



半姿勢・半拘束型力触覚インターフェース

A Semi-posture and Semi-constrained Haptic Interfaces

加藤聖人¹⁾, 脇田 航¹⁾

Kiyoto KATO and Wataru WAKITA

1) 広島市立大学大学院情報科学研究科 (〒 731-3194 広島市安佐南区大塚東 3-4-1, kato@vr.info.hiroshima-cu.ac.jp)

概要: 本研究では, 手指の各関節を中間姿勢(半姿勢)で半拘束することで, 手指からの力入力から視覚による位置出力によって力触覚を錯覚させる半姿勢・半拘束型力触覚インタフェースを提案し, 手始めに指を半姿勢で半拘束した状態において, 手指からの力入力から手首の位置推定と, 指の把持動作を推定し, 視覚呈示によって形状感や摩擦感, 重量感, 硬さの表現を行った. 実験の結果, 力入力に応じた位置変化を視覚的に呈示することができ, 疑似的なアドミタンス型の力覚呈示を実現することができた.

キーワード: 半姿勢, 半拘束, 力触覚, クロスモーダル

1. はじめに

より現実に近い VR 体験を実現するためには, ユーザが VR 空間内のオブジェクトを実際に触っているかのように感じられる技術が求められる. そのためには, 押下感, なぞり感, 摩擦感などの多様な力触覚を精密に再現することが重要であり, ユーザの手指からの入力を正確に読み取り, 適切な感覚を呈示するメカニズムが必要となる.

現状, 様々な力触覚呈示手法 [1] が提案されているが, 例として力覚呈示装置と Pseudo-haptics の二つの手法をあげる. 力覚呈示装置には位置入力-力出力のインピーダンス型と力入力-位置出力のアドミタンス型に大別できる. インピーダンス型はエンコーダ等でエンドエフェクタの位置を計測し, 反力をアクチュエータによって呈示するため, エンドエフェクタへの操作を軽くするためのバックドライバビリティの確保が重要となり, 操作範囲を広げることが困難となる. 一方, アドミタンス型は力センサでエンドエフェクタにかかる力から位置を決めるため, 大型化しやすく操作範囲を広くとりやすいが, その分様々なコストがかかってしまう. いずれも複雑な形状や質感を精密に再現することは困難である. 一方, Pseudo-haptics は, ユーザの入力と視覚呈示の不一致から生じる錯覚を利用する手法である. 必ずしも物理的な刺激を必要とせず, 振動や音等の呈示と組み合わせる等, 比較的簡素な機構で様々な感覚を知覚しやすく, 近年のモーションキャプチャによるフルボディ型 VR との相性もよいという利点がある.

しかしながら, モーションキャプチャベースの Pseudo-haptics では力触覚に重要な要素となる作用に対する反作用が生じないため, ユーザとアバタの状態が離れるほど違和感となってしまう. 力触覚呈示装置ほどの力触感を錯覚させることは困難である. また, 空間的な制約や, 感覚生起に個人差があること, 感覚の強度を高めるためには実空間

と VR 空間の位置の乖離を大きくする必要があるが, その乖離が過度に大きくなると不快感を招くというデメリット [2] も存在する.

そこで我々は, 手指の各関節を中間姿勢(半姿勢)で半拘束することで, 手指からの力入力から視覚による位置出力によって力触覚を錯覚させる半姿勢・半拘束型力触覚インタフェースを提案する. 本手法は Pseudo-haptics において作用反作用を生じさせるためにユーザの手指を拘束するが, 硬軟感を錯覚させるためにユーザの手指を硬いと柔らかいの中間硬さで拘束(半拘束)し, 手指であればグーとパーいづれの状態も錯覚しやすいようにユーザの各関節の可動域の中間状態(半姿勢)で半拘束する. 各指からの力入力からアバタの手指を操作することで, 力入力-視覚による位置出力となり, アドミタンス型の力覚呈示装置のような作用をクロスモーダル効果によって生起させることが期待できる. 本研究では手始めに指を半姿勢で半拘束した状態において, 手指からの力入力から手首の位置推定と, 指の把持動作を推定し, 視覚呈示によって形状感や摩擦感, 重量感, 硬さの表現を行う.

2. システム概要

図 1 に提案システムの概要図を示す.

2.1 力入力システム

力センサには片持ち梁タイプのビーム型ロードセルを用いて力入力とし, その値を用いて, バーチャル空間内のオブジェクトを動かす. また, ロードセル上に指を置くことで指を半姿勢で半拘束する. この状態でオブジェクトを操作する方法として, ロードセルを半拘束状態の指で押すことによって生まれるロードセルの歪みによって抵抗値を得る. この抵抗値を HX711 コンバータでデジタルデータに変換し, マイコン (Arduino) 経由で PC にシリアル通信で送信する. PC 上の Unity では, 受信したデータを C# スクリ

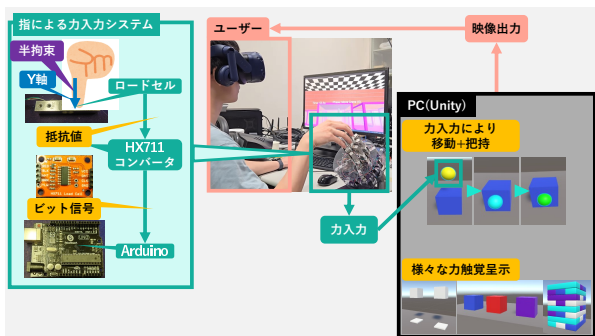
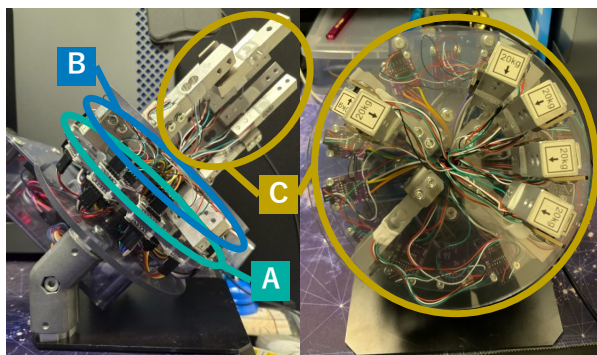


図 1: システム概要図

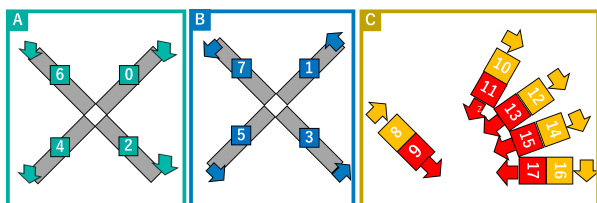
プトで数値に変換し、指からの入力値として利用する。

2.2 使用したデバイス

図 2 に使用するデバイスの外観とロードセルの配置を示す。



(a) デバイスの外観



(b) ロードセルの配置

図 2: 手指入力プラットフォーム

ロードセルの配置は全 3 層 (A,B,C) に分かれており、それぞれの数字が L_0 から L_{17} のロードセルに対応し、矢印の方向が入力値の正の向きである。本研究では、A 層である L_0, L_2, L_4, L_6 を使用し、緑色の矢印の方向を正方向の力入力値とする。また、C 層である $L_9, L_{11}, L_{13}, L_{15}, L_{17}$ を使用し、赤色の矢印の方向を指の屈曲動作の方向の力入力値として使用する。

2.3 手首の位置推定手法

VR 空間内での操作者の手の位置は、力入力から物理シミュレーションによって目標位置を逐次的に算出する。このプロセスは、1. 力の算出と非線形化、2. 質量に応じた移動減衰、そして 3. 最終的な目標位置の決定、という 3 つの主要なステップで構成される。算出された目標位置に対し、PD 制御を適用することで、応答性と安定性を両立した手の物理的な追従動作を実現する。

2.3.1 力の算出と非線形化

まず、ロードセルから得られる値 L_i を用いて、VR 空間における力の大きさ P_k ($k \in x, y, z$) を次式で算出する。

$$\begin{aligned} P_x &= L_0 + L_2 - L_4 - L_6 \\ P_y &= -(L_0 + L_2 + L_4 + L_6) \\ P_z &= L_0 - L_2 - L_4 + L_6 \end{aligned} \quad (1)$$

次に、算出された力の大きさ P_k を 2 乗することで、入力に対する応答を非線形化する。この処理により、操作者が加える力が小さいときには微細な制御を可能にしつつ、力が大きい領域ではより大きな動きを表現することが可能となる。力の方向を考慮するために $\text{sign}(P_k)$ を乗算する。さらに、この非線形化された力にゲイン値 T_h を乗算し、移動方向ベクトル $\vec{F}_k$ を次式によって決定する。

$$\vec{F}_k = \text{sign}(P_k) \cdot P_k^2 \cdot T_h \quad (2)$$

2.3.2 質量に応じた移動減衰

さらに、オブジェクトを把持している場合には、その質量 m_{grip} に基づいて移動量を減衰させる。これは、重い物体を操作する際の慣性を表現するための簡易的な処理である。影響係数 f_{mass} を用いて算出される減衰率 d_{mass} は、次式によって導出され、最終的な移動ベクトルに乗算される。

$$d_{\text{mass}} = 1 - \text{clamp}_{0,1}(m_{\text{grip}} \cdot f_{\text{mass}}) \quad (3)$$

2.3.3 最終的な目標位置の決定

これまでの計算結果を統合し、手の現在位置 $\vec{p}_{\text{current}}$ から、時間 Δt 後の最終的な目標位置 $\vec{p}_{\text{target}}$ を次式によって算出する。

$$\begin{aligned} \vec{F}_{\text{final}} &= \vec{F}_{\text{move}} \cdot d_{\text{mass}} \\ \vec{p}_{\text{target}} &= \vec{p}_{\text{current}} + \vec{F}_{\text{final}} \cdot \Delta t \end{aligned} \quad (4)$$

手の推定位置は (4) 式によって導出した目標位置に対し、PD 制御を用いて追従することによって求める。PD 制御を適用することにより、力入力の外れ値に起因するポイントの意図しない急峻な挙動を抑制し、滑らかで安定した追従動作を実現する。

2.4 オブジェクトの把持判定

物体把持の判定処理は、候補検出、距離測定、および力入力の 3 段階で実行される。これにより、ユーザは力入力によって対象物を「掴む/掴まない」を直感的に切り替えることができる。まず、操作者の手の周囲に設定された検出範囲内にある把持対象レイヤのオブジェクトを広域的に探索する。このときの手のカーソルは黄色であり、これがデフォルトの色である。次に、検出された各オブジェクトに対し、操作者の手のコライダ表面と対象オブジェクトのコライダ表面との間の最短距離を算出し、この距離が接続閾値以下で最も近いものを把持候補とする。このとき、手のカーソルが水色に変化する。最終的に、複数のロードセルから得られる指の力入力値が、対象オブジェクトの質量に応じて算

出される閾値を超えた場合に、FixedJoint コンポーネントを用いて対象オブジェクトとの物理的な接続を確立し、把持が完了する。把持が完了したとき、手のカーソルが緑色に変化する。図 3 に手のカーソルの変化の様子を示す。

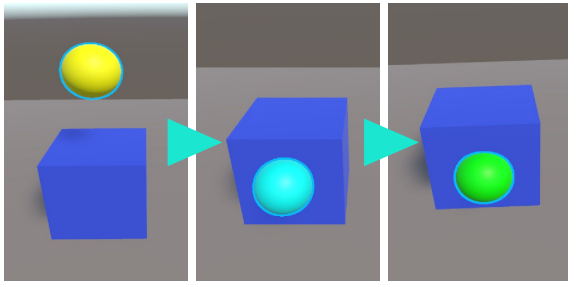


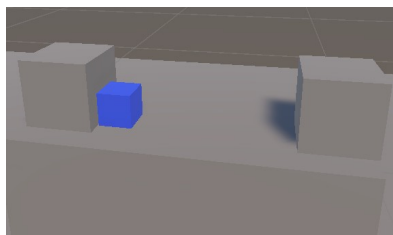
図 3: 手のカーソルの変化の様子

上記の処理によって Unity 上の手のカーソルを推定した手首の位置とし、距離判定や力入力値の閾値判定などを経て持つ 3 次元マウスのような操作感を実現する。

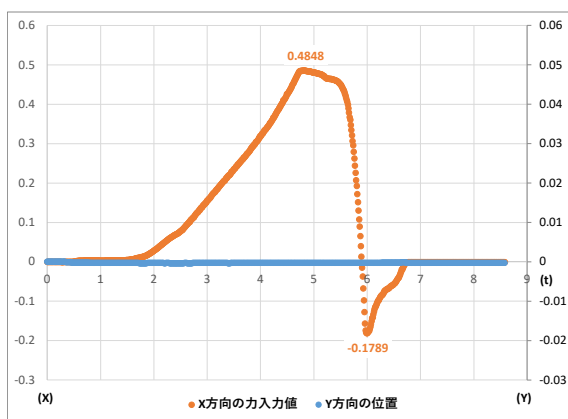
3. 評価実験

3.1 形状感に関する評価実験

Unity 内のオブジェクトを把持した状態で力を変化させながら左右にオブジェクトを移動させた。図 4 に実験環境と力入力と位置の結果を示す。



(a) 実験環境



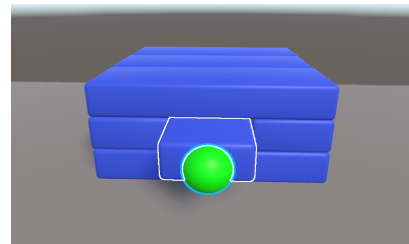
(b) 力入力と位置の結果

図 4: 形状感に関する評価実験結果

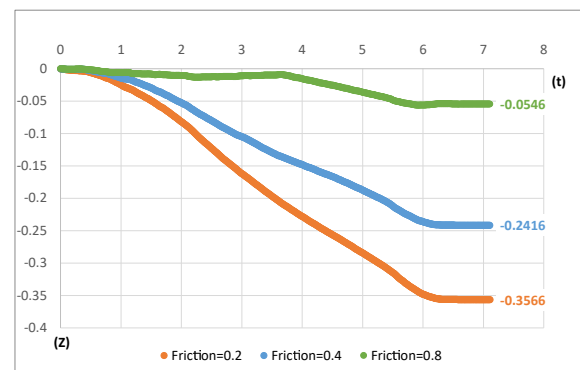
図 4(b) の左側の縦軸がなぞった方向である X 軸方向に対してのデバイスからの力入力値であり、右側の縦軸がなぞる動作に使用しているオブジェクトの Y 軸方向の位置である。実験結果のグラフから力を変化させながら表面をなぞったときも同じ高さで物体をなぞれることが確認できる。

3.2 摩擦感に関する評価実験

物体の摩擦感を擬似的に呈示する実験環境として、ジェンガを模した構造物を構築した。実験タスクは、この構造物から白色のアウトラインでハイライトされたブロックを引き抜くことである。ブロック間の摩擦感を変化させるため、Unity の Physic Material における動摩擦係数と静止摩擦係数を、0.2 (橙色)、0.4 (水色)、0.8 (黄緑色) の 3 段階に設定した。そして、各摩擦条件下でデバイスからの力入力を厳密に統一するため、まず基準となる摩擦係数 0.2 で引き抜き操作を行った際の力入力の時系列データを記録する。すべての条件下でそのデータを力入力値として再生し、実験を行った。図 5 に実験環境と実験結果を示す。



(a) 実験環境



(b) ブロックを引き抜く際の摩擦係数ごとの位置変化

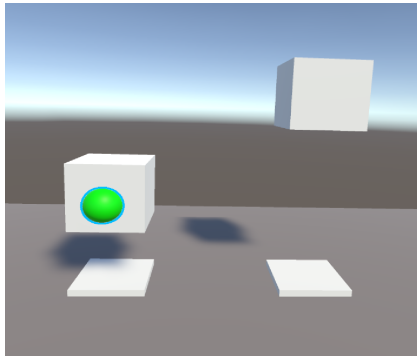
図 5: 摩擦感に関する評価実験結果

図 5(b) の縦軸がブロックを引き抜いた方向 (負) で、横軸が時間である。上記の実験結果から、動摩擦係数、静止摩擦係数を増加させることによって物体間の摩擦が増加し、物体の移動量が減少していることから視覚的な摩擦感を呈示できていることがわかる。

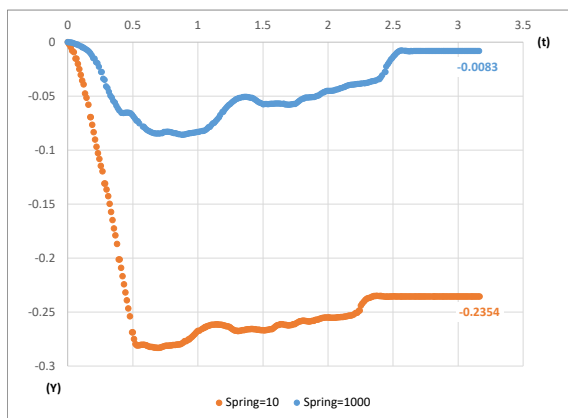
3.3 硬さに関する評価実験

物体の硬さを擬似的に呈示させるために図 5(a) の実験環境を用意した。実験タスクは上部の立方体のオブジェクトを手のカursorで把持し、下部の土台に向けて押し込むことである。押し込む硬さの表現と押し込んだ際に再び初期位置に戻すために Unity の Spring Joint コンポーネントを用いている。硬さの違いを表現するために Spring Joint コンポーネントのばね係数を 10 と 1000 の 2 種類を用意した。実験結果は図 6(b) の通りである。グラフの縦軸がオブジェクトを押し込んだ方向である Y 軸方向の変位で、横軸が時間である。各ばね係数下でデバイスからの力入力を厳密に統一するため、まず基準となるばね係数 10 で Y 軸方

向に押し込む操作を行った際の力入力の時系列データを記録し、すべての条件下でそのデータを力入力値として再生している。



(a) 実験環境



(b) 同じ力入力による位置の変化

図 6: 硬さに関する評価実験結果

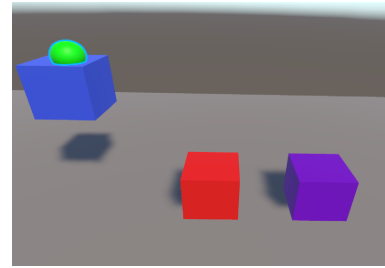
図 6(b) の結果から、同じ力入力をしているがばね係数が高いほど押し込み方向の変位量が減っており、視覚的な硬さを呈示できていることが確認できる。

3.4 重量感に関する評価実験

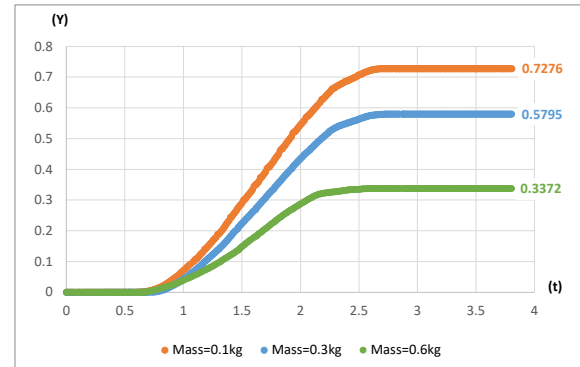
持ち上げるときの重量感の違いを疑似的に呈示するために図 7(a) に示す 3 種類の重さの違うオブジェクトを配置した実験環境を構築した。実験タスクはこのオブジェクトを持ち上げることである。

この 3 つのオブジェクトの質量をそれぞれ 0.1kg, 0.3kg, 0.6kg の 3 種類に設定した。持ち上げ動作時のデバイスの力入力を厳密に統一するために、まず基準となる質量=0.1kg で持ち上げ動作を行い、力入力の時系列データを記録する。すべての条件下でそのデータを力入力値として再生し、実験を行った。図 7(b) に実験結果を示す。

グラフの縦軸が持ち上げた高さである Y 軸方向の変位で、横軸が時間である。図 7(b) の結果から、同じ持ち上げ動作時の力入力値でも持ち上がった高さの最大値がオブジェクトの質量によって変化しており、重いオブジェクトほど持ち上がっていないことから、視覚的な重量感の違いを呈示できていることが確認できる。



(a) 実験環境



(b) 同じ力でオブジェクトを持ち上げる際の位置変化

図 7: 重量感に関する評価実験結果

4. むすび

本研究では、手指の各関節を中間姿勢（半姿勢）で半拘束することで、手指からの力入力から視覚による位置出力によって力触覚を錯覚させる半姿勢・半拘束型力触覚インタフェースを提案し、手始めに指を半姿勢で半拘束した状態において、手指からの力入力から手首の位置推定と、指の把持動作を推定し、視覚呈示によって形状感や摩擦感、重量感、硬さの表現を行った。実験の結果、力入力に応じた位置変化を視覚的に呈示することができ、疑似的なアドミタンス型の力覚呈示を実現することができた。

今後の課題として、現状では手をカーソルとして把持動作を掴む/掴まないの二値で行っているが、これを Inverse Kinematics を用いたアバタ制御と剛体による物理的インタラクションを用いた 5 指で制御可能にしていく。また、本システムを用いてユーザが様々な力触覚を知覚できるか主観評価を行い検証していく。

参考文献

- [1] C. Wee, K. M. Yap and W. N. Lim: Haptic Interfaces for Virtual Reality: Challenges and Research Directions, in IEEE Access, vol. 9, pp. 112145-112162, 2021.
- [2] A. Pusch and A. Lecuyer: Pseudo-haptics: From the theoretical foundations to practical system design guidelines, Proc. 13th Int. Conf. Multimodal Interfaces, pp. 57-64, 2011.