

エネルギー科学研究科
エネルギー社会・環境科学専攻修士論文
題目： アフォーダンスの概念に基づく
人体モーション合成システムの開発

指導教官： 吉川 榮和 教授

氏名： 市口 誠道

提出年月日： 平成12年2月7日(月)

論文要旨

題目：アフォーダンスの概念に基づく人体モーション合成システムの開発

吉川榮和研究室, 市口誠道

要旨：

近年のコンピュータ技術やソフトウェア技術の発達により、コンピュータを用いて3次元コンピュータグラフィックス(3DCG)により人の動作を合成することが可能となり、映画やゲーム等に利用されている。このように、コンピュータを用いて3DCGで表現された人体モデルの動作を合成する人体モーション合成システムは、特定の動作合成を目的に構築されているものがほとんどであり、合成された人体モデルが仮想空間内で周りのオブジェクトとインタラクションできるような機能を重視して開発されているシステムは少ない。

そこで本研究では、仮想空間内の人体モデルとオブジェクトとの自由なインタラクションが合成可能であるとともに、人体モデルの新たな動作を合成することが可能な人体モーション合成システムの開発を目的として研究を進めた。仮想空間に人体モーションを合成することは、実世界での人々の振舞いをコンピュータ上で再現することになるので、人の実世界での振舞いを取扱う認知科学から有用なヒントが得られる。認知科学でのアフォーダンスの概念は、情報は人間をとりまく環境そのものの中に実在しており、我々はその情報を直接知覚するとしている。この主張は、我々が深く考えることもなく環境の変化や未知の環境に無意識に適応していることをよく説明している。

本論文でははじめに、研究の背景として人体モーション合成について述べ、目的とするシステムを開発するために、認知科学の概念であるアフォーダンスを適用することを述べている。

次に、アフォーダンスの概念について概説し、実空間と仮想空間の対応づけを行った後、アフォーダンスの概念を人体モーション合成に利用する手法について述べている。本研究で用いた手法は、まず、仮想空間内のオブジェクトのデータを、システムと独立したオブジェクトデータベースに格納する。そして、人体モデルの動作に関する情報を操作対象となる仮想空間内のオブジェクトに保持させる。これにより、人体モデルの動作情報と人体モーション合成システムが独立となり、動作情報のみの追加・変更が可能となって、人体モデルの動作を自由に追加・変更することが可能となる。

そして、上記の手法に基づいて実際に開発したシステムの詳細を述べる。その後、開発したシステムの使用事例としてアニメーションを作製し、人体モデルと仮想空間内のオブジェクトとのインタラクションが合成可能であることを実証する。また、開発したシステムの評価実験を行い、本システムにより自然な動作を合成できることを確認する。

最後に、結論として本研究を総括し、今後の課題を挙げる。

目次

第 1 章 序論	1
第 2 章 研究の背景と目的	3
2.1 ヒューマンインタフェースとバーチャルコラボレータ	3
2.2 人体モーション合成	6
2.2.1 人体モーション合成の利用	6
2.2.2 人体モーション合成に関する従来手法とその問題点	6
2.3 研究の目的	9
第 3 章 アフォーダンスの概念とその人体モーション合成システムへの適用	11
3.1 認知科学におけるアフォーダンスの概念	11
3.1.1 アフォーダンスとは	11
3.1.2 アフォーダンスの知覚	11
3.1.3 アフォーダンスの特性	13
3.2 アフォーダンスの概念の人体モーション合成への応用	14
3.2.1 実空間と仮想空間との対応	14
3.2.2 アフォーダンスの概念の人体モーション合成システムへの適用	15
3.2.3 オブジェクト指向との関係	17
3.3 アフォーダンスの概念の導入による利点・欠点	17
3.3.1 アフォーダンス導入による利点	18
3.3.2 アフォーダンス導入による欠点	21
3.4 アフォーダンスの概念に基づく人体モーション合成システムの要求仕様	21
第 4 章 人体モーション合成システム (AHMSS) の開発	30
4.1 人体モデルの構成とその姿勢の表現	30
4.1.1 人体モデルの構成	30
4.1.2 姿勢の 4 元数表現	34
4.2 AHMSS の概要	37

4.2.1	ソフトウェア構成	37
4.2.2	ハードウェア構成	41
4.2.3	システム全体の処理の流れ	42
4.2.4	システムを構成する各要素における処理	44
第 5 章	AHMSS の使用例と評価及び考察	54
5.1	AHMSS の使用方法	54
5.1.1	アニメーションを合成する際の手順	54
5.1.2	仮想空間に新たにオブジェクトを追加する際の手順	55
5.1.3	バーチャル人間の新たな動作を追加する際の手順	56
5.2	AHMSS の使用例	57
5.3	動作の合成手法	59
5.3.1	モーションキャプチャデータ再生アルゴリズム	59
5.3.2	歩行動作のアルゴリズム	69
5.3.3	球状線形補間アルゴリズム	76
5.3.4	その他のアルゴリズム	78
5.3.5	アニメーション合成例	80
5.4	AHMSS の評価と考察	82
5.4.1	AHMSS の評価	82
5.4.2	考察	86
5.5	今後の展望	89
5.5.1	バーチャルコラボレータの将来構想への対応	89
5.5.2	アフォーダンスに基づく人体モーション合成の将来展望	91
第 6 章	結論	93
	謝辞	95
	参考文献	96

目 次

2.1	バーチャルコラボレータ構成図	4
3.1	包囲光配列とその変化 (J.J.Gibson: The Senses Considered as Perceptual Systems, Houghton Mifflin(1996) より)	12
3.2	仮想空間へのオブジェクトの追加	19
3.3	動作情報の分割	25
3.4	従来の人体モーション合成システムのシステム構成	27
3.5	アフォーダンスに基づく人体モーション合成システムのシステム構成	27
4.1	人体モデルと各部名称	31
4.2	指モデル	32
4.3	人体モデルのリンクの階層構造	32
4.4	ワールド座標と各リンクに設定したローカル座標	33
4.5	指モデルのローカル座標系	34
4.6	Quaternion が示す回転	35
4.7	ソフトウェア構成	39
4.8	ハードウェア構成図	42
4.9	バーチャル人間の動作合成時の処理の流れ	43
4.10	仮想空間へのオブジェクト追加時の処理の流れ	44
4.11	メインプロセスの詳細	46
4.12	オブジェクトデータの構造	51
5.1	アニメーションシナリオ	58
5.2	バーチャル人間の移動経路 (上から見た図)	58
5.3	合成するアニメーションのシナリオとアルゴリズムの対応	59
5.4	動作計測時の機器配置	61
5.5	トランスミッタ	62
5.6	専用パソコン	63
5.7	磁気センサの取り付け位置 (腕)	64

5.8	磁気センサの取り付け位置（足）	64
5.9	計測の様子	65
5.10	磁気センサとリンクの方向のずれ	66
5.11	直立姿勢	67
5.12	腰位置の算出方法 1	70
5.13	腰位置の算出方法 2	70
5.14	座標変換の方法	75
5.15	アニメーションの合成例	80
5.16	ドアを開ける動作の合成例	81
5.17	歩きながらコップを手にする動作の合成例	81

表 目 次

3.1	アフォーダンスの特性	14
3.2	実空間と仮想空間の対応	15
3.3	テーブル、椅子、コップのアフォーダンス	16
3.4	各領域の問題の対応関係	21
3.5	人体モーション合成システムの仕様	22
3.6	人体モーション合成システムに求められる仕様	29
4.1	仕様とソフトウェア構成の関係	40
4.2	入力データ例	42
4.3	ユーザからの入力情報	45
5.1	動作とそのアルゴリズム	60
5.2	ULTRATRAK システムの基本性能	61
5.3	直立姿勢時のワールド座標系から各ローカル座標系への変換 Quaternion	68
5.4	作成した移動用の動作データの一覧	73
5.5	アルゴリズム毎のアニメーションの表示のフレームレート	84
5.6	アルゴリズム毎の計算のフレームレート	84
5.7	バーチャル人間とオブジェクトのポリゴン数	85
5.8	アルゴリズム毎の起動時間	85
5.9	評価の対象とした動作とそれぞれの動作間の移行	87

第 1 章 序論

近年のコンピュータ技術の発達により、3次元コンピュータグラフィックス(3DCG)は、コマーシャルフィルムや映画、テレビゲーム等、我々の身の回りのありとあらゆる場面で利用されている。

映画において3DCGを利用することは、もはや当たり前のことになりつつあり、3DCGを用いていない方が珍しいくらいになっている。3DCGで表現する対象も、数年前までは恐竜や未知の生命体などの現存しない生物や架空のものであったが、近年では3DCGで表現するには最も難しい対象であるといわれる人間にまで広がっている。

映画「バットマン・フォーエバー」では、俳優の存在価値という問題から、3DCGで合成されたバットマンが歩くシーンがカットされたのは有名な話である。逆にいうと、それだけリアリティーのある映像を作り出すことが可能になったといえる。

一方で、ゲーム等においても現在では3DCGを利用しているものがほとんどである。映画ほどのリアリティーは無いものの、人のキャラクターを3DCGで表現し、それらをなめらかな高速のアニメーションとして表示している。

このように、コンピュータを用いて3DCGで表現された人体モデルの動作を合成することを人体モーション合成と呼ぶが、従来の人体モーション合成システムは、特定の動作合成を目的に構築されたものが多く、また、予め定まっている人間の動作をCGアニメーション化することを主眼にして開発されていて、人体モデルが仮想空間内で周りのオブジェクトとインタラクションできるような機能を重視して開発されているシステムは少ない。そのため、仮想空間に新たに追加したオブジェクトや既に存在しているオブジェクトに対して人体モデルの新たな動作を合成できるようにシステムを拡張することが困難である。

そこで、本研究では、仮想空間内の人体モデルとオブジェクトとの自由なインタラクションが合成可能であり、人体モデルの新たな動作を合成することが可能な人体モーション合成システムの開発を目指す。

仮想空間において人体モーションを合成することは、実世界での人の動作をコンピュータ上で実現することである。そのためには、人が動作を行う場合に、世界をどのように知覚し、認知して動作するのかを知ることが重要である。よって、本研究では認知科学における概念を利用することを試みる。

認知科学には様々な概念があるが、その一つであるアフォーダンス^{[1][2]}の概念では、情報は人間をとりまく環境そのものの中に実在しており、我々はその情報を直接知覚するとしている。この主張は、我々が深く考えることもなく環境の変化や未知の環境に無意識に適応していることをよく説明している。このアフォーダンスの概念を人体モーション合成に適用すれば、人体モデルと環境とのインタラクションを自由に合成することが可能になると期待される。

以上より、本研究では人体モーション合成にアフォーダンスの概念を利用し、仮想空間内で人体モデルとオブジェクトとの自由なインタラクションを合成することができ、人体モデルが新たな動作を行うことができる人体モーション合成システムを開発することを目的とする。

以下に、本論文の構成について述べる。まず、第2章では、本研究の背景について述べた後、本研究に関する従来研究についてまとめ、本研究の目的を明らかにする。第3章では、認知科学の分野におけるアフォーダンスの概念について説明し、それを人体モーション合成に適用する方法を述べる。そして、アフォーダンスの概念を利用した人体モーション合成システムに求められる仕様についてまとめる。第4章では、実際に人体モーション合成システムをどのように設計するかについて説明する。第5章では、構築したシステムを用いて実際にアニメーションを作成し、システムの評価を行い、今後の展望について述べる。最後に、第6章で本研究の結論を述べる。

第 2 章 研究の背景と目的

本章ではまず、本研究の背景となるヒューマンインタフェースについて説明し、本研究室で研究開発中の新しいインタフェース環境であるバーチャルコラボレータについて説明する。次に、バーチャルコラボレータにおける本研究の位置付けについて説明した後、人体モーション合成の利用対象について説明する。そして、人体モーション合成の従来手法とその問題点について説明し、最後に、本研究の目的をまとめる。

2.1 ヒューマンインタフェースとバーチャルコラボレータ

インタフェースはもともと機械と機械の接続部の意味で、接続端子、接続回路から信号の変換機、変換ソフトウェアもまたインタフェースと呼ばれる。ヒューマンインタフェース (HI) は、人と機械あるいはコンピュータの接続部を意味し、キーボードやマウス、またはこれを使った漢字変換ソフト、マウスとメニューを使った表計算やグラフ作成ソフトも HI である。しかし、コンピュータが目に見える形よりは、ものの影で力を発揮する場面が多くなっており、HI はコンピュータに限らず、様々な機械に組み込まれたデジタル技術の広い応用を視野に入れている。そこで HI とは、コンピュータやその他の機械を使って人々の交わりを支援する技術ということができる [3]。

この HI の変遷を述べると、コンピュータの操作は第一世代の文字・記号だけの入出力の不便なキャラクタユーザインタフェースから、第二世代のマウスで操作し、グラフやウインドウで表示表示するグラフィカルユーザインタフェースとなり、使い勝手が良くなった。しかし、マウスクリックによる直接指示は煩わしく、また誰にでも操作できるものではない。誰にとってもバリアフリーであり、人がしたいことだけを伝え、細かなことはコンピュータが意を汲み取って代行してくれる、間接指示ができることが理想である。これが第三世代の知的インタフェースである。マルチメディア (文字、図表、動画、音声、ジェスチャ、脳波、視線、ウインク、マウス等) によって、ユーザは自分の意志だけを伝え、それがたとえ曖昧でも、コンピュータはユーザの意図を汲み取って適応してくれる。この第三世代のインタフェースの一つとして、エージェントシステムが挙げられる。エージェントとは、自律的かつ協調的に振舞うことにより人間の活動の支援を行うソフトウェアのことである。

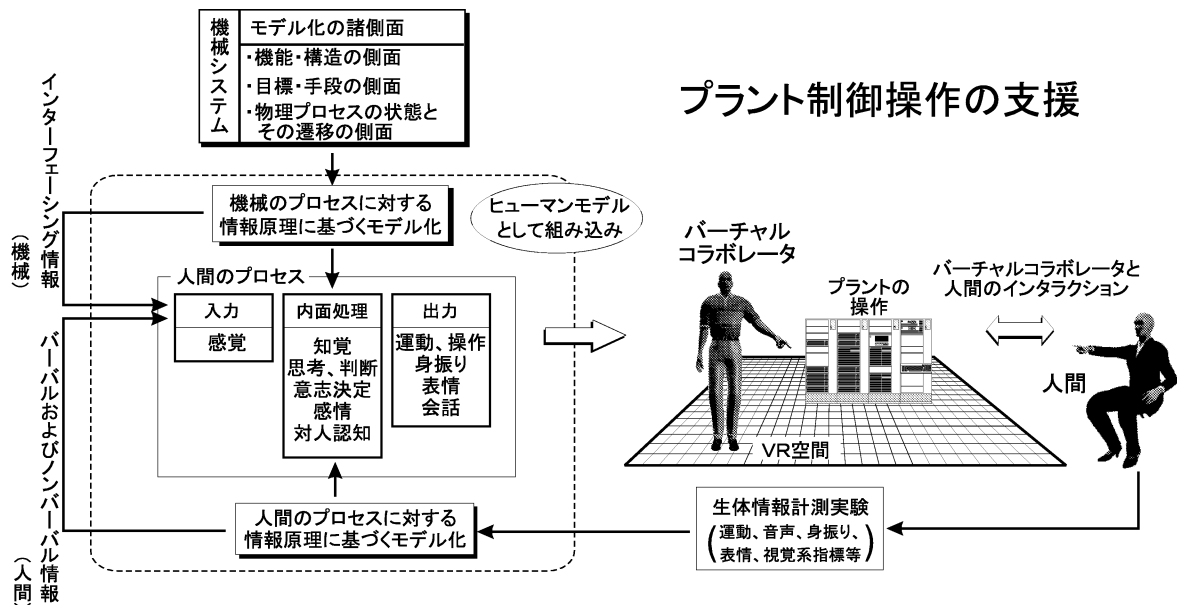


図 2.1: バーチャルコラボレータ構成図

本研究室では、このエージェントシステムをプラントシステム操作に適用した、仮想映像空間内の知的ロボットである「バーチャルコラボレータ」を構築することを目指している [4][5]。このバーチャルコラボレータは、仮想空間を通じて、人間と同じ形態で、同じように話し、動き、身振りをし、考え、感情を持ち、人間と自然に交流できる新しいコンセプトのインタフェースである。機械システムの具体的対象としては、原子力発電プラントを取り上げ、その制御室における制御盤操作を支援するインタフェースの構築を目指している。

このように、バーチャルコラボレータは仮想空間内に実現される知的ロボットであり、人間と同じように感覚器官、知能を持ち、仮想空間内で行動することができる。すなわち、バーチャルコラボレータは、仮想空間の中に存在するもう一人のプラント運転員であり、協調作業者である。このようなバーチャルコラボレータによるインタフェースシステムを構築するため、図 2.1 のような構成を考えている。実際に制御盤を操作する際には、図 2.1 の右側に示すように、運転員はプラント制御室を模擬した仮想空間に没入して、そのプラント制御室に配置されたバーチャルコラボレータと協調して制御盤操作を行う。本研究室では、このようなバーチャルコラボレータを実現するための要素技術として、次のような 4 つの要素技術に分けて研究している。

1. ヒューマンモデル技術

これは主にバーチャルコラボレータの頭脳を実現する部分であり、ここでのヒュー

マンモデルとは人間と同様に知覚（入力）、思考（内部処理）、行動（出力）の各機能を持ち、人間の知的活動を模擬するコンピュータモデルを意味している。バーチャルコラボレータの知的活動を駆動する内部モデルを構成するヒューマンモデルの研究を進めている。

2. インタフェースデバイス技術

インタフェースデバイスは、マン・マシンインタフェースとして使うハードウェアデバイスであり、バーチャルコラボレータと人間とが仮想空間内で円滑なコミュニケーションを図るための物理的接点となる機器である。各種情報の提示、及び人間の音声、身振り、各種生体情報を収集するデバイスの研究開発を進めている。

3. 人間情報行動認識技術

人間同士が対面してコミュニケーションを行う場合には、会話だけでなく、身振り、顔の表情など様々な「振る舞い」の観察から、相手の内面状態を認識している。人間情報行動認識技術は、このような人間の対人認知機能をバーチャルコラボレータに持たせるための技術であり、動画像処理による身振りや表情の認識、脳波測定や、皮膚温度、心拍、視点位置、瞳孔径等のリアルタイム計測である。これらの技術により人間の思考や感情を推定する方法を開拓している。

4. 仮想空間での人体モーション合成技術

ヒューマンインタフェースとしてのバーチャルコラボレータの動き、身振り、顔の表情等の立ち振る舞いが、人間が見て自然な動作に見えるほど、人とのコミュニケーションが円滑になると期待される。そこで、仮想空間内でバーチャルコラボレータが自然な動作や表情を実現するため、新しい人体モーション合成方法の研究を進めている。

本研究では、上記の要素技術のうち、4.の仮想空間での人体モーション合成技術に関するもので、バーチャルコラボレータの自然な動作の実現を目指す。具体的には、ヒューマンモデルからの出力に応じてバーチャルコラボレータの動作をリアルタイムに合成する人体モーション合成システムの開発を目指す。

2.2 人体モーション合成

本節では、まず、人体モーション合成がどのような分野で利用されているかを概説し、次いで従来の人体モーション合成手法とその問題点を述べる。

2.2.1 人体モーション合成の利用

コンピュータを用いる人体モーション合成の最も身近な利用対象には、映画やゲームが挙げられる。数年前までは、人体モーション合成を用いて人の動作を合成することは困難であったが、近年では様々な映画の製作に人体モーションが利用され、現実には撮影が不可能なシーンや危険を伴うシーンの作製に用いられている。ゲームに関しても、3DCGで表現されたキャラクターの動作を人体モーション合成を利用して作製することが一般的になってきている。

その他には、人体モーション合成を利用して人間の動作をシミュレーションし、様々な製品や環境の評価に利用されている。具体的には、仮想空間の中に評価対象とする製品や作業環境を精密に構築し、その中に精密に合成された人体モデルを配置する。そして、人体モデルが製品を操作する動作をシミュレーションすることによって、その製品の安全性や保守性の評価を行う。また、評価対象として製造ラインを仮想空間内に構築し、その中で人間の作業をシミュレーションすることによって製造ラインにおける作業者の安全性や肉体的な負荷を評価することもできる。このように、仮想空間内に製品や作業環境を構築し、人間の動作をシミュレーションすることによって、実際に製品や作業環境を製作・構築する前にそれを評価することができ、製品の改良や作業環境の改善、コスト削減等に役立たせることができる。

また、医学の分野においては、先天的に歩行障害を持つ人のリハビリテーションに利用されている。健常者の歩行動作を人体モーション合成によって仮想空間内で映像化し、歩行動作がどのように行われ、どのような筋肉がどのように用いられているのかを解析する。そして、その結果をリハビリテーションに応用するのである。

このように、人体モーション合成は様々な分野で利用されている。

2.2.2 人体モーション合成に関する従来手法とその問題点

2.2.1項で述べたように、人体モーション合成は様々な分野で利用されており、人体モーション合成手法もその利用目的に応じて様々である。人体モーション合成手法をまとめると、主に以下の3つの手法に分類される^[6]。

- アニメータベースの手法
- アクターベースの手法
- ユーザベースの手法

アニメータベースの手法は、コンピュータグラフィックス（Computer Graphics; 以下CG）アニメーションの製作者（アニメータ）がリアリティーのある人の動きをCGアニメーションとして合成する際に用いる手法であり、複雑で計算量の多いアルゴリズムが利用されている。この手法で作製されたアニメーションは、一旦フィルムかビデオに記録された後に再生される。そのため、リアルタイムでアニメーションを合成する必要はなく、より自然な動作を合成するための複雑なアルゴリズムを使用することができ、リアルなCGアニメーションを作成することができる。映画やCMで使用されている映像もこの手法によるものである。

アクターベースの手法は、コンピュータ世界に住む自律的なエージェントであるアクターを作成するためのものである。あらかじめ動作を合成するアルゴリズムや振る舞い方等をアクターに組み込んでおくので、アクターは自分自身の動作を自分で合成することができる。

また、ユーザベースの手法は、ユーザの指示に従い、リアルタイムに動作する“バーチャル人間”（仮想空間の中で振舞う人体模型）の動作を合成するためのものであり、動作を合成するアルゴリズムや動作データ等は予め用意しておき、ユーザは用意している動作の中から合成させたい動作を指定する。ゲームのキャラクタ等はこの方法により合成されている。

本研究では、バーチャルコラボレータの知能にあたるヒューマンモデルからの動作指示に従い、バーチャルコラボレータの動作をリアルタイムに合成する人体モーション合成システムの開発を目指しているので、上記分類の手法のうち、ユーザベースの手法が研究対象である。

このユーザベースの代表的な人体モーション合成手法としては、以下の3つの手法が挙げられる。

1. 人間の全身の動きをセンサーを用いてコンピュータに取込む3次元モーションキャプチャ技術を利用して実際の人の動きを計測し、アニメーションとして再生する^[7]。

2. ロボティクス分野における手法である逆運動学や動力学等の数式モデルを用いて、人間の姿勢や動きをシミュレーションしてアニメーションを合成する [8][9]。
3. 人間の一連の動作のうち、重要な姿勢をスポット的にキーフレームとしてあらかじめ作成しておき、キーフレームの間の動きをコンピュータにより補間計算する [10]。

1. の手法は、比較的容易に自然な動作を合成することは可能であるが、実際の人の動作の計測に手間がかかり、また計測した動作しか再生することができない。

2. の手法は、既にシミュレーション手法が確立されている動作はリアルタイムで合成することが可能であり、動作にある程度のバリエーションを持たせることも可能である。しかし、シミュレーション手法を確立することが困難であり、また自然な動作の合成も難しい。

3. の手法は、従来のセル画を用いたアニメーション分野で用いられている手法をコンピュータにより実現したものであり、簡単な動作の場合は比較的自然的な動作を合成できるが、複雑な動作ではキーフレームを作成したり、それらの姿勢の間を自然的な動作として補間するのが難しい。

これらの手法を用いた従来の人体モーション合成システムは、特定の動作合成を目的に構築されたものが多く、また、予め定まっている人間の動作をCGアニメーション化することを主眼にして開発されていて、仮想空間内で周りのオブジェクトとインタラクションできるような機能を重視して開発されているシステムは少ない。そのため、仮想空間に新たに追加したオブジェクトや既に存在しているオブジェクトに対して人体モデルが新たな動作をするような場合にも使用できるようにシステムを拡張することが困難である。また、人体モーション合成システムの利用目的を変更する場合に、その目的に応じた動作を合成可能にする必要があるが、これらの従来のシステムでは、そのシステムが用いている人体モーション合成手法に従って、新たに人の動きを計測したり、シミュレーション手法を確立しなければならず、多大な労力が必要である。

一方で、本研究室で研究開発中であるバーチャルコラボレータシステムは、仮想空間内のプラント運転員であるので、仮想空間内のプラント制御盤とのインタラクションは欠かすことはできない。その上、プラント制御盤におけるボタンや計器などの配置は、制御盤毎に異なり、その変化に対応可能なことが必要である。バーチャルコラボレータシステムに用いる人体モーション合成システムでは、仮想空間内で人体モデルとオブジェクトとの自由なインタラクションを柔軟に生成できる新しい人体モーシ

ン合成手法の創成が求められている。

2.3 研究の目的

仮想空間において人体モーションを合成することは、実世界での人の動作をコンピュータ上で実現することである。そのためには、人が動作を行う場合に、世界をどのように知覚し、認知して動作するのかが知ることが重要である。しかしながら、人体モーション合成においてはこれまで、人間の動作を如何に合成するかという観点から、生体力学やロボティクス等における手法は用いられてきたものの、認知科学で取り扱う人間の知覚や認知特性に着目して方法論を構成する試みがなかった。

そこで、本研究では、2.2.2項で述べた従来システムの問題点を解決し、仮想空間の中でバーチャルコラボレータが環境内のオブジェクトとの多様なインタラクション、多様な動作をリアルタイムで柔軟に合成することのできる新しい方法論を開拓するために、認知科学における概念を利用することを試みる。

認知科学には様々な概念があり、現在、認知過程を、人間が環境から刺激を受け、それを脳の中で処理して、外部に行動として出力するという、「情報」は人間の内面に生成されるとする情報処理モデルが主流である。これに対し、アフォーダンスの概念^[1]では、情報は人間をとりまく環境そのものの中に実在しており、我々はその情報を直接知覚するとしている。この主張は、我々が深く考えることもなく環境の変化や未知の環境に無意識に適応していることをよく説明している。

これより、本研究では認知科学の概念の中でも特に“アフォーダンスの概念”に着目し、この“アフォーダンスの概念”を上記の既存のユーザベースの人体モーション合成手法に適用することを検討する。

アフォーダンスは、視覚心理学者であった James J. Gibson により提唱された概念であり、「観察者との関係で存在する環境の特性であり、環境が観察者に提供するもの」と定義される、環境に内在する情報である^[11]。人はこのアフォーダンスを環境から直接知覚し、行動しているとされる。なお、アフォーダンスについては、後に3.1節で詳しく述べる。

以上、本研究では、2.2.2項で述べた従来の人体モーション合成システムの有する、仮想空間の拡張性に関する問題点を解決し、バーチャルコラボレータの多様な動作をリアルタイムに合成するために、認知科学におけるアフォーダンスの概念を用いた、新たな人体モーション合成手法を提案し、実際にこの手法を用いた人体モーション合成

システムを開発することを目的とする。

第 3 章 アフォーダンスの概念とその人体モーション合成システムへの適用

本章では、まずアフォーダンスの概念について説明し、次いでそれをどのように解釈することによって人体モーション合成にどのように応用するのかという基本的な考え方を述べる。そして、アフォーダンスの概念を人体モーション合成へ適用することによって生じる利点・欠点について説明する。最後に、アフォーダンスの概念に基づいた人体モーション合成システムに求められる仕様についてまとめる。

3.1 認知科学におけるアフォーダンスの概念

本節では、認知科学の分野におけるアフォーダンスの概念について説明し、アフォーダンスの知覚の仕方について説明する。

3.1.1 アフォーダンスとは

アフォーダンスは、視覚心理学者であった James J. Gibson により提唱された概念であり、「人は環境に存在する“アフォーダンス”を直接的に知覚することができる」とされる。アフォーダンスとは、“観察者との関係で存在する環境の特性であり、環境が観察者に提供するもの”と定義される、環境に内在する情報である。すなわち、アフォーダンスとは環境が人に提供する“価値”のことであり、良いものであれ、悪いものであれ、環境が人に与えるために備えているものである。

3.1.2 アフォーダンスの知覚

アフォーダンスの知覚には物体を構成する「面」が重要な役割を果たす。我々の周りの空間は、太陽や電灯などからの放射光が様々な物体の無数の面で反射し、散乱した光が充満している状態にある。散乱する光で照明されている空中のあらゆるところでは、あらゆる方向からくる光が交差している。つまり、環境の中の場所はどこでもすべての方向からの光によって“包囲”されていることになる。この環境を包囲する光を包囲光と呼ぶ。この包囲光は、それ自体が視覚にとっての“情報”であり、観察者が環

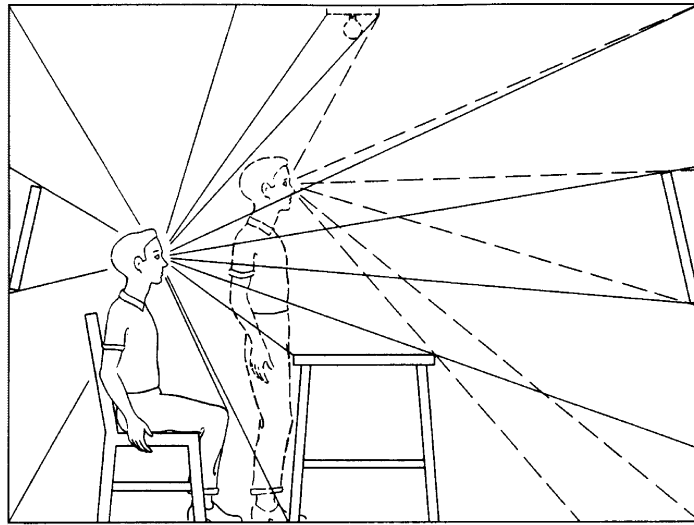


図 3.1: 包囲光配列とその変化 (J.J.Gibson: The Senses Considered as Perceptual Systems, Houghton Mifflin(1996) より)

境の中にいるとき、観察する位置を包囲する光には、図 3.1 に示したように、環境を構成する面に応じた構造があり、この包囲光の“異質性”は“包囲光配列”と呼ばれる。図 3.1 に示されるように、包囲光は対応する面や肌理を、観察者に対して1つの角度（これを“立体角”と呼ぶ）として表している。したがって、包囲光配列は、一群の立体角がなす構造でもある。観察者が動くと、この包囲光配列も変化して、面の形が変わるだけでなく、消失したり新しい面が出現したりする。

この包囲光配列の変化は環境の中の構造的に不変なものを明らかにする。例えば、ある視点からは窓がひずんだ四辺形に見えても、視点移動によっていずれ長方形という不変的性質を持つことが判明するのである。この不変なものは“不変項”と呼ばれる。見るということ観察者が行っていることは、包囲光配列から不変項をピックアップすることである。

我々を包囲する光の構造には、もう1つの重要な情報がある。それは観察者自身についての情報である。人は包囲光配列の変化から、自分自身の姿勢や移動の方向、速度や加速度に関する情報を得る。つまり、環境を知覚することと自己を知覚することは相補的であるといえる。たとえば、人がある隙間をすり抜けられるかどうかを知覚するのは、自分の肩幅の1.3倍を基準としていることが実験から示されている。この“1.3”という値は、生き物を基準にして表したものであり、“生態学的値”と呼ばれる。人はこのような生態学的値を多数っており、このことは、観察者が環境の面の配置から見ているものが、観察者の身体にとっての“意味”や“価値”であることを示している。

以上で述べたように、人は環境の中から包囲光配列の不変項をピックアップすると同時に、自分自身に関する情報を得ている。そして、環境から観察者自身にとっての意味や価値を直接知覚しているのである。この“観察者にとっての環境の意味や価値”が“アフォーダンス”である [2]。

3.1.3 アフォーダンスの特性

次に、アフォーダンスの特性について述べる。これまで述べてきたように、アフォーダンスは事物の物理的な性質ではない。それは“人にとっての環境の性質”である。アフォーダンスは観察者の主観が構成するものでもない。観察者にその意志があるか否かとは無関係に、眼前の溝はまたいで渡れるか否かどちらかが決まっているからである。それは環境の中に実在する、観察者にとって価値のある情報である。

物体、物質、場所、事象、他の動物、そして人工物など環境の中にあるすべてのものはアフォーダンスを持つ。環境にあるものは、すべてアフォーダンスの用語で記述することができる。

例えば、1枚の紙について考える。この紙は、破ることをアフォードする。しかし紙が“厚いダンボールの小さな切れ端”ならば破ることをアフォードしない。つまり破れないと知覚されるだろう。その他にも、紙は文字を書いたり絵を描いたりすることや、物を包んだり、あるいは丸めて遠くへ飛ばすことなど様々なことをアフォードする。このように、一つの物に対して非常に多くのアフォーダンスが存在する。

また、生態学的値は人によって異なるので、同じ物でも人によって異なるアフォーダンスが知覚される。ある幅の隙間について考えると、その隙間は太った人には通りぬけられないことをアフォードするが、やせた人には通りぬけられることをアフォードする。また、階段について考えると、これは昇り降りのアフォードするものであるが、乳幼児には行き止まりであることをアフォードする。すなわち、観察者の立場や特性が変わると環境のアフォーダンスも変化するのである。

このようなアフォーダンスが知覚できるためにはあらかじめ学習や経験が必要なことがある。また、これらのアフォーダンスは分析して得られるのではなくて直接知覚される。アフォーダンスの特性を表 3.1 にまとめる。

表 3.1: アフォーダンスの特性

	特性
1	物理的な性質ではない
2	環境が観察者に提供するもので、環境に内在する情報である
3	主観によって変わることはない
4	観察者にとって意味を持つ環境の価値である
5	環境の中にあるものすべてがアフォーダンスを持つ
6	一つの物に対して非常に多くのアフォーダンスが存在する
7	観察者の立場や特性が変わると環境のアフォーダンスも変化する
8	アフォーダンスは直接知覚される

3.2 アフォーダンスの概念の人体モーション合成への応用

3.1節で述べたように、アフォーダンスは環境に内在する情報であり、人はそれを直接知覚し、それに基づいて行動している。すなわち人はアフォーダンスに従うことによって行動しているといえる。そしてその行動の要因となるアフォーダンスは、人を取り囲む環境に存在しているのである。

本節では、このアフォーダンスの概念を、人体モーション合成へと応用する手法について述べる。

3.2.1 実空間と仮想空間との対応

まず、実空間において人を取り囲む“環境”は、仮想空間においては、まさに“仮想空間”そのものが人を取り囲む環境に相当し、仮想空間内に存在する物体であるオブジェクトはすべてそのような環境を構成するものとなる。なお、仮想空間内の“バーチャル人間”は、実空間における“人間”に相当する。実空間においては、人はアフォーダンスに従うことによって行動しており、アフォーダンスを知覚した後に実際に動きを生じさせる機構は人の体である。一方、仮想空間においては、バーチャル人間に動作を生じさせる機構は、人体モーション合成システムである。

次に、アフォーダンスについて考える。実空間では、人は環境の中から包囲光配列の不変項をピックアップすると同時に、自分自身に関する情報を得ることによって、アフォーダンスを知覚している。しかし、仮想空間においては、環境の中の不変項も自分

表 3.2: 実空間と仮想空間の対応

実空間	仮想空間
環境	仮想空間
人間	バーチャル人間
体	人体モーション合成システム
アフォーダンス (環境の価値)	動作情報 (動作データと動作合成手法)

自身に関する情報も、すべて人体モーション合成システムが数値データとして保持しているためこれらを知覚する必要はない。しかし、これらの環境や自分自身に関する情報は単なる数値データでしかなく、これらの数値データを、“バーチャル人間にとって意味を持つ環境の価値”、すなわち“アフォーダンス”に変換することが必要である。しかしながら、このような数値データを、“アフォーダンス”へと変換を行うことは非常に困難である。そこで本研究では、次のように実空間におけるアフォーダンスの概念を仮想空間へ拡張する。

アフォーダンスは、“観察者にとって意味を持つ環境の価値”であり、人に動作を誘発させるものである。仮想空間においては、バーチャル人間に動作を誘発させるもの、すなわちバーチャル人間の動作を駆動する元となるものは、バーチャル人間の動作情報、すなわち動作データと動作合成手法（動作合成アルゴリズム）と考えることができる。よって、実空間におけるアフォーダンスは、仮想空間におけるバーチャル人間の動作情報に相当するといえる。そこで、本研究ではバーチャル人間の動作情報を仮想空間におけるアフォーダンスと捉え、これらをバーチャル人間にすべて持たせるのではなく、バーチャル人間を取り巻く物体に持たせる方法で再構成することを考えた。

表 3.2 に、実空間と仮想空間との対応についてまとめる。

3.2.2 アフォーダンスの概念の人体モーション合成システムへの適用

3.2.1 項で述べたように、実空間と仮想空間を対応づけることにより、アフォーダンスの概念を人体モーション合成システムへと適用することが可能となる。

アフォーダンスは環境に内在する情報であるので、仮想空間において実空間のアフォーダンスに相当するバーチャル人間の動作情報を、仮想空間内に存在する個々のオブジェクトに保持させることによって、アフォーダンスの概念を人体モーション合成システ

表 3.3: テーブル、椅子、コップのアフォーダンス

物体の名称	アフォードする動作
テーブル	上に物を置く
椅子	座る
コップ	手に持つ

ムへと適用することができる。

しかし、このような適用方法では、アフォーダンスの概念の持つ特性のすべてを仮想空間へと拡張することはできない。アフォーダンスは、環境に内在する情報であり、ここでいう環境とは、その中に存在する物体のみならず、それらの位置関係や状態をも含むものである。そして、それらの位置関係や状態から知覚されるアフォーダンスも存在するのである。

たとえば、ある環境にテーブルと椅子、そしてコップがあるとすると。そして、それぞれの物体が個々に表 3.3 に示す動作をアフォードしているとする。しかし、テーブルが椅子のそばにある場合には、この環境は全体として、表 3.3 に示した動作とは別に、たとえば「椅子に座りつつテーブルの上にコップを置く」という、いわば「椅子に座る」動作と「テーブルの上にコップを置く」という動作を融合した動作をアフォードする。

このような、環境内の複数の物体の組み合わせが持つアフォーダンスは、上記の適用方法では表現することができない。

そこで本研究では、このような環境内の複数の物体の組み合わせが持つアフォーダンスを、環境を構成する物体がそれぞれ持つアフォーダンスを組み合わせたものとして捉える。そして、このようなアフォーダンスを人体モーション合成システムで表現する代替手法として、それぞれのオブジェクトがアフォードする複数の動作を融合することにより実現する。

この環境内の複数の物体の組み合わせが持つアフォーダンスは、実際には単純なものではなく、個々のアフォーダンスの組み合わせ以上のものをアフォードする場合もある。例えば、テーブルと椅子が適切な位置関係にあるときには、この椅子とテーブルの組み合わせは「椅子に座ってテーブルに足を架ける」動作をアフォードするが、このアフォーダンスは、個々のオブジェクトが持つアフォーダンスを単純に組み合わせるだけでは表現できない。本研究では、このようなアフォーダンスは考慮に入れずに、単なる個々のアフォーダンスの組み合わせであるアフォーダンスを複数動作の融合に

よって実現する。

3.2.3 オブジェクト指向との関係

本研究では、3.2.2項で述べたように、バーチャル人間の動作情報を仮想空間内に存在する個々のオブジェクトに保持させることによって、アフォーダンスの概念を人体モーション合成システムへと適用する。この手法は、ソフトウェア工学におけるオブジェクト指向の考え方と非常に似通っており、ここで、オブジェクト指向の考え方との関係を整理しておく。なお本項において以下では、オブジェクト指向におけるオブジェクトの概念を“「オブジェクト」”と表記し、仮想空間内のオブジェクトを“オブジェクト（物）”と表記する。

オブジェクト指向では、システムは相互にメッセージをやりとりする多くの「オブジェクト」（オブジェクトにはデータとそれを処理するプロセスとがカプセル化されている）の集まりとしてモデル化される^{[12][13]}。人体モーション合成システムをオブジェクト指向の考え方に沿ってモデル化すると、次のようになる。

一般的なモデル化では、バーチャル人間や仮想空間に存在するオブジェクト（物）を、それぞれ1つの「オブジェクト」として捉える。このようなモデル化では、バーチャル人間に関する処理やデータのすべてをバーチャル人間の「オブジェクト」に、仮想空間内のオブジェクト（物）に関する処理やデータのすべてをオブジェクト（物）の「オブジェクト」に持たせることになる。

一方、本研究の手法では、仮想空間内のオブジェクト（物）に関する処理やデータをオブジェクト（物）自身に保持させるのであるが、バーチャル人間に関する処理やデータも操作対象となるオブジェクト（物）に保持させるようにする。

このように、アフォーダンスの概念に沿ったモデル化と一般のオブジェクト指向は異なるものである。

3.3 アフォーダンスの概念の導入による利点・欠点

次に、3.2.2項のように、アフォーダンスの概念を人体モーション合成に適用することによって生じる利点・欠点について以下に述べる。

3.3.1 アフォーダンス導入による利点

人体モーション合成にアフォーダンスの概念を導入することによる利点としては、仮想空間に新たに追加したオブジェクトや既に存在しているオブジェクトに対するバーチャル人間の新たな動作を合成することが容易に行えるということである。

従来の人体モーション合成の考え方では、バーチャル人間が仮想空間内のオブジェクトに対する動作情報をすべて保持している。そのため、図 3.2 上に示すように、仮想空間に新たにオブジェクト椅子 2 を追加する際には、バーチャル人間に椅子 2 に関する動作情報を追加する必要がある。これは、人体モーション合成システムを変更しなければならないことを意味する。

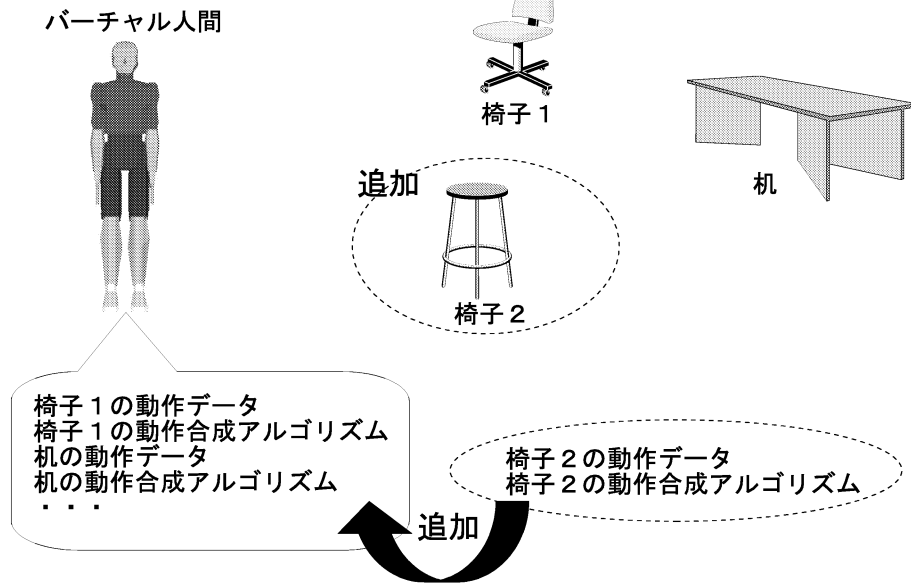
一方、本研究では、バーチャル人間の動作情報をその操作対象であるオブジェクトに持たせている。そのため、図 3.2 下に示すように、仮想空間に新たにオブジェクト椅子 2 を追加すれば、同時に椅子 2 に関する動作情報も追加される。このように、アフォーダンスの概念を導入することによって、仮想空間に新たに追加したオブジェクトや既に存在しているオブジェクトに対するバーチャル人間の新たな動作を合成することが容易に行える。これは、アフォーダンスの概念を利用することによって生じる最大の利点である。

その他の利点として、人体モーション合成における「組み合わせ爆発」を回避できることが挙げられる。

従来の人体モーション合成システム構築手法では、すべての動作情報をバーチャル人間に保持させているため、複雑な仮想空間内でバーチャル人間の複雑な動作を合成する際には、バーチャル人間に保持させる動作情報の量が非常に多くなる。例えば、仮想プラント運転室内で、バーチャル人間の「制御盤上の計器 A を見ながらボタン B を押す」という動作を合成する場合、制御盤上には、無数の計器とボタンが配置されている。これら計器とボタンのすべての組み合わせに関して、このような動作を合成しようとする、「組み合わせ爆発」が生じ、バーチャル人間に保持させる動作情報の量が莫大なものになってしまう。また、このように動作情報の量が非常に多い場合には、保持する動作情報の中から目的とする動作情報を検索するために必要な時間も長くなってしまふ。

一方、アフォーダンスの概念を導入し、バーチャル人間の動作情報を仮想空間内のオブジェクトに持たせると、バーチャル人間の動作情報は操作対象であるそれぞれのオブジェクトに分散することになる。これにより、各オブジェクトが保持する動作情報

従来の人体モーション合成



アフォーダンスに基づく人体モーション合成

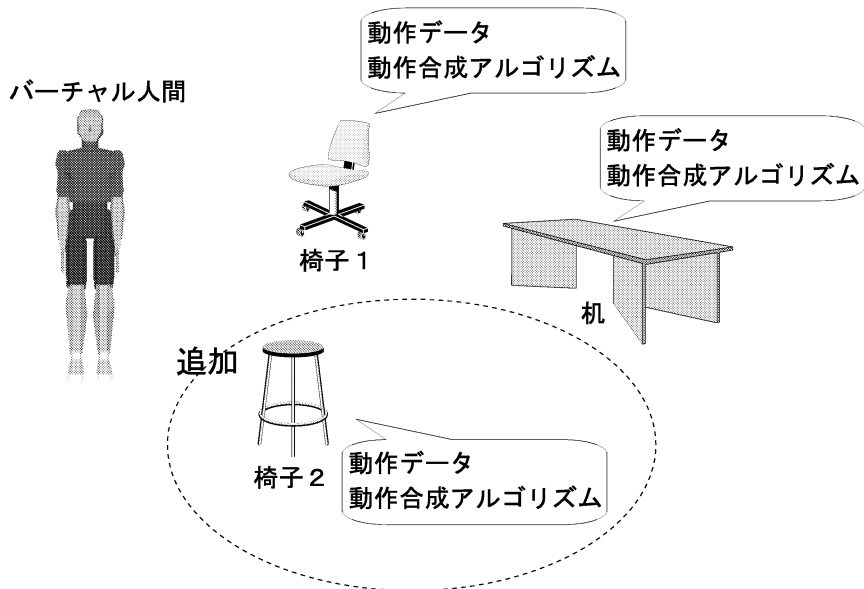


図 3.2: 仮想空間へのオブジェクトの追加

の量は、バーチャル人間にすべての動作情報を保持させた場合と比較すると、格段に少なく済む、「組み合わせ爆発」は生じない。また、動作情報の検索時間に関しても、1つのオブジェクトがアフォードする動作の種類はさほど多くはないので短くて済む。

上記のような、バーチャル人間が保持しなければならない動作情報の量が増大することによって生じる問題は、人工知能（AI）の分野におけるフレーム問題や、古典的な運動制御理論におけるベルンシュタイン問題に対応するものである。

AIの分野におけるフレーム問題とは、簡単に言うと、ある行為に関連することと、関連しないことを効率的に見分けるにはどのようにすればよいか、という問題である。AIの分野では、チェスのようなゲームや、定理の証明のように、問題の生ずる「環境」が限定されていて、「推論」の力が最大に発揮されるような課題を対象としている。このような場合には、「行為」（コマの動き）に先立って、その行為と直接には関係しない「環境」の状態（他のコマの盤面での配置）についての完全な目録を作り、それをいちいち書き換えて、次の動作のプランに用いていた。しかし、このような手法では、「環境」が限定されない場合、すなわち対象を「現実の世界」に移した場合に、ある行為に付随して生じる環境の変化の組み合わせをすべて推論しようとして、「組み合わせ爆発」が生じて計算量が爆発的に増加し、思考停止状態に陥ってしまう。

次に、ベルンシュタイン問題について述べる。古典的な運動制御の理論では、映画の原理と同様に、脳が1コマ毎の身体各部の配置を指令すれば、それをつなげることで身体の運動を制御できると考えられていた。これを基礎にする伝統的なモデルでは、運動についての記憶を貯蔵している脳の部位から、適切な運動のプログラムを検索してきて、皮質の運動野にある各筋・関節の動きに対応した「鍵盤」でそのプログラムを「演奏する」という形で運動が制御されると説明した。このモデルの特徴は、運動を制御する原因、すなわち運動のプランが実行に先立って脳内でつくられることを仮定している点である。しかし、個々の関節や筋肉の状態を特定化する方法で運動を制御しようとする、決定すべきことがあまりにも多くなってしまふ。例えば人間の腕では、筋の運動ユニットレベルでも2600の自由度があり、制御すべきことが膨大になって「組み合わせ爆発」が生じてしまい、現実的ではない。

以上、従来の人体モーション合成システムの構築方法における問題、フレーム問題、ベルンシュタイン問題の対応をまとめると、表3.4になる。

このように、アフォードダンスの概念を人体モーション合成手法へ導入することにより、以上に挙げたように、合成可能な動作の種類を容易に増やすことができる、フ

表 3.4: 各領域の問題の対応関係

問題名称	目的	生じる問題
人体モーション合成における問題	動作の選択	バーチャル人間が保持すべき情報量が莫大なものとなり、目的とする動作情報の検索時間が長くなる
フレーム問題	環境の理解	ある行為に付随して生じる環境の変化の組み合わせをすべて推論しようとして思考停止状態に陥る
ベルンシュタイン問題	運動制御	制御すべきことが膨大になってしまう

フレーム問題を回避できる、という2つの利点が生じる。

3.3.2 アフォーダンス導入による欠点

一方で、アフォーダンス導入による欠点は、仮想空間内に同じオブジェクトが複数存在する場合に、動作情報も同じ数だけ存在し、動作情報の量が非常に多くなってしまふという点である。例えば、仮想空間内にプラントの制御盤があるとする。制御盤には多数のボタンがあるが、そのボタン一つ一つが同じ“押す”という動作情報を重複して持っているのである。

しかしこの問題は、人体モーション合成システムを構築する際に、同種のオブジェクト間で、動作情報の共通部分を共有することによって、ある程度は解消することが可能である。

3.4 アフォーダンスの概念に基づく人体モーション合成システムの要求仕様

本節では、本研究で開発する人体モーション合成システムに要求される仕様について述べる。

本研究の目的は、バーチャルコラボレータの動作、すなわち発電プラントの運転員の行う様々な動作を仮想空間内で合成することである。これに基づき、本研究で開発する人体モーション合成システムに要求される仕様を表 3.5 にまとめる。

表 3.5: 人体モーション合成システムの仕様

仕様 1	発電プラントにおける運転員の動作を合成できる
仕様 2	バーチャル人間の動作に応じて仮想空間内のオブジェクトが移動する
仕様 3	バーチャルコラボレータの知能にあたるヒューマンモデルからの動作指示に従って動作を合成する
仕様 4	リアルタイムで動作を合成できる
仕様 5	プラント制御盤の設計変更に対応できる
仕様 6	複数のバーチャルコラボレータの異なる動作を同時に合成できる
仕様 7	複数の動作を融合できる

以下で、それぞれの仕様についてその仕様を実現しなければならない理由と本研究での実現方法について説明する。

- 仕様 1 : 発電プラントにおける運転員の動作を合成できる

仕様 1 は、本研究における人体モーション合成システム開発の直接の目的が、発電プラントの運転員を模擬するバーチャルコラボレータのプラント中央制御室内の制御動作の合成であることによる。

発電プラントの運転員が行う代表的な動作としては、以下のようなものが挙げられる。

- ボタンを押す
- レバーをスライドさせる
- キーボードを操作する
- 計器の値を確認する
- マニュアルを読む
- 指差し確認をする
- タッチディスプレイパネルを操作する
- 制御室内を移動する
- ジェスチャによって感情や意志を表出する

上記のように、発電プラントの運転員が行う動作の種類は少ないものの、発電プラントの制御盤には様々な位置や角度で多数のボタンやレバー、計器が配置されているため、これらの動作はその操作対象の位置や角度の変化に対応できることが必要となる。

この仕様を満たすためには、それぞれの機器に対する動作の動作データと動作合成手法を用意すれば良い。一方、本研究ではアフォーダンスの概念を導入し、バーチャル人間の動作情報を操作対象のオブジェクトに持たせている。そのため、同種のオブジェクトが多数存在する場合には、3.3.2項で述べたように、動作情報、すなわち動作データと動作合成手法の量が非常に多くなってしまいうという問題が生じる。さらに、オブジェクトに関する情報である、オブジェクトの形状データや大きさ、重さなどのデータも、同種のオブジェクト間では共通であり、同じデータが複数存在するという問題もある。

そこで、3.3.2項で述べたように、動作情報のすべてを操作対象のオブジェクトに保持させるのではなく、同種のオブジェクト間で動作情報の共通部分を共有すると共に、オブジェクトに関する情報も、同種のオブジェクト間で共通部分を共有させるようにする。

例として、仮想空間内に複数のボタンがあり、“ボタンを押す”という動作を合成する場合について考える。

動作情報は具体的には、動作データと、動作データを元に動作を合成する動作合成アルゴリズムから成る。そして、“ボタンを押す”という動作では、動作データは“ボタンの仮想空間内の位置・姿勢”であり、動作合成アルゴリズムは“指定された位置まで右手を伸ばし、その位置を指で押す”アルゴリズムであるとする。この場合、動作データは同種のオブジェクト間で共有することはできないが、動作合成アルゴリズムは同種のオブジェクト間で共有可能である。また、ボタン自身の情報であるボタンの大きさ、重さ、形状データも同種のオブジェクト間で共有することができる。

この手法の概念図を図 3.3 に示す。図 3.3 では、上側が単純に操作対象のオブジェクトにバーチャル人間の動作情報を持たせた場合の、2つの同種のオブジェクト A、B におけるデータ構成を示している。オブジェクト A、オブジェクト B はそれぞれ、動作情報とオブジェクト自身の情報を保持している。しかし、動作合成アルゴリズムと大きさ、重さ、形状データは2つのオブジェクト間での相違

はなく、同じデータを重複して保持あり、保持すべきデータ量が非常に多い。そこで、図 3.3 下側に示すように、動作情報やオブジェクトの情報の共通部分を共通データとして共有化し、データ量を削減する。オブジェクト A、オブジェクト B が個々に保持する情報としては動作データのみであり、他の情報はオブジェクト A とオブジェクト B の間で共有する。

このように、共通のデータを共有することによって、各オブジェクトが保持するデータ量を削減することができる。

- 仕様 2 : バーチャル人間の動作に応じて仮想空間内のオブジェクトが移動する

例えば、バーチャル人間が仮想空間内でプラントの運転マニュアルを手にとって移動する場合に、運転マニュアルは手に追隨して移動しなければならない。

この仕様を満たすためには、バーチャル人間の動作を合成するアルゴリズムに加えて、オブジェクトの動作を合成するアルゴリズムを用意すればよい。

- 仕様 3 : バーチャルコラボレータの頭脳にあたるヒューマンモデルからの動作指示に従って動作を合成する

仕様 3 は、本研究の目的が、バーチャルコラボレータの動作を合成する人体モーション合成システムの開発であることによる。

この仕様を満たすために、人体モーション合成システムの入力を、ヒューマンモデルからの動作指示に対応させる。ヒューマンモデルからの動作指示は、操作対象の名称、操作対象となるボタンや計器の位置 (x, y, z 座標) および操作の種類 (ボタンを押す、レバーをスライドさせる等) であり、人体モーション合成システムの入力もこれと同じものとする。

- 仕様 4 : リアルタイムで動作を合成できる

仕様 4 は、バーチャルコラボレータは人間とのインタフェースであり、人間と“実時間”でコミュニケーションしなければならないところから生じている。特に、バーチャルコラボレータの目に見える動きを映像化する人体モーションの合成はユーザへの出力インタフェースとして重要である。

ここでいうリアルタイムとは、毎秒 10 コマから 20 コマの割合で動作を合成することを意味する。

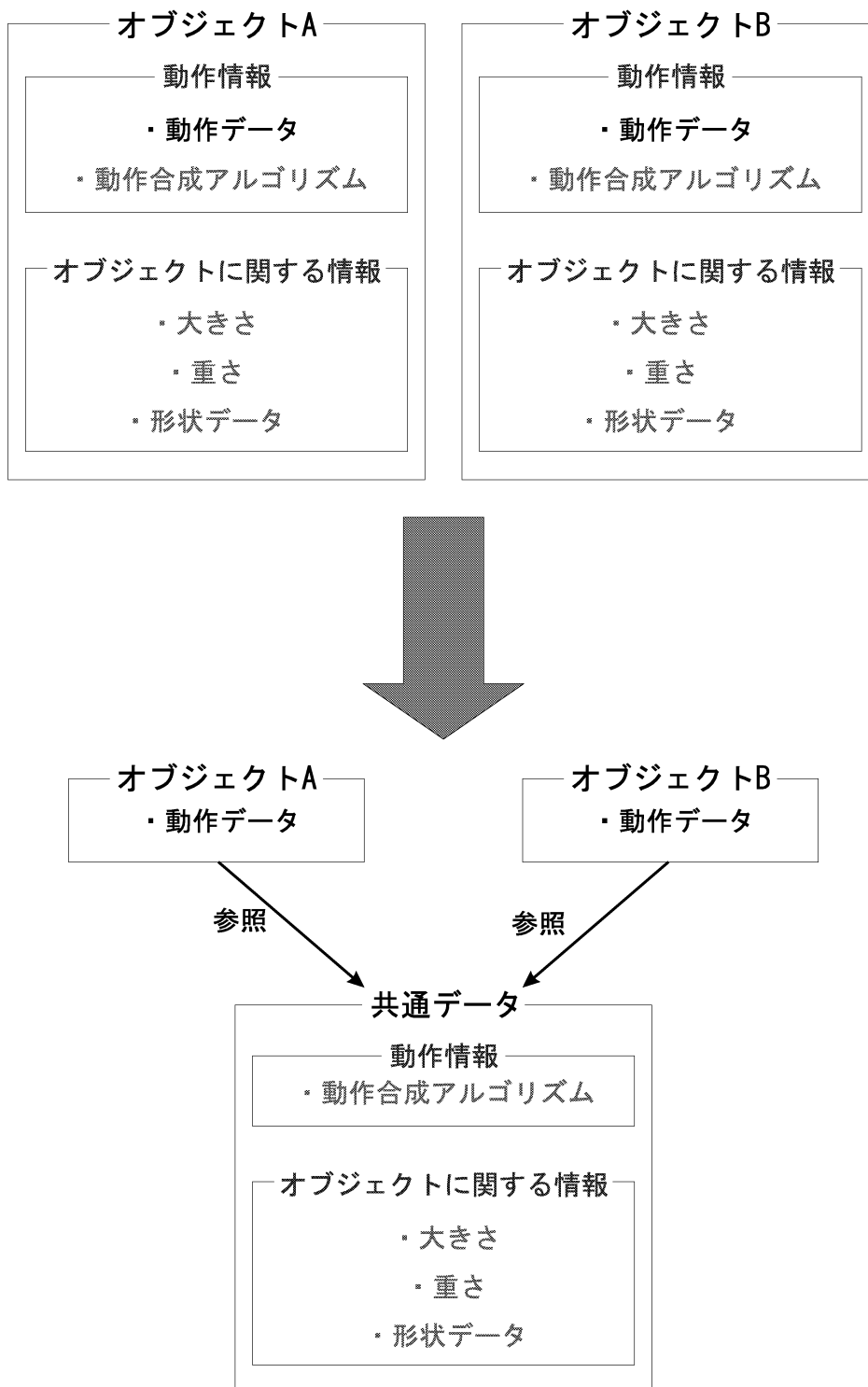


図 3.3: 動作情報の分割

この仕様を満たすために、動作データから実際にバーチャル人間の動作を合成する手法である動作合成アルゴリズムとして、短時間で動作を合成することが可能であるもの、もしくは1コマ毎に結果を画面に出力可能であり、かつ1秒間に10コマ程度の出力が可能であるものを用いる。また、計算負荷を分散させるために、動作を合成する部分と3次元画像を実際に描画する部分を分離し、異なる計算機上に実装する。

- 仕様5：プラント制御盤の設計の変更に対応できる

仕様5は、バーチャルコラボレータが発電プラントの制御盤操作を支援するインタフェースであり、操作対象となる制御盤として様々な種類のものが存在することによる。

この仕様を満たすために、本研究では人体モーション合成システムへアフォーダンスの概念を導入し、仮想空間に新たに追加したオブジェクトや既存のオブジェクトに対する新たな動作を合成できるようにする。

従来の人体モーション合成システムでは、2.2.2項で述べたように、仮想空間に新たに追加したオブジェクトや既存のオブジェクトに対する新たな動作を合成することは困難であった。これは、従来の人体モーション合成システムがこのような拡張を考慮して構築されたものではなかったからである。大きく分けると人体モーション合成システムは、ユーザからの入力、動作合成に必要なデータ（動作データ）、動作データから動作を合成する動作合成手法（動作合成アルゴリズム）、そして出力から成る。ここで、従来の人体モーション合成システムのシステム構成を図3.4に示す。

従来の人体モーション合成システムはこの図に示すように、動作データをデータベースとしてシステムとは独立に構築しているものの、動作合成アルゴリズムをシステム内部に含んでいた。故に、新たな動作を合成するためには、システム本体に含まれる動作合成アルゴリズムの部分を変更する必要がある。

そこで、本研究では3.2.2項で述べたように、人体モーション合成にアフォーダンスの概念を導入し、バーチャル人間の動作データおよび動作合成アルゴリズムをその操作対象であるオブジェクトに持たせるようにした。これにより、アフォーダンスに基づく人体モーション合成システムのシステム構成は図3.5に示すようになる。

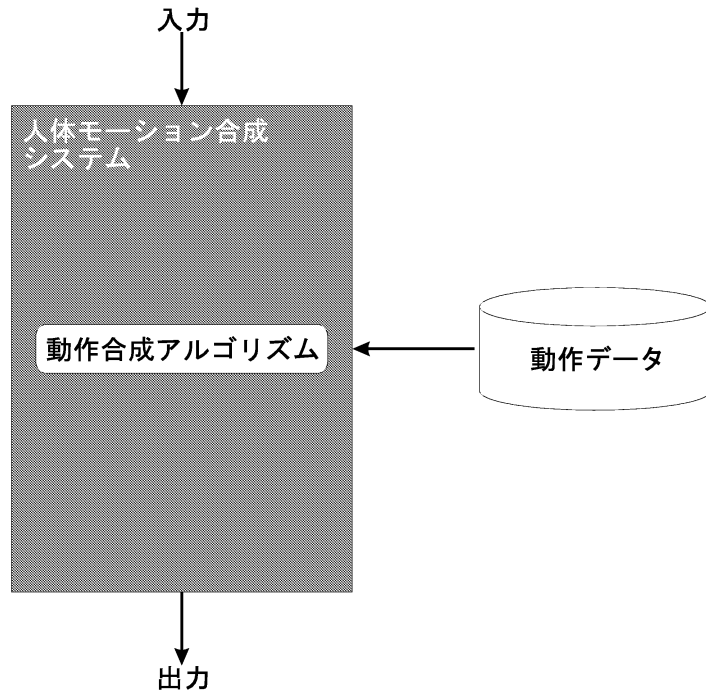


図 3.4: 従来の人体モーション合成システムのシステム構成

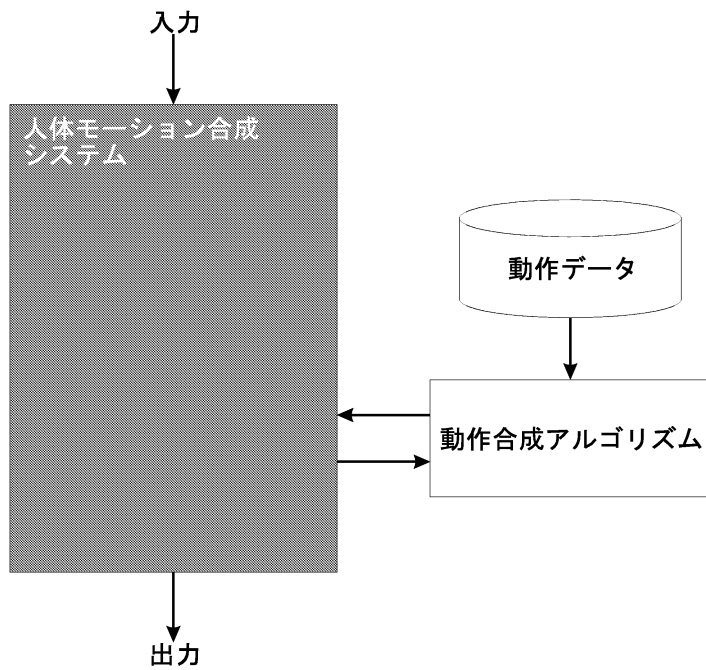


図 3.5: アフォーダンスに基づく人体モーション合成システムのシステム構成

図に示すように、アフォーダンスに基づく人体モーション合成システムでは、動作データと動作合成アルゴリズムの両方がシステムとは独立したものとなる。これによりシステムの役割は、ユーザからの入力を受け付け、データの流処理し、結果をディスプレイに出力するのみとなる。一方で、動作データと動作合成アルゴリズムがシステムと独立であることから、動作データや動作合成アルゴリズムの変更が容易に行えるようになるのである。この結果、仮想空間へ新たに追加したオブジェクトや、既存のオブジェクトに対する新たな動作の合成が、動作データや動作合成アルゴリズムを追加するだけで実現される。これは、人体モーション合成システムの構築にアフォーダンスの概念を適用することによって実現可能となるものである。

- 仕様6：複数のバーチャルコラボレータの異なった動作を同時に合成できる

仕様6は、バーチャルコラボレータの将来構想として、1つの仮想空間の中で複数のバーチャルコラボレータが協調してプラント運転員を支援するマルチバーチャルコラボレータシステムがあることによる。

そこで、この仕様を満たすために、複数のバーチャル人間の異なった動作を合成できるシステム設計をする必要がある。

- 仕様7：複数の動作を融合できる

仕様7は、3.2.2項で述べた、複数の物体の組み合わせが持つアフォーダンスを表現するためのものである。

この仕様を満たすためには、複数の物体のそれぞれがアフォードする動作を融合する機構を構築する必要がある。

以上より、アフォーダンスに基づく人体モーション合成システムに求められる仕様とその実現方法を表3.6にまとめる。

なお、表中の仕様5に関しては、アフォーダンスの概念を適用することによって実現可能となるものであり、かつアフォーダンスの概念を適用することによる最大の利点である。

表 3.6: 人体モーション合成システムに求められる仕様

	仕様	実現方法
仕様 1	プラント運転員の動作の合成	動作データと動作合成手法を、同種のオブジェクト間の共通部分と非共通部分に分離して用意
仕様 2	オブジェクトの動作の合成	オブジェクトの動作合成手法を用意
仕様 3	ヒューマンモデルの動作指示に従う	入力をヒューマンモデルの出力に対応させる
仕様 4	リアルタイムでの動作合成	計算負荷の少ないアルゴリズムを使用し、計算部と表示部を分離する
仕様 5	プラント制御盤の変更に対応	アフォーダンスの概念を導入し、仮想空間の拡張性を確保
仕様 6	複数のバーチャルコラボレータの異なった動作の合成	複数のバーチャルコラボレータの動作合成を考慮したシステム設計
仕様 7	複数の動作の融合	動作融合部分の構築

第 4 章 人体モーション合成システム (AHMSS) の開発

本章では、本研究で開発した、アフォーダンスの概念を利用した人体モーション合成システムである AHMSS (Affordance-based Human Motion Synthesizing System) について説明する。まず、バーチャル人間として仮想空間内に表示する人体モデルの構成と、その姿勢の表現手法について述べる。次に AHMSS の概要について述べる。そして、そのソフトウェア構成、ハードウェア構成、全体の処理の流れ、システムを構成する各要素における処理方法を述べる。

4.1 人体モデルの構成とその姿勢の表現

本節では、本研究でバーチャル人間として仮想空間内に表示する人体モデルの構成とその姿勢の表現方法について述べる。

4.1.1 人体モデルの構成

本研究で用いる人体モデルの構成を図 4.1 に示す。このモデルでは、人の全身の動きを表現するのに最低限必要と思われる 14 個の関節と 15 個の部位 (リンク) から構成されている。図中、二重丸 () が関節の位置を示す。また、左右の手は、図 4.2 に示すように、掌のリンクに接続された 14 個の関節と 14 個のリンクとしてモデル化している。すなわち人体モデルは、体幹の 14 関節と左右の指の各 14 関節の計 42 関節を持つ。それぞれの関節では、接続されている 2 つのリンクのうち、一方のリンクが他方のリンクに対して x, y, z 軸周りの回転の 3 つの自由度を持つ。

ここで、このモデルに図 4.3 に示すような、腰を頂点とする階層構造を設定する。図では体幹部分の階層構造のみを示しているが、掌の階層の下に、指の階層構造も同様に設定されている。このような階層構造を設定することにより、ある階層の姿勢をその 1 つ上の階層から見た相対的な姿勢として表すことが可能となる。

また、各リンクには、図 4.4 および図 4.5 に示すように、上位リンクとの接続部分である関節を原点とし、そのリンクに固定されるローカル座標系を設定する。また、ロー

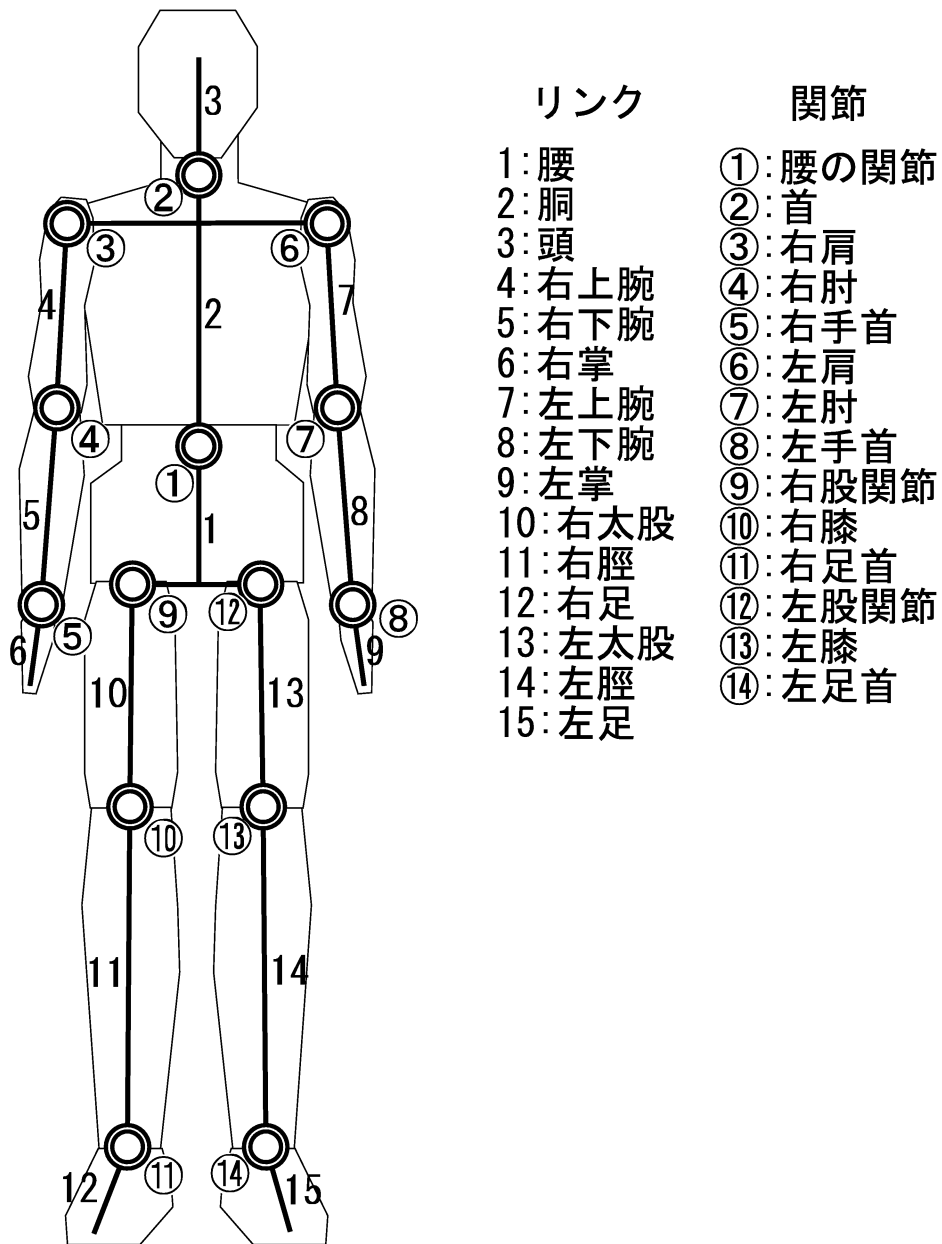


図 4.1: 人体モデルと各部名称

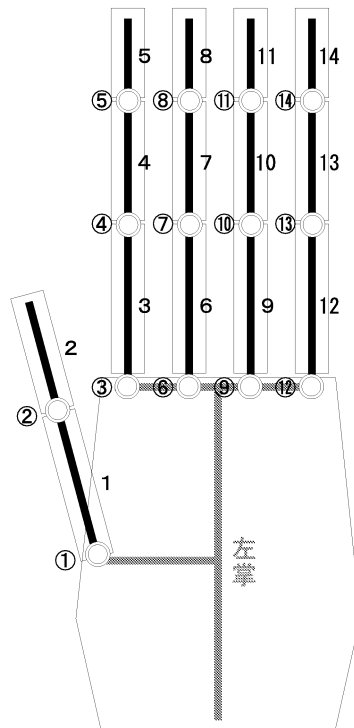


図 4.2: 指モデル

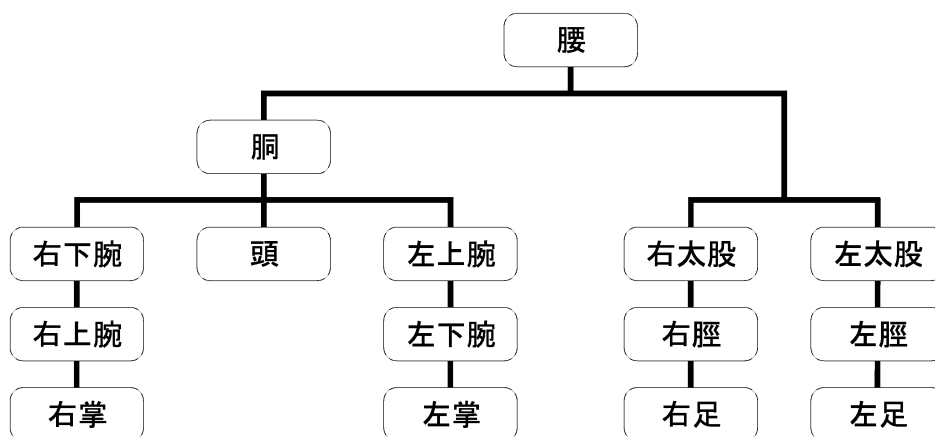


図 4.3: 人体モデルのリンクの階層構造

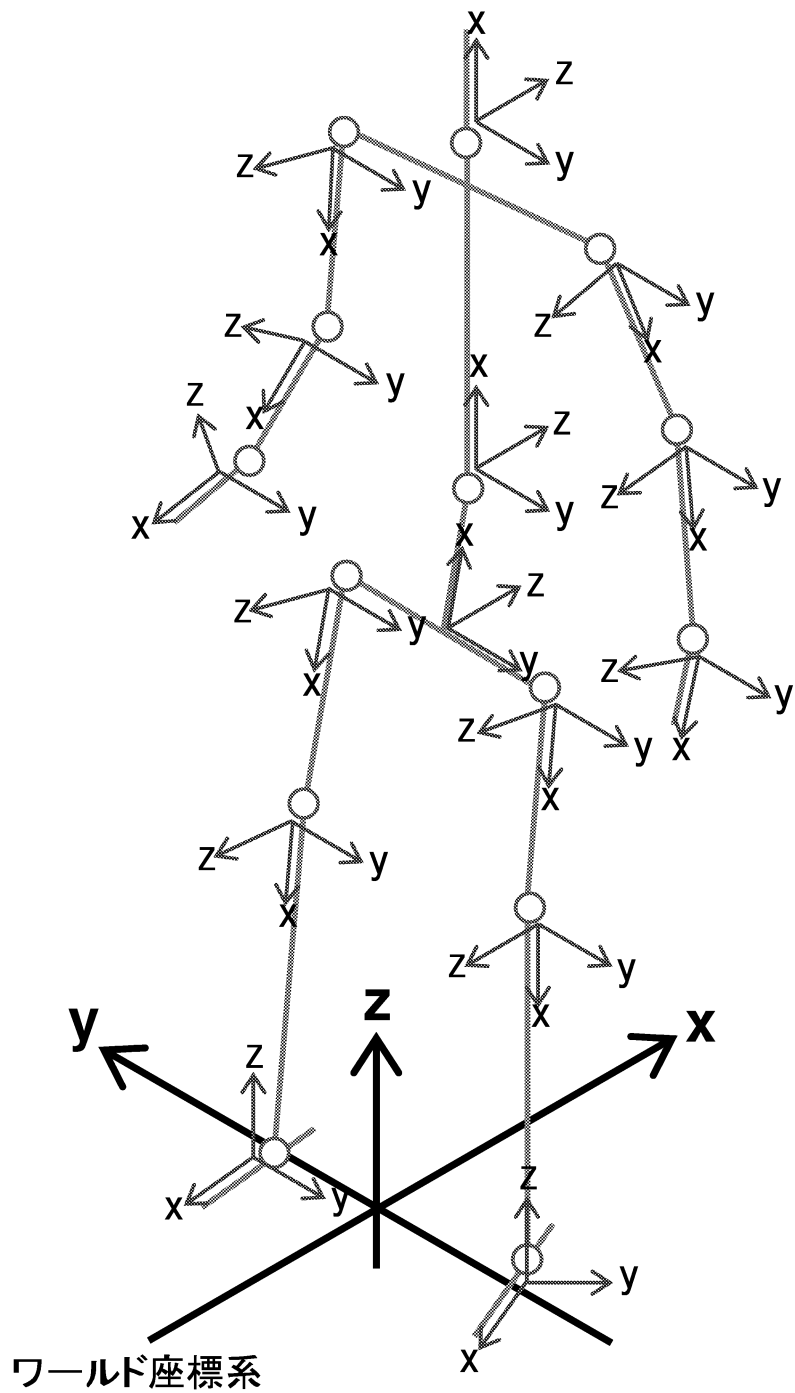


図 4.4: ワールド座標と各リンクに設定したローカル座標

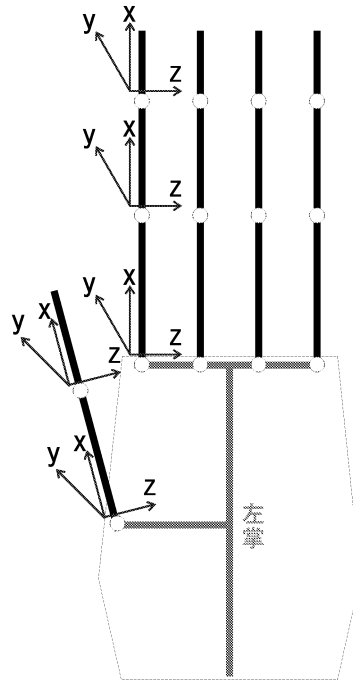


図 4.5: 指モデルのローカル座標系

カル座標系とは別に、上方向を z 軸とするワールド座標系を設定する。

このように、人体モデルに階層構造を設定し、各リンクにローカル座標系を設定することによって、人体モデルの姿勢を、ワールド座標系から腰のリンクのローカル座標系への変換と、隣接する 2 つのリンクのローカル座標系の間の変換の合計 43 個の変換を用いて記述することが可能となる。この 43 個の変換の組がアニメーションの 1 コマに相当する [14][15]。

4.1.2 姿勢の 4 元数表現

本項では、バーチャル人間を構成するリンクのうち、隣接する 2 つのリンクに設定されたローカル座標系の間の変換に関して説明する。2 つの座標系の間の変換を表現する方法は幾つか提案されているが、主なものとして以下の 3 つが挙げられる。

1. オイラー角を用いた表現方法
2. 方向余弦行列（回転行列）を用いた表現方法
3. 4 元数（Quaternion）を用いた表現方法

オイラー角を用いた表現方法は、概念が容易なため、様々な場面で用いられている。しかし、この表現方法には、一つの座標変換に対して複数通りの表現ができる特異点

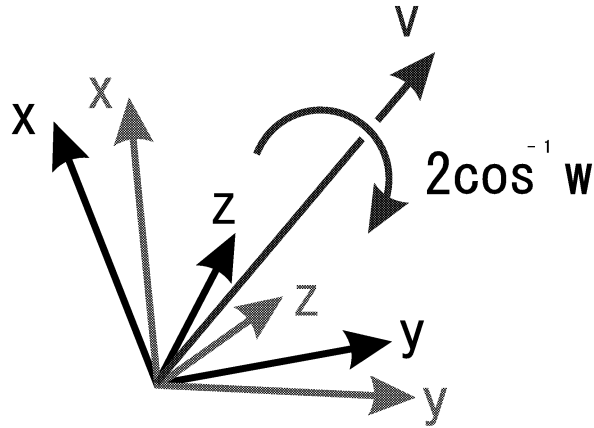


図 4.6: Quaternion が示す回転

が存在するため、表現方法が一意に定まらないという問題がある。方向余弦行列を用いた表現方法は、表現が直感的でなく、また 9 変数表現であるため非常に冗長である。

一方、Quaternion を用いた表現方法は、特異点がなく、4 変数表現であるため、冗長度が低く、Quaternion 同士の演算（補間、加算、減算等）が行い易い等の利点がある。以上の理由から、本研究では座標系の変換を表現する方法として Quaternion を採用した^{[16][17]}。

Quaternion q は、以下の 3 つの式

$$q = w + xi + yj + zk \quad (4.1)$$

$$i^2 = j^2 = k^2 = -1 \quad (4.2)$$

$$ij = -ji = k, jk = -kj = i, ki = -ik = j \quad (4.3)$$

で示される、1 つの実数 w に 3 つの虚数単位 i, j, k を付け加えたものである。以下では実数部が w 、虚数部が $\vec{v} = (x, y, z)$ の Quaternion q を、 $q = (w, x, y, z)$ もしくは $q = [w, \vec{v}]$ と記す。Quaternion は、 $|q|^2 = w^2 + x^2 + y^2 + z^2 = 1$ の時、図 4.6 に示すようにベクトル $\vec{v} = (x, y, z)$ を回転軸とした $2 \cos^{-1} w$ の回転変換を示す。

次に Quaternion に関する各種演算について以下で説明する。

- 乗算

乗算は以下のように定義される。

$$\begin{aligned} q_0 q_1 &= (w_0 + x_0 i + y_0 j + z_0 k)(w_1 + x_1 i + y_1 j + z_1 k) \\ &= (w_0 w_1 - x_0 x_1 - y_0 y_1 - z_0 z_1) \end{aligned}$$

$$\begin{aligned}
& (w_0x_1 + x_0w_1 + y_0z_1 - z_0y_1)i + \\
& (w_0y_1 - x_0z_1 + y_0w_1 + z_0x_1)j + \\
& (w_0z_1 + x_0y_1 - y_0x_1 + z_0w_1)k.
\end{aligned} \tag{4.4}$$

なお、乗算においては交換則は成り立たない。

- 共役 Quaternion

Quaternion q に共役な Quaternion q^* は次式によって定義される。

$$q^* = (w + xi + yj + zk)^* = w - xi - yj - zk. \tag{4.5}$$

- Quaternion の大きさ (ノルム)

Quaternion の大きさ (ノルム) $N(q)$ は次式によって定義される。

$$N(q) = N(w + xi + yj + zk) = w^2 + x^2 + y^2 + z^2. \tag{4.6}$$

- 逆 Quaternion

q を座標系 Σ_A から座標系 Σ_B への変換を表す Quaternion とすると、その逆変換である q^{-1} は Σ_B から座標系 Σ_A への変換を表し、次式によって定義される。

$$q^{-1} = \frac{q^*}{N(q)}. \tag{4.7}$$

- Quaternion によるベクトルの変換

ベクトル $\vec{u} = (\alpha, \beta, \gamma)$ は Quaternion で $q_u = [0, u]$ と表現でき、ベクトル \vec{u} の Quaternion q ($q^2 = 1$) による変換 \vec{u}' は、

$$q'_u = q^{-1}q_uq. \tag{4.8}$$

で計算できる。以下では、Quaternion q によるベクトル \vec{v} の変換を $q(\vec{v})$ と表す。

次に、実際に人体モデルの姿勢を仮想空間内に描画する際の計算方法について説明する。本研究では人体モデルの姿勢は、1. ワールド座標系から腰のリンクのローカル座標系への変換と、2. 隣接する2つのリンクのローカル座標系の間の変換および、3. ワールド座標系における腰の位置ベクトル、さらに4. 親リンクのローカル座標系における子リンクのローカル座標系の原点の位置ベクトルにより記述されている。

そして、人体モデルの姿勢を描画する際には、各リンクのローカル座標系のワールド座標系における位置と姿勢を計算し、その結果を元に人体モデルを構成している各リンクを描画している。

例えば、右上腕を描画する際には、以下のような計算が行われている。

まず、右上腕のローカル座標系 Σ_{rarm1} のワールド座標系 Σ_w における位置ベクトル \vec{v}_{w_rarm1} を次式により求める。

$$\vec{v}_{w_rarm1} = \vec{v}_{w_hip} + Q_{w_hip}^{-1}(\vec{v}_{hip_body}) + Q_{w_hip}^{-1}\{Q_{hip_body}^{-1}(\vec{v}_{body_rarm1})\} \quad (4.9)$$

次に、右上腕のローカル座標系 Σ_{rarm1} のワールド座標系 Σ_w における姿勢 Q_{w_rarm1} を次式により求める。

$$Q_{w_rarm1} = Q_{w_hip}Q_{hip_body}Q_{body_rarm1} \quad (4.10)$$

以上により、右上腕のローカル座標系 Σ_{rarm1} のワールド座標系 Σ_w における位置と姿勢が計算され、この計算結果を元にして右上腕のリンクが仮想空間内に描画される。

4.2 AHMSSの概要

本節ではまず、AHMSSの基本設計、システム構成およびハードウェア構成について説明する。そして、システムを構成する各要素における処理について説明した後、システム全体の処理の流れを述べる。

4.2.1 ソフトウェア構成

AHMSSで実現する機能と3.4節で述べた仕様を、もう一度システム設計の観点からまとめると、以下のようになる。

仕様1 バーチャル人間および仮想空間内のオブジェクトの動作を合成する

仕様2 入力、操作対象の名称、位置および操作の種類とする

仕様3 出力は、仮想空間内におけるバーチャル人間とオブジェクトの動作のアニメーションとし、ディスプレイに表示する

仕様4 動作合成部分と表示部分を分離して異なる計算機上に実装する

仕様5 バーチャル人間の動作情報を操作対象のオブジェクトに保持させる

仕様6 同種のオブジェクト間で、動作データと動作合成手法の共通部分を共有できるようにする

仕様7 仮想空間へオブジェクトを容易に追加できる

仕様8 バーチャル人間の動作を容易に追加できる

仕様9 複数動作の融合を行う

仕様10 同時に複数のバーチャル人間の動作を合成可能とする

これより、AHMSSのソフトウェア構成を、図4.7に示すものとする。なお、図中灰色で示すメインプロセス、人アルゴリズムプロセス、物アルゴリズムプロセス、仮想空間情報サーバ、画像表示プロセスはそれぞれ独立したプロセスである。図4.7に示したようにソフトウェアを構成する理由を次に述べる。

AHMSSでは、バーチャル人間の動作情報を操作対象のオブジェクトに保持させるために、動作情報はシステムの他の部分と独立に実装する必要がある。さらに、動作情報は、具体的には動作合成アルゴリズムと動作データから成るが、これらも互いに独立であることが望ましい。そして、それぞれのアルゴリズムは、独立したプロセスとして実装する。これは、アルゴリズムあるいは動作データのみを追加や変更を可能とし、また、これについては後に4.2.4項で具体的に説明するが、複数の動作の融合を可能するために必要である。同様の理由により、仮想空間内のオブジェクトの動作を合成するアルゴリズムも、それぞれ独立したプロセスとして実装する。

次に、人体モーション合成のリアルタイム性を確保するために、動作の合成結果を1コマ毎に出力する。これも、後に4.2.4項で具体的に説明するが、複数の動作を融合するためにも必要である。

また、計算負荷を分散させるために、動作合成などの計算を行う部分と、実際に仮想空間を描画し、ディスプレイ上に表示する部分をそれぞれ異なる計算機上に実装する。

そして、複数の動作を融合する機構として動作融合部を構築する。さらに、複数のバーチャル人間の動作を合成可能とするために、仮想空間の情報（バーチャル人間やオブジェクトの位置・姿勢）を一元管理する部分である仮想空間情報サーバを、複数のバーチャル人間の情報を保持できるように構築する。

以上、AHMSSの仕様とソフトウェア構成の関係を表4.1にまとめる。

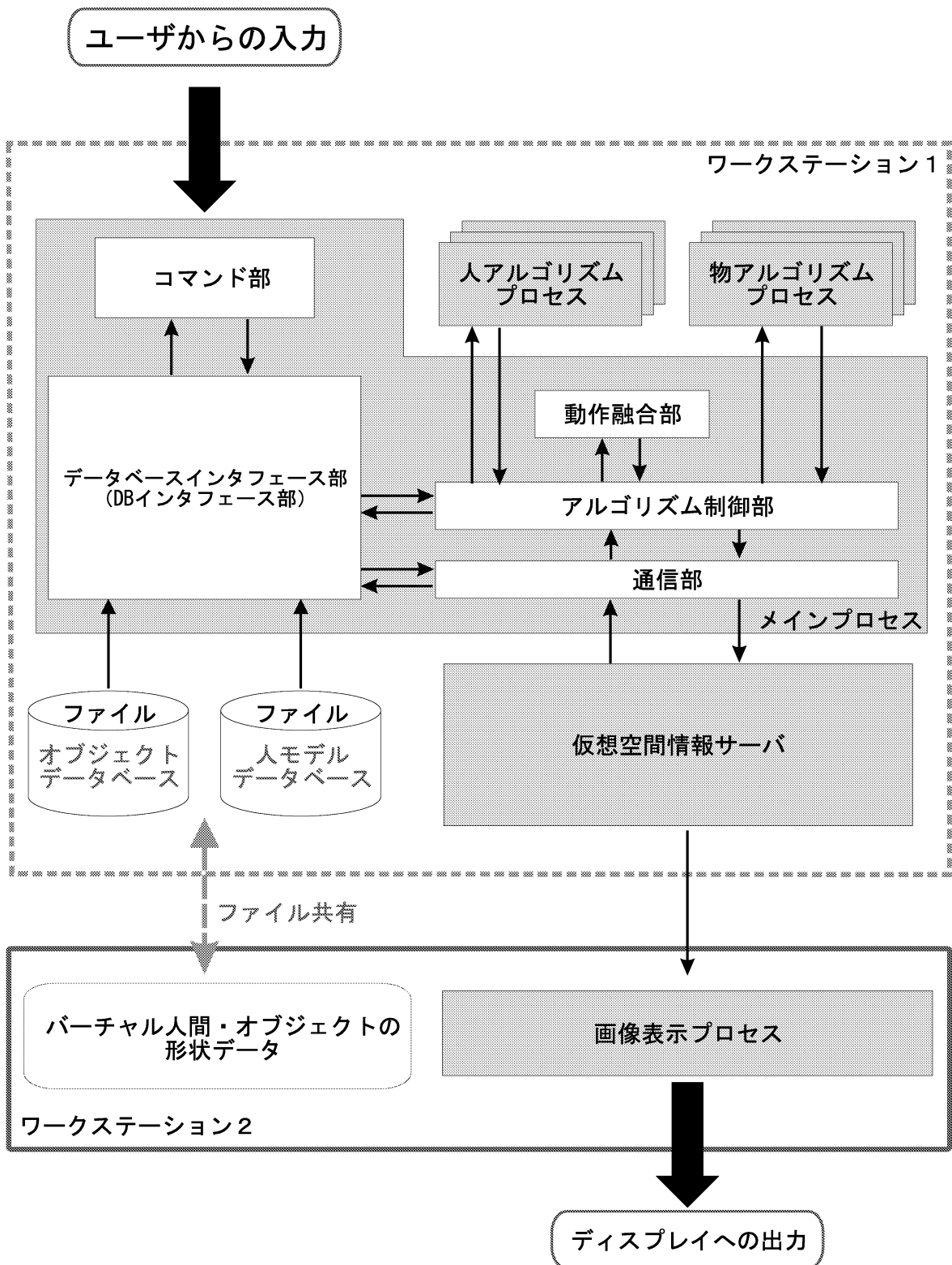


図 4.7: ソフトウェア構成

表 4.1: 仕様とソフトウェア構成の関係

仕様	ソフトウェア構成
1	バーチャル人間およびオブジェクトの動作データと動作合成アルゴリズムを用意する
2	メインプロセスでユーザからの入力を受け付ける
3	画像表示プロセスで実際に仮想空間を描画してディスプレイ上に出カする
4	動作合成部分と画像表示部分を異なる計算機上に構築する
5-8	動作情報を動作データと動作合成アルゴリズムに分割し、バーチャル人間の動作データはオブジェクトデータベースに格納し、動作合成アルゴリズムは人・物アルゴリズムプロセスとしてそれぞれ独立に構築する
9	動作融合部を構築する
10	仮想空間管理サーバを複数のバーチャル人間の情報を保持できるように構築する

次に、図 4.7 の構成要素の概要を述べる。なお、各構成要素の詳細な説明は 4.2.4 項で行う。

メインプロセスは、システム全体の制御を行う部分であり、ユーザからの入力を受け付けるコマンド部、データベースからの情報を読み込むデータベースインタフェース部（DB インタフェース部）、人・物アルゴリズムの制御を行うアルゴリズム制御部、複数動作の融合を行う動作融合部、仮想空間情報サーバとの通信を行う通信部から成る。

人アルゴリズムプロセスおよび物アルゴリズムプロセスはそれぞれ、バーチャル人間と仮想空間内のオブジェクトの動作を合成するプロセスであり、各々のアルゴリズムプロセスはそれぞれ別々のプロセスとして構築する。

オブジェクトデータベースには、オブジェクトの形状データや大きさ、質量等のデータに加え、そのオブジェクトに関するバーチャル人間の動作データが格納されている。人モデルデータベースには、バーチャル人間の形状データや各リンクの重さ、長さなどのデータが格納されている。

仮想空間情報サーバは、仮想空間の情報を一元管理する部分であり、複数のバーチャル人間の情報を保持できるように構築する。なお、複数のバーチャル人間の動作を合成する際には、複数のメインプロセスを 1 つの仮想空間情報サーバと接続する。

画像表示プロセスは、仮想空間を描画し、ディスプレイへと出力する部分である。なお、バーチャル人間や仮想空間内のオブジェクトの形状データは、2 つのワークステーション間でファイル共有している。

4.2.2 ハードウェア構成

次に、AHMSS のハードウェア構成について説明する。AHMSS では、4.2.1 項で述べたように、動作合成などの計算を行う部分と、実際に仮想空間を描画し、ディスプレイ上に表示する部分をそれぞれ異なる計算機上に実装して計算負荷を分散することによって、人体モーション合成のリアルタイム性を確保する。そのため、ハードウェア構成は、図 4.8 に示すものとする。すなわち、仮想空間の描画は、ワークステーション 2 で行い、それ以外の処理のすべてをワークステーション 1 で行う。なお、2 つのマシン間ではイーサネットによるソケット通信により、仮想空間の描画に必要なバーチャル人間の位置と姿勢、仮想空間内のオブジェクトの位置と姿勢のデータがそれぞれ通信される。

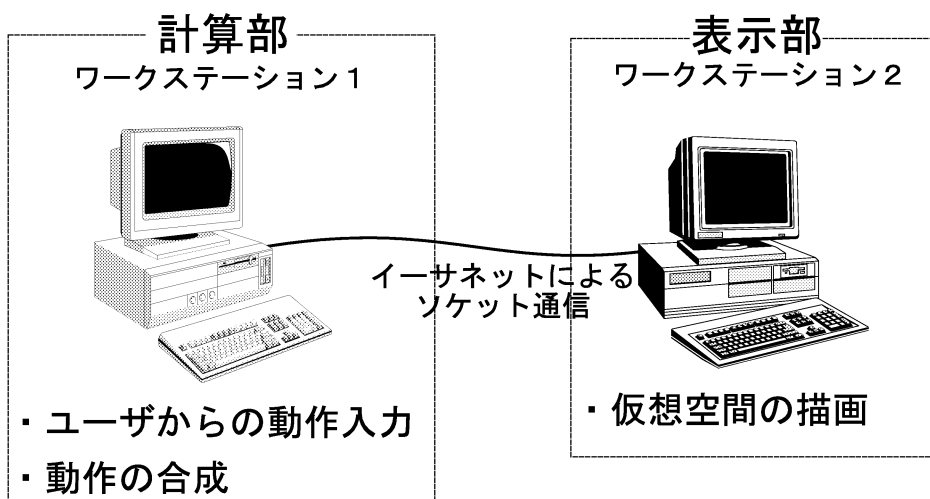


図 4.8: ハードウェア構成図

4.2.3 システム全体の処理の流れ

AHMSSが提供する機能は、1. バーチャル人間の動作の合成、と2. 仮想空間へのオブジェクトの追加の2つである。以下では、それぞれについて、処理の流れの概略を以下に説明する。

バーチャル人間の動作合成

バーチャル人間の動作を合成する際の処理の流れを図4.9に示す。

ユーザはまず、バーチャル人間に、仮想空間内のどのオブジェクトに対してどのような動作をさせるか指定する。このときに、動作の目標地点や動作時間（ユーザ指定コマ数）等も必要に応じて入力する。入力データの例を表4.2に示す。

次に、システムはユーザからの入力情報に基づき、バーチャル人間の動作データ、人・物アルゴリズム、形状ファイルなどをオブジェクトデータベースから読み込む。そして、人・物アルゴリズムプロセスを起動する。そして、システムは以下の処理をユー

表 4.2: 入力データ例

操作対象の名称	コップ
動作の目標地点	$x = 400, y = 250, z = 80$
動作の種類	手に取る
動作時間（ユーザ指定コマ数）	200

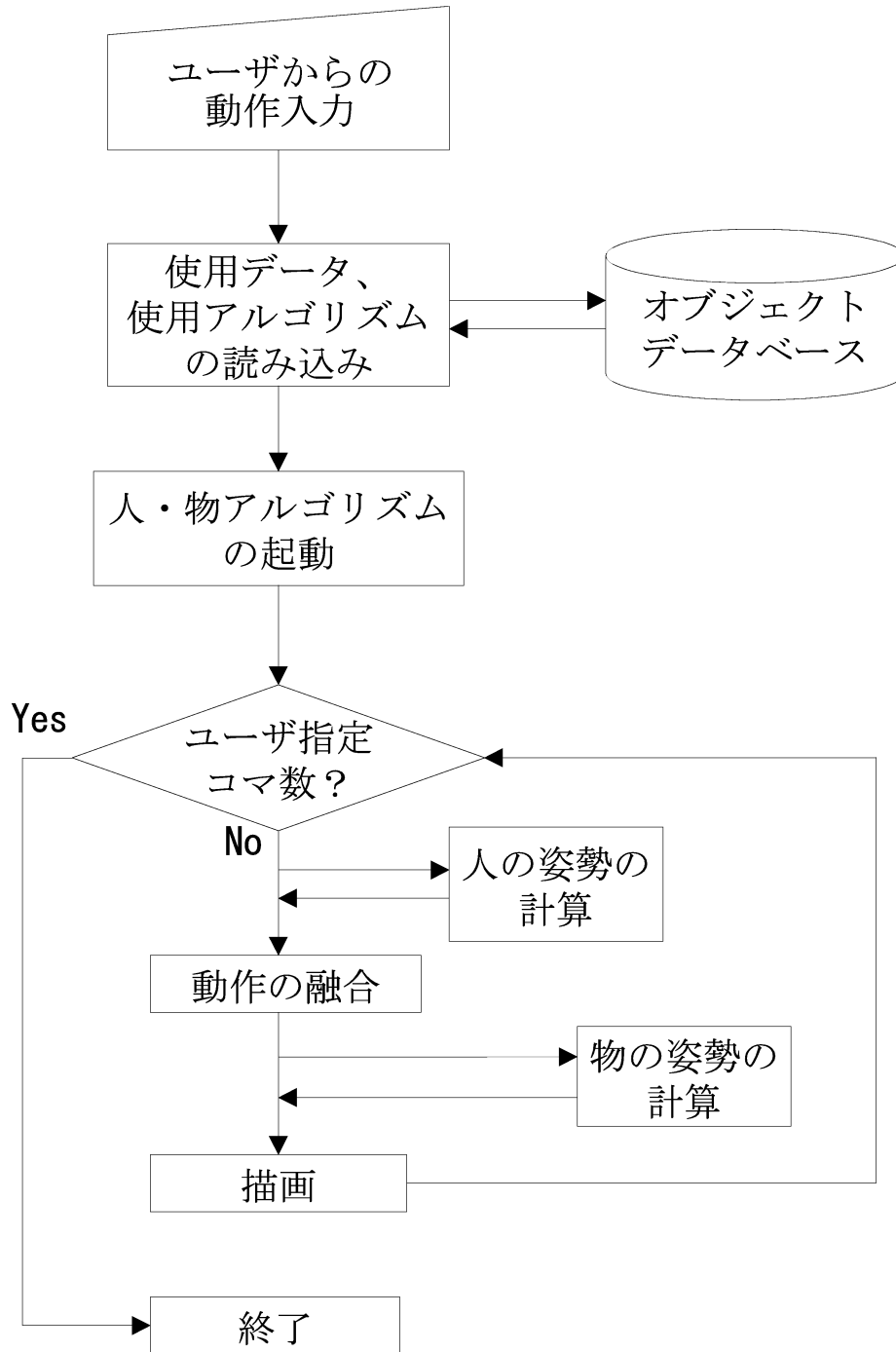


図 4.9: バーチャル人間の動作合成時の処理の流れ

ザ指定コマ数だけ繰り返す。

まず、人アルゴリズムプロセスで1コマ分のバーチャル人間の姿勢を計算する。複数動作を融合する場合には、複数の人アルゴリズムプロセスを並列に実行して、それぞれの動作におけるバーチャル人間の姿勢を計算し、それらを融合することによって、バーチャル人間の姿勢を決定する。次に、このバーチャル人間の姿勢を基に、物アルゴリズムプロセスでオブジェクトの姿勢を計算する。これで、1コマ分のバーチャル人間、オブジェクトの姿勢の計算が完了し、1コマ分の描画が行われる。この処理を毎秒10~20回の速さで、ユーザ指定コマ数だけ繰り返すことによって、バーチャル人間の動作がアニメーションとしてディスプレイ上に表示される。

仮想空間へのオブジェクト追加時

ユーザが仮想空間へオブジェクトを追加するときの処理の流れを図4.10に示す。

ユーザが仮想空間へ追加するオブジェクトの名称を入力すると、システムは、オブジェクトデータベースから追加するオブジェクトの形状データやオブジェクトの大きさ・質量等のデータを読み込む。そして、これら読み込んだデータを仮想空間情報サーバへと送ることによって仮想空間へオブジェクトを追加することができる。

4.2.4 システムを構成する各要素における処理

本項では、図4.7に示したシステムのそれぞれの構成要素について詳細な説明を行う。なお、4.2.3項で述べたように、AHMSSが提供する機能は、1. バーチャル人間の動作の合成、2. 仮想空間へのオブジェクトの追加の2つであり、以下ではそれぞれの機能に分けて説明する。

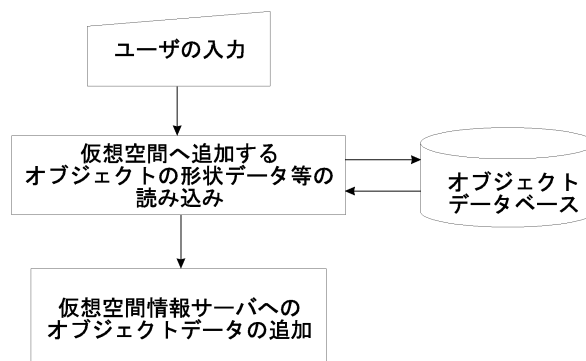


図 4.10: 仮想空間へのオブジェクト追加時の処理の流れ

表 4.3: ユーザからの入力情報

入力情報
操作対象の名称 (操作対象の位置)
(動作の目標地点)
(動作の目標姿勢)
操作の種類
(動作時間)
(融合開始タイミング)

バーチャル人間の動作の合成時の処理

以下では、バーチャル人間の動作合成時におけるシステムの各構成要素における処理およびデータベースに格納されているデータに関して説明する。

- ユーザからの入力

バーチャルコラボレータの知能部分であるヒューマンモデルから送られてくる情報は、操作対象の名称または操作対象の位置 (x, y, z 座標) および操作の種類 (ボタンを押す、レバーをスライドさせる等) であり、基本的にはこれらがユーザからの入力情報となる。この他にも動作の種類によっては、動作の目標地点や目標となるバーチャル人間の姿勢を指定する。また *AHMSS* では、複数のオブジェクトの組み合わせが持つアフォーダンスを表現するために、複数動作の融合を実現する。後に詳しく述べるが、*AHMSS* における複数動作の融合では、動作を融合するタイミングやそれぞれの動作の動作時間の調整は行っていない。そのため、動作時間や融合開始のタイミングはユーザが入力する必要がある、これらの値もユーザからの入力情報となる。ユーザからの入力情報を表 4.3 にまとめる。表中カッコ () で囲んだ項目は、動作の種類によっては入力する必要がある項目である。

- メインプロセス

図 4.11 にメインプロセスの構成図を、各構成要素間で入出力される情報と共に示す。図に示したようにメインプロセスは、コマンド部、DB インタフェース部、アルゴリズム制御部、通信部、動作融合部の 5 つの構成要素から成る。以下に、動作融合部以外の各構成要素について説明する。なお、動作融合部については後に説明する。

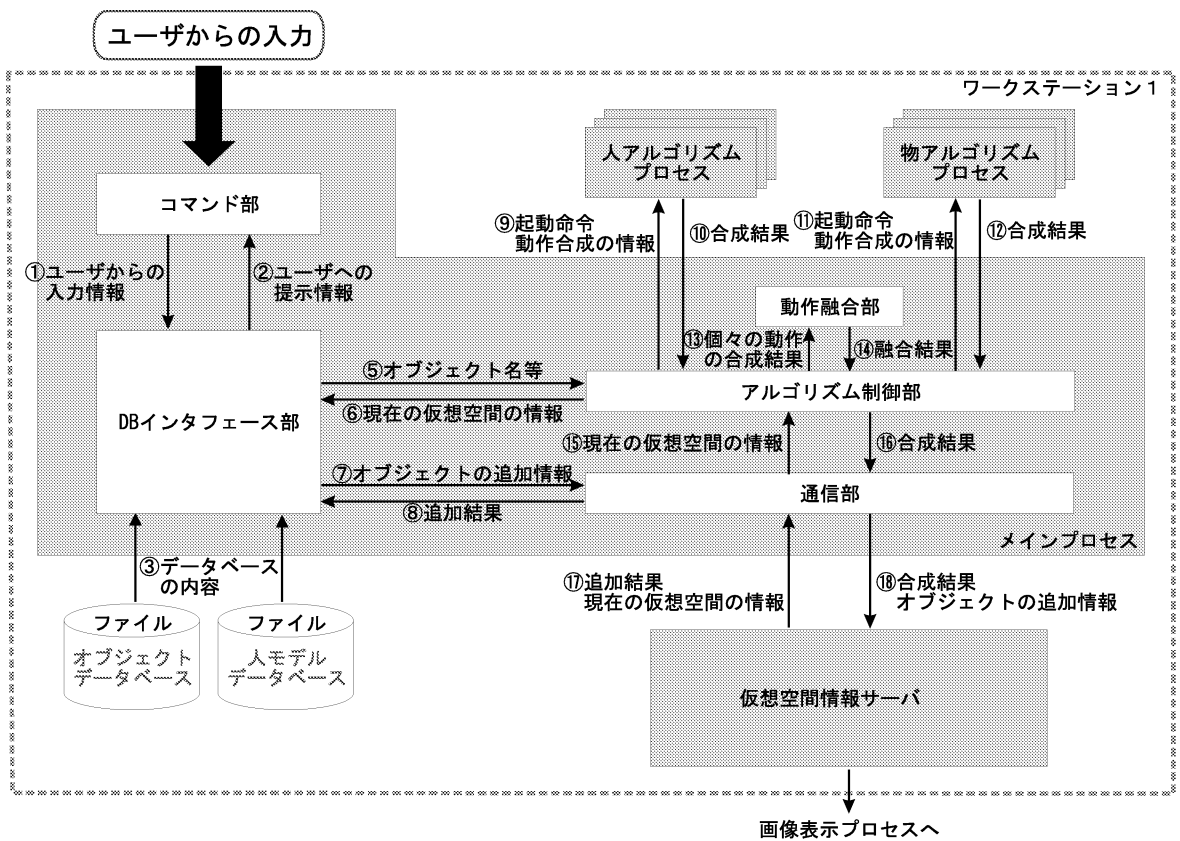


図 4.11: メインプロセスの詳細

– コマンド部

コマンド部は、ユーザとメインプロセスとの間のインタフェースの役割を担う部分であり、ユーザからの入力を受け付け、ユーザからの入力情報をDBインタフェース部へと送る。また、DBインタフェース部からは、バーチャル人間の動作を入力する際に必要となる情報である、仮想空間内に存在するオブジェクトの名称や各オブジェクトがアフォードする動作の名称等を受け取り、図 4.8 に示したワークステーション 1 のディスプレイ上へ提示する。

– DB インタフェース部

DBインタフェース部は、コマンド部から送られてくるユーザの入力情報に基づき、オブジェクトデータベースにアクセスし、データベースの内容を読み込む。動作合成時には、読み込んだデータベースの内容をユーザからの入力情報と共にアルゴリズム制御部へと送る。またアルゴリズム制御部からは、仮想空間内のオブジェクト名等のユーザへの提示情報が入力されるので、それをコマンド部へ出力する。

– アルゴリズム制御部

アルゴリズム制御部では、DBインタフェース部から送られてくる動作合成の情報に基づき、人アルゴリズムプロセスと物アルゴリズムプロセスの起動および制御を行う。以下でその処理について説明する。

アルゴリズム制御部ではまず、DBインタフェース部からの入力に従い、必要な人アルゴリズムプロセス、物アルゴリズムプロセスを起動する。また、通信部からも現在の仮想空間の情報（バーチャル人間・オブジェクトの位置・姿勢等）が入力される。次に、起動した人アルゴリズムプロセスに、DBインタフェース部および通信部から入力された動作合成の情報を出力する。そして、人アルゴリズムプロセスから 1 コマ分のバーチャル人間の姿勢の合成結果が入力される。複数の動作を融合する場合には、それぞれの人アルゴリズムプロセスからの計算結果を動作融合部へと出力する。そして、動作融合部での動作融合計算の後、融合結果が動作融合部から入力される。なお、複数の動作を融合しない場合にはこの処理は行われぬ。ここまでの処理により、1 コマ分のバーチャル人間の姿勢が決定される。

次に、決定されたバーチャル人間の姿勢およびオブジェクトの動作合成情報は、物アルゴリズムプロセスへ出力される。そして、物アルゴリズムプロ

セスから 1 コマ分のオブジェクトの姿勢の計算結果が入力される。

以上により、1 コマ分のバーチャル人間の姿勢とオブジェクトの姿勢が計算され、計算結果を通信部へと出力する。この計算処理をユーザ指定コマ数分行った後、アルゴリズム制御部は各アルゴリズムプロセスを停止させる。

– 通信部

通信部では、メインプロセスと仮想空間情報サーバとの間のデータ通信を行う。動作合成時には、通信部はアルゴリズム制御部に現在の仮想空間の情報（バーチャル人間・オブジェクトの位置・姿勢等）を出力する。また、入力されたバーチャル人間、オブジェクトの姿勢計算結果を仮想空間通信サーバへと出力する。

なお、このメインプロセスを複数実行し、1 つの仮想空間情報サーバへと接続することにより、複数のバーチャル人間の動作を合成することが可能である。

● 人アルゴリズムプロセス

人アルゴリズムプロセスは、アルゴリズム制御部によって起動され、アルゴリズム制御部から入力される動作合成情報を基に、バーチャル人間の動作を合成する。そして、アルゴリズム制御部から送られる制御信号にしたがって、合成結果をアルゴリズム制御部へと出力する。

それぞれのアルゴリズムプロセスは独立したプロセスであり、“モーションキャプチャシステムで取得されたデータを再生するアルゴリズム” や “線形補間アルゴリズム” 等の様々なアルゴリズムを独立プロセスとして用意しておく。個々の人アルゴリズムプロセスは、そのアルゴリズムにより、バーチャル人間の姿勢を 1 コマ毎に計算し、結果をアルゴリズム制御部に出力する。バーチャル人間の複数の動作を融合する際には、複数の人アルゴリズムプロセスにより個々の動作の姿勢を計算し、後に述べる動作融合部で融合した姿勢を計算する。

● 物アルゴリズムプロセス

物アルゴリズムプロセスも、人アルゴリズムプロセスと同様、アルゴリズム制御部によって起動される。アルゴリズム制御部から入力される動作合成情報を基に仮想空間内のオブジェクトの動きを計算する。そして、アルゴリズム制御部から送られる制御信号にしたがって、合成結果をアルゴリズム制御部へと出力する。

人アルゴリズムプロセスと同様に、物アルゴリズムプロセスもそれぞれ独立したプロセスであり、複数のオブジェクトの動きを同時に計算することが可能である。

- 動作融合部

次に、メインプロセスの一部である動作融合部について説明する。

複数動作の融合は、人アルゴリズムから出力される個々の動作の合成結果のデータを融合することによって実現する。具体的な動作融合手法を以下に述べる。

各動作の各関節には、その動作における各関節に動きの重要性に応じて予め重要度が設定されている。動作の種類によっては、動作の始めと終わりで各関節の動きの重要性が異なる場合もあるので、重要度は動作の1コマ毎に設定されている。

そして、動作を融合する際には、個々の動作の合成結果に重みを掛けて加重平均をとることによって、動作融合を行う。融合の計算式を(4.11)に示す。

$$q_j = \frac{\sum_{i=1}^n w_{ij} q_{ij}}{\sum_{i=1}^n w_{ij}} \quad (4.11)$$

ここで、 i は融合する動作の番号、 j はQuaternionの番号、 n は融合する動作数、 q_j は融合後の関節 j のQuaternion、 q_{ij} は i 番目の動作の関節 j におけるQuaternion、 w_{ij} は i 番目の動作の関節 j の重みとする。

なお、ユーザが指定した動作時間が、その動作に予め設定された動作時間と異なる場合には、各関節の重要度をユーザが指定した動作時間に合わせて変化させる。

- 仮想空間情報サーバ

仮想空間の状態を一元管理する部分であり、現在の仮想空間の情報(バーチャル人間・オブジェクトの位置・姿勢等)を1コマ分保持している。

動作合成時にはまず、現在の仮想空間の情報をメインプロセスの通信部へ出力する。そして、メインプロセスでの動作合成完了後、メインプロセスの通信部からバーチャル人間、オブジェクトの姿勢が入力され、そのデータを次のデータが入力されるまで保持する。また、画像表示プロセスからの要求があった場合には、画像表示プロセスへ現在保持しているバーチャル人間、オブジェクトの姿勢デー

タを出力する。なお、メインプロセス、画像表示プロセスとの入出力はソケット通信により行う。

また、同一仮想空間内で複数のバーチャル人間の動作を合成する場合には、複数のメインプロセスを同時に実行させることになるが、仮想空間情報サーバは、複数のメインプロセスから同時に入力が行われる等の場合にも正しく仮想空間の情報を更新するように構築されている。

- 画像表示プロセス

仮想空間情報サーバに保持されているデータを基に、仮想空間を描画する。画像表示プロセスは、1コマ分描画し終わる毎に仮想空間情報サーバの内容を読み込み、次の1コマを描画する。このため、描画の速度が仮想空間情報サーバの更新速度よりも遅い場合でも、常に最新の仮想空間の内容が描画され、リアルタイム性が保証される。

- オブジェクトデータベース

オブジェクトデータベースに格納されているオブジェクトデータの構造を図4.12に示す。図のように、オブジェクトデータには、オブジェクトの形状データファイル名、テクスチャファイル名、バーチャル人間の動作データファイル名、使用する人・物アルゴリズム名、物体情報ファイル名を保持する。動作データファイル名、人・物アルゴリズムプロセス名は、オブジェクトがアフォードする動作毎に用意する。また、物体情報ファイルは、オブジェクトの大きさや質量等のオブジェクト自体に関するデータを記録している。

- 人体モデルデータベース

人体モデルデータベースに格納されている人体モデルデータには、バーチャル人間の形状データファイル名、テクスチャファイル名、バーチャル人間の体の各部位の長さ・質量を保持から成る。

仮想空間へのオブジェクトの追加時の処理

以下では、仮想空間へのオブジェクトの追加時に関与するシステムの構成要素であり、動作合成時とその処理が異なるものに関して説明する。

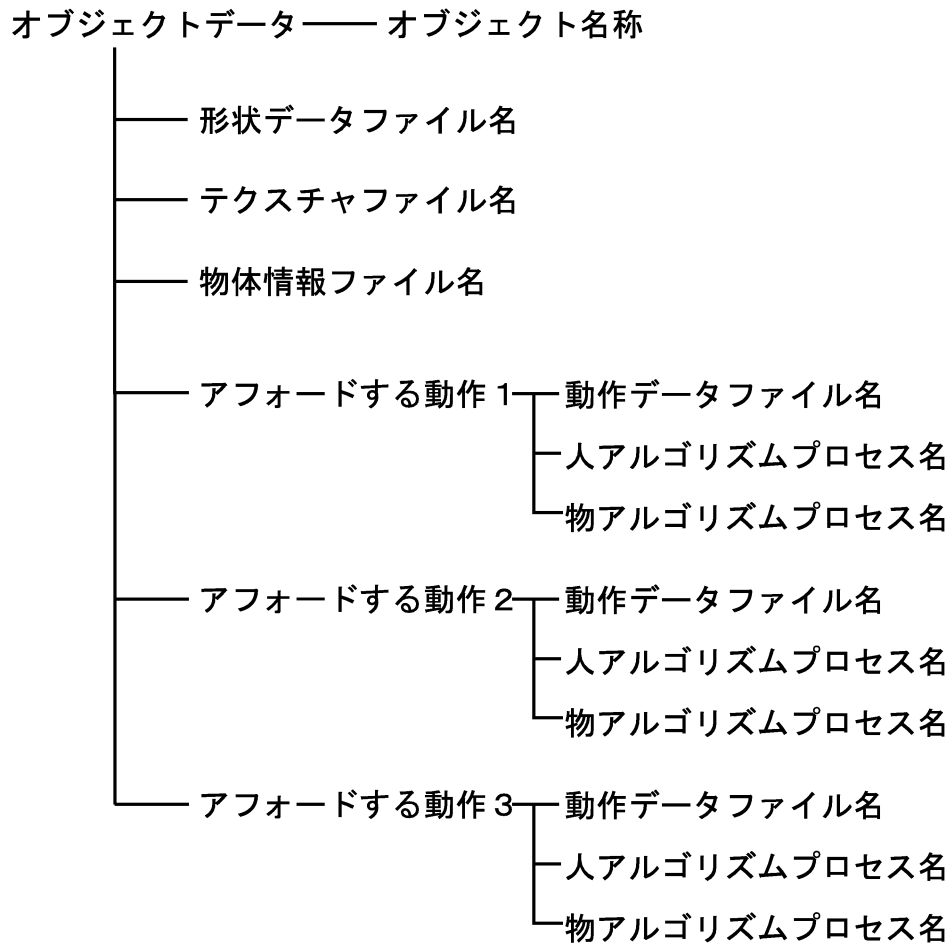


図 4.12: オブジェクトデータの構造

- ユーザからの入力

仮想空間へオブジェクトを追加する際には、ユーザは追加するオブジェクトの名称、仮想空間内での位置および姿勢を指定する。

- メインプロセス

仮想空間へのオブジェクトの追加時には、コマンド部、DB インタフェース部、通信部のみが関与するので、以下ではこの3つについて説明する。

- コマンド部

ユーザからの入力を受け付け、ユーザからの入力情報を DB インタフェース部へと送る。また、DB インタフェース部からは、仮想空間へオブジェクトを追加する際に必要となる情報を受け取り、図 4.8 に示したワークステーション 1 のディスプレイ上へと提示する。

- DB インタフェース部

DB インタフェース部は、コマンド部から送られてくるユーザの入力情報に基づき、オブジェクトデータベースにアクセスし、データベースの内容を読み込む。仮想空間へのオブジェクト追加時には、読み込んだデータベースの内容をユーザからの入力情報と共に通信部へと送る。また通信部からは、オブジェクト追加の成否の結果が入力され、コマンド部を介してユーザへと提示される。

- 通信部

通信部では、メインプロセスと仮想空間情報サーバとの間のデータ通信を行う。仮想空間へのオブジェクトの追加時には、DB インタフェース部から入力されるオブジェクトの追加情報を仮想空間情報サーバへと出力する。また、追加の成否の結果を仮想空間情報サーバから入力され、それを DB インタフェース部へと出力する。

- 仮想空間情報サーバ

仮想空間の状態を一元管理する部分であり、現在の仮想空間の情報（バーチャル人間・オブジェクトの位置・姿勢等）を保持している。仮想空間へのオブジェクト追加時には、メインプロセスからオブジェクトの追加に関する情報（追加するオブジェクトの名称、位置、姿勢等）が入力される。この入力に従い、仮想空

間情報サーバへのオブジェクトの登録が行われ、仮想空間にオブジェクトが追加される。また、通信部にはオブジェクト追加の成否が出力される。

第 5 章 *AHMSS* の使用例と評価及び考察

本章ではまず、本研究で開発した、アフォーダンスの概念を導入した人体モーション合成システムである *AHMSS* を用いてアニメーションを合成する際の手順について説明する。また、仮想空間に新たにオブジェクトを追加する際の手順およびバーチャル人間の新たな動作を追加する際の手順についても説明する。次に、*AHMSS* を用いて実際にアニメーションを合成した結果について説明する。そして、*AHMSS* の評価と考察を行い、今後の展望について述べる。

5.1 *AHMSS* の使用方法

本節では、*AHMSS* の使用方法について説明する。

5.1.1 アニメーションを合成する際の手順

本項では、*AHMSS* を用いてアニメーションを合成する際の手順を説明する。*AHMSS* では、対話形式のユーザインタフェースを構築しており、ユーザは以下に示すように簡単にアニメーションの合成を行うことができる。

1. 仮想空間内に存在するオブジェクトの一覧が提示される。
2. ユーザはバーチャル人間の操作対象とするオブジェクトの名称を入力する。
3. 入力したオブジェクトがアフォードする動作の一覧が提示される。
4. ユーザは、バーチャル人間に実行させる動作名を入力する。
5. 合成する動作の種類によっては、その他の入力事項（動作の目標地点、目標地点での方向等）が提示されるので、ユーザはそれらの項目について入力する。
6. 動作時間について入力要求があると、ユーザは合成する動作の動作時間をアニメーションのコマ数として入力する。
7. 動作を融合するか否かの入力要求があると、ユーザは、動作を融合する際には Yes を、しない場合には No を入力する。

8. Yes の場合は 1. に戻る。
9. No の場合は入力は完了し、入力した動作がアニメーションとしてディスプレイに表示される。

5.1.2 仮想空間に新たにオブジェクトを追加する際の手順

次に仮想空間に新たにオブジェクトを追加する際の手順について説明する。

オブジェクトデータが用意されている場合

まず、オブジェクトデータベースに追加するオブジェクトのデータが既に用意されている場合には、以下のように、アニメーションを合成する際と同様に、対話型インタフェースにより仮想空間へオブジェクトを追加する。

1. オブジェクトデータベース内のオブジェクトデータの一覧が提示される。
2. ユーザは、仮想空間に追加するオブジェクトの名称を入力する。
3. オブジェクトを追加する位置を入力するよう求められる。
4. ユーザは、仮想空間内のオブジェクトの座標 (x, y, z) を入力する。
5. オブジェクトの姿勢の入力要求があると、ユーザはオブジェクトの姿勢を入力する。

以上により、仮想空間へオブジェクトが追加される。

オブジェクトデータが用意されていない場合

次に、オブジェクトデータベースに追加するオブジェクトのデータが用意されていない場合について説明する。

まず、追加するオブジェクトの形状ファイルを用意する必要がある。AHMSSでは、形状データの形式として dxf 形式を採用している。dxf 形式とは、AutoDesk 社によって開発された CAD データのフォーマットのことであり、現在 CAD データの業界標準 (de-fact standard) となっている。そのため、このフォーマットを用いて作成されたオブジェクトは様々なアプリケーションソフトで読み込むことができる。

次に、オブジェクトデータファイルを作成する必要がある。オブジェクトデータファイルとは、4.2.4 項で説明したオブジェクトデータを記述したテキストファイルのこと

である。オブジェクトデータファイルには、オブジェクトの名称、形状ファイル名、テクスチャファイル名、オブジェクトの大きさや質量を記録した物体情報ファイル名、アフォードする動作名、動作データファイル名、人アルゴリズムプロセス名、物アルゴリズムプロセス名等を所定の形式に従って記述する。

そして、これら2つのファイルを所定のディレクトリへと移動することによって、このオブジェクトがオブジェクトデータベースへと追加されることとなる。後は前述の手順に従って、仮想空間へオブジェクトを追加する。

5.1.3 バーチャル人間の新たな動作を追加する際の手順

次に、バーチャル人間の新たな動作を追加する際の手順について説明する。

AHMSSでは、バーチャル人間の新たな動作を追加する際に、以下の3つの場合が考えられる。

1. 既存の動作合成アルゴリズムを用い、新たな動作データのみを追加する。
2. 既存の動作データを用い、新たな動作合成アルゴリズムのみを追加する。
3. 新たな動作合成アルゴリズム、動作データを追加する。

1.、2. の場合は、3. の場合の部分的な場合なので、ここでは3. の場合について説明する。

バーチャル人間の新たな動作合成アルゴリズムを追加する際には、動作合成アルゴリズムを実行形式のプロセスとして作成する必要がある。以下では、作成する動作合成アルゴリズムの入出力仕様について述べる。

動作合成アルゴリズムを実行する際には、入力としてシステムが保持する以下の情報が参照可能である。

- 仮想空間内のすべてのオブジェクトの現在の位置・姿勢
- 動作合成対象のオブジェクトの名称
- 動作の目標とする位置・姿勢
- 動作データのファイル名
- ユーザ指定コマ数

そして、これらの入力データを元に、バーチャル人間の位置・姿勢を計算し、出力する人アルゴリズムプロセスを作成し、追加する。

この際、動作合成アルゴリズムに対する制約は、以下の2点となる。

- リアルタイムに動作を合成できること
- 1コマ毎に結果を出力すること

従って、バーチャル人間の新たな動作を追加する際には、以上の要求を満たすアルゴリズムを作成し、その実行形式と動作データを所定のディレクトリへ追加すればよい。そして、仮想空間にオブジェクトを追加する際の手順で説明したオブジェクトデータファイルに新たに追加する動作データ名および動作合成アルゴリズム名を記述すれば、バーチャル人間の新たな動作の追加が完了する。

5.2 AHMSSの使用例

本節では、開発したAHMSSの機能を確認するために行った使用事例について述べる。

AHMSSの使用事例として、図5.1および図5.2に示すシナリオのアニメーションを作成する。図5.1は、バーチャル人間が合成する動作を時間順に示しており、図5.2は仮想空間内のオブジェクトの配置とバーチャル人間の移動経路を示している。

上記のアニメーションを実際に合成するための手順としては、以下のようになる。

1. 仮想空間情報サーバ、画像表示プロセス、メインプロセスを順に起動する。
2. メインプロセスを仮想空間情報サーバに接続する。
3. オブジェクトデータベースから必要なオブジェクトを選択して、仮想空間内に配置する。
4. 配置したオブジェクトに対する動作を入力する。

以上の手順の結果、バーチャル人間の人体モーションが仮想空間にアニメーションとして表示される。次節以降では、図5.1および図5.2に示したシナリオのアニメーションを合成するために使用する各種アルゴリズムについて説明する。

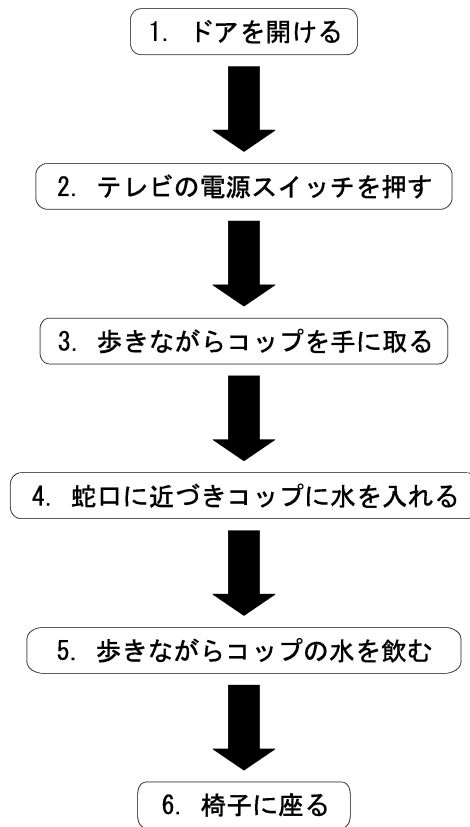


図 5.1: アニメーションシナリオ

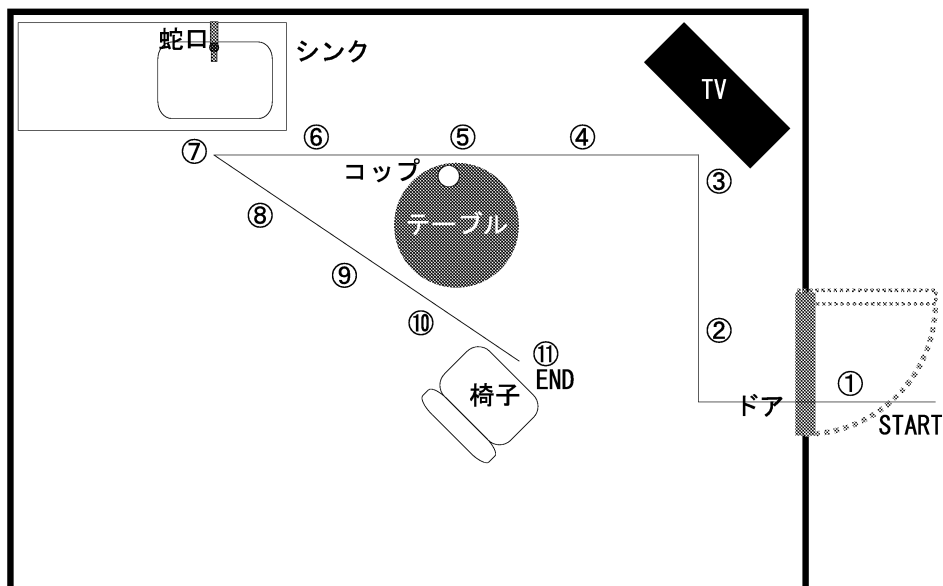


図 5.2: バーチャル人間の移動経路 (上から見た図)

時間	動作	人アルゴリズム	物アルゴリズム	図5.1における バーチャル人間の位置
	ドアの開閉	モーションキャプチャデータ 再生ALG	ドアの開閉を合成するALG	①
	歩行	歩行動作のALG		②
	テレビの電源 スイッチを押す	球状線形補間ALG		③
	歩行	歩行動作のALG		④
	歩きながらコップに 手を伸ばす	歩行動作のALG +球状線形補間ALG		⑤
	歩きながらコップを握る	歩行動作のALG +物を握る動作のALG	手に追従して移動するALG	⑤
	歩行	歩行動作のALG +腕の姿勢を維持するALG	手に追従して移動するALG	⑥
	蛇口をひねる	球状線形補間ALG	手に追従して移動するALG	⑦
	歩行	歩行動作のALG +腕の姿勢を維持するALG	手に追従して移動するALG	⑧
	歩きながら コップの水を飲む	歩行動作のALG +球状線形補間ALG	手に追従して移動するALG	⑨
	歩行	歩行動作のALG +腕の姿勢を維持するALG	手に追従して移動するALG	⑩
	椅子に座る	モーションキャプチャデータ 再生ALG +腕の姿勢を維持するALG	手に追従して移動するALG	⑪

図 5.3: 合成するアニメーションのシナリオとアルゴリズムの対応

5.3 動作の合成手法

本節では、5.2 節で説明したアニメーションを合成するために必要なアルゴリズムや動作データの作成方法について説明する。それぞれの動作に対して用いたアルゴリズムを表 5.1 および図 5.3 にまとめる。なお、図 5.3 では、アルゴリズムを略して ALG と表記した。また、図中のプラス（+）記号は動作の融合を表す。

これらのアルゴリズムの内容について、以下で説明する。

5.3.1 モーションキャプチャデータ再生アルゴリズム

モーションキャプチャデータ再生アルゴリズムは、3次元モーションキャプチャシステムを用いて取得されたデータを再生するアルゴリズムである。ここでは、まず3次元モーションキャプチャシステムについて説明し、取得したデータの補正方法について述べる。その後、モーションキャプチャデータ再生アルゴリズムの内容について説明する。

表 5.1: 動作とそのアルゴリズム

動作名	人アルゴリズム名	物アルゴリズム名
ドアの開閉	モーションキャプチャデータ再生アルゴリズム	ドアの開閉を合成するアルゴリズム
歩行動作	歩行動作のアルゴリズム	(なし)
対象に手を伸ばす	球状線形補間アルゴリズム	(なし)
コップを握る	物を握る動作のアルゴリズム	(なし)
コップの移動	腕の姿勢を維持するアルゴリズム	手に追従して移動するアルゴリズム
蛇口をひねる	球状線形補間アルゴリズム	(なし)
コップの水を飲む	球状線形補間アルゴリズム	(なし)
椅子に座る	モーションキャプチャデータ再生アルゴリズム	(なし)

3次元モーションキャプチャシステム

3次元モーションキャプチャシステムは、実際の人動作を計測し、数値化するシステムである。

動作を計測する際に用いる3次元モーションキャプチャシステムの構成を図5.4に示す。使用した3次元モーションキャプチャシステムは、POLHEMUS社のULTRATRAKである。ULTRATRAKは、磁界を利用した3次元位置、姿勢計測システムであり、磁界を発生するトランスミッタ、磁界を計測する15個の磁気センサと、専用の拡張ボードを備えたパソコンから構成されている。トランスミッタおよびパソコンの写真を図5.5、図5.6に示す。動作中の被計測者の姿勢は毎秒60コマ計測され、オイラー角で表現された各センサーのなす角度のデータが、リアルタイムにパソコン内のハードディスクに保存される。従って計測された姿勢の時間分解能は1/60秒となる。システムの基本性能を表5.2に示す。

次に、計測の方法について説明する。ULTRATRAKでは、磁界を利用して測定を行うため、金属等、磁界を乱すものに影響を受けやすい。そのため、計測は屋外の周囲に金属の無い場所で行い、被計測者には金属類の装着具をすべて外してもらった。また、磁気センサは、人の皮膚の上に固定する方式であるため、筋肉の動きの影響を受ける可能性がある。そこで、今回は、磁気センサを図5.7、図5.8に示すように、筋肉

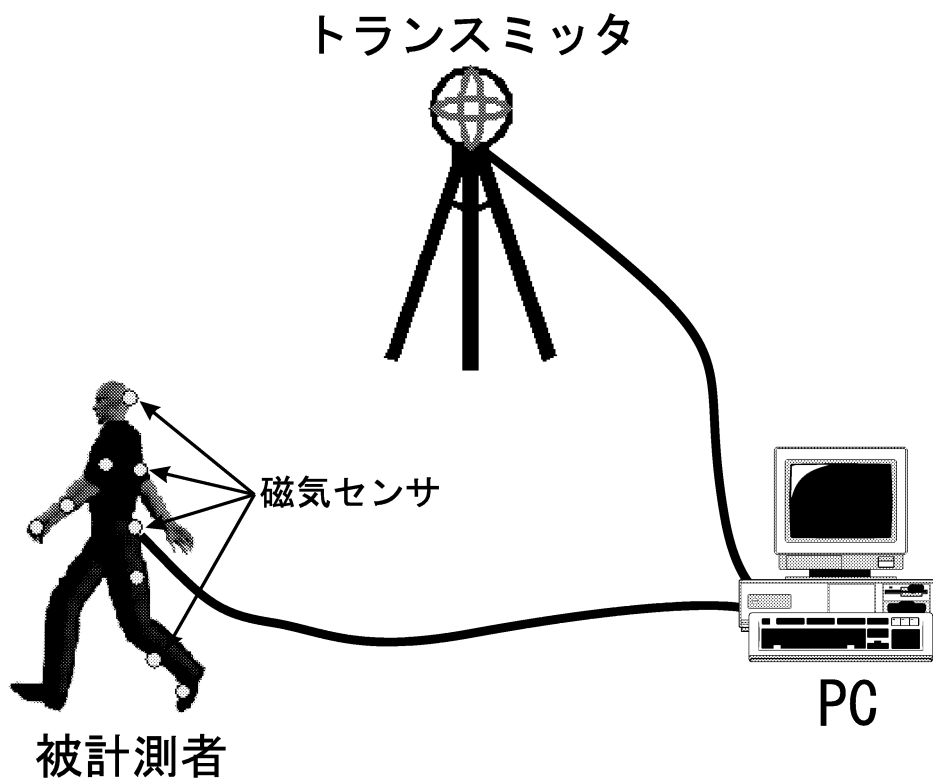


図 5.4: 動作計測時の機器配置

表 5.2: ULTRATRAK システムの基本性能

計測間隔	1/60 second	
測定可能範囲	最大	15 feet
	最小	2 feet
精度	角度	3 °RMS
	位置	1 inch RMS at 5 feet 3 inch RMS at 15 feet
分解能	角度	0.1 °RMS
	位置	0.05 inch RMS at 5 feet 0.25 inch RMS at 5 feet



図 5.5: トランスミッタ



図 5.6: 専用パソコン

がそれほど大きく動かないと思われる位置に固定した。なお、磁気センサは、図 4.1 に示した人体モデルの 15 個のリンクの動きを合成するために、人間の頭、胴、腰、左右上腕、左右下腕、左右掌、左右太股、左右脛、左右足の合計 15 箇所に取り付けた。計測時の様子を図 5.9 に示す。

なお、実験では、被計測者に、動作の最初か最後に直立の姿勢をしてもらった。直立の姿勢のデータは、次に説明する計測データの補正を行う際に用いるものである。また計測して得られるデータは、オイラー角で表現されたものであるため、Quaternion に変換する必要がある。

計測データの補正

ここでは、3次元モーションキャプチャシステムにより計測された動作データの補正を行う方法を述べる。人の体に磁気センサを取り付けて歩行動作を計測する場合、図 5.10 に示すように、皮膚上に取り付けた磁気センサの位置・方向と取り付けられたリンクの位置・方向は一致していない。従って、各部位の正確な位置・方向を知るためには、このずれの量を知る必要がある。このずれの量は、皮膚表面が変化しなければ一定であり、従って単一の一次変換で表すことができる。本研究では、このずれを補正する方法として、被計測者が直立姿勢時に計測したデータが図 5.11 に示すような直立姿勢を表すと仮定して補正を行う方法を用いた。

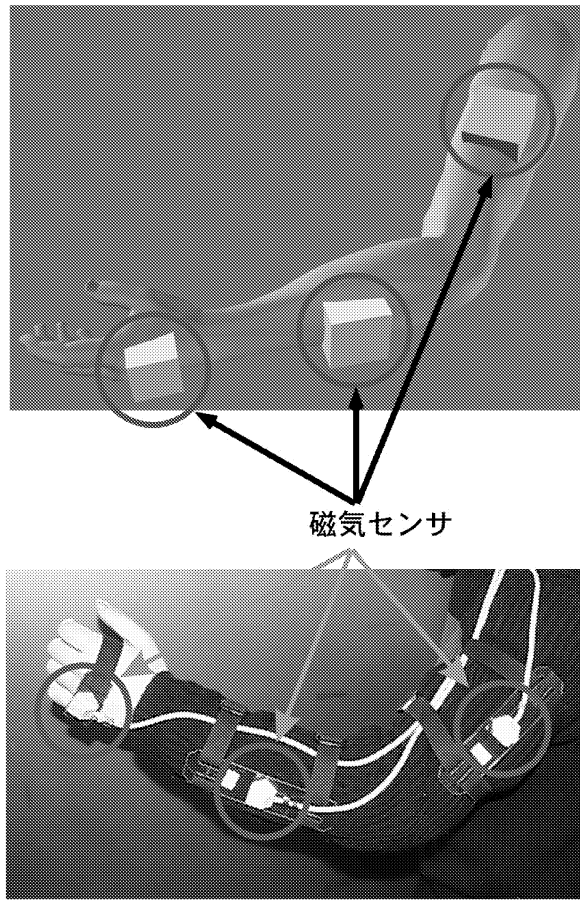


図 5.7: 磁気センサの取り付け位置 (腕)

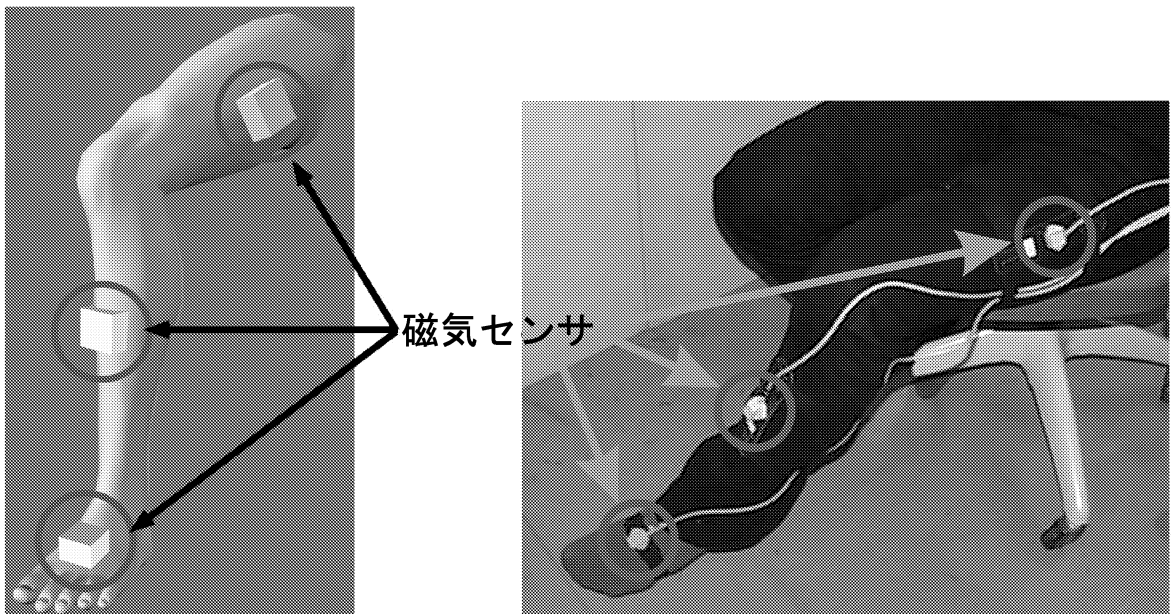


図 5.8: 磁気センサの取り付け位置 (足)



図 5.9: 計測の様子

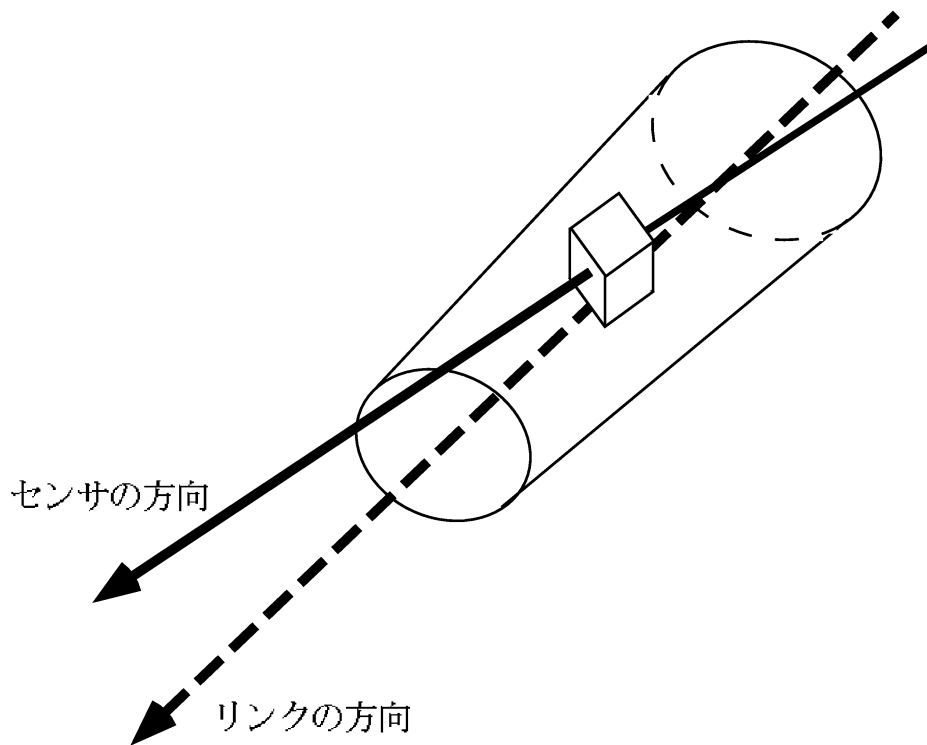


図 5.10: 磁気センサとリンクの方向のずれ

被計測者に取り付けた磁気センサ n から得られるデータをワールド座標系を基準とした Quaternion で表したものを ${}^W Q_{S_n}$ とし、ワールド座標系からリンク n のローカル座標系への変換を ${}^W Q_{L_n}$ とすると、磁気センサ n からリンク n のローカル座標系への変換 ${}^{S_n} Q_{L_n}$ は、次式で表すことができる。

$${}^{S_n} Q_{L_n} = ({}^W Q_{S_n})^{-1} {}^W Q_{L_n} \quad (5.1)$$

ここで、 ${}^W Q_{S_n}$ に被計測者が直立姿勢をとっているときの測定データを代入すると、このときの ${}^W Q_{L_n}$ は表 5.3 に示すような、直立を表す値が入るはずである。従って ${}^{S_n} Q_{L_n}$ が定数として求められる。ここで、リンクの向きとセンサの向きの関係が変化しないとすると、センサの座標系からリンクのローカル座標系への変換 ${}^{S_n} Q_{L_n}$ も変化しない。測定された任意のデータにおける、ワールド座標系からセンサの座標系への変換を ${}^W Q_{S_{n,d}}$ とすると、リンクのワールド座標系からの変換 ${}^W Q_{L_{n,d}}$ は、

$${}^W Q_{L_{n,d}} = {}^W Q_{S_{n,d}} {}^{S_n} Q_{L_n} \quad (5.2)$$

と、表すことができる。よって、(5.2) より、測定データを補正することができる。

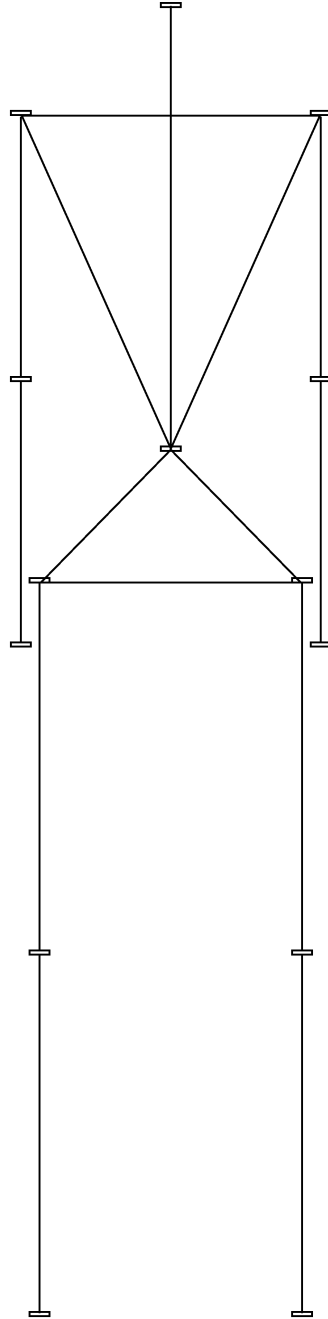


图 5.11: 直立姿势

表 5.3: 直立姿勢時のワールド座標系から各ローカル座標系への変換 Quaternion

${}^W Q_{L-n}$	$q = (w, x, y, z)$
${}^W Q_{\text{腰}}$	$(\sqrt{0.5}, 0, \sqrt{0.5}, 0)$
${}^W Q_{\text{胸}}$	$(\sqrt{0.5}, 0, \sqrt{0.5}, 0)$
${}^W Q_{\text{頭}}$	$(\sqrt{0.5}, 0, \sqrt{0.5}, 0)$
${}^W Q_{\text{右上腕}}$	$(0, \sqrt{0.5}, 0, \sqrt{0.5})$
${}^W Q_{\text{右下腕}}$	$(0, \sqrt{0.5}, 0, \sqrt{0.5})$
${}^W Q_{\text{右掌}}$	$(0, \sqrt{0.5}, 0, \sqrt{0.5})$
${}^W Q_{\text{左上腕}}$	$(0, -\sqrt{0.5}, 0, -\sqrt{0.5})$
${}^W Q_{\text{左下腕}}$	$(0, -\sqrt{0.5}, 0, -\sqrt{0.5})$
${}^W Q_{\text{左掌}}$	$(0, -\sqrt{0.5}, 0, -\sqrt{0.5})$
${}^W Q_{\text{右太股}}$	$(\sqrt{0.5}, 0, -\sqrt{0.5}, 0)$
${}^W Q_{\text{右脛}}$	$(\sqrt{0.5}, 0, -\sqrt{0.5}, 0)$
${}^W Q_{\text{右足}}$	$(1, 0, 0, 0)$
${}^W Q_{\text{左太股}}$	$(\sqrt{0.5}, 0, -\sqrt{0.5}, 0)$
${}^W Q_{\text{左脛}}$	$(\sqrt{0.5}, 0, -\sqrt{0.5}, 0)$
${}^W Q_{\text{左足}}$	$(1, 0, 0, 0)$

計測した動作の合成

次に、計測した動作データから、バーチャル人間の動作を合成する方法について説明する。モーションキャプチャシステムで取得したデータは、各リンクのなす角度であるため、仮想空間におけるバーチャル人間の位置を特定することができない。ただし、本研究では、図 4.3 に示したように、人体モデルに腰を頂点とする階層構造を設定している。そして、各リンクの長さは予め決めてあり、各リンク間の角度はモーションキャプチャシステムで取得したデータにより与えられるので、腰の位置を決定することによって、バーチャル人間の位置と姿勢が唯一に決まる。そこで、本研究では以下に述べる方法により、腰の位置を決定する。

手順 1 左右の足に対して、腰からの相対的な鉛直方向の位置を求め、位置が最も低い点を接地部分とする。

手順 2 図 5.12 に示すように、接地部分から腰の位置へのベクトル \vec{A} を計算する。

手順 3 \vec{A} にワールド座標系の原点から手順 1 で求めた接地部分までのベクトル \vec{B} を加えて、ワールド座標系における腰の位置を計算する。

手順 4 次の姿勢の接地部分を調べ、接地部分が手順 1 で求めた接地部分と同じ場合は、手順 2 に戻る。違う場合は、図 5.13 に示すように、手順 1 で求めた接地部分から新たに求めた接地部分へのベクトル \vec{C} を計算する。

手順 5 手順 4 で求めた \vec{C} を手順 3 で求めた \vec{B} に加え、新たな接地部分のワールド座標系の原点からの位置を求める。

以上により、バーチャル人間の姿勢が唯一に決まり、モーションキャプチャシステムで計測された動作が仮想空間内にアニメーションとして表示される。

5.3.2 歩行動作のアルゴリズム

本研究では、歩行動作のアルゴリズムとして、本研究室の先行研究で開発された安藤による手法^[14]を利用する。ここでは、アルゴリズムを簡単にするために、バーチャル人間の歩行を直線移動のみに限定している。アルゴリズムの詳細は付録 A に譲り、ここでは概要を説明する。

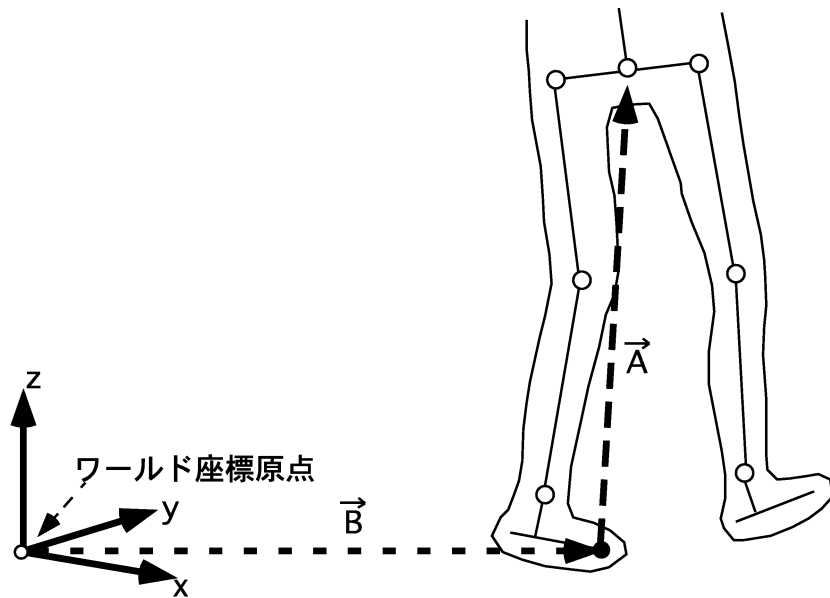


図 5.12: 腰位置の算出方法 1

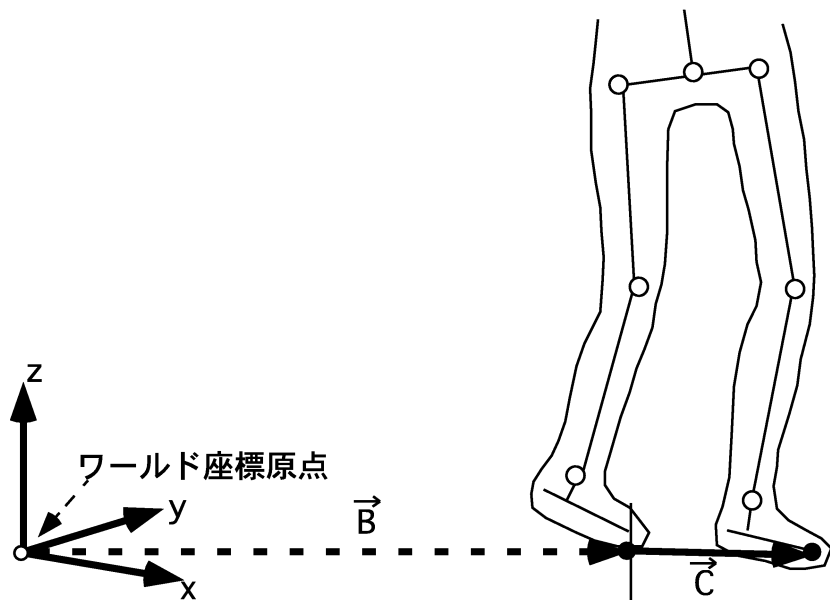


図 5.13: 腰位置の算出方法 2

歩行動作について

人の歩行動作を考えた場合、その歩数が1歩もしくは2歩の場合は、同じ動作が繰り返されることはないが、その歩数が3歩以上の場合は、途中、直進の動作が繰り返されると捉えることができる。そこで、バーチャル人間が2つの地点間を移動する時の動作を、距離が短い場合（歩数が1歩もしくは2歩の場合）と長い場合（歩数が3歩以上の場合）に分けて考える。

- 移動距離が短い場合（1歩もしくは2歩の場合）

移動距離が短い場合の歩行動作としては、1歩で目的の地点に辿り着く場合と2歩で目的の地点に辿り着く場合がある。そして、それぞれについて、移動方向と、目的の地点で向いている方向に応じて個々に動作データを作成しなければならない。

- 移動距離が長い場合（歩数が3歩以上の場合）

人が3歩以上で移動する場合、その一連の動作は「歩き始め」と「直進」、そして「歩き終わり」の3つの動作から構成されると捉えることができる。

- 歩き始め

「歩き始め」とは、直立の姿勢から移動方向に向かって足を踏み出しつつ体を回転させ、2歩歩いた後に、体の向いている方向が移動方向と一致し、体の姿勢が直進の動作の始めの姿勢と一致するまでの動作とする。「歩き始め」の動作データとしては、全方向に歩き始める動作データが必要である。

- 直進

人が直進する動作は、片足を踏み出す動作を左右逆にしながら交互に繰り返す動作と捉えることができる。「直進」の動作データとしては、この交互に繰り返される一連の動作の内、後方の足だけが地面と接触している状態から、前方の足が着地し、体が前進するにつれて足の前後が入れ替わり、そして、最初に後方にあった足が前に振り出されるまでの動作の動作データが必要となる。

- 歩き終わり

人は歩き終わりの段階において、歩幅や最終的に体を向ける方向を調整していると考えられる。また、歩き終わりにおいては、その移動距離に応じて、

右足で目的の地点に着地し、左足を添える場合と、左足で目的の地点に着地し、右足を添える場合の2つの場合があると考えられる。従って、バーチャル人間の歩き終わりの歩行動作を合成する場合には、歩き終わりの方向に応じた0度から180度までの動作データを、右足を最後に添える動作と、左足を最後に添える動作の両方の動作に対して作成する必要がある。

動作データベースの作成

以下に、上で述べた歩行動作を合成するために必要な動作データについてまとめる。

- 直進の動作データの作成

5.3.1項で説明した3次元モーションキャプチャシステムにより歩行動作を計測し、補正した歩行動作のデータから、一步分のデータを切り出す。歩行動作を合成する場合、直進の動作データを左右逆にしながら連続して表示するが、通常、切り出したデータは、最初の姿勢と最後の姿勢が完全には一致しないので、一致するようにデータを修正する。ここでは、最初の姿勢を基準にして、最後の姿勢を最初の姿勢の左右を反転させたものに一致させる。

- 歩き始めの動作データの作成

直進の動作データを作成した場合と同様に、補正した計測データから、2歩目までのデータを切り出す。歩き始めの動作データの最初の姿勢は直立でなければならない。また、歩き始めの動作データの最後の姿勢は、直進の動作データの最初の姿勢を、移動方向分だけ回転させた姿勢と一致していなければならない。よって、歩き始めの動作データの最後の姿勢を、直進の動作データの最初の姿勢に一致するように補正した計測データを修正する。

- 歩き終わりの動作データの作成

歩き終わりの動作データは、歩き始めの動作データとは逆に、最後の姿勢が直立であり、最初の姿勢が直進の動作の最後の姿勢に一致してはいなくてはならない。よって、歩き始めの動作データと同様に、補正した計測データから最後の直立の姿勢から2歩さがった姿勢までを切り出し、修正を行う。

- 1歩または2歩で到達する動作データの作成

表 5.4: 作成した移動用の動作データの一覧

動作の分類	各動作データの種類	データ数
1歩で到着	<ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ ・目的地点で向いている方向について0度から180度まで30度ステップ ・歩幅について普通の歩幅と小さい歩幅の2つ 	合計 98
2歩で到着	<ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ ・目的地点で向いている方向について0度から180度まで30度ステップ ・歩幅について普通の歩幅と小さい歩幅の2つ 	合計 98
歩き始め	<ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ 	合計 7
直進	<ul style="list-style-type: none"> ・1歩歩くデータ 	合計 1
歩き終り	<ul style="list-style-type: none"> ・目的地点で向いている方向について0度から180度まで30度ステップ ・最後に踏み出した足の左右 ・歩幅について普通の歩幅と小さい歩幅の2つ 	合計 28

1歩または2歩で到達する動作データは、最初の姿勢と最後の姿勢が共に直立でなくてはならない。これらの動作データもまた、補正した計測データを修正することによって作成する。

以上述べた方法で作成した移動用の動作データの一覧を表 5.4 に示す。移動方向と目的地点で向いている方向については、共に動作データは30度毎に用意する。さらに変形の度合いを小さくするために、1歩と2歩で到達する動作データと歩き終わりの動作データについては、歩幅が小さいものと普通のものの2種類の動作データを用意した。

移動動作の合成

次に、作成した動作データを用いて移動の動作を合成する方法について述べる。移動の動作を合成する場合、まず、移動の方向と距離に応じて適切な動作データを選択する。その後、選択した動作データの変形を行い、動作を合成する。

- 方向と距離に応じた動作データの選択

移動距離が長い場合は、移動動作を1つの動作ではなく、歩き始め、直進、歩き終わりといった複数の動作を組み合わせることにより合成する。また、予め準備する動作のデータベースに関しても、その方向や移動距離によって複数の種類の動作を用意している。また、移動距離が短い場合でも、その移動距離や到達地点で人体モデルが向いている方向によって複数の動作データを用意している。

移動動作を合成するためには、まず、移動方向と距離に応じて適切な動作データを選択する必要がある。

最初に、移動元と移動先の2地点間の距離に応じて、歩行動作を以下の4つのパターンに分類する。

1Step 1歩で移動先にたどり着く。

2Step 2歩で移動先にたどり着く。

2Parts 歩き始めと歩き終わりの動作データを直接連結した範囲で移動先にたどり着く。

3Parts 歩き始め、歩き終わりの間に直進のデータを含まなければ移動先にたどり着けない。

次に、分類されたパターンにおいて、変形率が最も小さくなるように、適切な動作データを組み合わせる。これにより、適切な動作データの組み合わせと動作データの変形率が決定される。なお、歩行動作の具体的な分類方法については、付録Aにて詳細に説明する。

- 動作データの変形

次に、移動動作をその方向や距離に応じて自由に合成するためには、移動方向の調整や移動距離の縮小といった動作データの変形が必要である。上記の手順によって決定された動作データの組み合わせを、変形率に基づいて変形する。以下ではその手法を簡単に説明する。なお、これについての詳細な説明は付録Aに譲る。

- 上半身の姿勢決定

まず、人体モデルのすべての部分について、注目している部分のローカル座標系に姿勢変化を与えた時の座標系 (\sum'_{Part}) について考える。この座標系

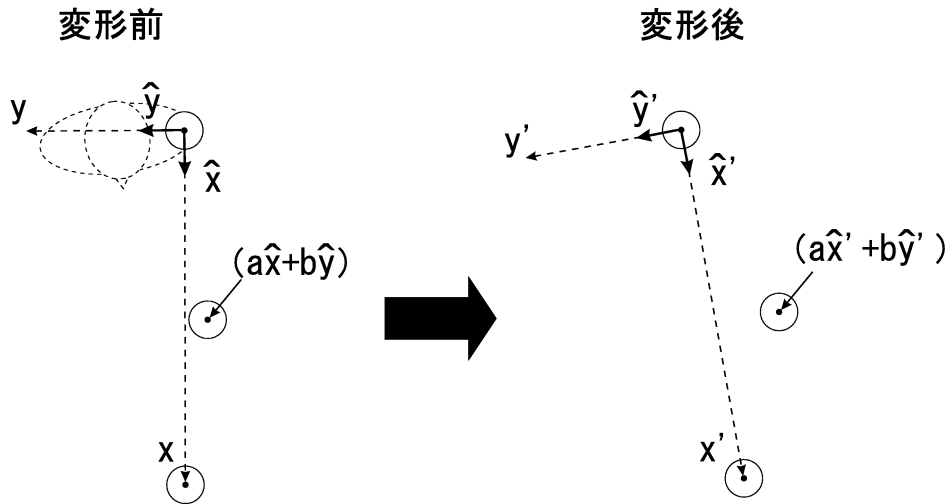


図 5.14: 座標変換の方法

は図 4.3 における階層構造において注目している部分の 1 つ下の部分ローカル座標系である。また、ここでは特に断らない限りワールド座標系の $x - y$ 平面内での位置のみを考え、 Σ'_{Part} の各フレームにおける原点の位置が高さ方向については同じであると仮定する。

変形前において、最初のフレームから最後のフレームの Σ'_{Part} の原点への直線を X 軸と仮定し、 X 軸上の単位ベクトルを \hat{X} とする。ワールド座標系の $x - y$ 平面内の Y 軸と \hat{Y} は X 軸と \hat{X} を 90° 回転させて求める。これにより、任意のフレームの Σ'_{Part} の原点を \hat{X} 及び \hat{Y} の線形結合 $a\hat{X} + b\hat{Y}$ として表現できる。変形後の最初のフレームと最後のフレームのフレームが既知であるから、同じ線形結合を施すことによって、図 5.14 に示すように任意のフレームの原点を求めることができる。

同様に、 Σ'_{Part} の原点を基準とする x と z 方向の単位ベクトルの先端の $x - y$ 平面における座標、即ち $x - y$ 平面への正射影も求めることができる。

これらの正射影から x 、 y 方向の単位ベクトルを求め、さらに y 方向の単位ベクトルを求めると、ワールド座標系からみた変形後の任意のフレーム Σ'_{Part} の原点、及び x 、 y 、 z 方向の単位ベクトルを求めることができる。

– 下肢の姿勢の決定

下肢の姿勢は接地している部分が滑ってはいけないという制約があるため、上半身と別の方法で姿勢を決めなければならない。両太股を Leg0、脛を Leg1、足を Leg2 として、まず Leg0 の姿勢を決め、次に地面と接触する Leg2、最後

に Leg1 の姿勢を決定する。

はじめに、上半身の姿勢を決める際と同じ方法を用いて座標変換した太股の付け根と膝の座標から Leg0 の姿勢を決める。

続いて Leg2 について、基本データの腰の Σ'_{Hip} の z 軸のワールド座標系の xy 平面への正射影と、変形後のデータの腰の Σ'_{Hip} の z 軸のワールド座標系の xy 平面への正射影のなす角を求め、これを θ として、基本データの Leg2 のワールド座標系から見た姿勢に対し、ワールド座標系の z 軸回りに θ だけ回転させ、これを変形後の Leg2 の姿勢とする。この時、Leg2 が地面と接触した場合、接地点 (踵かつま先) を支点として固定して姿勢変化を与え、その時の Leg2 の原点を Σ'_{Part} の原点とする。

しかし、Leg2 が地面から離れる瞬間や接地したまま最終フレームに至った場合、上記の方法によると、地面との接地点が変化し、求めた位置が接地点と一致しないため、足が滑ったような誤差が生じる。そこで、Leg2 が地面から離れた瞬間から次に接触するまでのフレームを求めて、その間でその誤差を徐々に吸収するように各フレームでの Leg2 の原点を求める。

最後に、求めた Leg0 と Leg2 の位置及び姿勢から Leg1 の姿勢を決定する。

5.3.3 球状線形補間アルゴリズム

球状線形補間アルゴリズム^{[17][18]}は、2つの回転変換を表す Quaternion の補間を行うアルゴリズムである。

4.1.2 項で述べたように、Quaternion q は、 $|q|^2 = w^2 + x^2 + y^2 + z^2 = 1$ の時、ベクトル $\vec{v} = (x, y, z)$ を回転軸とした $2 \cos^{-1} w$ の回転変換を意味する。そして、このような回転変換を表す Quaternion を線形補間する場合、その結果も回転変換を表していなければならない。しかし、Quaternion の各々の要素毎に線形補間を行っても、その結果の Quaternion の大きさは 1 ではなく、回転変換を表すものではない。また、この Quaternion を大きさが 1 になるように単位化したとしても、それは元の 2 つの Quaternion が表す回転変換を線形補間したものとはならない。

そこで、Quaternion の線形補間は、以下のように行う。まず、回転変換を表す Quaternion を 4 次元ベクトルとすると、すべての回転を表す Quaternion は、4 次元空間における単位超球上の点で表される。そして、Quaternion 同士の線形補間は、この球の表面上の弧に沿った経路で行われるとする。 q_0 、 q_1 を、この球の中心を原点とし、球表

面に向かう異なる単位ベクトルとすると、求める補間は次のように表すことができる。

$$q(t) = c_0(t)q_0 + c_1(t)q_1, \quad 0 \leq t \leq 1, \quad (5.3)$$

ここで、 $c_0(t)$ と $c_1(t)$ は実数値関数であり、 $q(t)$ は単位 Quaternion である。さらに、 $c_0(0) = 1$ 、 $c_0(1) = 0$ 、 $c_1(0) = 0$ 、 $c_1(1) = 1$ とする。これより、以下の式が導かれる。

$$\begin{aligned} 1 = q(t) \cdot q(t) &= c_0(t)^2 + 2c_0(t)c_1(t)(q_0 \cdot q_1) + c_1(t)^2 \\ &= \begin{bmatrix} c_0(t) & c_1(t) \end{bmatrix} \begin{bmatrix} 1 & d \\ d & 1 \end{bmatrix} \begin{bmatrix} c_0(t) \\ c_1(t) \end{bmatrix} = c^t M c \end{aligned} \quad (5.4)$$

ここで、 $d = q_0 \cdot q_1 = \cos \theta$ であり、 θ は 2 つの単位 Quaternion のなす角度である。 $|d| < 1$ であるので、行列 M は正定値となり、行列 M は、以下のように分解できる。

$$M = R^t D R, \quad D = \begin{bmatrix} 1+d & 0 \\ 0 & 1-d \end{bmatrix}, \quad R = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$u = \sqrt{D} R c$ とすると、(5.4) は、 $u^t u = 1$ と表すことができる。ゆえに、 u は単位ベクトルであり、 $u = (\cos \omega t, \sin \omega t)$ と表すことができる。 $c_0(t)$ について解くと、以下のようなになる。

$$c_0(t) = \frac{\cos \omega t}{\sqrt{2(1+d)}} - \frac{\sin \omega t}{\sqrt{2(1-d)}} = \frac{\sin(\omega t + \psi_0)}{\sqrt{1-d^2}} = \frac{\sin(\omega t + \psi_0)}{\sin \theta} \quad (5.5)$$

ただし、 ψ_0 は位相差である。同様に、 ψ_1 を位相差として、 c_1 について解くと以下のようなになる。

$$c_1 = \frac{\sin(\omega t + \psi_1)}{\sin \theta} \quad (5.6)$$

c_0 の境界条件を (5.5) に代入すると、

$$1 = \frac{\sin \psi_0}{\sin \theta}, \quad 0 = \frac{\sin(\omega + \psi_0)}{\sin \theta}$$

となり、これより、 $\psi_0 = \theta$ 、 $\omega = -\theta$ となる。よって、

$$c_0(t) = \frac{\sin((1-t)\theta)}{\sin \theta}$$

となる。同様に、 c_1 の境界条件を (5.6) に代入すると、

$$0 = \frac{\sin \psi_1}{\sin \theta}, \quad 1 = \frac{\sin(\omega + \psi_1)}{\sin \theta}$$

となり、これより、 $\psi_1 = 0$ 、 $\omega = \theta$ となる。よって、

$$c_1(t) = \frac{\sin(t\theta)}{\sin \theta}$$

となる。

以上より、球状線形補間は以下のように表される。

$$\begin{aligned} Slerp(q_0, q_1; t) &= \frac{q_0 \sin((1-t)\theta) + q_1 \sin(t\theta)}{\sin \theta} \\ \theta &= \cos^{-1}(q_0 \cdot q_1) \end{aligned} \quad (5.7)$$

5.3.4 その他のアルゴリズム

本項では、これまでに説明したもの以外のアルゴリズムについて説明する。また、ここで挙げたアルゴリズム以外に、手先の位置から腕の各関節の角度を求める逆運動学アルゴリズムもあり、これに関しては付録 B で説明する。

物を握る動作のアルゴリズム

物を握る動作のアルゴリズムは、物を握る際の指の動きを合成するためのアルゴリズムである。

ここでは握る対象として、コップのような円筒形の物とする。物を握る動作において、対象がコップのような円筒形の物である場合には、親指とそれ以外の指で対象物を包み込む形に指を曲げる。この場合、親指は他の 4 本の指とは異なった動きをする。ゆえに本アルゴリズムでは、物を握る際の指の動作を、親指の動きとそれ以外の指の動きに分けて合成する。具体的なアルゴリズムを以下に示す。

1. 握る対象である円筒形オブジェクトの直径を設定する
2. 設定した値から、目標とする指の曲げ角度を計算する
3. ユーザ指定コマ数で握る動作が完了するように、現在の指の曲げ角度と目標とする指の曲げ角度を球状線形補間する

球状線形補間は、 q_0 として現在の指の姿勢を、 q_1 として目標の指の姿勢を代入し、 t の初期値として、ユーザ指定コマ数の逆数を与える。そして、コマ毎に t の値をユーザ指定コマ数の逆数分だけ増やしていくことによって、物を握る動作を合成する。

腕の姿勢を維持するアルゴリズム

腕の姿勢を維持するアルゴリズムは、腕の各関節の角度を一定に保つことにより、腕の姿勢を一定に保つためのアルゴリズムである。このアルゴリズムは、バーチャル人間がコップを手を持って移動させる際の、コップを持った手の姿勢を合成するためのものである。

アルゴリズムとしては非常に単純であり、肩、肘、手首の関節の角度を一定に保つだけである。

ドアの開閉を合成するアルゴリズム

これも物アルゴリズムの1つであり、バーチャル人間がドアを開閉する動作に伴って、ドアを回転させるアルゴリズムである。アルゴリズムとしては、以下のようになる。

1. バーチャル人間の手とドアノブの距離を計算する
2. 距離が閾値以下である場合は、バーチャル人間がドアノブをつかんでいるものとして、手の動作に追隨してドアを回転させる
3. 距離が閾値以上である場合は、現在の姿勢を保持する

ドアは軸を中心とした回転運動のみ可能であり、手がドアノブの可動範囲である円弧から離れた場合には、ドアは手に追隨するのを止め、その場で停止する。

手に追隨して移動するアルゴリズム

手に追隨して移動するアルゴリズムは、オブジェクトを手の動きに追隨して移動させるアルゴリズムである。このアルゴリズムは、手に握ったオブジェクトをバーチャル人間が移動させる際に、オブジェクトの位置と姿勢を計算するために用いる。よって、このアルゴリズムの対象となるオブジェクトとしては、仮想空間内を自由に移動できるものとする。アルゴリズムとしては、以下のようになる。

1. バーチャル人間の手とオブジェクトの距離を計算する
2. 距離が閾値以下である場合は、手の動作に追隨してオブジェクトを移動させる

対象となるオブジェクトは、自由に動くことができるため、一度バーチャル人間の手を追隨を始めたオブジェクトは、本アルゴリズムが終了するまでは追隨を止めない。

5.3.5 アニメーション合成例

5.2節で述べたシナリオに基づき、5.3節で説明した動作合成手法を用いて合成したアニメーションの例を図5.15に示す。また、ドアを開ける動作の合成例を図5.16に、歩きながらコップを手にする動作の合成例を図5.17に示す。

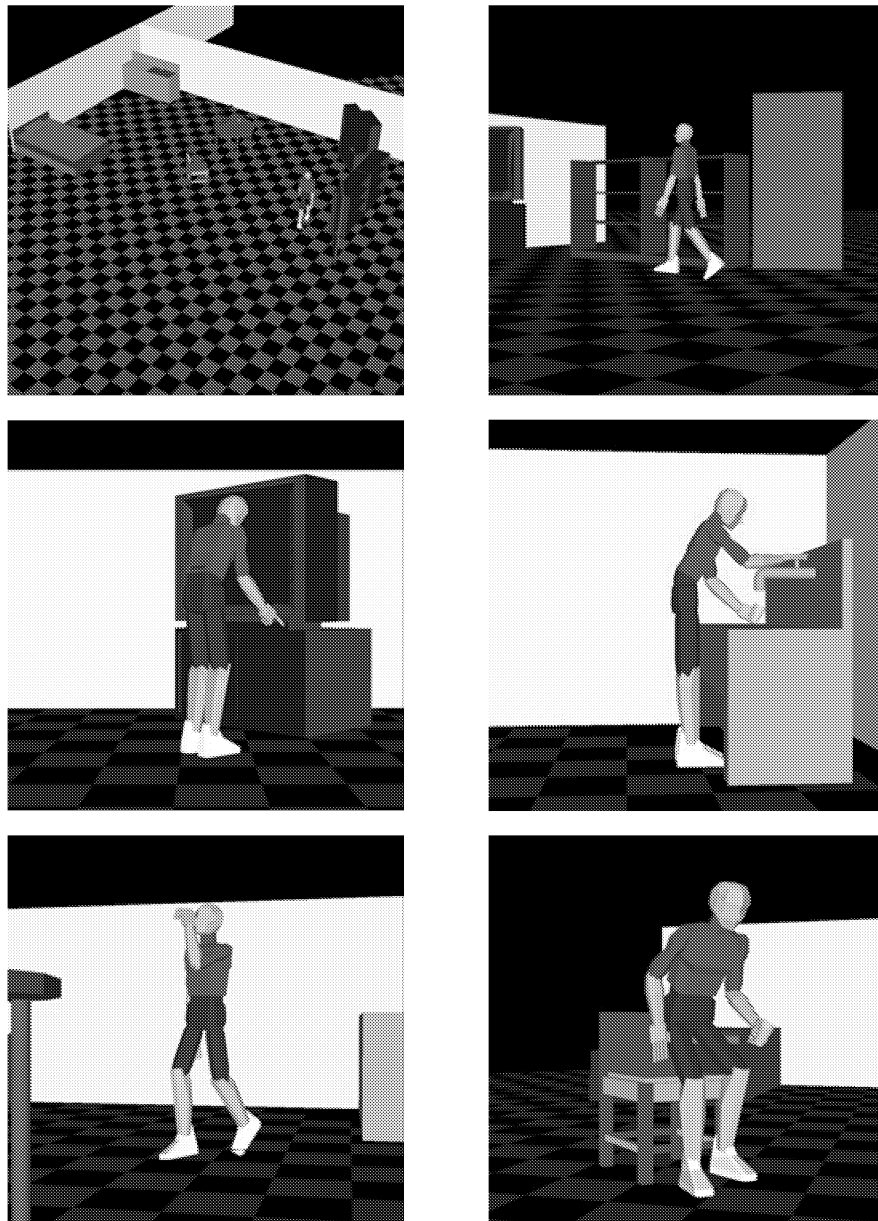


図 5.15: アニメーションの合成例

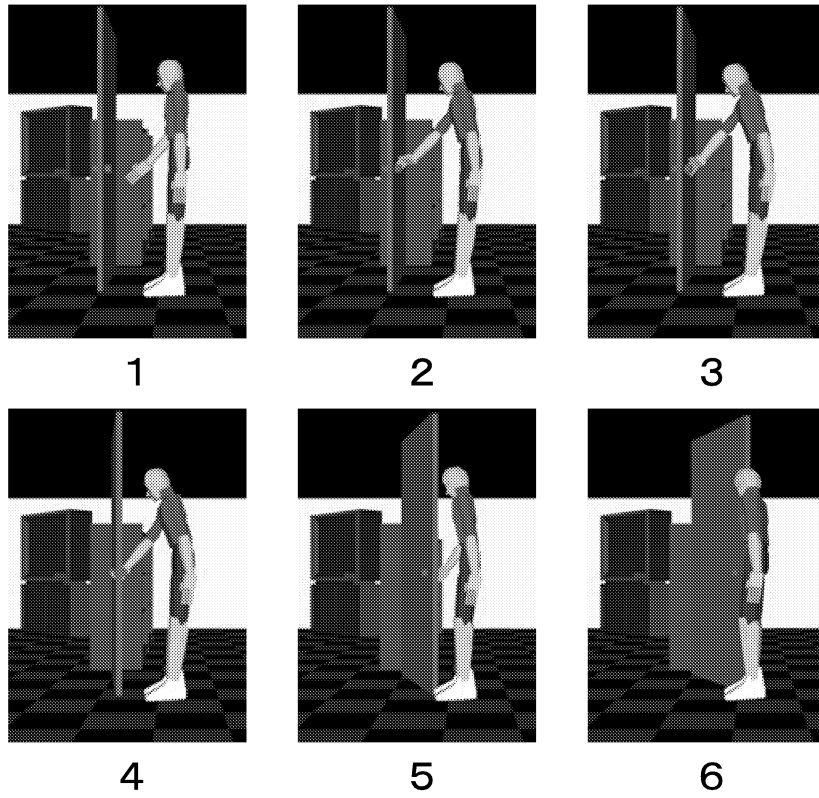


図 5.16: ドアを開ける動作の合成例

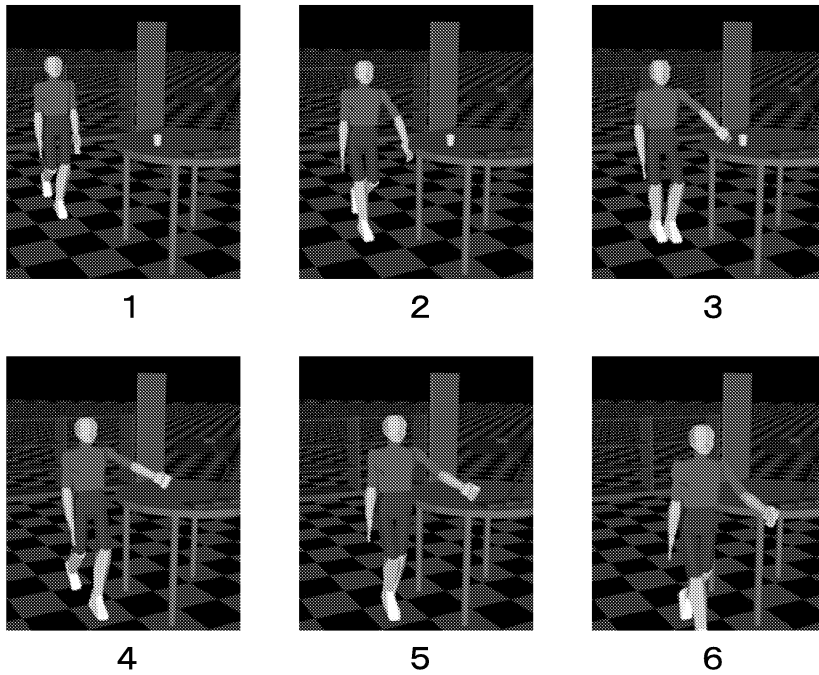


図 5.17: 歩きながらコップを手にする動作の合成例

5.4 AHMSSの評価と考察

本節では、AHMSSの評価と考察を行い、今後の展望について述べる。

5.4.1 AHMSSの評価

ここでは、AHMSSの評価を行う。評価手法としてはアルゴリズム毎のフレームレート（1秒間に計算・表示するコマ数）等を数値的に評価する他、AHMSSにより合成したアニメーションの見た目の主観評価を行う。

AHMSS評価時のシステム構成としては、図4.8のワークステーション1としてPCワークステーション（CPU:Pentium 700MHz×2、OS:Linux）を、ワークステーション2としてSGI社のグラフィックワークステーションOctane（CPU:R1000 250MHz）を用いた。

数値的評価

2.3節で述べたように、バーチャルコラボレータの動作はリアルタイムで合成する必要がある。そこで、AHMSSのリアルタイム性を検証するために、以下の3点に関して数値的評価を行った。

- 各アルゴリズムを用いて表示したアニメーションのフレームレート
- 各アルゴリズムの計算のフレームレート
- 各アルゴリズムプロセスの起動時間

4.2.4項で述べたように、AHMSSでは、仮想空間情報サーバと画像表示プロセスの動作は同期しておらず、画像表示プロセスは仮想空間情報サーバのデータ更新の如何に関わらず、1コマ分の描画が完了する毎に仮想空間情報サーバのデータを読み込む。すなわち、仮想空間情報サーバのデータ更新速度は実際の描画速度と異なる。よって、実際のアニメーションの表示速度と動作の計算速度の両方についてリアルタイム性を検証する必要がある。

また、AHMSSでは様々な動作合成アルゴリズムを用いているので、これらのフレームレートはアルゴリズムによっても異なる。よってアルゴリズム毎にもリアルタイム性を検証する必要がある。

さらに、4.2節で述べたように、AHMSSでは、バーチャル人間の動作合成アルゴリズムが複数同時に実行される場合や、オブジェクトの動作合成アルゴリズムが同時に実行される場合がある。よって、複数の動作合成アルゴリズムを同時に実行した場合のリアルタイム性を検証する必要がある。

以上より、次に示す6つの場合の表示のフレームレートと計算のフレームレートを計測した。

1. モーションキャプチャデータ再生アルゴリズムで動作を合成した場合
2. 歩行動作のアルゴリズムで動作を合成した場合
3. 球状線形補間アルゴリズムで動作を合成した場合
4. 歩行動作のアルゴリズムと球状線形補間アルゴリズムで動作を合成した場合（動作融合）
5. 歩行動作のアルゴリズムと手に追隨して移動するアルゴリズムで動作を合成した場合（オブジェクト操作）
6. 歩行動作のアルゴリズムと球状線形補間アルゴリズムと手に追隨して移動するアルゴリズムで動作を合成した場合（動作融合 + オブジェクト操作）

なお、4.2.3項で説明したように、物アルゴリズムプロセスはバーチャル人間の姿勢を基に動作合成を行うため、単独でのフレームレートを計測することはできない。結果を表5.5、表5.6に示す。ただし、表示のフレームレートは、仮想空間にバーチャル人間と12個のオブジェクトが存在する場合の計測結果である。バーチャル人間と各オブジェクトのポリゴン数を表5.7に示す。

また、AHMSSでは、動作合成アルゴリズムを動作毎に用意しており、それらはそれぞれ独立したプロセスとして構築している。そのため、実際に動作を合成する際には使用する動作合成アルゴリズムを起動する必要があるため、このアルゴリズム起動に時間がかかり、表示のリアルタイム性に影響を与えることが考えられる。そこで、各アルゴリズムの起動にかかる時間を計測した。計測対象とするアルゴリズムは、上記の表示・計算のフレームレートの計測の場合と同様とした。結果を表5.8に示す。

表 5.5: アルゴリズム毎のアニメーションの表示のフレームレート

アルゴリズム名	フレームレート (<i>frame/sec.</i>)	1フレーム当たり の時間 (<i>msec.</i>)
モーションキャプチャデータ 再生アルゴリズム	4.96	202
歩行動作のアルゴリズム	4.96	202
球状線形補間アルゴリズム	4.96	202
動作融合	4.96	202
オブジェクト操作	4.96	202
動作融合 + オブジェクト操作	4.96	202

表 5.6: アルゴリズム毎の計算のフレームレート

アルゴリズム名	フレームレート (<i>frame/sec.</i>)	1フレーム当たり の時間 (<i>msec.</i>)
モーションキャプチャデータ 再生アルゴリズム	24.98	40.03
歩行動作のアルゴリズム	24.93	40.11
球状線形補間アルゴリズム	24.97	40.05
動作融合	24.89	40.18
オブジェクト操作	12.48	80.13
動作融合 + オブジェクト操作	12.51	79.94

表 5.7: バーチャル人間とオブジェクトのポリゴン数

モデルの名称	ポリゴン数
バーチャル人間	1301
椅子	98
テーブル	320
ドア	268
床	10,368
流し台	185
テレビ	167
壁 1	2
壁 2	2
ベッド	94
電気スタンド	568
本棚	90
合計	13,679

表 5.8: アルゴリズム毎の起動時間

アルゴリズム名		起動に必要な時間 (msec.)
モーションキャプチャデータ再生アルゴリズム		5.2
歩行動作のアルゴリズム		5.3
球状線形補間アルゴリズム		5.2
動作融合	歩行動作のアルゴリズム	5.4
	球状線形補間アルゴリズム	6.1
オブジェクト操作	球状線形補間アルゴリズム	5.4
	手に追従して移動するアルゴリズム	5.9
動作融合 + オブジェクト操作	歩行動作のアルゴリズム	5.4
	球状線形補間アルゴリズム	6.8
	手に追従して移動するアルゴリズム	6.2

アニメーションの見た目の主観評価

2.1 節で述べたように、バーチャルコラボレータの動作は、人間が見て自然な動作であることが望ましい。そこで、AHMSSで合成したアニメーションの見た目の自然さの評価を行った。

アニメーションの見た目の主観評価は、5.2 節で述べたアニメーションを被験者に提示し、アニメーションを構成するそれぞれの動作について「自然である(4点)」「どちらかといえば自然である(3点)」「どちらかという和不自然である(2点)」「不自然である(1点)」の4段階で評価してもらうことにより行った。さらに、ある動作から他の動作への移行の仕方についても同様に4段階で評価してもらった。なお、被験者は男子大学院生4名および女子大学院生1名の計5名である。評価の対象とした動作とそれぞれの動作間の移行および全被験者の評価点の平均を表5.9に示す。

5.4.2 考察

本項では、5.4.1 項で行ったAHMSSの数値的評価とアニメーションの見た目の主観評価に対する考察を述べる。

数値的評価に対する考察

表5.5～表5.8より、以下のことが分かる。

- 表示のフレームレートはアルゴリズムに依存しない。
- バーチャル人間の動作合成とオブジェクトの動作合成を並列に行うと、計算のフレームレートは約半分になる。
- 各アルゴリズムの起動時間は10msec.以下と短い。

まず、表示のフレームレートについて考える。表5.5に示したように、表示のフレームレートは3.4節で示した人体モーション合成システムの要求仕様よりも小さく、改善する必要がある。計算のフレームレートは表示のフレームレートの倍以上であるため、計算の負荷は表示のフレームレートに影響を与えていないことが分かる。グラフィックワークステーションの描画能力は十分に高く、これも表示のフレームレートが小さい原因とはなり得ない。次に、4.7節に示した仮想空間情報サーバと画像表示プロセスの間の通信については、通信データ量は1kbyte以下と小さく、ハードウェアの問題ではない。

表 5.9: 評価の対象とした動作とそれぞれの動作間の移行

評価対象	評価点の平均
ドアの開閉	3.2
前の動作から次の動作への移行 1	2.4
歩行	3.0
前の動作から次の動作への移行 2	3.4
テレビの電源スイッチを押す	3.4
前の動作から次の動作への移行 3	2.8
歩行	2.8
前の動作から次の動作への移行 4	2.4
歩きながらコップに手を伸ばす	2.6
前の動作から次の動作への移行 5	2.4
歩きながらコップを握る	2.4
前の動作から次の動作への移行 6	2.8
歩行	2.8
前の動作から次の動作への移行 7	3.0
蛇口をひねる	2.2
前の動作から次の動作への移行 8	3.0
歩行	3.0
前の動作から次の動作への移行 9	3.4
歩きながらコップの水を飲む	3.4
前の動作から次の動作への移行 10	3.0
椅子に座る	3.6
平均	2.9

以上より、表示のフレームレートが小さい原因は、この2つのプロセス間の通信方式が問題であると予想される。現在、この2つのプロセス間の通信は、まず画像表示プロセスから仮想空間情報サーバにデータ送信リクエストを出す。仮想空間情報サーバは、このリクエストを受け取った後に、画像表示プロセスへとバーチャル人間の位置や姿勢等の描画に必要なデータを送信する。すなわち、1回のデータ通信のために2回の通信を行っており、これが表示のフレームレートが小さい原因であると考えられる。よって、表示のフレームレートを改善するためには、1回の通信で1回のデータ通信を行うようにシステムの仕様を変更すれば良い。

次に、計算のフレームレートについて考える。計算のフレームレートは、表5.6より、オブジェクトの操作を伴わない場合は、動作融合の場合も含めて $24\text{frame}/\text{sec}$. 以上であり、3.4節で示した人体モーション合成システムの要求仕様を満たすものである。しかし、オブジェクトの操作を伴う場合は、フレームレートは $12\text{frame}/\text{sec}$. 程度であり、オブジェクトの操作を伴わない場合のほぼ半分である。これは、AHMSSでは4.9節に示したように人の姿勢の計算と物の姿勢の計算を直列に処理しているためである。よって、人と物の姿勢の計算を並列に処理するようにシステムを変更すれば、この点は改善される。しかし、人と物の姿勢の計算を並列に処理した場合は、人の姿勢と物の姿勢の関係に矛盾が生じる可能性があり、この点を考慮しなければならない。

最後に、各アルゴリズムプロセスの起動に必要となる時間については、 10msec . 以下と非常に短いため、人体モーション合成のリアルタイム性にほとんど影響を与えないことが分かった。

アニメーションの見た目の主観評価に対する考察

表5.9より、評価点の平均が2.5点以下、すなわちどちらかというと不自然であると評価された項目は動作間の移行1、動作間の移行4、動作間の移行5、および歩きながらコップを握る動作、蛇口をひねる動作の5項目であることが分かる。動作間の移行1は、ドアの開閉から歩行への移行であり、バーチャル人間はドアを開閉した後直立姿勢をとり、その後歩行を開始する。被験者の感想から、評価点の低い原因がこの動作間の直立姿勢であることが分かった。この点に関しては、何らかの方法により改善する必要がある。動作間の移行4、動作間の移行5の評価点が低いことに関しては、被験者の感想から、コップを手にする動作自体がそれほど自然に感じられなかったこ

とが原因であることが分かった。実際、動作間の移行4～コップを握る動作の間は、評価点が全体的に低くなっている。また、コップを握る動作および蛇口をひねる動作の評価点が低い原因は、それぞれの動作自体が多少不自然であり、動作合成アルゴリズムを自然な動作を合成できるように改良する必要がある。アニメーション全体の評価点を平均すると2.9点であり、全体的にはほぼ自然な動作に見えたと思われ、AHMSSで自然な動作の合成が可能であるといえる。

5.5 今後の展望

本節では、バーチャルコラボレータの将来構想に基づき、今後の展望について述べる。また、本研究では人体モーション合成システムを認知科学の概念であるアフォーダンスに基づいて開発したが、これは人体モーション合成における新たな試みであり、アフォーダンスの観点からみた人体モーション合成システムの将来展望も併せて最後に述べる。

5.5.1 バーチャルコラボレータの将来構想への対応

バーチャルコラボレータの将来構想として、プラントを構成する機器の保修作業訓練の支援や複数のバーチャルコラボレータが協調して作業を行うマルチバーチャルコラボレータ等があり、これに対応するために必要な機能としては、以下のものが挙げられる。

- オブジェクト同士のインタラクションの実現
- バーチャル人間同士のコミュニケーション動作の実現

ここでは、これらの機能をAHMSSへ導入する際に解決すべき課題に関して述べる。

オブジェクト同士のインタラクションの実現

AHMSSでは、例えば、“レンチを使ってナットを回す”等のオブジェクト同士のインタラクションは実現していない。そのため、プラント内の機器保修作業訓練の支援を実現するために、仮想空間内のオブジェクト同士のインタラクションをシミュレーションする必要がある。たとえば、ボルトからナットを外すために、ナットを回さなければ外れないようにしたり、蓋がナットとボルトで固定されている場合は、ナット

をすべて外すまでは蓋が外れないようにしなければならない。これを、オブジェクトの3次元形状を基にオブジェクト間の完全な接触判定を行い、オブジェクトの動きを制限することによりシミュレーションする方法があるが、これは、形状データの精度や計算機の処理速度等の問題点があり、実現は困難である。よって、多くの場合、オブジェクトに「状態」の概念を導入し、各オブジェクトの取り得る状態に対してオブジェクト同士のインタラクションをシミュレーションするための計算機上での処理を予め定めておくことによって、バーチャル人間の動作に対応したオブジェクトの動作をシミュレーションする手法が採られている。この手法を実現するためには、以下のことが必要となる。

- 仮想空間内のオブジェクトの状態遷移を処理する機構

オブジェクト同士のインタラクションを実現するためには、バーチャル人間の動作に応じて仮想空間内のオブジェクトの状態がどのように変化するかについての情報も必要となる。たとえば、ペンが「机の上に置かれた状態」で、バーチャル人間が「右手でペンを握る」動作を行った場合、ペンの状態は「机の上に置かれた状態」から「右手に握られた状態」に遷移する必要がある。このような遷移の仕方を処理するための機構が必要となる。

- 訓練対象の機器に物理法則を適用する機構

上述の機構からの出力を基に物体の状態を決定すると、次にその状態に応じたオブジェクトの物理的な振る舞いを仮想空間内でシミュレーションする必要がある。たとえば、ペンが「空中で手を放され、机に向かって落ちつつある状態」にあれば、ペンが机と接触するまで自由落下させる必要がある。また、ナットが「ボルトにはまった状態」にあれば、ナットの動作はボルトを中心軸とした回転運動に限定する必要がある。このような物体相互間の拘束や動作をシミュレーションするための機構が必要である。

以上の機構を実現することによって、仮想空間内のオブジェクト間のインタラクションを実現することができる。

AHMSSは、アフォーダンスの概念に基づいており、その重要な特徴は、仮想空間へ新たに追加したオブジェクトや既存のオブジェクトに対するバーチャル人間の新たな動作の合成が可能なことである。そのため、上記の機構を構築する際には、この特徴を残すことができる方法を取らなければならない。その方法の1つとして、本研究室での石井らによる先行研究^{[19][20]}における方法を利用することができる。

石井らの研究では、仮想空間内での機器必修訓練を行う環境の構築支援システムを構築しており、このシステムは仮想空間へのオブジェクトの追加や追加したオブジェクトの状態遷移、オブジェクトの物理的な振る舞いを自由に設定することが可能となっている。

そこで、*AHMSS*を次のように改良することによって、オブジェクト同士のインタラクションを実現することが可能となる。

- オブジェクトの状態遷移を処理する機構は、石井らのシステムの機構を、仮想空間の情報を一元管理する仮想空間管理サーバに組み込むことによって実現する
- 訓練対象の機器に物理法則を適用する機構は、物アルゴリズムとして実現する
- バーチャル人間の動作情報である動作データや人アルゴリズム・物アルゴリズムは、オブジェクトに持たせるのではなく、オブジェクトのとり状態毎に持たせる

なお、バーチャル人間の動作データをオブジェクトの状態毎に持たせることは、アフォーダンスに沿った考え方であるといえる。

バーチャル人間同士のコミュニケーション動作の実現

マルチバーチャルコラボレータシステムでは、複数のバーチャルコラボレータが協調して作業を行う。そのため、バーチャルコラボレータ同士のコミュニケーション動作、すなわち身振りや手振りによるコミュニケーションを仮想空間内で実現することが必要となる。*AHMSS*では、複数のバーチャル人間の動作を合成することは可能であるが、バーチャル人間同士のコミュニケーション動作に関しては未だ実現していない。本研究では、アフォーダンスの概念を基に人体モーション合成システムを開発しているが、アフォーダンスの概念では、我々は他人からもアフォーダンスを知覚しているといえる。そこで、仮想空間内のオブジェクトと同様に、他のバーチャルコラボレータに動作データを持たせることによって、バーチャル人間同士のコミュニケーション動作を実現することができると思われる。

5.5.2 アフォーダンスに基づく人体モーション合成の将来展望

本研究では、人体モーション合成における新たな試みとして、認知科学の概念であるアフォーダンスに基づいて人体モーション合成システムを開発した。ここでは、このアフォーダンスの観点からみた人体モーション合成の将来展望を述べる。

本研究では、バーチャル人間の動作情報をその操作対象であるオブジェクトに持たせることにより、アフォーダンスの概念を人体モーション合成システムへと適用した。しかし、この方法はアフォーダンスの正確な適用とはいえない。

3.1.2項で述べたように、人は環境の中から面を知覚することによって包囲光配列の不変項をピックアップすると同時に、自分自身に関する情報を得ている。そして、環境から観察者自身にとっての意味や価値を直接知覚しており、これが元来のアフォーダンスの意味である。すなわち、面の知覚 = アフォーダンスの知覚ということが出来る。よって、アフォーダンスの概念を正確に仮想空間へと拡張するには、面を知覚し、それをバーチャル人間にとっての意味や価値であるアフォーダンスへと変換する機構が必要となる。

そこで、まず、面を知覚する機構であるが、これは仮想空間を構成するオブジェクトは複数の三角形ポリゴン、すなわち面から構成されている。この面に関する情報はすべてシステムが保持しているので、特に面を知覚するための機構は必要ない。よって、これらの面の構成をバーチャル人間にとっての意味や価値、すなわちバーチャル人間のアフォーダンスへと変換する機構が必要となる。このような機構を構築することは、膨大なデータの羅列から意味を抽出することを意味し、非常に困難である。しかし、このような機構と、そのアフォーダンスに対応するバーチャル人間の動作を合成する機構が実現されれば、どのような仮想空間内でもバーチャル人間の動作を合成することが可能となり、非常に汎用性の高い人体モーション合成システムを構築することが可能となる。

第 6 章 結論

本研究では、仮想空間へ新たに追加したオブジェクトや既存のオブジェクトに対する新たな動作の合成を容易に行うことができる、アフォーダンスの概念に基づく人体モーション合成システムである *AHMSS* (Affordance-based Human Motion Synthesizing System) の開発を行った。

第 2 章では、まず、マンマシンインタフェースの現状について述べ、本研究室で研究開発中のバーチャルコラボレータについて説明した。そして、バーチャルコラボレータの全体構成における本研究の位置付けを明らかにした。また、人体モーション合成に関する従来手法とその問題点について述べ、その解決策として、認知科学の概念であるアフォーダンスを利用することを提案し、本研究の目的を明らかにした。

第 3 章では、認知科学におけるアフォーダンスの概念について説明し、その人体モーション合成への適用手法について説明した。そして、アフォーダンスの概念を適用することによって生じる利点・欠点についてまとめた。また、アフォーダンスの概念に基づく人体モーション合成システムに求められる仕様をまとめた。

第 4 章では、人体モーション合成システムの基本となる人体モデルの構成とその姿勢の表現方法について述べた。そして、*AHMSS* の基本設計について説明し、そのハードウェア構成やソフトウェア構成、全体の処理の流れや各構成要素での処理について説明した。

第 5 章では、*AHMSS* を用いて作成したアニメーション例について説明した。また、アニメーションを作成するために必要となる動作の合成手法についても説明した。そして、*AHMSS* の評価を行い、今後の展望についてまとめた。

本研究では、人体モーションの合成に認知科学の概念を適用するという、これまでにない新しい試みを行ったが、実際にアフォーダンスの概念を適用した人体モーションの合成システムである *AHMSS* を開発することにより、その有効性を確認することができた。

認知科学においては、アフォーダンス以外にも様々な概念が提唱されており、今後それらの概念を含めて人体モーションの合成手法に適用することにより、これまでに解決できなかった様々な問題を解決する糸口が見つかる可能性は十分にあり、その適用を検討する有用性は高いといえる。

なお、今後の課題としては以下のことが挙げられる。

- アニメーションの表示のフレームレートの改善
- 動作間の移行を自然に合成する手法の開発
- オブジェクト同士のインタラクションの実現
- バーチャル人間同士のコミュニケーションの実現
- 仮想空間内のオブジェクトの面の構成をバーチャル人間のアフォーダンスへと変換する手法の開発

謝 辞

本研究を進めるにあたり、研究全般にわたってご指導を頂きました吉川榮和教授に深く感謝いたします。本研究を進めるにあたり、数々の貴重な助言を頂きました下田宏助教授に深く感謝いたします。本研究を進めるにあたり、数々の助言を頂き、協力して頂いた博士課程の石井裕剛氏に深く感謝いたします。本研究を進めるにあたり、プログラミング作業やその他様々な面で協力して頂いた修士課程 1 回生の小牧大輔君に深く感謝いたします。本研究の評価実験に協力して頂いた、修士課程 1 回生の伊藤京子さん、大坂融弘君、笹井寿郎君、米田賀一君、沖陽三君に心から感謝します。最後に、研究を進める上で何かとお世話頂いた谷友美秘書、藤岡美紀秘書および吉川研究室の学生の皆様にも心から感謝いたします。

参考文献

- [1] James J. Gibson : 生態学的視覚論, サイエンス社 (1995).
- [2] 佐々木正人 : 岩波科学ライブラリー 12 アフォーダンス - 新しい認知の理論, 岩波書店 (1994).
- [3] 田村博 : ヒューマンインタフェースとは, ヒューマンインタフェース (田村博編), オーム社, 第 1 章 (1998).
- [4] 下田宏, 吉川榮和 : 新しいヒューマンインタフェースとしてのバーチャルコラボレータの実験研究, 日本機械学会 第 6 回動力・エネルギー技術シンポジウム講演論文集, pp.171-176(1998).
- [5] 吉川榮和, 下田宏, Wu Wei, 石井裕剛, 小澤尚久 : 新しいインタフェース環境「バーチャルコラボレータ」の実験研究 (1) - 全体構想 -, 日本原子力学会 1999 年春の大会要旨集 第 分冊, E36, p.278(1999).
- [6] Douglas E. D. and Sudhanshu K. S. : Fast Techniques for Mixing and Control of Motion Unit for Human Animation, Proc. of Pacific Graphics '94, pp.229-242(1994).
- [7] Shimoda H. : A Computer-Aided Sensing and Design Methodology for the Simulation of Natural Human Body Motion and Facial Expression, 2nd International Conference on Engineering Design and Automation(CD-ROM), August 9-12, Hawaii(1998).
- [8] David T. Chen and David Zeltzer : Pump It Up : Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method, Computer Graphics, 26(2), pp.89-98(1992).
- [9] David C. Brogan, Ronald A. Metoyer and Jessica K. Hodgins : Dynamically Simulated Characters in Virtual Environments, IEEE Computer Graphics and Applications, Vol.18, No.5, pp.2-13(1998).

- [10] Jessica K. Hodgins and James F. O'Brien : Computer Animation, Encyclopedia of Computer Science, International Thomson Computer Press, Fourth Edition(1998).
- [11] 黒川隆夫 : 知覚と操作の相関, ヒューマンインタフェース (田村編), オーム社, 第3章 (1998).
- [12] Software Design 連載記事「〇〇って何?!」(<http://oobof.inarcadia.co.jp/library/SD/index.html>).
- [13] 春木良且 : オブジェクト指向への招待 - 思考表現のための新しい技法, 啓学出版 (1989).
- [14] 安藤寛 : 仮想空間内における自然な人体モーションの生成, 京都大学大学院工学研究科電気工学第二専攻修士論文 (1998).
- [15] 小牧大輔 : 人の歩行動作の計測と特徴分析に関する基礎研究, 京都大学工学部電気工学第二学科学士論文 (1999).
- [16] 宮本敏雄, 保坂秀正訳 : 数概念の拡張, 森北出版株式会社 (1995).
- [17] David Eberly : Quaternion Algebra and Calculus, Magic Software (<http://www.magic-software.com/src/graphics/quat/quat.pdf>).
- [18] Rotating Objects Using Quaternions (http://www.gamasutra.com/features/programming/19980703/quaternions_01.htm).
- [19] 石井裕剛 : 人工現実感技術を用いた機器保修訓練環境構築支援システムの開発, 京都大学大学院工学研究科電気工学第二専攻修士論文 (1998).
- [20] 石井裕剛, 手塚哲夫, 吉川榮和 : 人工現実感技術を用いた機器保修の訓練環境構築支援システムの開発, 日本バーチャルリアリティ学会論文誌, Vol.4, No.1, pp.303-312(1999).

付録目次

付録 A 歩行動作のアルゴリズム	付録 A-1
A.1 仮想空間内の移動	付録 A-1
A.2 動作データベースの作成	付録 A-3
A.2.1 直進の動作データの作成	付録 A-3
A.2.2 歩き始めの動作データの作成	付録 A-4
A.2.3 歩き終わりの動作データの作成	付録 A-5
A.2.4 1歩または2歩で到達する動作データの作成	付録 A-5
A.3 移動動作の合成	付録 A-7
A.3.1 方向と距離に応じた動作データの選択	付録 A-7
A.3.2 動作データの変形	付録 A-10
参考文献	付録 A-14
付録 B 手先の位置から手の姿勢を計算する逆運動学アルゴリズム	付録 B-1
B.1 手先の位置から手の姿勢を計算する逆運動学アルゴリズム	付録 B-1
B.1.1 肩の関節変数を固定する場合	付録 B-4
B.1.2 手首の関節変数を固定した場合	付録 B-7
B.1.3 適切な解の選択	付録 B-10
参考文献	付録 B-11

付録 図目次

A.1	移動の方向と目的地点で向いている方向	付録 A-2
A.2	動作データの選択から動作データの縮小率の決定までの流れ	付録 A-7
A.3	上半身の座標変換	付録 A-11
B.1	簡略化された腕のモデル	付録 B-2

付録表目次

A.1 作成した移動用の動作データの一覧	付録 A-6
--------------------------------	--------

付録 A 歩行動作のアルゴリズム

ここでは、5.5.2項で述べた、本研究室の先行の研究で開発された安藤の歩行動作のアルゴリズム^[1]を詳細に説明する。

なお、安藤の手法では人体モデルの姿勢を表現する手法として、本研究で用いている Quaternion ではなく、オイラー角 (*Yaw*、*Pitch*、*Roll*) を用いている。そのため、以下の説明では、姿勢はすべてオイラー角で表現されている。そして、実際に動作を合成する際には、アルゴリズムからの出力結果を Quaternion に変換して用いている。また、アルゴリズムを簡単にするために、バーチャル人間の歩行を直線移動のみに限定している。

A.1 仮想空間内の移動

人の歩行動作を考えた場合、その歩数が 1 歩もしくは 2 歩の場合は、同じ動作が繰り返されることはないが、その歩数が 3 歩以上の場合は、途中、直進の動作が繰り返されると捉えることができる。そこで、バーチャル人間が 2 つの地点間を移動する時の動作を、距離が短い場合 (歩数が 1 歩もしくは 2 歩の場合) と長い場合 (歩数が 3 歩以上の場合) に分けて考える。

- 距離が短い場合 (1 歩もしくは 2 歩の場合)

1 歩で目的の地点にたどり着くということは、まず直立の姿勢から片足を踏み出した段階で、その着地点がすでに目的の地点であるということである。そして、残っているもう片方の足は最初に踏み出した足に添えるように着地する。2 歩で目的の地点にたどり着く場合も同様に、2 歩目に踏み出した足の着地点が目的の地点になる。

次に移動方向と目的の地点で向いている方向に応じて歩行動作を分類する。ここで移動方向とは、最初の直立の姿勢で体全体が向いている方向のベクトルと、出発点から目的の地点までのベクトルとの相対的な角度のことである。図 A.1 の A に示すように、ここでは移動方向は -180 度から 180 度までの範囲となる。また、目的の地点で向いている方向とは、図 A.1 の B に示すように、歩行動作が終わっ

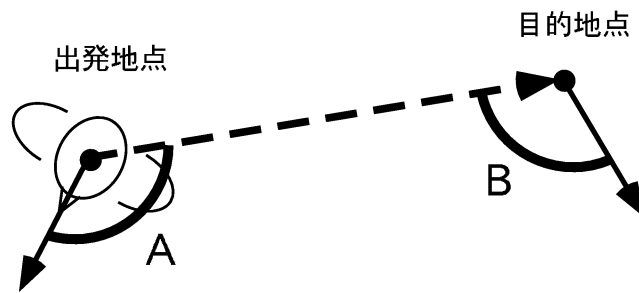


図 A.1: 移動の方向と目的地点で向いている方向

た時点で体全体が向いている方向のベクトルと移動方向との相対的な角度であり、同様に -180 度から 180 度までの範囲となる。

以上より、距離が短い場合の歩行動作としては、1歩と2歩のそれぞれについて、移動方向と、目的地点で向いている方向に応じて個々に動作データを作成する必要があることが分かる。ただし、 -180 度から 0 度までの動作データは、 180 度から 0 度までの動作データを反転させたものと考えることができるので、作成するのは移動方向と目的地点で向いている方向、共に、 $0\sim 180$ 度の動作データのみとなる。

- 距離が長い場合（3歩以上の場合）

人が3歩以上で移動する場合、その一連の動作は、まず移動方向に体を向けながら片方の足を振り出す「歩き始め」の動作、その後、移動距離に応じた回数、左右の足を前後に振り出す「直進」の動作、最後に、目的地点、方向に体を静止させる「歩き終わり」の動作の3つの動作から構成されると捉えることができる。以下では、歩き始め、直進、歩き終わりの3つの動作についてそれぞれ説明する。

- 歩き始め

ここでは、「歩き始め」とは、直立の姿勢から移動方向に向かって足を踏み出しつつ体を回転させ、2歩歩いた後に、体の向いている方向が移動方向と一致し、体の姿勢が直進の動作の始めの姿勢と一致するまでの動作とする。「歩き始め」の動作データとしては、全方向に歩き始める動作データが必要であるが、 -180 度から 0 度までの動作データは、 180 度から 0 度までの動作データを反転させたものと考えることができるため、「歩き始め」用の動作データとして、 0 度から 180 度の各方向に歩き出す動作データを作成する必要がある。

－ 直進

人が直進する動作は、片足を踏み出す動作を左右逆にしながら交互に繰り返す動作と捉えることができる。ここでは、この交互に繰り返される一連の動作の内、後方の足だけが地面と接触している状態から、前方の足が着地し、体が前進するにつれて足の前後が入れ替わり、そして、最初に後方にあった足が前に振り出されるまでの動作を「直進の単位動作」と呼ぶ。バーチャル人間が直進する動作を合成するには、右足が最初に前に来る直進の単位動作と、左足が最初に前に来る直進の単位動作が必要となるが、これら2つの動作は左右対称であるため、実際には右足が最初に前に来る直進の単位動作の動作データのみを作成する。

－ 歩き終わり

人がある程度長い距離を歩く場合、歩き始めや途中の直進の段階では、目的の地点に向かって歩くことを意識しているだけで歩幅の調整は行っておらず、歩き終わりの段階で初めて歩幅の調整を行っていると考えられる。また、最終的に体を向ける方向に関しても、歩き終わりの段階で、方向の調整を行っていると考えられる。また、人の歩行動作では、その移動距離に応じて、右足で目的の地点に着地し、左足を添える場合と、左足で目的の地点に着地し、右足を添える場合の2つの場合があると考えられる。従って、バーチャル人間の歩き終わりの歩行動作を合成する場合には、歩き終わりの方向に応じた0度から180度までの動作データを、右足を最後に添える動作と、左足を最後に添える動作の両方の動作に対して作成する必要がある。

A.2 動作データベースの作成

上で述べた歩行動作を合成するためには、動作データが必要である。以下に、必要な動作データについてまとめる。

A.2.1 直進の動作データの作成

3次元モーションキャプチャシステムにより計測し、補正した歩行動作のデータから、一步分のデータを切り出す。歩行動作を合成する場合、直進の動作データを左右逆にしながら連続して表示するが、通常、切り出したデータは、最初の姿勢と最後の姿勢が完全には一致しないので、一致するようにデータを修正する。ここでは、最初

の姿勢を基準にして、最後の姿勢を最初の姿勢の左右を反転させたものに一致させることにする。

最後の姿勢における、本文中の図4.3に示した階層構造の親子関係にある2つのリンクA、Bの間の座標変換を ${}^W R_{Last}$ 、最初の姿勢の左右を反転させた後の姿勢における同じリンク間の座標変換を ${}^W R_{Rev_First}$ とする。また、回転行列 R を $M(Yaw, Pitch, Roll)$ で表すことにする。ここで、 Yaw 、 $Pitch$ 、 $Roll$ はそれぞれ z 軸、 y 軸、 x 軸回りの回転角を表す。 ${}^W R_{Last}$ から ${}^W R_{Rev_First}$ への姿勢変化を表す回転行列 ${}^L R_F$ を求めると次式のようなになる。

$${}^L R_F = ({}^W R_{Last})^{-1} \cdot {}^W R_{Rev_First} = M(Y_{LF}, P_{LF}, R_{LF}) \quad (A.1)$$

次に、切り出したデータに含まれる姿勢の数を N 、 x 番目の姿勢のリンクA及びBの間の座標変換を R_{x-th} とし、修正後の直進の歩行におけるリンクA及びBの間の座標変換 R_{x-th}^{Modify} を以下のように定義する。

$$R_{x-th}^{Modify} = R_{x-th} \cdot M\left(\frac{Y_{LF}}{N}x, \frac{P_{LF}}{N}x, \frac{R_{LF}}{N}x\right) \quad (A.2)$$

最初の姿勢では $x = 0$ であり、 $R_{0-th}^{Modify} = R_{0-th}$ となり、最後の姿勢では、 $x = N$ であり、

$$R_{N-th}^{Modify} = R_{N-th} \cdot M(Y_{LF}, P_{LF}, R_{LF}) = {}^W R_{Rev_First} \quad (A.3)$$

となり、式(A.2)によって、最後の姿勢を最初の姿勢を左右反転させたものに一致させることができる。こうして得られたデータを直進の動作データとする。

A.2.2 歩き始めの動作データの作成

直進の動作データを作成した場合と同様に、補正した計測データから、2歩目までのデータを切り出す。作成する歩き始めの動作データの最初の姿勢は直立でなくてはならない。また、歩き始めの動作データの最後の姿勢は、直進の動作データの最初の姿勢を、移動方向分だけ回転させた姿勢と一致していなくてはならない。ここでは、歩き始めの動作データの最後の姿勢を、直進の動作データの最初の姿勢に一致するように修正する。

歩き始めの最初の姿勢における親子関係にある2つのリンクA及びBの間の座標変換を ${}^W R_{First}^{Start}$ 、最後の姿勢における座標変換を ${}^W R_{Last}^{Start}$ 、直進の動作の最初の姿勢を移

動方向だけ回転させた姿勢におけるリンク A 及び B の間の座標変換を ${}^W R_{First}^{Straight}$ 、直立姿勢におけるリンク A 及び B の間の座標変換を ${}^W R_{Stand}$ とする。式 (A.1) と同様に、以下の式を求める。

$$M_1(Y_F, P_F, R_F) = ({}^W R_{First}^{Start})^{-1} \cdot {}^W R_{Stand} \quad (A.4)$$

$$M_2(Y_L, P_L, R_L) = ({}^W R_{Last}^{Start})^{-1} \cdot {}^W R_{First}^{Straight} \quad (A.5)$$

そして、修正後の歩き始めの動作におけるリンク A 及び B の間の座標変換 R_{x-th}^{Modify} を次式で定義する。

$$R_{x-th}^{Modify} = R_{x-th} \cdot M \quad (A.6)$$

ただし、

$$M = (Y_F(\frac{N-x}{N})^{d1} + Y_L(\frac{x}{N})^{d2}, P_F(\frac{N-x}{N})^{d1} + P_L(\frac{x}{N})^{d2}, R_F(\frac{N-x}{N})^{d1} + R_L(\frac{x}{N})^{d2}) \quad (A.7)$$

$d1$ 、 $d2$ は 1 以上の適当な整数である。 $d1$ 及び $d2$ は、修正した後の 3 次元映像を見て、最も最適だと思われる値を決定した。

A.2.3 歩き終わりの動作データの作成

歩き終わりの動作データは、歩き始めの動作データとは逆に、最後の姿勢が直立であり、最初の姿勢が直進の動作の最後の姿勢に一致してはならない。具体的には、補正した計測データから最後の直立の姿勢から 2 歩さがった姿勢までを切り出し、歩き始めの動作データを作成した時と同じ方法で動作の修正を行う。

A.2.4 1 歩または 2 歩で到達する動作データの作成

1 歩または 2 歩で到達する動作データは、最初の姿勢と最後の姿勢が共に直立でなくてはならない。修正の方法は、歩き始めの動作データを作成した時と同じである。

以上述べた方法で作成した移動用の動作データの一覧を表 A.1 に示す。移動方向と目的地点で向いている方向については、共に動作データは 30 度毎に用意する。さらに変形の度合いを小さくするために、1 歩と 2 歩で到達する動作データと歩き終わりの動作データについては、歩幅が小さいものと普通のものの 2 種類の動作データを用意した。

表 A.1: 作成した移動用の動作データの一覧

動作の分類	各動作データの種類	データ数
1歩で到着	<ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ ・目的地点で向いている方向について0度から180度まで30度ステップ ・歩幅について普通の歩幅と小さい歩幅の2つ 	合計 98
2歩で到着	<ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ ・目的地点で向いている方向について0度から180度まで30度ステップ ・歩幅について普通の歩幅と小さい歩幅の2つ 	合計 98
歩き始め	<ul style="list-style-type: none"> ・移動方向について0度から180度まで30度ステップ 	合計 7
直進	<ul style="list-style-type: none"> ・1歩歩くデータ 	合計 1
歩き終り	<ul style="list-style-type: none"> ・目的地点で向いている方向について0度から180度まで30度ステップ ・最後に踏み出した足の左右 ・歩幅について普通の歩幅と小さい歩幅の2つ 	合計 28

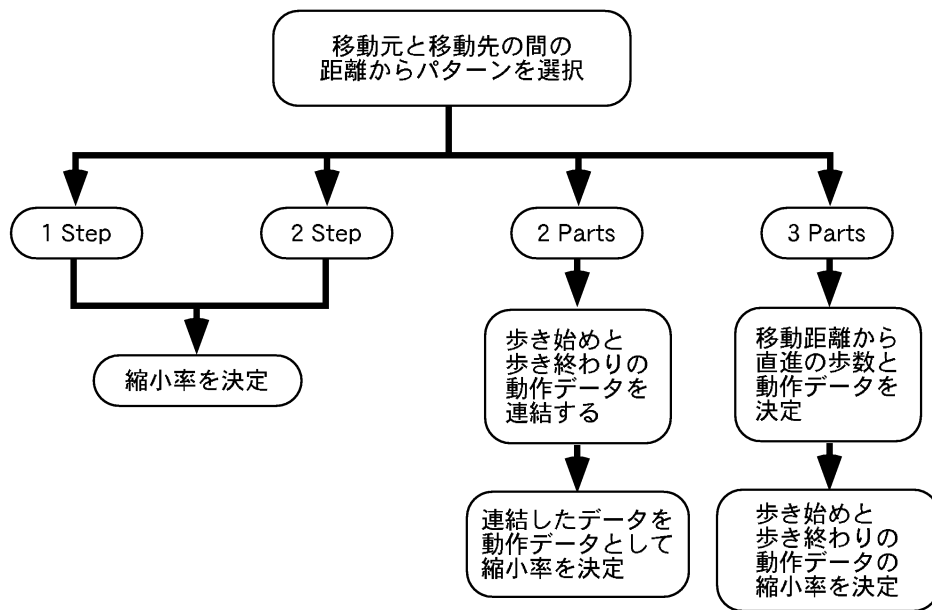


図 A.2: 動作データの選択から動作データの縮小率の決定までの流れ

A.3 移動動作の合成

次に、作成した動作データを用いて移動の動作を合成する方法について述べる。移動の動作を合成する場合、まず、移動の方向と距離に応じて適切な動作データを選択する。その後、選択した動作データの変形を行い、動作を合成する。

A.3.1 方向と距離に応じた動作データの選択

移動距離が長い場合は、移動動作を1つの動作ではなく、歩き始め、直進、歩き終わりといった複数の動作を組み合わせることにより合成する。また、予め準備する動作のデータベースに関しても、その方向や移動距離によって複数の種類の動作を用意している。また、移動距離が短い場合でも、その移動距離や到達地点で人体モデルが向いている方向によって複数の動作データを用意している。

動作データの選択から動作データの移動距離の縮小率の決定までの流れを図 A.2に示す。移動動作を合成するためには、まず、移動方向と距離に応じて適切な動作データを選択する必要がある。

最初に、移動元と移動先の2地点間の距離 (Len) に応じて、以下の4つのパターンに分類する。

1Step 1歩で移動先にたどり着く。

2Step 2歩で移動先にたどり着く。

2Parts 歩き始めと歩き終わりの動作データを直接連結した範囲で移動先にたどり着く。

3Parts 歩き始め、歩き終わりの間に直進のデータを含まなければ移動先にたどり着けない。

移動元と移動先の2地点間の距離に応じた具体的な分類方法を以下に述べる。

1. 合成する動作の移動方向に応じて、歩き始め、歩き終わり、1歩でたどり着くもの、2歩でたどり着くものの4種類の動作データを全て選択する。具体的には、動作データは、30度ごとに用意されているので、合成する動作の移動方向の ± 15 度以内に入る動作データを選択する。

2. 移動方向に応じて選択された、歩き始めの動作データの移動距離を Len_{First} 、普通の歩幅の歩き終わりの動作データの移動距離を Len_{LastL} とすると、

$$Len_{First} + Len_{LastL} < Len \quad (A.8)$$

の条件に当てはまる場合はこの移動動作を3Partsのパターンとする。

3. 移動方向に応じて選択された、2歩で普通の歩幅でたどり着く動作データの移動距離を Len_{2StepL} 、小さな歩幅の歩き終わりの動作データの移動距離を Len_{LastS} とすると、

$$Len_{2StepL} < Len \leq Len_{First} + Len_{LastL} \quad (A.9)$$

の条件に当てはまる場合はこの移動動作を2Partsのパターンとする。さらにここで、 $Len_{First} + Len_{LastS} < Len$ の場合は、歩き終わりの動作データは普通の歩幅のものを選び、そうでない場合は小さな歩幅の動作データを選択する。

4. 移動方向に応じて選択された、1歩で普通の歩幅でたどり着く動作データの移動距離を Len_{1StepL} 、2歩で小さな歩幅でたどり着く動作データの移動距離を Len_{2StepS} 、2歩で普通の歩幅でたどり着く動作データの移動距離を Len_{2StepL} とすると、

$$Len_{1StepL} < Len \leq Len_{2StepL} \quad (A.10)$$

の条件に当てはまる場合は、この移動動作を2Stepのパターンとする。ここで、 $Len_{2StepS} < Len$ の場合は、2歩で普通の歩幅でたどり着く動作データを選択する。そうでない場合は小さな歩幅の動作データを選択する。

5. 上記以外の場合は $Len \leq Len_{1StepL}$ なので、1Stepのパターンとなる。1歩で小さな歩幅でたどり着く動作データの移動距離を Len_{1StepS} とすると、 $Len_{1StepS} < Len$ の場合は1歩で普通の歩幅でたどり着く動作データを選択する。そうでない場合は小さな歩幅の動作データを選択する。

選択されたパターンが1Step、2Stepの場合、動作データの縮小率は Len を、選択された動作データの移動距離で割ったものである。2Partsのパターンの場合は、歩き始めと歩き終わりのデータを連結したデータを1つの動作データと考え、 Len をその動作データの移動距離で割ったものが縮小率となる。

3Partsのパターンについてはさらに細かく考える。人は通常、移動の際に歩き終わりの部分において距離の調整を行うものと考えられるので、基本的に、移動距離を調整するのは歩き終わりの部分とする。また、3Partsのパターンの場合は、他の3つのパターンとは異なり、要求されている移動距離に応じて直進の動作が繰り返される。そのため、直進部分の歩数も求める必要がある。そこで3Partsのパターンの場合は、以下のように動作データ及びその移動距離の縮小率を決定する。

1. まず、歩き始めの基本データの移動距離は変えないと考え、直進の部分の歩数を1つずつ増やしていく。そして、それによって歩き終りの部分で足の左右の着地順が変化するので、それを考慮に入れて、歩き終りの基本データは普通の歩幅のものを選ぶ。これらの合計の移動距離が Len を越えるまで直進の歩数を増やす。
2. 歩き終りの基本データの移動距離の縮小率 $ratio$ を算出する。これは、直進の移動距離を Len_M 、直進の歩数を num とすると、

$$ratio = (Len - Len_{First} - Len_M \times num) / Len_{LastL} \quad (A.11)$$

である。もし $ratio$ がある適当な値 $Rlimit$ 、例えば $Rlimit = 0.8$ よりも大きい場合はこのデータを選択する。 $Rlimit$ よりも小さい場合は、歩き終りの基本データを小さい歩幅のものに変えて $ratio$ を計算する。なぜなら、 $ratio$ が小さいということは変形の度合いが大きいということであり、普通の歩幅の基本データを採用すると動作が不自然になってしまうからである。小さい歩幅の歩き終りの基本データの移動距離を Len_{LastS} として式 (A.11) の Len_{LastL} を Len_{LastS} に置き換えて同様に計算する。 $Rlimit \leq ratio \leq 1$ の場合はこのデータを選択する。

ここで、 $ratio > 1$ の場合は直進の歩数を1つ増やす。この時、足の左右が切り替

わるため、選択する基本データも変わることにも注意する。そして再び上述の方法により $ratio$ を計算する。 $Rlimit \leq ratio \leq 1$ の場合はこの基本データを選択する。

3. 2. で適切な基本データを選択できなかった場合は、歩き始めの基本データの移動距離も縮小する。1. が終わった段階で算出されている直進の歩数を Num として、 $ratio1$ を以下のように計算する。

$$ratio1 = (Len - Len_M \times Num) / (Len_{First} + Len_{LastL}) \quad (A.12)$$

また、2. の最後の段階で算出されている直進の歩数、つまり、歩き終りの基本データの歩幅を小さいものとして $ratio$ が1より小さくなるように選択された直進の歩数を $Num2$ として、式 (A.12) と同様に、

$$ratio2 = (Len - Len_M \times Num2) / (Len_{First} + Len_{LastS}) \quad (A.13)$$

を計算する。 $ratio1$ と $ratio2$ を比較し、 $ratio1$ が大きい場合は歩き終りの基本データとして普通の歩幅のものを、そうでない場合は小さい歩幅のものを選ぶ。

ここまでの手順により適切な動作データ及び歩き始めと歩き終わりの部分の適切な縮小率が決定される。

A.3.2 動作データの変形

移動動作をその方向や距離に応じて自由に合成するためには、移動方向の調整や移動距離の縮小といった動作データの変形が必要である。このとき、下半身(腰よりも位置が下の部位、腰を含まない)は、足が地面と接触しているため、移動動作中に足が滑ったように見えないように動作を変形する必要がある。以下では動作の変形方法を、上半身と下半身に分けて述べる。

上半身の姿勢の決定

まず、人体モデルの上半身の各部位について、注目している部位のローカル座標系に姿勢変化を与えた時の座標系を Σ_{Part} 、さらに移動方向と移動距離の変形を加えた時の座標系を Σ'_{Part} とする。 Σ_{Part} と Σ'_{Part} の関係をワールド座標系の Z 軸の正から負の方向へ眺めた図を図 A.3 に示す。図では、左上腕に注目して、その変形前のローカル座標系と変形後のローカル座標系を示している。

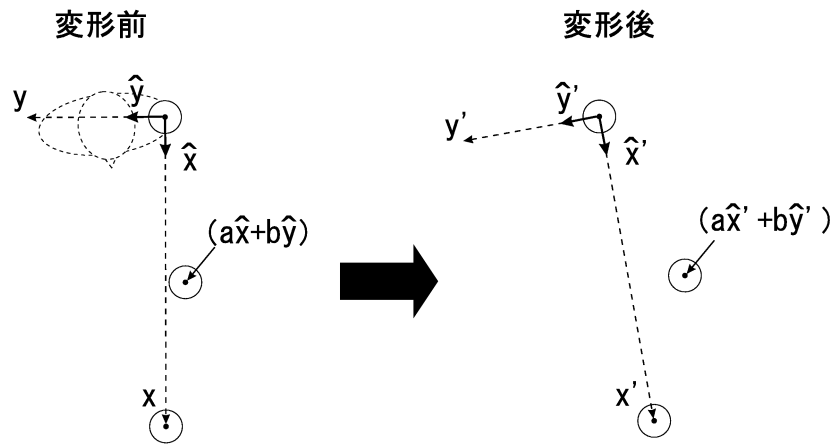


図 A.3: 上半身の座標変換

まず、 Σ'_{Part} の原点の座標を定める方法を説明する。以下では、特に断らない限り、ワールド座標系の x - y 平面のみで考える。つまり、各座標系の原点のワールド座標系における z 座標は同じものとする。

1. まず、図 A.3の左側の図において、動作データの最初の姿勢における Σ_{Part} の原点から最後の姿勢における原点へのベクトルの方向を X 軸とし、 X 軸方向の単位ベクトルを \hat{x} とする。次に、 X 軸をワールド座標系の z 軸に対して 90 度回転させたものを Y 軸とし、 Y 軸方向の単位ベクトルを \hat{y} とする。そして移動動作中の Σ_{Part} の原点の座標をこれら 2 つの単位ベクトルで表したものを $(a\hat{x} + b\hat{y})$ とする。
2. 次に、変形後の動作データについて、1.と同様に、最初の姿勢の Σ'_{Part} の原点から最後の姿勢の Σ'_{Part} の原点へのベクトルを X' 軸とし、 X' 軸方向の単位ベクトルを \hat{x}' とする。また X' 軸をワールド座標系の z 軸に対して 90 度回転させたものを Y' 軸とし、 Y' 軸方向の単位ベクトルを \hat{y}' とする。
3. 以上定めた a 、 b 、 \hat{x}' 、 \hat{y}' を用いて、変形した移動動作中の Σ'_{Part} の原点の座標を $(a\hat{x}' + b\hat{y}')$ と定める。

次に Σ'_{Part} の姿勢を定める方法を述べる。 Σ'_{Part} の姿勢は、 Σ'_{Part} を定める x 軸、 y 軸、 z 軸の 3 軸の内、 2 軸の方向が定まると求めることができる。まず Σ_{Part} の z 軸の原点を基準とする単位ベクトルの先端の座標について考える。

1. 動作データの最初の姿勢の Σ_{Part} の原点を始点とした z 軸方向の単位ベクトル（以下、 z 軸の単位ベクトル）の先端から最後の姿勢の z 軸の単位ベクトルの先端へ

のベクトルを X 軸とし、X 軸方向の単位ベクトルを \hat{x} とする。次に X 軸をワールド座標系の z 軸に対して 90 度回転させたものを Y 軸とし、Y 軸方向の単位ベクトルを \hat{y} とする。そして、移動動作中の Σ_{Part} の z 軸の単位ベクトルの先端の座標を $(a\hat{x} + b\hat{y})$ と表す。

2. 変形後の動作データについても同様に X' 軸と Y' 軸を定め、 \hat{x}' 、 \hat{y}' をその単位ベクトルとする。そして、変形後の移動動作中の Σ'_{Part} の z 軸の単位ベクトルの先端の座標を、 a 、 b を用いて $(a\hat{x}' + b\hat{y}')$ と定める。
3. 変形後の移動動作中の Σ'_{Part} の x 軸の単位ベクトルの先端の座標も同様に定める。
4. 変形後の動作データの Σ'_{Part} の x 軸の単位ベクトルの先端の座標をワールド座標系の x-y 平面へ正射影した点を Q とする。そして、単位ベクトルの先端の座標を正射影したものが点 Q に一致する Σ'_{Part} の原点を基準とする単位ベクトルを求め、これを Σ'_{Part} の x 軸の単位ベクトルとする。
5. 同様に、 Σ'_{Part} の z 軸が Σ'_{Part} の x 軸と直交していることを考慮しながら Σ'_{Part} の z 軸の単位ベクトルを求める。
6. Σ'_{Part} の x 軸と z 軸の単位ベクトルから、 Σ'_{Part} の姿勢を求める。

以上のようにして、移動距離と移動方向の変形後の上半身の各部位の位置と姿勢を定める。

下肢の姿勢の決定

下肢の姿勢を決定する場合、まず、太股 (Leg0) の姿勢を決定し、次に地面 (ワールド座標系の x-y 平面) と接触している部位である足 (Leg2) の姿勢を決定し、最後に脛 (Leg1) の姿勢を決定する。なお、以下の説明で用いる各リンクのローカル座標系の x,y,z の各軸は、本文中の図 4.4 に示した各リンクのローカル座標系の x,y,z の各軸を表している。

1. まず、太股 (Leg0) について考える。Leg0 のローカル座標系 (Σ_{Leg0}) の原点 (股関節) を、上半身の姿勢を決定した時に既に定まっている腰の姿勢を用いて求める。

次に、Leg0 の Σ_{Leg0} の x 軸を求める。Leg0 は Leg1 (脛) のローカル座標系 (Σ_{Leg1}) の原点に繋がっていることを利用し、上半身の姿勢を求めた時と同じ方法で求め

た Σ_{Leg1} の原点をワールド座標系のx-y平面へ正射影した座標と、先に求めたLeg0の Σ_{Leg0} の原点をワールド座標系のx-y平面へ正射影した座標を結んだベクトルを、Leg0の Σ_{Leg0} のx軸をx-y平面へ正射影したものに等しいとする。

そして、Leg0の長さを Σ_{Leg0} のx軸を表すベクトルの長さとして、そのベクトルの先端の座標のワールド座標系のx-y平面への正射影からLeg0の方向を求める。 Σ_{Leg0} のz軸に関しては、上半身の姿勢を求めた方法と同様にして求める。

以上の方法により、太股(Leg0)の姿勢が求まる。

- 次に足(Leg2)の姿勢を決定する。まず、変形前の動作データの腰(Hip)のローカル座標系 Σ_{Hip} のz軸を、ワールド座標系のx-y平面へ正射影したものと、変形後の動作データの腰のローカル座標系 Σ'_{Hip} のz軸を、ワールド座標系のx-y平面へ正射影したものとをなす角を求め、これを θ とする。

次に、変形する前の動作データのLeg2のワールド座標系から見た姿勢に対し、ワールド座標系のz軸回りに θ だけ回転させ、このときの姿勢を変形後のLeg2の姿勢とする。

- さらにLeg2のローカル座標系の原点を求める。まず、注目しているLeg2が空中にある場合、上半身の姿勢を求めた方法と同様にしてLeg2のローカル座標系の原点を求める。一方、Leg2が空中から地面に接触した場合、その接触地点が滑らないように接触地点を固定し、その接触地点と先に求めたLeg2の姿勢から、Leg2のローカル座標系の原点を求める。
- 最後に脛(Leg1)の姿勢を求める。Leg0の姿勢を求める際に求めたLeg1のローカル座標系 Σ'_{Leg1} の原点を、ワールド座標系のx-y平面へ正射影した点から、先に求めたLeg2の Σ'_{Leg2} の原点をx-y平面に正射影した点へのベクトルを、Leg1の Σ'_{Leg1} のx軸を表すベクトルのワールド座標系のx-y平面への正射影とし、後はLeg0の姿勢を求めた方法と同様にして、Leg1の姿勢を決定する。

以上のようにして、移動距離と移動方向の変形後の下半身の各部位の位置と姿勢を求める。

参考文献

- [1] 安藤寛：仮想空間内における自然な人体モーションの生成, 京都大学大学院工学研究科電気工学第二専攻修士論文 (1998).

付録 B 手先の位置から手の姿勢を計算する逆運動学アルゴリズム

B.1 手先の位置から手の姿勢を計算する逆運動学アルゴリズム

逆運動学アルゴリズムは、操作対象に手を伸ばす際の腕の動作を合成するために利用する。本研究では、逆運動学アルゴリズムとして、Tolaniらの手法^[1]を採用する。

人間の腕は非常に複雑であるが、肩に球状の、肘には外巻きの、そして手首にも球状の関節をもつ、7自由度の機構としてモデル化される。このモデルでは、肩甲骨の動きや前腕の回内運動を無視しているが、視覚的な要求を満たすのには十分な構造である。肩に固定座標系0を設定し、各関節に可動座標系 $i(i = 1 \dots 7)$ を設定する。 A_i を、フレーム $i-1$ からフレーム i への 4×4 の同次の座標系変換行列として定義する。この座標変換行列は、関節変数 θ_i の関数である。図B.1に示したモデルにおいて、 A_i の値は以下ようになる。

$$A_1 = R_z(\theta_1) = \begin{bmatrix} c1 & -s1 & 0 & 0 \\ s1 & c1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (B.1)$$

$$A_2 = R_x(\theta_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c2 & -s2 & 0 \\ 0 & s2 & c2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (B.2)$$

$$A_3 = R_z(\theta_3)T(0, 0, L1) = \begin{bmatrix} c3 & -s3 & 0 & 0 \\ s3 & c3 & 0 & 0 \\ 0 & 0 & 1 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (B.3)$$

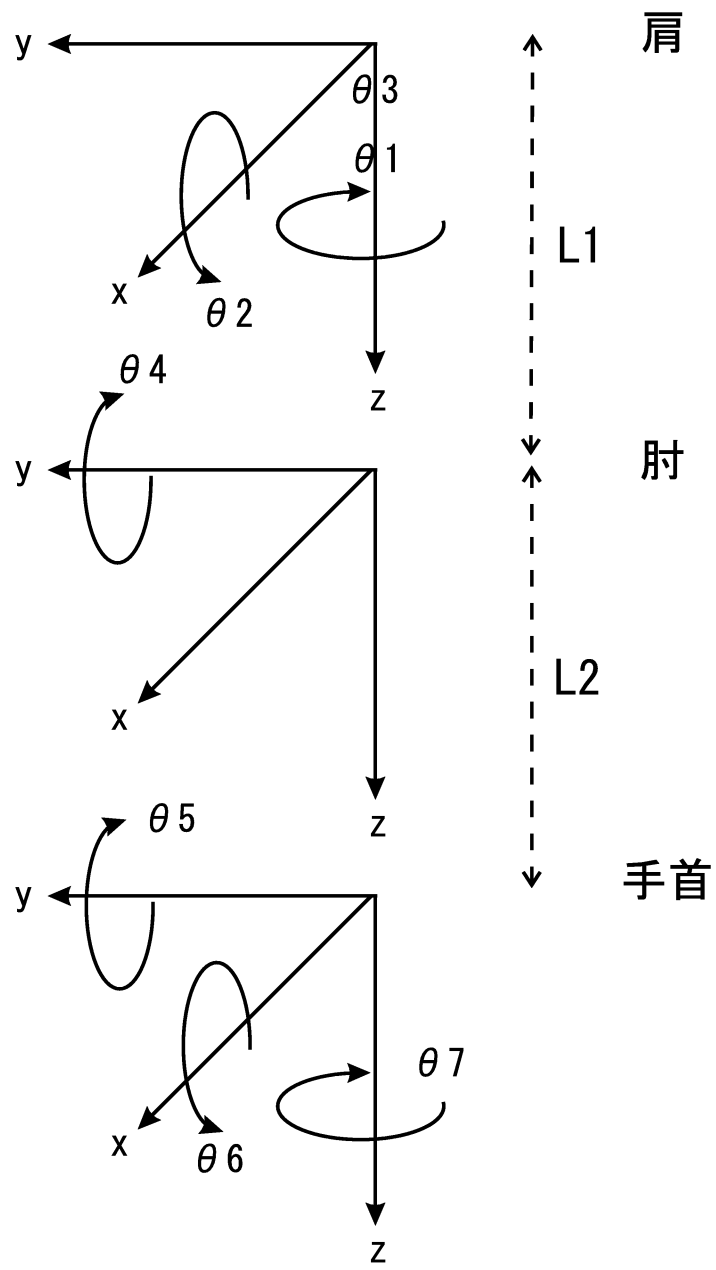


図 B.1: 簡略化された腕のモデル

$$\mathbf{A}_4 = \mathbf{R}_y(\theta_4)\mathbf{T}(0, 0, L_2) = \begin{bmatrix} c_4 & 0 & s_4 & s_4L_2 \\ 0 & 1 & 0 & 0 \\ -s_4 & 0 & c_4 & c_4L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.4})$$

$$\mathbf{A}_5 = \mathbf{R}_y(\theta_5) = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ 0 & 1 & 0 & 0 \\ -s_5 & 0 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.5})$$

$$\mathbf{A}_6 = \mathbf{R}_x(\theta_6) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_6 & -s_6 & 0 \\ 0 & s_6 & c_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.6})$$

$$\mathbf{A}_7 = \mathbf{R}_z(\theta_7) = \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.7})$$

ここで、 c_i と s_i は、 $\cos(\theta_i)$ と $\sin(\theta_i)$ を表し、 L_1 と L_2 は上腕と下腕の長さを表す定数である。

目的とする手首の座標系の、肩の座標系における位置と角度を表す行列を $\mathbf{A}_{\text{wrist}}$ とすると、逆運動学問題は、次の (B.8) を満たす $\theta_1, \dots, \theta_7$ の角度の組を導出することである。

$$\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6\mathbf{A}_7 = \mathbf{A}_{\text{wrist}} \quad (\text{B.8})$$

$\mathbf{A}_{\text{wrist}}$ の要素が (B.9) のように表せるとすると、

$$\mathbf{A}_{\text{wrist}} = \begin{bmatrix} \vartheta_{11} & \vartheta_{12} & \vartheta_{13} & \vartheta_{14} \\ \vartheta_{21} & \vartheta_{22} & \vartheta_{23} & \vartheta_{24} \\ \vartheta_{31} & \vartheta_{32} & \vartheta_{33} & \vartheta_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.9})$$

ベクトル $\mathbf{p} = [\vartheta_{14}, \vartheta_{24}, \vartheta_{34}]^T$ は、固定座標系における手首の位置を表す。手首の位置 \mathbf{p} が決まると、次の (B.10) によって手首と肩の距離から θ_4 は一意に決定される。

$$\theta_4 = \pi \pm \arccos\left(\frac{L1^2 + L2^2 - \|\mathbf{p}\|^2}{2L1L2}\right) \quad (\text{B.10})$$

θ_4 には2つの解が存在するが、肘の関節が1方向にしか曲がらないことを考慮すると、解は1つに決まる。これより、(B.8)において、 A_{wrist} は6つの独立変数によって決まるが、(B.8)を満たす $\theta_1, \theta_2, \theta_3, \theta_5, \theta_6, \theta_7$ の組み合わせは無数にある。そこで、有限個の解を導出するために、新たな拘束条件が必要となる。

この拘束条件として、手首もしくは肩の関節変数の1つを適当な値に固定することによって自由度を下げ、残りの関節変数について解を求める。この方法では、(B.8)は、手首もしくは肩のいずれかの関節が球状関節である6自由度のマニピュレータの逆運動学問題となる。これは、ロボティクスの分野では良く知られたものであり、以下に、肩の関節変数を固定する場合と手首の関節変数を固定する場合について、それぞれ3つ、合計6つの場合に分けて解の導出方法を述べる。

B.1.1 肩の関節変数を固定する場合

まず、手首の位置は最初の4つの関節変数にのみ依存するので、次の(B.11)が成立する。

$$A_1 A_2 A_3 A_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \vartheta_{14} \\ \vartheta_{24} \\ \vartheta_{34} \\ 1 \end{bmatrix} \quad (\text{B.11})$$

- θ_1 を固定した場合

θ_1 を既知とした場合には、上記の(B.11)の両辺に A_1^{-1} を掛けることによって、以下の式が得られる。

$$\begin{aligned} s_4 L_2 c_3 &= c_1 t \vartheta_{14} + s_1 \vartheta_{24} \\ s_4 L_2 s_3 c_2 + (-c_4 L_2 - L_1) s_2 &= -s_1 \vartheta_{14} + c_1 \vartheta_{24} \\ (c_4 L_2 + L_1) c_2 + s_4 L_2 s_3 s_2 &= \vartheta_{34} \end{aligned} \quad (\text{B.12})$$

これらより、2つの θ_3 が解として求められる。

$$\theta_3 = \pm \arccos \left(\frac{c_1 \vartheta_{14} + s_1 \vartheta_{24}}{s_4 L_2} \right) \quad (\text{B.13})$$

θ_3 に対して、 θ_2 は以下のように計算される。

$$\begin{aligned} \theta_2 &= \arctan 2(ad - bc, ac + bd) \\ a &= s_4 L_2 s_3 \\ b &= c_4 L_2 + L_1 \\ c &= -s_1 \vartheta_{14} + c_1 \vartheta_{24} \\ d &= \vartheta_{34}. \end{aligned} \quad (\text{B.14})$$

- θ_2 を固定した場合

θ_2 を既知とした場合には、(B.12)の最後の式から、 θ_3 の2つの解が求められる。

$$\begin{aligned} \theta_3 &= \begin{cases} \arcsin(\eta) \\ -[\pi - \arcsin(\eta)] \end{cases} \\ \eta &= \frac{\vartheta_{34} - (c_4 L_2 + L_1) c_2}{s_4 L_2 s_2} \end{aligned} \quad (\text{B.15})$$

そして、 θ_1 は次の(B.16)によって求められる。

$$\begin{aligned} \theta_1 &= \arctan 2(ad - bc, ac + bd) \\ a &= \vartheta_{24} \\ b &= \vartheta_{14} \\ c &= s_4 L_2 s_3 c_2 + (-c_4 L_2 - L_1) s_2 \\ d &= s_4 L_2 c_3. \end{aligned} \quad (\text{B.16})$$

- θ_3 を固定した場合

(B.11)の両辺に $(A_1 A_2)^{-1}$ を掛けることによって、以下の式が導出される。

$$s_4 L_2 c_3 = c_1 \vartheta_{14} + s_1 \vartheta_{24}$$

$$\begin{aligned}
s_4 L_2 c_3 &= (-s_1 \vartheta_{14} + c_1 \vartheta_{24}) c_2 + s_2 \vartheta_{34} \\
c_4 L_2 + L_1 &= c_2 \vartheta_{34} + (s_1 \vartheta_{14} - c_1 \vartheta_{14}) s_2
\end{aligned} \tag{B.17}$$

これらの式より、 θ_1 と θ_2 が以下のように求められる。

$$\begin{aligned}
\theta_1 &= \arctan 2(\vartheta_{24}, \vartheta_{14}) \\
&\quad \pm \arctan 2\left(\sqrt{\vartheta_{14}^2 + \vartheta_{24}^2 - (s_4 c_3 L_2)^2}, s_4 c_3 L_2\right) \\
\theta_2 &= \arctan 2(ad - bc, ac + bd) \\
a &= \vartheta_{34} \\
b &= (-s_1 \vartheta_{14} + c_1 \vartheta_{24}) \\
c &= c_4 L_2 + L_1 \\
d &= s_4 L_2 s_3.
\end{aligned} \tag{B.18}$$

- 手首の姿勢の計算

θ_1 、 θ_2 、 θ_3 そして θ_4 が決まると、手首の角度は、次の手法により計算される。

(B.8) は、次のように変形することができる。

$$\mathbf{A}_5 \mathbf{A}_6 \mathbf{A}_7 = (\mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 \mathbf{A}_4)^{-1} \text{wrist} \tag{B.19}$$

右辺の行列の回転成分を $r_{ij} (i = 1 \dots 3, j = 1 \dots 3)$ で表し、左辺を展開すると、(B.20) に示す 9 つの等式が得られる。

$$\begin{bmatrix} c_5 c_7 + s_5 s_6 s_7 & -c_5 s_7 + s_5 s_6 c_7 & s_5 c_6 \\ c_6 s_7 & c_6 c_7 & -s_6 \\ -s_5 c_7 + c_5 s_6 s_7 & s_5 s_7 + c_5 s_6 c_7 & c_5 s_6 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{B.20}$$

これより、 θ_6 に対して 2 つの解が求められる。

$$\theta_6 = \begin{cases} \arcsin(-r_{23}) \\ -[\pi + \arcsin(-r_{23})] \end{cases} \tag{B.21}$$

そして、 θ_5 と θ_7 は、適切な $\arctan 2$ を計算することによって求められる。

$$c_6 \neq 0 \begin{cases} \theta_5 = \arctan 2 \left(\frac{r_{13}}{c_6}, \frac{r_{33}}{c_6} \right) \\ \theta_7 = \arctan 2 \left(\frac{r_{21}}{c_6}, \frac{r_{22}}{c_6} \right) \end{cases}$$

$$c_6 = 0 \begin{cases} \theta_5 = \arctan 2(-r_{31}, r_{11}) \\ \theta_7 = 0 \end{cases} \quad (\text{B.22})$$

このように1組の肩と肘の関節変数に対して、手首の姿勢の解が2組存在する。よって、逆運動学方程式の解は2組存在する。

以上より、これら3つの場合においては、肩の関節変数の1つを固定することによって、他の2つの肩の関節変数に対して2組の解が導かれる。そして、それぞれの肩の関節変数に対して、上記のように2組の手首の関節変数が求められる。それゆえ、肩の関節変数の1つを固定することにより、(B.8)に対して最大4組の異なる解が導かれることになる。

B.1.2 手首の関節変数を固定した場合

次に、手首の関節変数のうちの1つを固定した場合について考える。肩の関節の位置は、肘と手首の関節のみの関数であるので、 θ_4 、 θ_5 、 θ_6 、 θ_7 のみ含む、3つの方程式を得ることが可能である。

$$\mathbf{A}_7(\mathbf{A}_{\text{wrist}})^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = (\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6)^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \alpha c_7 - \beta s_7 \\ \alpha s_7 + \beta c_7 \\ \gamma \\ 1 \end{bmatrix} = (\mathbf{A}_5\mathbf{A}_6)^{-1} \begin{bmatrix} s_4 L_1 \\ 0 \\ -c_4 L_1 - L_2 \\ 1 \end{bmatrix} \quad (\text{B.23})$$

ここで、

$$\begin{aligned}
 \alpha &= -(\vartheta_{11}\vartheta_{14} + \vartheta_{21}\vartheta_{24} + \vartheta_{31}\vartheta_{34}) \\
 \beta &= -(\vartheta_{12}\vartheta_{14} + \vartheta_{22}\vartheta_{24} + \vartheta_{32}\vartheta_{34}) \\
 \gamma &= -(\vartheta_{13}\vartheta_{14} + \vartheta_{23}\vartheta_{24} + \vartheta_{33}\vartheta_{34}).
 \end{aligned} \tag{B.24}$$

とする。

- θ_7 を固定した場合

(B.23)において、 A_5 、 A_6 を左辺に移項し、展開すると以下の式が得られる。

$$\begin{aligned}
 (\alpha c_7 - \beta s_7)c_5 + (\alpha s_7 + \beta c_7)s_5s_6 + \gamma c_6s_5 &= s_4L_1 \\
 (\alpha s_7 + \beta c_7)c_6 - \gamma s_6 &= 0 \\
 -(\alpha c_7 - \beta s_7)s_5 + (\alpha s_7 + \beta c_7)c_5c_6 + \gamma c_5c_6 &= -c_4L_1 - L_2.
 \end{aligned} \tag{B.25}$$

θ_7 が既知の場合は、(B.25)における未知数は θ_5 と θ_6 だけとなる。2番目の等式より、2組の θ_6 が求められる。

$$\theta_6 = \arctan 2(-\gamma, \alpha s_7 + \beta c_7) \pm \frac{\pi}{2} \tag{B.26}$$

θ_6 の値を(B.25)の最初と最後の式に代入することによって、2組の θ_5 が求められる。

$$\begin{aligned}
 \theta_5 &= \arctan 2(ad - bc, ac + bd) \\
 ac_5 - bs_5 &= c \\
 as_5 + bc_5 &= d \\
 a &= (\alpha s_7 + \beta c_7)s_6 + \gamma c_6 \\
 b &= (\alpha c_7 - \beta s_7) \\
 c &= -c_4L_1 - L_2 \\
 d &= s_4L_1.
 \end{aligned} \tag{B.27}$$

- θ_5 を固定した場合

θ_5 を固定した場合には、(B.23)は以下のように変形される。

$$\mathbf{A}_6\mathbf{A}_7(\mathbf{A}_{\text{wrist}})^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = (\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5)^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{B.28})$$

これは、以下のように展開される。

$$\begin{aligned} \alpha c_7 - \beta s_7 &= c_5 s_4 L_1 + s_5 (c_4 L_1 + L_2) \\ \alpha c_6 s_7 + \beta c_6 c_7 - \gamma s_6 &= 0 \\ \alpha s_6 s_7 + \beta s_6 c_7 + \gamma c_6 &= s_5 s_4 L_1 + c_5 (-L_2 - c_4 L_1). \end{aligned} \quad (\text{B.29})$$

(B.29)の最初の式より、 θ_7 に対して2組の解が得られる。

$$\begin{aligned} \theta_7 &= \arctan 2(-\beta, \alpha) \pm \arctan 2\left(\sqrt{\alpha^2 + \beta^2 - c^2}, c^2\right) \\ c &= c_5 s_4 L_1 + s_5 (c_4 L_1 + L_2). \end{aligned} \quad (\text{B.30})$$

それぞれの θ_7 について、 θ_6 は以下のようになる。

$$\begin{aligned} \theta_6 &= \arctan 2(ad, \gamma d) \\ a &= \alpha s_7 + \beta c_7 \\ d &= s_5 s_4 L_1 + c_5 (-L_2 - c_4 L_1). \end{aligned} \quad (\text{B.31})$$

- θ_6 を固定した場合

θ_6 を固定した場合には、(B.29)における未知数は θ_5 と θ_7 となる。2番目の式より、 θ_7 の2組の解が求められる。

$$\theta_7 = \arctan 2(\beta c_6, \alpha c_6) \pm \arctan 2\left(\sqrt{(\alpha c_6)^2 + (\beta c_6)^2 - (\gamma s_6)^2}, \gamma s_6\right) \quad (\text{B.32})$$

そして、 θ_5 は、以下の式により求められる。

$$\begin{aligned} \theta_5 &= \arctan 2(ad - bc, ac + bd) \\ a &= s_4 L_1 \\ b &= -c_4 L_1 - L_2 \\ c &= \alpha c_7 - \beta s_7 \\ d &= \alpha s_6 s_7 + \beta s_6 c_7 + \gamma c_6. \end{aligned} \quad (\text{B.33})$$

- 肩の姿勢の計算

次に、手首の関節の角度が決まると、 θ_1 、 θ_2 、 θ_3 は $A_{\text{wrist}}(A_4A_5A_6A_7)^{-1}$ の回転成分から求めることができる。(B.8) で、既知の項を右辺に移項して次のように変形する。

$$A_1A_2A_3 = A_{\text{wrist}}(A_4A_5A_6A_7)^{-1} \quad (\text{B.34})$$

右辺の回転成分を r_{ij} で表すと、以下のようになる。

$$\begin{bmatrix} c_1c_3 - s_1c_2s_3 & -c_1s_3 - s_1c_2c_3 & s_1s_2 \\ s_1c_3 + c_1c_2s_3 & -s_1s_3 + c_1c_2c_3 & -c_1s_2 \\ s_2s_3 & s_2c_3 & c_2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (\text{B.35})$$

これより、2組の θ_1 、 θ_2 、 θ_3 が求められる。

$$\begin{aligned} \theta_2 &= \pm \arccos(r_{33}) \\ \theta_1 &= \begin{cases} s_2 \neq 0 & \arctan 2 \left(\frac{r_{13}}{s_2}, \frac{-r_{23}}{s_2} \right) \\ s_2 = 0 & \arctan 2 (r_{21}, r_{11}) \end{cases} \\ \theta_3 &= \begin{cases} s_2 \neq 0 & \arctan 2 \left(\frac{r_{31}}{s_2}, \frac{-r_{32}}{s_2} \right) \\ s_2 = 0 & 0 \end{cases} \end{aligned} \quad (\text{B.36})$$

以上より、肩の関節変数の1つを固定した場合と同様、手首の関節変数の1つを固定した場合も、(B.8) に対して最大4組の独立した解が存在する。

B.1.3 適切な解の選択

以上で説明した通り、肩と手首のいずれかの関節変数の1つを固定した場合に、(B.8) に対して最大4組の独立した解が存在する。さらに、固定する関節が肩と手首に3つずつ、合計6つあるので、この手法では最大24組の解が存在することになる。この24組の解の中から適切な解を選択するのであるが、本アルゴリズムでは、腕の動作をスムーズに合成するために、腕の姿勢の変化を (B.37) により算出し、その値が最も小さい解の組を (B.8) の解として用いている。ただし、 θ_i^0 は直前の関節 i における関節変数の値を示す。

$$\sum_{i=1}^7 (\theta_i - \theta_i^0)^2 \quad (\text{B.37})$$

以上の手法により、手先の位置から腕の姿勢が計算される。よって、手先の位置のフレーム毎の変化を動作データとして保持していれば、各フレームにおける腕の姿勢が計算され、腕の動作をアニメーションとして合成することができる。

参 考 文 献

- [1] Deepak Tolani and Norman I. Badler : Real-Time Inverse Kinematics of Human Arm, Presence, **Vol.5**, No.4, pp.393-401(1996).