

シミュレーションキャッシングと 遠隔インタラクティブ流体シミュレーションへの応用

橋本 健介¹ 手塚 俊作¹ 森 眞一郎^{1,a)} 富田 眞治²

受付日 2012年1月17日, 採録日 2012年5月23日

概要: 我々は、遠隔地のスーパーコンピュータ上での科学技術計算に対して対話的な操作を可能とする体感型シミュレーションシステムの実現を目指した研究を行っている。このような遠隔インタラクティブシミュレーションを実現するためには、シミュレーションサーバが必要十分な計算スループットを持つとともに、操作端末とサーバとの間の通信遅延を隠蔽するメカニズムが必要不可欠となる。本論文では、この問題に対して、サーバ上で進められているシミュレーションの一部を操作端末側でも同時に計算し、操作端末側での簡易シミュレーションの結果を利用して通信遅延を隠蔽するシミュレーションキャッシングを提案し、当該手法をインタラクティブ流体シミュレータの遠隔操作に応用した結果を報告する。

キーワード: インタラクティブシミュレーション, 遠隔操作, インタラクティブ・スーパーコンピュータリング, 遅延隠蔽, シミュレーションキャッシング

Simulation Caching and Its Application to Remote Interactive Fluid Simulation

KENSUKE HASHIMOTO¹ SHUNSAKU TEZUKA¹ SHIN-ICHIRO MORI^{1,a)}
SHINJI TOMITA²

Received: January 17, 2012, Accepted: May 23, 2012

Abstract: In this paper, we showcase an interactive fluid simulator as an example of a next generation human in the loop scientific simulation environment. It has a capability of steering the simulation by interactively changing the boundary conditions, through the haptic device which is also used as a force display device to visualize the simulation results as force. Furthermore, in order to utilize the power of the simulation server which is located far away from the operator of this interactive fluid simulation, the remote steering framework is discussed. The results of the preliminary implementation is also discussed briefly.

Keywords: interactive simulation, remote steering, interactive supercomputing, latency hiding, simulation caching

1. まえがき

近年の計算機性能の急速な向上にともない、インタラクティブな実時間シミュレーションへの期待が高まっている。フライトシミュレーション [1] や航空管制シミュレーションのようにコンピュータ上のシミュレーション結果を操作者

が直接体感し、その反応として対話的にシミュレーションをステアリング可能なシミュレーションの形態は“human-in-the-loop simulation”あるいは“interactive simulation”と呼ばれる。近年さかんに研究が行われている、災害時の緊急避難シミュレーション等もその一例である。従来、このようなシミュレータ上で行われてきたシミュレーションは主に離散事象シミュレーションであったが、これをスーパーコンピュータ上の科学技術計算のシミュレーションにも拡張する試みも進められている [2], [3], [4], [5], [6]。しかしながら、実時間での対話的なステアリングまでを考慮した

¹ 福井大学
University of Fukui, Fukui 910-8507, Japan

² 京都大学
Kyoto University, Kyoto 606-8501, Japan

a) moris@u-fukui.ac.jp

研究は始まったばかりである [7]. 我々も, 遠隔地のスーパーコンピュータ上での科学技術計算に対して対話的な操作を可能とする体感型シミュレーションシステムの実現を目指した研究を行っている. このような遠隔インタラクティブシミュレーションを実現するためには, シミュレーションサーバが必要十分な計算スループットを持つとともに, 操作端末とサーバとの間の通信遅延を隠蔽するメカニズムが必要不可欠となる.

本論文では, この問題に対して, サーバ上で進められているシミュレーションの一部を操作端末側でも同時に計算し, 操作端末側の簡易シミュレーションの結果を利用することで通信遅延を隠蔽するシミュレーションキャッシングを提案する. さらに, 当該手法をインタラクティブ流体シミュレータの遠隔操作に応用した結果を報告する. 以下, 2章では関連研究を述べ, 3章でシミュレーションキャッシングを提案し, 4章でシミュレーションキャッシングをインタラクティブ流体シミュレーションの遠隔操作に応用した結果を報告する.

2. 関連研究

実行中のシミュレーションにユーザが介入し, シミュレーションのステアリングを行うという試みは, 文献 [8] のバーチャルマイクロスコープにも見られる. バーチャルマイクロスコープでは, シミュレーション結果をその場で可視化するとともに, 操作者からの対話的な要求でシミュレーションの詳細度や解析パラメータの変更を行うためのフレームワークが構築されている. シミュレーション結果を実時間でモニタリングするためのランタイムライブラリとしては NEC の RSVlib [9] や富士通の VisLink [10] がある. また, CAVE 等の仮想空間提示環境を用いてシミュレーション空間を仮想空間にマッピングし, 人がその仮想空間の中に入ってシミュレーション結果を体感する設備は数多く導入されているが, シミュレーション対象と人との直接的な双方向インタラクションが実現できているとはいえない. また, いずれも遠隔地からの実時間インタラクションに関しては十分な検討が行われていない. VisLink では, 計算の中断, 再開, 再実行, 軽微なパラメータの変更程度の簡単なシミュレーション・ステアリングは可能であり, この機能を使って京都大学のスーパーコンピュータ上のシミュレーションを岩手大学で遠隔操作する試みを上沢らが報告している [11]. しかしながら, 実時間性ならびにアプリケーションプログラムの中まで踏み込んだ複雑なシミュレーション・ステアリングという点ではまだ実現できていない [7].

本論文で取り上げるインタラクティブな流体計算に関しては, ベクトル処理向きの流体計算を高速化するアクセラレータとして GPU を採用する研究が近年数多く行われている. 従来から行われている反復法ベースの数値解析プロ

グラムをそのまま GPU に実装するアプローチ [12], [13] と近年注目を集めている粒子法ベースのアプローチに大別される. 後者は, 我々が参考にした格子ボルツマン法を用いた手法 [14] や, SPH 法を用いた手法 [15], [16], [17] 等が提案されており, 単精度浮動小数点演算という制約下でも実用に耐える応用問題の存在が示されている.

物理シミュレーション結果の 3 次元力覚提示に関する研究では, バーチャルリアリティ応用としてのゲーム感覚での攪拌シミュレーション [19] や, 仮想容器の動きに対応して容器内の流体の動きを SPH 法によりシミュレーションし仮想容器に加わる慣性力を提示するシステムの研究 [16], [17], [18] 等が行われている.

シミュレーションをともなう遠隔操作フレームワークに関する研究としては, 鈴木らが行ったロボット手術のための遠隔手術シミュレーションシステムがあり, 日本とタイとの間で遠隔手術シミュレーション実験を行っている [20]. この実験では, ネットワーク遅延が往復約 140 ms の環境下で, 腹腔鏡下での胆嚢摘出手術シミュレーションを実行させている. しかしながら, この実験では力覚デバイスは 3 次元座標の入力だけに用いられており, 力覚をともなったシミュレーション結果提示は行われていない. 一方, 山口らは遠隔地間の実時間触覚通信技術の研究として遠隔地点間で手術シミュレーションを行うことのできる触覚通信システムの研究 [21] を行っており, 肝臓モデルの実時間変形シミュレーションを実現している. ここでは性能の等しいシミュレータを 2 地点に設置し, 両方で同じシミュレーションを行うことで実時間性を確保している. これに対し, 我々が提案する遠隔操作フレームワークは遠隔地のサーバと手元のサーバの性能が非対称のものを想定しており, これにより手元のサーバのみでは実現不可能な大規模あるいは高精細なシミュレーションを可能とする.

遅延隠蔽を目的として操作者側でシミュレーションを行うという観点では, ロボットの遠隔操作を支援する「予測ディスプレイ (Predictive Display)」があり, 最終目的こそ異なるものの問題解決の方法論はほぼ等価であるといえる. 具体的には, 操作されるロボットのシミュレータを操作者側に用意し, シミュレータ上のロボットをあたかも現実のロボットと見なして操作命令を生成することで遠隔操作における遅延を隠蔽する手法であり, すでに実用化が進んでいる. ここで使われるシミュレータを予測ディスプレイ (Predictive Display) と呼ぶ [22], [23], [24]. 予測ディスプレイを用いたシステムは, 宇宙空間のロボットを地上から遠隔操作する場合 [23], [25], [26] のように遅延時間そのものが非常に大きな環境で, かつ, 移動物体の動作速度が比較的低い状況での利用が多く, 主に物体どうしの衝突防止やソフトタッチを目的として操作者側に注意を促した

めの視覚情報の提示*1が第一義的な目的である。

ネットワーク上でのオンライン対戦ゲームのように、多数の参加者がサーバ上で行われているシミュレーション(仮想環境 [Virtual Environment (VE)]) に対してインタラクションを行う際の通信量削減手法としては Dead Reckoning Model (DRM) [27], [28] がある。DRM では、通信路の回線容量や遅延時間に応じた一定の許容値をあらかじめ設定し、遠隔地のユーザの操作が定められた許容範囲を超えるまでサーバに情報を伝達しないことで通信量を削減する。一方で、サーバ上の情報が更新されない状況下でも、各参加者は他の参加者の動きを何らかの手法で予測し本来の仮想環境の状態を推定する。これにともない、各参加者が想定する仮想環境の状態には不一致が発生するが、これは全参加者が許容した不一致として容認する。しかしながら、定められた許容値を超えた時点で何らかの一貫性制御を行う。一時的な不一致を認めつつ、適切なタイミングで一貫性を維持するという点は本論文の提案も同様であるが、DRM の手法を VE における物体間の衝突判定等とは計算の質・量ともに本質的に異なる科学技術計算における物理場のシミュレーションへ応用することは非現実的である。

遠隔操作時の力覚に関する遅延隠蔽法に間して我々のアプローチに非常に近い研究としては文献 [29] の研究がある。ここでは、力覚情報の提示に手元で行う簡易シミュレーションの結果を利用することで力覚に関する遅延隠蔽を目指している。視覚と力覚の知覚特性の違いに注目した点は我々と同様であるが、操作者側の簡易シミュレーション結果の解の精度保証が行われない点、通信遅延時間のゆらぎ(以下、ジッタと呼ぶ)対策に間して言及がなされていない点が我々の提案との大きな違いである。

3. シミュレーションキャッシング

3.1 遠隔操作フレームワーク

本研究で想定する遠隔操作フレームワークは、1) 遠隔地に設置された大規模シミュレーションサーバ、2) オペレータの操作や結果の呈示を行うユーザインタフェース(操作端末)、それに加えて、3) シミュレーション空間を縮小した簡易シミュレーションを行う PC クラスタベースのクライアント(ローカルサーバ)で構成されるサーバクライアント型のシミュレーションシステムとする(図1参照)。このとき、大規模シミュレーションサーバは、インタラクティブなシミュレーションの実行に十分な計算スループットを持っているものとする。また、操作端末とローカルサーバ間の通信遅延は十分に小さいものとする。

このとき、最も基本的な遠隔インタラクティブシミュ

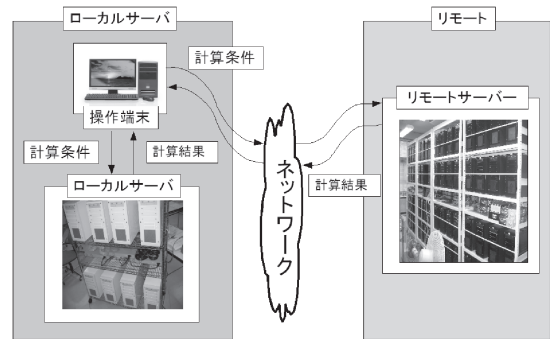


図1 遠隔シミュレーションシステム

Fig. 1 Remote simulation system.

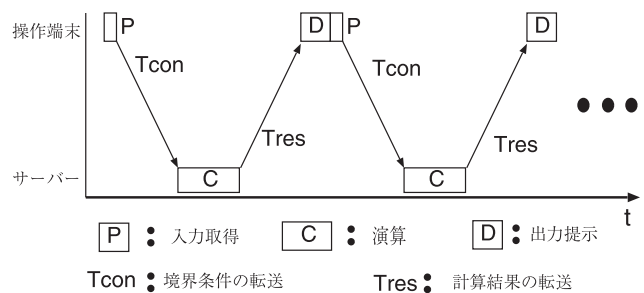


図2 遠隔シミュレーションの処理フロー

Fig. 2 Remote simulation flow.

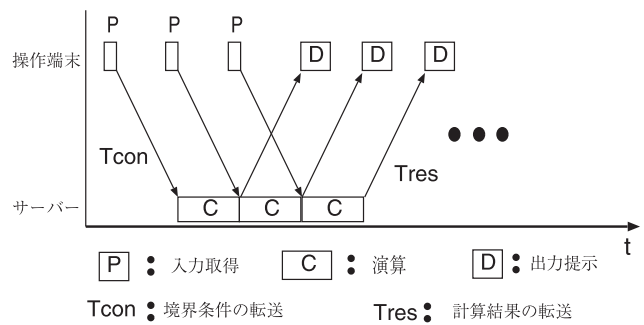


図3 パイプライン化した遠隔シミュレーションの処理フロー

Fig. 3 Pipelined remote simulation flow.

レーションの処理フローは図2に示すとおり、P:操作端末での入力操作、Tcon:シミュレーションに必要な境界条件の伝送、C:サーバでの数値演算、Tres:シミュレーション結果の伝送、D:出力提示の5つのフェーズで構成される。

これらの5つのフェーズをナイーブに逐次実行していたのでは、計算サーバのスループットは十分に活かしきれない。そこで、一連のステアリング入力に対してパイプライン処理を行うことでスループットを向上させる。図3にパイプライン化を行った場合の処理フローを示す。なお、Tcon ならびに Tres が変動せず、それら自体が複数ステージのパイプラインと見なせる場合、全体のパイプラインピッチは、P, C, D のうち最大のもので決まる。本研究では、P, C, および D に関してはインタラクションに支障がない程度に短いものとする [30], [31], [32]。

*1 遠隔地からの実画像に付加情報を加えるだけのものから、遠隔地の環境を VR 空間上にモデル化し完全な仮想環境での視覚情報を提示するものまで様々な提示方法がある。

3.2 シミュレーションキャッシング

パイプライン処理の適用によりシミュレーションのスループット向上は達成できたが、依然としてシミュレーション結果を提示するまでの遅延時間は残ったままである。ネットワークを介した遠隔操作の場合、五感にかかわる情報のリアルタイムな提供は非常に重要であり、通信の時間遅れをとまなう（リアルタイム性に欠ける）情報提供は、操作者を混乱させるものとなる。さらに、通信遅延の問題をさらに複雑にする要因として、通信遅延自体の動的な揺らぎ（以下、ジッタと呼ぶ）の問題がある。ジッタが発生すると出力提示の間隔が変動してしまい操作性が低減する。そこで、このような通信遅延を隠蔽する手法としてシミュレーションキャッシングを提案する。

シミュレーションキャッシングとは、リモートサーバと連携・協調してシミュレーションを行う簡易シミュレーションサーバ（ローカルサーバ）を操作者の近くに設置することで、操作者に対して、リモートサーバのみでは得られない効果（付加価値）を得る技術である。

期待される具体的な効果としては、遅延時間の隠蔽や局所的な注目領域に対するリモートサーバよりも高分解能な計算結果等が考えられる。計算機のキャッシュ・メモリとのアナログで、遅延時間の隠蔽という観点でシミュレーションキャッシングを議論すると以下のようになる。キャッシュメモリは主記憶上のデータのコピーを作って隠し持ち、必要に応じてそのデータを使うことで主記憶アクセスの遅延を隠蔽する。この際、キャッシュ・メモリ自体がデータを加工することはない。我々はこれをパッシブ（受動的）なキャッシングと考える。これに対して、本論文で提案するシミュレーションキャッシングはリモートサーバからデータをローカルサーバにコピーし、さらにリモートサーバで行われているデータの加工と同様な加工をローカルサーバにコピーしたデータに加えつつデータを隠し持つアクティブ（能動的）なキャッシングと考える。付加価値として高分解能な計算結果を得るという観点で考えると、本来リモートサーバで行うべき計算の一部をローカルサーバが代理で計算していると考えられることもできる。

シミュレーションキャッシングの方法は、環境やシミュレーションの種類によって様々なものが考えられる。以下では、1) 簡易シミュレーションの方法、2) 遅延隠蔽の方法、3) シミュレーションの一貫性制御手法、という3つの観点でシミュレーションキャッシングの典型的なモデルを分類する。

3.2.1 簡易シミュレーションの方法

ローカルサーバでリモートサーバ上で実行するシミュレーションの一部を並行して冗長計算するが、「シミュレーションの一部」という概念としては、たとえば次のようなものが考えられる。

- a) 粗いシミュレーション格子で構成される全体空間：二

次元のシミュレーションであれば、解像度を縦横 $1/N$ とすると、計算格子の格子点数が $1/N^2$ に減少する。さらに、アプリケーションによっては時間方向の計算量が $1/N$ となる場合もある。

- b) シミュレーション対象空間の一部の部分領域を抽出：二次元のシミュレーションであれば、縦横 $1/N$ の領域を簡易シミュレーションする場合計算量は $1/N^2$ に減少する。
- c) シミュレーション対象空間の一部の部分領域を高解像度抽出：注目領域のみに限定してリモートサーバよりもさらに高精度で計算を行う。
- d) 異なる数値計算モデルの利用：計算精度とのトレードオフで計算量の少ない計算モデル/アルゴリズムを利用。
- e) ハードウェア性能の違いを利用：たとえばリモートで 64 bit、ローカルで 32 bit の計算精度でシミュレーションを行えば、メモリ量は $1/2$ 倍に減少し、計算量は $1 \sim 1/2$ 倍に減少する。

3.2.2 遅延隠蔽手法

通信遅延の隠蔽手法としては、たとえば次のようなものが考えられる。

- a) つねにローカル側だけの情報を提示する。
- b) ローカル側の情報とリモート側の情報を融合して提示する。
- 性質の異なる複数の情報を提示する場合に、情報の性質に応じてローカルとリモートを使い分ける。
 - 原則リモートの情報を用い、ジッタ等で許容値を超えた場合にローカルの情報を用いる。
- c) ジッタ等で許容値を超えた場合、1 ステップ前の状態を提示し続ける。

次章で言及する流体シミュレーションへの応用では、視覚情報と力覚情報でリモート側とローカル側のデータを使い分けるとともに、ジッタ対策においても両者を併用している。このように複数の手法を組み合わせることも可能である。

3.2.3 シミュレーションの一貫性制御

シミュレーションが過去の状態に依存せず（履歴を持たない）、現在の入力から一意に計算が可能な場合等、リモートサーバの結果をローカルサーバに反映させる必要がまったくない場合が存在する。このようなとき、ローカルとリモートの一貫性制御は必要ない。このような、ローカルとリモートが一貫性という形での同期をとらずに非同期にシミュレーションを行うモデルを、特にシミュレーションブロキシと呼ぶ。

しかし、シミュレーションの対象となるシステムが時間項を含む形でモデル化されている場合、シミュレーション結果は過去の状態に依存する（履歴を持つ）。この場合、同一入力、同一初期条件であっても高解像度シミュレーションの結果と、低解像度シミュレーションの結果の誤差がシ

シミュレーションの時間経過とともに蓄積されていく。また、シミュレーションモデル自体は時間項を持たないが、境界条件としてリモート側の結果が必要な場合も存在する。このような場合は、リモートとローカルのシミュレーションを同期させ、一貫性制御を行う必要がある。

この、一貫性制御の実装に関しても種々の実装モデルが考えられる。まず、一貫性制御の頻度に関しては、シミュレーションの1ステップごとに一貫性制御を行うモデルが最も厳密なモデルであるが、遅延隠蔽という観点では十分な効果が期待できない。そのため、 N ステップごとに一貫性制御を行うモデルが標準的なモデルとなる。 N の値は、シミュレーションに要求される精度、実時間性の要求レベル、ローカル・リモート間の平均遅延時間等から適切な値を設定する必要がある。

N ステップごとに一貫性制御を行う場合、ローカル側とリモート側で N ステップのシミュレーションに要する時間が異なる（通常はリモート側が遅いと考えられる）。そのため、 N ステップ進めた後の一貫性制御の実装に関してもいくつかの実装方法が考えられる。

中断モデル： N ステップごとに入力を受付をいったん中断し、一貫性制御が終わるまでシミュレーションを停止するモデル。

ロールバックモデル： N ステップ経過後も入力を受付を中断せず、シミュレーションを継続する。

ロールバックモデルを用いる場合、ローカル側が $N + \alpha$ ステップまで進んだ後にリモートから一貫性制御のデータが到着する。この時点で初めて、 N ステップ時の時刻にロールバックしてシミュレーションを開始する。そのままの実装では、時間をさかのぼって結果を再提示することになるため違和感を感じることが想定される。そこで、現在提示しているタイムステップまでローカルサーバのシミュレーションを進め、追いついた時点で反映させる方法等を用いる必要がある。

4. インタラクティブ流体シミュレーションへの応用

シミュレーションによってシミュレーションキャッシングの実装方法は変わってくるが、本研究では流体シミュレーションを実時間シミュレーションの一例として取り上げ、これにシミュレーションキャッシングを実装する。まず、インタラクティブ流体シミュレーションについて述べ、次に具体的なシミュレーションキャッシングの実装方法について述べる。

4.1 インタラクティブ流体シミュレーションと遠隔操作フレームワークの実装

インタラクティブシミュレーションの実現のためには、入力に対する瞬時の応答が求められるため計算の高速化が

必須である。また、シミュレーションへのインタラクティブな入出力も実現しなければならない。そこで我々は、格子ボルツマン法 [14] を用いた流体計算処理と可視化処理を GPU で、抵抗力計算と力覚提示デバイスの制御を CPU でそれぞれ行うことによって、物体（円柱）にかかる抵抗力を力覚提示デバイスに提示するとともに、実時間で流体中の物体の位置を変動させることのできるインタラクティブ流体シミュレータを実装した [34], [35]。流体計算に格子ボルツマン法を適用した理由は、流体内を物体が移動する移動境界問題への適用の容易さと、並列性の抽出が容易であるためである [33], [37]。なおこの際、サーバクライアント型システムのフレームワークは MPI を用いて実装する。

4.1.1 処理の流れ

シミュレーション結果に基づき応力を力覚提示デバイスに提示するとともに、実時間で境界条件（流体中の障害物の位置）を変動させる流体攪拌シミュレータを開発した [34], [35]。ここでは、オペレータによるシミュレーションへのインタラクティブな入出力を実現するために、SensAble Technologies 社の PHANTOM Omni [36] を使用した。オペレータはこのデバイスのペン状のスタイラス部を移動させることによって 3 次元位置情報を入力できる。また、デバイスはスタイラス部を通じてオペレータに力覚を返すことができる。

まず、操作端末が触覚デバイスから攪拌棒の座標を取得し、ネットワークを介してリモートサーバに送信する。リモートサーバは受信した新座標を GPU に転送する。GPU は攪拌棒の新座標からシミュレーションを行い、シミュレーションの実行結果を CPU に送り、CPU で応力の計算を行い、視覚・力覚データとして操作端末に送信する。操作端末は受信したデータをもとに、モニタより視覚を、触覚デバイスより力覚をオペレータに返す（図 4）。

予備実験としてこの手法を遠隔操作せず表 1 の環境でローカルに実装した結果、1 タイムステップあたりの実行時間が 1.65 msec となり、2 次元での流体シミュレーション

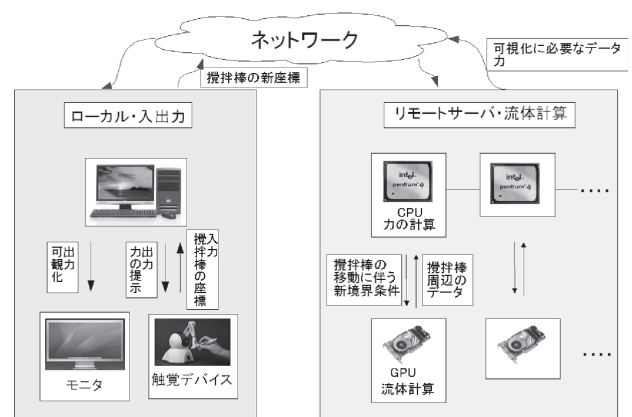


図 4 インタラクティブ流体シミュレータの処理フロー
Fig. 4 Interactive fluid simulation flow.

表 1 予備実験環境

Table 1 Environment for preliminary experiment.

CPU	Pentium4 3.6 GHz
OS	WindowsXP
GPU	GeForce8800Ultra

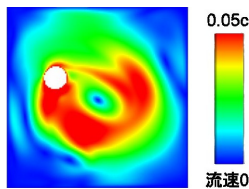
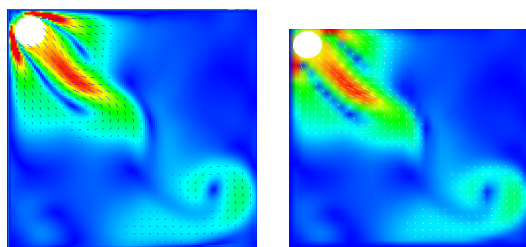


図 5 シミュレーションの様子

Fig. 5 Snapshot of fluid stirring simulation.



(a) 間引き前のデータ (b) 補間後のデータ

図 6 通信量削減のための空間間引の影響

Fig. 6 Effect of spatially curtailed image.

をインタラクティブに実行することができた。シミュレーションの様子を図 5 に示す。なお、本論文で使用したデータはレイノルズ数を 100 と設定した場合のデータである。

4.1.2 通信データ量の低減

インタラクティブな遠隔操作環境の構築にあたっては、利用可能な通信路の通信帯域を考慮したデータ量の削減が必要不可欠である。予備実験の際の計算結果の可視化(図 5)では、シミュレーション空間と同じ解像度の格子点上での流速を色で表現していたが、今回の実装ではシミュレーション空間を縦横それぞれ $1/n$ に間引いた格子点上で流速ベクトルを線で表示する可視化手法を採用することでデータ量の低減を図った。

このとき、流速ベクトルの線による可視化のみではシミュレーションの結果をとらえにくい。間引きされた流速ベクトルから線型補間した流速をカラーコンタとして併用することでデータ通信量を増加させることなく表示結果を改善した(図 6)。ここでは、シミュレーション空間を 128×128 として実装しているため、 $n = 4$ の場合可視化に必要なデータは 32×32 の格子点上の流速ベクトルでよいことになり、通信するデータ量は $1/16$ に低減されている。表示結果の比較のため、全シミュレーション空間の色と線を可視化した表示結果を図 6 に示す。静止画での比較では違いを感じられるが、実時間でシミュレーションを継続している最中にはその差は感じる事ができない程度であった [38]。

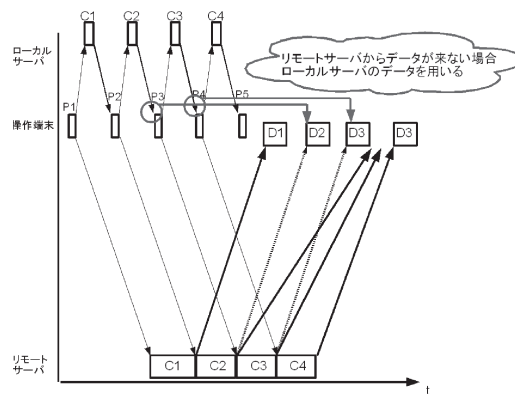


図 7 ジッタ対策

Fig. 7 Jitter compensation.

また、仮にリモートサーバ上のシミュレーション解像度に対応した高精細な静止画を観測する必要がある場合には、いったんシミュレーションを停止(ポーズ)し、その時点での間引きする前の画像を提示する。あるいは、サーバ側で画像を保存しておき、後で観測する等の方法が考えられる。

4.2 シミュレーションキャッシングの応用

まず、操作端末における一連のステアリング入力に対してパイプライン処理を行うことでスループットを向上させる。パイプライン化によって、流体シミュレーションにおける出力(流体の可視化による視覚提示と触覚デバイスへの力覚提示)が、現在の入力より数ステップ過去のものになる。しかしながら、攪拌流体シミュレーションにおける操作の連続性と、人間の視覚特性を考慮すると動画像であれば数 100 msec 程度の入出力遅延は許容できると考えられる。そのためパイプライン化による遅延の影響は少ない。一方で、人間の力覚に対する知覚特性は入出力の遅延に敏感であるため、力覚データをパイプライン化することは難しい。

よって、力覚情報に対してはローカルサーバで簡易シミュレーションを行った結果を提示し、視覚情報に関しては遠隔地の計算サーバの計算結果を提示することで、視覚、力覚ともに違和感のない情報提示を行う。簡易シミュレーションの方法としては、3.2.1 項の a) を行う。このとき、視覚提示における結果の遅延提示とシミュレーション解像度の低減による力覚提示への影響を調査する必要がある。

次に、ジッタを隠蔽する手法を図 7 に示す。ジッタが発生し、リモートサーバから可視化に必要なデータを一定間隔で受け取ることができなくなった場合、受け取るはずであったデータに対応するローカルサーバのデータを用いて可視化することで、視覚提示をジッタに影響されることなく一定に行う。このとき、対応するローカルサーバの可視化データを使用できるよう、ローカルサーバのシミュレーション結果をリングバッファに格納しておく。また、この

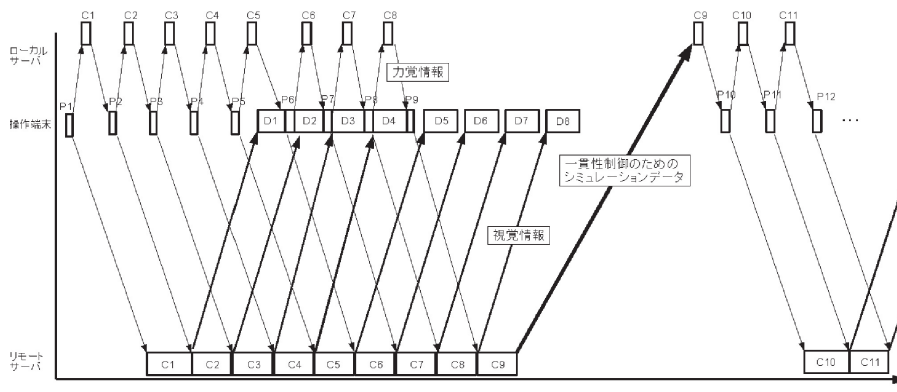


図 8 シミュレーションキャッシング (Jitter が許容値以下の場合)

Fig. 8 Simulation caching.

表 2 実装環境

Table 2 Implementation environment.

	操作端末	ローカルサーバ	リモートサーバ
CPU	Pentium4 2.53 GHz	Core2 2.66 GHz	PentiumD 3.6 GHz
GPU	GeForce 7800GS	GeForce 8800GTX	GeForce 8800GTX
OS	Linux		
MPI	MPICH2-1.0.5		

手法を用いて視覚提示した場合、すでに提示したリモートサーバの可視化データは用いず、提示した可視化データの次のデータを用いる。

このようにローカルサーバのシミュレーション結果を用いているが、流体シミュレーションはシミュレーションが過去の履歴に依存するため、解像度を落としたシミュレーションが進むごとに、リモートサーバとの誤差が蓄積されてしまう。そこで、リモートサーバ上のデータの一部を適宜ローカルサーバにコピーしそれを新たな初期値としてローカルサーバ上のシミュレーションを再開することで誤差の蓄積をキャンセルする。反映のタイミングや、反映中のシミュレーションの挙動は様々なものが考えられるが、ここでは比較の実装しやすい方法として、一定間隔で反映し、反映する間はシミュレーションを一時中断する（反映する直前に提示した視覚・触覚データ提示し続け、反映し終わるまで入力を受け付けない）ものとする。図 8 は 9 ステップごとに一貫性制御を行う中断モデルの例であり、入力 P9 の後、一貫性制御が終わるまで P10 を受け付けない。このため、反映によりシミュレーション精度の向上が望めるが、シミュレーションを中断することに対する違和感が生じる。よって、適度な反映頻度を調査する必要がある。

4.3 実装と評価

上述のシミュレーションキャッシングを用いたインタラクティブ流体シミュレーションの遠隔操作システムを実装する。実装環境を表 2 に示す。今回の実装では、環境構築の都合上、操作端末・ローカルサーバ・リモートサーバが物理的にはすべて福井大学の研究室にある。ただし、

リモートサーバと通信する場合、京都大学に設置した中継用のリモート PC を経由して通信を行うことで仮想的なリモートサーバを実現している。よって、操作端末とリモートサーバ間の 1 回の双方向通信では福井大学と京都大学の 2 往復分の通信遅延が生じることになる。このとき、操作端末とリモートサーバとの間の往復通信遅延は実測で約 60 ms であった。また、今回の環境ではリモートサーバとローカルサーバがほぼ同程度の性能を持っているが、実際のシミュレーションにおいて、リモートサーバではローカルサーバの約 8 倍の計算を実行した。リモートサーバからの通信データ量は、速度場を可視化するためのデータは 2 次元速度ベクトル (1 格子点あたり 8 B) で 8 KB、一貫性制御の際は 1 格子点あたり 32 B で 128 KB を送信した。

4.3.1 視覚特性と力覚特性の検証

まず、パイプライン化によって起こる結果の遅延提示の影響を調査するため、シミュレーション結果をいったん記憶し、一定時間後に画像と力覚の提示を行うソフトウェアによる簡易遅延シミュレータを作成し中継用のリモートサーバをバイパスして実験を行った。被験者にまず遅延がない状態でシミュレータを操作してもらい、次に遅延時間をランダムに設定して力覚・視覚提示における違和感を感じるか否かを実験した。6 名に対して実験を行い、各被験者が確実に違和感を訴える遅延時間を計測したところ、80 ms で違和感を感じた被験者が 1 名、100 ms で違和感を感じる被験者が 4 名、120 ms で違和感を感じた被験者が 5 名であった。なお、この時点で違和感を感じなかった被験者 1 名は過去にネットワーク経由での遠隔操作を経験した者で、500 ms 程度まで違和感を訴えなかった。これらの結

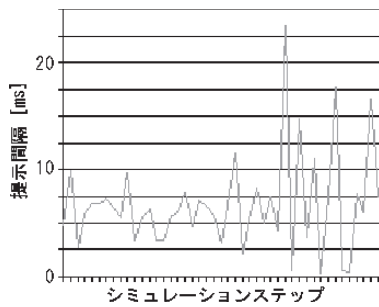


図 9 視覚情報の提示間隔の一例 (シミュレーションキャッシングなし)

Fig. 9 Effect of jitter on simulation (w/o simulation caching).

果は、鈴木らが行った遠隔手術シミュレーション実験 [20] で生じた遅延時間と同程度のものとなった。同様に、視覚提示とともに反力も提示した場合の遅延耐性を測定したところ、2 ms から 30 ms の間で異和感を感じた被検者が 4 名、100 ms 以上の遅延に対して異和感を感じなかった被検者が 2 名であった。また、反力データとしてシミュレーションの空間解像度が 4 倍異なるシミュレーション結果を提示した際に、違いを認識できた被検者はひとりもいなかった。このことから、力覚においては大きな個人差はあるものの、数 msec から数十 msec 程度の遅延で違和感を感じるという現象が確認された。さらに、力覚においては時間に対しては敏感であるが、一方で提示する力の大きさ自体の精度にはさほど敏感でないことも確認できた。これは、圧力刺激に対する心理的知覚が圧力刺激量の対数に比例するというウエーバー・フェヒナーの法則とも一致する。

4.3.2 パイプライン化の効果

まず、遠隔操作による通信時間がシミュレーションに与える影響を調査するため、ローカルサーバやパイプラインを用いずに、入力受付を出力の後に行う逐次実行処理 (図 2 参照) を行ったところ、シミュレーション結果の提示間隔の平均は約 60 msec となり、そのうち 98% を通信時間が占めていることが分かった。この通信遅延によってシミュレーション速度が大幅に低下し実時間処理が不可能になることを確認した。

次に、図 3 に示すパイプライン化した遠隔シミュレーション環境を構築した。その結果、比較的ジッタが少い時間帯では、シミュレーション結果の提示間隔の平均は演算時間 4 msec とほぼ等しくなり、通信時間が隠蔽されることが確認できた。ところが、図 9 のように、シミュレータを操作する時間帯によってはシミュレーション結果の提示間隔に著しい揺らぎが発生し、操作中に異和感を感じることが確認できた。

4.3.3 シミュレーションキャッシングの効果

最後に、リモートサーバとローカルサーバを協調させ、シミュレーションキャッシングの環境を構築しジッタの影響を含む通信遅延の隠蔽効果を測定した。ジッタの許容時

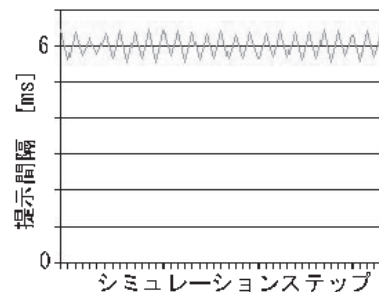


図 10 視覚情報の提示間隔の一例 (シミュレーションキャッシングあり)

Fig. 10 Effect of jitter on simulation (w/ simulation caching).

間を 6 ms に設定し実験を行った結果、図 10 のように提示間隔の変動を設定値の 10% 以下に抑制できた。

次に、リモートサーバとローカルサーバの一貫性を保証する頻度がローカルサーバでの計算結果にどのような影響を与えるかを評価する実験を行った。ただし、この実験では流体の攪拌動作を一定とするため、力覚提示デバイスからの位置情報にかかわらず、攪拌棒がシミュレーション空間の中心部分で等速円運動を行うように強制的に位置情報を与えた。シミュレーションを 10,000 ステップ進めた時点でのリモートサーバが提示した画像を図 11 (a) に示す。図 11 (b) および (c) は同じ時刻にローカルサーバが提示した画像である。(b) は 750 ステップごとにリモートサーバとローカルサーバで一貫性制御を行った場合であり、(c) は実行中に 1 度も一貫性制御を行わなかった場合である。一貫性制御を行わなかった (c) では、(a) とはまったく異なる画像になっていることが分かる。一方、一貫性制御を行った (b) では、(a) と同一ではないものの、ほぼ等価な結果が得られていることが分かる。

図 12 は、一貫性制御を行う頻度 (更新頻度) がシミュレーション結果に与える影響を時間を追って観測した結果である。各シミュレーションステップにおいて、リモートサーバが提示する画像を基準画像 (S) と考え、この基準画像とローカルサーバが提示する画像の差をノイズ画像 (N) として SN 比を計算したものである*2。更新を行った直後のシミュレーション・ステップでは SN 比が高く、時間が進むにつれて SN 比が低下していくことが分かる。また、更新頻度が 750 ステップの場合に比べ 100 ステップの場合が SN 比が高いことも分かった。しかしながら、今回の実験環境では 1 ステップが約 6 ms に相当するため、更新頻度を 100 ステップとした場合は約 600 ms ごとにシミュレーションが一時中断する。この中断の影響が速度の低下として感じられてしまう。これに対して更新頻度を 750 ステップとした場合は中断の間隔が 4.5 秒ごととなり中断の影響

*2 今回は速度を可視化しているため、格子点 (i, j) でのリモートサーバが提示する速度を $s_{(i,j)}$ 、速度誤差を $n_{(i,j)}$ としたときの SN 比 [dB] は $SN \text{ 比} = 10 \log_{10} \{ (\sum_{i,j} s_{(i,j)}^2) / (\sum_{i,j} n_{(i,j)}^2) \}$ として計算している。

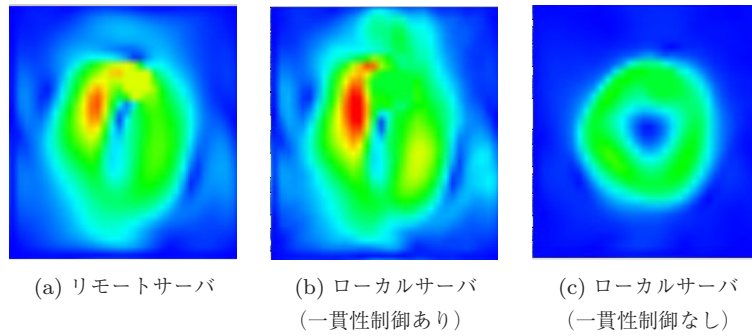


図 11 一貫性制御の効果

Fig. 11 Effect of coherence control.

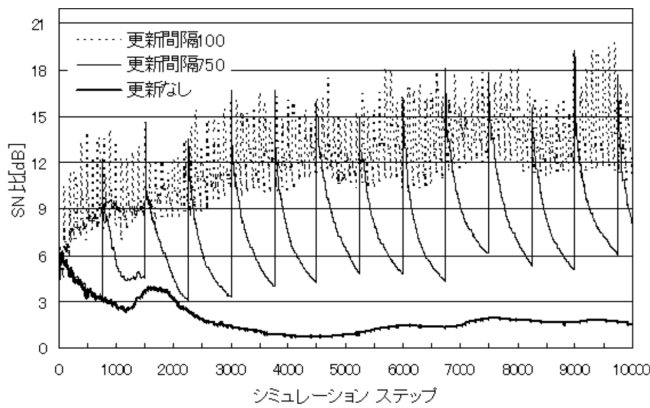


図 12 更新頻度が提示画像に与える影響

Fig. 12 Effect of update frequency.

はほぼ無視できるようになった。実際のシステムにおいては、利用環境に応じてユーザーが更新頻度を制御可能にする必要があると考える。

5. シミュレーションキャッシングの意義ならびに応用可能性に関する考察

現在のインターネットにおけるベストエフォート型通信と同様、本論文で提案するシミュレーションキャッシングはベストエフォート型のシミュレーションと考えることもできる。今回のアプリケーションでは実時間性と計算精度のトレードオフが必要な状況においては、時間が許さざりにおいて高精度の計算結果を提示するという形でのベストエフォート型シミュレーションである。たとえば反復型の数値計算アルゴリズムにおいて、計算終了条件として残差等を用いた収束条件と反復回数の上限を設定した場合に相当する。このとき、後者の終了条件（タイムリミットに相当）で計算が終了した場合に得られた解を「正しい」と見なすか「間違い」と見なすかは、その計算結果を利用する環境に依存する。また、十分な精度が得られていないと判断した場合には、反復回数の上限值を上げるという選択により時間と精度のトレードオフが可能である。今回実装した流体シミュレーションにおいても、ジッタの許容値を上げることで高精度な結果が視覚提示される確率を上げる

ことが可能である。なお、力覚提示に対する応答時間制約の充足に関しては、リモートサーバ単体では得られなかったシミュレーションキャッシングの付加価値である。

次に、他の応用可能性について検討を行う。3.2.1 項の簡易シミュレーションの方法で言及した注目領域に限定した高解像度計算については、拡散シミュレーションに関する予備実装を行っており [39]、今後評価検討を進めていく予定であるが、その応用例として、たとえば県の防災センターで行う県レベルのシミュレーションと、県内の複数の漁協レベルで独自に行う複数の特定地域での詳細シミュレーションとの連携等を考えており、県レベルではすべてをカバーすることのできない詳細シミュレーションを相互に連携しながら分散して進めていくようなシナリオが考えられる。さらに、その発展形として災害時等にリモートサーバへの接続が切れた状態においても、ローカルサーバの簡易シミュレーションを利用して精度は下がるが最新の状況を提供するという利用形態も考えることが可能である。このような状況においては、非専門家に対する情報提示の難しさの議論は残るものの、福島原発事故における SPEEDI [40] のシミュレーション結果の開示に関するシミュレーション結果の信頼性と実時間情報提示に関する議論が参考になるのではないかと考える。

6. まとめ

本論文では、リモートサーバ上でのシミュレーションをインタラクティブにステアリングする際の遅延隠蔽手法としてシミュレーションキャッシングを提案した。この手法を応用して遅延に敏感な力覚提示をともなう遠隔インタラクティブ流体シミュレーション環境のプロトタイプシステムを構築し、提案手法の技術的な有効性を検証した。

今後の課題としては、より多くの問題に適用可能な遠隔操作フレームワークの構築、シミュレーションの目的や操作者の嗜好等も考慮したユーザインタフェースの改良、クラウドコンピューティング環境への応用を視野に入れたフレームワークの仮想化等が考えられる。

謝辞 本研究の一部は、日本学術振興会科学研究費補助

金基盤研究 S (課題番号 16100001), 基盤研究 C (課題番号 22500044) ならびに, (財) 栢森情報科学振興財団研究助成による。

参考文献

- [1] McCarty, W.D., Sheasby, S., Amburn, P., Stytz, M.R. and Switzer, C.: A virtual cockpit for a distributed interactive simulation, *IEEE Computer Graphics and Applications*, Vol.14, Issue 1, pp.49-54 (1994).
- [2] Marshall, R., Kempf, J. and Dyer, S.: Visualization Methods and Simulation Steering for a 3D Turbulence Model of Lake Erie, *Proc. Symp. on Interactive 3D Graphics*, pp.89-97 (1990).
- [3] Johnson, C., Parker, S.G., Hansen, C., Kindlmann, G.L. and Livnat, Y.: Interactive Simulation and Visualization, Vol.32, No.12, pp.59-65, *IEEE Computer* (1999).
- [4] 小柳義夫: 10 年後の HPC, 筑波大学計算物理学研究センター 10 周年シンポジウム「計算科学の展望」講演スライド (2002).
- [5] Zhu, Q. and Agrawal, G.: Supporting Fault-Tolerance for Time-Critical Events in Distributed Environments, *Proc. Supercomputing*, Paper Nr.32, pp.1-12 (2009).
- [6] Malakar, P., Natarajan, V. and Vddhiyar, S.S.: An Adaptive Framework for Simulation and Online Remote Visualization of Critical Climate Applications in Resource-constrained Environment, *Proc. Supercomputing*, Paper Nr.295, pp.1-10 (2010).
- [7] Woodward, P.R., Porter, D.H., Greensky, J., Larson, A.J., Knox, M., Hanson, J., Ravindran, N. and Fuchs, T.: Interactive Volume Visualization of Fluid Flow Simulation Data, *PARA'06 Workshop on the State-of-the-Art in Scientific and Parallel Computing* (2006).
- [8] 関口智嗣, 高田俊和, 松原 聖, 逢坂俊郎, 平原幸男: バーチャルマイクロスコプのワークステーションクラスタによる高速実装, 情報処理学会研究報告 HPC-071, pp.29-34 (1998).
- [9] Doi, S. et al.: RVSLIB: A Library for Concurrent Network Visualization of Large Scale Unsteady Simulation, *SPEEDUP Journal*, 11, pp.59-65 (1997).
- [10] 富士通 (株): リアルタイム可視化用ソフトウェア VisLink, 入手先 (<http://jp.fujitsu.com/solutions/hpc/app/vislink/>).
- [11] 上沢和真, 酒井晃二, 小山田耕二: リアルタイム可視化ツール Vislink によるステアリングプログラム, 京都大学大型計算機センター広報, Vol.34, No.6, pp.327-338 (2001).
- [12] 藤井敬伸: グラフィクスプロセッサを用いた数値計算の高速化と計算誤差の影響に関する検討, 福井大学工学部卒業論文 (2007).
- [13] 成瀬 彰, 住元真司, 久門耕一: GPGPU 上での流体アプリケーションの高速化手法—1GPU で姫野ベンチマーク 60GFLOPS 超, 情報処理学会研究報告 HPC-117, pp.49-54 (2008).
- [14] Fan, Z., Qiu, F. and Kaufman, A.: GPU Cluster for High Performance Computing, *Proc. ACM/IEEE Conf. on Supercomputing* (2004).
- [15] 原田隆宏, 塚越誠一, 河口洋一郎: GPU を用いた Smoothed Particle Hydrodynamics の高速化, *Visual Computing グラフィクスと CAD 合同シンポジウム'07*, pp.175-180 (2007).
- [16] Monaghan, J.J.: Simulating free surface flows with SPH, *Journal of Computational Physics*, Vol.110, pp.399-406 (1994).
- [17] 川井昌之, 廣田功貴, 畔柳将吾: 3 次元流体を含む仮想環境構築のための SPH の圧力フィードバック補正, 日本バーチャルリアリティ学会論文誌, Vol.13, No.4, pp.469-475 (2008).
- [18] 松田憲彦, 川井昌之, 野田英希, 松下 洸: 3 次元流体を含む力覚提示装置, 電気関係学会北陸支部大会講演論文集, F-104 (2006).
- [19] 瀬崎勇一, 大内 農, 櫻井快勢, 瀬井大志: 谷本隼飛, 溝口敦士, 宮田一乗: Witch's Cauldron: 力覚インタフェースを利用した混ぜる VR アプリケーション, *インタラクティブ 2007 論文集*, pp.73-74 (2007).
- [20] 鈴木薫之, 鈴木直樹, 服部麻木, 小西晃造, 家入里志, 橋爪誠: 手術計画・訓練・教育のための遠隔手術シミュレーション, *MEDICAL IMAGING TECHNOLOGY*, Vol.25, No.1, pp.25-30 (2007).
- [21] 山口 哲, 田中孝英, Viet, H.Q.H., 高間康文, 辻野圭則, 山下裕礼, 小澤美有紀, 田中弘美: ポリウムベース臨場感通信のための遠隔地間実時間触覚共有システムの開発, 立体映像技術研究会, NiCT (2007).
- [22] 神徳徹雄: 環境モデルを用いた遠隔マニピュレーションシステム, 工業技術院機械技術研究書 NEWS, 1991-No.11 (1991).
- [23] Mathan, S., Hyndman, A., Fischer, K., Blatz, J. and Brams, D.: Efficacy of a Predictive Display, Steering Device and Vehicle Body Representation in the Operation of a Lunar Vehicle, *Proc. CHI '96*, pp.71-72 (1996).
- [24] Burkert, T., Leupold, J. and Passig, G.: A Photorealistic Predictive Display, *Presence*, Vol.13, No.1, pp.22-43 (2004).
- [25] Hirzinger, G., Brunner, B., Dietrich, J. and Heindl, J.: ROTEX - The First Remotely Controlled Robot in Space, *Proc. Int'l Conf. on ROBOTICS and AUTOMATION*, pp.2604-2611 (1994).
- [26] Lapointe, J.F.: COSMOS: A VR-Based Proof-of-Concept Interface for Advanced Space Robot Control, *Proc. Int'l Conf. on Augmented tele-existence*, pp.215-219 (2005).
- [27] Capin, T.K., Esmerado, J. and Thalmann, D.: A Dead-Reckoning Technique for Streaming Virtual Human Animation, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.9, No.3, pp.411-415 (1999).
- [28] Delaney, D., Ward, T. and McLoone, S.: On Consistency and Network Latency in Distributed Interactive Applications: A Survey -PartII, *Presense*, Vol.15, No.4, pp.465-482 (2006).
- [29] アダ イمام カヤディ, 山本佳男: 力覚フィードバック型微細作業システムを用いた遠隔操作実験, 東海大学紀要工学部, Vol.47, No.2, pp.93-100 (2007).
- [30] 依藤 逸, 野田裕介, 糸 真人, 嶋田 創, 中尾 恵, 森真一郎, 中島 浩, 富田真治: 操作の連続性を考慮した投機計算を利用するインタラクティブシミュレータ, 先進的計算基盤システムシンポジウム, pp.229-238 (2009).
- [31] 岩永翔太郎, 福岡慎治, 森真一郎: 実時間シミュレーションへの応用を前提とした SMW 公式を用いた逆行列計算のマルチコア並列処理, 電子情報通信学会論文誌, Vol.J94-D, No.7, pp.1165-1168 (2011).
- [32] 坂井陽平, 浅野琢也, 福岡慎治, 森真一郎: 高精細タイルディスプレイを用いた並列ポリウムレンダリングシステムの実装, 先進的計算基盤システムシンポジウム, pp.2-10 (2011).
- [33] Bouzidi, M., Firdaouss, M. and Lallemand, P.: Momentum transfer of a Boltzmann-lattice fluid with boundaries, *Phys. Fluids*, 13, pp.3452-3459 (2001).
- [34] 小松原誠: インタラクティブ流体シミュレーション, 京都大学大学院情報科学研究科修士論文 (2006).

- [35] 橋本健介, 嶋田 創, 三輪 忍, 幡生安紀, 森真一郎, 富田眞治: インタラクティブシミュレーションにおける遠隔操作フレームワークの実装, 情報処理学会研究報告 2007-HPC-111, Vol.2007, No.80, pp.103-108 (2007).
- [36] SensAble Technologies, available from <http://www.sensable.com/> (accessed 2010-01-07).
- [37] 山口明德: GPU を用いたインタラクティブ流体・構造連成シミュレータの構築, 京都大学大学院情報学研究所修士論文 (2009).
- [38] 関連動画ファイル, 入手先 (<http://www.youtube.com/user/HPCfuis>) (参照 2012-04-03).
- [39] 西村裕介: シミュレーションキャッシングフレームワークの実装, 福井大学大学院工学研究科修士論文 (2012).
- [40] 文部科学省原子力安全課原子力防災ネットワーク (環境防災 N ネット), 入手先 (<http://www.bousai.ne.jp/vis/torikumi/index0301.html>) (参照 2012-04-03).



富田 眞治 (フェロー)

1945 年生. 1968 年京都大学工学部電子工学科卒業. 1973 年京都大学大学院博士課程修了. 工学博士. 同年京都大学工学部情報工学教室助手. 1978 年同助教授. 1986 年九州大学大学院総合理工学研究科教授, 1991 年京都大学工学部教授, 1998 年同大学大学院情報学研究科教授, 2009 年京都大学名誉教授. 京都大学物質-細胞統合システム拠点特定拠点教授/事務部門長. 現在に至る. 計算機アーキテクチャ, 並列処理システム等に興味を持つ. 著書「並列コンピュータ工学」, 「コンピュータアーキテクチャ (第 2 版)」等. IEEE, ACM 各会員. 電子情報通信学会, 情報処理学会各フェロー.



橋本 健介

1984 年生. 2007 年福井大学工学部情報・メディア工学科卒業. 2010 年福井大学大学院工学研究科情報・メディア工学専攻修士課程修了. 現在, PFU ソフトウェア株式会社勤務.



手塚 俊作

1985 年生. 2009 年福井大学工学部情報・メディア工学科卒業. 2011 年福井大学大学院工学研究科情報・メディア工学専攻修士課程修了. 現在, 株式会社富士通ミッションクリティカルシステムズ勤務.



森 眞一郎 (正会員)

1963 年生. 1987 年熊本大学工学部電子工学科卒業. 1986 年九州大学大学院総合理工学研究科情報システム学専攻修士課程修了. 1992 年同専攻博士後期課程単位取得退学. 同年京都大学工学部情報工学科助手. 1995 年同助教授. 1998 年同大学大学院情報学研究科助教授. 2006 年福井大学大学院工学研究科情報・メディア工学専攻教授. 博士 (工学). 並列処理, 可視化, 計算機アーキテクチャの研究に従事. 電子情報通信学会, IEEE-CS, ACM 各会員.