

有限な操作領域をもつ3次元インタフェースデバイスの操作方法に関する研究

瀬崎崇浩¹ 赤羽克仁² 一色正晴² 橋本直己² 佐藤誠²

(1) 東京工業大学大学院 総合理工学研究所

(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)

Abstract: クラッチの状態を自動で切り替えるシステム「Dual Shell Method」を提案した。クラッチ機構というのは3次元インタフェースデバイスとVR空間の連結を制御することで、任意のVR空間を操作出来る方法である。従来はスイッチなどの外部入力によって行われていたが、システム側がクラッチの切り替えを自動で行うことで、ユーザにクラッチを意識させずにインタフェースを操作させるのが研究の目的である。

Dual Shell Method for interacting VR world with limited device space

Takahiro Sezaki¹ Katuhito Akahane²
Masaharu Issiki² Naoki Hashimoto² Makoto Sato²

Abstract: In the field of 3D user interface devices, there is often a mismatch in VR world space and device space. Rate control is one method of solving this problem, which works by translating device position into velocity of a pointer in VR world. However, rate control is less intuitive to use than position control. One method for using position control is by introducing a clutching mechanism to address the mismatch between VR and device space. In this paper, we introduce an automatic clutching method, named "Dual Shell Method," which is easier and more intuitive to use than manual clutching methods.

1 背景

表.1 アイソメトリック型とアイソトニック型デバイス

	isotonic	isometric
2D	mouse touch pad	track point joystick
3D	PHANToM	space navigator

インタフェースデバイスにはアイソメトリック型デバイスとアイソトニック型デバイスの2種類に分類 [2] できる。(表1)

2種類のデバイスの違いは操作中にデバイス変位を初期位置に戻す力が発生しているかどうかである。アイソメトリック型デバイスには初期位置に戻そうとする力が働いており、アイソトニック型デバイスには発生していない。

インタフェースデバイスとポインタを結びつけるカップリング方法はポジションコントロールとレートコントロールの2種類がある。通常ポジションコントロールはアイソトニック型デバイスを使用するとき用いられる。ポジションコントロールの例としてマウスがある。ディ

スプレイ中のポインタ位置と、マウスの位置は1対1に対応している。そしてマウスを右に動かせば、マウスの移動量に比例定数(スケール)を掛けた値(ディスプレイのドット)に変換され、ポインタがディスプレイ上を右に移動するのである。

レートコントロールはアイソトニック型デバイスを使用するとき用いられる。レートコントロールの例としてジョイスティックの操作がある。デバイス変位量はスティックの傾斜量とその方向であるので3次元ベクトルで表せられる。このデバイス変位ベクトルをポインタの速度としてポインタに与える方法がレートコントロールである。

1.1 3次元インタフェースデバイスの特徴

1.1.1 アイソメトリック型

1. メリット

レートコントロールで使用することが出来るので、無限のVR空間中の領域を操作可能である。

2. デメリット

レートコントロールは積分処理をしていることになるので空間的精度に問題がある。またユーザにとって直感的な操作が出来ない。

1.1.2 アイソトニック型

1. メリット

ポジションコントロールで使用するために空間的精度が高く、またグリップ位置がポインタ位置に対応しているため直感的（実際の物体を扱うような）操作ができる。

2. デメリット

デバイス操作領域の VR 空間への対応範囲のみポインタが移動可能であり、無限の VR 空間全域を移動することはできない。

1.2 アイソトニック型デバイス改善の必要性

ユーザビリティを考えると誰にでも使い易いシステムが求められている。そして直感的に正確な動作を3次元空間で行うにはアイソトニック型デバイスが向いている。しかし3次元アイソトニック型デバイスには操作可能領域が有限である問題があり操作領域の問題を解決できるシステムが求められている。

2 関連研究

2.1 クラッチ機能付きポジションコントロール

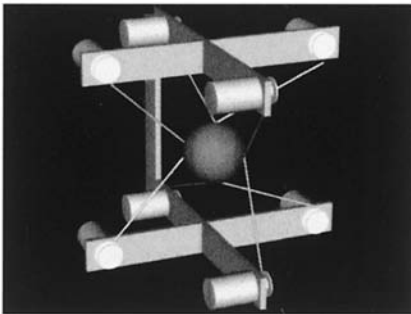


図1 SPIDAR-G

この方法はSPIDAR-G[3]を用いて実装された。(図1)インタフェースデバイスの変位量とは、球形グリップの中心位置を示す3次元ベクトルとグリップ姿勢を示すクォータニオンで表現できる。SPIDAR-Gを起動するとグリップ位置・姿勢情報がデバイスからPCに送られて



図2 SPIDAR-Gのグリップ

くる。またグリップに付加されたスイッチ(図2)からは、ON-OFFの2値情報が送られてくる。通常動作ではクラッチ状態がONであり、グリップ位置・姿勢情報がVR空間中のポインタ位置に対応している。そして、スイッチを押すとクラッチ状態がOFFとなり、グリップ位置・姿勢の変化がVR空間中のポインタに反映されなくなりポインタの位置・姿勢は変化しなくなる。再びスイッチを押すとクラッチ状態がONになりポインタとグリップが連動する。クラッチ状態の切り替えにより任意のVR空間全域を操作することが可能となった。

マウス操作との3次元デバイスの対比を示すと、クラッチ状態をONからOFFに切り替えるとはマウスを空中に持ち上げマウスとポインタ位置の連動を切るということである。クラッチ状態をOFFからONに切り替えるとは空中に持ち上げたマウスをマウスパッド上に戻すという行為に対応している。

問題点

1. 理解の難しさ

クラッチの原理をユーザが理解していないと使えない。

2. 低ユーザビリティ

この方法はユーザの任意のデバイス領域内で使用することが出来る。よってデバイスの空間精度の低い場所で操作することも許容してしまう。精度の低いデバイス領域での使用は避けるべきである。

3. デバイスへの危険

グリップがフレームの外側に出ても、使用が出来てしまうのでデバイスを壊す恐れがある。デバイスの仕様にしたがっての使用を促す必要がある。

2.2 バブルテクニック

関連研究にバブルテクニック [1]がある。(図3)インタフェースデバイスにはアイソトニック型デバイスであるPHANToMを使用している。PHANToMのグリップを動かしVR空間中ポインタを移動することで操作をする。VR空間中にはポインタとそれを取り囲む「バブル」と

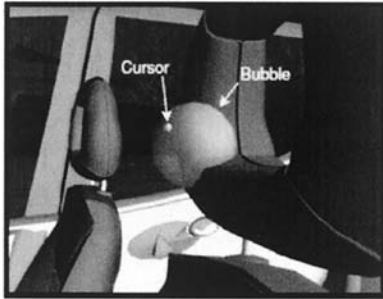


図3 バブルテクニック

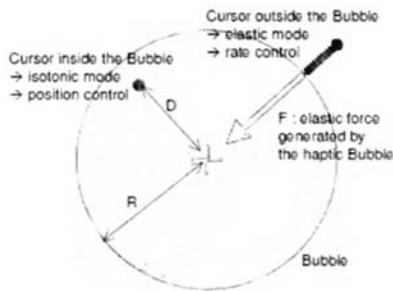


図4 バブルテクニックの概念図

いう球体が表示され、ポインタがバブルの内側に位置するときにはポジションコントロールで動く。そして、ポインタがバブルの境界を越えて移動しようとするときポインタとバブルが同期して移動する。この状態においては、ポインタはレートコントロールで動き、デバイスにはデバイス領域中心にグリップを移動しようとする力が発生するのでアイソメトリック型デバイスとして動作していることになる。(図4)

問題点

1. 低ユーザビリティ

バブルを動かすときには、レートコントロールなので空間的精度が悪い。またポジションコントロールとレートコントロールが切り替わるのでユーザを混乱させる恐れがある。

2. ユーザへのストレス

正確なポインティングをするときは、バブルを作業領域が中心となるように移動しそれからポインタをバブルの中心域まで動かして操作するといった二度手間が生じる。

3. デバイスへの危険

PHANTOM グリップがフレームの外側に出てしまっている場合、使用が出来てしまうのでデバイスを壊す恐れがある。

3 Dual Shell Method の提案

関連研究の問題点を解決する方法として Dual Shell Method(以下 DSM と略す)がある。この方法は自動でクラッチ状態を切り替えることで、外部入力式クラッチ方法の問題点が解決できる。また、バブルテクニックのようにレートコントロールを使うことなく、精度が良いポジションコントロールのみで VR 空間中を操作することが出来る。

Dual Shell Method の原理

ここでは VR 空間にあるポインタをデバイス空間にあるグリップを使って動かす様子を説明する。例としてグリップを x 軸方向に動かしている状況を説明する。図5の上段のグラフは時間と VR 空間中のポインタ位置、中段は時間とデバイス空間中のグリップ位置、下段は時間とクラッチ状態の関係を示したものである。

1. 図5.①の状態ではクラッチ状態=ON である。x 軸正の方向にグリップを動かしている。この状況ではグリップの動きに連動してポインタも VR 空間を X 軸方向に移動している。デバイス空間におけるグリップ位置は図6.①である。
2. 図5.②の状況でグリップを外側 Shell の半径 R まで動かすとクラッチ=OFF となる。クラッチ=OFF なのでグリップの動きがポインタの動きに反映されなくなる。(図5.②)
3. クラッチ=OFF の状態ではグリップをデバイス中心方向(x 軸負の方向)に動かすと誘導力が発生する。デバイス空間におけるグリップ位置は図6.②である。
4. 内側 Shell までグリップを動かすとクラッチ=ON となりグリップに連動してポインタが動かせるようになる。(図5.③)

クラッチ状態が ON のときは外側 Shell がクラッチ状態を切り替えるスイッチの役割を果たし、クラッチ状態が OFF のときは内側 Shell がクラッチ状態を切り替えるスイッチの役割を果たしている。

3.1 操作の分かりやすさ

システムから発生するグリップを SPIDAR-G フレーム中心に戻そうとする誘導力覚により、ユーザはクラッ

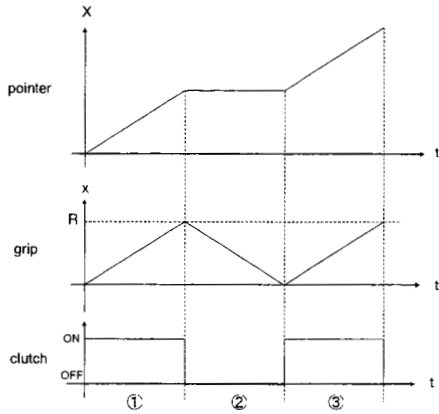


図 5 DSM の流れ

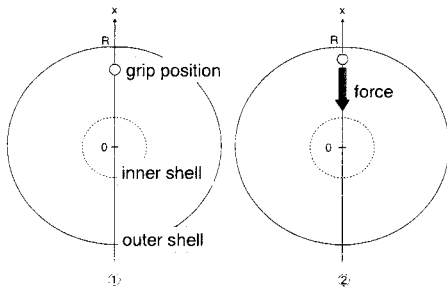


図 6 DSM の概念図・デバイス

手操作を理解していなくても自然にグリップをデバイス中心に動かすことが出来る。そして、クラッチ状態を再び ON に切り替えることが出来る。

3.2 デバイス無依存性

デバイスに依存しないシステムはとても大切である。SPIDAR-G はグリップの大きさや形状の自由度が高く外部入力素子をグリップに付加することは容易である。しかし PHANTOM など、他のアイソトニック型 3次元入力インタフェースにおいては、付加出来る外部入力素子は制約がある。また、ユーザの操作性を考えてもデバイスに付加されているボタンなどの外部入力素子はアプリケーション中の動作（VR 空間中の物体を指定するなど）において使うべきであり、ポインタの移動は外部入力に頼らずに行うべきであると考える。

4 実験 (1)

DSM と外部入力式クラッチの比較実験をする。外部入力式クラッチでは、キーボードの space キーによってクラッチの状態を切り替えるモード、グリップに付加されたボタンスイッチによってクラッチの状態を切り替えるモードの 2 種類を準備した。比較実験は外部入力クラッチ (2 種類) と DSM の計 3 つの方法で比較する。

4.1 実験タスク

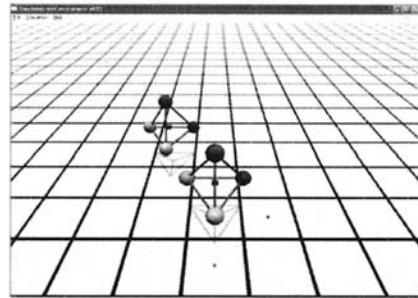


図 7 実験タスク

実験は並進運動におけるドッキングタスクである。オブジェクトは画面手前中央にある 4 面体であり (図 7), SPIDAR-G グリップに運動して動く。このオブジェクトの初期位置を VR 空間の中心位置とする。実験手順について示す。

1. ターゲットがランダムに選ばれる位置に出現
2. オブジェクトを動かしてターゲットに 2[s] ドッキング
3. ターゲットが消え VR 空間の中心位置に新しいターゲットが出現
4. オブジェクトを動かしてターゲットに 2[s] ドッキング
5. 1. の状態に戻る

この 1.-5. の動作を繰り返す。

ドッキング判定 (接触判定) はオブジェクト及びターゲットの 4 面体の頂点の球体で行う。4 つの球体は、それぞれ幾何的位置により色分けされていて、オブジェクトとターゲットはそれぞれ同じ色同士の球体をドッキングさせなくてはならない。ドッキング完了に必要な保持時間は 2[s] とした。

図 8 に示すように、ターゲットは VR 空間の中心位置から 3 つの距離に出現する。それら 3 種類はタスク完了

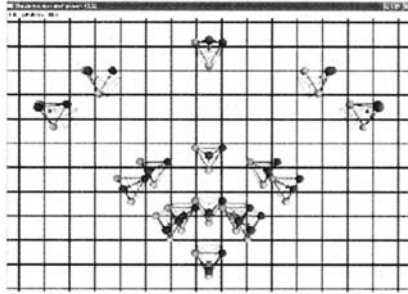


図 8 ターゲット出現位置

表. 2 オブジェクト位置と必要なクラッチ回数

ターゲット位置	距離 [m]	クラッチ回数
遠い	8[m]	4,5 回
中間	4[m]	1,2 回
近い	2[m]	必要なし

に必要なクラッチの回数に応じて選ばれている。(表 2) それぞれの距離に応じて同心円上にターゲットが5つずつ配置されるようにしてあり、ターゲット出現位置は全部で15個ある。

SPIDAR-G グリップの可動範囲はグリップ中心位置がキャリブレーション位置 (SPIDAR-G のフレーム中心) から約5[cm]の距離である。SPIDAR-G グリップとVR空間中オブジェクトのスケールの関係は、SPIDAR-G グリップを5[cm]移動させたときVR空間中オブジェクトが約2.0[m]移動するように設定してある。Dual Shell Methodの外側はキャリブレーション位置から半径5[cm]の位置に張る球面であり、内側はキャリブレーション位置から半径1.5[cm]の位置に張る球面とした。

実験の順番は

1. 外部入力式クラッチ (space キー)
2. 外部入力式クラッチ
(グリップに付加されたボタン)
3. Dual Shell Method

とした。

5 結果 (1)

実験結果を図9, 図10に示す。被験者は20代の男性5人である。

図9はVR空間の中心位置から、ランダムに出現位置が選ばれたターゲットに対してドッキングタスクを行っ

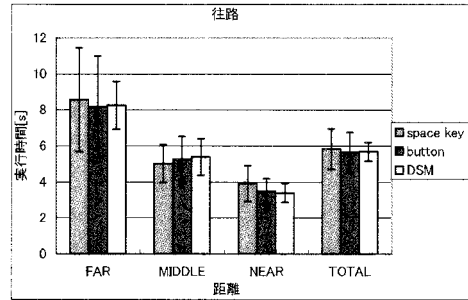


図 9 ドッキングタスク往路結果

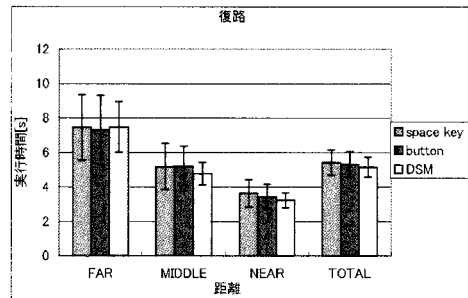


図 10 ドッキングタスク復路結果

たときの実行時間である。そして図10はVR空間の中心位置に現れたターゲットに対してドッキングタスクを行ったときの実行時間である。

図の縦軸はFAR(8[m]),MIDDLE(4[m]),NEAR(2[m]),それぞれの距離にある5つのターゲットのポインティングに要した実行時間の平均である。TOTALは全部(15体)のターゲットに対する平均の実行時間である。

実行時間に関しては差異が見られなかった。

5.1 Dual Shell Methodの分散

DSMに関しては他の方法よりも分散が小さいという傾向が見られる。他の方法は、被験者により操作方法に慣れる度合いが違うためにタスク実行時間における分散が比較的大きくなったのだと考察できる。また、DSMは被験者の習熟度によらず、より均一的な操作性を提供したと言える。よって、DSMは初めて3次元インタフェースデバイスを使用するユーザにとって有効性があると考えられる。

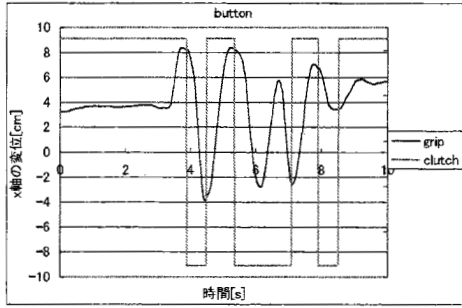


図 11 外部入力(グリップのボタン)

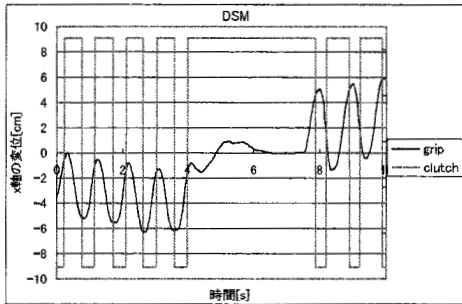


図 12 Dual Shell Method

5.2 クラッチ操作の失敗について

図 11, 図 12 は SPIDAR-G グリップの x 軸方向の時間的な変位をプロットしたものである。図 11 は外部入力(グリップに付加したボタン), 図 12 は DSM である。x 軸方向とは実験タスクにおける奥行き方向であり, そのグリップ奇跡を読み取ることでクラッチの失敗について考察する。図にはグリップ奇跡の x 軸方向成分と併に, クラッチ状態を表す矩形波も描かれている。矩形波が上辺の値を取るときにクラッチ=ON であり, 下辺の値のときはクラッチ=OFF である。外部入力(グリップに付加したボタン)ではクラッチ=OFF のときのグリップ操作が誤っていることが認められる。正しいクラッチ操作では, クラッチを切った後, クラッチを切る前と逆方向に変位させクラッチ状態を ON に戻さなくてはならない。図 11 ではクラッチ操作が間違っていることが確認される。そして図 12 の DSM ではクラッチ操作が確実に成功していることが分かる。そして, 表 3 のクラッチ失敗率を見ると DSM が他の方法よりクラッチミスが少ないのが分かる。

表. 3 クラッチ失敗率

mode	clutch error rate[%]
外部入力 (space key)	24.5
外部入力(グリップのボタン)	19.0
Dual Shell Method	5.8

5.3 クラッチ操作によるストロークについて

一回のクラッチ切り替え ON → OFF → ON の動作でポインタが進む距離をクラッチのストロークと呼ぶことにする。

DSM と外部入力式クラッチの実行時間に大差が無い理由にストロークの問題が挙げられる。オブジェクトを動かすときに DSM の方が外部入力式クラッチより約 2 倍ほど多くクラッチの切り替えをしているからである。

6 Dual Shell Method - high stroke の提案

実験 (1) では DSM の有効性を示す実験を行った。しかし, DSM にはストロークが小さいという問題が見つかった。ここでは, ストロークが小さい問題を解決する為に内側 Shell 位置を可変にすることを提案する。

デバイス中心から外側 Shell までの距離つまりは外側 Shell の半径を R とする。クラッチが切れる瞬間のグリップ位置ベクトルを \mathbf{P} ($|\mathbf{P}| = R$) とおく。内側 Shell の中心位置を表す位置ベクトルを \mathbf{Q} とすると, 比例定数 a ($a < 1$) を用いて

$$\mathbf{Q} = -a \cdot \mathbf{P} \quad (1)$$

で表せられる。この方法により 1 回のクラッチ操作で進む距離は $1 + a$ 倍大きくなる。

7 実験 (2)

Dual Shell Method - high stroke (以下 DSM-HS と略す) の評価実験を行った。ストロークの大きさを決める (1) 式の比例定数は $a = 0.7$ とした。

実験タスクについては実験 (1) と同じことを行った。比較対象としては, 外部入力式クラッチ(グリップに付加したボタン)と DSM を選び, DSM-HS を含めた 3 種類で比較実験を行った。

8 結果 (2)

被験者は 20 代の男性 5 人で実験を行った。結果は図 13, 図 14 である。図 13 は VR 空間の中心位置から, ラン

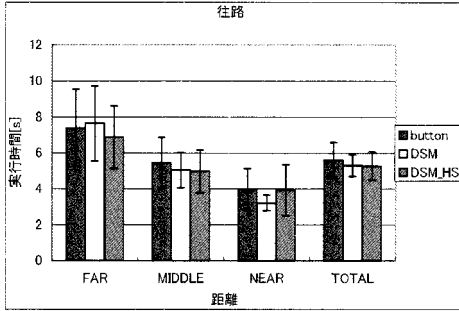


図 13 ドッキングタスク往路結果

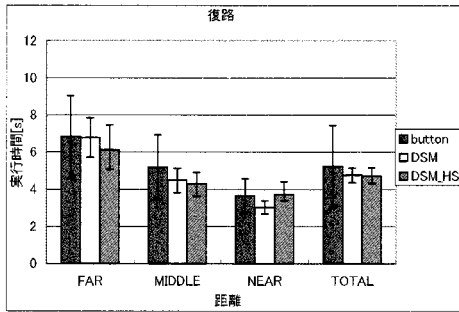


図 14 ドッキングタスク復路結果

ダムに出現位置が選ばれたターゲットに対してドッキングタスクを行ったときの実行時間(往路)である。その後、VR空間の中心位置に現れたターゲットに対してドッキングタスクを行ったときの実行時間は図14に示す。図の縦軸はFAR(8[m]),MIDDLE(4[m]),NEAR(2[m])のそれぞれの距離にある5つのターゲットのポインティングに要した実行時間の平均である。TOTALは全部(15体)のターゲットに対する平均の実行時間である。

全体の平均を見るとDSMとDSM-HSで差がないように見えるが、DSMは近距離において実行時間が比較的短くDSM-HSは遠距離においてDSMより実行時間が短くなっている。

9 まとめ

Dual Shell Methodを提案し、初心者のユーザにとって使い易い3次元インタフェースデバイスの操作方法を提案した。その後、Dual Shell Methodの一回のクラッチ操作におけるストロークが小さいという問題を解決するために、Dual Shell Method - high strokeを提案し実験により評価した。

今後はDual Shell Methodストロークの大きさの最適化を検討し、より使い易いDual Shell Methodを目指

す。また、回転におけるDual Shell Methodを実装し、6自由度の操作をDual Shell Methodで行える環境を構築する。

参考文献

- [1] Lionel Dominjon, Anatole Lecuyer, Jean-Marie Burkhardt, Guillermo Andrade-Barroso, and Simon Richir. The "bubble" technique: Interacting with large virtual environments using haptic devices with limited workspace. In *WHC '05: Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 639-640, Washington, DC, USA, 2005. IEEE Computer Society.
- [2] Shumin Zhai. User performance in relation to 3d input device design. *SIGGRAPH Comput. Graph.*, Vol. 32, No. 4, pp. 50-54, 1998.
- [3] 赤羽克仁, 長谷川唱一, 小池康晴, 佐藤誠. 10khz以上の高更新周期を実現する高解像度ハプティックコントローラの提案. 日本バーチャルリアリティ学会第8回大会論文集, pp. 7-10, 2003.