特性が存在しないことを示す場合、音響環境データは、上記少なくとも 2 つの特性のさらなるアニメーションインジケーションを含まない。

【 0 0 4 4 】

一部の実施形態では、アニメーションインジケーションが、第 1 の音声要素の少なくとも 1 つの特性が時間間隔中に変化することを示す場合、少なくとも 2 つの特性のためのさらなるアニメーションインジケーションが存在し、さらなるアニメーションインジケーションは、対応する特性がアニメーション化されているか否かを示し、アニメーションインジケーションが、時間間隔中に変化する、第 1 の音声要素の少なくとも 1 つの特性が存在しないことを示す場合、アニメーションインジケーションは除外される。

【 0 0 4 5 】

一部の実施形態では、音響環境データは、時間間隔中に変化する少なくとも 1 つの特性の少なくとも 2 つの値と、これらの少なくとも 2 つの値の間を補間するための時間的補間を記述する補間データとを含む。

【 0 0 4 6 】

本発明の任意選択の特徴によれば、音声要素は複数の音響効果要素を含み、音響環境データは、ユーザによって制御される環境への変化を、複数の音響効果要素のうちの第 1 の音響効果要素とリンクさせるデータを含む。

【 0 0 4 7 】

一部の実施形態では、音響環境データは連続したデータセットとして配置され、各データセットは時間間隔にわたるデータを含む。

【 0 0 4 8 】

時間間隔はデータセットごとに異なる。

【 0 0 4 9 】

本発明の任意選択の特徴によれば、音響環境データは複数の連続したデータセットとして配置され、各データセットは、時間間隔にわたるデータを含み、連続したデータセットのうちの第 1 のデータセットは、音に影響を及ぼす特性のうちの少なくとも 1 つの特性の第 1 の特性値と、第 1 のデータセットによって表される時間間隔内の時間を示す、第 1 の特性値の時間インジケーションとを含む。

【 0 0 5 0 】

一部の実施形態では、連続したデータセットのうちの第 1 のデータセットは、音に影響を及ぼす特性のうちの少なくとも 1 つの特性の少なくとも 2 つの特性値と、これらの少なくとも 2 つの特性値の時間インジケーションとを含み、時間インジケーションは、第 1 のデータセットによって表される時間間隔内の時間を示す。

【 0 0 5 1 】

本発明の任意選択の特徴によれば、音響環境データは複数の連続したデータセットとして配置され、各データセットは、時間間隔にわたるデータを含み、ビットストリーム生成器は、音の伝播に影響を及ぼす特性のうちの第 1 の特性の特性値が、第 1 のデータセット

10

20

30

40

50

によって表される時間間隔内のデフォルト時間のために提供されているかを判定し、そうである場合は、時間インジケーションなしで第1のデータセット内に第1の特性値を含め、そうでない場合は、第1の特性値のための時間インジケーションとともに第1のデータセット内に第1の特性値を含める。

【0052】

一部の実施形態では、音響環境データは、環境の音声をレンダリングするために必要なデータを全て含むいくつかの完全なレンダリングデータセットと、環境の音声をレンダリングするために他のデータセットからのデータを追加で必要とするいくつかの部分的レンダリングデータセットとを含む。

【0053】

一部の実施形態では、環境の少なくとも一部の要素の音響環境データは、環境のシングラフの要素の識別データおよび親識別データを含む。

【0054】

これらの少なくとも一部の要素は、例えば、オブジェクト、音源（音声要素）、および/または環境の音響特性であり得る。

【0055】

本発明の任意選択の特徴によれば、第1の音声要素の音響環境データは、音の伝播に影響を及ぼす特性のうちの第1の特性の第1の特性値についての第1の適用可能性領域および第2の適用可能性領域のインジケーションを含み、第1の適用可能性領域は、第1の特性値が適用される第1の音声要素の位置の領域を示し、第2の適用可能性領域は、第1の特性値が適用されるリスニング位置の領域を示す。

【0056】

本発明の態様および任意選択の特徴によれば、レンダリング音声を生成するための装置が提供され、装置は、環境内の音源を表す複数の音声要素の音声データを受信する第1の受信機（303）と、環境内の音源を表す複数の音声要素の音声データのメタデータを含むビットストリームを受信する第2の受信機（305）であって、メタデータは環境の音響環境データを含み、音響環境データは、環境内の音源の音の伝播に影響を及ぼす特性を記述し、音響環境データの少なくとも一部は、環境内の複数のリスニングポーズに適用可能であり、特性は静的特性および動的特性の両方を含む、第2の受信機と、音声データおよび音響環境データに応答して環境の出力音声データを生成するレンダラー（307）とを含む。

【0057】

本発明の態様および任意選択の特徴によれば、環境内の音源を表す複数の音声要素の音声データのメタデータを含むビットストリームが提供され、メタデータは環境の音響環境データを含み、音響環境データは、環境内の音源の音の伝播に影響を及ぼす特性を記述し、音響環境データの少なくとも一部は、環境内の複数のリスニングポーズに適用可能であり、特性は静的特性および動的特性の両方を含む。

【0058】

本発明の態様および任意選択の特徴によれば、ビットストリームを生成する方法が提供され、方法は、環境内の音源を表す複数の音声要素の音声データのメタデータを生成するステップであって、メタデータは環境の音響環境データを含み、音響環境データは、環境内の音源の音の伝播に影響を及ぼす特性を記述し、音響環境データの少なくとも一部は、環境内の複数のリスニングポーズに適用可能であり、特性は静的特性および動的特性の両方を含む、生成するステップと、メタデータを含むビットストリームを生成するステップと、を含む。

【0059】

本発明の態様および任意選択の特徴によれば、レンダリング音声を生成する方法が提供され、方法は、環境内の音源を表す複数の音声要素の音声データを受信するステップと、環境内の音源を表す複数の音声要素の音声データのメタデータを含むビットストリームを受信するステップであって、メタデータは環境の音響環境データを含み、音響環境データ

10

20

30

40

50

は、環境内の音源の音の伝播に影響を及ぼす特性を記述し、音響環境データの少なくとも一部は、環境内の複数のリスニングポーズに適用可能であり、特性は静的特性および動的特性の両方を含む、受信するステップと、音声データおよび音響環境データに応答して環境の出力音声データを生成するステップと、を含む。

【0060】

本発明の上記および他の側面、特徴、および利点は、以下に記載される実施形態を参照しながら説明され、明らかになるであろう。

【図面の簡単な説明】

【0061】

以下、本発明の単なる例に過ぎない実施形態について、以下の図面を参照しながら説明する。

【図1】図1は、視聴覚分配システムの例を示す。

【図2】図2は、本発明の一部の実施形態に係る符号化装置の要素の例を示す。

【図3】図3は、本発明の一部の実施形態に係る復号装置の要素の例を示す。

【図4】図4は、本発明の一部の実施形態に係るビットストリームのデータ構造の例を示す。

【図5】図5は、本発明の一部の実施形態に係る、ビットストリーム内で表される変化する特性値の例を示す。

【図6】図6は、本発明の一部の実施形態に係るビットストリームのデータ構造を示す。

【発明を実施するための形態】

【0062】

以下の説明は、仮想現実（VR）アプリケーションなどの視聴覚アプリケーションに焦点を当てるが、説明される原理および概念は他の多くのアプリケーションおよび実施形態でも使用できることが理解されるであろう。

【0063】

以下の説明は、環境の音声の表現を提供するビットストリームの生成に焦点を当てる。多くの例では、音声表現は視覚環境の表現によって補完され、VRアプリケーションは、ユーザに提示する音声および映像の両方を生成するように構成されている。音声表現が、仮想シーン／環境の視覚表現を補足するか、またはビットストリームが、例えば、現実世界もしくはハイブリッド環境の表現を提供し得る。ビットストリームは、例えば音源やオブジェクトなどの環境／シーン内の個々の要素を表すデータを含む可能性がある。さらにより一般的な音響または視覚データ（例えば、残響や背景色などを記述するデータ）など、より一般的な情報が提供されてもよい。

【0064】

しかし、説明される原理および概念が他の多くのアプリケーションおよび実施形態でも使用できることが理解されるであろう。

【0065】

仮想現実体験は、コンピュータなどのデバイスが、仮想シーン用の3次元音声および映像をレンダリングし、それをユーザに提示することにより、ユーザに対して仮想体験を生成することを可能にし得る。ユーザは通常、動き回る可能性があるため、観察者／リスニングポーズが動的に変化する可能性がある。多くの実施形態では、仮想シーン／環境は、例えばオブジェクトが移動するか、またはその形状を変化させ、複数の音源が異なる時点で音声を発する動的なシーンであってもよい。

【0066】

当該技術分野では、配置およびポーズという用語が、位置および／または方向／向きを表す一般的な用語として使用される。例えばオブジェクト、カメラ、頭部、またはビューの位置および方向／向きの組み合わせがポーズまたは配置と呼ばれる。したがって、配置またはポーズのインジケーションは6つの値／成分／自由度を備え得る。各値／成分は典型的には、対応するオブジェクトの位置／場所または方向／向きの個々の特性を記述する。当然ながら、多くの状況において、配置またはポーズはより少ない成分で考慮または

10

20

30

40

50

表現され得、例えば、1つ以上の成分が一定または無関係であると見なされる場合が該当する（例えば、全てのオブジェクトが同じ高さを有し、かつ向きが水平であると見なされる場合、4つの成分でオブジェクトのポーズを完全に表現することが可能であり得る）。以下、ポーズという用語は1～6個（可能な最大自由度に対応）の値で表すことができる位置および／または向きを指すために使用される。ポーズという用語は、配置という用語に置き換えることができる。ポーズという用語は、位置および／または向きという用語に置き換えることができる。ポーズという用語は、位置および向きという用語（ポーズが位置および向きの両方の情報を提供する場合）、位置という用語（ポーズが位置の情報（場合によってはそれのみ）を提供する場合）、または向き（ポーズが向きの情報（場合によってはそれのみ）を提供する場合）に置き換えることができる。

10

【0067】

多くのアプローチにおいて、VRアプリケーションは例えば、リモートVRサーバを一切使用しないか、または、場合によってはリモートVRサーバへのアクセスすら有さないスタンドアロンデバイスによって視聴者にローカルに提供され得る。しかし、他のアプリケーションでは、VRアプリケーションはリモートサーバまたは中央サーバから受信したデータに基づき得る。例えば、リモート中央サーバから音声または視覚データがVRデバイスに提供され、ローカル処理されて、所望のVR体験が生成されてもよい。

【0068】

図1は、リモートVRクライアントデバイス101が、例えばインターネットなどのネットワーク105を介してVRサーバ103と連絡するVRシステムの例を示す。サーバ103は、多数存在する可能性があるクライアントデバイス101を同時にサポートするように構成され得る。

20

【0069】

VRサーバ103は、例えば、仮想環境の要素、および仮想オブジェクトを定義するデータをクライアントデバイス101に送信することによって、仮想現実体験をサポートすることができる。データは、ユーザに提示可能なグラフィックスを生成するためにクライアントデバイス101によって使用され得る多数の仮想オブジェクトの視覚的特徴および幾何学的特性を具体的に記述し得る。一部の実施形態では、データはまた、ユーザに提示可能な様々な情報を含み得る。さらに、サーバ103は、ユーザ体験、特に没入感をさらに高める可能性がわる仮想サウンド／音声をローカルに生成するために使用できる音声データをクライアントデバイス103に提供することができる。

30

【0070】

以下の説明は、音響シーンおよび環境の表現（典型的には、音源の表現および環境の音響特性の表現の両方を含む）を提供する音声ビットストリームの生成に焦点を当てる。

【0071】

図2は、ビットストリームを生成する装置の一例を示しており、具体的には、装置は図1のサーバ103であってもよい（に含まれてもよい）。装置は、具体的にはエンコーダ／送信機であってもよい。図3は、図2の装置によって生成されたビットストリームなどのビットストリームを受信して処理する装置の一例を示す。図3の装置はデコーダ／受信機であってもよく、具体的には、図1のクライアントデバイス101（の一部）であってもよい。以下において、図2の装置はエンコーダとも呼ばれ、図3の装置はデコーダとも呼ばれる。

40

【0072】

この例では、エンコーダは音声環境を記述するビットストリームを生成する。この具体例では、ビットストリームは、環境内の音源によって生成される音声の音声データと、音響環境を記述するメタデータ（典型的には、個々の音源用のメタデータおよび音響環境用のメタデータの両方を含む）との両方を含む。しかし、一部の実施形態では、音声データおよびメタデータは別々のビットストリームで提供されてもよく、具体的には、音声データ用のメタデータのみを含む別個のビットストリームが生成され、実際の音声データは別個のビットストリームで別個に提供されてもよい。実際、一部の実施形態では、音声データ

50

は1つのソース／プロバイダから提供され、追加のメタデータは別のソース／プロバイダから提供され得る。

【0073】

エンコーダは、環境内の複数の音源を表す複数の音声要素用の音声データを生成するように構成された音声データ生成器201を含む。音声生成器は、例えば、個々の音源もしくは例えばマイクが拾った音を表す音声要素からの音声を記述する受信音声データから音声データを生成するか、または、例えばそのシーンの音声モデルに基づいて、音声データ自体を生成してもよい。一部の実施形態では、音声データ生成器201は、例えば、様々な音源からの音声の適切な表現を含むローカルストアから特定の音源用の音声データを抽出することができる。他の実施形態では、代わりにまたは追加で、音声データは、例えばライブ音声を取得するマイク入力から生成されてもよい。

10

【0074】

音声データ生成器201は、任意の適切なデータフォーマットに従うように音声データを生成することができ、任意の適切な符号化および表現などが使用され得る。したがって、音声データは任意の適切な方法で生成され得、任意の適切な方法で音声要素を表し得る。

【0075】

音声要素は、音声オブジェクト、音声クリップ、音声チャネル、一次もしくは高次アンビソニックス（FOA、HOA）、または他の音声要素であり得る。具体的には、各音声要素は、環境内で生成され得る音声を特徴付ける音声データのセットによって表されてもよい。音声データは一般に、複数の異なる音声要素の音声データのセットを複数組含むように生成され、具体的には、個々の音声要素によって表される複数の異なる音源の音声データのセットを複数組含む。

20

【0076】

エンコーダは、メタデータを生成するように構成されたメタデータ生成器203をさらに含む。メタデータは、環境の音響環境データを含み、音響環境データは、環境内の音源の音の伝播に影響を与える特性を記述する。音響環境データの特性は、環境の音響特性（例えば、残響、反射特性など）、音の伝播に影響を与える可能性がある環境内のオブジェクトの特性（例えば、オブジェクトの位置、向き、サイズ、材料、減衰、反射など）、または音源／音声要素の特性（例えば、位置、向き、サイズ、音量など）を含み得る。

30

【0077】

ビットストリームは、環境内で知覚される音声に影響を与える様々な特性の特性値のインジケーションを一緒に提供するいくつかのデータシンボル／ビットのデータグループを含み得る。また、様々な他のデータ、例えば、音響環境データの他のデータ用の補助パラメータまたはフォーマットの定義などを提供するデータグループが含まれてもよい。

【0078】

データグループは、データ値／フォーマットなどを示す単純なビットシーケンスであってもよく、または適切な情報を提供するデータのより複雑な組み合わせであってもよい。データグループは、例えば、多くのシナリオにおいて、ビットストリームの1つ以上のデータフィールドに対応するとみなされ得る。データフィールドはサブデータフィールドを含んでもよく、すなわち、それら自体がデータフィールドの組み合わせであるデータフィールドの階層的構成が適用されてもよい。

40

【0079】

音響環境データは、例えば、音声要素によって表される1つ以上の音源のポーズ（位置および／または向き）、環境または環境内の個々のオブジェクトの音響特性（例えば、（場合によっては周波数に依存する）減衰、残響特性、音波の伝播に影響を与える可能性のあるオブジェクトの形状、吸音、音響反射、音響散乱、音響結合、または音響伝達パラメータなどの材料特性など）、信号リファレンス、リファレンス距離、レンダリング制御フラグ、音響効果特性、ユーザインタラクションメタデータなどを記述するメタデータを含むことができる。

50

【 0 0 8 0 】

音響環境データは、複数のリスニングポーズに適用可能であり得るデータを含み得、具体的には、異なるリスナーポーズ（具体的には、異なる位置および／または向き）に依存するように（リスナーポーズごとに異なるように）、レンダラーが音声要素のレンダリングを適合させるために使用できるデータを含み得る。さらに、音響環境データは、静的特性および動的特性の両方のデータを含み得る。具体的には、音響環境データは、時間に依存せず、少なくとも、データが提供される時間間隔において時不变特性（値）を記述するデータを含むことができる（したがって、そのような特性は静的であり得る）。音響環境データは、時間に依存し、少なくとも、データが提供される時間間隔において時変特性（値）を記述するデータをさらに含むことができる（したがって、そのような特性は動的であり得る）。少なくとも1つの特性について、音響環境データは、特性（の値）の時間変化を記述するデータを含むことができる。特性は、環境内の音源、音声要素、および／または音響伝播／音の伝播のうちの1つの特性であってもよい。

10

【 0 0 8 1 】

エンコーダは、音声データおよびメタデータ（または一部の実施形態ではメタデータのみ）を含むビットストリームを生成するように構成されたビットストリーム生成器205をさらに含む。ビットストリームは、後に詳細に説明する特定のデータフォーマットの1つ以上の側面および特徴を満たすように生成され得る。

20

【 0 0 8 2 】

ビットストリーム生成器205は、生成されたビットストリームを出力するように構成された出力プロセッサ207に結合される。出力プロセッサ207は、特定のアプリケーションの要望に応じてビットストリームを通信するために必要な機能を有し得る。例えば、出力プロセッサは、ネットワークインターフェース、無線機能、Wi-Fi回路、インターネット接続機能などを有することができる。

20

【 0 0 8 3 】

図3に示すデコーダは、エンコーダによって生成されたビットストリームを受信するように構成された受信機または入力プロセッサ301を含む。入力プロセッサ301は、特定のアプリケーションの要望に応じてビットストリームを受信するために必要な機能を有し得る。例えば、入力プロセッサ301は、ネットワークインターフェース、無線機能、Wi-Fi回路、インターネット結合機能などを有し得る。多くの実施形態において、入力プロセッサ301は、エンコーダの出力プロセッサ207の機能に対して補完的な機能を有し得る。

30

【 0 0 8 4 】

入力プロセッサ301は、ビットストリームから音声データを抽出するように構成された音声データプロセッサ303に結合される。したがって、音声データプロセッサ303は、環境の多数の音声要素を表す音声データを抽出し、そして多くの場合、処理するように構成される。通常、各音声要素は環境内の1つ以上の音源に対応し得る。

30

【 0 0 8 5 】

説明されている例では、ビットストリームは、音声要素を記述する音声データを含む。他の実施形態では、音声データはビットストリームに含まれておらず、別のソースから受信されてもよく、例えば場合によっては、デコーダ／クライアントデバイスの内部ソースなどであってもよい。例えば、内部メモリは音声データを保存し得、サーバ103は、例えば、環境内で音源が移動する動的なアニメーションを提示するために音声を強化する情報を提供することによって、強化された体験を提供し得る追加のメタデータを提供することができる。一部の実施形態では、入力プロセッサ301は、サーバとは異なる外部ソースから音声データを受信するように構成されてもよく、例えば、音声データは、メタデータを提供するサーバ103とは異なるサーバによって提供される異なるビットストリームの一部として受信されてもよい。

40

【 0 0 8 6 】

デコーダは、ビットストリームからメタデータを抽出するように構成されたメタデータ

50

プロセッサ 305 をさらに含む。したがって、メタデータプロセッサ 305 は、音声要素 / 環境のメタデータを抽出し、そして多くの場合、処理するように構成される。メタデータプロセッサ 305 は、データを抽出し、メタデータによって記述される特性の 1 つ、複数、または全てに対して適切な特性値を生成するように構成され得る。

【0087】

この例におけるデコーダは、メタデータおよび特性値に基づいて音声データを処理するプロセッサを有する。具体的には、デコーダは、メタデータによって表される特性のうちの少なくとも 1 つの特性の値に基づいて 1 つ以上の音声要素をレンダリングするように構成されたレンダラー 307 を有し得、ここで、特性の値はメタデータから決定される。通常、レンダラー 307 は、メタデータに基づいて 1 つ以上の音声要素をレンダリングすることによって、環境に対応する音声シーンをレンダリングするように構成され得、例えば、メタデータによって決定されるポーズ（およびその変化）、環境および環境内のオブジェクトを記述するメタデータに基づく周波数依存減衰、メタデータによって記述される環境の残響特性を表す残響などが使用され得る。

【0088】

環境データおよびコンテキストデータに基づいて音声データをレンダリングするための多くのアルゴリズム、アプローチ、および技術が当業者に知られており（例えば、HRTF レンダリング、残響モデリングなどを含む）、簡潔かつ明確にするために、これについてはこれ以上詳細に説明しないことを理解されたい。

【0089】

以下に説明する特徴および側面のうちの 1 つ以上を含むデータフォーマットに従って、エンコーダはビットストリームを生成するように構成され、デコーダはビットストリームを受信、復号、および処理するように構成される。具体的な特徴および特性のうちの大部分または全てを含み得る特定のビットストリーム / フォーマットを参照してアプローチを説明するが、多くの実施形態では、例えば一部の特徴またはアプローチしか含まない可能性があるビットストリームが生成されることが理解されよう。実際、説明されるビットストリームの複数の異なる機能および要素は必ずしも併用されず、個別に使用されるか、または他の説明される機能と任意の方法で組み合わせができる個別の機能である。

【0090】

具体例では、連続した複数のデータセット（データセットはデータフレームとも呼ばれ得る）として構成された音響環境データを含むメタデータビットストリームが生成される。各データセットは、ある時間間隔についてのデータを含む。音響環境データは、ある時間間隔のデータ値がグループ化されるように構成され得、具体的には、単一の時間間隔にのみ適用される全ての符号化された音声データストリームがそのデータセットに含まれる連続データセットとしてグループ化され得、データセットは、その単一の時間間隔とは異なる時間間隔にのみ適用される音響環境データを含まない。所与の時間間隔に固有の全てのデータを取得するには、デコーダはそのデータセットを復号するだけで済む。一部の実施形態では、所与の時間間隔の音響環境データは他の時間間隔にも適用され得る（例えば、一部の静的音響環境データは全ての時間間隔に適用され得る）。そのようなマルチ間隔データは、複数の時間間隔のうちの 1 つの時間間隔のデータセット内、複数の時間間隔のうちの 2 つ以上の時間間隔のデータセット内、全てのデータセット内、またはデータセット外（例えば、個々の時間間隔のデータセットとは別に提供される共通データセットなど）に含まれ得る。

【0091】

したがって、ビットストリームは、複数の異なる時間間隔のデータセットを含み得る。以下では、各データセットはビットストリームのデータポイントとも呼ばれる。

【0092】

生成されるメタデータビットストリームは、多くの場合比較的静的なメタデータであって、特性（例えば、位置、向き、信号ゲイン）のアニメーション / 变化が比較的遅いメタデータを含む可能性がある。したがって、この例では、ビットストリームは、数ミリ秒の

10

20

30

40

50

範囲の継続時間有するフレームとして編成されず、はるかに長い時間スケールの時間間隔を表すデータセットまたはデータポイントに編成される。データセット/データポイントの時間間隔は、例えば0.5秒、1秒、2秒、または5秒以上であってもよい。

【0093】

ビットストリームは、ランダムに読み取りを始める場合、または複数のビットストリームが結合されている場合にビットストリームの復号を開始するのに十分なデータを含む独立したデータポイントを含み得る。このような独立したデータポイントのデータは、先行するデータポイントのデータに依存しない。このアプローチを使用したビットストリームの例を図4に示す。

【0094】

各データセット/ポイントによって表される時間間隔の継続時間はフレキシブルであってもよく、グループごとに異なってもよい。典型的には、継続時間は数秒程度であってもよい。データポイントはタイムスタンプに対して定義されてもよく、具体的には、データポイントのための始点、終点、または中間点であってもよい。データポイントのメタデータは、時間間隔の音響環境データの特性に関するデータ（例えば、具体的には、具体的な特性値を示すデータ）を表すことができる。例えば、示される値は、時間間隔の終わり（例えば、次のデータポイントに引き継がれる場所、またはシーンが終了する場所）の値であってもよい。さらに、値が時間間隔内で時間変化する場合、時間間隔内の他の時点での特性値を再構成することを可能にするデータが含まれてもよい。例えば、時間間隔内の値は補間によって決定され得る。

10

20

【0095】

この具体例では、時間間隔は、ビットストリームのデータ値によって示され得る。同様に、データフィールド/値によって示される特定の時間間隔に適用されるように、時間間隔内のデータ値が参照され、示されてもよい。具体的には、`targetOffset[n]`として参照される値が使用されてもよく、ここで、`n`はインデックスである。

【0096】

このアプローチでは、時間変化が起こるとき、例えばアニメーションが開始するとき、停止するとき、またはその範囲内で速度が変更するとき、以下のような様々なアプローチを使用してこれをサポートすることができる。

- ウェイポイント

30

- データポイントを更新する

【0097】

メタデータフィールドがアニメーション化され、データポイントの時間間隔内でアニメーションを複数回更新する必要がある場合、データポイントの時間間隔内に複数の`targetOffset`を含めることによって複数のウェイポイントを含めることができる。例えば、

- `TargetOffset[0] = 48000`
- `TargetOffset[1] = 20000`
- `TargetOffset[2] = 38643`

【0098】

データポイントは、時間間隔内の特定の時点にリンクされた1つ以上の特性のターゲット値を含み得る。その時点は、時間インジケーションによって示され、具体的には`targetOffset`フィールドの形態の時間インジケーションで示される。それこれが`targetOffset`リファレンスとペアにされた複数のターゲット値を提供することにより、値が補間される複数のウェイポイントを生成できる。図5は、これを使用して変化するパラメータ値を提供する方法を示す。

40

【0099】

代わりにまたは追加で、独立していない複数のデータポイントであり得る複数の更新データポイントによって、動的な変化がサポートされてもよい。更新データポイントは、その時間間隔の完全なレンダリングを可能にするのに十分なデータを提供せず、ビットスト

50

リーム内の他の場所に設けられたデータに依拠する可能性がある。更新データポイントは、更新データポイントによって表される時間間隔中の特性を記述する音響環境データのサブセットのみを含む可能性がある。一部の実施形態では、更新データポイントは常に、少なくとも1つの独立したデータポイントとリンクされ得る。

【0100】

更新データポイントは、場合によっては、時変特性（時間間隔中に変化する）に関するデータのみを含み得る。また、更新データポイントは、更新データポイントが適用される時間間隔を示すデータ、例えば開始時点、終了時点、中間点などの関連付けられたデータを有してもよい。更新データポイントは、次の独立したデータポイントを越さない最大時間間隔継続時間有してもよい。

10

【0101】

更新データポイントの利点は、対応するフィールドが独立したデータポイントよりも高いレートで送信されるシーン要素のライブストリーミング（例えば、他のユーザの動きおよびアクション）に有用であることあり得る。

【0102】

したがって、ビットストリームは、音の伝播に影響を及ぼす環境の特性についての1つ以上の特性値を含む複数の連続したデータセット／データポイントを含むことができる。さらに、ある特性値が適切である時間を示すための時間インジケーション（例えば、`targetOffset`）が含まれてもよい。一部の実施形態では、データセットは、1つ以上の特性について2つ以上の特性値を含むことができる。データセットは、可変特性について、複数の特性値がいつ適用されるべきかを示す時間インジケーションを含むことができる。

20

【0103】

一部の実施形態では、時間インジケーションは、時間間隔内のデフォルト時間と異なる場合にのみ含められてもよい。例えば、特性値はデフォルトでは時間間隔の終了時に示されており、時間間隔の終了時に提供される特性値には特定の時間インジケーションは提供されない可能性がある。しかし、時間間隔内の別の時間について提供される特性値は時間インジケーションに関連付けられ得る。

【0104】

したがって、一部の実施形態では、ビットストリーム生成器205は、所与の特性の特性値が、表される時間間隔内のデフォルト時間のために提供されたものであるか否かを判定するように構成される。そうである場合、第1の特性値は時間インジケーションなしでデータポイントに含められ、そうでない場合は、特性値に時間インジケーションが含まれる。

30

【0105】

ビットストリームは、それぞれ独立していて、レンダリングに追加のデータを必要としないデータセットと、それぞれ独立しておらず、音響環境の完全なレンダリングを可能にするために他のデータセットからの音響環境データを必要とするデータセットとの両方を含むように生成されてもよい。したがって、音響環境データは、環境の音声をレンダリングするために必要なデータを全て含むいくつかの完全なレンダリングデータセットと、環境の音声をレンダリングするために他のデータセットからのデータに追加で依拠し、そのようなデータを必要とするいくつかの部分的レンダリングデータセットとを含む可能性がある。

40

【0106】

データポイントが独立しており、最大数秒のギャップをカバーする可能性があるという概念によれば、データポイントの時間間隔の始点および終点の両方を含むと有用である可能性がある（例えば、データポイントの開始位置と、補間間隔の終了位置とを示す）。多くの場合で終点のみを指定することが許容され、また、デコーダがストリームに割り込むとき、最初に復号されるデータポイントの間隔の継続時間にかけてのみ潜在的な逸脱が発生する可能性があることから、これは任意選択である。

50

【0107】

多くの場合、ソースポーズの終点が最も重要な情報とみなされる可能性があり、これは位置および向きのデータを含み得る。加えて、任意選択の開始位置 / 向きが提供されてもよい。より効率的なコード化によってビットレートを節約するために、これは終点に対して差動コード化されてもよい。補間不可能なデータ（例えば、フラグ、識別子、インデックス）の場合、変化が瞬時に発生するオフセットが存在する可能性があり、データポイントの開始時の値が示される可能性がある（したがって、差動ではない）。フラグの場合、開始時の値は、提供されたインジケーションに続く値の反対であるとみなされるため、示されない可能性がある。

【0108】

10

JSON構造では、このようなアプローチの例は次の通りである。

- ObjectSource []
 - ID
 - SignalIndex
 - PreGain
 - PreGainInterpMethod (任意選択)
 - PreGainInterpLengthIdx (任意選択)
 - PreGainDelta (任意選択) % 存在はそのデータポイント内にアニメーションがあることを伝える

20

- 位置
 - PositionInterpMethod (任意選択)
 - PositionInterpLengthIdx (任意選択)
 - PositionDelta (任意選択) % 存在はそのデータポイント内にアニメーションがあることを伝える

20

- 向き
 - OrientationInterpMethod (任意選択)
 - OrientationInterpLengthIdx (任意選択)
 - OrientationDelta (任意選択) % 存在はそのデータポイント内にアニメーションがあることを伝える

30

- レンダリング
 - RenderUpdateOffsetIdx (任意選択) % 存在はそのデータポイント内にアニメーションがあることを伝える
- 指向性ID
 - DirectivityIDUpdateOffsetIdx (任意選択)
 - DirectivityIDStart (任意選択) % 存在はそのデータポイント内にアニメーションがあることを伝える

【0109】

ここで、終点は [位置 , 方向] であり、始点は [位置 - PositionDelta , 向き - OrientationDelta] である。これは、終了位置が、必要に応じて開始位置を再構築することを可能にする絶対値としても相対値としても提供され得ることを意味する。

40

【0110】

例えば、次のいずれかを定義するフィールドInterpMethodなどによって、様々な補間方法を指示示すことができる。

- 'linear' <デフォルト>
- 'instant' (ターゲットのタイムスタンプにおいて、補間なしで瞬時に変化させる。例えば、SignalIndex、ExtentID、または任意のBooleanフィールドの変形に有用である)
- 'spherical'
- 'logarithmic'

50

【0111】

終点のターゲットは、次の独立したデータポイントの始点、任意の中間更新データポイント、または中間ウェイポイントであり得る。ビットストリームの一般部分の中に複数の潜在的なターゲットが列挙され、ビットストリームの他の部分がそれを効率的に参照することで、他の部分がカバーするデータのために妥当なターゲットが示されてもよい。ターゲットは、所与の時点でのターゲット値 / 意図された値であってもよい。

【0112】

以下、単一のデータポイント / データセットのデータのフォーマットおよび構文の例を説明する。この説明は、MPEG-H 3D Audio (ISO/IEC 23008-3) などの MPEG Audio 規格で使用されるビットストリームフォーマットを記述するアプローチに沿ったものである。構文記述は疑似コードとして構造化されており、関数呼び出しは、その関数の下で記述された構文が、関数呼び出しが行われた場所に挿入されることを示す。ビットストリーム内のビットを占めるフィールドは太字で表示されており、2列目および3列目は二進数を使用してビット数およびビットフォーマットを示している。一部のフィールドは、表される値に応じて、可変数 n_r のビットを有し得る。これらのフィールドは、コードワードおよび対応する値を記述するルックアップテーブルに関連付けられている。コードワードは、ビットごとに読み取りを開始するときに、より短いコードワードがより長いコードワードの最初の方のビットと決して重ならないよう設計されている。これは、可逆データ符号化またはエントロピー符号化として当業者に知られており、例えば、ハフマン符号化、ランレンジス符号化、または算術符号化が挙げられる。構文の前半で読み取られたデータが、他の部分の復号を制御するために、構文の後半の部分で使用されてもよい。例えば、ビット数、要素数、符号化の方法、特定のデータの存在などを通知することによってである。

10

20

30

30

40

【0113】

記述では、以下の特定の頭字語が使用される (ISO/IEC 23008-3、MPEG-H 3D Audio仕様から引用)。

- **bs1bf** - ビット列、左ビットが最初 (bit string, left bit first)。「左」は、ISO/IEC 14496 でビット列が書き込まれる順序である。ビット列は、例えば '1000 0001' のように、一重引用符で囲まれた 1 と 0 の文字列として記述される。ビット列内の空白は読みやすくするためのものであり、特に意味はない。

- **boolean** に使用される

- **tcimsbf** - 2 の補数の整数、最上位 (符号) ビットが最初 (two's complement integer, most significant (sign) bit first)。

- **uimsbf** - 符号なし整数最上位ビットが最初 (Unsigned Integer Most Significant Bit First)。

- **uint** に使用される

- **v1clbf** - 可変長コード、左ビットが最初 (variable length code, left bit first)。「左」は可変長コードが書き込まれる順序を指す。

【0114】

データセット / ポイントは、次のフォーマット / 構文に従って提供され得る。

DataPoint()

【表 1】

構文	ビット数	タイプ
DataPoint()		
{		
dpType;	2	uimsbf
DEBUG_DATA();		
[fs, isSmallScene] = GeneralData();		
TargetData(fs);		
FreqGridData();		
DirectivityPatternData();		
AcousticElementData(isSmallScene, fs);		
AudioSourceData();		
AcousticEnvironmentData(fs);		
MaterialData();		
VoxelData();		
GeometricElementData(isSmallScene);		
UserInteractionData(isSmallScene, fs);		
reservedBits;	16	
}		

【0 1 1 5】

以下では、そのようなデータポイントの様々なフィールド / 部分 / 要素がより詳細に説明されるが、説明されるフィールド / 部分 / 要素は具体的なデータポイントまたは構文に限定されず、音響環境データを含む任意のビットストリーム内で個別に、または任意の組み合わせで使用され得ることが理解されるであろう。

【0 1 1 6】

データポイントの機能 / フィールド / 部分要素のさらなる説明。

【0 1 1 7】

d p T y p e

データポイントのタイプ。データポイントのタイプを示す。

10

20

30

40

50

【表2】

値	タイプ
0	独立したデータポイント – ビットストリームへのランダムアクセスのために使用される
1	更新データポイント – 独立したデータポイント間でデータ更新を運ぶために使用される。先行する独立したデータポイントからの情報に依拠する。
2	予約済み
3	予約済み

10

20

30

40

50

【0118】

reserved Bits

予約済みビット。これらを使用して、拡張機構を導入することができる。

【0119】

したがって、データポイントは、データポイントの特定のタイプを示す2つのビットで始まり得る。2つのビットは、具体的には、データポイントが独立したデータポイントであるか、または更新ポイントであるかを示す。

【0120】

General Data()

フィールドGeneral Data()は、一般的な構成またはレンダリングデータでいくつかのパラメータに関する情報を提供するデータを提供し得る。

【表3】

構文	ビット数	タイプ
[fs, isSmallScene] = GeneralData()		
{		
bsVersion;	4	uimsbf
isSmallScene;	1	bslbf
fs = LUT(usacSamplingFrequencyIndex);	5	bslbf
if (usacSamplingFrequency == 0x1f)		
{		
fs = usacSamplingFrequency;	24	uimsbf
}		
if (useDefaultSpeedOfSound == false)	1	bslbf
{		
speedOfSound = 343;		
} else		
{		
speedOfSound;	14	uimsbf
}		
return [fs, isSmallScene];		
}		

【0121】

40

bsVersion

ビットストリームバージョン。

【表4】

値	記述
0	CfP応答用の第1のバージョン
1...15	予約済み

50

【 0 1 2 2 】

i s S m a l l S c e n e

真に設定されている場合、[- 1 0 0 . . 1 0 0] メートルの範囲外の座標が存在しないことを示す。

【 0 1 2 3 】

一部の実施形態では、メタデータ生成器 2 0 3 は、環境が、空間的に制約された環境であることを示すグローバルインジケータを含む音響環境データを生成するように構成される。例えば、グローバルインジケータは、真または偽に設定され得る *i s S m a l l S c e n e* の形態であり得る。真に設定すると、*i s S m a l l S c e n e* は [- 1 0 0 . . 1 0 0] メートルを超えない座標で表現されるように制限される。

10

【 0 1 2 4 】

空間的に制限された環境を示すグローバルインジケータが設定されている場合、データ値の所定の制限付きフォーマットに準拠するように、多くのデータ値が制限される。具体的には、空間座標（具体的には位置値）は、[- 1 0 0 . . 1 0 0] メートルの間隔に制限されるなど、閾値を超えないように制限され得る。一部の実施形態では、空間的に制限された環境を示すようにグローバルインジケータが設定されているとき、他のパラメータが、限定された範囲に制限され得る。例えば、小さい環境の場合、残響の最大継続時間または時定数（例えば、*T₆₀* 値）が制限される可能性がある。

【 0 1 2 5 】

空間的に制限された環境は、閾値未満の空間的広がりを有する環境であり得る。所定の制限付きフォーマットは、環境が空間的に制限された環境でない場合にビットストリームに使用されるデータフォーマットよりも少ないビットを使用する、音の伝播に影響する特性の少なくともいくつかの値のデータフォーマットであり得る。

20

【 0 1 2 6 】

一部の実施形態では、1つ以上のパラメータ値の制限を示すグローバルフラグまたはインジケータが含まれてもよい。

【 0 1 2 7 】

u s a c S a m p l i n g F r e q u e n c y I n d e x

音声信号に使用されるサンプルレートのインデックス。2 3 0 0 3 - 3 の定義に基づき、

30

40

50

【表 5】

usacSamplingFrequencyIndex	サンプリング周波数
0x00	96000
0x01	88200
0x02	64000
0x03	48000
0x04	44100
0x05	32000
0x06	24000
0x07	22050
0x08	16000
0x09	12000
0x0a	11025
0x0b	8000
0x0c	7350
0x0d	予約済み
0x0e	予約済み
0x0f	57600
0x10	51200
0x11	40000
0x12	38400
0x13	34150
0x14	28800
0x15	25600
0x16	20000
0x17	19200
0x18	17075

10

20

30

40

50

0x19	14400
0x1a	12800
0x1b	9600
0x1c	予約済み
0x1d	予約済み
0x1e	予約済み
0x1f	エスケープ値

注: `UsacSamplingFrequencyIndex` の値 `0x00` ~ `0x0e` は、ISO/IEC 14496-3:2009 で規定されている `AudioSpecificConfig()` に含まれる `samplingFrequencyIndex` の値 `0x0` ~ `0xe` と同一である。

10

20

30

40

【0128】

`usacSamplingFrequency`
`usacSamplingFrequencyIndex = 0` である場合、デコーダの出力サンプリング周波数は、符号なし整数値としてコード化される。

【0129】

`useDefaultSpeedOfSound`
 材料なし媒体に対してデフォルトの音速 (20°C での値である 343 m/s) を使用するか、またはカスタム値を指定するかを示すフラグ。

【0130】

`speedOfSound`
 材料なし媒体用のカスタム音速値。材料なし媒体は、異なる材料が割り当てられたジオメトリによって占有されていない空間とみなされる。

【0131】

`TargetData()`
 フィールド `TargetData()` は、データポイントの他の部分からインデックスによって参照される、属性のアニメーションのターゲットオフセットを含み得る。

【表 6】

構文	ビット数	タイプ
TargetData(fs)		
{		
tdNrTargets = GetCountOrIndex();		
for t = 0:tdNrTargets - 1		
{		
targetOffset[t] = GetDuration(fs);		
}		
}		

10

20

30

40

50

【0 1 3 2】

FreqGridData()

フィールドは、データポイントの他の部分からインデックスによって参照される周波数グリッド定義を含み得る。これは通常、一部のビットストリームパーサに、次にコード化される周波数依存要素の数を通知する。

【0 1 3 3】

フィールド FreqGridData() は、データポイントの他の部分からインデックスによって参照される周波数グリッド定義に関する情報を提供するデータを提供し得る。具体的には、周波数グリッド / 複数の周波数範囲への細分割が記述されてもよく、定義された周波数グリッドに従って、ビットストリームの他のパラメータ / 他のデータが提供され得る。例えば、周波数グリッドの定義された周波数の複数の異なる範囲に対して減衰を示すことによって、異なる周波数依存フィルタリングを提供することができる。したがって、これらの減衰値に対応する周波数を明示的に記述する必要なく、フィルタを単純に減衰値のセットとして記述することができる。

【0 1 3 4】

一部の実施形態では、音響環境データは、周波数グリッドを記述する 1 つ以上のデータグループ / データフィールドと、周波数グリッドを使用して特性のうちの周波数依存特性を記述するデータをそれが含む複数のデータグループ / データフィールドとを含むように構成され得る。周波数グリッドを使用して特性のうちの周波数依存特性を記述するデータは、例えば、周波数値および / または周波数グリッドによって定義される周波数値のうちの 1 つ以上の周波数値についてデータ値を提供することによって、これを行うことができる。

【0 1 3 5】

周波数グリッドは、例えば周波数部分範囲の中心周波数を定義することによって、周波数範囲を周波数部分範囲に細分化したものであってもよい。

【0 1 3 6】

一部の実施形態では、ビットストリームは、ビットストリームが、周波数グリッドを記述するデータグループを含むか否かを示すインジケータを含むことができる。例えば、单一のビットである bFlagPresent が、周波数グリッド定義が (例えば、次のビットに) 含まれているか否かを示し得る。

【0 1 3 7】

一部の実施形態では、ビットストリームは、周波数グリッドを記述するデータのフォーマットのインジケーションを含むことができる。例えば、ビットストリームは、例えば所定の／事前定義された周波数グリッドのセットのうちのグリッドのインデックスを示すことによって、周波数グリッドが事前定義されたグリッドへの参照によって記述されているか否かを記述するインジケーションを含むことができる。別の例として、ビットストリームは、周波数グリッドの少なくとも一部の部分範囲について、典型的には周波数グリッドの全ての部分範囲について、開始周波数および周波数範囲を示すデータを含むことができる。一部の実施形態では、周波数グリッドは、周波数範囲／周波数間隔の境界周波数を示す遷移周波数のセットによって示され得る。例えば、第1の周波数間隔／範囲が、第1の周波数間隔／範囲の開始周波数および次の周波数間隔／範囲の開始周波数を示すデータによって示されてもよい。

10

【0138】

多くの実施形態では、周波数範囲は、周波数スケールの対数表現において一定であり得る。

【0139】

周波数バンディング (banding) および範囲／間隔への分割は、オクターブに基づいて行われてもよい。1オクターブの差は周波数（例えば、125、250、500、1000 Hz）を2倍することを表す。ビットストリームは、例えば、オクターブバンドまたは別の下位区分（例えば、オクターブバンド間にさらに2つの値を配置する1/3オクターブバンド（125、160、200、250、315、400、500、630、800、1000））におけるバンディングがあるか否かを示す。一部の実施形態では、オクターブバンドの分数に位置合わせされたデータによって、少なくとも1つの部分範囲または個別の周波数が示されてもよい。

20

【0140】

一部の実施形態では、ビットストリームは、個別の周波数、例えば複数の個別の周波数のセットを示すデータを含むことができる。その場合、ビットストリーム内において、他の周波数依存特性は、これらの個別の周波数を特性ごとに明示的に記述する必要なく、これらの個別の周波数のための特性値のセットによって単純に表され得る。

【0141】

多くの実施形態では、その場合、周波数グリッドはビットストリームのデータによって記述／定義され得る。例えば、周波数グリッドの異なるモードが使用されてもよく、データは、周波数グリッドを記述するために使用されるモードを示してもよい。

30

【0142】

例えば、フィールド `fgdMethod` は、例えば次のうちのどのモードが使用されるかを示してもよい。

- デフォルトのグリッド
 - ・例えば、オクターブバンドの分数に位置合わせされる
- 開始周波数 + 周波数ホップサイズ + バンドの量
 - ・例えば、オクターブバンドの分数に位置合わせされる
- 個別の周波数
 - ・例えば、オクターブバンドの分数に位置合わせされる

40

【0143】

周波数グリッドのフォーマット／構文の例は以下の通りである。

50

【表 7】

構文	ビット数	タイプ
FreqGridData()		
if (bFgdPresent)	1	bs\bf
{		
fgdNrGrids = GetCountOrIndex();		
for g = 0:fgdNrGrids - 1		
{		
fgdMethod;	2	uimsbf
if (fgdMethod == 'Individual frequencies')		
{		
fgdNrBands[g] = GetCountOrIndex();		
for b = 0:fgdNrBands[g] - 1		
{		
fgdCenterFreq[g][b] = GetFrequency();		
}		
} elseif (fgdMethod == 'Start-Hop-Amount')		
{		
fgdNrBands[g] = GetCountOrIndex();		
fgdCenterFreq[g][0] = GetFrequency();		
frequencyHop = LUT(frequencyHopCode);	var	vlclbf
for b = 1:fgdNrBands[g] - 1		
{		
fgdCenterFreq[g][b] = fgdCenterFreq[g][b - 1] * frequencyHop;		
}		
} elseif (fgdMethod == 'Default banding')		
{		

10

20

30

40

50

fgdDefaultBanding;	4	
}		
}		
}		
}		

10

【 0 1 4 4 】

b F g d P r e s e n t

周波数グリッドが定義されているかを示すフラグ。

【 0 1 4 5 】

f g d M e t h o d

周波数グリッドがコード化されている方法。

【 表 8 】

値	名称
0	個別の周波数
1	開始一ホップー量
2	デフォルトのバンディング
3	予約済み

20

【 0 1 4 6 】

f g d C e n t e r F r e q

30

各周波数グリッドの各バンドの中心周波数を H z で示す。

【 0 1 4 7 】

f r e q u e n c y H o p C o d e

周波数バンディングのホップ係数を示すコード。

40

50

【表 9】

コードワード	値
1100	$2^{\wedge}(1/8)$
1101	$2^{\wedge}(1/7)$
0010	$2^{\wedge}(1/6)$
0011	$2^{\wedge}(1/5)$
0000	$2^{\wedge}(1/4)$
01	$2^{\wedge}(1/3)$
0001	$2^{\wedge}(1/2)$
10	$2^{\wedge}1$
111	$2^{\wedge}2$

10

20

【0 1 4 8】

`f g d D e f a u l t B a n d i n g 1` についくつかのデフォルトバンディングスキームを定義する。

【表 10】

値	<code>f g d C e n t e r F r e q [g]</code>
0	[100 200 500 1000 2000 5000 10000]
1	[1000 2000 3000 4000 5000 6000 7000 8000]
2..15	予約済み

30

【0 1 4 9】

したがって、一部の実施形態では、ビットストリームは、音響環境データの周波数依存特性を記述するために他のデータグループにおいて使用される周波数グリッドを記述するデータグループを含むように生成される。したがって、周波数グリッドがデータストリーム内に提供され、周波数変化を記述するために使用される。さらに、ビットストリームは、ビットストリームが周波数グリッドの記述を含むか否かを示す特定のインジケータを含む。したがって、通信される具体的な特性および周波数依存性のために最適化できる任意選択でカスタマイズ可能な周波数グリッドを使用するようにビットストリーム生成を適合可能な柔軟なアプローチが提供される。単純な例として、周波数分解能または周波数範囲が柔軟に適合できてもよく、実際には、周波数グリッド記述全体が任意選択で、記述が含まれるか否かのインジケータを適切に設定することによって省略できる。

40

【0 1 5 0】

また、データグループが、周波数グリッドの記述を含むだけでなく、周波数グリッドの記述に用いられるフォーマットを具体的に示すような構成であってもよい。したがって、受信機が周波数グリッド記述データを解釈することを可能にするインジケーションが提供されてもよく、これにより、ビットストリーム生成器は、具体的な特性およびそれらの特性の具体的な周波数依存性に特に適した周波数グリッドの記述のためのフォーマットを自由に選択できるようになる。

【0 1 5 1】

周波数グリッドを記述するデータグループは、所定のデフォルトグリッド、周波数グリ

50

ッドの少なくとも一部の部分範囲の開始周波数および周波数範囲、ならびに / または個別の周波数を示すデータを含み得る。

【 0 1 5 2 】

したがって、このアプローチは、このような音響環境データをビットストリーム内に符号化して表現するための具体的な狭いアプローチを使用することにより、ビットストリームに周波数変化特性を含めることを可能にすることができる。このアプローチは、上記のような周波数依存の音響環境特性をビットストリームに含めるための、非常に効率的で、適合性があり、柔軟なアプローチを提供することができる。

【 0 1 5 3 】

Acoustic Element Data()

10

構文のこの部分は、音響要素を記述するデータをカバーする。これらは通常、シーディングラフの階層要素として機能する汎用要素である。通常、このような要素の特性は子ノードに引き継がれる。すなわち、音響要素のプリゲイン値は、対応する音響要素の下に編成された音源にも適用される。これらのノードの位置および向きは、親の向きまたはポーズを無視するようにポーズ依存関係が設定されているノードでない限り、子ノードの位置および向きを変換する。

【 0 1 5 4 】

一部の実施形態では、音響環境データは、少なくとも 1 つの音声要素のためのアニメーションインジケーションを含む。アニメーションインジケーションは、ある時間間隔中に音声要素の特性が変化するか否かを示し、したがって、特性が動的特性であるか静的特性であるかを示す。具体的には、あるデータポイント / グループについて、そのデータポイント / グループの時間間隔内で音声要素が静的であるか時間変化するかを示すフラグまたはインジケーションが含まれてもよい。

20

【 0 1 5 5 】

対応する音声が時間変化することをアニメーションインジケーションが示している場合、音響環境データは、変化する特性の変化を記述するデータをさらに含むことができる。このような音声要素の音響環境データは、音声要素の少なくとも 1 つの特性について、その特性が時間変化する特性であるか否かを示すデータを含むことができる。したがって、(データセット / ポイントの時間間隔内で) どの特性が時間変化し、どの特性が時間変化しないかを示す、特性に関するインジケーションが含まれてもよい。

30

【 0 1 5 6 】

多くの実施形態では、音響環境データは、時間間隔内の異なる時間での特性値を決定する方法を記述するデータを含むことができ、具体的には、音響環境データは、異なる時間での特性値を決定するために適用される補間アプローチを記述するデータを含むことができる。

【 0 1 5 7 】

音響環境データは、特定の時間変化する特性について 2 つ以上の値を含み得、各値は特定の時点のものであり得る。そして、他の時間についての特性値は補間によって決定され得る。符号化された音声データストリーム、特に個別のデータポイントは、時間変化する特性について実行されるべき時間的補間の特性を記述するデータを含み得る。

40

【 0 1 5 8 】

場合によっては、データポイントの時間間隔内で補間に使用される値のうちの 1 つ以上が、時間間隔 / データポイント外で提供されるか、提供される値から決定されてもよい。例えば、1 つ以上の値が、例えばビットストリーム内で先に送信されたものであってもよい別のデータポイントのデータから導出され得る。このようなデータ値は、補間が適用される (および補間がそれを対象として定義されている可能性がある) データポイントの時間間隔の前または後の時間であり得る特定の時点 (例えば、タイムスタンプによって示される) に関連付けられる可能性がある。

【 0 1 5 9 】

一部の実施形態では、データポイントごとに異なる補間が記述され、したがって、時間

50

間隔ごとに異なる補間が適用され得る。また、異なる特性（同じデータポイント／時間間隔の異なる特性を含む）に対して異なる補間が記述されてもよい。

【0160】

補間方法は、例えば、フラグ * `InterpMethod` によって表されるコードによって示され得る。コードは、例えば、以下の表 1 に示す通りであり得る。

【表 11】

コード	方法
0	線形
1	瞬間
2	予約済み
3	予約済み
4	予約済み
5	予約済み
6	予約済み
7	予約済み

10

20

30

【0161】

一部の実施形態では、環境の少なくとも一部の要素の音響環境データは、環境のシングラフの要素の識別データおよび親識別データを含むことができる。これらの少なくとも一部の要素は、例えば、オブジェクト、音源（音声要素）、および / または環境の音響特性であり得る。

【0162】

例えば、複数の異なるオブジェクトおよび / または音源がシングラフ内に配置され得、音響環境データは、このシングラフを示すデータを含み得る。これは、具体的には、個別の要素の識別データおよび親識別情報を提供することによって実現することができる。なぜなら、これによってシングラフの再構築が可能になり、したがって階層を表すように構成され得る。

【0163】

一部の実施形態では、要素のアニメーションが、音響環境データ内に示されてもよい（場合によっては記述されてもよい）。

【0164】

例えば、音響環境データは、要素 / ソースごとに（時間セグメント内で）アニメーション化されているか否かを示す `ElementAnimated`、`SourceAnimated`、および `AttributeAnimated` フラグ / インジケーションを含むことができる。その場合、属性 / 特性がアニメーション化されているか否かを示すために、属性 / 特性ごとにフラグ / インジケーションを含めることができる。その場合、アニメーション / 時間変化を記述するデータがさらに含まれ得る。

40

【0165】

上記機能の例として、次のフォーマットが使用されてもよい。

50

【表 1 2】

構文	ビット数	タイプ
AcousticElementData(isSmallScene)		
{		
if (aedPresent)	1	bslbf
{		
aedNrElements = GetCountOrIndex();		
for e = 0:aedNrElements - 1		
{		
elementAnimated;	1	bslbf
aedID = GetID();		
aedParentID = GetID();		
aedPoseDependency;	3	uimsbf
aedPosition =GetPosition(isSmallScene);		
if (elementAnimated)		
{		
if (attributeAnimated)	1	bslbf
{		
aedPositionInterpMethod;	3	uimsbf
aedPositionInterpTargetIdx = GetCountOrIndex();		
aedPositionDelta = GetPositionDelta(isSmallScene);		
while (attributeAnimated)	1	bslbf
{		
aedPosition =GetPosition(isSmallScene);		
aedPositionInterpTargetIdx = GetCountOrIndex();		
}		
}		

10

20

30

40

50

}		
aedOrientation = GetOrientation();		
if (elementAnimated)		
{		
if (attributeAnimated)	1	bs\bf
{		
aedOrientationInterpMethod;	3	uimsbf
aedOrientationInterpTargetIdx =		
GetCountOrIndex();		
aedOrientationDelta = GetOrientation();		
while (attributeAnimated)	1	bs\bf
{		
aedOrientation = GetOrientation();		
aedOrientationInterpTargetIdx =		
GetCountOrIndex();		20
}		
}		
}		
aedPregain = GetGain();		
if (elementAnimated)		
{		
if (attributeAnimated)	1	bs\bf
{		
aedPregainInterpMethod;	3	uimsbf
aedPregainInterpTargetIdx =		
GetCountOrIndex();		
aedPregainDelta = GetGainDelta();		
while (attributeAnimated)	1	bs\bf
{		
aedPregain = GetGain();		40

10

20

30

40

50

aedPregainInterpTargetIdx = GetCountOrIndex();		
}		
}		
}		
aedRender;	1	bs bf
if (elementAnimated)		
{		
while (attributeAnimated)	1	bs bf
{		
aedRenderUpdateTargetIdx = GetCountOrIndex();		
}		
}		
}		
}		
}		

10

20

30

【0166】

a e d P r e s e n t

音響要素が定義されているか否かを示すフラグ。

【0167】

a e d N r E l e m e n t s

定義された音響要素の数を通知する。

【0168】

e l e m e n t A n i m a t e d

そのデータポイントにおいて、対応する要素がアニメーション化されているか否かを示す
フラグ。

【0169】

a t t r i b u t e A n i m a t e d

そのデータポイントにおいて、対応する属性がアニメーション化されているか否かを示す
フラグ。また、より多くのウェイポイントについてのデータが存在するか否かを示すため
にも使用される。

【0170】

a e d I D

音響要素の I D。

【0171】

40

50

a e d P a r e n t I D

シーングラフ内の音響要素の親の I D。

【 0 1 7 2 】

a e d P o s e D e p e n d e n c y

音響要素のポーズが何に相対的であるかを示す。

【 0 1 7 3 】

a e d P o s i t i o n

音響要素の位置座標 (x , y , z) をメートル単位で示す。

同じ要素が複数回出現する場合、これは、データポイントの範囲内に複数のウェイポイントがあることを示す。

【 0 1 7 4 】

a e d P o s i t i o n I n t e r p M e t h o d

位置アニメーションに使用する補間方法を示す (表 1 参照) 。

【 0 1 7 5 】

a e d P o s i t i o n I n t e r p T a r g e t I d x

T a r g e t O f f s e t へのインデックス。先行する位置ターゲット値が有効である、データポイントのタイムスタンプからのオフセットを示す。

データポイントの範囲内のウェイポイントごとに 1 つのターゲットインデックスが存在するように、複数のターゲットインデックスが提供されてもよい。

【 0 1 7 6 】

a e d P o s i t i o n D e l t a

データポイントのタイムスタンプにおける位置の値を再構築することを可能にする位置デルタ値。

PositionAtDPSStart = a e d P o s i t i o n - a e d P o s i t i o n D e l t a

複数のターゲット値が存在する場合、最初のターゲット値を基準とする。

【 0 1 7 7 】

a e d O r i e n t a t i o n

音響要素の向き (ヨー , ピッチ , ロール) をラジアン単位で示す。

同じ要素が複数回出現する場合、これは、データポイントの範囲内に複数のウェイポイントがあることを示す。

【 0 1 7 8 】

a e d O r i e n t a t i o n I n t e r p M e t h o d

向きアニメーションに使用する補間方法を示す (表 1 参照) 。

【 0 1 7 9 】

a e d O r i e n t a t i o n I n t e r p T a r g e t I d x

T a r g e t O f f s e t へのインデックス。先行する向きターゲット値が有効である、データポイントのタイムスタンプからのオフセットを示す。

データポイントの範囲内のウェイポイントごとに 1 つのターゲットインデックスが存在するように、複数のターゲットインデックスが提供されてもよい。

【 0 1 8 0 】

a e d O r i e n t a t i o n D e l t a

データポイントのタイムスタンプにおける向きの値を再構築することを可能にする向きデルタ値。

OrientationAtDPSStart = a e d O r i e n t a t i o n - a e d O r i e n t a t i o n D e l t a

複数のターゲット値が存在する場合、最初のターゲット値を基準とする。

【 0 1 8 1 】

a e d P r e g a i n

対応する音響要素の下に階層的に配置された全てのソースのプリゲイン値 (d B) 。

10

20

30

40

50

同じ要素が複数回出現する場合、これは、データポイントの範囲内に複数のウェイポイントがあることを示す。

【0182】

`a e d P r e g a i n I n t e r p M e t h o d`

プリゲインアニメーションに使用する補間方法を示す（表1参照）。

【0183】

`a e d P r e g a i n I n t e r p T a r g e t I d x`

`T a r g e t O f f s e t`へのインデックス。先行するプリゲインターゲット値が有効である、データポイントのタイムスタンプからのオフセットを示す。

データポイントの範囲内のウェイポイントごとに1つのターゲットインデックスが存在するように、複数のターゲットインデックスが提供されてもよい。

【0184】

`a e d P r e g a i n D e l t a`

データポイントのタイムスタンプにおけるプリゲインの値を再構築することを可能にするプリゲインデルタ値。

`P r e g a i n A t D P S t a r t = a e d P r e g a i n - a e d P r e g a i n D e l t a`

複数のターゲット値が存在する場合、最初のターゲット値を基準とする。

【0185】

`a e d R e n d e r`

レンダリングフラグ。偽の場合、対応する音響要素の下に階層的に配置された全てのソースをレンダリングしないことを示す。

【0186】

`a e d R e n d e r U p d a t e T a r g e t I d x`

`T a r g e t O f f s e t`へのインデックス。先行する `a e d R e n d e r` 値へとレンダリングフラグが状態を反転する、データポイントのタイムスタンプからのオフセットを示す。

データポイントの範囲内のウェイポイントごとに1つのターゲットインデックスが存在するように、複数のターゲットインデックスが提供されてもよい。各ターゲットインデックスは、フラグの状態のバイナリ反転を示す。

【0187】

`A u d i o S o u r c e D a t a () A u d i o S o u r c e D a t a ()`

構文のこの部分は、様々なタイプの音源を集めている。

【表13】

構文	ビット数	タイプ
<code>AudioSourceData(isSmallScene, fs)</code>		
{		
<code>ObjectSourceData(isSmallScene);</code>		
<code>HOASourceData(isSmallScene);</code>		
<code>SoundEffectData(fs);</code>		
}		

【0188】

`O b j e c t S o u r c e D a t a ()`

10

20

30

40

50

構文のこの部分は、オブジェクトソースの特性を記述する。

【表 14】

構文	ビット数	タイプ
ObjectSourceData(isSmallScene)		
{		
if (osdPresent)	1	bslbf
{		
osdNrElements = GetCountOrIndex();		
for e = 0:osdNrElements - 1		
{		
sourceAnimated;	1	bslbf
osdID = GetID();		
osdParentID = GetID();		
osdSignalIndex = GetCountOrIndex();		
osdIsContinuousSource;	1	bslbf
osdReferenceDistance = GetDistance(isSmallScene);		
osdPoseDependency;	3	uimsbf
osdPosition =GetPosition(isSmallScene);		
if (sourceAnimated)		
{		
if (attributeAnimated)	1	bslbf
{		
osdPositionInterpMethod;	3	uimsbf
osdPositionInterpTargetIdx = GetCountOrIndex();		
osdPositionDelta = GetPositionDelta(isSmallScene);		
while (attributeAnimated)	1	bslbf
{		
osdPosition =GetPosition(isSmallScene);		

10

20

30

40

50

osdPositionInterpTargetIdx = GetCountOrIndex();		
}		
}		
}		
osdOrientation = GetOrientation();		
if (sourceAnimated)		
{		
if (attributeAnimated)	1	bslbf
{		
osdOrientationInterpMethod;	3	uimsbf
osdOrientationInterpTargetIdx = GetCountOrIndex();		
osdOrientationDelta = GetOrientation();		
while (attributeAnimated)	1	bslbf
{		
osdOrientation = GetOrientation();		
osdOrientationInterpTargetIdx = GetCountOrIndex();		
}		
}		
}		
osdPregain = GetGain();		
if (sourceAnimated)		
{		
if (attributeAnimated)	1	bslbf
osdPregainInterpMethod;	3	uimsbf
osdPregainInterpTargetIdx = GetCountOrIndex();		
osdPregainDelta = GetGainDelta();		

10

20

30

40

50

while (attributeAnimated)	1	bs\bf
{		
osdPregain = GetGain();		
osdPregainInterpTargetIdx =		
GetCountOrIndex();		
}		
}		
}		
osdRender:	1	bs\bf
if (sourceAnimated)		
{		
while (attributeAnimated)	1	bs\bf
{		
osdRenderUpdateTargetIdx =		
GetCountOrIndex();		
}		
}		
if (osdIsOmniDirectional == false)	1	bs\bf
{		
osdDirectivityPatternID = GetID();		
}		
if (osdIsPointSource == false)	1	bs\bf
{		
osdExtentID = GetID();		
}		
GetRenderingConditions();		
}		
}		
}		

10

20

30

40

【 0 1 8 9 】

o s d P r e s e n t

オブジェクトソースの有無を示すフラグ。

【 0 1 9 0 】

o s d N r E l e m e n t s

後続するオブジェクトソース要素の数を示す。

【 0 1 9 1 】

s o u r c e A n i m a t e d

そのデータポイントにおいて、対応するソースがアニメーション化されているか否かを示

50

すフラグ。

【0192】

a e d I D

音響要素の I D。

【0193】

a e d P a r e n t I D

シーングラフ内の音響要素の親の I D。

【0194】

o s d S i g n a l I n d e x

ソースに対応する信号の信号入力バッファ内へのインデックスを示す。

10

【0195】

o s d I s C o n t i n u o u s S o u r c e

連続的な信号に関連付けられているか、またはデコーダ側でトリガされる音響効果に関連付けられているかを示すフラグ。

【0196】

o s d R e f e r e n c e D i s t a n c e

オブジェクトソースの基準距離を示す。

20

【0197】

o s d P o s e D e p e n d e n c y

ソースのポーズが何に相対的であるかを示す。

30

【0198】

a t t r i b u t e A n i m a t e d

そのデータポイントにおいて、対応する属性がアニメーション化されているか否かを示すフラグ。また、より多くのウェイポイントについてのデータが存在するか否かを示すためにも使用される。

【0199】

o s d P o s i t i o n

オブジェクトソースの位置座標 (x , y , z) をメートル単位で示す。

同じソースが複数回出現する場合、これは、データポイントの範囲内に複数のウェイポイントがあることを示す。

30

【0200】

o s d P o s i t i o n I n t e r p M e t h o d

位置アニメーションに使用する補間方法を示す (表 1 参照) 。

40

【0201】

o s d P o s i t i o n I n t e r p T a r g e t I d x

TargetOffsetへのインデックス。先行する位置ターゲット値が有効である、データポイントのタイムスタンプからのオフセットを示す。

データポイントの範囲内のウェイポイントごとに 1 つのターゲットインデックスが存在するように、複数のターゲットインデックスが提供されてもよい。

40

【0202】

o s d P o s i t i o n D e l t a

データポイントのタイムスタンプにおけるオブジェクトソース位置の値を再構築することを可能にする位置デルタ値。

PositionAtDPSstart = o s d P o s i t i o n - o s d P o s i t i o n D e l t a

複数のターゲット値が存在する場合、最初のターゲット値を基準とする。

【0203】

o s d O r i e n t a t i o n

オブジェクトソースの向き (ヨー , ピッチ , ロール) をラジアン単位で示す。

同じソースが複数回出現する場合、これは、データポイントの範囲内に複数のウェイポイ

50

ントがあることを示す。

【0204】

`osdOrientationInterpMethod`

向きアニメーションに使用する補間方法を示す（表1参照）。

【0205】

`osdOrientationInterpTargetIdx`

TargetOffsetへのインデックス。先行する向きターゲット値が有効である、データポイントのタイムスタンプからのオフセットを示す。

データポイントの範囲内のウェイポイントごとに1つのターゲットインデックスが存在するように、複数のターゲットインデックスが提供されてもよい。

10

【0206】

`osdOrientationDelta`

データポイントのタイムスタンプにおけるオブジェクトソースの向きの値を再構築することを可能にする向きデルタ値。

`OrientationAtDPSstart = osdOrientation - osdOrientationDelta`

複数のターゲット値が存在する場合、最初のターゲット値を基準とする。

【0207】

`osdPregain`

オブジェクトソースのプリゲイン値（dB）。

20

同じ要素が複数回出現する場合、これは、データポイントの範囲内に複数のウェイポイントがあることを示す。

【0208】

`osdPregainInterpMethod`

プリゲインアニメーションに使用する補間方法を示す（表1参照）。

【0209】

`osdPregainInterpTargetIdx`

TargetOffsetへのインデックス。先行するプリゲインターゲット値が有効である、データポイントのタイムスタンプからのオフセットを示す。

データポイントの範囲内のウェイポイントごとに1つのターゲットインデックスが存在するように、複数のターゲットインデックスが提供されてもよい。

30

【0210】

`osdPregainDelta`

データポイントのタイムスタンプにおけるオブジェクトソースのプリゲイン値を再構築することを可能にするプリゲインデルタ値。

`PregainAtDPSstart = osdPregain - osdPregainDelta`

複数のターゲット値が存在する場合、最初のターゲット値を基準とする。

【0211】

`osdRender`

レンダリングフラグ。偽の場合、オブジェクトソースをレンダリングすべきではないことを示す。

40

【0212】

`osdRenderUpdateTargetIdx`

TargetOffsetへのインデックス。先行する`osdRender`値へとレンダリングフラグが状態を反転する、データポイントのタイムスタンプからのオフセットを示す。

データポイントの範囲内のウェイポイントごとに1つのターゲットインデックスが存在するように、複数のターゲットインデックスが提供されてもよい。各ターゲットインデックスは、フラグの状態のバイナリ反転を示す。

50

【 0 2 1 3 】

o s d I s O m n i D i r e c t i o n a l

ソースが無指向性であるかを示すフラグ。

【 0 2 1 4 】

o s d D i r e c t i v i t y P a t t e r n I D

使用すべき指向性パターンの I D。

【 0 2 1 5 】

S o u n d E f f e c t D a t a ()

構文のこの部分は、音響効果の特性を記述する。これらは、連続的にレンダリングされる音声信号ではなく、ユーザインタラクションなどによってトリガされる音声信号の短いセグメントを表す。

【 0 2 1 6 】

一部の実施形態では、音響環境データは、音響効果、クリップ、および他の音声要素に関連付けられたメタデータを提供することができ、メタデータは、特に、特定のユーザアクションまたはインタラクションに関連付けられた音響効果音声要素のデータを含むことができる。例えば、きしむ音を出すドアが開かれることに対して音響効果要素が設けてられてもよい。このような音響効果は、環境内でドアが開かれることに対応する入力をユーザが提供した場合に、レンダラーがドアの軋む音響効果を抽出してレンダリングできるように、ユーザアクションに関連付けられてもよい。

【 0 2 1 7 】

一部の実施形態では、音声要素は、多数の音響効果要素を含んでもよく、音響環境データは、ユーザによって制御される環境変化を、そのような音響効果要素のうちの 1 つ（または複数）とリンクさせるデータを含んでもよい。ユーザによって制御される環境変化は、例えば、ユーザ入力もしくはユーザインタラクションであってもよく、またはこれらに基づいて決定されてもよい（またはこれらに由来してもよい）。

【 0 2 1 8 】

音響環境データは、例えば、（要素を識別する）ターゲット I D、ターゲット属性、および / または対応するターゲット値を参照することによって、特定のユーザインタラクションのための変更を示すことができる。これらの変化のリストは、トリガ I D に応じて実行されてもよい。

【 0 2 1 9 】

具体例として、音源の属性として S o u n d E f f e c t I D を使用することで音響効果のレンダリング開始を示されてもよく、再生されるべき音響効果の音響効果 I D がターゲット値として含められてもよい。

【 0 2 2 0 】

音響効果データは、例えば、次の構文に従って提供され得る。

10

20

30

40

50

【表 15】

構文	ビット数	タイプ
SoundEffectData(fs)		
{		
if (sedPresent)	1	bs\bf
{		
sedNrElements = GetCountOrIndex();		
for e = 0:sedNrElements - 1		
{		
sedID = GetID();		
sedParentID = GetID();		
sedDuration = GetDuration();		
sedPregain = GetGain();		
}		
}		
}		

【0221】

sedPresent

音響効果データの有無を示すフラグ。

【0222】

sedNrElements

次に定義される音響効果要素の数を示す。

【0223】

sedID

音響効果のID。

【0224】

acdParentID

シーングラフ内の音響効果の親のID。

【0225】

sedDuration

10

20

30

40

50

音響効果の継続時間。

【 0 2 2 6 】

 s e d P r e g a i n

音響効果のプリゲイン値。

【 0 2 2 7 】

一部の実施形態では、音響環境データは、特定の領域に関連付けられたデータを含むことができる。例えば、特性値が、特性値が適用される特定の領域のインジケーションとともに提供されてもよい。例えば、所与の音源に関して、特性（例えば、周波数依存の減衰）は、リスナーが音源と同じ部屋にいるか、またはユーザが別の部屋にいるかに依存し得る。したがって、特性は、値が有効であるためには、リスニング位置がその中に存在しなければならないリンクされた領域のインジケーションを有し得る。リスニング位置が適用ゾーン外である場合、特性を正当に使用することができない。代わりに、例えば、現在のリスニング位置を含む異なる適用範囲を有する別の特性値が含まれてもよい。

10

【 0 2 2 8 】

多くの実施形態では、メタデータは、1つ以上の特性について、複数のリンクされた領域を含むことができ、具体的には、第1の適用領域および第2の適用領域を含むことができる。第1の領域は音声要素／音源の位置のために提供され、第2の領域はリスニング位置のために提供され得る。したがって、特性値は、一方はリスニング位置に関連し、もう一方は音源位置に関連する2つの領域に関連付けられ得る。その場合、デコーダ／レンダラーは、リスニング位置および音源位置が適切な領域内にあるかを評価する。そうでない場合、別の特性値が使用されてもよく、例えば、デフォルト値、リスニング位置および音源位置を含む他の有効性領域に関連付けられた値、またはオリジナルの値の代替として提供され、有効性領域が満たされない場合に使用するために示される代替特性値が使用されてもよい。

20

【 0 2 2 9 】

具体例として、ビットストリームは、リスニング位置および音源／音声要素位置それぞれの有効性領域をそれぞれが表す `applicableRegion` および `internalSourceRegion` で示されるフィールド／データを含むことができる。

【 0 2 3 0 】

一例として、ビットストリームは、リスニング位置と、リスニング位置がデータフィールド `applicableRegion` の値によって示される領域内にあるか否かとに依存する第1の特性値のデータを含むことができる。その場合、この値はさらに、データフィールド `internalSourceRegion` の値によって示される領域内にある音源に直接適用可能であり、データフィールド `internalSourceRegion` によって示される領域外にある場合は別の特性値が適用される。

30

【 0 2 3 1 】

このようなアプローチを使用したビットストリームのデータの例が以下に含まれている。

【 0 2 3 2 】

 A c o u s t i c E n v i r o n m e n t D a t a ()

40

構文のこの部分は、（一般的な）音響環境／周囲の（全体的な）特性を記述する。具体的には残響特性。

【表 16】

構文	ビット数	タイプ
AcousticEnvironmentData(fs)		
{		
if (acdPresent)	1	bslbf
{		
acdNrElements = GetCountOrIndex();		
for e = 0:acdNrElements - 1		
{		
acdID = GetID();		
acdParentID = GetID();		
applicableRegionID = GetID();		
internalSourceRegionID = GetID();		
freqGridIdx = GetCountOrIndex();		
dsrOffset = GetDuration(fs);		
for b = 0:fgdNrBands[freqGridIdx]		
{		
T60[e][b] = GetDuration(fs);		
dsr[e][b] = LUT(dsrCode);	var	vlclbf
}		
}		
}		
}		

10

20

30

40

【0 2 3 3】

a c d P r e s e n t

音響環境データの有無を示すフラグ。

【0 2 3 4】

a c d N r E l e m e n t s

定義されている音響環境の数を示す。

【0 2 3 5】

a c d I D

音響環境定義の I D。

50

【 0 2 3 6 】

a c d P a r e n t I D

シーングラフ内の音響環境定義の親の I D。

【 0 2 3 7 】

a p p l i c a b l e R e g i o n I D

パラメータが適用される領域を記述する幾何学的要素の I D。ポーズ依存性がグローバルでない限り、位置および向きは音響環境の親によってオフセットされる。

【 0 2 3 8 】

i n t e r n a l S o u r c e R e g i o n I D

全ての包含されているソースの全てのエネルギーが残響に寄与する領域を記述する幾何学的要素の I D。ポーズ依存性がグローバルでない限り、位置および向きは音響環境の親によってオフセットされる。 10

【 0 2 3 9 】

f r e q G r i d I d x

F r e q G r i d D a t a () 内に定義されている周波数グリッドのリストのインデックス。

【 0 2 4 0 】

d s r O f f s e t

R I R において D S R が計算される場所からのオフセット(秒)。オフセット = 0 はソースで放たれることに等しい。したがって、オフセットはそれより高いはずである。 20

【 0 2 4 1 】

T 6 0

初期減衰後の、E D C の直線部分の 0 ~ - 3 0 d B ポイントから計算される T 6 0 時間。

【 0 2 4 2 】

D S R

拡散エネルギーとソースエネルギーの比率 (D i f f u s e t o S o u r c e e n e r g y R a t i o)。拡散残響エネルギーは、1人のユーザのサンプルポイントのために、R I R ラグ d s r O f f s e t 後に計算される。ソースエネルギーは、その拡散エネルギーを引き起こす放出されたソースエネルギーの合計である。 30

【 0 2 4 3 】

d s r C o d e

D S R 値を示すコード。

【表 17】

コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値
10001100	-150	011100101	-121	011000110	-92	110001	-63	010010	-34
10001101	-149	011111010	-120	011000111	-91	110110	-62	010011	-33
10001110	-148	011111011	-119	011000100	-90	110111	-61	010000	-32
10001111	-147	011111000	-118	011000101	-89	110100	-60	010001	-31
100011100	-146	011111001	-117	011011010	-88	110101	-59	010110	-30
100011101	-145	011111110	-116	011011011	-87	001010	-58	011010010	-29
10000010	-144	011111111	-115	011011000	-86	001011	-57	011010011	-28
10000011	-143	011111100	-114	011011001	-85	001000	-56	011010000	-27
10000000	-142	011111101	-113	011011110	-84	001001	-55	011010001	-26
10000001	-141	0111110010	-112	011011111	-83	001110	-54	011010110	-25
10000110	-140	0111110011	-111	011011100	-82	001111	-53	011010111	-24
10000111	-139	0111110000	-110	011011101	-81	001100	-52	011010100	-23
10000100	-138	0111110001	-109	010100	-80	001101	-51	011010101	-22
10000101	-137	0111110110	-108	010101	-79	000010	-50	010111010	-21
011101010	-136	0111110111	-107	100110	-78	000011	-49	010111011	-20
011101011	-135	0111110100	-106	100111	-77	000000	-48	010111000	-19
011101000	-134	0111110101	-105	100100	-76	000001	-47	010111001	-18
011101001	-133	011001010	-104	100101	-75	000110	-46	010111110	-17
011101110	-132	011001011	-103	111010	-74	000111	-45	010111111	-16
011101111	-131	011001000	-102	111011	-73	000100	-44	010111100	-15
011101100	-130	011001001	-101	111000	-72	000101	-43	010111101	-14
011101101	-129	011001110	-100	111001	-71	101010	-42	10001010	-13
011100010	-128	011001111	-99	111110	-70	101011	-41	10001011	-12
011100011	-127	011001100	-98	111111	-69	101000	-40	10001000	-11
011100000	-126	011001101	-97	111100	-68	101001	-39	10001001	-10
011100001	-125	011000010	-96	111101	-67	101110	-38		

コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値
011100110	-124	011000011	-95	110010	-66	101111	-37		
011100111	-123	011000000	-94	110011	-65	101100	-36		
011100100	-122	011000001	-93	110000	-64	101101	-35		

10

20

30

40

50

【 0 2 4 4 】
G e o m e t r i c E l e m e n t D a t a ()

構文のこの部分は、幾何学的要素を記述する。

【表 18】

構文	ビット数	タイプ
GeometricElementData(isSmallScene)		
{		
if (gedPresent)	1	bs bf
{		
gedNrElements = GetCountOrIndex();		
for e = 0:gedNrElements - 1		
{		
gedID = GetID();		
gedType;	3	uimsbf
if (gedType == 0)		
{		
cornerPos1 = GetPosition(isSmallScene);		
cornerPos2 = GetPosition(isSmallScene);		
}		
elseif (gedType == 1)		
{		
boxParentID = GetID();	3	uimsbf
boxPoseDependency;		
boxPosition = GetPosition(isSmallScene);		
boxOrientation = GetOrientation();		
boxXDim = GetDistance(isSmallScene);		
boxYDim = GetDistance(isSmallScene);		
boxZDim = GetDistance(isSmallScene);		
}		
}		
}		
}		

10

20

30

40

【 0 2 4 5 】

g e d P r e s e n t

音響環境データの有無を示すフラグ。

【 0 2 4 6 】

g e d N r E l e m e n t s

定義された幾何学的要素の数を示す。

50

【0247】

g e d I D

幾何学的要素の I D。

【0248】

g e d T y p e

次に伝達される幾何学的要素のタイプを定義する。

【表19】

値	意味
0	単純な境界ボックス
1	ボックス
2	球
3	円柱
4	メッシュ
5	予約済み
6	予約済み
7	予約済み

10

20

30

40

50

【0249】

c o r n e r P o s 1

座表軸に平行なボックスの1つの角を含む。位置はグローバル座標を表す。

【0250】

c o r n e r P o s 2

c o r n e r P o s 1 の対角である、座標軸に平行なボックスの第2の角を含む。位置はグローバル座標を表す。

【0251】

b o x P a r e n t I D

対応するボックス幾何学的要素の親ノードの I D。

【0252】

b o x P o s e D e p e n d e n c y

音響要素のポーズが何に相対的であるかを示す。

【0253】

b o x P o s i t i o n

ボックスジオメトリの位置座標 (x, y, z) をメートル単位で示す。

【0254】

b o x O r i e n t a t i o n

ボックスジオメトリの向き (ヨー, ピッチ, ロール) をラジアン単位で示す。

【0255】

b o x X D i m

回転前の x 軸に沿ったボックスジオメトリの寸法。

【0256】

b o x Y D i m

回転前の y 軸に沿ったボックスジオメトリの寸法。

【0257】

b o x Z D i m

回転前の z 軸に沿ったボックスジオメトリの寸法。

【 0 2 5 8 】**U s e r I n t e r a c t i o n D a t a ()**

構文のこの部分は、ユーザインタラクションデータを記述する。どの要素について、どの属性をどの値に変更する必要があるか、またはこの値が外部エンティティから提供されるか（例えば、ソースを拾ってシーン内で移動させるなど、完全にユーザによって制御されるインタラクションの場合）を記述することによって、ユーザインタラクションをどのようにしてトリガできるか、およびトリガに応じてどのような変更を加えるべきかを記述する。

10

【 0 2 5 9 】

視覚、音声、触覚などの 6 D o F メディアレンダリングの全ての側面をカバーする、ユーザインタラクションのよりセマンティックな定義が、より高いビットストリームレベルで定義され得る。ユーザインタラクションは、これらの側面の全てまたは一部に影響を与える可能性がある。このシステムレイヤー（M P E G - I の場合、M P E G S y s t e m s ワーキンググループWG 0 3 によってカバーされる）は、ユーザがどのようにして特定のインタラクションをトリガできるかを定義する。これは、シーンの特定の空間的領域内でコントローラのトリガボタンを作動することを定義するという意味であってもよいし、または、他のレイヤーが、抽象的な意味をハードウェア依存制御にリンクさせるというより抽象的な意味であってもよい。例えば、ユーザがドア 5 を開けた場合、ユーザインタラクション G を作動する。

20

【 0 2 6 0 】

システムレベルでのそのようなユーザインタラクショントリガは、専用のユーザインタラクショントリガをそれぞれのメディアレンダーに送信する可能性がある。例えば、システムレベルのユーザインタラクション G は音声ユーザインタラクション 1 2 にリンクされ得、音声レンダラーはその後、t r i g g e r I D = 1 2 を有するユーザインタラクションに関連付けられた変更を実行し得る。

30

【 0 2 6 1 】

多くの実施形態では、そのようなユーザインタラクショントリガは、より没入型のユーザインタラクションのためのさらなるパラメータを伴うことができる。例えば、ユーザによって拾い上げられ、動かされている音源の位置座標が提供されてもよい。

【 0 2 6 2 】

音声レンダラーの外部からトリガされるこのようなユーザインタラクションは、外部トリガとも呼ばれる可能性がある。他のユーザインタラクションは、音声レンダラー自体によってトリガされることに依拠してもよい。このようなトリガ間の区別は、ユーザインタラクションの変化を記述するビットストリーム内の t r i g g e r T y p e 特性によって示されてもよい。

40

50

【表 2 0】

構文	ビット数	タイプ
UserInteractionData(isSmallScene, fs)		
{		
if (uidPresent)	1	bs\bf
{		
uidNrElements = GetCountOrIndex();		
for e = 0:uidNrElements - 1		
{		
triggerType;	3	uimsbf
if (triggerType < 2)		
{		
triggerIdx = GetCountOrIndex();		
}		
if (triggerType == 2)		
{		
triggerID = GetID();		
condTransition;	1	bs\bf
updateDelay = GetDuration(fs);		
}		
nrChanges = GetCountOrIndex();		
for u = 0:nrChanges - 1		
{		
if (u > 0)		
{		
usePreviousID;	1	bs\bf
}		
else		

10

20

30

40

50

{			
usePreviousID = 0;			
}			
if (!usePreviousID)			
{			
previousTargetID = targetID;			
targetID = GetID();			
}			
else			
{			
targetID = previousTargetID;			
}			
if (!immediateChange)	1	bs\bf	10
{			
duration = GetDuration(fs);			
}			
else			
{			
duration = 0;			
}			
targetAttribute = LUT(attribCode);	var	vl\cl\bf	20
if (triggerType == 1)			
{			
externalParameter;	1	bs\bf	30
}			
else			
{			
externalParameter = 0;			
}			40

if (externalParameter)		
{		
parameterIdx = LUT(paramIdxCode) ;	var	vlclbf
while (moreParameters)	1	bslbf
{		
parameterIdx = LUT(paramIdxCode) ;	var	vlclbf
}		
}		
else		
{		
switch (targetAttribute)		
{		
case Position:		
targetValue =		20
GetPosition(isSmallScene) ;		
case Orientation:		
targetValue = GetOrientation() ;		
case Pregain:		
targetValue = GetGain() ;		
case Render:		
targetValue = render ;	1	bslbf
case SignalIndex:		
targetValue = GetCountOrIndex() ;		
case SoundEffectID:		
targetValue = GetID() ;		
case :		
targetValue = <TBD. >		
}		
}		40

}		
}		
}		
}		

10

20

30

40

50

【0263】

u i d P r e s e n t

ユーザインタラクションの有無を示すフラグ。

【0264】

t r i g g e r T y p e

ユーザインタラクションがどのようにシーン更新をトリガするかを定義する。

【表21】

値	意味
0	外部（パラメータなし）—外部から提供されたパラメータを使用せずに、外部プロセスによってシーン更新がトリガされる。
1	外部（パラメータあり）—外部から提供されたパラメータを使用して、外部プロセスによってシーン更新がトリガされる。
2	P r o x i m i t y C o n d i t i o n —ユーザの近さの状態を確認する内部プロセスによってトリガされる更新。
3	予約済み
4	予約済み
5	予約済み
6	予約済み
7	予約済み

10

20

30

40

50

【0265】

t r i g g e r I d x

この特定のシーンの更新をトリガするために外部ソースからのユーザインタラクションメッセージ内で使用されるインデックス。

【0266】

t r i g g e r I D

この特定のシーン更新を記述する状態を記述する状態要素の I D。

【0267】

c o n d T r a n s i t i o n

状態が指定された遷移値に変化すると、更新がトリガされる。

【0268】

u p d a t e D e l a y

トリガ後に更新をどれだけ遅らせるべきか。

【0269】

n r C h a n g e s

このシーン更新の変化の数を示す。

【0270】

u s e P r e v i o u s I D

変化が前の変化と同じ I D についてのものであるか否かを示すフラグ。

【0271】

i m m e d i a t e C h a n g e

変化が即時であるか、または特定の期間にわたってターゲット値まで補間されるかを示すフラグ。

【0272】

duration

変化の補間期間。

【0273】

targetAttribute

変更する属性の名称。

【0274】

attributeCode

変更可能な属性を示すコード。

【表22】

コード ワード	値
01	位置
11	向き
10	プリゲイン
001	レンダリング
0001	SignalIndex
0000	SoundEffectID

10

20

30

40

50

【0275】

externalParameter

パラメータ値が外部プロセスによって提供されるか否かを示すフラグ。

【0276】

parameterIdx

この属性にマッピングされる外部プロセスからのメッセージ内のパラメータのインデックス。

【0277】

paramIdxCode

パラメータインデックスを示すコード。

【表 2 3】

コードワード	値
000	0
111	1
101	2
100	3
0111	4
0101	5
0100	6
0011	7
1101	8
1100	9
01101	10
00101	11
00100	12
011001	13
0110001	14
0110000	15

10

20

30

40

50

【0 2 7 8】

more Parameters

追加パラメータの有無を示すフラグ。

【0 2 7 9】

targetValue

変化のターゲット値。ターゲット属性のタイプに応じて、構文内の switch 文に従い、値のコード化が異なり得る。

【0 2 8 0】

サポート要素

ビットストリームは、特性値を表すデータの提供をサポートできる様々なサポート要素を含み得る。

【0 2 8 1】

また、多くの実施形態では、特性の値を示すために使用されるビット数は、ビットストリーム内で可変であってもよい。特に、特性値は、所定の数のビットを含むフィールドによって示され得る。ただし、特性値のための追加ビットを提供する1つ以上の拡張フィールドが含まれていることを示すフラグインジケータが含められてもよい。拡張フィールドは、具体的には、特性値の範囲を拡張するための追加ビットを提供してもよい。特に、拡

張フィールドは、（動的）範囲を有する特性値を生成するために、デフォルトフィールドのデータビットと組み合わせられる最上位（／上位）ビットをさらに含み得る（具体的には、より高い値を表現することを可能にする追加ビットを提供し得る）。

【0282】

他のシナリオでは、拡張フィールドは、具体的には、特性値の精度を拡張するための追加ビットを提供し得る。特に、拡張フィールドは、より高精度の特性値を生成するために、デフォルトフィールドのデータビットと組み合わせられる追加の最下位（／下位）ビットをさらに含み得る。

【0283】

一部の実施形態では、音響環境データは、所与の特性の値を表す第1のビットを提供する第1のデータフィールドと、第1のデータフィールド／所与の特性のためのインジケーションを提供し得る第2のデータフィールドとを含み得る。このインジケーションは、第1の特性の値を表すさらなるビットを提供するさらなる拡張データフィールドが含まれるか否かのインジケーションを含み得る。このインジケーションは、例えば、提供されるデータ値の範囲を拡張するためのビットを拡張フィールドが含むことのインジケーション、および／または提供されるデータ値の精度／分解能を高めるためのビットを含むことのインジケーションであり得る。具体的には、このインジケーションは、第1のデフォルトフィールドおよび拡張フィールドの両方からのビットの組み合わせから得られる、より大きなデータワードによって値が表されることを示し得る。

【0284】

一部の実施形態では、さらなるビットは、特性が取り得る値の範囲を拡張する。さらなるビットは、特性の値を表す、データワードの第1のビットよりも上位のビットであってもよい。一部の実施形態では、さらなるビットは、特性が取り得る値の分解能を増加させる。さらなるビットは、第1の特性の値を表す、データワードの第1のビットよりも下位のビットであってもよい。さらなるビットは、第1の特性が取り得る値の精度を拡張し得る。

【0285】

多くの実施形態では、デフォルトビットおよびさらなるビットは、必ずしもビットを連結することによって組み合わせられるわけではなく、他の方法で組み合わせられる値を各々が表してもよい。

【0286】

したがって、一部の実施形態では、音響環境データは、特性の値を決定するためにさらなるビットが使用されることを示すインジケーション／フラグを含むことができる。このアプローチは、例えば、空間的特性、量、ゲイン特性、容量特性、周波数特性、インデックス特性、および識別特性などの多様な特性値について使用できる。

【0287】

例えば、より広い範囲が提供されていることを示すために以下のフィールドが使用されてもよい。

- 時間の尺度の場合 : a d d S e c o n d s
- 空間の尺度の場合 : a d d H e c t o m e t e r s
- 量の尺度の場合 : i s L a r g e r N u m b e r
- (整数) ID番号の場合 : l a r g e r V a l u e

【0288】

より正確な（より分解能が高い）値が提供されていることを示すために、以下のフィールドが使用されてもよい。

- 時間の尺度の場合 : a d d M i l l i s e c o n d s
- 空間の尺度の場合 : a d d C e n t i m e t e r s
- 周波数の尺度の場合 : m o r e A c c u r a c y
- 角度の尺度の場合 : a d d F i n e A n g l e
- ゲインの尺度の場合 : a d d F i n e G a i n .

10

20

30

40

50

【0289】

ビットストリームのための様々なサポート要素を以下に示す。上記したように、サポート要素は可変範囲 / 分解能値を使用する可能性がある。

【0290】

GetID()

整数のIDを返す。

【表24】

構文	ビット数	タイプ
GetID()		
{		
id = idVal;	7	uimsbf
while (largerValue)	1	bslbf
{		
idVal;	7	uimsbf
id = id * 128 + idVal;		
}		
}		

10

20

30

40

50

【0291】

idVal

ID値または部分的なID値。

【0292】

largerValue

ID値がより大きいかを示すフラグ。

【0293】

GetCountOrIndex()

[0 . . 1023] の範囲内の数値を返す。

【表 2 5】

構文	ビット数	タイプ
number = GetCountOrIndex()		
{		
number = LUT(countOrIndexLoCode);	var	v clbf
if (isLargerNumber)	1	bs bf
{		
numberHi = LUT(countOrIndexHiCode);	var	v clbf
number = number + numberHi * 64;		
}		
return number;		
}		

10

20

30

40

50

【0 2 9 4】

c o u n t O r I n d e x L o C o d e

カウントまたはインデックス値の下位ビットを示すコード。

【表 2 6】

コード ワード	値								
0111	0	001010	13	111101	26	1101000	39	1011011	52
100	1	001001	14	111100	27	1100111	40	1011010	53
01100	2	001000	15	111011	28	1100110	41	1011001	54
01101	3	000111	16	111010	29	1100101	42	1011000	55
01010	4	000110	17	111001	30	1100100	43	1010111	56
01011	5	000101	18	111000	31	1100011	44	1010110	57
01000	6	000100	19	1101111	32	1100010	45	1010101	58
01001	7	000011	20	1101110	33	1100001	46	1010100	59
001111	8	000010	21	1101101	34	1100000	47	1010011	60
001110	9	000001	22	1101100	35	1011111	48	1010010	61
001101	10	000000	23	1101011	36	1011110	49	1010001	62
001100	11	111111	24	1101010	37	1011101	50	1010000	63
001011	12	111110	25	1101001	38	1011100	51		

10

20

30

【0 2 9 5】

i s L a r g e r N u m b e r

より大きな数値を示すためにさらに多くのビットが送信されるか否かを示すフラグ。

【0 2 9 6】

c o u n t O r I n d e x H i C o d e

カウントまたはインデックス値の上位ビットを示すコード。

40

50

【表 2 7】

コードワード	値	
001	1	
000	2	
110	3	
101	4	10
100	5	
0111	6	
0101	7	
1111	8	
1110	9	
01101	10	20
01001	11	
01000	12	
011001	13	
0110001	14	
0110000	15	30

【0 2 9 7】

GetDuration()

時間的継続時間をサンプルで返す。

【表 2 8】

構文	ビット数	タイプ
duration = GetDuration(fs)		
{		
deciSeconds = LUT(deciSecondsCode);	var	vlclbf
duration = deciSeconds * fs;		
if (addMilliseconds)	1	bslbf
{		
milliSeconds = LUT(milliSecondsCode);	var	vlclbf
duration = duration + milliSeconds * fs;		
if (addSamples)	1	bslbf
{		
samples = LUT(samplesCode);	var	vlclbf
duration = duration + samples;		
}		
}		
if (addSeconds)	1	bslbf
{		
seconds = LUT(secondsCode);	var	vlclbf
duration = duration + seconds * fs;		
}		
duration = round(duration);		
return duration;		
}		

10

20

30

40

【0 2 9 8】

LUT()

値が引数として指定されたフィールドに対応して、ルックアップテーブルへのクエリを実行する。

【0 2 9 9】

deciSecondsCode

デシ秒継続時間オフセットを示すコード。

50

【表 2 9】

コードワード	値
110	0
100	0. 1
101	0. 2
0110	0. 3
0111	0. 4
111	0. 5
0100	0. 6
0101	0. 7
0010	0. 8
0011	0. 9
000	1

10

20

30

40

50

【0 3 0 0】

add Milli seconds

次に、ミリ秒継続時間オフセットが送信されるか否かを示すフラグ。

【0 3 0 1】

milli Seconds Code

ミリ秒継続時間オフセットを示すコード。

【表30】

コード ワード	値								
1111010	0	10010	0.02	10000	0.04	10110	0.06	10100	0.08
1111011	0.001	0101001	0.021	0111111	0.041	0001101	0.061	0010011	0.081
1111000	0.002	0101110	0.022	0111100	0.042	0000010	0.062	0010000	0.082
1111001	0.003	0101111	0.023	0111101	0.043	0000011	0.063	0010001	0.083
1111110	0.004	0101100	0.024	0110010	0.044	0000000	0.064	0010110	0.084
1111111	0.005	0101101	0.025	0110011	0.045	0000001	0.065	0010111	0.085
1111100	0.006	0100010	0.026	0110000	0.046	0000110	0.066	0010100	0.086
1111101	0.007	0100011	0.027	0110001	0.047	0000111	0.067	0010101	0.087
1110010	0.008	0100000	0.028	0110110	0.048	0000100	0.068	1101010	0.088
1110011	0.009	0100001	0.029	0110111	0.049	0000101	0.069	1101011	0.089
11001	0.01	10011	0.03	10001	0.05	10111	0.07	10101	0.09
1110000	0.011	0100110	0.031	0110100	0.051	0011010	0.071	1101000	0.091
1110001	0.012	0100111	0.032	0110101	0.052	0011011	0.072	1101001	0.092
1110110	0.013	0100100	0.033	0001010	0.053	0011000	0.073	1101110	0.093
1110111	0.014	0100101	0.034	0001011	0.054	0011001	0.074	1101111	0.094
1110100	0.015	0111010	0.035	0001000	0.055	0011110	0.075	1101100	0.095
1110101	0.016	0111011	0.036	0001001	0.056	0011111	0.076	1101101	0.096
0101010	0.017	0111000	0.037	0001110	0.057	0011100	0.077	1100010	0.097
0101011	0.018	0111001	0.038	0001111	0.058	0011101	0.078	1100011	0.098
0101000	0.019	0111110	0.039	0001100	0.059	0010010	0.079	110000	0.099

【0302】

addSamples

次に、サンプルベースの継続時間オフセットが送信されるか否かを示すフラグ。

【0303】

samplesCode

サンプル数継続時間オフセットを示すコード。

10

20

30

40

50

【表31】

コード ワード	値								
11000	1	001111	20	010010	39	110110000	58	111000111	77
11001	2	001100	21	010011	40	110110001	59	111000100	78
111110	3	001101	22	010000	41	110110110	60	111000101	79
111111	4	000010	23	010001	42	110110111	61	111011010	80
111100	5	000011	24	010110	43	110110100	62	111011011	81
111101	6	000000	25	010111	44	110110101	63	111011000	82
10010	7	000001	26	010100	45	111001010	64	111011001	83
10011	8	000110	27	010101	46	111001011	65	111011110	84
10000	9	000111	28	11010	47	111001000	66	111011111	85
10001	10	000100	29	110111010	48	111001001	67	111011100	86
10110	11	000101	30	110111011	49	111001110	68	111011101	87
10111	12	011010	31	110111000	50	111001111	69	111010010	88
10100	13	011011	32	110111001	51	111001100	70	111010011	89
10101	14	011000	33	110111110	52	111001101	71	111010000	90
001010	15	011001	34	110111111	53	111000010	72	111010001	91
001011	16	011110	35	110111100	54	111000011	73	111010110	92
001000	17	011111	36	110111101	55	111000000	74	111010111	93
001001	18	011100	37	110110010	56	111000001	75	111010100	94
001110	19	011101	38	110110011	57	111000110	76	111010101	95

10

20

30

40

【0304】

addSeconds

次に、秒継続時間オフセットが送信されるか否かを示すフラグ。

【0305】

secondsCode

秒継続時間オフセットを示すコード。

50

【表3-2】

コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値
0011	1	1011	7	01011	13	10101	19	101001	25
0001	2	1001	8	01001	14	011111	20	0101001	26
0000	3	1000	9	01000	15	011110	21	0101000	27
1111	4	01110	10	00101	16	010101	22	1010001	28
1101	5	01101	11	11101	17	001001	23	10100001	29
1100	6	01100	12	11100	18	001000	24	10100000	30

【0306】

```
GetFrequency()
```

[16 . . . 49717] の範囲内の周波数 (Hz) を返す。

【表3-3】

構文	ビット数	タイプ
GetFrequency()		
{		
frequency = LUT(frequencyCode);	var	v\cl\bf
if (moreAccuracy)	1	bs\lbf
{		
frequency = frequency * 2^((frequencyRefine + 1) / 51);	4	uimsbf
}		
}		

【0307】

```
LUT()
```

値が引数として指定されたフィールドに対応して、ルックアップテーブルへのクエリを実行する。

【0308】

```
frequencyCode
```

1 / 3オクターブバンドの中心周波数 (Hz) を示すコード。

10

20

30

40

50

【表34】

コードワード	値	コードワード	値	コードワード	値	コードワード	値	コードワード	値
100011	16	011010	80	011111	400	1101	2000	010100	10000
001110	20	011011	100	1111	500	010000	2500	010101	12500
001111	25	0001	125	011100	630	010001	3150	0010	16000
1001	31.5	011000	160	011101	800	1010	4000	10000	20000
001100	40	011001	200	1100	1000	010110	5000	10001010	25000
001101	50	1110	250	010010	1250	010111	6300	10001011	31500
0000	63	011110	315	010011	1600	1011	8000	1000100	40000

【0309】

more Accuracy

より正確な周波数のためのデータが送信されるか否かを示すフラグ。

【0310】

frequency Refine

周波数値を調整するための値を示すフィールド。

【0311】

GetPosition()

位置 [x, y, z] をメートル単位で返す。

10

20

30

40

50

【表 3 5】

構文	ビット数	タイプ
position = GetPosition(isSmallScene)		
{		
for c = 0:2		
{		
coord = GetDistance(isSmallScene);		
isNegative;	1	bs lbf
if (isNegative)		
{		
position[c] = -1 * coord;		
} else		
{		
position[c] = coord;		
}		
}		
return position;		
}		

10

20

30

40

50

【0 3 1 2】

GetPositionDelta()

位置 [dx, dy, dz] をメートル単位で返す。

【表 3 6】

構文	ビット数	タイプ
positionDelta = GetPositionDelta(isSmallScene)		
{		
positionDelta = GetPosition(isSmallScene);		
< >		
return positionDelta;		
}		

10

20

30

40

50

【0 3 1 3】

GetDistance()

距離をメートル単位で返す。

【表 3 7】

構文	ビット数	タイプ
distance = GetDistance(isSmallScene)		
{		
meters = LUT(metersCode);	var	vlclbf
distance = meters;		
if (isSmallScene == false)		
{		
if (addHectometers)	1	bslbf
{		
hectometers = LUT(hectometersCode);	var	vlclbf
distance = distance + 100 * hectometers;		
while (addKilometers)	1	bslbf
{		
kilometers = LUT(kilometersCode);	var	vlclbf
distance = distance + kilometers * 1000;		
}		
}		
}		
if (addCentimeters)	1	bslbf
{		
centimeters = LUT(centimetersCode);	var	vlclbf
distance = distance + centimeters / 100;		
if (addMillimeters)	1	bslbf
{		
millimeters = LUT(millimetersCode);	var	vlclbf
distance = distance + millimeters / 1000;		
}		
}		
return distance;		
}		

10

20

30

40

【0 3 1 4】

LUT()

値が引数として指定されたフィールドに対応して、ルックアップテーブルへのクエリを実行する。

【0 3 1 5】

m e t e r s C o d e

メートル座標オフセットを示すコード。

50

【表38】

コードワード	値	コードワード	値	コードワード	値	コードワード	値	コードワード	値
111101	0	000001	20	010101	40	10011110	60	11100010	80
110010	1	000110	21	101010	41	10011111	61	11100011	81
110011	2	000111	22	101011	42	10011100	62	11100000	82
110000	3	000100	23	101000	43	10011101	63	11100001	83
110001	4	000101	24	101001	44	10010010	64	11100110	84
110110	5	011010	25	101110	45	10010011	65	11100111	85
110111	6	011011	26	101111	46	10010000	66	11100100	86
110100	7	011000	27	101100	47	10010001	67	11100101	87
110101	8	011001	28	101101	48	10010110	68	11111010	88
001010	9	011110	29	100000	49	10010111	69	11111011	89
001011	10	011111	30	1000100	50	10010100	70	11111000	90
001000	11	011100	31	1000101	51	10010101	71	11111001	91
001001	12	011101	32	10001110	52	11101010	72	11111110	92
001110	13	010010	33	10001111	53	11101011	73	11111111	93
001111	14	010011	34	10001100	54	11101000	74	11111100	94
001100	15	010000	35	10001101	55	11101001	75	11111101	95
001101	16	010001	36	10011010	56	11101110	76	11110010	96
000010	17	010110	37	10011011	57	11101111	77	11110011	97
000011	18	010111	38	10011000	58	11101100	78	11110000	98
000000	19	010100	39	10011001	59	11101101	79	11110001	99

【0316】

add Hectometers

次に、ヘクトメートル座標オフセットが送信されるか否かを示すフラグ。

【0317】

hectometers Code

ヘクトメートル座標オフセットを示すコード。

10

20

30

40

50

【表 3 9】

コードワード	値
010	1
011	2
000	3
001	4
110	5
111	6
1010	7
1011	8
100	9

10

20

【0 3 1 8】

a d d K i l o m e t e r s

次に、キロメートル座標オフセットが送信されるか否かを示すフラグ。

【0 3 1 9】

k i l o m e t e r s C o d e

キロメートル座標オフセットを示すコード。10 kmを超える距離の場合、複数回の出現が提供される可能性がある。

【表 4 0】

コードワード	値
10	1
011	2
001	3
000	4
111	5
0101	6
0100	7
1101	8
11001	9
11000	10

30

40

【0 3 2 0】

50

add Centimeters

次に、センチメートル座標オフセットが送信されるか否かを示すフラグ。

【0321】

centimeters Code

センチメートル座標オフセットを示すコード。

【表41】

コード ワード	値								
110010	0	0100110	20	0001010	40	0011110	60	100010	80
110011	1	0100111	21	0001011	41	0011111	61	100011	81
110000	2	0100100	22	0001000	42	0011100	62	100000	82
110001	3	0100101	23	0001001	43	0011101	63	100001	83
110110	4	0111010	24	0001110	44	0010010	64	100110	84
110111	5	0111011	25	0001111	45	0010011	65	100111	85
110100	6	0111000	26	0001100	46	0010000	66	100100	86
110101	7	0111001	27	0001101	47	0010001	67	100101	87
0101010	8	0111110	28	0000010	48	0010110	68	1111010	88
0101011	9	0111111	29	0000011	49	0010111	69	1111011	89
0101000	10	0111100	30	0000000	50	0010100	70	1111000	90
0101001	11	0111101	31	0000001	51	0010101	71	1111001	91
0101110	12	0110010	32	0000110	52	101010	72	1111110	92
0101111	13	0110011	33	0000111	53	101011	73	1111111	93
0101100	14	0110000	34	0000100	54	101000	74	1111100	94
0101101	15	0110001	35	0000101	55	101001	75	1111101	95
0100010	16	0110110	36	0011010	56	101110	76	111010	96
0100011	17	0110111	37	0011011	57	101111	77	111011	97
0100000	18	0110100	38	0011000	58	101100	78	111000	98
0100001	19	0110101	39	0011001	59	101101	79	111001	99

【0322】

add Millimeters

次に、ミリメートル座標オフセットが送信されるか否かを示すフラグ。

【0323】

10

20

30

40

50

millimeters Code
ミリメートル座標オフセットを示すコード。

【表 4 2】

コードワード	値
000	0
001	1
0110	2
0111	3
0100	4
0101	5
110	6
111	7
100	8
101	9

10

20

30

40

50

【0 3 2 4】

向き特性の表現

音響環境データは多くの場合において、実際にはほとんど場合において、1つ以上の向き特性の値を含み得る。

【0 3 2 5】

一部の実施形態では、これは、向き値の表現を含む音響環境データによって有利に達成され得る。

【0 3 2 6】

一部の実施形態では、音響環境データは、向き特性を表すための向き表現フォーマットを記述するデータグループ / データフィールドを含む。多くの実施形態では、音響環境データは、複数の異なる向き表現フォーマットを記述するまたは定義するデータを(同じまたは複数の異なるデータセット / ポイント内に)含むことができる。各向き表現フォーマットは、向き値を表現するための(データ / ビット)フォーマットを提供する可能性がある。その場合、複数のデータセット / ポイントは、定義された向き表現フォーマットのうちの1つを使用することによって向き特性を記述するデータを含む可能性がある。

【0 3 2 7】

一部の実施形態では、音響環境データは、ビットストリームが、向き表現フォーマットを記述するデータグループ / データフィールドを含むか否かを示すインジケータを含むことができる。例えば、向き表現フォーマットを記述するデータフィールドが含まれていることを示すフラグを示すことができる。

【0 3 2 8】

また、音響環境データ内に定義されている向き表現フォーマットに従って向き値が提供されているか否かを示すフラグ / インジケータが、個別の向き値について提供されてもよい。フラグ / インジケータは、例えば複数の向き表現フォーマットのうちのいずれがその特定の値に使用されるかを示すために、個別の値について含まれてもよい。

【0 3 2 9】

向き表現フォーマットは、例えば次のうちの1つ以上を含み得る。

【0330】

所定のデフォルト向き表現のインジケーション。デフォルト向き表現の数は事前に決定されていてもよい（例えば、規格の定義によって）。このようなデフォルト向きを参照するデータが含まれていてもよい。例えば、フィールド orientationCode がデフォルト向きを示してもよい。

【0331】

所定の角度のセット。向き表現フォーマットは所定の角度のセットを定義してもよく、向き値は、例えば、そのような所定の角度のうちの1つを単純に参照することによって示され得る。例えば、各所定の角度がインデックスによって表され、適切なインデックスを示すことによって所与の向き値が表されてもよい。例えば、向き値の各角度は、狭い範囲の角度からのデフォルト角度によって示されてもよい。

10

【0332】

（例えば、量子化グリッド上の）角度のセット。角度の表現は、明示的な角度値によって行われてもよい。角度値は、所与のワード長によって表されてもよく、したがって所与の量子化レベルを有することができる。したがって、各向きの角度は、より大きな範囲の角度のうちの1つによって示される。

20

【0333】

上記の例に従って向き値を提供するアプローチのいくつかの例を以下に提供する。

【0334】

GetOrientation()

向き [ヨー, ピッチ, ロール] をラジアン単位で返す。

30

40

50

【表 4 3】

構文	ビット数	タイプ
orientation = GetOrientation()		
{		
orientation = LUT(orientationCode);	var	vlclbf
if (orientationCode == 'Default angles')		
{		
orientation[0] = LUT(defaultYawCode);	var	vlclbf
orientation[1] = LUT(defaultPitchCode);	var	vlclbf
orientation[2] = LUT(defaultRollCode);	var	vlclbf
} elseif (orientationCode == 'Explicit angles')		
{		
for (o = 0:2)		
{		
coarseAngle = LUT(coarseAngleCode);	var	vlclbf
angle = coarseAngle;		
if (addFineAngle)	1	bslbf
{		
fineAngle = LUT(fineAngleCode);	var	vlclbf
angle = angle + fineAngle;		
}		
orientation[o] = angle;		
}		
}		
return orientation;		
}		

【0 3 3 5】

L U T ()

値が引数として指定されたフィールドに対応して、ルックアップテーブルへのクエリを実行する。

【0 3 3 6】

o r i e n t a t i o n C o d e

デフォルト向きか、またはさらなるデータによって向きが定義される 2 つのエスケープ値のいずれかを示す向きのコード。

10

20

30

40

50

【表 4 4】

コードワード	値	記述
01	[0, 0, 0]	デフォルト向き：（ヨー, ピッチ, ロール）
1100	[0.5 pi, 0, 0]	デフォルト向き：（ヨー, ピッチ, ロール）
1101	[1.0 pi, 0, 0]	デフォルト向き：（ヨー, ピッチ, ロール）
0010	[-0.5 pi, 0, 0]	デフォルト向き：（ヨー, ピッチ, ロール）
0011	[0, 0.5 pi, 0]	デフォルト向き：（ヨー, ピッチ, ロール）
0000	[0, -0.5 pi, 0]	デフォルト向き：（ヨー, ピッチ, ロール）
0001	[0, 0, 1.0 pi]	デフォルト向き：（ヨー, ピッチ, ロール）
111	N/A	デフォルト角度が送信される
10	N/A	明示的な角度が送信される

10

20

30

40

【0 3 3 7】

d e f a u l t Y a w C o d e

デフォルトのヨー角のコード。

【表 4 5】

コードワード	値
010	-0.75 pi
011	-0.5 pi
000	-0.25 pi
001	0
110	0.25 pi
111	0.5 pi
100	0.75 pi
101	pi

【0 3 3 8】

d e f a u l t P i t c h C o d e

デフォルトのピッチ角のコード。

50

【表 4 6】

コードワード	値
11	-0.5 pi
100	-0.25 pi
00	0
101	0.25 pi
01	0.5 pi

10

【0 3 3 9】

default Role Code
デフォルトのロール角のコード。

【表 4 7】

コードワード	値
010	-0.75pi
011	-0.5 pi
000	-0.25 pi
001	0
110	0.25 pi
111	0.5 pi
100	0.75 pi
101	pi

20

30

【0 3 4 0】

coarse Angle Code
1 / 36 pi 刻みでの粗い角度インジケーションのコード。

40

50

【表 4 8】

コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値
010010	-1 pi	011101	-0.58333 pi	101100	-0.16667 pi	111111	0.25 pi	001110	0.66667 pi
010011	-0.97222 pi	0110010	-0.55556 pi	101101	-0.13889 pi	111100	0.27778 pi	001111	0.69444 pi
010000	-0.94444 pi	0110011	-0.52778 pi	100010	-0.11111 pi	111101	0.30556 pi	001100	0.72222 pi
010001	-0.91667 pi	0110000	-0.5 pi	100011	-0.08333 3 pi	110010	0.33333 pi	001101	0.75 pi
010110	-0.88889 pi	0110001	-0.47222 pi	100000	-0.05555 6 pi	110011	0.36111 pi	000010	0.77778 pi
010111	-0.86111 pi	0110110	-0.44444 pi	100001	-0.02777 8 pi	110000	0.38889 pi	000011	0.80556 pi
010100	-0.83333 pi	0110111	-0.41667 pi	100110	0 pi	110001	0.41667 pi	000000	0.83333 pi
010101	-0.80556 pi	0110100	-0.38889 pi	100111	0.02777 8 pi	110110	0.44444 pi	000001	0.86111 pi
0111010	-0.77778 pi	0110101	-0.36111 pi	100100	0.05555 6 pi	110111	0.47222 pi	000110	0.88889 pi
0111011	-0.75 pi	101010	-0.33333 pi	100101	0.08333 3 pi	110100	0.5 pi	000111	0.91667 pi
0111000	-0.72222 pi	101011	-0.30556 pi	111010	0.11111 pi	110101	0.52778 pi	000100	0.94444 pi
0111001	-0.69444 pi	101000	-0.27778 pi	111011	0.13889 pi	001010	0.55556 pi	000101	0.97222 pi
0111110	-0.66667 pi	101001	-0.25 pi	111000	0.16667 pi	001011	0.58333 pi		

10

20

30

40

50

コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値
0111111	- 0. 63889 pi	101110	- 0. 22222 pi	111001	0. 19444 pi	001000	0. 61111 pi		
0111100	- 0. 61111 pi	101111	- 0. 19444 pi	111110	0. 22222 pi	001001	0. 63889 pi		

【 0 3 4 1 】

a d d F i n e A n g l e

より細かい粒度の角度データが送信されるか否かを示すフラグ。

【 0 3 4 2 】

f i n e A n g l e C o d e

1 / 1 8 0 0 p i 刻みでの細かい角度インジケーションのコード。

【表 4 9】

コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値
11100	0. 0005555 6 pi	001010	0. 0061111 pi	000000	0. 01166 7 pi	011110	0. 01722 2 pi	010100	0. 02277 8 pi
11101	0. 0011111 pi	001011	0. 0066667 pi	000001	0. 01222 2 pi	011111	0. 01777 8 pi	010101	0. 02333 3 pi
10010	0. 0016667 pi	001000	0. 0072222 pi	000110	0. 01277 8 pi	011100	0. 01833 3 pi	11010	0. 02388 9 pi
10011	0. 0022222 pi	001001	0. 0077778 pi	000111	0. 01333 3 pi	011101	0. 01888 9 pi	11011	0. 02444 4 pi
10000	0. 0027778 pi	001110	0. 0083333 pi	000100	0. 01388 9 pi	010010	0. 01944 4 pi	11000	0. 025 pi
10001	0. 0033333 pi	001111	0. 0088889 pi	000101	0. 01444 4 pi	010011	0. 02 pi	11001	0. 02555 6 pi
10110	0. 0038889 pi	001100	0. 0094444 pi	011010	0. 015 pi	010000	0. 02055 6 pi	111110	0. 02611 1 pi
10111	0. 0044444 pi	001101	0. 01 pi	011011	0. 01555 6 pi	010001	0. 02111 1 pi	111111	0. 02666 7 pi
10100	0. 005 pi	000010	0. 010556 pi	011000	0. 01611 1 pi	010110	0. 02166 7 pi	111110	0. 02722 2 pi
10101	0. 0055556 pi	000011	0. 011111 pi	011001	0. 01666 7 pi	010111	0. 02222 2 pi		

【 0 3 4 3 】

G e t G a i n ()

ゲイン値 (d B) を返す。

10

20

30

40

50

【表 5 0】

構文	ビット数	タイプ
gain = GetGain()		
{		
gain = LUT(coarseGainCode);	var	vlclbf
if (addFineGain)	1	bslbf
{		
fineGain = LUT(fineGainCode);	var	vlclbf
gain = gain + fineGain;		
}		
return gain;		
}		

10

20

30

40

50

【0 3 4 4】

c o a r s e G a i n C o d e

粗いゲイン値 (dB) のコード。

【表 5 1】

コードワード	値	
0110100	-60	10
0110101	-30	
011110	-24	
011111	-18	
01110	-15	
1100	-12	
111	-9	
100	-6	
101	-3	
00	0	
010	3	20
1101	6	
01100	9	
01101110	12	
01101111	15	
0110110	24	30

【0 3 4 5】

add Fine Gain

より細かい分解能ゲイン値を提供するために、さらなるデータが送信されるか否かを示すフラグ。

【0 3 4 6】

fine Gain Code

より細かいゲイン分解能（1 dB 分解能）のコード。

40

50

【表 5 2】

コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値	コード ワード	値
000110	-29	010000	-23	001010	-17	001100	-11	01110	-5
000111	-28	010001	-22	001011	-16	001101	-10	01111	-4
000100	-27	010110	-21	001000	-15	011010	-9	0000	-3
000101	-26	010111	-20	001001	-14	011011	-8	10	-2
010010	-25	010100	-19	001110	-13	011000	-7	11	-1
010011	-24	010101	-18	001111	-12	011001	-6		

10

【0 3 4 7】

GetGainDelta()

ゲインデルタ値 (dB) を返す。

【表 5 3】

20

構文	ビット数	タイプ
gain = GetGainDelta()		
{		
gain = GetGain();		
return gain;		
}		

30

【0 3 4 8】

GetRenderingConditions()

ソースをレンダリングする方法に関する情報を含む。

40

50

【表 5 4】

構文	ビット数	タイプ
GetRenderingConditions()		
{		
if (isNormalConditions == false)	1	bs bf
{		
doReverb;	1	bs bf
doEarlyReflections;	1	bs bf
doDoppler;	1	bs bf
doDistanceAtt;	1	bs bf
doDirectPath;	1	bs bf
if (regionDependentActivation)	1	bs bf
{		
activateRegionID = GetID();		
activateGoingIn;	1	bs bf
}		
if (regionDependentDeactivation)	1	bs bf
{		
deactivateRegionID = GetID();		
deactivateGoingIn;	1	bs bf
}		
}		
}		

【0349】

isNormalConditions

レンダリング条件がノーマルか否かを示すフラグ。これは、いずれの音響特徴が明示的にオフにされず、ソース、またはソースの特定の音響特徴のレンダリングがデコーダによって決定されることを意味する。

【0350】

doReverb

対応するソースの残響をレンダリングするか否かを示すフラグ。

【0351】

10

20

30

40

50

d o E a r l y R e f l e c t i o n s

対応するソースの初期反射をレンダリングするか否かを示すフラグ。

【 0 3 5 2 】

d o D o p p l e r

対応するソースのドップラー硬化をレンダリングするか否かを示すフラグ。

【 0 3 5 3 】

d o D i s t a n c e A t t

対応するソースの距離減衰をレンダリングするか否かを示すフラグ。

【 0 3 5 4 】

d o D i r e c t P a t h

対応するソースの直接経路をレンダリングするか否かを示すフラグ。

【 0 3 5 5 】

r e g i o n D e p e n d e n t A c t i v a t i o n

ユーザの位置に応じて、ソースをアクティブにするか否かをさらならデータが指定するか否かを示すフラグ。

【 0 3 5 6 】

a c t i v a t e G o i n g I n

真の場合、ユーザが指定された領域内に移動したときにソースがアクティブ化されることを示すフラグ。

【 0 3 5 7 】

r e g i o n D e p e n d e n t D e a c t i v a t i o n

ユーザの位置に応じて、ソースを非アクティブにするか否かをさらならデータが指定するか否かを示すフラグ。

【 0 3 5 8 】

d e a c t i v a t e G o i n g I n

真の場合、ユーザが指定された領域内に移動したときにソースが非アクティブ化されることを示すフラグ。

【 0 3 5 9 】

上記では、音声および音源という用語が使用されているが、これは音 / サウンドおよびサウンドソースという用語と等価であることが理解されるであろう。「音声」という用語への言及は、「音 / サウンド」という用語への言及に置き換えることができる。

【 0 3 6 0 】

明瞭さのために、上記の説明は、異なる機能的回路、ユニット、およびプロセッサに関連して本発明の実施形態を説明している。しかしながら、本発明を損なうことなく、異なる機能的回路、ユニット、またはプロセッサ間で、機能が適切に分散され得ることが理解されよう。例えば、複数の別々のプロセッサまたはコントローラによって実行されるように説明された機能が、同じプロセッサまたはコントローラによって実行されてもよい。したがって、特定の機能的ユニットまたは回路への言及は、厳密な論理的または物理的な構造または組織を示すものではなく、説明される機能を提供するための適切な手段への言及であると考えられたい。

【 0 3 6 1 】

本発明は、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み合わせを含む任意の適切な形態で実施することができる。本発明は、1つ以上のデータプロセッサおよび / またはデジタル信号プロセッサ上で動作するコンピュータソフトウェアとして少なくとも部分的に実装されてもよい。本発明の実施形態の要素および構成要素は、任意の適切な方法で物理的、機能的、および論理的に実装され得る。実際には、機能は、単一のユニット、複数のユニット、または他の機能ユニットの一部として実装されてもよい。したがって、本発明は、単一のユニットとして実装されてもよく、または複数の異なるユニット、回路、およびプロセッサの間で物理的および機能的に分散されてもよい。

【 0 3 6 2 】

10

20

30

40

50

いくつかの実施形態に関連して本発明を説明したが、本発明は明細書に記載される具体的形態に限定されない。本発明の範囲は添付の特許請求の範囲によってのみ限定される。さらに、ある特徴が特定の実施形態に関連して記載されているように見えたとしても、当業者は、上記実施形態の様々な特徴が本発明に従って組み合わせられ得ることを認識するであろう。請求項において、備える、含む等の用語は他の要素またはステップの存在を排除するものではない。

【 0 3 6 3 】

さらに、個別に列挙されていたとしても、複数の手段、要素、回路、または方法ステップは、例えば、単一の回路、ユニット、またはプロセッサによって実施されてもよい。さらに、個々の特徴が異なる請求項に含まれていたとしても、これらは好適に組み合わされ得、異なる請求項に含まれていることは、特徴の組み合わせが実現不可能であるおよび/または有利でないことを意味するものではない。また、1つのクレームカテゴリー内にある特徴が含まれているからといって、その特徴がこのカテゴリーに限定されるとは限らず、特徴は適宜、他のクレームカテゴリーに等しく適用可能である。さらに、請求項における特徴の順序は、特徴が作用すべき特定の順序を指すものではなく、特に、方法クレームにおける個々のステップの順序はステップをその順序で実行しなければならないことを意味しない。ステップは任意の適切な順序で実行され得る。また、単数形の表現は複数形を排除するものではない。したがって、「a」、「a n」、「第1の」、「第2の」などの表現は複数を排除するものではない。特許請求の範囲内の参照符号は明瞭さのための例に過ぎず、請求項の範囲を如何ようにも限定するものではない。

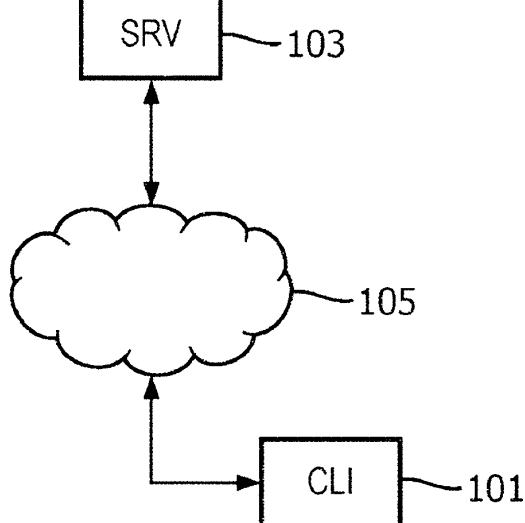
10

20

30

40

50



【図 2】

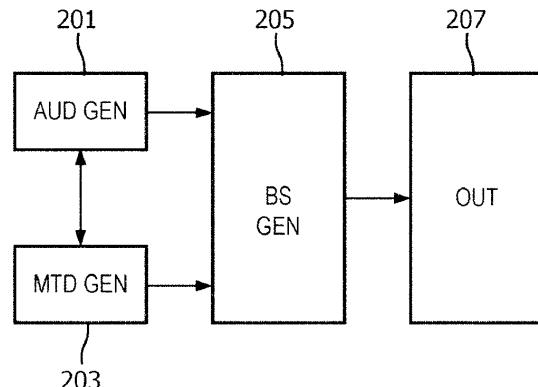


図 2

図 1

【図3】

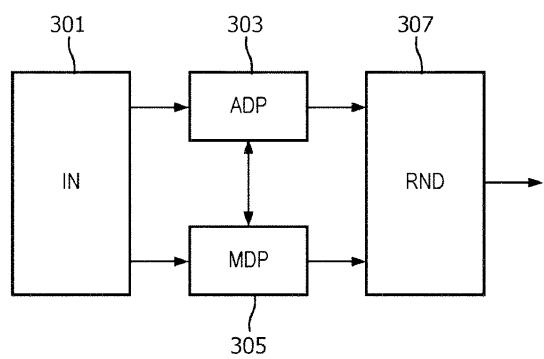


図3

【図5】

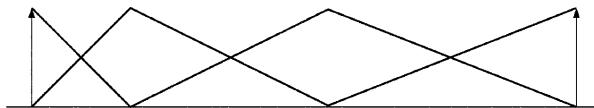


図5

【図6】

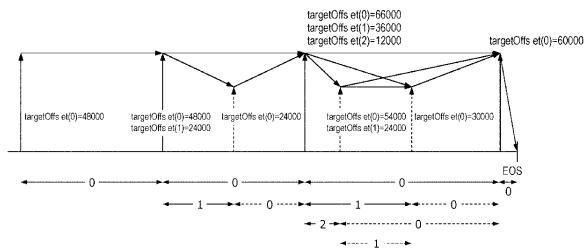


図6

【外国語明細書】

2024124401000073.pdf

10

20

30

40

50

フロントページの続き

ドーフェン ハイ テック キャンパス 52 フィリップス インターナショナル ビー. ヴィ. イ
ンテレクチュアル プロパティー アンド スタンダーズ
F ターム (参考) 5D162 AA05 AA09 AA11 CC18 CC36 CD07 EG04