

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5455191号
(P5455191)

(45) 発行日 平成26年3月26日 (2014. 3. 26)

(24) 登録日 平成26年1月17日 (2014. 1. 17)

(51) Int. Cl.

F 1

A 6 3 F 13/428 (2014. 01)

A 6 3 F 13/00 2 0 0

A 6 3 F 13/211 (2014. 01)

A 6 3 F 13/00 1 0 6

請求項の数 24 (全 46 頁)

(21) 出願番号 特願2009-93985 (P2009-93985)
 (22) 出願日 平成21年4月8日 (2009. 4. 8)
 (65) 公開番号 特開2010-240242 (P2010-240242A)
 (43) 公開日 平成22年10月28日 (2010. 10. 28)
 審査請求日 平成24年2月20日 (2012. 2. 20)

(73) 特許権者 000233778
 任天堂株式会社
 京都府京都市南区上鳥羽鉾立町 1 1 番地 1
 (74) 代理人 100158780
 弁理士 寺本 亮
 (74) 代理人 100121359
 弁理士 小沢 昌弘
 (74) 代理人 100130269
 弁理士 石原 盛規
 (72) 発明者 鈴木 一郎
 京都府京都市南区上鳥羽鉾立町 1 1 番地 1
 任天堂株式会社内

審査官 柴田 和雄

最終頁に続く

(54) 【発明の名称】 姿勢算出装置、姿勢算出プログラム、姿勢算出システム、および姿勢算出方法

(57) 【特許請求の範囲】

【請求項 1】

角速度センサと加速度センサとを少なくとも備えた入力装置から角速度データと加速度データとを含む操作データを取得し、当該入力装置の姿勢を算出する姿勢算出装置であって、

前記入力装置から見た重力方向を表す重力ベクトルを、前記操作データを用いて算出する重力方向算出手段と、

前記入力装置の運動による加速度を表す運動加速度ベクトルを前記加速度データにより示される加速度と前記重力ベクトルとに基づいて算出する運動加速度算出手段と、

前記入力装置が所定の運動を行う場合における、前記入力装置の運動による運動加速度と角速度との予め定義された関係に基づいて、前記角速度データにより示される角速度に対して当該関係を満たす運動加速度へと前記運動加速度ベクトルを近づける方向へ前記重力ベクトルを補正する第 1 補正手段と、

前記第 1 補正手段が補正した重力ベクトルに対応する前記入力装置の姿勢を算出する第 1 姿勢算出手段とを備え、

前記所定の運動は、所定位置を中心とした回転運動であり、

前記第 1 補正手段は、

前記回転運動を行う場合において前記入力装置の運動加速度の方向と直交する直交ベクトルを前記角速度データに基づいて算出する直交ベクトル算出手段と、

前記入力装置の運動加速度と角速度との関係として、前記運動加速度ベクトルが前記

10

20

直交ベクトルに直交する関係を用いて、前記重力ベクトルを補正するベクトル補正手段とを含む、姿勢算出装置。

【請求項 2】

前記角速度データから前記入力装置の姿勢を算出する第 2 姿勢算出手段をさらに備え、前記重力方向算出手段は、前記第 2 姿勢算出手段が算出した姿勢に基づいて前記重力ベクトルを算出し、

前記第 1 姿勢算出手段は、前記第 1 補正手段が補正した重力ベクトルを用いて前記第 2 姿勢算出手段が算出した姿勢を補正することによって前記入力装置の姿勢を算出する、請求項 1 に記載の姿勢算出装置。

【請求項 3】

前記直交ベクトル算出手段は、前記入力装置から取得される前記角速度データの推移に基づいて角加速度を算出し、前記角速度データにより示される角速度と当該角加速度とに基づいて前記直交ベクトルを算出する、請求項 1 に記載の姿勢算出装置。

【請求項 4】

前記所定の運動は、所定位置を中心とした等角速度の回転運動であり、

前記直交ベクトル算出手段は、前記回転運動の中心軸を表す角速度ベクトルを前記直交ベクトルとして算出する、請求項 1 に記載の姿勢算出装置。

【請求項 5】

前記ベクトル補正手段は、前記直交ベクトルと前記運動加速度ベクトルとの内積が小さくなるように前記重力ベクトルを補正する、請求項 1 から請求項 4 のいずれか 1 項に記載の姿勢算出装置。

【請求項 6】

前記第 1 補正手段は、前記角速度データにより表される角速度が小さいほど前記重力ベクトルを補正する量が小さくなるように前記重力ベクトルを補正する、請求項 1 から請求項 5 のいずれか 1 項に記載の姿勢算出装置。

【請求項 7】

前記第 1 補正手段は、前記加速度データにより示される加速度の単位時間あたりの変化量が大きいほど前記重力ベクトルを補正する量が小さくなるように前記重力ベクトルを補正する、請求項 1 から請求項 6 のいずれか 1 項に記載の姿勢算出装置。

【請求項 8】

前記第 1 補正手段は、前記重力ベクトルを所定角度だけ回転させたときに、回転後の重力ベクトルから算出される運動加速度ベクトルが、前記角速度データにより示される角速度に対して当該関係を満たす運動加速度へ最も近づく方向へ、前記重力ベクトルを回転させる補正を行う、請求項 1 から請求項 7 のいずれか 1 項に記載の姿勢算出装置。

【請求項 9】

前記加速度データが示す加速度の向きに近づける方向へ前記重力ベクトルを補正する第 2 補正手段をさらに備え、

前記第 1 姿勢算出手段は、前記第 1 補正手段および第 2 補正手段による補正後の重力ベクトルを用いて前記入力装置の姿勢を算出する、請求項 1 から請求項 8 のいずれか 1 項に記載の姿勢算出装置。

【請求項 10】

前記第 2 補正手段は、前記角速度データにより表される角速度が大きいほど前記重力ベクトルを補正する量が小さくなるように前記重力ベクトルを補正する、請求項 9 に記載の姿勢算出装置。

【請求項 11】

前記第 2 補正手段は、前記加速度データにより示される加速度の単位時間あたりの変化量が大きいほど前記重力ベクトルを補正する量が小さくなるように前記重力ベクトルを補正する、請求項 9 または請求項 10 に記載の姿勢算出装置。

【請求項 12】

角速度センサと加速度センサとを少なくとも備えた入力装置から角速度データと加速度

10

20

30

40

50

データとを含む操作データを取得し、当該入力装置の姿勢を算出する姿勢算出装置のコンピュータにおいて実行される姿勢算出プログラムであって、

前記入力装置から見た重力方向を表す重力ベクトルを前記操作データを用いて算出する重力方向算出手段と、

前記入力装置の運動による加速度を表す運動加速度ベクトルを前記加速度データにより示される加速度と前記重力ベクトルとに基づいて算出する運動加速度算出手段と、

前記入力装置が所定の運動を行う場合において当該運動により前記入力装置に加えられる運動加速度と角速度との関係を定義しておき、前記角速度データにより示される角速度に対して当該関係を満たす運動加速度へと前記運動加速度ベクトルを近づける方向へ前記重力ベクトルを補正する第1補正手段と、

前記第1補正手段が補正した重力ベクトルを用いて前記入力装置の姿勢を算出する第1姿勢算出手段として前記姿勢算出プログラムは前記コンピュータを機能させ、

前記所定の運動は、所定位置を中心とした回転運動であり、

前記第1補正手段は、

前記回転運動を行う場合において前記入力装置の運動加速度の方向と直交する直交ベクトルを前記角速度データに基づいて算出する直交ベクトル算出手段と、

前記入力装置の運動加速度と角速度との関係として、前記運動加速度ベクトルが前記直交ベクトルに直交する関係を用いて、前記重力ベクトルを補正するベクトル補正手段とを含む、姿勢算出プログラム。

【請求項13】

前記角速度データから前記入力装置の姿勢を算出する第2姿勢算出手段として前記コンピュータをさらに機能させ、

前記重力方向算出手段は、前記第2姿勢算出手段が算出した姿勢に基づいて前記重力ベクトルを算出し、

前記第1姿勢算出手段は、前記第1補正手段が補正した重力ベクトルを用いて前記第2姿勢算出手段が算出した姿勢を補正することによって前記入力装置の姿勢を算出する、請求項12に記載の姿勢算出プログラム。

【請求項14】

前記直交ベクトル算出手段は、前記入力装置から取得される前記角速度データの推移に基づいて角加速度を算出し、前記角速度データにより示される角速度と当該角加速度とに基づいて前記直交ベクトルを算出する、請求項12に記載の姿勢算出プログラム。

【請求項15】

前記所定の運動は、所定位置を中心とした等角速度の回転運動であり、

前記直交ベクトル算出手段は、前記回転運動の中心軸を表す角速度ベクトルを前記直交ベクトルとして算出する、請求項12に記載の姿勢算出プログラム。

【請求項16】

前記ベクトル補正手段は、前記直交ベクトルと前記運動加速度ベクトルとの内積が小さくなるように前記重力ベクトルを補正する、請求項12から請求項15のいずれか1項に記載の姿勢算出プログラム。

【請求項17】

前記第1補正手段は、前記角速度データにより表される角速度が小さいほど前記重力ベクトルを補正する量が小さくなるように前記重力ベクトルを補正する、請求項12から請求項16のいずれか1項に記載の姿勢算出プログラム。

【請求項18】

前記第1補正手段は、前記加速度データにより示される加速度の単位時間あたりの変化量が大きいほど前記重力ベクトルを補正する量が小さくなるように前記重力ベクトルを補正する、請求項12から請求項17のいずれか1項に記載の姿勢算出プログラム。

【請求項19】

前記第1補正手段は、前記重力ベクトルを所定角度だけ回転させたときに、回転後の重力ベクトルから算出される運動加速度ベクトルが、前記角速度データにより示される角速

10

20

30

40

50

度に対して当該関係を満たす運動加速度へ最も近づく方向へ、前記重力ベクトルを回転させる補正を行う、請求項 1 2 から請求項 1 8 のいずれか 1 項に記載の姿勢算出プログラム。

【請求項 2 0】

前記加速度データが示す加速度の向きに近づける方向へ前記重力ベクトルを補正する第 2 補正手段として前記コンピュータをさらに機能させ、

前記第 1 姿勢算出手段は、前記第 1 補正手段および第 2 補正手段による補正後の重力ベクトルを用いて前記入力装置の姿勢を算出する、請求項 1 2 から請求項 1 9 のいずれか 1 項に記載の姿勢算出プログラム。

【請求項 2 1】

前記第 2 補正手段は、前記角速度データにより表される角速度が大きいほど前記重力ベクトルを補正する量が小さくなるように前記重力ベクトルを補正する、請求項 2 0 に記載の姿勢算出プログラム。

【請求項 2 2】

前記第 2 補正手段は、前記加速度データにより示される加速度の単位時間あたりの変化量が大きいほど前記重力ベクトルを補正する量が小さくなるように前記重力ベクトルを補正する、請求項 2 0 または請求項 2 1 に記載の姿勢算出プログラム。

【請求項 2 3】

角速度センサと加速度センサとを少なくとも備えた入力装置から角速度データと加速度データとを含む操作データを取得し、当該入力装置の姿勢を算出する姿勢算出システムであって、

前記入力装置から見た重力方向を表す重力ベクトルを、前記操作データを用いて算出する重力方向算出手段と、

前記入力装置の運動による加速度を表す運動加速度ベクトルを前記加速度データにより示される加速度と前記重力ベクトルとに基づいて算出する運動加速度算出手段と、

前記入力装置が所定の運動を行う場合における、前記入力装置の運動による運動加速度と角速度との予め定義された関係に基づいて、前記角速度データにより示される角速度に対して当該関係を満たす運動加速度へと前記運動加速度ベクトルを近づける方向へ前記重力ベクトルを補正する第 1 補正手段と、

前記第 1 補正手段が補正した重力ベクトルに対応する前記入力装置の姿勢を算出する第 1 姿勢算出手段とを備え、

前記所定の運動は、所定位置を中心とした回転運動であり、

前記第 1 補正手段は、

前記回転運動を行う場合において前記入力装置の運動加速度の方向と直交する直交ベクトルを前記角速度データに基づいて算出する直交ベクトル算出手段と、

前記入力装置の運動加速度と角速度との関係として、前記運動加速度ベクトルが前記直交ベクトルに直交する関係を用いて、前記重力ベクトルを補正するベクトル補正手段とを含む、姿勢算出システム。

【請求項 2 4】

角速度センサと加速度センサとを少なくとも備えた入力装置から角速度データと加速度データとを含む操作データを取得し、当該入力装置の姿勢を算出する姿勢算出装置において実行される姿勢算出方法であって、

前記入力装置から見た重力方向を表す重力ベクトルを、前記操作データを用いて算出する重力方向算出ステップと、

前記入力装置の運動による加速度を表す運動加速度ベクトルを前記加速度データにより示される加速度と前記重力ベクトルとに基づいて算出する運動加速度算出ステップと、

前記入力装置が所定の運動を行う場合における、前記入力装置の運動による運動加速度と角速度との予め定義された関係に基づいて、前記角速度データにより示される角速度に対して当該関係を満たす運動加速度へと前記運動加速度ベクトルを近づける方向へ前記重力ベクトルを補正する第 1 補正ステップと、

10

20

30

40

50

前記第1補正ステップにおいて補正された重力ベクトルに対応する前記入力装置の姿勢を算出する第1姿勢算出ステップとを備え、

前記所定の運動は、所定位置を中心とした回転運動であり、

前記第1補正ステップは、

前記回転運動を行う場合において前記入力装置の運動加速度の方向と直交する直交ベクトルを前記角速度データに基づいて算出する直交ベクトル算出ステップと、

前記入力装置の運動加速度と角速度との関係として、前記運動加速度ベクトルが前記直交ベクトルに直交する関係を用いて、前記重力ベクトルを補正するベクトル補正ステップを含む、姿勢算出方法。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、姿勢算出装置または姿勢算出プログラムに関し、より特定的には、入力装置の姿勢を算出する姿勢算出装置または姿勢算出プログラムに関する。

【背景技術】

【0002】

従来、加速度センサおよびジャイロセンサを用いて入力装置の姿勢を算出する技術が考えられている。例えば、特許文献1には、加速度センサおよびジャイロセンサを備えた入力制御装置を使用するゲーム装置が記載されている。このゲーム装置は、ゲームキャラクタが持つ刀を入力制御装置の動きに応じて制御するものである。具体的には、加速度センサの出力に基づいて刀を振る動作のデータが作成され、ジャイロセンサの出力に基づいて刀の姿勢のデータが作成される。

20

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2000-308756号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

30

上記特許文献1のようにジャイロセンサを用いて姿勢を算出する場合、算出される姿勢と入力制御装置の実際の姿勢との間に誤差が生じる。入力制御装置の動きが遅い場合に入力制御装置の角速度がジャイロセンサによって検知されなかったり、逆に、入力制御装置の動きが激しい場合に入力制御装置の角速度がジャイロセンサの検知可能な範囲を超えてしまったりすること等が生じる場合があるからである。また、角速度データの出力の間隔よりも短い期間中に角速度が急激に変化するような場合にも、誤差が発生する可能性がある。また、ジャイロセンサは温度等の要因によりセンサの出力にドリフトと呼ばれる誤差が発生することもある。そして、角速度から算出される姿勢には、時間とともに角速度の誤差が蓄積されていくことになるので姿勢の誤差が大きくなっていくことになる。特許文献1では、ジャイロセンサから算出される姿勢の誤差を想定していないために、姿勢を正

40

【0005】

それ故、本発明の目的は、ジャイロセンサ等の角速度センサを用いて入力装置の姿勢を正確に算出することができる姿勢算出装置または姿勢算出プログラムを提供することである。

【課題を解決するための手段】

【0006】

本発明は、上記の課題を解決するために、以下の構成を採用した。なお、本欄における括弧内の参照符号および補足説明等は、本発明の理解を助けるために後述する実施形態との対応関係を示したものであって、本発明を何ら限定するものではない。

50

【0007】

本発明は、角速度センサ（ジャイロセンサ55、56）と加速度センサ（37）とを少なくとも備えた入力装置（8）から角速度データ（63）と加速度データ（64）とを含む操作データ（62）を取得し、当該入力装置の姿勢を算出する姿勢算出装置（ゲーム装置3）である。姿勢算出装置は、重力方向算出手段（ステップS11を実行するCPU10。以下、単にステップ番号のみを記載する。）と、運動加速度算出手段（S24）と、第1補正手段（S34）と、第1姿勢算出手段（S15）とを備えている。重力方向算出手段は、入力装置から見た重力方向を表す重力ベクトル（ V_g ）を操作データを用いて算出する。運動加速度算出手段は、入力装置の運動による加速度を表す運動加速度ベクトル（ V_A ）を加速度データにより示される加速度（検出加速度ベクトル V_a ）と重力ベクトル（ V_g ）とに基づいて算出する（図10）。第1補正手段は、入力装置が所定の運動（回転運動）を行う場合における、前記入力装置の運動による運動加速度と角速度との予め定義された関係（式（7）または（15））に基づいて、角速度データにより示される角速度に対して当該関係を満たす運動加速度（図11に示すベクトル V_A' ）へと運動加速度ベクトルを近づける方向へ重力ベクトルを補正する（図11、図12）。第1姿勢算出手段は、第1補正手段が補正した重力ベクトルに対応する入力装置の姿勢を算出する。

10

【0008】

上記によれば、姿勢算出装置は、入力装置が所定の運動を行う場合における角速度と運動加速度との関係を用いて、運動加速度ベクトルが当該関係を満たすように、運動加速度ベクトルの算出元である重力ベクトルを補正する。これによって、入力装置が動かされている場合であっても、加速度データと角速度データとを用いて、重力ベクトルを正しく補正することができる。そして、正確な重力ベクトルを用いて入力装置の姿勢を算出することによって、姿勢を正確に算出することができる。

20

【0009】

また、姿勢算出装置は、角速度データから入力装置の姿勢（第1の姿勢）を算出する第2姿勢算出手段（S4）をさらに備えていてもよい。このとき、重力方向算出手段は、第2姿勢算出手段が算出した姿勢に基づいて重力ベクトルを算出する。第1姿勢算出手段は、第1補正手段が補正した重力ベクトルを用いて第2姿勢算出手段が算出した姿勢を補正することによって入力装置の姿勢を算出する。

【0010】

上記によれば、ジャイロセンサの検出結果である角速度データから算出された入力装置の姿勢を、正確な重力ベクトルを用いて補正することができる。これによって、ジャイロセンサを用いて入力装置の姿勢を正確に算出することができる。

30

【0011】

上記所定の運動は、所定位置を中心とした回転運動であってもよい。このとき、第1補正手段は、直交ベクトル算出手段（S3、S42）と、ベクトル補正手段（S34）とを含んでいる。直交ベクトル算出手段は、回転運動を行う場合において入力装置の運動加速度の方向と直交する直交ベクトル（ V_ω または W ）を角速度データに基づいて算出する。ベクトル補正手段は、入力装置の運動加速度と角速度との関係として、運動加速度ベクトルが直交ベクトルに直交する関係（式（7）または（15））を用いて、重力ベクトルを補正する。

40

【0012】

上記によれば、入力装置が回転運動を行う場合において、運動加速度ベクトルが直交ベクトルに直交する関係を用いることにより、入力装置が回転運動を行う場合における入力装置の姿勢を容易に算出することができる。

【0013】

上記直交ベクトル算出手段は、入力装置から取得される角速度データの推移に基づいて角加速度を算出し、角速度データにより示される角速度と当該角加速度とに基づいて直交ベクトルを算出してもよい。

【0014】

50

上記によれば、直交ベクトルを角速度データから容易に算出することができる。

【0015】

上記所定の運動は、所定位置を中心とした等角速度の回転運動であってもよい。このとき、直交ベクトル算出手段は、回転運動の中心軸を表す角速度ベクトルを直交ベクトルとして算出する。

【0016】

上記によれば、所定の運動を等角速度の回転運動とすることによって、角速度ベクトルを直交ベクトルとして用いることができる。したがって、直交ベクトルを容易に算出することができ、重力ベクトルの補正処理を容易に行うことができる。

【0017】

上記ベクトル補正手段は、直交ベクトルと運動加速度ベクトルとの内積が小さくなるように重力ベクトルを補正してもよい。

【0018】

上記によれば、直交ベクトルと運動加速度ベクトルとの内積の大きさを評価して重力ベクトルを補正することによって、補正を容易に行うことができる。

【0019】

上記第1補正手段は、角速度データにより表される角速度が小さいほど重力ベクトルを補正する量が小さくなるように重力ベクトルを補正してもよい（S32）。

【0020】

上記によれば、入力装置の角速度が小さい場合には重力ベクトルの補正量は小さくなり、角速度が大きい場合には重力ベクトルの補正量は大きくなる。ここで、角速度が小さい場合とは、入力装置が回転運動を行っていない可能性が高い場合であると推測され、第1補正手段による補正が正しくない可能性がある。つまり、上記によれば、第1補正手段による補正が正しくない可能性が高い場合には、補正量が小さくなるので、第1補正手段によって不正確な補正が行われることを防止することができる。その結果、重力ベクトルの補正を正しく行うことができ、入力装置の姿勢を正確に算出することができる。

【0021】

上記第1補正手段は、加速度データにより示される加速度の単位時間あたりの変化量（加加速度J）が大きいほど重力ベクトルを補正する量が小さくなるように重力ベクトルを補正してもよい（S33）。

【0022】

上記によれば、入力装置の加加速度が大きい場合には重力ベクトルの補正量は小さくなり、加加速度が小さい場合には重力ベクトルの補正量は大きくなる。ここで、加加速度が大きい場合とは、入力装置が回転運動を行っていない可能性が高い場合であると推測され、第1補正手段による補正が正しくない可能性がある。つまり、上記によれば、第1補正手段による補正が正しくない可能性が高い場合には、補正量が小さくなるので、第1補正手段によって不正確な補正が行われることを防止することができる。その結果、重力ベクトルの補正を正しく行うことができ、入力装置の姿勢を正確に算出することができる。

【0023】

上記第1補正手段は、重力ベクトルを所定角度だけ回転させたときに、回転後の重力ベクトルから算出される運動加速度ベクトルが、角速度データにより示される角速度に対して当該関係を満たす運動加速度へ最も近づく方向へ、重力ベクトルを回転させる補正を行ってもよい。

【0024】

上記によれば、重力ベクトルを補正すべき方向を決定するだけでよいので、重力ベクトルの補正を容易に行うことができる。また、仮に、補正後の重力ベクトルが、運動加速度ベクトルが上記関係を満たす場合の重力ベクトルに一致するように補正するとすれば、1回の補正における補正量が大きくなり、重力ベクトルが急激に変化してしまうおそれがある。重力ベクトルが急激に変化すると、算出される入力装置の姿勢が急激に変化することになり、入力装置の姿勢を入力操作に用いる場合において操作性が悪くなるおそれがある

10

20

30

40

50

。これに対して、上記によれば、1回の補正における補正量を一定量以下に制限することができるので、重力ベクトルの急激な変化を防止し、入力装置の操作性が悪くなることを防止することができる。

【0025】

また、姿勢算出装置は、加速度データが示す加速度（検出加速度ベクトル V_a ）の向きに近づける方向へ重力ベクトルを補正する第2補正手段（S12）をさらに備えていてもよい。このとき、第1姿勢算出手段は、第1補正手段および第2補正手段による補正後の重力ベクトルを用いて入力装置の姿勢を算出する。

【0026】

上記によれば、重力ベクトルは、加速度データが示す加速度の向きに近づける方向へ補正される。ここで、入力装置が静止している場合には、当該加速度の向きは重力方向を指すので、重力ベクトルを正確に重力方向と一致させることができる。したがって、上記によれば、入力装置が動いている場合に加えて、静止している場合にも重力ベクトルを正しく補正することができ、入力装置の姿勢を正確に算出することができる。

10

【0027】

上記第2補正手段は、角速度データにより表される角速度が大きいほど重力ベクトルを補正する量が小さくなるように重力ベクトルを補正してもよい（S23）。

【0028】

上記によれば、入力装置の角速度が大きいほど、第2補正手段による重力ベクトルの補正量が小さくなる。ここで、加速度センサが検出する加速度は、角速度が大きいほど、入力装置が動いている可能性が高く、重力方向を正確に表していないと推測される。したがって、上記によれば、第2補正手段による補正が正しくない可能性が高い場合には、補正量が小さくなるので、第2補正手段によって不正確な補正が行われることを防止することができる。その結果、重力ベクトルの補正を正しく行うことができ、入力装置の姿勢を正確に算出することができる。

20

【0029】

また、上記第1補正手段および第2補正手段の両方において、角速度の大きさに応じて重力ベクトルの補正量を変化する場合には、次のような利点がある。すなわち、この場合、角速度が大きい期間では第1補正手段によって主に補正が行われ、角速度が小さい期間では第2補正手段によって主に補正が行われる。したがって、角速度が大きい場合でも小さい場合でも重力ベクトルの補正が行われることとなり、補正が行われる機会を多くすることができる。その結果、重力ベクトルを正確に算出することができ、入力装置の姿勢を正確に算出することができる。

30

【0030】

上記第2補正手段は、加速度データにより示される加速度の単位時間あたりの変化量（加加速度 J ）が大きいほど重力ベクトルを補正する量が小さくなるように重力ベクトルを補正してもよい（S26）。

【0031】

上記によれば、入力装置の加加速度が大きいほど、第2補正手段による重力ベクトルの補正量が小さくなる。ここで、加速度センサが検出する加速度は、加加速度が大きいほど、入力装置が動いている可能性が高く、重力方向を正確に表していないと推測される。したがって、上記によれば、第2補正手段による補正が正しくない可能性が高い場合には、補正量が小さくなるので、第2補正手段によって不正確な補正が行われることを防止することができる。その結果、重力ベクトルの補正を正しく行うことができ、入力装置の姿勢を正確に算出することができる。

40

【0032】

また、本発明は、情報処理装置のコンピュータを上記各手段として機能させる姿勢算出プログラムの形態で実施されてもよい。

【発明の効果】

【0033】

50

本発明によれば、入力装置が所定の運動を行う場合における角速度と運動加速度との関係を用いて、運動加速度ベクトルが当該関係を満たすように、運動加速度ベクトルの算出元である重力ベクトルを補正する。これによって、入力装置が動かされている場合であっても、加速度データと角速度データとを用いて、重力ベクトルを正しく補正することができる。

【図面の簡単な説明】

【0034】

【図1】 ゲームシステムの外観図

【図2】 ゲーム装置の機能ブロック図

【図3】 入力装置の外観構成を示す斜視図

10

【図4】 コントローラの外観構成を示す斜視図

【図5】 コントローラの内部構造を示す図

【図6】 コントローラの内部構造を示す図

【図7】 入力装置の構成を示すブロック図

【図8】 入力装置に関する重力ベクトルおよび検出加速度ベクトルを示す図

【図9】 静的補正処理における重力ベクトルの補正を示す図

【図10】 入力装置に関する重力ベクトル、検出加速度ベクトル、および運動加速度ベクトルを示す図

【図11】 入力装置に関する角速度ベクトルおよび運動加速度ベクトルを示す図

【図12】 動的補正処理における重力ベクトルの補正を示す図

20

【図13】 ゲーム装置のメインメモリに記憶される主なデータを示す図

【図14】 ゲーム装置において実行される処理の流れを示すメインフローチャート

【図15】 図14に示す加速度に基づく補正処理（ステップS5）の流れを示すフローチャート

【図16】 図15に示す静的補正処理（ステップS12）の流れを示すフローチャート

【図17】 図15に示す動的補正処理（ステップS13）の流れを示すフローチャート

【図18】 動的補正処理における重力補正ベクトルの算出方法を示す図

【図19】 動的補正処理における重力補正ベクトルの算出方法を示す図

【図20】 図15に示す動的補正処理（ステップS13）の変形例を示すフローチャート

【図21】 図14に示す撮像画像に基づく補正処理（ステップS6）の流れを示すフローチャート

30

【図22】 撮像画像に対応する2次元座標を示す図

【発明を実施するための形態】

【0035】

[ゲームシステムの全体構成]

図1を参照して、本発明の一実施形態に係る姿勢算出装置の一例であるゲーム装置を含むゲームシステム1について説明する。図1は、ゲームシステム1の外観図である。以下、据置型のゲーム装置を一例にして、本実施形態のゲーム装置およびゲームプログラムについて説明する。図1において、ゲームシステム1は、テレビジョン受像器（以下、単に「テレビ」と記載する）2、ゲーム装置3、光ディスク4、入力装置8、およびマーカ部6を含む。本システムは、入力装置8を用いたゲーム操作に基づいてゲーム装置3でゲーム処理を実行するものである。

40

【0036】

ゲーム装置3には、当該ゲーム装置3に対して交換可能に用いられる情報記憶媒体の一例である光ディスク4が脱着可能に挿入される。光ディスク4には、ゲーム装置3において実行されるためのゲームプログラムが記憶されている。ゲーム装置3の前面には光ディスク4の挿入口が設けられている。ゲーム装置3は、挿入口に挿入された光ディスク4に記憶されているゲームプログラムを読み出して実行することによってゲーム処理を実行する。

【0037】

50

ゲーム装置3には、表示装置の一例であるテレビ2が接続コードを介して接続される。テレビ2は、ゲーム装置3において実行されるゲーム処理の結果得られるゲーム画像を表示する。また、テレビ2の画面の周辺（図1では画面の上側）には、マーカ部6が設置される。マーカ部6は、その両端に2つのマーカ6Rおよび6Lを備えている。マーカ6R（マーカ6Lも同様）は、具体的には1以上の赤外LEDであり、テレビ2の前方に向かって赤外光を出力する。マーカ部6はゲーム装置3に接続されており、ゲーム装置3はマーカ部6が備える各赤外LEDの点灯を制御することが可能である。

【0038】

入力装置8は、自機に対して行われた操作の内容を示す操作データをゲーム装置3に与えるものである。本実施形態では、入力装置8はコントローラ5とジャイロセンサユニット7とを含む。詳細は後述するが、入力装置8は、コントローラ5に対してジャイロセンサユニット7が着脱可能に接続されている構成である。コントローラ5とゲーム装置3とは無線通信によって接続される。本実施形態では、コントローラ5とゲーム装置3との間の無線通信には例えばBluetooth（ブルートゥース）（登録商標）の技術が用いられる。なお、他の実施形態においてはコントローラ5とゲーム装置3とは有線で接続されてもよい。

【0039】

[ゲーム装置3の内部構成]

次に、図2を参照して、ゲーム装置3の内部構成について説明する。図2は、ゲーム装置3の構成を示すブロック図である。ゲーム装置3は、CPU10、システムLSI11、外部メインメモリ12、ROM/RTC13、ディスクドライブ14、およびAV-IC15等を有する。

【0040】

CPU10は、光ディスク4に記憶されたゲームプログラムを実行することによってゲーム処理を実行するものであり、ゲームプロセッサとして機能する。CPU10は、システムLSI11に接続される。システムLSI11には、CPU10の他、外部メインメモリ12、ROM/RTC13、ディスクドライブ14およびAV-IC15が接続される。システムLSI11は、それに接続される各構成要素間のデータ転送の制御、表示すべき画像の生成、外部装置からのデータの取得等の処理を行う。システムLSIの内部構成について後述する。揮発性の外部メインメモリ12は、光ディスク4から読み出されたゲームプログラムや、フラッシュメモリ17から読み出されたゲームプログラム等のプログラムを記憶したり、各種データを記憶したりするものであり、CPU10のワーク領域やバッファ領域として用いられる。ROM/RTC13は、ゲーム装置3の起動用のプログラムが組み込まれるROM（いわゆるブートROM）と、時間をカウントするクロック回路（RTC: Real Time Clock）とを有する。ディスクドライブ14は、光ディスク4からプログラムデータやテクスチャデータ等を読み出し、後述する内部メインメモリ11eまたは外部メインメモリ12に読み出したデータを書き込む。

【0041】

また、システムLSI11には、入出力プロセッサ（I/Oプロセッサ）11a、GPU（Graphics Processor Unit）11b、DSP（Digital Signal Processor）11c、VRA 1d、および内部メインメモリ11eが設けられる。図示は省略するが、これらの構成要素11a～11eは内部バスによって互いに接続される。

【0042】

GPU11bは、描画手段の一部を形成し、CPU10からのグラフィクスコマンド（作画命令）に従って画像を生成する。VRAM11dは、GPU11bがグラフィクスコマンドを実行するために必要なデータ（ポリゴンデータやテクスチャデータ等のデータ）を記憶する。画像が生成される際には、GPU11bは、VRAM11dに記憶されたデータを用いて画像データを作成する。

【0043】

DSP11cは、オーディオプロセッサとして機能し、内部メインメモリ11eや外部メインメモリ12に記憶されるサウンドデータや音波形（音色）データを用いて、音声データを生成する。

【0044】

上述のように生成された画像データおよび音声データは、AV-IC15によって読み出される。AV-IC15は、読み出した画像データをAVコネクタ16を介してテレビ2に出力するとともに、読み出した音声データを、テレビ2に内蔵されるスピーカ2aに出力する。これによって、画像がテレビ2に表示されるとともに音がスピーカ2aから出力される。

【0045】

入出力プロセッサ11aは、それに接続される構成要素との間でデータの送受信を実行したり、外部装置からのデータのダウンロードを実行したりする。入出力プロセッサ11aは、フラッシュメモリ17、無線通信モジュール18、無線コントローラモジュール19、拡張コネクタ20、およびメモリカード用コネクタ21に接続される。無線通信モジュール18にはアンテナ22が接続され、無線コントローラモジュール19にはアンテナ23が接続される。

【0046】

入出力プロセッサ11aは、無線通信モジュール18およびアンテナ22を介してネットワークに接続し、ネットワークに接続される他のゲーム装置や各種サーバと通信することができる。入出力プロセッサ11aは、定期的にフラッシュメモリ17にアクセスし、ネットワークへ送信する必要があるデータの有無を検出し、当該データが有る場合には、無線通信モジュール18およびアンテナ22を介してネットワークに送信する。また、入出力プロセッサ11aは、他のゲーム装置から送信されてくるデータやダウンロードサーバからダウンロードしたデータを、ネットワーク、アンテナ22および無線通信モジュール18を介して受信し、受信したデータをフラッシュメモリ17に記憶する。CPU10はゲームプログラムを実行することにより、フラッシュメモリ17に記憶されたデータを読み出してゲームプログラムで利用する。フラッシュメモリ17には、ゲーム装置3と他のゲーム装置や各種サーバとの間で送受信されるデータの他、ゲーム装置3を利用してプレイしたゲームのセーブデータ（ゲームの結果データまたは途中データ）が記憶されてもよい。

【0047】

また、入出力プロセッサ11aは、コントローラ5から送信される操作データをアンテナ23および無線コントローラモジュール19を介して受信し、内部メインメモリ11eまたは外部メインメモリ12のバッファ領域に記憶（一時記憶）する。

【0048】

さらに、入出力プロセッサ11aには、拡張コネクタ20およびメモリカード用コネクタ21が接続される。拡張コネクタ20は、USBやSCSIのようなインターフェースのためのコネクタであり、外部記憶媒体のようなメディアを接続したり、他のコントローラのような周辺機器を接続したり、有線の通信用コネクタを接続することによって無線通信モジュール18に替えてネットワークとの通信を行ったりすることができる。メモリカード用コネクタ21は、メモリカードのような外部記憶媒体を接続するためのコネクタである。例えば、入出力プロセッサ11aは、拡張コネクタ20やメモリカード用コネクタ21を介して外部記憶媒体にアクセスし、外部記憶媒体にデータを保存したり、外部記憶媒体からデータを読み出したりすることができる。

【0049】

ゲーム装置3には、電源ボタン24、リセットボタン25、およびイジェクトボタン26が設けられる。電源ボタン24およびリセットボタン25は、システムLSI11に接続される。電源ボタン24がオンされると、ゲーム装置3の各構成要素に対して、図示しないACアダプタを経て電源が供給される。リセットボタン25が押されると、システムLSI11は、ゲーム装置3の起動プログラムを再起動する。イジェクトボタン26は、

10

20

30

40

50

ディスクドライブ14に接続される。イジェクトボタン26が押されると、ディスクドライブ14から光ディスク4が排出される。

【0050】

〔入力装置8の構成〕

次に、図3～図6を参照して、入力装置8について説明する。図3は、入力装置8の外観構成を示す斜視図である。図4は、コントローラ5の外観構成を示す斜視図である。図3は、コントローラ5の上側後方から見た斜視図であり、図4は、コントローラ5を下側前方から見た斜視図である。

【0051】

図3および図4において、コントローラ5は、例えばプラスチック成型によって形成されたハウジング31を有している。ハウジング31は、その前後方向（図3に示すZ軸方向）を長手方向とした略直方体形状を有しており、全体として大人や子供の片手で把持可能な大きさである。プレイヤは、コントローラ5に設けられたボタンを押下すること、および、コントローラ5自体を動かしてその位置や姿勢を変えることによってゲーム操作を行うことができる。

10

【0052】

ハウジング31には、複数の操作ボタンが設けられる。図3に示すように、ハウジング31の上面には、十字ボタン32a、1番ボタン32b、2番ボタン32c、Aボタン32d、マイナスボタン32e、ホームボタン32f、プラスボタン32g、および電源ボタン32hが設けられる。本明細書では、これらのボタン32a～32hが設けられるハウジング31の上面を「ボタン面」と呼ぶことがある。一方、図4に示すように、ハウジング31の下面には凹部が形成されており、当該凹部の後面側傾斜面にはBボタン32iが設けられる。これらの各操作ボタン32a～32iには、ゲーム装置3が実行するゲームプログラムに応じた機能が適宜割り当てられる。また、電源ボタン32hは遠隔からゲーム装置3本体の電源をオン／オフするためのものである。ホームボタン32fおよび電源ボタン32hは、その上面がハウジング31の上面に埋没している。これによって、プレイヤがホームボタン32fまたは電源ボタン32hを誤って押下することを防止することができる。

20

【0053】

ハウジング31の後面にはコネクタ33が設けられている。コネクタ33は、コントローラ5に他の機器（例えば、ジャイロセンサユニット7や他のコントローラ）を接続するために利用される。また、ハウジング31の後面におけるコネクタ33の両側には、上記他の機器が容易に離脱することを防止するために係止穴33aが設けられている。

30

【0054】

ハウジング31上面の後方には複数（図3では4つ）のLED34a～34dが設けられる。ここで、コントローラ5には、他のメインコントローラと区別するためにコントローラ種別（番号）が付与される。各LED34a～34dは、コントローラ5に現在設定されている上記コントローラ種別をプレイヤに通知したり、コントローラ5の電池残量をプレイヤに通知したりする等の目的で用いられる。具体的には、コントローラ5を用いてゲーム操作が行われる際、上記コントローラ種別に応じて複数のLED34a～34dのいずれか1つが点灯する。

40

【0055】

また、コントローラ5は撮像情報演算部35（図6）を有しており、図4に示すように、ハウジング31前面には撮像情報演算部35の光入射面35aが設けられる。光入射面35aは、マーカ6Rおよび6Lからの赤外光を少なくとも透過する材質で構成される。

【0056】

ハウジング31上面における1番ボタン32bとホームボタン32fとの間には、コントローラ5に内蔵されるスピーカ49（図5）からの音を外部に放出するための音抜き孔31aが形成されている。

【0057】

50

次に、図5および図6を参照して、コントローラ5の内部構造について説明する。図5および図6は、コントローラ5の内部構造を示す図である。なお、図5は、コントローラ5の上筐体（ハウジング31の一部）を外した状態を示す斜視図である。図6は、コントローラ5の下筐体（ハウジング31の一部）を外した状態を示す斜視図である。図6に示す斜視図は、図5に示す基板30を裏面から見た斜視図となっている。

【0058】

図5において、ハウジング31の内部には基板30が固設されており、当該基板30の上主面上に各操作ボタン32a～32h、各LED34a～34d、加速度センサ37、アンテナ45、およびスピーカ49等が設けられる。これらは、基板30等に形成された配線（図示せず）によってマイクロコンピュータ（Micro Computer：マイコン）42（図6参照）に接続される。本実施形態では、加速度センサ37は、X軸方向に関してコントローラ5の中心からずれた位置に配置されている。これによって、コントローラ5をZ軸回りに回転させたときのコントローラ5の動きが算出しやすくなる。また、加速度センサ37は、長手方向（Z軸方向）に関してコントローラ5の中心よりも前方に配置されている。また、無線モジュール44（図6）およびアンテナ45によって、コントローラ5がワイヤレスコントローラとして機能する。

【0059】

一方、図6において、基板30の下主面上の前端縁に撮像情報演算部35が設けられる。撮像情報演算部35は、コントローラ5の前方から順に赤外線フィルタ38、レンズ39、撮像素子40、および画像処理回路41を備えている。これらの部材38～41はそれぞれ基板30の下主面に取り付けられる。

【0060】

さらに、基板30の下主面上には、上記マイコン42およびバイブ्रेータ48が設けられている。バイブ्रेータ48は、例えば振動モータやソレノイドであり、基板30等に形成された配線によってマイコン42と接続される。マイコン42の指示によりバイブ्रेータ48が作動することによってコントローラ5に振動が発生する。これによって、コントローラ5を把持しているプレイヤの手にその振動が伝達される、いわゆる振動対応ゲームを実現することができる。本実施形態では、バイブ्रेータ48は、ハウジング31のやや前方寄りに配置される。つまり、バイブ्रेータ48がコントローラ5の中心よりも端側に配置することによって、バイブ्रेータ48の振動によりコントローラ5全体を大きく振動させることができる。また、コネクタ33は、基板30の下主面上の後端縁に取り付けられる。なお、図5および図6に示す他、コントローラ5は、マイコン42の基本クロックを生成する水晶振動子、スピーカ49に音声信号を出力するアンプ等を備えている。

【0061】

また、ジャイロセンサユニット7は、3軸回りの角速度を検知するジャイロセンサ（図7に示すジャイロセンサ55および56）を有する。ジャイロセンサユニット7は、コントローラ5のコネクタ33に着脱可能に装着される。ジャイロセンサユニット7の前端（図3に示すZ軸正方向側の端部）には、コネクタ33に接続可能なプラグ（図7に示すプラグ53）が設けられる。さらに、プラグ53の両側にはフック（図示せず）が設けられる。ジャイロセンサユニット7がコントローラ5に対して装着される状態では、プラグ53がコネクタ33に接続されるとともに、上記フックがコントローラ5の係止穴33aに係止する。これによって、コントローラ5とジャイロセンサユニット7とがしっかりと固定される。また、ジャイロセンサユニット7は側面（図3に示すX軸方向の面）にボタン51を有している。ボタン51は、それを押下すれば上記フックの係止穴33aに対する係止状態を解除することができるように構成されている。したがって、ボタン51を押下しながらプラグ53をコネクタ33から抜くことによって、ジャイロセンサユニット7をコントローラ5から離脱することができる。

【0062】

また、ジャイロセンサユニット7の後端には、上記コネクタ33と同形状のコネクタが設けられる。したがって、コントローラ5（のコネクタ33）に対して装着可能な他の機

10

20

30

40

50

器は、ジャイロセンサユニット7のコネクタに対しても装着可能である。なお、図3においては、当該コネクタに対してカバー52が着脱可能に装着されている。

【0063】

なお、図3～図6に示したコントローラ5およびジャイロセンサユニット7の形状や、各操作ボタンの形状、加速度センサやバイブレータの数および設置位置等は単なる一例に過ぎず、他の形状、数、および設置位置であっても、本発明を実現することができる。また、本実施形態では、撮像手段による撮像方向はZ軸正方向であるが、撮像方向はいずれの方向であってもよい。すなわち、コントローラ5における撮像情報演算部35の位置（撮像情報演算部35の光入射面35a）は、ハウジング31の前面でなくてもよく、ハウジング31の外部から光を取り入れることができれば他の面に設けられてもかまわない。

10

【0064】

図7は、入力装置8（コントローラ5およびジャイロセンサユニット7）の構成を示すブロック図である。コントローラ5は、操作部32（各操作ボタン32a～32i）、コネクタ33、撮像情報演算部35、通信部36、および加速度センサ37を備えている。コントローラ5は、自機に対して行われた操作内容を示すデータを操作データとしてゲーム装置3へ送信するものである。

【0065】

操作部32は、上述した各操作ボタン32a～32iを含み、各操作ボタン32a～32iに対する入力状態（各操作ボタン32a～32iが押下されたか否か）を示す操作ボタンデータを通信部36のマイコン42へ出力する。

20

【0066】

撮像情報演算部35は、撮像手段が撮像した画像データを解析してその中で輝度が高い領域を判別してその領域の重心位置やサイズなどを算出するためのシステムである。撮像情報演算部35は、例えば最大200フレーム／秒程度のサンプリング周期を有するので、比較的高速なコントローラ5の動きでも追跡して解析することができる。

【0067】

撮像情報演算部35は、赤外線フィルタ38、レンズ39、撮像素子40、および画像処理回路41を含んでいる。赤外線フィルタ38は、コントローラ5の前方から入射する光から赤外線のみを通過させる。レンズ39は、赤外線フィルタ38を透過した赤外線を集光して撮像素子40へ入射させる。撮像素子40は、例えばCMOSセンサやあるいはCCDセンサのような固体撮像素子であり、レンズ39が集光した赤外線を受光して画像信号を出力する。ここで、テレビ2の表示画面近傍に配置されるマーカ部6のマーカ6Rおよび6Lは、テレビ2の前方に向かって赤外光を出力する赤外LEDで構成される。したがって、赤外線フィルタ38を設けることによって、撮像素子40は、赤外線フィルタ38を通過した赤外線だけを受光して画像データを生成するので、マーカ6Rおよび6Lの画像をより正確に撮像することができる。以下では、撮像素子40によって撮像された画像を撮像画像と呼ぶ。撮像素子40によって生成された画像データは、画像処理回路41で処理される。画像処理回路41は、撮像画像内における撮像対象（マーカ6Rおよび6L）の位置を算出する。画像処理回路41は、算出された位置を示す座標を通信部36のマイコン42へ出力する。この座標のデータは、マイコン42によって操作データとしてゲーム装置3に送信される。以下では、上記座標を「マーカ座標」と呼ぶ。マーカ座標はコントローラ5自体の向き（傾斜角度）や位置に対応して変化するので、ゲーム装置3はこのマーカ座標を用いてコントローラ5の向きや位置を算出することができる。

30

40

【0068】

なお、他の実施形態においては、コントローラ5は画像処理回路41を備えていない構成であってもよく、撮像画像自体がコントローラ5からゲーム装置3へ送信されてもよい。このとき、ゲーム装置3は、画像処理回路41と同様の機能を有する回路あるいはプログラムを有しており、上記マーカ座標を算出するようにしてもよい。

【0069】

加速度センサ37は、コントローラ5の加速度（重力加速度を含む）を検出する、すな

50

わち、コントローラ 5 に加わる力（重力を含む）を検出する。加速度センサ 37 は、当該加速度センサ 37 の検出部に加わっている加速度のうち、センシング軸方向に沿った直線方向の加速度（直線加速度）の値を検出する。例えば、2 軸以上の多軸加速度センサの場合には、加速度センサの検出部に加わっている加速度として、各軸に沿った成分の加速度をそれぞれ検出する。例えば、3 軸または 2 軸の加速度センサは、アナログ・デバイゼズ株式会社（Analog Devices, Inc.）または ST マイクロエレクトロニクス社（STMicroelectronics N.V.）から入手可能である種類のものでもよい。なお、加速度センサ 37 は、例えば静電容量式の加速度センサであるとするが、他の方式の加速度センサを用いるようにしてもよい。

【0070】

本実施形態では、加速度センサ 37 は、コントローラ 5 を基準とした上下方向（図 3 に示す Y 軸方向）、左右方向（図 3 に示す X 軸方向）および前後方向（図 3 に示す Z 軸方向）の 3 軸方向に関してそれぞれ直線加速度を検出する。加速度センサ 37 は、各軸に沿った直線方向に関する加速度を検出するものであるため、加速度センサ 37 からの出力は 3 軸それぞれの直線加速度の値を表すものとなる。すなわち、検出された加速度は、入力装置 8（コントローラ 5）を基準に設定される XYZ 座標系（コントローラ座標系）における 3 次元のベクトル（ a_x , a_y , a_z ）として表される。以下では、加速度センサ 37 によって検出される 3 軸に関する各加速度値を各成分とするベクトルを加速度ベクトルと呼ぶ。また、以下では、当該加速度ベクトルを、後述する運動加速度ベクトルと明確に区別する目的で、「検出加速度ベクトル」と呼ぶことがある。

【0071】

加速度センサ 37 が検出した加速度を示すデータ（加速度データ）は、通信部 36 へ出力される。なお、加速度センサ 37 が検出した加速度は、コントローラ 5 自体の向き（傾斜角度）や動きに対応して変化するので、ゲーム装置 3 は加速度データを用いてコントローラ 5 の向きや動きを算出することができる。本実施形態では、ゲーム装置 3 は、加速度データに基づいてコントローラ 5 の姿勢を判断する。

【0072】

加速度センサ 37 が検出した加速度（加速度ベクトル）を示すデータ（加速度データ）は、通信部 36 へ出力される。本実施形態において、加速度センサ 37 は、コントローラ 5 の傾斜角度を判断するためのデータを出力するセンサとして用いられる。

【0073】

なお、加速度センサ 37 から出力される加速度の信号に基づいて、ゲーム装置 3 のプロセッサ（例えば CPU 10）またはコントローラ 5 のプロセッサ（例えばマイコン 42）等のコンピュータが処理を行うことによって、コントローラ 5 に関するさらなる情報を推測または算出（判定）することができることは、当業者であれば本明細書の説明から容易に理解できるであろう。例えば、加速度センサ 37 を搭載するコントローラ 5 が静止状態であることを前提としてコンピュータ側の処理が実行される場合（すなわち、加速度センサによって検出される加速度が重力加速度のみであるとして処理が実行される場合）、コントローラ 5 が現実的に静止状態であれば、検出された加速度に基づいてコントローラ 5 の姿勢が重力方向に対して傾いているか否かまたはどの程度傾いているかを知ることができる。具体的には、加速度センサ 37 の検出軸が鉛直下方向を向いている状態を基準としたとき、1 G（重力加速度）がかかっているか否かによって、コントローラ 5 が基準に対して傾いているか否かを知ることができるし、その大きさによって基準に対してどの程度傾いているかも知ることができる。また、多軸の加速度センサ 37 の場合には、さらに各軸の加速度の信号に対して処理を施すことによって、重力方向に対してコントローラ 5 がどの程度傾いているかをより詳細に知ることができる。この場合において、プロセッサは、加速度センサ 37 からの出力に基づいてコントローラ 5 の傾斜角度を算出してもよいし、当該傾斜角度を算出せずに、コントローラ 5 の傾斜方向を算出するようにしてもよい。このように、加速度センサ 37 をプロセッサと組み合わせて用いることによって、コントローラ 5 の傾斜角度または姿勢を判定することができる。

【0074】

一方、コントローラ5が動的な状態（コントローラ5が動かされている状態）であることを前提とする場合には、加速度センサ37は重力加速度に加えてコントローラ5の動きに応じた加速度を検出するので、検出された加速度から重力加速度の成分を所定の処理により除去することによってコントローラ5の動き方向を知ることができる。また、コントローラ5が動的な状態であることを前提とする場合であっても、検出された加速度から、加速度センサの動きに応じた加速度の成分を所定の処理により除去することによって、重力方向に対するコントローラ5の傾きを知ることが可能である。なお、他の実施例では、加速度センサ37は、内蔵の加速度検出手段で検出された加速度信号をマイコン42に出力する前に当該加速度信号に対して所定の処理を行うための、組込み式の処理装置または他の種類の専用の処理装置を備えていてもよい。組込み式または専用の処理装置は、例えば、加速度センサ37が静的な加速度（例えば、重力加速度）を検出するために用いられる場合、加速度信号を傾斜角（あるいは、他の好ましいパラメータ）に変換するものであってもよい。

10

【0075】

通信部36は、マイコン42、メモリ43、無線モジュール44、およびアンテナ45を含んでいる。マイコン42は、処理を行う際にメモリ43を記憶領域として用いながら、マイコン42が取得したデータをゲーム装置3へ無線送信する無線モジュール44を制御する。また、マイコン42はコネクタ33に接続されている。ジャイロセンサユニット7から送信されてくるデータは、コネクタ33を介してマイコン42に入力される。以下、ジャイロセンサユニット7の構成について説明する。

20

【0076】

ジャイロセンサユニット7は、プラグ53、マイコン54、2軸ジャイロセンサ55、および1軸ジャイロセンサ56を備えている。上述のように、ジャイロセンサユニット7は、3軸（本実施形態では、XYZ軸）周りの角速度を検出し、検出した角速度を示すデータ（角速度データ）をコントローラ5へ送信する。

【0077】

2軸ジャイロセンサ55は、X軸周りの角速度およびY軸周りの（単位時間あたりの）角速度を検出する。また、1軸ジャイロセンサ56は、Z軸周りの（単位時間あたりの）角速度を検出する。なお、本明細書では、コントローラ5の撮像方向（Z軸正方向）を基準として、XYZ軸周りの回転方向を、それぞれ、ロール方向、ピッチ方向、ヨー方向と呼ぶ。すなわち、2軸ジャイロセンサ55は、ロール方向（X軸周りの回転方向）およびピッチ方向（Y軸周りの回転方向）の角速度を検出し、1軸ジャイロセンサ56は、ヨー方向（Z軸周りの回転方向）の角速度を検出する。

30

【0078】

なお、本実施形態では、3軸回りの角速度を検出するために、2軸ジャイロセンサ55と1軸ジャイロセンサ56とを用いる構成としたが、他の実施形態においては、3軸回りの角速度を検出することができればよく、用いるジャイロセンサの数および組み合わせはどのようなものであってもよい。

【0079】

また、本実施形態では、後述する姿勢算出処理における計算を容易にする目的で、各ジャイロセンサ55および56が角速度を検出する3つの軸は、加速度センサ37が加速度を検出する3つの軸（XYZ軸）と一致するように設定される。ただし、他の実施形態においては、各ジャイロセンサ56および57が角速度を検出する3つの軸と、加速度センサ37が加速度を検出する3つの軸とは一致しなくてもよい。

40

【0080】

各ジャイロセンサ56および57で検出された角速度を示すデータは、マイコン54に出力される。したがって、マイコン54には、XYZ軸の3軸回りの角速度を示すデータが入力されることになる。マイコン54は、上記3軸回りの角速度を示すデータを角速度データとしてプラグ53を介してコントローラ5へ送信する。なお、マイコン54から

50

コントローラ 5 への送信は所定の周期毎に逐次行われるが、ゲームの処理は 1 / 60 秒を単位として（1 フレーム時間として）行われることが一般的であるので、この時間以下の周期で送信を行うことが好ましい。

【0081】

コントローラ 5 の説明に戻り、操作部 32、撮像情報演算部 35、および加速度センサ 37 からマイコン 42 へ出力されたデータ、ならびに、ジャイロセンサユニット 7 からマイコン 42 へ送信されてきたデータは、一時的にメモリ 43 に格納される。これらのデータは、上記操作データとしてゲーム装置 3 へ送信される。すなわち、マイコン 42 は、ゲーム装置 3 の無線コントローラモジュール 19 への送信タイミングが到来すると、メモリ 43 に格納されている操作データを無線モジュール 44 へ出力する。無線モジュール 44 は、例えば Bluetooth (ブルートゥース) (登録商標) の技術を用いて、所定周波数の搬送波を操作データで変調し、その微弱電波信号をアンテナ 45 から放射する。つまり、操作データは、無線モジュール 44 で微弱電波信号に変調されてコントローラ 5 から送信される。微弱電波信号はゲーム装置 3 側の無線コントローラモジュール 19 で受信される。受信された微弱電波信号について復調や復号を行うことによって、ゲーム装置 3 は操作データを取得することができる。そして、ゲーム装置 3 の CPU 10 は、取得した操作データとゲームプログラムとに基づいて、ゲーム処理を行う。なお、通信部 36 から無線コントローラモジュール 19 への無線送信は所定の周期毎に逐次行われるが、ゲームの処理は 1 / 60 秒を単位として（1 フレーム時間として）行われることが一般的であるので、この時間以下の周期で送信を行うことが好ましい。コントローラ 5 の通信部 36 は、例えば 1 / 200 秒に 1 回の割合で各操作データをゲーム装置 3 の無線コントローラモジュール 19 へ出力する。

【0082】

上記コントローラ 5 を用いることによって、プレイヤは、各操作ボタンを押下する従来の一般的なゲーム操作に加えて、コントローラ 5 を任意の傾斜角度に傾ける操作を行うことができる。その他、上記コントローラ 5 によれば、プレイヤは、コントローラ 5 によって画面上の任意の位置を指示する操作、および、コントローラ 5 自体を動かす操作を行うこともできる。

【0083】

[姿勢算出処理の概要]

次に、ゲーム装置 3 において実行される、入力装置 8 の姿勢を算出する姿勢算出処理の概要を説明する。本実施形態では、入力装置 8 を用いてカヌーを漕ぐゲーム操作を行うゲームを想定する。すなわち、本実施形態では、ゲーム装置 3 は、プレイヤが入力装置 8 をパドルのようにして用いて、パドルで水を掻くように入力装置 8 を動かすことで、仮想のゲーム空間内のパドルを動かしてカヌーを漕ぐゲームを実行するものとする。より具体的には、ゲーム装置 3 は、入力装置 8 の姿勢を算出し、ゲーム空間内のパドルの姿勢を入力装置 8 の姿勢に応じて変化させる。

【0084】

本実施形態では、ゲーム装置 3 は、ジャイロセンサ 55 および 56 と、加速度センサ 37 と、撮像手段（撮像素子 40）とを備えた入力装置 8 から操作データを取得し、入力装置 8 の姿勢を算出する。なお、本実施形態では、入力装置 8 は、ジャイロセンサ 55 および 56 と、加速度センサ 37 と、撮像素子 40 とを備える構成とするが、他の実施形態においては、ジャイロセンサと加速度センサとを少なくとも備える構成であればよい。

【0085】

ゲーム装置 3 は、ジャイロセンサ 55 および 56 が検出する角速度に基づいて入力装置 8 の姿勢を算出する。以下では、当該角速度に基づいて算出される入力装置 8 の姿勢を、「第 1 の姿勢」と呼ぶ。第 1 の姿勢は、ジャイロセンサ 55 および 56 の検出結果に生じる誤差が原因で、実際の姿勢とは異なるおそれがある。そこで、本実施形態では、ゲーム装置 3 は、加速度センサ 37 によって検出される加速度を用いて、上記第 1 の姿勢を補正する。また、撮像素子 40 によって撮像される画像（撮像画像）を用いて、第 1 の姿勢を

補正する。

【0086】

以下、図8～図12を参照して、加速度センサ37が検出する加速度を用いた補正処理の概要について説明する。ゲーム装置3は、上記ジャイロセンサ55および56が検出する角速度から入力装置8の姿勢を算出することができる他、加速度センサ37が検出する加速度からも入力装置8の姿勢を算出することができる。すなわち、加速度センサ37は入力装置8に加えられる重力加速度を検出する（加速度センサ37が検出する加速度には重力加速度が含まれる）ので、ゲーム装置3は、検出される加速度を用いて、入力装置8を基準にした重力方向、すなわち、重力方向に対する入力装置8の傾き（姿勢）を算出することができる。以下では、加速度センサ37が検出する加速度に基づいて算出される姿勢を、「第2の姿勢」と呼ぶ。本実施形態では、上記角速度に基づく第1の姿勢は、上記加速度から算出される重力方向（第2の姿勢）を用いて補正される。

10

【0087】

ここで、本実施形態では、ゲーム装置3は、上記加速度を用いた補正として、静的補正処理と、動的補正処理という2種類の補正処理を行う。静的補正処理とは、入力装置8が主に静止しているまたは静止状態に近い場合に入力装置8の第1の姿勢を補正することを目的とした補正処理である。動的補正処理とは、入力装置8が主に動いている場合に入力装置8の第1の姿勢を補正することを目的とした補正処理である。本実施形態では、静的補正処理および動的補正処理の2種類の補正を行うことによって、入力装置8が静止している場合と動いている場合との両方の状況で、加速度を用いた補正を可能としている。以下、静的補正処理および動的補正処理の概要について説明する。

20

【0088】

まず、静的補正処理について図8および図9を用いて説明する。図8は、入力装置8に関する重力ベクトルおよび検出加速度ベクトルを示す図である。図8に示す重力ベクトル V_g は、入力装置8を基準とした（入力装置8から見た）重力方向を表す。当該重力方向は、空間に設定される鉛直下方向であるので、上記角速度に基づいて第1の姿勢が算出されれば、当該姿勢に対応して入力装置8を基準とした重力方向が算出できる。また、図8に示す検出加速度ベクトル V_a は、加速度センサ37が検出した加速度の向きを表す。なお、図8に示すベクトル V_z は、入力装置8のコントローラ座標系（XYZ座標系）におけるZ軸正方向、すなわち、入力装置8の姿勢を表すベクトルである。

30

【0089】

図8において、入力装置8が静止またはほぼ静止している状態である場合、検出加速度ベクトル V_a は重力方向を表すと推測することができる。したがって、この場合、ゲーム装置3は、角速度から算出された重力ベクトル V_g を、より正確に重力方向を表すと推測される検出加速度ベクトル V_a を用いて補正する。図9は、静的補正処理における重力ベクトル V_g の補正を示す図である。図9においては、ベクトル V_g は補正前の重力ベクトルを示し、ベクトル V_g' は補正後の重力ベクトルを示す。図9に示すように、本実施形態においては、補正前の重力ベクトル V_g は、検出加速度ベクトル V_a に近づく方向へ補正される。これによって、重力ベクトルがより正確に重力方向を表すように補正されたことになり、補正された重力ベクトルを用いて入力装置8の第1の姿勢を補正することによって姿勢を正確に算出することができる。

40

【0090】

なお、詳細は後述するが、静的補正処理において重力ベクトル V_g が検出加速度ベクトル V_a に近づく度合は、入力装置8が静止状態に近いほど大きくなる。つまり、本実施形態においては、検出加速度ベクトル V_a が重力方向を表すものとして信頼できる場合ほど、重力ベクトル V_g が検出加速度ベクトル V_a に近くなるように補正される。これによれば、入力装置8の姿勢をより正確に算出することができる。また、本実施形態においては、入力装置8が静止状態からかけ離れている（激しく動かされている）場合には、静的補正処理における補正が行われない。これによっても、入力装置8の姿勢をより正確に算出することができる。

50

【0091】

なお、入力装置8が静止しているか、あるいはほぼ静止している場合には、加速度センサ37が検出する加速度には重力加速度のみが含まれるので、上記静的補正処理によって重力方向を正確に算出することができる。一方、入力装置8がプレイヤーによって動かされている場合には、重力加速度だけでなく、入力装置8の運動に起因する加速度が加速度センサ37によって検出される。例えば、加速度センサ37が検出する加速度には、遠心力等の慣性力による加速度が含まれる。そのため、入力装置8が動いている場合には、加速度センサ37の出力をそのまま用いても重力方向を正確に判断することができず、上記静的補正処理によって重力方向を正確に算出することができないおそれがある。また、入力装置8が動いている場合には上記静的補正処理を実行しないようにする方法も考えられる。しかし、この方法では、例えば上記カヌーのゲームの場合のように、プレイヤーが入力装置8を絶えず動かしてゲーム操作を行うことが想定される場合には、上記静的補正処理が実行される機会が無く、重力方向の補正処理が実行されないことになる。その結果、入力装置8の姿勢を正確に算出することができない。以上のように、静的補正処理のみでは、入力装置8が動いている場合に対応することができない可能性がある。そこで、本実施形態では、入力装置8が動いている場合にも重力方向の補正処理を行うことを可能にするべく、静的補正処理に加えて、動的補正処理を行うものとする。以下、動的補正処理について図10～図12を用いて説明する。

10

【0092】

図10は、入力装置8に関する重力ベクトル、検出加速度ベクトル、および運動加速度ベクトルを示す図である。図10に示す運動加速度ベクトル V_A は、入力装置8の運動加速度を表すベクトルである。ここで「運動加速度」とは、入力装置8の運動により入力装置8に加えられる加速度のことであり、入力装置8が回転運動を行っている場合に生じる遠心力による加速度や、入力装置8が速度変化を伴って運動している場合に生じる慣性力による加速度を含む。運動加速度ベクトル V_A は、加速度センサ37が検出した加速度から、重力加速度を除いた加速度のベクトルとして算出される。すなわち、運動加速度ベクトル V_A は、上記検出加速度ベクトル V_a から上記重力ベクトル V_g を引いたベクトルである。このように、運動加速度ベクトル V_A は、重力ベクトル V_g から決まるベクトルである。重力ベクトル V_g は角速度に基づく第1の姿勢から算出されるものである。運動加速度ベクトル V_A については重力ベクトル V_g と同様に、ジャイロセンサ55および56の誤検出等の原因で実際の値に比べて誤差が生じるおそれがある。

20

30

【0093】

図11は、入力装置8に関する角速度ベクトルおよび運動加速度ベクトルを示す図である。図11に示す角速度ベクトル V_ω は、ジャイロセンサ55および56が検出する角速度で入力装置8が回転するときの回転軸を表し、ジャイロセンサ55および56の検出結果から得ることができる。ここで、入力装置8が特定の運動を行う場合には、ジャイロセンサ55および56が検出する角速度と運動加速度との間には所定の関係が成立する。ここで、本実施形態のようにプレイヤーがパドルで水を掻くように入力装置8を動かす場合には、入力装置8は所定位置を中心とした回転運動を行うと考えられる。入力装置8を動かす運動は、プレイヤーの肘、肩等の関節を中心とした回転運動であることが多いので、回転運動と見なすことができる（常に回転運動を行うとは限らないが、少なくとも一部の期間では回転運動を行う）。詳細は後述するが、入力装置8が例えば等角速度で回転運動を行う場合には、角速度に対応する回転軸（角速度ベクトル V_ω ）と、入力装置8の運動加速度とが直交する関係が成立する。したがって、図11においては、運動加速度は、ベクトル V_A' のように、角速度ベクトル V_ω と垂直になるべきであるとの推定が成り立つ。例えば、入力装置8が、コントローラ座標系のY軸正方向が鉛直上向きとなる姿勢で、Y軸方向に平行な軸を回転軸として、X軸負方向に向かって回転移動する回転運動を行っている（図10～図12に示す矢印参照）ものとするれば、運動加速度は、ベクトル V_A' のようにY軸方向に垂直な向きになると推定される。

40

【0094】

50

動的補正処理では、ゲーム装置3は、上記の推定を用いて重力ベクトル V_g を補正する。図12は、動的補正処理における重力ベクトルの補正を示す図である。図12に示すように、ゲーム装置3は、入力装置8の現在の角速度（図11に示す角速度ベクトル V_ω ）に対して上記関係を満たす運動加速度（ベクトル $V_{A'}$ ）へと運動加速度ベクトル V_A を近づける方向へ重力ベクトル V_g を補正する。換言すれば、動的補正処理では、加速度センサ37の検出結果である検出加速度ベクトル V_a から算出される運動加速度（ベクトル V_A ）を、ジャイロセンサ55および56の検出結果である角速度ベクトル V_ω から決められる運動加速度（ベクトル $V_{A'}$ ）へと近づけるように、重力ベクトル V_g が補正される。

【0095】

以上のように、動的補正処理では、入力装置8が特定の運動（ここでは、等角速度の回転運動）を行うものとして、当該運動を行う場合における角速度と運動加速度との関係（ここでは、「運動加速度ベクトル V_A が角速度ベクトル V_ω と垂直になる」という関係）を予め定義しておく。そして、運動加速度ベクトル V_A が当該関係を満たすように、運動加速度ベクトル V_A の算出元である重力ベクトル V_g を補正する。これによって、入力装置8が動かされている場合であっても、加速度センサ37が検出した加速度を用いて重力方向を補正することができる。

【0096】

また、本実施形態では、ゲーム装置3は、上記静的補正処理および動的補正処理によって補正された重力ベクトル V_g を用いて入力装置8の第1の姿勢を補正する。すなわち、第1の姿勢を、上記静的補正処理および動的補正処理によって補正された重力ベクトル V_g に対応する姿勢（第2の姿勢）へと近づける補正を行う。本実施形態によれば、入力装置8が静止している場合には静的補正処理によって、入力装置8が動いている場合には動的補正処理によって、それぞれ重力ベクトルが補正され、補正された重力ベクトルを用いて第1の姿勢が補正される。したがって、入力装置8が静止している場合でも動かされている場合でも、入力装置8の第1の姿勢を補正することができ、入力装置8の姿勢を正確に算出することができる。また、本実施形態では、入力装置8が動かされている場合でも姿勢を補正することができるので、プレイヤーが入力装置8を常に動かすような操作が想定される場合に特に有効である。

【0097】

なお、本実施形態においては、加速度センサ37が検出した加速度を用いた補正処理の後、撮像素子40によって撮像される画像（撮像画像）を用いた補正処理が実行される。撮像画像を用いた補正処理では、ゲーム装置3は、まず撮像画像に基づいて入力装置8の姿勢を算出する。すなわち、撮像画像内におけるマーカの画像の位置（マーカ座標）から入力装置8の姿勢を算出する。以下では、撮像画像に基づいて算出される入力装置8の姿勢を、「第3の姿勢」と呼ぶ。次に、ゲーム装置3は、上記角速度に基づく第1の姿勢を、撮像画像から算出された第3の姿勢へと近づける補正を行う。また、本実施形態においては、ゲーム装置3は、加速度に基づく補正処理を先に行い、撮像画像に基づく補正処理を後に行うこととするが、他の実施形態においては、加速度に基づく補正処理と撮像画像に基づく補正処理とのいずれを先に行うようにしてもよい。

【0098】

以上のように、本実施形態によれば、ジャイロセンサ55および56によって検知された角速度から算出される入力装置8の第1の姿勢を、加速度センサ37によって検知された加速度を用いて補正し、また、撮像手段による撮像画像を用いて補正する。これによって、ジャイロセンサから算出される姿勢の誤差を減少することができ、入力装置8の姿勢をより正確に算出することができる。

【0099】

なお、加速度センサ37の検出結果からは、重力方向を軸とした回転（ヨー方向の回転）を検知することはできないので、加速度に基づく補正処理では、ヨー方向に関しては補正を行うことができない。しかし、加速度に基づく補正処理は、入力装置8の姿勢がどの

10

20

30

40

50

ような姿勢であっても（常に加速度を検出することができるので）可能であるという特長を有する。一方、入力装置 8 の撮像方向にマーカ部 6 がなければマーカ座標が検出されない。撮像画像に基づく補正処理は、入力装置 8 の姿勢によっては行うことができない。しかし、撮像画像に基づく補正処理は、姿勢（特にロール方向の姿勢）を正確に算出することができるという特長を有する。本実施形態においては、このように特長の異なる 2 種類の補正を行うことによって、入力装置 8 の姿勢をより正確に算出することができる。

【0100】

なお、他の実施形態においては、ゲーム装置 3 は、撮像画像を用いた補正処理を実行しなくてもよい。この場合、入力装置 8 は、撮像情報演算部 35 を備えていない構成であってもよい。

10

【0101】

[ゲーム装置 3 における処理の詳細]

次に、ゲーム装置 3 において実行される処理の詳細について説明する。まず、ゲーム装置 3 における処理において用いられる主なデータについて図 13 を用いて説明する。図 13 は、ゲーム装置 3 のメインメモリ（外部メインメモリ 12 または内部メインメモリ 11e）に記憶される主なデータを示す図である。図 13 に示すように、ゲーム装置 3 のメインメモリには、ゲームプログラム 60、操作データ 62、およびゲーム処理用データ 67 が記憶される。なお、メインメモリには、図 13 に示すデータその他、ゲームに登場する各種オブジェクトの画像データや、オブジェクトの各種パラメータを示すデータ等、ゲーム処理に必要なデータが記憶される。

20

【0102】

ゲームプログラム 60 は、ゲーム装置 3 に電源が投入された後の適宜のタイミングで光ディスク 4 からその一部または全部が読み込まれてメインメモリに記憶される。ゲームプログラム 60 には、姿勢算出プログラム 61 が含まれる。姿勢算出プログラム 61 は、入力装置 8 の姿勢を算出するための姿勢算出処理を実行するためのプログラムである。

【0103】

操作データ 62 は、コントローラ 5 からゲーム装置 3 へ送信されてくる操作データである。上述したように、コントローラ 5 からゲーム装置 3 へ 1/200 秒に 1 回の割合で操作データが送信されるので、メインメモリに記憶される操作データ 62 はこの割合で更新される。

30

【0104】

操作データ 62 には、角速度データ 63、加速度データ 64、マーカ座標データ 65、および操作ボタンデータ 66 が含まれる。角速度データ 63 は、ジャイロセンサユニット 7 のジャイロセンサ 55 および 56 によって検出された角速度を示すデータである。ここでは、角速度データ 63 は、図 3 に示す X Y Z 座標系の 3 軸回りのそれぞれの角速度を示す。3 軸周りの角速度（時計回り）のそれぞれの大きさを 3 次元のベクトルとして表すと、角速度データ 63 は、回転面に垂直な方向を向き（より具体的には、右ネジを角速度の方向へ回転させた場合における右ネジの進行方向を向き）、その大きさが角速度 ω の大きさを表す角速度ベクトル $V\omega$ を示す。なお、本実施形態では、入力装置 8 の角加速度を算出するために、最後に取得された操作データの前に取得された操作データに含まれる角速度データが角速度履歴データとしてメインメモリに記憶される。

40

【0105】

加速度データ 64 は、加速度センサ 37 によって検出された加速度（検出加速度ベクトル）を示すデータである。ここでは、加速度データ 64 は、図 3 に示す X Y Z の 3 軸の方向に関する加速度を各成分とする 3 次元の加速度ベクトルを示す。また、本実施形態においては、コントローラ 5 が静止している状態で加速度センサ 37 が検出する加速度ベクトルの大きさを“1”とする。つまり、加速度センサ 37 によって検出される重力加速度の大きさは“1”である。

【0106】

マーカ座標データ 65 は、撮像情報演算部 35 の画像処理回路 41 によって算出される

50

座標、すなわち上記マーカ座標を示すデータである。マーカ座標は、撮像画像に対応する平面上の位置を表すための２次元座標系（図１７に示す x' 、 y' 座標系）で表現される。なお、撮像素子４０によって２つのマーカ６Ｒおよび６Ｌが撮像される場合には、２つのマーカ座標が算出される。一方、撮像素子４０の撮像可能な範囲内にマーカ６Ｒおよび６Ｌのいずれか一方が位置しない場合には、撮像素子４０によって１つのマーカのみが撮像され、１つのマーカ座標のみが算出される。また、撮像素子４０の撮像可能な範囲内にマーカ６Ｒおよび６Ｌの両方が位置しない場合には、撮像素子４０によってマーカが撮像されず、マーカ座標は算出されない。したがって、マーカ座標データ６５は、２つのマーカ座標を示す場合もあるし、１つのマーカ座標を示す場合もあるし、マーカ座標がないことを示す場合もある。

10

【０１０７】

操作ボタンデータ６６は、各操作ボタン３２ａ～３２ｉに対する入力状態を示すデータである。

【０１０８】

ゲーム処理用データ６７は、後述するゲーム処理（図１４）において用いられるデータである。ゲーム処理用データ６７は、第１姿勢データ６８、重力方向データ６９、加速度大きさデータ７０、加加速度データ７１、第１補正度データ７２、第２補正度データ７３、運動加速度データ７４、重力補正ベクトルデータ７５、補正行列データ７６、ロール姿勢成分データ７７、ヨー姿勢成分データ７８、ピッチ姿勢成分データ７９、および第３姿勢データ８０を含む。なお、図１３に示すデータの他、ゲーム処理用データ６７は、ゲーム処理において用いられる各種データ（ゲームパラメータを示すデータ等）を含む。

20

【０１０９】

第１姿勢データ６８は、角速度データ６３を用いて算出される入力装置８の上記第１の姿勢を示すデータである。本実施形態では、入力装置８の姿勢は、以下の式（１）に示す 3×3 の行列 $M1$ で表現される。

【数１】

$$M1 = \begin{bmatrix} X_x & Y_x & Z_x \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{bmatrix} \cdots (1)$$

30

上記行列 $M1$ は、所定の基準姿勢から現在の入力装置８の姿勢への回転を表す回転行列である。以下では、上記第１の姿勢を示す行列 $M1$ を、「第１姿勢行列 $M1$ 」と呼ぶ。なお、第１姿勢行列 $M1$ により表される姿勢は、入力装置８が存在する空間の所定位置を基準とした x 、 y 、 z 座標系（上記空間座標系）における姿勢である。ここでは、 x 、 y 、 z 座標系は、入力装置８がマーカ部６の正面に位置することを前提とし、入力装置８の位置からマーカ部６を向く方向を z 軸正方向とし、鉛直上向き（重力方向の逆方向）を y 軸正方向とし、入力装置８の位置からマーカ部６を向いた場合の左方向を x 軸正方向とした座標系であるとする。また、上記所定の基準姿勢は、マーカ部６の正面に位置する入力装置８の撮像方向がマーカ部６の中央を向き、かつ、コントローラ５のボタン面が鉛直上向きとなる姿勢（すなわち、入力装置８を基準とした X 軸、 Y 軸、 Z 軸が、それぞれ x 軸、 y 軸、 z 軸の向きと一致する姿勢）であるとする。したがって、回転行列 $M1$ は、入力装置８の X 軸、 Y 軸、 Z 軸の向きを示す単位ベクトルを、それぞれ x 、 y 、 z 座標系で示したものを並べたものでもある。なお、本実施形態では、行列を用いて入力装置８の姿勢を表現することとしたが、他の実施形態においては、入力装置８の姿勢は、３次のベクトルまたは３つの角度によって表現されてもよい。

40

【０１１０】

重力方向データ６９は、入力装置８から見た重力方向、すなわち、上記重力ベクトル V_g を示すデータである。重力ベクトル V_g は、上記コントローラ座標系（ X 、 Y 、 Z 座標系）における方向を表す単位ベクトルである。

50

【0111】

加速度大きさデータ70は、加速度データ64により示される検出加速度ベクトル V_a の大きさ（長さ） L を示すデータである。

【0112】

加加速度データ71は、入力装置8の加速度の単位時間あたりの変化量（加加速度） J を示すデータである。本実施形態では、加加速度は、最後に算出された運動加速度ベクトルと、その前に算出された運動加速度ベクトルとの差として算出される。加加速度は3次元のベクトルとして算出可能であるが、本実施形態では加加速度の大きさのみを用いるので、加加速度データ71は、加加速度の大きさを示すスカラー量を示す。加加速度データ71は、後述する第1補正度Aおよび第2補正度Bを算出するために用いられる。

10

【0113】

第1補正度データ72は、静的補正処理によって上記重力ベクトル V_g を補正する度合（第1補正度A）を示すデータである。また、第2補正度データ73は、動的補正処理によって上記重力ベクトル V_g を補正する度合（第2補正度B）を示す。詳細は後述するが、各補正度の値が大きいくほど補正量は大きくなる。

【0114】

運動加速度データ74は、上記運動加速度ベクトル V_A を示すデータである。運動加速度ベクトル V_A は、重力ベクトル V_g と同様、上記コントローラ座標系（XYZ座標系）における方向を表すベクトルである。運動加速度ベクトル V_A は、上記加速度データ64および重力方向データ69に基づいて算出される。

20

【0115】

重力補正ベクトルデータ75は、上記動的補正処理において重力ベクトルを補正するために用いられる重力補正ベクトル ΔV_g を示すデータである。詳細は後述するが、動的補正処理による補正後の重力ベクトルは、補正前の重力ベクトルに重力補正ベクトル ΔV_g を加算した方向を向く。

【0116】

補正行列データ76は、補正行列 M_a を示すデータである。補正行列 M_a は、加速度に基づく補正処理において、入力装置8の第1の姿勢を補正するために用いられる回転行列である。つまり、上記補正処理においては、上記第1姿勢行列 M_1 に補正行列 M_a を掛けることで第1の姿勢が補正される。

30

【0117】

ロール姿勢成分データ77は、撮像画像から算出される入力装置8の第3の姿勢に含まれる姿勢成分のうち、ロール方向に関する姿勢成分（ロール姿勢成分） M_{3r} を示すデータである。また、ヨー姿勢成分データ78は、上記第3の姿勢に含まれる姿勢成分のうち、ヨー方向に関する姿勢成分（ヨー姿勢成分） M_{3y} を示すデータであり、ピッチ姿勢成分データ79は、上記第3の姿勢に含まれる姿勢成分のうち、ピッチ方向に関する姿勢成分（ピッチ姿勢成分） M_{3p} を示すデータである。なお、ここで言うロール方向、ヨー方向、およびピッチ方向とは、入力装置8の撮像方向（Z軸正方向）を基準とした場合の回転方向である。本実施形態においては、各姿勢成分 M_{3r} 、 M_{3y} 、および M_{3p} は、第1姿勢データ68により示される姿勢と同様、 3×3 の行列で表現される。

40

【0118】

第3姿勢データ80は、上記第3の姿勢を示すデータである。本実施形態では、第3の姿勢は、第1の姿勢と同様、 3×3 の行列 M_3 で表現される。後述するステップS6の補正処理では、上記行列 M_3 が第1姿勢行列 M_1 を補正するために用いられる。以下では、第3の姿勢を表す行列 M_3 を「第3姿勢行列 M_3 」と呼ぶ。本実施形態では、入力装置8から操作データとしてマーカ座標データが送信されてくるので、第3姿勢行列 M_3 は、マーカ座標データ65に基づいて算出される。具体的には、第3姿勢行列 M_3 は、上記各姿勢成分 M_{3r} 、 M_{3y} 、および M_{3p} を合成することによって得られる。

【0119】

次に、ゲーム装置3において行われる処理の詳細を、図14～図17を用いて説明する

50

。図14は、ゲーム装置3において実行される処理の流れを示すメインフローチャートである。ゲーム装置3の電源が投入されると、ゲーム装置3のCPU10は、図示しないブートROMに記憶されている起動プログラムを実行し、これによってメインメモリ等の各ユニットが初期化される。そして、光ディスク4に記憶されたゲームプログラムがメインメモリに読み込まれ、CPU10によって当該ゲームプログラムの実行が開始される。図14に示すフローチャートは、以上の処理が完了した後に行われる処理を示すフローチャートである。

【0120】

まず、ステップS1において、CPU10は、ゲームに関する初期化処理を実行する。この初期化処理においては、ゲーム処理に用いられる各種パラメータの値が初期化されたり、仮想のゲーム空間が構築されたり、プレイヤオブジェクトや他のオブジェクトがゲーム空間の初期位置に配置されたりする。以上のステップS1の次にステップS2の処理が実行される。

10

【0121】

ステップS2において、CPU10は初期姿勢設定処理を実行する。具体的には、プレイヤが所定の操作（例えば、Aボタン32dを押下する操作）を行ったことに応じて、入力装置8の初期姿勢として所定の値が設定される。ここでは、基準となる姿勢は、Z軸が鉛直方向と平行となり、かつ、入力装置8の撮像方向がマーカ部6の中央（マーカ6Rおよび6Lの中央）を向く姿勢としているので、プレイヤは、初期姿勢が上記基準となる姿勢となるように入力装置8を把持した状態で、上記所定の操作を行うことが望ましいが、入力装置が静止状態に近く、かつマーカ部を撮像可能であれば、初期姿勢を算出することが可能である。CPU10は、所定の操作が行われると、上記初期姿勢を表す行列を示すデータを第1姿勢データ68としてメインメモリに記憶する。この時点での重力ベクトルである、空間座標系における重力方向（0，-1，0）を、コントローラ座標系で表したベクトル（-Xy，-Yy，-Zy）を示すデータを、重力方向データ69としてメインメモリに記憶する。以上のステップS2の後、ステップS3～S8の処理ループが、ゲームが実行される間繰り返し実行される。なお、1回の当該処理ループは、1フレーム時間（例えば1/60秒）に1回の割合で実行される。

20

【0122】

なお、本実施形態では、初期姿勢設定処理（ステップS2）は、ゲーム開始前（ステップS3～S8の処理ループが実行される前）に1回実行されるのみであるとしたが、他の実施形態においては、ゲーム中における任意のタイミングで初期姿勢設定処理が実行されるようにしてもよい。すなわち、CPU10は、ゲーム中においてプレイヤが上記所定の操作を行ったことに応じて初期姿勢設定処理を実行するようにしてもよい。

30

【0123】

ステップS3において、CPU10は操作データを取得する。すなわち、コントローラ5から送信されてくる操作データが無線コントローラモジュール19を介して受信される。そして、受信された操作データに含まれる角速度データ、加速度データ、マーカ座標データ、操作ボタンデータがそれぞれメインメモリに記憶される。ステップS3の次にステップS4の処理が実行される。

40

【0124】

ステップS4において、CPU10は、メインメモリに記憶されている角速度データ63に基づいて入力装置8の第1の姿勢（第1姿勢行列M1）を算出する。第1の姿勢を角速度から算出する方法はどのような方法であってもよいが、本実施形態においては、第1の姿勢は、前回の入力装置8の第1の姿勢（前回のステップS3～S8の処理ループにおいて算出された姿勢）と、今回の角速度（今回の処理ループにおいて取得された角速度）とを用いて算出される。具体的には、CPU10は、メインメモリに記憶されている角速度データ63および第1姿勢データ68を読み出し、第1姿勢データ68により示される前回の姿勢と角速度データ63により示される今回の角速度とを取得する。そして、前回の姿勢を今回の角速度で単位時間分だけ回転させた姿勢を、新たな第1の姿勢とする。ス

50

テップ S 4 で算出された姿勢（ 3×3 の行列）を示すデータが、第 1 姿勢データ 6 8 としてメインメモリに新たに記憶される。上記ステップ S 4 の次にステップ S 5 の処理が実行される。

【0125】

ステップ S 5 において、CPU 10 は加速度に基づく補正処理を実行する。加速度に基づく補正処理は、加速度データを用いて入力装置 8 の第 1 の姿勢を補正する処理である。以下、図 1 5 を参照して、加速度に基づく補正処理の詳細を説明する。

【0126】

図 1 5 は、図 1 4 に示す加速度に基づく補正処理（ステップ S 5）の流れを示すフローチャートである。加速度に基づく補正処理においては、まずステップ S 1 1 において、CPU 10 は重力ベクトル V_g を算出する。重力ベクトル V_g は、前回のステップ S 3 ~ S 8 の処理ループで算出された重力ベクトルと、今回のステップ S 3 ~ S 8 の処理ループで取得された角速度とに基づいて算出される。具体的には、CPU 10 は、メインメモリに記憶されている角速度データ 6 3 および重力方向データ 6 9 を読み出し、角速度データ 6 3 により示される今回の角速度と重力方向データ 6 9 により示される前回の重力ベクトルとを取得する。そして、前回の重力ベクトルを、今回の角速度の回転方向とは逆方向に単位時間分だけ回転させることによって新たな重力ベクトル V_g を算出する。新たに算出された重力ベクトル V_g を示すデータが、新たな重力方向データ 6 9 としてメインメモリに記憶される。以上のステップ S 1 1 の次にステップ S 1 2 の処理が実行される。

【0127】

なお、本実施形態では、入力装置 8 の姿勢は上記空間座標系で表現されるのに対して、重力ベクトルは上記コントローラ座標系で表現されるので、空間座標系における入力装置 8 の姿勢が変化することによって空間座標系における重力ベクトルの方向が変化してしまう。つまり、今回の処理ループにおける上記ステップ S 4 において新たな姿勢が算出されて姿勢が変化すると、姿勢の変化に応じて（実際には変化していない）重力ベクトルも変化してしまう。上記ステップ S 1 1 の処理は、重力ベクトルを正しく表すために、入力装置 8 の姿勢の変化による重力ベクトルの変化をキャンセルするための処理である。

【0128】

ステップ S 1 2 においては、CPU 10 は、静的補正処理を実行する。入力装置 8 が主に静止している（または静止状態に近い）場合に入力装置 8 の第 1 の姿勢を補正する補正処理である。以下、図 1 6 を参照して、静的補正処理の詳細を説明する。

【0129】

図 1 6 は、図 1 5 に示す静的補正処理（ステップ S 1 2）の流れを示すフローチャートである。静的補正処理においては、まずステップ S 2 1 において、CPU 10 は、加速度センサ 3 7 によって検出された加速度の大きさ L を算出する。すなわち、メインメモリに記憶されている加速度データ 6 4 を読み出し、当該加速度データ 6 4 により示される検出加速度ベクトル V_a について大きさ L を算出する。算出された大きさ L を示すデータは加速度大きさデータ 7 0 としてメインメモリに記憶される。ステップ S 2 1 の次にステップ S 2 2 の処理が実行される。

【0130】

ステップ S 2 2 において、CPU 10 は、静的補正処理において入力装置 8 の姿勢を補正する度合を示す上記第 1 補正度 A を算出する。第 1 補正度 A は、ステップ S 2 1 で算出された検出加速度ベクトル V_a の大きさ L に基づいて算出される。具体的には、CPU 10 は、メインメモリに記憶されている加速度大きさデータ 7 0 を読み出す。そして、当該加速度大きさデータ 7 0 により示される大きさ L を用いて、次の式（2）に従って第 1 補正度 A を算出する。

$$\begin{aligned} 0 \leq |L - 1| \leq R \text{ のとき、} & A = 1 - (|L - 1| / R) \\ R < |L - 1| \text{ のとき、} & A = 0 \end{aligned} \quad \dots (2)$$

上式（2）において、定数 R は、予め定められており、例えば $R = 0.4$ に設定される。上式（2）は、検出加速度ベクトル V_a の大きさ L が重力加速度の大きさ（ $= 1$ ）に近い

10

20

30

40

50

ほど大きくなるように、第1補正度Aを算出する式である。つまり、上式(2)によって、検出加速度ベクトルV aの大きさLが重力加速度の大きさに近いほど、重力ベクトルの補正量が大きくなる。なお、他の実施形態においては、上記大きさLが1に近いほど大きい重みが付されるように、上式(2)で算出されるAをさらに2乗した値を第1補正度として用いてもよい。上式(2)で算出された第1補正度Aを示すデータは、第1補正度データ72としてメインメモリに記憶される。なお、ステップS22で算出された第1補正度Aは、最終的な値ではない演算中の値であって、以降のステップS23～S26で値が修正されることによって、最終的な第1補正度Aの値が得られる。ステップS22の次にステップS23の処理が実行される。

【0131】

10

ここで、入力装置8が動かされている状態では、重力加速度以外に、入力装置8が動かされることによって生じる慣性による加速度が加速度センサ37によって検出される。そのため、入力装置8が動かされている状態では、検出加速度ベクトルV aの大きさLが“1”とは異なる値となり、入力装置8が激しく動かされている場合ほど上記差が大きくなる傾向がある。また、入力装置8が激しく動かされている場合には、検出加速度ベクトルV aに重力加速度以外の成分(上記慣性による加速度の成分)が多く含まれているので、検出加速度ベクトルV aは、重力方向を示す値としては信頼できないものと推測される。つまり、上記差が大きいほど、検出加速度ベクトルV aは信頼できないものと考えられる。

【0132】

20

そこで、本実施形態では、上記ステップS22により、検出加速度ベクトルV aの大きさLが重力加速度の大きさ(=1)に近いほど、重力ベクトルの補正量が大きくなるようにしている。また、検出加速度ベクトルの大きさLと重力加速度の大きさとの差が所定の基準(所定値R)よりも小さい場合にのみ補正が行われるようにしている。これによれば、検出加速度ベクトルV aが信頼できる場合ほど重力ベクトルが検出加速度ベクトルV aに近い値となり、また、検出加速度ベクトルV aの値が信頼できない場合には補正が行われない。このように、本実施形態によれば、検出加速度ベクトルV aの信頼度に応じて補正量が決まるので、重力ベクトルV gの補正量を当該信頼度に応じて適切に決定することができる。これによって、ゲーム装置3は、重力方向を正確に算出することができ、入力装置8の姿勢をより正確に算出することができる。

30

【0133】

ステップS23において、CPU10は、入力装置8の角速度に応じて第1補正度Aの値を修正する。具体的には、CPU10は、メインメモリに記憶されている角速度データ63および第1補正度データ72を読み出し、角速度データ63により示される角速度と第1補正度データ72により示される第1補正度Aとを取得する。そして、上記角速度を用いて、次の式(3)に従って修正後の第1補正度Aを算出する。

$$0 \leq |\omega| \leq T1 \text{ のとき、 } A = A' \times \{1 - (\omega / T1)\} \\ T1 < |\omega| \text{ のとき、 } A = 0 \quad \dots (3)$$

上式(3)において、定数T1は、予め定められており、例えば、 $T1 = 60 [\text{deg} / \text{sec}]$ に設定される。また、変数 ω は、角速度の大きさ、すなわち、角速度ベクトルV ω の大きさである。上式(3)は、角速度が大きいほど第1補正度Aが小さくなるように、第1補正度Aを修正するものである。つまり、上式(3)によって、角速度が大きいほど、重力ベクトルの補正量が小さくなる。なお、他の実施形態においては、角速度が小さいほど大きい重みが付されるように、上式(3)における“ $\{1 - (\omega / T1)\}$ ”を、“ $\{1 - (\omega / T1)\}^2$ ”としてもよい。上式(3)で修正された第1補正度Aを示すデータは、第1補正度データ72としてメインメモリに記憶される。ステップS23の次にステップS24の処理が実行される。

40

【0134】

ここで、入力装置8の角速度が大きい場合とは、入力装置8が激しく動かされている場合であると推測され、この場合、検出加速度ベクトルV aの値は、重力方向を示す値とし

50

ては信頼できないものと推測される。逆に、角速度が小さい場合とは、入力装置 8 があまり動かされていない場合であると推測され、検出加速度ベクトル V_a の値は信頼できると推測される。したがって、上記ステップ S 2 3 によって、上記ステップ S 2 2 と同様、検出加速度ベクトル V_a が信頼できる場合（角速度が小さい場合）ほど、第 1 補正度 A の値が大きくなるように設定することができる。その結果、上記ステップ S 2 2 の場合と同様、検出加速度ベクトル V_a の信頼度に応じて補正量を適切に決定することができるので、重力方向および入力装置 8 の姿勢をより正確に算出することができる。

【0135】

また、上記ステップ S 2 2 では、入力装置 8 が動かされている度合（検出加速度ベクトル V_a の信頼度）を、入力装置 8 の加速度に基づいて判断しているのに対して、上記ステップ S 2 3 では、入力装置 8 の角速度に基づいて判断している。このように、ステップ S 2 2 とは異なる変数で上記信頼度を判断するので、ステップ S 2 2 で上記信頼度を正しく判断できない場合（例えば、入力装置 8 を動かしている場合において、検出加速度ベクトル V_a の大きさが重力加速度の大きさと偶然等しくなる場合）でも、ステップ S 2 3 によって上記信頼度を正しく判断することができる。

【0136】

ステップ S 2 4 において、CPU 10 は、上記運動加速度ベクトル V_A を算出する。上述のように、運動加速度ベクトル V_A は、検出加速度ベクトル V_a および重力ベクトル V_g に基づいて算出される。具体的には、CPU 10 は、メインメモリに記憶されている加速度データ 6 4 および重力方向データ 6 9 を読み出し、加速度データ 6 4 により示される検出加速度ベクトル V_a から、重力方向データ 6 9 により示される重力ベクトル V_g を引くことによって、運動加速度ベクトル V_A を算出する。算出された運動加速度ベクトル V_A を示すデータは、運動加速度データ 7 4 としてメインメモリに記憶される。なお、本実施形態では、入力装置 8 の加加速度を算出するために、前回の処理ループで算出された運動加速度ベクトル V_A を示すデータは、加速度履歴データとしてメインメモリに記憶される。ステップ S 2 4 の次にステップ S 2 5 の処理が実行される。

【0137】

ステップ S 2 5 において、CPU 10 は、入力装置 8 の加加速度 J を算出する。加加速度 J は、今回の処理ループで算出された運動加速度ベクトルと、前回の処理ループで算出された運動加速度ベクトルとの差分として算出される。具体的には、CPU 10 は、メインメモリに記憶されている運動加速度データ 7 4 および上記加速度履歴データを読み出し、運動加速度データ 7 4 により示される運動加速度ベクトルから、加速度履歴データにより示される運動加速度ベクトルを引いたベクトルの大きさを加加速度（の大きさ） J として算出する。算出された加加速度を示すデータは、加加速度データ 7 1 としてメインメモリに記憶される。なお、他の実施形態においては、CPU 10 は、運動加速度ベクトルに代えて、加速度センサ 3 7 が検出した加速度ベクトルから加加速度を算出するようにしてもよい。すなわち、今回のステップ S 3 - S 8 の処理ループにおいて取得された検出加速度ベクトルから、前回のステップ S 3 - S 8 の処理ループにおいて取得された検出加速度ベクトルを引いたベクトルの大きさを加加速度（の大きさ） J として算出してもよい。ステップ S 2 5 の次にステップ S 2 6 の処理が実行される。

【0138】

ステップ S 2 6 において、CPU 10 は、入力装置 8 の加加速度 J に応じて第 1 補正度 A の値を修正する。具体的には、CPU 10 は、メインメモリに記憶されている加加速度データ 7 1 および第 1 補正度データ 7 2 を読み出し、加加速度データ 7 1 により示される加加速度 J と第 1 補正度データ 7 2 により示される第 1 補正度 A とを取得する。そして、上記加加速度 J を用いて、次の式（4）に従って修正後の第 1 補正度 A を算出する。

$0 \leq J \leq U1$ のとき、 $A = A' \times \{1 - (J/U1)\} \times C1$

$U1 < J$ のとき、 $A = 0$ …（4）

上式（4）において、定数 $U1$ は、予め定められており、例えば、 $U1 = 40$ [G/sec] に設定される。また、定数 $C1$ は、補正量を調整するために適宜設定され、例えば、

10

20

30

40

50

C1 = 0.03に設定される。上式(4)は、加加速度Jが大きいほど第1補正度Aが小さくなるように、第1補正度Aを修正するものである。つまり、上式(4)によって、加加速度Jが大きいほど、重力ベクトルの補正量が小さくなる。なお、他の実施形態においては、加加速度Jが小さいほど大きい重みが付されるように、上式(4)における“ $\{1 - (J/U1)\}$ ”を、“ $\{1 - (J/U1)\}^2$ ”としてもよい。上式(4)で修正された第1補正度Aを示すデータは、第1補正度データ72としてメインメモリに記憶される。上記ステップS26によって、最終的な第1補正度Aが算出されたことになる。ステップS26の次にステップS27の処理が実行される。

【0139】

ここで、入力装置8の加加速度Jが大きい場合とは、入力装置8が激しく動かされている場合であると推測され、上述したように、検出加速度ベクトルVaの値が信頼できないものと推測される。逆に、上記加加速度Jが小さい場合とは、入力装置8があまり動かされていない場合であると推測され、検出加速度ベクトルVaの値が信頼できるものと推測される。したがって、上記ステップS26によって、上記ステップS22およびS23と同様、検出加速度ベクトルVaが信頼できる場合（加加速度Jが小さい場合）ほど、第1補正度Aの値を大きく設定することができる。その結果、上記ステップS22の場合と同様、検出加速度ベクトルVaの信頼度に応じて補正量を適切に決定することができるので、重力方向および入力装置8の姿勢をより正確に算出することができる。

【0140】

また、上記ステップS26では、入力装置8が動かされている度合（検出加速度ベクトルVaの信頼度）を、上記ステップS22およびS23とは異なる変数である加加速度Jを用いて判断している。そのため、ステップS22およびS23で上記信頼度を正しく判断できずに第1補正度Aが大きい値のままになる場合（例えば、入力装置8が回転せずに平行移動している場合で、検出加速度ベクトルVaの大きさが重力加速度の大きさと偶然等しくなる場合）であっても、ステップS26の処理によって第1補正度Aを小さく変更することができるので、上記信頼度を正しく判断することができる。

【0141】

以上のように、本実施形態においては、ステップS22、S23、S26によって、検出加速度ベクトルVaの信頼度を、入力装置8の状態を表す3つの異なる変数を用いて判断している。これによって、検出加速度ベクトルVaの信頼度をより正確に判断することができ、信頼度に応じて補正量を適切に決定することができる。そのため、ゲーム装置3は、重力方向および入力装置8の姿勢をより正確に算出することができる。なお、他の実施形態においては、CPU10は、ステップS22、S23、S26の全ての処理を実行する必要はなく、S22、S23、S26の処理のうち1つまたは2つの処理のみを実行するようにしてもよい。

【0142】

また、上記ステップS22、S23、またはS26の結果、第1補正度Aが“0”となった場合、CPU10は、それ以降の処理をスキップして、静的補正処理を終了するようにしてもよい。上記の場合、実質的には重力ベクトルの補正が行われないので、それ以降の処理を省略することにより、処理を短縮することができるからである。

【0143】

ステップS27において、CPU10は、検出加速度ベクトルVaに近づけるように重力ベクトルVgを補正する（図9参照）。本実施形態では、補正前の重力ベクトルVgを、上記第1補正度Aの大きさに応じた割合で検出加速度ベクトルVaに近づけることによって、補正後の重力ベクトルVg'が算出される。具体的には、CPU10は、メインメモリに記憶されている加速度データ64、重力方向データ69、および第1補正度データ72を読み出し、加速度データ64により示される検出加速度ベクトルVaと、重力方向データ69により示される（補正前の）重力ベクトルVgと、第1補正度データ72により示される第1補正度データAとを取得する。そして、補正前の重力ベクトルVgと検出加速度ベクトルVaとのなす角度を $\theta1$ [deg]とすると、検出加速度ベクトルVaの

方向へ“ $A \cdot \theta 1$ [deg]”だけ補正前の重力ベクトル V_g を回転させることによって、補正後の重力ベクトル V_g' を算出する。算出された重力ベクトル V_g' を示すデータは、新たな重力方向データ69としてメインメモリに記憶される。なお、補正後の重力ベクトル V_g' の算出方法はどのような方法であってもよい。例えば、他の実施形態においては、補正前の重力ベクトル V_g の終点から検出加速度ベクトル V_a の終点までを結ぶ線分を A ： $(1-A)$ に内分する点を終点とするベクトルを算出し、算出されたベクトルを正規化することによって、補正後の重力ベクトル V_g' を得るようにしてもよい。以上のステップS21～S27によって、静的補正処理において重力ベクトルが補正されたこととなる。上記ステップS27の後、CPU10は静的補正処理を終了する。

【0144】

10

図15の説明に戻り、ステップS12の次のステップS13において、CPU10は、動的補正処理を実行する。動的補正処理は、入力装置8が主に動いている場合に入力装置8の姿勢を補正する補正処理である。

【0145】

上述のように、動的補正処理においては、CPU10は、入力装置8が等角速度で回転運動を行うものと推定し、等角速度の回転運動を行う場合における角速度と運動加速度との関係を用いて、重力ベクトル V_g を補正する。ここで、回転運動を行う入力装置8の運動加速度ベクトルは、一般的に、次の式(5)によって表される。

【数2】

20

$$VA = \frac{d}{dt}(V\omega \times Vr) = \dot{V}\omega \times Vr + V\omega \times (V\omega \times Vr) \quad \dots (5)$$

上式(5)において、ベクトル $V\omega$ は、ジャイロセンサ55および56が検出した角速度を表す角速度ベクトルであり、ベクトル Vr は、入力装置8の位置、より厳密には加速度センサの位置37の位置から回転の中心位置を指すベクトル(位置ベクトル)である。ここでは、入力装置8は等角速度の回転運動を行うと推定するので、上式(5)の右辺の第1項が“0”となり、さらに上式(5)を整理すると、次の式(6)が得られる。

$$VA = (-V\omega^2 [I] + V\omega V\omega^T) Vr \quad \dots (6)$$

上式(6)において、行列 $[I]$ は 3×3 の単位行列であり、行列 $V\omega^T$ はベクトル $V\omega$ を 1×3 の行列であるとみなした場合における行列 $V\omega$ の転置行列である。上式(6)の両辺それぞれに対してベクトル $V\omega$ との内積を算出する演算を行うと、右辺が“0”となるので、下記の式(7)が成り立つ。

30

$$V\omega \cdot VA = 0 \quad \dots (7)$$

上式(7)は、角速度ベクトル $V\omega$ と運動加速度ベクトル VA とが直交することを表している。つまり、入力装置8が等角速度の回転運動を行う場合、角速度ベクトル $V\omega$ と運動加速度ベクトル VA とは直交する関係が成り立つ。なお、以下では、入力装置8が回転運動を行う場合において運動加速度ベクトルと直交するベクトルを、「直交ベクトル」と呼ぶことがある。本実施形態では、角速度ベクトル $V\omega$ が直交ベクトルである。

【0146】

40

上記関係を用いることによって、重力ベクトル V_g を補正することができる。すなわち、運動加速度ベクトル VA は重力ベクトル V_g から算出される(上記ステップS24参照)ので、ジャイロセンサ55および56が検出した角速度ベクトル $V\omega$ に対して運動加速度ベクトル VA が上式(7)を満たすように、重力ベクトル V_g を補正すればよい。なお、本実施形態における動的補正処理では、入力装置8が正確に等角速度の回転運動を行っていない場合もあり得ることや、補正によって重力ベクトル V_g が急激に変化することを防止すること等を考慮して、上記関係を満たす重力ベクトルを1回の補正処理で得るのではなく、1回の補正処理では、上記関係を満たす重力ベクトルへと近づける補正を行う。すなわち、運動加速度ベクトル VA が角速度ベクトル $V\omega$ に対して垂直に近づくように、重力ベクトル V_g を補正する。

【0147】

50

補正前の重力ベクトル V_g を、上記関係を満たす重力ベクトルへと近づけるためには、直交ベクトル W と運動加速度ベクトル V_A との内積が小さくなるように重力ベクトル V_g を補正すればよい。すなわち、関数 $\phi = V_\omega \cdot V_A = V_\omega \cdot (V_a - V_g)$ を、 $|V_g|^2 = 1$ の条件下で最小にする重力ベクトル V_g を得ればよい。このような重力ベクトル V_g を得る方法はあるどのような方法であってもよいが、本実施形態における動的補正処理では、重力補正ベクトル ΔV_g を用いた方法によって処理を行う。重力補正ベクトル ΔV_g は、予め定められた長さであり、その方向へ重力ベクトル V_g を回転させるベクトルである。CPU 10 は、重力補正ベクトル ΔV_g の方向、すなわち、重力ベクトル V_g をどの向きへ回転させるかを決定する。つまり、本実施形態では、重力ベクトル V_g を所定角度だけ（重力補正ベクトル ΔV_g の長さだけ）回転させたときに、回転後の重力ベクトルから算出される運動加速度ベクトルが、上記関係を満たす運動加速度へ最も近づく方向へ、重力ベクトル V_g を回転させる補正を行う。

10

【0148】

以下、図17を参照して、動的補正処理の詳細を説明する。図17は、図15に示す動的補正処理（ステップS13）の流れを示すフローチャートである。動的補正処理においては、まずステップS31において、CPU 10 は、重力ベクトル V_g を補正するために用いられる上記重力補正ベクトル ΔV_g を算出する。重力補正ベクトル ΔV_g は、角速度ベクトル V_ω および運動加速度ベクトル V_A に基づいて算出される。以下、図18を参照して、重力補正ベクトル ΔV_g の算出方法について説明する。

【0149】

20

図18および図19は、動的補正処理における重力補正ベクトル ΔV_g の算出方法を示す図である。図18においては、運動加速度ベクトル V_A と角速度ベクトル V_ω とのなす角 θ_2 は鋭角である。当該角 θ_2 が鋭角である場合、上記の関係を満たすべく、運動加速度ベクトル V_A を角速度ベクトル V_ω を直交させる（直交状態に近づける）ためには、運動加速度ベクトル V_A を角速度ベクトル V_ω の逆方向へ回転させる（角速度ベクトル V_ω から遠ざける）ように補正すればよい。つまり、運動加速度ベクトル V_A の終端を角速度ベクトル V_ω の逆方向（ベクトル V_1 の方向）へ回転させればよい。したがって、重力ベクトル V_g については、その終端を角速度ベクトル V_ω の同方向へ回転させればよい。以上より、重力補正ベクトル ΔV_g は、角速度ベクトル V_ω と同じ方向を向くベクトル $-V_1$ を重力ベクトル V_g と垂直な平面 P に投影したベクトルとして算出することができる。一方、図19に示すように、運動加速度ベクトル V_A と角速度ベクトル V_ω とのなす角 θ_2 が鈍角である場合は、角 θ_2 が鋭角である場合とは逆に、運動加速度ベクトル V_A の終端を角速度ベクトル V_ω と同方向へ回転させればよい。したがって、重力補正ベクトル ΔV_g は、角速度ベクトル V_ω の逆方向を向くベクトル $-V_1$ を上記平面 P に投影したベクトルとして算出することができる。

30

【0150】

上記ステップS31の具体的な処理としては、CPU 10 は、まず、角速度ベクトル V_ω および運動加速度ベクトル V_A に基づいて上記ベクトル V_1 を算出する。ベクトル V_1 は、運動加速度ベクトル V_A と角速度ベクトル V_ω とのなす角が鋭角である場合に角速度ベクトル V_ω と逆方向を向き、当該角が鋭角である場合に角速度ベクトル V_ω と同方向を向く。すなわち、CPU 10 は、メインメモリに記憶されている角速度データ63および運動加速度データ74を読み出し、角速度データ63により示される角速度ベクトル V_ω と運動加速度データ74により示される運動加速度ベクトル V_A とを取得する。そして、角速度ベクトル V_ω および運動加速度ベクトル V_A を用いて、次の式(8)に従ってベクトル V_1 を算出する。

40

$$V_1 = (V_\omega \cdot V_A) V_\omega \quad \cdots (8)$$

さらに、CPU 10 は、上式(8)で算出されたベクトル V_1 を上記平面 P （図18参照）に投影したベクトルを得るべく、次の式(9)に従って重力補正ベクトル ΔV_g を算出する。

$$\Delta V_g = V_1 - (V_g \cdot V_1) V_g \quad \cdots (9)$$

50

上式（９）は、上記Ｖ１から重力ベクトルＶｇと平行な成分を除いたベクトルを算出するものである。ＣＰＵ１０は、さらに、上式（９）で得られた重力補正ベクトル $\Delta V g$ を正規化した後、予め定められた定数を掛ける。この定数は、補正量を表すものであり、例えば０．０３に設定される。これによって、所定の長さの重力補正ベクトル $\Delta V g$ を得ることができる。以上によって算出された重力補正ベクトル $\Delta V g$ を示すデータは、重力補正ベクトルデータ７５としてメインメモリに記憶される。ステップＳ３１の次にステップＳ３２の処理が実行される。

【０１５１】

なお、本実施形態においては、角速度ベクトル $V \omega$ の方向に基づいて上記重力補正ベクトル $\Delta V g$ の方向を決定した。ここで、他の実施形態においては、ＣＰＵ１０は、上式（７）の関係を満たす重力ベクトルを算出し、算出された重力ベクトルへ補正前の重力ベクトルを近づける方向として、上記重力補正ベクトル $\Delta V g$ の方向を決定してもよい。

【０１５２】

ステップＳ３２において、ＣＰＵ１０は、動的補正処理において入力装置８の姿勢を補正する度合を示す上記第２補正度Ｂを算出する。第２補正度Ｂは、入力装置８の角速度に応じた大きさになるように算出される。具体的には、ＣＰＵ１０は、メインメモリに記憶されている角速度データ６３を読み出す。そして、角速度データ６３により示される角速度を用いて、次の式（１０）に従って第２補正度Ｂを算出する。

$0 \leq |\omega| \leq T2$ のとき、 $B = \omega / T2$

$T2 < |\omega|$ のとき、 $B = 1 \quad \dots (10)$

上式（１０）において、定数Ｔ２は、予め定められており、例えば、 $T2 = 10 [deg / sec]$ に設定される。上式（１０）は、角速度 ω の大きさ（角速度ベクトル $V \omega$ の大きさ）が小さいほど第２補正度Ｂが小さくなるように、第２補正度Ｂを算出するものである。つまり、上式（１０）によって、角速度 ω が小さくほど、重力ベクトルの補正量が小さくなる。なお、他の実施形態においては、角速度 ω が大きいほど大きい重みが付されるように、上式（１０）における“ $\omega / T2$ ”を、“ $(\omega / T2)^2$ ”としてもよい。上式（１０）で算出された第２補正度Ｂを示すデータは、第２補正度データ７３としてメインメモリに記憶される。ステップＳ３２の次にステップＳ３３の処理が実行される。

【０１５３】

ここで、入力装置８の角速度が小さい場合、入力装置８が回転運動を行っていない可能性が高いと推測される。動的補正処理における補正は、入力装置８が（等角速度の）回転運動を行っていることを前提として行われるが、入力装置８の角速度が小さい場合には、この前提が成り立たない可能性が高いと考えられる。そのため、本実施形態では、入力装置８の角速度が小さい場合には、動的補正処理における補正が正確でないおそれがあることを考慮して、補正量を小さくしている。これによれば、入力装置８が回転運動を行っていないことから補正を正しく行うことができない場合に、動的補正処理によって不正確な補正が行われることを防止することができる。その結果、動的補正処理において重力ベクトルの補正を正しく行うことができる。

【０１５４】

ステップＳ３３において、ＣＰＵ１０は、入力装置８の加加速度Ｊに応じて第２補正度Ｂの値を修正する。具体的には、ＣＰＵ１０は、メインメモリに記憶されている加加速度データ７１および第２補正度データ７３を読み出し、加加速度データ７１により示される加加速度Ｊと第２補正度データ７３により示される第２補正度Ｂとを取得する。そして、上記加加速度Ｊを用いて、次の式（１１）に従って修正後の第２補正度Ｂを算出する。

$0 \leq J \leq U2$ のとき、 $B = B' \times \{1 - (J / U1)\}$

$U2 < J$ のとき、 $B = 0 \quad \dots (11)$

上式（１１）において、定数Ｕ２は、予め定められており、例えば、 $U2 = 40 [G / sec]$ に設定される。上式（１１）は、加加速度Ｊが大きいほど第２補正度Ｂが小さくなるように、第２補正度Ｂを修正するものである。つまり、上式（１１）によって、加加速度Ｊが大きいほど、重力ベクトルの補正量が小さくなる。なお、他の実施形態においては

、加加速度 J が小さいほど大きい重みが付されるように、上式 (11) における “ $\{1 - (J/U2)\}$ ” を、“ $\{1 - (J/U2)\}^2$ ” としてもよい。上式 (11) で修正された第2補正度 B を示すデータは、第2補正度データ73としてメインメモリに記憶される。上記ステップS33によって、最終的な第2補正度 B が算出されたことになる。ステップS33の次にステップS34の処理が実行される。

【0155】

ここで、入力装置8の加加速度 J が大きくなるのは、入力装置8を動かし始めた時点、(動かしている状態から) 停止させた時点、または入力装置8を往復移動させている場合における折り返しの時点等である。このような時点では、入力装置8は回転運動を行っていないと推測される。したがって、入力装置8の角速度が小さい場合と同様、入力装置8の加加速度 J が大きくなる場合には、「入力装置8が(等角速度の) 回転運動を行っている」という前提が成り立たない可能性が高いと考えられ、この場合には動的補正処理における補正が不正確になるおそれがある。そのため、本実施形態では、入力装置8の加加速度 J が大きい場合には、動的補正処理における補正が正確でないおそれがあることを考慮して、補正量を小さくしている。これによれば、上記ステップS32の場合と同様、動的補正処理によって不正確な補正が行われることを防止することができる。その結果、動的補正処理において重力ベクトルの補正を正しく行うことができる。

【0156】

また、上記ステップS33では、入力装置8が回転運動を行っているか否かを、上記ステップS32とは異なる変数である加加速度 J を用いて判断している。そのため、入力装置8が回転運動を行っているか否かをステップS32では正しく判断できずに第2補正度 B が大きい値のままになる場合であっても、ステップS26の処理によって第1補正度 A を小さく変更することができるので、上記信頼度を正しく判断することができる。

【0157】

ステップS34において、CPU10は、重力ベクトル V_g を重力補正ベクトル ΔV_g で補正する。具体的には、CPU10は、メインメモリに記憶されている重力方向データ69および重力補正ベクトルデータ75を読み出し、重力方向データ69により示される重力ベクトル V_g に、重力補正ベクトルデータ75により示される重力補正ベクトル ΔV_g を加算する。さらに、加算結果のベクトルを正規化することによって、補正後の重力ベクトル V_g を算出する。補正後の重力ベクトル V_g を示すデータは、重力方向データ69としてメインメモリに記憶される。これによって、動的補正処理において重力ベクトル V_g の補正が行われたこととなる。上記ステップS34の後、CPU10は動的補正処理を終了する。

【0158】

上記動的補正処理によれば、CPU10は、入力装置8が等角速度の回転運動を行うものとして、「運動加速度ベクトル V_A が角速度ベクトル V_ω と垂直になる」という関係(上式(7))を予め定義しておく。そして、補正前の重力ベクトル V_g を、上記関係を満たす重力ベクトルに近づける補正を行う。これによれば、ゲーム装置3は、入力装置8が動いている場合であっても重力ベクトルを補正することができ、重力方向および入力装置8の姿勢を正確に算出することができる。

【0159】

また、本実施形態におけるカヌーのゲームでは、プレイヤはパドルで水を掻くように入力装置8を動かすので、入力装置8が回転運動を行う期間があると想定される。その一方、入力装置8の振り始めや振り終わりの期間や、入力装置8が正確に回転運動を行っておらずに往復運動に近い運動を行っている期間があるおそれがある。そのため、本実施形態では、入力装置8の角速度および加加速度を用いて、入力装置8が回転運動を行っているか否かを判断し、回転運動を行っている可能性が高い場合には動的補正処理における補正量を大きくし、回転運動を行っている可能性が低い場合には補正量を小さくしている(ステップS32およびS33)。したがって、本実施形態によれば、入力装置8が回転運動を行う期間には重力方向が正しく補正される一方、振り始めや振り終わりの期間や、往復

運動を行っている期間には補正が行われないので、不正確な補正が行われることがなく、重力方向を正確に算出することができる。

【0160】

〔動的補正処理の変形例〕

なお、上記実施形態においては、入力装置 8 が等角速度で回転運動を行うものと推定して、運動加速度と角速度との関係（上式（7））を定義した。ここで、他の実施形態においては、入力装置 8 が、等角速度でない場合も含む回転運動を行うものと推定して、運動加速度と角速度との関係を定義することも可能である。以下、入力装置 8 が（等角速度でない場合も含む）回転運動を行うものと推定する場合の変形例について説明する。

【0161】

まず、入力装置 8 が等角速度でない回転運動を行う場合における、運動加速度と角速度との関係について説明する。入力装置 8 が等角速度でない回転運動を行う場合においても、上式（5）は成り立つ。この場合、式（5）の右辺の第 1 項が“0”とならないので、右辺の第 1 項を削除せずに式（5）を整理すると、次の式（12）が得られる。

【数 3】

$$\mathbf{V}\mathbf{A} = \left(\left[\dot{\mathbf{V}}\boldsymbol{\omega}^* \right] - \mathbf{V}\boldsymbol{\omega}^2[\mathbf{I}] + \mathbf{V}\boldsymbol{\omega}\mathbf{V}\boldsymbol{\omega}^T \right) \mathbf{V}\mathbf{r} \quad \dots (12)$$

上式（12）は、上式（6）と比べて、右辺の括弧内の第 1 項が追加されている点で異なる。なお、当該第 1 項は、外積の行列表現を表し、演算結果が外積と同様になる行列（反対称行列）を表すものとする。上述のように、入力装置 8 を動かす運動は回転運動であると見なすことができるので、入力装置 8 の位置から回転中心を向く位置ベクトル $\mathbf{V}\mathbf{r}$ と角速度ベクトル $\mathbf{V}\boldsymbol{\omega}$ とが直交することを前提とすれば、上式（12）において右辺の括弧内の第 3 項に位置ベクトル $\mathbf{V}\mathbf{r}$ を掛けたものは“0”になる。さらに、上式（12）を整理して両辺にベクトル $\mathbf{V}\boldsymbol{\omega}$ との内積を算出する演算を行うと、下記の式（13）が成り立つ。

【数 4】

$$\mathbf{V}\boldsymbol{\omega}^T \left(\left[\dot{\mathbf{V}}\boldsymbol{\omega}^* \right] - \mathbf{V}\boldsymbol{\omega}^2[\mathbf{I}] \right)^{-1} \mathbf{V}\mathbf{A} = \mathbf{V}\boldsymbol{\omega} \cdot \mathbf{V}\mathbf{r} = 0 \quad \dots (13)$$

ここで、下記の式（14）のようにベクトル $\mathbf{W}$ を定義する。

【数 5】

$$\mathbf{W} = \left\{ \left(\left[\dot{\mathbf{V}}\boldsymbol{\omega}^* \right] - \mathbf{V}\boldsymbol{\omega}^2[\mathbf{I}] \right)^{-1} \right\}^T \mathbf{V}\boldsymbol{\omega} \quad \dots (14)$$

上式（14）のようにベクトル $\mathbf{W}$ を定義すると、上式（13）の右辺は、ベクトル $\mathbf{W}$ とベクトル $\mathbf{V}\mathbf{A}$ との内積として表すことができるので、上式（13）は下記の式（15）のように表すことができる。

$$\mathbf{W} \cdot \mathbf{V}\mathbf{A} = 0 \quad \dots (15)$$

上式（15）は、角速度ベクトル $\mathbf{V}\boldsymbol{\omega}$ とベクトル $\mathbf{W}$ とが直交することを表している。つまり、入力装置 8 が回転運動を行う場合、運動加速度ベクトル $\mathbf{V}\mathbf{A}$ とベクトル $\mathbf{W}$ とは直交する関係が成り立ち、ベクトル $\mathbf{W}$ が上記直交ベクトルとなる。本変形例では、CPU 10 は、上式（7）に代えて、上式（15）で表される関係に基づいて、重力ベクトルを補正する。

【0162】

以下、図 20 を参照して、上記変形例における動的補正処理の詳細について説明する。図 20 は、図 15 に示す動的補正処理（ステップ S13）の変形例を示すフローチャートである。なお、図 20 において、図 17 と同じ処理ステップには図 17 と同じステップ番号を付す。

【0163】

10

20

30

40

50

本変形例における動的補正処理においては、まずステップS41において、CPU10は入力装置8の角加速度ベクトルを算出する。角加速度ベクトルは、例えば、今回の処理ループで算出された角速度ベクトル $V\omega$ と、前回の処理ループで算出された角速度ベクトル $V\omega$ との差分のベクトルとして算出される。具体的には、CPU10は、メインメモリに記憶されている角速度データ63および角速度履歴データを読み出し、角速度データ63により示される角速度ベクトルから、角速度履歴データにより示される角速度ベクトルを引いたベクトルを角加速度ベクトルとして算出する。算出された角加速度ベクトルを示すデータはメインメモリに記憶される。ステップS41の次にステップS42の処理が実行される。

【0164】

10

ステップS42において、CPU10は上記直交ベクトル W を算出する。直交ベクトル W は、上式(14)に従って、角速度ベクトル $V\omega$ と角加速度ベクトルとから算出することができる。具体的には、CPU10は、メインメモリに記憶されている角速度データ63および角加速度ベクトルを示すデータを読み出し、角速度データ63により示される角速度ベクトルと当該角加速度ベクトルを用いて、上式(14)に従って直交ベクトル W を算出する。算出された直交ベクトル W を示すデータは、直交ベクトルデータとしてメインメモリに記憶される。ステップS42の次にステップS43の処理が実行される。

【0165】

ステップS43において、CPU10は重力補正ベクトル ΔVg を算出する。重力補正ベクトル ΔVg は、上記ステップS31では角速度ベクトル $V\omega$ と運動加速度ベクトル VA とに基づいて算出されたが、本変形例では上記直交ベクトル W と運動加速度ベクトル VA とに基づいて算出される。なお、ステップS43における重力補正ベクトル ΔVg の具体的な算出方法は、角速度ベクトル $V\omega$ に代えて直交ベクトル W を用いる点を除いてステップS31と同様の方法である。

20

【0166】

本変形例では、ステップS43の次に、上記実施形態と同様のステップS32～S34の処理が実行される。その結果、重力補正ベクトル ΔVg によって重力ベクトル Vg が補正される。本変形例によれば、等角速度でない回転運動にも対応した角速度ベクトル $V\omega$ と運動加速度ベクトル VA と関係を用いて重力ベクトル Vg の補正を行うので、入力装置8が等角速度でない回転運動を行う場合にも重力ベクトル Vg を正しく補正することができる。したがって、CPU10は、上記実施形態に比べてより正確に重力方向を算出することができ、入力装置8をより正確に算出することができる。

30

【0167】

図15の説明に戻り、ステップS13の次のステップS14において、CPU10は、静的補正処理および動的補正処理で補正された重力ベクトルに基づいて補正行列 Ma を算出する。補正行列 Ma は、入力装置8の第1の姿勢を補正するための回転行列である。具体的には、補正行列 Ma は、現在の入力装置8の第1の姿勢から決められる重力方向を、静的補正処理および動的補正処理で補正された重力ベクトル Vg が表す方向と一致するように回転させる回転行列である。ここで、現在の入力装置8の第1の姿勢から決められる重力方向は、上記空間座標系における重力方向 $(0, -1, 0)$ をコントローラ座標系に変換したベクトル、すなわち、 $(0, -1, 0)$ を第1姿勢行列 $M1$ (上式(1)参照)で回転させたベクトル $(-Xy, -Yy, -Zy)$ によって表される。

40

【0168】

ステップS14の具体的な処理としては、CPU10はまず、メインメモリに記憶されている第1姿勢データ68を読み出し、第1姿勢データ68により示される第1姿勢行列 $M1$ を用いて、上記空間座標系における重力方向 $(0, -1, 0)$ をコントローラ座標系に変換したベクトル Vh を算出する。次に、CPU10は、メインメモリに記憶されている重力方向データ69を読み出し、上記ベクトル Vh を、重力方向データ69により示される重力ベクトル Vg に一致するように回転させる回転行列を補正行列 Ma として算出する。算出された補正行列 Ma は、補正行列データ76としてメインメモリに記憶される。

50

ステップS 1 4の次にステップS 1 5の処理が実行される。

【0 1 6 9】

ステップS 1 5において、CPU 1 0は、入力装置8の第1の姿勢（第1姿勢行列M 1）を補正行列M aで補正する。具体的には、メインメモリに記憶されている第1姿勢データ6 8および補正行列データ7 6を読み出し、第1姿勢データ6 8により示される補正前の第1姿勢行列M 1を、補正行列データ7 6により示される補正行列M aで回転させることによって、補正後の第1姿勢行列M 1'を算出する。なお、このとき、CPU 1 0は、補正行列M aに所定の定数C 2を掛け、補正量を調節するようにしてもよい。この定数C 2は、 $0 < C 2 \leq 1$ の範囲で予め定められており、例えば $C 2 = 0.3$ に設定される。補正後の第1姿勢行列M 1'を示すデータが、新たな第1姿勢データ6 8としてメインメモリに記憶される。以上のステップS 1 5によって、加速度に基づく補正処理において入力装置8の第1の姿勢が補正されたこととなる。ステップS 1 5の後、CPU 1 0は加速度に基づく補正処理を終了する。

10

【0 1 7 0】

以上のように、加速度に基づく補正処理においては、静的補正処理（ステップS 1 2）および動的補正処理（ステップS 1 3）において重力方向を補正し、補正後の重力方向を用いて入力装置8の第1の姿勢を補正した。このように、静的補正処理と動的補正処理との両方の補正処理を行うことによって、入力装置8が静止している場合であっても動かされている場合であっても、加速度センサ3 7の検出結果を用いて第1の姿勢を補正することができる。

20

【0 1 7 1】

図1 4の説明に戻り、ステップS 5の次のステップS 6において、CPU 1 0は、前述した撮像画像に基づく補正処理を実行する。撮像画像に基づく補正処理は、撮像画像から得られるマーカ座標データを用いて入力装置8の第1の姿勢を補正する処理である。以下、図2 1を参照して、撮像画像に基づく補正処理の詳細を説明する。

【0 1 7 2】

図2 1は、図1 4に示す撮像画像に基づく補正処理（ステップS 6）の流れを示すフローチャートである。撮像画像に基づく補正処理においては、まずステップS 5 1において、CPU 1 0は、入力装置8の撮像手段（撮像素子4 0）によってマーカ部6が撮像されているか否かを判定する。ステップS 5 1の判定は、メインメモリに記憶されているマーカ座標データ6 5を参照することによって行うことができる。ここでは、マーカ座標データ6 5が2つのマーカ座標を示す場合、マーカ部6が撮像されていると判定し、マーカ座標データ6 5が1つのマーカ座標のみを示す場合、または、マーカ座標がないことを示す場合、マーカ部6が撮像されていないと判定する。ステップS 5 1の判定結果が肯定である場合、以降のステップS 5 2～S 5 7の処理が実行される。一方、ステップS 5 1の判定結果が否定である場合、以降のステップS 5 2～S 5 7の処理がスキップされ、CPU 1 0は、撮像画像に基づく補正処理を終了する。このように、撮像素子4 0によってマーカ部6が撮像されていない場合には、撮像素子4 0から得られるデータを用いて入力装置8の姿勢を算出することができないので、この場合には撮像画像に基づく補正処理において補正は行われない。

30

40

【0 1 7 3】

ステップS 5 2において、CPU 1 0は、マーカ座標データに基づいてロール姿勢成分M 3 rを算出する。ロール姿勢成分M 3 rは、撮像画像内でのマーカ部6の向きに基づいて、すなわち、マーカ座標データ6 5により示される2つのマーカ座標を結ぶ線の傾きに基づいて算出される。以下、図2 2を参照して、ロール姿勢成分M 3 rの算出方法の例を説明する。

【0 1 7 4】

図2 2は、撮像画像に対応する2次元座標を示す図である。図2 2に示されるように、本実施形態においては、撮像画像における位置を表すための2次元座標系（ x' y' 座標系）は、撮像画像の範囲を $-1 \leq x' \leq 1$ 、 $-1 \leq y' \leq 1$ で表すものとする。 x' y'

50

座標系は、入力装置 8 が基準姿勢（入力装置 8 の撮像方向がマーカ部 6 の中央を向き、かつ、コントローラ 5 のボタン面が鉛直上向きとなる姿勢）にある場合に、撮像画像において鉛直下向きが y' 軸正方向となり、右向きが x' 軸正方向となるように設定されるものとする。また、図 22 に示す点 P1 および点 P2 はマーカ座標の位置を示し、点 P3 は点 P1 と点 P2 との midpoint である。図 22 に示すベクトル $v10$ は、点 P1 を始点とし、点 P2 を終点とするベクトルである。

【0175】

ロール姿勢成分 $M3r$ を算出するために、CPU10 はまず、マーカ座標データ 65 を読み出し、当該マーカ座標データ 65 により示される 2 つのマーカ座標から上記ベクトル $v10$ を算出する。さらに、ベクトル $v10$ を正規化したベクトル (hx, hy) を算出する。このベクトル (hx, hy) は、入力装置 8 が上記基準姿勢にある場合に x 軸正方向を向き、入力装置 8 のロール方向の回転に応じて向きが変化する。ベクトル (hx, hy) がロール方向の姿勢に対応しているので、ロール姿勢成分 $M3r$ は、このベクトル (hx, hy) に基づいて算出することができる。具体的には、CPU10 は、次の式 (16) に従ってロール姿勢成分 $M3r$ を算出する。

【数 6】

$$M2r = \begin{bmatrix} hx & -hy & 0 \\ hy & hx & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdots (16)$$

上式 (16) によって算出された行列を示すデータは、ロール姿勢成分データ 77 としてメインメモリに記憶される。ステップ S52 の次にステップ S53 の処理が実行される。

【0176】

ステップ S53 において、CPU10 は、マーカ座標データに基づいてヨー姿勢成分 $M3y$ を算出する。ヨー姿勢成分 $M3y$ は、撮像画像内でのマーカ部 6 の向きおよび位置に基づいて算出される。以下、図 22 を参照して、ヨー姿勢成分 $M3y$ の算出方法の例を説明する。

【0177】

まず、CPU10 は、マーカ座標データ 65 を読み出し、当該マーカ座標データ 65 により示される 2 つのマーカ座標の midpoint を算出する。本実施形態では、マーカ部 6 の位置として当該 midpoint の位置を用いる。さらに、CPU10 は、算出された midpoint の座標を、 $x'y'$ 座標系の原点を中心として、入力装置 8 のロール方向に関する回転角度だけ（入力装置 8 の回転方向とは逆方向に）回転させた座標 (px, py) を算出する。換言すれば、midpoint の座標は、原点を中心として、上記ベクトル (hx, hy) が x 軸正方向を向くように回転される。入力装置 8 が水平方向（ x 軸方向）に関してマーカ部 6 と同じ位置（すなわち、マーカ部 6 の正面の位置）にあるとすれば、上記のようにして得られた回転後の座標 (px, py) から、ヨー方向に関する姿勢を算出することができる。

【0178】

次に、CPU10 は、上記 midpoint の回転後の座標 (px, py) と、マーカ部 6 が x' 軸方向の端に位置する場合のヨー方向の角度（限界角度） $\theta y'$ とに基づいて、ヨー方向に関する回転角度 θy を算出する。ここで、上記限界角度 $\theta y'$ と、上記限界角度 $\theta y1$ となる場合における上記 midpoint の回転後の x 座標値 px' とは予め求めておくことができる。したがって、 px と px' との比が、 θy と $\theta y'$ との比に等しくなることを用いて、ヨー方向に関する回転角度 θy を算出することができる。具体的には、ヨー方向に関する回転角度 θy は、次の式 (17) によって算出することができる。

$$\theta y = px \times \theta y' / px' \cdots (17)$$

なお、マーカ部 6 の水平方向の長さを無視する場合、上記限界角度 $\theta y'$ は、コントローラ 5 の画角の $1/2$ とし、上記 px' の値は“1”とすることができる。

【0179】

最後に、CPU 10は、上式(17)により算出された角度 θ_y の回転を行う回転行列をヨー姿勢成分M3yとして算出する。具体的には、ヨー姿勢成分M3yは次の式(18)によって算出される。

【数7】

$$M2y = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \cdots (18)$$

上式(18)によって算出された行列を示すデータが、ヨー姿勢成分データ78としてメインメモリに記憶される。以上のステップS53の次にステップS54の処理が実行される。

10

【0180】

ステップS54において、CPU 10は、ロール姿勢成分M3rとヨー姿勢成分M3yとを合成する。すなわち、メインメモリからロール姿勢成分データ77およびヨー姿勢成分データ78を読み出し、各データ77および78により示されるロール姿勢成分M3rとヨー姿勢成分M3yとを積算する。ステップS54の次にステップS55の処理が実行される。

【0181】

ステップS55において、CPU 10は、第1の姿勢に基づいてピッチ姿勢成分M3pを算出する。なお、本実施形態の処理とは異なるが、ピッチ姿勢成分M3pについても、ヨー姿勢成分M3yと同様の方法で、上記座標(p_x , p_y)のy座標値に基づいて算出することが可能である。ただし、上記座標(p_x , p_y)を用いてヨー方向(ピッチ方向)の姿勢を算出する方法は、入力装置8が水平方向(ピッチ方向の場合は鉛直方向)に関してマーカ部6と同じ位置にあることを前提として成り立つ方法である。本実施形態のゲームシステム1においては、プレイヤーは、水平方向に関してマーカ部6(テレビ2)のほぼ正面の位置で入力装置8を操作すると考えられるので、「入力装置8が水平方向に関してマーカ部6と同じ位置にある」ことを前提して、上記ステップS53の方法によってヨー方向の姿勢を算出することが可能である。一方、プレイヤーは、立って入力装置8を操作することも座って入力装置8を操作することも考えられ、また、マーカ部6の位置もテレビ2の画面の上側に配置されることも下側に配置されることも考えられる。そのため、本実施形態のゲームシステム1においては、「入力装置8が鉛直方向に関してマーカ部6と同じ位置にある」ことを必ずしも前提にできないため、上記座標(p_x , p_y)を用いてピッチ方向の姿勢を算出することを行わないようにしてもよい。

20

30

【0182】

そこで、本実施形態においては、ピッチ姿勢成分M3pについては、加速度に基づく補正(ステップS5)後の姿勢をそのまま用いることとする(したがって、撮像画像に基づく補正処理においては、ピッチ方向に関しては補正が行われない)。具体的には、CPU 10は、メインメモリから第1姿勢データ68を読み出す。そして、第1姿勢データ68により示される第1姿勢行列M1の各要素を用いて、ピッチ方向に関する回転角度 θ_p を、次の式(19)に従って算出する。

40

$$\begin{aligned} \cos(\theta_p) &= (Z_x \times Z_x + Z_z \times Z_z)^{1/2} \\ \sin(\theta_p) &= Z_y \end{aligned} \cdots (19)$$

上式(19)における変数 Z_x , Z_y , Z_z は、上式(1)において示した第1姿勢行列M1の要素である。なお、ここで用いられる第1姿勢行列M1は、今回の処理ループにおいて加速度に基づく補正処理(ステップS5)が行われた後の第1姿勢行列M1である。さらに、CPU 10は、上式(19)で算出した $\cos(\theta_p)$ および $\sin(\theta_p)$ を用いて、次の式(20)に従ってピッチ姿勢成分M3pの行列を算出する。

【数 8】

$$M2p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta p & \sin \theta p \\ 0 & -\sin \theta p & \cos \theta p \end{bmatrix} \cdots (20)$$

上式(20)によって算出された行列を示すデータが、ピッチ姿勢成分データ79としてメインメモリに記憶される。以上のステップS55の次にステップS56の処理が実行される。

【0183】

10

ステップS56において、CPU10は、ロール方向、ヨー方向、およびピッチ方向の各姿勢成分に基づいて第3姿勢行列M3を算出する。第3姿勢行列M3は、ロール姿勢成分M3rとヨー姿勢成分M3yとの合成結果に、ピッチ姿勢成分M3pをさらに合成することによって得られる。具体的には、CPU10は、メインメモリからピッチ姿勢成分データ79を読み出し、当該ピッチ姿勢成分データ79により示されるピッチ姿勢成分M3pを、ステップS54で算出された行列に積算する。積算の結果得られた第3姿勢行列M3を示すデータは、第3姿勢データ80としてメインメモリに記憶される。ステップS56の次にステップS57の処理が実行される。

【0184】

20

ステップS57において、CPU10は、第3姿勢行列M3を用いて入力装置8の第1の姿勢(第1姿勢行列M1)を補正する。ステップS57における補正は、第1姿勢行列M1を第3姿勢行列M3に所定の割合(下記の定数C3)で近づけることによって行われる。CPU10は、メインメモリから第1姿勢データ68および第3姿勢データ80を読み出す。そして、第1姿勢データ68により示される第1姿勢行列M1と、第3姿勢データ80により示される第3姿勢行列M3とを用いて、次の式(21)に従って補正を行う。

$$M1 = (M3 - M1') \times C3 + M1' \cdots (21)$$

上式(21)において、変数M1'は補正前の第1姿勢行列である。また、定数C3は、 $0 < C3 \leq 1$ の範囲で予め設定され、例えば0.1に設定される。上式(21)によって算出された補正後の第1姿勢行列M1を示すデータが、新たな第1姿勢データ68としてメインメモリに記憶される。上記ステップS57の後、CPU10は撮像画像に基づく補正処理を終了する。

30

【0185】

以上のように、撮像画像に基づく補正処理においては、撮像画像に基づく第3の姿勢を算出し、角速度に基づく第1の姿勢を第3の姿勢に近づけるように補正することとした。この補正によって、第1の姿勢をより正確な値となるように補正することができる。なお、本実施形態においては、ロール方向およびヨー方向の姿勢についてのみ撮像画像から第3の姿勢を算出したが、上述のように、ピッチ方向についても撮像画像から第3の姿勢を算出することは可能であり、他の実施形態においては、ロール方向、ヨー方向、およびピッチ方向について撮像画像から第3の姿勢を算出してもよい。また、撮像画像に基づく補正処理においては、ロール方向、ヨー方向、およびピッチ方向のうち少なくとも1方向について第3の姿勢を算出すればよい。

40

【0186】

なお、他の実施形態においては、撮像画像に基づく補正処理は、入力装置8がマーカ部6を撮像していると推測される場合にのみ、実行されるようにしてもよい。具体的には、CPU10は、撮像画像に基づく補正処理を実行する前に、入力装置8(撮像手段)がマーカ部6を撮像可能な向きを向いているか否かを判定する。この判定は、撮像画像に基づく補正処理を行う前の姿勢を用いて行うことができる。例えば、当該姿勢が、入力装置8の撮像方向が入力装置8からマーカ部6への向きに対して同じ向きかそれとも逆向きかを判定するようにしてもよい。また、上記の判定に用いられる姿勢は、前回の処理ループに

50

において最終的に得られた姿勢であってもよいし、今回の処理ループにおいて加速度に基づく補正処理が行われた姿勢であってもよい。

【0187】

上記の判定の結果、CPU10は、入力装置8がマーカ部6を撮像可能な向きを向いていると判定される場合、撮像画像に基づく補正処理を実行し、マーカ部6を撮像可能な向きを向いていないと判定される場合、撮像画像に基づく補正処理をスキップする。なお、マーカ部6でない物（例えば、部屋の電灯や窓の外の太陽光等）がマーカ部6と誤検出されることがあり、誤検出によって得られたマーカ座標を用いて第3の姿勢が算出される場合、かかる第3の姿勢を用いて補正処理を行っても正しく補正を行うことができない。これに対して、上記の判定処理を行うことにより、誤検出によって得られたマーカ座標から算出される第3の姿勢を用いて補正処理が行われることを防止することができる。これによって、撮像画像に基づく補正処理を正しく行うことができる。

10

【0188】

図14の説明に戻り、ステップS6の次のステップS7において、CPU10は、補正後の入力装置8の姿勢（第1姿勢行列M1）を用いたゲーム処理を実行する。本実施形態では、ゲーム空間内に配置されるカヌーのパドルの姿勢を、入力装置8の姿勢に応じて制御する処理が実行される。さらに、パドルの動きに応じてカヌーを移動させる処理が実行される。なお、他の実施形態においては、ゲーム処理は、補正後の入力装置8の姿勢を表す第1姿勢行列M1を入力値としてゲーム結果に反映させる処理であればどのような処理であってもよい。例えば、仮想のゲーム空間内のオブジェクトを、第1姿勢行列M1により示される姿勢となるように制御して表示する処理であってもよいし、第1姿勢行列M1により示される姿勢と所定の姿勢との角度に応じた速度で上記オブジェクトを移動させるように制御して表示する処理であってもよい。ステップS7の次にステップS8の処理が実行される。

20

【0189】

ステップS8において、CPU10は、ゲームを終了するか否かを判定する。ステップS8の判定は、例えば、ゲームがクリアされたか否か、ゲームオーバーとなったか否か、プレイヤーがゲームを中止する指示を行ったか否か等によって行われる。ステップS8の判定結果が否定である場合、ステップS3の処理が再度実行される。以降、ステップS8でゲームを終了すると判定されるまで、ステップS3～S8の処理ループが繰り返し実行される。一方、ステップS8の判定結果が肯定である場合、CPU10は、図14に示すゲーム処理を終了する。以上で、ゲーム処理の説明を終了する。

30

【0190】

以上のように、本実施形態においては、ジャイロセンサ55および56によって検出された角速度から入力装置8の第1の姿勢を算出し（ステップS4）、第1の姿勢を加速度に基づく補正処理（S5）および撮像画像に基づく補正処理（S6）によって補正した。そして、補正後の第1の姿勢を用いてゲーム処理を実行する（ステップS7）ので、CPU10は、入力装置8の正確な姿勢に基づいたゲーム処理を実行することができる。これによれば、例えば、入力装置8の姿勢をゲーム空間内のオブジェクトの姿勢に対して正確に反映させることができるので、ゲーム操作の操作性を向上することができる。

40

【0191】

[変形例]

なお、上記実施形態においては、動的補正処理によって補正された重力ベクトルは、ジャイロセンサ55および56が検出した角速度に基づいて算出された入力装置8の姿勢を補正するために用いられた。つまり、上記実施形態においては、ゲーム装置3は、動的補正処理よりも先に算出された入力装置8の第1の姿勢を上記重力ベクトルを用いて補正することによって、入力装置8の姿勢を算出するものであった。ここで、上記重力ベクトルは、先に算出された姿勢を補正するために用いられる必要はなく、入力装置8の姿勢を算出するために用いられればよい。例えば、他の実施形態においては、ゲーム装置3は、次の（1）～（4）の処理によって入力装置8の姿勢を算出することも可能である。

50

(1) 検出加速度ベクトル V_a に基づいて重力ベクトル V_g を算出する (検出加速度ベクトル V_a を重力ベクトル V_g とする)。

(2) 上記ステップ S 2 4 と同様の方法で、重力ベクトル V_g と検出加速度ベクトル V_a とに基づいて運動加速度ベクトル V_A を算出する。

(3) 上記動的補正処理と同様の処理によって重力ベクトル V_g を補正する。

(4) 補正された重力ベクトル V_g に対応する入力装置 8 の姿勢を算出する。

上記 (1) ~ (4) の処理によれば、ジャイロセンサ 5 5 および 5 6 が検出した角速度等から先に入力装置 8 の姿勢を算出しなくとも、動的補正処理によって補正された重力ベクトルに基づいて入力装置 8 の姿勢を算出することができる。

【0192】

10

また、上記実施形態においては、動的補正処理において補正の対象となる重力ベクトルを、角速度データ 6 3 から算出される第 1 の姿勢に基づいて算出した。ここで、他の実施形態においては、動的補正処理において補正の対象となる重力ベクトルは、入力装置 8 が備える各種センサの出力 (すなわち、操作データ) を用いて算出されればよく、例えば上記 (1) の処理のように、検出加速度ベクトル V_a を用いて算出されてもよい。また、動的補正処理において補正の対象となる重力ベクトルの算出方法は、動的補正処理における補正方法とは異なる方法であれば、どのような方法であってもよい。

【産業上の利用可能性】

【0193】

以上のように、本発明は、ジャイロセンサを用いて入力装置の姿勢を正確に算出すること等を目的として、例えば入力装置の姿勢に応じたゲーム処理を行うゲーム装置またはゲームプログラム等として利用することが可能である。

20

【符号の説明】

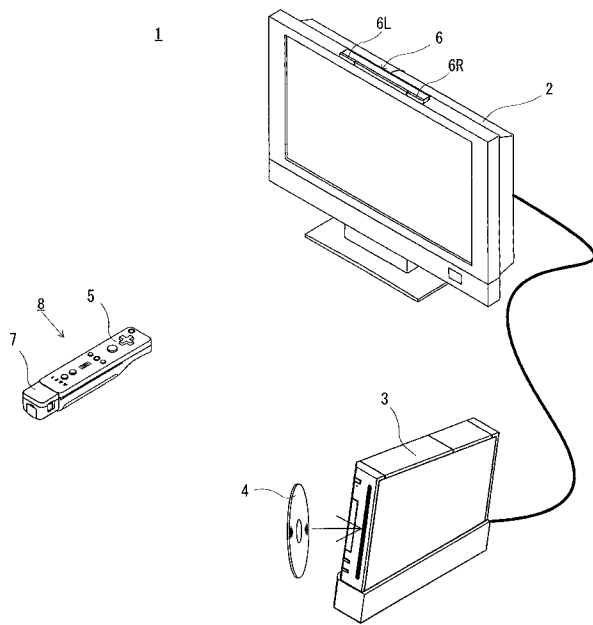
【0194】

- 1 ゲームシステム
- 2 テレビ
- 3 ゲーム装置
- 4 光ディスク
- 5 コントローラ
- 6 マーカ部
- 7 ジャイロセンサユニット
- 8 入力装置
- 10 CPU
- 11c GPU
- 11e 内部メインメモリ
- 12 外部メインメモリ
- 37 加速度センサ
- 55 2 軸ジャイロセンサ
- 56 1 軸ジャイロセンサ
- 60 ゲームプログラム
- 61 姿勢算出プログラム
- 63 角速度データ
- 64 加速度データ
- 68 第 1 姿勢データ
- 69 重力方向データ
- 74 運動加速度データ

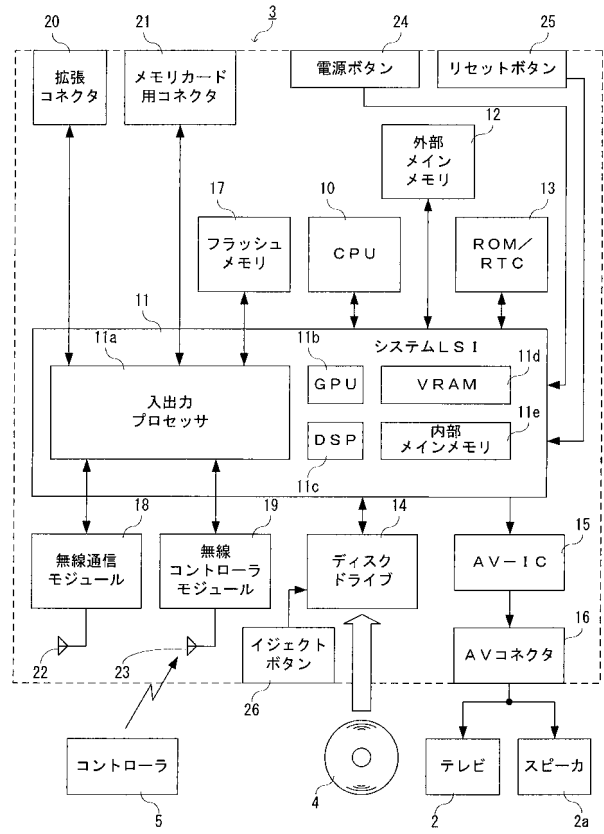
30

40

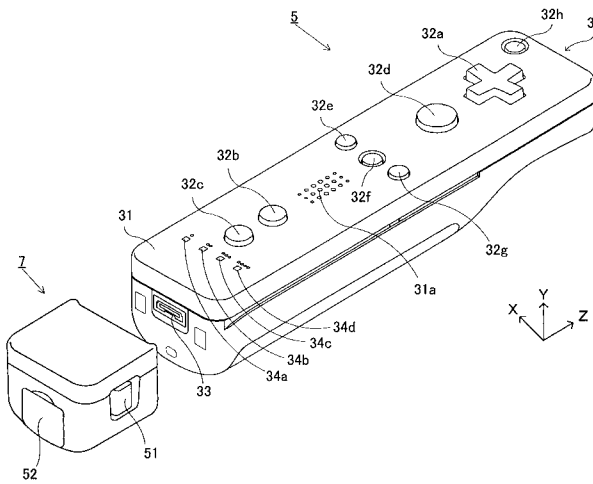
【図 1】



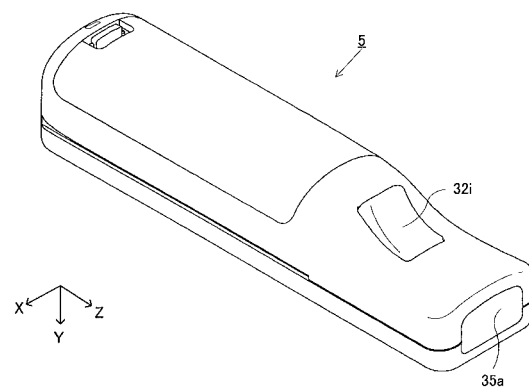
【図 2】



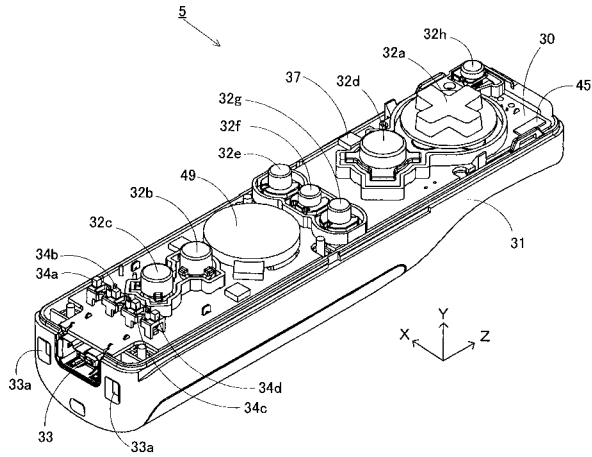
【図 3】



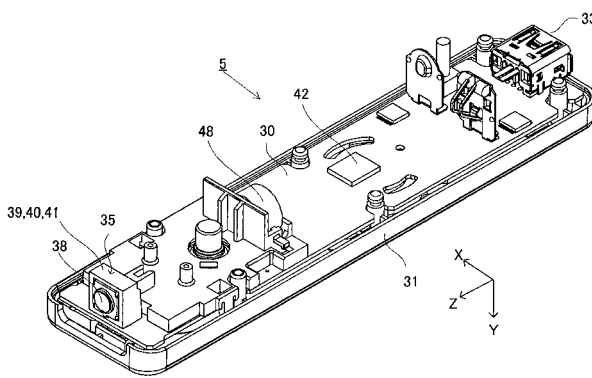
【図 4】



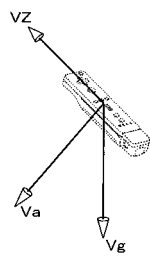
【図 5】



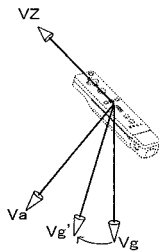
【図 6】



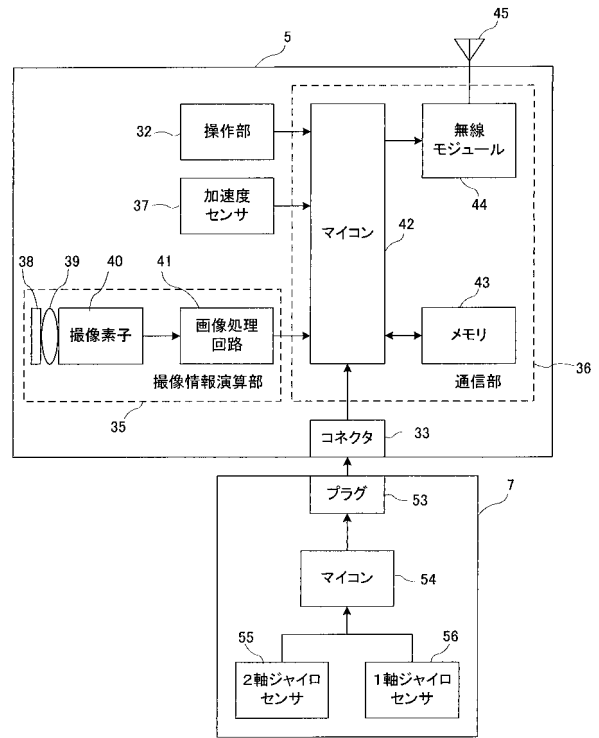
【図 8】



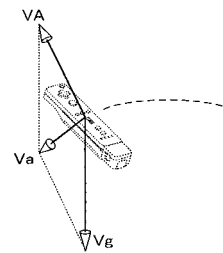
【図 9】



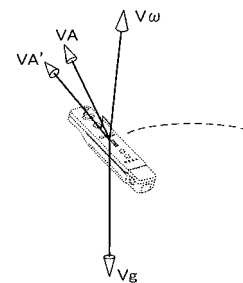
【図 7】



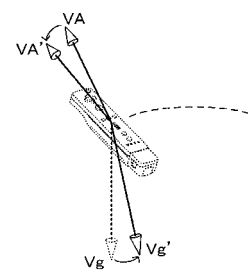
【図 10】



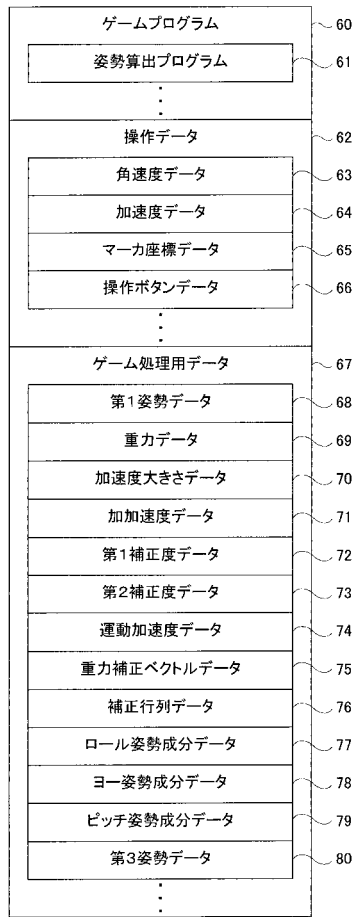
【図 11】



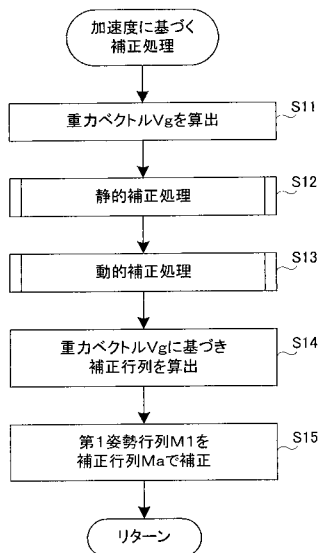
【図 12】



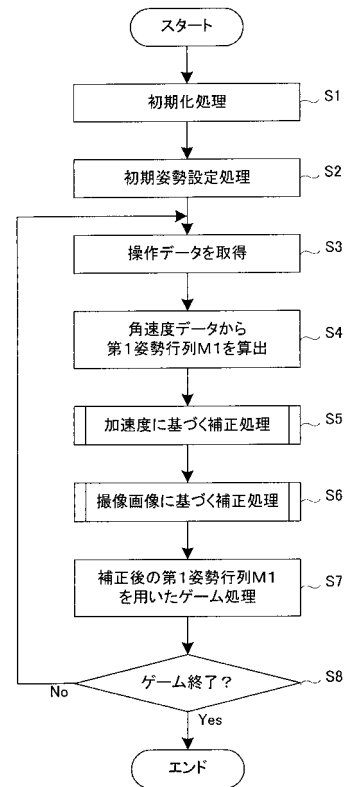
【図 13】



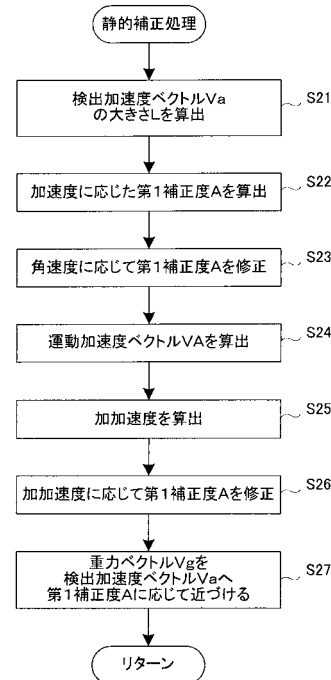
【図 15】



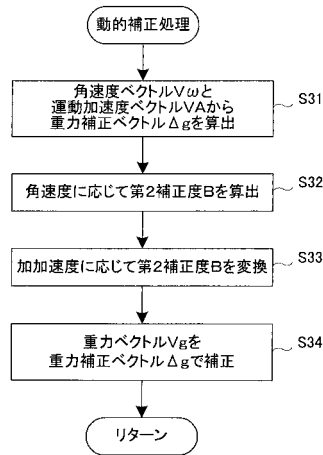
【図 14】



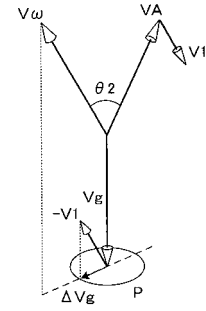
【図 16】



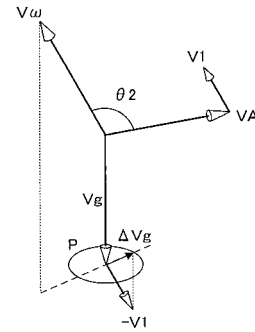
【図 17】



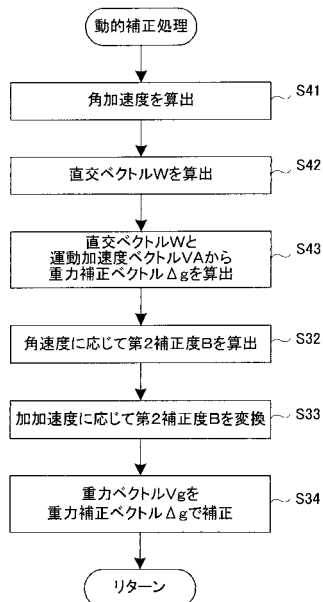
【図 18】



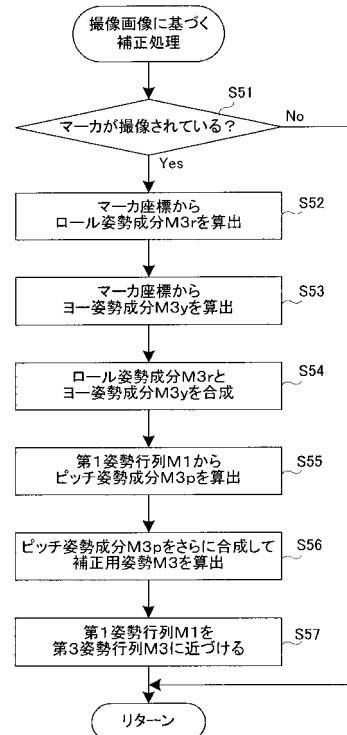
【図 19】



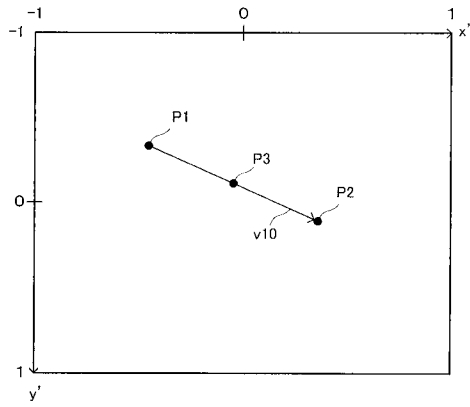
【図 20】



【図 21】



【図 22】



フロントページの続き

(56)参考文献 特開2008-002992 (JP, A)
特開2007-232662 (JP, A)
特開2000-308756 (JP, A)
国際公開第2008/026357 (WO, A1)

(58)調査した分野(Int.Cl., DB名)
A63F 13/00 - 13/12
A63F 9/24