

Title	共有仮想現実環境内での力覚を伴う協調作業による物体操作システム
Author(s)	藤井, 大悟
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1771
Rights	
Description	Supervisor:堀口 進, 情報科学研究科, 修士

修 士 論 文

共有仮想現実環境内での力覚を伴う
協調作業による物体操作システム

北陸先端科学技術大学院大学
情報科学研究科 情報システム学専攻

藤井 大悟

2004 年 3 月

修士論文

共有仮想現実環境内での力覚を伴う 協調作業による物体操作システム

指導教官 堀口 進 教授

審査委員主査 堀口 進 教授
審査委員 Shen Hong 教授
審査委員 松澤 照男 教授

北陸先端科学技術大学院大学
情報科学研究科 情報システム学専攻

210079 藤井 大悟

提出年月: 2004年2月

目次

第1章	序論	1
1.1	研究の目的と背景	1
1.2	本論文の構成	2
第2章	仮想現実環境と仮想力触覚提示	3
2.1	仮想現実環境	3
2.2	仮想力触覚提示装置	4
2.3	共有仮想現実環境	9
2.3.1	共有仮想現実環境	9
2.3.2	触覚を含む共有仮想現実環境	11
2.4	まとめ	12
第3章	ネットワーク上での力触覚	13
3.1	ネットワーク上での力覚の伝送	13
3.1.1	システム構成	13
3.2	力覚の伝送実験	18
3.2.1	ネットワークの更新周期による操作への影響	18
3.2.2	ネットワークの遅延による操作への影響	25
3.3	まとめ	30
第4章	協調作業による物体操作システムの構築	31
4.1	はじめに	31
4.2	物体操作に適応した力覚提示法	31
4.3	システム構成	33
4.4	リアルタイム物理モデル	35
4.5	システムの評価実験	39
4.5.1	サーバ1台とクライアント1台でのシステムの評価	39
4.5.2	サーバ1台とクライアント2台でのシステムの評価	43
4.6	まとめ	47
第5章	まとめ	48
5.1	まとめ	48

5.2 今後の課題	49
---------------------	----

目次

2.1	仮想力覚提示デバイスの分類	5
2.2	SPIDAR	6
2.3	Cyber Grasp	6
2.4	PDMMF	6
2.5	FEELEX	7
2.6	体積型ハプティックディスプレイ	8
2.7	WYSIWYF ディスプレイ	8
2.8	Collaborative CyberMirage	10
2.9	DIVE	10
2.10	握手システム	11
3.1	システム構成	13
3.2	処理の流れ	14
3.3	PHANToM の外観	15
3.4	サーバ クライアントモデル	17
3.5	Spring-Damper Model	18
3.6	操作画面	18
3.7	クライアント 1 台の場合のカーソルの距離 (30Hz)	20
3.8	クライアント 1 台の場合のカーソルの距離 (60Hz)	20
3.9	クライアント 1 台の場合のカーソルの距離 (100Hz)	21
3.10	クライアント 1 台の場合のカーソルの距離 (150Hz)	21
3.11	クライアント 1 台の場合のカーソルの距離 (200Hz)	22
3.12	クライアント 1 台の場合のカーソルの距離 (300Hz)	22
3.13	クライアント 1 台の場合のカーソルの距離 (500Hz)	23
3.14	クライアント 1 台の場合の更新周期による平均と標準偏差	23
3.15	クライアント 2 台の場合の更新周期による平均と標準偏差 (PC1)	24
3.16	クライアント 2 台の場合の更新周期による平均と標準偏差 (PC2)	24
3.17	通信遅延がある場合: 60Hz	25
3.18	通信遅延がある場合: 100Hz	25
3.19	カーソルの軌道 遅延 0 [ms]	26
3.20	カーソルの軌道 遅延 30 [ms]	26
3.21	カーソルの軌道 遅延 50 [ms]	27

3.22	カーソルの軌道 遅延 80 [ms]	27
3.23	カーソルの軌道 遅延 100 [ms]	28
3.24	カーソルの軌道 遅延 150 [ms]	28
3.25	カーソルの軌道 遅延 200 [ms]	29
4.1	通常の反力の生成	31
4.2	遅延がある場合の反力の生成	32
4.3	シーケンス図	34
4.4	処理の流れ	35
4.5	抗力の場合	37
4.6	ペナルティ法	38
4.7	解析法とペナルティ法の計算速度の比較	39
4.8	物体操作タスク	39
4.9	サーバ1台、クライアント1台の場合での仮想物体の位置 (X 軸)、遅延時間 0[ms]	40
4.10	サーバ1台、クライアント1台の場合での仮想物体の位置 (X 軸)、遅延時間 30[ms]	40
4.11	サーバ1台、クライアント1台の場合での仮想物体の位置 (X 軸)、遅延時間 50[ms]	41
4.12	サーバ1台、クライアント1台の場合での仮想物体の位置 (X 軸)、遅延時間 80[ms]	41
4.13	サーバ1台、クライアント1台の場合での仮想物体の位置 (X 軸)、遅延時間 100[ms]	42
4.14	サーバとクライアントの通信遅延による位置誤差	42
4.15	システム環境2	44
4.16	サーバ1台、クライアント2台の場合での仮想物体の位置 (X 軸)、遅延時間 0[ms]	44
4.17	サーバ1台、クライアント2台の場合での仮想物体の位置 (X 軸)、遅延時間 30[ms]	45
4.18	サーバ1台、クライアント2台の場合での仮想物体の位置 (X 軸)、遅延時間 50[ms]	45
4.19	サーバ1台、クライアント2台の場合での仮想物体の位置 (X 軸)、遅延時間 80[ms]	46
4.20	サーバ1台、クライアント2台の場合での仮想物体の位置 (X 軸)、遅延時間 100[ms]	46

表 目 次

3.1	構成システムの環境	14
3.2	PHANToM 性能表	15
4.1	構成システムの環境 2	43

第1章 序論

1.1 研究の目的と背景

近年、注目されているバーチャルリアリティ (VR) とは、コンピュータにより創り出された仮想環境を、現実環境と同じように体感・行動できる技術のことである。VR 技術は、コンピュータ性能の向上やネットワークの高速化により、急速に進歩してきた。中でも遠隔地間で共有仮想現実環境を構築し協調作業を行う研究が盛んに行われており、医療分野や教育分野など多様な分野での応用が期待されている。

近年では、ネットワークを介して接続される共有仮想現実環境で力覚を表現する研究が行われている。これまで VR 技術が主に対象としていたのは、視覚や聴覚であった。それらに力覚を加えることによってこれまでの視覚や聴覚に作用する VR システムより高い現実感が得られることが期待されている。力覚を提示するためには力触覚デバイスである PHANTOM や CyberGrasp が必要となる。このような触覚デバイスをネットワークを介して接続することにより、互いに離れた場所においても、力触覚を伴う、より現実感のあるインタラクションが可能となる。例えば、映像や音声に触覚を加えることによってそれぞれ異なる場所にいる複数の医師達が、一人の患者に対して外科手術を行うシミュレーションが可能になる。

このような力覚を伴う共有仮想現実環境では、それぞれのユーザが違和感なく力触覚を介した仮想環境へ対するインタラクションが行うことが出来る必要がある。しかしながら、ネットワークを介した力覚情報の制御には以下に示す解決すべき問題がある。

力覚デバイスのフォースフィードバック機構の制御更新周期は 1kHz 程度と高速であるため、ネットワークの遅延の影響を受け、安定したインタラクションの実現は困難である。

その理由は、人間が何かを触り続けているという感覚を感じる周波数が約数百 Hz だからである。したがって、フォースフィードバックは数百 Hz から 1kHz の高速な提示が必要となり、フィードバックの計算にはその制御周期に収まる高速な計算処理が要求されることになる。しかし、IP ネットワーク等の通信環境では 100 ミリ秒から数秒の通信遅延が存在するため、力触覚のフィードバック処理をネットワーク上で行った場合、更新は十分に行えず、安定したフォースフィードバックは困難であるため、安定したインタラクションを行うことができない。

本研究では、共有された一つの仮想環境上で力触覚デバイスを用いて物体の操作によるインタラクションを行うシステムを構築することを目的とする。その場合、本研究で力覚の高速な更新周期に対する通信遅延の問題を解決するために、力覚のフィードバック情報

をネットワーク上での通信から切り離し、ユーザに近い場所で計算する手法を採用することで、安定した力覚の提示を行う。またその場合に、仮想物体の運動を現実に近い物理システムで高速に計算することによって、違和感の少ない、力覚の伴う協調作業システムを実現する。

1.2 本論文の構成

1章は研究の背景と目的について述べる。

2章では本研究と関連した従来研究について述べ、これらの特徴と問題点について検討する。

3章では、力触覚デバイスである PHANTOM を用いて、ネットワーク上で力覚の伝送を行い、その場合にネットワークの状態がユーザ操作にどの程度影響があるかを調べる。

4章では、力覚を伴う協調作業による物体操作システムの構築に向けて、通信遅延に対処するためのシステム制御法を提案する。

5章にまとめを述べる。

第2章 仮想現実環境と仮想力触覚提示

2.1 仮想現実環境

近年、注目されているバーチャルリアリティ (VR) とは、コンピュータにより創り出された仮想環境に現実環境と同じような現実感を与え、ユーザに体感・行動できる技術のことである。VR 技術は、コンピュータ性能の向上やネットワークの高速化により、急速に進歩してきた。

単に「人工的な現実感」といった場合には、例えば小説や映画といったメディア表現も含まれるが、VR の構成要件としては以下の要素が必要とされる。

- 体験可能な仮想空間 (virtual world) の構築
- 五感 (のうちのいくつか) に働きかけて得られる没入感 (immersion)
- 対象者の位置や動作に対する感覚へのフィードバック (sensory feedback)
- 対象者が世界に働きかけることができる対話性 (interactivity)

うしろの3つを構成要件とする場合もある。この基準に照らせば、例えば小説には視聴覚による没入感が欠け、映画には対話性が欠けるため、VR とはみなされない。

Virtual Reality という用語は、1987 年に NASA(米航空宇宙局) が、VPL Research 社に発注して開発した「VIEW」(仮想環境ワークステーション) というシステムの開発プロジェクトで使い始めた用語で、語感の先進的な響きとシンプルさが受け入れられて、コンピュータシステムで現実感を作り出す技術の総称として定着した。

計算機が生成した仮想の世界で擬似的に多様な状況を体験できる仮想現実システムは、今日では様々な分野で応用されており、特にシミュレータやアミューズメントの分野では積極的に利用されている。

仮想現実システムはコンピュータと入出力機器の組み合わせによって構築される。頭に装着して視界を覆うヘッドマウントディスプレイ (HMD : Head Mount Display) や、手の動きを入力したり擬似的に触覚を与える手袋上のデータグローブ、体を包み込む衣服状の入出力装置データスーツなど、様々な機器が考案され、いくつかは実用化されており、現在でも研究、開発が行われている。

2.2 仮想力覚提示装置

人は何かに触れたり、操ったりするなどには手を使って空間に働きかける。手には感覚があり、脳は手と空間の状況を捉えながら手を制御する。力覚提示装置は手の感覚の一部を満足させることで、脳が手を動かすのを助け、仮想現実環境の操作性を大きく向上させる。初期段階の仮想現実技術は、主に視覚情報や音声情報などの仮想再現が中心であったが、仮想力覚提示装置 (Haptic Display) による力覚や触覚の仮想的な再現も行われている。

さらなる技術の発展により、画像や音声に加えて力覚を仮想的に提示する力覚提示型仮想現実システムの需要が高まっている。仮想力覚提示を伴った仮想現実システムの応用分野として、以下のような分野が存在する。

- 医療・福祉分野
 - － 医療シミュレータ, リハビリ支援システム
- 科学技術分野
 - － 可視化ツール, プロダクトデザインシステム
- 芸術分野
 - － インタラクティブアート, 造形支援システム
- アミューズメント分野
 - － 対戦型コンピュータゲーム

これまで、様々な種類の仮想力覚提示装置が提案および開発されているが、野間ら [1] によると使用形態から図 2.1 のような分類をすることができる。

把持型

操作型に分類される把持型仮想力覚提示装置は、一般的には通常のロボットアームの先端部にグリップを設け、利用者はグリップ部分を握り保持することで操作を行う。利用者が操作を行う際に、グリップ部分を常に把持する必要があるこの形態の機構では、自由な入力動作を阻害しないためには並進移動と回転運動について 6 自由度以上を確保する必要がある。円滑な操作を実現するためには、利用者の手先部の動作範囲を十分にカバーした機構の稼働範囲を実現し、利用者の腕力を支持可能な剛性を持たせることが重要である。さらに、機構の自由度と動作範囲や提示可能反力など、実際の構築において同時に実現することができない要素があるため、使用目的に応じた装置の選択が求められる。

Salisbury ら [5] [6] は、ジンバル機構を応用したペン型の仮想力覚提示装置 PHANTOM を開発した。この装置は操作者に対して、任意の方向 (3DOF) の仮想力覚を提示するこ



図 2.1: 仮想力覚提示デバイスの分類

とができる。それにより，操作者に対して仮想空間中のオブジェクトの形、堅さ、柔らかさ、素材感、重さなどを操作者に対して力覚・触覚提示することができる。

装着型

装着型仮想力覚提示装置は、デバイスを利用者の体に固定することで、利用者の身体動作に直接的に作用し拘束力を提示する。一般的に、この形式の装置には手先腕部全体の自由度を満足させる機構が必要であり、システム構成が複雑かつ大型になりやすい。人間の骨格に対応した外骨格マニピュレータは大型であり、利用者への装着も身体的な負担が大きいため、現在では握持型仮想力覚提示装置のグリップ部分を身体に固定する形式が主流である。

佐藤ら [7] [8] は SPIDAR(図 2.2) と呼ばれる糸の張力を利用した力覚提示装置を開発した。この装置は 4 本の糸で支持されたリングによって指先を拘束し、並進力を指先に提示することができる。

Virtual Technologies 社は、アクチュエータ付のリング機構をデータグローブの上から装着する Cyber Grasp [18](図 2.3) を製品化した。このデバイスは各指に力をフィードバックするワイヤで接続されたリングを 2 本ずつ装着し、力を指に提示できる。

Burdia ら [13] が開発した PDMFF(図 2.4) では、小型の空気圧ピストンを指先と手のひらの部分に配置し、握り込み動作を行う際の指先への反力提示を実現した。この形式の装置では、物体の握持感のみの再現であるが、装置を小型化することが可能であり身体への装着という面では有効である。



☒ 2.2: SPIDAR



☒ 2.3: Cyber Grasp



☒ 2.4: PDMMF

形状入力型

形状入力型仮想力覚提示装置は仮想物体の形状の一部を機構によって実世界に再現し、体験者が実物体と同じように指や手全体で直接触れることで力触覚を提示する。この形式の装置は、装置操作者に面の力触覚提示が可能である。また装置を装着したり、外したりする必要がない。しかしながら、装置自体の実現が困難であり、再現できる形状に限界があるという問題点がある。

岩田ら [14] は、直動機構のロッドを用いて、面の凹凸を表現できる FEELEX(図 2.5) を開発した。面の凹凸を表現するために、直動機構のロッドをアレイ状に配置して駆動させ、操作者に面の触覚を提示している。



図 2.5: FEELEX

阿部ら [15] は、空気圧バルーンを用いて、多方向から直接触れることが可能なディスプレイ（体積型ハプティックディスプレイ）(図 2.6) を開発した。この装置は、空気圧バルーンを複数個用いることで、連続した「面」を呈示するだけでなく、把持行動のように多方向からのコンタクトも可能になっている。

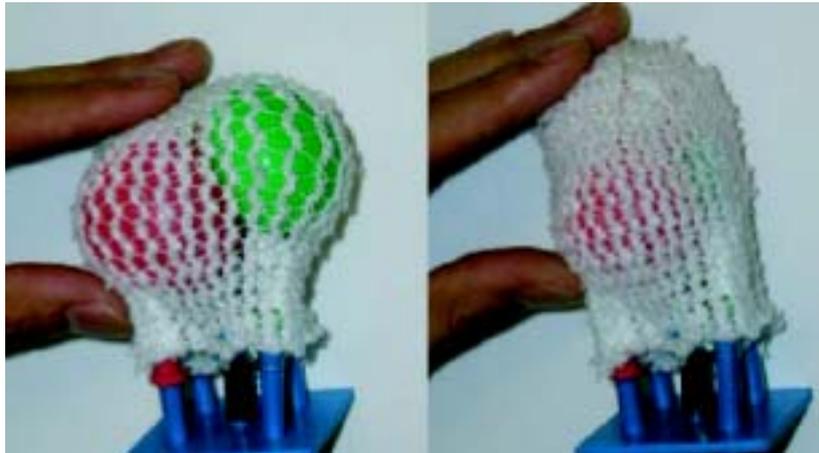


図 2.6: 体積型ハプティックディスプレイ

横小路ら [10] [11] [12] は、視覚情報と力覚情報の融合を目指した WYSIWYF ディスプレイ (図 2.7) を開発した。この研究では、画像ベースの位置トラッキング、クロマキーにより実画像と CG 画像との混合、遭遇型ハプティックインタフェースなどを提案している。

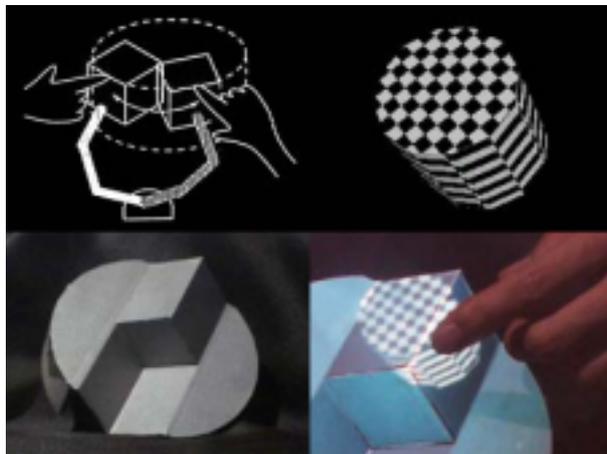


図 2.7: WYSIWYF ディスプレイ

以上のように、仮想現実システムで仮想力覚提示を行うために、これまで様々な形態の仮想力覚提示装置が提案されてきた。

仮想力覚の提示は人間の身体の一部を直接的に拘束することで、入力装置としての機能と出力装置として機能を両立させている。そのため、使用する目的や構築する仮想現実環境の種類に応じて、適切な形態の装置を選択することが重要である。

2.3 共有仮想現実環境

2.3.1 共有仮想現実環境

仮想現実環境を共有するということは、いままで単独で動作していた仮想現実システムを複数のユーザで利用できるように拡張し、一つの仮想的な空間で何らかの作業を行うことである。現実環境でも複数の人が互いに協力し合うことで、単独では不可能な作業を行ったり、単独の作業よりも効率良く行ったりしているものが数多くある。このような協調作業の例として、システムメンテナンス、施設の建設、外科手術などが挙げられる。このような作業を仮想現実環境によって共有することで、

- 距離の克服
- 物理的制約からの解放
- 積極的な操作支援

などの利点が得られると考えられる。

協調仮想環境 (CVEs: Collaborative Virtual Environments) の研究は、1台のコンピュータの中で行われていたゲームやシミュレーションなどを、ネットワークに接続された複数のコンピュータ上で実行されるように拡張した、対戦型シューティングゲームや軍事シミュレーションなどから始まった。初めは高速な LAN 上に構築された閉じたシステムであったが、インターネットの普及とともに WAN 上に公開のサイバースペースを開設するサイトが多く見られるようになった。

1990年代頃半ばには、マルチメディアサービスの実験を目的としたもの、チャットやフォーラムを拡張した新しいコミュニケーションサービスをめざしたもの、電子商取引と結び付けたものなどさまざまな目的のサイバースペースが登場した。その中には公開実験の形式をとるものから有料の会員制のものまで、いろいろな参加形態のものがある。

共有仮想現実環境の研究

仮想空間共有システムの初期として、1980年代に米国防総省高等研究局 (DARPA) によって開発された SIMNET [20] がある。このシステムは、複数の参加者が戦車やヘリコプターなどの搭乗者となり、チームで戦術シミュレーションすることを目的として開発された。

ATR の臨場感通信会議システム [22] では、「互いに異なる複数の場所にいる多くの人々があたかも一堂に会した面談の感覚で会議することができる通信」を実現した。このシステムでは、スクリーン面を超えたオブジェクトの指さし動作や、複数人による仮想オブジェクト組み立て作業を実現している。

MR システム研究所によって開発された Collaborative CyberMirage [23](図 2.8) は、近未来のインターネットショッピングを題材に作成されたシステムであり、写実的な仮想環境を複数人で共有することができる。



図 2.8: Collaborative CyberMirage

仮想空間管理技術として Carlsson らによって開発された DIVE [21](図 2.9) は、マルチユーザ分散仮想環境として開発されたシステムである。このシステムでは、Aura と呼ばれる概念が提唱された。Aura とは、仮想空間内のオブジェクトの周囲に広がる一定範囲の空間を指す。複数のオブジェクトの Aura 同士が重なったときにのみ、そのオブジェクト同士のインタラクションを行わせることで、通信しなければならないオブジェクトの数を減らすことができ、通信量を削減することができる。



図 2.9: DIVE

2.3.2 触覚を含む共有仮想現実環境

仮想空間共有システムにおいて、ユーザに触覚を与えることが、音声・動画像のみによって構成されるシステムと比較し、ユーザに仮想空間をより臨場感があると感じさせることが示されている。

Basdogan ら [2] の研究においては、遠隔地にいる複数のユーザが触覚デバイス、グラフィックスディスプレイを用いてタスクを行っている。このタスクにおいてユーザに、動画像のみを与える場合、動画像と触覚を与える場合に対して、ユーザ間の“ a sense of togetherness ”について主観評価に基づく測定を行い、触覚メディアがそのような感覚を向上させることを示している。

また、ネットワークを通じて共有される仮想空間内の仮想物体を触覚デバイスを用いて操作することの出来るシステムが既にいくつか開発が行われている。

Buttolo ら [4] は“ one user at a time architecture ”に基づく触覚を含むネットワークスキャッシュを実現している。しかしながら、“ one user at a time architecture ”においては、複数のユーザは同時に一つのオブジェクトの触ることが不可能である。

Alhalabi ら [3] は力覚デバイスを用いてフォースフィードバックによる遠隔地間での握手システム(図 2.10)を構築している。この研究では力覚として各ユーザでの握手の振りを伝送し、ネットワークの通信状態に影響を受けにくいシステムを構築した。

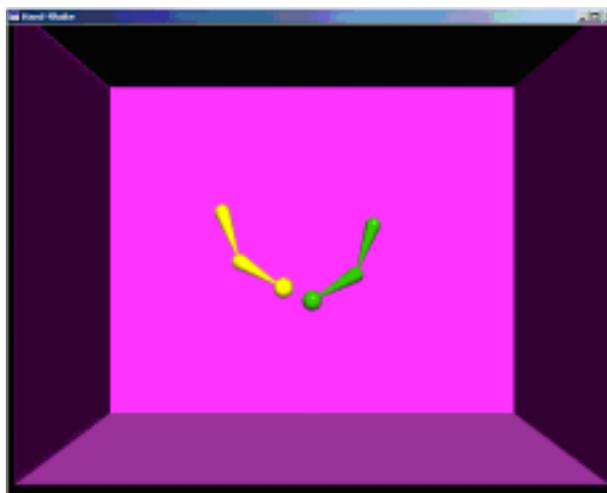


図 2.10: 握手システム

引地ら [17] は共有仮想空間上で力触覚デバイスを用いて、力覚を含むコラボレーションシステムを実装して、QoS について検討を行っている。この研究ではカーソル位置をサーバに送信して、その位置から仮想環境の更新をおこない、更新された環境とカーソルの位置から反力を計算するシステムを構築している。

また触覚を含む共有仮想現実環境技術の応用の1つとして、仮想現実環境を利用した3次元造形支援システムが従来から研究されてきた。仮想現実環境内で自由かつ直観的な物

体変形操作を実現するために、幾何関数によって曲面を近似表現する手法や、現実物体の変形特性に注目した手法などが提案されている。

しかしながら触覚を含む共有仮想現実環境の研究について、実用化されるまでにはいたっておらず、まだこの研究は、発展途上である。

2.4 まとめ

本章では本研究に関連する研究について述べた。

このような力覚情報のネットワークを介した利用には解決すべき問題がある。力覚デバイスのフォースフィードバック機構の制御更新周期は1kHz程度と高速であり、そのフィードバックの計算にはその制御周期に収まる高速な計算処理が必要である。しかし、IPネットワーク等の通信環境では100ミリ秒から数秒の通信遅延が存在し、そのため、力触覚のフィードバック処理をネットワーク上においた場合、安定した力覚の伝送は困難である。このように通信遅延に対して何らかの対策をとる必要がある。

そこで次章では実際に力覚情報をネットワークで扱った場合の操作への影響を示す。

第3章 ネットワーク上での力触覚

3.1 ネットワーク上での力覚の伝送

本章では力覚の伝送を実際にネットワークで伝送した場合においてどのような影響があるかを調べる。この際に構築したシステムを示す。システムは、クライアント-サーバ構成である。このシステム構成で共有仮想環境内で各ユーザが仮想物体を操作するシステムを構築する。

3.1.1 システム構成

システム構成を図 3.1 に示す。

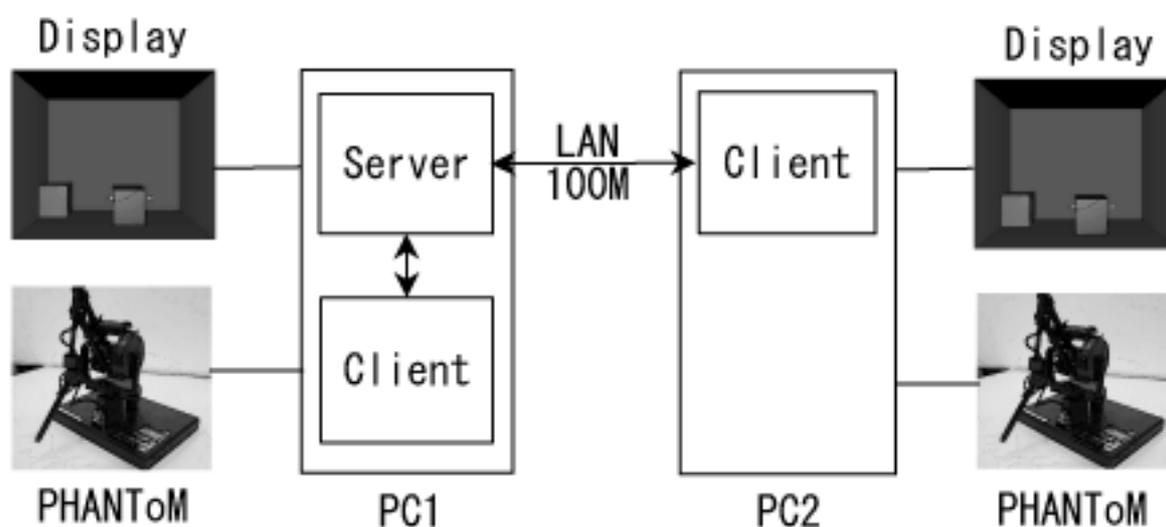


図 3.1: システム構成

本章で用いるシステムはPC1とPC2の2台のPCを使用する。各PCには力触覚デバイスとして位置入力-力出力型のPHANToM (SensAble社)を専用インターフェイス(PCI)を介して接続され、また仮想現実環境をユーザに表示するグラフィックディスプレイを接続している。PC1上では仮想現実環境を管理するサーバプロセスとユーザの操作する環

境を描くクライアントプロセスを実行する。PC2 上ではクライアントプロセスのみ実行し、PC1 上のサーバと通信を行うために 100Mbps の LAN で PC1 と PC2 を接続する。それぞれの PC の環境は表 3.1 のようになっている。サーバプロセスは仮想現実環境を管理し、ユーザのカーソルを操作に基づいて、反力の計算を行い、クライアントに送信するプロセスである。クライアントプロセスは、ユーザが操作する仮想現実環境の描画を行い、サーバから送信されてきた反力をユーザに提示するプロセスである。

	CPU	Main Memory	OS
PC1	Pentium4 2.53GHz	1GB	WindowsXP Pro
PC2	Pentium3 450MHz	256MB	Windows2000 Pro

表 3.1: 構成システムの環境

この環境でシステムの構築を行い、実験を行う。図 3.2 はこのシステムでのサーバとクライアントそれぞれの処理の流れを示したものである。まずサーバからの命令でクライアントに接続された PHANTOM を初期化し、サーバ側に力覚を管理する仮想現実環境を生成する。次にクライアントでは、ユーザに提示する仮想現実環境の画像情報を生成し、力覚デバイス进行操作する。その力覚デバイスの操作を元に、サーバでは操作に対する力覚情報を生成し、クライアントに送信する。サーバで求められた力覚情報から、クライアントに接続された力覚デバイスに操作に対する反力を生成する。このようにして、ユーザに力覚を提示する。

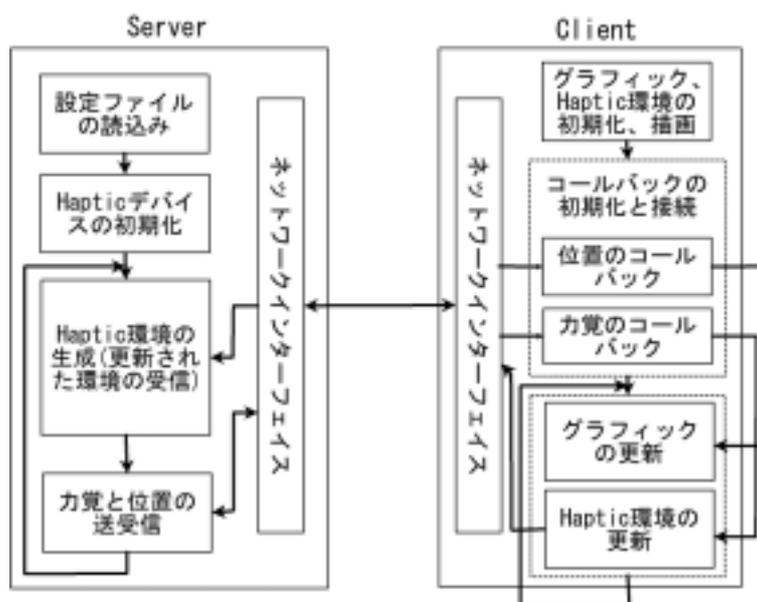


図 3.2: 処理の流れ

PHANToM

MITのTouch Labで開発されたPHANToMは現在最も普及している力触覚デバイスである。図3.3にPHANToMの外観を示す。PHANToMはポイント型で位置入力、力出力のデバイスである。入力は位置の3自由度とポインタの回転の3自由度の計6自由度で、出力は位置のみの3自由度と回転も含めた6自由度のタイプがある。ユーザは先端部分をペン状に持ち、仮想環境に対して位置を入力し、操作の結果を反力を出力として得る。本研究で用いたものは、X, Y, Zの3自由度にピッチ・ヨー・ロールの3自由度を加えたものである。表3.2は本システムで利用したPHANToMのスペックである。このデバイスをPHANToM用アンプに接続し、PCに組み合わされたPCI専用インターフェイスと接続する。



図 3.3: PHANToM の外観

位置精度	800dpi (0.03mm)
最大フォース	8.5N (約 0.8kg 重)
見かけの質量	100g 以下
動作空間 (奥 × 高 × 横)	130 × 180 × 250mm

表 3.2: PHANToM 性能表

VRPN

本章ではVRPNを利用してサーバクライアントシステムを構築した。VRPN(The Virtual Reality Peripheral Network) [19] は、仮想現実システムで使用されるデバイスのアプリケーションプログラムとネットワークを透過的に実行できるように設計されたライブラリである。これによってユーザは haptic デバイス、サウンド、マウスなどのデバイスをネットワークトポロジを意識せずに対応するデバイスを扱うことができる。対応するクライアントライブラリ環境は次のものがある。

- SGI/Irix
- PC/Win32
- PC/Cygwin
- HP700/Hpux
- Sparc/Solaris
- PC/Linux
- Zaurus/Linux

また、サーバ側ライブラリは、次のものがある。

- SGI/Irix
- PC/Win32
- PC/Linux

ネットワークモデル

ネットワークモデルにはいくつかの形態があるが、本研究ではサーバ - クライアントモデルを用いて、システムを構築する。次にサーバ - クライアントモデルについて概要を述べる。

サーバ - クライアントモデルは図 3.4 に示されるように、ネットワークにおいて、クライアントと呼ばれるコンピュータがサーバと呼ばれるコンピュータに対してサービスを依頼するようなシステムである。アプリケーション処理の一部をクライアントで行なうので入力に対する応答が速い。サーバの処理中はほとんど通信は行なわれず、結果のみがクライアントに送られるので効率がよい。

本研究の目的として、共有仮想現実空間での物体操作ということを考える場合、ユーザの操作によってそれぞれのクライアントで仮想環境の状態に差異が生じてしまうため、各クライアントで同期処理を行う必要がある。このような作業では物体の管理を一元的に

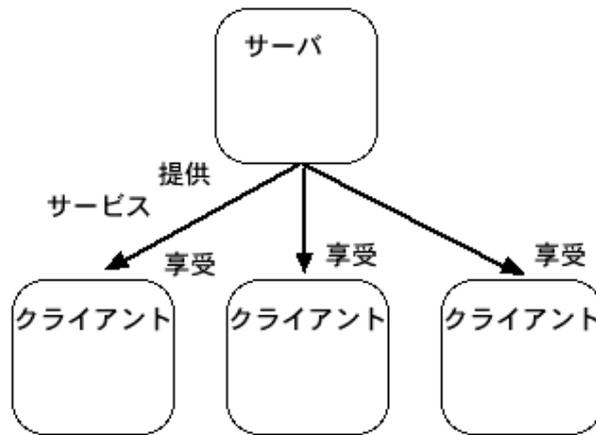


図 3.4: サーバ クライアントモデル

行ったほうが同期処理に都合がよく、またシステムの構成が容易である。そのため本研究ではサーバ - クライアントモデルを採用し、ネットワークを構築する。

スプリング ダンパモデル

人間の力触覚の特性上、力覚の計算は非常に高速で行う必要がある。そこで、スプリング ダンパモデルによる近似を用いて、物体接触時の力覚的な感触を仮想的に表現する。図 3.5 はスプリング ダンパモデルについての各パラメータの関係を示したものである。

スプリング ダンパモデルでは、物体表面の位置座標 X とその微分を用いて接触時に得られる反力 F を表現する。

スプリングモデルにより得られる力を F_k とし、バネ定数を K_s 、物体表面の位置変位量を ΔX とすれば

$$F_k = K_s \Delta x \quad (3.1)$$

で表される。ダンパモデルは力を F_d とし、ダンパ定数を K_d 、物体とカーソルの相対速度を V とすれば

$$F_d = -K_d V \quad (3.2)$$

と表される。これら 2 つを用いて、得られる力覚 F を

$$F = F_k + F_d \quad (3.3)$$

で表す。一般にバネ定数は仮想オブジェクトの硬さや剛性感などに対応し、ダンパ定数は仮想オブジェクトの弾力感や粘りなどと関係している。

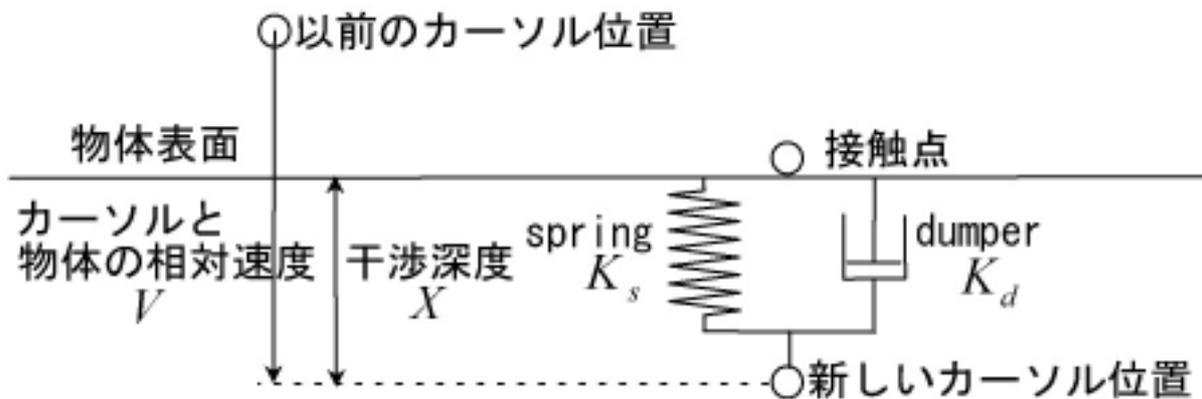


図 3.5: Spring-Damper Model

3.2 力覚の伝送実験

3.2.1 ネットワークの更新周期による操作への影響

これらのシステムで、新しい制御手法を提案する予備実験として、ネットワークで接続されたサーバとクライアント1、2台でタスクを行い、更新周期を変化させることで操作にどのように影響があるかを調べる。

実験方法として、仮想環境内で図 3.6 に示すような半径 5cm の球のオブジェクトを構築し、ユーザがクライアントで球の表面を PHANToM でなぞった時の、球の中心からのカーソルの距離を計測する。クライアントの数が1台の場合と、2台の場合で実験を行う。

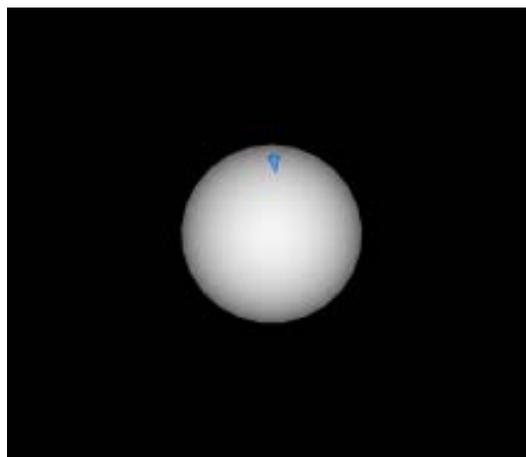


図 3.6: 操作画面

クライアント 1 台での実験結果

図 3.7 にクライアントが 1 台で、ネットワーク更新周期が 30Hz の場合でのカーソルの球の中心からの距離を示す。図 3.7、図 3.8、図 3.9、図 3.10、図 3.11、図 3.12、図 3.13 はクライアントが 1 台で、ネットワーク更新周期が 60、100、150、200、300、500Hz の場合での結果を示す。

実験方法から、球の表面をなぞり続けることが出来ることが望ましいため、カーソルの距離が 5cm の付近に収束してあるのが良い。

まず更新周期が遅い場合は、5cm より大きい値が現れている。これは、カーソルが球に侵入した場合、即座に反力を生成して、球への侵入を防ぐ力が働かなければいけないが、更新周期が遅いため、反力の生成が遅れる。そのためカーソルが過剰にめり込み、スプリング ダンパモデルの計算方法より、予定より強い反力が出力されるため、カーソルが球の表面を適切になぞることが難しい。更新周期が速くなってくると、カーソルの中心からの距離が 5cm を下回るようになるが、これは更新周期が速くなる事によって、反力の生成に遅れがなくなるためである。しかし、まだ更新速度が十分でないために、波形の上下が激しい。更新周期が十分速くなると、5cm を少し下回るぐらいで安定して、球の表面をなぞることが出来る。図 3.14 は図 3.7、図 3.8、図 3.9、図 3.10、図 3.11、図 3.12、図 3.13 のグラフから求められた、クライアント 1 台の場合の更新周期の変化による中心からの距離の平均と標準偏差をあらわすグラフである。Client1 台の結果から、更新周期が速くなると平均は良くなり、標準偏差も小さくなるので球の表面に近いところをなぞることが出来る。またこれらの結果から操作精度が 1cm 程度の操作を行う場合は 500Hz 必要である。

クライアント 2 台での実験結果

図 3.15 は、PC1 でのクライアント 2 台の場合の更新周期の変化による球の中心からの距離の平均と標準偏差のグラフ、図 3.16 は PC2 でのクライアント 2 台の場合の更新周期の変化による球の中心からの距離の平均と標準偏差のグラフである。

図 3.15、図 3.16 より、PC1 が 100Hz 辺りをピークに徐々に悪くなるのに対して、PC2 は 30Hz が一番よく、それ以上は変化が激しい。これは PC1 と PC2 の構成による違いによるものと考えられる。またクライアント 1 台の場合と異なる傾向のグラフを示しているのは、PC1 でサーバとクライアントプロセスが同時実行されているためで、その負荷によって更新周期が高速になっても、結果が良くなり、反対に結果が悪くなってしまふ。またそのために、クライアントプロセスしか実行されていない PC2 でも、同じように更新周期の高速化が良い結果を導いていない。これらから、使用環境ではサーバで反力生成を行うには 100Hz 程度がシステムの限界である。

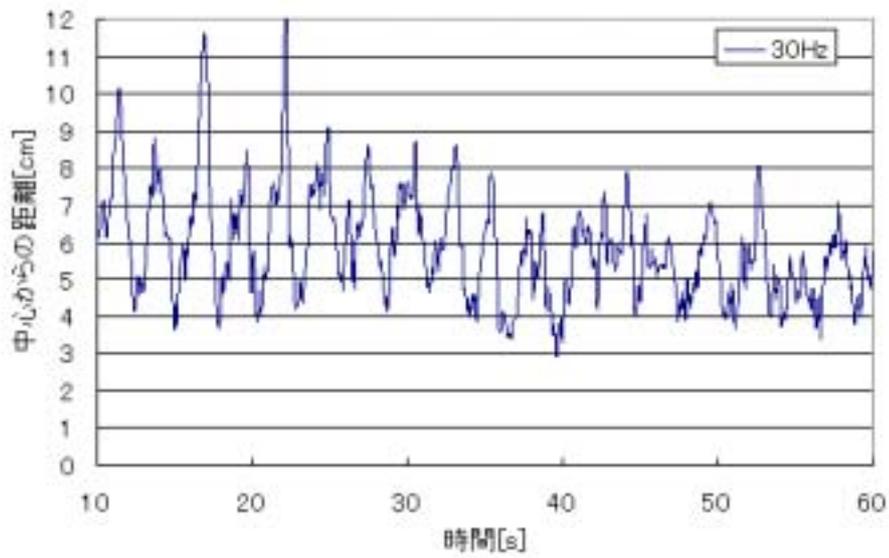


図 3.7: クライアント 1 台の場合のカーソルの距離 (30Hz)

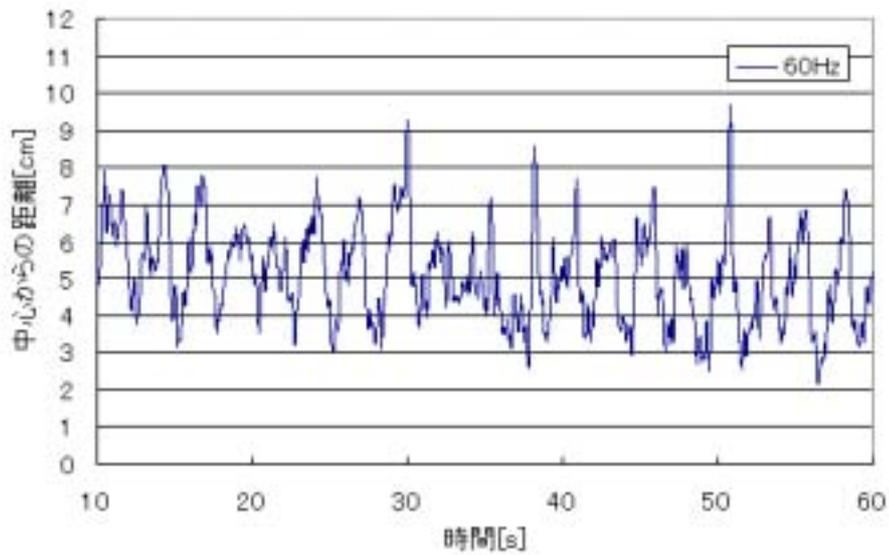


図 3.8: クライアント 1 台の場合のカーソルの距離 (60Hz)

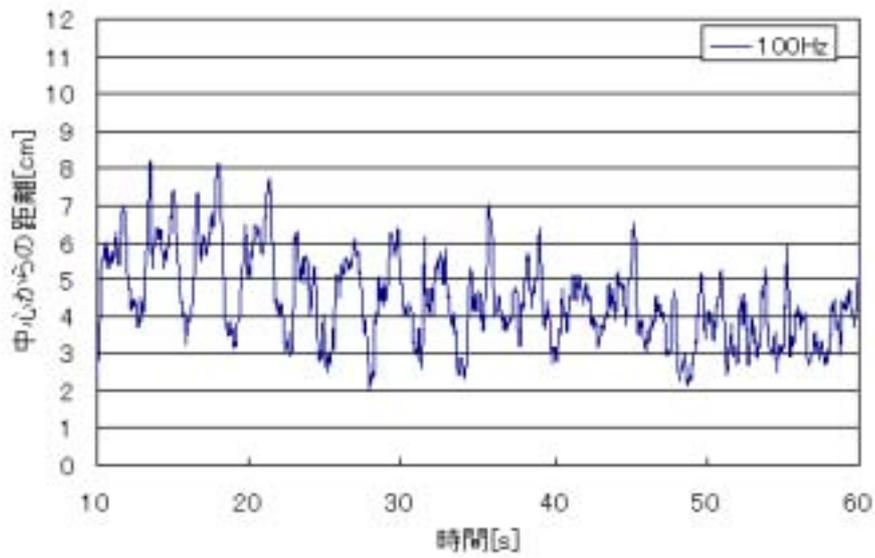


図 3.9: クライアント 1 台の場合のカーソルの距離 (100Hz)

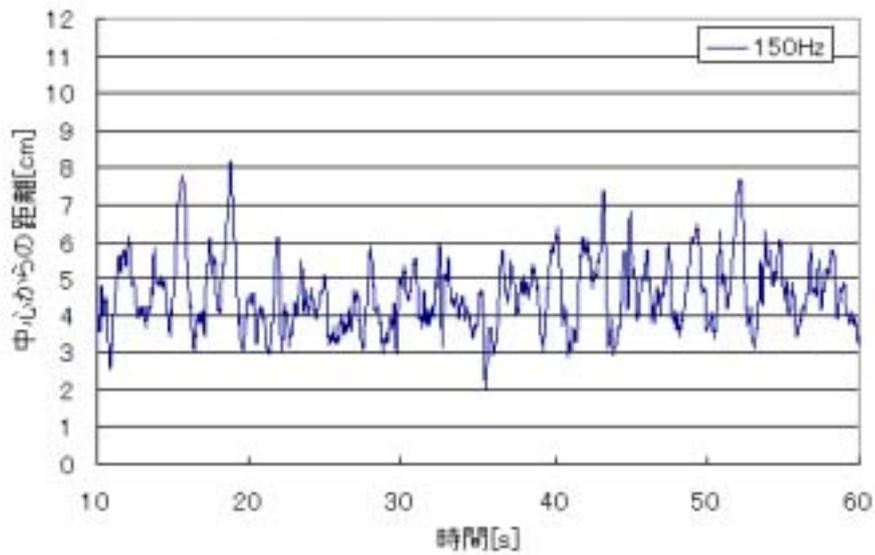


図 3.10: クライアント 1 台の場合のカーソルの距離 (150Hz)

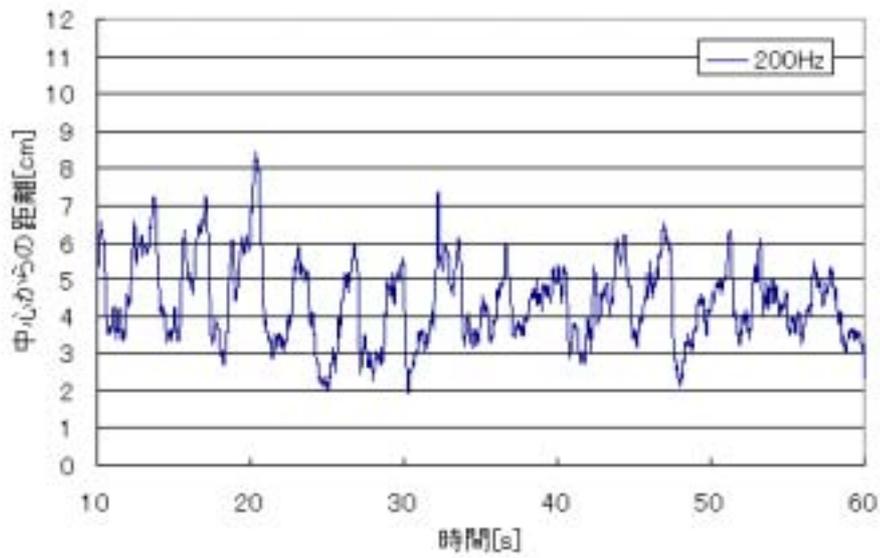


図 3.11: クライアント 1 台の場合のカーソルの距離 (200Hz)

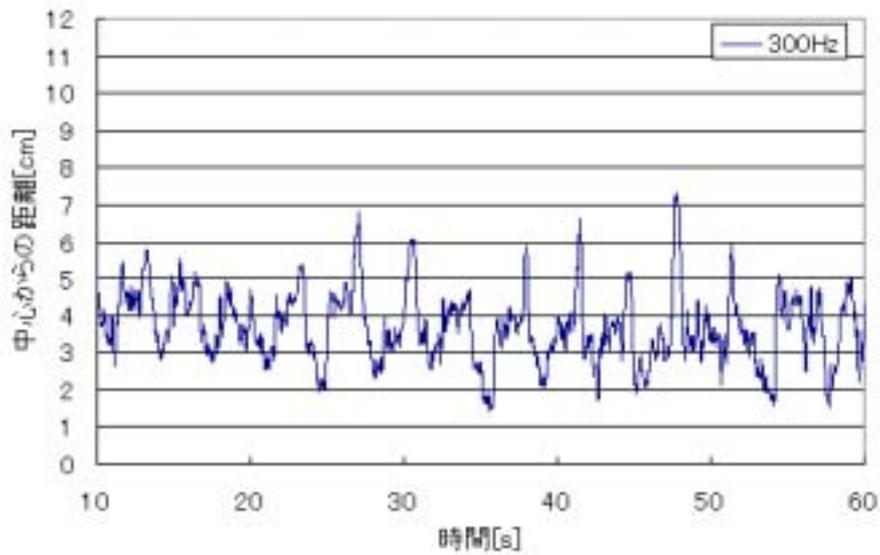


図 3.12: クライアント 1 台の場合のカーソルの距離 (300Hz)

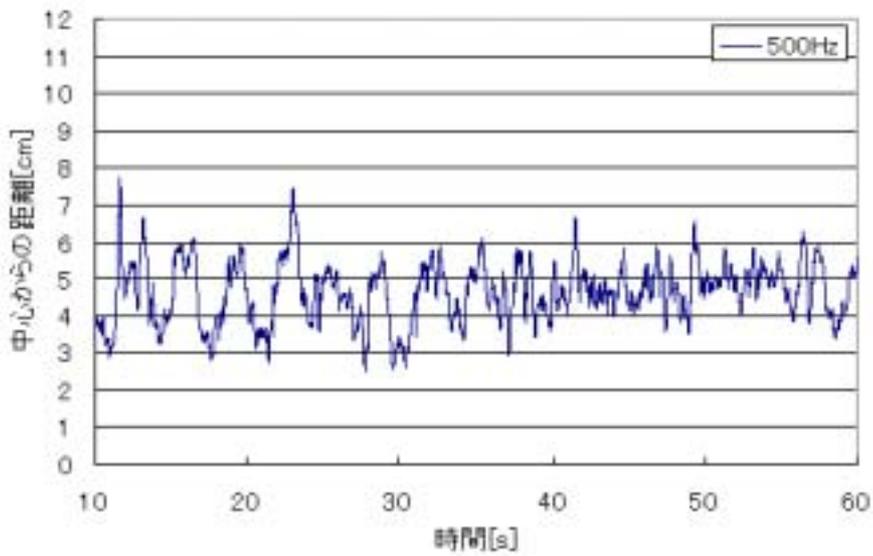


図 3.13: クライアント 1 台の場合のカーソルの距離 (500Hz)

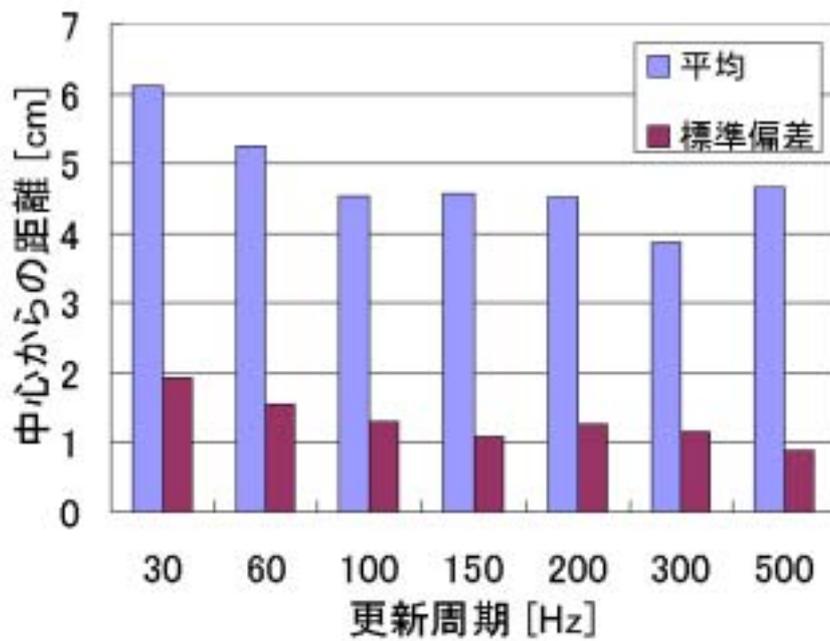


図 3.14: クライアント 1 台の場合の更新周期による平均と標準偏差

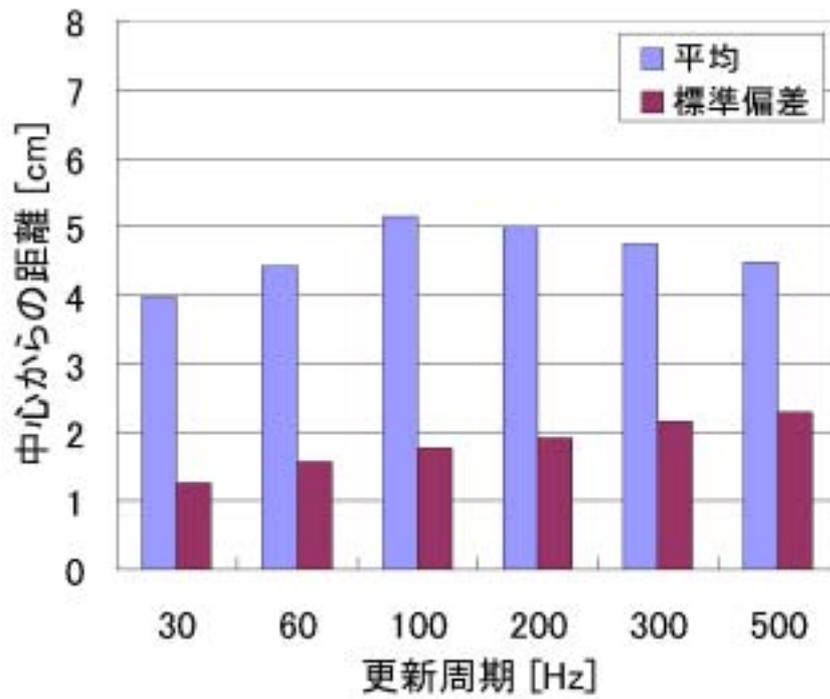


図 3.15: クライアント 2 台の場合の更新周期による平均と標準偏差 (PC1)

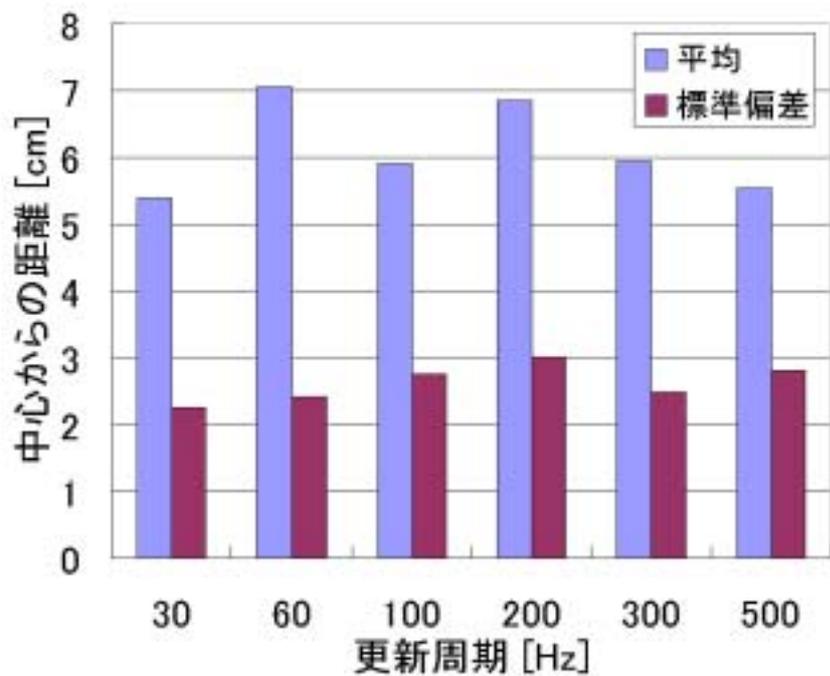


図 3.16: クライアント 2 台の場合の更新周期による平均と標準偏差 (PC2)

3.2.2 ネットワークの遅延による操作への影響

サーバ1台とクライアント1台接続した場合、通信遅延（固定）がある条件で、カーソルの操作がどのようになるかを調べる。

実験は前節と同様に、半径5cmの球の表面をPHANToMを使用してなぞり、そのときの球の中心からのカーソルの距離を計測する。ネットワークの更新周期を60Hz, 100Hzに変化させて繰り返す。

また、通信遅延（固定）は0, 30, 50, 80, 100, 150, 200 [ms] の場合で実験を行う。

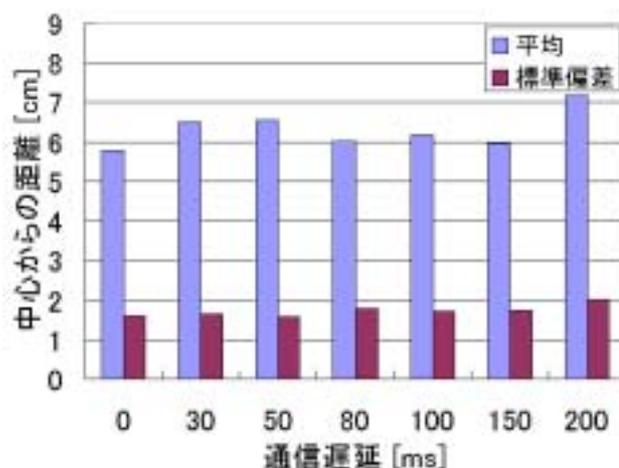


図 3.17: 通信遅延がある場合: 60Hz

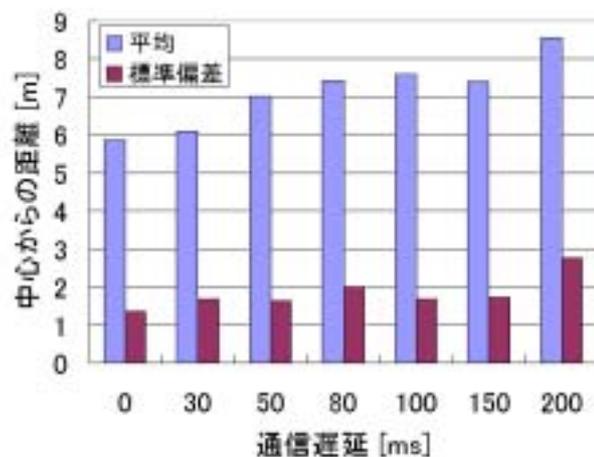


図 3.18: 通信遅延がある場合: 100Hz

次に、ネットワークの更新周期 100Hz の場合のカーソルの軌跡を示す。

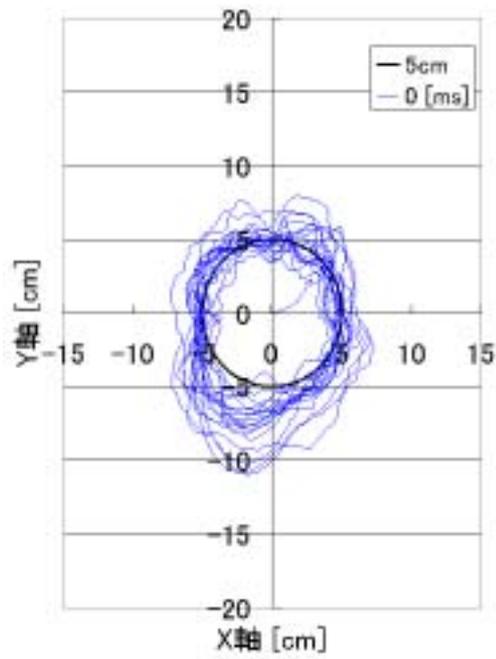


図 3.19: カーソルの軌道 遅延 0 [ms]

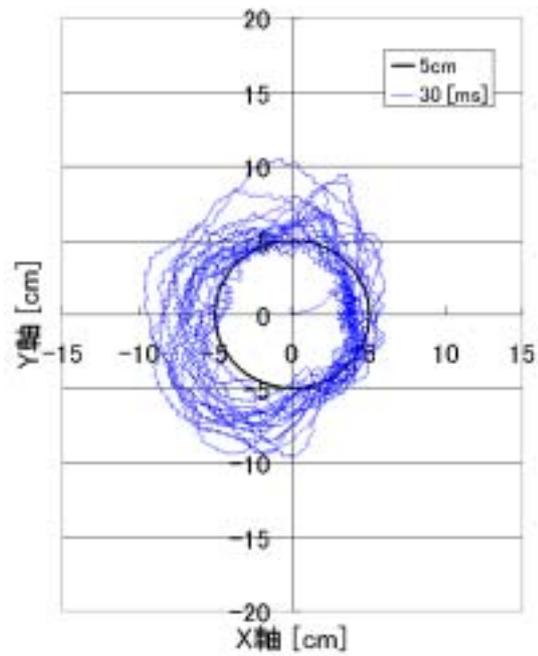


図 3.20: カーソルの軌道 遅延 30 [ms]

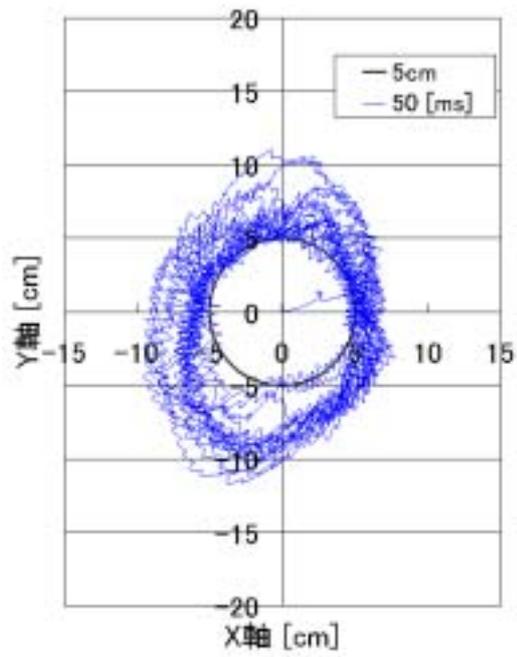


図 3.21: カーソルの軌道 遅延 50 [ms]

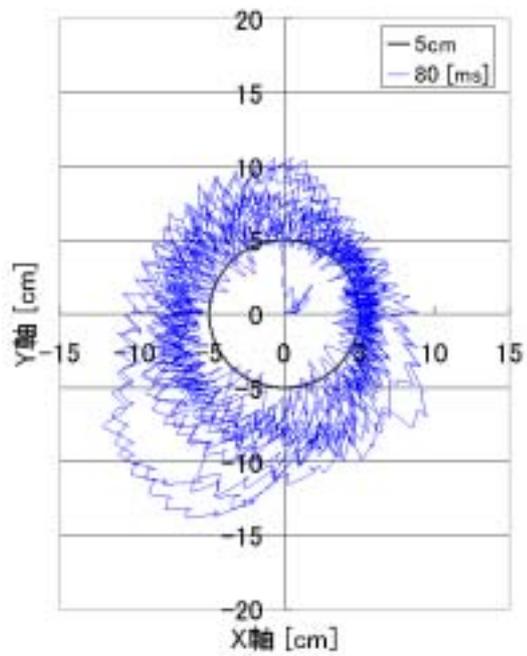


図 3.22: カーソルの軌道 遅延 80 [ms]

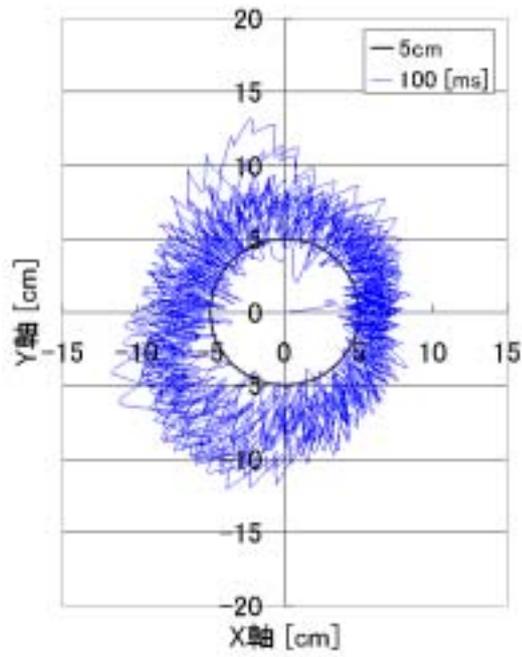


図 3.23: カーソルの軌道 遅延 100 [ms]

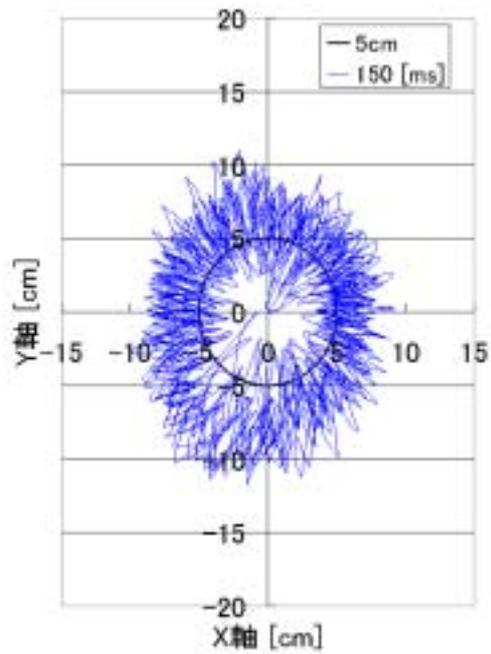


図 3.24: カーソルの軌道 遅延 150 [ms]

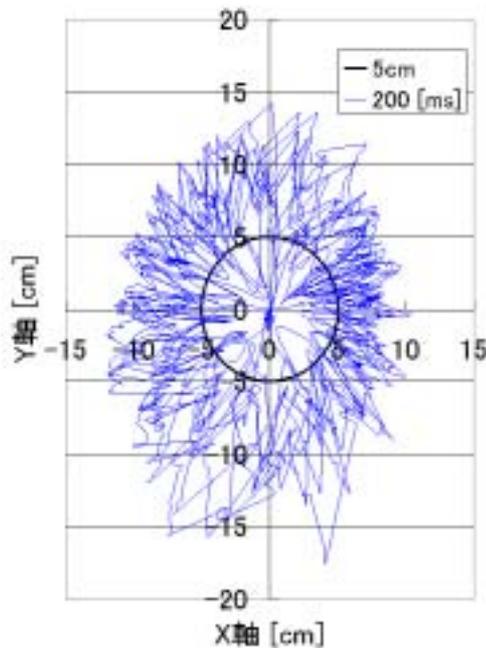


図 3.25: カーソルの軌道 遅延 200 [ms]

通信遅延がある場合の実験結果

図 3.17 は更新周期 60Hz で通信遅延がある場合の、通信遅延の変化によるカーソルの球の中心からの距離から求めた、平均と標準偏差のグラフ、図 3.18 は更新周期 100Hz で通信遅延がある場合の、通信遅延の変化によるカーソルの球の中心からの距離から求めた、平均と標準偏差のグラフである。

図 3.17、図 3.18 より、通信遅延が小さい場合は 100Hz のほうが標準偏差が小さく、通信遅延が大きい場合は 60Hz のほうが標準偏差が小さい。これから、高い更新周期の方が、通信遅延の影響を受けやすいことが分かる。

これらの結果からサーバ-クライアント間の更新周期を下げる必要がある。

図 3.19 は、通信遅延が 0ms の場合の、カーソルの軌跡を X-Y 平面に表したものである。同様に、図 3.20、図 3.21、図 3.22、図 3.23、図 3.24、図 3.25 は、遅延時間がそれぞれ、30、50、80、100、150、200ms の場合の、カーソルの軌跡を X-Y 平面に表したものである。軌跡からも分かるように、ネットワークの通信遅延によって、安定したカーソル操作が維持できないことがわかる。サーバで反力を計算する場合、遅延時間が大きくなると操作は大きく乱れる。よって反力をサーバで計算しては滑らかな力覚を提示できない。

3.3 まとめ

プロトタイプシステムとして力覚の伝送と仮想環境の管理をすべてサーバで行った場合には、クライアントの性能によって異なるが通信の周期が高速になってもユーザの操作性が良くなるとは限らず、逆に操作が不安定になることを検証した。また通信遅延がある場合、力覚の更新周期を高速にしても、それが操作の安定に繋がらないことが確かめられた。それらを踏まえて、次章では、システムの制御法を提案する。

第4章 協調作業による物体操作システムの構築

4.1 はじめに

これまでに実験した結果より、仮想現実環境で力覚を扱う場合、ネットワークを利用する際に通信回線の影響を大きく受けることが明らかになった。これから、仮想現実環境化での新しいシステム制御法を用いた物体操作システムを構築することとする。この章ではその手法を述べ、本研究で構築する、力覚を伴う協調作業による物体操作システムの構築に向けて、通信遅延に対処するためのシステム制御法を提案する。また仮想物体の物理計算を高速に行うための計算手法について検討を行う。これらを踏まえ、構築したシステムの実装、評価を行い、提案手法の妥当性を検討する。

4.2 物体操作に適応した力覚提示法

サーバ側で反力を算出する場合、ユーザが触覚デバイス进行操作して位置情報を入力してから、それに対する反力が出力される一連の処理の間にネットワークが介在する。その結果、ユーザの操作に対する反応が遅れ、入力に対して遅延が生じてしまう。

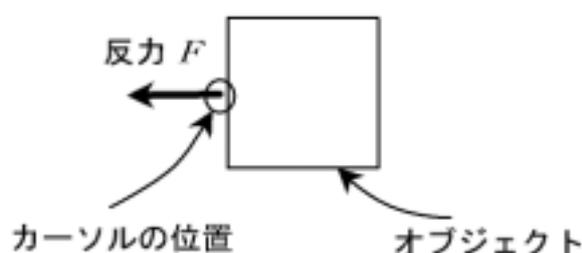


図 4.1: 通常の反力の生成

図 4.1 に示すように、遅延がない場合では物体に接触した場合にすぐに反力を生成して、物体への過剰な侵入を防がれる。

図 4.2 のような、反力の生成に遅延がある場合は、ユーザのカーソルが物体の中に侵入していても反力の生成が行われず、そのままカーソルが物体の中に侵入し続ける。その結

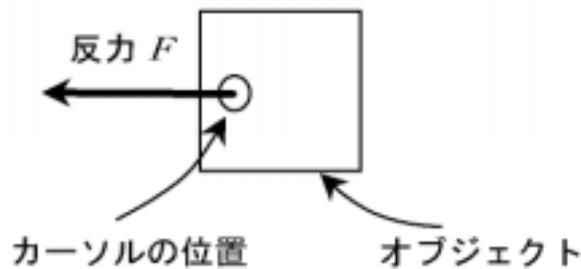


図 4.2: 遅延がある場合の反力の生成

果、反力計算手法であるスプリング - ダンパモデルの特性より、生成される反力は物体の侵入量によって求められる。そのため計算される反力は本来の位置ではなく、遅れた位置でのパラメータを用いてしまうため、ユーザが想定するよりも過剰な反力が生成される。このことは、前章で行った、力覚生成をサーバで行う実験で示された通りである。

実験からネットワークを介することによって生じる遅延による反力生成の問題がわかったので、そこで次の手法を提案する。

1. サーバの仮想環境をクライアントに複製する。
2. ユーザはクライアントに複製された環境で仮想物体を操作し、反力の計算をクライアントで行う。
3. ユーザが操作した仮想物体の位置とカーソル操作履歴をサーバに送信し、それによってサーバの仮想環境を更新する。この時、各クライアントで操作した仮想物体の位置に矛盾が生じるので、サーバで仮想物体位置とカーソル操作履歴を用いて仮想物体の位置を定める。
4. 更新した仮想物体の位置を再びクライアントに送信して、各クライアント間の仮想物体の位置の同期を取る。

この手法によってユーザ間の仮想環境の一貫性を保ちながら通信遅延による操作の乱れを防ぐ。特徴としてはサーバで反力を計算するのではなく、クライアントで反力を生成することで、反力提示からネットワークの更新周期や、ネットワークの通信遅延の影響を取り除くことが出来る。また仮想環境の状態をネットワークで送受信することで、力覚情報をネットワークで通信するよりも更新周期を遅くすることが可能である。そして、サーバで仮想物体の位置情報を一括して管理することで、仮想物体の位置情報の矛盾を修正することが可能である。

ここでクライアント側で反力を計算する処理を詳細に説明する。

1. ユーザはクライアント側の仮想環境に対して操作を行う。

2. ユーザの操作に対して、クライアント側でカーソル、オブジェクト間の反力の算出、オブジェクトの状態更新を行う。
3. 更新されたオブジェクトの位置をサーバに送信する。
4. サーバ側においては、クライアントより受信した各クライアント側の仮想環境内のオブジェクトの情報より、サーバでのオブジェクトの位置情報を定め、位置情報オブジェクトの状態更新を行う。
5. 更新されたオブジェクトの状態をサーバからクライアントの送信する。
6. クライアント側において、受信したオブジェクトの状態を元にクライアントで管理している仮想空間を更新する。
7. クライアント側において、受信したオブジェクトの状態をもとにグラフィックスを更新する。

4.3 システム構成

本研究で構築する、物体操作システムは次の4つの要素からなる。

- 力覚管理
- オブジェクト管理
- グラフィックス管理
- ネットワーク管理

(1) 力覚管理

この要素は力覚情報を処理し、ユーザに提示する要素である。ユーザはこの要素で管理された力覚デバイス PHANTOM を使用して、仮想環境内でのカーソル動かし、環境に接触することで利用者に入力に対する反力として力覚を出力する。

(2) オブジェクト管理

この要素は仮想環境内でのオブジェクトの位置情報を管理しており、ユーザの操作による振る舞い、また仮想物体間の衝突や物理法則に従った運動を計算している。

(3) グラフィクス管理

この要素はオブジェクト管理部で扱う仮想物体の位置をユーザに提示する。グラフィックスの描画には OpenGL を利用しており、この更新は力覚管理部やオブジェクト管理部から独立の更新タイミングである。

(4) ネットワーク管理

このシステムもサーバ クライアントモデルを採用しており、その相互接続にはこのネットワーク管理部が処理する。今回のネットワークは IP ネットワークを利用して接続し、通信プロトコルとして TCP/IP を使用した。図 4.3 はサーバ クライアント間の通信シーケンスを表したもので、このシーケンスに従って、サーバ クライアント間で通信を行う。

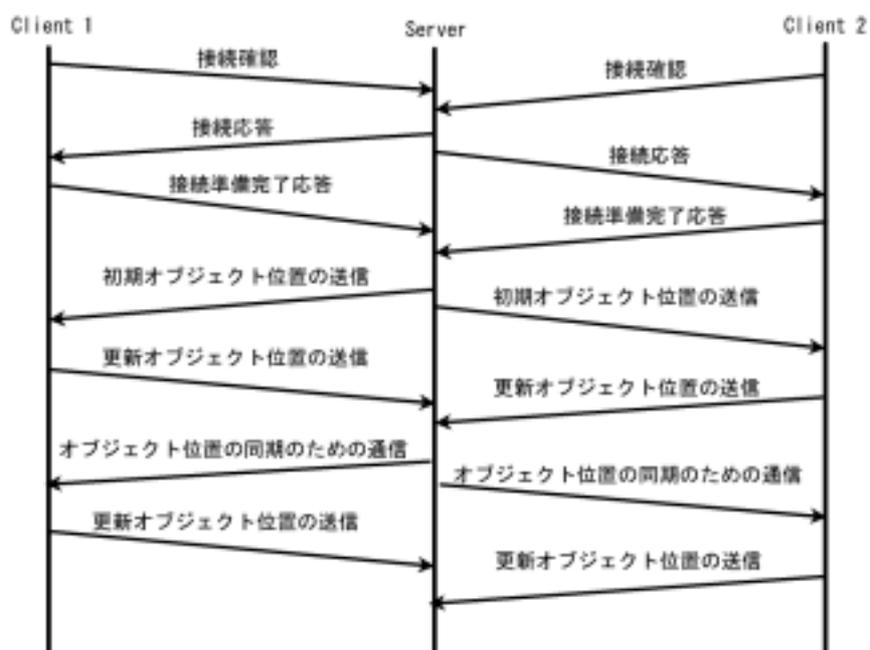


図 4.3: シーケンス図

本システムはこれらの要素から構築されたサーバとクライアントから出来ている。

OpenGL

Silicon Graphics 社が中心となって開発した 3 次元グラフィックスのためのプログラムインターフェイスである。グラフィック表示は OpenGL を用いて描画を行った。本研究ではグラフィックス管理の部分で OpenGL を用いて、仮想現実環境の描画を行う。

GHOST

GHOST は PHANToM を用いて仮想環境を構築するための開発キットである。ユーザは GHOST を用いることによって簡単に触覚の提示を含む仮想環境を構築できる。本研究では、力覚管理部で GHOST を利用して、力覚を含む仮想環境を構築した。

マルチスレッドライブラリ

システムを高速化するためにそれぞれのプログラムをマルチスレッド化した。その際には、Mcl というマルチスレッドプログラミング用 C++ クラスライブラリを利用した。本研究では、4つの要素を同時実行するため、それぞれの負荷が互いに悪影響を与えるのを避けるために、各要素を別スレッドにして、プログラムを実行している。図 4.4 はサーバ、クライアントプロセスと各要素によるスレッドの関係を示したものである。

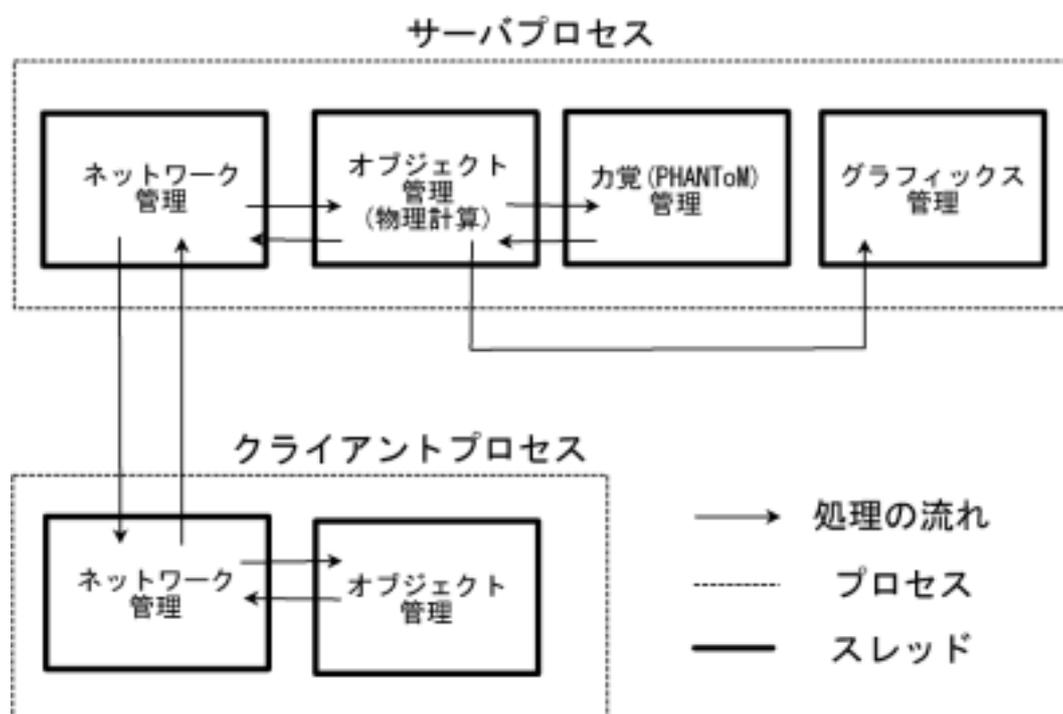


図 4.4: 処理の流れ

4.4 リアルタイム物理モデル

力覚提示を行うためには、高い更新周期 (300Hz ~ 1kHz 以上) で制御を行う必要がある。また、力覚インタラクションでは、シミュレーションの更新に必要な計算時間が制御周期

を超えることがあると、安定な力覚提示が出来ず、問題となる。

物体の運動は、物体に働く力が求まれば、運動方程式の数値解として求まる。

$$F_{object} = F_{reaction} - K_{damper} \times V_{in} \quad (4.1)$$

$$A = \frac{F_{object}}{m_{object}} \quad (4.2)$$

$$V_{out} = V_{in} + A\Delta t \quad (4.3)$$

$$\Delta X_{out} = V_{out}\Delta t \quad (4.4)$$

この式によって計算することで、 Δt での仮想物体の位置が求まる。ここで F_{object} は実際に物体に働く力、 $F_{reaction}$ は操作によって物体に加える力、 K_{damper} は、物体の弾力感によって決定するダンパ係数、 A は物体の加速度、 m_{object} は、物体の質量、 V_{out} は、物体の速度、 V_{in} は操作による力を加えるものの速度、 X_{out} は物体の位置である。重力、バネによる力、動摩擦など、場の力や物体の位置・速度によって決まる力は簡単な計算で求まる。しかし、抗力や静止摩擦など複数物体の位置関係を部分的に拘束する力（拘束力）の計算は難しい。それを解決する物理モデルを用いた研究についていくつか述べる。

解析的な手法

Baraffら [24] [25] は、運動方程式と拘束条件から接触力を解く手法を用いることを提案している。Baraffらの手法は接触の数 n に対して $O(n^3)$ の計算時間を要する。Mirtichら [26] は衝突時に2物体に働く撃力によって抗力を表現する手法を提案している。この手法は衝突が発生した時刻を求め、2物体の衝突を順に処理する。この手法は、1衝突の処理にかかる計算時間は短いですが、短期間に多数の接触が起きる場合、多くのステップを処理する必要がある。そのため、一定期間の問題に多大な計算時間を要することがある。

解析的な手法による解法

剛体の運動は以下の式で与えられる。

$$f = \frac{dmv}{dt} \quad (4.5)$$

$$\tau = \frac{dI\omega}{dt} \quad (4.6)$$

$$mv(t + \Delta t) = mv(t) + f\Delta t \quad (4.7)$$

とで拘束違反を解消する方法である。接触力の計算にペナルティ法を用いると、接触の数 n に対して計算時間が $O(n)$ となり、高速処理が可能である。また、運動方程式と無関係に、物体の位置、速度だけから接触力を求めることができるので、力覚ポインタを物体モデルと区別なく扱える。

ペナルティ法による解法

ペナルティ法 (図 4.6) は式 4.9 を解かずに計算する方法で、拘束を侵した分だけペナルティとして力を加えて問題を解決する。

$$f = kr + b\ddot{r} + c \quad (4.11)$$

から

$$f = kl + bv \quad (4.12)$$

とし、そのうち拘束を満たすだろうということで拘束を解いている。ここでバネの伸び l 、物体の相対速度 v とする。この機構をオブジェクト管理部に組み込む。

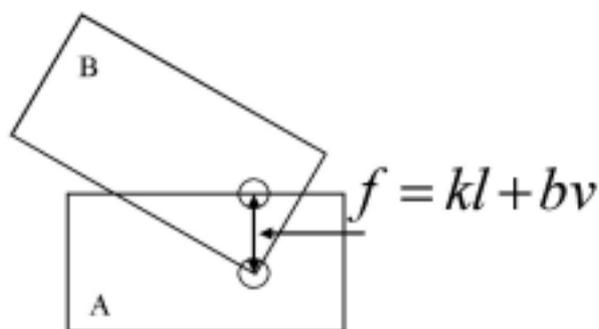


図 4.6: ペナルティ法

参考までに、図 4.7 に長谷川ら [16] によって調べられた解析的な手法とペナルティ法を用いた場合の計算速度の比較を示す。このようにグラフからみてもペナルティ法が高速なことが分かる。

このような研究から力覚を扱う物理モデルとしてペナルティ法は非常に望ましいといえる。よって本研究でもペナルティ法を採用する。

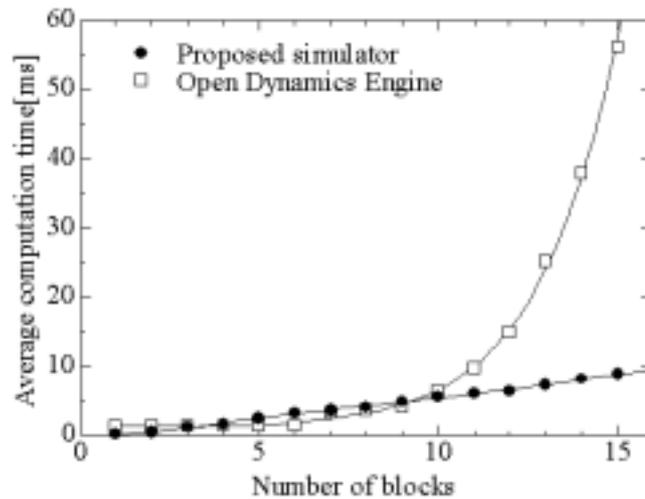


図 4.7: 解析法とペナルティ法の計算速度の比較

4.5 システムの評価実験

4.5.1 サーバ1台とクライアント1台でのシステムの評価

構築したシステムを評価するために、実際に操作を行って、どのように仮想物体が振る舞うか検討する。サーバとクライアント1台の場合で接続したとき、通信遅延が存在する条件で、操作にどのような影響があるか調べる。実験方法として、あらかじめ指定した軌道にオブジェクトを動かす操作を行った際の、サーバとクライアントの仮想物体の位置を計測する。

この実験では以下の仮定をおく。

- 通信遅延は固定値とし、0, 30, 50, 80, 100 [ms] の値をとるものとする。
- 表現できる物体は剛体のみで回転は考慮しない。
- 操作タスクを容易にするため、物体の移動方向は図 4.8 のように X 軸のみに限定。

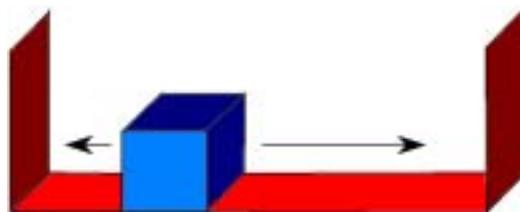


図 4.8: 物体操作タスク

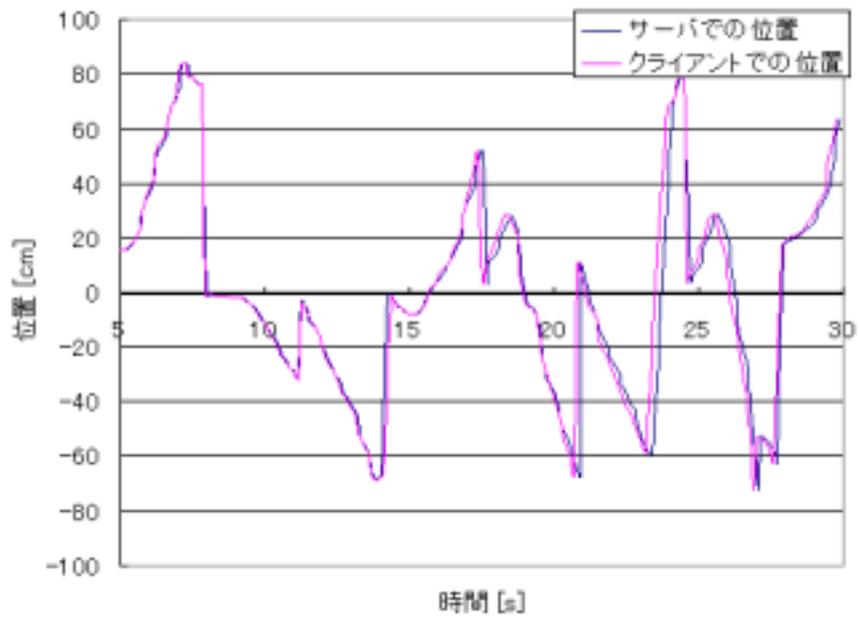


図 4.9: サーバ 1 台、クライアント 1 台の場合での仮想物体の位置 (X 軸)、遅延時間 0[ms]

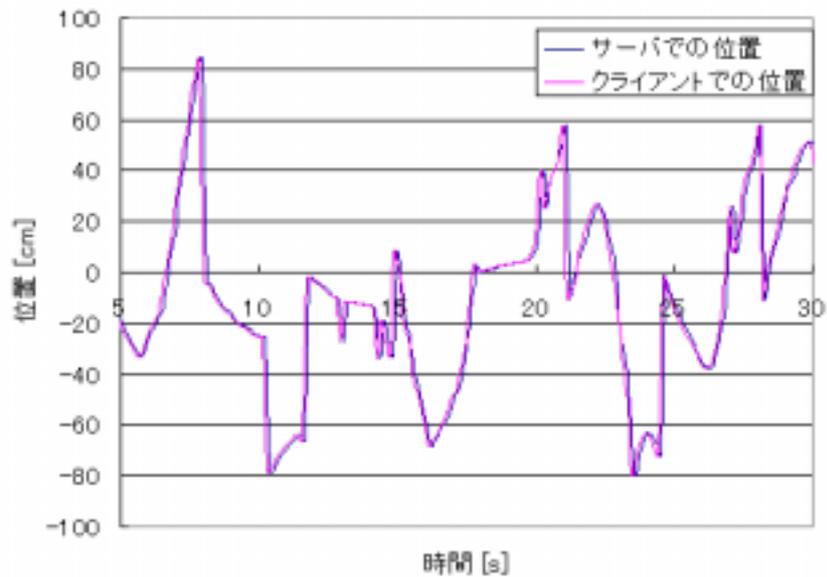


図 4.10: サーバ 1 台、クライアント 1 台の場合での仮想物体の位置 (X 軸)、遅延時間 30[ms]

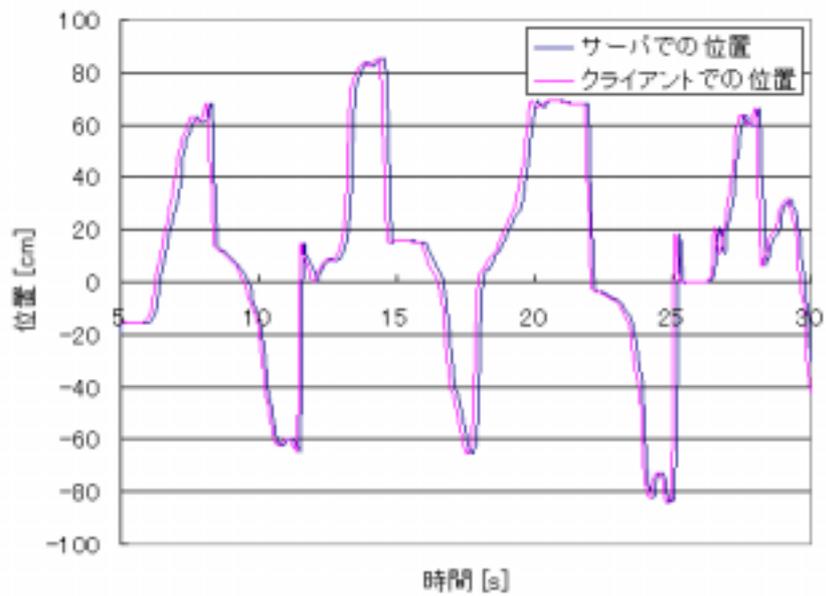


図 4.11: サーバ 1 台、クライアント 1 台の場合での仮想物体の位置 (X 軸)、遅延時間 50[ms]

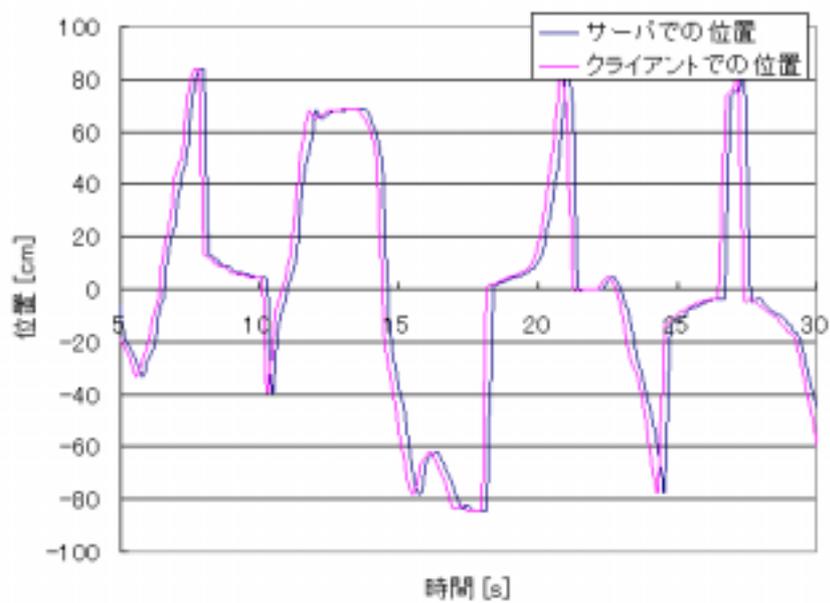


図 4.12: サーバ 1 台、クライアント 1 台の場合での仮想物体の位置 (X 軸)、遅延時間 80[ms]

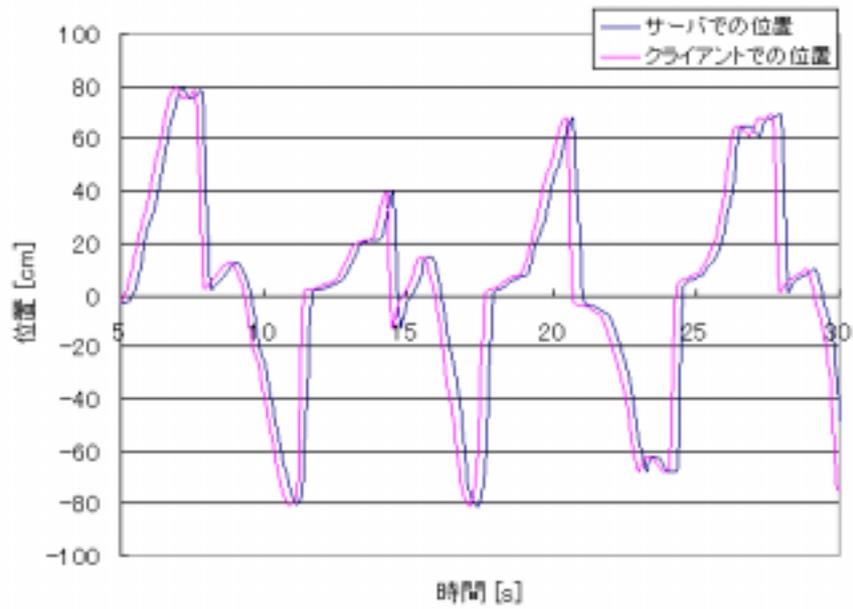


図 4.13: サーバ 1 台、クライアント 1 台の場合での仮想物体の位置 (X 軸)、遅延時間 100[ms]

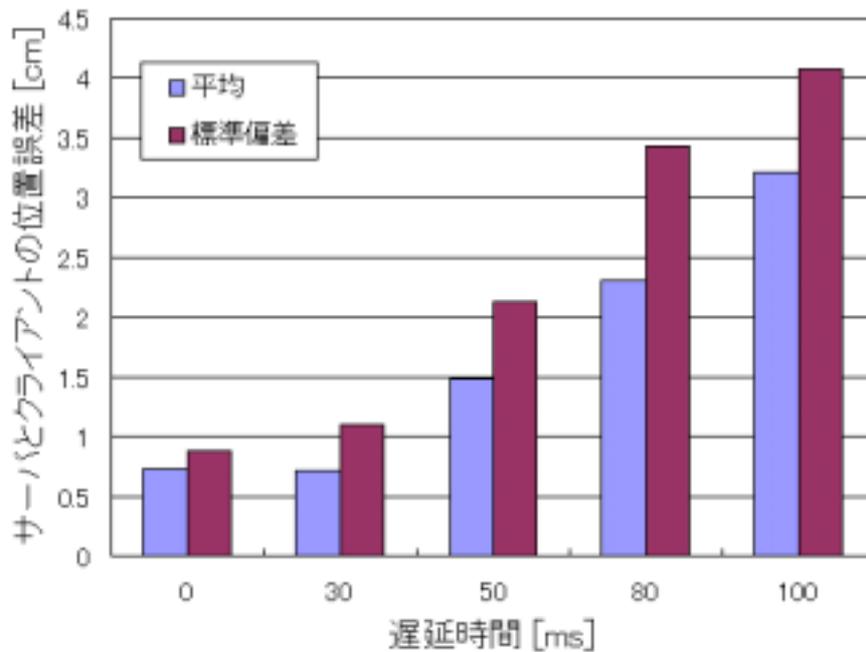


図 4.14: サーバとクライアントの通信遅延による位置誤差

サーバ1台とクライアント1台でのシステムの結果

図 4.9 は通信遅延が 0ms の場合のサーバ1台、クライアント1台の場合でのサーバの仮想環境の仮想物体の位置とクライアントの仮想環境の仮想物体の位置を比較したグラフである。グラフから、サーバの仮想環境とクライアントの仮想環境が一致しているのがわかる。X はワークスペースの中心を基準として、右側を + 方向、左側を - 方向として計測している。図 4.10、図 4.11、図 4.12、図 4.13 は、通信遅延が 30、50、80、100ms と変化させた場合のサーバの仮想環境の仮想物体の位置とクライアントの仮想環境の仮想物体の位置を比較したグラフである。グラフから通信遅延が大きくなると、サーバとクライアントの仮想環境のずれが大きくなるのが分かる。図 4.14 はサーバとクライアントの同時刻での通信遅延による位置誤差である。このグラフからも通信遅延が大きくなると、仮想環境のずれが大きくなるのが分かる。しかし、遅延時間が増加することでサーバとクライアント間の同時刻での仮想物体の位置誤差は増大するが、クライアントの位置誤差が蓄積されて大きくなるわけではない。

4.5.2 サーバ1台とクライアント2台でのシステムの評価

サーバとクライアント2台の場合で接続したとき、通信遅延がある条件で、操作にどのような影響があるか調べる。実験方法として、前節と同様にあらかじめ指定した軌道にオブジェクトを動かす操作を行った際の、サーバとクライアントの仮想物体の位置を計測する。

前節と同様にこの実験では以下の仮定をおく。

- 通信遅延は固定値とし、0, 30, 50, 80, 100 [ms] の値をとるものとする。
- 表現できる物体は剛体のみで回転は考慮しない。
- 操作タスクを容易にするため、物体の移動方向は X 軸のみに限定する。

またこのときのシステム環境を変更する。

	CPU	Main Memory	OS
PC1	Pentium4 2.53GHz	1GB	WindowsXP Pro
PC2	Pentium3 450MHz	256MB	Windows2000 Pro
PC3	PentiumM 1.5GHz	768MB	WindowsXP Pro

表 4.1: 構成システムの環境 2

表 4.1 のように PC3 を追加し、PC3 でサーバプロセスを実行する。これにより、各 PC で実行されるプログラムは、1 つのみになる。また変更されたシステム構成は図 4.15 のように変更した。なお更新周期は特に定めず、可能な限り高速に実行されるように設定している。

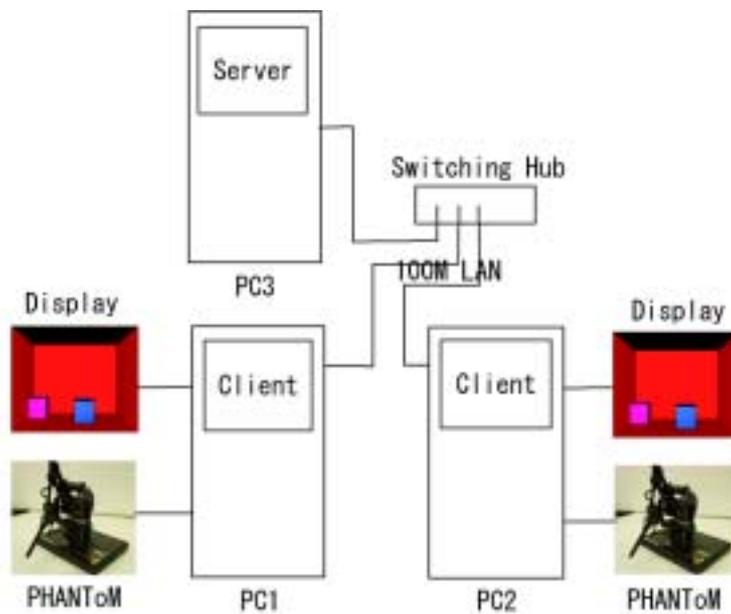


図 4.15: システム環境 2

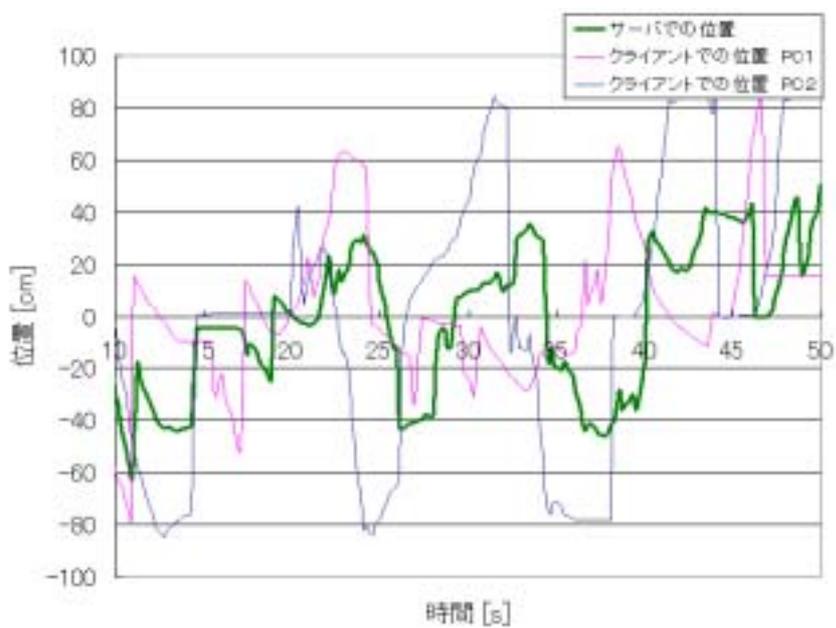


図 4.16: サーバ 1 台、クライアント 2 台の場合での仮想物体の位置 (X 軸)、遅延時間 0[ms]

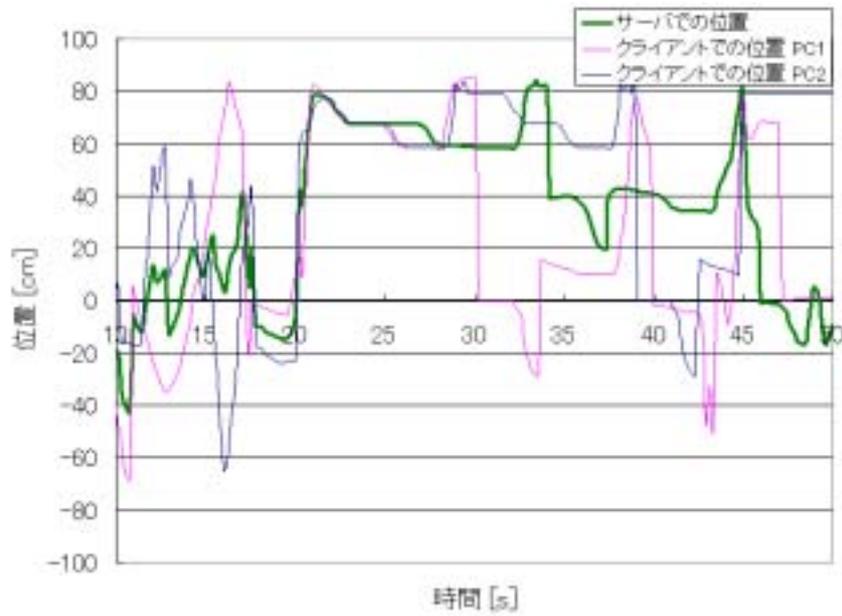


図 4.17: サーバ1台、クライアント2台の場合での仮想物体の位置(X軸)、遅延時間 30[ms]

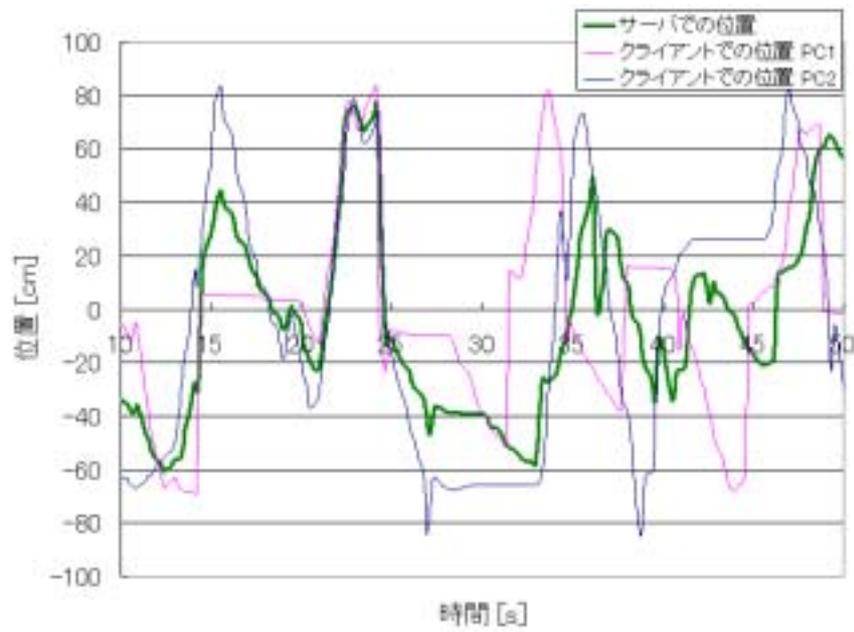


図 4.18: サーバ1台、クライアント2台の場合での仮想物体の位置(X軸)、遅延時間 50[ms]

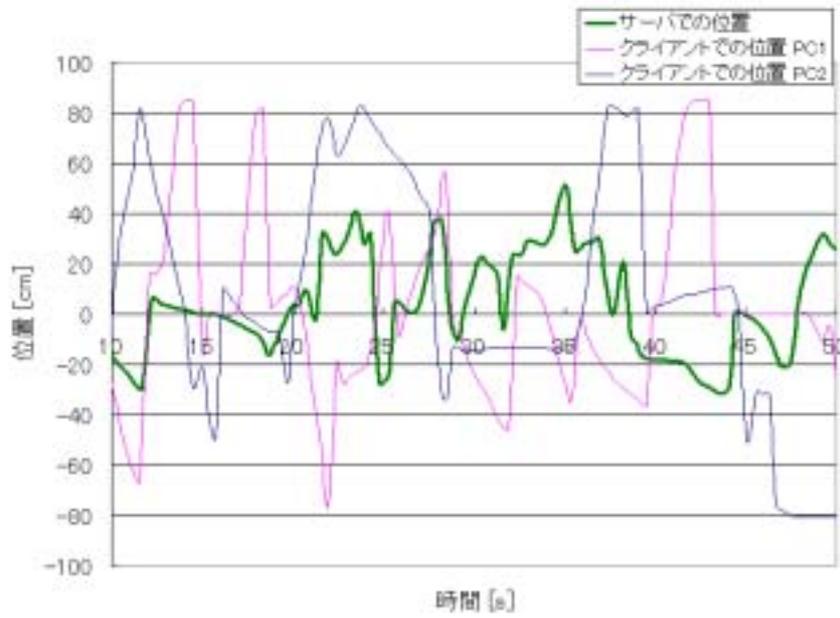


図 4.19: サーバ1台、クライアント2台の場合での仮想物体の位置 (X 軸)、遅延時間 80[ms]

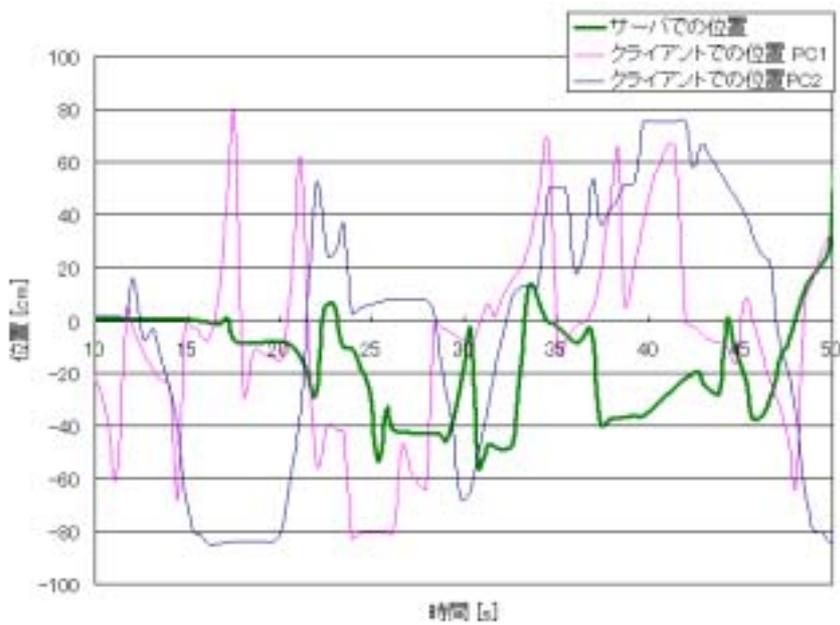


図 4.20: サーバ1台、クライアント2台の場合での仮想物体の位置 (X 軸)、遅延時間 100[ms]

サーバ1台とクライアント2台でのシステムの結果

図4.16は通信遅延が0msの場合のサーバ1台、クライアント2台の場合でのサーバの仮想環境の仮想物体の位置とクライアントの仮想環境の仮想物体の位置を比較したグラフである。グラフから、各クライアントが完全に同期しているとはいえないが、各クライアントの仮想物体の軌跡は互いに近い傾向が見られる。図4.17、図4.18、図4.19、図??は、通信遅延が30、50、80、100msと変化させた場合のサーバの仮想環境の仮想物体の位置とクライアントの仮想環境の仮想物体の位置を比較したグラフである。通信遅延が大きくなると、各クライアントの仮想環境のずれは大きくなるが、通信遅延が50msまでは比較的、各クライアントの仮想物体の軌跡が近い。この結果から、本システムでは通信遅延が50ms程度までは、高い力覚の更新周期を維持しながら、仮想環境の操作が行えることが明らかになった。

4.6 まとめ

本章では新しいシステム制御法を提案した。また、物体操作のための高速な物理計算法を検討し、それらを組み込んだ物体操作システムを構築した。クライアントで反力を計算することによって、高い力覚の更新周期を維持しながら、50ms程度までの通信遅延に耐えるシステムであることを確かめた。

しかしながら、いくつかの問題点が明らかになった。まず現状では仮想現実環境の同期をとるために、仮想物体の位置情報を用いているが、力覚デバイスを生かした、仮想物体を両側から支えて持ち上げるといったような操作が行えない。また、各クライアントの同期をとるために、サーバから送られてきた仮想物体の位置情報を元に、クライアントの仮想現実環境を更新しているが、この頻度を速くしても、クライアントの仮想環境の変化が頻繁になり、仮想物体の操作に乱れが生じることが明らかになった。また、違和感の少ない力覚の更新周期は出るが、1000Hzで動作するわけではなく、さらなる高速化が必要であることがわかった。

第5章 まとめ

5.1 まとめ

近年、注目されているバーチャルリアリティ (VR) とは、コンピュータにより創り出された仮想環境を、現実環境と同じように体感・行動できる技術のことである。VR 技術は、コンピュータ性能の向上やネットワークの高速化により、急速に進歩してきた。中でも遠隔地間で共有仮想現実環境を構築し協調作業を行う研究が盛んに行われており、医療分野や教育分野など多様な分野での応用が期待されている。

近年では、ネットワークを介して接続される共有仮想現実環境で力覚を表現する研究が行われている。これまで VR 技術が主に対象としていたのは、視覚や聴覚であった。それらに力覚を加えることによってこれまでの視覚や聴覚に作用する VR システムより高い現実感が得られることが期待されている。

PHANToM や CyberGrasp といった力触覚デバイスをネットワークを介して接続することにより、互いに離れた場所においても、力触覚を伴う、より現実感のあるインタラクションが可能となる。

力覚を伴う共有仮想現実環境では、それぞれのユーザが違和感なく力触覚を介した仮想環境へ対するインタラクションが行うことが出来る必要がある。しかしながら、力覚情報のネットワークを介した制御には解決すべき問題がある。

力覚デバイスのフォースフィードバック機構の制御更新周期は 1kHz 程度と高速であるため、ネットワークの遅延の影響を受け、安定したインタラクションの実現は困難である。

その理由は、人間が何かを触り続けているという感覚を感じる周波数が約数百 Hz だからである。したがって、フォースフィードバックは数百 Hz から 1kHz の高速な提示が必要となり、フィードバックの計算にはその制御周期に収まる高速な計算処理が要求されることになる。しかし、IP ネットワーク等の通信環境では 100 ミリ秒から数秒の通信遅延が存在するため、力触覚のフィードバック処理をネットワーク上で行った場合、更新は十分に行えず、安定したフォースフィードバックは困難であるため、安定したインタラクションを行うことができない。

本研究では、共有された一つの仮想環境上で力触覚デバイスを用いて物体の操作によるインタラクションを行うシステムを構築することを目的とする。その場合、本研究で力覚の高速な更新周期に対する通信遅延の問題を解決するために、力覚のフィードバック情報をネットワーク上での通信から切り離し、ユーザに近い場所で計算する手法を採用することで、安定した力覚の提示を行う。またその場合に、仮想物体の運動を現実に近い物理シ

システムで高速に計算することによって、違和感の少ない、力覚の伴う協調作業システムを実現する。

本論文では、従来のシステムに比べて、力覚を伴う共有仮想現実環境で、安定した力覚提示が可能な物体操作システムを提案した。提案したシステムでは、ネットワーク遅延に対応するためのシステム制御法を提案し、これによって安定した物体操作環境を提供する。また計算負荷軽減のための物理計算手法を採用した。これらから、共有仮想現実での力覚を伴う物体操作についての構築を行った。

第2章では、本研究に関連する従来研究と、問題点について述べた。

第3章では、従来通り、サーバで力覚を計算した場合のネットワークを介しての利用について調べ、クライアントに必要な力覚の更新周期、サーバ-クライアント間の更新周期についてそれぞれ検証した。

第4章では、ネットワーク遅延に対処するために、仮想環境の制御法を提案した。また仮想環境を更新する高速な物理シミュレーションについて検討を行い、ペナルティ法を用いて仮想現実環境を更新する手法を採用した。これらの提案手法を用いてサーバ、クライアントからなる物体操作システムを構築した。

本システムでは従来のシステムとは異なり、システムの管理法を変えて、評価実験よりユーザの操作に対して安定したフォースフィードバックの生成に成功し、違和感の少ない力覚を伴う協調作業が可能であることが分かった。また各クライアントからの操作が物体操作に反映されることを確かめた。

本研究で構築された共有仮想現実環境での物体操作システムは、力覚情報と、仮想物体情報を切り離して考えることが出来る。そのため視覚情報のみの共有仮想現実環境の技術が応用できると考える。よって比較的、現状の共有仮想現実環境システムに組み合わせやすい。

このシステムを利用することによって、従来の共有仮想現実環境に力覚情報を追加することができ、より臨場感の高い仮想現実環境を構築することができる。

5.2 今後の課題

本システムはユーザの操作に対して安定したフォースフィードバックの生成を行い、違和感の少ない力覚を伴う物体操作システムである。しかし、4.6節でも触れたように、いくつか問題点がある。

第1の問題点として、現状では仮想現実環境の同期をとるために、仮想物体の位置情報を用いているが、力覚デバイスを生かした、仮想物体を両側から支えて持ち上げるといったような操作が行えない。これは、カーソルの位置情報を有効利用して、カーソルの動きによる仮想物体に加えられる力を、サーバ側で計算することで、操作が行えるようになると考えられる。また PHANToM には、手元にボタンがあり、これを利用することも考えられる。

第2の問題点として、各クライアントの同期をとるために、サーバから送られてきた仮

想物体の位置情報を元に、クライアントの仮想現実環境を更新しているが、この頻度を速くしても、クライアントの仮想環境の変化が頻繁になり、仮想物体の操作に乱れが生じることが明らかになった。また各クライアントの同期処理に十分な検討を行っていない。これはクライアント側の仮想環境をサーバのデータを基に修正する場合に、急に変更するために起こる問題である。解決法としては、仮想物体の位置情報をそのまま使用するのではなく、力として、その位置へ徐々に動かすような機構を組み込むことで、改善できると考えられる。現状では各クライアント間の同期は、まだ十分ではなく、各クライアントの操作の間を取ることで、それぞれの操作を反映しているが、仮想物体の移動履歴を利用して、予測するような処理を行うことで、より細かい精度で物体操作が出来る。

第3の問題点として、力覚更新周期のさらなる高速化が必要である。これは、更新周期を高くすると、仮想物体の位置を管理する、スレッド間で共有される変数がロックする頻度が高くなるために生じる問題だと考えられる。プログラムにロックを避ける機構を組み込む必要がある。

これらが今後の研究の課題となる。

謝辞

本研究を行なうにあたり，御指導，御鞭撻をいただいた北陸先端科学技術大学院大学 堀口 進教授に深く感謝致します．

北陸先端科学技術大学院大学 田島 敬史助教授には，サブテーマで熱心に御指導いただき，深く感謝申し上げます．

北陸先端科学技術大学院大学 林 亮子 助手には，貴重なコメントをいただき，深く感謝申し上げます．

北陸先端科学技術大学院大学 福土 将 助手には，有益な御助言をいただき，深く感謝申し上げます．

最後に，日頃よりお世話になったマルチメディア統合システム講座の皆様に厚くお礼申し上げます．

参考文献

- [1] 野間, 宮里, “仮想空間におけるハプティクス”, 映像情報メディア学会技術報告, Vol.21, No.47, pp.1- 6,(1997).
- [2] BASDOGAN, HO, SLATER, SRINIVASAN, “ The Role of Haptic Communication in shared Virtual Environments ”, Proc. of Fourth PHANToM UserGroup '98, (1998).
- [3] ALHALABI, HORIGUCHI, “ Tele-Handshake: A Cooperative Shared Haptic Virtual Environment ”, Eurohaptics Conference, Birmingham, UK, pp.60-64, July 1-4, (2001).
- [4] Buttlo, Oboe, Hannaford, McNeely, “ Force Feedback in Shared Virtual Simulation ”, MI-CAD, (1996).
- [5] Massie, T.and K.Salisbury, “The PHANToM Haptic Interface : A Device for Probing Virtual Objects”, ASME Winter Annual Meeting,DSC-Vol.55-1,ASME,New York,pp.295-300, (1994).
- [6] K.Salisbury, et al., “Haptic Rendering: Programming Touch Interaction with Virtual Object”, Proc.of ACM Symposium on Interactive 3D Graphics, pp.123-130, (1995).
- [7] 佐藤, 平田, 河原田, “空間インタフェース装置 SPIDAR の提案”, 電子情報通信学会論文誌, D-II, Vol.J74-D-II, No.7, pp.887-894 (1991).
- [8] 佐藤, 平田, 河原田, “仮想空間のためのインターフェースデバイス - SPIDAR -”, 信学技報, PRU 89-88, pp.51-58 (1993).
- [9] 高橋, 金井, 位守, “仮想クレイモデリングシステム (VCMS) の開発”, Proc. of 9th Symposium on Human Interface, pp.423-430 (1993).
- [10] 横小路, ホリス, 金出, “仮想環境への視覚/力覚インタフェース:WYSIWYF ディスプレイ”, 日本 VR 学会論文集 , Vol.2 No.4, pp. 17-24 (1997).
- [11] Yokokohji, “A visual/haptic interface to virtual environment(WYSIWYF Display) and its application”, *Proceedings of 1998 IEEE and ATR Workshop on Computer Vision for Virtual Reality Based Human Communications* , pp. 99-104 (1998).

- [12] Yokokohji, Hollis, Kanade, “WYSIWYF Display: A Visual/Haptic Interface to Virtual Environment”, *Presence*, Vol.8 No.4, pp. 412-434 (1999).
- [13] Burdea, G. et al. “A Portable Dextrous Master with Force Feedback”, *PRESENCE*, Vol.1, No.1, pp.15-23, (1992).
- [14] Iwata, Yano, Nakaizumi, Kawamura, “ Project FEELEX: Adding Haptic Surface to Graphics ”, *Proceeding of SIGGRAPH2001* (2001).
- [15] 阿部, 矢野, 岩田, “ 空気圧バルーンを用いた体積型ハプティックディスプレイ ”, *HIS 研究会報告集 Vol.4 No.3* (2002).
- [16] 長谷川, 田崎, 佐藤, “ ハプティックインタラクションのための物理ベースモデリング ”, *日本バーチャルリアリティ学会第8回大会論文集* (2003).
- [17] 引地, 森野, 福田, 松本, 瀬崎, 安田, “ 触覚を含む仮想空間共有システムの提案と評価 ”, *電子情報通信学会論文誌, B, Vol.J86-B, No.2*, pp.268-278 (2003).
- [18] Immersion Corporation: 3D Technologies Cyber Grasp:
<http://www.immersion.com/products/3d/interaction/cybergrasp.shtml>.
- [19] VRPN : Virtual Reality Peripheral Network:
<http://www.cs.unc.edu/Research/vrpn/index.html>.
- [20] Sandeep Singhal, Michael Zyda, “ Networked Virtual Environments Design and Implementation ”, Addison Wesley (2000).
- [21] Emmanuel Frècon, Mårten Stenius, “ DIVE:A scalable network architecture for distributed virtual environments ”, *Distributed Systems Engineering Journal*, Vol.5, No.3, pp.91-100 (1998).
- [22] 宮里, 岸野, “ 臨場感通信会議における仮想空間を介した指さし指示動作知覚の評価 ”, *電子情報通信学会論文誌, D-II, Vol.J80-D-II, No.5*, pp.1221-1230 (1997).
- [23] 片山, 内山, 熊谷, 田村, 苗村, 金子, 原島, “ Collaborative CyberMirage: リアリティと相互ウェアネスを追求した仮想区間共有体験システム ”, *情報処理学会論文誌, Vol.39, No.5*, pp.1484-1493 (1998).
- [24] D.Baraff, “ Analytical methods for dynamic simulation of non-penetrating rigid bodies ”, *Computer Graphics Proceedings(SIGGRAPH 89) Vol. 23*, pp. 223-232, (1989).
- [25] D.Baraff, “ Fast contact force computation for nonpenetrating rigid bodies ”, *Computer Graphics Proceedings(SIGGRAPH 94)*, pp. 23-34, (1994).

- [26] B.Mirtich, J.Canny, “ Impulse-based Simulation of Rigid Bodies ”, Proceedings of 1995 Symposium on Interactive 3D Graphics, (1995).
- [27] M.Moore, J.Wilhelms, “ Collision Detection and Response for Computer Animation ”, Computer Graphics (SIGGRAPH 88), Vol. 22, pp. 289-298, August (1988).
- [28] M.McKenna, D.Zeltzer, “ Dynamic simulation of autonomous legged locomotion ”, Computer Graphics (SIGGRAPH 90), Vol. 24, pp. 29-38, August (1990).

研究業績

藤井大悟, 堀口進, ”共有仮想現実環境での力覚を伴う協調作業による物体操作システム”, 電気関係学会北陸支部連合大会, pp.236-236, (2003 Sep.).