

力覚を伴うボリュームベース遠隔仮想協働環境における 通信データの時刻同期制御

Synchronous Processing of Toward Real-time Volume-based Haptic Collaborative Virtual Environment with Realistic Sensation

山岡 励[†] 田中孝英^{††} 山口哲^{††,†††} 島田伸敬^{††} 李周浩^{††} 田中弘美^{††}
 Rei Yamaoka, Takahide Tanaka, Satoshi Yamaguchi, Nobutaka T. Shimada, Lee Johoo and Hiromi T. Tanaka

立命館大学 情報理工学部 知能情報学科[†] 立命館大学大学院 理工学研究科^{††}
 (〒525-8577 滋賀県草津市野路東1-1-1, hiromi@cv.ci.ritsume.ac.jp)
 大阪大学大学院 歯学研究科 統合機能口腔科学専攻 顎口腔機能再建学講座^{†††}
 (〒565-0871 大阪府吹田市山田丘1-8, e-mail: yamagu@dent.osaka-u.ac.jp)

Abstract: We have developed a novel volume-based haptic collaborative virtual environment system, which allows participants at remote sites on the network to simultaneously interact with the same target object in virtual environments, by only exchanging a small set of manipulation parameters for the target object. We first developed an online-remesh volume model, we call dynamic adaptive grids, for simulating deformable objects such as soft tissues in each remote site. Then haptic sensation during interaction with the target object, is achieved by rendering the reflection force from the object, simulated with the online-remesh volume model from the manipulation parameters exchanged among all remote sites. This time, we improved the synchronization precision of the system.

Key Words: Haptic Collaborative Virtual Environment, Haptic Communication, Dynamic Adaptive Grids

1. はじめに

近年、バーチャルリアリティ技術の進歩に伴い、遠隔地間でネットワークを介して様々な協働作業を行う研究が盛んに行われており、医療、教育、産業分野への応用が期待されている。仮想協働環境の医療分野における研究として、生体や臓器等柔軟物を対象としたサーフェイスモデルで幾何ベースの柔軟物シミュレーションが一般的である。そのため、変形モデルで力学的物理特性がパラメータ化されておらず、対象臓器の粘弾性特性の違いや摩擦に応じた任意の物理法則に忠実なシミュレーションは困難であり、高速高精度の物理ベース変形モデルを開発することが課題となっている。

本研究ではハプティックコミュニケーションと呼ぶ、実世界における力覚を実時間で提供する物理モデルでボリュームベースの変形モデルの臨場感通信を提案する[3]。実世界に忠実な情報提示をするために、タイムスタンプを用いて同期用サーバで同期処理を行っている。本稿では遠隔仮想協働環境の同期手法について述べる。従来システムではOSをWindowsで構成していたが、タイムスタンプを用いてシステムを同期させる場合において、時刻取得の精度が悪いためPCの内部時間を各地点で精度よく一致させることは難しい。また、サーバでの同期処理にも不具合が生じる。そこで、従来システムのOSを時刻取得精度のよいLinuxに変更し、同期アルゴリズムを説明した上で実験により同期精度の向上を検証する。

2. システムの概要と構成

本システムでは図1に示すように各地点で同一ボリュームモデルを持ち、操作パラメータとタイムスタンプを送受信して、これらのパラメータを基に各地点で同様の変形計算を行う。各地点のユーザはハプティックデバイス (PHANTOM Omni[1]) のペン型インタフェース (以下、スタイラス) を操作して仮想空間内の物体の任意の点を引っ張る/押し込むことができる。また、各ユーザは同時にインタラクション可能であり、互いに力覚を感じられる。

3. 力覚情報の通信

本研究では、HapticCommunicationToolkit (以下、HCT) [2] を用いる。HCTは、力覚情報の通信を目的として開発されたネットワークライブラリである。また、力覚通信に必要な1000[Hz]という高フレームレートを実現するために、パケットサイズが256[byte]までという制限があり、この制限を超えるよう拡張することも可能であるが、データサイズに比例して高速な通信回

線が必要となるため、現在の一般的な回線 (100[Mbps] ~ 1[Gbps]) では実現困難である。そこで、この制限内に収まるように、力覚通信に必要な最少の操作パラメータのみを送受信することにより実時間力覚通信を実現する。

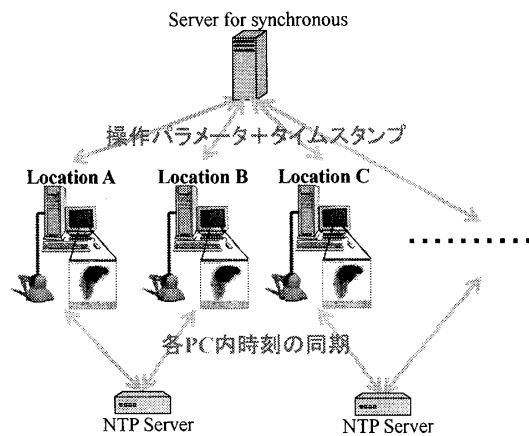


図1 N地点でのシステム構成

3.1 同期手法

各地点間で直接パラメータを送受信する場合、通信遅延によりパラメータの組み合わせに差異が生じる。そこで、本研究では図1に示すように同期用サーバを設けて各地点のパラメータを集め、タイムスタンプを用いて揃えてから送信することで各地点における変形状態の同期を実現する。まず、各地点のPCの時刻をNTP (Network Time Protocol) サーバと同期しておき、通信開始後各地点からタイムスタンプを含むパラメータをサーバに送信する。サーバでは各地点から受信したタイムスタンプを用いて時刻の最も近いパラメータ同士を組み合わせ各地点に配信する。参照するNTPサーバはクライアントのPCからの地理的距離が近いものを選ぶ。Location A, B と Location C の地理的距離が離れている場合、それぞれに最も近いNTPサーバを参照する。なお、NTPサーバのIPアドレスは一般に公開されている。

3.2 Linux へのシステム移行

従来の本研究ではタイムスタンプの取得に Windows 標準ライブラリの関数 `Getlocaltime` を使用した。この時刻取得精度は約 10[msec] であり 100[msec] 単位の精度の範囲内であれば使用可能

である。しかし、本研究では 1000[Hz]で更新されている値が異なる操作パラメータに同じタイムスタンプをつけてしまうことになる。したがって HCT は 1000[Hz]で送受信を行っているので、時刻取得の精度は 1[msec]以上の精度が必要となる。そこで OS を Linux に変更し、標準ライブラリ関数である Gettimeofday 用いるためシステムをすべて移行した。この関数の時刻取得精度は μsec のオーダーである(図 2)。また、NTP サーバと同期した際 Linux における各地点の PC の時刻誤差は 1[msec]以下である。

Windows:Getlocaltime()	Linux:Gettimeofday()
Time=42276.15	Time=1232091589.831715
Time=42276.62	Time=1232091589.831716
Time=42276.78	Time=1232091589.831717
Time=42277.09	Time=1232091589.831718
Time=42277.40	Time=1232091589.831719
Time=42277.40	Time=1232091589.831720
Time=42277.71	Time=1232091589.831721

図 2 Windows と Linux のタイムスタンプの精度

3.3 N地点での同期アルゴリズム

以下の図3はN=3の時、配列に格納したタイムスタンプの値のみを示している。アルゴリズムの説明のため、タイムスタンプの値を簡略化してある。図3において上側に配列番号、左側に要素番号を示す。

Array1	Array2	Array3
0 1354.003	0 1354.001	0 1354.008
1 1354.004	1 1354.010	1 1354.012
2 1354.006	2 1354.012	2 1354.015
3 1354.009	3 1354.013	3 1354.019
4 1354.011	4 1354.014	4 1354.022
5 .	5 .	5 .
6 .	6 .	6 .
7 .	7 .	7 .

図 3 同期アルゴリズムの説明

- すべてのクライアント 1-N とサーバとの通信を確立
- 各クライアントはサーバにタイムスタンプ付きの操作パラメータを送信
- サーバはあらかじめ配列を N 個(領域を 1000 とする)用意して、クライアント 1 のデータを配列 1 に、クライアント 2 のデータを配列 2 に、クライアント N のデータを配列 N に入れる
 - この配列の領域が満たされるまでクライアントに 0 を送信
- サーバはタイムスタンプの順(時刻の早いもの順)で配列をソート
- クライアント数が奇数の時は(N+1)/2、偶数の時は(N+2)/2 番目の配列のデータを基準に設定
- 基準となった配列の値(タイムスタンプ)と、それぞれのクライアントの配列のタイムスタンプとの差分が閾値(5msec)以内だったら、送られてきた順に送り返す。さらにそれぞれの配列の添字を 1 増加
- 差分が閾値よりも大きかった場合
 - 基準となった配列の値(タイムスタンプ)より、時刻の早いタイムスタンプの値をもつクライアントの配列は、そのクライアントの配列の添字を 1 増加
 - 逆に基準よりも時刻の遅いタイムスタンプの値を持つクライアントでは、配列の添字を 1 減少
 - 条件が 7.1 で、差分が閾値以内になり送信したあとに残りの添字も 1 増加
 - 条件が 7.2 で、差分が閾値以内になり送信したあとすべての配列の添字を 1 増加
- 2-7 を繰り返す(3.1 は省略)

4. 実験

本章では、同期手法の有効性を検証するために腹腔鏡下胆嚢摘出術のような基本的な手術で重要となる「2人のユーザ(術者と助手)が柔軟な臓器を引っ張り合う」というタスクを想定し、各地点の操作パラメータの組み合わせ精度の比較をシステ

ム構成が Windows である場合と Linux である場合で行う。

4.1 実験条件

クライアント2台と同期用サーバの計3台のPCをOSがWindows, Linuxであるものをそれぞれ用意する。ボリュームモデルの、左右任意の2点に正弦波状の入力を与えて引っ張り合うタスクを再現した。その際組み合わせられた操作パラメータのタイムスタンプの差分を計測した。また、実験はLAN環境で行い、遠隔地間を想定して100[msec]の遅延をかけた。実験に用いたPCのスペックを以下の表1, 2に示す。

表 1 Windows 版システムシステム構成

	OS	CPU[GHz]	Memory[GB]
Server	Windows XP SP2 32bit	Intel(R)Core2Duo3.40	2
Client	Windows XP SP2 64bit	Xeon X5355 2.66	8

表 2 Linux 版システム構成

	OS	CPU[GHz]	Memory[GB]
Server	Fedora core 6-i386	Intel(R)Core2Duo3.40	2
Client	Fedora core 6-X86_64	Xeon E5450 3.0	16

4.2 結果・考察

組み合わせられた操作パラメータのタイムスタンプを差分した結果が以下の図 4 である。

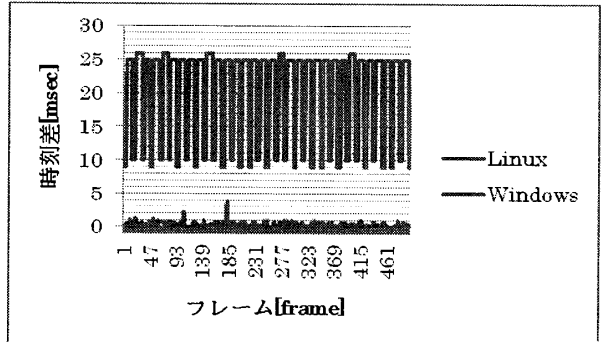


図 4 時刻誤差の比較

システム構成が Windows の時刻誤差の平均は 13[msec]であった。各クライアントは図 4 の時刻差がある操作パラメータを使用して変形計算を行う。ただし、各クライアントに組み合わせられて送られてきたタイムスタンプの差が小さいほど実時間に忠実な情報提示をしていることになる。一方、Linux でシステムを再構成した場合、組み合わせられたタイムスタンプの時刻誤差が平均 0.35[msec]の情報を使って変形計算を行っている。HCT が 1000[Hz]で操作パラメータを更新しているのに対し、時刻誤差 1[msec]以下で組み合わせることができているので、OS を Windows から時刻取得精度のよい Linux に変更したことで同期の精度が向上したことがわかる。なお、Windows のグラフの軌跡が規則的なのは OS に依存するものと考えられる。

5. まとめ

本稿では、ハプティックコミュニケーションと呼ぶ、物理ベースボリュームモデルを用いた遠隔協働環境を実現する同期手法について述べ、実験によりLinuxでシステムを構成することで同期精度が向上し、実時間を考慮した操作パラメータを各クライアントの入力として与えることにより実世界により忠実に情報を提示することができた。今後の課題として、組み合わせるパラメータにどの程度の時刻誤差があると違和感があるのか主観評価をすることや、各地点での変形計算の更新周期の同期などが挙げられる。

参考文献

- [1] Phantom Omni User's Guide, 2004.
- [2] Haptic Communication Toolkit User's Guide, 2006.
- [3] Takahide Tanaka, Satoshi Yamaguchi et al "Toward Volume-Based Haptic Collaborative Virtual Environment with Realistic Sensation" Proceedings of ISUC2008 2nd International Symposium on Universal Communication, pp268-273, 2008
- [4] 高間康文, 辻野圭則, 堀洋介, 田中弘美, "柔軟物体の適応的四面体ボリュームメッシュのオンラインリメッシュ法" 日本バーチャルリアリティ学会論文誌, Vol. 13, No. 1, pp.69-70, 2008