(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7612700号
(P7612700)

(45)発行日　令和7年1月14日(2025.1.14)

(24)登録日　令和6年12月27日(2024.12.27)

(51)国際特許分類
G06T 19/00 (2011.01)

FI
G06T 19/00 A

請求項の数 20 （全60頁）

(21)出願番号　特願2022-548609(P2022-548609)
(86)(22)出願日　令和3年2月12日(2021.2.12)
(65)公表番号　特表2023-513304(P2023-513304A)
(43)公表日　令和5年3月30日(2023.3.30)
(86)国際出願番号　PCT/US2021/017967
(87)国際公開番号　WO2021/163570
(87)国際公開日　令和3年8月19日(2021.8.19)
審査請求日　令和6年1月29日(2024.1.29)
(31)優先権主張番号　16/792,080
(32)優先日　令和2年2月14日(2020.2.14)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　517160525
バルブ コーポレーション
アメリカ合衆国、９８００４ ワシントン州、ベルビュー、ノースイースト ４ティーエイチ ストリート １０４００スイート １４００
(74)代理人　110001737
弁理士法人スズエ国際特許事務所
(72)発明者　ライビー、アーロン
アメリカ合衆国、ワシントン州 ９８００４、ベルビュー、ノースイースト・フォース・ストリート １０４００
審査官　渡部 幸和

最終頁に続く

(54)【発明の名称】　分散システムにおける動き平滑化

(57)【特許請求の範囲】

【請求項１】

ヘッドマウントディスプレイ（ＨＭＤ）であって、

１つ以上のディスプレイパネルと、

プロセッサと、

コンピュータ実行可能命令を記憶するメモリと、を備え、前記命令が、前記プロセッサによって実行されると、前記ＨＭＤに、

前記ＨＭＤに通信可能に連結されたホストコンピュータから、圧縮されたピクセルデータを表す動きベクトルアレイを受信することと、

前記圧縮されたピクセルデータを解凍してピクセルデータを取得することと、

レンダリングメッシュを生成することと、なお、前記レンダリングメッシュはテッセレーションメッシュを含む、

前記動きベクトルアレイ間の非ゼロ動きベクトルに少なくとも部分的に基づいて、前記レンダリングメッシュの頂点を、移動された頂点として前記レンダリングメッシュ内の異なる位置に移動させることと、

前記ピクセルデータを修正して、修正されたピクセルデータを取得することであって、前記ピクセルデータが、

前記ＨＭＤの予測姿勢、および

前記移動された頂点、なお、前記動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを修正することが、前記レンダリングメッシュの前記移動された頂点

に従って前記ピクセルデータのピクセル値を移動させて、前記修正されたピクセルデータを取得することを含む、に少なくとも部分的に基づいて修正される、取得することと、
前記修正されたピクセルデータに少なくとも部分的に基づいて、前記１つ以上のディスプレイパネル上に画像を提示することと、を行わせる、ＨＭＤ。
【請求項２】
前記頂点が、
前記非ゼロ動きベクトルの方向に、かつ
前記非ゼロ動きベクトルの大きさに対応する量だけ、移動される、請求項１に記載のＨＭＤ。
【請求項３】
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ＨＭＤに、さらに、符号化されたデータストリームから前記動きベクトルアレイを抽出させる、請求項１に記載のＨＭＤ。
【請求項４】
前記ホストコンピュータが、前記ＨＭＤに無線で連結され、
前記動きベクトルアレイが、前記ホストコンピュータから無線で受信される、請求項１に記載のＨＭＤ。
【請求項５】
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ＨＭＤに、
さらに、前記動きベクトルアレイにフィルタを適用して、修正された動きベクトルアレイを取得させ、
前記ピクセルデータを修正することが、前記修正された動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを修正することを含む、請求項１に記載のＨＭＤ。
【請求項６】
方法であって、
ヘッドマウントディスプレイ（ＨＭＤ）によって、ホストコンピュータから、圧縮されたピクセルデータを表す動きベクトルアレイを受信することと、
前記ＨＭＤによって、前記圧縮されたピクセルデータを解凍して、ピクセルデータを取得することと、
前記ＨＭＤによってレンダリングメッシュを生成することと、なお、前記レンダリングメッシュはテッセレーションメッシュを含む、
前記動きベクトルアレイ間の非ゼロ動きベクトルに少なくとも部分的に基づいて、前記レンダリングメッシュの頂点を、移動された頂点として前記レンダリングメッシュ内の異なる位置に移動させることと、
前記ＨＭＤによって、前記移動された頂点に少なくとも部分的に基づいて前記ピクセルデータを修正して、修正されたピクセルデータを取得することと、なお、前記移動された複数の頂点に少なくとも部分的に基づく前記ピクセルデータの前記修正が、前記レンダリングメッシュの前記移動された頂点に従って前記ピクセルデータのピクセル値を移動させて、前記修正されたピクセルデータを取得することを含む、
前記修正されたピクセルデータに少なくとも部分的に基づいて、前記ＨＭＤの１つ以上のディスプレイパネル上に画像を提示することと、を含む、方法。
【請求項７】
前記ピクセルデータの前記修正の前に、
前記ＨＭＤのメモリ内の前記ピクセルデータをキャッシュされたピクセルデータとしてキャッシュすることと、
前記キャッシュされたピクセルデータが、前記ＨＭＤに利用可能な最後に解凍されたピクセルデータを表すことを決定することと、
前記ＨＭＤの前記メモリから前記キャッシュされたピクセルデータを取り出すことと、
をさらに含む、請求項６に記載の方法。

【請求項８】
前記動きベクトルアレイを符号化されたデータストリームから抽出することをさらに含む、請求項６に記載の方法。
【請求項９】
前記画像の前記提示の前に、
前記ＨＭＤによって、前記ホストコンピュータから、前記ホストコンピュータ上で実行されるアプリケーションによって、前記ピクセルデータを生成するために使用された前記ＨＭＤの予測姿勢を示す姿勢データを受信することをさらに含み、
前記ピクセルデータの前記修正が、前記ＨＭＤの前記予測姿勢と更新された姿勢の予測との間の比較にさらに少なくとも部分的に基づいている、請求項６に記載の方法。
【請求項１０】
前記画像の前記提示の前に、
前記ＨＭＤによって、前記ホストコンピュータから、前記ホストコンピュータ上で実行されるアプリケーションによって出力されるＺバッファデータを表す深度データを受信することをさらに含み、
前記ピクセルデータの前記修正が、前記深度データにさらに少なくとも部分的に基づいている、請求項６に記載の方法。
【請求項１１】
前記ＨＭＤによって、前記動きベクトルアレイにフィルタを適用して、修正された動きベクトルアレイを取得することをさらに含み、
前記ピクセルデータの前記修正が、前記修正された動きベクトルアレイに基づいて、前記ピクセルデータを修正することを含む、請求項６に記載の方法。
【請求項１２】
ディスプレイシステムであって、
１つ以上のディスプレイパネルと、
プロセッサと、
コンピュータ実行可能命令を記憶するメモリと、を備え、前記命令が、前記プロセッサによって実行されると、前記ディスプレイシステムに、
ホストコンピュータから、圧縮されたピクセルデータを表す動きベクトルアレイを受信することと、
前記圧縮されたピクセルデータを解凍してピクセルデータを取得することと、
レンダリングメッシュを生成することと、なお、前記レンダリングメッシュはテッセレーションメッシュを含む、
前記動きベクトルアレイ間の非ゼロ動きベクトルに少なくとも部分的に基づいて、前記レンダリングメッシュの頂点を、移動された頂点として前記レンダリングメッシュ内の異なる位置に移動させることと、
前記移動された頂点に少なくとも部分的に基づいて前記ピクセルデータを修正して、修正されたピクセルデータを取得することと、なお、前記移動された複数の頂点に少なくとも部分的に基づく前記ピクセルデータの前記修正が、前記レンダリングメッシュの前記移動された頂点に従って前記ピクセルデータのピクセル値を移動させて、前記修正されたピクセルデータを取得することを含む、
前記修正されたピクセルデータに少なくとも部分的に基づいて、前記１つ以上のディスプレイパネル上に画像を提示することと、を行わせる、ディスプレイシステム。
【請求項１３】
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ピクセルデータを修正する前に、前記ディスプレイシステムに、
前記メモリ内の前記ピクセルデータを、キャッシュされたピクセルデータとしてキャッシュすることと、
前記キャッシュされたピクセルデータが、前記ディスプレイシステムに利用可能な最後に解凍されたピクセルデータを表すことを決定することと、

前記メモリから前記キャッシュされたピクセルデータを取り出すことと、を行わせる、請求項１２に記載のディスプレイシステム。
【請求項１４】
前記ホストコンピュータが、前記ディスプレイシステムに無線で連結され、
前記動きベクトルアレイが、前記ホストコンピュータから無線で受信される、請求項１２に記載のディスプレイシステム。
【請求項１５】
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記画像を提示する前に、前記ディスプレイシステムに、
前記ホストコンピュータから、前記ホストコンピュータ上で実行されるアプリケーションによって出力されるＺバッファデータを表す深度データを受信させ、
前記ピクセルデータを修正することが、前記深度データにさらに少なくとも部分的に基づいている、請求項１２に記載のディスプレイシステム。
【請求項１６】
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ディスプレイシステムに、
前記動きベクトルアレイにフィルタを適用して、修正された動きベクトルアレイを取得させ、
前記ピクセルデータを前記修正することが、前記修正された動きベクトルアレイに基づいて前記ピクセルデータを修正することを含む、請求項１２に記載のディスプレイシステム。
【請求項１７】
前記ディスプレイシステムが、仮想現実（ＶＲ）ヘッドセットまたは拡張現実（ＡＲ）ヘッドセットのうちの少なくとも１つを含む、請求項１２に記載のディスプレイシステム。
【請求項１８】
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ディスプレイシステムに、さらに、符号化されたデータストリームから前記動きベクトルアレイを抽出させる、請求項１２に記載のディスプレイシステム。
【請求項１９】
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記画像を提示する前に、前記ＨＭＤに、
前記ホストコンピュータから、前記ホストコンピュータ上で実行されるアプリケーションによって出力されるＺバッファデータを表す深度データを受信させ、
前記ピクセルデータを修正することが、前記深度データにさらに少なくとも部分的に基づいている、請求項１に記載のＨＭＤ。
【請求項２０】
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ピクセルデータを修正する前に、前記ＨＭＤに、
前記メモリ内の前記ピクセルデータを、キャッシュされたピクセルデータとしてキャッシュすることと、
前記キャッシュされたピクセルデータが、前記ＨＭＤに利用可能な最後に解凍されたピクセルデータを表すことを決定することと、
前記メモリから前記キャッシュされたピクセルデータを取り出すことと、を行わせる、請求項１に記載のＨＭＤ。
【発明の詳細な説明】
【技術分野】
【０００１】
関連出願の相互参照
本出願は、２０２０年２月１４日に出願された「ＭＯＴＩＯＮ　ＳＭＯＯＴＨＩＮＧ　ＩＮ　Ａ　ＤＩＳＴＲＩＢＵＴＥＤ　ＳＹＳＴＥＭ」と題する同時係属の本発明の譲受人に譲

渡された米国特許出願第１６／７９２，０８０号の優先権を主張するＰＣＴ出願であり、当該米国特許出願は、参照によりその全体が本明細書に組み込まれる。

【背景技術】

【０００２】

仮想現実（ＶＲ）システムは、ビデオゲーム業界の内部および外部の両方で使用されている。ＶＲシステムは、オールインワン（またはスタンドアロン）ＶＲヘッドセットとして、または分散システムとしてセットアップされ得る。分散セットアップでは、ＶＲヘッドセットは、物理的にテザリングされるか、またはホストコンピュータに無線で接続され得る。この分散セットアップでは、ホストコンピュータは、通常、フレームを出力するビデオゲームなどのグラフィックスベースのアプリケーションを実行し、ＶＲヘッドセットは、フレームがホストコンピュータからストリーミングされるときにフレームを表示する。このタイプのセットアップは、ホストコンピュータの高演算能力を活用して、「シンクライアント」デバイスのように機能する軽量のＶＲヘッドセットに高品質の画像を表示する。さらに、ＶＲヘッドセットに内蔵されているようなＶＲシステム用ディスプレイは、通常、ＶＲアプリケーションに好適である最小リフレッシュレートで動作する。例えば、９０ヘルツ（Ｈｚ）はＶＲディスプレイの一般的なリフレッシュレートである。「ライブレンダリング」シナリオでは、グラフィックスベースのアプリケーションは、ディスプレイのリフレッシュレートに適合するフレームレートでレンダリングするためのフレームを出力する。このシナリオでは、十分なデータ転送レートでフレームがホストコンピュータからＶＲヘッドセットに転送されていると仮定すると、画面がリフレッシュされるたびに、アプリケーションによって出力される新しいフレーム（本明細書では「実際のフレーム」と称する）が表示され得る。そのようなライブレンダリングシナリオは、多くの場合、アプリケーションが「フレームレートに到達する」と称される。

【０００３】

実際には、アプリケーションは、様々な理由で常にフレームレートに到達するとは限らない。例えば、アプリケーションがフレームを断続的にドロップする場合があり、および／またはアプリケーションが一時的により遅いレートでフレームを出力する場合がある（例えば、理想的なフレームレートが１秒当たり９０フレームの場合は１秒当たり４５フレーム）。さらに、分散システムでは、ネットワークの混雑は、ホストコンピュータからＶＲヘッドセットにデータが転送されるレートにおけるレイテンシをもたらし得る。これらの状況では、「回転のみの再投影」と呼ばれる技術が、ユーザの頭部の回転を考慮するように、欠落しているフレームを再投影フレームで置き換えるために使用され得、これは、欠落しているフレームがないかのようにユーザに見せる。例えば、再投影しない場合、アプリケーションからのフレームレートが不足する、またはフレームがＶＲヘッドセットに遅れて到着すると、ゲーム内でスタッタまたはヒッチが発生する場合がある。ユーザが仮想環境に完全に没入しているＶＲアプリケーションでは、フレームが欠落し、欠落しているフレームを補償するための再投影が存在しない場合、ユーザは、不快になり得る。したがって、再投影は、フレームが欠落したときにユーザ体験をより良好にすることを可能にする技術である。アプリケーションが理想的なフレームレートの半分でフレームを出力している例を検討する（例えば、理想的なフレームレートが１秒当たり９０フレームの場合は１秒当たり４５フレーム）。この例では、最後にレンダリングされた実際のフレームのピクセルデータを使用して、シーンを変換する再投影フレームを作成し（例えば、回転および再投影の計算を通して）、再投影シーンをユーザの現在の頭部の向きに一致させることで、１つおきにフレームを再投影させることができる。これは、再投影フレームが欠損フレームを補償するために使用される場合でさえも、ユーザの頭部の回転を考慮して予想される方式でシーンが移動しているようにユーザに見えるようにする。

【０００４】

回転のみの再投影は、ゲーム内のスタッタまたはヒッチを防止するが、少なくとも、低持続性ディスプレイを使用するＶＲシステム（例えば、ディスプレイがフレーム時間のうちのごくわずかな間、点灯される場合）では、頭部の回転中に、回転のみの再投影自体の

望ましくない視覚的アーチファクトを生成する。例えば、回転のみの再投影は、頭部の回転を考慮するが、フレーム間でシーン内を移動するかまたは動く仮想オブジェクトを考慮しない。これは、移動しているかまたは動いているオブジェクトに関して、「ジャダー」と呼ばれる望ましくない視覚的アーチファクトを発生させ得る。ジャダーは、移動しているオブジェクト（例えば、画面上を移動する弾丸またはボール）が２つの位置の間で（またはそれ自体から離れて）フレーム間で跳ね返るように見える、「ダブルゴースト効果」をユーザに知覚させる。したがって、再投影が使用されている間にユーザがその頭部を回転させると、シーン内で任意に移動しているかまたは動いているオブジェクトがジャダーすることになる。

【０００５】

本明細書では、これらのシステムおよび他のシステムを改善および強化するための技術的解決法を提供する。

【図面の簡単な説明】

【０００６】

詳細な説明を、添付の図面を参照して説明する。各図において、参照番号の左端数字は、参照番号が最初に現れる図を識別する。異なる図における同じ参照番号の使用は、類似または同一の構成要素または特徴を示す。

【０００７】

【図１】本明細書に開示される実施形態による、フレームが、ヘッドマウントディスプレイ（ＨＭＤ）に通信可能に連結されたホストコンピュータ上で実行されるアプリケーションによって出力されるときに、ＨＭＤ上にフレームをレンダリングするための例示的なプロセスのフロー図である。図１は、シーン内で移動しているかまたは動いているオブジェクトを考慮するために再投影中に実装され得る動き平滑化技術を示す。

【図２】本明細書に開示される実施形態による、ホストコンピュータおよびＨＭＤのそれぞれのレンダリングワークロードを示す２つの例示的なタイムラインを示す図である。

【図３】再投影がＨＭＤなどのディスプレイ上のフレームをレンダリングするために使用されているときに、シーン内で移動しているかまたは動いているオブジェクトを考慮するための例示的な動き平滑化技術を示す図である。

【図４】例示的なレンダリングメッシュ、および再投影フレームの動き平滑化にレンダリングメッシュをどのように使用することができるかを示す図である。

【図５】本明細書に開示される実施形態による、動き平滑化技術の一部としてグラフィックスプロセッシングユニット（ＧＰＵ）によって生成される動きベクトルを使用して再投影フレームを生成するための例示的なプロセスのフロー図である。

【図６】本明細書に開示される実施形態による、ピクセルデータが動きベクトル推定のためにＧＰＵに入力される前に、以前にレンダリングされたフレームのピクセルデータを位置合わせするための例示的なプロセスのフロー図である。

【図７】本明細書に開示される実施形態による、以前にレンダリングされたフレームのいくつかのピクセルデータを除外し、動きベクトル推定のためにＧＰＵへの入力としてピクセルデータの残りの部分を提供するための例示的なプロセスのフロー図である。

【図８】本明細書に開示される実施形態による、以前にレンダリングされたフレームのピクセルデータのパネルマスク部分を除外し、動きベクトル推定のためにＧＰＵへの入力としてピクセルデータの残りの部分を提供する、以前にレンダリングされたフレームのピクセルデータを位置合わせするための例示的なプロセスのフロー図である。

【図９】本明細書に開示される実施形態による、動き平滑化技術の一部として再投影フレームを生成するために使用される前にＧＰＵから出力される動きベクトルを閾値化するための例示的なプロセスのフロー図である。

【図１０】本明細書に開示される実施形態による、動き平滑化技術の一部として再投影フレームを生成するために使用される前にＧＰＵから出力される動きベクトルを減衰させるための例示的なプロセスのフロー図である。

【図１１】本明細書に開示される実施形態による、減衰テクスチャを生成するための例示

的なプロセスのフロー図であり、減衰テクスチャは、動き平滑化技術の一部として再投影フレームを生成するために使用される前に、ＧＰＵから出力される動きベクトルを減衰させるために使用され得る。
【図１２】本明細書に開示される実施形態による、結果として生じる動きベクトルのセットが動き平滑化技術の一部として再投影フレームを生成するために使用される前に、ほとんどまたはまったくない色変化の領域に対応する動きベクトルをゼロ化するための例示的なプロセスのフロー図である。
【図１３】本明細書に開示される実施形態による、結果として生じる動きベクトルのセットが動き平滑化技術の一部として再投影フレームを生成するために使用される前に、１つ以上のフィルタを使用して動きベクトル場を「クリーンアップする」ための例示的なプロセスのフロー図である。
【図１４】本明細書に開示される実施形態による、ピクセルデータが動きベクトル推定のためにＧＰＵに入力される前に、以前にレンダリングされたフレームを回転させるための例示的なプロセスのフロー図である。
【図１５】本明細書に開示される実施形態による、以前にレンダリングされたフレームのルマデータおよびクロマデータに基づいて生成される動きベクトルアレイ間で選択するための例示的なプロセスのフロー図である。
【図１６】本明細書に開示される実施形態による、動きベクトルの複数のアレイを取得し、アレイ間の差を決定し、動き平滑化のための決定された差に基づいて最終的な動きベクトルアレイを生成するための例示的なプロセスのフロー図である。
【図１７】本明細書に開示される実施形態による、画像領域の異なる部分について異なる解像度で複数の動きベクトルアレイを取得するための例示的なプロセスのフロー図である。
【図１８Ａ】本明細書に開示される実施形態による、ＨＭＤおよびホストコンピュータを利用するシステムの２つの代替的なセットアップを示す。
【図１８Ｂ】本明細書に開示される実施形態による、ＨＭＤおよびホストコンピュータを利用するシステムの２つの代替的なセットアップを示す。
【図１９】本明細書に開示される技術が実装され得る、ＶＲヘッドセットなどの装着用デバイスの例示的なコンポーネントを示す。
【図２０】本明細書に開示される技術が実装され得る、ホストコンピュータの例示的なコンポーネントを示す。
【発明を実施するための形態】
【０００８】
本明細書には、とりわけ、分散システムのディスプレイ上でフレームをレンダリングするときに、シーン内を移動しているかまたは動いているオブジェクトを考慮するための動き平滑化技術が記載されている。本明細書に記載される動き平滑化技術は、移動しているかまたは動いているオブジェクトの前述のジャダーアーチファクトなど、移動しているかまたは動いているオブジェクトに関して望ましくない視覚的アーチファクトを軽減する。ヘッドマウントディスプレイ（ＨＭＤ）は、ディスプレイ上に画像をレンダリングするときに開示された動き平滑化技術を実装することができるディスプレイまたはディスプレイシステムの例示的なタイプである。ＨＭＤは、ユーザが仮想現実（ＶＲ）環境または拡張現実（ＡＲ）環境に没入する目的で、ユーザによって装着され得る。ＨＭＤなどのディスプレイまたはディスプレイシステムは、ホストコンピュータから分離され得るが、なおもこれに通信可能に連結され得る。アプリケーション（例えば、ビデオゲーム）は、ホストコンピュータ上で実行され、一連のフレームの個々のフレームについてのピクセルデータを生成する。ピクセルデータは、フレームごとにＨＭＤに送信され、ＨＭＤの１つ以上のディスプレイパネルは、受信したピクセルデータに基づいて画像をレンダリングする。これらの画像は、ＨＭＤに含まれる光学系を通してユーザによって視認され、ユーザに、ユーザがＶＲまたはＡＲ環境に没入しているかのように画像を知覚させる。
【０００９】
記載されるように、ＨＭＤなどのディスプレイシステムは、「再投影」と呼ばれる技術

を利用して、フレームレートに達していないアプリケーションを補償することができる。例えば、再投影フレームは、実際のフレーム間でレンダリングされて理想的なフレームレートを達成し得、各再投影フレームは、ホストコンピュータ上で実行されているアプリケーションによって出力された最近にレンダリングされた実際のフレーム（例えば、最後にレンダリングされた実際のフレーム）からのピクセルデータを使用して生成され得る。再投影フレームでは、以前の実際のフレームでレンダリングされたシーンが、ユーザの頭部の回転を考慮した方式で変換される（例えば、回転および再投影の計算を通して）。

【００１０】

本明細書には、フレーム間で移動するかまたは動くオブジェクトの動きをさらに考慮する方式で、フレームのピクセルデータを修正するために分散システムで使用可能な動き平滑化技術が記載される。本明細書に記載される動き平滑化技術は、コンピュータビジョンアルゴリズムを使用して、動きベクトルの形態で複数のフレームにわたるオブジェクトの動き（例えば、方向および大きさ）を推定する。例えば、動きベクトルアレイは、ピクセルデータがＨＭＤなどのディスプレイシステムに送信される前に、ピクセルデータを圧縮する結果（例えば、副産物として）として、ホストコンピュータによって生成され得る。したがって、動きベクトルアレイは、少なくとも一連のフレーム内のいくつかのフレームについて、圧縮されたピクセルデータを表し得る。用例では、ホストコンピュータ上のグラフィックス処理ユニット（ＧＰＵ）のビデオエンコーダ（例えば、ビデオ符号化チップ）は、複数の以前にレンダリングされたフレームのピクセルデータを分析して、所与のフレームの圧縮されたピクセルデータを表す動きベクトルアレイを生成し得る。動きベクトルは、ＨＭＤに送信され得、動きベクトルを使用して解凍されたピクセルデータを取得することに加えて、動きベクトルは、ＨＭＤによって、移動しているかまたは動いているオブジェクトを考慮する方式で解凍されたピクセルデータを修正するためにも使用され得る。別の言い方をすれば、ホストコンピュータから受信した動きベクトルをＨＭＤ上で使用して、移動するオブジェクトのジャダーが軽減されるようにオブジェクトがレンダリングされる（例えば、再投影される）フレーム内のどこに位置するべきかを（以前にレンダリングされたフレーム内のオブジェクトの動きから）外挿することができる。

【００１１】

例示的な動き平滑化プロセスでは、関連付けられたピクセルデータは、ホストコンピュータ上で圧縮され得る。ホストコンピュータ上のグラフィックス処理ユニット（ＧＰＵ）のビデオエンコーダは、ピクセルデータを圧縮するために使用され得る。ピクセルデータを圧縮する結果として、ビデオエンコーダは、圧縮されたピクセルデータを表す動きベクトルアレイを生成し得る。ホストコンピュータは、動きベクトルアレイをＨＭＤなどのディスプレイシステムに送信し得る。いくつかの実施形態では、ホストコンピュータは、ＨＭＤに無線で連結され得るが、本明細書では有線接続も企図される。ＨＭＤは、ホストコンピュータから、動きベクトルアレイを受信し得る。ＨＭＤは、圧縮された第１のピクセルデータを解凍して、ピクセルデータ（または、解凍もしくは復号アルゴリズムを使用したピクセルデータの近似値）を取得し得る。ＨＭＤはさらに、ホストコンピュータから受信した動きベクトルアレイに少なくとも部分的に基づいてピクセルデータを修正し、修正されたピクセルデータを取得し得る。この修正されたピクセルデータは、シーン内で移動しているかまたは動いているオブジェクトに関するいかなるジャダーも、排除しないとしても、軽減するために、「動き平滑化」される。次いで、動き平滑化フレームは、修正されたピクセルデータに少なくとも部分的に基づいて、ＨＭＤのディスプレイパネル上に画像を提示することによってレンダリングされ得る。

【００１２】

本明細書に記載される動き平滑化技術は、フレームが欠落した場合でも、予想される様式で、オブジェクトがシーン内で移動するかまたは動く、より現実的で、より高忠実な視聴体験を提供する。前述のように、アプリケーションがフレームレートに到達できなかった場合、ネットワークの渋滞が発生した場合、または他の理由の可能性がある場合、フレームは欠落し得る。したがって、本明細書に記載される動き平滑化技術は、フレームが時

々欠落する可能性がある場合に、分散システム内の高忠実度のユーザ体験を維持することによって、これらの発生を補償する。動き平滑化技術はまた、フレームが欠落しないとき（すなわち、フレームがＨＭＤでタイムリーな様式で受信されるとき）のライブレンダリングシナリオにおいても、フレームの送信における固有のレイテンシを補償する。さらに、無線の実装形態において、本明細書に記載される技術は、ＨＭＤにより多くのコンポーネントを追加する必要がなく、無線ＨＭＤにおける動き平滑化補正を可能にし、それによって、今日市場に出回っている多くのオールインワン（またはスタンドアロン）ヘッドセットと比較して、高温になりすぎず、装着しやすい軽量ヘッドセットを可能にする。さらに、本明細書に記載される技術およびシステムは、ホストコンピュータからＨＭＤにデータを圧縮および送信するためにすでに生成された動きベクトルを活用することができ、これは、ＨＭＤ上の再投影フレームの動き平滑化のための動きベクトルを使用する際に追加の輸送コストがないことを意味する。

【００１３】

修正された第２のピクセルデータは、第１のピクセルデータをフレームバッファに出力した後にＨＭＤ上のフレームバッファに出力されるため、ホストコンピュータから受信された動きベクトルアレイは、ＨＭＤによって将来のフレーム（例えば、再投影フレーム）に外挿するために使用され得ることを理解されたい。このようにして、本明細書に記載される例では、第１のフレームは、再投影フレームの前にレンダリングされる。この外挿技術は、フレーム間の内挿と対比され得、本明細書に説明される技術およびシステムは、フレーム間に内挿するために動きベクトルを使用するのではなく、ホストコンピュータから受信した動きベクトルを使用して将来のフレームに外挿することに関係することを理解されたい。

【００１４】

本明細書では、本明細書に開示される技術およびプロセスを実装するように構成されたシステム、例えば、ディスプレイシステム（例えば、ＨＭＤ）を含むシステム、ならびに本明細書に開示される技術およびプロセスを実装するためのコンピュータ実行可能命令を記憶する非一時的コンピュータ可読媒体もまた、開示される。本明細書に開示される技術およびシステムは、例として、ビデオゲームアプリケーション、特にＶＲゲームアプリケーションの文脈で議論されているが、本明細書に説明される技術およびシステムは、非限定的に、非ＶＲアプリケーション（例えば、ＡＲアプリケーション）、および／または産業機械アプリケーション、防衛アプリケーション、ロボットアプリケーションなどの非ゲームアプリケーションを含む、他のアプリケーションに利益を提供し得ることを理解されたい。さらに、ＨＭＤは、画像を表示するためのディスプレイシステムの例として提供されるが、他のタイプのディスプレイシステムは、ホストコンピュータからビデオコンテンツをストリーミングするハンドヘルドディスプレイデバイス、比較的大型の壁掛け式ディスプレイシステム、またはビルボード式ディスプレイシステムなど、本明細書に説明される動き平滑化技術から利益を得ることができることを理解されたい。

【００１５】

本明細書に記載のプロセスは、ロジックフローグラフ内のブロックの集合として例解され、ハードウェア、ソフトウェア、ファームウェア、またはそれらの組み合わせ（つまり、ロジック）に実装することができる一連の動作を表す。ソフトウェアの文脈では、ブロックは、コンピュータ実行可能命令を表し、コンピュータ実行可能命令は、１つ以上のプロセッサによって実行されると、列挙された動作を実行する。一般に、コンピュータ実行可能命令は、特定の機能を実行するか、または特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含む。動作が記載される順序は、限定として解釈されることを意図するものではなく、任意のいくつかの記載されたブロックは、プロセスを実装するために任意の順序でおよび／または並行して組み合わされ得る。

【００１６】

図１は、本明細書に開示される実施形態による、（ユーザ１０４によって装着される）

ヘッドマウントディスプレイ（ＨＭＤ）１０２上にフレームをレンダリングするための例示的なプロセス１００のフロー図であり、フレームは、ＨＭＤ１０２に通信可能に連結されるホストコンピュータ１０６上で実行されるアプリケーションによって出力される。図１は、シーン内で移動しているかまたは動いているオブジェクトを考慮するために再投影中に実装され得る動き平滑化技術を示す。

【００１７】

図１の上部において、ＨＭＤ１０２は、ユーザ１０４によって装着されているように示され、ホストコンピュータ１０６は、ＨＭＤ１０２に通信可能に連結されているように示されている。例えば、ホストコンピュータ１０６は、これらに限定されないが、パーソナルコンピュータ（ＰＣ）、ラップトップコンピュータ、デスクトップコンピュータ、ポータブルデジタルアシスタント（ＰＤＡ）、モバイルフォン、タブレットコンピュータ、セットトップボックス、ゲームコンソール、サーバコンピュータ、ウェアラブルコンピュータ（例えば、スマートウォッチなど）、またはデータを送受信することができる任意の他の電子デバイスを含む、任意のタイプのコンピューティングデバイスおよび／または任意の数のコンピューティングデバイスとして実装することができる。

【００１８】

ホストコンピュータ１０６およびＨＭＤ１０２は、アプリケーション（例えば、ビデオゲーム）を実行し、ディスプレイ上に関連付けられた画像をレンダリングするための分散システムを集合的に表す。いくつかの実施形態では、ホストコンピュータ１０６は、ＨＭＤ１０２を装着するユーザ１０４の世帯など、ＨＭＤ１０２と同じ環境に併置され得る。代替的に、ホストコンピュータ１０６は、ＨＭＤ１０２の地理的位置に関して遠隔地理的位置に位置するサーバコンピュータの形態でホストコンピュータ１０６など、ＨＭＤ１０２に関して遠隔に位置することができる。リモートホストコンピュータ１０６の実装形態では、ホストコンピュータ１０６は、インターネットなどの広域ネットワークを介してＨＭＤ１０２に通信可能に連結され得る。ローカルホストコンピュータ１０６の実装形態では、ホストコンピュータ１０６は、ＨＭＤ１０２と環境（例えば、世帯）内に併置され得、それによって、ホストコンピュータ１０６およびＨＭＤ１０２は、直接的に、または仲介ネットワークデバイスを介してローカルエリアネットワーク（ＬＡＮ）を通して通信可能に一緒に連結され得る。

【００１９】

一緒に通信可能に連結されることによって、ＨＭＤ１０２およびホストコンピュータ１０６は、ＨＭＤ１０２のディスプレイパネル上に対応する画像を提示するために使用されるピクセルデータを生成することによって、所与のフレームをレンダリングするために協働的に一緒に働くように構成されている。ホストコンピュータ１０６およびＨＭＤ１０２は、無線および／または有線接続を介して通信可能に一緒に連結され得る。例えば、デバイス１０２／１０６は、Ｗｉ－Ｆｉ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、無線周波数（ＲＦ）、および／または任意の他の好適な無線プロトコルを使用してデータを交換することができる。追加的または代替的に、デバイス１０２／１０６は、それらの間のデータ転送のための有線接続を容易にするための１つ以上の物理ポートを含み得る。

【００２０】

ＨＭＤ１０２は、開示された動き平滑化技術を実装し得る例示的な「ディスプレイシステム」として本明細書に提示されるが、「ディスプレイシステム」の他のタイプおよび／または実装形態が、本明細書に説明される動き平滑化技術を実装し得ることを理解されたい。したがって、ＨＭＤは、本明細書に記載される技術を実装するための単なる例示的なタイプのディスプレイまたはディスプレイシステムであることを理解されたいが、本明細書における「ＨＭＤ」への任意の言及は、用語「ディスプレイ」または「ディスプレイシステム」で置き換えられ得ることを理解されたい。いくつかの例では、ＨＭＤ１０２は、ＶＲゲームシステムで使用するためなど、ＶＲシステムで使用するためのＶＲヘッドセットを表し得る。しかしながら、ＨＭＤ１０２は、追加的に、または代替的に、ＡＲアプリケーションで使用するためのＡＲヘッドセット、またはゲーム関連ではない（例えば、産

業アプリケーション）ＶＲおよび／またはＡＲアプリケーションで使用可能なヘッドセットとして実装され得る。ＡＲでは、ユーザ１０４は、実世界環境上に重ね合わされた仮想オブジェクトを見るが、ＶＲでは、ユーザ１０４は、典型的には実世界環境を見ることはなく、ＨＭＤ１０２のディスプレイパネルおよび光学系（例えば、レンズ）を介して知覚されるような仮想環境に完全に没入する。一部のＶＲシステムでは、ユーザ１０４の現実世界の環境のパススルー画像は、仮想画像と併せて表示されて、ＶＲシステム内に拡張ＶＲ環境を作成し得、それによって、ＶＲ環境は、現実世界の画像で拡張される（例えば、仮想世界上にオーバーレイされる）ことを理解されたい。本明細書で説明する例は、主にＶＲベースのＨＭＤ１０２に関係するが、ＨＭＤ１０２は、ＶＲアプリケーションでの実装形態に限定されないことを理解されたい。

【００２１】

さらに、ＨＭＤ１０２は、ディスプレイパネルのステレオ対のうちの左ディスプレイパネルおよび右ディスプレイパネルなどの、単一のディスプレイパネルまたは複数のディスプレイパネルを含み得る。ＨＭＤ１０２の１つ以上のディスプレイパネルは、ＨＭＤ１０２を装着しているユーザ１０４によって視認可能である一連の画像フレーム（本明細書では、「フレーム」と称される）を提示するために使用され得る。ＨＭＤ１０２は、任意の数のディスプレイパネル（例えば、３つ以上のディスプレイパネル、一対のディスプレイパネル、または単一のディスプレイパネル）を含み得ることを理解されたい。したがって、「ディスプレイパネル」という用語は、本明細書で単数形で使用される際、２パネルＨＭＤ１０２の一対のディスプレイパネルのいずれかのディスプレイパネルを指し得るか、または任意の数のディスプレイパネル（例えば、単一パネルＨＭＤ１０２またはマルチパネルＨＭＤ１０２）を有するＨＭＤ１０２の単一のディスプレイパネルを指し得る。２パネルＨＭＤ１０２では、ステレオフレームバッファは、例えば、ＨＭＤ１０２の両方のディスプレイパネル上で２１６０×１２００ピクセル（例えば、ディスプレイパネル当たり１０８０×１２００ピクセル）をレンダリングし得る。

【００２２】

さらに、ＨＭＤ１０２のディスプレイパネルは、ディスプレイパネル上でのフレームの提示中に発光素子（例えば、発光ダイオード（ＬＥＤ））を利用して光を放出する発光ディスプレイなど、任意の好適なタイプのディスプレイ技術を利用し得る。例として、ＨＭＤ１０２のディスプレイパネルは、液晶ディスプレイ（ＬＣＤ）、有機発光ダイオード（ＯＬＥＤ）ディスプレイ、無機発光ダイオード（ＩＬＥＤ）ディスプレイ、またはＨＭＤアプリケーションのための任意の他の好適なタイプのディスプレイ技術を含み得る。

【００２３】

ＨＭＤ１０２のディスプレイパネルは、固定リフレッシュレートまたはリフレッシュレートの範囲にわたって動的に変化する可変リフレッシュレートであり得る９０ヘルツ（Ｈｚ）リフレッシュレートなどの任意の好適なリフレッシュレートで動作し得る。ディスプレイの「リフレッシュレート」は、ディスプレイが画面を再描画する１秒当たりの回数である。１秒当たりに表示されるフレームの数は、固定リフレッシュレートをした場合、ディスプレイのリフレッシュレートによって制限され得る。したがって、一連のフレームは、一連のフレームのうちの１つのフレームが画面の更新ごとに表示されるように、処理され（例えば、レンダリングされ）、ディスプレイ上に画像として表示され得る。すなわち、ＨＭＤ１０２上に一連の画像を提示するために、ＨＭＤ１０２のディスプレイパネルは、一連のフレーム内で、ディスプレイのリフレッシュレートでフレーム間に遷移し得、画面のリフレッシュごとにピクセルを点灯させる。いくつかの実施形態では、フレームレートを抑制することができ、および／またはアプリケーションがターゲットフレームレートに到達することができず、および／またはネットワークの混雑は、データ送信にレイテンシをもたらし得る。これらのシナリオでは、再投影フレーム（本明細書では「幻のフレーム」と称されることもある）は、アプリケーションレンダリングされたフレーム（本明細書では「実際のフレーム」と称されることもある）の間に挿入することができる。

【００２４】

概して、ホストコンピュータ１０６上で実行するアプリケーションは、グラフィックスベースのアプリケーション（例えば、ビデオゲーム）であり得る。アプリケーションは、ＨＭＤ１０２のディスプレイパネル上に画像を提示するために使用され得る一連のフレームを出力するように構成される。例えば、アプリケーションは、一連のフレームに関するピクセルデータを生成し得、ピクセルデータは、ＨＭＤ１０２のディスプレイパネル上に対応する画像を提示するために使用される。いくつかの実施形態では、フレームがＨＭＤ１００のディスプレイパネル上にレンダリングされる前にターゲットにレンダリングされ得るように、画面外レンダリングが利用される。したがって、「レンダリング」は、本明細書で使用される際、ディスプレイ以外のターゲットへのレンダリングおよびディスプレイ自体へのレンダリングの前、および／またはディスプレイへのレンダリング（例えば、異なるターゲットへの画面外レンダリングを伴わずに、またはその後に）を含むことができる。

【００２５】

ここでプロセス１００を参照すると、１１２において、ホストコンピュータ１０６のロジック（例えば、ソフトウェア、ハードウェア、および／またはファームウェアなど）は、ホストコンピュータ１０６上で実行されているアプリケーションから、一連のフレームの第１のフレームのピクセルデータを取得（または受信）し得る。ピクセルデータは、ＨＭＤ１０２のディスプレイパネルのピクセルアレイ内の個々のピクセルのためのピクセル値を含み得る。実行中のアプリケーションから取得されるピクセルデータはいくつかの実施形態では、ピクセルごとの値（例えば、色値）の二次元アレイを含み得る。いくつかの実施形態では、ピクセルデータは、深度値、輝度値などの追加のデータまたはメタデータをさらに含む。いくつかの実施形態では、ピクセルデータは、色およびアルファ値の単一のセットによって表される各ピクセルのデータを含み得る（例えば、赤チャネルの１つの色値、緑チャネルの１つの色値、青チャネルの１つの色値、および１つ以上のアルファチャネルの１つ以上の値）。

【００２６】

いくつかの実施形態では、ホストコンピュータ１０６は、まず、ＨＭＤ１０２から頭部追跡データを受信してもよく、これは、フレームについてのピクセルデータを生成するためのＨＭＤ１０２の予測姿勢を決定するために使用されてもよい。すなわち、ホストコンピュータ１０６は、ＨＭＤ１０２から受信した頭部追跡データを使用して、アプリケーションによってレンダリングされることになるフレームに対してＨＭＤ１０２のディスプレイパネルの発光素子が点灯する時間にＨＭＤ１０２があるであろう予測姿勢を示す姿勢データを生成し得る。例えば、ＨＭＤ１０２の頭部追跡システムは、ＨＭＤ１０２の最大６自由度（例えば、三次元（３Ｄ）位置、ロール、ピッチ、およびヨー）を追跡するように構成され得、これは、ＨＭＤ１０２の予測姿勢（例えば、ＨＭＤ１０２の将来の姿勢に起因する予測頭部移動を考慮する）を決定するために、頭部追跡データとしてホストコンピュータ１０６に送信され得る。したがって、予測姿勢を示す姿勢データは、フレームをレンダリングするためのアプリケーションへの入力として提供され得、アプリケーションは、姿勢データに基づいてピクセルデータを出力し得る。例えば、アプリケーションは、姿勢データを受信するための関数を呼び出し得、要求された姿勢データ（フレームの目標点灯時間に予測される）は、アプリケーションが姿勢データに従ってフレームをレンダリングすることができるようにアプリケーションに提供され得、これは、シーンをレンダリングするために使用される仮想カメラの姿勢に対応する。

【００２７】

いくつかの実施形態では、アプリケーションは、フレームおよび／または余分なピクセルデータ（本明細書では「境界外ピクセルデータ」または「追加のピクセルデータ」と称されることがある）のための深度データ（例えば、Ｚバッファデータ）を生成するようにさらに指示され得、それに応答して、ブロック１１２において、ホストコンピュータ１０６のロジックは、上述したピクセルデータに加えて、アプリケーションから深度データおよび／またはフレームに関連付けられている余分なピクセルデータを取得し得る。アプリ

ケーションによって出力される深度バッファ（またはＺバッファ）からなどの深度データは、シーン内の閉塞されたオブジェクトを示し得る。したがって、深度データは、とりわけ、シーン内のオブジェクトの視差を調整するために使用され得る（例えば、世界空間で遠く離れている船は、クローズアップオブジェクトが同じ頭部の移動で動くほど頭部の移動で動かない場合がある）。シーン内の仮想オブジェクトに対応するピクセルの深度を知ることは、ＨＭＤ１０２への再投影中にそのような視差を調整する方法を知ることに役立つ。深度情報を知ることはまた、ＨＭＤ１０２の回転に基づくだけでなく、空間内のＨＭＤ１０２の並進に基づいて、シーンを歪めることも可能にする。いかなる深度データも伴わずに、システムは、約２メートルの平均深度を仮定し得る。しかしながら、シーン内の平均深度のより良い推定値を提供する任意の好適な深度の解像度は、ＨＭＤ１０２上で再投影を実行する際に有益であり得、利用可能な帯域幅に含まれ得る深度の解像度が高いほど、ＨＭＤ１０２上でより良い再投影調整を行うことができる。とはいっても、低解像度の深度データを送信することで、低レイテンシなどの利点がある場合もある。深度データの解像度がＨＭＤ１０２上の再投影調整を改善するのに好適である限り、深度データを含むことが有益であり得る。

【００２８】

アプリケーションによって出力される余分なピクセルデータは、ＨＭＤ１０２のディスプレイパネルのピクセルのアレイの境界の外側に余分なピクセル値を含み得る。例えば、ＨＭＤ１０２のディスプレイパネルが２１６０×１２００ピクセルのアレイを有する場合、ブロック１１２において取得されたピクセルデータは、ピクセルの２１６０×１２００アレイ内のピクセル値に対応し得、一方で、余分なピクセルデータは、２１６０×１２００アレイの境界の外側にあるピクセルに対応し得る。したがって、ピクセルデータおよび余分なピクセルデータは、例えば、２４００×１４００ピクセルのより大きなアレイなどのより多くの数のピクセル値を構成し得る。ターゲットよりも大きいビューでレンダリングする能力は、視野（ＦＯＶ）のいかなる効果的な減少も伴わずに、ディスプレイ領域の周辺の周りにパネルマスクを適用することを可能にし得る。

【００２９】

１１４において、ホストコンピュータ１０６のロジックは、ブロック１１２においてアプリケーションから取得されたピクセルデータを圧縮し得る。ブロック１１４においてピクセルデータを圧縮するための例示的な理由は、ホストコンピュータ１０６とＨＭＤ１０２との間のデータ接続の帯域幅の制限に起因する。例えば、所望のフレームレート（例えば、９０Ｈｚ以上のフレームレート）でフレームを表示するために割り当てられた時間量で非圧縮ピクセルデータ（例えば、ファイル）を送信するのに十分な帯域幅がないことが多い。圧縮は、（ピクセルデータを圧縮しないことと比較して）エンドツーエンドのグラフィックスパイプラインにある程度のレイテンシを導入するが、ピクセルデータを圧縮するためのこの追加の時間は、姿勢予測に考慮され得、圧縮のためにもたらされたレイテンシを考慮するためにＨＭＤ１０２上で調整が行われ得る。

【００３０】

いくつかの実施形態では、ブロック１１４において実行される圧縮は、ピクセルデータのフォーマットを変更することを伴う。例えば、ブロック１１４における圧縮は、Ｈ．２６５および／またはＭＰＥＧ－Ｈパート２と称されることもある、高効率ビデオコーディング（ＨＥＶＣ）および／またはＨＥＶＣの拡張などのビデオ圧縮規格を利用し得る。ＨＥＶＣは、ブロック１１４において利用され得るビデオ圧縮規格の例であるが、ブロック１１４においてピクセルデータを圧縮するために、他の好適なビデオ圧縮規格を利用できることを理解されたい。ＨＥＶＣは、ピクセルデータを圧縮するために動き推定を利用し、帯域幅の制約が存在するシステムにおいて非圧縮ピクセルデータよりも少ないデータを送信し、かつ圧縮データが受信側（例えば、ＨＭＤ１０２において）で元の非圧縮データを近似するのに十分であるという目標を有する。ブロック１１４における圧縮動作において利用される動き推定の一部として、動きベクトルアレイが生成され得、これは、サブブロック１１６において示される。概して、「動きベクトル」は、方向のためのＸおよびＹ

成分、ならびに大きさ（通常は、２Ｄ矢印の長さによって表される）を有する二次元（２Ｄ）矢印である。動きベクトルの大きさは、ＸおよびＹ成分方向の両方でのピクセル数など、任意の好適な測定単位で指定され得る。いくつかの例では、ホストコンピュータ１０６のＧＰＵのビデオエンコーダ（例えば、ビデオ符号化チップ）は、ブロック１１２において取得された第１のピクセルデータに基づいて、および１つ以上の他のフレーム（例えば、以前にレンダリングされたフレーム）のピクセルデータに基づいて、サブブロック１１６において動きベクトルアレイを生成し得る。サブブロック１１６において動きベクトルアレイを生成するために、ビデオエンコーダは、圧縮アルゴリズムに従って、ビデオエンコーダが分析することを課せられる各フレームのピクセルデータ間でピクセルごとの値（例えば、輝度値）を比較し得る。追加的に、または代替的に、ビデオエンコーダは、圧縮アルゴリズムに従って、マクロブロック（例えば、１６ピクセルのブロック（すなわち、４×４ピクセルのマクロブロック）、６４ピクセルのブロック（すなわち、８×８ピクセルのマクロブロック）、４０９６ピクセルのブロック（すなわち、６４×６４ピクセルのマクロブロックなど））を、ビデオエンコーダが分析することを課される各フレームのピクセルデータ間で比較し得る。ホストコンピュータ１０６のＧＰＵは、圧縮動作の一部として、任意の好適な解像度で一対のフレーム間のピクセルデータの部分を比較し得る。

【００３１】

ＨＥＶＣを含む多くのビデオ圧縮アルゴリズムでは、Ｉフレーム、Ｐフレーム、およびＢフレームなどの異なるタイプのフレームがあり得る。Ｉフレームは、解凍中にフレームを再構築するのに十分なデータを含むという意味で、自立することができる。このため、Ｉフレームは、ＰフレームおよびＢフレームと比較して、比較的大量のデータを含み、最も圧縮性が低い。Ｐフレームは、元のピクセルデータを圧縮し、次いで解凍し、再構築するために以前のフレーム（例えば、参照フレーム）からのデータを使用する。Ｐフレームは、動きベクトルの形態など、前のフレームとの差を符号化するため、Ｐフレームは、Ｉフレームと比較して相対的に小さい。Ｂフレームの「Ｂ」は双方向を表し、これは、Ｂフレームが元のピクセルデータを圧縮し、次いで解凍し、そして再構築するために、先行するフレームと将来のフレームとの両方を使用することを意味する。Ｉフレームは相対的大きいため、Ｉフレームを送信するインスタンスは、本開示において減少または最小化され得る。しかしながら、いくつかのシナリオでは、ホストコンピュータ１０６とＨＭＤ１０２との間で無線接続が一時的に失われ、その後再確立される場合など、Ｉフレームを送信することが有益である場合がある。いずれの場合でも、共通の一連のフレームは、Ｉフレームから始まり、一連のＰフレームが続くことがある。ブロック１１４におけるＰフレームの圧縮は、画像のブロックが参照フレーム内のそれらの位置に対して新しいフレーム内で移動した場所に関して、参照フレーム（例えば、Ｉフレーム、および／または先行するＰフレームなど）との差を符号化することを伴い得る。これらの相違は、サブブロック１１６において生成された動きベクトルによって表され得る。

【００３２】

さらに、１１４において、図６のプロセス６００を参照してより詳細に説明されるように、ホストコンピュータ１０６のロジックは、それらのフレームに関するピクセルデータがホストコンピュータ１０６のＧＰＵへの入力として提供される前に、圧縮に使用されるフレームを位置合わせし得る（または別様に修正し得る）。これは、複数の連続したフレームをレンダリングする過程でオブジェクト（静止しているオブジェクトおよび移動しているオブジェクトの両方）が位置間を移動させられ得る、フレーム間のＨＭＤ１０２の移動のために行うことができる。一方のフレームを他方のフレームに位置合わせすることによって、またはその逆によって、または両方のフレームを互いに位置合わせするように調整することによって、シーン内の特定の静止しているオブジェクトを表すピクセルデータを、静止しているオブジェクトに対応するピクセルデータが、ＧＰＵのビデオエンコーダによって移動しているオブジェクトと誤認されないように、２つのフレーム間の概ね同じ位置に移動させることができる。ブロック１１４において実行される位置合わせは、頭部の移動に起因する動きベクトル内の差分（または差）を減少させるために、フレーム（例

えば、フレームのピクセルデータ）のうちの一方または両方に対する回転再投影修正を含み得る。動きベクトルの差分（または差）のこの減少は、本明細書に記載するように、ホストコンピュータ１０６からＨＭＤ１０２に動きベクトルを送信する際の帯域幅の減少の追加の利点を提供し得る。したがって、修正されたピクセルデータは、サブブロック１１６において動きベクトルを生成するときに、ブロック１１４における圧縮のためのＧＰＵへの入力として提供され得る。

【００３３】

１１８において、ホストコンピュータ１０６は、圧縮されたピクセルデータ１１０（１）を含み得るデータ１１０をＨＭＤ１０２に送信し得る。Ｐフレームに関して上述したように、圧縮されたピクセルデータ１１０（１）のほとんどの時間は、サブブロック１１６において生成された動きベクトル１１０（２）のアレイの形態で送信され得る。すなわち、一連のＰフレームについて、動きベクトルアレイ１１０（２）は、圧縮されたピクセルデータを表し得る。しかしながら、ＩフレームがＨＭＤ１０２に送信される場合、圧縮されたピクセルデータ１１０（１）は、動きベクトルを含まない場合がある。ブロック１１８において送信されたデータ１１０は、フレームのピクセルデータ１１０（１）を生成するためにアプリケーションによって使用される姿勢データ１１０（３）をさらに含み得る。いくつかの実施形態では、ブロック１１８において送信されたデータ１１０は、姿勢データ１１０（３）を省略し得、および／またはデータ１１０は、深度データ１１０（４）、余分なピクセルデータ（例えば、ＨＭＤ１０２のディスプレイパネルの境界の外側にある）、視差閉塞データ、および／またはキューブマップデータ（例えば、ＨＭＤ１０２が、いかなるデータも有さない暗いピクセルを提示すること以外の他の選択肢を有するように、急激かつ大規模な頭部移動のためのもの）などの追加データを含み得る。データ１１０は、ブロック１１８において、無線で、有線接続を介して、広域ネットワークなどを介してなど、実装形態に応じて様々な方式でＨＭＤ１０２に送信され得る。いくつかの実施形態では、データ１１０の一部または全部は、圧縮されたピクセルデータ１１０（１）、例えば、動きベクトル１１０（２）とともに、符号化されたデータストリームの一部として、帯域内または帯域外に送信され得る。例えば、姿勢データ１１０（３）、深度データ１１０（４）、および／または追加データは、動きベクトル１１０（２）などの圧縮されたピクセルデータ１１０（１）から帯域外に送信され得る。

【００３４】

１２０において、ＨＭＤ１０２は、限定されないが、動きベクトル１１０（２）（受信時に、圧縮されたピクセルデータを表す）、姿勢データ１１０（３）、および／または深度データ１１０（４）などの、圧縮されたピクセルデータ１１０（１）を含むデータ１１０をホストコンピュータ１０６から受信し得る。

【００３５】

１２２において、ＨＭＤ１０２のロジック（例えば、ソフトウェア、ハードウェア、および／またはファームウェアなど）は、例えば、動きベクトル１１０（２）および参照フレーム（例えば、以前のフレームのピクセルデータ１１０（１））を使用することによってなど、圧縮されたピクセルデータ１１０（１）を解凍して、アプリケーションによって出力されるフレームのピクセルデータ、または少なくとも元のピクセルデータの近似を取得し得る。ブロック１２２において圧縮されたピクセルデータを解凍することは、ＨＥＶＣアルゴリズムなどの解凍または復号アルゴリズムを使用し得る。さらに、サブブロック１２４によって示されるように、例えば、少なくとも動きベクトル１１０（２）が圧縮されたピクセルデータ１１０（１）を表す場合（例えば、Ｐフレームに対して）、ＨＭＤ１０２のロジックは、例えば、符号化されたデータストリーム（例えば、ＨＥＶＣストリーム）からなど、データストリームから動きベクトル１１０（２）を抽出し得る。解凍されたピクセルデータ１１０（１）、動きベクトル１１０（２）、および場合によっては追加のデータ（例えば、姿勢データ１１０（３）、深度データ１１０（４）など）は、ＨＭＤ１０２のメモリにキャッシュされ得、その結果、追加のデータは、再投影フレームをレンダリングする際、および／または将来のデータを解凍する際に使用するためなど、後の時

点でアクセスされ得る。ブロック１２０～１２４は、フレームがホストコンピュータ１０６から継続的に受信されるにつれて反復し得ることを理解されたい。データ１１０は、ある期間ＨＭＤ１０２のメモリ内に維持（例えば、キャッシュ）され得、その後、データ１１０は、将来のデータ１１０を記憶するための空間を作るために破棄され得ることを理解されたい。

【００３６】

１２６において、ＨＭＤ１０２のロジックは、ＨＭＤ１０２のディスプレイパネル上に画像を提示する際に使用するために、解凍されたピクセルデータ１１０（１）などの最新の解凍データ１１０を取り出し得る。例えば、データ１１０がブロック１２０において受信され、ブロック１２２において解凍されると、ブロック１２６における取り出し動作は、ＨＭＤ１０２のメモリに記憶されているキャッシュされたピクセルデータ１１０（１）が、ＨＭＤ１０に利用可能な最後に解凍されたピクセルデータを表すと決定し得、ＨＭＤ１０２のロジックは、最後に解凍されたデータ、例えば、最後に受信された動きベクトル１１０（２）を解凍することから取得されたピクセルデータ１１０（１）をメモリから取り出し得る。用例では、ＨＭＤ１０２が９０Ｈｚのリフレッシュレートで動作している場合、フレームは、およそ１１．１１ミリ秒ごとに１回表示されることになる。この文脈では、ＨＭＤ１０２が最後のフレームが表示されてからの最後の１１．１１ミリ秒で新しいデータを受信および解凍した場合、新しい解凍されたピクセルデータ１１０（１）は、次のフレームをレンダリングするために使用され得る。一方、最後のフレームが表示されてからの最後の１１．１１ミリ秒間にＨＭＤ１０２が新しいデータを受信および解凍していない場合、最後にレンダリングされたフレームのピクセルデータ１１０（１）を次のフレームのレンダリングに使用してもよい。

【００３７】

１２８において、ＨＭＤ１０２のロジックは、ＨＭＤ１０２の予測姿勢に少なくとも部分的に基づいて、フレームのピクセルデータを修正して、フレームの修正されたピクセルデータを取得し得る。例えば、ＨＭＤ１０２のロジックは、ブロック１２６において取り出されたピクセルデータを生成するためにアプリケーションによって使用されたＨＭＤ１０２の元の予測姿勢と、フレームの点灯時間により近い時間にＨＭＤ１０２のロジックによって予測された更新された姿勢との比較に基づいて、ピクセルデータに調整を適用し得る。この比較は、アプリケーションがフレームをレンダリングした時点での元の姿勢予測と、ディスプレイパネル上にフレームをレンダリングする前のＨＭＤ１０２における更新された姿勢予測との間の差分（または差）を明らかにし得、ブロック１１８において適用される調整は、この差分を補償するために（例えば、２つの姿勢決定の間の差分に応じて、ピクセルデータをある方向にもしくは別の方向にシフトおよび／または回転させることによって）回転計算を含み得る。

【００３８】

さらに、１２８において、ＨＭＤ１０２のロジックはまた、ブロック１２４において抽出され、ブロック１２６において取り出された動きベクトル１１０（２）のアレイに少なくとも部分的に基づいて、フレームのピクセルデータを修正し得る。ブロック１２８における修正の結果、再投影フレームの修正されたピクセルデータが取得される。本開示は、「実際の」フレームと再投影（または、「幻の」フレーム）との間の区別を行うが、この区別は、ＨＭＤ１０２上での提示の前に、実際のフレームがしばしばブロック１２８で調整／修正されないことを暗示することを意図しない。すなわち、ＨＭＤ上で提示されたフレームは、「実際の」フレームと「幻の」フレームとの両方でブロック１２８において実行され得る修正動作に起因して、合成されている（すなわち、ホストコンピュータ１０６上で実行されているアプリケーションによって出力された元のフレームと同じではない）と考えることができる。この意味で、ブロック１２８において修正されるすべてのフレームは、本明細書で使用されるように、「再投影」フレームであると考えることができる。ピクセルデータが修正されるかどうか、および／またはブロック１２８において修正される程度は、システムの動作性能に依存する。例えば、アプリケーションによって出力され

るすべてのフレームがＨＭＤ１０２においてタイムリーに受信されるライブレンダリングシナリオにおいて、すべてがそれと同じように進む場合、元の姿勢予測と更新された姿勢予測との間の差分は、ゼロではないにしても、それに近いものとなり、この場合、ブロック１２８において適用される修正は、もしあれば、最終的な出力に影響をほとんどまたはまったく及ぼさない場合がある。しかしながら、アプリケーションがフレームレートに到達できなかった、またはネットワーク渋滞に起因してフレームが遅れて到着するか、もしくは転送中にドロップされるシナリオでは、ブロック１２８においてピクセルデータ１１０（１）に適用される修正は、元の姿勢予測と更新された姿勢予測との間のより大きな差分を補償するために有意であり得る。以下でより詳細に説明するように、動きベクトル１１０（２）に基づく再投影フレームのピクセルデータの修正は、レンダリングメッシュを利用し得る。例えば、レンダリングメッシュは、再投影フレームに対して生成され得る。レンダリングメッシュは、複数の頂点を有するテッセレーションメッシュを含み得、レンダリングメッシュの頂点は、移動された頂点としてレンダリングメッシュ内の異なる位置に移動され得る（例えば、頂点を（ｉ）非ゼロ動きベクトル１１０（２）の方向に移動させることによって、および（ｉｉ）非ゼロ動きベクトル１１０（２）の大きさに対応する量だけ移動させることによって）。したがって、ブロック１２８におけるピクセルデータの修正は、再投影フレームの修正されたピクセルデータを取り出すために、ブロック１２６において取得されたピクセルデータのピクセル値を移動させることなどによって、レンダリングメッシュの移動された頂点に従って行われ得る。例えば、ピクセル値は、レンダリングメッシュ内の移動された頂点に従って、修正されたピクセルデータ内の新しい位置に、左へ４ピクセル、および上へ４ピクセル移動され得る。いくつかの実施形態では、修正された動きベクトル１１０（２）をピクセルデータに適用する前に、動きベクトル１１０（２）が修正され得る。例えば、以下でより詳細に説明されるように、フィルタ（例えば、Ｎ×Ｎスカラーメディアンフィルタ、Ｍ×Ｍ平均最大ぼかしフィルタなど）を動きベクトルアレイ１１０（２）に適用し得、動きベクトルの修正されたアレイを取得し得、動きベクトルの修正された（例えば、フィルタリングされた）アレイを使用して、ブロック１２８においてピクセルデータを修正し得る。

【００３９】

さらに、１２８において、ＨＭＤ１０２のロジックはまた、ブロック１２６において取り出された深度データ１１０（４）に少なくとも部分的に基づいて、フレームのピクセルデータを修正し得る。深度データ１１０（４）は、ブロック１２８において、とりわけ、シーン内のオブジェクトの視差を調整するために使用され得る（例えば、世界空間で遠く離れている船は、クローズアップオブジェクトが同じ頭部の移動で動くほど頭部の移動で動かない場合がある）。シーン内の仮想オブジェクトに対応するピクセルの深度を知ることは、ＨＭＤ１０２への再投影中にそのような視差を調整する方法を知ることに役立つ。深度情報を知ることはまた、ＨＭＤ１０２の回転だけでなく、空間内のＨＭＤ１０２の並進に基づいて、シーンを歪めることも可能にする。深度データ１１０（４）がホストコンピュータ１０６から受信されない場合、ＨＭＤ１０２のロジックは、約２メートルの平均深度を想定し得る。

【００４０】

１３０において、画像は、修正されたピクセルデータに少なくとも部分的に基づいて、ＨＭＤ１０２のディスプレイパネル上に提示され得る。例えば、ＨＭＤ１０２のロジックは、修正されたピクセルデータをフレームバッファに出力し得、フレームバッファに出力された修正されたピクセルデータに基づいて、画像をＨＭＤ１０２のディスプレイパネルに提示させ得る。これは、修正されたピクセルデータをＨＭＤ１０２のディスプレイパネルにスキャンアウトし、ディスプレイパネルの発光素子を点灯させて、ディスプレイパネル上のピクセルを点灯させることを伴い得る。一対のディスプレイパネルを有するＨＭＤ１０２に関して、この修正されたピクセルデータは、一対のディスプレイパネル上に表示されることになる一対の画像を表すフレームに対応し得、それに応じて、ステレオフレームバッファに出力され、スキャンアウトされ得る。再投影フレームに対応する結果として

生じる画像は、動きベクトル１１０（２）に従ってピクセルデータの修正によって「動き平滑化」される。

【００４１】

図２は、本明細書に開示される実施形態による、２つの例示的なタイムライン２００（１）および２００（２）を示す図であり、ホストコンピュータ１０６およびＨＭＤ１０２のそれぞれのレンダリングワークロードを示している。図２の例は、ホストコンピュータ１０６と関連付けられた第１のタイムライン２００（１）に関して、フレーム「Ｆ」、フレーム「Ｆ＋１」、およびフレーム「Ｆ＋２」の３つの例示的なフレームを描いている。この第１のタイムライン２００（１）は、ホストコンピュータ１０６のＧＰＵを使用して、ホストコンピュータ１０６上で実行されるアプリケーションによって、フレームをどのように直列にレンダリングすることができるかを示す。ここで、アプリケーションは、第１のタイムライン２００（１）上で左から右に、順番に、フレームＦ、次いで、フレームＦ＋１、さらに次いで、フレームＦ＋２をレンダリングする。第１のタイムライン２００（１）上の楕円は、アプリケーションが実行を続けるにつれて、任意の数のフレームに対してこれが継続され得ることを示す。第１のタイムライン２００（１）はまた、水平タイムライン２００（１）に直交するように向けられた垂直線によって、アプリケーションがターゲットフレームレート（例えば、垂直線が約１１．１１ミリ秒だけ分離される９０Ｈｚのフレームレート）をターゲットにしていることを暗示する。図２の例では、ホストコンピュータ１０６上で実行されているアプリケーションは、たまたまフレームＦおよびＦ＋１上でターゲットフレームレートに到達しているが、アプリケーションはフレームＦ＋２のフレームターゲットフレームレートに到達しない。例えば、フレームＦ＋２のシーンが、多数の移動しているオブジェクトもしくは複雑なテクスチャを含む場合があり、これらの複雑さ、および／もしくは他の理由により、アプリケーションは、フレームＦ＋２をレンダリングするために割り当てられた時間よりも長い時間を要する。ホストコンピュータ１０６は、ＨＭＤ１０２の移動に関する頭部追跡データ２０８を受信して、タイムライン２００（１）上の各フレームをレンダリングするためのＨＭＤ１０２の予測姿勢を決定し得る。

【００４２】

ＨＭＤ１０２に関連付けられている図２の第２のタイムライン２００（２）は、個々のフレームのＨＭＤ１０２のレンダリングワークロード２０２（ａ）、２０２（ｂ）、および２０２（ｃ）を示している。所与のフレームのＨＭＤ１０２の個々のレンダリングワークロード２０２は、ＨＭＤ１０２のディスプレイパネル上に最終的な画像が提示される前に、ピクセルデータ１１０（１）に適用される調整を表し得る。そのような調整は、ＨＭＤ１０２上に最終画像をレンダリングする前に、ピクセルデータ１１０（１）に適用される、幾何学的歪み、色収差、頭部移動などのための調整を含み得るが、これらに限定されない。フレームＦおよびＦ＋１について、これらの調整は、ホストコンピュータ１０６上で実行されるアプリケーションによって生成されたピクセルデータに適用され、これらの調整のうちの少なくともいくつかは、ＨＭＤ１０２の元の予測姿勢と、ＨＭＤ１０２の更新された姿勢の予測との間の差分を考慮することによって、ホストコンピュータ１０６から受信された姿勢データ１１０（３）を利用し得る。したがって、第２のタイムライン２００（２）のフレームＦおよびＦ＋１は、ライブレンダリング状況中にアプリケーションからリアルタイムで出力されるピクセルデータ１１０（１）の修正バージョンであるという意味で「実際の」フレームの修正バージョンを表すことを意味する。対照的に、フレームＦ＋２をレンダリングするために、ＨＭＤ１０２は、先行するフレームの姿勢予測およびＨＭＤ１０２によって行われた更新された姿勢予測に基づいて再投影（または「幻の」）フレームを生成するために、先行するフレームについて以前に受信されたピクセルデータ１１０（１）（例えば、フレームＦ＋１のピクセルデータ１１０（１））を使用し得る。さらに、レンダリングワークロード２０２（ｃ）は、フレームＦ＋１のデータ１１０とともに受信した動きベクトル１１０（２）に基づいて、フレームＦ＋１からのピクセルデータを修正して、本明細書に記載するように、フレームＦ＋２を「動き平滑化」すること

を含み得る。いずれにせよ、レンダリングワークロード２０２の結果は、フレームバッファ（例えば、ステレオフレームバッファ）に出力され得る修正されたピクセルデータの生成である。本明細書において、「実際の」フレームと「幻の」フレームとの間のこの区別は、実際のフレームがＨＭＤ１０２上で調整されないことを暗示することを意味するものではなく、この意味で、ＨＭＤ側で生成されたフレームはすべて、効果的に合成される（すなわち、ホストコンピュータ１０６上で実行されるアプリケーションによって出力される元のフレームと同じではない）。

【００４３】

図２の第２のタイムライン２００（２）はまた、各フレームのスキャンアウト時間２０４（ａ）、２０４（ｂ）、および２０４（ｃ）、ならびに各フレームの点灯時間２０６（ａ）、２０６（ｂ）、および２０６（ｃ）を示す。所与のフレームのためのスキャンアウト時間２０４の間に、（修正されたピクセルデータの）ピクセル値のサブセットは、ディスプレイポート（例えば、高解像度マルチメディアインターフェース（ＨＤＭＩ）（登録商標））を介してＨＭＤ１０２のディスプレイパネルにスキャンアウトされ、所与のフレームのための点灯時間２０６の間に、ＨＭＤ１０２のディスプレイパネルの発光素子は、ディスプレイパネルのピクセルを点灯させるために点灯される。図２は、ＬＣＤパネルとともに使用されて、ＨＭＤ１０２のリフレッシュレートでディスプレイパネルの発光素子から光を同時に放出し得る、グローバルフラッシュタイプのディスプレイ駆動方式の例を示している。用例では、ＨＭＤ１０２が９０Ｈｚのリフレッシュレートで動作している場合、各フレームの点灯時間２０６は、およそ１１．１１ミリ秒で分離され得る。

【００４４】

図３は、再投影がＨＭＤ１０２などのディスプレイ上のフレームをレンダリングするために使用されているときに、シーン内で移動しているかまたは動いているオブジェクトを考慮するための例示的な動き平滑化技術を示す図である。図３の例は、一連のフレーム３００として順番にレンダリングされることになる３つの例示的なフレーム３００（１）、３００（２）、および３００（３）を描いている。図３の例では、オブジェクト３０２は、フレーム３００（１）～３００（３）の過程にわたって左方向（すなわち、右から左へ）にシーンを横切って移動していることが示されている。フレーム３００（２）および３００（３）のオブジェクト３０２の点線輪郭は、以前にレンダリングされたフレーム３００にオブジェクト３０２が位置していた場所を表している。ここで、フレーム３００（１）は、最初にレンダリングされ、次いで、フレーム３００（２）は、２番目にレンダリングされ、次いで、フレーム３００（３）は、３番目にレンダリングされる。

【００４５】

一連のフレーム３００のフレーム３００のうちの少なくともいくつかは、それらがビデオゲームアプリケーション、またはフレーム３００のピクセルデータに基づいてＨＭＤ１０２上に対応する画像を提示するのに十分な時間内に任意の他のタイプのグラフィックスベースアプリケーションなどのアプリケーションから出力されるという意味で「実際の」フレームであり得る。アプリケーションは、個々のフレーム３００をレンダリングするためにピクセルデータ１１０（１）をフレームバッファに出力するグラフィックスパイプラインで実行され得る。

【００４６】

実行中、ＨＭＤ１０２の頭部追跡モジュールは、ユーザの１０４の頭部姿勢に従って一連のフレーム３００の次のフレーム３００をどのようにレンダリングするかに関してアプリケーションに知らせるために、アプリケーションを実行するホストコンピュータ１０６に提供されるＨＭＤ１０２の位置および向き（姿勢）についてのデータを生成し得る。これは、ユーザ１０４がオブジェクト（移動オブジェクト３０２などの静止しているオブジェクトと移動しているオブジェクトとの両方）を含む仮想環境を見回しているとユーザ１０４に信じさせる方式で、アプリケーションがＨＭＤ１０２上で画像をレンダリングするためのピクセルデータ１１０（１）を出力することを可能にする。アプリケーションがフレームレートに到達している場合、静止および移動しているオブジェクトの両方が、ユー

ザ１０４の頭部の動きとともに、予想される様式で、シーン内で移動するように知覚される。本明細書に記載される動き平滑化技術は、フレームレートに到達しなかったアプリケーションを補償して、移動しているオブジェクトに関して同様の視覚的知覚が達成されるようにする方式である。

【００４７】

図３の例では、第１のフレーム３００（１）は、アプリケーションから受信される第１の「実際の」フレームを表し得、第２のフレーム３００（２）は、アプリケーションから受信され、第１のフレーム３００（１）の後にレンダリングされる第２の「実際の」フレームを表し得、第３のフレーム３００（３）は、第２のフレーム３００（２）に関連付けられているピクセルデータ１１０（１）（２）から生成される再投影フレームを表し得る。したがって、図３における「第３のフレーム３００（３）」は、本明細書において「再投影フレーム３００（３）」と称されることもある。図３の例では、第２のフレーム３００（２）（フレーム２）を圧縮するために、ホストコンピュータ１０６のロジックは、ホストコンピュータ１０６のグラフィックスプロセッシングユニット（ＧＰＵ）３０４への入力として、第１のフレーム３００（１）に関連付けられている第１のピクセルデータ１１０（１）（１）および第２のフレーム３００（２）に関連付けられている第２のピクセルデータ１１０（１）（２）を提供し得る。

【００４８】

ＧＰＵ３０４のビデオエンコーダ（例えば、ビデオ符号化チップ）は、ＧＰＵ３０４に入力された第１のピクセルデータ１１０（１）（１）および第２のピクセルデータ１１０（１）（２）に基づいて、動きベクトル１１０（２）（２）のアレイを生成し得る。動きベクトル１１０（２）（２）のアレイを生成するために、ＧＰＵ３０４のビデオエンコーダは、入力として提供された各フレーム３００のピクセルデータ１１０（１）間のピクセルごとの値（例えば、輝度値）を比較し得る。追加的に、または代替的に、ＧＰＵ３０４のビデオエンコーダは、入力として提供された各フレーム３００のピクセルデータ１１０（１）の間のマクロブロック（例えば、１６ピクセルのブロック（すなわち、４×４ピクセルのマクロブロック）、６４ピクセルのブロック（すなわち、８×８ピクセルのマクロブロック））を比較し得る。このようにして、ＧＰＵ３０４は、任意の好適な解像度で、一対のフレーム３００間のピクセルデータ１１０（１）の部分を比較し得る。いくつかの実施形態では、入力ピクセルデータ１１０（１）（１）および１１０（１）（２）は、ＧＰＵ３０４にダウンサンプリングされたフレームを入力するために、より低い解像度にダウンサンプリングされる。いくつかの実施形態では、動きベクトル１１０（２）（２）は、ＨＥＶＣアルゴリズムなどの任意の好適な圧縮アルゴリズムに従って生成される。

【００４９】

ＧＰＵ３０４から出力される動きベクトル１１０（２）（２）のアレイは、本明細書では「動きベクトル場」と呼ばれることがある。この動きベクトル場１１０（２）（２）はまた、任意の好適な解像度で出力され得、および／またはダウンサンプリング／アップサンプリングされ得る。例えば、動きベクトルアレイ１１０（２）（２）は、ピクセルごとの単一の動きベクトル、ピクセルのグループごとの単一の動きベクトル（例えば、４×４のマクロブロックについての１つの動きベクトル、８×８のマクロブロック、ピクセルの任意の形状のパッチなど）、または所与のフレーム３００のすべてのピクセルについての単一の動きベクトルを含み得る。

【００５０】

ＧＰＵ３０４のビデオエンコーダによる入力ピクセルデータ１１０（１）の比較に基づいて、第２のフレーム３００（２）の一部分が第１のフレーム３００（１）の一部分に類似している場合（例えば、閾値輝度値内で）、および各フレーム３００内の類似した部分が距離（例えば、ＸおよびＹ成分方向のピクセル数）でオフセットされる場合、これは、動きベクトル１１０（２）（２）のアレイに含まれる動きベクトルによって表すことができる。フレーム３００（１）および３００（２）内のオブジェクト３０２に対応するピクセル値が、ＧＰＵ３０４のビデオエンコーダによって類似していると決定される（例えば

、ピクセルデータ１１０（１）内のピクセル値に基づいて、いくつかの類似性メトリックを満たす一致部分）例を考える。このオブジェクト３０２の動きベクトルは、オブジェクト３０２の移動と同じ方向を指す方向を有し得、または動きベクトルは、オブジェクト３０２の移動の方向と反対方向を指し得る。言い換えれば、動きベクトルは、後続のフレーム３００（２）におけるオブジェクト３０２の位置からオフセットされた、以前のフレーム３００（１）におけるオブジェクト３０２の位置を参照する方向を指し得る。したがって、アレイ１１０（２）（２）内の動きベクトルは、第２のフレーム３００（２）内の座標から第１のフレーム３００（１）内の座標へのオフセットを提供する。オフセットは、オブジェクト３０２などのオブジェクトを移動させるかまたは動かすことに関して、第１のフレーム３００（１）の画像から第２のフレーム３００（２）の画像への変換を説明している。

【００５１】

上述したように、図３の第３のフレーム３００（３）は、再投影フレームを表し得、これは、第３のフレーム３００（３）に関するピクセルデータ１１０（１）（３）が、以前にレンダリングされた実際のフレーム（この場合、第２のフレーム３００（２））に関連付けられているピクセルデータ１１０（１）（２）から導出され得ることを意味する。言い換えれば、再投影フレーム３００（３）のピクセルデータ１１０（１）（３）は、ホストコンピュータ１０６からリアルタイムで受信されるのではなく、アプリケーション生成フレームのピクセルデータ１１０（１）から生成され、アプリケーションがフレームレートに到達していないとき、および／またはネットワーク渋滞が発生したときに、欠落しているフレームの間隔を「埋める」ために使用される。この場合、図３の例では、第２のフレーム３００（２）は、再投影フレーム３００（３）の前に最後にレンダリングされたフレーム３００であるため、再投影フレーム３００（３）に関するピクセルデータ１１０（１）（３）は、第２のフレーム３００（２）に関連付けられたピクセルデータ１１０（１）（２）から生成される。いくつかの実施形態では、回転および再投影変換は、再投影フレーム３００（３）の第３のピクセルデータ１１０（１）（３）を生成するために、第２のフレーム３００（２）に関連付けられている第２のピクセルデータ１１０（１）（２）を修正するために計算および使用され得、第２のフレーム３００（２）がレンダリングされたときからＨＭＤ１０２の回転を考慮するような様式で、第２のフレーム３００（２）内でレンダリングされたシーンを効果的に回転させ、並進させ、および／または他の方法で移動させる。例えば、ユーザ１０４は、第２のフレーム３００（２）がレンダリングされた時点から自身の頭部を回転させ得、これは、シーンがこの頭部の移動に従って提示されるように、再投影フレーム３００（３）の第３のピクセルデータ１１０（１）（３）の生成に考慮される。

【００５２】

次いで、図３の動き平滑化技術は、第２のフレーム３００（２）と関連付けられてホストコンピュータ１０６から受信された動きベクトル１１０（２）（２）のアレイに少なくとも部分的に基づいて、第３のピクセルデータ１１０（１）（３）を修正し、再投影フレーム３００（３）の修正された第３のピクセルデータ１１０（１）（３）’を取得する。いくつかの実施形態では、修正された第３のピクセルデータ１１０（１）（３）への第３のピクセルデータ１１０（１）（３）のこの修正は、特定のピクセルまたはピクセルのグループに対応する非ゼロ動きベクトル１１０（２）（２）に基づいて、第３のピクセルデータ１１０（１）（３）のピクセル値を異なる位置に移動させることを含む。移動は、一定の方向、および一定の量（例えば、水平方向（＋／－）方向および垂直方向（＋／－）方向にいくつかのピクセルを移動させる）であり得る。次いで、再投影フレーム３００（３）は、修正された第３のピクセルデータ１１０（１）（３）’に少なくとも部分的に基づいて、ディスプレイ（例えば、ＨＭＤ１０２のディスプレイパネル）上でレンダリングされる。したがって、再投影フレーム３００（３）は、動きベクトル１１０（２）（２）（動きベクトル１１０（２）（２）は、第２のフレーム３００（２）のピクセルデータ１１０（１）（２）を圧縮する結果（例えば、副産物）として、以前に受信された（および／ま

たは以前にレンダリングされた）実際のフレーム３００（１）および３００（２）のピクセルデータ１１０から生成された）に基づいて修正され、オブジェクト３０２を予想される位置にレンダリングするために「動き平滑化」される。

【００５３】

図４は、レンダリングメッシュ４００の例、ならびに再投影フレームの動き平滑化にレンダリングメッシュ４００をどのように使用することができるかを示す図である。例えば、レンダリングメッシュ４００は、ホストコンピュータ１０６のＧＰＵ３０４によって出力された動きベクトル１１０（２）のアレイに基づいて、図３の再投影フレーム３００（３）のための第３のピクセルデータ１１０（１）（３）を修正するために、ＨＭＤ１０２によって使用され得る。この例では、ＨＭＤ１０２のロジックは、レンダリングメッシュ４００を生成し得、レンダリングメッシュ４００の頂点４０２は、動きベクトル１１０（２）のアレイに従って移動され得る。例えば、頂点４０２は、非ゼロ動きベクトル４０４の方向に、および非ゼロ動きベクトル４０４の大きさに対応する量だけ移動され得る。例えば、図３に示される左方向に移動するオブジェクト３０２を例にとると、動きベクトル４０４は、レンダリングメッシュ４００のコンテキストで適用されて、頂点４０２を特定の数のピクセル（動きベクトル４０４の大きさに対応する）を左方向（または負のＸ方向）に移動させ得る。

【００５４】

レンダリングメッシュ４００は、複数の頂点４０２（１）、４０２（２）、…、４０２（Ｎ）（まとめて４０２）を有するテッセレーションメッシュとして示される。レンダリングメッシュ４００のテッセレーションは、任意の好適な幾何学的パターンであり得る。図４のレンダリングメッシュ４００の例は、三角形４０６の繰り返しパターンとして示されるが、任意の好適な幾何学的形状は、正方形（「四角形」と称されることがある）、六角形（例えば、ハニカムパターンの場合）などを含むが、これらに限定されない、レンダリングメッシュ４００のために使用され得る。この例では、正方形（または四角形）の左下隅から正方形（または四角形）の右上隅までの対角線は、特定の向きを有する繰り返し三角形４０６のレンダリングメッシュ４００を作成するために使用される。レンダリングメッシュ４００の異なる向きは、図４に示される向きの代わりに、正方形（または四角形）の左上隅から正方形（または四角形）の右下隅までの対角線を使用して、正方形（または四角形）を三角形４０６に分割することによって作成され得る。いくつかの実施形態では、これらの異なる向きの混合物はまた、正方形（または四角形）の左下隅から正方形（または四角形）の右上隅までの対角線を使用して他のすべての正方形を分割し、正方形（または四角形）の左上隅から正方形（または四角形）の右下隅までの対角線を使用して間の正方形を分割するなどによって、単一のレンダリングメッシュ４００のために使用され得る。いくつかの実施形態では、ＨＭＤ１０２のロジックは、ホストコンピュータ１０６から受信した動きベクトル場１１０（２）に基づいて、レンダリングメッシュ４００の生成に使用する、これらの複数の向きのうちどの向きを使用するかを動的に決定するように構成され得る。これは、レンダリングメッシュ４００内の幾何学的形状（例えば、三角形４０６）に最良の向きを選択するために行われ得、これは、最も滑らかに見え、動き平滑化された画像をもたらす。

【００５５】

レンダリングメッシュ４００はまた、任意の好適な解像度で生成され得る。例えば、最高解像度のレンダリングメッシュ４００は、ピクセル当たり２つの隣接する三角形４０６であり得、各正方形（または四角形）は、単一のピクセルにマッピングされる。より低い解像度は、１６ピクセルのグループなどのピクセルのグループごとに２つの隣接する三角形４０６であり得る。あるいは、ピクセルは、任意の好適な解像度でレンダリングメッシュ４００の頂点４０２にマッピングされ得る。例えば、各頂点４０２は、最高解像度で単一のピクセルに関連付けられ得、または各頂点４０２は、より低い解像度で、１６ピクセルのグループなどのピクセルのグループに関連付けられ得る。いくつかの実施形態では、レンダリングメッシュ４００の解像度は、動きベクトル１１０（２）のアレイにおける単

一の動きベクトル４０４が頂点４０２または正方形（もしくは四角形）（例えば、２つの隣接する三角形４０６）にマッピングするように、動きベクトル１１０（２）のアレイの解像度と同じ解像度である。レンダリングメッシュ４００と動きベクトルアレイ１１０（２）との間の一致解像度を達成することは、例えば、ＧＰＵ３０４から、レンダリングメッシュ４００の解像度に一致する特定の解像度で動きベクトルアレイ１１０（２）を要求することによって、レンダリングメッシュ４００として解像度に一致するように動きベクトルアレイ１１０（２）をダウンサンプリングもしくはアップサンプリングすることによって、またはホストコンピュータ１０６から受信される動きベクトルアレイ１１０（２）の解像度に一致する解像度でレンダリングメッシュ４００を生成することによってなど、様々な方法で達成され得る。

【００５６】

図４は、４つの非ゼロ動きベクトル４０４（１）、４０４（２）、４０４（３）、および４０４（４）がレンダリングメッシュ４００の４つの頂点４０２に対応する例を示している。これらの４つの動きベクトル４０４（１）～（４）は、図３に描かれる移動しているオブジェクト３０２に基づいてＧＰＵ３０４が検出した動きベクトルに対応し得る。したがって、例示的な動きベクトル４０４（１）～（４）は、オブジェクト３０２の方向性の動きに対応する左方向を指し得るが、上述したように、方向性は、オブジェクト３０２の方向性の動きの方向とは反対（例えば、右方向）であり得る。動きベクトル４０４の方向性は、動き平滑化アルゴリズムにおいて、ピクセルデータ１１０（１）を所望の方向に修正するために考慮され得る。動きベクトル１１０（２）の場が４つの例示的な動きベクトル４０４（１）～（４）を含み、動きベクトル１１０（２）のアレイの残りの動きベクトルのすべてがゼロベクトルである基本的な例を考える。この例では、再投影フレーム３００（３）のピクセルデータ１１０（１）（３）は、頂点４０８（１）、４０８（２）、４０８（３）、および４０８（４）（図４の下部に示される）が移動するにつれて、非ゼロ動きベクトル４０４（１）～（４）に対応する頂点４０２をレンダリングメッシュ４００内の異なる位置に移動させることによって、非ゼロ動きベクトル４０４（１）～（４）に基づいて修正され得る。図４の下部は、動き平滑化が適用された後のレンダリングメッシュ４００を示しており、移動された頂点４０８（１）～（４）は、動き平滑化の前の頂点４０２の位置と比較して、レンダリングメッシュ４００内の異なる位置にある。このようにして動きベクトル４０４（１）～（４）が適用されるとき、移動された頂点４０８（１）～（４）は、レンダリングメッシュ４００内の幾何学的形状（例えば、三角形４０６）のうちの特定のものを引き伸ばすこと、または歪めることなどによって、レンダリングメッシュ４００の１つ以上の部分を歪ませる。図４の例では、図４の下部に示されるように、引き伸ばされた三角形４１０を作成するための動き平滑化の結果として、三角形４０６のいくつかが引き伸ばされる。移動された頂点４０８（１）～（４）に対応する（再投影フレーム３００（３）のピクセルデータ１１０（１）（３）の）ピクセル値は、レンダリングメッシュ２００内の移動された頂点４０８（１）～（４）の位置に対応する異なるピクセル位置でレンダリングされる。移動された頂点２０６（１）～（４）と移動されていない頂点４０２との間のピクセル位置は、（例えば、移動された頂点４０８（１）～（４）と移動されていない頂点４０２との間のピクセル値を内挿するなどによって、勾配を適用することによって）混合され得る。いくつかの実施形態では、深度バッファを利用して、再投影フレーム３００（３）の修正されたピクセルデータ１１０（１）（３）’のフレームバッファに出力される最終的なピクセル値のセットを決定し得る。すなわち、動きベクトル４０４（１）～（４）をレンダリングメッシュ４００に適用した結果、移動された頂点４０８（１）～（４）に対応する画像内の位置に複数のピクセル値が存在し得る。この場合、「より近い」（より小さい）深度値に関連付けられているピクセル値のいずれかは、「より遠い」（より大きい）深度値に関連付けられているその位置で別のピクセル値をレンダリングする代わりにレンダリングされ得る。

【００５７】

図５は、本明細書に開示される実施形態による、動き平滑化技術の一部としてグラフィ

ックスプロセッシングユニット（ＧＰＵ）によって生成される動きベクトルを使用して再投影フレームを生成するための例示的なプロセス５００のフロー図である。考察を目的として、プロセス５００は、以前の図を参照して説明される。

【００５８】

５０２において、ホストコンピュータ１０６のロジックは、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００と関連付けられているピクセルデータ１１０（１）を提供し得る。例えば、２つの最後にレンダリングされたフレーム３００に関連付けられているピクセルデータ１１０（１）は、圧縮されたピクセルデータ１１０（１）の送信前にピクセルデータ１１０（１）を圧縮するための圧縮アルゴリズムの一部としてＧＰＵ３０４への入力として提供され得る。これらのフレーム３００は、過去にレンダリングされた第１のフレーム３００（１）、および第１のフレーム３００（１）の後にレンダリングされた第２のフレーム３００（２）などのアプリケーション（例えば、ビデオゲームアプリケーション）から受信された実際のフレームであり得る。したがって、第２のフレーム３００（２）は、ホストコンピュータ１０６上で実行中のアプリケーションによって出力された最後にレンダリングされたフレームを表し得、第１のフレーム３００（１）および第２のフレーム３００（２）は、一連のフレーム３００内でアプリケーションによって連続して出力され得るが、ブロック５０２において入力として提供されるピクセルデータ１１０（１）は、連続してレンダリングされたフレームのピクセルデータ１１０（１）である必要はない。例えば、中間フレーム３００は、第１のフレーム３００（１）と第２のフレーム３００（２）との間でレンダリングされ得、ブロック５０２において入力として提供されるピクセルデータ１１０（１）は、第１のフレーム３００（１）および第２のフレーム３００（２）に関連し得る。

【００５９】

５０４において、動きベクトルアレイ１１０（２）は、ＧＰＵ３０４から受信され得る。ブロック５０４において受信された動きベクトル１１０（２）のアレイは、第１のフレーム３００（１）に関連付けられた第１のピクセルデータ１１０（１）（１）および第２のフレーム３００（２）に関連付けられた第２のピクセルデータ１１０（１）（２）に少なくとも部分的に基づいて（例えば、第１のピクセルデータ１１０（１）（１）と第２のピクセルデータ１１０（１）（２）との間の比較に基づいて）、ＧＰＵ３０４のビデオエンコーダによって生成され得る。ＧＰＵ３０４のビデオエンコーダは、例えば、比較されたピクセル値間の差が閾値差よりも小さいどうかを決定することによってなど、ピクセル値（またはピクセル値のグループ）間の類似性を探す好適なコンピュータビジョンおよび／またはビデオ符号化アルゴリズム（例えば、ＨＥＶＣ）を使用するように構成され得る。そのような類似性メトリック内のいずれも、２つのフレーム３００間で一致するピクセルデータ１１０（１）であると考えられ得る。

【００６０】

５０６において、ＨＭＤ１０２のロジックは、第２のフレーム３００（２）の第２のピクセルデータ１１０（１）（２）に少なくとも部分的に基づいて、再投影フレーム３００（３）の第３のピクセルデータ１１０（１）（３）を生成し得る。この場合、第２のフレーム３００（２）は、再投影フレーム３００（３）の直前のレンダリングされたフレームを表す。例えば、ブロック５０４と５０６との間で、ＨＭＤ１０２は、第２のフレーム３００（２）の圧縮された第２のピクセルデータ１１０（１）（２）を受信し、ピクセルデータ１１０（１）（２）を解凍し、ＨＭＤ１０２の更新された姿勢予測を考慮するためにピクセルデータ１１０（１）（２）を修正し、第２のフレーム３００（２）のＨＭＤ１０２上に画像を提示し得、ブロック５０６において、ＨＭＤ１０２は、次のフレームとして再投影フレーム３００（３）をレンダリングする準備をしている。

【００６１】

５０８において、ＨＭＤ１０２のロジックは、再投影フレーム３００（３）の修正された第３のピクセルデータ１１０（１）（３）’を取得するために、動きベクトル１１０（２）（２）のアレイに少なくとも部分的に基づいて、第３のピクセルデータ１１０（１）（

３）を修正し得る。サブブロック５１０および５１２によって示されるように、再投影フレーム３００（３）のピクセルデータ１１０（１）（３）の修正は、レンダリングメッシュ４００を利用し得る。

【００６２】

したがって、５１０において、ＨＭＤ１０２のロジックは、再投影フレーム３００（３）のレンダリングメッシュ４００を生成し得る。レンダリングメッシュ４００は、複数の頂点４０２を有するテッセレーションメッシュを含み得る。いくつかの実施形態では、レンダリングメッシュ４００の解像度は、動きベクトル４０４とレンダリングメッシュ４００の「要素」（例えば、レンダリングメッシュ４００の頂点４０２、レンダリングメッシュ４００の正方形（または四角形）などの要素）との間に１対１の対応があるように、動きベクトルアレイ１１０（２）（２）の解像度と一致し得る。動きベクトル場１１０（２）（２）とレンダリングメッシュ４００との間の一致解像度を取得することは、ＧＰＵ３０４が特定の解像度で動きベクトル場１１０（２）（２）を出力することを要求すること、動きベクトル場１１０（２）（２）の解像度をダウンサンプリングもしくはアップサンプリングすること、および／またはＧＰＵ３０４によって出力される動きベクトル場１１０（２）（２）の解像度に一致する解像度でレンダリングメッシュ４００を生成することなど、本明細書に記載される技術のいずれかを含み得る。

【００６３】

５１２において、ロジックは、レンダリングメッシュ４００の（複数の頂点４０２の）頂点４０２を、頂点４０８が移動するにつれて、レンダリングメッシュ４００内の異なる位置に移動させ得る。頂点４０２は、（ｉ）非ゼロ動きベクトル４０４の方向に、および（ｉｉ）非ゼロ動きベクトル４０４の大きさに対応する量だけ移動され得る。したがって、ブロック５０８における第３のピクセルデータ１１０（１）（３）の修正は、例えば、再投影フレーム３００（３）の修正された第３のピクセルデータ１１０（１）（３）’を取得するために、移動された頂点４０８に従って第３のピクセルデータ１１０（１）（３）のピクセル値を移動させることによって、レンダリングメッシュ４００の移動された頂点４０８に従って行われ得る。例えば、第３のピクセルデータ１１０（１）（３）のピクセル値は、レンダリングメッシュ４００内の移動された頂点４０８に従って、修正された第３のピクセルデータ１１０（１）（３）内の新しい位置に、左側に４ピクセル、および上方に４ピクセル移動され得る。

【００６４】

いくつかの実施形態では、複数の動きベクトル場１１０（２）は、ブロック５０２において入力された以前にレンダリングされたフレーム３００の異なるセットに基づいてブロック５０４において受信され得、追加の動き関連パラメータは、ブロック５０８における再投影フレームの動き平滑化で使用するために複数の動きベクトル場１１０（２）に基づいて決定され得る。例えば、ブロック５０４において受信された動きベクトル１１０（２）（２）のアレイをもたらす前の２つのフレーム３００（１）および３００（２）に加えて、図５のアルゴリズムは、いくつかの実施形態では、ＧＰＵ３０４への入力として第１のフレーム３００（１）および第１のフレーム３００（１）の前にレンダリングされた「ゼロ番目の」フレーム３００（０）を提供し、その異なる一対の入力フレーム３００に基づいて、追加の動きベクトル１１０（２）のアレイを受信することなどによって、１つ以上の追加のフレームを戻してもよい。次いで、動きベクトルの複数のアレイを比較して、フレーム間で動いているオブジェクトの加速度などの動き関連パラメータを決定してもよく、これらの動き関連パラメータは、動き平滑化調整に関してピクセル値を多かれ少なかれ移動させるように、第３のピクセルデータ１１０（１）（３）に適用される最終動きベクトル４０４の大きさを修正（例えば、増加／減少）することなどによって、ブロック５０８に適用され得る。

【００６５】

５１４において、ＨＭＤ１０２のロジックは、修正された第３のピクセルデータ１１０（１）（３）’に少なくとも部分的に基づいて、再投影フレーム３００（３）をディスプレ

イ上（例えば、ＨＭＤ１０２のディスプレイパネル上）にレンダリングし得る。ブロック５１４においてレンダリングされる結果として生じる再投影フレーム３００（３）は、ＧＰＵ３０４から受信された動きベクトル１１０（２）に従って、第３のピクセルデータ１１０（１）（３）の修正によって「動き平滑化」される。ＧＰＵ３０４が１つ以上のＧＰＵ３０４を表し得ることを理解されたい。例えば、複数のＧＰＵ３０４は、ＨＭＤ１０２のステレオディスプレイパネル上に所与のフレーム３００をレンダリングするために利用され得、これらのＧＰＵ３０４に入力されるフレーム３００のピクセルデータ１１０（１）は、それに応じて分割され得る（例えば、ピクセルデータ１１０（１）の左半分は、第１のＧＰＵ３０４への入力として提供され得、ピクセルデータ１１０（１）の右半分は、第２のＧＰＵ３０４への入力として提供され得る）。

【００６６】

図６は、本明細書に開示される実施形態による、ピクセルデータが動きベクトル推定のためにＧＰＵに入力される前に、以前にレンダリングされたフレームのピクセルデータを位置合わせするための例示的なプロセス６００のフロー図である。考察を目的として、プロセス６００は、以前の図を参照して説明される。さらに、図５および図６のページ外参照「Ａ」によって示されるように、プロセス６００は、図５のブロック５０４における動作の前に実行される動作を表し得、プロセス５００は、いくつかの実施形態では、ブロック５０４～５１４の動作を継続し得る。

【００６７】

６０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＨＭＤ１０２から受信した回転データに基づいて、ＨＭＤ１０２が、以前にレンダリングされたフレーム３００をレンダリングする間に第１の向きから第２の向きに回転したと決定し得、そのピクセルデータ１１０（１）は、ＧＰＵ３０４への入力として提供される。例えば、ユーザ１０４は、第１のフレーム３００（１）および第２のフレーム３００（２）をレンダリングする時間に対応し得る、時間ｔ１と時間ｔ２との間で右方向に頭部を回転させ得る。

【００６８】

６０４において、以前にレンダリングされたフレーム３００は、それらのフレーム３００のピクセルデータ１１０（１）がホストコンピュータ１０６のＧＰＵ３０４への入力として提供される前に位置合わせされ得る。これは、２つのフレーム３００（１）および３００（２）をレンダリングする過程でオブジェクト（静止しているオブジェクトおよび移動しているオブジェクトの両方）を位置間で移動させ得る、フレーム３００間のＨＭＤ１０２の移動のために行われる。一方のフレーム３００（１）を他方のフレーム３００（２）に位置合わせすることによって、またはその逆によって、シーン内の特定の静止しているオブジェクトを表すピクセルデータ１１０（１）を、静止しているオブジェクトに対応するピクセルデータ１１０（１）が、ＧＰＵ３０４のビデオエンコーダによって移動しているオブジェクトと間違われないように、２つのフレーム３００（１）と３００（２）との間の概ね同じ位置に移動させることができる。ブロック６０４における位置合わせは、（ｉ）第２のフレーム３００（２）をレンダリングする時間に、第１のフレーム３００（１）内のシーンをＨＭＤ１０２の第２の向きに位置合わせする修正された第１のピクセルデータ６１０（１）（１）を取得するための第１のピクセルデータ１１０（１）（１）（第１のフレーム３００（１）に関連付けられている）、または（ｉｉ）第１のフレーム３００（１）をレンダリングする時間に、第２のフレーム３００（２）内のシーンをＨＭＤ１０２の第１の方向に位置合わせさせる修正された第２のピクセルデータを取得するための第２のピクセルデータ１１０（１）（２）（第２のフレーム３００（２）に関連付けられている）のうちの少なくとも１つを修正することを含み得る。図６の下部の図は、前者の場合を示しており、第１のピクセルデータ１１０（１）（１）は、第１のフレーム３００（１）内のシーンをＨＭＤ１０２の第２の向きに位置合わせする修正された第１のピクセルデータ６１０（１）（１）を取得するように修正される。しかしながら、第１のピクセルデータ１１０（１）（１）または第２のピクセルデータ１１０（１）（２）のいずれ

かは、位置合わせ目的のために修正され得ることを理解されたい。ブロック６０４における位置合わせは、頭部の移動に起因する動きベクトル１１０（２）内の差分を減少させるために、フレームのピクセルデータのうちの一方または両方に対する回転再投影修正を含み得る。動きベクトル１１０（２）内の差分のこの減少は、本明細書に記載するように、ホストコンピュータ１０６からＨＭＤ１０２に動きベクトル１１０（２）を送信する際の帯域幅の減少という追加の利点を提供し得る。

【００６９】

６０６において、ホストコンピュータ１０６のロジックは、以前にレンダリングされたフレーム３００のうちの１つの修正されたピクセルデータ、およびＧＰＵ３０４への入力として他のフレーム３００の元のピクセルデータ１１０（１）を提供し得る。図６の図に示される例は、修正された第１のピクセルデータ６１０（１）（１）および元の第２のピクセルデータ１１０（１）（２）をＧＰＵ３０４への入力として提供することを描いている。述べたように、プロセス６００は、プロセス５００のブロック６０６からブロック５０４まで継続し得る（ページ外参照「Ａ」によって示されるように）。したがって、ブロック６０６においてＧＰＵ３０４への入力として提供されるピクセルデータに基づいて、ブロック５０４においてＧＰＵ３０４から動きベクトルアレイ１１０（２）を受信し得、図５の動き平滑化アルゴリズムの残りの動作を実行して、動き平滑化される再投影フレーム３００（３）をレンダリングし得る。

【００７０】

図７は、本明細書に開示される実施形態による、以前にレンダリングされたフレームのいくつかのピクセルデータを除外し、動きベクトル推定のためにＧＰＵへの入力としてピクセルデータの残りの部分を提供するための例示的なプロセス７００のフロー図である。考察を目的として、プロセス７００は、以前の図を参照して説明される。さらに、図５および図７のページ外参照「Ａ」によって示されるように、プロセス７００は、図５のブロック５０４における動作の前に実行される動作を表し得、プロセス５００は、いくつかの実施形態では、ブロック５０４～５１４の動作を継続し得る。

【００７１】

７０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＨＭＤ１０２から受信した回転データに基づいて、ＨＭＤ１０２が、ＧＰＵ３０４への入力として提供されることになる以前にレンダリングされた（例えば、ピクセルデータの圧縮のために）フレーム３００をレンダリングする間に第１の向きから第２の向きに回転したと決定し得る。以前にレンダリングされたフレーム３００は、図３に示される第１のフレーム３００（１）および第２のフレーム３００（２）であり得る。

【００７２】

７０４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として利用可能なピクセルデータ１１０（１）の一部分を提供し得る。ＧＰＵ３０４への入力として提供される利用可能なピクセルデータ１１０（１）の部分は、例えば、第１のピクセルデータ１１０（１）（１）の一部分７０６（１）および第２のピクセルデータ１１０（１）（２）の一部分７０６（２）を含み得、各々、ＨＭＤ１０２のディスプレイパネルの１つ以上のエッジにおけるピクセルのサブセット以外のピクセルに対応する。例えば、図７に示すように、ＧＰＵ３０４への入力として提供されるピクセルデータ１１０（１）の部分７０６は、ディスプレイパネルの左右のエッジ（図７に黒色で示される）でのピクセルデータ１１０（１）の残りの部分を除外する。言い換えれば、ディスプレイパネルの左右エッジにおけるピクセルデータ１１０（１）は、ＧＰＵ３０４が、ディスプレイパネルの左右エッジ間のピクセルデータ１１０（１）の中心部分７０６に動き推定努力を集中させるように、ＧＰＵ３０４に提供されない。正および／または負の垂直方向におけるＨＭＤ１０２の回転のために、ピクセルデータ１１０（１）の除外された部分は、画像の上端および下端であり得る。いずれにしても、画像のエッジでピクセルデータ１１０（１）の一部分を除外し、ピクセルデータ１１０（１）の残りの部分７０６をＧＰＵ３０４に排他的に提供することは、画像のエッジで効果的にゼロ動きベクトル

４０４を結果として生じ得、これは、不要な視覚的アーチファクトが画像のエッジ付近で顕現し、それが別様には移動しているオブジェクト３０２の特徴ではない外れ値動きベクトル４０４を結果として生じる可能性がある状況で有用であり得る。述べたように、プロセス７００は、ブロック７０４においてＧＰＵ３０４への入力として提供されるピクセルデータに基づいて、動きベクトルアレイ１１０（２）がＧＰＵ３０４から受信される、プロセス５００のブロック７０４からブロック５０４まで続行し得、図５の動き平滑化アルゴリズムは、動きが平滑化される再投影フレーム３００（３）をレンダリングするために実行され得る。

【００７３】

図８は、本明細書に開示される実施形態による、以前にレンダリングされたフレームのピクセルデータのパネルマスク部分を除外し、動きベクトル推定のためにＧＰＵへの入力としてピクセルデータの残りの部分を提供する、以前にレンダリングされたフレームのピクセルデータを位置合わせするための例示的なプロセス８００のフロー図である。考察を目的として、プロセス８００は、以前の図を参照して説明される。さらに、図５および図８のページ外参照「Ａ」によって示されるように、プロセス８００は、図５のブロック５０４における動作の前に実行される動作を表し得、プロセス５００は、いくつかの実施形態では、ブロック５０４～５１４の動作を継続し得る。

【００７４】

８０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＨＭＤ１０２から受信した回転データに基づいて、ＨＭＤ１０２が、ＧＰＵ３０４への入力として提供されることになる以前にレンダリングされたフレーム３００をレンダリングする間に第１の向きから第２の向きに回転したと決定し得る。以前にレンダリングされたフレームは、図３に示される第１のフレーム３００（１）および第２のフレーム３００（２）であり得る。

【００７５】

８０４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、第１のフレーム３００（１）に関連付けられている第１のピクセルデータ１１０（１）（１）を修正して、第１のフレーム３００（１）のシーンをＨＭＤ１０２の第２の方向に位置合わせする修正された第１のピクセルデータ８０８（１）を取得し得る。図８の例では、第１のピクセルデータ１１０（１）（１）の一部分は、ＨＭＤ１０２のディスプレイパネルの周辺においてレンダリングされたパネルマスク（黒色で示される）に対応するデータを各フレームで表している。したがって、修正された第１のピクセルデータ８０８（１）のパネルマスク部分８１４（１）（すなわち、図８の黒色で示される部分）は、修正された第１のピクセルデータ８０８（１）のパネルマスクに対応するデータを表している。述べたように、アプリケーションは、フレームをレンダリングするときに余分なピクセルデータを生成するように指示され得、余分なピクセルデータはパネルマスクに対応する。この方式では、パネルマスクを適用する際のＦＯＶの効果的な低減はない。

【００７６】

８０６において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、第２のフレーム３００（２）に関連付けられている第２のピクセルデータ１１０（１）（２）を修正して、第２のフレーム内のシーンをＨＭＤ１０２の第１の向きに位置合わせする修正された第２のピクセルデータ８０８（２）を取得し得る。ここでも、第２のピクセルデータ１１０（１）（２）の一部分は、ＨＭＤ１０２のディスプレイパネルの周辺においてレンダリングされたパネルマスク（黒色で示される）に対応するデータを各フレームで表している。したがって、修正された第２のピクセルデータ８０８（２）のパネルマスク部分８１４（２）（すなわち、図８の黒色の部分）は、修正された第２のピクセルデータ８０８（２）のパネルマスクに対応するデータを表している。

【００７７】

８１０において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、修正された第１のピクセルデータ８０８（１）のパネルマスク部分８１４（１）を修正された

第２のピクセルデータ８０８（２）のパネルマスク部分８１４（２）と組み合わせて、パネルマスク８１６に対応する共通値を有するピクセルのサブセットを決定し得る。これは、一種のベン図と考えることができ、パネルマスク８１６に対応するピクセルのサブセットは、修正された第１のピクセルデータ８０８（１）および修正された第２のピクセルデータ８０８（２）における修正されたパネルマスクの組み合わされたバージョンである。

【００７８】

８１２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、各々がパネルマスク８１６に対応するピクセルのサブセット以外のピクセルに対応する第１のピクセルデータ１１０（１）（１）の特定の部分（例えば、中心部分）および第２のピクセルデータ１１０（１）（２）の特定の部分（例えば、中心部分）を提供し得る。これは、ＧＰＵ３０４が、パネルマスク８１６ピクセルによって覆われている各々の以前にレンダリングされたフレーム内のピクセルデータ１１０（１）の部分の動き推定を無視することを可能にし、ＧＰＵ３０４は、その動き推定の努力を、各フレーム３００のピクセルデータ１１０（１）の中心、非パネルマスク部分に集中させることができる。述べたように、プロセス８００は、ブロック８１２においてＧＰＵ３０４への入力として提供されるピクセルデータに基づいて、動きベクトルアレイ１１０（２）がＧＰＵ３０４から受信される、プロセス５００のブロック８１２からブロック５０４まで続行し得、図５の動き平滑化アルゴリズムは、動きが平滑化される再投影フレーム３００（３）をレンダリングするために実行され得る。

【００７９】

図９は、本明細書に開示される実施形態による、動き平滑化技術の一部として再投影フレームを生成するために使用される前にＧＰＵから出力される動きベクトルを閾値化するための例示的なプロセス９００のフロー図である。考察を目的として、プロセス９００は、以前の図を参照して説明される。

【００８０】

９０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００と関連付けられているピクセルデータ１１０（１）を提供し得る。ブロック９０２において実行される動作は、プロセス５００のブロック５０２に関して説明される動作と同様であり得る。

【００８１】

９０４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から動きベクトルアレイ１１０（２）を受信し得る。ブロック９０４において実行される動作は、プロセス５００のブロック５０４に関して説明される動作と同様であり得る。

【００８２】

９０６において、ロジック（例えば、ホストコンピュータ１０６からの動きベクトル１１０（２）の受信後のＨＭＤ１０２のロジック）は、動きベクトル１１０（２）のアレイ内の個々の動きベクトルの大きさ（または長さ）を第１の閾値の大きさと比較して、第１の閾値の大きさよりも大きい大きさを有する動きベクトル１１０（２）の第１のサブセットを決定し得る。第１の閾値の大きさは、仮想オブジェクト３０２の移動または動き以外の何かであるフレーム間の変化を表すことができる高い大きさの外れ値動きベクトル１１０（２）のこのサブセットの影響を軽減するために利用され得る。異常に高い大きさの動きベクトルは、様々な理由で生じ得る。ＧＰＵ３０４は、第２のフレーム３００（２）の右上のピクセルが、第１のフレーム３００（１）の左下のピクセルに十分に類似していることを見出し得、この動きベクトル４０４がフレーム３００間の仮想オブジェクト３０２の移動または動きを表すものではないにもかかわらず、アレイ内の他の動きベクトル４０４と比較して比較的高い大きさで結果として生じる動きベクトル４０４を出力し得る。場合によっては、ビデオゲームは、ユーザがフレーム３００間でシーンが激しく変化する異なる位置にテレポートすることを可能にし得、大きな動きベクトル４０４をＧＰＵ３０４のビデオエンコーダによって生成させる。これらおよび他の場合において、これらの大き

な動きベクトル４０４を閾値化することは有用であり得る。
【００８３】
９０７において、複数の以前にレンダリングされたフレーム３００が第１のフレーム３００（１）および第２のフレーム３００（２）を含む場合、ロジック（例えば、ＨＭＤ１０２のロジック）は、第１のフレーム３００（１）がレンダリングされた第１の時間と、第２のフレーム３００（２）がレンダリングされた第２の時間との間の期間（または間隔）を決定し得る。
【００８４】
９０９において、ブロック９０６における比較で使用される第１の閾値の大きさは、第１の時間と第２の時間との間の時間周期に少なくとも部分的に基づいて選択され得る。用例では、単位時間当たりのユーザの視点からの移動度で測定された第１の閾値の大きさは、２つのフレーム３００（１）と３００（２）との間の時間の１１．１１ミリ秒当たりの移動度６である。したがって、２つのフレーム３００（１）および３００（２）が離れている時間が長いほど、第１の閾値の大きさが大きくなり、したがって、動きベクトル１１０（２）は（大きさが）許容されるだけ長くなる。これは、ある速度を超えた動きがあると、動き平滑化がそれほど効果的でない場合があり、実際に有害な視覚的アーチファクト（例えば、シーンの一部を「泳いでいる」ように見せることによって）を引き起こす場合があるためである。
【００８５】
９０８において、ロジック（例えば、ＨＭＤ１０２のロジック）は、第１の閾値の大きさよりも大きい大きさを有すると決定された動きベクトル１１０（２）の第１のサブセットの大きさを減少させ得、その結果、大きさは、第１の閾値の大きさで制限される。これは、第１の修正された動きベクトルアレイ１１０（２）を作成する。言い換えれば、第１の閾値の大きさを超えるそれらの動きベクトル１１０（２）について、ロジックは、動きベクトル１１０（２）の第１の修正されたアレイが第１の閾値の大きさ以下の大きさを含み、第１の閾値の大きさより大きい大きさを含まないように、それらの動きベクトル１１０（２）の大きさを第１の閾値の大きさに制限（または限定）するように構成される。いくつかの実施形態では、第１の閾値の大きさで動きベクトル１１０（２）の大きさを制限し、動き平滑化のために非ゼロ動きベクトルのすべてを使用する代わりに、ロジックは、第１の閾値の大きさを満たすかまたは超える大きさを有する動きベクトル１１０（２）の第１のサブセットを破棄することができ、これにより、それらは動き平滑化において全く使用されない。
【００８６】
９１０において、ロジック（例えば、ＨＭＤ１０２のロジック）は、動きベクトル１１０（２）のアレイにおける個々の動きベクトルの大きさ（または長さ）を第２の閾値の大きさと比較して、第２の閾値の大きさよりも小さい大きさを有する動きベクトル１１０（２）の第２のサブセットを決定し得る。第２の閾値の大きさは、低い大きさの外れ値の動きベクトル４０４のこのサブセットの影響を軽減するために利用され得、これは、ユーザの一定の頭部の移動によってしばしば引き起こされるフレーム間の変化、および／または正確であるが絶対ゼロ度の動きまで正確ではない頭部追跡を表し得る。このため、ＧＰＵ３０４の出力は、ゼロ長の動きベクトル４０４を提供することはほとんどない。むしろ、上述のように、ＧＰＵ３０４の出力は、周囲の頭部の動きおよび／または追跡ジッタに起因するかなりの量のノイズを有することがより一般的である。言い換えれば、２つの連続したフレーム間のピクセルは、１００％一致することは稀である。用例では、動きのピクセルで測定される第２の閾値の大きさは、動きの２ピクセルの閾値である。この例では、２ピクセル未満の動きの大きさ（または長さ）を有する任意の動きベクトルは、低い大きさの外れ値ベクトルであると考えられる。
【００８７】
９１２において、ロジック（例えば、ＨＭＤ１０２のロジック）は、第２の閾値よりも小さい大きさを有すると決定された動きベクトル１１０（２）の第２のサブセットの大き

さ（例えば、２ピクセルよりも小さい動きの長さを有する低い大きさの外れ値動きベクトル）をゼロの長さ／大きさに減少させ得る。これは、第２の閾値の大きさよりも小さい大きさを有する任意の動きベクトルを含まない第２の修正された動きベクトルアレイ１１０（２）を作成する。これは、本明細書では、動きベクトル１１０（２）に小さなデッドゾーンを適用すると称されることもある。

【００８８】

９１４において、ロジック（例えば、ＨＭＤ１０２のロジック）は、第１の閾値の大きさで制限された大きさを有する動きベクトル１１０（２）の第２の修正されたアレイに基づいて、および第２の閾値よりも小さい大きさを有するいかなる動きベクトル４０４も伴わずに、再投影フレームのピクセルデータ１１０（１）（３）を修正し得る。いくつかの実施形態では、第１の閾値の大きさで制限される代わりに、第１の動きベクトルのサブセットが破棄される場合などに、ブロック９１４において、破棄された動きベクトルの第１のサブセット１１０（２）以外の動きベクトルの残りのサブセット１１０（２）は、再投影フレームのピクセルデータ１１０（１）（３）を修正するために使用される。ピクセルデータ１１０（１）（３）の修正は、本明細書に記載されている技術（例えば、プロセス５００のブロック５０８～５１２を参照して記載されているもの）のいずれかを含み得る。

【００８９】

９１６において、ロジック（例えば、ＨＭＤ１０２のロジック）は、再投影レーム３００（３）の修正されたピクセルデータ１１０（１）（３）’に少なくとも部分的に基づいて、ディスプレイ上（例えば、ＨＭＤ１０２のディスプレイパネル上）における再投影フレーム３００（３）をレンダリングし得る。ブロック９１６において実行される動作は、プロセス５００のブロック５１４に関して説明される動作と同様であり得る。

【００９０】

図１０は、本明細書に開示される実施形態による、動き平滑化技術の一部として再投影フレームを生成するために使用される前にＧＰＵから出力される動きベクトルを減衰させるための例示的なプロセス１０００のフロー図である。考察を目的として、プロセス１０００は、以前の図を参照して説明される。

【００９１】

１００２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００と関連付けられているピクセルデータ１１０（１）を提供し得る。ブロック１００２において実行される動作は、プロセス５００のブロック５０２に関して説明される動作と同様であり得る。

【００９２】

１００２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から動きベクトルアレイ１１０（２）を受信し得る。ブロック１００４において実行される動作は、プロセス５００のブロック５０４に関して説明される動作と同様であり得る。

【００９３】

１００６において、ロジック（例えば、ホストコンピュータ１０６からの動きベクトル１１０（２）を受信した後のＨＭＤ１０２のロジック）は、動きベクトル１１０（２）の個々のものの大きさを減衰（例えば、短縮、減少など）する目的で、動きベクトル１１０（２）のアレイ上にオーバーレイされた減衰テクスチャを生成し得る（例えば、それらの動きベクトル１１０（２）の品質に低い信頼性がある場合）。減衰テクスチャは、減衰テクスチャが複数のテクセル（例えば、テクセルのグリッド）を含むように、任意の好適な解像度で生成され得る。減衰テクスチャの解像度は、複数の動きベクトルが減衰テクスチャの個々のテクセル内にあるように、動きベクトルアレイ１１０（２）の解像度よりも低い解像度であり得る。例では、減衰テクスチャは、ユーザの視野（ＦＯＶ）の中心にあるテクセルが約６度×６度（水平および垂直）であるような解像度を有することができる。テクセル当たりの度数は減衰テクスチャ全体で一定でない場合があるので、テクセル当たりの解像度（水平方向および垂直方向）はおおよそのものであってよい。例えば、ＨＭＤ

１０２のＦＯＶは、領域においておよそ１００度×１００度（例えば、プラスまたはマイナス約１０度）に及ぶことがある。減衰テクスチャは、解像度が減衰テクスチャのテクセルごとに変化するが、解像度がおおよそ６～１０度×６～１０度であるテクセルを提供するように、非線形投影行列に基づいて生成され得る。さらに、減衰テクスチャの各テクセルは、減衰値を割り当てられ得る（例えば、０．０～１．０の範囲内）。所与のテクセルに割り当てられた減衰値は、そのテクセルにおける動きベクトル４０４の大きさが減少（短縮）される量を制御する。例えば、１．０の減衰値は、減衰しないことに対応し得るので、１．０の減衰値を動きベクトル４０４に適用することは、動きベクトル４０４をその大きさに関してそのまま（例えば、減少させない、短縮させないなど）にしておくことを意味する。しかしながら、０．０の減衰値は、完全な減衰に対応し得るので、０．０の減衰値を動きベクトル４０４に適用することは、動きベクトル４０４がその大きさに関してゼロに減少することを意味する。０．０～１．０の減衰値は、部分的な減衰に対応し得るので、例えば、０．５の減衰値を動きベクトル４０４に適用することは、動きベクトルが、その元の長さ（大きさ）の５０％（半分）に減少することを意味する。したがって、減衰テクスチャは、動きベクトル４０４の大きさを減少（短縮）することができるが、動きベクトル４０４の大きさを増大（延長）することはできない。

【００９４】

図１０は、２つの連続してレンダリングされたフレーム３００の間の画面の下部における輝度（または色）の急激な変化に基づいてブロック１００６において生成され得る、例示的な減衰テクスチャ１００７を示している。例えば、ビデオゲームをプレイしている間、ＨＭＤ１０２のユーザは、ハンドヘルドコントローラ上で、次のレンダリングされたフレーム３００上の画面の下半分に大きな青色のメッシュを出現させるボタンを選択し得る。この例示的なシナリオでは、ＧＰＵ３０４は、このフレーム３００および以前にレンダリングされたフレーム３００を入力として処理するときに、画面の下半分の大きな青色のメッシュ内のピクセルの輝度値に最も一致する輝度値を有する、シーンの上部でレンダリングされる空のピクセルを見つけることができる。この結果、ＧＰＵ３０４は、大きさが突然非常に大きくなり、シーン内の実際の動きを表さないシーンの下部の動きベクトル４０４を有する動きベクトル場１１０（２）を出力することになる。例示的な減衰テクスチャ１００７では、白色のテクセルは、１．０の減衰値を表し得る一方で、黒色のテクセルは、０．０の減衰値を表し得、濃淡のある灰色のテクセルは、０．０～１．０の減衰値を表し得る。

【００９５】

１００８において、ロジック（例えば、ＨＭＤ１０２のロジック）は、ブロック１００６において生成された減衰テクスチャを使用して（例えば、減衰値に対応する量だけ動きベクトル１１０（２）の大きさを減少させるために減衰テクスチャの減衰値を使用して）動きベクトル１１０（２）のアレイにおける個々の動きベクトル４０４の大きさを減少させ得る（例えば、スケールダウンし得る）。これは、減衰テクスチャにおける１．０未満の少なくとも１つの減衰値を仮定して、修正された動きベクトルアレイ１１０（２）を作成する。その目的は、フレーム３００の間で閾値量を超えて変化した動きベクトル４０４をスケールダウンすることであり、これは、これらの動きベクトル４０４がシーン内の実際の動きを表しておらず、動きを表していると勘違いしてはいけないことを最もよく示している。減衰テクスチャは、それらの領域内の動きベクトルが単一のフレーム内で極端である（その大きさに関して）画面の領域を見つけるために使用され、これは、動きベクトルが実際の動きを表すために信頼できないことを示し、または、それらがシーン内の実際の動きを表す場合、そのような動きはとにかく動き平滑化が目立った影響を有するには速すぎ、それらの動きベクトルを使用して動き平滑化を試みるよりゼロ動きベクトルを有することが好ましいことを示している。現実的には、フレーム間で動きがある場合、ディスプレイのリフレッシュレートは通常９０Ｈｚ程度であり、フレーム間のその短い時間で何かが大きく動くことは難しいので、動きは通常、以前のフレームとそれほど大きく変わらない。したがって、多くの場合、減衰テクスチャによって減衰されたこれらの極端な動き

ベクトル４０４は、実際の動きを表さない。

【００９６】

サブブロック１００９によって示されるように、いくつかの実施形態では、ブロック１００８において減衰テクスチャを使用して動きベクトルの大きさを減少させることは、減衰テクスチャを使用して動きベクトルの大きさを減少させる前に、最小Ｎ×Ｎフィルタを減衰テクスチャに適用することを含み得る。式中、Ｎは任意の好適な数であり得る。Ｎ＝３は、最低３×３のフィルタ（すなわち、３×３ブロックのテクセル）を使用する。Ｎ＝３の例を使用して、ブロック１００９において最小３×３フィルタを減衰テクスチャに適用する際に、ロジック（例えば、ＨＭＤ１０２のロジック）は、減衰テクスチャ１００７からテクセル単位で、対応するテクセル（例えば、対応するテクセルおよびその８つの隣接するテクセルを含むテクセルのブロック）を中心とするテクセルの各３×３ブロックにおける９個の減衰値のうちの最小値を取得し得る。ブロック１００９においてこの最小Ｎ×Ｎフィルタを適用すると、通常、非ゼロ減衰値が割り当てられているが、隣接するテクセルがゼロの減衰値を割り当てられている減衰テクスチャの任意の対応するテクセルが、その対応するテクセルが減衰テクスチャ１００７において非ゼロ減衰値を割り当てられているにもかかわらず、その対応するテクセルの対応する動きベクトルにゼロの減衰値を適用し、それによって対応するテクセルの動きベクトルをゼロに減少させることになるため、より多くの動きベクトル４０４をゼロ化させることになる。言い換えれば、減衰テクスチャの所与のテクセル内の動きベクトル４０４は、隣接するテクセルにゼロの減衰値が割り当てられている場合、ゼロ化され、これは、隣接するテクセルが多くの高い大きさの動きベクトルを有することを意味する。図１０は、最小Ｎ×Ｎフィルタを減衰テクスチャ１００７に適用した結果である例示的な減衰テクスチャ１０１１を示している。結果として生じる減衰テクスチャ１０１１は、減衰テクスチャ１００７よりも多くの黒色のテクセルを含み、これは、結果として生じる減衰テクスチャ１０１１を使用してより多くの動きベクトルがゼロ化されることを意味する。

【００９７】

１０１０において、ロジック（例えば、ＨＭＤ１０２のロジック）は、減衰テクスチャ内の１．０よりも小さい減衰値について減衰されたおそらくいくつかの大きさを伴う、修正された動きベクトルアレイ１１０（２）に基づいて、再投影フレームのピクセルデータ１１０（１）（３）を修正し得る。ピクセルデータ１１０（１）（３）の修正は、本明細書に記載されている技術（例えば、プロセス５００のブロック５０８～５１２を参照して記載されているもの）のいずれかを含み得る。

【００９８】

１０１２において、ロジック（例えば、ＨＭＤ１０２のロジック）は、再投影レーム３００（３）の修正されたピクセルデータ１１０（１）（３）’に少なくとも部分的に基づいて、ディスプレイ上（例えば、ＨＭＤ１０２のディスプレイパネル上）における再投影フレーム３００（３）をレンダリングし得る。ブロック１０１２において実行される動作は、プロセス５００のブロック５１４に関して説明される動作と同様であり得る。

【００９９】

図１１は、本明細書に開示される実施形態による、減衰テクスチャを生成するための例示的なプロセス１１００のフロー図であり、減衰テクスチャは、動き平滑化技術の一部として再投影フレームを生成するために使用される前に、ＧＰＵから出力される動きベクトルを減衰させるために使用され得る。したがって、プロセス１１００は、プロセス１０００のブロック１００６において減衰テクスチャを生成するために実行され得る動作のサブプロセスを表し得る。考察を目的として、プロセス１１００は、以前の図を参照して説明される。

【０１００】

１１０２において、ホストコンピュータ１０６から動きベクトルアレイ１１０（２）を受信した後、ＨＭＤ１０２のロジックは、動きベクトル１１０（２）の複数の以前に取得されたアレイに少なくとも部分的に基づいて、減衰テクスチャの各テクセルについての差

分ベクトルのセットを決定し得る。例えば、動きベクトル１１０（２）のアレイの履歴は、一連のフレームが処理されるにつれて維持され得、動きベクトル１１０（２）の最後の２つのアレイ（例えば、動きベクトル１１０（２）の最後２つのアレイ）は、ブロック１１０２における差分ベクトルのセット（例えば、２つのアレイにおける対応する動きベクトルの対間の差（例えば、大きさに関して）を示す異なるベクトル）を決定するために比較され得る。

【０１０１】

１１０４において、ロジック（例えば、ＨＭＤ１０２のロジック）は、閾値の大きさよりも大きい大きさを有する減衰テクスチャの各テクセル内の差分ベクトルのパーセンテージを計算し得る。例えば、ブロック１１０４において、ロジックは、減衰テクスチャの所与のテクセル内の差分ベクトルを評価し、テクセル内の「外れ値」ベクトルを、閾値の大きさよりも大きい差分（例えば、大きさ）を有するそれらの差分ベクトルとして識別し、テクセル内の外れ値ベクトルのパーセンテージを計算し得る。ブロック１１０４において「外れ値」ベクトルを識別する目的のために、任意の好適な閾値、例えば、フレーム当たり（または時間１１．１１ミリ秒当たり）約３度の移動／動き（ユーザの視点から）の閾値を使用することができる。言い換えれば、ブロック１１０４において使用される閾値は、図９のプロセス９００のブロック９０７および９０９を参照して上述したように、第１のフレーム３００（１）がレンダリングされたときの第１の時間と、第２のフレーム３００（２）がレンダリングされたときの第２の時間との間の時間周期（または間隔）に基づき得る。

【０１０２】

１１０６において、ロジック（例えば、ＨＭＤ１０２のロジック）は、減衰テクスチャの各テクセルについて、「外れ値」差分ベクトルであるテクセル内の差分ベクトルのパーセンテージが、２５％などの閾値パーセンテージを満たすかまたはそれを超えるかを決定し得る。ブロック１１０６において、「外れ値」差分ベクトルであるテクセル内の差分ベクトルのパーセンテージが閾値パーセンテージを満たすかまたは超える場合、プロセス１１００は、ブロック１１０６からブロック１１０８への「はい」経路をたどり得る。

【０１０３】

１１０８において、ロジック（例えば、ＨＭＤ１０２のロジック）は、閾値パーセンテージを満たすかまたは超えるいくつかの「外れ値」差分ベクトルを有するテクセルについて、減衰値をゼロとして計算し得る。言い換えれば、ロジックは、ある特定の量（例えば、２５％以上）の差分ベクトルが最大長さよりも大きい場合、所与のテクセルの減衰値をゼロとして計算し得る。プロセス１０００に記載されるように、ゼロの減衰値を所与のテクセル内の動きベクトル４０４に適用することは、それらの動きベクトル４０４の大きさをゼロに減少させる（例えば、そのテクセル内の動きベクトルを完全に減衰させるために）。

【０１０４】

１１０６において、「外れ値」差分ベクトルの閾値パーセンテージ未満を有するテクセルについて、プロセス１１００は、ブロック１１０６からブロック１１１０への「いいえ」経路をたどり得、ロジックは、ブロック１１０４において計算されたパーセンテージに少なくとも部分的に基づいている減衰値として、それらのテクセルの減衰値を計算し得る。例えば、所与のテクセルの閾値の大きさよりも大きい大きさを有する差分ベクトルのパーセンテージが、閾値パーセンテージ未満（例えば、２５％）である場合、減衰値は、閾値パーセンテージ未満の値の範囲内（例えば、０％～２５％の範囲内）で計算されたパーセンテージに線形にマッピングする値として設定され得る。ブロック１１１０において計算された減衰値は、ゼロ化されていない残りのベクトル内のノイズを減少させるために、（減衰テクスチャが動きベクトル場１１０（２）に適用されたときに）有効である。

【０１０５】

１１１２において、計算された減衰値は、以前にレンダリングされたフレーム３００に基づいて、そのテクセルの既存の減衰値と比較され得る。ブロック１１１２において減衰

値が既存の値未満である場合、プロセス１１００は、ブロック１１１２からブロック１１１４への「はい」経路をたどり得、そこでテクセルの減衰値は、ブロック１１０８または１１１０で計算された減衰値に設定される。例えば、以前のフレーム内の減衰テクスチャのそのテクセルに対する減衰がなかった場合、既存の減衰値は１．０に等しくてもよく、ブロック１１０８または１１１０において計算された新しい減衰値が１．０未満である場合、この例では、減衰値は、所与のテクスチャ内の動きベクトルを減衰させるために新たに計算された減衰値に設定される。

【０１０６】

１１１２において、計算された減衰値が問題のテクセルの既存の値以上である場合、プロセス１１００は、ブロック１１１２からブロック１１１６への「いいえ」経路に従い得、既存の減衰値は、減衰がないことに対応する最大減衰値（例えば、１．０）に向かって経時的に増分される。例えば、いくつかの状況は、フレーム間で変化する不規則な動きベクトルを引き起こす可能性があるため、新しい減衰値は、減衰テクスチャのテクセルの以前の（古い）減衰値の上に「混合」され得、これは、計算された減衰値が、その期間にわたってテクセルの既存の減衰値よりも決して小さくないと仮定して、減衰値が、ある期間にわたって（例えば、１秒の期間にわたって）増分的に増加して減衰がなくなることを意味する。

【０１０７】

図１２は、本明細書に開示される実施形態による、結果として生じる動きベクトルのセットが動き平滑化技術の一部として再投影フレームを生成するために使用される前に、色変化がほとんどまたはまったくない領域に対応する動きベクトルをゼロ化するための例示的なプロセス１２００のフロー図である。考察を目的として、プロセス１２００は、以前の図を参照して説明される。

【０１０８】

１２０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００と関連付けられているピクセルデータ１１０（１）を提供し得る。ブロック１２０２において実行される動作は、プロセス５００のブロック５０２に関して説明される動作と同様であり得る。

【０１０９】

１２０４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から動きベクトルアレイ１１０（２）を受信し得る。ブロック１２０４において実行される動作は、プロセス５００のブロック５０４に関して説明される動作と同様であり得る。

【０１１０】

１２０６において、ロジック（例えば、ホストコンピュータ１０６から動きベクトル１１０（２）を受信した後のＨＭＤ１０２のロジック）は、動きベクトル１１０（２）のアレイにおける動きベクトルごとの色差値を決定するために、ＧＰＵ３０４に入力された複数の以前にレンダリングされたフレーム３００の間で色ピクセル値（ＲＧＢベクトルに関して）を比較し得る。これらの色差値は、閾値色差と比較されて、それらの色差値が閾値色差よりも小さいため、フレーム間の色変化がほとんどまたはまったくないことに関連付けられている動きベクトル１１０（２）のサブセットを決定し得る。いくつかの実施形態では、ロジックは、ＲＧＢピクセル値をベクトルとして扱う。次いで、色の差（例えば、差分ベクトル）を決定するために、以前にレンダリングされたフレーム３００間の対応するＲＧＢベクトルを比較する。動きベクトル１１０（２）のアレイ内の各動きベクトル４０４は、ピクセルのブロック（例えば、８×８ピクセルのブロック）に対応し得るため、このような色比較は、フレーム３００間の８×８ピクセルのブロック内のＲＧＢピクセル値の一部または全部を比較することによって、動きベクトルベースで実行され得る。いくつかの実施形態では、ロジック（例えば、ＨＭＤ１０２のロジック）は、フレーム３００間の色変化の量（例えば、色差値）を決定するために、単一の動きベクトル４０４に対応する、ピクセルの各８×８ブロック内のピクセルの他の行ごとおよび他の列ごとの色値を

比較し、色変化の量は、特定の動きベクトル４０４に関連付けられている。したがって、所与の動きベクトル１１０（２）に対応するピクセルのブロックに関して、フレーム間の色比較は、処理リソースを保存するために、対応するピクセルのブロック内のピクセルの一部分（例えば、２５％）を比較することを伴い得る。

【０１１１】

１２０８において、ロジック（例えば、ＨＭＤ１０２のロジック）は、色差値が閾値色差未満である、動きベクトル１１０（２）のサブセットの大きさをゼロの長さ／大きさに減少させ得る。これは、以前にレンダリングされたフレーム間で色変化がほとんどまたはまったくなかったシーンの領域内の任意の動きベクトルを含まない、修正された動きベクトルアレイ１１０（２）を作成する（ゼロ動き色差と呼ばれることもある）。例えば、テキストが単色の背景の前にあるときに、テキストがぼんやりして見えたり、エッジの周りがねじれて見えたりするのを防ぐのに役立ち得る。また、シーン内の点灯または他のユーザインターフェース要素が徐々に明るくなったり暗くなったりするようにみえる、視覚障害を引き起こすアーチファクトを防ぐのにも役立ち得る。

【０１１２】

１２１０において、ロジック（例えば、ＨＭＤ１０２のロジック）は、修正された動きベクトル１１０（２）のアレイに基づいて再投影フレームについてのピクセルデータ１１０（１）（３）を修正し得、修正された動きベクトル１１０（２）は、色が以前にレンダリングされたフレーム３００間で効果的に同じである領域において任意の動きベクトル４０４をもはや含まない。ピクセルデータ１１０（１）（３）の修正は、本明細書に記載されている技術（例えば、プロセス５００のブロック５０８～５１２を参照して記載されているもの）のいずれかを含み得る。

【０１１３】

１２１２において、ロジック（例えば、ＨＭＤ１０２のロジック）は、再投影レーム３００（３）の修正されたピクセルデータ１１０（１）（３）’に少なくとも部分的に基づいて、ディスプレイ上（例えば、ＨＭＤ１０２のディスプレイパネル上）における再投影フレーム３００（３）をレンダリングし得る。ブロック１２１２において実行される動作は、プロセス５００のブロック５１４に関して説明される動作と同様であり得る。

【０１１４】

図１３は、本明細書に開示される実施形態による、結果として生じる動きベクトルのセットが動き平滑化技術の一部として再投影フレームを生成するために使用される前に、１つ以上のフィルタを使用して動きベクトル場を「クリーンアップする」ための例示的なプロセス１３００のフロー図である。考察を目的として、プロセス１３００は、以前の図を参照して説明される。

【０１１５】

１３０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００と関連付けられているピクセルデータ１１０（１）を提供し得る。ブロック１３０２において実行される動作は、プロセス５００のブロック５０２に関して説明される動作と同様であり得る。

【０１１６】

１３０４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から動きベクトルアレイ１１０（２）を受信し得る。ブロック１３０４において実行される動作は、プロセス５００のブロック５０４に関して説明される動作と同様であり得る。

【０１１７】

１３０６において、ロジック（例えば、ホストコンピュータ１０６から動きベクトル１１０（２）を受信した後のＨＭＤ１０２のロジック）は、Ｎ×Ｎスカラーメディアンフィルタを動きベクトル１１０（２）のアレイに適用して、動きベクトル１１０（２）の第１の修正されたアレイを取得し得る。式中、Ｎは、５×５のスカラーメディアンフィルタを適用するＮ＝５などの任意の好適な数であり得る。Ｎ＝５の例では、５×５スカラーメデ

ィアンフィルタは、選択された動きベクトルを取り囲む５×５の動きベクトルのセット１１０（２）を見ることによって、動きベクトル１１０（２）のアレイ内の各動きベクトルに適用される。選択された動きベクトルについて、選択されたベクトルを囲む５×５の領域は、２５個の動きベクトルを有する。選択された動きベクトルに適用されるスカラーメディアンフィルタは、（２５のｘ成分値のうちの）中央ｘ成分値、および（２５のｙ成分値のうちの）中央ｙ成分値を計算し、次いで、中央ｘ成分値と中央ｙ成分値とを組み合わせて、スカラー中央値ベクトルを取得し、選択された動きベクトルは、スカラーメディアンベクトルで置き換えられ得る。このプロセスは、動きベクトル１１０（２）のアレイ内の各ベクトルについて反復する。ここでも、Ｎに関する任意の適切な値を使用することができ、Ｎ＝５は単なる例である。ブロック１３０６においてスカラーメディアンフィルタを適用することは、動きベクトル場１１０（２）内のノイズを低減し、外れ値動きベクトルを除去するのに役立つ。ブロック１３０６においてスカラーメディアンフィルタを使用することはまた、より計算集約的であるより複雑な距離アルゴリズムに依拠するベクトルメディアンフィルタなどのより複雑なフィルタを適用するよりも計算上高速である。

【０１１８】

１３０８において、ロジック（例えば、ＨＭＤ１０２のロジック）は、Ｍ×Ｍ平均最大ぼかしフィルタを動きベクトル１１０（２）の第１の修正されたアレイに適用して、動きベクトル１１０（２）の第２の修正されたアレイを取得し得る。式中、Ｍは、３×３平均最大ぼかしフィルタを適用するＭ＝３などの任意の好適な数であり得る。平均最大ぼかしフィルタは、独立して２つの別個のフィルタ（平均ベクトルフィルタおよび最大長ベクトルフィルタ）を適用し、次いで、これらの２つの独立して適用されたフィルタから得られたベクトルのベクトル平均を取得する。Ｍ＝３を使用する例では、３×３平均最大ぼかしフィルタは、選択された動きベクトルを取り囲む３×３セットの動きベクトル１１０（２）を見ることによって、動きベクトル１１０（２）のアレイ内の各動きベクトルに適用される。選択された動きベクトルについて、選択されたベクトルを囲む３×３の領域は、９個の動きベクトルを有する。まず、平均ベクトルフィルタを選択された動きベクトルに適用し、（９個のｘ成分値のうちの）平均ｘ成分値、および（９個のｙ成分値のうちの）平均ｙ成分値を計算し、次いで、平均ｘ成分値と平均ｙ成分値とを組み合わせて平均ベクトルを取得する。次に、最大長ベクトルフィルタが、（９個の動きベクトルのうちの）最長ベクトルを決定する、選択された動きベクトルに独立して適用される。次いで、結果として生じるベクトルは、平均ベクトルおよび最長ベクトルのベクトル平均を取ることによって決定され、選択された動きベクトルは、結果として生じるベクトルに置き換えられ得る。このプロセスは、動きベクトル１１０（２）のアレイ内の各ベクトルについて反復する。ここでも、Ｍに関する任意の好適な値を使用することができ、Ｍ＝３は単なる例である。ブロック１３０８において平均最大ぼかしフィルタを適用することは、動きベクトルのグループのサブエリア内の不連続を除去する平滑化された動きベクトル場を提供する。いくつかの実施形態では、ブロック１３０８において平均最大ぼかしフィルタを適用することは、本明細書に記載するように、動きベクトル場１１０（２）が動き平滑化に使用される前に、動きベクトル場１１０（２）上での最終的な「クリーンアップ」ステップである。例えば、動きベクトルの閾値化、減衰テクスチャの使用、色差がほとんどまたはまったくない領域におけるベクトルのゼロ化など、本明細書に記載する他の技術と組み合わせて使用される場合、これらの様々な動作は、平均最大ぼかしフィルタを適用する前に実行され得る。

【０１１９】

１３１０において、ロジック（例えば、ＨＭＤ１０２のロジック）は、フィルタリングされた動きベクトルを用いて修正された動きベクトルアレイ１１０（２）に基づいて、再投影フレームのピクセルデータ１１０（１）（３）を修正し得る。ピクセルデータ１１０（１）（３）の修正は、本明細書に記載されている技術（例えば、プロセス５００のブロック５０８～５１２を参照して記載されているもの）のいずれかを含み得る。

【０１２０】

１３１２において、ロジック（例えば、ＨＭＤ１０２のロジック）は、再投影レーム３００（３）の修正されたピクセルデータ１１０（１）（３）’に少なくとも部分的に基づいて、ディスプレイ上（例えば、ＨＭＤ１０２のディスプレイパネル上）における再投影フレーム３００（３）をレンダリングし得る。ブロック１３１２において実行される動作は、プロセス５００のブロック５１４に関して説明される動作と同様であり得る。

【０１２１】

図９～図１３のプロセスで説明される様々な技術（ＧＰＵ３０４によって出力される動きベクトル場を修正する様々な技術）は、任意の組み合わせおよび／または任意の順序で使用され得ることを理解されたい。例えば、図９を参照して説明した技術は、図１０および図１１を参照して説明した技術と組み合わせて使用することができ、それらの技術の一部または全部は、図１２を参照して説明した技術と組み合わせて使用することができ、それらの技術の一部または全部は、図１３を参照して説明した技術と組み合わせて使用することができる。すなわち、最適化された動きベクトル場を取得する堅牢な技術は、動きベクトル場の堅牢な修正において、図９～図１３を参照して一緒に記載される技術のすべてを利用し得る。さらに、ホストコンピュータ１０６および通信可能に連結されたＨＭＤ１０２を含む分散システムにおいて、図９～図１３のプロセスに説明された様々な技術（ＧＰＵ３０４によって出力される動きベクトル場を修正する様々な技術）の一部または全部が、ホストコンピュータ１０６上で実行され得ることを理解されたい。例えば、ホストコンピュータ１０６は、（例えば、減衰テクスチャを使用して）動きベクトルを閾値化し、かつ／または、修正された動きベクトルアレイを動き平滑化で使用するためにＨＭＤ１０２に送信する前に、フィルタを動きベクトルに適用するように構成され得る。このシナリオでは、修正された動きベクトルアレイ１１０（２）は、圧縮されたピクセルデータ１１０（１）から帯域外に送信され得、かつ／または修正された動きベクトルアレイ１１０（２）内のゼロ動きベクトルは、動きベクトルをＨＭＤ１０２に送信する前に動きベクトルデータを圧縮するようにランレングス符号化され得る。

【０１２２】

図１４は、本明細書に開示される実施形態による、ピクセルデータが動きベクトル推定のためにＧＰＵに入力される前に、以前にレンダリングされたフレームを回転させるための例示的なプロセス１４００のフロー図である。考察を目的として、プロセス１４００は、以前の図を参照して説明される。

【０１２３】

１４０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ピクセルデータ１１０（１）がＧＰＵ３０４への入力として提供されることになる複数の以前にレンダリングされたフレーム３００のうちの第１のフレーム３００（１）を、第１の回転されたフレーム１４０４（１）を取得するための回転量だけ回転させ得る。

【０１２４】

１４０６において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、複数の以前にレンダリングされたフレーム３００のうちの第２のフレーム３００（２）を、第２の回転されたフレーム１４０４（２）を取得するための回転量だけ回転させ得る。

【０１２５】

１４０８において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、回転されたフレーム１４０４（１）および１４０４（２）のピクセルデータをＧＰＵ３０４への入力として提供し得る。プロセス１４００は、プロセス５００のブロック１４０８からブロック５０４まで継続し得る（ページ外参照「Ａ」によって示されるように）。したがって、ブロック１４０８においてＧＰＵ３０４への入力として提供されるピクセルデータに基づいて、ブロック５０４においてＧＰＵ３０４から動きベクトルアレイ１１０（２）を受信し得、図５の動き平滑化アルゴリズムの残りの動作を実行して、動き平滑化される再投影フレーム３００（３）をレンダリングし得る。

【０１２６】

プロセス１４００は、移動しているかまたは動いているオブジェクトを真に表していな

い「奇妙な」動きベクトルを引き起こすことから、シーン内の水平および／または垂直のアーキテクチャのエッジの効果を軽減し得る。複数の以前にレンダリングされたフレーム３００（１）および３００（２）は、ブロック１４０２および１４０６で同じ量の回転だけ回転され得、または各フレーム３００（１）および３００（２）は、ブロック１４０２および１４０６においてそれぞれ異なる量の回転だけ回転され得る。いくつかの実施形態では、ブロック１４０２および１４０６における各フレームの回転量は、事前定義される（例えば、入力フレーム３００（１）および入力フレーム３００（２）を４５度回転させる）。フレーム３００（１）および３００（２）を４５度回転させることにより、シーン内の水平および垂直エッジが斜めに向けられ得、その結果、これらの斜めのエッジは、ＧＰＵ３０４のビデオエンコーダによって生成されるそれほど多くの「奇妙な」動きベクトルを引き起こさないことがある。いくつかの実施形態では、ブロック１４０２および１４０６における各フレームの回転量は、各フレーム３００のためのランダムな回転量であり得る。これは、４５度に位置合わせされ得る任意のエッジを一時的に隠し得る。

【０１２７】

図１５は、本明細書に開示される実施形態による、以前にレンダリングされたフレームのルマデータおよびクロマデータに基づいて生成される動きベクトルアレイ間を選択するための例示的なプロセス１５００のフロー図である。考察を目的として、プロセス１５００は、以前の図を参照して説明される。さらに、図５および図１５のページ外参照「Ｂ」によって示されるように、プロセス１５００は、図５のブロック５０６における動作の前に実行される動作を表し得、プロセス５００は、いくつかの実施形態では、ブロック５０６～５１４の動作を続行し得る。

【０１２８】

１５０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００に関連付けられているルマデータを提供し得る。例えば、第１のフレーム３００（１）に関連付けられている第１のピクセルデータ１１０（１）（１）は、第１のルマデータ、第１のクロマデータ、および／または追加のタイプのデータを含み得る。同様に、第２のフレーム３００（２）に関連付けられている第２のピクセルデータ１１０（１）（２）は、第２のルマデータ、第２のクロマデータなどの類似のタイプのデータを含み得る。したがって、ブロック１５０２において、第１のルマデータおよび第２のルマデータは、ＧＰＵ３０４への入力として提供され得る。

【０１２９】

１５０４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から、第１のルマデータおよび第２のルマデータに少なくとも部分的に基づいて、ＧＰＵ３０４のビデオエンコーダによって生成された第１の動きベクトルアレイ１１０（２）を受信し得る。

【０１３０】

１５０６において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、第１のフレーム３００（１）に関連付けられている第１のクロマデータおよび第２のフレーム３００（２）に関連付けられた第２のクロマデータをＧＰＵ３０４への入力として提供し得る。

【０１３１】

１５０８において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から、第１のクロマデータおよび第２のクロマデータに少なくとも部分的に基づいて、ＧＰＵ３０４のビデオエンコーダによって生成された第２の動きベクトルアレイ１１０（２）を受信し得る。

【０１３２】

１５１０において、ロジック（例えば、ホストコンピュータ１０６のロジック、または動きベクトル１１０（２）の複数のアレイを受信した後のＨＭＤ１０２のロジック）は、動きベクトル１１０（２）の第１のアレイおよび動きベクトル１１０（２）の第２のアレ

イの分析に基づいて、動きベクトル１１０（２）の第１のアレイまたは動きベクトル１１０（２）の第２のアレイのうちの１つを選択し得る。例えば、選択アルゴリズムは、動きベクトル１１０（３）の各場における動きベクトル１１０（２）の大きさおよび／または方向に基づいて、例として、最も不規則でないものを選択し得る。例えば、第２のフレーム３００（２）内で明るい点滅光（例えば、仮想爆発）が発生するとき、ルマデータは、ＧＰＵ３０４に、ある程度の閾値の大きさを超える大きさで動きベクトル１１０（２）を出力させ得、一方、同じ以前にレンダリングされたフレーム３００のクロマデータは、ＧＰＵ３０４に、極めて大きな大きさを有するそのような動きベクトルを含まない動きベクトル１１０（２）を出力させ得る。この場合、ブロック１５１０において、クロマデータから生成された動きベクトル１１０（２）のアレイが選択され得る。これは、選択アルゴリズムの１つの例に過ぎず、他のものが使用されてもよい。述べたように、プロセス１５００は、再投影フレーム３００（３）のピクセルデータ１１０（１）（３）が、ブロック１５１０からの動きベクトル１１０（２）の選択されたアレイに基づいて修正される、プロセス５００のブロック１５１０からブロック５０６に続行し得る。

【０１３３】

図１６は、本明細書に開示される実施形態による、動きベクトルの複数のアレイを取得し、アレイ間の差を決定し、動き平滑化のための決定された差に基づいて最終的な動きベクトルアレイを生成するための例示的なプロセス１６００のフロー図である。考察を目的として、プロセス１６００は、以前の図を参照して説明される。さらに、図５および図１６のページ外参照「Ｂ」によって示されるように、プロセス１６００は、図５のブロック５０６における動作の前に実行される動作を表し得、プロセス５００は、いくつかの実施形態では、ブロック５０６～５１４の動作を続行し得る。

【０１３４】

１６０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００に関連付けられている第１のピクセルデータ１１０（１）（１）を提供し得る。ブロック１６０２において実行される動作は、プロセス５００のブロック５０２に関して説明される動作と同様であり得る。

【０１３５】

１６０４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ブロック１６０２において入力として提供された第１のピクセルデータ１１０（１）（１）に基づいて、ＧＰＵ３０４から第１の動きベクトルアレイ１１０（２）を受信し得る。

【０１３６】

１６０６において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００に関連付けられている第２のピクセルデータ１１０（１）（２）を提供し得る。

【０１３７】

１６０８において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から、第２のピクセルデータ１１０（１）（２）に少なくとも部分的に基づいて、第２の動きベクトルアレイ１１０（２）を受信し得る。

【０１３８】

１６１０において、ロジック（例えば、ホストコンピュータ１０６のロジック、またはホストコンピュータ１０６から動きベクトル１１０（２）のアレイを受信した後のＨＭＤ１０２のロジック）は、第１のアレイ１１０（２）と第２のアレイ１１０（２）との間の動きベクトル１１０（２）の方向および／または大きさの違いを決定し得る。例えば、第１のアレイ１１０（２）の動きベクトル４０４と、第２のアレイ１１０（２）の対応する動きベクトル４０４との間で比較を行い、動きベクトルが、動きベクトルの方向または大きさのいずれかまたは両方に関して異なるかどうか、およびその場合、それらが異なるだけの量を決定し得る。

【０１３９】

１６１２において、ロジック（例えば、ホストコンピュータ１０６のロジック、またはＨＭＤ１０２のロジック）は、ブロック１６１０において決定されるように、動きベクトル１１０（２）の第１のアレイと動きベクトル１１０（２）の第２のアレイとの間の差に少なくとも部分的に基づいて、動きベクトル１１０（２）の最終アレイを生成し得る。例えば、動きベクトル１１０（２）の最終アレイは、第１のアレイ１１０（２）または第２のアレイ１１０（２）内の動きベクトルの個々のものを減衰させることから生じる、または各アレイ１１０（２）内の対応する動きベクトルの平均を表す平均動きベクトルを減衰させることから生じる動きベクトルのセットを表し得る。述べたように、プロセス１６００は、ブロック１６１２において生成された動きベクトル１１０（２）の最終アレイに基づいて、再投影フレーム３００（３）のピクセルデータ１１０（１）（３）が修正されるプロセス５００のブロック１６１２からブロック５０６に続行し得る。

【０１４０】

プロセス１６００がどのように使用され得るかの用例として、第１の動きベクトルアレイ１１０（２）が、入力フレーム３００（１）および３００（２）のピクセルデータ１１０（１）に基づいて、それらの元の（「ｙ上」）向きでブロック１６０４において受信され、第２の動きベクトルアレイ１１０（２）が、４５度回転された入力フレーム３００（１）および３００（２）のピクセルデータ１１０（１）に基づいてブロック１６０８において受信される例を考える。入力画像フレーム３００を回転させるこの概念は、図１４を参照して説明された。したがって、この用例では、第１のフレーム３００（１）および第２のフレーム３００（２）のピクセルデータ１１０（１）は、第１の動きベクトルアレイ１１０（２）を生成するためにブロック１６０２において入力として提供され得、第１の回転されたフレーム１４０４（１）および第２の回転されたフレーム１４０４（２）のピクセルデータ１１０（１）は、第２の動きベクトルアレイ１１０（２）を生成するためにブロック１６０６において入力として提供され得る。この用例では、２つの動きベクトルアレイ１１０（２）は、（例えば、第２のアレイ１１０（２）を逆方向に４５度回転させることによって）再位置合わせされ得、次いで、両方のアレイ１１０（２）における対応するベクトル間の角度差（方向）および／または長さ差（大きさ）を見ることによって不一致を決定するために比較され得る。実際の動き（例えば、シーン内で移動しているかまたは動いているオブジェクト）は、回転された入力フレーム３００に対して非常に類似した動きベクトルアレイを生成し得る一方で、誤って検出された動きは、非常に異なる動きベクトルアレイを生成し得る場合があり得る。この用例では、最終的な動きベクトルアレイ１１０（２）は、ブロック１６１２において、２つの正規化されたベクトルのドット積によって各アレイ１１０（２）における２つの対応するベクトルの平均の長さを減衰させることによって生成され得る。いくつかの実施形態では、図１０および図１１を参照して説明されるもののような減衰テクスチャは、ブロック１６１２において利用されて、最終的なアレイを生成し得る。例えば、プロセス１１００のブロック１１０２において決定された差分ベクトルのセットは、ブロック１６０４において受信された動きベクトル１１０（２）の第１のアレイと、ブロック１６０８において受信された動きベクトル１１０（２）の第２のアレイとの間の差分に基づき得、結果として生じる減衰テクスチャは、ブロック１６１２において、第１のアレイ１１０（２）、第２のアレイ１１０（２）、または２つのアレイ１１０（２）間のベクトルの組み合わせ（例えば、平均）を減衰させるために使用され得る。

【０１４１】

プロセス１６００がどのように使用され得るかの別の用例として、複数の以前にレンダリングされたフレーム３００の入力テクスチャのために第１のミップレベル（ミップマップレベル）が生成され、この第１のミップレベルのための対応するピクセルデータ１１０（１）が、ブロック１６０２において、動きベクトル１１０（２）の第１のアレイを生成するためにＧＰＵ３０４への入力として提供される例を考える。一方、第２のミップレベルは、複数の以前にレンダリングされたフレーム３００の入力テクスチャに対して生成され得、この第２のミップレベルに対する対応するピクセルデータ１１０（１）は、第２の

動きベクトルアレイ１１０（２）を生成するためにブロック１６０６でＧＰＵ３０４への入力として提供され得る。これは、任意の数の動きベクトルの対応するアレイを生成する任意の数のミップレベルに対しても行うことができる。いくつかの実施形態では、３つまたは４つの動きベクトルアレイ１１０（２）がブロック１６１０の前に受信されるように、入力テクスチャの３つまたは４つのミップレベルを生成することができる。いくつかの実施形態では、各ミップレベルは、以前のミップレベルの幅および高さの半分（例えば、面積の２５％）である。ブロック１６１０において、この用例では、異なるミップレベルから生成されたアレイ１１０（２）の対間の差が決定され得、ブロック１６１２において、決定された差に基づいて、最終的な動きベクトルアレイ１１０（２）が生成され得る。例えば、ロジックは、異なるミップレベルについての複数の動きベクトルアレイ１１０（２）にわたる異常を検出し得、異常な動きベクトルを減衰させる（例えば、ゼロに減少させる）ことができる。入力フレーム３００の異なるミップレベルを使用して異なる動きベクトルアレイ１１０（２）を生成するというこの例は、シーン内に繰り返しパターンの大きな領域（例えば、壁紙）が存在し、ＧＰＵ３０４のビデオエンコーダが、移動しているかまたは動いているオブジェクトが存在しないにもかかわらず、動きベクトル解像度に対する繰り返しパターンの頻度に起因して多くの動きがあることを考慮して、多くの高い大きさの動きベクトルを生成し得る状況において役立ち得る。したがって、プロセス１６００は、異なるミップレベルの動きベクトル１１０（２）のアレイを生成するために使用されるとき、繰り返しパターンの異なる周波数にわたる異常を検出するのに役立ち得、検出されると、減衰テクスチャを使用して、それらの動きベクトル１１０（２）を減衰させる（例えば、ゼロ化する）ことができる。

【０１４２】

プロセス１６００がどのように使用され得るかのさらに別の例として、複数のアレイ１１０（２）が異なる解像度で取得され、ブロック１６１０において、第１のアレイ１１０（２）内の単一の動きベクトルを、第２のアレイ１１０（２）内の複数の対応する動きベクトルと比較することによって（例えば、第２のアレイ１１０（２）内の複数の対応する動きベクトルの方向および／または大きさに関して平均と比較することによって）差が決定される例を考える。異なる解像度で動きベクトルアレイ１１０（２）を使用することは、ブロック１６１２において動きベクトルの最終的なアレイを生成するために有用な差をもたらし得る。

【０１４３】

図１７は、本明細書に開示される実施形態による、画像領域の異なる部分について異なる解像度で複数の動きベクトルアレイを取得するための例示的なプロセス１７００のフロー図である。考察を目的として、プロセス１７００は、以前の図を参照して説明される。さらに、図５および図１７のページ外参照「Ｂ」によって示されるように、プロセス１７００は、図５のブロック５０６における動作の前に実行される動作を表し得、プロセス５００は、いくつかの実施形態では、ブロック５０６～５１４の動作を続行し得る。

【０１４４】

１７０２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、以前にレンダリングされたフレーム３００と関連付けられているピクセルデータ１１０（１）を提供し得る。ブロック１７０２において実行される動作は、プロセス５００のブロック５０２に関して説明される動作と同様であり得る。

【０１４５】

１７０４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から第１の動きベクトルアレイ１１０（２）（１）を受信し得、第１の動きベクトルアレイ１１０（２）（１）は、第１の解像度で受信されるか、または別様に、第１の解像度にアップサンプリングまたはダウンサンプリングされる。

【０１４６】

１７０６において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、動きベクトル１１０（２）（１）の第１のアレイに少なくとも部分的に基づいて、ディスプ

レイにわたる画像領域１７０５の第１の部分１７０５（Ａ）内の動きの不在を検出し得、動きベクトル１１０（２）（１）の第１のアレイに少なくとも部分的に基づいて、画像領域１７０５の第２の部分１７０５（Ｂ）内のオブジェクト３０２の動きを検出し得る。例えば、非ゼロ動きベクトルは、画像領域１７０５の右半分で検出され得、一方、画像領域１７０５の左半分は、任意の非ゼロ動きベクトルを含まないゼロ値の動きベクトルを含み得る。

【０１４７】

１７０８において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、第１のピクセルデータ１１０（１）（１）（第１のフレーム３００（１）に関連付けられている）の第１の部分および第２のピクセルデータ１１０（１）（２）（第２のフレーム３００（２）に関連付けられている）の第１の部分を提供し得る。第１のピクセルデータ１１０（１）（１）の第１の部分および第２のピクセルデータ１１０（１）（２）の第１の部分は、各々、ブロック１７０６において動きのないことが検出された画像領域１７０５の第１の部分１７０５（Ａ）に対応し得る。

【０１４８】

１７１０において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から、第１のピクセルデータ１１０（１）（１）の第１の部分および第２のピクセルデータ１１０（１）（２）の第１の部分に少なくとも部分的に基づいて、ＧＰＵ３０４のビデオエンコーダによって生成された第２の動きベクトルアレイ１１０（２）（２）を受信し得る。この第２のアレイの動きベクトル１１０（２）（２）は、第１のアレイの動きベクトル１１０（２）（１）の第１の解像度よりも高い第２の解像度で生成され得る。動きの不在が検出された画像領域１７０５の第１の部分１７０５（Ａ）についてのより高い解像度の動きベクトル場１１０（２）（２）を取得することは、より高い解像度での動きベクトル場１１０（２）（２）が、大規模な移動が検出されなかった画像領域１７０５の第１の部分１７０５（Ａ）の小規模な移動を検出するのに役立ち得るという考えに基づき得る。

【０１４９】

１７１２において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４への入力として、第１のピクセルデータ１１０（１）（１）（第１のフレーム３００（１）に関連付けられている）の第２の部分および第２のピクセルデータ１１０（１）（２）（第２のフレーム３００（２）に関連付けられている）の第２の部分を提供し得る。第１のピクセルデータ１１０（１）（１）の第２の部分および第２のピクセルデータ１１０（１）（２）の第２の部分は、各々、ブロック１７０６においてオブジェクト３０２の動きが検出された画像領域１７０５の第２の部分１７０５（Ｂ）に対応し得る。

【０１５０】

１７１４において、ロジック（例えば、ホストコンピュータ１０６のロジック）は、ＧＰＵ３０４から、第１のピクセルデータ１１０（１）（１）の第２の部分および第２のピクセルデータ１１０（１）（２）の第２の部分に少なくとも部分的に基づいて、ＧＰＵ３０４のビデオエンコーダによって生成された第３の動きベクトルアレイ１１０（２）（３）を受信し得る。この第３の動きベクトルアレイ１１０（２）（３）は、第２の動きベクトルアレイ１１０（２）（２）の第２の解像度よりも低い解像度である第３の解像度で生成され得る。動きが検出された画像領域１７０５の第２の部分１７０５（Ｂ）に対するこの比較的低解像度の動きベクトル場１１０（２）（３）は、第１の動きベクトル場１１０（２）（１）の第１の解像度ですでに大規模な移動が検出された画像領域１７０５の第２の部分１７０５（Ｂ）の移動を検出するために、より高い解像度の動きベクトル場が必要でない場合があるという考えに基づき得る。

【０１５１】

１７１６において、ロジック（例えば、ホストコンピュータ１０６のロジック、またはホストコンピュータ１０６からの動きベクトルアレイ１１０（２）を受信した後のＨＭＤ１０２のロジック）は、動き平滑化のために動きベクトルの第２のアレイ１１０（２）（

２）および動きベクトルの第３のアレイ１１０（２）（３）を使用し得る。プロセス１７００は、再投影フレーム３００（３）のピクセルデータ１１０（１）（３）は、ブロック１７１６からの動きベクトル１１０（２）に基づいて修正される、ブロック１７１６からプロセス５００のブロック５０６に続行し得る。例では、動き平滑化のためにブロック１７１６で動きベクトルの複数のアレイを使用することは、プロセス１６００のブロック１６１２を参照して説明される動作を含み得る。

【０１５２】

図１８Ａおよび図１８Ｂは、本明細書に開示される実施形態による、ＨＭＤ１０２およびホストコンピュータ１０６を利用するシステムの２つの代替的なセットアップを示す。図１に描かれているように、例示的な実装形態は、ホストコンピュータ１０６が、ユーザ１０４によって装着されるＨＭＤ１０２と環境内に併置される。例えば、ホストコンピュータ１０６は、ホストコンピュータ１０６がＨＭＤ１０２と同じ部屋または異なる部屋に位置しているかどうかにかかわらず、ユーザ１０４がハウス内でＨＭＤ１０２を使用している間、ユーザ１０４のハウス内に位置し得る。代替的に、モバイルコンピューティングデバイス（例えば、タブレットまたはラップトップ）の形態のホストコンピュータ１０６は、ユーザ１０４の背面のバックパック内で運ばれてもよく、それによって、より大きなモビリティを可能にする。例えば、ユーザ１０４は、そのようなシステムを使用しながら、公共の公園内に位置することができる。

【０１５３】

図１８Ａは、ホストコンピュータ１０６が、ＨＭＤ１０２に対して地理的に離れた場所に位置する１つ以上のサーバコンピュータを表す、代替の実装形態を示している。この場合、ＨＭＤ１０２は、無線ＡＰ（ＷＡＰ）、基地局などのアクセスポイント（ＡＰ）１８００を介してホストコンピュータ１０６に通信可能に連結され得る。用例では、データは、ホストコンピュータ１０６とＨＭＤ１０２との間で、ＡＰ１８００を介して、例えばインターネットを介してデータをストリーミングすることによって、交換される（例えば、ストリーミングされる）。

【０１５４】

図１８Ｂは、ホストコンピュータ１０６が、ラップトップまたはタブレットコンピュータなどの中間コンピューティングデバイス１８０２を介してＨＭＤ１０２に通信可能に連結される、さらに別の代替の実装形態を示している。図１８Ａと図１８Ｂとの間の差は、図１８ＡのＡＰ１８００が、レンダリングを行わないデータルーティングデバイスとして単純に作用し得、一方で、図１８Ｂの中間コンピューティングデバイス１８０２は、レンダリングワークロードの一部分を実行し得る。すなわち、ホストコンピュータ１０６とＨＭＤ１０２との間でレンダリングワークロードを分岐させる代わりに、レンダリングワークロードは、ホストコンピュータ１０６、中間コンピューティングデバイス１８０２、およびＨＭＤ１０２といった３つのデバイスなどの３つ以上のデバイスの間で区分けされ得る。図１８Ｂのシナリオにおいて、ホストコンピュータ１０６は、本明細書に記載されるように、ピクセルデータ１１０（１）を生成し得、中間コンピューティングデバイス１８０２は、ピクセルデータ１１０（１）を修正するためのレンダリング動作の第１のセットを実行し得、ＨＭＤ１０２は、修正されたピクセルデータを修正するためのレンダリング動作の最終セットを実行し得る。

【０１５５】

図１９は、本明細書に開示される実施形態による、ＨＭＤ１９００の例示的なコンポーネントを示しており、例えば、ＶＲヘッドセットが埋め込まれ得る。ＨＭＤ１０２は、ユーザ１０４によって（例えば、ユーザ１０４の頭部に）装着されることになるデバイスとして実装され得る。いくつかの実施形態では、ＨＭＤ１０２は、ユーザ１０４が、ユーザ１０４の頭部の周囲に収まるようにサイズ決めされる固設機構（例えば、調整可能バンド）を使用して、ＨＭＤ１０２をユーザの頭部に固設することを可能にすることなどによって、頭部に装着可能であり得る。いくつかの実施形態では、ＨＭＤ１０２は、ニアアイまたはニアツーアイディスプレイを含む仮想現実（ＶＲ）または拡張現実（ＡＲ）ヘッドセ

ットを含む。したがって、「ウェアラブルデバイス」、「ウェアラブル電子デバイス」、「ＶＲヘッドセット」、「ＡＲヘッドセット」、および「ヘッドマウントディスプレイ（ＨＭＤ）」という用語は、本明細書では互換的に使用されて、デバイス１０２を指し得る。しかしながら、これらのタイプのデバイスは、ＨＭＤ１０２の単なる例であることを理解されたい、ＨＭＤ１０２は、様々な他のフォームファクタで実装され得ることを理解されたい。

【０１５６】

例示された実装形態では、ＨＭＤ１０２は、１つ以上のプロセッサ１９００およびメモリ１９０２（例えば、コンピュータ可読媒体１９０２）を含む。いくつかの実装形態では、プロセッサ１９００は、中央処理装置（ＣＰＵ）、ＧＰＵ１９００（１）、ＣＰＵおよびＧＰＵ１９００（１）の両方、マイクロプロセッサ、デジタル信号プロセッサ、または当該技術分野で既知の他の処理ユニットもしくは構成要素を含み得る。代替的にまたは追加的に、本明細書に記載されている機能性は、１つ以上のハードウェアロジックコンポーネントによって、少なくとも部分的に実行することができる。例えば、非限定的に、使用され得るハードウェアロジックコンポーネントの例示的なタイプとしては、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、特定用途向け集積回路（ＡＳＩＣ）、特定用途向け標準製品（ＡＳＳＰ）、システムオンチップシステム（ＳＯＣ）、複合プログラマブル論理デバイス（ＣＰＬＤ）などが挙げられる。加えて、プロセッサ１９００の各々は、プログラムモジュール、プログラムデータ、および／または１つ以上のオペレーティングシステムも記憶し得る、その独自のローカルメモリを保有し得る。

【０１５７】

メモリ１９０２は、揮発性および不揮発性メモリ、コンピュータ可読命令、データ構造、プログラムモジュール、または他のデータなどの情報を記憶するための任意の方法または技術で実装された取り外し可能および取り外し不可能な媒体が挙げられ得る。そのようなメモリとしては、限定されるものではないが、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリもしくは他のメモリ技術、ＣＤ－ＲＯＭ、デジタル多目的ディスク（ＤＶＤ）もしくは他の光学ストレージ、磁気カセット、磁気テープ、磁気ディスクストレージもしくは他の磁気ストレージデバイス、ＲＡＩＤストレージシステム、または所望の情報を記憶するために使用され得、コンピューティングデバイスによってアクセスされ得る、任意の他の媒体が挙げられる。メモリ１９０２は、コンピュータ可読記憶媒体（「ＣＲＳＭ」）として実装され得、それは、メモリ１９０２上に記憶された命令を実行するために、プロセッサ１９００によってアクセス可能である任意の利用可能な物理媒体であってもよい。１つの基本的な実装形態では、ＣＲＳＭは、ランダムアクセスメモリ（「ＲＡＭ」）およびフラッシュメモリを含み得る。他の実装形態では、ＣＲＳＭには、読み出し専用メモリ（「ＲＯＭ」）、電気的に消去可能なプログラマブル読み出し専用メモリ（「ＥＥＰＲＯＭ」）、または所望の情報を記憶するために使用され得、プロセッサ１９００によってアクセスされ得る、任意の他の有形媒体が含まれ得るが、これらに限定されない。

【０１５８】

一般に、ＨＭＤ１０２は、本明細書に説明される技術、機能、および／または動作を実装するように構成されるロジック（例えば、ソフトウェア、ハードウェア、および／またはファームウェアなど）を含み得る。コンピュータ可読媒体１９０２は、命令、データストアなどの、様々なモジュールを含むものとして示され、これらのモジュールは、本明細書に説明される技術、機能、および／または動作を実施するためにプロセッサ１９００上で実行するように構成され得る。コンピュータ可読媒体１９０２に記憶され、かつプロセッサ１９００上で実行可能なものとして、数個の例示的な機能モジュールが示されているが、同じ機能が、代替的に、ハードウェア、ファームウェア内に、またはシステムオンチップ（ＳＯＣ）および／もしくは他のロジックとして実装されてもよい。

【０１５９】

オペレーティングシステムモジュール１９０４は、他のモジュールの便宜のために、ＨＭＤ１０２内に連結されたハードウェアを管理するように構成され得る。加えて、いくつ

かの事例では、ＨＭＤ１０２は、メモリ１９０２に記憶されているか、またはさもなければＨＭＤ１０２にアクセス可能な１つ以上のアプリケーション１９０６を含み得る。この実装では、アプリケーション１９０６は、ゲームアプリケーション１９０８を含む。しかしながら、ＨＭＤ１０２は、任意の数またはタイプのアプリケーションを含み得、ここに示される特定の例に限定されない。コンポジタ１９１０は、ＨＭＤ１０２の他のロジックと組み合わせて、動きが平滑化され、再投影フレーム３００をレンダリングするために、本明細書に説明記載される動き平滑化技術を実行するように構成され得る。

【０１６０】

一般的に、ＨＭＤ１０２は、入力デバイス１９１２および出力デバイス１９１４を有する。入力デバイス１９１２は、制御ボタンを含み得る。いくつかの実装形態では、１つ以上のマイクロフォンが、ユーザ音声入力などのオーディオ入力を受信するための入力デバイス１９１２として機能し得る。いくつかの実装形態では、１つ以上のカメラまたは他のタイプのセンサ（例えば、慣性測定ユニット（ＩＭＵ））が、ユーザ１０４の手および／または頭部の動きなどの、ジェスチャ入力を受信するための入力デバイス１９１２として機能し得る。いくつかの実施形態において、追加の入力デバイス１９１２が、キーボード、キーパッド、マウス、タッチスクリーン、ジョイスティックなどの形態で提供され得る。他の実施形態において、ＨＭＤ１０２は、キーボード、キーパッド、または他の同様の形態の機械的入力を省略してもよい。代わりに、ＨＭＤ１０２は、入力デバイス１９１２の比較的単純な形態、ネットワークインターフェース（無線または有線ベース）、電源、および処理／メモリ機能で実装され得る。例えば、１つ以上の入力部品の限定されたセット（例えば、構成を開始する、電源をオン／オフするなどのための専用ボタン）が用いられ得、ＨＭＤ１０２は、その後に使用され得る。一実装形態では、入力デバイス１９１２は、音量を増加／減少させるための基本的な音量制御ボタン、ならびに電源およびリセットボタンなどの制御機構を含み得る。

【０１６１】

出力デバイス１９１４は、ディスプレイ１９１６を含み得、これは、１つ以上のディスプレイパネル（例えば、ディスプレイパネルのステレオ対）を含み得る。出力デバイス１９１４は、非限定的に、発光素子（例えば、ＬＥＤ）、触覚感覚を生み出すためのバイブレータ、および／またはスピーカ（例えば、ヘッドフォン）などをさらに含み得る。例えば、電源がオンであるときなどの状態を示すための単純な発光素子（例えば、ＬＥＤ）も存在し得る。

【０１６２】

ＨＭＤ１０２は、ネットワークへの無線接続を容易にするために、アンテナ１９２０に連結された無線ユニット１９１８をさらに含み得る。無線ユニット１９１８は、Ｗｉ－Ｆｉ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、無線周波数（ＲＦ）などの、様々な無線技術のうちの１つ以上を実装し得る。ＨＭＤ１０２は、ネットワーク、接続された周辺機器（ＰＣ、ゲームコンソールなどのホストコンピュータ１０６を含む）、または他の無線ネットワークと通信するプラグインネットワークデバイスへの有線接続を容易にするための物理ポートをさらに含み得ることを理解されたい。

【０１６３】

ＨＭＤ１０２は、１つ以上の光学要素を使用して、電子ディスプレイ１９１６からユーザの眼に光を向ける光学サブシステム１９２２をさらに含み得る。光学サブシステム１９２２は、非限定的に、アパーチャ、レンズ（例えば、フレネルレンズ、凸レンズ、凹レンズなど）、フィルタなどを含む、異なる光学素子の様々なタイプおよび組み合わせを含み得る。いくつかの実施形態では、光学サブシステム１９２２内の１つ以上の光学要素は、反射防止コーティングなどの１つ以上のコーティングを有し得る。光学サブシステム１９２２による画像光の拡大は、電子ディスプレイ１９１６が、より大型のディスプレイよりも物理的により小型で、軽量で、より低消費電力であることを可能にする。加えて、画像光の拡大は、表示されたコンテンツ（例えば、画像）の視野（ＦＯＶ）を増加させ得る。例えば、表示されたコンテンツのＦＯＶは、表示されたコンテンツが、ユーザのＦＯＶの

ほぼすべて（例えば、対角１２０～１５０度）、場合によってはすべてを使用して提示されるようなものである。ＡＲアプリケーションは、より狭いＦＯＶ（例えば、約４０度のＦＯＶ）を有していてもよい。光学サブシステム１９２２は、非限定的に、樽型歪み、ピンクッション歪み、長手方向色収差、横収差、球面収差、像面湾曲、非点収差などの、１つ以上の光学誤差を補正するように設計され得る。いくつかの実施形態では、表示のために電子ディスプレイ１９１６に提供されるコンテンツは、事前に歪められており、光学サブシステム１９２２は、コンテンツに基づいて生成された、電子ディスプレイ１９１６からの画像光を受光する場合にその歪みを補正する。

【０１６４】

ＨＭＤ１０２は、動き、位置、および方向データを生成するために使用されるセンサなどの、１つ以上のセンサ１９２４をさらに含み得る。これらのセンサ１９２４は、ジャイロスコープ、加速度計、磁力計、ビデオカメラ、カラーセンサ、または他の動き、位置、および方向センサであり得るか、またはそれらを含み得る。センサ１９２４はまた、動き、位置、および方向データを生成するためにカメラまたはカラーセンサによって外部から視認され得る一連の能動または受動マーカなどの、センサのサブ部分を含み得る。例えば、ＶＲヘッドセットは、その外部に、外部カメラで見た場合、または光（例えば、赤外光または可視光）で照らした場合に、反射器または光（例えば、赤外光または可視光）などの複数のマーカを含み得、動き、位置、および方向データを生成するために、ソフトウェアによる解釈のための１つ以上の基準点を提供し得る。ＨＭＤ１０２は、ＨＭＤ１０２の環境内の基地局によって投影または散布される光（例えば、赤外光または可視光）に対して感応性である光センサを含み得る。

【０１６５】

一例では、センサ１９２４は、慣性測定ユニット（ＩＭＵ）１９２６を含み得る。ＩＭＵ１９２６は、加速度計、ジャイロスコープ、磁力計、および／もしくは動きの検出、ＩＭＵ１９２６と関連付けられた誤差の補正、またはそれらのいくつかの組み合わせに好適な他のセンサから受信された測定信号に基づいて、較正データを生成する電子デバイスであり得る。測定信号に基づいて、ＩＭＵ１９２６のような、そのような動きベースのセンサは、ＨＭＤ１０２の初期位置に対するＨＭＤ１０２の推定された位置を示す較正データを生成し得る。例えば、複数の加速度計が並進運動（前方／後方、上／下、左／右）を測定し、複数のジャイロスコープが回転運動（例えば、ピッチ、ヨー、およびロール）を測定してもよい。ＩＭＵ１９２６は、例えば、測定信号を迅速にサンプリングし、サンプリングされたデータからＨＭＤ１０２の推定された位置を計算し得る。例えば、ＩＭＵ１９２６は、経時的に加速度計から受信された測定信号を統合して速度ベクトルを推定し得、経時的に速度ベクトルを統合して、ＨＭＤ１０２上の基準点の推定された位置を決定する。基準点は、ＨＭＤ１９００の位置を説明するために使用され得る点である。基準点は、一般的に、空間内の点として定義され得るが、様々な実施形態では、基準点は、ＨＭＤ１０２内の点（例えば、ＩＭＵ１９２６の中心）として定義される。代替的に、ＩＭＵ１９２６は、サンプリングされた測定信号を外部コンソール（または他のコンピューティングデバイス）に提供し、外部コンソールが較正データを決定する。

【０１６６】

センサ１９２４は、センサデータを高レートで提供するために、比較的高い周波数で動作し得る。例えば、センサデータは、１０００Ｈｚのレート（または１ミリ秒ごとに１つのセンサの読み取り）で生成され得る。このようにして、１秒間に１０００回の読み取りが行われる。センサがこの速度で（またはそれ以上の速度で）これだけのデータを生成する場合、動きを予測するために使用されるデータセットは、数十ミリ秒ほどの比較的短い期間でも非常に大きくなる。

【０１６７】

言及されるように、いくつかの実施形態では、センサ１９２４は、３Ｄ空間内のＨＭＤ１０２の位置および／または方向、姿勢などを追跡する目的で、ＨＭＤ１０２の環境内の基地局によって放出される光に対して感応性である光センサを含み得る。位置および／ま

たは方向の計算は、光パルスのタイミング特性、およびセンサ１９２４によって検出される光の有無に基づき得る。

【０１６８】

ＨＭＤ１０２は、眼追跡モジュール１９２８をさらに含み得る。ＨＭＤ１０２の内部のカメラまたは他の光学センサは、ユーザの眼の画像情報を捕捉し得、眼追跡モジュール１９２８は、捕捉された情報を使用して、ねじれおよび回転の大きさ（すなわち、ロール、ピッチ、およびヨー）ならびに各眼の注視方向を含む、瞳孔間距離、ＨＭＤ１０２に対する各眼の３次元（３Ｄ）位置を決定し得る（例えば、歪み調整の目的で）。一例では、赤外光が、ＨＭＤ１０２内で放出され、各眼から反射される。反射光は、ＨＭＤ１０２のカメラによって受光または検出され、各眼によって反射された赤外光の変化から眼球回転を抽出するために分析される。ユーザ１０４の眼を追跡するための多くの方法が、眼追跡モジュール１９２８によって使用され得る。したがって、眼追跡モジュール１９２８は、各眼の最大６自由度（すなわち、３Ｄ位置、ロール、ピッチ、およびヨー）を追跡し得、追跡された量の少なくともサブセットが、注視点（すなわち、ユーザが見ている仮想シーン内の３Ｄ場所または位置）を推定するためにユーザ１０４の２つの眼から組み合わせられ得る。例えば、眼追跡モジュール１９２８は、過去の測定値からの情報、ユーザ１０４の頭部の位置を識別する測定値、および電子ディスプレイ１９１６によって提示されるシーンを記述する３Ｄ情報を統合し得る。したがって、ユーザ１０４の眼の位置および向きの情報は、ユーザ１０４が見ている、ＨＭＤ１０２によって提示された仮想シーン内の注視点を決定するために使用される。

【０１６９】

ＨＭＤ１０２は、頭部追跡モジュール１９３０をさらに含み得る。頭部追跡モジュール１９３０は、上述のように、センサ１９２４のうちの１つ以上を利用して、ユーザ１０４の頭部の回転を含む頭部の動きを追跡し得る。例えば、頭部追跡モジュール１９３０は、ＨＭＤ１０２の最大６自由度（すなわち、３Ｄ位置、ロール、ピッチ、およびヨー）を追跡し得る。これらの計算は、一連のフレーム３００のすべてのフレーム３００において行われ得るため、アプリケーション（例えば、ビデオゲーム）は、次のフレーム３００内でシーンをどのようにレンダリングするかを決定し得、かつ／またはコンポジタ１９１０は、頭部の位置および向きに応じて、再投影フレーム３００をどのようにレンダリングするかを決定し得る。いくつかの実施形態では、頭部追跡モジュール１９３０は、現在および／または過去のデータに基づいて、ＨＭＤ１０２の将来の位置および／または配向を予測するように構成される。これは、ユーザ１０４がディスプレイ１９１６上で光（したがって、画像）を実際に見る前に、アプリケーションがフレーム３００をレンダリングするように求められるためである。したがって、次のフレーム３００は、フレーム３００をレンダリングする前のおよそ２５～３０ミリ秒（ｍｓ）などのより早い時点で行われた、頭部の位置および／または配向のこの将来の予測に基づいて、レンダリングすることができる。ホストコンピュータ１０６がＨＭＤ１０２に通信可能に（例えば、無線で）連結される分散システムでは、頭部姿勢の将来の予測は、ネットワークレイテンシ、圧縮動作などを考慮するために、フレーム３００の点灯時間の３０ｍｓ以上前に行われ得る。頭部追跡モジュール１９３０によって提供される回転データは、任意の好適な測定単位で、ＨＭＤ１０２の回転方向およびＨＭＤ１０２の回転量の両方を決定するために使用され得る。例えば、回転方向は、左、右、上、および下に対応する正または負の水平方向および正または負の垂直方向の観点で単純化されて出力され得る。回転量は、度、ラジアンなどの単位であり得る。角速度が、ＨＭＤ１０２の回転速度を決定するために計算されてもよい。

【０１７０】

図２０は、本明細書に開示される技術が実装され得る、ホストコンピュータ１０６の例示的なコンポーネントを示す。例示された実装形態では、ホストコンピュータ１０６は、１つ以上のプロセッサ２０００およびメモリ２００２（例えば、コンピュータ可読媒体２００２）を含む。いくつかの実装形態では、プロセッサ２０００は、ＣＰＵ、ＧＰＵ３０４、ＣＰＵおよびＧＰＵ３０４の両方、マイクロプロセッサ、デジタルシグナルプロセッ

サ、または当該技術分野で既知の他の処理ユニットもしくはコンポーネントを含み得る。代替的にまたは追加的に、本明細書に記載されている機能性は、１つ以上のハードウェアロジックコンポーネントによって、少なくとも部分的に実行することができる。例えば、非限定的に、使用することができる例示的なタイプのハードウェアロジックコンポーネントとしては、ＦＰＧＡ、ＡＳＩＣ、ＡＳＳＰ、ＳＯＣ、ＣＰＬＤなどが挙げられる。加えて、プロセッサ２０００の各々は、プログラムモジュール、プログラムデータ、および／または１つ以上のオペレーティングシステムも記憶し得る、その独自のローカルメモリを保有し得る。

【０１７１】

メモリ２００２は、揮発性および不揮発性メモリ、コンピュータ可読命令、データ構造、プログラムモジュール、または他のデータなどの情報を記憶するための任意の方法または技術で実装された取り外し可能および取り外し不可能な媒体が挙げられ得る。そのようなメモリは、これらに限定されるものではないが、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリもしくは他のメモリ技術、ＣＤ－ＲＯＭ、ＤＶＤ、もしくは他の光メモリ、磁気カセット、磁気テープ、磁気ディスクストレージもしくは他の磁気ストレージデバイス、ＲＡＩＤストレージシステム、または所望の情報を記憶するために使用することができ、かつコンピューティングデバイスからアクセスすることができる任意の他の媒体を含む。メモリ２００２は、ＣＲＳＭとして実装され得、それは、メモリ２００２上に記憶された命令を実行するために、プロセッサ２０００によってアクセス可能である任意の利用可能な物理媒体であってもよい。１つの基本的な実装形態では、ＣＲＳＭには、ＲＡＭおよびフラッシュメモリを含み得る。他の実装形態では、ＣＲＳＭには、ＲＯＭ、ＥＥＰＲＯＭ、または所望の情報を記憶するために使用され得、プロセッサ２０００によってアクセスされ得る、任意の他の有形媒体が含まれ得るが、これらに限定されない。

【０１７２】

一般に、ホストコンピュータ１０６は、本明細書に説明される技術、機能、および／または動作を実装するように構成されるロジック（例えば、ソフトウェア、ハードウェア、および／またはファームウェアなど）を含み得る。コンピュータ可読媒体２００２は、命令、データストアなどの、様々なモジュールを含むものとして示され、これらのモジュールは、本明細書に説明される技術、機能、および／または動作を実施するためにプロセッサ２０００上で実行するように構成され得る。オペレーティングシステム２００４、ビデオゲーム（またはゲームアプリ２０１０）などのアプリケーション２００８を含むビデオゲームクライアント２００６、およびレンダリングコンポーネント２０１２の形態の例示的な機能モジュールが、コンピュータ可読媒体２００２に記憶され、プロセッサ２０００上で実行可能であるように示される。いくつかの実施形態では、追加のまたは異なる機能モジュールが、コンピュータ可読媒体２００２に記憶され、かつプロセッサ２０００上で実行可能であり得る。

【０１７３】

オペレーティングシステム２００４は、他のモジュールの利益のために、ホストコンピュータ１０６内の、およびホストコンピュータ１０６に連結されたハードウェアを管理するように構成され得る。このビデオゲームクライアント２００６は、ビデオゲーム（またはビデオゲームプログラム）などのプログラムを起動および実行するように構成されている実行可能クライアントアプリケーションを表し得る。言い替えると、ビデオゲームクライアント２００６には、ＨＭＤ１０２およびホストコンピュータ１０６を含むシステム上でビデオゲームをプレイするために使用可能なゲームソフトウェアが含まれ得る。ビデオゲームクライアント２００６がインストールされた状態で、ホストコンピュータ１０６は、コンピュータネットワーク（例えば、インターネット）経由で遠隔システムからビデオゲームを受信（例えば、ダウンロード、ストリーミングなど）し、ビデオゲームクライアント２００６を介してビデオゲームを実行するための能力を有し得る。この目的のために、ホストコンピュータ１０６上でのダウンロードおよび実行のために別々に購入することができる直接購入モデル、サブスクリプションベースモデル、ビデオゲームがある期間の

レンタルまたはリースされるコンテンツ配信モデルなどの任意のタイプのコンテンツ配信モデルを利用することができる。したがって、ホストコンピュータ１０６は、ビデオゲームライブラリ２０１４内に、１つ以上のビデオゲームを含み得る。これらのビデオゲームは、ビデオゲームクライアント２００６をロードすることによって、取り出され、そして実行され得る。例では、ユーザ１０４は、ビデオゲームクライアント２００６をロードし、ビデオゲームを選択してビデオゲームの実行を開始することによって、購入してビデオゲームライブラリ２０１４にダウンロードした複数のビデオゲームのうちの１つをプレイすることを選択し得る。ビデオゲームクライアント２００６は、ユーザが資格情報（例えば、ユーザアカウント、パスワードなど）を使用してビデオゲームサービスにログインすることを可能にし得る。

【０１７４】

ホストコンピュータ１０６上で実行するアプリケーション２００８は、グラフィックスベースのアプリケーション２００８（例えば、ビデオゲーム２０１０）であり得る。アプリケーション２００８は、一連のフレームに関するピクセルデータを生成するように構成され、ピクセルデータは、最終的に、ＨＭＤ１０２のディスプレイパネル１９１６上に対応する画像を提示するために使用される。ランタイム中、所与のフレームについて、レンダリングコンポーネント２０１２は、フレームのための予測「点灯時間」を決定し得る。フレームのためのこの予測「点灯時間」は、ＨＭＤ１０２のディスプレイパネル１９１６の発光素子がフレームのために点灯する時間を表す。この予測は、とりわけ、ホストコンピュータ１０６とＨＭＤ１０２との間の無線通信リンクの固有のレイテンシ、ならびにフレームバッファからのピクセルの予測されるレンダリング時間および／または既知のスキャンアウト時間を考慮することができる。言い換えれば、無線通信リンクについての予測は、有線通信リンクについての予測とは異なり得る。例えば、レンダリングコンポーネント２０１２は、有線通信リンクの場合、将来の第１の時間量である点灯時間を予測し得る（例えば、将来の約２２ミリ秒）が、一方で、レンダリングコンポーネント２０１２は、無線通信リンクの場合、無線接続に対して有線接続を介してデータを転送するときのレイテンシの固有の違いに起因して、将来の第２の、より大きな時間量である点灯時間を予測し得る（例えば、将来の約４４ミリ秒）。

【０１７５】

本明細書で説明したように、ホストコンピュータ１０６は、ＨＭＤ１０２から、ＨＭＤ１０２の頭部追跡ジュール１９３０によって生成された頭部追跡データ２０８を受信することもできる。この頭部追跡データ２０８は、ターゲットフレームレートおよび／またはＨＭＤ１０２のリフレッシュレートに対応する周波数などの任意の適切な周波数、または１０００Ｈｚ（または１ミリ秒ごとの１つのセンサの読み取り）などの異なる（例えば、より速い）周波数で生成および／または送信され得る。レンダリングコンポーネント２０１２は、頭部追跡データ２０８に少なくとも部分的に基づいて、ＨＭＤ１０２が予測点灯時間に存在する予測姿勢を決定するように構成されている。次いで、レンダリングコンポーネント２０１２は、予測姿勢に基づいてフレームをレンダリングする（例えば、フレームについてピクセルデータを生成する）ために、予測姿勢を示す姿勢データを実行アプリケーション２００８に提供し得、レンダリングコンポーネント２０１２は、アプリケーション２００８から、フレーム３００と関連付けられたピクセルデータ１１０（１）を取得し得る。

【０１７６】

ホストコンピュータ１０６は、ネットワークおよび／またはＨＭＤ１０２などの第２のデバイスへの無線接続を容易にするために、アンテナ２０２０に連結された無線ユニット２０１８を含むが、これらに限定されない、通信インターフェース２０１６をさらに含み得る。無線ユニット２０１８は、Ｗｉ－Ｆｉ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、無線周波数（ＲＦ）などの、様々な無線技術のうちの１つ以上を実装し得る。ホストコンピュータ１０６は、ネットワークおよび／またはＨＭＤ１０２などの第２のデバイスへの有線接続を容易にする物理ポートをさらに含み得ることを理解されたい。

【０１７７】
一般的に、ホストコンピュータ１０６は、入力デバイス２０２２および出力デバイス２０２４を有する。入力デバイス２０２２は、キーボード、キーパッド、マウス、タッチスクリーン、ジョイスティック、制御ボタン、マイク、カメラなどを含み得る。出力デバイス２０２４は、非限定的に、ディスプレイ、発光素子（例えば、ＬＥＤ）、触覚感覚を生み出すためのバイブレータ、スピーカ（例えば、ヘッドフォン）などを含み得る。いくつかの実施形態では、ＨＭＤ１０２上に実装されていると示されているコンポーネントのサブセットは、ホストコンピュータ１０６またはＨＭＤ１０２から分離された別のコンピューティングデバイス上に実装され得、かつ／またはホストコンピュータ１０６上に実装されていると示されているコンポーネントのサブセットは、ＨＭＤ１０２またはホストコンピュータ１０６から分離された別のコンピューティングデバイス上に実装され得ることを理解されたい。
【０１７８】
本主題は、構造的特徴に固有の言語で説明されているが、添付の特許請求の範囲で定義された主題が必ずしも説明された特定の特徴に限定されないことを理解されたい。むしろ、特定の特徴は、特許請求の範囲を実装する例示的な形態として開示される。
以下に、本願出願時の特許請求の範囲に記載された発明を付記する。
[１]
ヘッドマウントディスプレイ（ＨＭＤ）であって、
１つ以上のディスプレイパネルと、
プロセッサと、
コンピュータ実行可能命令を記憶するメモリと、を備え、前記命令が、前記プロセッサによって実行されると、前記ＨＭＤに、
前記ＨＭＤに通信可能に連結されたホストコンピュータから、圧縮されたピクセルデータを表す動きベクトルアレイを受信することと、
前記圧縮されたピクセルデータを解凍してピクセルデータを取得することと、
前記ピクセルデータを修正して、修正されたピクセルデータを取得することであって、前記ピクセルデータが、
前記ＨＭＤの予測姿勢、および
前記動きベクトルアレイ、に少なくとも部分的に基づいて修正される、取得することと、
前記修正されたピクセルデータに少なくとも部分的に基づいて、前記１つ以上のディスプレイパネル上に画像を提示することと、を行わせる、ＨＭＤ。
[２]
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ＨＭＤに、
複数の頂点を有するテッセレーションメッシュを含むレンダリングメッシュを生成することと、
前記動きベクトルアレイ間の非ゼロ動きベクトルに少なくとも部分的に基づいて、前記複数の頂点のうちの頂点を、移動された頂点として前記レンダリングメッシュ内の異なる位置に移動させることと、をさらに行わせ、
前記動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを修正することが、前記レンダリングメッシュの前記移動された頂点に従って前記ピクセルデータのピクセル値を移動させて、前記修正されたピクセルデータを取得することを含む、[１]に記載のＨＭＤ。
[３]
前記頂点が、
前記非ゼロ動きベクトルの方向に、かつ
前記非ゼロ動きベクトルの大きさに対応する量だけ、移動される、[２]に記載のＨＭＤ。
[４]

前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ＨＭＤに、さらに、符号化されたデータストリームから前記動きベクトルアレイを抽出させる、[１]に記載のＨＭＤ。

[５]

前記ホストコンピュータが、前記ＨＭＤに無線で連結され、

前記動きベクトルアレイが、前記ホストコンピュータから無線で受信される、[１]に記載のＨＭＤ。

[６]

前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ＨＭＤに、

さらに、前記動きベクトルアレイにフィルタを適用して、修正された動きベクトルアレイを取得させ、

前記動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを修正することが、前記修正された動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを修正することを含む、[１]に記載のＨＭＤ。

[７]

方法であって、

ヘッドマウントディスプレイ（ＨＭＤ）によって、ホストコンピュータから、圧縮されたピクセルデータを表す動きベクトルアレイを受信することと、

前記ＨＭＤによって、前記圧縮されたピクセルデータを解凍して、ピクセルデータを取得することと、

前記ＨＭＤによって、前記動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを修正して、修正されたピクセルデータを取得することと、

前記修正されたピクセルデータに少なくとも部分的に基づいて、前記ＨＭＤの１つ以上のディスプレイパネル上に画像を提示することと、を含む、方法。

[８]

前記ピクセルデータの前記修正の前に、

前記ＨＭＤのメモリ内の前記ピクセルデータをキャッシュされたピクセルデータとしてキャッシュすることと、

前記キャッシュされたピクセルデータが、前記ＨＭＤに利用可能な最後に解凍されたピクセルデータを表すことを決定することと、

前記ＨＭＤの前記メモリから前記キャッシュされたピクセルデータを取り出すことと、をさらに含む、[７]に記載の方法。

[９]

前記ＨＭＤによって、複数の頂点を有するテッセレーションメッシュを含むレンダリングメッシュを生成することと、

前記動きベクトルアレイ間の非ゼロ動きベクトルに少なくとも部分的に基づいて、前記複数の頂点のうちの頂点を、移動された頂点として前記レンダリングメッシュ内の異なる位置に移動させることと、をさらに含み、

前記動きベクトルアレイに少なくとも部分的に基づく前記ピクセルデータの前記修正が、前記レンダリングメッシュの前記移動された頂点に従って前記ピクセルデータのピクセル値を移動させて、前記修正されたピクセルデータを取得することを含む、[７]に記載の方法。

[１０]

前記動きベクトルアレイを符号化されたデータストリームから抽出することをさらに含む、[７]に記載の方法。

[１１]

前記画像の前記提示の前に、

前記ＨＭＤによって、前記ホストコンピュータから、前記ホストコンピュータ上で実行されるアプリケーションによって、前記ピクセルデータを生成するために使用された前記

ＨＭＤの予測姿勢を示す姿勢データを受信することをさらに含み、
前記ピクセルデータの前記修正が、前記ＨＭＤの前記予測姿勢と更新された姿勢の予測との間の比較にさらに少なくとも部分的に基づいている、[７]に記載の方法。
[１２]
前記画像の前記提示の前に、
前記ＨＭＤによって、前記ホストコンピュータから、前記ホストコンピュータ上で実行されるアプリケーションによって出力されるＺバッファデータを表す深度データを受信することをさらに含み、
前記ピクセルデータの前記修正が、前記深度データにさらに少なくとも部分的に基づいている、[７]に記載の方法。
[１３]
前記ＨＭＤによって、前記動きベクトルアレイにフィルタを適用して、修正された動きベクトルアレイを取得することをさらに含み、
前記動きベクトルアレイに少なくとも部分的に基づく前記ピクセルデータの前記修正が、前記修正された動きベクトルアレイに基づいて、前記ピクセルデータを修正することを含む、[７]に記載の方法。
[１４]
ディスプレイシステムであって、
１つ以上のディスプレイパネルと、
プロセッサと、
コンピュータ実行可能命令を記憶するメモリと、を備え、前記命令が、前記プロセッサによって実行されると、前記ディスプレイシステムに、
ホストコンピュータから、圧縮されたピクセルデータを表す動きベクトルアレイを受信することと、
前記圧縮された第１のピクセルデータを解凍してピクセルデータを取得することと、
前記動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを修正して、修正されたピクセルデータを取得することと、
前記修正されたピクセルデータに少なくとも部分的に基づいて、前記１つ以上のディスプレイパネル上に画像を提示することと、を行わせる、ディスプレイシステム。
[１５]
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ピクセルデータを修正する前に、前記ディスプレイシステムに、
前記メモリ内の前記ピクセルデータを、キャッシュされたピクセルデータとしてキャッシュすることと、
前記キャッシュされたピクセルデータが、前記ＨＭＤに利用可能な最後に解凍されたピクセルデータを表すことを決定することと、
前記メモリから前記キャッシュされたピクセルデータを取り出すことと、を行わせる、[１４]に記載のディスプレイシステム。
[１６]
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ディスプレイシステムに、
複数の頂点を有するテッセレーションメッシュを含むレンダリングメッシュを生成することと、
前記動きベクトルアレイ間の非ゼロ動きベクトルに少なくとも部分的に基づいて、前記複数の頂点のうちの頂点を、移動された頂点として前記レンダリングメッシュ内の異なる位置に移動させることと、を行わせ、
前記動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを修正することが、前記レンダリングメッシュの前記移動された頂点に従って前記ピクセルデータのピクセル値を移動させて、前記修正されたピクセルデータを取得することを含む、[１４]に記載のディスプレイシステム。

[１７]
前記ホストコンピュータが、前記ディスプレイシステムに無線で連結され、
前記動きベクトルアレイが、前記ホストコンピュータから無線で受信される、[１４]に記載のディスプレイシステム。
[１８]
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記画像を提示する前に、前記ディスプレイシステムに、
前記ホストコンピュータから、前記ホストコンピュータ上で実行されるアプリケーションによって出力されるＺバッファデータを表す深度データを受信させ、
前記ピクセルデータを修正することが、前記深度データにさらに少なくとも部分的に基づいている、[１４]に記載のディスプレイシステム。
[１９]
前記コンピュータ実行可能命令が、前記プロセッサによって実行されると、前記ディスプレイシステムに、
前記動きベクトルアレイにフィルタを適用して、修正された動きベクトルアレイを取得させ、
前記動きベクトルアレイに少なくとも部分的に基づいて前記ピクセルデータを前記修正することが、前記修正された動きベクトルアレイに基づいて前記ピクセルデータを修正することを含む、[１４]に記載のディスプレイシステム。
[２０]
前記ディスプレイシステムが、仮想現実（ＶＲ）ヘッドセットまたは拡張現実（ＡＲ）ヘッドセットのうちの少なくとも１つを含む、[１４]に記載のディスプレイシステム。
【図面】
【図１】

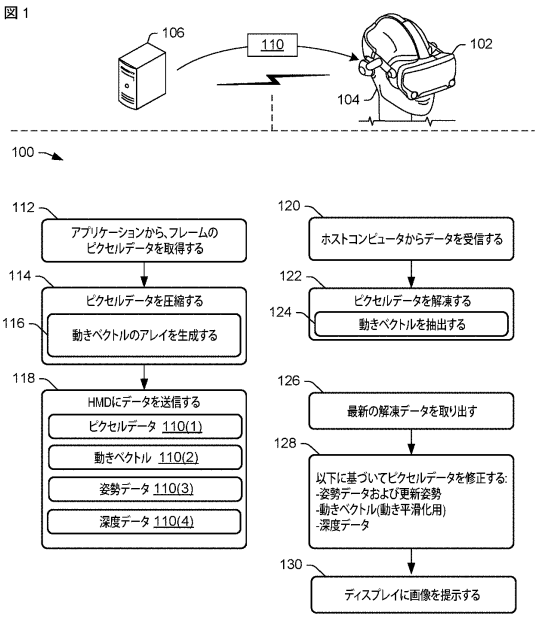

FIG. 1

【図２】

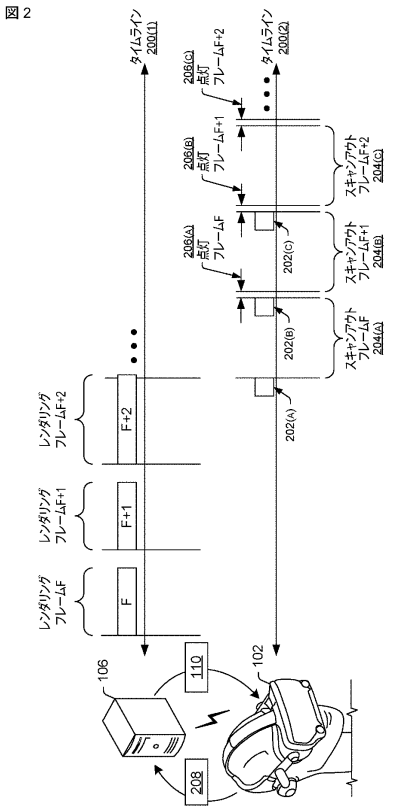

FIG. 2

【図３】

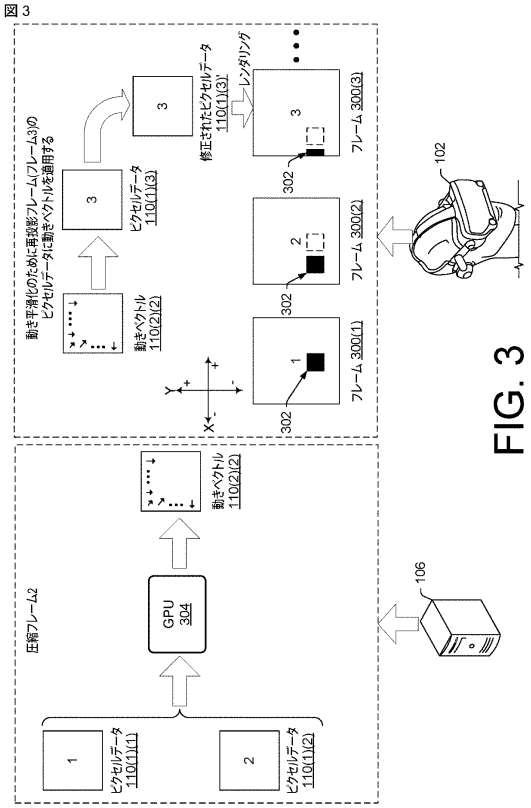

FIG. 3

【図４】

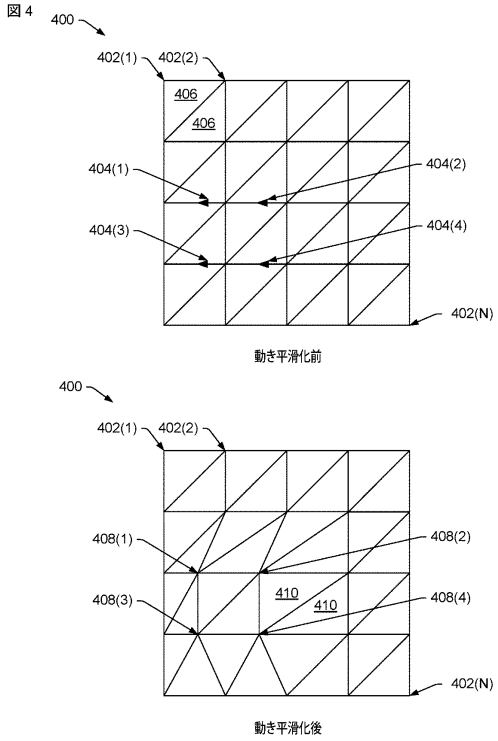

FIG. 4

【図５】

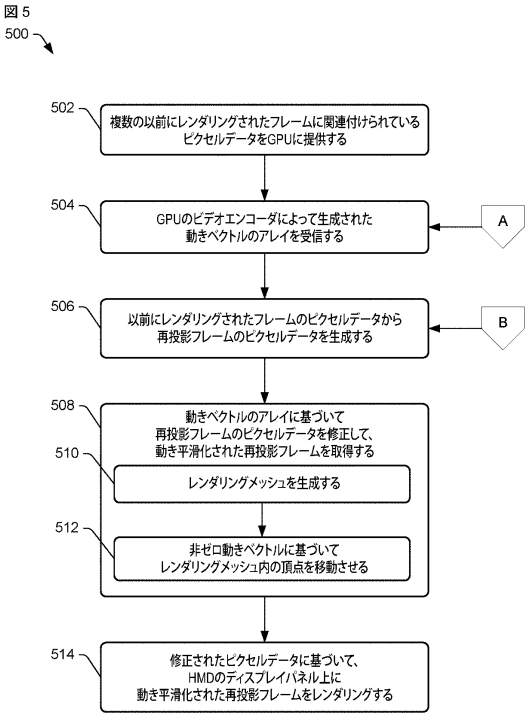

FIG. 5

【図６】

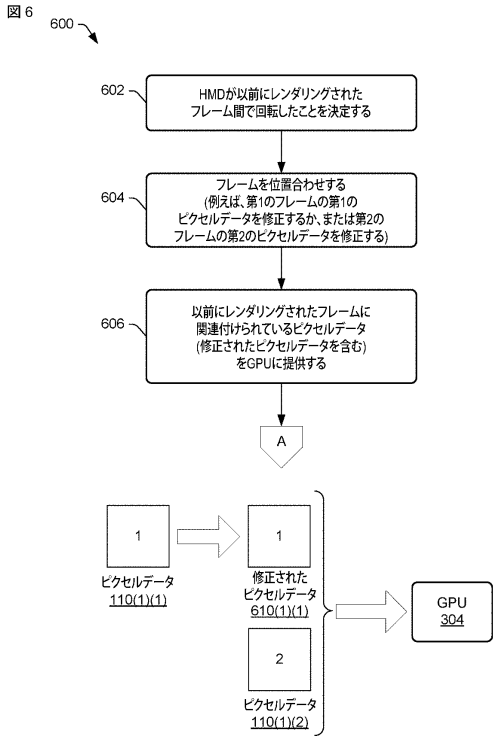

FIG. 6

【図７】

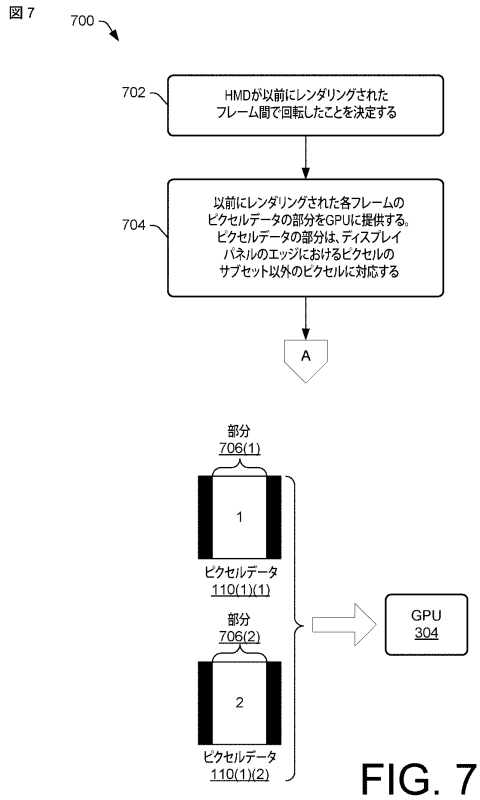

FIG. 7

【図８】

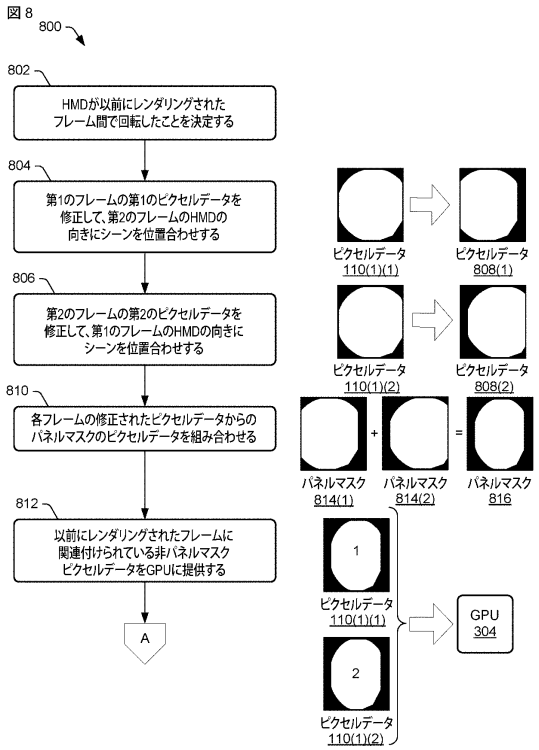

FIG. 8

【図９】

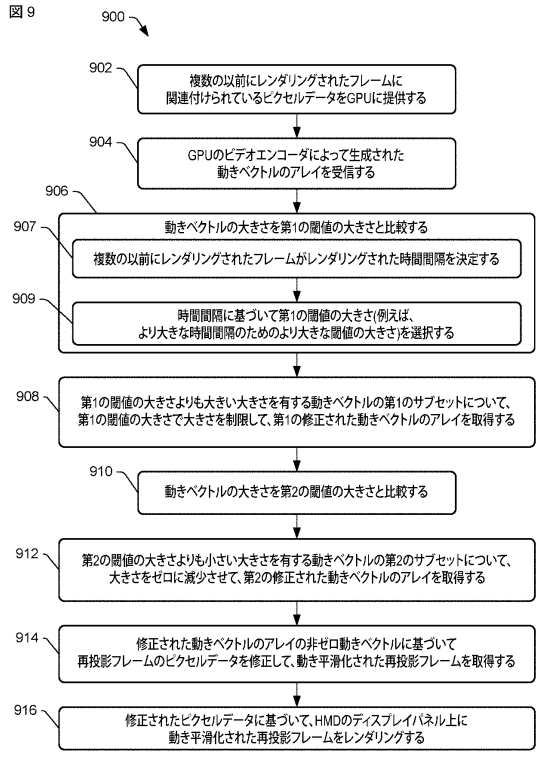

FIG. 9

【図１０】

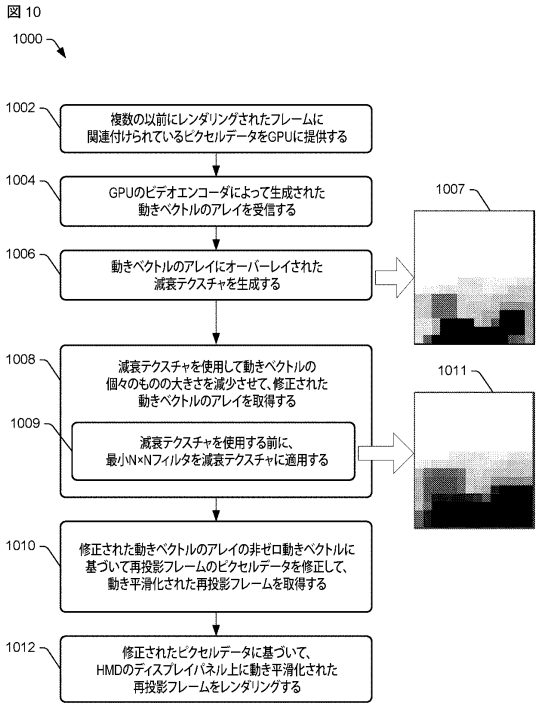

FIG. 10

【図１１】

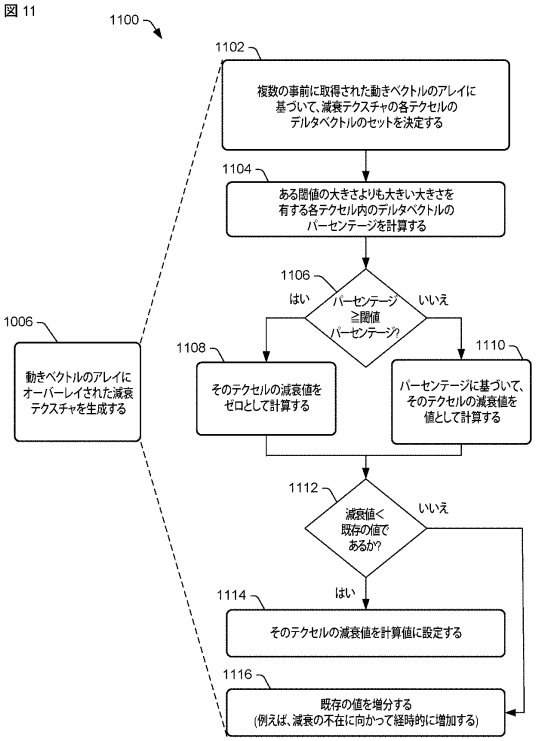

FIG. 11

【図１２】

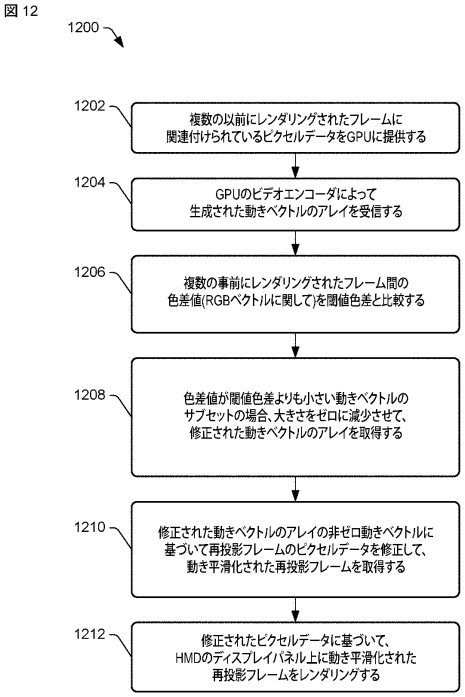

FIG. 12

【図１３】

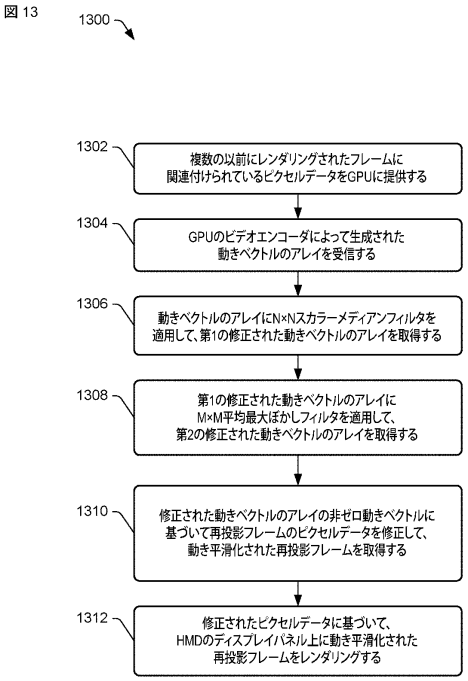

FIG. 13

【図１４】

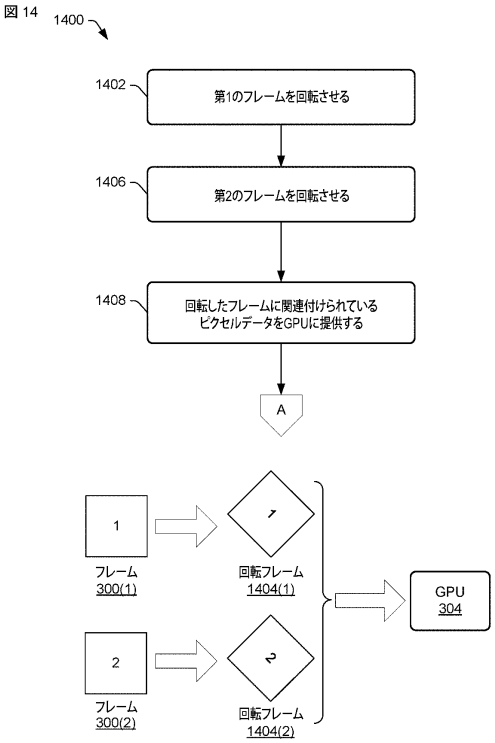

FIG. 14

【図１５】

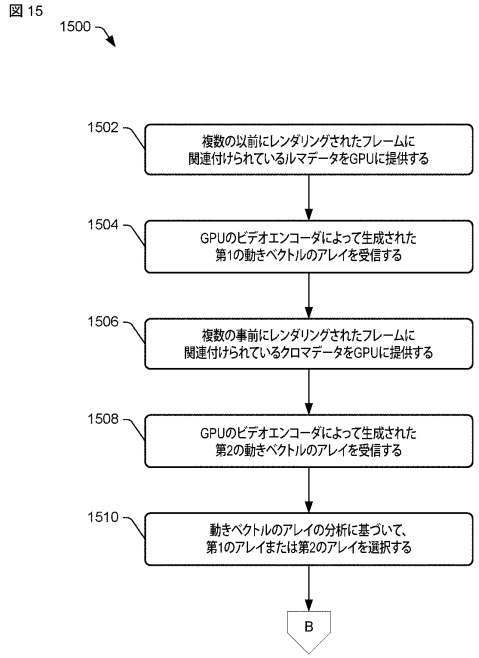

FIG. 15

【図１６】

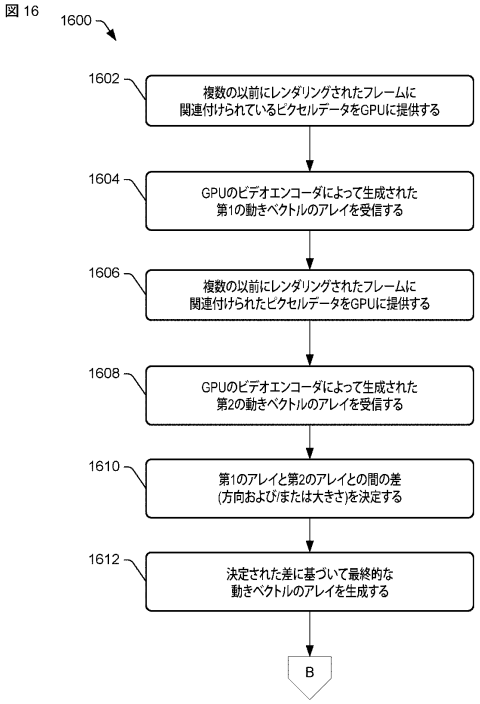

FIG. 16

【図１７】

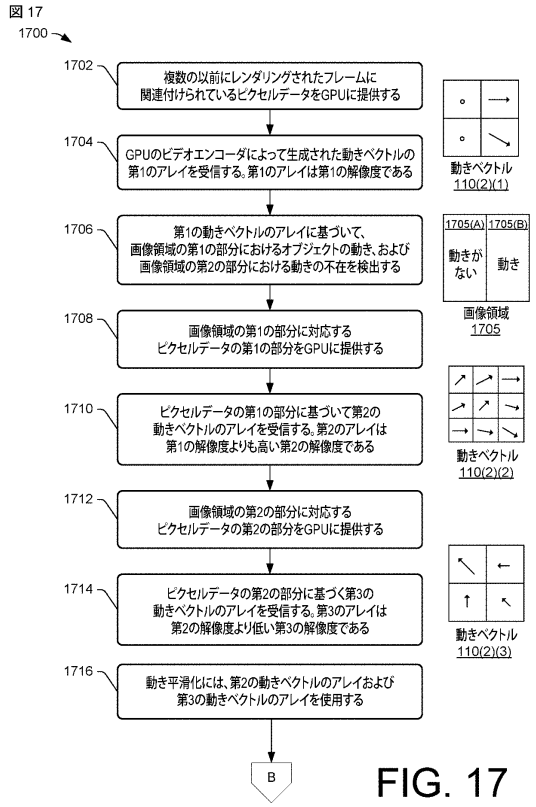

FIG. 17

【図１８Ａ】

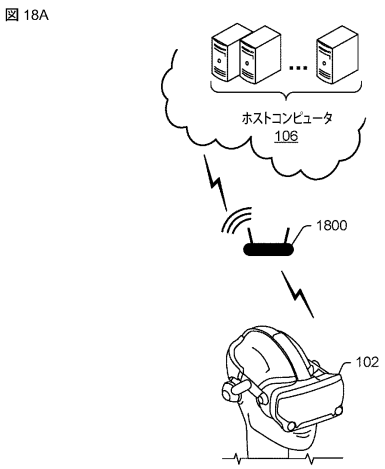

FIG. 18A

【図１８Ｂ】

図 18B

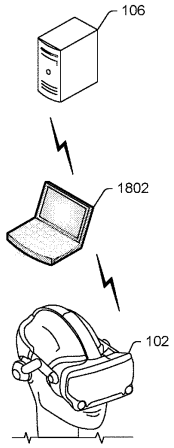

FIG. 18B

【図１９】

図 19

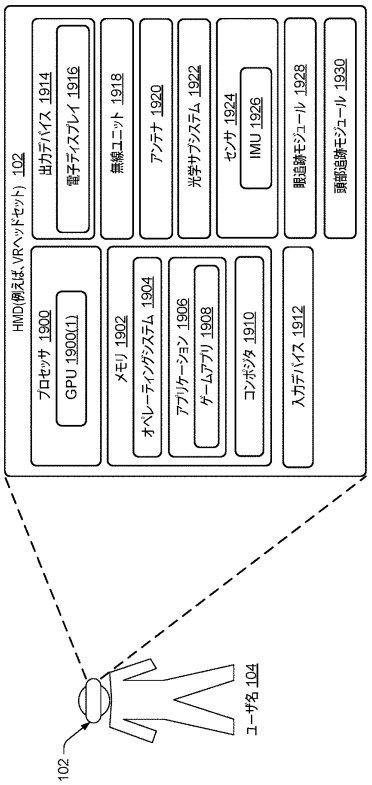

FIG. 19

【図２０】

図 20

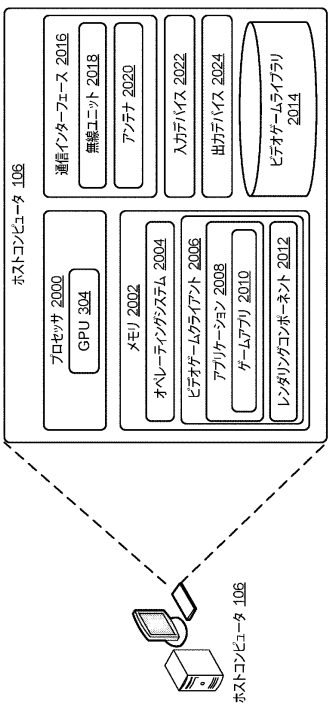

FIG. 20

フロントページの続き

(56)参考文献　特表２０１９－５０６０１５（ＪＰ，Ａ）
米国特許出願公開第２０１８／０２２０１１９（ＵＳ，Ａ１）
米国特許第０９９７８１８０（ＵＳ，Ｂ２）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｔ　１９／００