

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
積み木メタファを用いた
題目： インタラクティブ仮想空間構築
インタフェースの開発

指導教官： 吉川 榮和 教授

氏名： 遠藤 啓介

提出年月日： 平成15年2月5日（水）

目次

第 1 章 序論	1
第 2 章 研究の背景と目的	3
2.1 Virtual Reality の現状とその応用分野	3
2.2 VR 技術を用いた訓練環境の構築	5
2.3 仮想空間を構築する手法の従来研究とその問題点	7
2.4 本研究の目的と意義	11
第 3 章 積み木メタファに基づく仮想空間の構築手法の提案	14
3.1 構築可能とする訓練環境	14
3.2 仮想空間内における仮想物体の挙動計算手法	15
3.2.1 Physically-based 手法	15
3.2.2 Rule-based 手法	17
3.2.3 本研究におけるシミュレーション手法	18
3.3 ActCube を用いた仮想空間の構築手法	21
3.3.1 仮想空間の構築に必要な設定	21
3.3.2 ActCube の概要	25
3.3.3 ActCube の機能	28
3.3.4 ActCube の種類	33
3.3.5 ActCube を用いた仮想空間の定義例	35
3.3.6 ActCube 手法の利点	37
3.3.7 ActCube 手法の欠点	40
第 4 章 仮想空間管理シミュレーションシステム (Act System) の開発	41
4.1 Act System が具備すべき機能	41
4.2 Act System の概要	44
4.2.1 ソフトウェア構成	44
4.2.2 ハードウェア構成	47

4.2.3	インタフェースの構成要素	50
4.3	3次元環境下での直接操作について	54
4.4	仮想空間の構築について	57
4.4.1	仮想空間の構築者が実行可能な操作	57
4.4.2	仮想機器構築の手順	61
4.5	仮想空間のシミュレーションについて	63
4.5.1	シミュレーション中の訓練生が実行可能な操作	63
4.5.2	シミュレーションの流れ	64
第 5 章	仮想空間の構築例及び Act System の評価	66
5.1	仮想空間の構築例	66
5.1.1	開発した ActCube	66
5.1.2	Act System を用いて構築した仮想空間	72
5.2	Act System を評価するための被験者実験	77
5.2.1	Act System の操作性を評価するための被験者実験	77
5.2.2	ActCube 手法を評価するための被験者実験	81
5.3	まとめと今後の課題	84
第 6 章	結論	88
	謝 辞	90
	参 考 文 献	91

目 次

2.1	VR 技術の訓練への応用	4
2.2	仮想空間とユーザの相互作用	7
2.3	インタラクティブ仮想空間の構築に必要な設定作業	8
2.4	仮想空間の構築支援システムの画面	9
2.5	モデリング作業の変遷	10
2.6	積み木メタファによる仮想空間の構築イメージ図	11
3.1	Physically-based simulation	16
3.2	ポリゴンによる球の表現	16
3.3	Rule-based simulation	17
3.4	衝突により実現可能な事象	19
3.5	衝突が多発する場面	20
3.6	衝突では実現困難な事象	20
3.7	本研究でのシミュレーション手法の特徴	21
3.8	プリンタのシミュレーション	22
3.9	ねじとボルトにおけるシミュレーション	22
3.10	ねじにおける拘束関係の設定箇所	24
3.11	プリンタにおける拘束関係	24
3.12	プリンタにおける設定	25
3.13	ActCube 基本外観	26
3.14	ねじ ActCube と蝶番 ActCube	26
3.15	ActCube のボルトへの付着	29
3.16	ActCube の大きさの調節	29
3.17	ActCube をねじに付着させる場合	29
3.18	凹属性と凸属性	30
3.19	ActCube の結合条件	31
3.20	ActCube の機能発動	32
3.21	シミュレーション中の ActCube の外観	33

3.22	ActCube を含んだシミュレーション	34
3.23	ActCube の種類	34
3.24	プリンタへの ActCube 付着	35
3.25	ActCube によるプリンタのシミュレーション	36
3.26	ボルトに締まっているナットをレンチで回す作業の定義例	36
3.27	ボルトにナットが2つ締まっている時の関係	37
3.28	ActCube の開発	39
4.1	Act System のシステム構成	44
4.2	モーションキャプチャシステム装置構成	48
4.3	手の開閉情報を取得するためのボタンデバイス	49
4.4	ボタンとセンサ	49
4.5	Act System のインタフェース	51
4.6	キャリブレーション時のユーザの姿勢	55
4.7	接触によるハイライト表示	56
4.8	3次元オブジェクトの移動	58
4.9	ActCube を3次元形状に付着させる作業	59
4.10	ActCube を3次元形状から引き離す作業	59
4.11	ActCube の拡大縮小	60
4.12	ActCube をひねる作業	61
4.13	仮想機器の構築手順	62
4.14	シミュレーションフローチャート	65
5.1	平行移動 ActCube の外観	66
5.2	平行移動凸 ActCube における挙動制御方法	67
5.3	平行移動凹 ActCube における挙動制御方法	67
5.4	蝶番 ActCube の外観	69
5.5	蝶番 ActCube における挙動制御方法	69
5.6	ねじ ActCube の外観	70
5.7	ねじ ActCube における挙動制御方法	71
5.8	把持 ActCube の外観	71
5.9	スウィング式逆止弁の構造	72
5.10	スウィング式逆止弁構築の様子	74

5.11 スウィング式逆止弁シミュレーションの様子	74
5.12 被験者実験のプリンタ完成図	78
5.13 3次元デバイスを用いた被験者実験の様子	79
5.14 ActCube 付着可能性の視覚的補助	82
5.15 レールを用いた平行移動の設定	84

表目次

3.1	実現する物理現象	15
3.2	拘束関係と ActCube の関係	27
4.1	ULTRATRAK システムの基本性能	48
5.1	スウィング式逆止弁の分解の作業手順 (一部)	73
5.2	スウィング式逆止弁のために必要な 3 次元形状	73
5.3	スウィング式逆止弁のために必要な ActCube	73
5.4	スウィング式逆止弁の構築手順	73
5.5	仮想空間構築支援システムと Act System の対比	75
5.6	3 次元デバイスを用いた被験者実験のアンケート結果	80
5.7	キーボード・マウスを用いた被験者実験のアンケート結果	83

第 1 章 序論

「自宅にいながら、宇宙旅行が体験できるようになるかもしれない。」「今の薄っぺらいデスクトップ環境が、3次元の部屋のような動的な空間に置き換わる日が来るかもしれない。」

これらの願望を実現できる可能性がある技術が Virtual Reality(以下、VR) 技術である。ディスプレイの画面に映し出されるコンピュータ・グラフィックスが描き出す VR の世界は無限の可能性を秘めていると VR に携わる研究者達は信じ、1990 年代に入ってから、各地で研究活動が活発に行われた。1990 年代後半に入ると、Web 環境上で 3D によるグラフィックス環境を構築するために VRML が策定され、仮想 3D 世界でチャット等ができるバーチャルコミュニティのサービスも始まった。これらの技術が登場した当時は、コンピュータのインタフェースの次のパラダイムは VR によって牽引され、VR は徐々に人々の生活の中に浸透していくものだと期待が持たれていた。

しかし、インターネット環境の貧弱さと個人用 PC の処理能力では本格的な 3D 空間を再現することが難かったため、期待に反し、利用者は伸び悩み、VR 技術は一般の人々の間に普及することは無く失速していき、現在では、一部の専門分野での使用に留まっている。

しかし、現在、PC の 3D 処理性能は大きく向上した上、ブロードバンド環境の普及もあり、かつて VR 技術の浸透を阻害していたハードルが急速に取り除かれていっており、再び VR 技術が人々の間に浸透する期待が高まっている。その中で、VR 技術が本当に、人々を惹きつけるものとするためには、人々が用意された環境を楽しむだけでなく、自分たちがその世界を構築でき、自分の世界を創り出す楽しみを体験できるようにする必要があるだろう。しかし、仮想空間内の物体が操作できる仮想空間を構築するためには、仮想物体の動きや、仮想物体間および仮想物体と人間の間の相互作用を定義する必要がある。そのため、現状では、仮想空間は、プログラミング等の専門知識を有する人しか構築できない。誰もが、仮想世界の構築に参画できるようになるには、専門知識を要求せずに構築できる手法が必要である。

一方、近年、3次元形状のモデリング作業を、実世界での物体に関する知識を活用して 3次元環境下での直接操作により行うことができるシステムが実現されている。このシステムは、3次元形状を直接 3次元的に操作でき、日常の実空間での作業感覚に近

い直感的な感覚で作業が行えるといった利点を持つ。仮想空間の構築作業を3次元環境下で実現可能にすることにより、仮想空間の構築作業も同様の効果が期待でき、プログラミング等の専門知識を排除できる可能性がある。

以上により、本研究では、誰もが仮想空間を構築できるようになることを最終目標とし、まず、初期段階として、構築する仮想空間の対象を訓練環境に限定し、専門知識を持たない人でも3次元環境での直接操作だけで訓練環境を構築できる手法を提案し、その手法に従って訓練環境を構築・シミュレーションすることができる仮想空間管理シミュレーション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、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。3章では、3次元環境下での直接操作だけで積み木を組み立てる感覚で仮想空間を構築できる手法を提案する。そして、提案した手法で仮想空間を構築することによる利点をまとめる。4章では、提案した手法に従って仮想空間を構築し、シミュレーションするために開発したシステムについて説明する。5章では、提案した手法に従って実際に仮想空間を構築し、提案する手法の効果を評価し、今後の展望について述べる。最後に、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、Virtual Reality(以下、VR)技術の現状について展望する。次いで、機器保守の訓練システム等への VR 技術の応用の可能性について述べる。そして、様々な分野へ VR 技術を応用するために必要な要素技術の中で、特に仮想空間の構築手法に注目し、仮想空間の構築手法についての研究開発の現状と問題点を展望し、最後に、本研究の目的を述べる。

2.1 Virtual Reality の現状とその応用分野

VR とは、「計算機によって作り出された合成情報を人間の感覚器官に直接提示することにより、仮想世界の能動的な疑似体験を可能にする」ための技術である^{[1][2]}。人の身体(頭や手)の動きを計測し、その動きに応じて、コンピュータグラフィックスにフィードバックをかけることにより、人は現実のイメージに近い仮想的な世界を体験できる。「Virtual Reality」という単語は1980年代後半、VPL社のLanierにより生み出された言葉であり、この新語が発表されたことを契機に、各地で活発に研究開発が進められるようになり、今日では広く社会にも認知される技術となっている。

その VR 技術は、主に次の 3 つの要素技術で構成される^[3]。

表示システム 仮想世界を臨場感あふれる方法で人間の主に視覚、聴覚、触覚へ表示するデバイス

計測システム 人間が仮想世界と相互作用するために、人間の動作をリアルタイムで計測するセンサ

情報処理システム 仮想世界そのものを記述すると共に、計測システムからの入力に基づいて仮想世界の中で起こる様々な事象をシミュレートし、その結果をもっともらしい形で表現して表示システムに出力するソフトウェア

図 2.1 に示すような仮想世界を人が体験する時は、まず、人間の動きをより自然な形でシステムに伝達するために、データグローブ、3次元マウス、ジョイスティック等の計測システムを用いて人間の動きを計測しコンピュータに入力する。これらの入力デ

バイスから得られたデータを、情報処理システムが、仮想世界の内部状況に合わせて、仮想の手の位置や姿勢情報等に変換し、仮想世界内部の仮想物体の位置や姿勢などを更新する。この結果は、コンピュータグラフィックスとして、ヘッドマウントディスプレイや立体視ディスプレイを通じて人の視覚に提示される。また、フォースフィードバックを返すことができる PHANToM 等の触覚への表示システムを用いることにより、仮想物体を実際に触っている感覚を人に提示することも可能である。

VR 技術は、表示・計測システムといった機械と人とのインタフェースの開発が中心に進められてきたが、1990 年代後半に、Web 環境上に仮想世界を構築するための VRML(Virtual Reality Modeling Language)^[4] が登場し、現在では、その VRML の次世代規格である X3D^[5] の策定が進められている。このような技術が提案されていることから分かるように、将来的に、現在のホームページに近い感覚で人々が仮想の空間を公開する日が来ることが期待されている。そのためには、誰もが仮想の世界を手軽に構築できる技術が必要とされる。

また、近年の計算機能力の飛躍的な向上やインタフェース技術の発展といった仮想世界を構成する要素技術の充実に従い、VR 技術は様々な分野に応用されている^[6]。設計・製造分野への応用として、仮想空間内に住空間を設計し、性能を疑似体験できるシステム、遠隔地からあたかもロボットを自分の分身のごとく操作できるロボット遠隔操作システムや、外科手術のシミュレーションを仮想空間内で行うなどの医療分野への応用がある。

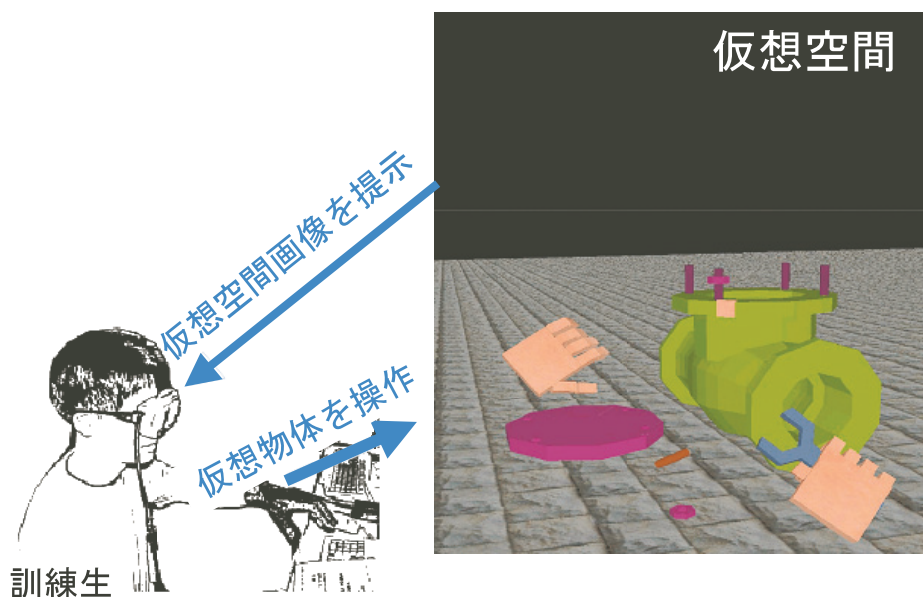


図 2.1: VR 技術の訓練への応用

2.2 VR技術を用いた訓練環境の構築

2.1節で述べた応用例の他に、VR技術を用いて機器保守の訓練環境を構築する試みも行われている。例えば、1990年代中頃には、NASAで宇宙空間内で宇宙望遠鏡の修理作業を行う訓練環境が構築され^[7]、最近では、建設機器、林業機械訓練のシミュレータ等が開発されている^[6]。

VR技術を用いて構築した訓練環境は、実機を用いた訓練環境に比べて次のような利点がある。

1. 事故が起こる可能性が低く、安全である
2. 現実世界では実現困難な事柄が模擬できる
3. 実機を用意する必要がなく、費用が安く済む可能性がある
4. 訓練対象や作業環境を自由に設定できる

特に、本研究室では、工学プラントの機器保守作業の訓練環境をVR技術により構築する試みを積極的に行っている。工学プラントが益々大規模複雑化するにあたり、機器保守作業の重要性が増大しており、原子力発電プラント等の大規模エネルギー機器の保守作業や運転作業を体験できる環境をVR技術を用いて構築することができれば、訓練機会の増加、エネルギー利用技術の向上、エネルギー機器取扱上の安全性・信頼性の向上を期待できる。

VR技術を用いて工学プラントの訓練環境を構築する研究としては、南雲らによる原子炉ドライウェルにおけるウォークスルー環境^[8]、阿部らによる機器の修復支援システム^[9]、新井らによる変電所保守員向け体感型シミュレータ^[10]、吉川らによるスイング式逆止弁の分解・組み立て作業の訓練システム^[11]等を挙げることができる。

これらの先行研究より、VR技術を用いた機器作業の訓練環境は、作業の手順や機器の構造などを覚えることには効果的であることが報告されている。

しかし、VR技術により構築された訓練環境は、現在盛んに実用化されるには至っていない。その原因としては、表示・計測のためのデバイス技術がまだ未熟であるというハードウェア的側面の問題と、構築された訓練環境で訓練生の操作の選択肢が少なく、何度訓練を行っても訓練生は同じ体験しかできないため、実機を用いた訓練に比べて低い訓練効果しか挙げられず、また、VR技術を用いた訓練環境を構築する労力が大きく、多様な訓練環境のラインナップを揃えることができない、といったソフトウェア的側面の問題が挙げられる。

本研究では、特に、ソフトウェア的側面の問題に着目し、従来の仮想空間の構築における問題点を克服できる手法の確立を目指す。

そこで、まず、高い訓練効果を挙げることができる訓練環境を実現するために必要なユーザインタフェースのレベルを考察する。機器保修の訓練環境をそのユーザインタフェースにより、段階的に分類すると以下ようになる^[12]。そのイメージ図を図 2.2 に示す。

分類 1 ウォークスルー式訓練システム

ウォークスルー式訓練システムは、訓練生がマウスやキーボード、ジョイスティック等を用いて、仮想空間内を歩き回ることにより、訓練対象機器の構造や各種名称を学習する訓練システムである。訓練対象機器の 3 次元モデルを作成し、訓練生の視点を変更する機能を構築することにより比較的容易に実現できるが、作業手順や機器の操作方法等の学習に用いることはできない。

分類 2 メニュー選択式訓練システム

メニュー選択式訓練システムは、ウォークスルー式訓練システムに、仮想空間内に配置された物体間の物理的拘束関係を論理的にモデル化してシミュレーションする機能を付加したもので、訓練生はマウスやキーボードで訓練対象機器に対する作業の種類を選択することで、機器の分解や組立作業を行う。主に作業手順の学習に用いることができる。

分類 3 ジェスチャ式訓練システム

ジェスチャ式訓練システムは、メニュー選択式訓練システムで、マウス等を用いて作業の種類を選択する代わりに、データグローブや 3 次元マウス等を用いて、実際の作業に近いジェスチャで機器を操作して作業を進める訓練システムである。仮想物体間の物理的拘束関係を実世界の物理法則に近い形でシミュレーションし、結果を視覚的に表現する必要がある。作業手順に加え、具体的な機器の操作方法も学習できる。

以上に述べた訓練システムの分類の内、分類 1 の訓練システムは、その構築のために必要な労力が他の分類と比べて最も少なく、技術的にも最も容易に構築できるが、訓練生自らが作業を体験できないために、訓練が受け身となり、訓練効率が最も低い訓練システムである。

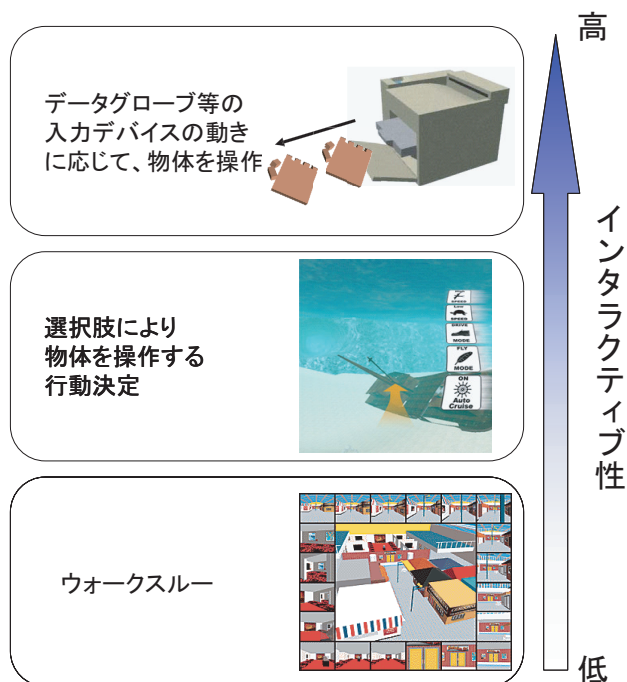


図 2.2: 仮想空間とユーザの相互作用

一方、分類 3 の訓練システムは、最も構築の労力が大きく、技術的にも最も構築が困難であるが、実機を用いた訓練に最も訓練環境が類似しており、訓練効率が最も高い訓練システムである。さらに、そのジェスチャ式訓練システムでは、訓練生の動作を制限せずに力覚を提示可能なデバイスが実装されれば、現実に近い環境を提供でき、仮想空間内で機器保守作業の手順といった知識を詰め込むだけでなく、作業の技能そのものを習得できるようになる可能性がある。

以上の点から、VR 技術を用いて構築する訓練環境としては、ジェスチャ式訓練システムが望ましい。しかし、ジェスチャ式訓練システムでは、構築の労力が大きいという問題があるため、容易に構築するための手法の登場が望まれる。

以下、本論文では、このジェスチャ式訓練システムのような、ジェスチャで仮想物体を自由に操作することができる仮想空間を「インタラクティブ仮想空間」と定義する。

2.3 仮想空間を構築する手法の従来研究とその問題点

2.2 節で述べたようにジェスチャ式訓練システムは訓練効率が高いにも関わらず、ウォークスルー式の訓練システムに比べて実用化例は少ない。

ウォークスルー式訓練システムは、訓練の対象となる 3 次元形状やテクスチャを作

成し、そのデータを基にした仮想物体を仮想空間内に配置する作業を行えば、仮想空間が構築可能であることに対し、ジェスチャ式訓練システムを構築する際には、3次元形状等の作成の他に、訓練生の入力に対しての仮想物体の反応や仮想物体同士が相互作用しあう時の挙動等を定義する必要がある。つまり、2.1節で述べた情報処理システムの担う役割が大きくなるため、その規模が大きくなってしまい、構築作業に非常に労力がかかってしまう。

インタラクティブ仮想空間を構築するために必要な主な設定をまとめると、図2.3に示すようになり、この中でも、仮想物体の挙動設定作業は、仮想空間の構築特有の作業であり、インタラクティブ仮想空間を構築する際には、この挙動設定作業をいかに簡単に行えるか、が重要なポイントとなる。

以上のような状況から、VRを使用した訓練環境は、VRに精通した専門家によるプログラミングで構築されていることが一般的である。この事態は、以下のような問題を生み、VR技術を用いた訓練環境の普及を妨げている。

1. 専門家でなければ構築できない
2. 大規模な仮想空間の構築が難しい
3. 新たに訓練環境を構築する際に、プログラムを修正または作り直す必要があるため、その労力が大きい

特に、1.の問題を解決し、逆に、専門知識を持たない人でも、簡単に仮想空間の構築が可能ならば、現場の指導者が実際に訓練環境を構築できるようになり、訓練の現場の意見を訓練システムに直接反映させることができる。さらに、仮想空間を構築する人の絶対数の増加が期待され、その結果、多様な仮想空間が生み出され、体験可能な仮想空間のリストが充実することにもつながる。

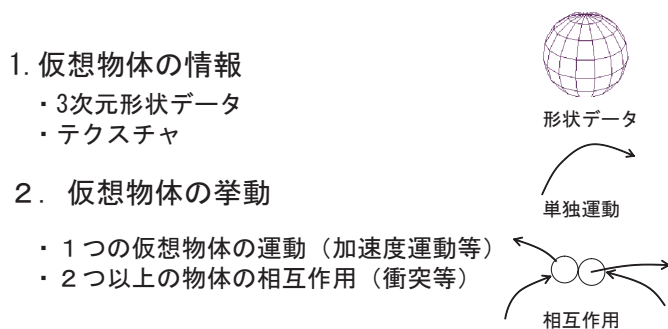


図 2.3: インタラクティブ仮想空間の構築に必要な設定作業

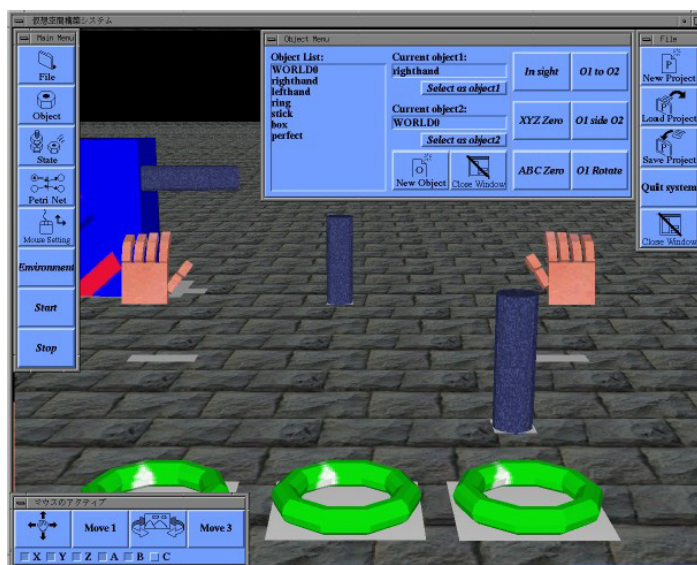


図 2.4: 仮想空間の構築支援システムの画面

そこで、上記の問題を解決し、VRソフトウェアの生産性を向上させるためには、次の方策が考えられる。

方策1 プログラミングの専門知識を必要とせずに、仮想空間を構築できる

方策2 複雑な仮想空間を構築する際に、対象を分割して個々に構築可能にし、データの再利用性を高める

方策2を実現するための研究としては、オブジェクト指向に基づく仮想空間の構築手法^[13]が提案され、大規模仮想空間構築の方針を示されたが、仮想空間の構築のためには、プログラミング等の専門知識を要求し初心者には構築が困難であった。

方策1を実現するための研究としては、図2.4に示すようなGraphical User Interface(以下、GUI)を介して仮想空間を構築できる仮想空間構築支援システム^[14]の開発があり、労力の軽減につながっている。しかし、多様な仮想空間を初心者でも構築できるためには、GUI環境下でのインタフェースが複雑すぎ、仮想物体間の相互作用をモデル化する作業に特別な知識を要した。

また、商用の代表的な仮想空間構築支援システムとして、レクサーリサーチ社の「AsseyWORK^[15]」、SENSE8社の「World Tool Kit^[16]」、米国EON社の「EON^[17]」等が発売されており、これらのソフトウェアを用いて、訓練環境を構築できる。これらのソフトウェアは構築のためのGUI環境が準備されており、プログラミングを行わなくても仮想空間を構築できる。また、仮想空間の構築で開発のネックとなりやすい3

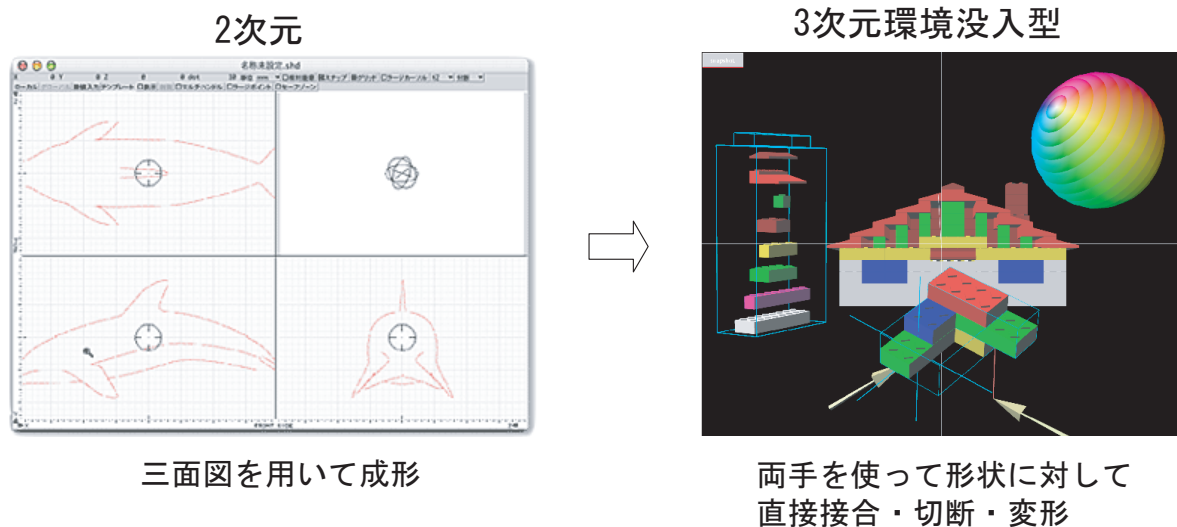


図 2.5: モデリング作業の変遷

次元空間描画エンジンを搭載していることや、VR用のデータグローブ等のデバイスを容易にシステムに組み込めること等から、全てプログラミングで仮想空間を構築する場合に比べて、少ない労力、少ない時間で仮想空間を構築することが可能である。しかし、仮想空間の構築者は、その多様なダイアログの意味を理解する必要性があり、また、インタラクティブ仮想空間を構築するために必要な仮想物体間の相互作用や仮想物体の挙動の定義において、スクリプトの記述が必要になる等、やはり専門知識を有していない者では仮想空間の構築が難しく、結局、既存のソフトウェアでは、仮想空間の構築に専門知識を要求されるのが現状である。

そこで、本研究では、GUI環境よりもさらに誰でも理解しやすい形で仮想空間を構築できる手法を開発するために、同様に専門知識を有していないと製作が困難な事例である3次元形状のモデリング作業に注目した。モデリング作業は従来、図2.5に示すように3面図を用いた作成手法が一般的であり、その操作に習熟していない人では形状の構築が難しかったが、近年、3次元環境に没入して粘土や積み木をいじくる感覚で直接3次元形状を切断、変形させてモデリング作業を行える3次元モデラ^{[18][19]}が開発されている。この3次元モデラのモデリング作業を行う3次元環境下には、ダイアログなどなく、3次元環境下に存在する3次元オブジェクトの操作だけでモデリング作業を完了できる。

3次元環境下での作業は、人が日常生活で既に習得しているメタファで物体を操作するため、新たに情報操作のためのインターフェースの習得が必要でないという利点がある。さらに、3次元形状を扱った作業では、その3次元形状への操作がすぐに画面に反

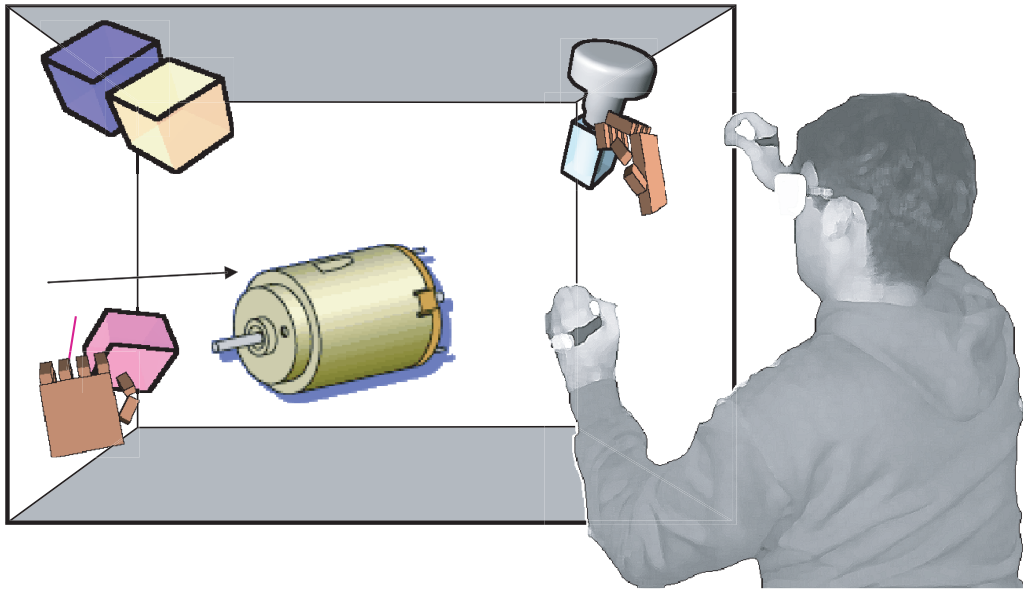


図 2.6: 積み木メタファによる仮想空間の構築イメージ図

映されるため、より直感的に作業を進められる。

そこで、本研究では、この3次元環境への没入というアイデアを、インタラクティブ仮想空間の構築に適用する。すなわち、インタラクティブ仮想空間の構築作業において、専門知識が必要になる仮想物体の挙動の設定作業を3次元環境下で積み木を組み立てる感覚で行える手法を考案する。

2.4 本研究の目的と意義

本研究の目的

本研究では、2.3節で述べたように、3次元環境内で3次元モデラのように3次元オブジェクトを操作するだけで、インタラクティブ仮想空間を構築できる手法を提案し、その手法に基づいて実際に仮想空間を構築可能であるシステムを試作した後、提案する手法の効果を評価することを目的とする。

具体的には、図2.6に示すように3次元環境内で3次元形状に、他の3次元形状と相互作用を行う性質を付加する ActCube と呼ぶ3次元オブジェクトをインタラクティブ仮想空間の構築のために新たに導入し、3次元環境下でこの ActCube を積み木のように操作することにより、仮想空間の構築において特に専門知識が要される仮想物体の挙動を設定する作業が行える手法を提案する。そして、その手法を基に仮想空間を構築、管理できる試作システムとして Act System を開発する。

本研究の意義

3次元環境内で仮想空間を構築することにより、情報操作のための知識が必要なくなり、コンピュータの操作に習熟していない人でもインタラクティブ仮想空間を構築できることが期待される。

また、通常、仮想空間の構築時には、構築作業とその構築した仮想空間のテスト体験を交互に繰り返し、作業を進める。仮想空間のテスト体験は3次元下で実行されるため、従来のプログラミング作業などで構築していたときは、3次元デバイスの着脱や3次元アプリケーションの起動などインタフェース間の移行が煩雑であったが、本研究の手法では、3次元下で構築作業が行われるため、テスト体験と構築作業をシームレスに移行でき組み立てた効果がすぐに確かめられる利点がある。

さらに、積み木を組み立てる感覚で、仮想空間の構築作業が実現できれば、誰でもが気軽に遊び感覚で仮想世界を構築できるようになり、そのエンターテインメント効果により、多くの人々がVRに興味を向けることを期待できる。

本研究で開発するシステムは、訓練環境の構築だけでなく、様々な分野に応用可能である。まず、製品の仮想モックアップに適用可能である。仮想空間内で機器を組み立てて、実機によるモックアップを行わずに、問題の洗い出しができる。その際、操作可能な機器が構築されれば、操作性の観点からも製品の評価ができる。また、製品の3次元カタログの開発にも応用できる。e-commerceにおける電子商店などで購入前にその製品の3次元カタログを参照することができれば、ユーザは商品イメージをつかみやすくなる。インタラクティブな仮想物体であるならば、更にユーザの興味を惹きつけやすくなる。

関連研究と本研究の特徴

3次元環境下で仮想空間を構築するという着想は、2.3節で述べたように、3次元モデラ、特に、VLEGO^[18]に強い影響を受けている。

一方、本研究の関連研究としては、仮想空間内で特殊な働きを果たす3次元オブジェクトを3D Widgetと呼び、その3D Widgetを組み合わせて、スイッチやライトの実現、または、ピアノ演奏を実現できるような仮想空間を構築する研究がある^{[20][21]}。本研究で導入するActCubeもこの3D Widgetであると言える。従来の研究では、その3D Widgetの配置等はプログラミングで実現することが一般的であり、本研究のように、仮想空間内で直接その3D Widgetを組み合わせるだけで、仮想世界を構築できる研究はほとんどない。

機器を操作可能な仮想空間を構築する研究開発事例としては、既に述べた商用ソフトウェアの他に、技能教育システム^[22]の開発を行った研究がある。この研究では、機器が操作される際の動線を記述しておくことにより、シミュレーション時にその動線に従った剛体の組立を実現している。

本研究で導入する ActCube のような固有の機能を持つ 3 次元オブジェクトを画面上での直接操作によって組合せて機能を合成するという考え方は、IntelligentBox^[23]に近い。本手法との違いは、IntelligentBox は、インタラクティブな仮想環境を構築することを目的としていないこと、3 次元オブジェクト間の親子関係等をダイアログ画面で設定する必要があることが挙げられる。

また、本研究のように、物理的拘束関係を仮想空間のシミュレーションに積極的に取り入れている研究としては、仮想空間の構築時に、従来無視されがちである CAD データ内に含まれている物理的拘束関係に関する記述を活用し、仮想空間を構築する研究^{[24][25]}がある。CAD データを仮想空間シミュレーションシステムに取り込む際、その物理的拘束関係に関する記述に基づいたデータ構造を作成し、物理的拘束関係に関するシミュレーションを可能にしている。しかし、この手法では、設定を変更したい場合は、CAD の知識を要求するため、専門知識が必要になる。

本研究では、3 次元環境下で、3 次元オブジェクトを組み合わせるだけで、仮想物体の挙動が設定可能になり、また、3 次元環境下の作業だけで、全ての構築作業を完了できることが最大の特徴であり、他の仮想空間構築に関する事例とは、この点で大きく異なっている。

第 3 章 積み木メタファに基づく仮想空間の構築 手法の提案

本章ではまず、本研究で構築を想定している訓練環境の機能について述べる。次に、その訓練環境内における仮想物体の挙動計算手法について説明する。その後、その挙動計算手法と従来手法における仮想空間の構築の困難性を考慮して、3次元環境に没入した状態で積み木を組み立てるような感覚で仮想空間を構築可能にする手法を提案する。

3.1 構築可能とする訓練環境

本研究が対象とする訓練作業は、プラント等における機器の保守作業とし、機器の分解組立作業を主な訓練の実施目的とする。分解組立作業での主な訓練生の動作や物体同士の物理的拘束関係等として、訓練生が手に物体を把持する、訓練生がドライバーでねじをはずす、訓練生が手で蓋を回転させて開ける、訓練生が治具を使用しないと蓋を開けられない、モータが回転を続ける、等がある。

それらの操作が可能なインタラクティブな訓練環境を構築するためには、仮想空間が以下の仕様を満たしている必要がある。

- 仮想空間内での作業においては、例えば人がスパナを持つというような人と機器との相互作用を実現するだけでなく、人が持っているスパナでナットを外す等の機器と機器との相互作用も実現する。また、ねじが他の部品を固定するような、ある部品が他の部品を拘束する場面も再現されなければならない。
- モータの回転や自由落下等、訓練生の操作に依存しない形での仮想物体の自律的な運動も存在し、これらの事象も再現されなければならない。
- 実世界と同様の物理法則には従うが、物理法則を完全に反映した正確なシミュレーションである必要はなく、訓練生が違和感を覚えられないレベルに到達することができれば、十分機能を満たすことができると言える。また、VR 技術を用いた訓練環境での訓練時に、違和感を覚えさせないために、必要な条件として、ジェスチャ

表 3.1: 実現する物理現象

機器補修の訓練環境を実現する際に必要な物理現象	
本研究で実現する物理現象	本研究で実現しない物理現象
物体の衝突	物体の変形
物体の接触	物体の分割
重力	電磁気など

入力からディスプレイへのフィードバックの間の遅延が小さい必要があり、この点には留意しなければならない。

- 対象とする仮想物体は、その形状が変化しない剛体と仮定し、仮想物体の形状変化や切断は起こらないものとする。表 3.1 に本研究で実現する物理現象を記載する。
- 普通、保修作業中に機器とインタラクションする身体パーツは、ほとんど訓練生の手であることから、機器とインタラクションする身体パーツは、訓練生の両手のみとする。
- 実世界と同様の手による操作を実現するために、訓練生の手は 3 軸方向の平行と回転の 6 自由度を持つ。また、手を開閉することにより、機器を把持できる。

3.2 仮想空間内における仮想物体の挙動計算手法

インタラクティブ仮想空間を構築する場合、シミュレーション方法が異なれば、構築のために設定する項目も変化する。

ここでは、まず、仮想物体の挙動を計算する一般的な 2 つのシミュレーション手法を説明し、その後、その 2 つの手法を照らし合わせ、3.1 節で述べた具備すべき機能を満たし、また、構築労力が少なくなる適切なシミュレーション手法を考案する。

3.2.1 Physically-based 手法

Physically-based 手法とは、現実の物理法則に基づき、厳密に仮想物体の挙動を計算する手法である [26][27][28]。本研究では、特に、他の物体との間の相互作用を衝突に限定し、仮想物体の挙動を計算する手法を採用している。Physically-based 手法は、自由落

下時の仮想物体の挙動だけでなく、衝突時における仮想物体の挙動の変化なども物理法則に従い計算する。仮想物体の衝突時の挙動変化は、仮想物体の重さなどの物理パラメータ、および、衝突時の仮想物体の速度と衝突点から一意に計算される。詳細を付録 A. に示す。シミュレーションの様子を図 3.1 に示す。

この手法では、仮想空間内での仮想物体の挙動を実世界に近づけることができるため、リアリティがあるシミュレーションが可能である。また、VR 空間構築者が入力する情報は、相互作用を全て衝突に限定することにより、仮想物体の物理パラメータ(形状、摩擦係数等)だけでよくなり、仮想空間構築者が入力する設定項目が少ないというメリットがある。

しかし、仮想物体は図 3.2 に示すように一般的にポリゴンと呼ばれる三角または四角平面の集合として表されており、例えば、ボルトにはまったナットを回転させる場合等を、仮想物体の形状情報等を用いて計算するには、正確な衝突情報を必要とするた

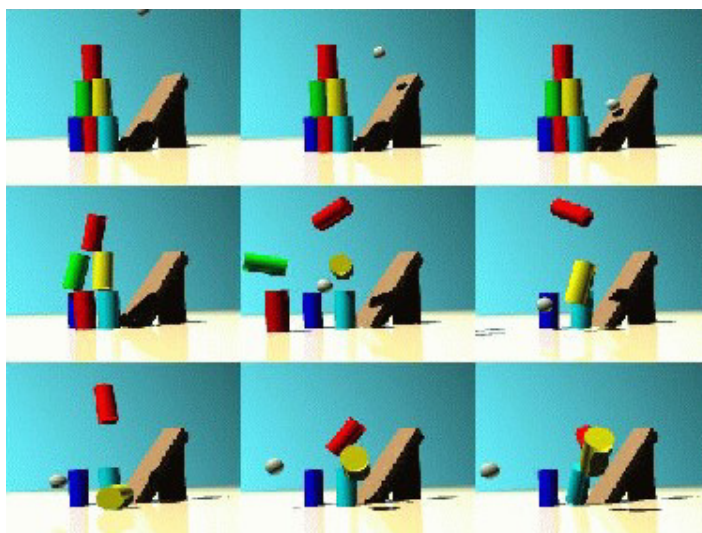


図 3.1: Physically-based simulation



球



ポリゴンによる球



図 3.2: ポリゴンによる球の表現

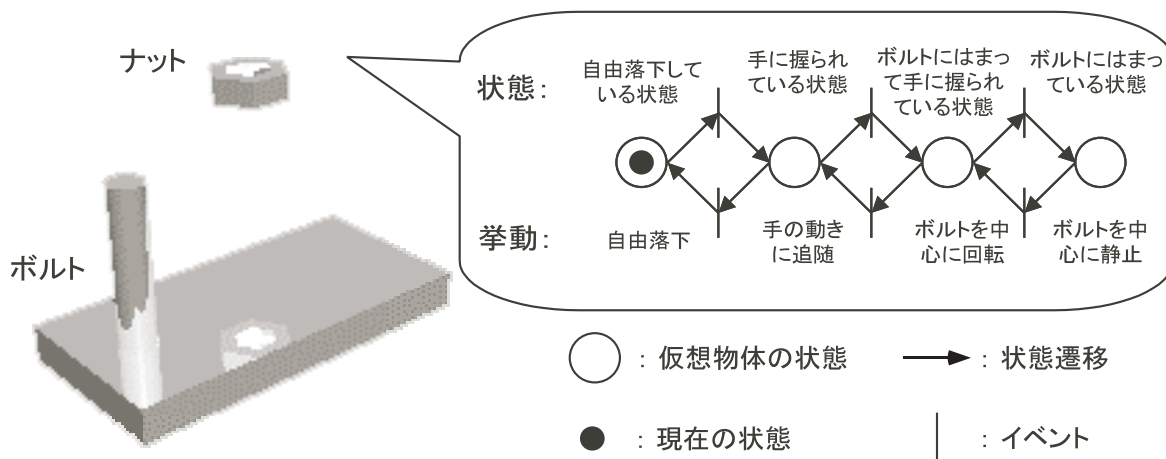


図 3.3: Rule-based simulation

め、精密な形状情報および非常に高精度の衝突判定アルゴリズム等が要求される。現状では、ボルトがナットにはまっている状態で、短時間に正確に衝突情報を返すことが可能なアルゴリズムは存在せず、ボルトをナットに締める状況をこの手法を用いてリアルタイムにシミュレーションすることはできない。リアルタイム性を求めない製品設計の分野などで、この手法を用いて製品の形状の稼動検証等が行われている。

3.2.2 Rule-based 手法

Rule-based 手法は、図 3.3 に示すような仮想物体が取りうる状態と、各状態における訓練生の入力動作に応じた仮想物体の挙動を、あらかじめデータベースとして定めておき、仮想空間のシミュレーションを行う際には、そのデータベースを用いて、仮想物体の挙動を合成する^[29]。例えば、「ナットを手で掴んでボルトに締める」作業を行える仮想空間を構築する場合は、図 3.3 に示す様に、ナットが取りうる状態として「自由落下している状態」「手に握られている状態」「ボルトにはまって手に握られている状態」「ボルトにはまっている状態」を定める。そしてナットが「自由落下している状態」の時には、ナットが他の物体に接触するまでは、訓練生の動作入力の内容に関わらず、常に自由落下する様に定め、「手に握られている状態」にある時には、訓練生の手を開く動作までは常に訓練生の手挙動にナットの挙動が追従する様に定める。

しかし、「ナットを手で掴んでボルトに締める」という単純な例では、仮想物体の状態を管理しその状態における挙動を実現する手法が有効であるが、この手法では構築が難しい訓練環境も数多く存在する。例えば「机の上に転がり落ちるナット」を仮想空間内で実現する場合、実空間と同様の挙動をさせるためには、落下速度や落下角度に

応じたナットの跳ね返りを考慮しなければならず、この手法で実現するためには、非常に多くの状態と各状態に応じたナットの挙動をあらかじめ定めなくてはならない。この様に、仮想物体が様々な挙動をとる可能性がある仮想空間を実現する際には、その全てを個別の状態として定義するのではなく、Physically-based 手法を用いて仮想物体の挙動を計算する方が適切である。

また、この手法では、仮想物体が取りうる状態を前もってデータベース化しておく必要があるが、この作業は、仮想空間をモデル化するための専門知識を必要とするため、初心者がこのデータベースを作成して仮想空間を構築することは、非常に困難であるというデメリットも有する。

以上のような短所も持つが、仮想物体の挙動を形状情報等を用いて厳密に計算することを回避でき、状況に応じて計算負荷が小さい計算手法を採用できるため、実時間性を損なわないシミュレーションが可能である。例えば、ボルトにはまったナットを回転させる場合等は、ナットはボルトの長軸を中心とした回転平行移動に限定すれば、仮想物体は実世界とほぼ同様の物理法則に従いながらも、リアルタイム性を確保したシミュレーションを行える。

3.2.3 本研究におけるシミュレーション手法

本研究でのシミュレーション手法の提案

3.1 節で述べたように訓練環境では、機器と機器のインタラクションを実現しなければならないが、現状では、Physically-based 手法のみでのシミュレーションは、不可能であると言える。

本研究では、3次元環境に仮想空間の構築者が没入して、仮想空間を構築する手法の提案を目的にしているが、Rule-based 手法における仮想物体の状態をデータベース化する作業を3次元環境内で可能にしても、仮想空間モデル化のために専門知識を要求する事実は変わらず、Rule-based 手法の全面的な採用も得策ではない。

よって、一方の手法のみを用いて仮想空間全てをシミュレーションすることは現実的でなく、本研究では、Physically-based 手法と Rule-based 手法を組み合わせたシミュレーション手法を用いる。

そのためには、Physically-based 手法と Rule-based 手法の組み合わせ方を考案する必要があり、そのために、まず、Physically-based 手法のみでシミュレーションした場合に、実現が困難な場面を分析する。

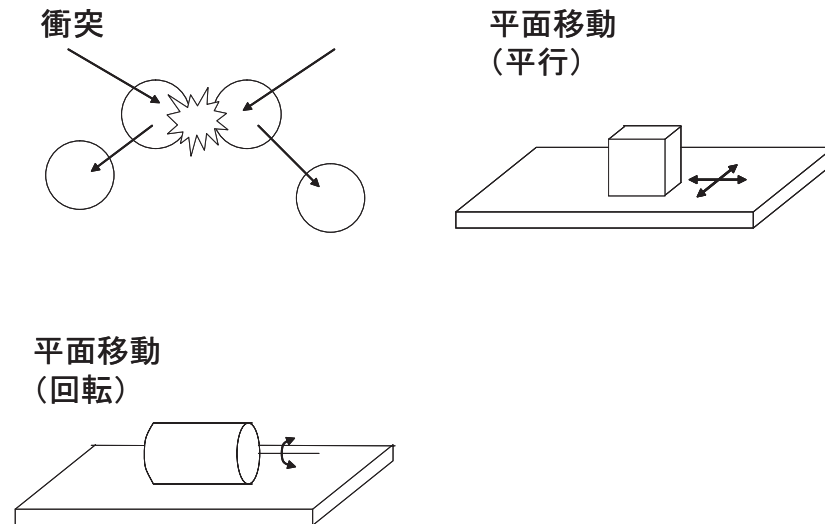


図 3.4: 衝突により実現可能な事象

仮想物体の自由落下等の単独での運動は、Physically-based 手法で十分に実現可能である。また、単純な衝突も Physically-based 手法で実現可能である。また、仮想物体が床の上を平面移動する運動も、摩擦を実装できれば、Physically-based 手法で実現可能なレベルである。Physically-based 手法で実現可能な場面を図 3.4 に示す。

しかし、図 3.5 に示すようなねじの挙動のシミュレーションでは、ねじとねじ穴の内壁が密接に接触しあっており、正確にねじとねじ穴の間での衝突情報を獲得し、ねじとねじ穴の間で発生する力を計算することは難しい。また、机と引き出しの関係を考えても、密接に接触し合っているため、衝突情報を獲得することは難しく、Physically-based 手法でのシミュレーションは実現困難である。Physically-based 手法で、実現困難な場面を図 3.6 に示す。

このように、Physically-based の手法を用いると、正確な衝突情報が獲得しにくい状況、つまり、2 物体が密接に接触しあう場面で問題が生じる。また、このような密接に接触し合う場面とは、2 物体が接触しながら、相互作用しあっている場面であり、見方を変えると、一方の物体と他方の物体の間で物理的拘束関係が生じている場面である。以下、2 物体が密接に接触し合う物理的拘束関係を、衝突等の物理的拘束関係と区別するために、「密接拘束関係」と呼ぶ。

従って、密接拘束関係が物体間で生じている場面は Physically-based 手法で挙動計算すべきでないと言える。また、密接拘束関係時の物体の挙動に注目すると、仮想物体の挙動は平行移動・軸回転といった規則的な動きであり、これらの挙動は Rule-based 手法でその挙動を再現しても、実世界の挙動と大きく変わらず訓練生は違和感を覚え

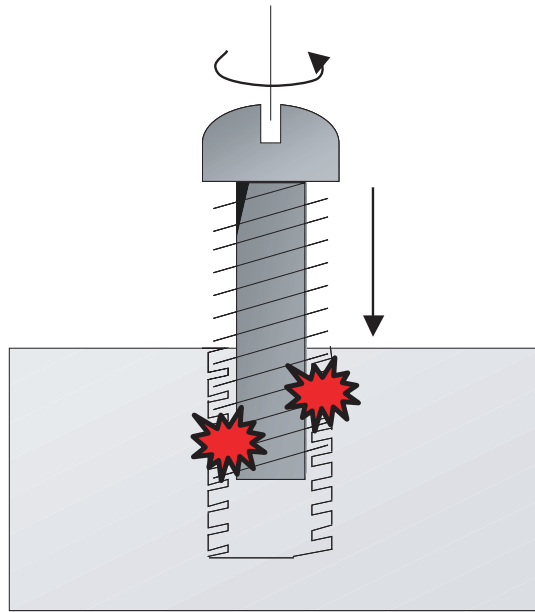


図 3.5: 衝突が多発する場面

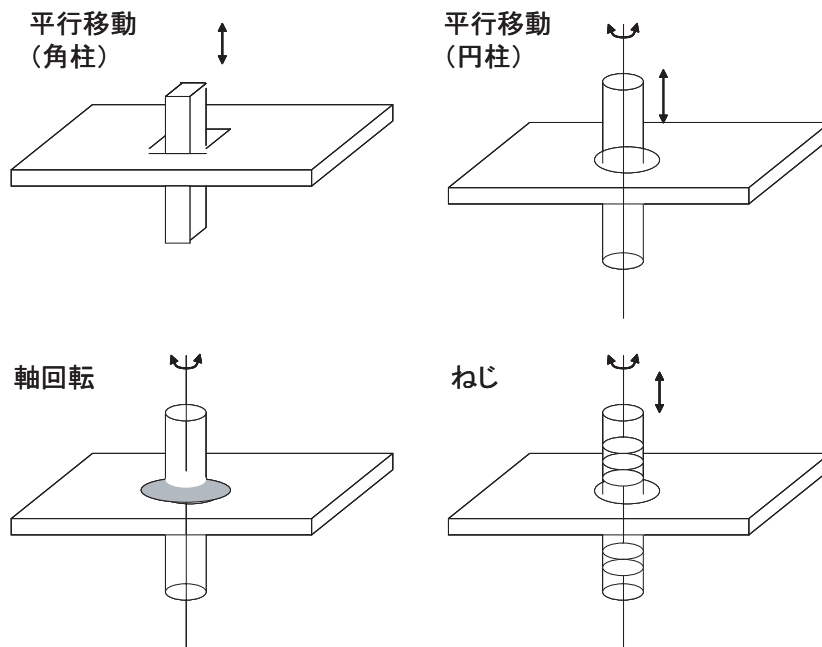


図 3.6: 衝突では実現困難な事象

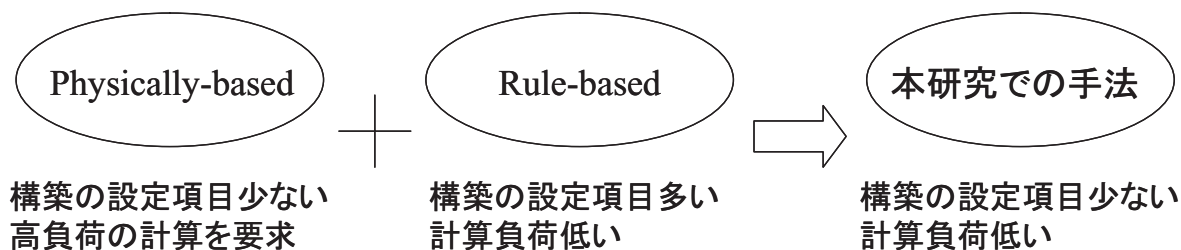


図 3.7: 本研究でのシミュレーション手法の特徴

ないと考えられる。

以上の分析を鑑みて、本研究では、平行移動や軸回転等の3次元形状間の密接拘束関係時の仮想物体の挙動は Rule-based 手法で、それ以外の場面の仮想物体の挙動は Physically-based 手法でシミュレーションする。本研究で用いるシミュレーション手法の特徴を図 3.7 に示す。

本研究でのシミュレーション手法の具体例

具体例を用いて、本研究でのシミュレーション手法を説明する。

図 3.8 に示すプリンタのトナーの交換の例では、トナーをプリンタに挿入する動き、前蓋を開ける動きは、密接拘束関係が生じている時の動きであり、これらの動きは Rule-based 手法により実現する。そのため、これらの拘束関係が存在することを仮想空間の構築時に、データベース化する必要がある。一方、飛んできたボールがプリンタに当たる動きや、前蓋が閉まっている時に人の手がプリンタ内部のトナーに接触不可能なことは Physically-based 手法で実現する。

また、図 3.9 の例では、ナット - ボルトの間でねじの拘束関係が生じているため、ナットの動きは Rule-based 手法により螺旋運動するように決定される。一方、ボルトにはまっているナット - ナットの間は密接拘束関係が生じていないため、ボルトにはまっている状態でのナット同士の衝突は Physically-based 手法で実現する。

3.3 ActCube を用いた仮想空間の構築手法

3.3.1 仮想空間の構築に必要な設定

3.2.3 項で述べたシミュレーション手法に基づく仮想空間は、Rule-based 手法により仮想物体が密接拘束関係にある時の挙動をシミュレートする。そのため、仮想空間の構築時に密接拘束関係に関する情報を設定する必要がある。一方、仮想物体が密接拘

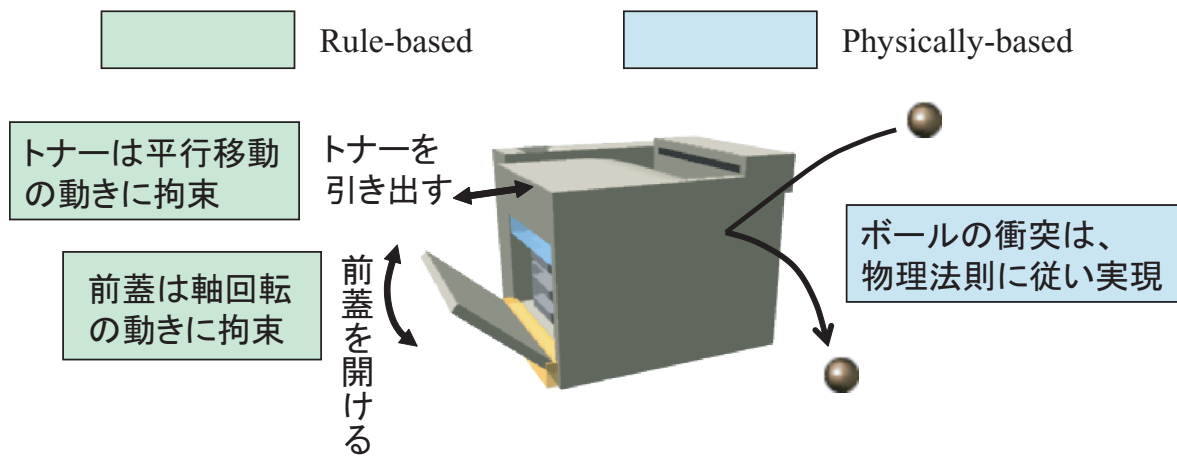


図 3.8: プリンタのシミュレーション

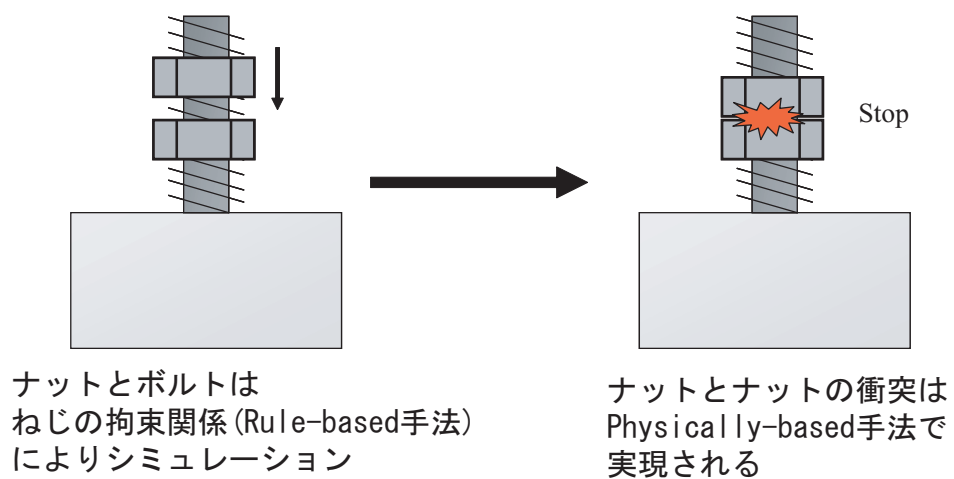


図 3.9: ねじとボルトにおけるシミュレーション

束関係にない時の挙動は、全て Physically-based 手法で計算するが、Physically-based 手法は、形状データおよび質量等の物理パラメータがあれば、シミュレーション実行可能である。そこで、本研究でのシミュレーション手法を採用した場合、仮想空間の構築に必要な情報をまとめると、以下のようになる。

1. 3次元形状データとテクスチャの作成

3次元形状データは、摩擦係数や密度を含んだ状態で用意される必要がある。なお、本研究では、仮想空間の構築に特有な仮想物体の挙動設定に主眼を置いているため、3次元形状データおよびテクスチャは用意されているものとする。

2. 密接拘束関係に関する設定

密接拘束関係を Rule-based 手法によりシミュレートするために、具体的には以下の設定項目が必要になる。

対象物体 拘束関係を設定する仮想物体

種類 設定する拘束関係の種類

箇所 拘束関係を設定する箇所

図 3.10 に示すように、拘束関係は仮想物体の形状全体に設定するだけでなく、形状の一部に設定することもある。従って、仮想物体の形状において、拘束関係を設定する箇所を記述する必要がある。

軸 拘束関係時の軸

軸回転やねじの拘束関係時における回転の中心軸のように、密接拘束関係を実現するためには、軸情報を必要とすることがある。

相手 拘束関係が発生する相手

発生条件 拘束関係の発生条件

どのような条件下になれば、対象物体と相手との間で密接拘束関係が生じている状態に移行するか、また、逆に、どのような条件下になれば、その密接拘束関係が解けるのか、を設定する。

挙動 拘束関係にある時の仮想物体の挙動の定義

仮想物体が密接拘束関係にある時の仮想物体の挙動を定義する関数を設定する必要がある。挙動は、種類に応じて一意に決まることが多い。

拘束関係はねじ全体でなく、
ねじ先の部分にのみ設定する



図 3.10: ねじにおける拘束関係の設定箇所

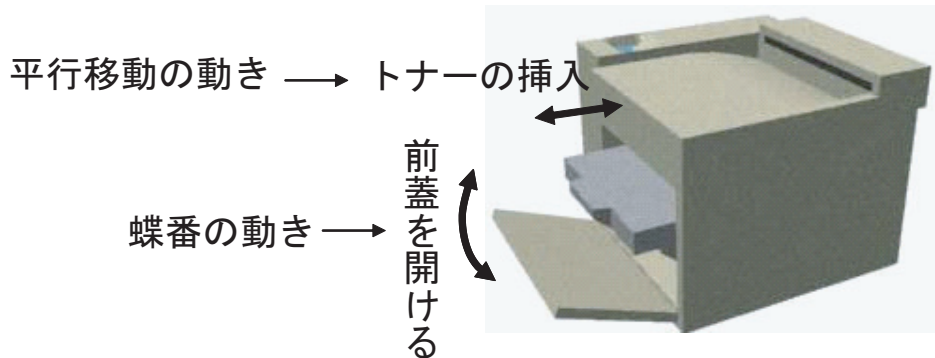


図 3.11: プリンタにおける拘束関係

図 3.11 のプリンタを具体例にとり、密接拘束関係に関する設定項目及び、これらの項目がシミュレーションのために必要になる理由を説明する。

プリンタの例では、プリンタ本体とトナーの間で平行移動の拘束関係、プリンタ本体と前蓋の間で蝶番の拘束関係が生じているが、以下、平行移動の拘束関係に絞って説明する。

まず、この関係では、「対象物体」はプリンタ本体とトナーであり、この2つの対象に「種類」として平行移動の拘束関係を設定する。さらに、プリンタ本体においてトナーと拘束関係が生じる「箇所」を、トナー受けと設定し、トナーにおいてプリンタと拘束関係が生じる「箇所」をプリンタと接する箇所に設定する。トナーにおいて、トナー受けに進入していく方向を「軸」として設定し、プリンタにおいては、トナーが進入してくる方向を「軸」として設定する。プリンタ本体から見た場合、拘束関係が生じる「相手」はトナーであり、逆に、トナーの「相手」はプリンタ本体である。さらに、種類の異なるトナーがこのプリンタのトナー受けに挿入されないようにするため、トナー受けに挿入可能な「相手」のトナーを特定して記述する必要がある。また、トナーがトナー受けに挿入されている時は、トナーは密接拘束関係にあり、Rule-based 手法でシミュレートされるが、非挿入時は、Physically-based 手法により挙動が計算されなければならない。そこで、どのような状況下になれば、密接拘束関係が生じ、

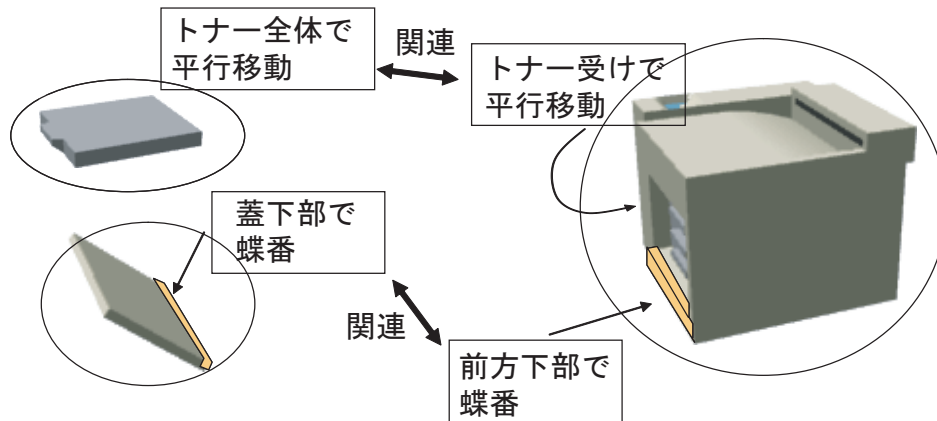


図 3.12: プリンタにおける設定

Physically-based 手法から、Rule-based 手法にシミュレーション手法が切り替わるかの「発生条件」を記述する必要がある。トナーの場合は、トナー受けにはまった時に、シミュレーション手法が、Physically-based 手法から Rule-based 手法に切り替わる必要があり、逆にトナー受けから取り去られた後は、シミュレーション手法が Rule-based 手法から Physically-based 手法に戻される必要があり、これらの「発生条件」を明記する必要がある。また、トナーがトナー受けに挿入されている時の「拳動」として、トナーが平行移動する動きを計算するための関数が必要である。

平行移動と蝶番の両方を含めた場合、プリンタに関して図 3.12 のような項目を設定する。具体的には、トナーにおいて、平行移動の拘束関係が発生し、その相手はプリンタであり、プリンタのトナー受けでも平行移動の拘束関係が生じ、その相手がトナーであると設定する。さらに、前蓋においては、蓋下部で蝶番の拘束関係が生じ、その相手はプリンタであり、プリンタの前方下部でも蝶番の拘束関係が生じ、その相手は蓋であると設定する。

本研究は、3次元モデラのように3次元環境に仮想空間構築者が没入して、その3次元環境下での構築者の3次元オブジェクトに対する直接操作だけで、仮想空間の構築を可能にする手法の提案を目的としている。これを実現するためには、3次元環境内で以上述べた密接拘束関係の設定項目を記述できる工夫が必要であり、逆に、この項目さえ3次元環境下で設定できれば、仮想空間が構築可能になる。

3.3.2 ActCube の概要

本研究では、3次元環境内での作業だけで密接拘束関係に関する情報の設定を可能とする「ActCube」と呼ぶ3次元オブジェクトを導入する手法を新たに提案する。

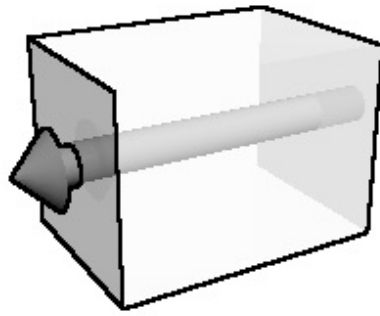


図 3.13: ActCube 基本外観

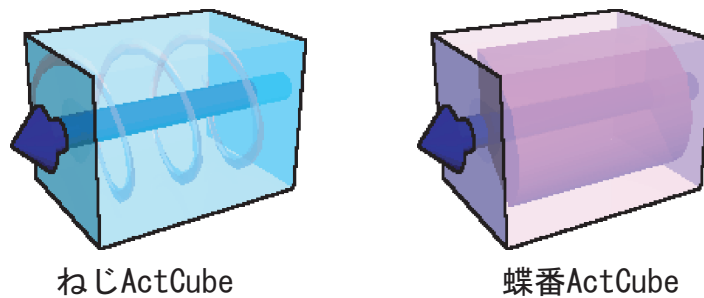


図 3.14: ねじ ActCube と蝶番 ActCube

ActCube とは、「3次元形状に付着することにより、その付着した3次元形状に他の3次元形状と拘束関係を発生させる性質を与える3次元オブジェクト」である。

ActCube は、拘束関係の「種類」とそれに伴う「挙動」に関する情報を保有する。ActCube の3次元オブジェクトとしての基本的な外観を図3.13に示す。ActCube は、保有する種類毎に色が異なり、また、基本的に直方体形状をしている。さらに、ActCube は、上面と底面があり、その上面と底面の中心を貫いて、方向を持った軸がある。色情報だけでは、ActCube の種類が認識しにくい時は、図3.14のように直方体形状の上に、その ActCube 特有の形状を重畳することも可能である。

ActCube 自体は、予めプログラミングの知識に精通した専門家が開発し、仮想空間の構築者は、専門家から提供された ActCube を使用して、拘束関係の設定作業を行う。ActCube の開発者は、以下の情報を ActCube 自身に組み込む。

1. 拘束関係の「種類」
2. 拘束関係時の仮想物体の「挙動」
3. 外観情報 (色等)

表 3.2: 拘束関係と ActCube の関係

密接拘束関係	ActCube の特性
対象物体	付着した物体
種類	色
箇所	付着した位置
軸	方向
相手	断面形状など

ActCube の大きな特徴として、ActCube は 3 次元形状に付着する性質を有している。ActCube は、この 3 次元形状に付着するという性質と図 3.13 のような外観をしているため、3 次元環境下で ActCube が持つ特性としては、次の項目が挙げられる。

1. 色
2. 付着先の 3 次元形状、付着位置
3. 軸方向の長さ、軸が貫く断面形状
4. 方向
5. 重畳形状

2.~4. は、3 次元環境内で ActCube を平行移動、回転、または拡大縮小等の直接操作により変更可能な特性である。また、4.4.1 項で詳しく述べるが、5. の重畳形状も、3 次元環境内の直接操作で変更可能である。

3.3.1 項で述べた仮想空間の構築のために必要な拘束関係の設定項目を、数値や選択項目等により記述するのではなく、本研究では、この ActCube の持つ特性を拘束関係の設定に活用することを考える。

そこで、ActCube の持つ特性と拘束関係の設定項目を表 3.2 に示すように対応付ける。対応付けの基本方針として、構築者は積み木を操作する感覚で ActCube を 3 次元形状に付着させる作業により、拘束関係の設定が完了するようにしてある。ActCube を付着させた先の 3 次元形状が拘束関係の「対象物体」であり、また、拘束関係の「箇所」が ActCube が付着した位置というように、構築者が予め一般生活の中で獲得している知識を用いれば、容易に想像できるように対応付けた。したがって、構築者は、一

般生活の中で獲得している知識を参考に ActCube の特性を直接操作により仮想空間内で設定すれば、仮想空間を構築できる。

また、ActCube は、上記で述べた 3 次元形状に付着する機能に加えて、以下の ActCube 特有のルールが存在する。

1. ActCube は凹または凸の対属性を有する
2. 凹凸は結合する
3. 凹凸結合時には、付着している 3 次元形状の挙動を制御する機能を発動する
4. シミュレーション時には、見えなくなる

以下、ActCube の 3 次元形状への付着とこれらのルールについて詳細に説明し、この ActCube を導入したことにより、前節で述べた密接拘束関係の設定が 3 次元オブジェクトの操作だけで実行可能になり、また、3.2.3 項で述べたシミュレーションが可能になることを述べる。

3.3.3 ActCube の機能

「ねじの拘束関係」を具体例にとり、ActCube の機能について述べる。

3 次元形状への付着

ActCube は、3 次元形状に付着する機能を持つ。図 3.15 に示すように、構築者は、ActCube を 3 次元形状に付着させて、拘束関係に関する設定を行う。ActCube が付着した先の 3 次元形状が拘束関係の「対象物体」であり、ActCube が覆う範囲が拘束関係が発生する「箇所」である。

もし、ActCube が、拘束関係が発生する箇所を覆うには十分な大きさではないならば、図 3.16 に示すように拘束関係の範囲を覆う大きさに拡大縮小させてから、付着作業を行う。

また、ねじの拘束関係では、ねじが回転する際の「軸」の情報が必要であり、ActCube が持つ方向を使用してこの軸を設定する。ねじの場合、図 3.17 に示すように、ActCube の軸方向とねじの進行方向とを一致させ、ねじ先にねじ ActCube をねじ先の長さに合わせて付着させる。

この付着作業により、3 次元形状に拘束関係の「対象物体」と「箇所」、「軸」が指定でき、また、ActCube が最初から持つ「種類」と「挙動」も指定したことになる。

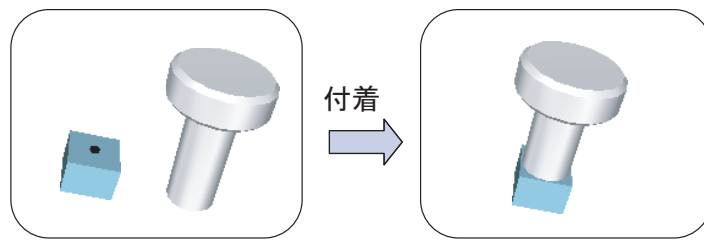


図 3.15: ActCube のボルトへの付着

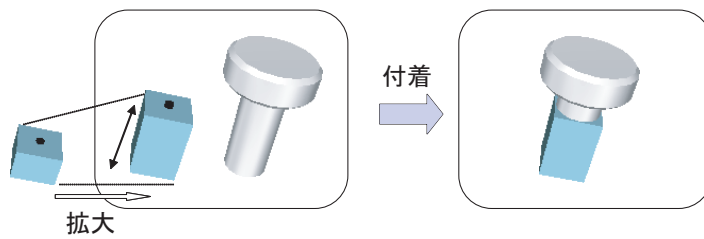


図 3.16: ActCube の大きさの調節

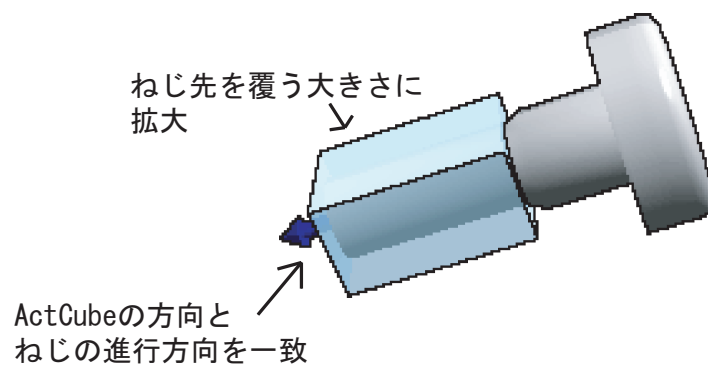


図 3.17: ActCube をねじに付着させる場合

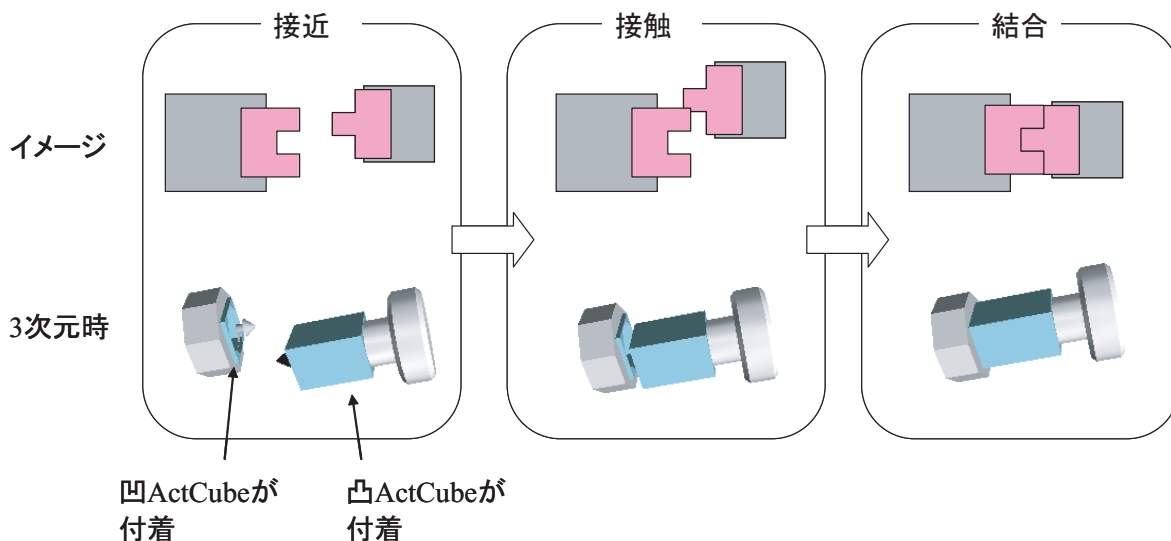


図 3.18: 凹属性と凸属性

凹属性と凸属性

拘束関係の相手を定義する場面をねじを例に考える。仮想空間にねじ A とナット A、ナット B が存在する場面を想定する。

例えば、ねじ A の拘束関係の相手をナット A というように、拘束関係の相手を直接指定して定義すると、ねじ A に関してナット B との間でも拘束関係が発生するように定義する必要があり煩わしい。もし、仮想物体の数が多いと、新しい仮想物体を追加する際の定義作業が膨大になってしまう。

そこで、拘束関係の相手を直接指定するのではなく、例えば、ねじはねじ穴の性質を持つ仮想物体と拘束関係を持つというように、間接的に「特定の性質を持っている仮想物体と拘束関係を持つ」という形で拘束関係の相手を定義する方が望ましい。新しい仮想物体を導入する際に、性質 A を持たせれば、既存の仮想物体を変更せずに、新しい仮想物体と既存の仮想物体が拘束関係を持てるようになる。

この特定の性質を指定可能にするために、ActCube に拘束関係の種類が存在する。しかし、密接拘束関係を実現することを考えた場合、その拘束関係に携わる 2 つの仮想物体は、対等な関係ではないことが多い。例えば、ねじの拘束関係の場合、関係ある仮想物体はねじとねじ穴を有する仮想物体であり、立場を入れ替えることはできない。つまり、1 つの ActCube に対して、役割の異なる凹凸の 2 つの属性が必要となる。そこで、ActCube は、1 つの種類に対して、凹 ActCube と凸 ActCube が存在する。

また、1 つの密接拘束関係に関わる 2 つの仮想物体は、双方が双方を拘束しあうとい

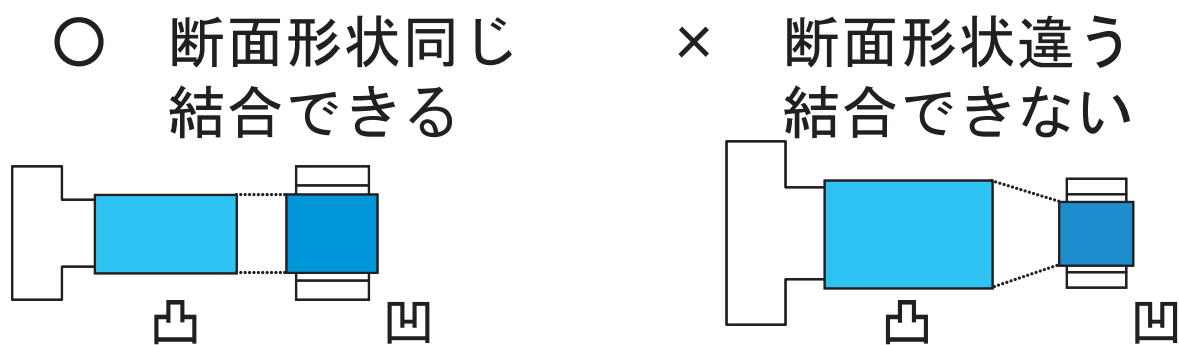


図 3.19: ActCube の結合条件

う関係であることはなく、普通、拘束する側と拘束を受ける側に分けられる。例えば、ねじの拘束関係の場合、ねじ穴がねじの動きを拘束しており、拘束する側はねじ穴、拘束を受ける側はねじと区別できる。そこで、本研究では、ActCube の凹凸属性に関して、凹 ActCube を拘束する側、凸 ActCube を拘束を受ける側に付着することとする。

なお、凸 ActCube と凹 ActCube は外見上、保有する軸の色が異なる。

結局、ねじの場合、構築者は、ねじ側に凸 ActCube、ねじ穴側に凹 ActCube をそれぞれ付着させることになる。

密接拘束関係は一般的に 2 物体間で生じ、一対一の関係であることが多い。しかし、一対多の関係である密接拘束関係も存在する。例えば、ギアを考えると、1 つのギアに複数のギアがかんでいることもある。このような一対多の関係の場合、密接拘束関係を凹凸と分けることはできない。一対多の密接拘束は出現機会が少ないこともあり、本研究では、まず、一対一の密接拘束関係のみを取り扱うこととし、一対多の密接拘束の実現は将来課題としている。

凹と凸の結合

ActCube は、図 3.18 に示すように同種の凸 ActCube と凹 ActCube 同士が接触すると結合する機能を持つ。この結合は、3 次元形状が、密接拘束関係にある状態に移行したことを意味する。この結合機構そのものが密接拘束関係の「発生条件」を満たす。

また、ネジには太さがあり、2cm のねじ穴に 3cm のねじを締められないことを実現するために、ActCube の断面形状の属性を 2 つの ActCube の結合条件に加える。例えば、図 3.19 に示すように凸 ActCube と凹 ActCube の断面形状が同じ場合、結合可能だが、断面形状が異なる時は結合できない。この断面形状の違いにより、トナーとトナー受けの関係やねじの太さの違い等が表現できる。

挙動制御

ActCube は、凸と凹が結合していない時には、付着している 3 次元形状の移動に対して何ら影響を与えない。しかし、ActCube は、凸と凹が結合している時には、付着している 3 次元形状の位置を制御する機能を果たす。具体的には、以下の 2 つの機能を果たす。

1. 付着先の 2 物体同士間で Physically-based 手法による衝突処理が実行されることを禁止する
2. 入力された移動量を、適切な 3 次元形状の移動量に変換する

Act-Cube が結合している時は、1. の機能により、2 物体間で Physically-based 手法により挙動がシミュレートされるのを防ぎ、2. の機能により、Rule-based 手法によるシミュレーションを実行する。図 3.20 に示すように、ねじとナットの場合では、ActCube 結合時、ねじに軸回転方向の移動量に加わると、ActCube はその移動量をねじの動きを模擬させるような移動量に変換し、ねじを螺旋状に移動させる。このように ActCube は、仮想物体をあたかも拘束関係にあるように挙動を決定する働きをする。

シミュレーション時の ActCube の外見

構築者が構築した訓練環境を、訓練生が体験する時にも、ActCube の姿が見えていない必要はない。そこで、図 3.21 に示すように、ActCube は、シミュレーション中は見えなくさせることにする。

ねじの動きを模擬

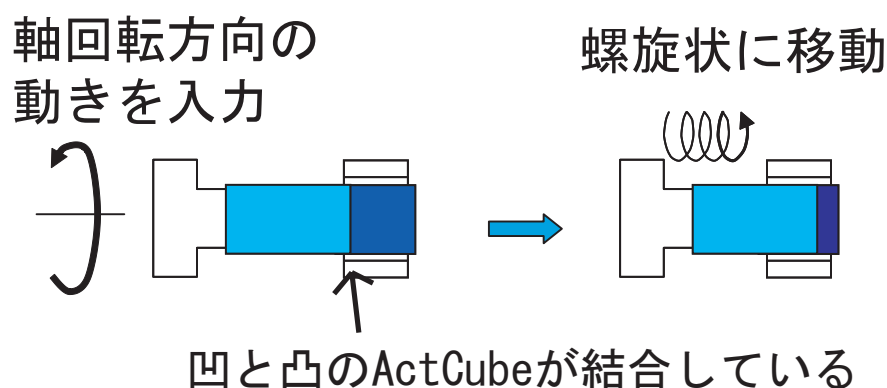


図 3.20: ActCube の機能発動

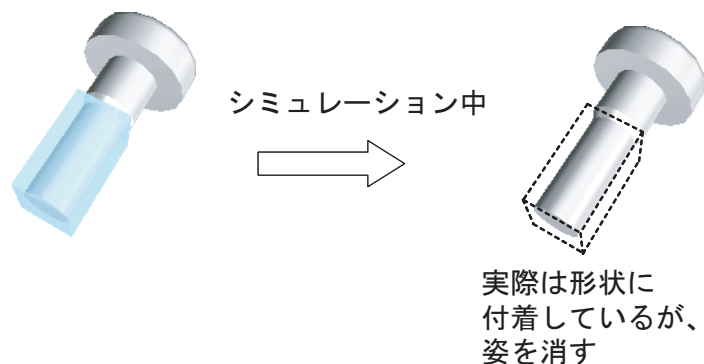


図 3.21: シミュレーション中の ActCube の外観

ActCube を含んだシミュレーション

ActCube を含んだ仮想世界のシミュレーション手法を図 3.22 を用いてまとめる。

まず、ActCube が介在しない時は、2 物体同士の接触等は、Physically-based 手法で計算される。

次に、ActCube が介在し、凹と凸が結合できる時は、ActCube が仮想物体同士を結びつけ、2 つの物体が拘束関係にあるように Rule-based 手法で挙動が決定される。

つぎに、ActCube が介在するが、凹と凸が結合できない時は、ActCube は、仮想物体の挙動に影響を与えず、2 物体同士の相互作用は、Physically-based 手法で計算される。つまり、ActCube が介在しない時と同じ結果になる。

3.3.4 ActCube の種類

ActCube により、2 物体間が拘束関係にある時の形状の位置姿勢計算が行われる。その密接拘束関係としては、平行移動、軸回転等が考えられ、ここでは代表的な ActCube を説明する。いくつかのイメージ図を図 3.23 に示す。

把持 ActCube 形状を他の形状で把持可能にする場面で使用する。例えば、レンチでナットの頭をはさむ場面等に用いられる。

平行移動 ActCube 形状が他の形状に拘束されて平行移動しかできない場面で用いる。

例えば、引き出しやペンキャップをペンにはめる操作等に用いられる。

軸回転 ActCube 形状が他形状の軸を中心とする回転運動に制限される場面で使用する。例えば、ハンドルやバルブの動きのために用いられる。

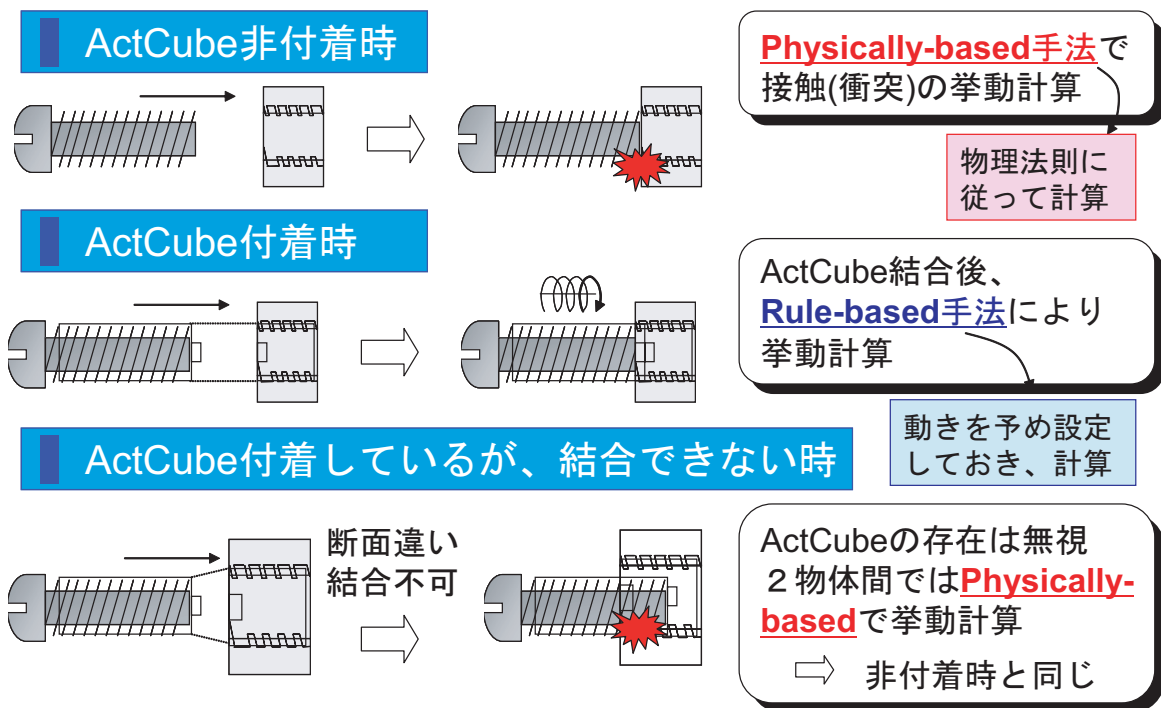


図 3.22: ActCube を含んだシミュレーション

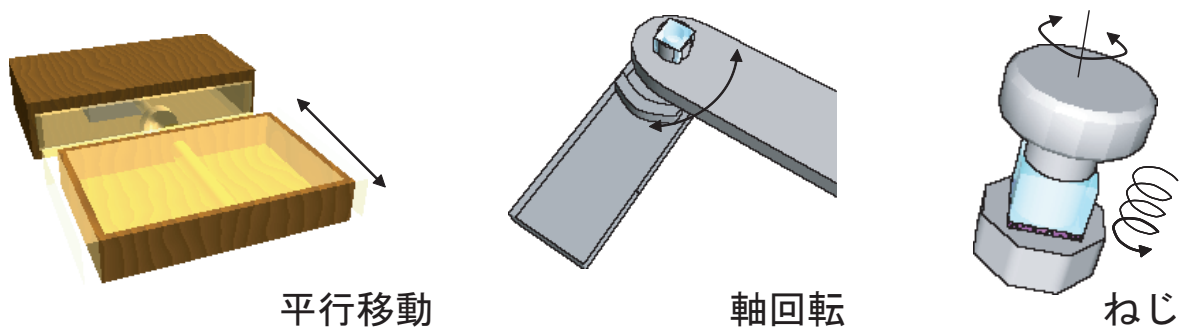


図 3.23: ActCube の種類

蝶番 ActCube 回転角度が 360 度以内の軸回転に動きが制限される場面で用いる。例えば、ドア等に用いられる。

ねじ ActCube ねじのように軸回転につれて、軸方向に平行移動する場面で用いる。その名称通り、ねじの動きを表現するために用いられる。

本論文では、密接拘束関係を ActCube により実現していることを述べたが、密接拘束関係以外の物体間の関係も実世界には多く存在する。その例として、スイッチやつまみ調節等が考えられるが、これらは、訓練環境をより多様化させるために実現すべき関係である。本研究では、研究の初期段階としてこれらは実現しなかったが、実現

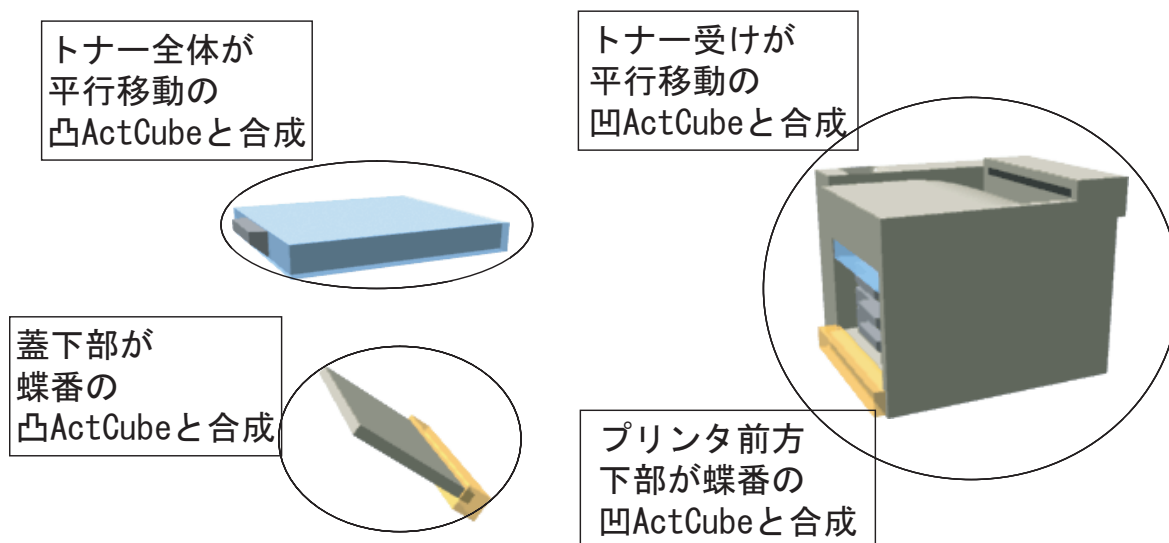


図 3.24: プリンタへの ActCube 付着

するための構想を付録 B. で述べる。

3.3.5 ActCube を用いた仮想空間の定義例

実際に、ActCube を用いてプリンタを定義した例を図 3.24 に示す。トナーとトナー受けの平行移動の関係においては、拘束を受けて平行移動をする側はトナーであるため、トナーに凸 ActCube、トナー受けに凹 ActCube を付着させる。定義する際は、まず、トナー全体を平行移動の凸 ActCube で覆う。また、プリンタのトナー受けの箇所に平行移動の凹 ActCube を付着させる。この際、凸 ActCube の軸方向をトナー受けに入っていく方向と一致させる必要がある。また、凸 ActCube と凹 ActCube の上面の長方形形状を同じ形にしておく。また、シミュレーション時、図 3.25 のように、トナーがトナー受けに挿入されていない時は自由に移動させることができるが、トナーとトナー受けの平行移動の ActCube が接触すると、ActCube 同士が結合し、拘束関係が発生している状態に移行し、トナーは平行移動の動きに制限される。

ボルトに締まっているナットをレンチで回す作業を実現するために必要な ActCube の定義例を図 3.26 に示す。ナットがボルトに締まるナット - ボルト間の関係はねじ ActCube で、レンチでナットを動かすレンチ - ナット間の関係は把持 ActCube で実現する。この定義により、把持 ActCube の働きによりレンチでナットを動かし、ねじ ActCube の働きによりナットがボルトに締まることが実現できる。この場合、ナットには、2 つの ActCube が付着する。

また、ボルトにナットが 2 つ締まっている状況というのは、図 3.27 に示すような関係

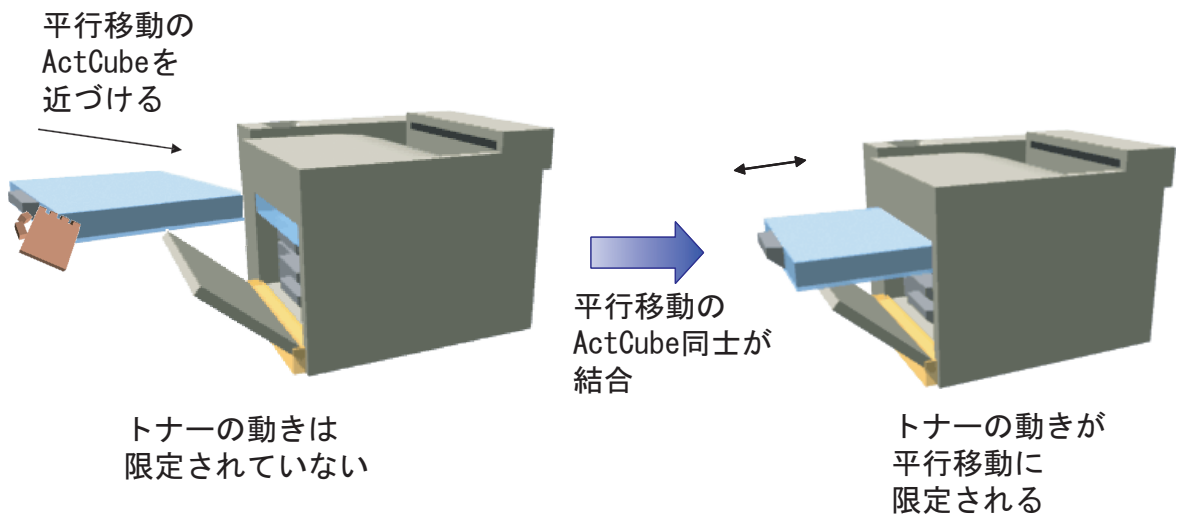


図 3.25: ActCube によるプリンタのシミュレーション

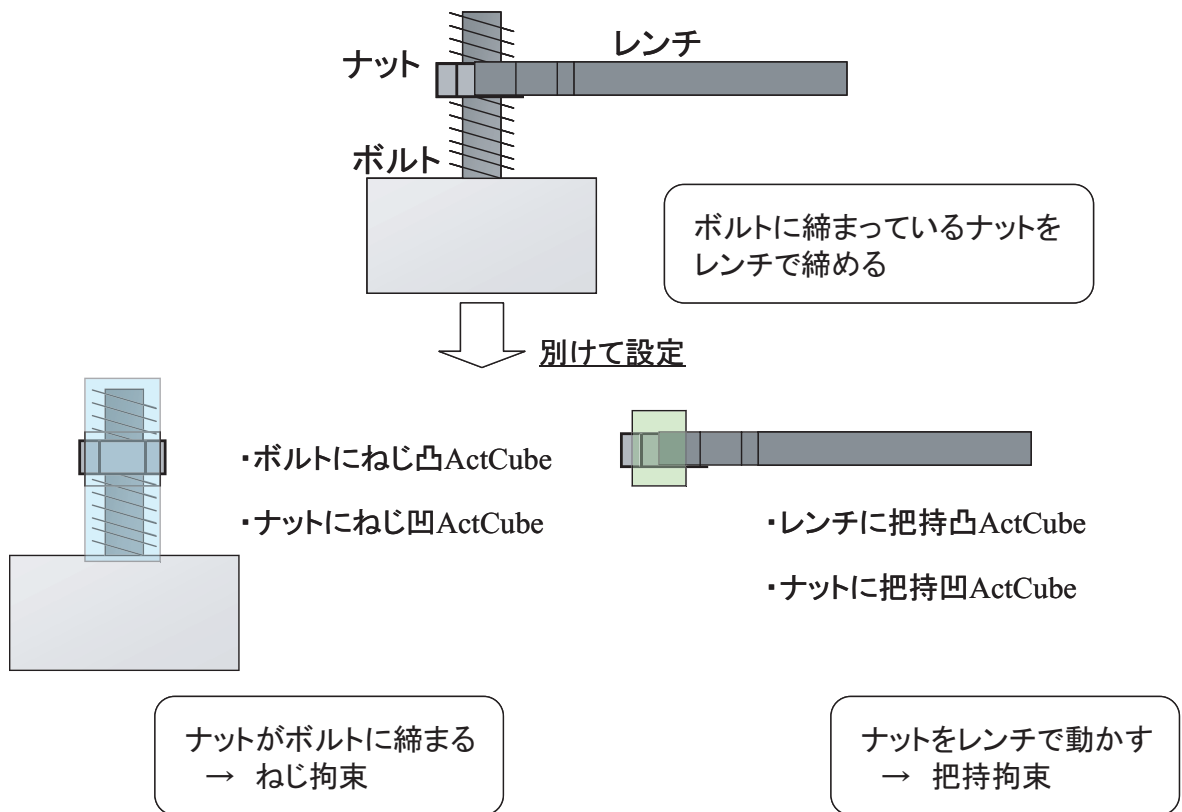


図 3.26: ボルトに締まっているナットをレンチで回す作業の定義例

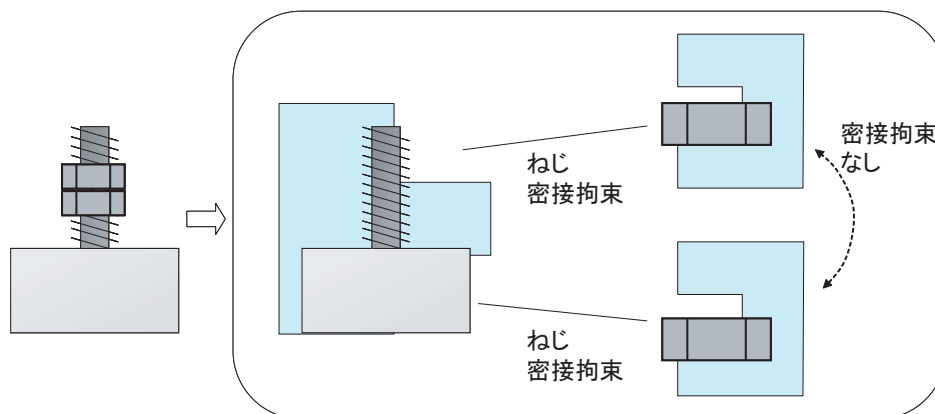


図 3.27: ボルトにナットが2つ締まっている時の関係

になる。ねじ凸 ActCube は、複数の凹 ActCube と結合する機構を持たせてあり、同時に2つ以上のナットがボルトに締まっている状況が実現できる。この際、ボルトに締まっているナット同士は、密接拘束の関係にないため、ナット同士の衝突は、Physically-based 手法の衝突処理によって実現される。

3.3.6 ActCube 手法の利点

3次元環境内での構築

ActCube という拘束関係の設定を行うことができる特殊な3次元オブジェクトの導入により、3次元形状と ActCube の直接操作のみで仮想物体の挙動を設定可能になる。

3.3節で、仮想空間を構築する場合、拘束関係に関する種類、発生箇所等の設定が必要であることを述べた。本来、これらの項目は、数値設定、定義ファイルの書き換えにより行うことが普通である。しかし、ActCube の特性とこれらの項目を対応付けることにより、本来数値設定等で行われていた作業が ActCube の移動と変形作業だけで実現可能となった。

本研究で採用したシミュレーション手法では、密接拘束関係の設定のみをシミュレーションのために必要な情報として求められるため、ActCube により密接拘束関係の設定が実現できるということは、3次元環境内での3次元オブジェクトの直接操作だけで仮想空間の構築が行えることになる。

仮想空間を構築する際に必要な専門知識について

この ActCube を用いて、仮想空間を構築する構築者は、ActCube の内部機構に関する知識がなくても構築可能である。

図 3.28 に示すように、仮想物体を構築する際は、専門知識を持つ人が ActCube を開発し、VR 空間構築者は仮想物体の 3 次元形状にその ActCube を視覚的に付着して、仮想物体を構築していく。このように、仮想空間の構築に際して専門知識を要求する部分は、その専門知識を持つ人に任せ、実際に VR 空間を構築する人は、専門知識を必要としない構築形態にすることにより、構築者は容易に仮想物体を開発できるようになる。

本研究では、3 次元形状に密接拘束関係を設定するための ActCube を専門家が開発し、構築者は、その ActCube を 3 次元形状に付着させて密接拘束関係を設定し、仮想空間を構築する方式を採っている。一方、専門家が 3 次元形状に他の形状と密接拘束関係を発する機能を直接組み込んだ 3 次元形状を提供し、構築者はその 3 次元形状を配置するだけで仮想空間を構築する方式も考えられる。しかし、その方式では、結局、専門家でなければ、仮想物体の挙動を定義できない、つまり、仮想空間が構築できないという状況になってしまい、本研究の誰もが仮想空間を構築できるようにするという目的を達成できない。また、様々な仮想空間を構築する場合、仮想物体は多種多様に渡るため、開発者がそれら全てに組み込む作業を行わなければならない、開発者側の負担が大きくなってしまう。

また、構築者が仮想空間を構築するために必要な ActCube が提供されていない場合などに、構築者側から ActCube 開発者側へ要求を伝えるフィードバックがかかることにより、より質の高い仮想空間が構築されることが期待される。

情報の再利用

ActCube は 1 つの拘束関係に対し凹と凸の対属性がある。このルールにより、拘束関係の相手を直接指定しなくてよくなるため、新たな仮想物体の追加・削除が比較的容易に行える。つまり、既存のデータを新しく構築する他の仮想空間でも再利用することができ、構築労力の削減につながることを期待される。

これは 2.3 節で述べた仮想空間の構築を容易にするための方策のひとつである仮想空間を分割して構築することに適うものであり、本手法で大規模仮想空間の構築も実現可能である。

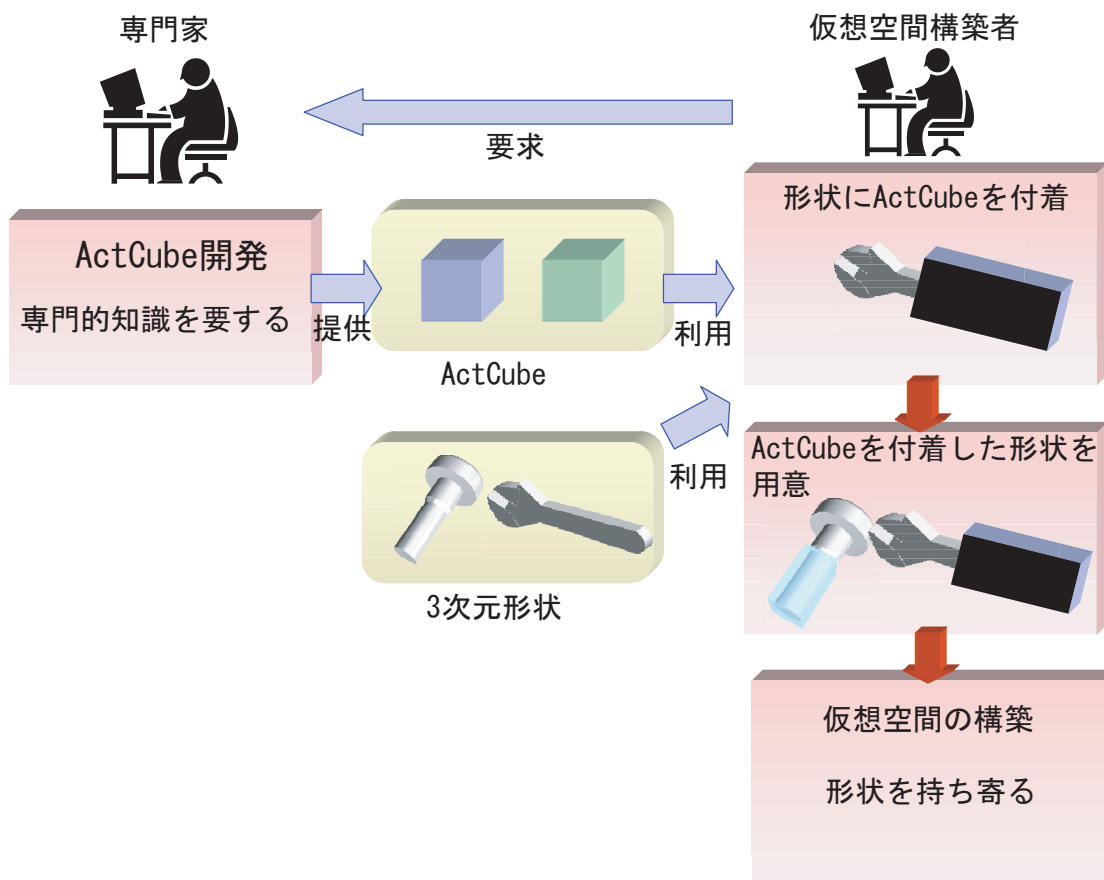


図 3.28: ActCube の開発

3.3.7 ActCube 手法の欠点

詳細な設定が難しい

普通、コンピュータのインタフェースには、インタフェースの煩雑さと機能の多さのジレンマがある。インタフェースをシンプルにしようとするソフトウェアの機能が減り、細かい設定ができなくなる。逆に機能を増やすとインタフェースが複雑になる。本研究は、このジレンマの中で、仮想空間を構築する際に必要な最低限の設定項目を抽出し、それらを可能な限り直感的に設定できるようにするために、3次元インタフェースを導入したと捉えることができる。

そのため、ActCube 手法では、仮想空間の構築に最低限必要な項目しか設定できないため、例えば、訓練生の入力した力に対して、ねじが回転する角度を細かく変更したい場合等のように、仮想空間を細かくカスタマイズしたいという要望に対応することは難しい。

複雑な仮想物体の構築

構築者が3次元形状に ActCube を付着させる作業を実行する時に、その3次元形状が他の3次元形状と多くの拘束関係を生じる仮想物体を構築する場合は、1つの3次元形状に多くの ActCube を付着させる作業を行わなければならない、その数が増えると構築者に多大な労力をかける結果になってしまう。

この問題を解決する方策としては、コンピュータが3次元形状を解析し、拘束関係が生じるであろう形態をしている箇所に自動的に ActCube を配置するツールを開発する策が挙げられる。

第 4 章 仮想空間管理シミュレーションシステム (Act System) の開発

本章では、3章で述べた ActCube を用いた仮想空間の構築手法を用いて、実際に仮想空間を構築し、シミュレーションするための試作システムとして開発した仮想空間管理シミュレーションシステム Act System について述べる。まずはじめに、ActCube を用いた仮想空間の構築とシミュレーションを実現するシステムとして Act System が具備すべき機能について述べる。その後、Act System のインタフェースについて説明し、次に Act System を構成する各サブシステムの役割を説明する。その後、仮想空間の構築者が Act System で実際に仮想空間を構築する際の操作手順を説明し、最後に Act System が ActCube を含む仮想空間をシミュレーションする際の流れ等について説明する。

また、説明の前に、便宜的に以下の単語を定義する。仮想空間の構築・シミュレーション、両方の場面において共通する事項について述べる時は、構築者・訓練生を統一的に「ユーザ」と呼ぶ。また、ActCube が付着した 3次元形状を「仮想部品」、その仮想部品に付着している凸と凹の ActCube が結合することにより、つながりあった複数の仮想部品の集合を「仮想機器」と呼ぶことにする。

4.1 Act System が具備すべき機能

3章で、ActCube を用いることにより 3次元オブジェクトの直接操作だけで、拘束関係の設定ができ仮想空間の構築を可能にする手法を説明したが、ここでは、その手法に基づき、仮想空間を構築でき、かつ、その構築した仮想空間のシミュレーションを実行可能にするために、Act System が具備すべき機能について考察する。

3次元環境への没入を実現するために、Act System は、まず、ユーザの身体の動きを計測する機能、仮想空間をユーザの視覚・触覚へ表示する機能、が必須である。また、3章で述べた手法に基づき、仮想空間の構築を実現する機能、その構築した仮想空間をシミュレーションする機能も必要となる。以下でそれぞれの機能について要求仕様をまとめる。

要求仕様 1 人間の身体を計測する機能

一般的な VR ソフトウェア同様、Act System でも、人間の手の動きを計測する必要がある。仮想空間の対話性を高めるためには、データグローブ等のデバイスを用いることにより、手の指の動作まで計測する必要がある。しかし、ActCube 手法による仮想空間の構築作業は、ActCube の移動・変形操作で実現できることから、構築者から入力される情報としては、手の移動量・開閉状態で十分である。よって、Act System は、ActCube による仮想空間の構築・シミュレーションのために、手の平行と回転の 6 自由度の移動量と手の開閉状態を計測できる機能が最低限必要である。

また、仮想物体の操作を円滑に行うための手法として、両手による操作を用いる手法が有効である^{[18][30]}。両手操作は、日常の様々な場面で行われており、片手操作に比べて作業効率等の面で優れている。そこで、Act System でも、両手作業を可能にするために、片手でなく、両手を計測する。

要求仕様 2 仮想空間を視覚・触覚等に表示する機能

仮想空間を構築・シミュレーションするためにディスプレイに仮想空間を描画する機能が必要である。また、視覚情報だけではなく、力覚情報・聴覚情報を同時に提示することにより、ユーザはより高い臨場感で仮想空間の構築・シミュレーションが行える^[31]。

また、ディスプレイへの表示方法としては、CrystalEyes、HMD 等の立体視ができる表示デバイスを実装できれば、ユーザのさらなる臨場感の向上が期待できる。

要求仕様 3 ActCube を用いて仮想空間を構築する機能

ActCube 手法において仮想空間の構築のためには、ActCube を 3 次元形状に付着し、拘束関係の設定を行う必要がある。この付着作業を実現するためには、構築者が ActCube 及び 3 次元形状に対して以下の操作を実行できる必要がある。

1. 構築者が ActCube 及び 3 次元形状を移動させる機能
2. 構築者が ActCube を 3 次元形状に付着させる機能
3. 構築者が ActCube を拡大縮小する機能
4. 構築者が開発者の用意した ActCube を利用可能にする機能
5. 構築者が用意した 3 次元形状を読み込む機能

要求仕様 4 仮想空間をシミュレーションする機能

ActCube 手法に基づく仮想空間のシミュレーションを行うためには、ActCube の機能に従った 2 物体間の密接拘束関係の Rule-based 手法によるシミュレーションと、仮想物体が拘束関係にない時の Physically-based 手法による自由落下や 2 物体の衝突等の運動をシミュレーションする必要がある。このシミュレーションを実現するためには、以下の機能が求められる。

1. 訓練生が 3 次元形状を移動させる機能
2. 3 次元形状間で Physically-based 手法により衝突処理を行う機能
3. ActCube 同士の衝突・結合を行う機能
4. ActCube 結合時、ActCube の内部情報として定義されている挙動制御情報に従い、Rule-based 手法により付着している 3 次元形状の挙動を制御する機能

また、これらの基本的な機能のほかに、VR 技術による訓練環境を実現するためには、以下の機能が求められる。

要求仕様 5 仮想機器を保存する機能

仮想空間内での訓練を実現するために、仮想空間の構築者が構築した仮想機器をデータベースに保存しておく機能が必要である。また、訓練時に途中の状態を一時保存しておく場合にも、この機能が使用される。

要求仕様 6 保存してある仮想機器を読み込む機能

訓練環境の実現のためには、データベースに保存してある仮想機器のデータを読み込み、仮想機器を計算機上に再構成する機能が必要である。

さらに、ActCube 開発者のために、次の機能が必要である。

要求仕様 6 ActCube 開発者が ActCube を開発するためのスクリプト

アプリケーションを変更せずに機能を追加する機能拡張ファイルを一般にプラグインと呼ぶが、ActCube もこのプラグインのように、Act System を変更することなく追加可能にするため、ActCube 開発のための専用スクリプトを定義することが望まれる。

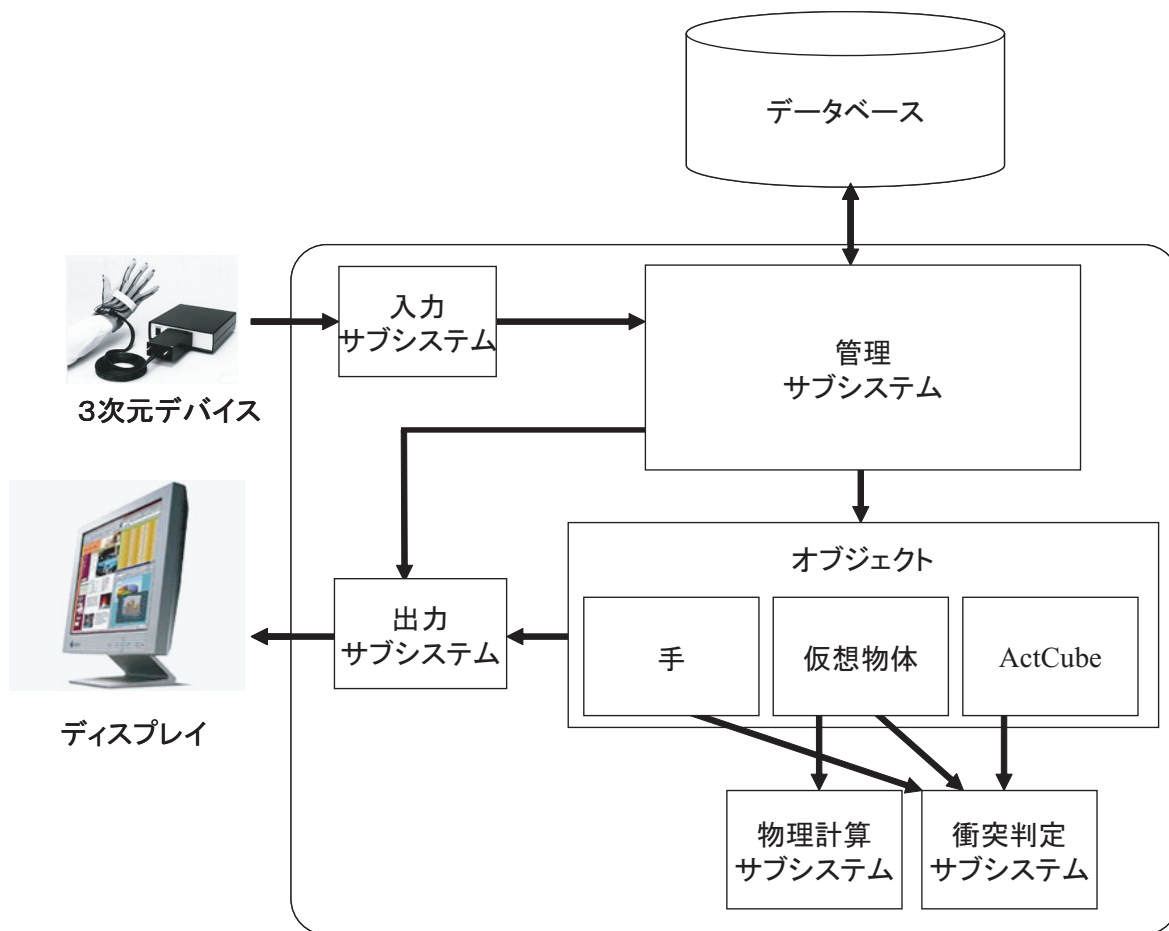


図 4.1: Act System のシステム構成

4.2 Act System の概要

本節では、前節で述べた要求仕様に従い開発した Act System のシステム構成と、仮想空間構築時のユーザインタフェースについて概説する。

4.2.1 ソフトウェア構成

4.1 節で述べた要求仕様に従って開発した ActCube のシステム構成を図 4.1 に示す。Act System は、C++言語を用いて、システム構成図上の全てのサブシステムを単一計算機上 (DOS/V 機; OS: Linux; CPU: PentiumIII 1266MHz × 2) 上に実装した。以下、各サブシステムの概要を説明する。

入力サブシステム

データグローブ等の3次元デバイスが計測したユーザの手の動作情報を受け取り、管理サブシステムに入力する役割を果たす。また、3次元デバイスからの情報は手ぶれ等のノイズを含むため、それらをフィルタリングする機構も含む。

4.2.2項で後述するように、Act Systemでは、要求仕様1を満たすために、磁気式モーションキャプチャシステムをユーザの手の位置の計測、ボタンをユーザの手の開閉状態の計測に使用しており、ユーザの両手の動きおよび手の開閉が入力パラメータである。

なお、他の3次元デバイスへの変更は、この入力サブシステムを置換するだけでよい。

出力サブシステム

3次元形状の形状データ、位置・姿勢やテクスチャ情報等に基づき、3次元グラフィックスのアニメーションをディスプレイに描画する役割を担う。

視覚情報だけでなく、力覚デバイスを用いて接触等を実現する方法も考えられるが、力覚ディスプレイは規模が大きくなりやすく、また、操作の可動域にユーザの手の動作を制限するため、本研究では採用しない。

なお、本研究では、ディスプレイへ描画する機能は、SGI社から発売されているOpenGL Performer^[32]ライブラリを使用し開発した。

管理サブシステム

システム全体の制御を行う部分である。主な機能を以下に示す。

1. システム全体の処理の流れを制御する

制御部は、システム全体を制御し、データの流とその処理方法を決定する。また、一定時間を経過する毎に出力サブシステムに対して画面を更新するように、オブジェクトに対して各自固有の処理を行うように命令する。

2. 入力サブシステムからの手の位置情報を補正する

入力サブシステムから送られてくる手の位置情報の座標系は、Act System内に構築された仮想世界の座標系とは一致しない。そのため、管理システムは、Act System起動時、入力サブシステムのデータの座標系と仮想世界の座標系を一致さ

せるため、キャリブレーションを行い、その後は、そのキャリブレーションで得られた補正情報に基づき、入力データを補正する。

3. データベースから 3 次元形状・ActCube・仮想機器のデータを読み込む機能
4. 仮想機器をデータベースに保存する機能

オブジェクト

Act System において仮想空間内に存在する 3 次元オブジェクトの最小単位は、ユーザの仮想の手・3 次元形状・ActCube である。Act System を起動中、管理サブシステムから処理を行う命令が送信されてきた時にそれぞれが以下で述べる役割を果たす。

- ・仮想の手 管理サブシステムは、ユーザの手の位置情報、開閉状態の変化を検出すると、その情報をこの仮想の手に伝達する。仮想の手は、その情報に基づき、仮想空間内での位置姿勢を更新し、また、手の開閉状態に応じて 3 次元オブジェクトの把持・開放動作を行う。
- ・3 次元形状 シミュレーション時は、仮想空間内に存在する他の 3 次元形状と衝突判定を行い、衝突が検出された時は、物理計算サブシステムを利用して、衝突に発生する力を計算し、Physically-based 手法によるシミュレーションを実現する。
- ・ActCube ActCube は、構築時、仮想の手の動作に応じて、自身を拡大縮小させたり、また、3 次元形状との衝突を検知した場合、その 3 次元形状に付着する処理を行う。

シミュレーション中は、付着先の 3 次元形状が移動した場合に、他の 3 次元形状に付着している ActCube との間で衝突判定を行い、同じ種類で属性が異なる (凸の場合は凹、凹の場合は凸)ActCube と衝突していれば、結合処理を行う。その後は、ActCube の内部に組み込まれている 3 次元形状の挙動を決定する関数に従い、3 次元形状を平行移動等の動きに拘束する。

衝突判定サブシステム

衝突判定サブシステムは、3 次元オブジェクト間の衝突判定を行う。まず、球近似による簡易な衝突判定を行い、衝突している可能性があれば、厳密な衝突判定を実行

し、衝突している 3 次元オブジェクトの組み合わせと 3 次元形状間の接点を正確に算出する。

衝突判定アルゴリズム^[33]は様々な手法が提案されているが、本研究では、衝突判定ルーチン PQP ライブラリを使用した^[34]。これは、PQP ライブラリが以下の特徴を有するためである。

1. 凸多面体だけでなく、非凸多面体に適用可能であり、非凸多面体間の最短距離計算が実行可能である
2. 判定が比較的高速であること
3. 特別な前処理を必要としないため、実装が容易である

凸多面体とは、多面体の 2 頂点を結ぶ線分が必ず形状内に含まれる多面体である。凹多面体の衝突判定は一般的に難しく、Physically-based 手法によるシミュレーションを行う時は、仮想物体の形状を凸多面体と仮定することが多い。しかし、非凸多面体に衝突判定アルゴリズムが適用可能であることにより、Act System で使用する 3 次元形状を凸多面体に限定する必要がなくなる。

物理計算サブシステム

物理計算部は、3 次元形状同士が衝突した時に発生する力の算出、3 次元形状にかかる力から加速度・速度等の算出、を行う^[35]。Physically-based 手法に基づき、3 次元形状の位置姿勢を計算する役割を果たす。

データベース

データベースには、3 次元形状データのファイル、仮想機器の情報を保存したテキストファイル等が格納されている。これらデータベース内の情報は、Act System 開始時に管理サブシステムにより計算機のメモリ上に自動的に読み込まれる。

本来、要求仕様 7 に従うのならば、ActCube は Act System の内部に組み込まず、ActCube 開発のためのスクリプトを用いて、ファイルの形で外部定義できることが望ましい。しかし、本研究では、スクリプト定義の労力が大きいことから、スクリプトの定義を将来課題とし、ActCube をシステム内部に組み込む形で提供する。

表 4.1: 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

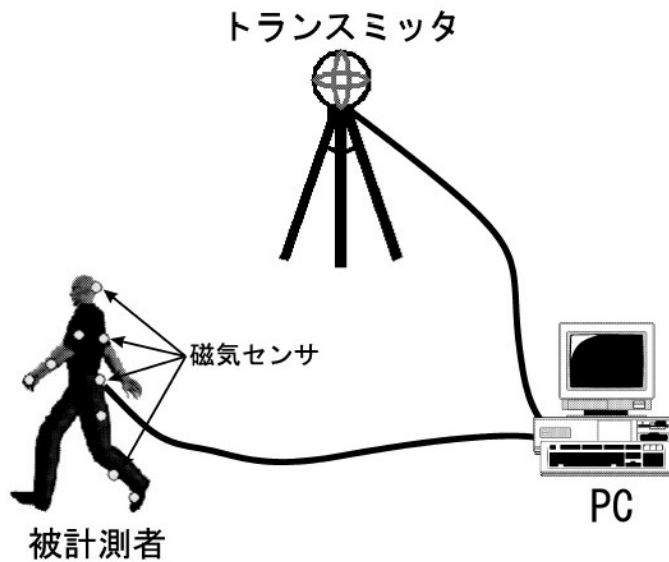


図 4.2: モーションキャプチャシステム装置構成



図 4.3: 手の開閉情報を取得するためのボタンデバイス

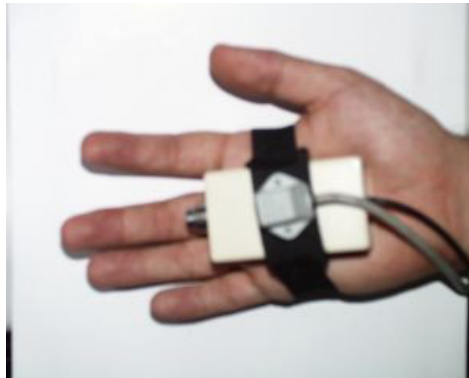


図 4.4: ボタンとセンサ

4.2.2 ハードウェア構成

3次元環境下で仮想空間を構築可能にするための利点は、日常生活で習得しているメタファで3次元オブジェクトを操作できることであり、この利点を活用するためには、実世界のジェスチャがそのまま3次元環境下に反映されることが望ましい。そのため、入力デバイスとして、3次元マウスやジョイパッド等は、3次元データを計測できるが、実世界のジェスチャと大きく異なり有効でない。

そこで、本研究では、人体動作を入力値とするために、3次元デバイスとして、センサのリアルタイムポジションのデータを取り込むことができる磁気式3次元空間モーションキャプチャシステム (Polhemu 社:ULTRATRAK)^[36] を使用し、ユーザの手の位置情報を取得することにした。

ULTRATRAK は、図 4.2 に示すように、磁界を発生するトランスミッタ、磁気を計測する磁気センサ、専用の計算機、という装置構成をしており、トランスミッタを基準としたセンサの角度と位置を計測する。また、ULTRATRAK は、ネットワークを通じて、他のコンピュータに計測データを随時送信できる。センサの位置情報の計測間隔は1/60秒であり、手の位置情報の取得がボトルネックとなり仮想空間の描画更新が遅延する恐れはない。平均平方誤差 (以下、RMS) は、トランスミッタからセンサの距離が約 1.5m(5 feet) で、角度に対しては3度、位置に対して3cm(1 inch) であり、誤差も十分小さいと言える。ULTRATRAK の基本性能を表 4.1 に示す。

また、図 4.3 に示す ON/OFF が入力できるボタンデバイスを開発し、ユーザに手を握った状態ではボタンを押印してもらうことにより手の開閉状態を計測する。このボタンデバイスのデータは、serial port からコンピュータに取り込める。

本研究では、このように手の開閉情報のみをユーザの指が持つ情報として利用する。現実には、指の持つ情報としては、手が物体を把持する状態としても、ON/OFF だけでは表現できず、軽く握る・強く握る・平行移動をしやすい形で握る等、さまざまな形態がある。より対話性の高い仮想空間を提供するためには、これらの情報を正確に読み取る必要がある。しかし、そのためには、指の動きを非常に正確に計測できるデバイスと、その指の情報から手の状態を分析するシステムが必要になり、開発労力が大きい。そこで、本研究では、その指の情報の活用は将来課題とし、指の情報としては、手の開閉情報のみを利用している。

結局、Act System 利用中、両手の位置情報・開閉情報を計測するために、ユーザは、両手に図 4.4 に示すようなセンサとボタンデバイスを組み合わせたものを持つことになる。

4.2.3 インタフェースの構成要素

図 4.5 に、Act System の構築インタフェースを示す。Act System のインタフェースは、次の要素で構成される。

- 仮想空間の構築と、シミュレーションテストを行うためのワークスペース
- 3次元形状、ActCube を操作するためのハンドカーソル
- 仮想部品を構成するための3次元形状
- 3次元形状の振る舞いを定める ActCube
- ActCube の読み込みや仮想機器の保存のための Widget 類

ワークスペース

ハンドカーソルで3次元形状・ActCube を操作するための中央の場所を「ワークスペース」と呼ぶ。

Act System を利用するユーザは、このワークスペースに没入する。仮想空間の構築と、構築した仮想空間をテストシミュレーションする作業の両方が、このワークスペースで行われる。仮想空間の構築作業中は、ワークスペース内は無重力の状態である。

ハンドカーソル

ハンドカーソルは手の形状をした仮想物体で、右手・左手に対応して2つ存在するユーザの仮想の手である。ハンドカーソルはユーザの右手・左手それぞれの動きに追従してスペース内を自由に移動する。また、ハンドカーソルは普段は手を開いた状態であるが、ユーザが手を握ると、ハンドカーソルも手を握った状態になり、逆にユーザが手を開くとハンドカーソルも再び手を開いた状態に戻る。ハンドカーソルが3次元形状や ActCube に触れた状態で手を握るとそれらを掴み、ハンドカーソルに追従して空間内を移動させることができる。

3次元形状

Act System を利用するためには、仮想部品を構成する3次元形状が必要であり、構築者は Act System を起動する前に、3次元形状ファイルを dxf 形式等で用意しておき、データベースに登録しておく。

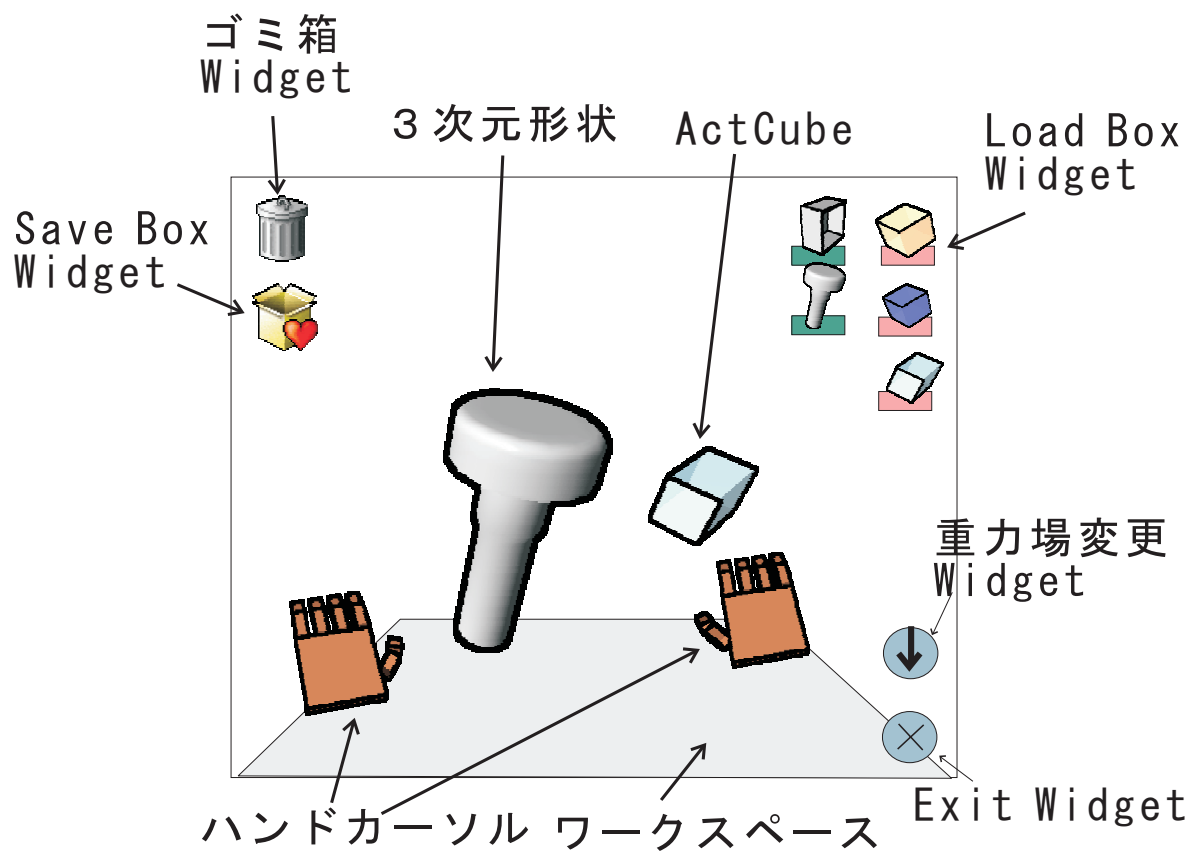


図 4.5: Act System のインターフェース

Act System では、構築作業中、構築者は、3次元形状を自由に移動させることはできるが、形状自体を拡大縮小させるような変形作業はできない。

ActCube

3.3節で説明したように、3次元形状に拘束関係を設定するための3次元オブジェクトである。構築者は、この ActCube を3次元形状において拘束関係が発生する箇所を覆うように、大きさを調節して、付着させる作業を行う。

Widget 類

Widget 類は、Act System インタフェースの最前面に表示される。2次元 GUI におけるアイコンに相当する。構築者は、Widget を握ったり、Widget を握ったままワークスペース内へドラッグすることにより、Act System の終了、ActCube のワークスペースへの読み込み作業等が行える。Act System のインタフェースには、次の Widget が配置されている。

1. Exit Widget

ハンドカーソルで握ることにより、Act System を終了できる。

2. 重力場変更 Widget

ハンドカーソルで握ることにより、構築モードとテストシミュレーションモードを切り替えることができる。テストシミュレーションモードでは、3次元形状に重力が働くようになり、また、ActCube は見えなくなり構築者は操作できなくなる。

3. Load Box Widget

Load Box Widget には、ActCube 用、3次元形状用、仮想機器用の3種類が存在する。それぞれ、Load Box Widget を握ったまま、ワークスペース内にドラッグすることにより、ワークスペース内にその Widget が保有する3次元オブジェクトの複製が作成される。管理サブシステムは、起動時データベースをスキャンし、3次元形状、仮想機器のファイルを発見すると、そのファイルに対応した Load Box Widget を自動的に画面上に作成する。

4. Save Box Widget

構築した仮想機器を保存したい場合等に、Save Box Widget を使用する。仮想機器を保存するためには、仮想機器をハンドカーソルで Save Box Widget 上に移動させ、ドロップする。この操作により、データベースに仮想機器のデータがテキストファイルとなって出力される。

5. ゴミ箱 Widget

ゴミ箱 Widget 上に 3 次元オブジェクトをドロップすることにより、仮想機器や ActCube を削除できる。

Act System では、以上の Widget を標準装備したが、以下の Widget を実装できれば、システムの作業効率がさらに向上することが期待できる。

1. 複製 Widget

仮想機器等をこの Widget 上にドロップすることにより、複製がワークスペース内に作成できる。

2. TrackBall Widget ・ズーム Widget

ワークスペースを見る方向を変更したい時に TrackBall Widget を握り、ドラッグすることにより、そのドラッグの移動量に応じて、ワークスペースが回転する。また、小さな仮想部品を扱う時等は、ズーム Widget を握ることにより、ワークスペースに対するズーム比率を変更できる。

4.3 3次元環境下での直接操作について

本節では、まず、Act System における位置情報のキャリブレーション方法について説明し、その後、3次元環境下での直接操作を円滑に行うために使用している操作手法について述べる。

キャリブレーション

ULTRATRAK から送信されてくる位置情報は、ULTRATRAK のトランスミッタを中心としたワールド座標系のデータである。そこで、このデータをワークスペース内

のローカル座標系に変換するための補正情報を導出するために、Act System は起動時に一度だけキャリブレーションを行う。

キャリブレーション方法としては、図 4.6 に示すようにユーザに両手を広げてもらった状態で、その状態で両手同時にボタンを押してもらい、両手を広げた状態での右手・左手の世界座標系での位置情報を獲得する。Act System は、ユーザの右手・左手を結ぶ直線の中点をユーザの体の中心位置と仮定して、両手の位置情報を元にワールド座標系における体の中心位置を計算する。また、中心位置と手の距離を計算する。

Act System は、ワークスペースにおける初期状態でのユーザの仮想の体の中心位置、及び、仮想の手の長さが設定してあり、この情報とキャリブレーションによって得られた実際の体の位置情報と手の長さを照らし合わせて、補正情報を算出する。より詳しい説明は付録 C. に譲る。

キャリブレーションを実行後、ULTRATRAK から送信されてくる情報は、この補正情報を元にワールド座標系のデータからワークスペースにおけるローカル座標系のデータに変換される。

操作補助手法

3次元環境下での直接操作は実世界の知識が活用できる一方で、一般的に、VR 特有の問題として以下の問題が発生する。

1. 人間は空間指示能力が低く、実生活においては触覚・力覚がこの問題を補い、実物体との正確な接触を可能にしているが、3次元環境下での作業はフォースフィードバックがかからないため操作性に難があり、精密な指示が困難となる

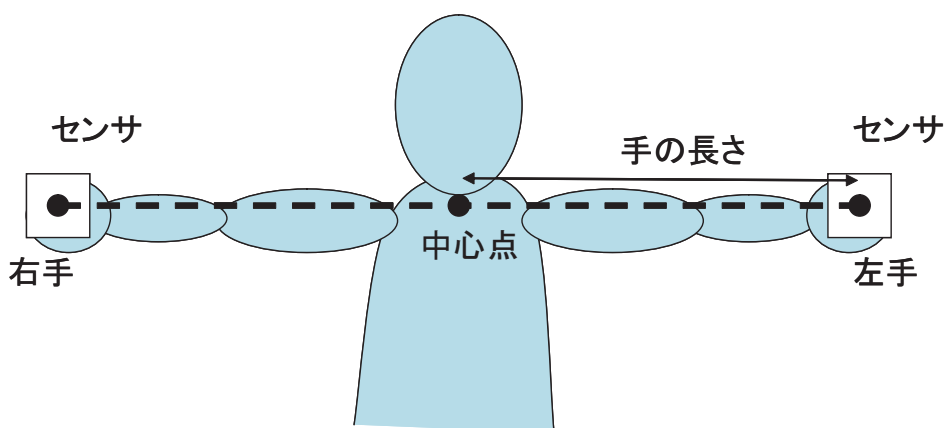


図 4.6: キャリブレーション時のユーザの姿勢

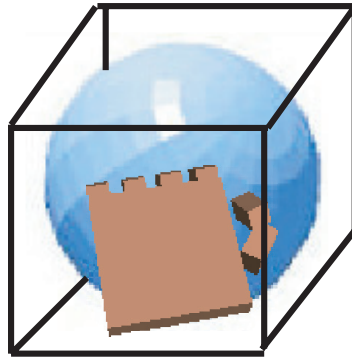


図 4.7: 接触によるハイライト表示

2. 3次元環境下では、実物体を観察する場合に比べて距離感の把握が困難であり、特に、奥行き情報が知覚しにくい

本研究では、このVR特有の問題の解消を目指し、操作補助手法として、従来研究により有効と報告されている、離散配置制約・接触情報の提示・影の提示、を実装している。

離散配置制約 問題1を改善するために、3次元オブジェクトを連続的に移動させるのではなく、2次元描画ソフトにおけるスナップと同様の考え方に従い、3次元オブジェクトを仮想の格子に沿わせる、または他の3次元オブジェクトに自動的に接続する、といったように、離散的な配置に制限することにより、視覚的に3次元環境下での直接操作を補助する研究が行われている^{[18][37]}。Act Systemでも、このアイデアを用い、3次元オブジェクトの位置を離散的に制限する。

接触情報の提示 3次元環境下でハンドカーソルが3次元オブジェクトに接触した時に、接触したことを視覚的に提示するために、図4.7に示すようにハンドカーソルが接触している3次元オブジェクトをワイヤフレームのバウンディングボックスで囲み、ハンドカーソルが接触している3次元オブジェクトにハイライトが当たるようにしている。

影の提示 問題2の状況を改善する方策として、影を提示する手法が有効である。そこで、本研究でも、ワークスペースの床に格子状に線を描画し、3次元オブジェクトの影を提示し、奥行き知覚を補助している。

4.4 仮想空間の構築について

本節では、仮想空間構築時に構築者が行える操作について述べた後、実際に仮想機器を構築する際の流れについて説明する。

4.4.1 仮想空間の構築者が実行可能な操作

仮想空間の構築者は、4.2.2 項で述べたセンサとボタンが組み合わせたデバイスを持ち、ジェスチャを行うことにより、3次元オブジェクトである3次元形状と ActCube を両手を用いて直接操作する。構築者は、構築作業中、ハンドカーソルで3次元形状、ActCube にインタラクションでき、また、必要ならば、Widget 類への操作を行う。

Act System における構築者の操作を、片手作業、両手作業という観点から分類する。

1. 片手作業

片手作業では、3次元形状と ActCube を移動させる作業が実行可能である。特に、ActCube を移動して3次元形状上に配置させることにより、3次元形状に ActCube を付着させる作業が実現できる。また、Widget 類への操作は片手作業で行う。

2. 両手作業

両手で行える作業は、ActCube の拡大縮小作業と、ActCube のひねり作業である。また、ActCube と3次元形状を近づける作業等は、両手による協調作業で実行可能である。

以下、これらの作業を詳しく説明する。

3次元オブジェクトを移動させる作業 3次元オブジェクトをハンドカーソルで把持し、移動させる際は、まず、ハンドカーソルをその3次元オブジェクトと接触する場所まで移動させる。ハンドカーソルと3次元オブジェクトの接触が検知されると、3次元オブジェクトにハイライトが当たる。その状態で、手を握れば、3次元オブジェクトを把持でき、3次元オブジェクトの移動が可能になる。3次元オブジェクトを握っている状態で、手を開けば、3次元オブジェクトを離し、ハンドカーソルは初期状態に戻る。この様子を図 4.8 に示す。

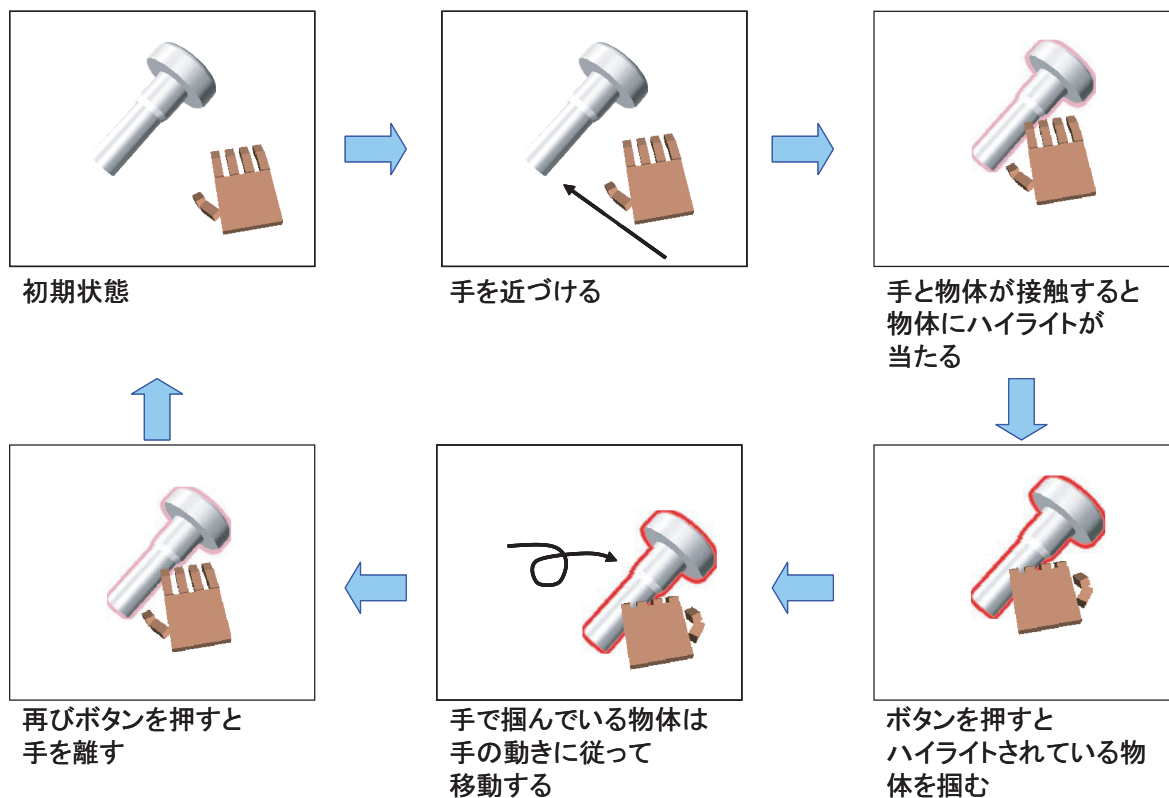


図 4.8: 3次元オブジェクトの移動

ActCube を 3次元形状に付着させる作業 ActCube を 3次元形状に付着させる作業は、ActCube を移動させる作業を応用して実現される。ActCube を付着させる時は、図 4.9 に示すように ActCube を 3次元形状上に移動させ、そこで、手を離す操作を行う。ActCube は、手の把持から開放された瞬間、接触している 3次元形状が存在していれば、自動的にその 3次元形状に付着する。3次元形状に付着した ActCube は、引き離されるまで 3次元形状の移動に付随し続ける。

また、逆に 3次元形状に付着している ActCube を 3次元形状から引き離す操作は、図 4.10 に示すように、付着している ActCube をハンドカーソルで把持し、3次元形状と接触しない場所まで解放することにより完了する。

ActCube の拡大縮小作業 ActCube が 3次元形状の拘束関係箇所を覆う適切大きさにない場合、その大きさを調節するために ActCube の拡大縮小作業を行う必要があり、その拡大縮小作業は、両手を用いて実行される。図 4.11 に示すように、両手で 1つの ActCube を持ち、両手の間隔を広げると ActCube は拡大縮小する。しかし、3次元デバイスの操作の困難性から、適切な大きさに ActCube を調節する作業は非常に難しい。そこで、ActCube の大きさは、連続的な値をとらず、離散的な数値しか持ち得ないよ

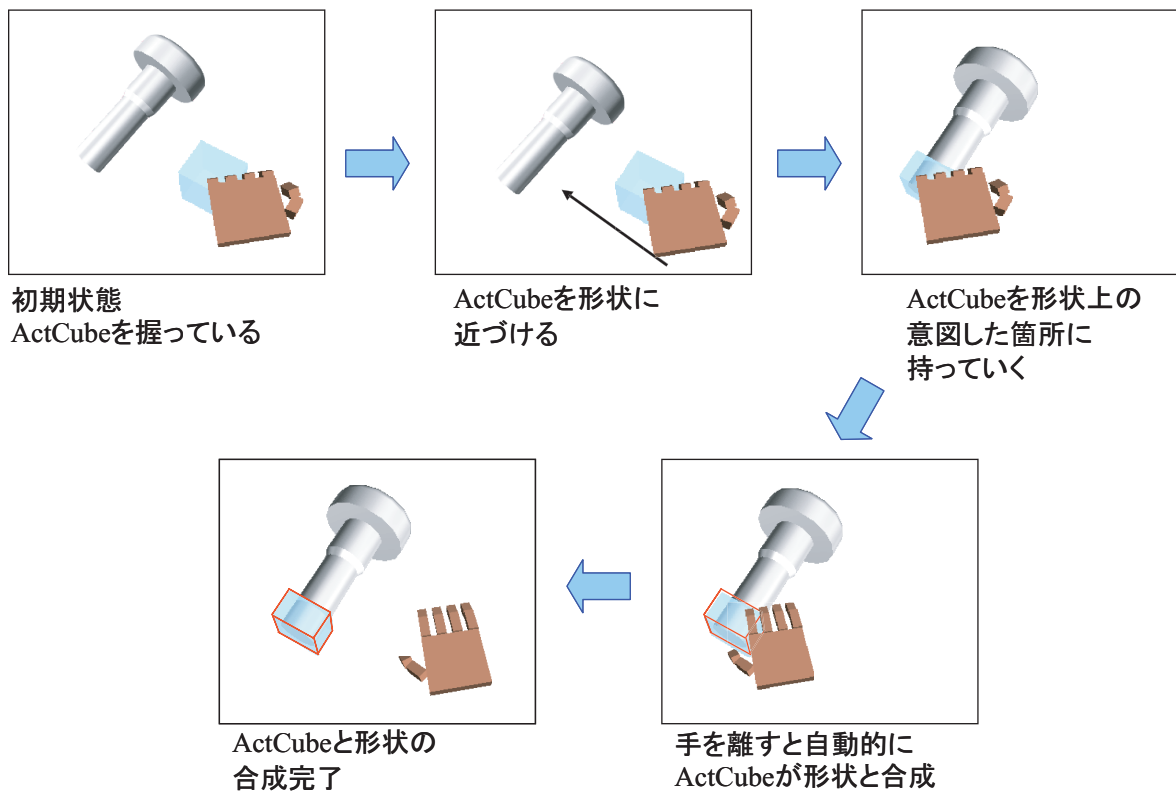


図 4.9: ActCube を 3 次元形状に付着させる作業

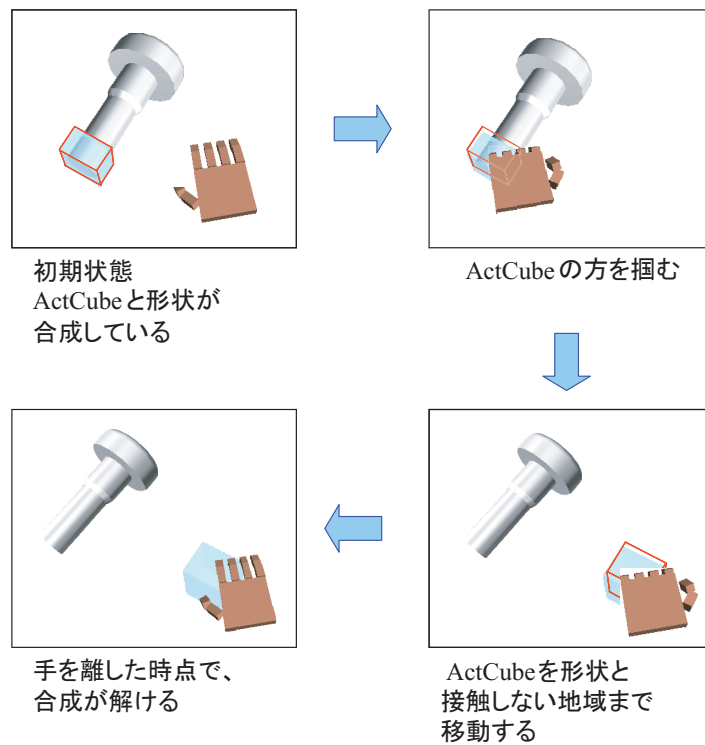


図 4.10: ActCube を 3 次元形状から引き離す作業

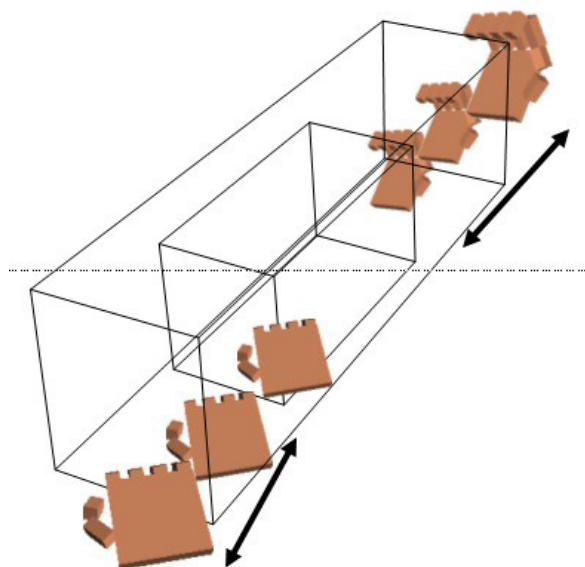


図 4.11: ActCube の拡大縮小

うに制約することとした。その ActCube の大きさを離散的に制御する際の具体的な値は、仮想空間に存在する 3 次元形状の大きさに影響を受けるため、一概に定義できないが、ユーザが使いやすい値に適切に設定する。

ActCube のひねり作業 ActCube は、図 3.14 に示すように特有の形状が重畳されていることがある。ActCube の種類によっては、両手での ActCube をひねる操作により、その重畳形状を変化させることができる。重畳形状は、ActCube の固有パラメータとして関連付けられていることがあり、凹凸結合後の仮想物体の挙動制御にそのパラメータは影響を与える。例えば、蝶番 ActCube であれば、回転を許容する角度の範囲がこの重畳形状と結びついている。両手でのひねり作業は、図 4.12 のように実現される。また、拡大縮小作業同様、ひねりの角度も連続値でなく、離散値に制限される。その ActCube のひねり角度を離散的に制御する際の具体的な値は、3 次元デバイスの操作性に影響を受けるため、一概に定義できないが、ユーザが使いやすい値に適切に設定する。

両手での協調作業 両手操作を実現しているため、ActCube と 3 次元形状を近づける作業を行う際に、右手で ActCube、左手で 3 次元形状を持ち、近づけるといった両手協調作業が実行可能であり、作業効率の向上が期待できる。

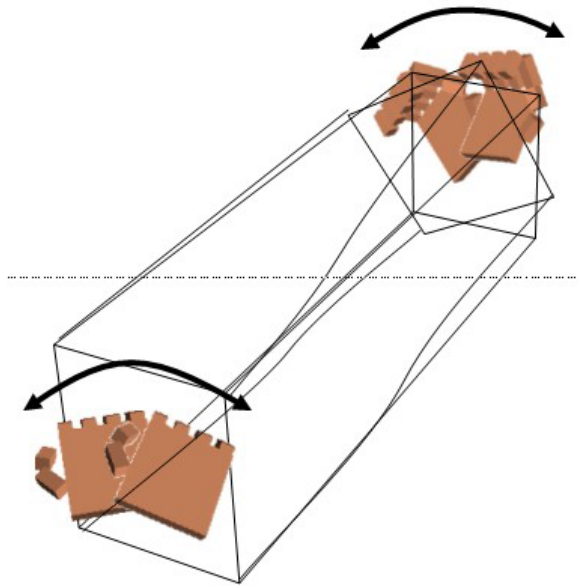


図 4.12: ActCube をひねる作業

4.4.2 仮想機器構築の手順

実際に仮想機器を構築するために必要な構築作業手順について述べる。

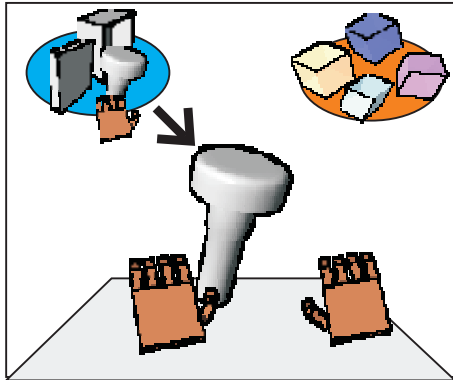
Act System を用いて仮想機器を構築するためには、以下の 2 つの作業を行う必要がある。

- a. ActCube を 3 次元形状に付着させて仮想部品を定義する作業
- b. 仮想部品を持ち寄り、ActCube の凹凸結合機能を利用して 3 次元形状同士を結合する作業

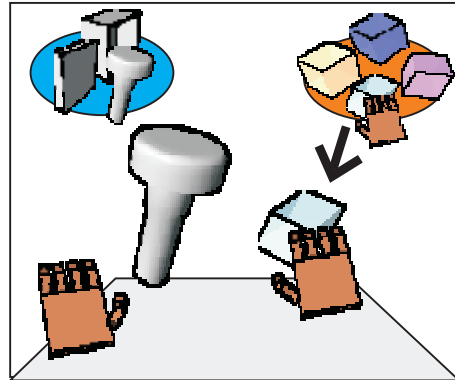
a. は 4.4.1 項で述べた操作方法により実行される構築作業に特有の操作である。また、ActCube は、仮想機器構築中であっても、3.3.2 項で述べた凹凸結合機能、挙動制御機能等を果たす。つまり、重力はかからないが、構築中もシミュレーション訓練中の訓練生の操作と共通する作業を行える。b. はその凹凸結合機能を利用して、3 次元形状同士を結びつける作業である。

実際の構築作業手順としては、以下の通りになり、図 4.13 にその様子を示す。1～6 は a. に属し、7～8 は b. に属する作業である。

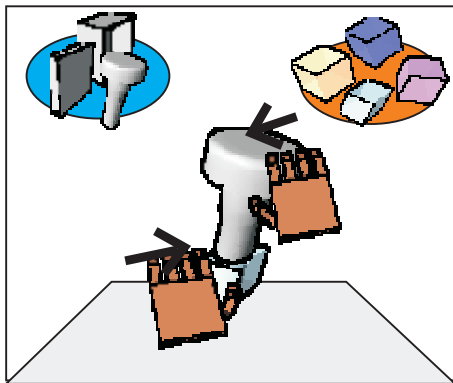
1. 必要な 3 次元形状をワークスペースへ読み込む
2. 必要な ActCube をワークスペースへ読み込む



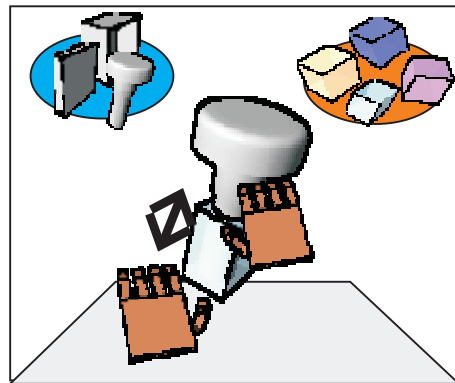
① 3次元形状を読み込む



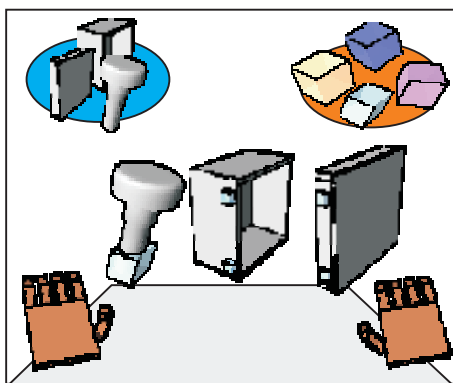
② ActCube を読み込む



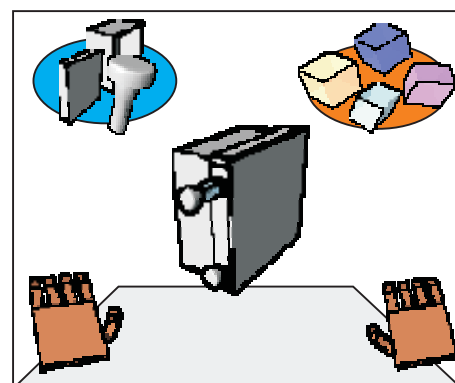
③ 形状に ActCube を付着



④ ActCube のサイズを調節



⑤ 仮想部品を用意



⑥ 仮想機器を組み上げる

図 4.13: 仮想機器の構築手順

3. 3次元形状上に ActCube を移動させて 3次元形状に ActCube を付着する
4. 両手を用いて ActCube の大きさを調節する
5. 2~4を繰り返し、必要な ActCube を全て 3次元形状に付着して、仮想部品を定義する
6. ActCube が付着された仮想部品を一旦ファイルに保存する
7. 仮想部品をワークスペースに全て読み込む
8. 仮想部品を ActCube を介して結合させて、仮想機器を組み立てる

機器の組立作業を実現したい時は、仮想部品を用意した時点で構築終了であり、解体作業を行いたい時は、仮想部品を全て組み上げて仮想機器が構築できた時点が構築終了となる。

4.5 仮想空間のシミュレーションについて

本節では、まず、Act System のシミュレーション中に訓練生が実行できる操作について述べる。その後、ActCube が存在する場合の仮想物体の挙動計算シミュレーションの処理の流れを述べる。

4.5.1 シミュレーション中の訓練生が実行可能な操作

シミュレーション時、ワークスペース内の作業等で構築時と大きく異なる点は、以下の2点である。

1. ActCube がワークスペース内に描画されなくなり、ハンドカーソルがインタラクションできる対象が 3次元形状だけになる
2. 3次元形状に重力がかかり、自由落下する

ActCube は描画されなくなるが、ワークスペース内には実際には存在しており、凹凸結合機能や挙動制御機能により 3次元形状を物理的拘束関係に制御する働きをする。また、3次元形状に対して実行可能な操作は、移動作業だけであり、この移動作業は、前節で述べた構築中の 3次元オブジェクトの移動作業と同じ方法で実現される。

4.5.2 シミュレーションの流れ

ここでは、図 4.14 のフローチャートを用いて、Act System シミュレーション時の処理の流れを説明する。

1. ユーザが、仮想空間中の 3 次元形状を選択し、その 3 次元形状を移動させる。
2. 操作された 3 次元形状に付着している ActCube(以下、名前を ActCube[I] と呼ぶ)が、他の 3 次元形状に付着している ActCube と結合中であるか調べる。
3. 他の 3 次元形状に付着している ActCube(以下、名前を ActCube[II] と呼ぶ) と結合していれば、ActCube[I] にユーザの操作情報(移動量)を送信する。ActCube[I] は、ActCube[II] と協力し、そのユーザの移動量を拘束関係に応じた適切な移動量に変換し、その値を ActCube[I][II] が付着している 3 次元形状の位置姿勢に反映させ、1. に戻る。
4. 結合中の ActCube が無ければ、ユーザの操作情報に基づく 3 次元形状の位置と姿勢を求め、その位置と姿勢で新たに他の ActCube 同士の接触が起こるか判定する。
5. 4. の結果、新たに ActCube 同士の接触が起これば、その種類と断面形状を調べ、結合の条件が成立すれば 2 つの ActCube を結合させ、1. に戻る。
6. 4. の結果、新たな ActCube 同士の接触が無ければ、Physically-based 手法で 3 次元形状の位置と姿勢を計算し、その結果を仮想空間に反映して 1. に戻る。

以上のステップを繰り返すことにより、ユーザの両手入力操作に応じた仮想空間内の仮想物体の位置と姿勢を更新し、仮想空間のシミュレーションを行う。

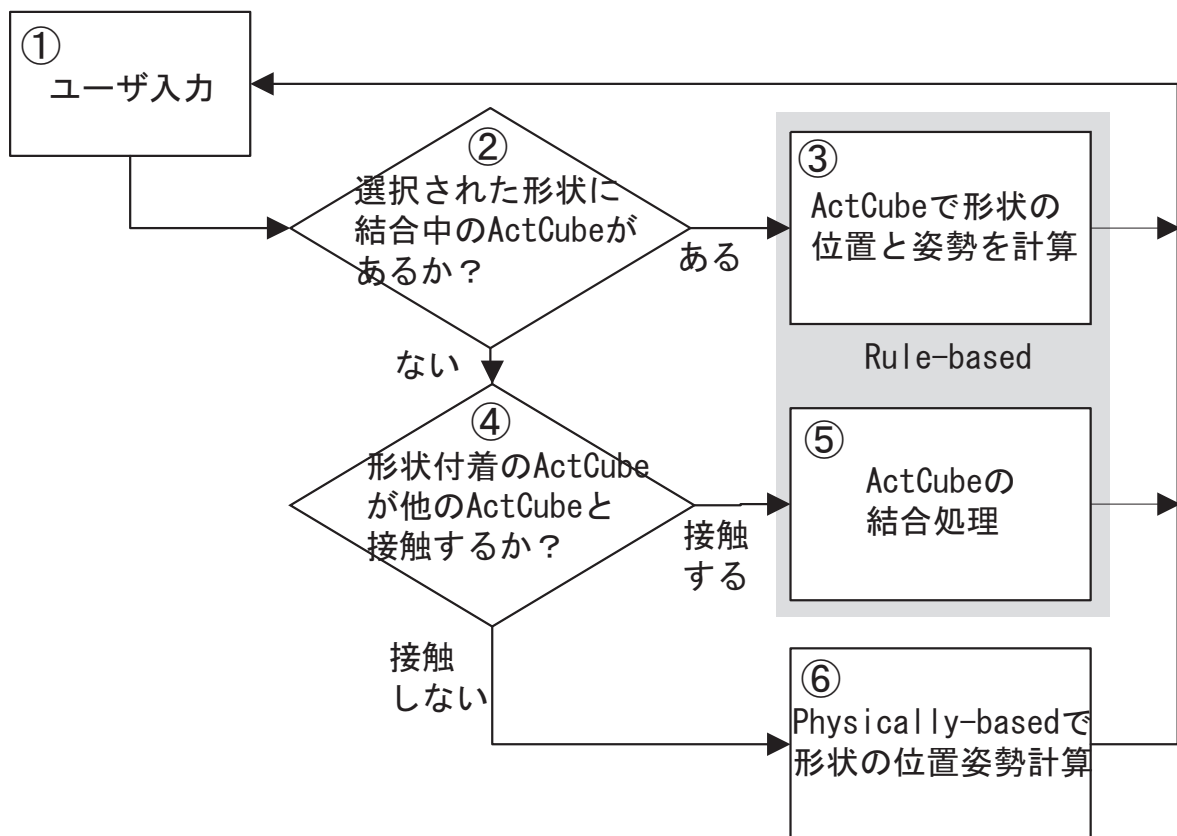


図 4.14: シミュレーションフローチャート

第 5 章 仮想空間の構築例及び Act System の評価

本研究では、3章で述べた ActCube により拘束関係の設定を行う手法を提案し、その手法に基づき、4章で述べた仮想空間管理シミュレーションシステム Act System を開発した。

本章では、まず、Act System を用いて構築した仮想空間の構築例を示す。次いで、ActCube 手法の有効性と Act System の操作性の評価のために行った被験者実験について述べる。

5.1 仮想空間の構築例

本節では、まず、密接拘束関係を実現するために開発した代表的な ActCube の役割やその使用例を示した後、Act System のデモンストレーションとして構築した仮想機器について述べ、Act System での仮想空間の構築労力等を検討する。

5.1.1 開発した ActCube

平行移動 ActCube 平行移動 ActCube は、3次元形状が他の3次元形状により平行移動の動きに拘束される場面で用いられ、引き出しの動き等を実現するために使用される。平行移動 ActCube の外観は、図 5.1 に示すように、直方体の上に両方向矢印が重

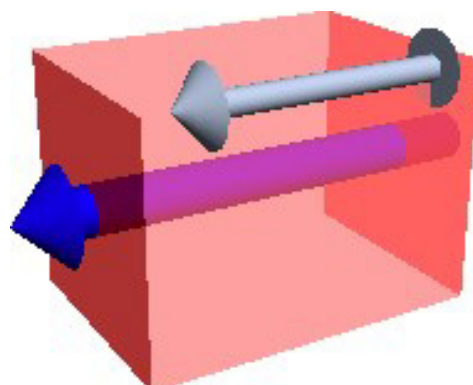


図 5.1: 平行移動 ActCube の外観

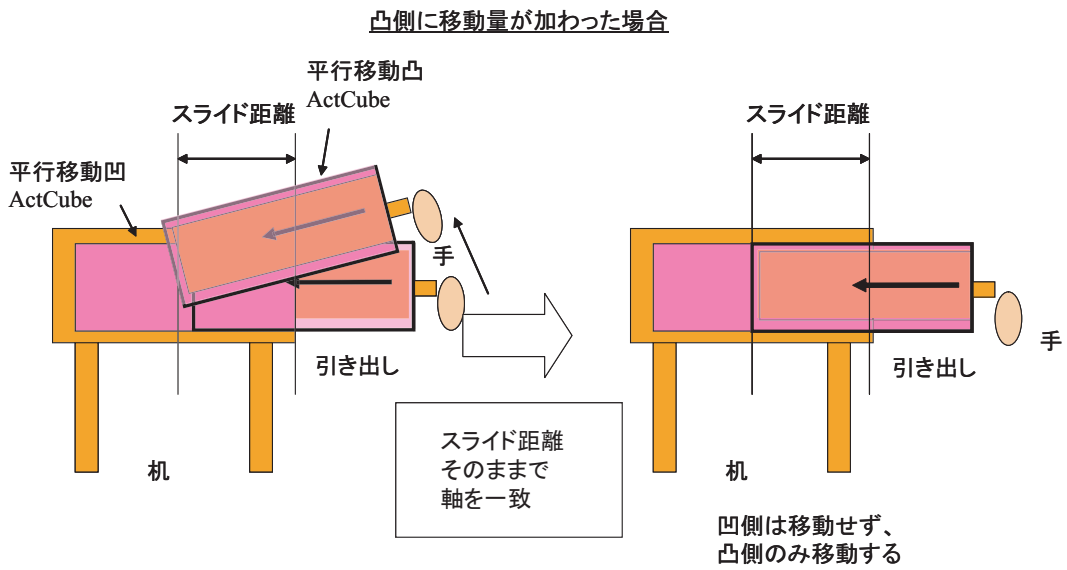


図 5.2: 平行移動凸 ActCube における挙動制御方法

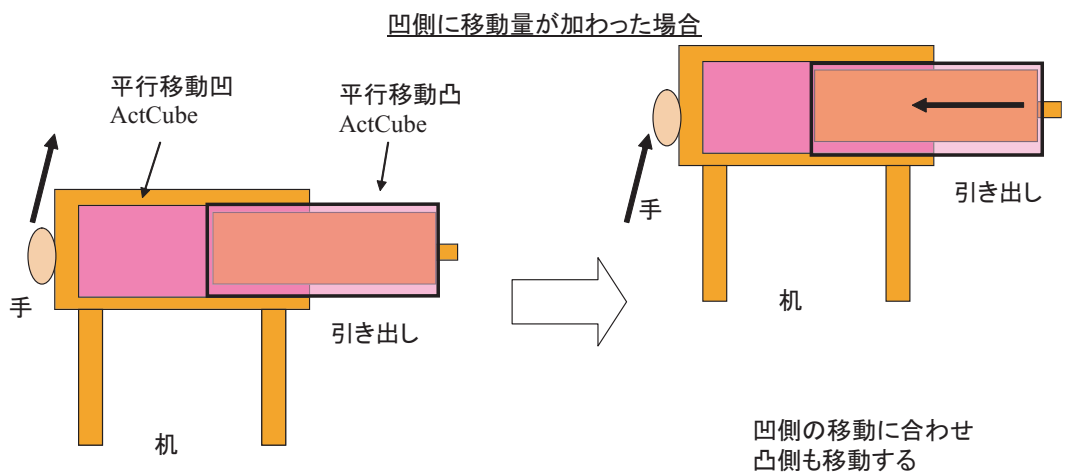


図 5.3: 平行移動凹 ActCube における挙動制御方法

置されている。平行移動 ActCube では、平行移動させられる 3 次元形状に凸 ActCube が付着し、平行移動させる 3 次元形状に凹 ActCube が付着する。引き出しの例で言えば、凸 ActCube は引き出し側、凹 ActCube は机側に付着させる。

凹凸の ActCube 結合時、凸 ActCube 側の 3 次元形状に位置情報が伝達されると、図 5.2 に示すように、その位置情報から凸 ActCube の凹 ActCube の軸方向にそったスライド距離を算出し、そして、その値を逆算し、凸 ActCube 側の 3 次元形状の移動量を求める。

逆に、凹と凸の ActCube 結合時、凹 ActCube 側の 3 次元形状に位置情報が伝達されると、図 5.3 に示すように、凹 ActCube の移動によって、凸 ActCube も移動させる。

凹と凸が結合した際は、スライド距離が 0 になる位置に凸 ActCube を移動させる。また、凸 ActCube の移動により、スライド距離が 0 以下になると結合が切れる。

蝶番 ActCube 蝶番 ActCube は、3 次元形状が他 3 次元形状の軸を中心とする回転角度が 360 度以内の軸回転に制限される場面で使用され、代表的な場面としてはドアの動き等がある。蝶番 ActCube の外観は、図 5.4 に示すように、直方体の上に扇形状が重畳されている。回転させられる方の 3 次元形状に凸 ActCube が付着し、回転軸を有する方の 3 次元形状に凹 ActCube が付着する。ドアの例で言えば、凸 ActCube はドア側、凹 ActCube はドアを固定する壁側に付着させる。重畳している扇形状は、3 次元形状を制限する回転角度に対応しており、両手によるひねり作業により調節できる。ただし、回転角度のパラメータとして使用されるのは、凹 ActCube 側のパラメータのみである。

凹と凸の ActCube 結合時、凸 ActCube 側の 3 次元形状に位置情報が伝達されると、図 5.5 に示すように、その位置情報から凸 ActCube が凹 ActCube の軸中心に回転する角度を算出し、そして、その値を逆算し、凸 ActCube 側の 3 次元形状の移動量を求める。なお、回転する角度が、凹 ActCube 側の制限角度より大きい時は、回転角度を制限値を超えないように調節する。逆に、凹と凸の ActCube 結合時、凹 ActCube 側の 3 次元形状に位置情報が伝達されたると、平行移動の場合と同じように、凹 ActCube の移動によって、凸 ActCube も移動させる。

凹と凸が結合した際は、軸を一致させるように凸 ActCube を移動させる。また、蝶番 ActCube は、シミュレーション中は、結合が切れることはない。

ねじ ActCube ねじ ActCube は、名称通り、3 次元形状が他 3 次元形状の軸中心に固定されて、ねじの動作をする場面に使用される。ねじ ActCube の外観は、図 5.6 に示す

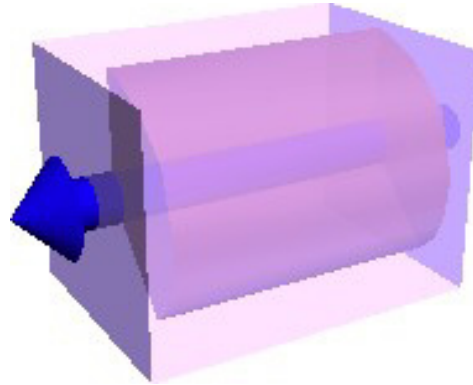


図 5.4: 蝶番 ActCube の外観

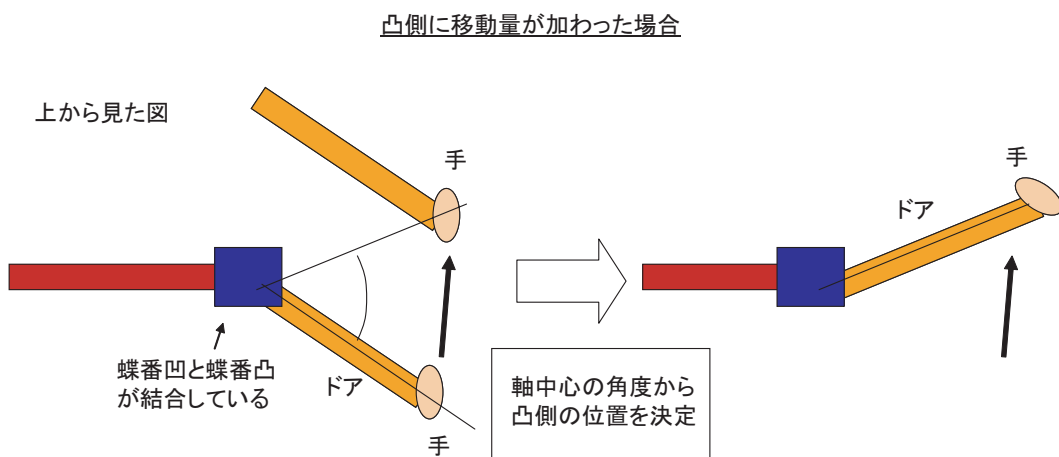


図 5.5: 蝶番 ActCube における挙動制御方法

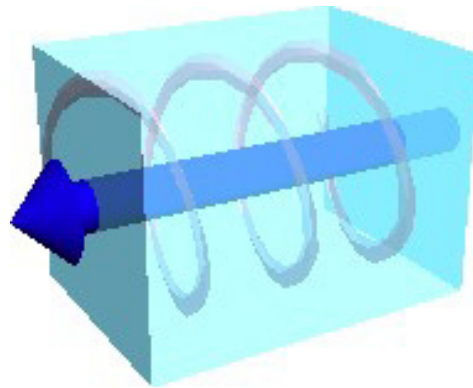


図 5.6: ねじ ActCube の外観

ように、直方体の上に螺旋形状が重畳されている。ねじの方の3次元形状に凸 ActCube が付着し、ねじ穴の方の3次元形状に凹 ActCube が付着する。重畳している螺旋形状は、ねじ一回転当たりでの締め度合いを表す。つまり、重畳されている形状の螺旋がゆるい時は、ねじを回す量が小さくても締まるが、螺旋がきつい時は、ねじを回す量が大きくなないと締まっていけない。

凹と凸の ActCube 結合時、凸 ActCube 側の3次元形状に位置情報が伝達されると、図 5.7 に示すように、その位置情報から凸 ActCube が凹 ActCube の軸中心の回転角度と軸方向の平行移動量を算出することにより、凸 ActCube 側の3次元形状の移動量を算出する。逆に、凹と凸の ActCube 結合時、凹 ActCube 側の3次元形状に位置情報が伝達されると、平行移動の場合と同じように、凹 ActCube の移動によって、凸 ActCube も移動させる。

結合時の処理、結合切断の条件は平行移動 ActCube と同一である。

把持 ActCube 把持 ActCube は、3次元形状において、手によって把持する箇所を明確に指定したい時に使用する。例えば、ユーザが引き出しを握る作業をする時、取っ手のみ握れるようにしたい場面に用いる。もし、この把持 ActCube が設定されていないと、手は引き出しのどの場所をも掴むことを許容されるが、把持 ActCube が3次元形状の一部に付着している時は、把持 ActCube によって指定された箇所しか握ることができない。図 5.8 に示すように、直方体の上に拳骨の形状が重畳されている。

また、把持 ActCube は、これまで述べてきた ActCube と大きく異なり、凹 ActCube しか存在しない。これは、ハンドカーソル自体が、凸 ActCube の役割を果たすためである。

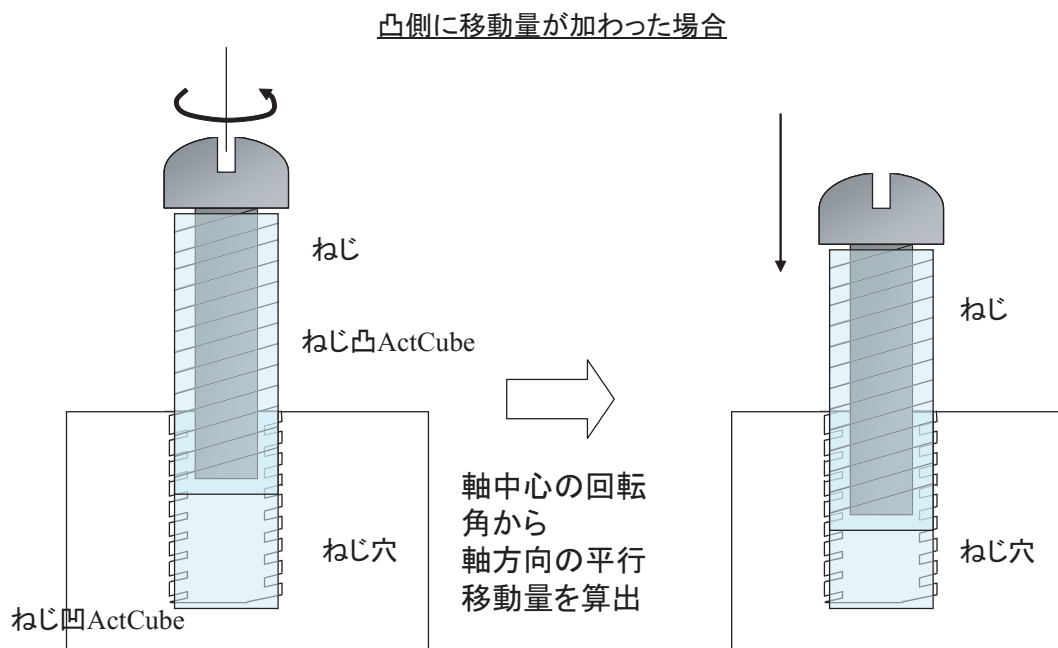


図 5.7: ねじ ActCube における挙動制御方法

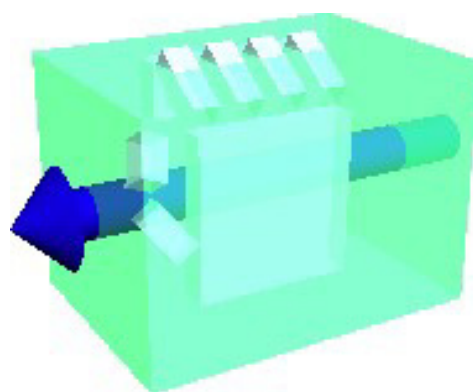


図 5.8: 把持 ActCube の外観

5.1.2 Act System を用いて構築した仮想空間

スウィング式逆止弁を対象に Act System を用いて構築した訓練環境について説明する。その後、その構築作業結果から、Act System の構築作業の労力等を評価する。

逆止弁の分解組立

Act System を用いた場合の仮想空間の構築にかかる労力等を評価することを目的として、原子力発電所の保守訓練機器の一つであるスウィング式逆止弁の分解組立作業を対象に仮想空間を構築した。スウィング式逆止弁は、図 5.9 に示すような構造をしており、分解組立作業の中で表 5.1 に示す範囲の作業工程を行えるように構築作業を行った。

スウィング式逆止弁の構築に必要な 3 次元形状および ActCube を表 5.2、表 5.3 に示す。構築に必要な 3 次元形状は既に作成されていて、ActCube も同様に既に開発されている状態から構築を始めた。構築の作業手順を表 5.4 に示す。スウィング式逆止弁の構築に要した時間は、30 分程度であり、短時間で済んだ。構築の様子を図 5.10 に、その構築したスウィング式逆止弁の実行画面を図 5.11 に示す。なお、Act System 開発者自身が構築を担当した。また、5.2 節で後述するが、本研究で用いた 3 次元デバイスの操作性が著しく悪かったため、ActCube の付着作業などを 2 次元マウスとキーボードで行えるように改良した Act System にて構築を行った。

評価

スウィング式逆止弁は、先行研究^[14]で開発された 2 次元 GUI による仮想空間構築支援システムを用いて構築されており、その結果と Act System で構築した結果を比較

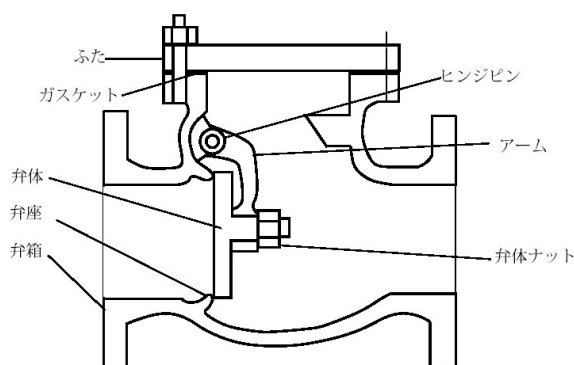


図 5.9: スウィング式逆止弁の構造

表 5.1: スウィング式逆止弁の分解の作業手順 (一部)

作業工程	作業内容
工程 1	ふたボルトのナットをレンチでゆるめる
工程 2	ふたを少し持ち上げ、内圧がかかっていないことを確認する
工程 3	ふたボルトからナットを全て取り外す
工程 4	ふたを取り外す

表 5.2: スウィング式逆止弁のために必要な 3 次元形状

1	逆止弁本体
2	ふた
3	ナット
4	レンチ

表 5.3: スウィング式逆止弁のために必要な ActCube

	ActCube の種類	働き
1	平行移動 ActCube	ふたを逆止弁のボルトにはめる
2	ねじ ActCube	ナットが逆止弁のふたボルトに締める
3	軸回転 ActCube	レンチでナットを回転させて締める

表 5.4: スウィング式逆止弁の構築手順

1	逆止弁のボルトに平行移動凹 ActCube を軸方向に拡大して付着
2	ふたの穴の位置に平行移動凸 ActCube を付着
3	ナット 4 つにねじ凸 ActCube を付着
4	逆止弁のボルト 4 つにねじ凹 ActCube を付着
5	ナット 4 つに軸回転凸 ActCube を付着
6	レンチに軸回転凹 ActCube を付着
7	ふたを逆止弁のボルトにはめる
8	ナットを逆止弁のボルトに締める

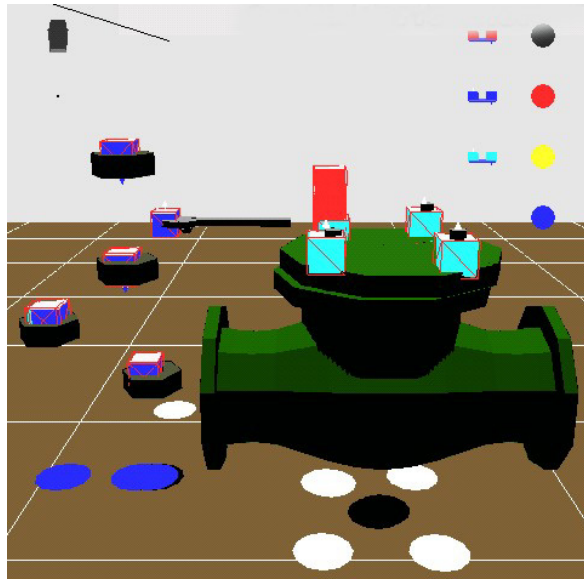


図 5.10: スウィング式逆止弁構築の様子

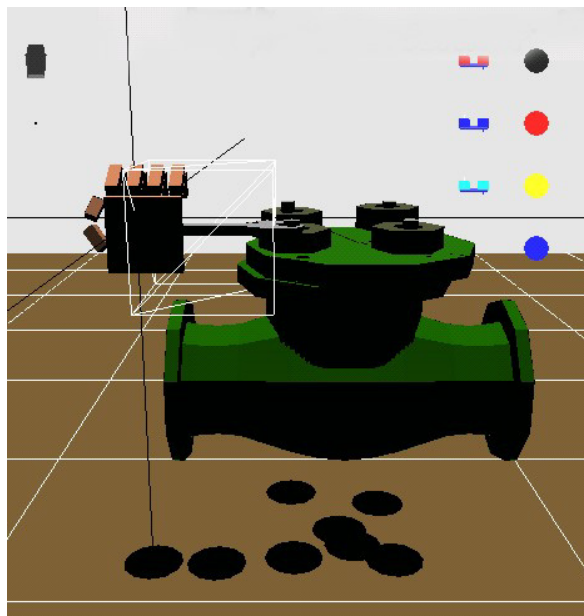


図 5.11: スウィング式逆止弁シミュレーションの様子

表 5.5: 仮想空間構築支援システムと Act System の対比

	仮想空間構築支援システム	Act System
シミュレーション手法	全て Rule-based 手法	Rule-based 手法で拘束関係を、それ以外の現象は Physically-based 手法
構築に必要な作業	ペトリネットによる仮想空間のモデル化	ActCube による拘束関係の設定
構築インタフェース	2次元 GUI	3次元環境での直接操作
逆止弁の構築時間	約 12 時間	約 30 分
実現している機能	物理的拘束関係、蓋にマーキングする作業、操作手順のモデル化	物理的拘束関係

し、Act System を構築労力等の点について評価を行う。

先行研究の仮想空間構築支援システムでは、構築者は、仮想空間内の作業をペトリネットという有向グラフを用いてモデル化し、そのペトリネットおよびその他の仮想空間内の物体に対する様々な情報を 2次元 GUI を用いて入力し、編集するだけで、目的とする訓練環境を柔軟に構築できる。

表 5.5 に、Act System および先行研究により構築された逆止弁の分解組立作業を実現するために必要となる作業や構築した仮想空間の特徴等についてまとめる。

構築作業時間に注目すると、構築作業時間が先行研究の 12 時間からわずか 30 分へと短縮し大幅な作業労力の時間削減につながっている。先行研究の手法では、仮想空間の仮想物体間の相互関係等をペトリネットという有向グラフを用いてモデル化する作業が必要であった。そのため、モデル化に関する専門知識が必要であり、モデル化作業に大きく時間を割かれていた。しかし、ActCube 手法を用いることにより、そのモデル化作業の束縛から解放され、直感的な作業だけで仮想空間を構築可能にしたことが作業効率向上の一因である。また、本研究では、構築の際に設定が容易である Physically-based 手法をシミュレーション手法の一部に取り入れたことも構築負荷削減に寄与した。

しかし、先行研究の支援システムは、ペトリネットという仮想空間全体を統括的に管理する内部モデルが存在するため、蓋にマーキングしなければ次のステップに進めないというように機器の操作手順の情報を含んだ形で訓練環境を構築することも可能

であった。Act System では、仮想空間全体を管理する内部モデルが存在しないため、訓練環境における操作手順を表現することはできない。そのため、Act System の訓練環境の構築で、操作手順を考慮していないことが、Act System における構築時間短縮の一因となってしまっていると考えられる。しかし、大幅に構築時間が短縮されていることから、それを差し引いても、Act Cube 手法は短時間で仮想空間を構築できることは明らかである。なお、Act System で操作手順を実現するためには、仮想物体の挙動を外部から監視し、現在の仮想空間の状況を把握するシステムを開発する必要がある。

また、Act System で開発したスウィング式逆止弁では、先行研究によって実現されていた鉛筆により蓋にマーキングする作業が実現できない。現段階では、Act System には鉛筆で仮想物体にマークをつける現象をシミュレーションする機構が組み込まれていないため実現できていないが、Physically-based 手法で鉛筆でのマーキングを実現する機構を組み込む、または、鉛筆でマークをつけるための ActCube を開発することにより、この問題は解決できる。

なお、Act System では、ハンドカーソルの位置を離散的な値をとるように制御しているため、正確な位置合わせが困難となっている。そのため、図 5.11 にあるように、蓋の位置と弁本体の位置が多少ずれている。3次元環境下での直接操作による作業では、正確な位置あわせは困難であり、3次元環境下で構築を進める場合、この問題は必ず発生してしまう。この問題を解決するためには、3次元環境下で構築を行うシステムを用意するだけでなく、それをサポートする2次元 GUI における構築システムも並列的に用意する策が考えられる。

Act System を用いた構築可能な仮想空間の可能性

本研究では、構築を行っていないが、Act System を用いてその他の様々な環境を構築することが可能である。例えば、機器保守作業として、モータの分解組立環境等が構築可能である。それだけでなく、自分の部屋の環境も Act System を用いて構築でき、応用すれば、仮想の都市も構築できる。

この Act System を用いて構築した仮想空間の使用用途としては、訓練環境への応用がまず第一に挙げられる。現段階では、力覚表示用のデバイスが使用されておらず、また、シミュレーション中も仮想物体の移動量を中心に仮想物体の制御を行っているため、機器の分解組立の手順や機器の構造習得のための使用に制限されてしまう。しかし、力覚デバイスを使用し、シミュレーション中も仮想物体間に働く力を基に仮想物

体の位置を制御するように改良することにより、機器保守作業の技能を習得できる仮想空間を実現できるはずである。

また、Act System で構築した仮想物体を、利用者が実際に操作できるインタラクティブな機器の 3 次元マニュアルとして利用可能である。

しかし、ActCube は、その付着先の仮想物体が変形して、3 次元形状に発生する物理的拘束関係が変化する可能性がある場面には対応することができない。つまり、仮想物体が変形するような仮想空間は、Act System を用いては構築できない。例えば、木材の切断等を含む大作業を Act System を用いて再現することは不可能である。

5.2 Act System を評価するための被験者実験

5.2.1 Act System の操作性を評価するための被験者実験

本項では仮想空間構築に対しては初心者の第三者に Act System を利用してもらい、ActCube 手法の理解のしやすさ、及び Act System の操作性に関する評価を行った結果について述べる。

実験の目的

被験者に Act System を用いて仮想機器を構築するタスクを課し、そのタスクにかかった時間、タスクの達成度、構築時の様子、および被験者へのアンケート結果から、4.2.2 項で述べた 3 次元デバイスの 3 次元環境下での操作性、及び、被験者の ActCube 手法の理解度を評価する。

実験の方法

- 被験者

被験者は、仮想空間の構築経験のない 20 歳代の男性 2 名とした。2 名とも、プログラミングの知識があり、キーボードやマウス等日ごろからコンピュータの操作には慣れている。しかし、ジェスチャ入力型の 3 次元デバイスは本研究で初めて使用する。

- 実験手順

被験者には、ActCube の概念、Act System の各機能の説明と操作方法を約 20 分

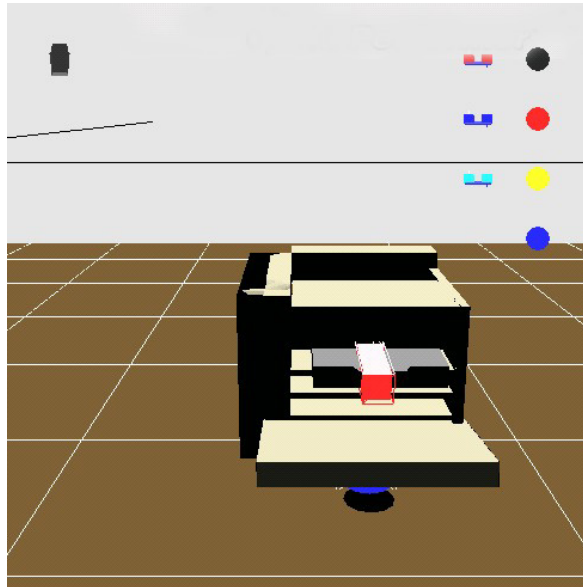


図 5.12: 被験者実験のプリンタ完成図

間説明し、開発済みの全ての ActCube における付着作業とシミュレーションの様子を見学させる。

その後、作業に慣れてもらうため、操作練習してもらい、操作に慣れた後、構築を行ってもらう仮想機器を説明し、作業を始めてもらう。

評価のために、作業開始から作業終了までにかかった時間を計測し、また、作成した仮想機器の内容も記録する。さらに、作業終了後、Act System を使用した感想を述べてもらう。

なお、必要な 3 次元形状データは予め作成されているものとする。

- 作業タスク

被験者には、トナーの交換作業が実行できるプリンタを構築してもらう。プリンタのトナー交換作業のためには、プリンタの前蓋を開閉する操作、トナーをトナー受けに出し入れする操作が実現できる必要がある。最終的なプリンタの完成図は、図 5.12 のようになる。3 次元形状として、プリンタ本体、プリンタ前蓋、トナーの 3 種類を用意しておき、この 3 次元形状に適切な ActCube に付着させて、トナーの交換が実現可能なプリンタが構築できた時点で作業終了とする。

この仮想機器を構築するためには、トナーの挙動の実現のために平行移動 ActCube、前蓋の挙動の実現のために、蝶番 ActCube を用いる必要がある。



図 5.13: 3次元デバイスを用いた被験者実験の様子

実験の結果

二人の被験者(被験者 A, 被験者 B)は共にタスクを終了した。トナー交換できるプリンタの構築に必要とした作業時間は、被験者 A では 14 分、被験者 B では 18 分であった。実験の様子を図 5.13 に示す。実験後、実施した被験者へのアンケート結果を表 5.6 に示す。このアンケート結果にもあるように、被験者の手の動きの逆方向にハンドカーソルが移動してしまう等、ハンドカーソルと被験者の手の動きの間に乖離があり、3次元デバイスの操作性は著しく悪かった。また、実験後、被験者 A の構築したプリンタは、軸回転 ActCube を付着させる方向を間違えているため、プリンタの蓋が実際とは異なる回転方向で開くようになっていることが判明した。実験後、被験者 A に理由を尋ねたところ、間違いには気づいていたが、操作性が悪いので、修正する作業が煩わしかったためであると回答した。

考察

被験者 2 名ともに、アンケート結果で述べているように、3次元環境下で自分の思い通りハンドカーソルを操作できなかったようであり、Act System の 3次元デバイスの操作性には難があった。また、アンケート結果にもあるように、ActCube が 3次元形

表 5.6: 3次元デバイスを用いた被験者実験のアンケート結果

被験者 A	被験者 B
操作性が悪すぎる。自分の思い通りに動かせないので、いらいらする。	疲れた。操作に難がある。
どうすればいいかは分かるけど、そのための操作ができなかった。	ActCube の働きは理解できた。
どの物体とどの影が対応しているのかわかりづらい。	ActCube が 3 次元形状に付着しているかどうかの判断を、いちいち付着させたであろう 3 次元形状を移動させるのがめんどくさい。

状に付着したことを視覚的に通知する補助手段が不足していたようである。

まとめると、Act System は、以下の 2 つの問題があることが分かった。

1. 実際の手の位置姿勢とワークスペース内におけるハンドカーソルの位置姿勢との不一致
2. 視覚的に操作を補助する手法の不足

まず、1. に関してであるが、Act System で用いた 3 次元デバイスは、フォースフィードバックがかからないことも原因として考えられるが、それ以上に被験者の手の位置計測のために使用している磁気式モーションキャプチャシステムの誤動作が主要因である。本実験では、モーションキャプチャシステムを室内で使用したことにより、室内にある PC から出る電磁波の影響や金属製品による電磁波の反射等が重なりキャプチャシステムが正確な手の位置を計測できなかったため、誤動作したようである。

手を平行に移動する操作は、手とハンドカーソルの不一致があっても、何度か移動を繰り返すことにより、その誤差の情報を推測、学習して、自分の思い通りにハンドカーソルが移動できたようだが、特に、手を回転する移動に関しては、誤差情報の推測が難しいようで、まったく自分の思い通りにならなかったようである。この問題を解決するためには、キャプチャシステムではなく、他のジェスチャ入力を計測できるデバイスを使用する必要がある。

また、2. に関してであるが、Act System では、ハンドカーソルが 3 次元オブジェクトと接触した時にその 3 次元オブジェクトのバウンディングボックスの表示、3 次元オ

プロジェクトの奥行き情報の認識を補助するための床への3次元オブジェクトの影の描画、を行っている。実験の様子から、これらが操作を補助するのに役立っていることは分かったが、ActCubeを3次元オブジェクトに付着させる作業を行う際は、この補助情報だけでは不十分であることが分かった。ActCubeを3次元オブジェクトに付着させる作業は、ハンドカーソルでActCubeを3次元オブジェクトと接触する位置まで移動させ、そこで手を離すことにより、自動的にActCubeが3次元オブジェクトに付着し完了するが、3次元環境内での操作では、奥行き情報の欠如等により、ActCubeを付着先の3次元形状に接触させたつもりでも、実は接触していないことや、3次元形状に付着させたつもりがなくても、ある3次元形状にいつのまにか付着していることが起こっていた。そこで、この問題を解決するために、ハンドカーソルが握っているActCubeが3次元形状に付着する可能性がある時に、何か特殊な視覚的補助を行うことが必要である。また、ActCubeが3次元形状に付着した時に音を鳴らすことも効果的であると思われる。

以上、ActCube手法により、3次元環境下での直接操作により、直感的な構築を進められるはずが、3次元環境での操作性がネックとなってしまう、仮想空間の構築を直感的に進められるとまでは、行かなかった。

5.2.2 ActCube手法を評価するための被験者実験

3次元デバイスを用いた実験では、その操作性の悪さが際立ってしまい被験者のActCube手法そのものへの理解度に関する評価が難しい。そこで、3次元デバイスを使用しないで、マウスとキーボード操作でActCubeを用いた仮想空間の構築が実行できるようにAct Systemを改良し、被験者実験を再び行った。

実験の目的

被験者にマウスとキーボードの操作で3次元オブジェクトの操作ができるように改良したAct Systemを用いて仮想機器を構築するタスクを課し、そのタスクにかかった時間、タスクの達成度、構築時の様子、および被験者へのアンケート結果から、ActCube手法の理解度を評価する。また、前実験で問題となったActCubeの3次元形状への付着可能性の有無を視覚的に補助するために、図5.14に示すようにハンドカーソルで握られているActCubeと3次元形状が衝突している時にActCubeを赤色のバウンディングボックスで覆うように改良したため、その効果も調べる。

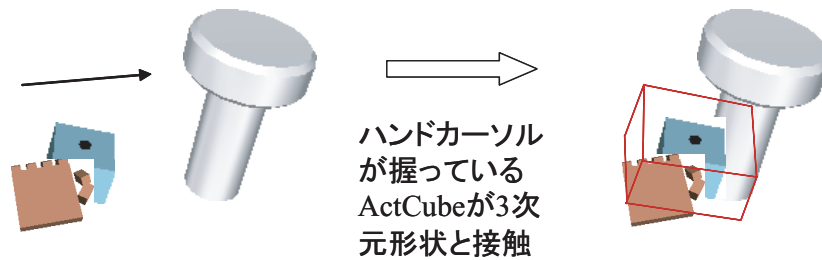


図 5.14: ActCube 付着可能性の視覚的補助

実験の方法

- 被験者

被験者は、仮想空間の構築経験のない20歳代の男性2名とした。なお、前実験とは異なる人間を対象とした。2名とも、キーボードやマウス等日ごろからコンピュータの操作には慣れているが、被験者Cは、プログラミングの知識がなく、被験者Dはプログラミングの知識がある。

- 実験手順

前実験と同様の手順で実験を実行した。

- 作業タスク

作業タスクも前実験と同様のタスクとした。

実験の結果

二人の被験者は共にタスクを完了した。トナー交換できるプリンタの構築に必要な作業時間は、被験者Cでは12分、被験者Dでは7分であった。実験後、実施した被験者へのアンケート結果を表5.7に示す。被験者Cは、アンケート結果でも述べているように、プリンタとトナーの3次元形状があれば、それだけで仮想空間を実世界と同様にシミュレーション可能であると考えていたように、ActCubeの機能の理解に時間を要したことや、キーボードでの3次元オブジェクトの回転操作の習熟に手間取ったこと等から、被験者Dの作業の2倍近い時間を要してしまった。前実験に比べて、全体的に短い時間でタスクが完了しているのは、インタフェースを3次元デバイスから2次元マウス・キーボードに変更したため、被験者が自分の思い通りに3次元オブジェクトを操作できるようになったためである。

表 5.7: キーボード・マウスを用いた被験者実験のアンケート結果

被験者 C	被験者 D
どうして、ActCube を付着させる作業が必要なのか分からない。形状さえ用意してあれば、仮想物体の動きは実現できる気がする。	ActCube の機能はすぐに分かった。
個々の ActCube の軸の意味等を考えていたら、時間がかかった。	付着した時に音が鳴った方がいい
ActCube の概念に慣れるのに、時間がかかった。慣れれば、構築できるようになると思う。	視点変更できるようにしてほしい。
ActCube の色の違いだけだと、区別が難しい。	

考察

実験の結果、被験者 C が、直ぐには ActCube 手法を理解できなかったように、誰しもが、ActCube 手法を簡単に理解できるとまでは行かないようである。ActCube は、本研究で導入した 3 次元オブジェクトで、実世界には存在しないものであり、ActCube が 3 次元環境内でどのような役割を果たし、どのような拘束関係を発生するかという情報は、ユーザが持つ実世界の知識にはないものである。プログラミング等の専門知識をバックボーンとして持つ者は、シミュレーションのために ActCube が果たす役割を直ぐに理解できるが、コンピュータに関する知識に疎いものは、実世界の知識をそのまま 3 次元環境内での作業にも当てはめてしまうため、3 次元形状が用意されていれば、3 次元環境内でもシミュレーション可能と思ってしまい、ActCube の存在理由や機構に戸惑うようである。

この ActCube の機能が理解しにくいという問題を解決するためには、密接拘束関係を設定する 3 次元オブジェクトを直方体形状のような抽象的な形状でなく、構築者が実世界の知識からその役割のイメージを喚起しやすい具体的な形状にすることが方策として考えられる。例を挙げて説明すると、平行移動の拘束関係は、図 5.15 に示すようににルールを用いて設定し、また、蝶番の拘束関係であれば、本当の蝶番のようなもので仮想物体間をつなぐことにより設定する、というように実世界でその拘束関係

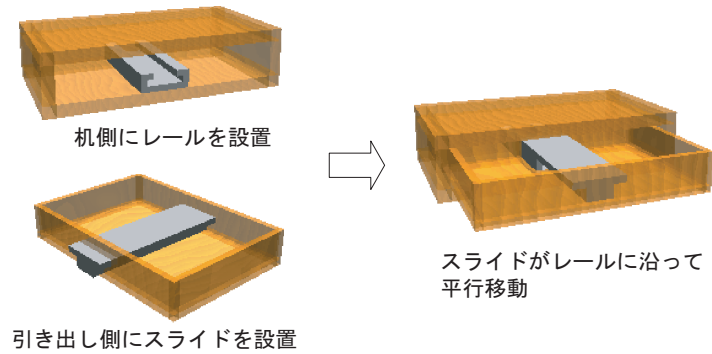


図 5.15: レールを用いた平行移動の設定

が生じる特徴的な器具を使用し設定作業を行うようにする。

しかし、その被験者 C も戸惑いながらも、しばらく試行錯誤し、操作に慣れていくことにより、なんとかタスクを完了していることから、直ぐに ActCube の機構を理解できない人でも、Act System を使用して、しばらく練習を繰り返すことにより、その概念に慣れて仮想空間が構築できるようになると思われる。また、コンピュータの知識をある程度もっている者ならば、どのように ActCube を付着させればいいのかは理解が早く、比較的短時間で仮想空間を構築でき、仮想空間の構築を初めて行う者でも仮想空間が構築できる手法であることは示されたと思われる。

ActCube が付着する可能性がある時に赤色のバウンディングボックスを描画した効果は、ActCube が上手く付着しなかったというデメリットの意見が出ていないことから、一応の成果を上げたように思われる。

また、ActCube の外観情報だけでは、ActCube が持っている拘束関係の種類認識が難しく、凹 ActCube と凸 ActCube の区別が付きにくいという問題も分かった。この問題を解決するためには、3次元環境下での操作において、外観等の3次元情報のみに頼るのではなく、ハンドカーソルが接触した ActCube の種類を文字情報で表示する等、操作の補助のために2次元的情報を効果的に導入していく策が考えられる。

5.3 まとめと今後の課題

スウィング式逆止弁の構築作業、および、被験者実験を通して、Act System は、効率良く仮想空間を構築できる手法であることは分かったが、3次元デバイスの操作性等の問題が判明した。そこで、それらの問題を解決し、さらに効率的な仮想空間構築を行えるようにするための今後の課題としては、以下の項目が挙げられる。

- 3次元デバイスの改良
- ActCubeの形状に関する検討
- ActCubeの種類の見直しと多様なActCubeの開発
- 操作補助手法の充実
- 2次元構築インタフェースとの連携
- 拘束関係発生形状解析ツールの開発

3次元デバイスの改良

Act Systemの3次元デバイスは、被験者実験により、操作性に難があることが分かった。そこで、まず、今後の課題としては、ジェスチャ入力を室内環境で実現できる3次元デバイスに変更し、再度システムの評価実験を行うことが挙げられる。

また、Act Systemでは、ユーザの指の動きは、手の開閉情報を取得しているだけであり、この情報だけでは、構築者は自然な操作で仮想空間が構築できない。指というのは、本来、物体を回転させる場合、平行移動させる場合等でその動き方が異なるものであり、また、物体を強く握る、軽く握るという違いもある。そこで、手の位置計測が実現できるだけでなく、ユーザの指の動きも計測できる3次元デバイスの開発が望まれる。

以上のように、より現実に近い直接操作での構築を実現するためには、実用的な3次元デバイスの開発が必要とされる。また、構築した仮想空間を訓練目的で使用する場合、訓練生が機器に関する知識だけでなく技能習得ができるようになるためには、実世界と同様のジェスチャで仮想機器を操作する必要があり、そのために、3次元デバイスは必須となる。

しかし、現実には3次元デバイスは、規模が大きくなりやすく、また、その設置費用も高価になりやすい。そこで、仮想空間の構築作業だけに限定するならば、3次元環境の直接操作を2次元デバイス、または、ゲームパッド等を用いて実現できるようにした方が、デバイスの携帯性等から多くの人に使用してもらえる期待がある。それらの身近にあるデバイスでの3次元環境操作を実現するような手法の開発も行う必要がある。

ActCube 形状の外観に関する検討

被験者実験の結果、ActCube は直方体形状をしているが、その形状が抽象的であるため、初めて Act System を利用する人では、ActCube の機能の理解に戸惑う恐れがあることが判明した。

そこで、今後の課題としては、初めて Act System を利用する人でも、その外見情報から、すぐに ActCube の機能を理解できるような形状として、どのような形状が望ましいかの検討を行う必要がある。

ActCube の種類の検討と多様な ActCube の開発

本研究では、スウィング式逆止弁の訓練環境の構築を試み、短時間で構築することができた。しかし、現時点では、試作システムで構築した訓練環境は、機能が乏しく、十分な訓練効果を上げられるほど、現実の細かい事象を模擬できておらず、現場の保守作業の訓練にますます適用可能であるとは言い難い。

そこで、本研究で提案した ActCube 手法を実際の訓練に適用するためには、今後、現場の機器の保守作業を事細かに検証し、本当に必要な ActCube の種類や機能の洗い出しを行っていく必要がある。

また、本研究では、密接拘束関係の ActCube のみを開発し、スイッチや、つまみ調節等の機能を果たすような情報を伝達する ActCube は開発しなかった。そのため、本研究での構築できる仮想空間には制限が生じてしまった。しかし、これらの情報伝達 ActCube を開発できれば、構築できる仮想空間の幅が広がり、例えば、スイッチを入れれば、音が鳴るような機器等も ActCube の付着作業だけで構築可能になることが期待される。

操作補助手法の充実

Act System でも、ハンドカーソルが接触している 3 次元オブジェクトをバウンディングボックスで覆う等の操作補助手法を実現しているが、ActCube の付着作業等を構築者がさらに快適かつ直感的に行うためには、さらなる操作補助手法の充実が必要である。その操作補助手法としては、ハンドカーソルが触れている ActCube の種類情報等を文字情報として画面に描画したり、ActCube を拡大縮小する時に、ActCube の近傍にものさしのような ActCube の大きさを揃えるための指標となるものを提示する等 2 次元 GUI のインタフェースにおけるユーザ補助のためのアイデアを 3 次元環境下

でも使用する方策が考えられる。

2次元構築インタフェースとの連携

3次元環境下での構築作業では、仮想空間のおおまかな骨組みは設定できるが、細かい位置合わせ等の作業を行うには、不向きである。そこで、この問題を解決するために、3次元環境下での構築を行うためのツールだけでなく、並列的に2次元 GUI でも仮想空間を構築できるツールを用意することが望ましい。

拘束関係発生形状解析ツールの開発

ActCube 手法では、ユーザに ActCube を 3次元形状における密接拘束関係が発生する箇所に付着させる作業を行ってもらう必要がある。しかし、3次元形状で発生する拘束関係が非常に多数存在する場合は、構築者がこの付着作業を実現するのは大きな構築負荷になる。そこで、今後 Act System を用いて仮想機器を構築していくことにより、3次元形状において密接拘束関係が発生する箇所というものの特徴を分析し、その結果を用いて、3次元形状をスキャンして自動的に密接拘束関係が発生する箇所に ActCube を配置してくれるツールの開発が望まれる。

第 6 章 結論

本研究では、仮想空間を構築する際に必要となる専門知識を排除し、3次元環境下での3次元オブジェクトの直接操作のみで仮想空間を構築する手法として、物理的拘束関係に関する情報を設定することができる「ActCube」と呼ぶ3次元オブジェクトを仮想空間の構築に導入する手法を提案した。また、そのActCube手法に基づき仮想空間を、実際に構築しシミュレートできるシステムとしてAct Systemを開発した。

2章では、まず、機器操作・保守作業に注目してVirtual Realityの研究と応用の現状について述べた。次いで、VR技術を用いて訓練システムを構築するための従来手法とその問題点について述べ、その解決策として、近年モデリングの分野で行われている3次元環境に没入しての実世界の知識を用いた3次元モデラが存在性に注目し、仮想空間の構築も3次元環境内での直接操作により実現することの可能性を述べ、本研究の目的を明らかにした。

3章では、まず、仮想空間内における仮想物体の動きをシミュレートする一般的な方法について述べ、その問題点を明らかにした。そして、本研究で採用するシミュレーション方法について説明し、そのシミュレーション手法を行うために必要となる物理的拘束関係の設定を3次元環境下で設定できるActCubeを導入した。そして、ActCubeの機能について説明し、ActCubeの3次元形状への付着作業だけで、仮想空間が構築できることを述べた。そして、具体的にActCubeを用いた仮想空間の構築例について述べた後、ActCubeを用いて仮想空間を構築することによる利点・欠点についてまとめた。

4章では、ActCube手法を用いて仮想空間を、実際に構築し、シミュレートするためのシステムとして開発したAct Systemについて述べた。まず、はじめに、ActCube手法を実現するためにAct Systemが備えるべき機能について述べた後、開発したAct Systemの概要について説明し、次にAct Systemのシステム構成および構築インタフェースについて説明した。その後、Act Systemを用いて仮想空間を構築する際の構築者の操作方法と手順について説明し、最後にAct Systemで構築した仮想空間をシミュレーションする際の処理の流れについて説明した。

5章では、本研究で開発した各種ActCubeの機能や、その使用例について説明し、その後、ActCube手法を用いた仮想空間の実現例として、スウィング式逆止弁の分解組

立作業ができる仮想空間を構築し、Act System の構築労力等について評価した。その後、Act System の操作性、提案した ActCube 手法の有効性を評価するために行った被験者実験の結果について述べた後、今後の展望についてまとめた。

本研究では、3次元環境内で積み木を操作する感覚で3次元形状に ActCube という3次元オブジェクトを付着させることにより、3次元形状の物理的拘束関係が設定できる手法を提案した。従来仮想空間の構築インタフェースは多機能化し、複雑で専門知識を要求する傾向にあったものを、この手法では、仮想空間の構築に必要な最低限の設定のみを抽出し、それを3次元環境での直接操作で設定できることを目指した。被験者実験の結果、次のような結果が得られた。

- 仮想空間構築の初心者でも、容易に仮想空間が構築できた
- 構築者によっては、ActCube の機能の認識が難しい

この結果を踏まえ、今後の課題としては、以下の項目が挙げられる。

- ActCube の3次元環境における外観の検討作業
- 多様な ActCube の開発

まず、誰もが仮想物体の挙動を設定できることを目指し、ActCube の機能がその外観から直感的に判断しやすい形状を検討する作業を行う必要がある。

また、現段階の試作システムで構築した訓練環境は、実際の訓練にそのまま適用することは難しい。今後は、実際の訓練・保守機器の訓練に関して訓練の現場に赴き話を聞くなどして、実際に VR で訓練環境を構築する際に実現しなければならない事象を検討し、それを実現するために必要な ActCube の種類・機能の検討および開発を行っていく必要がある。

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。

本研究室に配属されてより3年間、常に研究に協力していただき、本年度の研究においても研究の進行から論文の執筆に到るまで多大なるご協力を頂いた石井裕剛助手に心より感謝いたします。

修士課程の2年間、楽しい時も苦しい時もその経験を分かち合い、修士論文の執筆にあたって最後まで共に闘い貫いた、修士課程2回生の越智和弘君、小林隆君、鮫島良太さん、新田和弘君に感謝いたします。

本研究を進めるにあたり、論文の校正等、様々な面で協力していただいた博士課程2回生の伊藤京子さん、修士課程1回生の城田莉菜さん、富田大輔君、藤野秀則君、また、眠い中、早朝の被験者実験に参加してくれた研究室に宿泊するのが大好きな修士課程2回生の服部貴司君に深く感謝いたします。

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、吉川万里子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] 廣瀬通孝：電子情報通信学会編「ヒューマンコミュニケーション工学シリーズ」バーチャルリアリティ，オーム社(1995).
- [2] 廣瀬通孝：バーチャルリアリティって何だろう，ダイヤモンド社(1997).
- [3] 野村淳二，澤田一哉編著：日本ファジイ学会編「ソフトコンピューティングシリーズ」バーチャルリアリティ，朝倉書店(1997).
- [4] 広内 哲夫：Web3D グラフィックス VRML で創るバーチャルワールド，ピアソン・エデュケーション(2001).
- [5] <http://www.web3d.org/>(2003年2月3日現在).
- [6] 日高 俊明著：VR 革命 仮想を現実とした技術者たち，オーム社(2000).
- [7] Loftin, R.B. and Kenney , P.J.: Training the Hubble Space Telescope Flight Team. IEEE, Computer Graphics and Applications, Vol.15, No.5, pp.21-37(1995).
- [8] 南雲俊喜, 中山功, 甘利治雄, 岡田幹夫:小型円筒面スクリーンによる運転・保守作業訓練環境の構築, 日本バーチャルリアリティ学会第3回大会論文集, pp.207-208(1998).
- [9] 天野友博, 田中和明, 鄭絳宇, 安部憲広:仮想機械を用いる機器修復法の教示と誤りの検出・修正機構, 日本バーチャルリアリティ学会第2回大会論文集, pp.89-92(1997).
- [10] 新井浩一, 阿部慶子, 上地登:VR 技術を用いた変電所保守員向け集合教育用体感型シミュレータの開発, 日本バーチャルリアリティ学会論文誌, Vol.2, No.4, pp.7-16(1997).
- [11] 吉川榮和, 手塚哲央, 柏健一郎, 石井裕剛:仮想空間における機器保守訓練シミュレーション, 日本原子力学会誌, Vol.39, No.12 , pp.72-83(1997).

- [12] 石井裕剛, 吉川榮和: プラント技能訓練へのバーチャルリアリティ技術適用の展望, 計測自動制御学会システム情報部門シンポジウム 2000, pp.55-60(2000).
- [13] 小牧大輔: オブジェクト指向に基づく仮想空間構築手法に関する研究, 京都大学大学院エネルギー科学研究科修士論文 (2001).
- [14] 石井裕剛, 手塚哲央, 吉川榮和: 人工現実感技術を用いた機器保守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).
- [15] <http://www.lexer.co.jp/>(2003年2月3日現在).
- [16] <http://www.sense8.com/>(2003年2月3日現在).
- [17] <http://www.eonreality.com/>(2003年2月3日現在).
- [18] 清川 清, 竹村 治雄, 片山 喜章, 岩佐 英彦, 横矢 直和: ”両手操作を用いた仮想物体モデラ VLEGO”, 電子情報通信学会論文誌 (A), Vol.J80-A, No.9, pp.1517-1526(1997).
- [19] 大芝崇, 田中二郎: 3次元モデリングツール ”Claymore”: 付加情報によって強化された直接操作: 日本ソフトウェア科学会第15回大会論文集, pp.161-164(1998).
- [20] Conner DB, Snibbe SS, Hemdon KP, Robbins DC, Zeleznik RC, Van Dam A: ThreeDimensional Widgets: SIGGRAPH Symposium on Interactive 3D Graphics, pp.183-188(1992).
- [21] <http://www.nime.ac.jp/it3d/>(2003年2月3日現在).
- [22] 吉川恒夫: 力覚人工現実感技術を用いた技能教育システム: 人工現実感の設計 (館監修 佐藤編), 第4.2.3項 (pp.145-150), 培風館 (2000).
- [23] 岡田義広, 田中譲: 対話型3Dソフトウェア構築システム-IntelligentBox-: コンピュータソフトウェア, Vol.12, No.4, pp.84-94(1995).
- [24] Terrence Fernando, Norman Murray, Kevin T W Tan, Prasad Wimalaratne : Software Architecture for a Constraint-based Virtual Environment:ACM International Symposium on Virtual Reality Software and Technology, VRST 99, London. UK, December, pp. 147-154(1999).

- [25] Terrence Fernando, Luis Marcelino, Prasad Wimalaratne : Constraint-based Immersive Environment for Supporting Assembly and Maintenance Tasks: HCII 2001, New Orleans (USA), August 2001, Volume 1, pp. 943-947(2001).
- [26] David Baraff : Analytical methods for dynamic simulation of non-penetrating rigid bodies : In Computer Graphics Proceedings, vol 23, pp.223-232(1989).
- [27] David Baraff : contact force computation for nonpenetrating rigid bodies : Computer Graphics Proceedings, Annual Conference Series, pp.23-34(1994).
- [28] 川地克明 : 機械機構の挙動の直感的な把握を目的とした剛体運動シミュレーション手法東京大学工学部精密機械工学科修士論文 (1997).
- [29] 石井裕剛, 柏健一郎, 手塚哲央, 吉川榮和 : 人工現実感を用いた機器保修訓練システムへのペトリネットの適用, 第40回システム制御情報学会研究発表講演会予稿集, pp.239-240(1996).
- [30] 舟橋 健司, 安田 孝美, 横井 茂樹, 鳥脇 純一郎 : 仮想空間における両手による協調操作モデル, 情報処理学会論文誌, Vol.39, No.5, pp.1334-1342 (1998).
- [31] 橋本 渉, 岩田洋夫 : ボリュームデータに基づく術中危険領域の力覚表現, 日本バーチャルリアリティ学会論文誌, Vol. 3, No. 4, pp.197-202(1998).
- [32] OpenGL Performer Programming Guides & Reference Manuals, <http://www.sgi.com/software/performer/manuals.html>, Silicon Graphics, (2002).
- [33] M. Lin and S. Gottschalk. : Collision Detection between Geometric Models, A Survey, Appeared in the Proceedings of IMA Conference on Mathematics of Surfaces (1998).
- [34] ric Larsen, Stefan Gottschalk, Ming C. Lin, and Dinesh Manocha : Fast Proximity Queries with Swept Sphere Volumes, TR99-018, (1999).
- [35] David M. Bourg : Physics for Game Developers, O'Reilly & Associates (2001).
- [36] ULTRATRAK PRO OPERATING MANUAL, Polhemus, (1996).

- [37] Bier, Eric A.: Snap-dragging in three dimensions. In Proceedings of the 1990 Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics, Vol. 24, pp.193-205 (1990).