2007－HPC－111 (18)
2007／8／1

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

橋本健介[†] 嶋田創[††] 三輪忍[††]
幡生安紀[†] 森眞一郎[†] 富田眞治[††]

あらまし 近年の計算機性能の急速な向上に伴い，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．また，通信速度の飛躍的な高速化に伴い，インタラクティブシミュレーションの遠隔操作の研究も行われている．本報告では，このようなインタラクティブシミュレーションの一例として，力覚提示を伴う流体シミュレーション環境を構築し，さらに当該シミュレータを遠隔操作するフレームワークのプロトタイプ実装を行った結果を報告する．流体運動の計算手法としては格子ボルツマン法を用い，触覚デバイスを用いた高速計算のためにGPUによる計算を行った．シミュレーション結果はリアルタイムに流速を画像表示するとともに，流体内の障害物に加わる応力を触覚デバイスを用いて力覚提示した．遠隔操作するフレームワークを実装し，研究室内の異なる２台の計算機を用いた環境において遠隔操作を実験で行った．

Mounting of remote-controlled framework in interactive simulation

KENSUKE HASHIMOTO,[†] TUKURU SHIMADA,[††]
SHINOBU MIWA ,[††] YASUNORI HATABU ,[†]
SHIN-ICHIRO MORI [†] and SHINJI TOMITA[††]

abstract
The expectation for an interactive real-time simulation has risen as the computer performance in recent years improves rapidly. Moreover, the remote control of an interactive simulation is researched speeding up the telecommunication speed rapid. Then, the interactive fluid simulation and the remote control are described in this text. It simulated by using lattice Boltzmann method and GPU and sense of touch device. The fluid movement was calculated by the lattice Boltzmann method. It calculated by GPU for a high-speed calculation. With a sense of touch device as the sense of force presentation. And the framework that was the remote control of this was mounted. The remote control of an interactive simulation was achieved in the laboratory.

1. はじめに

近年の計算機性能の急速な向上に伴い，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．ここでは，オペレータによるシミュレーション対象へのインタラクティブな操作に対応して実時間でシミュレーションするとともに，即刻その結果を視覚その他の手段により提示することが求められる．我々は，インタラクティブシミュレーションの一例として流体シミュレーションを扱っている．

流体シミュレーションは，工学，医学等の分野で重要視されており，研究が進められているが，計算量が多くオペレータの操作に伴う動的な環境条件の変化にリアルタイムに対応できる数値計算モデルを使用する必要がある．

また，遠隔手術シミュレーションのようにインタラクティブシミュレーションを遠隔操作する研究も行われている．インタラクティブシミュレーションを遠隔操作する場合，オペレーターがシミュレーション結果から入力を決定し，その入力を元に次のシミュレーションを行うため，通信遅延をいかに隠蔽するかが重要となる．

そこで本研究では，インタラクティブシミュレーションを実装し，これを遠隔操作することを目的とする．遠隔操作として，シミュレーションを行うPCクラスタベースの計算サーバと，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とシミュレーション

† 福井大学
University of Fukui
†† 京都大学
Kyouto University

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を用いたサーバークライアント型のシミュレーションシステムを構想している(図1).

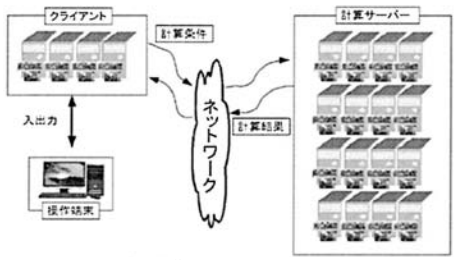

図 1　遠隔操作シミュレーションシステム

本稿では，流体の運動を計算するモデルとして格子ボルツマン法，高速計算のためにGPUによる計算，力覚提示として触覚デバイスを用いたインタラクティブ流体シミュレーションについて述べ，これを遠隔操作するフレームワークを実装した報告を行う．

2. 関連研究

GPUを用いた流体計算の高速化については，我々が参考にした格子ボルツマン法を用いた手法[1]や，SPH法を用いた手法[2]などが提案されており，単制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[3]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[4]等が行われている．

医療の高度化，多様化が進む中，新たな教育方法の確立が求められている．このような中，現実の手術を模擬した仮想環境を用いて手術に必要な知識や技術の習得を効率的かつ安全に行う目的で手術シミュレータの実現に向けた研究が進められている．手術シミュレータの全計算処理時間において生体の変形計算が占める割合は高く，大規模な生体モデルデータを実時間で変形処理するには高い演算性能が求められる．これをGPUによって高速に処理する研究が進められている．菊川らは，生体変形シミュレーションモデルとして粘性や慣性を付加した有限要素モデルを適用した手術シミュレーションを，GPUを装着したPCクラスタへ実装することで，約13000ノードの有限要素モデルにおいて手術シミュレーションの実時間性を確認した[5]．また中尾らは，実時間切開シミュレーションにおいて，切開前後で頂点数を変化させないことで，計算コストの増加を抑制しつつ，合成マトリクスの高速な部分更新を可能にする新しい有限要素モデリング方法を提案し，この手法が従来のものより高速化されることを確認した[6]．

さらに，手術シミュレーションを遠隔操作可能にすることで，手術シミュレーションが世界各地で実行可能になり，遠隔地にいる専門医の指導を受けることや，遠隔地同士の医師による手術手技の検討や議論が可能になる．鈴木らは，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を開発し，日タイ間遠隔手術シミュレーション実験を行った[7]．この実験では，ネットワーク遅延が往復約140msecかかるなかで，腹鏡下での胆嚢摘出手術シミュレーションを実行させた．

3. インタラクティブ流体シミュレーション

3.1 流体運動の数値シミュレーション

3.1.1 格子ボルツマン法

流体の運動を解析する方法としてよく知られている，ナビエストークス方程式を離散化して解く手法を用いた場合，境界形状に応じた適切なメッシュを作る必要があり，オペレーターの操作に対するメッシュの自動生成は難しく，実行時間が低下する．よって，流体運動を解析する手法として格子ボルツマン法を用いる．

格子ボルツマン法では，流れ場を規則的な格子で切り，仮想的な粒子を格子に沿って運動させることで流れをモデル化するため，オペレーターの操作に対してメッシュを動的に変更する必要がない．格子上での移動(並進)と衝突の特性を決定するボルツマン方程式の衝突演算子の適切な設定により，巨視的な流れ場に対する微視的粒子運動を規定する．この時，粒子を個別に捉えるのではなく，仮想的な粒子の集まりと捉える．

3.1.2 空間の離散化と分布関数

本研究では空間の離散化方法として，2次元空間において静止状態を含めて9つの移動方向を考える2D9Vモデルを用い，このモデルにおける仮想粒子の速度ベクトル $e_i(i = 0 \sim 8)$ を図2に示すように定義する．速度ベクトル e_i の方向に移動する仮想粒子の密度分布は，分布関数 $f_i(i = 0 \sim 8)$ で表す．

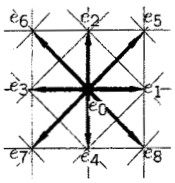

図 2　仮想粒子の速度ベクトル e_i

各格子点において，以下にて説明する衝突，並進，巨視化過程の繰り返しにより流体シミュレーションが進行する．

3.1.3 衝　　突

本研究では，衝突モデルとして格子ボルツマン法を用いた流体シミュレーションに関する多くの文献で用いられる格子 BGK モデルを採用する．このモデルでは簡素化された衝突モデルを採用し，衝突を次式でモデル化する．

$$f_i' = f_i - \frac{1}{\phi}\left(f_i - f_i^{eq}\right) \quad (i = 0, 1, \cdots, 8) \qquad (1)$$

ここで，f_i は衝突前の分布関数，f_i' は衝突後の分布関数である．ϕ は単一時間緩和係数と呼ばれる定数で，動粘性係数から求められる．f_i^{eq} は局所平衡分布関数と呼ばれる．格子点における密度を ρ，流速ベクトルを V とすれば，f_i^{eq} は次式で与えられる．

$$f_i^{eq} = \omega_i \rho \left\{1 + 3e_i \cdot V + \frac{9}{2}(e_i \cdot V)^2 - \frac{3}{2}V^2\right\} \qquad (2)$$

ここで $\omega_0 = \frac{4}{9}$，$\omega_{1\sim4} = \frac{1}{9}$，$\omega_{5\sim8} = \frac{1}{36}$ である．

3.1.4 並進および境界条件

並進では，衝突によって更新された分布関数 f_i に比例する数の粒子が e_i 方向の隣接格子点へ移動すると考える．ただし，移動方向に隣接する格子点が固体壁に相当する場合，Bouzidi らが提案した手法[8] を用いて分布関数を決定する．

3.1.5 巨　視　化

境界条件を考慮した並進の過程を経た分布関数を用いて，各格子点での巨視的変数を求める．密度 ρ，流速ベクトル V は，以下のように求められる．

$$\rho = \sum_{i=0}^{8} f_i \qquad (3)$$

$$\rho V = \sum_{i=0}^{8} f_i e_i \qquad (4)$$

衝突，並進，巨視化の一連の過程を 1 タイムステップとし，これを繰り返す．

3.2 実　　装

3.2.1 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のプロトタイプシステムを開発した[9]．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[10] 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，CPU が触覚デバイスから攪拌棒の座標を取得し，これを 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モニタに可視化する．また，攪拌棒周辺の局所データを CPU に送り，CPU で応力の計算を行い，触覚デバイスよりオペレーターに力覚を返す（図 3）．

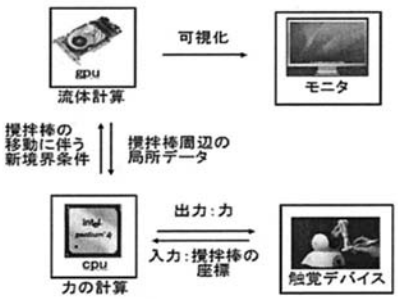

図 3　処理の流れ

3.2.2 GPU と CPU の連携

GPU では，流体シミュレーションと描画を行う．GPU は頂点プロセッサとフラグメントプロセッサを持ち，両プロセッサの処理を動的に制御するための高級言語である Cg[11] を用いることで，各プロセッサによる数値計算が可能になる．本研究では，フラグメントプロセッサを制御するためのフラグメントプログラムで主要な計算処理を行った．計算で用いるデータの初期値を CPU で作成し，32bit 浮動小数点テクスチャとして GPU に転送し，テクスチャの各ピクセルと各格子点を対応させ，テクスチャマッピングの機能を利用して計算する．

CPU では，触覚デバイスの制御と応力の計算を行う．応力は，物体周辺の圧力を積分し，Batchelor らが水の表現に用いた状態方程式を用いて，密度が平均密度以下の部分の圧力は小さく，平均密度以上の部分は圧力を大きくする補正により求める．この力の計算に必要な値を GPU から読み出し，触覚デバイスで力覚を提示する[12]．

3.3 可視化方法

シミュレーションの表示方法として，流速を色によって可視化することにより，全体の流れを直感的に理解することができた (図 4)．しかし，これだけでは具体的な流体の流れの把握が困難である．そこで，流速ベクトルを表示し，流体の具体的な流れがわかるようにした．線を表示する場合，方向と大きさを表現するための空間が必要となるため，空間を縦に 1/n，横に 1/n 間引きする．n=4 とし，流速と流速ベクトルの可視化を併用した結果を図 5 に示す．この流速ベクトルの線による表示は，3 次元シミュレーションの表示方法として利用でき，さらに可視化に必要なデータ量

が少ないため遠隔操作に利用可能である．

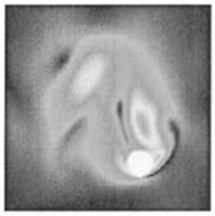

図 4　流速の可視化

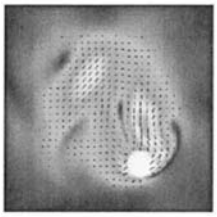

図 5　流速と流速ベクトルの可視化

4. 遠隔操作フレームワーク

インタラクティブシミュレーションによる仮想現実感を得る目的を実現するためには，3次元空間を扱う必要がある．ここで，大規模なインタラクティブシミュレーションを効果的に実行するには1台の PC では限界があり，PC クラスタにより並列処理が必要となる．しかし，PC クラスタは可搬性に乏しく，遠隔教育においては，シミュレーションの遠隔操作が求められる．本章では，PC クラスタの遠隔操作によるシミュレーションを目指し，MPI[13] を用いた遠隔操作フレームワークの実装について報告する．

4.1 遠隔操作フレームワーク

本研究では，手元にある計算機では実行不可能な大規模インタラクティブ流体シミュレーションを，遠隔に設置された高性能な計算機 (例えば PC クラスタ) を遠隔操作して実行することを目的とする．具体的には福井大学の操作端末および PC クラスタから京都大学の PC クラスタ[14] を遠隔操作することを目的としている．ローカルの操作端末でシミュレーションをステアリングし，リモートのサーバーでステアリングされたシミュレーションの新しい境界条件に基づいて数値計算を行い，そのシミュレーション結果をローカルのオペレータにフィードバックする．これにより，手元にある計算機だけでは不可能な大規模シミュレーションを実現する (図 6)．一連の動作を前章の流体シミュレーションの例に適用した場合の処理の流れを図 7 に示す．ここで，P:位置座標取得，C:数値演算，D:描画，Tcom:攪拌棒の新座標情報の伝送時間，Tres:可視化・力覚提示情報の伝送時間である．

4.2 研究室内での遠隔操作実験

本来，福井大学から京都大学にあるクラスタを遠隔操作することでシミュレーションを行うことを目的としているが，ここでは予備実験として，研究室内の計算機を遠隔操作することでシミュレーションを行う．リモート計算機にクラスタを用いずに，1台の計算機を用い，可視化に流速を色で表す方法を用いて実行した．実装環境を表 1 に示す．

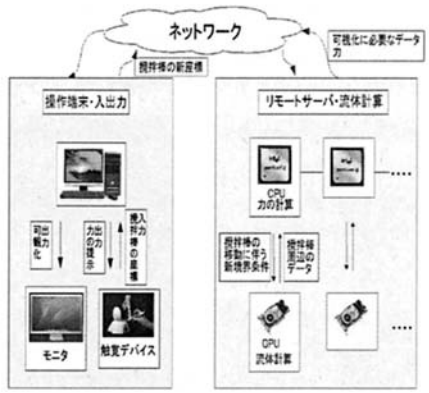

図 6　遠隔操作フレームワーク

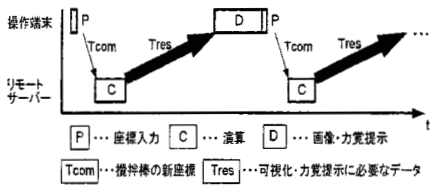

図 7　処理の流れ

表 1　実行環境

	操作端末	リモートサーバー
CPU	Pentium4 2.53GHz	PentiumD 3.6GHz
OS	Linux(FC3)	Linux(FC6)
GPU	GeForce 7800GS (AGP)	GeForce 8800GTX (PCI-e)
MPI	mpich2-1.0.5	mpich2-1.0.5

操作端末は，力覚デバイスのドライバの都合により FedoraCore3 が動作する PC を使用した．この際，グラフィックバスが AGP であったため，操作端末とリモートサーバーでは，描画性能や CPU-GPU 間の転送時間に性能差がある．

まず，予備実験としてリモートサーバーのみでシミュレーションを行った結果，実行時間 (シミュレーション結果が提示されてから，次のシミュレーション結果が提示されるまでの時間．つまり P+Tcom+C+Tres+D) は 1msec であった．

次に，操作端末とリモートサーバーを，1 つのハブを解して 100Mbps イーサネットで接続し，シミュレーションした実行結果を表 4.2 に示す．可視化に必要なデータの通信時間 (Tcom+Tres) に 11msec かかり，環境の違いから画像・力覚提示が D' から D に 0.87msec 増加したため，リモート操作での実行時間 (P+Tcom+C+Tres+D) が，ローカル操作での実行

時間 (P+C+D') から約 12msec 増加し，実時間性が低減した．

表 2 遠隔操作の実行時間

ローカル操作での実行時間 (P+C+D')	1msec
画像・力覚提示：D'(リモート操作)	0.03msec
画像・力覚提示：D(ローカル操作)	0.9msec
リモート操作の通信時間 (Tcom+Tres)	11msec
リモート操作での実行時間 (P+Tcom+C+Tres+D)	13msec

4.3 実時間性の改善

4.3.1 可視化データの低減

表示に流速の色による可視化方法を用いているため，可視化に必要なデータは，全シミュレーション空間の流速ベクトルになる．ここでは，シミュレーション空間を 128 × 128 として実装しており，浮動小数点演算を行っているため，通信しなければならない可視化に必要な流速ベクトルは約 130KB になり，実効転送速度約 10MB/s の専用回線を用いた研究室内での通信でも 11msec と長い通信時間を必要とする．実効転送速度数百 KB/s の専用回線ではない一般のインターネットを用いた場合，通信時間の実行時間への影響はさらに大きなものとなるため，通信するデータを削減しなければならない．

そこで，シミュレーション結果の表示方法を，流速を色によって可視化する方法から，流速ベクトルを線によって可視化する方法に変更する．流速ベクトルの可視化は，全シミュレーション空間ではなく，シミュレーション空間を 1/n に間引きした空間を扱えばよいため，通信データ量を削減できる．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なので約 8KB でよいことになり，通信時間が表 3 のように減少し，実行時間が短縮されたためインタラクティブにシミュレーションすることができた．

表 3 可視化方法変更による実効時間の変化

画像・力覚提示：D'	0.43msec
リモート操作の通信時間 (Tcom+Tres)	1msec
リモート操作での実行時間 (P+Tcom+D+Tres+C)	1.98msec

しかし，流速ベクトルの線による可視化のみでは図 8 のようにシミュレーションの結果をとらえにくい．そこで，間引きされた流速ベクトルから流速を補間し，色を用いた表示を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することを考えた．(図 9)．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 10 に示す．但し，実装の都合により線形補間は行っていないため，予稿ではわかりにくいが，わずかながら結果は異なっている．しかし，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ことができない程度と考えられる．また，今回は可視化に必要なデータの間引きを一様に行ったが，シミュレーション結果の空間内の重要度にあわせて間引きの仕方を動的に変更する手法も検討している．

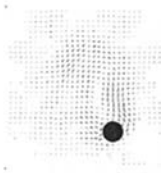

図 8 流速ベクトルによる可視化

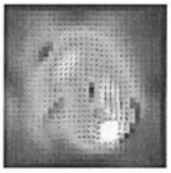

図 9 保間された流速データの併用

図 10 元データ

通信するデータを削減するための手段として可視化方法を変更し，通信するデータそのものを削減したが，例えば描画を 33msec 間隔に行うようなシミュレーション結果の提示を間引く方法も考えられる．今までのシミュレーションではシミュレーションが進む度に結果を提示していたが，結果の提示を間引いても違和感がないようなシミュレーションであれば，結果の提示を間引くことによって通信回数を削減し，通信するデータを削減することができる．本稿では，結果の提示を間引くことからこの手法を Curtailed Display(CD) 法とよぶことにする．

4.3.2 パイプライン化

研究室内での通信は遅延時間がほとんどかからないが，一般的なネットワークを用い，例えば京都-福井間で通信すると，15〜20msec 程度の遅延時間が生じインタラクティブにシミュレーションできないため，より効率的なシミュレーションモデルを検討する必要がある．

今までの処理の流れでは，出力が提示された後，新たな入力を送り，次のシミュレーションを進めていたため入力と出力は同じ時間軸のものであるが，シミュレーションによっては厳密に入力と出力が同じ時間軸のものでなくてもよい場合が考えられる．そのようなシミュレーションだと仮定すると，出力が提示される前に入力を送ることでパイプライン化できる (図 11)．これは，投機実行と考えることもできる．本稿ではこの手法を，シミュレーション結果が遅れて提示されることから，Delayed Display(DD) 法とよぶことにする．

この手法を実装し，表示方法に流速ベクトルによる可視化を用いて実験したところ，実行時間を約半分 (0.98msec) に高速化できた．これは C と P+D の処理時間が同程度なためである．

この実験では，約 1msec 前のシミュレーション結果が提示されていることになるが，入力と出力のズレに

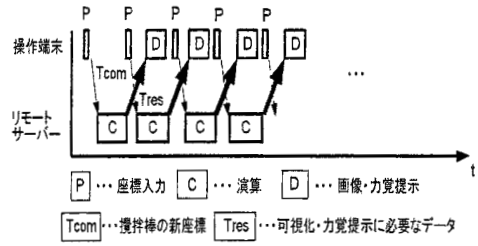

図 11 Delayed Display 法による
パイプライン化

よる違和感はなかった．しかし，遅延時間が長くなるとズレが大きくなるため，その影響を調査しなければならない．そこで，シミュレーション結果を一旦記憶し，一定時間後に画像と力覚の提示を行うことで，擬似的に遅延によるズレを作り出す実験を行った．その結果，100msec 程度のズレまで違和感を感じることなくシミュレーションすることができた．これは，鈴木らが行った遠隔手術シミュレーション実験[7] で生じた遅延時間と同程度のものとなった．

さらに，Delayed Display 法と Curtailed Display 法の併用が許容できるシミュレーションであれば，図 12 のような処理も可能である．我々の流体シミュレーションの例では，操作端末から攪拌棒の新座標を送り続け，リモートサーバーで随時シミュレーションしてシミュレーションの状態を更新していき，33msec 間隔で描画されるようシミュレーションの結果を操作端末に返すシミュレーションを検討している．

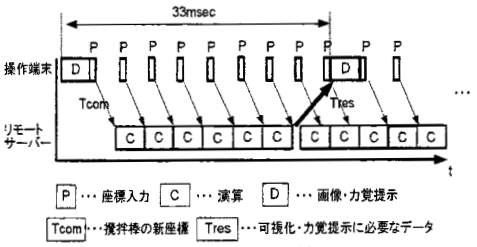

図 12 高速化の構想

5. ま と め

流体の運動を計算するモデルとして格子ボルツマン法，高速計算のために GPU による計算，力覚提示として触覚デバイスを用いたインタラクティブ流体シミュレーションを実装し，これを研究室内で遠隔操作することができた．さらにパイプライン化により実行速度を向上させ，遠隔操作によってインタラクティブシミュレーションを実行する可能性を示した．今後，実環境での実験を行うとともに動的な遅延変動によるシミュレーションへの影響等を調査する必要がある．

また，サーバークライアント間の通信遅延の影響を隠蔽する目的で，操作端末側でもローカルサーバーを構築し，簡易シミュレーションを行うことで (これをシミュレーションキャシングと呼ぶ) インタラクティブ性を向上する手法を検討している．

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), 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) Zhe Fan, Feng Qiu, Arie Kaufman, GPU Cluster for High Performance Computing Proc. ACM/IEEE Conf. on Supercomputing, 2004.
2) 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)
3) 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ション 2007 論文集, pp.73–74(2007)
4) 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.
5) 菊川 孝明, 長坂 学, 緒方 正人, 三菱プレシジョン (株) 開発部, RD Division,Mitsubisi Precision Co., LTd. : 実時間物理シミュレーション- 手術シミュレータにおける生体の実時間変形計算法-, Visual Computing グラフィクスと CAD 合同シンポジウム '07, pp129-134 (2007)
6) 中尾 恵, 河本 敏孝, 奏 小太郎: 柔らかい物体に対する切開の実時間シミュレーション方法, Visual Computing グラフィクスと CAD 合同シンポジウム '07, pp125-128 (2007)
7) 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 Vol25 No.1, pp.25-30(2007)
8) M. Bouzidi, M. Firdaouss and P. Lallemand, Momentum transfer of aBoltzmann-lattice fluid with boundaries, Phys. Fluids 13, pp.3452-3459 (2001)
9) 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学研究科, 修士論文 (2006)
10) SensAble Technologies, http://www.sensable.com/
11) NVIDIA Corporation, Cg Toolkit User's Manual Release 1.2 (2004)
12) 山口 明徳: インタラクティブ流体シミュレーションにおける力覚提示の実装, 京都大学工学部情報学科, 卒業論文 (2007)
13) P. パチェコ: MPI 並列プログラミング, 培風館 (2001)
14) 吉村知晋, 三輪忍, 嶋田創, 中島康彦, 森眞一郎, 富田眞治 中規模コモディティクラスタ向け相互結合網 Three Quads の提案情処研報 (2006-ARC-167), Vol.2006, No.20, pp.79–84(2006)