

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4848515号
(P4848515)

(45) 発行日 平成23年12月28日(2011.12.28)

(24) 登録日 平成23年10月28日(2011.10.28)

(51) Int.Cl. F I
G06T 19/00 (2011.01) G06T 17/40 D

請求項の数 6 (全 23 頁)

| | |
|---|--|
| <p>(21) 出願番号 特願2006-162666 (P2006-162666) (22) 出願日 平成18年6月12日(2006.6.12) (65) 公開番号 特開2007-334443 (P2007-334443A) (43) 公開日 平成19年12月27日(2007.12.27) 審査請求日 平成21年3月6日(2009.3.6)</p> | <p>(73) 特許権者 504174135 国立大学法人九州工業大学 福岡県北九州市戸畑区仙水町1番1号 (74) 代理人 100099634 弁理士 平井 安雄 (72) 発明者 尾下 真樹 福岡県飯塚市大字川津680-4 九州工業大学内 審査官 千葉 久博</p> |
|---|--|

最終頁に続く

(54) 【発明の名称】 アバター動作制御システム、そのプログラム及び方法

(57) 【特許請求の範囲】

【請求項1】

モーションキャプチャー装置と、
前記モーションキャプチャー装置から得られた利用者の入力姿勢とアバターの現在の状態から前記アバターの動作を追従制御する第1の追従制御手段と、
前記アバターの動作情報を予め記憶する動作情報記憶手段と、
前記動作情報記憶手段に記憶されている動作情報に基づいて、前記アバターの動作を生成する動作生成手段と、
前記動作生成手段が生成した動作を追従制御する第2の追従制御手段と、
前記アバターの現在の状態に基づいて、当該アバターの現在の状態が前記利用者の入力姿勢のみからは制御できない受動的な状態であり、前記利用者の入力姿勢とは異なる動作を前記アバターに行わせると判断される場合、又は、前記利用者の入力姿勢に基づいて、当該入力姿勢と異なる動作を前記アバターに行わせると判断される場合に、前記アバターに行わせる目標となる動作を特定して前記動作生成手段に制御信号を送ると共に、前記第1の追従制御手段による追従制御と、前記第2の追従制御手段による追従制御との混合比率を決定する制御指令手段と、
前記制御指令手段が決定した混合比率にしたがって、前記アバターの状態を変更する状態変化計算手段とを備え、
前記制御指令手段が、前記目標となる動作の動作変化に応じて、前記混合比率を変化させて決定することを特徴とするアバター動作制御システム。

10

20

【請求項 2】

請求項 1 に記載のアバター制御システムにおいて、
前記制御指令手段が、
前記アバターの動作状態が前記利用者の入力姿勢による制御の困難性を示す段階が大きくなるにしたがって、又は、前記入力姿勢と前記アバターに行わせる目標となる動作とが異なる度合いが大きくなるにしたがって、前記第 2 の追従制御手段が追従制御する割合を大きくすることを特徴とするアバター制御システム。

【請求項 3】

請求項 1 又は 2 に記載のアバター制御システムにおいて、
前記制御指令手段が、
前記アバターの身体の部位ごとに前記混合比率を異ならせて決定することを特徴とするアバター動作制御システム。

10

【請求項 4】

請求項 1 ないし 3 のいずれかに記載のアバター制御システムにおいて、
前記制御指令手段が、
前記目標となる動作が、前記アバターへの外的な作用に対するリアクション動作である場合に、前記目標となる動作として前記リアクション動作を特定し、前記動作生成手段に前記特定されたリアクション動作に関する制御信号を送ると共に、前記アバターが受けた外的な作用に応じた前記アバターの動作状態に対応する混合比率を決定することを特徴とするアバター制御システム。

20

【請求項 5】

請求項 1 ないし 3 のいずれかに記載のアバター制御システムにおいて、
前記制御指令手段が、
前記目標となる動作が、前記アバターを移動させる動作である場合に、前記目標となる動作として前記移動させる動作を特定し、前記動作生成手段に前記特定された移動させる動作に関する制御信号を送ると共に、前記目標となる動作の動作変化に応じて前記混合比率を決定することを特徴とするアバター制御システム。

【請求項 6】

モーションキャプチャー装置から得られた利用者の入力姿勢とアバターの現在の状態から前記アバターの動作を追従制御する第 1 の追従制御手段、
前記アバターの動作情報を予め記憶する動作情報記憶手段、
前記動作情報記憶手段に記憶されている動作情報に基づいて、前記アバターの動作を生成する動作生成手段、

30

前記動作生成手段が生成した動作を追従制御する第 2 の追従制御手段、
前記アバターの現在の状態に基づいて、当該アバターの現在の状態が前記利用者の入力姿勢のみからは制御できない受動的な状態であり、前記利用者の入力姿勢とは異なる動作を前記アバターに行わせると判断される場合、又は、前記利用者の入力姿勢に基づいて、当該入力姿勢と異なる動作を前記アバターに行わせると判断される場合に、前記アバターに行わせる目標となる動作を特定して前記動作生成手段に制御信号を送ると共に、前記第 1 の追従制御手段による追従制御と、前記第 2 の追従制御手段による追従制御との混合比率を決定する制御指令手段、

40

前記制御指令手段が決定した混合比率にしたがって、前記アバターの状態を変更する状態変化計算手段としてコンピュータを機能させ、

前記制御指令手段が、前記目標となる動作の動作変化に応じて、前記混合比率を変化させて決定することを特徴とするアバター動作制御プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、3次元空間内での利用者の動きを測定し、コンピュータに取り込むモーションキャプチャー装置を入力装置として利用し、仮想空間内に描画されるアバター（利用者

50

が操作する仮想人間)のアバター動作制御に関する。

【背景技術】

【0002】

モーションキャプチャー技術は色々な領域で用いられており、例えば、映画、動作分析、ゲームで使用されている。現在は、これらのオフライン向けに主に利用されているが、コンピュータゲームなどにおける操作インターフェースとしての期待も大きい。

【0003】

モーションキャプチャー装置を入力装置として用いることで、利用者は仮想空間内でのアバター(利用者が操作する仮想人間)の全身の動作を制御し、さまざまな動作を行わせることが可能となる。

10

【0004】

発明者は、このようなアプリケーションにおいては、第三者視点を利用することが最適であると考え(図16参照)。現在、多くの仮想現実アプリケーションでは、まるで利用者が仮想空間内にいるかのように感じる本人視点が用いられている。しかし、キャラクターの動きが重要となる格闘ゲーム等のアプリケーションにおいて、本人視点は適切ではない。現実世界では、我々人間は、感覚を通じて自分の体の動きや、他のキャラクターや物体との物理的な接触を感じることができる。しかしながら、現在の仮想現実アプリケーションでは、視覚的な情報や限られた触覚デバイスしか用いることができないため、そのよう感触を利用者が感じることはできない。そこで、利用者自身をアバターとして見る第三者視点を用いることで、利用者は動きと他のオブジェクトとの可視的な相互作用を感じることができる。これが、発明者が、第三者視点がエンターテインメントアプリケーションにより適しているとする理由である。

20

【0005】

そのようなアプリケーションを実現するために解決すべき問題として、2つの主な問題がある。1つ目の問題は、アバターはモーションキャプチャー装置からの入力にもとづき制御されるため、例えばアバターが他のキャラクターと衝突したときのリアクション動作など、環境からアバターへの物理的相互作用を実現することは難しい、ということである。これは、仮想空間内でアバターに物理的な影響が加えられても、利用者にもその力を作用させることはできないからである。また、2つ目の問題として、仮想環境と比べモーションキャプチャーの領域は狭いため利用者が歩き回るとは困難である、という問題がある。現在の仮想現実システムでは、ナビゲーション向けのゲームパッド、3Dポインティングインターフェースのような追加的な装置が用いることでこのような問題を解決しているが、発明者が想定しているようなモーションキャプチャー装置を使用して全身動作を制御するシステムでは、このような追加的な装置を使用することは難しい。

30

【0006】

ところで、モーションキャプチャーを入力装置としたキャラクター制御については、いくつかの研究がある(非特許文献1)。しかしながら、多くのシステムでは、入力姿勢をそのままキャラクターの動作の制御に用いており、キャラクターと環境の物理的相互作用は考慮されていない。

これに対し、非特許文献2で本発明に関連するシステムが提案されている。つまり、第

40

【0007】

また、非特許文献3では、マーシャルアーツの格闘ゲームにおいて、横から見た第三者視点を用いている。モーションキャプチャー装置から取得した利用者の姿勢をもとにアバターを制御し、コンピュータグラフィックスを用いてアバターを画面に描画する代わりに、カメラからリアルタイムの取得された利用者の動画を直接画面に描画している。

【非特許文献1】Shin, H.J., Lee, J., Gleicher, M., Shin, S.Y. Computer Puppetry: An Importance-Based Approach, ACM Transactions on Graphics. 20 (2), 67-94, 2001

【非特許文献2】Hasegawa, S., Ishikawa, T., Naoki Hashimoto, N. Human Scale Hapt

50

ic Interaction with a Reactive Virtual Human in a Realtime Physics Simulator. In Proc. of Advances in Computer Entertainment Technology 2005.

【非特許文献3】Hamalainen, P., Ilmonen, T., Hoysniemi, J., Lindholm, M., Nykane n, A. Martial Arts in Artificial Reality, Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 005), 781-790, 2005.

【発明の開示】

【発明が解決しようとする課題】

【0008】

前記非特許文献2のシステムでは第三者視点及び追従制御を用いているものの、入力姿勢に追従するものであり、自動的な反応を生成するものではない。したがって、キャラクターに大きな衝撃が加わった場合には、アバターの動作は不自然になる。

10

【0009】

前記非特許文献3の手法自体はユニークであるが、敵からの打撃に対してアバターが反応することができない。ここで、このシステムは、2つめの課題として述べたモーションキャプチャー領域の問題を、利用者の水平方向の移動を拡大し、垂直方向の動作に強調を加えることで、解決している。しかし、この手法では、モーションキャプチャー領域を多少拡大することはできるが、やはり広い仮想空間を利用者が歩き回るようなことはできない。

つまり、本発明は、前記非特許文献2及び3では解決することができなかった前述した2つの課題を解決する。

20

【0010】

ところで、コンピュータアニメーションにおいて、与えられた衝撃に対するリアクション動作を生成することは最も難しい技術の1つである。人の制御モデルの仕組みはとて複雑であり、まだ十分にモデル化されていないため、自然な人間のリアクション動作を計算機を用いて生成することは大変難しい。

【0011】

これに対して、複数の研究者により、追従制御及び/又は動作データベースを用いる手法が提案されている。

文献1 (Zordan, V. B., Chiu, B., Majkowska, A., Fast, M. Dynamic Response for Motion Capture Animation. ACM Transactions of Graphics (Proc. of SIGGRAPH 2005), 24(3), 697-701, 2005.) では、仮想的なキャラクターに衝撃が与えられた後の動力学シミュレーションの結果にもとづき、データベースから適切な動作データを探索して、動力学シミュレーションの結果と検索された動作データを混合することによって、衝撃に応じた自然なリアクション動作を生成する方法を提案している。

30

【0012】

文献2 (Komura T., Ho E. S.L., Lau R. W.H. Lau. Animating Reactive Motion Using Momentum-based Inverse Kinematics. The Journal of Computer Animation and Virtual Worlds, 16(3-4), 213-223, Wiley, 2005.) では、入力された衝撃情報に応じて適切なリアクション動作データを検索し、衝撃の大きさに応じて動作データを変形した後に、モーメント変化量を考慮しながら動作データを追従制御を行うことで、衝撃に応じた自然なリアクション動作を生成する方法を提案している。

40

【0013】

文献3 (Oshita, M., Makinouchi, A. A Dynamic Motion Control Technique for Human-like Articulated Figures. Computer Graphics Forum (EUROGRAPHICS 2001), 20(3), 192-202, 2001.) では、全身のバランスや関節負荷を考慮しつつ、追従制御の出力を修正するような動的制御アルゴリズムを開発することで、あらかじめ用意されたリアクション動作を使用することなく、自然な動作を生成している。

【0014】

これらの手法は、モーションキャプチャーからの利用者の動作を用いるのではなく、あらかじめ作成されたリアクション動作データを入力として用いて、任意の衝撃を与えたと

50

きのリアクション動作を生成するように設計されている。

【0015】

一方で、本発明で提案する手法は、モーションキャプチャーからの入力姿勢と文献2と文献1に示す同じような手法でデータベースから検索されたリアクション動作にもとづきアバターを制御する。本発明では、検索されたリアクション動作をそのまま目標動作として使用するが、文献2及び文献1に記述されるような手法を本発明に組み合わせて、より自然なリアクション動作を使用することができる。

【0016】

モーションキャプチャー以外の直感的な入力装置を用いる制御インターフェースとして、さまざまなものが開発されている(文献4:Yin, K. K., Pai, D. K. FootSee: an Interactive Animation System. In Proc. of ACM SIGGRAPH / Eurgraphics Symposium on Computer Animation 2003, 329-338, 2003.)。しかしながら、そのようなインターフェースはキャラクターと環境の相互作用が考慮されていない。また、本発明では利用者による全身制御が必要であるため、それらをナビゲーション等の追加的インターフェースとして本発明と同時に用いることも困難である。

【0017】

動作認識(文献5:Emering, L., Boulic, R., Thalmann, D. Live Participant's Action Recognition for Virtual Reality Interactions. In Proc. of Pacific Graphics, 15-21, 1997.)には多数の手法がある。しかしながら、現存の手法はどの種の動作実行がなされているのかを簡単に認識するものである。一方、本発明においては、単に利用者が歩行動作を行っているかという判定を行うだけではなく、歩行速度などを利用者の動作に合わせて制御する必要がある。そのため、本発明では、歩行動作をいくつかの状態に分け、利用者の動きからファジールールを用いて現在の状態を認識することで、状態遷移の速度から歩行速度を計算する手法を用いる。

【0018】

本発明は前記課題を解決するためになされたものであり、モーションキャプチャーを入力として、衝撃等を受けたときのアバターの自然なリアクション動作を実現する、第三者視点を用いた仮想現実アプリケーションのためのアバター動作制御システムを提供することを目的とする。

また、本発明は、限られた利用者の移動範囲の中でその移動範囲より広い仮想空間内を移動することができるアバター動作制御システムを提供することを目的とする。

【課題を解決するための手段】

【0019】

本発明に係るアバター動作制御システムは、モーションキャプチャー装置と、前記モーションキャプチャー装置から得られた利用者の入力姿勢とアバターの現在の状態から前記アバターの動作を追従制御する第1の追従制御手段と、前記アバターの動作情報を予め記憶する動作情報記憶手段と、前記動作情報記憶手段に記憶されている動作情報に基づいて、前記アバターの動作を生成する動作生成手段と、前記動作生成手段が生成した動作を追従制御する第2の追従制御手段と、前記アバターの現在の状態に基づいて、当該アバターの現在の状態が前記利用者の入力姿勢のみからは制御できない受動的な状態であり、前記利用者の入力姿勢とは異なる動作を前記アバターに行わせると判断される場合、又は、前記利用者の入力姿勢に基づいて、当該入力姿勢と異なる動作を前記アバターに行わせると判断される場合に、前記アバターに行わせる目標となる動作を特定して前記動作生成手段に制御信号を送ると共に、前記第1の追従制御手段による追従制御と、前記第2の追従制御手段による追従制御との混合比率を決定する制御指令手段と、前記制御指令手段が決定した混合比率にしたがって、前記アバターの状態を変更する状態変化計算手段とを備え、前記制御指令手段が、前記目標となる動作の動作変化に応じて、前記混合比率を変化させて決定するものである。

【0020】

このように本発明によれば、目標となる動作の動作変化に応じて、第1の追従制御手段

10

20

30

40

50

による追従制御と、第2の追従制御手段による追従制御との混合比率を変化するので、アバターの現在の状態が入力姿勢のみに従うだけでなく、また、アバターの現在の状態が生成動作のみに従うだけでなく、つまり、アバターの現在の状態が入力姿勢だけでなく生成動作にも従うことで、モーションキャプチャー装置によってアバターを動作させ、且つ、生成動作でアバターを動作させ、より現実世界に合致した自然な動作を実現することができる。

【0021】

入力姿勢の追従制御の比率と生成動作の追従制御の比率を合わせて1になる必要は必ずしもない。例えば、比率の和が1よりも小さければ、アバターの追従性は鈍くなり、ゆっくりとした動作になる。

10

【0022】

混合比率の決定に用いる入力姿勢、アバターの現在の状態は、ひとつの入力姿勢、アバターの現在の状態であってもよいし、複数の入力姿勢（現在の入力姿勢、過去の入力姿勢）、アバターの状態（現在のアバターの状態、過去のアバターの状態）であってもよい。複数のものから導き出せる所定部位の速度、加速度であってもよい。

【0023】

制御指令手段は、入力姿勢及び/又はアバターの現在の状態から追従制御の混合比率を決定しているが、入力姿勢、アバターの現在の状態に加え、動作生成手段が生成した動作も参照して追従制御の混合比率を決定する構成であってもよい。つまり、入力姿勢、アバターの現在の状態及び/又は生成動作からアバターに対する入力姿勢の追従制御手段による追従制御と生成動作の追従制御手段による追従制御の混合比率を決定する構成であってもよい。

20

【0025】

前記動作生成手段が動作を生成する場合には、動作データを記録する動作記憶手段から動作データを読み出して動作を生成する。実施形態においては動作記憶手段は動作データベースとして説示している。この動作記憶手段に様々は動作データを記録することで動作生成手段が様々な動作を生成することができる。また、動作記憶手段をデータベースで構築することもでき、動作データの検索性及び格納性に優れ、より迅速に動作生成手段が動作を生成することができる。

【0026】

[発明の下位概念その1]

アバターの状態から所定条件（衝撃を受ける、武器を投げられる、声をかけられる等の受動的動作）に該当することを判別したとき、適切なりアクション動作データを生成し、生成されたりアクション動作に追従制御する。そうすることで、他のキャラクターから攻撃を受けた場合に衝撃に対する適切なりアクション動作をとることができ、前記第1の課題を解決して環境とアバターとの物理的な相互作用を実現する。

30

【0027】

本発明では、モーションキャプチャー装置からの入力動作を単純に再生するのではなく、モーションキャプチャーからの入力姿勢に追従するようにアバターを制御する（図1参照）。ここで、発明の下位概念その1を第1の発明とする。例えば、アバターに衝撃が与えられたとき（例えば、他のキャラクターと衝突したとき）、システムはアバターが転倒するようなりアクション動作データを生成し、アバターはそのリアクション動作に追従する。リアクション動作中でも、利用者は僅かではあるがアバターを制御することができる。本発明により、システムはモーションキャプチャー装置からの入力された利用者の入力姿勢とシステムによって生成されたりアクション動作の間の自然な遷移を実現することができる。

40

【0028】

現実世界では、衝撃を受けたからといって人間はまったく自分の体をコントロールできなくなるのではなく、衝撃に応じて衝撃を受けつつ自分の体をコントロールすることができる。具体的には、ボクサーは相手のボクサーの攻撃を受けて身が反り返りながらも反撃

50

のパンチを出すことができる。ただし、あまりも相手の攻撃が強すぎて空に浮いた場合にはもはや自分の体をほとんどコントロールすることができない。本発明により仮想現実の中でも受動的動作、能動的動作のどちらか一方の動作に陥るのではなく、どちらの動作も含んだより現実世界に合致した動作が可能となる。

【0029】

[発明の下位概念その2]

入力姿勢から、利用者が行っているしぐさが所定条件（例えば、歩きの身ぶり、走るの身ぶり、早歩きの身ぶり、泳ぐ、登る、飛ぶ、よける、しゃがむ、振り返る等の能動的動作）に該当することを判別したとき、適切な動作を生成し、生成動作に追従制御する。そうすることで、例えば、利用者は、歩きの身ぶり、走るの身ぶり、早歩きの身ぶりをする
10
ことで、それぞれ仮想空間内で、アバターに、歩き、走る、早歩きの動作を行わせることができ、第2の課題を解決してモーションキャプチャーの領域よりも広くアバターが移動することができる。このとき、生成動作にアバターを完全に追従させるのではなく、入力姿勢への追従制御の比率を適切に制御することで、例えば、歩きながら手を振ったり等の、生成動作と利用者の動作を同時に実行させることができる。

【0030】

すなわち、本発明は歩行等の身ぶりによるナビゲーションインターフェースも有する（図2参照）。ここで、発明の下位概念その2を第2の発明とする。このインターフェースにより、利用者が足踏みをすると、つまり、水平的な移動なしに一定の場所で歩行の身ぶりをする
20
と、システムが歩行動作を生成し、アバターはその歩行動作に追従する。システムが生成する歩行動作は、利用者の身ぶりによって制御される。例えば、利用者が手足を素早く動かすと、その動作速度にもとづいて、システムは走る動作を生成する。つまり、利用者の歩行のしぐさに応じて、アバターの歩行速度や曲がる方向も制御することができる。本インターフェースを用いることで、利用者は、容易且つ直感的に、アバターの歩行動作の速度や方向を制御することができる。

【0031】

本発明は第三者視点を用いた仮想現実アプリケーションで用いることが望ましい。このように適用することで、利用者は、アバターの動作を通して、特別な力覚装置を用いること
30
なくとも、アバターへ加えられた衝撃などの環境からの物理的な相互作用を視覚的に認識することができる。

なお、発明者は、発明技術を実際に実装したプロトタイプシステムを開発した（図3参照）。

【0032】

本発明は、システム以外に、アバター動作制御プログラム、アバター動作制御装置、アバター動作制御方法としても把握することができる。

これら前記の発明の概要は、本発明に必須となる特徴を列挙したものではなく、これら複数の特徴のサブコンビネーションも発明となり得る。

【発明を実施するための最良の形態】

【0033】

本発明は、多くの異なる形態で実施可能である。従って、下記の実施形態の記載内容のみで解釈すべきではない。また、実施形態の全体を通して同じ要素には同じ符号を付けて
40
いる。

実施形態では、主にシステムについて説明するが、所謂当業者であれば明らかな通り、本発明はコンピュータで使用可能なプログラム及び方法としても実施できる。また、本発明はハードウェア、ソフトウェア、またはソフトウェア及びハードウェアの実施形態で実施可能である。プログラムは、ハードディスク、CD-ROM、DVD-ROM、光記憶装置または磁気記憶装置等の任意のコンピュータ可読媒体に記録できる。さらに、プログラムはネットワークを介した他のコンピュータに記録することができる。

【0034】

[1. システム構成]

10

20

30

40

50

アバター状態は、シミュレーションの各時刻において、モーションキャプチャーからの入力姿勢にもとづき更新される。

アバターは、通常の状態では、モーションキャプチャー装置からの入力姿勢に追従する。

大きな衝撃がアバターに与えられたとき、主にリアクション動作データベースから検索されたリアクション動作に追従する。

利用者が歩行の身ぶりを行ったときには、モーションブレンディング技術を用いることで歩行動作が生成され、アバターはそれに応じて歩行動作に追従する。

リアクション動作又は歩行動作中でも、状況に応じてアバターは入力姿勢にも追従することが可能であるため、利用者はアバターを制御することができる。

10

【0035】

[1.1 追従制御]

文献6 (Hodgins, J. K., Wooten, W. L., Brogan, D. C. and O'Brien, J.F. Animating Human Athletes. SIGGRAPH '95 Proceedings, 71-78, 1995.)、文献7 (Laszlo J., van de Panne M., Fiume E. Limit Cycle Control and Its Application to the Animation of Balancing and Walking. In Proc. of SIGGRAPH 1996, 155-162, 1996.)の標準的な物理的シミュレーションにおいては、追従制御部は関節トルクを演算し、シミュレーションはその関節トルクから関節加速度を演算しているが、発明者は制御性の観点から角加速度を用いることを選択した。

【0036】

20

トルクにもとづく追従制御では、安定した制御を実現するためには、適切なゲイン計数などのパラメータをキャラクターの骨格や目標動作に対して慎重に調整する必要がある。そのため、本発明のように、複数の追従制御部からの出力を混合してアバターの制御を行おうとすると、不安定な動作が生成されることになる。したがって、発明者は文献3、文献8 (Yamane, K., Nakamura, Y. Dynamics Filter - Concept and Implementation of Online Motion Generator for Human Figures. IEEE Transactions on Robotics and Automation, 19(3), 2003.)に記載されている方法と同じやり方で、角加速度を制御することにした。

【0037】

トルクにもとづく制御は一般的により物理的に妥当な動作を生成すると考えられているが、本実施形態のシステムにおいては必要ない。というのは、入力姿勢データとデータベース上のリアクション動作の両方は、実際の人間の動作をモーションキャプチャーしたものであり、それらはもともと物理的に妥当であるからである。文献6や文献7とは異なり、本実施形態では、物理的に正しい動作を生成するためでなく、複数の目標動作にもとづいて動作を生成することを目的として追従制御を用いる。

30

【0038】

[1.2 人体モデル]

アバター、及び、システムが操作するキャラクターなどの仮想人間は、多関節体としてモデル化される。本実施形態の人体モデルは、図4に示すように、20の体節、19の関節を有する多関節体によってモデル化されている。これは、コンピュータゲームで一般的

40

に使用されるものと同様の、典型的な人体モデルである。

人体モデルの姿勢は、次の数式で表現される。

【0039】

【数1】

$$M = \{P_{root}, q_{root}, q_0, \Lambda, q_i, \Lambda, q_{n-1}\} \quad (1)$$

【0040】

人体モデルの姿勢は、腰の位置・方向、各関節の回転からなる。ここで、nは関節の番

50

号である。本実装形態では、腰の方向や関節の回転を、回転ベクトルで示す。回転ベクトルの方向は親の体節に対する相対的な回転の回転軸を示し、回転ベクトルの長さはその回転軸まわりの回転角度を示す。回転ベクトルを用いることで、任意の回転を表現することができる。

人モデルの速度 M' 及び加速度 M'' も、(1) で示したものと同様の形式で示される。

【0041】

本実施形態では、アバター状態は、姿勢 M と速度 M' からなる。追従制御部は、角加速度 M'' を出力する。アバターの姿勢 M と速度 M' の変化は、各シミュレーションステップにおけるアバターの加速度 M'' をもとに、積分計算によって計算される。アバターが、他のキャラクター又は物体と衝突したときには、文献9 (Moore, M., and Wilhelms, J. Collision Detection and Response for Computer Animation. Computer Graphics (SIGGRAPH '88 Proceedings), 22(3), 289-298, 1988.) で述べられている方法により、全体節間の衝撃と速度変化の関係を表す線形問題を解くことで、衝突によるアバターの速度 M' の変化を算出することができる。

10

【0042】

[2 . アバター制御]

本実施形態では、アバターの現在状態と目標動作から得られる目標状態にもとづいて、追従制御のアバターの加速度を求める、PD制御法を使用する。

【0043】

【数2】

20

$$q_i'' = k(q_{target,i} - q_i) + d(q_{target,i}' - q_i') \quad (2)$$

【0044】

ここで、 q_i'' は i 番目の関節の出力回転加速度であり、 q_i 、 q_i' はそれぞれ現在の関節角度、回転速度であり、 $q_{target,i}$ 、 $q_{target,i}'$ はそれぞれ目標動作から得られる目標の関節角度、回転速度である。ゲイン計数 k 、ダンピング計数 d には、それぞれ適切な値を手動で設定している。本実施形態では、追従制御に関節トルクではなく角加速度を用いるので、全ての関節で同じ k 、 d を用いることができる。

30

PD制御部で関節回転を制御するために、回転及び回転速度を回転ベクトルとして扱う。

関節回転に加え、腰及び地面に接触する末端部位(足先、手先など)も制御される。

【0045】

本システムは、腰や末端部位の位置及び方向を制御し、それらの値から、逆運動学(inverse kinematics)をもちいて手足の各関節の回転を計算する。本実施形態では非特許文献1の解析的運動学手法を使用する。

腰(末端部位)の位置 p_j は、次のように制御される。

【0046】

【数3】

40

$$p_j'' = k_{pelvis}(p_{target,j} - p_j) + d_{pelvis}(p_{target,j}' - p_j') \quad (3)$$

前記(2)で示した関節回転の制御と同様の方法で、腰の方向も制御される。

【0047】

[3 . 衝撃時のリアクション動作]

本実施形態では、発明の本質的要素としての、動作生成部におけるリアクション動作生成処理と、制御司令部における入力姿勢及び生成動作の追従制御の混合比率の決定処理を

50

工夫することで、衝撃時のリアクション動作を実現する。

【0048】

本実装形態では、実際の人間の動作を観察することで得られた、人のリアクション動作は3つの段階に分けることができるという考えにもとづき、自然なリアクション動作を実現する。3つの段階は、能動制御段階、ステップ付き能動制御段階、受動制御段階である（図5参照）。一般に、衝撃が人に与えられたとき、その人は手や上半身を動かしてバランスをとろうとする（能動制御段階）。衝撃が小さい場合には、能動制御段階中に、バランスを回復することができる。もし、与えられた衝撃が比較的大きく、能動制御段階中にバランスを回復することができない場合には、通常、転倒を防ぐために、倒れようとする方向に足を踏み出す（ステップ付き能動制御段階）。このとき、足を数歩分動かすことで

10

【0049】

本実施形態では、上記のようなリアクション動作を実現する。まず、アバターに衝撃が加えられたときに、3種類のリアクション動作のどれを実行するかを、衝撃が加えられた後のアバターの状態にもとづいて決定する。入力姿勢に対する追従制御とリアクション動作に対する追従制御の混合比率は、実行するリアクションの種類及び3つの段階のどの段階であるかにもとづき、制御される。図5の各リアクション動作の下の実線と点線は、各種種類のリアクション動作の各段階における、入力姿勢に対する追従制御とリアクション動作に対する追従制御の混合比率をどのように変化させるかを、それぞれ表している。

20

【0050】

衝撃が与えられた直後のアバター状態の変化が小さいときには（図5（a）参照）、動作データベースに格納されたリアクション動作データ（以降、衝撃によって実現される結果の動作を「リアクション動作」、それを生成するために追従制御の目標動作として使用する、あらかじめ動作データベースに格納されている動作データのことを「リアクション動作データ」と呼称する。）は使用せず、入力姿勢の追従制御に小さな比率を使用することで、衝撃の影響によって利用者がアバターを完全に制御することができない能動制御段階を実現する。一方、大きな衝撃がアバターに与えられたときには、適切なリアクション動作データを動作データベースから検索して、追従制御の目標動作として使用する。アバターの状態をもとに動作データベースから検索されたリアクション動作データに設定された情報にもとづいて、実行するリアクション動作の種類を決定する。アバターがそれほど大きくバランスを崩していなければ、ステップ付き能動制御段階でバランスを回復する（図5（b）参照）。アバターが大きくバランスを崩していれば、受動制御段階まで実行する（図5（c）参照）。これらのリアクション動作を実行する場合も、小さな衝撃が与えられた場合（能動制御段階でバランスを回復する場合）と同様に、最初は入力姿勢の追従制御に小さな比率を付与する。ステップ付き能動制御段階にバランスを回復する場合には、ステップ付き能動制御段階中に比率を徐々に大きくすることで、当該段階の終了時には、アバターは入力姿勢の追従に戻る。他方、受動制御段階まで実行する場合には、ステップ付き能動制御段階で、リアクション動作データへの追従制御の比率を徐々に大きくし、入力姿勢の追従制御の比率を徐々に小さくしていく。受動制御段階の間は、入力姿勢への追従制御の比率は0にセットされ、アバターはリアクション動作データに完全に追従する。

30

40

【0051】

受動制御段階まで実行し、アバターが転倒してしまうと、リアクション動作が終了した時点で、アバターの姿勢と利用者の姿勢は異なる。このとき、アバターを利用者の姿勢に追従させると、アバターが不自然に立ち上がるような動作が生成されてしまう。このよ

50

うな問題の回避をするために、リアクション動作が終了した後に、アバターが利用者の姿勢に近い姿勢になるように、適切な遷移動作を実行する。前記遷移動作の実行方法については、後説する。

【 0 0 5 2 】

[3 . 1 リアクション動作データベース]

動作データベースには、あらかじめ、いくつかのリアクション動作データを格納しておく。大きな衝撃がアバターに加えられ、アバターがバランスを崩したとき、本実施形態では、動作データベースから適切なリアクション動作データを検索し、追従制御の目標動作として用いる。加えられた衝撃やそのときのアバターの状態に応じて、最も自然なリアクション動作を実現するのに最適なリアクション動作データを検索するため、それぞれのリアクション動作データには、前もって自動的に算出したインデックスキーを付与しておく。このインデックスキーは、リアクション動作が開始された際のアバターのバランス状態と姿勢状態が含まれる。

10

【 0 0 5 3 】

実行するリアクション動作データを決定する上で、リアクション動作を開始する時点でのアバターの運動方向や速度が最も重要となる。なぜなら、これらの値がリアクション動作の方向を決定するからである。リアクション動作データの検索において、重心の位置及び速度を考慮するため、本実施形態では、文献 1 3 (Hof A. L., Gazendam, M. G. J., Sinke, W. E. The condition for dynamic stability. Journal of Biomechanics, 38, 1-8, 2005.) で提案されている推定重心位置 (X c o M) の考え方を利用した。X c o M は、重心の位置と速度から計算される位置で、文献 1 3 では人間のバランス状態を評価するために用いられている。

20

X c o M は逆振り子モデルにもとづいており、次の式により算出される。

【 0 0 5 4 】

【数 4】

$$P_{XcoM} = P_{CoM} + \sqrt{l/g} P'_{CoM}, \quad (4)$$

【 0 0 5 5 】

P_{CoM} 、 P'_{CoM} はそれぞれ重心の位置、速度であり、 l は足の長さであり、 g は重力加速度である。ここで、X c o M を使用する代わりに、重心の位置及び速度を独立した変数として検索時の評価に用いることも可能であるが、この場合は、位置と速度をどれくらいの割合で勘当することという適切な比率を何らかの方法で決定することが必要になる。そこで、本実施形態では、このような比率の決定を避けるために、文献 1 3 で使用されているモデルを用いることとした。

30

【 0 0 5 6 】

適切なリアクション動作データを選択するためには、リアクション動作を開始するときの重心の位置や速度だけではなく、アバターの全身の姿勢や速度も重要な要素である。しかしながら、全ての部位の位置、速度を検索時の評価として用いるのは冗長である。そこで、本実施形態では、手先足先などの人体の末端部位の位置、速度のみを考慮することとした。一般に、バランスを失おうとするときのアバターの運動速度は大きいので、末端部位の位置だけではなく、速度も考慮する必要がある。そこで、X c o M と同様の考え方にもとづき、位置と速度の両方を考慮するため、手先足先の末端部位についても、式 (4) を用いて、推定末端位置 (X E E) を計算することにする。X E E は、各手足の元となる部位 (手の場合は肩、足の場合は股関節) を原点とするローカル座標系で示される。ローカル座標系の向きは、Z 軸が人体モデルの前方方向を向き、Y 軸はワールド座標系の Y 軸を向いている。以上で説明した通り、X c o M (3 次元ベクトル) と各手足の X E E (3 次元ベクトル) を使用し、インデックスキーは 1 5 次元のベクトルとして表される (図 6) 。

40

50

【 0 0 5 7 】

動作データベース中の各リアクション動作データのインデックスキーは、あらかじめ計算しておく。各リアクション動作データにおいて、キャラクターに衝撃が与えられた後に X c o M 長さが最も大きくなったときのキャラクターの状態にもとづいて、インデックスキーを計算する。各リアクション動作データごとに、開発者が大まかなりアクション開始時刻を指定しておく、指定時刻の前後の一定時間間隔内 (0 . 2 [秒]) を探索して、X c o M が最も大きくなる時刻を算出する。

【 0 0 5 8 】

[3 . 2 リアクション動作の探索]

アバターに衝撃が与えられると、本実施形態では、まずはその衝撃が小さいものと仮定し、図 5 (a) に示すように、リアクション動作データは使用せず、追従制御の混合比率を単純に制御し始める。その後、各シミュレーションステップにおいて、アバターのバランス状況 (X c o M 及び X E E s) を算出する。一般に、X c o M がアバターのサポートポリゴンの外部にある場合、アバターはバランスを失っていると考えられることができる (文献 1 3) 。このとき、最もアバターの状態に近いインデックスキーを有するリアクション動作データが用いられる。なお、サポートポリゴンとは、両足などのアバターが地面に接触している範囲全体を含む最小の凸多角形のことである。サポートポリゴンは、アバターの姿勢及び描画用の幾何形状モデルから、アバターと地面の間の接触面を計算し、その全ての接触面を含む凸包を計算することで、算出することができる。凸包計算の方法については、文献 1 4 などでも説明されている (文献 1 4 : O'Rourke, J. Computational Geometry in C (Cambridge Tracts in Theoretical Computer Science), Cambridge University Press, 1998.) 。アバターがバランスを失い始めたとき、つまり、アバターの X c o M がサポートポリゴンの外に移動したときに、そのときのアバターの状態から計算されたインデックスキーと、動作データベース中の各リアクション動作データのインデックスキーとを比較して、次の式で計算される、2つのインデックスキーの差が最も小さくなるリアクション動作データを検索する。

【 0 0 5 9 】

【 数 5 】

$$E_i = w_{XcoM} |p_{XcoM}^i - P_{XcoM}| + \sum w_j |p_j^i - p_j|, \quad (5)$$

【 0 0 6 0 】

ここで、 P_j^i 、 P_{XcoM}^i はそれぞれ X c o M の位置、j 番目の端部 (手、足、頭) の位置であり、 w_j 、 w_{XcoM} はそれぞれ X c o M の重み、j 番目の端部の重みである。たとえば、現在の実装においては、 $w_{XcoM} = 5 . 0$ 、 $w_j = 1 . 0$ を用いている。

衝撃がアバターに与えられた直後は、X c o M の長さは小さく、その後、徐々に大きくなる。そのため、あまり早くリアクション動作データの検索を行ってしまうと、適切なリアクション動作データが検索されない。そこで、X c o M がサポートポリゴンを超えた後、指定された持続期間の間、バランス状況を算出し続け、初めて X c o M の長さが極大値となったときの状態が、リアクション動作データの検索に用いられる。

【 0 0 6 1 】

[3 . 3 混合比率制御]

前説した考え方にもとづき、図 5 に示すように、入力姿勢への追従制御とリアクション動作への追従制御の混合比率は、実行しているリアクション動作の種類、現在の段階、リアクション動作開始からの経過時間より決定される。それぞれのリアクション動作データには、前もって各段階の切り替わるタイミングを示す時刻が設定されているものとする。最初の能動制御段階では、比率は線形に変化する。2 番目のステップ付き能動制御段階では、しばらくは一定の比率を使用し、次の段階に切り替わる時刻の一定時間前から、同じく比率を線形に変化させる。

【 0 0 6 2 】

本実施形態では、図 5 で示される線形関数を単純に使用するのではなく、入力姿勢の上半身の速度にもとづいて、入力姿勢への追従制御の比率を変化させる。図 5 の比率を単純に使用した場合、もし利用者が上半身をほとんど動かさなければ、アバターはその静止した姿勢に追従するため、追従制御の結果を混合して生成された動作は、元々のリアクション動作データに比べて動きが小さくなってしまふ。そこで、この問題を解決するために、利用者の両手と頭の速度の合計値があらかじめ決められた閾値よりも小さい場合には、図 5 の重み関数 $f(t)$ に、両手と頭の速度を掛け合わせたものを比率として使用する。

【 0 0 6 3 】

【 数 6 】

$$w_{input}(t) = \begin{cases} f(t) & \text{if } (|v_{right_hand}| + |v_{left_hand}| + |v_{head}|) \leq v \\ f(t) \times (|v_{right_hand}| + |v_{left_hand}| + |v_{head}|) / v & \text{otherwise.} \end{cases} \quad (6)$$

10

【 0 0 6 4 】

リアクション動作の種類とリアクション動作データが決定された後は、動作司令部は、リアクション動作が続く間、混合比率の制御を続ける。リアクション動作の実行が終了し、さらに必要であれば遷移動作の実行が終了すると、システムは標準の制御状態に戻る。

【 0 0 6 5 】

[3 . 4 入力姿勢の遷移]

前記したように、アバターが転倒した後に、不自然な動作が生成されることを防ぐため、本実施形態では遷移動作を実行する。ここでは、利用者は常に両足で立っていると仮定し、転倒した状態から立ち上がるような動作データを目標動作として追従制御を行う。このとき、なるべく自然な遷移動作を実行するためには、入力姿勢（利用者の状態）及び現在の状態（アバター状態）にもとづき、動作データベースから適切な動作データを動的に計算することが望ましい。しかし、現在の実装では、それぞれのリアクション動作データに、適切な立ち上がる動作をあらかじめ手動で割り当てている。

20

【 0 0 6 6 】

[4 . 歩行動作]

本実施形態では、図 2 に示すように、利用者が歩行のしぐさを行うことで、追従制御の目標動作として歩行動作を生成する。アバターの歩行動作を開始するときには、利用者は、普段歩くときと同様に、手足を同時に動かして足踏みのしぐさを行う必要がある。アバターが歩行動作を始めた後は、利用者は必ずしも足に合わせて手を動かす必要はなく、利用者が足踏みを続ける限り、アバターも歩行動作を続ける。そのため、アバターが歩行動作を開始した後は、利用者は、アバターの上半身の動作を自由に操作することができる。例えば、利用者は、アバターに、歩きながら手を振ったり、他のキャラクターを打撃させたりすることができる。このようなインターフェースを実現するために、本実施形態では、利用者の体全体のしぐさ（歩行動作の開始判定に使用）と、利用者の下半身のしぐさ（歩行動作の継続判定に使用）を認識する。入力姿勢への追従制御と歩行動作への追従制御の比率もまた、この動作認識の結果にもとづいて制御される。

30

40

【 0 0 6 7 】

[4 . 1 歩行動作の生成]

本実施形態では、動作生成部において、利用者の動作にもとづき、生成される歩行の速度や曲がる方向を制御するために、歩行動作の生成にモーションブレンディング技術を用いる（文献 15 : Park S. I., Shin H. J., Shin S. Y. On-line Locomotion Generation Based on Motion Blending. In Proc. of ACM SIGGRAPH Symposium on Computer Animation 2002, 113-120, 2002.）。もちろん、どのような動作生成の技術（例えば、モーショングラフ、ダイナミックシュミレーション等）も、本発明と組み合わせて使用することができる。

50

【 0 0 6 8 】

モーションブレンディングを行うために、まず、さまざまな速度、曲がる方向のいくつかの歩行動作（サンプル動作）を用意する。各サンプル動作には、あらかじめ手動で動作の節目を表すキー時刻を設定しておく。このキー時刻の情報は、速度の異なるサンプル動作を同期して、モーションブレンディングを行うために作融合処理の不可欠である。加えて、それぞれのサンプル動作の速度及び曲がる方向も、前もって計算される。生成する歩行動作のパラメータ（速度及び曲がる方向）が与えられたとき、この情報を用いて、与えられたパラメータを満たす歩行動作を生成するための、サンプル動作の混合比率が計算される。この混合比率にもとづきサンプル動作をブレンディングすることで、歩行動作が生成される。モーションブレンディングに関してより詳細な事項は文献 15 に記載されている。

10

【 0 0 6 9 】

[4 . 2 歩行の身ぶりの検出]

利用者のしぐさに応じてアバターの歩行動作を制御するためには、利用者が足踏みのしぐさを行っているかの判定だけではなく、利用者のしぐさから歩行動作の速度や曲がる方向などのパラメータを計算する必要がある。歩行方向（曲がる方向の角度）は利用者の腰の向きの変化から容易に算出できる。歩行速度を計算するために、図 7 に示すように、歩行動作を四つの状態に分け（右足上げ、右足下げ、左足上げ、左足下げ）、利用者が現在の状態の動きを行っているかを検出することで、状態遷移の速度から歩行速度を計算する。利用者がどの歩行状態を行っているかは、利用者の腰、足及び手の速度にもとづいて（図 8 参照）、ファジールールを用いて判定する。

20

【 0 0 7 0 】

全身のしぐさによる歩行状態を検出するために、腰、足及び手の速度を用いる。例えば、右足が上がっている状態（図 7 「右足上げ」）では、右足の垂直速度 $v_y^{right_foot}$ は大きな正の値（上向きの運動）になる。また、右手の水平 $v_z^{right_hand}$ は負の値（後方への運動）、左手速度 $v_z^{left_hand}$ は正の値（前方への運動）になる。また、足踏みのしぐさ中は、腰の水平速度 $v_{|x+z|}^{pelvis}$ は常に小さな値となる。

【 0 0 7 1 】

図 9 に示すように、あらかじめ与えられた速度の平均値と分散により定義されるメンバーシップ関数にもとづいて、各速度がどれだけ足踏みしぐさ中の値に近いかを表す評価値を計算する。例えば、右足 $v_y^{right_foot}$ の評価値（0 から 1 の間の値をとる）は次の式により算出される。

30

【 0 0 7 2 】

【数 7】

$$e_y^{right_foot} = \begin{cases} (v_y^{right_foot} - a^{right_foot}) / d^{right_foot} & \text{if } |v_y^{right_foot} - a^{right_foot}| < d^{right_foot} \\ 0 & \text{if } |v_y^{right_foot} - a^{right_foot}| \geq d^{right_foot} \end{cases} \quad (7)$$

40

【 0 0 7 3 】

同様の方法で計算された各部位の速度の評価値にもとづき、例えば、利用者の全身のしぐさが、右足が上がっている状態（図 7 Right Leg Up）かどうかの確率を表す評価値は、次の式により算出される。

【 0 0 7 4 】

【数 8】

$$e^{\text{right_leg_up(full)}} = \min \left\{ e_y^{\text{right_foot}}, e_{|x+z|}^{\text{pelvis}}, e_z^{\text{right_hand}}, e_z^{\text{left_hand}} \right\}. \quad (8)$$

同様に、下半身のしぐさによる歩行状態を検出するためには、手の速度は考慮せず、腰及び足の速度のみを用いる。

【0075】

【数 9】

$$e^{\text{right_leg_up(lower)}} = \min \left\{ e_y^{\text{right_foot}}, e_y^{\text{left_foot}}, e_{|x+z|}^{\text{pelvis}} \right\}. \quad (9)$$

【0076】

他の3つの歩行状態の認識(図7)にも、同様の関数を用いる。

$e^{\text{right_leg_up(full)}}$ 又は $e^{\text{left_leg_up(full)}}$ のどちらかの確率が閾値を超えると、アバターは歩行動作を開始する。

歩行動作中に、次の歩行状態(例えば、現在の状態が右足上げであれば、次は右足下げ) $e^{\text{right_leg_down(lower)}}$ の可能性が閾値を越えたら、次の歩行状態へ遷移する。

【0077】

歩行状態がある状態から次の状態へ遷移する瞬間は、手足の速度はゼロに近づくため、歩行状態を正しく認識することができない。この問題を解決するため、一定時間の間、どの歩行状態でもない状態が続いても、前の歩行状態を継続する。一定時間以上、どの歩行状態でもない状態が続いたら、歩行のしぐさは停止したと認識する(図7の「不歩行」)。

【0078】

利用者のしぐさが次の歩行状態であると検出されたら、動作認識部は、利用者の歩行状態を次の状態に遷移させる。動作状態の遷移速度は、以前のいくつかの歩行状態への遷移時刻から、算出される。また、現在の動作状態と状態遷移速度にもとづき、歩行動作の標準時間(1周期の歩行動作の開始時刻を0、終了時刻を1とし、0から1の間で変化する時間)も算出される。

【0079】

[4.3 歩行動作制御]

歩行動作の実行中は、入力姿勢に対する追従制御の比率は、腰を含む下半身については常に0である。上半身については、利用者が全身で歩行動作のしぐさを行っている間、又は、利用者が手を全く動かしていない間は、比率は0である。一方、(6)の計算と同様に、利用者が手を動かす速度が一定値以上であれば、上半身の比率には1に近い値を設定する。このような比率制御により、歩行動作が開始された後は、利用者の上半身が歩行以外の動きを行ったときのみ、アバターは利用者の上半身の姿勢に追従する。このように本実施形態では上半身と下半身で追従制御の比率を変更したが、複数の部位で追従制御の比率を変更することができる。たとえば、衝撃を受けた近傍部位は入力姿勢に対する追従制御の比率を小さくし、その衝撃を受けた近傍部位から離れる部位程入力姿勢に対する追従制御の比率を大きくすることもできる。

【0080】

[5.ハードウェア構成の具体例]

図10は本実施形態に係るモーションキャプチャーシステムのハードウェア構成図である。

本実施形態に係るモーションキャプチャーシステムは、コンピュータ10、表示装置20及びカメラ30からなる。その他、利用者に取り付けるマーカがある。

10

20

30

40

50

【 0 0 8 1 】

コンピュータは、CPU (Central Processing Unit) 1 1、DRAM (Dynamic Random Access Memory) 1 2等のメインメモリ、外部記憶装置であるHD (hard disk) 1 3、入力装置であるキーボード 1 5及びマウス 1 6、ネットワークに接続するための拡張カードであるLANカード 1 7、CD-ROMドライブ 1 8等からなる。なお、本実施形態ではLANカードを使用しないが、当業者であれば明らかな通り、システム構成によってはLANを構築して本発明を実現することができる。

例えば、CD-ROMに格納されているアバター動作制御プログラムがHD 1 3上に複製(インストール)され、必要に応じてアバター動作制御プログラムがメインメモリ 1 2に読み出され、CPU 1 1がかかるプログラムを実行することでアバター動作制御装置を構成する。

10

【 0 0 8 2 】

[6 . ブロック構成図の説明図]

図 1 1 は本実施形態に係るシュミレーションシステムのブロック構成図である。

モーションキャプチャー部 4 1 は、カメラ 2 0 からの出力信号を受け、利用者の姿勢を計算し、入力姿勢として出力する。

入力姿勢用の追従制御部 4 2 は、出力された入力姿勢(目標とする入力姿勢)及びアバター状態を受けて制御加速度を求めて出力する。なお、たとえば、PD制御法を用いて制御加速度を算出する。

【 0 0 8 3 】

動作認識部 4 3 は、出力された入力姿勢及び動作状態を受けて利用者の動作状態を認識して動作状態を出力する。

制御指令部 4 4 は、出力された動作状態及びアバター状態が所定状態に合致するか否かを判断し、合致している場合には対応する動作を動作生成部 4 6 に生成するように制御信号を出力する。また、制御指令部 4 4 は、所定状態に対応する混合比率を出力する。

20

【 0 0 8 4 】

動作生成部 4 6 は、制御信号を受けて動作データベース 4 7 から適切なりアクション動作データを読み出し、必要に応じて変更を加えて生成動作として出力する。

生成動作の追従制御部 4 8 は、出力された生成動作の姿勢(目標とする生成動作の姿勢)及びアバター状態を受けて制御加速度を求めて出力する。より詳細には、各時刻ごとに、生成動作の該当時刻の姿勢を目標姿勢として追従制御を行う。なお、たとえば、PD制御法を用いて制御加速度を算出する。

30

【 0 0 8 5 】

状態変化計算部 4 5 は入力姿勢に係る制御加速度、生成動作に係る制御加速度及び混合比率を受けて、混合比率にもとづいて入力姿勢に係る制御加速度及び生成動作に係る制御加速度を用いてアバター状態を変更する。

キャラクター制御部 5 0 はアバター以外の仮想空間内に存在するキャラクターの動作を制御する。

衝突処理部 6 0 はアバターとキャラクターの衝突を検出し、アバター状態とキャラクター状態を衝突に応じて更新する。

40

【 0 0 8 6 】

[7 . データ構造の具体例]

アバター状態は、腰の位置・向き・並進速度・回転速度、各関節の回転行列・回転速度(又は各回転軸の回転角度・回転角速度)からなる。

制御加速度は、腰の並進加速度・回転加速度、各関節の回転加速度(又は各回転軸の回転角加速度)からなる。ただし、足が地面に接触しているときは、足先の並進加速度・回転加速度も含める。

動作状態は、現在の状態を表すフラグ(通常状態・歩行動作中・リアクション動作中)、歩行状態(4種類)、リアクション動作のタイプ(3種類)、実行中のリアクション動作の各段階の切り替わりの時刻からなる。

50

【 0 0 8 7 】

[8 . 動作フローチャートの具体例]

図 1 2 ないし図 1 5 は本実施形態に係るシュミレーションシステムの動作フローチャートである。

モーションキャプチャー部 4 1 がカメラ 3 0 からの入力情報から入力姿勢を出力する (ステップ 1 0 1)。

入力姿勢の追従制御部 4 2 において、アバター状態が入力姿勢に近づくような、制御加速度を計算する (ステップ 1 1 1)。

動作認識部 4 3 において、入力姿勢・前の動作状態に応じて、現在の動作状態を認識する (ステップ 1 2 1)。

制御指令部 4 4 が現在の動作状態が歩行動作であるか否かを判断し (ステップ 1 3 1)、歩行動作でなければ制御指令部 4 4 が現在の動作状態がリアクション動作であるか否かを判断する (ステップ 1 3 2)。

【 0 0 8 8 】

ステップ 1 3 2 でリアクション動作でないと判断した場合には、制御司令部 4 4 は、入力姿勢にもとづく追従制御の割合が 1 になるような混合比率を出力する (ステップ 1 4 1)。

状態変化計算部 4 5 において、追従加速度・混合比率にもとづき、アバター状態を更新する (ステップ 1 5 1)。

ステップ 1 5 1 の後、ステップ 1 0 1 に戻る。

前記ステップ 1 3 1 で歩行動作であると判断した場合には、制御指令部 4 4 において、動作状態にもとづき、歩行動作のパラメタを決定する (ステップ 2 0 1)。

【 0 0 8 9 】

制御指令部 4 4 において、混合比率を計算する (ステップ 2 1 1)。

動作生成部 4 6 において、指定された歩行動作のパラメタと、動作データベース 4 7 に格納された動作データをもとに、動作ブレンディング処理を行って適切な歩行動作を生成する (ステップ 2 2 1)。

生成動作の追従制御部 4 8 において、アバター状態が、生成動作の現在姿勢に近づくような、制御加速度を計算する (ステップ 2 3 1)。

【 0 0 9 0 】

状態変化計算部 4 5 において、追従加速度・混合比率にもとづき、アバター状態を更新する (ステップ 2 4 1)。歩行動作が終了した後に、ステップ 1 0 1 に戻る。

制御指令部 4 4 において、バランスを大きく崩した場合は、動作状態をリアクション動作中とし、リアクション動作のパラメタを動作生成部 4 6 に渡す (ステップ 3 0 1)。

制御指令部 4 4 において、混合比率を計算する (ステップ 3 1 1)。

動作生成部 4 6 は、制御指令部 4 4 から渡されたパラメタ (バランス状態) をもとに、適切なリアクション動作を生成する (ステップ 3 2 1)。

【 0 0 9 1 】

追従制御部 4 8 において、アバター状態が、生成動作の現在姿勢に近づくような、制御加速度を計算する (ステップ 3 3 1)。

状態変化計算部 4 5 において、追従加速度・混合比率にもとづき、アバター状態を更新する (ステップ 3 4 1)。リアクション動作が終了した後に、ステップ 1 0 1 に戻る。

処理の終了は例えば試合形式の格闘ゲームであれば一試合終了と同時に終了する。試合の開始とともに再度処理がなされる。

【 0 0 9 2 】

アバターとキャラクターの衝突は次の処理による。なお、ステップ 1 0 1 ないしステップ 3 4 1 の処理と、ステップ 4 0 1 及びステップ 4 1 1 の処理は別々のプロセス、スレッドとして並列処理させることもできるが、たとえば、ステップ 1 0 1 ないしステップ 3 4 1 におけるステップ 1 5 1、ステップ 2 4 1、ステップ 3 4 1 の後にステップ 4 0 1 及びステップ 4 1 1 の処理を実行する構成であっても実装することができる。

10

20

30

40

50

【 0 0 9 3 】

アバターとキャラクターが衝突したか否かを衝突処理部 6 0 が判断する（ステップ 4 0 1）。衝突していると判断した場合には、アバター状態・キャラクタ状態を更新する（ステップ 4 1 1。衝突に応じて両者の速度を変化させる）。衝突していないと判断した場合、ステップ 4 1 1 の後、ステップ 4 0 1 に戻る。このアバターとキャラクターの衝突の処理は前記ステップ 1 0 1 ないし 3 4 1 の処理とは別プロセス、別スレッドで並行に処理することが望ましい。なお、衝突が生じる例としては、キャラクター制御部 5 0 がキャラクタ状態を更新してキャラクターが動作した場合、前説した処理によりアバターが動作する場合がある。

【 0 0 9 4 】

以上の前記実施形態により本発明を説明したが、本発明の技術的範囲は実施形態に記載の範囲には限定されず、これら実施形態に多様な変更又は改良を加えることが可能である。そして、かような変更又は改良を加えた実施の形態も本発明の技術的範囲に含まれる。このことは、特許請求の範囲及び課題を解決する手段からも明らかなことである。

【 図面の簡単な説明 】

【 0 0 9 5 】

【 図 1 】本発明の第 1 の発明概念図である。標準時アバターは入力姿勢に追従する。生成動作を要するとき、アバターはどちらもとぎれることなく追従する。

【 図 2 】本発明の第 2 の発明概念図である。

【 図 3 】本発明の試作システムの使用外観図である。

【 図 4 】本発明の実施形態に係る人体モデルの正面図である。

【 図 5 】本発明の実施形態に係るリアクション動作の説明図である。リアクション動作は 3 つの段階からなる。アバターがどの段階でバランスを回復するかによってリアクション動作は 3 種類に分類される。実線が入力姿勢の比率であり、点線がリアクション動作の比率である。

【 図 6 】本発明の実施形態に係るリアクション動作のインデックスキーの説明図である。適切なリアクション動作は重心及び末端部位にもとづき検索される。

【 図 7 】本発明の実施形態に係る歩行状態の遷移図である。

【 図 8 】本発明の実施形態に係る歩行の身ぶりの検出説明図である。

【 図 9 】本発明の実施形態に係る状態評価に使用するメンバーシップ関数の一例である。

【 図 1 0 】本発明の実施形態に係るモーションキャプチャーシステムのハードウェア構成図である。

【 図 1 1 】本発明の実施形態に係るモーションキャプチャーシステムのブロック構成図である。

【 図 1 2 】本発明の実施形態に係るモーションキャプチャーシステムの動作フローチャートその 1 である。

【 図 1 3 】本発明の実施形態に係るモーションキャプチャーシステムの動作フローチャートその 2 である。

【 図 1 4 】本発明の実施形態に係るモーションキャプチャーシステムの動作フローチャートその 3 である。

【 図 1 5 】本発明の実施形態に係るモーションキャプチャーシステムの動作フローチャートその 4 である。

【 図 1 6 】背景技術の各視点の説明図である。

【 符号の説明 】

【 0 0 9 6 】

- 1 0 コンピュータ
- 1 1 CPU
- 1 2 DRAM
- 1 3 HD
- 1 5 キーボード

10

20

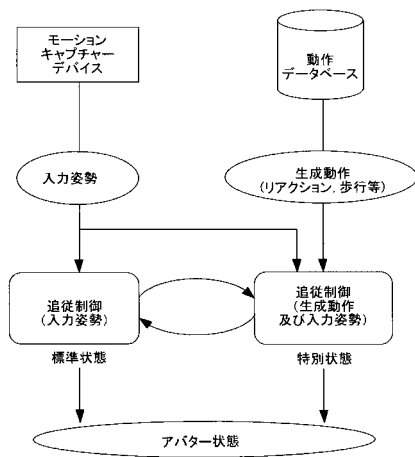
30

40

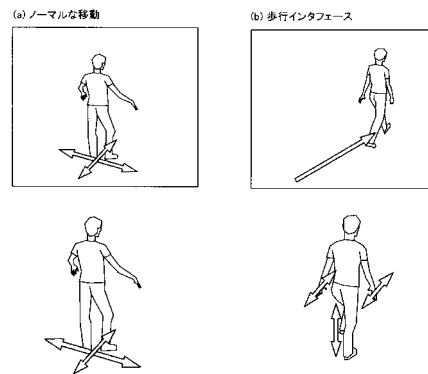
50

- 1 6 マウス
- 1 7 L A Nカード
- 1 8 C D - R O Mドライブ
- 2 0 表示装置
- 3 0 カメラ
- 4 1 モーションキャプチャー部
- 4 2 (入力姿勢の)追従制御部
- 4 3 動作認識部
- 4 4 制御指令部
- 4 5 状態変化計算部
- 4 6 動作生成部
- 4 7 動作データベース
- 4 8 (生成動作の)追従制御部
- 5 0 キャラクタ制御部
- 6 0 衝突処理部

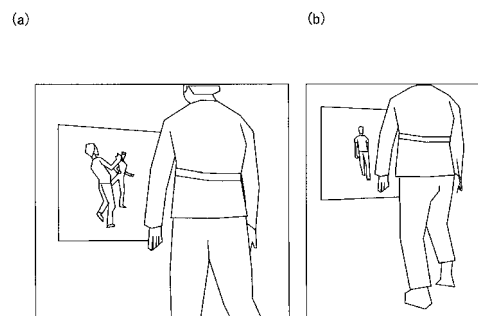
【 図 1 】



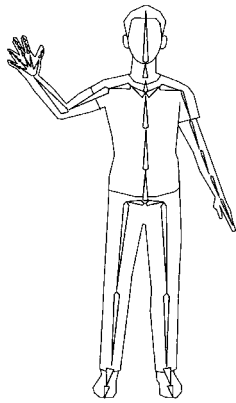
【 図 2 】



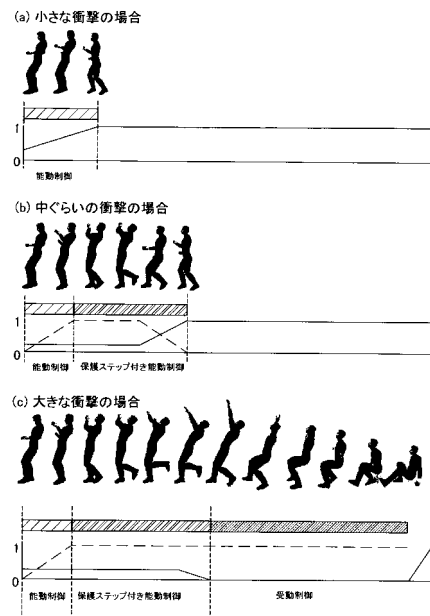
【 図 3 】



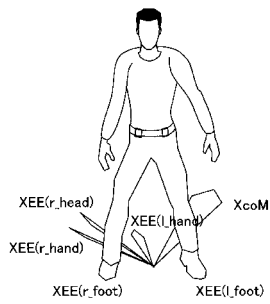
【 図 4 】



【 図 5 】



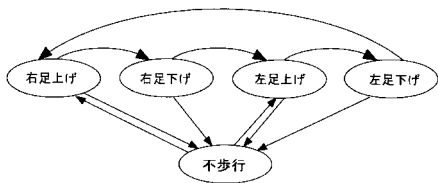
【 図 6 】



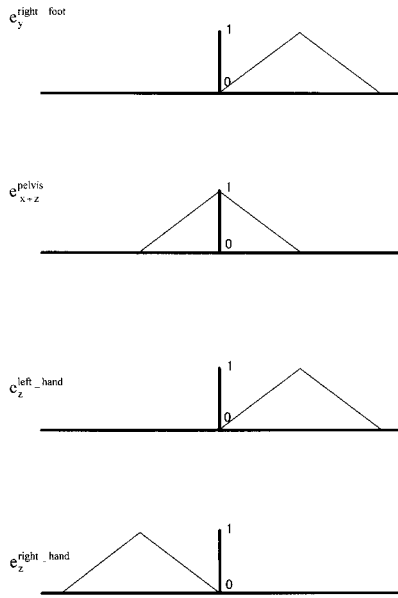
【 図 8 】



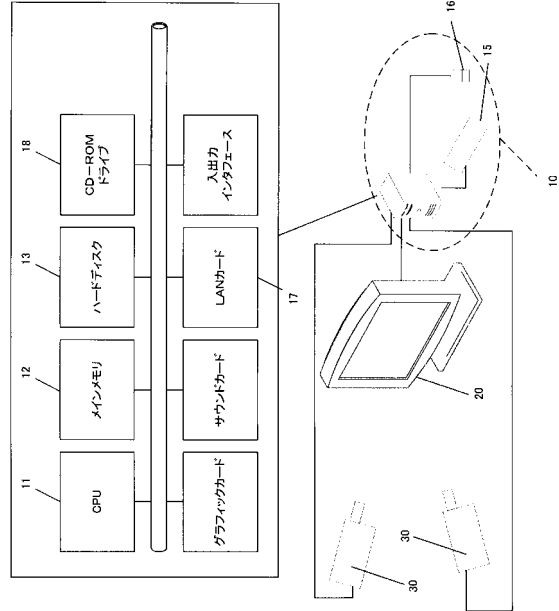
【 図 7 】



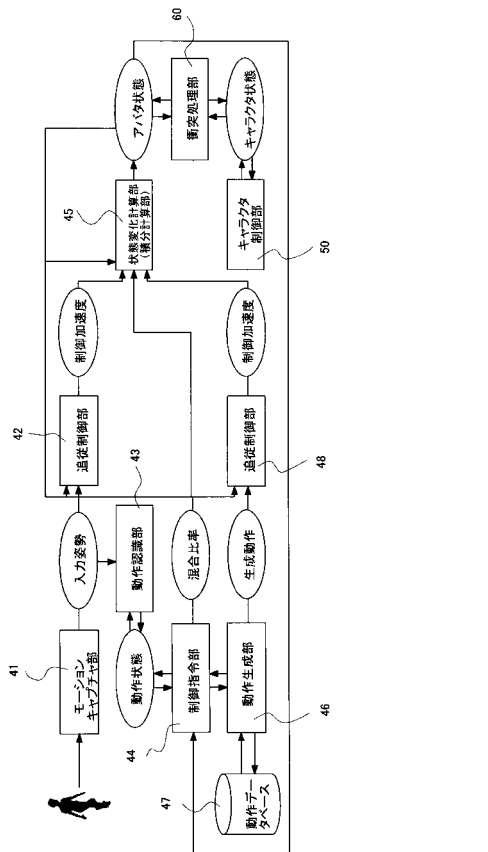
【図 9】



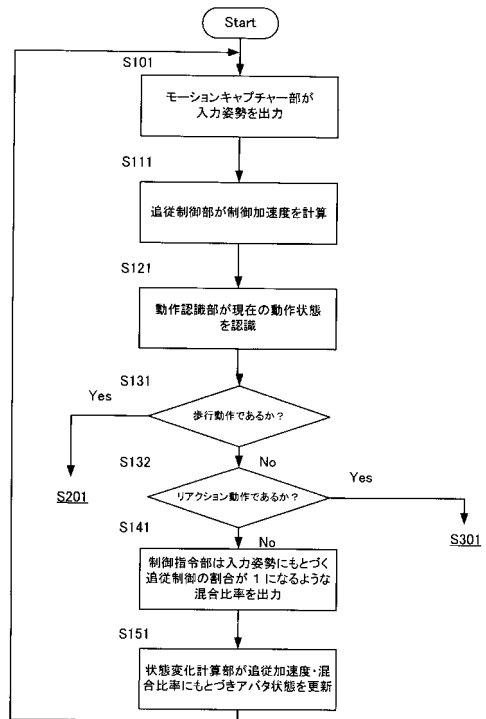
【図 10】



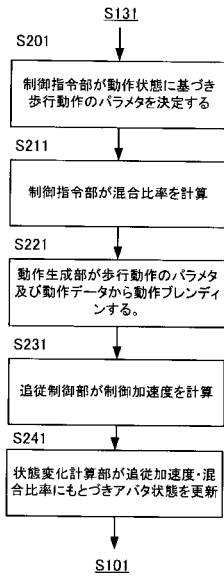
【図 11】



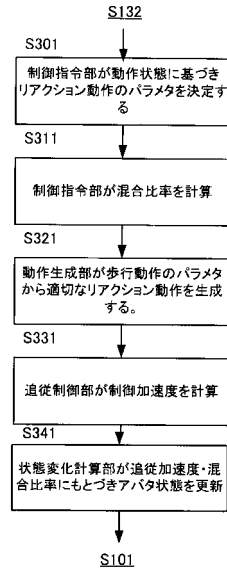
【図 12】



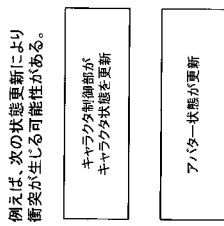
【図13】



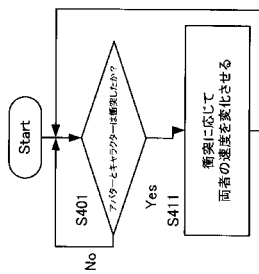
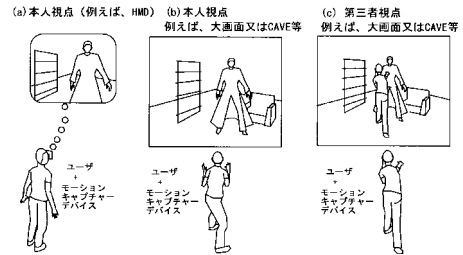
【図14】



【図15】



【図16】



フロントページの続き

(56)参考文献 特開2004-030502(JP,A)

特開2002-092643(JP,A)

特開2002-083286(JP,A)

特開2001-006000(JP,A)

工藤俊亮, 外2名, "全身を用いたバランス保持動作のモデル化とその生成", 電子情報通信学会論文誌, 日本, 社団法人電子情報通信学会, 2005年 8月 1日, 第J88-D-II巻, 第8号, p.1583-1591

(58)調査した分野(Int.Cl., DB名)

G06T 19/00, 13/00 - 13/80, 19/20

A63F 9/24, 13/00 - 13/12

G06T 15/00 - 15/87

G09G 5/00 - 5/42