



多重解像度モデルを活用した遭遇型力覚提示システムの開発

Development of encounter type haptic system using multiresolution model

曾根順治¹⁾, 夏目未来¹⁾, 正田大知¹⁾, 中島乃樹¹⁾, 長谷川晶一²⁾

Junji SONE, Mirai NATSUME, Daichi SHODA, Daigo NAKAJIMA, and Shoichi HASEGAWA

- 1) 東京工芸大学 工学部 (〒243-0207 神奈川県厚木市飯山 1583, sone@cs.t-kougei.ac.jp)
2) 東京工業大学 精密工学研究所 (〒226-8503 神奈川県横浜市緑区長津田町 4259, hase@pi.titech.ac.jp)

概要: 3本の指に力覚を提示するシステムを開発している。このシステムは、力覚提示位置や力覚の角度を変更できる遭遇型のシステムである。力覚提示位置を効率良く実現するために、多重解像度モデルを活用して、接触前から接触位置を推定しながら、力覚提示位置を変更する機能を開発した。

キーワード: 力覚提示システム, 遭遇型, 提示位置変更, 多重解像度

1. はじめに

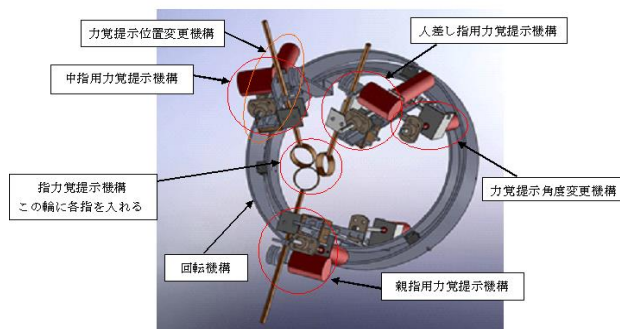
本近年, 3D-CG 技術の発展によって多くの力覚・触覚技術の研究が行われ, 今後の研究開発の発展により医療, エンターテインメントにおいて力覚提示を用いた VR シミュレーションが可能となると考える。現在まで, SPIDAR⁽¹⁾ のような多くのハプティックデバイスが考案されている。また, Spidar8 や Cyber Grasp においても物体と指の接触点固定されている問題がある。日常の作業においては, 指の接触点が増減しており, 高い精度の人工現実感を得るためには, 横小路先生の提唱されている遭遇型⁽²⁾のシステムが必要である。本研究においては, 力覚提示点を変更できる力覚提示機構を設計し, 力覚提示位置を変更する機構を開発している⁽³⁾。従来では, Springhead⁽⁴⁾ と Unity⁽⁶⁾ を連携して, VR シミュレーションと力覚提示制御を行っていたが, 本システムの特徴である遭遇型の力覚提示位置の変更や力覚の角度変更の機能までは開発できていなかった。力覚提示の精度向上のため, 力覚提示とは別のモータ制御システムを開発して, 力覚提示位置の変更する機能を開発し, さらに, Unity 上でも, 多重解像度モデルを採用して, 接触位置を予測しながら, 接触位置位置を接触前から変更する機能を開発した。

2. システム開発概要

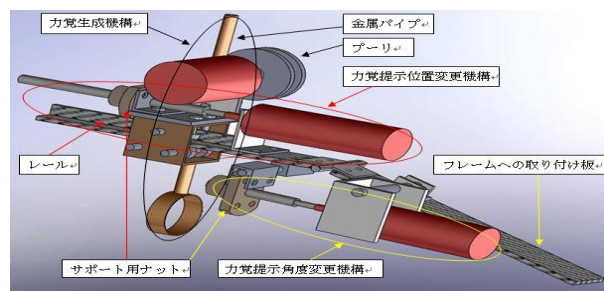
2.1 力覚提示デバイス

図 1 a) は力覚提示デバイスを示す。リングに取り付けられた 3つの力覚提示機構はそれぞれ親指, 人差し指, 中指に対応する。また全ての力覚提示機構は内リングに

固定されている。各指には力覚提示機構が備わっており, 糸を巻きつけたプーリを装備したアクチュエータ制御用モータで駆動される。また, 力覚提示がない場合は指力覚提示部を指で自由に移動させることが可能である⁽⁵⁾。



a) デバイスの全体構成



b) 力覚位置, 角度変更機構

図 1. 多指力覚提示デバイス

また、図 1 b) は力覚位置、角度変更機構を示す⁵⁾。

2.2 システム開発方法

システムは、Unity と SpringLib から構成される。SpringLib は、SpringHead のインターフェイス制御、デバイス制御、数学ライブラリを抽出して DLL 形式にライブラリ化している。その概要を図 2 に示す。SpringLib は指のデバイス制御の Hand3 クラスと、手首の位置と姿勢を制御する SpidarG6 クラスから構成される。

Unity では、グラフィック、衝突判定、物理計算を行う。力覚提示デバイスで得た指の位置情報を SpringLib の関数から受け取り、3D-CG 空間内に反映する。この実装により、3D-CG の指が、被験者の指に装着したデバイスの動きに合わせて動かすことが可能になる。また、3D-CG 空間内で指と物体が衝突した際に、抗力を計算し SpringLib のモータ制御機能を使い、力覚を発生させる。これは、モータと糸で指フックの力制御を行う。

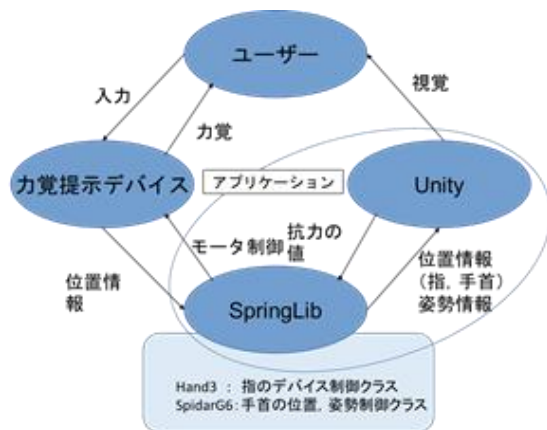


図 2. 開発システムの構成

力覚位置変更システムの開発は、制御プログラム(C#言語で記述)から直接、モータ制御が行える P C I 制御ボード(株式会社コスモテックス, PXP-G-88)を導入し、各指における力覚提示位置変更機構と力覚提示角度変更機構を自作アンプによって制御する方法を導入した。自作アンプはモータ制御ボードからのモータの回転数と方向の命令を受け、回転方向の方向制御回路を経て、電流制御アンプの高出力電圧をモータに送り、モータの回転方向、回転を制御して位置決めを行う。

Unity を用いて開発した仮想空間システムで、指と物体との衝突位置を予測するプログラムを追加開発し、触覚の生成位置を変更する機能を開発した。衝突位置の予測は力覚位置変更機構と力覚角度変更機構のモータの回転の速度が遅いため、衝突位置を予測しながら、指が、接触物体に近づいた状態から、徐々に接近するように工夫した。このプログラムでは多重解像度で使われる八分木を利用している。図 3 は八分木を使用して衝突位置の予測する説明図である。把持物体(Obj)と指(Yubi)オブジェクトの二種類の衝突位置を予測するために、八分割を行う回数の MaxLevel を設定する。設定したオブジェクトの

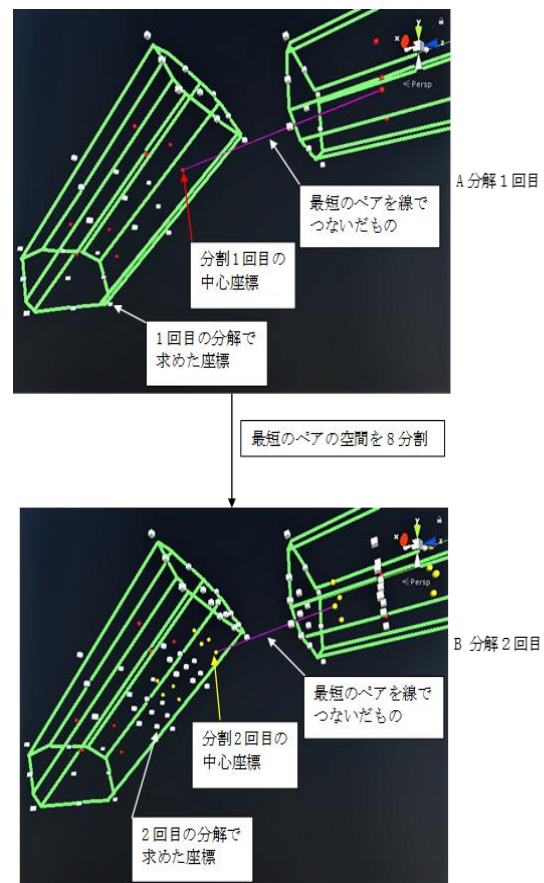


図 3 3次元空間で動かしたときの最短距離計算

x,y,z 軸の最大値と最小値を求め、そのオブジェクトを覆う直方体のサイズを求める。求めた直方体を八等分し、8つの直方体の空間を生成し、各空間の x,y,z 軸の値が異なる頂点 2 つを使用し中心の座標を求め、Obj の分割直方体の中心座標と Yubi の分割直方体の中心座標から、8×8 の組み合わせで距離を計算する。そして、Obj と Yubi の距離が一番近い組み合わせを探す。先ほど中心座標を求めるときに使用した 2 つの頂点を元にその空間を、さらに八等分する。そこでも、最小距離の組み合わせを求め、一番近いものの組み合わせを判別する。これを設定した MaxLevel 回数行いながら、Obj と Yubi がどの位置で衝突するかを推定する。そして、触覚生成プログラムに位置情報を渡す。白色の四角形は分割したときに求めた座標を表し、赤色の丸は分解 1 回目中心座標を表し、黄色の丸は分解 2 回目中心座標を表し、紫色の線が一番近い組み合わせを表す。

3. むすび

遭遇型多指力覚提示機構を実現するために、モータ制御機能を拡張して、力覚提示時に力覚提示位置変更機能を同時に動作する機能を、Unity 上で開発している仮想現実システムに追加開発した。また、力覚提示位置を、高速に計算するために、CG モデルを多重解像度表現により、階層的に接触位置の精度を高めながら、少ないステップで、接触位置を計算する機能を開発できた。今後は、触

覚の方向を制御する機能の開発を進める予定である。

謝辞

本研究は、科学研究助成事業(JSPS KAKENHI 17K00285, 21K12002)の助成を受けており、感謝の意を表す。

参考文献

- [1] M. Sato: Development of string-based force display:Spidar. Proc. The Eighth International Conference on Virtual Systems and Multi Media VSMM 2002, pp.1034–1039, 2002.
- [2] Y. Yokokohji Y.: Design and path planning of an encountered-type haptic display for multiple fingertip contacts based on the observation of human grasping behavior, IEEE International Conference on Robotics and Automation, pp.1986—1991, 2004.
- [3] J. Sone, T. Mori, O. Itamoti, S. Hasegawa, M. Sato: Development of Mechanism in Multi-finger Haptic Display - Development of Haptic mechanism and fusion with Spidar - Proceeding of ASIAGRAPH 2008, pp.91-96, 2008.
- [4] Springhead Project Website : <http://springhead.info/>.
- [5] J. Sone, S. Hasegawa, et. al. Development of MEMS Tactile Sensation Device for Haptic Robot, Journal of Robotics and Mechatronics, Vol.32, No.2 Special Issue on MEMS for Robotics and Mechatronics, pp. 315-322, 2020.
- [6] Jonathan Linowes : Unity による VR アプリケーション開発, オライリー・ジャパン, 2016.