

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-20554

(P2010-20554A)

(43) 公開日 平成22年1月28日(2010.1.28)

(51) Int.Cl.	F I	テーマコード (参考)
G 0 6 F 3/038 (2006.01)	G 0 6 F 3/038 3 1 0 Y	2 C 0 0 1
A 6 3 F 13/00 (2006.01)	A 6 3 F 13/00 F	2 F 1 0 5
G 0 1 C 19/00 (2006.01)	G 0 1 C 19/00 Z	5 B 0 8 7

審査請求 未請求 請求項の数 17 O L (全 39 頁)

(21) 出願番号	特願2008-180604 (P2008-180604)	(71) 出願人	000233778
(22) 出願日	平成20年7月10日 (2008. 7. 10)		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	100098291
			弁理士 小笠原 史朗
		(74) 代理人	100130269
			弁理士 石原 盛規
		(72) 発明者	太田 敬三
			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
		(72) 発明者	佐藤 賢太
			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
		F ターム (参考)	2C001 CA01 CA09 CB01 CB03 CB06
			CB08 CC01 CC08

最終頁に続く

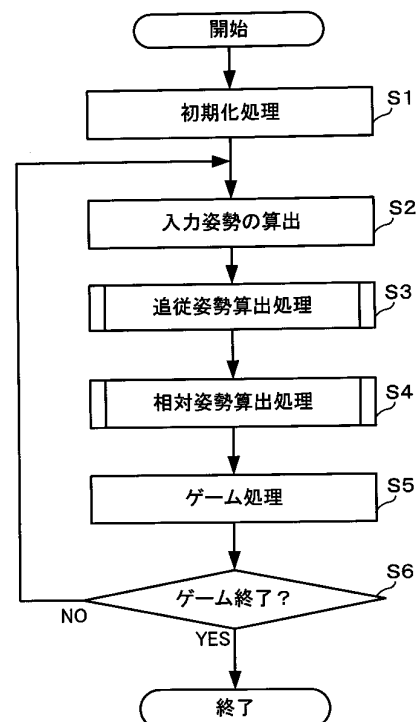
(54) 【発明の名称】 情報処理プログラムおよび情報処理装置

(57) 【要約】

【課題】 ジャイロ信号から高い精度で姿勢を算出することのできる情報処理プログラムおよび情報処理装置を提供すること。

【解決手段】 第1の姿勢算出手段で、動き検出センサを備えた入力装置から出力される動き検出信号に基づいて入力装置自体の姿勢を示す第1の姿勢を算出する。動き算出手段で、第1の姿勢の動きを算出する。接近動作判定手段で、動き算出手段によって算出された第1の姿勢の動きが所定の姿勢に近づく動きか否かを判定する。入力姿勢設定手段で、第1の姿勢の動きが所定の姿勢に近づく動きであれば第1の姿勢を補正した姿勢を入力姿勢として設定し、近づく動きでないときは第1の姿勢を入力姿勢として設定する。処理実行手段で、入力姿勢に基づいて所定の情報処理を行う。

【選択図】 図 1 6



【特許請求の範囲】

【請求項 1】

自身の動きを検出する動き検出センサを備えた入力装置から出力される動き検出信号に基づいて所定の情報処理を行う情報処理装置のコンピュータに実行させる情報処理プログラムであって、

前記コンピュータを、

前記動き検出信号に基づいて前記入力装置の姿勢を示す第 1 の姿勢を繰り返し算出する第 1 の姿勢算出手段と、

前記第 1 の姿勢算出手段によって繰り返し算出された第 1 の姿勢に基づいて当該第 1 の姿勢の動きを算出する動き算出手段と、

前記動き算出手段によって算出された前記第 1 の姿勢の動きに基づいて、当該第 1 の姿勢が所定の姿勢に近づく動きをしているか否かを判定するための接近動作判定手段と、

前記第 1 の姿勢が前記所定の姿勢に近づく動きをしていれば、当該第 1 の姿勢を前記所定の姿勢方向に一致するように、あるいは、近づくように補正した姿勢を入力姿勢として設定し、近づく動きでなければ前記第 1 の姿勢を入力姿勢として設定する入力姿勢設定手段と、

前記入力姿勢に基づいて所定の情報処理を行う処理実行手段として機能させる、情報処理プログラム。

【請求項 2】

前記動き算出手段は、前記第 1 の姿勢算出手段によって繰り返し算出された第 1 の姿勢の変化に基づいて当該第 1 の姿勢の移動ベクトルを算出する移動ベクトル算出手段を含み、

前記接近動作判定手段は、前記移動ベクトルに基づいて前記第 1 の姿勢の動きが前記所定の姿勢に近づく動きか否かを判定する、請求項 1 に記載の情報処理プログラム。

【請求項 3】

前記情報処理プログラムは、前記コンピュータに、前記第 1 の姿勢と関連する所定の姿勢である第 2 の姿勢を算出する第 2 の姿勢算出手段として更に機能させ、

前記第 2 の姿勢を前記所定の姿勢として用いる、請求項 2 に記載の情報処理プログラム。

【請求項 4】

前記第 2 の姿勢算出手段は、前記第 1 の姿勢に徐々に近づいていくような姿勢である追従姿勢を前記第 2 の姿勢として算出する、請求項 3 に記載の情報処理プログラム。

【請求項 5】

前記動き算出手段は、

前記追従姿勢が前記第 1 の姿勢に向かうベクトルである追従ベクトルを算出する追従ベクトル算出手段を含み、

前記接近動作判定手段は、前記移動ベクトルと前記追従ベクトルとの成す角度に基づいて前記第 1 の姿勢が前記第 2 の姿勢に近づく動きをしているか否かを判定する、請求項 4 に記載の情報処理プログラム。

【請求項 6】

前記接近動作判定手段は、

前記移動ベクトルと前記追従ベクトルとの内積を算出する内積算出手段を含み、

当該内積算出手段で算出された内積に基づいて前記第 1 の姿勢が前記第 2 の姿勢に近づく動きをしているか否かを判定する、請求項 5 に記載の情報処理プログラム。

【請求項 7】

前記入力姿勢設定手段は、

前記所定の姿勢を基準とした座標系における前記第 1 の姿勢を示す第 3 の姿勢を算出する第 3 の姿勢算出手段を含み、

前記第 1 の姿勢が前記所定の姿勢に近づく動きをしていれば、当該所定の姿勢を第 1 の姿勢に近づけてから前記第 3 の姿勢を算出して前記入力姿勢として設定し、近づく動き

10

20

30

40

50

でなければ、当該所定の姿勢を動かさずに第 3 の姿勢を算出して前記入力姿勢として設定する、請求項 1 に記載の情報処理プログラム。

【請求項 8】

前記入力姿勢設定手段は、

予め定められた姿勢である基本姿勢を基準とした座標系における前記第 3 の姿勢を示す第 4 の姿勢を算出する第 4 の姿勢算出手段を含み、

当該第 4 の姿勢を前記入力姿勢として設定する、請求項 7 に記載の情報処理プログラム。

【請求項 9】

前記情報処理プログラムは、前記コンピュータに、前記第 1 の姿勢と関連する所定の姿勢である第 2 の姿勢を算出する第 2 の姿勢算出手段として更に機能させ、

前記第 2 の姿勢を前記所定の姿勢として用いる、請求項 1 に記載の情報処理プログラム。

【請求項 10】

前記第 2 の姿勢算出手段は、前記第 1 の姿勢に徐々に近づいていくような姿勢である追従姿勢を第 2 の姿勢として算出する、請求項 3 に記載の情報処理プログラム。

【請求項 11】

自身の動きを検出する動き検出センサを備えた入力装置から出力される動き検出信号に基づいて所定の情報処理を行う情報処理装置のコンピュータに実行させる情報処理プログラムであって、

前記コンピュータを、

前記動き検出信号に基づいて前記入力装置の姿勢を示す入力姿勢を繰り返し算出する入力姿勢算出手段と、

前記入力姿勢算出手段によって繰り返し算出された入力姿勢に基づいて、所定軸を中心とした当該入力姿勢の回転方向を算出する回転方向算出手段と、

前記入力姿勢の回転方向が前記所定の基準回転方向に一致するように、あるいは、近づくように前記入力姿勢を補正する入力姿勢補正手段と、

前記入力姿勢補正手段で補正された入力姿勢に基づいて所定の情報処理を行う処理実行手段として機能させる、情報処理プログラム。

【請求項 12】

前記回転方向算出手段は、

前記入力姿勢算出手段によって繰り返し算出された入力姿勢から、当該入力姿勢の回転に対する回転軸を算出する回転軸算出手段を含み、

前記入力姿勢補正手段は、前記回転軸算出手段で算出された前記回転軸が、前記所定の基準回転方向の回転軸に一致するように、あるいは、近づくように前記入力姿勢を補正する、請求項 11 に記載の情報処理プログラム。

【請求項 13】

前記回転軸算出手段は、

前記入力姿勢算出手段によって繰り返し算出された入力姿勢の時系列順で前後する入力姿勢に対して直行する入力回転軸をそれぞれ算出する入力回転軸算出手段と、

前記入力回転軸算出手段で算出された各入力回転軸の平均となる平均回転軸を算出する平均回転軸算出手段を含み、

前記平均回転軸算出手段で算出された前記平均回転軸を、前記入力姿勢の回転に対する回転軸とする、請求項 12 に記載の情報処理プログラム。

【請求項 14】

前記情報処理プログラムは、前記コンピュータに、前記回転軸算出手段で算出された前記回転軸を示すベクトルと、真上方向を示すベクトルである真上ベクトルとの外積として示され、前記入力装置の前方方向を示す前方ベクトルを算出する前方ベクトル算出手段として更に機能させ、

前記入力姿勢補正手段は、前記前方ベクトル算出手段で算出された前記前方ベクトルが

10

20

30

40

50

前記所定の基準回転方向に一致するように、あるいは、近づくように前記入力姿勢を補正する、請求項 1 2 に記載の情報処理プログラム。

【請求項 1 5】

前記情報処理プログラムは、前記コンピュータに、前記入力姿勢の回転方向と、鉛直線と垂直になる軸を中心に回転する方向との近似度を算出する近似度算出手段として更に機能させ、

入力姿勢補正手段は、前記近似度算出手段で算出された近似度に応じて補正を行う、請求項 1 1 に記載の情報処理プログラム。

【請求項 1 6】

自身の動きを検出する動き検出センサを備えた入力装置から出力される動き検出信号に基づいて所定の情報処理を行う情報処理装置であって、

前記動き検出信号に基づいて前記入力装置の姿勢を示す第 1 の姿勢を繰り返し算出する第 1 の姿勢算出手段と、

前記第 1 の姿勢算出手段によって繰り返し算出された第 1 の姿勢に基づいて当該第 1 の姿勢の動きを算出する動き算出手段と、

前記動き算出手段によって算出された前記第 1 の姿勢の動きに基づいて、当該第 1 の姿勢が所定の姿勢に近づく動きをしているか否かを判定するための接近動作判定手段と、

前記第 1 の姿勢が前記所定の姿勢に近づく動きをしていれば、当該第 1 の姿勢を前記所定の姿勢方向に一致するように、あるいは、近づくように補正した姿勢を入力姿勢として設定し、近づく動きでなければ前記第 1 の姿勢を入力姿勢として設定する入力姿勢設定手段と、

前記入力姿勢に基づいて所定の情報処理を行う処理実行手段とを備える、情報処理装置。

【請求項 1 7】

自身の動きを検出する動き検出センサを備えた入力装置から出力される動き検出信号に基づいて所定の情報処理を行う情報処理装置であって、

前記動き検出信号に基づいて前記入力装置の姿勢を示す入力姿勢を繰り返し算出する入力姿勢算出手段と、

前記入力姿勢算出手段によって繰り返し算出された入力姿勢に基づいて、所定軸を中心とした当該入力姿勢の回転方向を算出する回転方向算出手段と、

前記入力姿勢の回転方向が前記所定の基準回転方向に一致するように、あるいは、近づくように前記入力姿勢を補正する入力姿勢補正手段と、

前記入力姿勢補正手段で補正された入力姿勢に基づいて所定の情報処理を行う処理実行手段とを備える、情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、動き検出センサを備えた入力装置自体の姿勢を検出し、当該検出された姿勢を仮想空間内の所定のオブジェクトの動きに反映させる情報処理装置のコンピュータに実行させる情報処理プログラムに関し、より特定的には、検出された入力装置の姿勢を補正して仮想空間内の所定のオブジェクトの動きに反映させる情報処理プログラム、情報処理装置に関する。

【背景技術】

【0002】

従来より、3 軸型ジャイロや 3 軸型加速計から出力される信号に基づいて姿勢角を算出する姿勢角検出装置が知られている。この姿勢角検出装置では、加減速時や旋回時において発生する余分な加速度が原因となり、算出される姿勢角に誤差が生じるという問題があった。そのため、上記姿勢角検出装置に更に速度センサを搭載した姿勢角検出装置が開示されている（例えば、特許文献 1）。この姿勢角検出装置では、当該速度センサから出力される速度信号を利用することで、上述のような誤差を補正して姿勢角の算出を行って

る。

【特許文献1】特開2004-132769号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

しかしながら、上述したような姿勢角検出装置においては、以下に示す問題点があった。すなわち、上記の姿勢角検出装置では、加減速時や旋回時において精度の高い姿勢角の算出が可能な反面、3軸型ジャイロのみを用いて姿勢角を算出する場合と比べると、3軸加速度計、更には速度センサといったセンサが必要となる。そのため、姿勢角算出装置にかかる費用効率の観点からすると、不利なものとなっていた。

10

【0004】

それ故に、本発明の目的は、費用効率面で有利であり、ジャイロ信号から高い精度で姿勢を算出することのできる情報処理プログラムおよび情報処理装置を提供することである。

【課題を解決するための手段】

【0005】

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、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の一例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。

【0006】

20

第1の発明は、自身の動きを検出する動き検出センサを備えた入力装置から出力される動き検出信号に基づいて所定の情報処理を行う情報処理装置のコンピュータに実行させる情報処理プログラムであって、コンピュータを、第1の姿勢算出手段(S2)と、動き算出手段(S21)と、接近動作判定手段(S22~S24)と、入力姿勢設定手段(S27, S28, S4)と、処理実行手段(S5)として機能させる。第1の姿勢算出手段は、動き検出信号に基づいて入力装置の姿勢を示す第1の姿勢を繰り返し算出する。動き算出手段は、第1の姿勢算出手段によって繰り返し算出された第1の姿勢に基づいて当該第1の姿勢の動きを算出する。接近動作判定手段は、動き算出手段によって算出された第1の姿勢の動きに基づいて、当該第1の姿勢が所定の姿勢に近づく動きをしているか否かを判定する。入力姿勢設定手段は、第1の姿勢が所定の姿勢に近づく動きをしていれば、当該第1の姿勢を所定の姿勢方向に一致するように、あるいは、近づくように補正した姿勢を入力姿勢として設定し、近づく動きでなければ第1の姿勢を入力姿勢として設定する。処理実行手段は、入力姿勢に基づいて所定の情報処理を行う。

30

【0007】

第1の発明によれば、入力装置から算出される姿勢の補正について、ユーザに違和感を与えることがないような精度の高い補正を実現することができる。

【0008】

第2の発明は、第1の発明において、動き算出手段は、第1の姿勢算出手段によって繰り返し算出された第1の姿勢の変化に基づいて当該第1の姿勢の移動ベクトルを算出する移動ベクトル算出手段を含む。そして、接近動作判定手段は、移動ベクトルに基づいて第1の姿勢の動きが所定の姿勢に近づく動きか否かを判定する。

40

【0009】

第2の発明によれば、移動ベクトルを用いて入力姿勢の動きを判定するため、判定処理を簡易なものにすることができ、処理負荷を軽減することが可能となる。

【0010】

第3の発明は、第2の発明において、情報処理プログラムは、コンピュータに、第1の姿勢と関連する所定の姿勢である第2の姿勢を算出する第2の姿勢算出手段として更に機能させ、第2の姿勢を所定の姿勢として用いる。

【0011】

第4の発明は、第3の発明において、第2の姿勢算出手段は、第1の姿勢に徐々に近づ

50

いていくような姿勢である追従姿勢を第2の姿勢として算出する。

【0012】

第3乃至第4の発明によれば、第1の姿勢の動きが第2の姿勢に近づく動きか否かの判定処理を簡易なものにすることができ、処理負荷を軽減することが可能となる。

【0013】

第5の発明は、第4の発明において、動き算出手段は、追従姿勢が第1の姿勢に向かうベクトルである追従ベクトルを算出する追従ベクトル算出手段(S22)を含む。そして、接近動作判定手段は、移動ベクトルと追従ベクトルとの成す角度に基づいて第1の姿勢が第2の姿勢に近づく動きをしているか否かを判定する。

【0014】

第5の発明によれば、ベクトル間のなす角度に基づいて処理を行うため、判定処理を簡易なものにすることができ、処理負荷を軽減することが可能となる。

【0015】

第6の発明は、第5の発明において、接近動作判定手段は、移動ベクトルと追従ベクトルとの内積を算出する内積算出手段(S23)を含む。そして、接近動作判定手段は、当該内積算出手段で算出された内積に基づいて第1の姿勢が第2の姿勢に近づく動きをしているか否かを判定する。

【0016】

第6の発明によれば、ベクトル間の内積に基づいて処理を行うため、判定処理を簡易なものにすることができ、処理負荷を軽減することが可能となる。

【0017】

第7の発明は、第1の発明において、入力姿勢設定手段は、所定の姿勢を基準とした座標系における第1の姿勢を示す第3の姿勢を算出する第3の姿勢算出手段(S41)を含む。そして、入力姿勢設定手段は、第1の姿勢が所定の姿勢に近づく動きをしていれば、当該所定の姿勢を第1の姿勢に近づけてから第3の姿勢を算出して入力姿勢として設定し、近づく動きでなければ、当該所定の姿勢を動かさずに第3の姿勢を算出して入力姿勢として設定する。

【0018】

第7の発明によれば、第1の姿勢について直接補正せずに、補正後の姿勢として代わりに第3の姿勢をゲームパラメータとして用いるため、補正前の第1の姿勢について保持しておくことができ、他の処理で利用することが可能となる。

【0019】

第8の発明は、第7の発明において、入力姿勢設定手段は、予め定められた姿勢である基本姿勢を基準とした座標系における第3の姿勢を示す第4の姿勢を算出する第4の姿勢算出手段(S42)を含み、当該第4の姿勢を入力姿勢として設定する。

【0020】

第8の発明によれば、ユーザに対して与える違和感を更に軽減することが可能となる。

【0021】

第9の発明は、第1の発明において、情報処理プログラムは、コンピュータに、第1の姿勢と関連する所定の姿勢である第2の姿勢を算出する第2の姿勢算出手段として更に機能させ、第2の姿勢を所定の姿勢として用いる。

【0022】

第10の発明は、第3の発明において、第2の姿勢算出手段は、第1の姿勢に徐々に近づいていくような姿勢である追従姿勢を第2の姿勢として算出する。

【0023】

第9乃至第10の発明によれば、第1の姿勢の動きが第2の姿勢に近づく動きか否かの判定処理を簡易なものにすることができ、処理負荷を軽減することが可能となる。

【0024】

第11の発明は、自身の動きを検出する動き検出センサを備えた入力装置から出力される動き検出信号に基づいて所定の情報処理を行う情報処理装置のコンピュータに実行させ

10

20

30

40

50

る情報処理プログラムであって、コンピュータを、入力姿勢算出手段（Ｓ２）と、回転方向算出手段（Ｓ５７）と、入力姿勢補正手段（Ｓ５８）と、処理実行手段（Ｓ５９）として機能させる。入力姿勢算出手段は、動き検出信号に基づいて入力装置の姿勢を示す入力姿勢を繰り返し算出する。回転方向算出手段は、入力姿勢算出手段によって繰り返し算出された入力姿勢に基づいて、所定軸を中心とした当該入力姿勢の回転方向を算出する。入力姿勢補正手段は、入力姿勢の回転方向が所定の基準回転方向に一致するように、あるいは、近づくように入力姿勢を補正する。処理実行手段は、入力姿勢補正手段で補正された入力姿勢に基づいて所定の情報処理を行う。

【００２５】

第１１の発明によれば、入力装置から算出される姿勢の補正について、ユーザに違和感を与えることがないような精度の高い補正を実現することができる。

10

【００２６】

第１２の発明は、第１１の発明において、回転方向算出手段は、入力姿勢算出手段によって繰り返し算出された入力姿勢から、当該入力姿勢の回転に対する回転軸を算出する回転軸算出手段（Ｓ７８）を含む。そして、入力姿勢補正手段は、回転軸算出手段で算出された回転軸が、所定の基準回転方向の回転軸に一致するように、あるいは、近づくように入力姿勢を補正する。

【００２７】

第１２の発明によれば、プレイヤーに違和感を与えることなく、入力姿勢について補正を行い、ゲーム処理に反映させることが可能となる。

20

【００２８】

第１３の発明は、第１２の発明において、回転軸算出手段は、入力姿勢算出手段によって繰り返し算出された入力姿勢の時系列順で前後する入力姿勢に対して直行する入力回転軸をそれぞれ算出する入力回転軸算出手段（Ｓ７４～Ｓ７７）と、入力回転軸算出手段で算出された各入力回転軸の平均となる平均回転軸を算出する平均回転軸算出手段（Ｓ７８）を含む。そして、回転軸算出手段は、平均回転軸算出手段で算出された平均回転軸を、入力姿勢の回転に対する回転軸とする。

【００２９】

第１３の発明によれば、プレイヤーに違和感を与えずに入力姿勢の回転軸の補正を行うことができる。

30

【００３０】

第１４の発明は、第１２の発明において、情報処理プログラムは、コンピュータに、回転軸算出手段で算出された回転軸を示すベクトルと、真上方向を示すベクトルである真上ベクトルとの外積として示され、入力装置の前方方向を示す前方ベクトルを算出する前方ベクトル算出手段（Ｓ８１）として更に機能させる。そして、入力姿勢補正手段は、前方ベクトル算出手段で算出された前方ベクトルが所定の基準回転方向に一致するように、あるいは、近づくように入力姿勢を補正する。

【００３１】

第１４の発明によれば、入力姿勢の向いている方向について、プレイヤーに違和感を与えることなく所定の方向を向くように補正を行い、ゲーム処理に反映させることが可能となる。

40

【００３２】

第１５の発明は、第１１の発明において、情報処理プログラムは、コンピュータに、入力姿勢の回転方向と、鉛直線と垂直になる軸を中心に回転する方向との近似度を算出する近似度算出手段として更に機能させる。そして、入力姿勢補正手段は、近似度算出手段で算出された近似度に応じて補正を行う。

【００３３】

第１５の発明によれば、ユーザに違和感を与えることなく、入力装置から算出される姿勢の向きを補正して情報処理を実行することができる。

【００３４】

50

第16の発明は、自身の動きを検出する動き検出センサを備えた入力装置(5)から出力される動き検出信号に基づいて所定の情報処理を行う情報処理装置(3)であって、第1の姿勢算出手段(10)と、動き算出手段(10)と、接近動作判定手段(10)と、入力姿勢設定手段(10)と、処理実行手段(10)とを備える。第1の姿勢算出手段は、動き検出信号に基づいて入力装置の姿勢を示す第1の姿勢を繰り返し算出する。動き算出手段は、第1の姿勢算出手段によって繰り返し算出された第1の姿勢に基づいて当該第1の姿勢の動きを算出する。接近動作判定手段は、動き算出手段によって算出された第1の姿勢の動きに基づいて、当該第1の姿勢が所定の姿勢に近づく動きをしているか否かを判定する。入力姿勢設定手段は、第1の姿勢が所定の姿勢に近づく動きをしていれば、当該第1の姿勢を所定の姿勢方向に一致するように、あるいは、近づくように補正した姿勢を入力姿勢として設定し、近づく動きでなければ第1の姿勢を入力姿勢として設定する。処理実行手段は、入力姿勢に基づいて所定の情報処理を行う。

10

【0035】

第16の発明によれば、上記第1の発明と同様の効果を得ることができる。

【0036】

第17の発明は、自身の動きを検出する動き検出センサを備えた入力装置(5)から出力される動き検出信号に基づいて所定の情報処理を行う情報処理装置(3)であって、入力姿勢算出手段(10)と、回転方向算出手段(10)と、入力姿勢補正手段(10)と、処理実行手段(10)とを備える。入力姿勢算出手段は、動き検出信号に基づいて入力装置の姿勢を示す入力姿勢を繰り返し算出する。回転方向算出手段は、入力姿勢算出手段によって繰り返し算出された入力姿勢に基づいて、所定軸を中心とした当該入力姿勢の回転方向を算出する。入力姿勢補正手段は、入力姿勢の回転方向が所定の基準回転方向に一致するように、あるいは、近づくように入力姿勢を補正する。処理実行手段は、入力姿勢補正手段で補正された入力姿勢に基づいて所定の情報処理を行う。

20

【0037】

第17の発明によれば、上記第11の発明と同様の効果を得ることができる。

【発明の効果】

【0038】

本発明によれば、例えばジャイロセンサのみを搭載した入力装置であっても、当該入力装置の姿勢の補正について、ユーザに違和感を感じさせないような精度の高い補正を実現することができる。

30

【発明を実施するための最良の形態】

【0039】

以下、本発明の実施の形態について、図面を参照して説明する。尚、この実施例により本発明が限定されるものではない。

【0040】

[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

【0041】

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

50

る。

【0042】

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。

【0043】

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。

【0044】

[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。

【0045】

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。

【0046】

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、VRAM11d、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。

【0047】

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

10

20

30

40

50

ータを用いて画像データを作成する。

【0048】

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。

【0049】

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。

10

【0050】

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。

【0051】

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。

20

30

【0052】

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。

【0053】

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。

40

【0054】

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

50

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、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。

【0055】

〔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。

【0056】

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。

【0057】

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、マイナスイボタン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。

【0058】

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。

【0059】

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。

【0060】

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。

【0061】

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

10

20

30

40

50

31aが形成されている。

【0062】

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。

【0063】

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。

【0064】

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。

【0065】

さらに、基板30の下主面上には、上記マイコン42およびバイブレータ48が設けられている。バイブレータ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レータ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ー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、バイブレータ48は、ハウジング31のやや前方寄りに配置される。つまり、バイブレータ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、バイブレータ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。

【0066】

また、ジャイロセンサユニット7は、3軸周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。

【0067】

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 3 3 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 3 3) に対して装着可能な他の機器は、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 5 2 が着脱可能に装着されている。

【0068】

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 3 5 の位置 (撮像情報演算部 3 5 の光入射面 3 5 a) は、ハウジング 3 1 の前面でなくてもよく、ハウジング 3 1 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。

【0069】

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 3 2 (各操作ボタン 3 2 a ～ 3 2 i) 、コネクタ 3 3 、撮像情報演算部 3 5 、通信部 3 6 、および加速度センサ 3 7 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。

【0070】

操作部 3 2 は、上述した各操作ボタン 3 2 a ～ 3 2 i を含み、各操作ボタン 3 2 a ～ 3 2 i に対する入力状態 (各操作ボタン 3 2 a ～ 3 2 i が押下されたか否か) を示す操作ボタンデータを通信部 3 6 のマイコン 4 2 へ出力する。

【0071】

撮像情報演算部 3 5 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 3 5 は、例えば最大 2 0 0 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。

【0072】

撮像情報演算部 3 5 は、赤外線フィルタ 3 8 、レンズ 3 9 、撮像素子 4 0 、および画像処理回路 4 1 を含んでいる。赤外線フィルタ 3 8 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 3 9 は、赤外線フィルタ 3 8 を透過した赤外線を集光して撮像素子 4 0 へ入射させる。撮像素子 4 0 は、例えば C M O S センサやあるいは C C D センサのような固体撮像素子であり、レンズ 3 9 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 6 R および 6 L は、テレビ 2 の前方に向かって赤外光を出力する赤外 L E D で構成される。したがって、赤外線フィルタ 3 8 を設けることによって、撮像素子 4 0 は、赤外線フィルタ 3 8 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 6 R および 6 L 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 4 0 によって撮像された画像を撮像画像と呼ぶ。撮像素子 4 0 によって生成された画像データは、画像処理回路 4 1 で処理される。画像処理回路 4 1 は、撮像画像内における撮像対象 (マーカ 6 R および 6 L) の位置を算出する。画像処理回路 4 1 は、算出された位置を示す座標を通信部 3 6 のマイコン 4 2 へ出力する。この座標のデータは、マイコン 4 2 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。

【0073】

加速度センサ 3 7 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 3 7 は、当該加速度センサ 3 7 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

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・デバイセズ株式会社（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。

【0074】

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、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 $\langle a_x, a_y, a_z \rangle$ として表される。以下の説明では、3次元ベクトルについては、例えば、 $\langle 1, 0, 1 \rangle$ のように、 $\langle X$ 軸成分、Y軸成分、Z軸成分 $\rangle$ という形式で表記する。また、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。

【0075】

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。

【0076】

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。

【0077】

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CPU10）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、1G（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。

【0078】

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 37 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

【0079】

なお、本実施形態では、コントローラ 5 の動きに応じて変化する値を出力するセンサとして、例えば静電容量式の加速度センサを用いることとするが、他の方式の加速度センサを用いるようにしてもよい。

【0080】

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

【0081】

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。

【0082】

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

【0083】

なお、本実施形態では、3 軸周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 軸周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。

40

【0084】

また、本実施形態では、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 55 および 56 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。

【0085】

各ジャイロセンサ 55 および 56 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 軸周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 軸周りの角速度を示すデータを角速度

50

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。

【0086】

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CPU10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。

【0087】

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。

【0088】

次に、本実施形態で想定するゲームの概要について説明する。本実施形態で想定するゲームは、剣劇ゲームである。具体的には、ゲーム画面は、3D仮想ゲーム空間を一人称視点で描画したものが表示される。そして、3D仮想ゲーム空間内においてプレイヤオブジェクトは剣オブジェクトを手にした状態であり、一人称視点のゲーム画面上には、当該剣オブジェクトが描画される。そして、上記コントローラ5を剣に見立て、プレイヤがコントローラ5を把持したまま振ることで、仮想ゲーム空間内の剣オブジェクトを振って楽しむ剣劇ゲームである。

【0089】

本実施形態では、上記入力装置8を剣に見立ててプレイヤに楽しませるために、上記ジャイロセンサユニット7の出力から算出されるコントローラ5の姿勢を仮想ゲーム空間内の剣オブジェクトの姿勢に反映させる。例えば、実空間内でプレイヤが入力装置8を把持した状態で、当該入力装置8を振り上げてから振り下ろす動作を行うと、この動作に連動して、仮想ゲーム空間内の剣オブジェクトも振り上げられ、振り下ろされる様が描画される。ここで、本実施形態における入力装置8の姿勢算出の方法について簡単に説明する。本実施形態では、X軸、Y軸、Z軸という3軸周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)を利用して、入力装置8の姿勢を算出する。より具体的には、当該ジャイロセンサで検出された角速度に基づいて、上記3軸それぞれの空間内での入力装置8の向きを示す3次元ベクトルを算出する。図8は、入力装置8の姿勢を表す3次元ベクトルの概念を説明するための図であり、入力装置8(コントローラ5)の前面(撮像情報演算部74が撮像する光の入射口側)が奥行き方向を向いている状

態での各軸に対応する当該3次元ベクトルの状態を示している（すなわち、基準座標系のX, Y, Z軸の単位ベクトルを示す）。図8において、各ベクトルの値を1～-1の範囲とすると、この状態では、X軸についてのベクトル（以下、X軸ベクトルと呼ぶ）は<1, 0, 0>で示される。また、Y軸についてのベクトル（以下、Y軸ベクトルと呼ぶ）は<0, 1, 0>で示され、Z軸についてのベクトル（以下、Z軸ベクトルと呼ぶ）は<0, 0, 1>で示される。そして、これら3軸それぞれの向きを示すベクトルを合成することで、最終的な入力装置8の姿勢が算出される。換言すれば、入力装置8の姿勢は上記3つの3次元ベクトルで示されることになる（後述する追従姿勢や相対姿勢等も同様）。例えば、図8に示した姿勢は、以下のような姿勢行列で示すこともできる。

【数1】

$$\text{入力装置8の姿勢} = \begin{bmatrix} 1, 0, 0 \\ 0, 1, 0 \\ 0, 0, 1 \end{bmatrix}$$

【0090】

ところで、上記のような剣オブジェクトの姿勢を算出するためには、理論上は、上記ジャイロセンサユニット7の出力から算出される入力装置8の姿勢をそのまま反映すれば良いが、実際は、上記ジャイロセンサ部品の取り付け向きの微妙なズレや、センサのアナログデータが特定のビット数のデジタル値に置き換えられること等が原因で、ジャイロセンサの検出精度には誤差が生じる。そのため、ジャイロセンサユニット7の出力から算出される姿勢をそのまま利用しても、入力装置8の実際の姿勢と仮想空間内の剣オブジェクトの姿勢との間で徐々にズレが生じてしまう。そこで、本実施形態では、以下のような補正処理を実行してこのズレを軽減し、プレイヤーに違和感を与えることを防いでいる。

【0091】

以下、本実施形態における補正処理について説明する。まず、当該補正処理の実行に際し、本実施形態におけるゲームでは、上記剣オブジェクトに「基本姿勢」というものを定義している。当該基本姿勢は、剣オブジェクトの元の位置、ベースとなる位置を示すものであり、人が剣を持って構えている姿勢（正面斜め上向きの状態）を想定したものである。そのため、例えば、本実施形態で想定するゲームを野球ゲームと仮定した場合は、剣オブジェクトの代わりにバットオブジェクトを用い、上向きの状態を基本姿勢として定義する。また、料理ゲームを想定し、剣オブジェクトの代わりにフライパンオブジェクトを用いる場合は、正面向きの状態を基本姿勢とする。本実施形態では、図9に示すような座標系（仮想ゲーム空間）において、剣オブジェクト101の切っ先が水平から5度上向きの状態（Z-Y間の角度が5度）を基本姿勢として定義する。そして、例えば、入力装置8を振り上げてから振り下ろすという一連の動作を行った場合、この入力装置8が当該基本姿勢に近づくような動き（振り下ろし）のときだけ、ジャイロセンサユニット7から得られる入力値を、より基本姿勢に近づくような値に補正してゲーム処理を行う（仮想空間内の剣オブジェクトの姿勢に反映させる）という処理を実行する。

【0092】

上記のように、本実施形態では、剣を振るように入力装置8を振る操作を行った際に、基本姿勢に近づくような動きのときだけ、より基本姿勢に近づけるように、ジャイロセンサユニット7から得られる入力値の補正を行うような処理を実行する。このような補正を行うのは、次のような理由による。例えば、入力装置8を動かしていない状態（止めている状態）であるにもかかわらず入力値の補正が行われると、入力装置8を動かしていないにも関わらず、仮想ゲーム空間内の剣オブジェクトは補正にかかる分だけ動いてしまうという状況となり、プレイヤーに違和感を与えてしまう。また、基本姿勢から遠ざかるような動き（振り上げ）を行っているときに、基本姿勢に近づけるような補正を行うと、プレイヤーが思っているよりも剣オブジェクトが動かない（振り上がらない）という印象を与えてしまう。つまり、入力装置8の反応が悪いというような印象や、違和感、不満感をプレイ

10

20

30

40

50

ヤに与えてしまう。これに対して、基本姿勢に近づくような動きの時に、より基本姿勢に近づくような補正を行った場合は、上記のような違和感よりは、むしろ「やりすぎた（振り下ろしすぎた）」「すごく反応が良い」というような、相手側に責任を求めにくい心証を与え、心理的に人間に違和感を感じさせにくい。このような観点から、本実施形態では、基本姿勢に近づく／遠ざかるという一連の動作（入力装置 8 を剣のように振るという動作）において、基本姿勢に近づく動作のときのみ入力値の補正を行うことで、プレイヤーに違和感を与えずに入力値を補正している。

【0093】

次に、以下の説明（本実施形態）で用いられる各種「姿勢」の概念について簡単に説明する。以下の説明では、上記のような基本姿勢の他、「入力姿勢」、「追従姿勢」、「相
10 対姿勢」という概念を用いる。入力姿勢とは、上述したようなジャイロセンサユニット 7 から算出される入力装置 8 の姿勢を示す。換言すれば、入力装置 8 の姿勢を直接的に示すものであるが、上記のようなジャイロセンサユニット 7 の検出精度の誤差等により、現実には、必ずしも入力装置 8 の姿勢とぴったり一致するとは限らない。

【0094】

次に、追従姿勢とは、本実施形態の処理において随時算出される仮想の姿勢であり、入力姿勢を徐々に追いかけていくような姿勢である。図 10 は、入力姿勢と追従姿勢の基本的な概念を説明するための図である。まず、本実施形態では、1 フレーム（1 / 60 秒）単位で後述するような処理が繰り返し実行される。また、初期状態として、入力姿勢および追従姿勢は共に水平な状態を前提とする。そして、図 10 に示されるように、1 フレーム
20 ム目で初期状態である入力姿勢（の先端）が、2 フレーム目で 90 度振り上げられた状態になったとする。このような場合、追従姿勢（の先端）は、1 ～ 5 フレーム分かけて、徐々に入力姿勢を追いかける（近づいていく）ことになる。

【0095】

但し、本実施形態においては、入力姿勢が追従姿勢に近づくような動き（以下、接近動作と呼ぶ）のときのみ、追従姿勢を入力姿勢に近づける処理を行う。逆に、入力姿勢が追従姿勢から遠ざかるような動き（以下、離反動作と呼ぶ）のときは、追従姿勢は動かさないような処理を行う。また、入力姿勢が静止している状態のときも追従姿勢は動かさないような処理を行う。

つまり、入力姿勢の先端が追従姿勢に向かって移動するような動きのときのみ、追従姿勢を入力姿勢に近づけるような処理を行う。図 11 は、本実施形態における入力姿勢と追従姿勢との関係を説明するための図である。図 11 に示されるように、入力姿勢が 1 フレーム目で初期状態であり、2 フレーム目で 90 度振り上げられた状態になった場合、入力姿勢の先端は、追従姿勢から遠ざかっていることになる。この場合は、追従姿勢は動かさない。その後、3 フレーム目において、入力姿勢が少し振り下ろされたような状態となった場合、入力姿勢の先端は追従姿勢に近づいていることになる。この場合は、追従姿勢の先端を入力姿勢に少し近づける。その後、4 フレーム目において、入力姿勢が水平な状態になったとする。すなわち、追従姿勢を追いついて、追従姿勢から遠ざかるような動きになったとする。この場合、追従姿勢は動かさない。つまり、入力姿勢が追従姿勢に近づいていき、追従姿勢に追いつき、追いついた時点で、追従姿勢の入力姿勢へ近づいていくという動きが停止することになる。更に、4 フレーム目から 5 フレーム目にかけては、入力姿勢が静止している状態である。この場合も、追従姿勢は動かさない。このように、本実施形態においては、入力姿勢の動きが追従姿勢に近づくような動きの時だけ、追従姿勢も入力姿勢に近づけるという処理を行う。

【0096】

次に、相対姿勢とは、上記入力姿勢と追従姿勢とに基づいて算出される仮想的な姿勢である。より具体的には、相対姿勢とは、追従姿勢から見た入力姿勢、すなわち、追従姿勢を座標軸の基準とした場合の入力姿勢を示す。図 12 は、相対姿勢の基本概念を示す図である。まず、図 12（a）に示すような入力姿勢と追従姿勢があるとする。そして、図 12（b）に示すように、これらの姿勢の原点を合わせた状態で、追従姿勢が水平になる位
50

置まで一緒に回転させる。つまり、追従姿勢を基準とした座標系に入力姿勢を変換する。そして、図12(c)に示すように、当該変換後の入力姿勢を相対姿勢として扱う。基本的には、当該相対姿勢を剣オブジェクトの姿勢としてゲーム処理で用いることで、補正のかけられた入力値を用いた処理が実行可能である。つまり、図12(d)に示すように、入力姿勢と相対姿勢とのなす角度分だけ、入力値に補正がかかっていることになる。但し、本実施形態では、後述するように、当該相対姿勢に対して更なる補正を行う。

【0097】

なお、本実施形態において、上記のような追従姿勢や相対姿勢という概念を用いるのは、ゲーム処理における他の処理（剣オブジェクトを動かす事以外の処理）で上記入力姿勢を利用する場合を考慮したものである。すなわち、本実施形態で説明する処理において上記入力姿勢を直接補正すると、当該他の処理において影響が及ぶため、入力姿勢そのものについては直接手を加えないようにしたものである。

【0098】

以下、入力装置8（仮想ゲーム空間内での剣オブジェクト101）の先端が正面を向いた状態から入力装置8を90度上に振り上げ、再度先端が正面に向く位置まで振り下ろすという一連の動作を行う場合を例として、本実施形態での補正処理の処理概要を説明する。

【0099】

本実施形態では、次のような一連の処理が1フレーム毎に繰り返し行われる。すなわち、入力姿勢等の算出処理、入力姿勢が接近動作をしているか否かの判定処理、接近動作の場合は追従姿勢を入力姿勢に近づける処理、相対姿勢の算出処理、そして、上記相対姿勢と基本姿勢との関係に基づく最終的な補正処理が実行される。

【0100】

より具体的には、まず、直前のフレームにおける処理ループでの入力姿勢と現在のフレームの処理ループでの入力姿勢に基づいて入力姿勢の移動ベクトルを算出する。更に、追従姿勢の移動すべき方向を示すベクトルも算出する。そして、両ベクトルを比較して、入力姿勢の移動ベクトルが追従姿勢に近づくような動きベクトルか否かを判定する。図13および図14は、当該処理の概念を説明するための図である。図13は、入力姿勢が接近動作を行っている例を示している。図13においては、現在のフレームで算出された入力姿勢 $Gdir$ と、直前のフレームでの処理ループにおける入力姿勢である Old と、追従姿勢 $Fdir$ （直前の処理ループにおける状態である）とが示されている。このような状況において、まず、入力姿勢 $Gdir$ と Old とに基づいて入力姿勢の移動ベクトル $\langle vx, vy, vz \rangle$ を算出する。次に、追従姿勢 $Fdir$ が入力姿勢に向かう（近づく）ベクトルである追従ベクトル $\langle nx, ny, nz \rangle$ を算出する。そして、図13のような状況で移動ベクトルと追従ベクトルとの内積を算出すると、マイナスの値が算出されることになる。この場合は、入力姿勢は追従姿勢に近づいていると判定する。

【0101】

一方、図14は、入力姿勢が離反動作の例を示している。図14のような状況においても同様に移動ベクトル $\langle vx, vy, vz \rangle$ と追従ベクトル $\langle nx, ny, nz \rangle$ との内積を算出すると、プラスの値が算出されることになる。この場合は、入力姿勢は追従姿勢から遠ざかっていると判定する。つまり、移動ベクトルと追従ベクトルとの移動方向が同じ方向であるか、互いに反対方向であるかを判定することで、接近動作か離反動作であるかを判定する。

【0102】

上記のような判定を行った結果、現在のフレームにおける入力姿勢の動きが接近動作の時は、追従姿勢を入力姿勢に近づける処理が実行される。一方、離反動作の時は、追従姿勢は動かさない。

【0103】

続いて、追従姿勢と入力姿勢に基づいて、上述したような相対姿勢（図12参照）を算出する。上記接近動作の場合は、上記のように追従姿勢を入力姿勢に近づけたうえで相対

10

20

30

40

50

姿勢の算出が行われることになるため、離反動作の場合における相対姿勢に比べると、少し追従姿勢に近づいた姿勢として算出されることになる。つまり、入力装置 8 を振り下ろしているときのみ、より基本姿勢に近づくような補正がかけられることになる。

【0104】

そして最後に、上記相対姿勢に対して上記基本姿勢に基づく最終的な補正を行う（以下、当該補正のことをオフセット補正と呼ぶ）。図 9 に示したような座標系において、上記のような処理を繰り返すと、最終的には相対姿勢は真正面（Z-Y 間の角度が 0 度の状態）の位置に収束する。その一方で、基本姿勢として上記のように 5 度上向きの状態を定義しているため、最終的に基本姿勢の 5 度分だけ相対姿勢をより戻すようにオフセット補正する（そのため、仮に基本姿勢を水平な状態（Z-Y 間の角度が 0 度の状態）として定義していれば、当該オフセット補正は必ずしも行う必要はない）。そして、当該オフセット補正を行った姿勢が最終的に剣オブジェクト 101 の姿勢として用いられることになる。

10

【0105】

このように、本実施形態では、接近動作のときのみ補正を行うようにすることで、プレイに違和感を与えることなく入力姿勢の補正を行うことが可能となる。

【0106】

次に、本実施形態の補正処理実行時に外部メインメモリ 12 に格納されるデータについて説明する。図 15 は、図 2 に示した外部メインメモリ 12 のメモリマップを示す図である。図 15 において、外部メインメモリ 12 は、プログラム記憶領域 121 およびデータ記憶領域 123 を含む。プログラム記憶領域 121 およびデータ記憶領域 123 のデータは、光ディスク 4 に予め記憶されたものが、外部メインメモリ 12 にコピーされたものである。

20

【0107】

プログラム記憶領域 121 は、CPU 10 によって実行されるゲームプログラムを記憶し、このゲームプログラムは、メイン処理プログラム 122 などによって構成される。

【0108】

データ記憶領域 123 には、角速度データ 124、入力姿勢データ 128、旧入力姿勢データ 132、追従姿勢データ 136、相対姿勢データ 140、基本姿勢データ 144、移動ベクトル 148、追従ベクトル 149、最終姿勢データ 150 などのデータが記憶される。

30

【0109】

角速度データ 124 は、コントローラ 5 を経由してジャイロセンサユニット 7 から送信されてくる角速度データであり、得られた角速度データを所定フレーム分（例えば、ゲーム処理間隔である 1 フレーム（1/60 秒）に対して 30 フレーム分）格納する。角速度データ 124 には、X 軸、Y 軸、および Z 軸の 3 軸成分に分けてそれぞれ検出した X 軸方向角速度 125、Y 軸方向角速度 126、および Z 軸方向角速度 127 が含まれる。なお、ゲーム装置 3 に備える無線コントローラモジュール 19 は、コントローラ 5 から所定間隔例えば 5 ms ごとに送信される操作情報に含まれる角速度データ 124 を受信し、無線コントローラモジュール 19 に備える図示しないバッファに蓄えられる。その後、ゲーム処理間隔である 1 フレーム毎に読み出されて外部メインメモリ 12 に記憶される。

40

【0110】

入力姿勢データ 128 は、上記角速度データ 124 に基づいて算出される入力装置 8 の姿勢を示すためのデータであり、上記図 13 で示した Gdir に相当する情報である。入力姿勢データ 128 は、入力 X 軸ベクトル 129、入力 Y 軸ベクトル 130、入力 Z 軸ベクトル 131 で構成されている。これらのベクトルは、入力装置 8 の姿勢を示す 3 軸それぞれの空間内での向きを示す 3 次元ベクトルである（上記図 8 参照）。以下の説明においては、例えば、入力 X 軸ベクトル 129 については <Gdir Xx, Gdir Xy, Gdir Xz> と示すこともある。同様に、入力 Y 軸ベクトル 130 については、<Gdir Yx, Gdir Yy, Gdir Yz> と示し、入力 Z 軸ベクトル 131 については、<Gdir Zx, Gdir Zy, Gdir Zz> と示すこともある。

50

【0111】

旧入力姿勢データ132は、直前のフレームでの処理ループにおける入力姿勢を示すデータであり、上記図13で示したOldに相当する情報である。旧入力姿勢データ132は、旧入力X軸ベクトル133、旧入力Y軸ベクトル134、旧入力Z軸ベクトル135で構成されている。以下の説明においては、例えば、旧入力X軸ベクトル133については $\langle OldXx, OldXy, OldXz \rangle$ と示すこともある。同様に、旧入力Y軸ベクトル134については、 $\langle OldYx, OldYy, OldYz \rangle$ と示し、旧入力Z軸ベクトル135については、 $\langle OldZx, OldZy, OldZz \rangle$ と示すこともある。

【0112】

追従姿勢データ136は、上述したような追従姿勢を示すためのデータであり、上記図13で示したFdirに相当する情報である。追従姿勢データ136は、追従X軸ベクトル137、追従Y軸ベクトル138、追従Z軸ベクトル139で構成されている。これらのベクトルは、上記入力姿勢の場合と同様に、追従姿勢を構成する3軸のそれぞれの向きを示すためのベクトルである。なお、以下の説明においては、追従姿勢をFdirと示すこともある。また、追従X軸ベクトル137については、 $\langle FdirXx, FdirXy, FdirXz \rangle$ のように示し、追従Y軸ベクトル138については、 $\langle FdirYx, FdirYy, FdirYz \rangle$ 、追従Z軸ベクトル139については、 $\langle FdirZx, FdirZy, FdirZz \rangle$ と示すこともある。

【0113】

相対姿勢データ140は、上述したような相対姿勢を示すためのデータであり、相対X軸ベクトル141、相対Y軸ベクトル142、相対Z軸ベクトル143で構成されている。これらのベクトルは、上記入力姿勢の場合と同様に、相対姿勢を構成する3軸のそれぞれの向きを示すためのベクトルである。なお、以下の説明においては、相対姿勢をRdirと示すこともある。また、相対X軸ベクトル141については、 $\langle RdirXx, RdirXy, RdirXz \rangle$ のように示し、相対Y軸ベクトル142については、 $\langle RdirYx, RdirYy, RdirYz \rangle$ 、相対Z軸ベクトル143については、 $\langle RdirZx, RdirZy, RdirZz \rangle$ と示すこともある。

【0114】

基本姿勢データ144は、上述したような基本姿勢（図9参照）を示すためのデータであり、基本X軸ベクトル145、基本Y軸ベクトル146、基本Z軸ベクトル147で構成されている。これらのベクトルは、上記入力姿勢の場合と同様に、基本姿勢を構成する3軸のそれぞれの向きを示すためのベクトルである。なお、以下の説明においては、基本姿勢をBdirと示すこともある。また、基本X軸ベクトル145については、 $\langle BdirXx, BdirXy, BdirXz \rangle$ のように示し、基本Y軸ベクトル146については、 $\langle BdirYx, BdirYy, BdirYz \rangle$ 、基本Z軸ベクトル147については、 $\langle BdirZx, BdirZy, BdirZz \rangle$ と示すこともある。

【0115】

移動ベクトル148は、1フレームの間に入力姿勢がどの程度移動したか（コントローラ5がどれだけ動かされたか）を示すためのベクトルデータである。上記図13で示したようなベクトル $\langle vx, vy, vz \rangle$ に相当する。

【0116】

追従ベクトル149は、追従姿勢が移動すべき方向（入力姿勢を追従する方向）を示すためのベクトルデータである。上記図13で示したようなベクトル $\langle nx, ny, nz \rangle$ に相当する。

【0117】

最終姿勢データ150は、最終的に剣オブジェクト101に反映させる姿勢を示すデータである。すなわち、補正後の姿勢を示すデータであり、最終X軸ベクトル151、最終Y軸ベクトル152、最終Z軸ベクトル153で構成されている。これらのベクトルは、上記入力姿勢の場合と同様に、基本姿勢を構成する3軸のそれぞれの向きを示すためのベクトルである。なお、以下の説明においては、最終姿勢をSdirと示すこともある。ま

10

20

30

40

50

た、最終X軸ベクトル151については、 $\langle SdirXx, SdirXy, SdirXz \rangle$ のように示し、最終Y軸ベクトル152については、 $\langle SdirYx, SdirYy, SdirYz \rangle$ 、最終Z軸ベクトル153については、 $\langle SdirZx, SdirZy, SdirZz \rangle$ と示すこともある。

【0118】

次に、図16～図19を参照して、ゲーム装置3によって実行される補正処理について説明する。図16は、ゲーム装置3によって実行される補正処理の全体処理を示すフローチャートである。

【0119】

まず、ステップS1において、以降の処理において用いられる各種データの初期化処理が実行される。具体的には、CPU10は、入力姿勢が正面向きになるように入力姿勢データ128を初期化する。そして、追従姿勢データ136を入力姿勢データ128と一致する姿勢となるように設定する。つまり、ゲーム開始直後の入力装置8の入力姿勢を検出し、追従姿勢を入力姿勢に一致させておく処理が実行される。その他、各種変数等の初期化も行われる。以降、ステップS2～S5の処理ループが1フレーム毎に繰り返されることによって、ゲームが進行していく。

【0120】

次に、ステップS2において、入力姿勢の算出が行われる。すなわち、CPU10は、入力装置8から取得された角速度データ124に基づいて、上記入力X軸ベクトル129、入力Y軸ベクトル130、入力Z軸ベクトル131を算出することで入力姿勢データ128を算出して外部メインメモリ12に格納する。なお、ジャイロセンサの出力に基づく姿勢算出の基本概念や基本的な算出手法等については当業者には既知であるため、本処理の詳しい説明は省略する。

【0121】

次に、ステップS3において、追従姿勢算出処理が実行される。図17は、当該ステップS3で示した追従姿勢算出処理の詳細を示すフローチャートである。図17において、まず、ステップS11で、追従姿勢を構成する3軸成分のうち、X軸を移動させる処理のための処理である、追従姿勢X軸移動処理が実行される。

【0122】

図18は、当該追従姿勢X軸移動処理の詳細を示すフローチャートである。まず、ステップS21において、上記入力X軸ベクトル129についての移動ベクトル $\langle vx, vy, vz \rangle$ を算出する処理が実行される。より具体的には、CPU10は、次のような数式を用いて、直前のフレームから現在のフレームまでの入力X軸ベクトル129の移動ベクトル $\langle vx, vy, vz \rangle$ を算出する。また、後述の処理で用いるための変数であるspeedも算出する。当該変数speedは、追従姿勢が入力姿勢に近づく早さ（つまり、追従姿勢の移動速度）を示すための情報である。

【数2】

$$vx = GdirXx - OldXx$$

$$vy = GdirXy - OldXy$$

$$vz = GdirXz - OldXz$$

$$speed = \sqrt{vx^2 + vy^2 + vz^2}$$

【0123】

続くステップS22において、以下の数式を用いて、追従X軸ベクトル137についての追従ベクトル $\langle nx, ny, nz \rangle$ が算出される。

10

20

30

40

【数 3】

$$nx = GdirXx - FdirXx$$

$$ny = GdirXy - FdirXy$$

$$nz = GdirXz - FdirXz$$

【0 1 2 4】

続くステップ S 2 3 で、上記移動ベクトルと追従ベクトルとの内積 a が以下の式を用いて算出される。

【数 4】

$$a = nx \times vx + ny \times vy + nz \times vz$$

10

【0 1 2 5】

次に、ステップ S 2 4 において、入力 X 軸ベクトル 1 2 9 が追従 X 軸ベクトル 1 3 7 に近づいているか否か（接近動作であるか否か）が判定される。すなわち、ステップ S 2 3 で算出した内積 a が 0 より小さいか否かが判定される。当該判定の結果、0 以上のときは（ステップ S 2 4 で NO）、入力 X 軸ベクトル 1 2 9 は離反動作を行っている、あるいは、静止状態であることになるため、処理が後述するステップ S 2 9 に進められる。

【0 1 2 6】

一方、0 未満のときは（ステップ S 2 4 で YES）、接近動作が行われていると考えられるため、追従 X 軸ベクトル 1 3 7 を入力 X 軸ベクトル 1 2 9 に近づけるための処理が実行される。具体的には、まず、追従姿勢 $Fdir$ を回転運動させるための準備（すなわち、移動させるための準備）として、ステップ S 2 5 において、追従ベクトル $\langle nx, ny, nz \rangle$ を追従姿勢 $Fdir$ と直角な成分のみにする処理が実行される。そのために、次のような数式を用いて、上記平行な成分を取り除く処理が実行される。

20

【数 5】

$$a = FdirXx \times nx + FdirXy \times ny + FdirXz \times nz$$

$$nx = nx - a \times FdirXx$$

$$ny = ny - a \times FdirXy$$

$$nz = nz - a \times FdirXz$$

30

【0 1 2 7】

続くステップ S 2 6 で、以下の数式を用いてベクトルの長さを算出する処理が実行される。つまり、追従 X 軸ベクトル 1 3 7 を移動させるべき距離が算出される。

【数 6】

$$a = \sqrt{nx^2 + ny^2 + nz^2}$$

【0 1 2 8】

次のステップ S 2 7 で、算出された長さが 0 であるか否かが判定される。つまり、追従 X 軸ベクトル 1 3 7 を動かす必要があるか否かが判定される。長さが 0 の場合は（ステップ S 2 7 で YES）、追従 X 軸ベクトル 1 3 7 と入力 X 軸ベクトル 1 2 9 が一致している状態であるため、動かす必要がないことになる。そのため、処理が後述するステップ S 2 9 に進められる。一方、長さが 0 ではないときは（ステップ S 2 7 で NO）、以下の数式を用いて、追従 X 軸ベクトル 1 3 7 を入力 X 軸ベクトル 1 2 9 に近づける処理が実行される。

40

【数 7】

$$b = speed \times C3 \div a$$

$$FdirXx = FdirXx + b \times nx$$

$$FdirXy = FdirXy + b \times ny$$

$$FdirXz = FdirXz + b \times nz$$

ここで、変数 b は回転割合を示す。また、係数 $C3$ は、追従 X 軸ベクトル 137 を入力 X 軸ベクトル 137 に近づける強さを示す係数であり、 $(0 < C3 \leq 1)$ の範囲の値である。つまり、この処理の結果、 $Fdir$ を追従ベクトル分だけ一気に移動させるのではなく、回転割合 b の値ずつ移動させることが可能となる。

10

【0129】

次に、ステップ S29 で、現在の入力 X 軸ベクトル $\langle GdirXx, GdirXy, GdirXz \rangle$ を旧入力 X 軸ベクトル 133 として外部メインメモリ 12 に保存する処理が実行される。以上で、追従姿勢 X 軸移動処理が終了する。

【0130】

図 17 に戻り、続いて、ステップ S12 において、追従 Y 軸ベクトルについての移動処理が実行される。この処理は、図 18 を用いて上述したのと同様の処理を、Y 軸を対象として実行するものである。具体的には、上述した各数式等において、X 軸ベクトルについての X, Y, Z 軸成分（例えば $\langle GdirXx, GdirXy, GdirXz \rangle$ ）を用いていたところを、Y 軸ベクトルについての X, Y, Z 軸成分 $\langle GdirYx, GdirYy, GdirYz \rangle$ に置き換えて、上述したような処理を行う。そのため、処理の詳細説明は省略する。

20

【0131】

次に、ステップ S13 において、追従 Z 軸ベクトルについての移動処理が実行される。この処理も、図 18 を用いて上述したのと同様の処理を、Z 軸を対象として実行するものである。具体的には、上述した各数式等において、X 軸ベクトルについての X, Y, Z 軸成分を用いていたところを、Z 軸ベクトルについての X, Y, Z 軸成分 $\langle GdirZx, GdirZy, GdirZz \rangle$ に置き換えて、上述したような処理を行う。そのため、処理の詳細説明は省略する。以上の処理を行うことで、移動後の追従姿勢 $Fdir$ が算出される。

30

【0132】

次に、ステップ S14 において、上記の処理で算出された追従姿勢 $Fdir$ が直交正規化される。以上で、追従姿勢算出処理は終了する。

【0133】

図 16 に戻り、次に、ステップ S4 において、相対姿勢算出処理が実行される。図 19 は、上記ステップ S4 で示した相対姿勢算出処理の詳細を示すフローチャートである。図 19 において、まず、ステップ S41 において、上記入力姿勢 $Gdir$ と追従姿勢 $Fdir$ に基づいて、上記相対姿勢 $Rdir$ が算出される。具体的には、以下の数式を用いて、相対姿勢 $Rdir$ が算出される。

40

【数 8】

$$RdirXx = FdirXx \times GdirXx + FdirXy \times GdirXy + FdirXz \times GdirXz$$

$$RdirXy = FdirYx \times GdirXx + FdirYy \times GdirXy + FdirYz \times GdirXz$$

$$RdirXz = FdirZx \times GdirXx + FdirZy \times GdirXy + FdirZz \times GdirXz$$

$$RdirYx = FdirXx \times GdirYx + FdirXy \times GdirYy + FdirXz \times GdirYz$$

$$RdirYy = FdirYx \times GdirYx + FdirYy \times GdirYy + FdirYz \times GdirYz$$

$$RdirYz = FdirZx \times GdirYx + FdirZy \times GdirYy + FdirZz \times GdirYz$$

10

$$RdirZx = FdirXx \times GdirZx + FdirXy \times GdirZy + FdirXz \times GdirZz$$

$$RdirZy = FdirYx \times GdirZx + FdirYy \times GdirZy + FdirYz \times GdirZz$$

$$RdirZz = FdirZx \times GdirZx + FdirZy \times GdirZy + FdirZz \times GdirZz$$

【0134】

次に、ステップ S 4 2 において、基本姿勢 B d i r と相対姿勢 R d i r とに基づいて、上述したようなオフセット補正（基本姿勢の位置に近づける補正）を行うための処理が実行される。具体的には、以下の数式を用いて上述したようなオフセット補正が実行される。

20

【数 9】

$$SdirXx = BdirXx \times RdirXx + BdirYx \times RdirXy + BdirZx \times RdirXz$$

$$SdirXy = BdirXy \times RdirXx + BdirYy \times RdirXy + BdirZy \times RdirXz$$

$$SdirXz = BdirXz \times RdirXx + BdirYz \times RdirXy + BdirZz \times RdirXz$$

$$SdirYx = BdirXx \times RdirYx + BdirYx \times RdirYy + BdirZx \times RdirYz$$

$$SdirYy = BdirXy \times RdirYx + BdirYy \times RdirYy + BdirZy \times RdirYz$$

$$SdirYz = BdirXz \times RdirYx + BdirYz \times RdirYy + BdirZz \times RdirYz$$

30

$$SdirZx = BdirXx \times RdirZx + BdirYx \times RdirZy + BdirZx \times RdirZz$$

$$SdirZy = BdirXy \times RdirZx + BdirYy \times RdirZy + BdirZy \times RdirZz$$

$$SdirZz = BdirXz \times RdirZx + BdirYz \times RdirZy + BdirZz \times RdirZz$$

このような処理が実行されることで、最終的に剣オブジェクト 1 0 1 に反映する姿勢を示す最終姿勢 S D i r （最終姿勢データ 1 5 0）が求められることになる。以上で、相対姿勢算出処理は終了する。

40

【0135】

図 1 6 に戻り、相対姿勢算出が終われば、ステップ S 5 において、ゲーム処理が実行される。すなわち、算出された最終姿勢 S d i r で示される姿勢で剣オブジェクト 1 0 1 を描画する処理が実行される。また、その他のゲーム処理も適宜実行される。ステップ S 5 の後、ステップ S 6 において、ゲーム終了か否かが判断され、Y E S の場合、ゲーム処理を終了し、N O の場合、ステップ S 2 に戻って、ゲーム処理が繰り返される。以上で、本実施形態にかかる補正処理の説明を終了する。

【0136】

このように、本実施形態では、入力姿勢の動きが接近動作となっているときのみ、追従姿勢を入力姿勢に近づけてから最終的な姿勢の算出を行っている。つまり、入力姿勢が接

50

近動作のときのみ補正を行っている。これにより、プレイヤーに違和感を与えることなく入力姿勢の補正を行うことが可能となる。また、ジャイロセンサのみで入力姿勢を算出するような場合であっても、精度の高い入力姿勢を算出することが可能となる。また、ジャイロセンサのみで算出処理を実行することで、入力装置にかかる費用効率面で有利となる。

【0137】

なお、上記実施形態では、上記補正処理以外の他のゲーム処理に及ぼす影響を考慮して、上記のような追従姿勢や相対姿勢という概念を用い、入力姿勢（入力姿勢データ128）については直接的に変更を加えないようにしていた。これに限らず、他のゲーム処理に影響を及ぼさないことを前提とすれば、上記追従姿勢や相対姿勢を用いずに、上記基本姿勢と入力姿勢との関係のみで処理するようにしてもよい。すなわち、入力姿勢が基本姿勢に接近動作をしているか否かを判定する。この判定の結果、接近動作である場合のみ、当該入力姿勢をより基本姿勢に近づけるように直接的に入力姿勢データ128を補正する。そして、当該補正後の入力姿勢データ128に基づいて剣オブジェクト101の姿勢を変更するようにしてもよい。

【0138】

また、入力姿勢の算出に関し、上記実施形態では、X軸、Y軸、Z軸という3軸それぞれの空間内でのコントローラ5の向きを示す3次元ベクトルを算出することで入力姿勢を算出する場合を例にしたが、これに限らず、コントローラ5の入力姿勢が算出できれば、どのような処理方法を用いても良い。そして、いずれかの処理方法で算出された入力姿勢の動作が、仮想的な姿勢である上記基本姿勢あるいは追従姿勢に対しての接近動作か離反動作かを判定し、接近動作の場合にのみ入力姿勢を補正する処理をおこなうようにすればよい。

【0139】

（第2の実施形態）

次に、図20から図24を参照して、本発明の第2の実施形態について説明する。第2の実施形態では、上述の第1の実施形態で行ったような補正処理に加え、更にコントローラ8の振り方を判定し、これに基づいて、コントローラ8の向いている方向に関する補正を実行する処理を行う。より具体的には、プレイヤーによるコントローラ8の振り始めから振り終わりまでの一連の振り操作（以下、一振り分操作と呼ぶ）を検出し、その振り方が「縦振り（上から下に振り下ろす、または、下から上へ振り上げるような振り方）」であれば、コントローラ8の向いている方向がゲーム内においては正面方向となるように補正する処理を実行する。

【0140】

例えば、図20に示すように、プレイヤーがモニタ2を対向した状態でコントローラを振っている状態が正面を向いている状態とする。そして、図21に示すように、モニタ2に対してプレイヤーが90度右を向いてコントローラ8を振っているような場合、縦振り操作が行われたタイミングで、ゲーム処理においては、テレビ2のほうを向いている、つまり、正面を向いてコントローラ8を振っているものとして扱うような補正を実行する。より具体的には、図21のような状態で、縦振り操作が検出されたときに、そのときのプレイヤーの向いている方向（以下、前方方向と呼ぶ）と、正面方向（図20の状態）との角度のズレを算出する。そして、剣オブジェクト101に、そのズレ角度の分だけ正面方向に向けてY軸周りの回転を加えて描画するという処理を実行する。

【0141】

これは、例えば、ゲーム画面として真正面に敵オブジェクトが表示されていることを前提とすると、コントローラ8を振っている内に、計算誤差の蓄積などにより、プレイヤーが気付かない間に剣オブジェクト101の切っ先が斜め前に向いてしまう状態があり得る。あるいは、プレイヤー自身の体の向き（コントローラ8の先端の向き）が多少斜めを向いてしまうような場合もあり得る（極端な例は、図21に示したような状態である）。この場合、厳密には剣オブジェクト101が真正面の敵オブジェクトに当たらないような状態となり得る。そこで、このような状態となっても、以下に説明するような処理を実行するこ

とで剣オブジェクト101の向きを正面向きに補正し、真正面に表示される敵オブジェクト等に剣オブジェクト101を当てるようにするものである。

【0142】

ここで、縦振り操作が検出されたときに補正を行うのは、横振り操作の場合は正面方向を正確に算出することが困難であり、その結果、正面からのズレ角度を正確に算出することが困難であるためである。

【0143】

なお、当該実施形態に係るゲーム装置3は、上述した第1の実施形態と同様であるため、同一の参照符号を付して詳細な説明を省略する。

【0144】

次に、第2の実施形態にかかる処理の概要を説明する。第2の実施形態では、コントローラ8が振られている間（一振り分操作の間）、上述の第1の実施形態で説明したような補正が行われた後の剣オブジェクト101の姿勢を示すデータ（上記最終姿勢データ150）をフレーム毎にバッファに蓄積する。そして、一振り分操作が終わったことが検出されたときに、蓄積されたデータを解析して、一振り分操作の内容が「縦振り操作」にどの程度近い操作であったかを示す、信頼度係数を算出する。当該信頼度係数は、真っ直ぐ縦に振った場合は「1」という値となり、真横に振った場合は「0」という値になる係数である。更に、上記前方方向を示す前方ベクトルを算出し、正面方向を示す正面ベクトルとの角度の差を算出する。そして、当該角度に基づいて、剣オブジェクト101の姿勢をY軸周りに回転させることで剣オブジェクトの姿勢を補正する。但し、このとき、上記信頼度係数に応じて補正する角度を調整する。すなわち、信頼度係数が小さいときは、補正する幅も小さくするような調整を行う。また、本実施形態では、一度に上記補正角度分の補正を行わずに、数フレームかけて徐々に補正していく。例えば、ズレ角度が40度あった場合は、1フレーム毎に10度ずつ補正し、4フレーム分の処理をかけて補正するような処理を行う。これは、一度で補正を行ってしまうことでプレイヤに違和感を感じさせることを防ぐためである（換言すれば、補正を行っていることをプレイヤに気付かせないようにするためである）。

【0145】

次に、第2の実施形態で用いられるデータについて説明する。図22は、第2の実施形態にかかる外部メインメモリ12のメモリマップを示す図である。図22において、第2の実施形態での外部メインメモリ12には、第1の実施形態で説明した各種データに加えて、姿勢バッファ201、回転軸ベクトル202、正面ベクトル203、真上ベクトル204、前方ベクトル205、信頼度係数206、第1バッファデータ207、第2バッファデータ208、Z軸差分ベクトル209、外積データ210、カウンタ211、総合補正角度212、今回補正角度213が記憶される。

【0146】

姿勢バッファ201には、上記一振り分操作にかかるフレーム毎の剣オブジェクトの姿勢を示すデータ（第1の実施形態において1フレーム毎に算出された最終姿勢データ150）が時系列で記憶される。以下の説明では、姿勢バッファ201内に格納される各フレームの姿勢データをbuf[n]と示す（[n]は0から始まる整数であり、配列の添字の役割を果たす。また、数字が小さいほど古いデータであることを示す）。

【0147】

回転軸ベクトル202は、一振り分操作における回転軸を示すベクトルである。例えば、図8に示したような座標系で、真上から真下にコントローラ8を振りおろした場合は、コントローラ8の先端がX軸を中心に回転するような動作になるため、X軸方向のベクトルが回転軸ベクトルに相当する。

【0148】

正面ベクトル203は、コントローラ8が正面を向いた状態を定義した単位ベクトルであり、本実施形態では<0, 0, 1>であるとする。

【0149】

真上ベクトル 204 は、真上方向を示すための単位ベクトルであり、本実施形態では、上記図 8 の状態における Y 軸ベクトル $\langle 0, 1, 0 \rangle$ であるとする。

【0150】

前方ベクトルは、コントローラ 8 の先端が向いている方向（プレイヤーが向いている方向）を示すためのベクトルである。例えば、図 20 の状態では、前方ベクトルは $\langle 0, 0, 1 \rangle$ で示され、図 21 のような、右を向いた状態であれば $\langle -1, 0, 0 \rangle$ で示される。

【0151】

信頼度係数 206 は、上記のように、一振り分操作がどの程度「縦振り操作」に近いかを示す情報である。当該信頼度係数は、1～0 の範囲の値を採り、後述する処理においては、コントローラ 8 を真っ直ぐ縦に振った場合（つまり、縦振り操作）は "1" という値となり、真横に振った場合は "0" という値になる。

10

【0152】

第 1 バッファデータ 207、第 2 バッファデータ 208、Z 軸差分ベクトル 209 および外積データ 210 は、以下の処理において、上記姿勢バッファ 201 に格納された姿勢データを解析するために用いられる変数である。また、カウンタ 211 は、後述する処理においてカウンタとして用いられる変数である。

【0153】

総合補正角度 212 および今回補正角度 213 は、上述したような補正すべき角度を示す。上記のように、本実施形態では、一度に上記補正角度分の補正を行わずに、数フレームかけて徐々に補正していく。そして、今回補正角度 213 は、総合補正角度 212 で示される角度の内、1 フレーム分の処理において補正すべき角度を示す。

20

【0154】

以下、第 2 の実施形態にかかる処理の詳細を説明する。図 23 は、第 2 の実施形態にかかる処理の全体処理を示すフローチャートである。図 23 において、まず、ステップ S51 において、第 1 次姿勢算出処理が実行される。この処理では、上記第 1 の実施形態で説明したような処理が実行される。すなわち、ジャイロセンサからのデータに基づいて入力姿勢を算出し、上記追従姿勢や相対姿勢、基本姿勢を用いて上述したような補正が行われた後の剣オブジェクト 101 の姿勢（上記最終姿勢データ 150）がこの処理で算出される。

【0155】

30

次に、ステップ S52 において、コントローラ 8 が振られているか否かの判定が実行される。すなわち、一振り分操作の途中であるか、コントローラ 8 を振っていない状態（プレイヤーが手を止めている状態）であるかが判定される。これは、例えば、角速度データ 124 の X-Y 成分の長さが所定の閾値を超えているか否かで判定すればよい。

【0156】

ステップ S52 の判定の結果、コントローラ 8 が一振り分操作中であると判定されたときは（ステップ S52 で YES）、次に、ステップ S53 において、直前のフレームでの処理ループにおいても、一振り分操作中であったか否かが判定される。すなわち、一振り分操作が開始された直後であるのか、一振り分操作が継続中の状態であるのかが判定される。この判定に際しては、上記ステップ S52 の判定の結果を外部メインメモリ 12 にフレーム毎に記憶するようにしておき、この判定結果を参照することで一振り分操作の継続中の判定を行えばよい。

40

【0157】

当該判定の結果、一振り分操作が開始された直後のときは（ステップ S53 で NO）、ステップ S54 において、姿勢バッファ 201 がクリアされる。一方、直前のフレームにおける処理ループにおいて一振り分操作が行われていたときは（ステップ S53 で YES）、ステップ S54 の処理は行わずに、ステップ S55 に処理が進められる。

【0158】

次に、ステップ S55 において、上記ステップ S51 で算出された最終姿勢データ 150 が姿勢バッファ 201 に格納される。その後、後述するステップ S58 へ処理が進めら

50

れる。

【0159】

一方、上記ステップS52の判定の結果、コントローラ8の一振り分操作中ではないときは（ステップS52でNO）、ステップS56において、直前のフレームにおける処理ループにおいて、一振り分操作が行われていたか否かが判定される。つまり、一振り分操作が終わったか、一振り分操作が行われていなかったかが判定される。当該判定の結果、一振り分操作が行われていないと判定されたときは（ステップS56でNO）、後述するステップS58に処理が進められる。一方、直前のフレームにおける処理ループにおいて、一振り分操作が行われていた（一振り分操作が終わった）と判定されたときは（ステップS56でYES）、続くステップS57において、補正角度算出処理が実行される。

10

【0160】

図24は、上記ステップS57で示した補正角度算出処理の詳細を示すフローチャートである。本処理では、姿勢バッファ201に溜まっている姿勢データを解析し、剣オブジェクト101の姿勢（向いている方向）を補正するための角度を算出する処理が実行される。図24において、まず、ステップS71において、カウンタ211に“0”が設定される。また、このとき、姿勢バッファ201内に記憶された姿勢データの数（以下、姿勢データ数と呼ぶ）が算出される。

【0161】

次に、ステップS72において、回転軸ベクトル202が<0, 0, 0>に初期化される。

20

【0162】

次に、ステップS73において、姿勢バッファ201内の最初のデータであるbuf[0]が第1バッファデータ207として設定される。続く、ステップS74において、カウンタ211に1が加算される。

【0163】

続くステップS75において、カウンタ211が上記ステップS71で算出された姿勢データ数以上か否かが判定される。つまり、姿勢バッファ201内のデータを全てについて以下に述べるような処理が行われたか否かが判定される。当該判定の結果、カウンタ211が上記姿勢データ数未満のときは（ステップS75でNO）、次に、ステップS76において、姿勢バッファ201中から上記カウンタ211で示される姿勢データ、すなわち、buf[カウンタ211]が取得され、第2バッファデータ208として設定される。例えば、当該処理が最初に実行されたときは、buf[1]で示される姿勢データが第2バッファデータ208として設定される。

30

【0164】

次に、ステップS77において、以下の数式を用いて、Z軸差分ベクトル209が算出される。

Z軸差分ベクトル＝第2バッファデータのZ軸ベクトル－第1バッファデータのZ軸ベクトル

【0165】

次に、ステップS78において、以下の数式を用いて、回転軸ベクトル202が算出される。

40

外積データ＝Z軸差分ベクトル×第2バッファデータのZ軸ベクトル

回転軸ベクトル＝回転軸ベクトル＋外積データ

すなわち、上記Z軸差分ベクトルと第2バッファデータのZ軸ベクトルとの外積を算出することで、上記2フレーム分の姿勢にかかる回転軸が算出される。たとえば、本処理が最初に実行されたときは、buf[0]とbuf[1]で示される姿勢データの変化における回転軸が算出される。そして、当該算出された回転軸が上記回転軸ベクトル202に加算される。

【0166】

次に、ステップS79において、次の処理ループでの処理のために、第2バッファデー

50

タ 208 が第 1 バッファデータ 207 として設定される。そして、上記ステップ S 74 の処理に戻り、処理が繰り返される。

【0167】

一方、上記ステップ S 75 の判定の結果、カウンタ 211 が上記ステップ S 71 で算出された姿勢データ数以上と判定されたときは（ステップ S 75 で YES）、姿勢バッファ 201 内のデータ全てについて上記のような回転軸の算出及び回転軸ベクトル 202 への加算処理が行われたことになる。このときは、ステップ S 80 において、回転軸ベクトル 202 の正規化が実行される。

【0168】

次に、ステップ S 81 において、前方ベクトル 205 が算出される。これは、回転軸ベクトル 202 と上記真上ベクトル 204 との外積を求めることで算出される。例えば、上記図 8 に示すような座標系を例にすると、真っ直ぐ縦振り操作を行った場合は、コントローラ 8 の姿勢の変化は X 軸周りを回転するような変化となる。すなわち、回転軸ベクトルは X 軸方向のベクトルになり、真上方向を示す Y 軸ベクトルとの外積を求めると、前方ベクトル 205 として Z 軸方向のベクトルが算出されることになる。

【0169】

次に、ステップ S 82 において、上記信頼度係数 206 が算出される。これは、前方ベクトル 205 の長さに所定の定数 α を乗じることで求められる。

信頼度係数 = 前方ベクトルの長さ $\times$ 定数 α

上記のように、信頼度係数 206 は、コントローラ 8 の振り方が縦振り操作にどれだけ近かったかを示す係数、いわば「縦振り度」であり、上記前方ベクトル 205 の長さが “1” に近いほど、縦振り度が大きいことになる。例えば、コントローラ 8 を真っ直ぐ縦に振り下ろした動きであれば、上記のように回転軸は X 軸方向のベクトルとなり、前方ベクトル 205 の長さは 1 となるが、仮に真横に振ったとすると、回転軸は Y 軸方向のベクトルとなり、前方ベクトル 205 の長さは “0” となる。

【0170】

次に、ステップ S 83 において、補正角度の更新処理が実行される。具体的には、以下のような処理が実行される。まず、CPU 10 は、前方ベクトル 205 の逆余弦を算出することで、正面ベクトル 203 と前方ベクトル 205 との角度差（すなわち、正面からのズレ角度）を算出する。次に、CPU 10 は、前方ベクトル 205 の X 軸成分が “0” より大きいか否かを判定し、“0” より大きい場合、すなわち、コントローラ 8 の先端が左向き（左寄り）の場合は、上記算出したズレ角度の符号（プラス／マイナス）を反転する。つまり、コントローラ 8 が正面から右向きにずれているか左向きにずれているかを判定して、いずれの場合も、補正する方向が正面方向になるように調整される（例えば、上記逆余弦の算出の結果、角度差が 45 度と算出されたとき、コントローラ 8 が左向きなら -45 度、右向きなら +45 度補正する）。更に、CPU 10 は、以下の式を用いて、信頼度係数 206 が小さい場合は、補正する幅も小さくするような調整を加えることで、総合補正角度 212 を算出する。

総合補正角度 = 算出した補正角度 $\times$ 信頼度係数

以上で、補正角度算出処理は終了する。

【0171】

図 23 に戻り、次に、ステップ S 58 において、今回補正角度 213 が算出される。上記のように、本実施形態では、一度に総合補正角度 212 分の補正を行わずに、数フレームかけて徐々に補正していくため、現在のフレームでの処理でどの程度補正するのかを算出する。具体的には、以下の式を用いて、総合補正角度 212 の数割だけ補正するような調整が行われる。

今回補正角度 = 総合補正角度 $\times$ 0.03

総合補正角度 = 総合補正角度 - 今回補正角度

【0172】

次に、ステップ S 59 において、上記今回補正角度 213 で示される分だけ、剣オブジ

ェクト101の姿勢がY軸回りに回転される。そして、回転後の姿勢が反映された剣オブジェクト101がゲーム画面として描画される。以上で、第2の実施形態にかかる処理が終了する。

【0173】

このように、第2の実施形態では、縦振り操作が行われたときに、コントローラ8の向きが正面向きとなるような補正を行う。これにより、プレイヤーに違和感を与えることなく、入力姿勢について補正を行い、ゲーム処理に反映させることが可能となる。

【0174】

なお、上記第2の実施形態では、第1の実施形態で説明したような補正処理が実行された後の姿勢を更に補正する場合を例にしたが、第1の実施形態にかかる補正処理は行わずに、入力姿勢に対して上記のような向きの補正のみを行うようにしてもよい。

10

【0175】

また、上記各実施形態では、ジャイロセンサからの出力に基づいてコントローラ5の姿勢を算出する場合を例としていたが、例えば加速度センサからの出力に基づいてコントローラ5の姿勢を算出する場合にも本発明は適用可能である。もちろん、加速度センサとジャイロセンサの双方の出力を用いてコントローラ5の姿勢を算出する場合であっても適用可能であることはいうまでもない。

【産業上の利用可能性】

【0176】

本発明にかかる情報処理プログラムおよび情報処理装置は、ユーザに違和感を感じさせることなく、検出された入力装置の姿勢を補正して情報処理に反映することができ、ゲーム装置等に有用である。

20

【図面の簡単な説明】

【0177】

【図1】ゲームシステム1の外観図

【図2】ゲーム装置3の構成を示すブロック図

【図3】入力装置8の外観構成を示す斜視図

【図4】コントローラ5の外観構成を示す斜視図

【図5】コントローラ5の内部構造を示す図

【図6】コントローラ5の内部構造を示す図

30

【図7】入力装置8の構成を示すブロック図

【図8】入力装置8の姿勢を表す3次元ベクトルの概念を説明するための図

【図9】基本姿勢を説明するための図

【図10】入力姿勢と追従姿勢の基本的な概念を説明するための図

【図11】本実施形態における入力姿勢と追従姿勢との関係を説明するための図

【図12】相対姿勢の基本概念を示す図

【図13】接近動作か否かを判定する処理の概念を説明するための図

【図14】接近動作か否かを判定する処理の概念を説明するための図

【図15】外部メインメモリ12のメモリマップを示す図

【図16】ゲーム装置3によって実行される補正処理の全体処理を示すフローチャート

40

【図17】図16のステップS3で示した追従姿勢算出処理の詳細を示すフローチャート

【図18】図17のステップS11で示した追従姿勢X軸移動処理の詳細を示すフローチャート

【図19】図16のステップS4で示した相対姿勢算出処理の詳細を示すフローチャート

【図20】第2の実施形態について説明するための図

【図21】第2の実施形態について説明するための図

【図22】第2の実施形態にかかる外部メインメモリ12のメモリマップを示す図

【図23】第2の実施形態にかかる処理の全体処理を示すフローチャート

【図24】図23のステップS57で示した補正角度算出処理の詳細を示すフローチャート

50

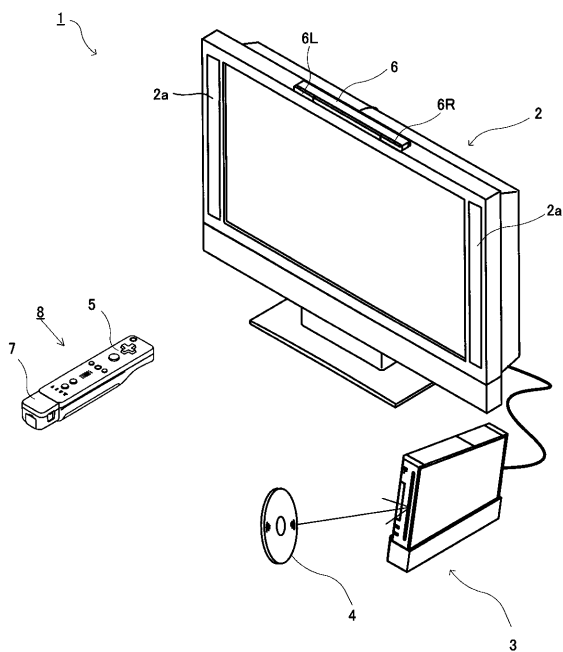
【符号の説明】

【0178】

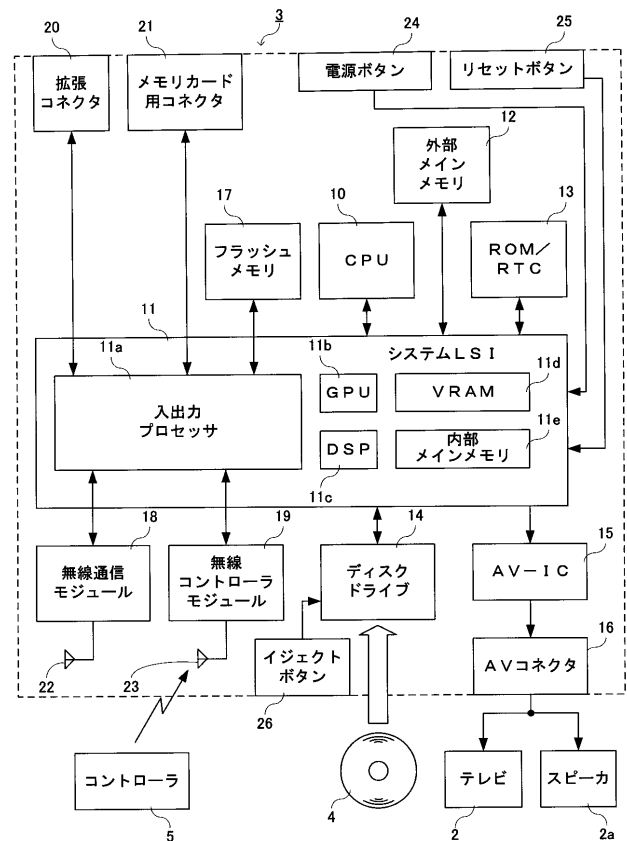
1 …ゲームシステム	
2 …モニタ	
2 a …スピーカ	
3 …ゲーム装置	
4 …光ディスク	
5 …コントローラ	
7 …ジャイロセンサユニット	
8 …入力装置	10
10 …CPU	
11 …システムLSI	
11 a …入出力プロセッサ	
11 b …GPU	
11 c …DSP	
11 d …VRAM	
11 e …内部メインメモリ	
12 …外部メインメモリ	
13 …ROM／RTC	
14 …ディスクドライブ	20
15 …AV-IC	
16 …AVコネクタ	
17 …フラッシュメモリ	
18 …無線通信モジュール	
19 …無線コントローラモジュール	
20 …拡張コネクタ	
21 …外部メモリカード用コネクタ	
22 …アンテナ	
23 …アンテナ	
24 …電源ボタン	30
25 …リセットボタン	
26 …イジェクトボタン	
30 …基板	
31 …ハウジング	
32 …操作部	
33 …コネクタ	
34 …LED	
35 …撮像情報演算部	
36 …通信部	
37 …加速度センサ	40
38 …赤外線フィルタ	
39 …レンズ	
40 …撮像素子	
41 …画像処理回路	
42 …マイコン	
43 …メモリ	
44 …無線モジュール	
45 …アンテナ	
48 …バイプレータ	
49 …スピーカ	50

- 5 3 … プラグ
 5 4 … マイコン
 5 5 … 2 軸 ジャイロセンサ
 5 6 … 1 軸 ジャイロセンサ

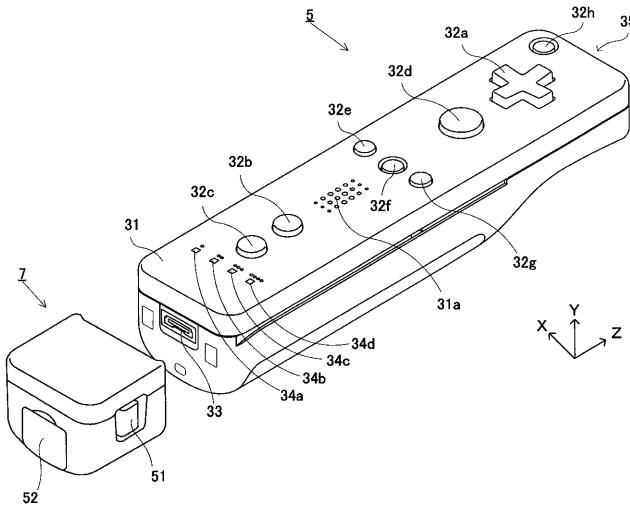
【図 1】



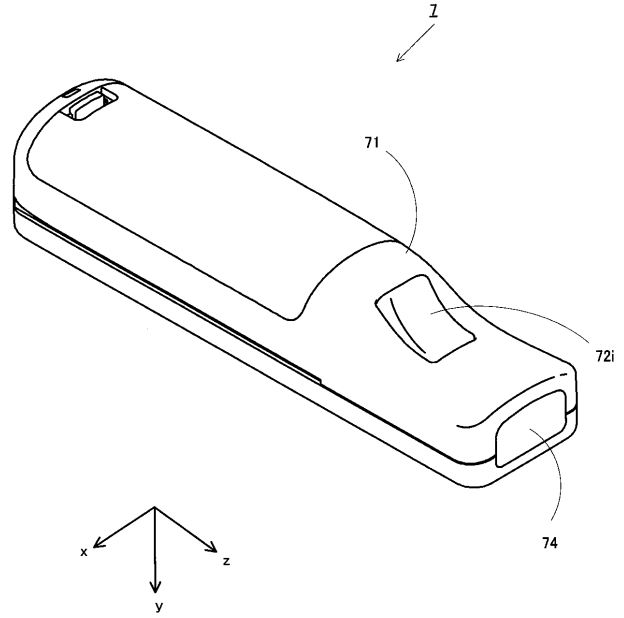
【図 2】



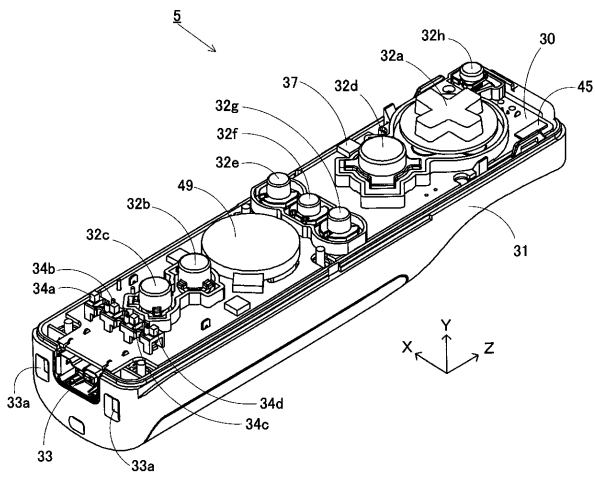
【図 3】



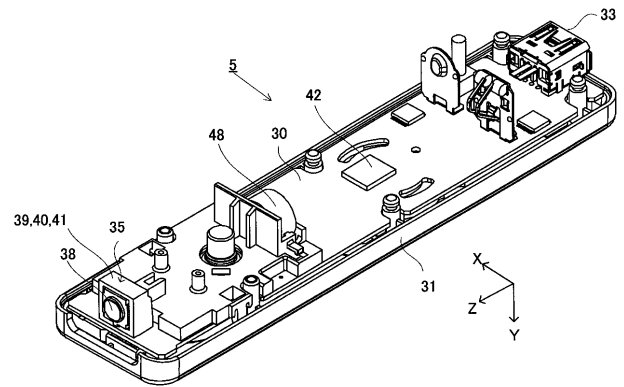
【図 4】



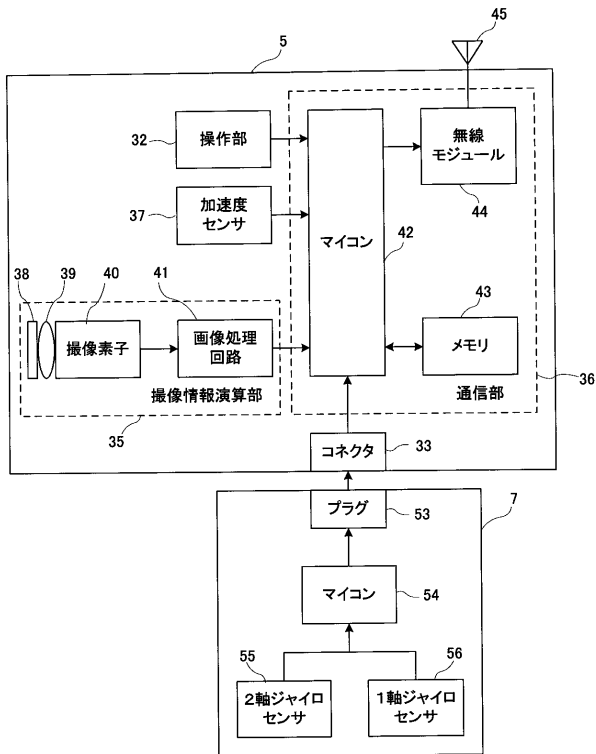
【図 5】



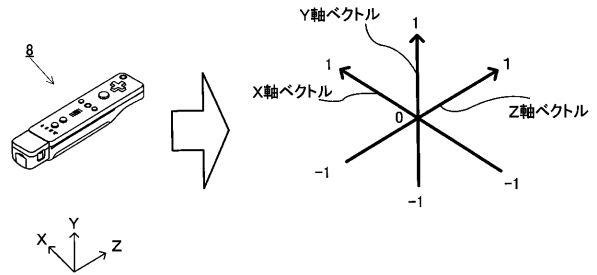
【図 6】



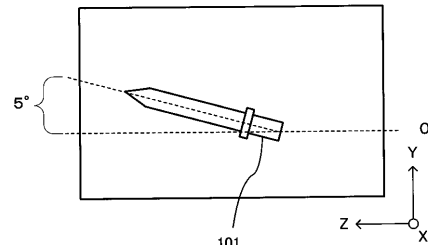
【図 7】



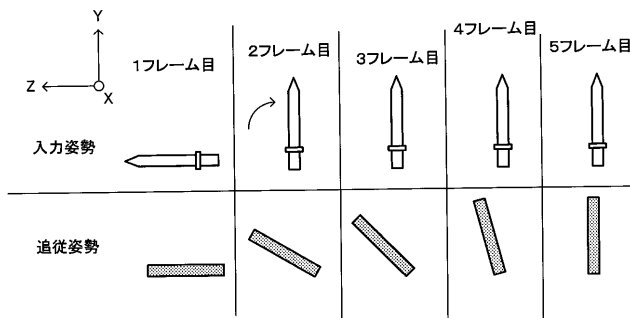
【図 8】



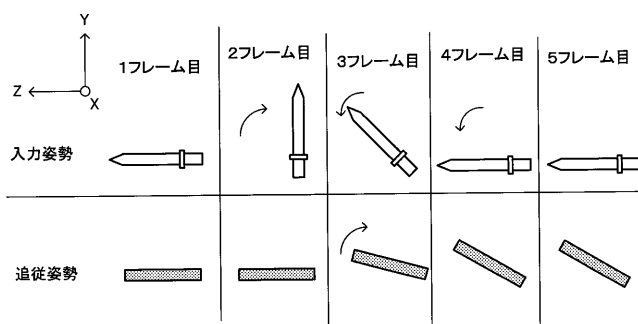
【図 9】



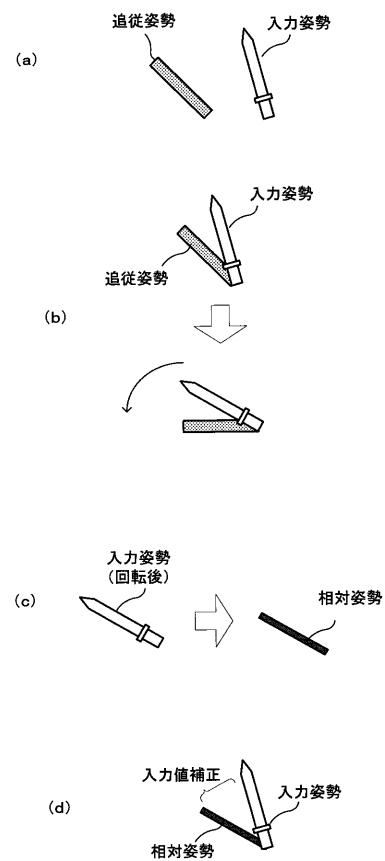
【図 10】



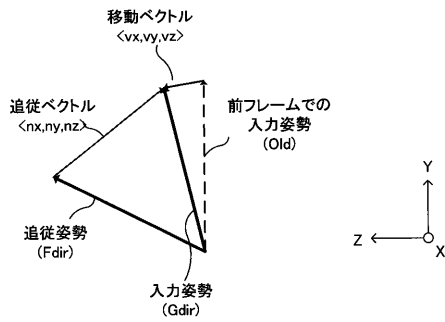
【図 11】



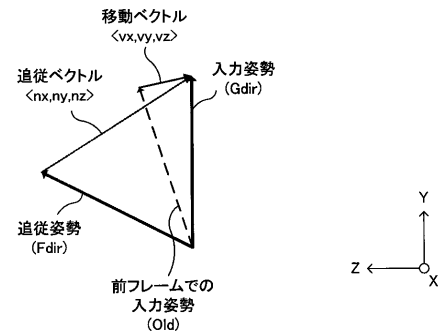
【図 12】



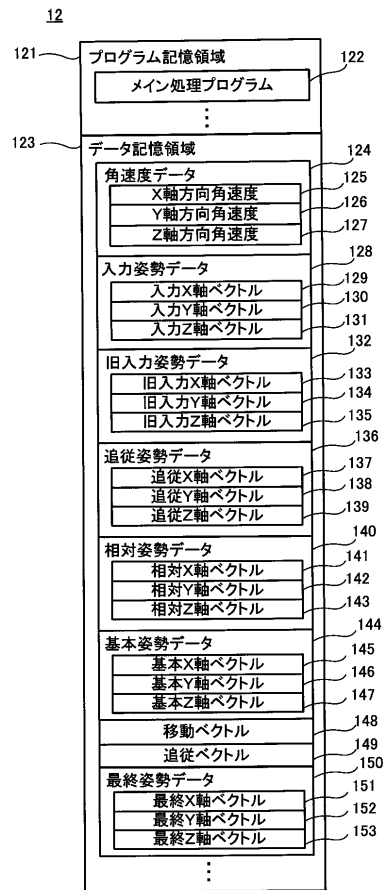
【図 1 3】



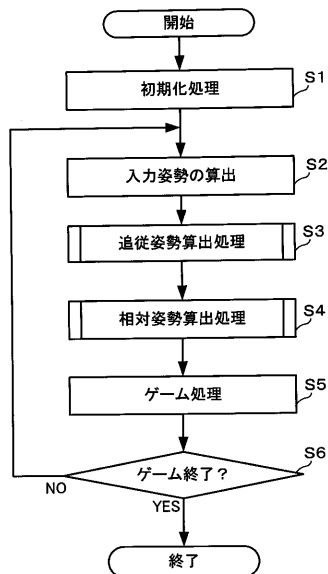
【図 1 4】



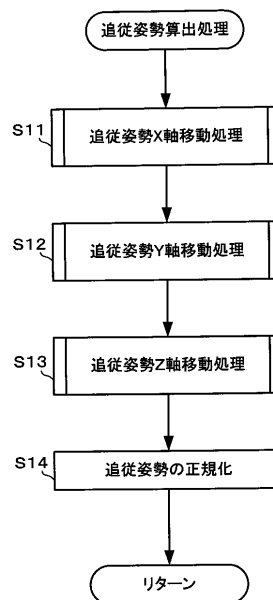
【図 1 5】



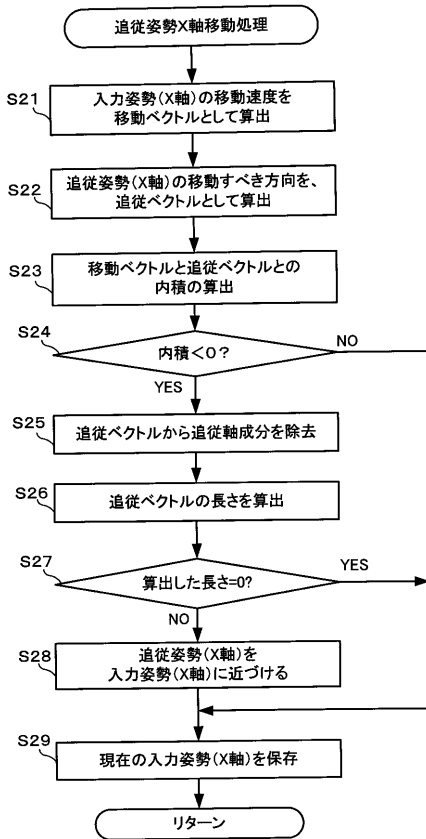
【図 1 6】



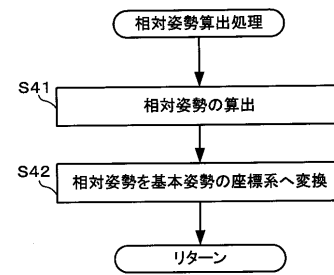
【図 1 7】



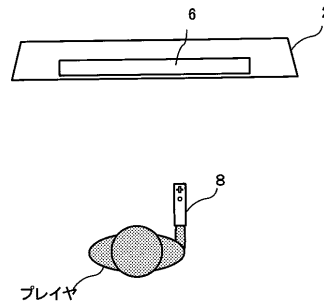
【図 18】



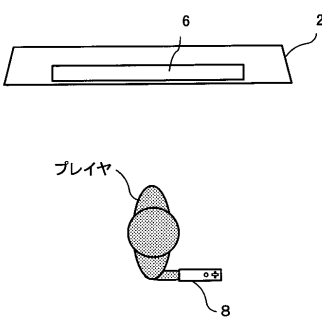
【図 19】



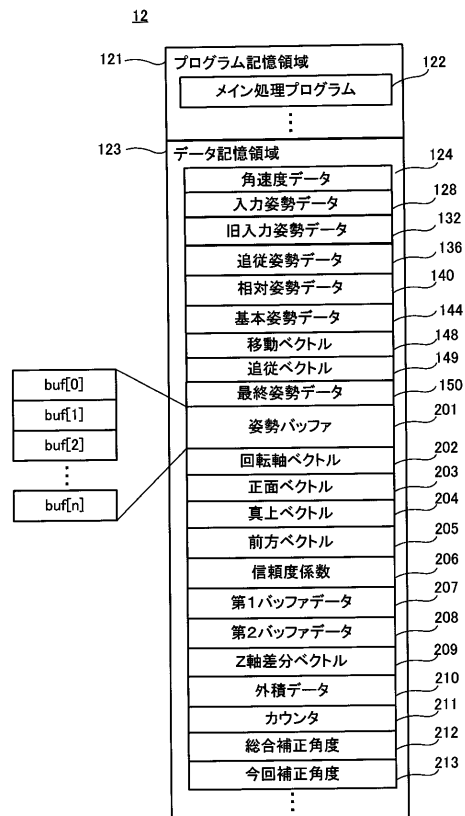
【図 20】



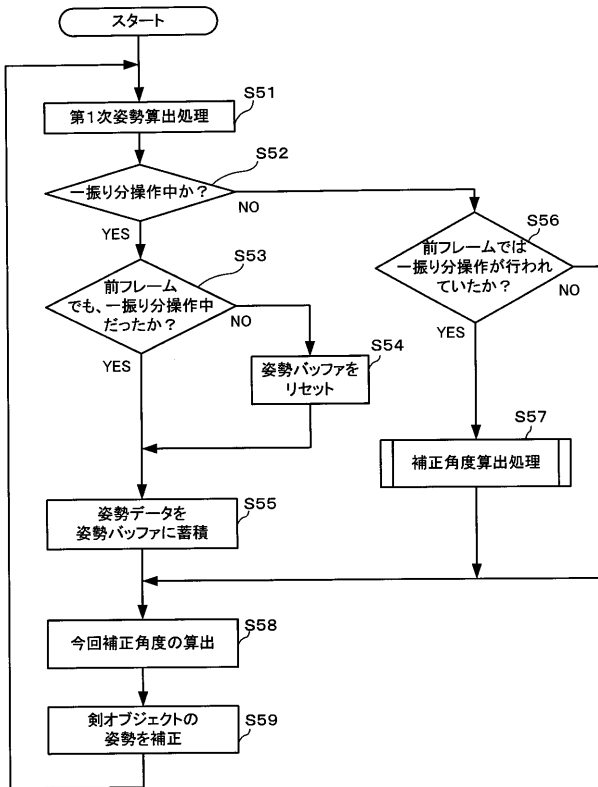
【図 21】



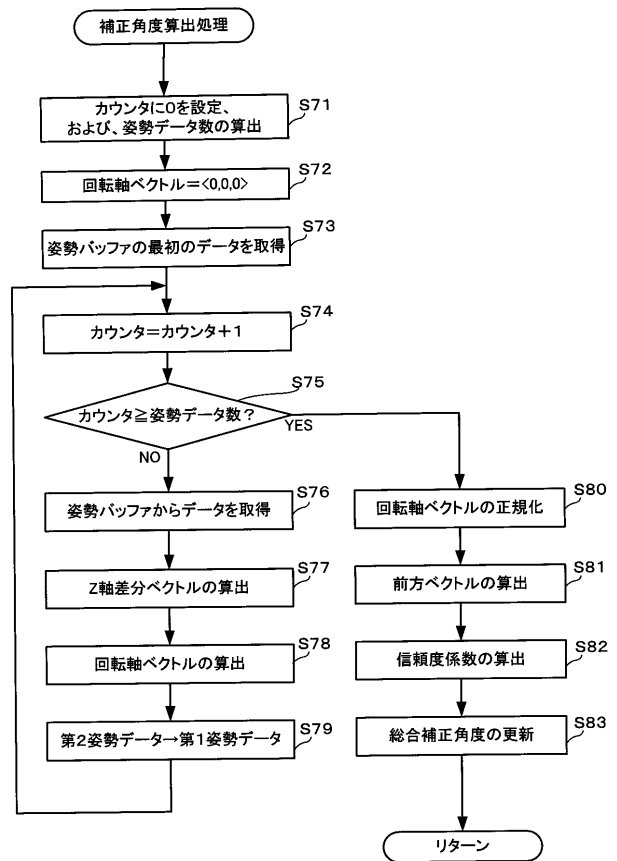
【図 22】



【図 23】



【図 24】



フロントページの続き

Fターム(参考) 2F105 AA10 BB17 CC11
5B087 AA02 AD00 BC26 DD03 DG02