

3次元仮想空間でのロープワーク実現のための インタフェースの提案

川畑 淳也^{1,a)} 北須賀 輝明¹ 眞鍋 雄貴¹ 有次 正義¹

概要: 仮想空間中にある物体を操作する技術が提案, 実用化されている. 力を加えても形が変化しないブロックなどのいわゆる剛体を操作対象とする技術は成熟しているが, 力を加えると形が変形する柔軟物体を操作対象とする技術の研究はまだ途に就いたばかりである. 本研究では柔軟物体の一種であるロープや糸などの紐状の物体を操作対象とし, 仮想的な紐を結ぶ操作と解く操作を実現できるような3次元入力インタフェースの開発を目的とする. そのために実空間中の紐の操作を観察し, 操作を細分化する. その操作を個別に実装し, 組み合わせることで目的の達成を目指す. 3次元入力デバイスとして, 空中ジェスチャ操作デバイスである Leap Motion を使用し, 手指の3次元位置をトラッキングする. 操作対象の仮想紐は物理エンジン JBullet を使用して作成する. 評価実験では, 4人の被験者に基礎的な操作のタスクと, 結ぶ・解く操作のタスクをしてもらい, 結果として, 全てのタスクの達成とインタフェースの習熟度が高いことを示した.

1. はじめに

コンピュータの中に作られた仮想的な世界を, 五感を刺激することであたかも現実のように人々に体験させる仮想現実感 (VR:Virtual Reality) 技術が幅広く研究されている. その一つとして, 入力デバイスを用いて仮想空間上のオブジェクト (仮想オブジェクト) を操作する技術がある. 実用化されている身近な例として, 任天堂の家庭用ゲーム機である Wii における Wii リモコンやマイクロソフトの Kinect が挙げられる. どちらも手軽に仮想空間の体験やオブジェクトの操作が可能である. 研究分野においては, 舟橋らによる, 仮想手を用いたオブジェクト操作の手法 [1] や, Song らによるハンドルバーの形状のメタファを用いた仮想オブジェクト操作手法 [2] やステレオカメラを利用した操作手法 [3] などが提案されている. 仮想オブジェクトの操作は様々な場面での応用が考えられる. もちろんゲームなど娯楽としての応用も考えられるが, その他にも, 手術や高所での作業など特殊な状況のシミュレーションにも役立つと考えられる. 既存の仮想オブジェクトの操作では, 力を加えても変形しない物体である剛体が操作の対象となっている [1]. 一方, 力を加えると変形する物体である柔軟物体に関する研究はあまり多くない [4]. 柔軟物体の操作は, 剛体の操作に比べて複雑である. また, 入力デバ

イスで複雑な操作を入力することが求められる場合, 利用者の負担は大きい.

そこで本研究では, 仮想空間上での柔軟物体, 特に紐に対し, 「結ぶ」, 「解く」という操作ができるようなインタフェースを提案する. 入力デバイスには, 3次元で手指の座標を取得できるデバイスを使用する.

入力インタフェースを設計するにあたり, 予備実験として現実空間で実際の紐を結ぶ操作と解く操作を撮影, 分析することで一連の流れを解析し, 操作を個々の動作に細分化する. 各動作に対し, それらの動作を仮想空間上で実行できるように実装を行い, 組み合わせることによって現実世界の操作に近い操作で仮想空間の紐を操作できるようにした. この時, 個人差が現れた部分, 例として右手や左手の使い方の差, 紐を結ぶ際に紐を通すのに必要な輪を作る位置の違いなどの影響が小さくなるように考慮する. また, 簡略化できる部分については, インタフェースを簡略化することでより容易に操作できるようにする.

実装後の評価実験として, 4名の被験者が, 把持する等の基礎的な操作をするタスクと, 結ぶ・解く操作をするタスクを行った. 基礎的な操作のタスクでは, 提案するインタフェースを用いて, 仮想空間上で簡単なタスクが達成できるかを調べた. 一方, 結ぶ・解く操作をするタスクについては, 現実空間と仮想空間それぞれでタスクを実行し, その結果から達成度, 所要時間, 習熟度を評価した. 結果として, 全タスクの達成と, 被験者がインタフェースに習

¹ 熊本大学

Kumamoto University

^{a)} kawabata@dbms.cs.kumamoto-u.ac.jp

熟することを示した。

2. 関連研究

2.1 仮想手を用いた操作の研究

船橋ら [1] は、本研究と同様に仮想の手を利用して仮想空間中のオブジェクトを操作するという手法を実現している。この手法はセンサを取り付けたグローブを入力装置として仮想空間上の仮想手を操作することで、仮想オブジェクトを把持したり移動させたりしている。グローブ型のデバイスを用いることにより、指の動作まで含めた操作ができるようになり、細かい作業も可能になっている。

それ以前の研究と比べ、把持する指先の指定や、3以上の指での把持ができない等の制約条件を課すことなく、どの指のどの部分を使用しても剛体を操作することが可能になっている。この制約条件の緩和は、本研究の目的である現実世界と同じ動きによる操作の実現を考える上でも必要であると考えられる。この研究で扱っている仮想オブジェクトはブロックのような剛体であり、紐や糸のような柔軟物体に関しては扱っていないため、本研究とは操作対象が異なる。

2.2 仮想の柔軟物体を操作対象とした研究

五十嵐ら [4] は、仮想の柔軟物体を操作することに焦点を当て、紐の結び目や紐と紐の重なりを扱っている。五十嵐らは仮想空間内の柔軟物体を2次元的に捉え、物体相互の重なりを、マウスを利用し容易に変更できるインタフェースを提案している。例えば、畳んだ布の重なっている部分をクリックすることで重なりを前後関係を入れ替えつつ、その前後関係を満たす畳み方に変形させることができる。また、物体のドラッグ中に新たな重なりが生じた際にユーザのキーボード操作によって重なりを上下関係を指定できる。このシステムを利用して贈答品や祝儀袋によく用いられる「水引」のCGモデル構築を目標とした研究 [5] もなされている。

この一連の研究は仮想の柔軟物体を操作するという点で本研究と同じ目的であるが、これらの研究が入力デバイスとしてマウスとキーボードを使用し、単純な操作を用いることで容易に柔軟物体を操作することを目的としているのに対し、本研究では手指の3次元座標を取得できるデバイスを用いて、現実空間と同様の手指の動きで、仮想空間中の柔軟物体を操作することを目的としている、つまり、使用するデバイスおよび、現実空間での操作の再現を目指しているか否かという点が異なる。

2.3 実空間の柔軟物体操作に関する研究

山川ら [6] や Saha ら [7] は、ロボットアームを利用して現実空間の紐を結ぶ事を目的としている。山川ら [6] は、本研究と同様に実際に人が紐を結ぶ操作を解析し、輪の作

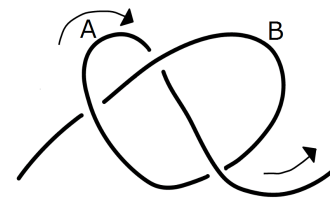


図1 ひとむすび

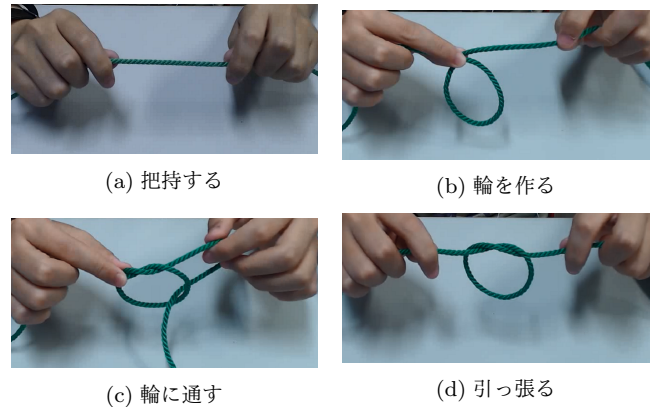


図2 結ぶ操作の流れ

成、紐の入れ替え、紐の引き抜きの3つに分割し、これらを基本的なスキルとしている。それらの基本的なスキル一つひとつを1本の多指ロボットアームに実装し、スキルどうしを統合させることで結ぶ操作を実現している。また、触覚センサ等を用いることで把持力も制御している。

Saha ら [7] は、2本のロボットアームと他の静的なオブジェクトをいくつか使用し、結ぶ操作を実現している。この研究でも人が結ぶ操作を参考にして同じ流れで結ぶ操作を再現している。

これらの研究はロボットアームにより実空間の柔軟物体である紐の操作をすることを目的としており、用途が遠隔操作などであるのに対して、本研究では仮想空間でのシミュレーションを想定しているので、扱う空間が異なる。

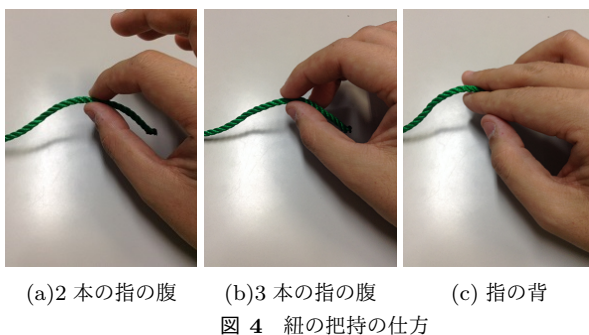
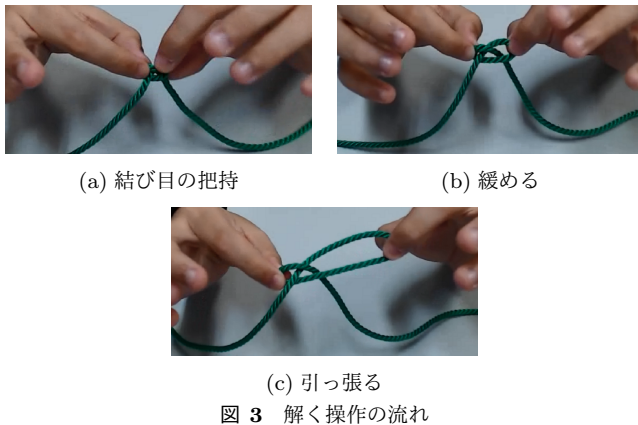
3. 操作解析

本章では、紐を結ぶ操作と解く操作の一連の流れの解析について述べる。6人の被験者(20代の男性5人、40代の男性1人、全員右利き)に現実空間上で、「ひとむすび」(図1)[8]を結ぶ操作と解く操作をしてもらい解析する。

3.1 結ぶ操作と解く操作の流れ

結ぶ操作はまず、結び目を作ろうとしている点をさむように左右の手指2点で紐を把持する(図2(a))。その後、把持した部分を動かしながら輪を作る(図2(b))。次に作った輪の中に紐の一方を通し(図2(c))、紐の両端を引っ張る(図2(d))という流れで構成される。

解く操作は図1の点A、Bをそれぞれ左右の手指2点で把持し(図3(a))、把持した部分を動かしながら結び目を緩



める(図3(b)).そのまま把持する位置をスライドさせながら紐を引っ張る(図3(c))という流れで構成される。

次節から、結ぶ操作と解く操作において見られた個人差や特徴的な動作について解析する。

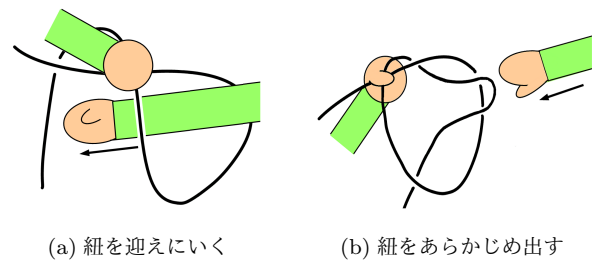
3.2 結ぶ操作

(a) 把持する操作

把持する際に使用する指の本数は2~5本で、操作中にも本数を変えながら把持していた。把持する部分としては、図4(a), (b)のように、指の腹で紐を挟み込んで把持する被験者が多かった(6人中5人)が、(c)のように指の背と腹で挟み込んで把持する被験者もいた。把持する操作の中で被験者に共通していたのは、人差し指を必ず使用しているという点であり、薬指や小指を使う場合は、親指、人差し指、中指の補助として使われていた。これらの観察から、把持する操作は人差し指と親指、もしくは人差し指と中指ではさむ操作と定義する。

(b) 輪を作る操作

輪を作る際には、紐を右手と左手の2点で把持しながら紐を重ね合わせる操作をする。重ねる場所は、被験者全員が把持している2点間であり、重ね合わせ方は、大きく分けて2つのパターンが見られた。1つは把持していない部分を重ねるパターンで、もう1つは把持している部分をそのまま重ねるパターンである。いずれの場合も紐どうしを近づけている操作なので、輪を作る操作は、紐を2点で把持しながら、把持して



いる点の間で紐を重ね合わせる操作と定義することができる。

輪を作った後は被験者全員が重なる部分を把持していたが、把持する際に6人中4人が非利き手で把持し、2人は利き手で把持していた。しかし、どの被験者についても、輪の重なり部分を把持している方の手は指を少し動かすのみで、手全体を動かして次の操作に移ることはなかったため、あくまでも重なり部分の把持は、固定する補助的な操作である。そのため重なりを把持する操作は重なり部分を固定することで対応する。

(c) 輪に紐を通す操作

紐の通し方も大きく分けて2つのパターンが見られた。1つは輪を把持していないほうの手で、輪の中に手を入れて紐を引っ張り出すパターンである(図5(a))。このパターンでは、紐を通すスペースを確保するために全ての指を使用して輪を広げるといった操作をする被験者も見られた。もう1つは、輪を把持している方の手指で紐をあらかじめ輪の外に少しだけ出しておき、把持していない方の手で引っ張るパターンである(図5(b))。輪の外に紐を出す方法は、指で紐を押し出すようにしていた。押し出す際に使用する手は、輪を把持している方の手を使用する被験者と把持していないほうの手を使用する被験者の両方がいた。輪に通す紐が長い場合は前者、短い場合は後者が見られた。このように2つのパターンが見られたが、どちらも輪を把持していないほうの手で紐を把持し引っ張る操作という部分は共通している。つまり輪の中に紐を通す操作は、一方の手で輪を把持しながら他方の手で紐を把持し引っ張る操作と定義できる。同様に、紐の重なりを把持しているほうの手で開き具合に連動した輪を広げる操作、重なりを把持しているほうの手では紐を押し出す操作が必要になる。

(d) 引っ張る操作

紐を引っ張る際に、被験者全員が輪に紐を通した位置をはさむようにして両手2点で把持し、左右の手を反対方向に引っ張る操作をした。つまりこの操作は、上述した把持する操作が可能であれば実現可能である。

3.3 解く操作

(a) 結び目を把持する操作

結び目を把持する操作は、3.2節で定義した操作と同じ操作で実現できる。

(b) 緩める操作

結び目を緩めるときには、図1の紐の結び目の点A, Bを両手2点でそれぞれ把持し、逆方向に動かすことで緩めている。この際には少し引っ張る動作と、把持する位置を変えるという動作を交互にやっている。このような大まかな流れは被験者間で違いは見られなかったが、緩め方には2つのパターンが見られた。1つは左右の手を両方動かすパターン。もう1つはどちらかの手(主に非利き手)を固定したまま、片方の手のみを動かして緩めるパターンである。いずれのパターンでも、把持しながら紐を動かす操作と把持しながら紐を動かさない操作の2つができればよく、これらは3.2節の把持の操作と同様である。

(c) 引っ張る操作

次に紐を引っ張る操作を解析する。緩める操作をする際の紐を引っ張る操作は結ぶ操作の把持の操作とは違う特徴が見られる。結ぶ際には把持する場所を変えることはほとんどなく、一つの点を把持したまま紐の移動や引っ張る操作をしていた。把持する場所を変える場合には一旦把持の操作をやめて、紐から指をはずすことで把持する場所を変えていた。解く際にも結ぶ操作と同様の操作もするが、解く際に特有の操作として紐から指をはずさずに、はさむ強さを調整することで把持する場所をずらし、紐の上の指先を滑らせて移動させるような操作が見られた。この指先を滑らせて紐を引っ張る操作は、把持と把持の解除を連続的にする操作と定義する。

4. 提案インタフェース

本章では3章での操作解析を基に、本研究で提案するインタフェースについて述べる。本研究で提案するインタフェースは、3次元で手指の座標が取得できるデバイスと、オブジェクトの表示および質量や速度、摩擦などといった物理演算が可能な物理エンジンを使うと仮定する。これらにより、現実空間と同じ物理特性を持った仮想紐とそれを操作する仮想手が生成できるものとする。

3章で解析した操作は全部で7つ(図6左)に細分化された。それらを実現するためには図6右の5つの操作、つまり紐を把持する、輪を作る、紐を押す、輪を広げる、紐を引っ張る操作が可能になればよい。また、これら5つの操作は表1の3つの機能によって構成されるため、これらの機能を実現する手法を提案する。

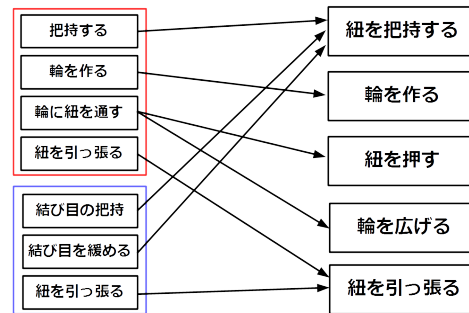


図6 細分化した操作(左)とグループ分けした操作(右)

表1 操作に必要な機能

	距離の算出	移動	接続
紐を把持する	○	○	○
輪を作る	○	○	○
紐を押す	○		○
輪を広げる	○	○	
紐を引っ張る	○	○	○

4.1 必要機能

まずは距離の算出について述べる。左右の手の座標や仮想空間上に存在するオブジェクトの座標をそれぞれ取得できるとした場合、距離を算出する2点の座標をそれぞれ (x_1, y_1, z_1) , (x_2, y_2, z_2) とすると、2点間の距離は式(1)で算出することができる。

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (1)$$

指先どうしの距離、指先と仮想紐の距離、仮想紐の近づいた部分の距離は全て式(1)を使用して算出する。

次にオブジェクトの移動について述べる。オブジェクトはフレームごとに生成と削除を繰り返すのではなく、一度オブジェクトを生成したら移動させる。移動方法は仮想手とその他のオブジェクト(仮想紐など)によって異なる。物理エンジンにより、仮想手以外のオブジェクトはその質量などに応じて重力などの物理法則に従い移動する。一方、仮想手は入力デバイスから取得した座標に表示する。仮想手の移動量はベクトルである。現フレームの入力デバイスから取得する手の座標 (x_n, y_n, z_n) と1フレーム前の仮想空間上で仮想手を構成するオブジェクトの座標 (x_p, y_p, z_p) から移動ベクトル $(x_n - x_p, y_n - y_p, z_n - z_p)$ を算出する。算出した移動ベクトルを1フレーム前の座標に加えることで仮想手を移動する。

最後に接続について述べる。接続とは仮想空間に存在しているオブジェクトどうしをつなぎ合わせ、同様の動きをするようにすることである。輪を作ったときの紐の交点で仮想紐の近づいた部分を接続する。この場合には、接続する点どうしを移動させるときに同じ座標上を動かすことで直接接続しているように見せる。一方仮想紐と仮想手の接

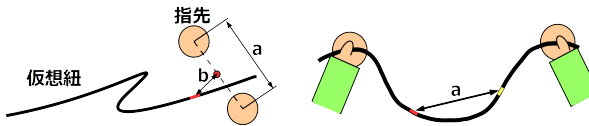


図 7 把持する操作 (提案手法)

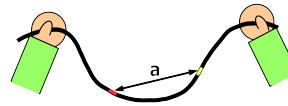


図 8 紐間の距離

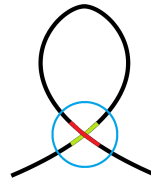


図 9 重なり前後の塗りわけ

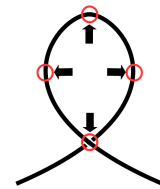


図 10 輪の広げ方

続をする際には、仮想手と直接接続するのではなく適切な座標を設定し、その場所に接続する。接続する際の適切な座標の設定については後述する。

4.2 各操作への適用

本節では 4.1 節で提案した機能を使い、解析した操作を実現する手法について述べる。本研究では、操作対象である仮想紐を短い棒状の剛体の集合として扱う。

把持する操作

把持する操作の概要図を図 7 に示す。把持する操作は、仮想紐の近くで、人差し指と人差し指に隣接する指を近づける操作とする。人差し指と人差し指に隣接する 2 つの指の間の距離を求め、この距離が閾値 A 以下のときにはさむ操作として検出する。はさむ操作を検出した際に、仮想紐をはさんでいる指先の midpoint から仮想紐の最も近い部分までの距離を求め、この距離が閾値 B 以下のときに接触状態として検出する。2 つの状態を同時に検出した場合に、指で紐をはさんだのみなし、はさむ操作をしている指先の midpoint 座標に仮想紐中の最も近い部分を接続し、把持の操作を実現する。把持する操作で指先の midpoint 座標と仮想紐を接続することで、紐をはさんでいる間は常に 2 つの指先の間に紐が存在するようになる。また両手についてそれぞれ把持の操作をしている間は把持フラグを立てる。

輪を作る操作

輪を作る操作は上述した把持フラグが両方立っている場合にのみ実行可能な操作とする。把持している 2 点の間にある、最も近くにある剛体 2 点の距離 (図 8 の a) を求め、この距離が閾値 C 以下になると剛体どうしが接触している状態として検出する。接触状態を検出したら 2 つの剛体を直接接続する。接続を解除するのは、仮想紐を引っ張る操作をした時とする。また、重なり部分について、仮想空間では前後関係を認識しにくいので、図 9 のように接続した部分の周囲を塗り分ける。

紐を押す操作

紐を押す操作は仮想手の指先と仮想紐の最も近い部分との距離を求め、この距離が閾値 D を下回ったときに接触状態として検出する。検出している間は、仮想手の移動ベクトルと同じベクトル分だけ仮想紐の接触

部分を動かすことで押す操作を実現する。押す操作によって移動する距離は把持して動かす距離に比べて小さいと考えられるので、指と紐は接続しない。

輪を広げる操作

輪を広げる操作は、親指から小指までの隣接する指先の距離 5 つを求め、これら 5 つの距離全てが閾値 E を超えたときに手の開きとして検出する。手の開きを検出したら、輪の部分から図 10 のような 4 等分した点を選び、それぞれ輪を広げるような方向に座標を移動させる。これにより、手を開いたら輪も同時に開くようにする。この時、輪の重なりを固定した部分は維持したままである。

紐を引っ張る操作

通常時の把持場所を変えずに紐を引っ張る操作は、把持する操作で実現可能である。ここで述べる紐を引っ張る操作とは、解く操作時に見られる把持場所をずらしながら紐を引っ張る操作のことである。ずらす操作は把持の操作で設定した閾値 A より小さい値の閾値 F を用いる。実際の距離が閾値 F から閾値 A の間では接続と解除をフレームごとに繰り返す。つまり仮想紐をつまんでいる指先の間の距離が、閾値 F より小さい時は把持位置をずらさずに引っ張る操作が、閾値 A と閾値 F の間の時はずらしながら引っ張る操作が可能になる。また、把持位置をずらさずに引っ張る操作をする時には、同時に輪の重なり固定も解除するが、ずらしながら引っ張る操作をする時は重なり固定は解除しない。

5. 実装

本章では提案インタフェースの実装について述べる。本研究では物理エンジンとして JBullet を、3 次元の手指の座標を取得するデバイスとして Leap Motion をそれぞれ使用する。JBullet は、剛体や柔軟物体に対して衝突判定演算を行うオープンソースライブラリである Bullet Physics を Java 環境に移植したものである。ただし、最新版の Bullet Physics では柔軟物体の生成や衝突判定は可能であるが、本研究で使用するバージョンは 20101010-1 で Bullet Physics の 2.72 に相当するため、柔軟物体の生成や衝突判定はできない。Leap Motion は Leap Motion 社によって開発された空中ジェスチャ操作デバイスであり、指先や関節の位置

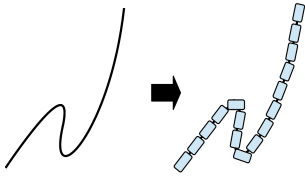


図 11 仮想紐の構成

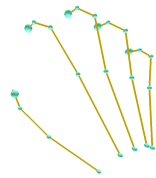


図 12 仮想右手

の3次元座標の取得や、指や手の種類の判断、つまみの強さなど様々なデータを取得することが可能である。本研究で使用する Leap Motion 用の SDK は Leap DeveloperKit であり、バージョンは 2.1.0+18736 である。

5.1 仮想紐

仮想空間上に生成する仮想紐の実装方法について述べる。本研究で操作する仮想紐は、4.2 節で述べたように複数の剛体の集合体として定義する。具体的には、複数の小さい剛体を用意しそれらをつなぎ合わせることで仮想紐を作成する(図 11)。このように定義する理由としては、JBullet が剛体のみを生成できる仕様であることと、仮想手に最も近い仮想紐の部分を探す場合などに、「剛体 = 要素」と考えることができ、接続の操作等がやりやすいためである。接続点には減衰 (damping) や曲げる角度、要素どうしの線形距離の上限・下限等が設定でき、要素の長さも変更可能であるため、さまざまな種類の紐や糸が表現できる。

5.2 仮想手

本研究では、仮想空間上の仮想紐等のオブジェクトを操作するために仮想手を作成する。仮想手を作成してオブジェクトを操作することで、現実世界に近い動きで目的を達成できる。仮想手を作成するには、3次元で手の座標を取得する必要がある。仮想手を作成する際にも、仮想紐と同じように JBullet 中の剛体を利用する。Leap Motion から取得した指先の位置と関節の位置にそれぞれ剛体を配置し、手の動きに合わせて仮想空間上の仮想手を移動する。作成した仮想手のうち右手を図 12 に示す。

5.3 必要機能

まず距離の算出の実装について述べる。JBullet には仮想空間に生成したオブジェクトの中心の座標を取得するメソッドがある。このメソッドで仮想手と仮想紐を形成しているオブジェクトの座標を取得し、距離を算出する。

次に移動の実装について述べる。Leap Motion には、両手の親指から小指の全ての指の指先と関節の座標を取得するメソッドがある。このメソッドとともに、仮想オブジェクトの中心座標を使用して、移動の機能を実装する。

次に、接続の実装について述べる。接続は JBullet 内にある接続用のメソッドを使用して 2 つ以上のオブジェクト

を接続する。このメソッドにはオブジェクトの接続位置をオブジェクトの中心からの相対座標で与えるので、接続するオブジェクトの種類によって個別に相対座標を与えるように実装する。

最後に、4.2 節で定義した閾値として次の値を用いる。紐をはさむ際に使用する閾値 A は Leap Motion SDK のつまみの強さの値で 0.7 とする。つまみの強さは 0 から 1 までの値をとり、pinchStrength() で得られる。接続用のオブジェクトから仮想紐までの距離に使う閾値 B は 20mm に設定する。輪を作るときの仮想紐間の距離に使う閾値 C は 9.5mm、紐を押すときに使用する指先と仮想紐までの距離に使用する閾値 D は 1cm に設定する。手の開きを検出する際に使用する閾値 E は Leap Motion SDK の手のひらの握りの強さで 0 とする。握りの強さは 0 から 1 までの値をとり、grabStrength() で得られる。閾値 A より小さい値である閾値 F は pinchStrength() の値で 0.3 に設定する。

6. 評価

6.1 評価方法

被験者には現実空間と仮想空間で以下のタスクをしてもらい、それぞれの達成度、かかった時間、習熟度について比較する。評価に使用する結び方は図 1 のひとつむすびである。またタスク終了後には、被験者にインタビューし、使い勝手や現実空間との操作の違いなどを聞き取り主観評価する。タスクは「基礎的な操作のタスク」と「結ぶ・解く操作のタスク」の 2 つに大きく分ける。それぞれのグループのタスクを以下に示す。

- 基礎的な操作のタスク
 - 仮想紐を両手で把持 (図 13(a))
 - 仮想紐で輪を作る (図 13(b))
 - 把持位置をずらす (図 13(a))
 - 仮想紐を穴に通す (図 13(d))
- 結ぶ・解く操作のタスク

- [タスク 1] 紐に結び目をつける (図 14(a))
- [タスク 2] 紐を固定ブロックに結ぶ (図 14(b))
- [タスク 3] 紐を動くブロックに結ぶ (図 14(c))
- [タスク 4] 紐の結び目を解く (図 14(a))
- [タスク 5] 紐を固定ブロックから解く (図 14(b))
- [タスク 6] 紐を動くブロックから解く (図 14(c))

また、基礎的な操作のタスクに関しては、達成の有無のみを記録し、かかった時間などは記録しない。結ぶ・解く操作のタスクについては、タスク 1 からタスク 6 の順に 1 回ずつ行うことを 1 試行とする。3 回試行することで習熟度を見る。

被験者は 20 代の男性 4 人 (いずれも右利き) である。被験者をそれぞれ A,B,C,D とする。実験をする前に、被験者には本研究で提案するインタフェースの機能や使い方について口頭で説明した。その後、第一著者が実演すること

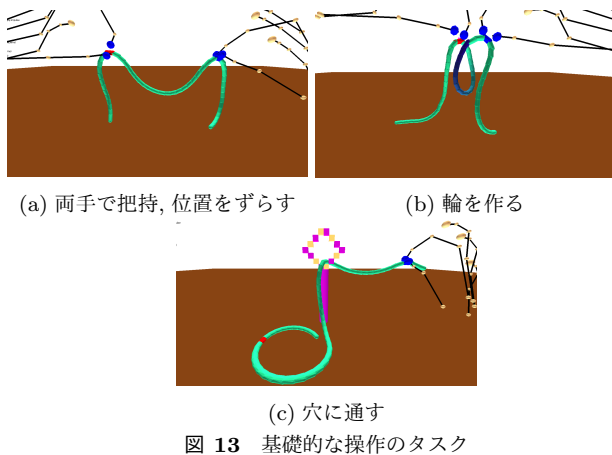


図 13 基礎的な操作のタスク

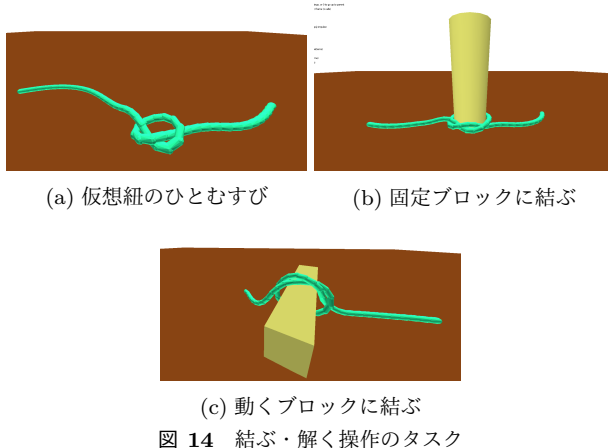


図 14 結ぶ・解く操作のタスク

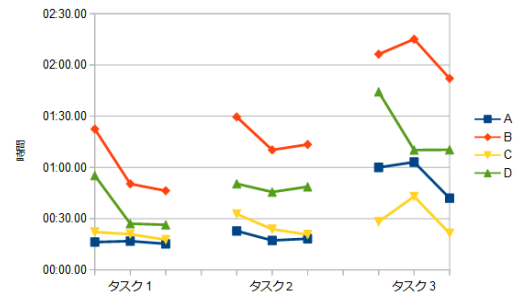
で動きを確認した。説明後に被験者にインタフェースを使用してもらい、基礎的な操作のタスクを練習する時間とした。練習の時間は被験者に任せ、5分から10分程度であった。評価実験は練習の時間が終了次第開始した。

6.2 評価結果

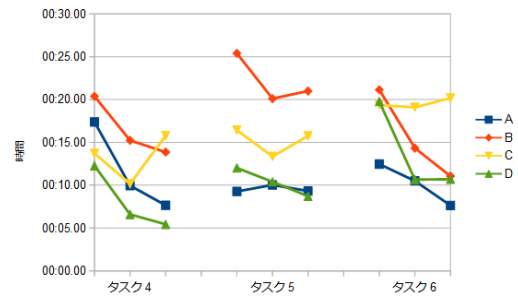
まず、基礎的な操作のタスクについて被験者全員が達成できた。被験者を観察したところ、把持位置をずらす操作に少し時間がかかっていたが、何度もやり直さないといけないというような被験者はいなかった。問題なく操作できるという被験者が多かった。

次に、結ぶ・解く操作のタスクについて結果を述べる。仮想空間での試行回数と時間の関係のグラフを図15に示す。

4名の被験者共に、6つのタスクを達成できた。図15のグラフにタスクにかかった時間を示す。タスクごとに左から1回目の試行、2回目の試行、3回目の試行を表している。図15を見ると、結ぶ操作のタスクは15秒から130秒程度の時間で達成することができ、解く操作のタスクは5秒から25秒ほどの時間で達成できていることがわかる。また、試行回数が増え操作に慣れていくにつれて、結ぶ操作では15秒から120秒程度で、解く操作では5秒から20秒程度で全員がタスクを達成できるようになり、ほとんどの被験者で試行回数が増えるとタスクにかかる時間が短



(a) 結ぶ操作の結果



(b) 解く操作の結果

図 15 試行回数と時間の関係のグラフ

くなった。この事から被験者が、少ない使用回数でインタフェースに習熟したことがわかる。

次に、(a)と(b)を比較すると、仮想空間では解く操作よりも結ぶ操作に時間がかかる被験者が多い。現実空間では2つの操作の差が、最大6秒程度、平均でも2秒程度しかないのに対し、仮想空間では最大110秒程度、平均では35秒程度の差が生じる。これらより、現実空間に比べて仮想空間では、紐が扱いにくいことがわかる。

次に、解く操作に関しては、タスクの内容が変わってもかかる時間は最大でも6秒程度しか変わらないが、結ぶ操作はタスクの内容が変わると最大60秒近く変化する。さらに、タスク3では被験者間で最大120秒ほどの差が生じている。これらはタスク3で使用した動く仮想オブジェクトの制御が不十分であり、仮想紐や仮想手と衝突した時に挙動が大きくなったことが理由であると考えられる。

最後に、被験者へのインタビューで得られた肯定的意見として、操作説明を聞くとわかりやすいという声や慣れたら使いやすいという意見があった。一方、否定的な意見として、奥行きが認知しづらい時があるという声や、仮想手が振動したり思った動きをしないためにストレスが溜まったという意見があった。これらについて6.3節で考察する。

6.3 考察

6.2節で述べたように、本研究で提案したインタフェースを使用し、全被験者が全てのタスクを達成することができた。しかし、被験者間ではタスクの達成時間に差が見ら

れた。タスクの達成時間に差が出た理由として、本研究で提案した手法に起因するものと、使用したデバイスの性能に起因するもの大きく2つがある。

提案した手法に起因するものとしては、紐を把持する手法が影響していると考えられる。紐の把持を検出する際に、つまむ操作の検出と仮想紐と指先間の距離を使用している。これは図13(a)のように紐が一本だけで重なっていないときには問題ないが、輪を作った時のように紐が重なっているときに、把持したい場所が思うように把持できない被験者が多かった。その結果、把持するつもりのない紐を把持することになり、紐を移動させることが難しくなる。今後の課題として、複数の仮想紐が重なった場合に目標である部分を把持できるように、閾値の設定などを見直すことが挙げられる。

次に、デバイスに起因するものとして、Leap Motionで手の座標を取得する際に、取得しにくい手の状況が存在する。例えば、手のひらをLeap Motionに対して平行にして指を動かした場合や、つまむ動作をする際に指どうしを近づけすぎた場合である。これらの場合に、正しい座標が取れなくなる現象が見られた。手の座標が取れないと、把持している途中で仮想手が急に移動する、あるいは消滅するため、操作を一からやり直すこととなり、タスクの達成に時間がかかる。解決策としては、2章で紹介した、舟橋ら[1]のグローブ型のデバイスを使用すること、あるいは複数のLeap Motionを用いることなどが考えられる。グローブ型のデバイスは手に装着して座標データ等を取得するため煩雑さは増えるものの、Leap Motionで取得しにくい角度での手指の座標も取得することができるようになり、誤作動が減ると考えられる。

7. まとめ

本研究では仮想空間上で現実空間と変わらない操作でロープワークができるようなインタフェースを提案した。インタフェース提案のために、実空間でのロープワークを解析することで、結ぶ操作を4つ、解く操作を3つの動作に細分化した。さらにこれらの動作を5つの基本的な動作に分類し、それらを実現するために必要な3つの機能を抽出した。それらの操作を実装し評価実験を実施したところ、被験者4名全員がひとむすびを結ぶタスクと解くタスクを達成できた。また評価から、提案したインタフェースを使用する回数が増えるごとに、習熟度が上がることを示した。その反面、被験者間でタスクにかかる時間に差が見られるなどの課題も確認された。

今後の課題としては、本研究で使用したひとむすび以外の結び方に対するアプローチと、オブジェクトからの力をフィードバックできるような装置との統合が挙げられる。提案したインタフェースの実用化に向けては、ひとむすび以外にも、例えば靴紐を結ぶ際の蝶結びなど多種多様な結

び方に対応する必要がある。結び方が増えるたびに操作解析をして実装をするようなインタフェースではなく、2本以上の紐の把持や、複数の輪を同時に作成するなど共通性が見られる操作を備え、それらを組み合わせることで様々な結び方に対応できるようなインタフェースの提案が求められる。また、Toyouraら[9]やMinamizawaら[10]などが研究している、手に力覚をフィードバックするデバイスと統合し、仮想紐の質量や弾性力を実空間の手にフィードバックできるようにしていきたい。このようなフィードバックにより、仮想空間中にある仮想紐を実空間中で結んでいるような体験ができ、より現実に近い仮想体験が可能になる。

参考文献

- [1] 舟橋健司, 安田考美, 横井茂樹, 鳥脇純一郎, “3次元仮想空間における仮想手による物体操作モデルと一実現法,” 電子情報通信学会論文誌 D-II, Vol.J81-D-II, No.5, pp.822-831, 1998.
- [2] P. Song, W. B. Goh, W. Hutama, C.-W. Fu, and X. Liu, “A Handle Bar Metaphor for Virtual Object Manipulation with Mid-Air Interaction,” Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12), pp.1297-1306, 2012.
- [3] P. Song, H. Yu, and S. Winkler, “Vision-based 3D finger interactions for mixed reality games with physics simulation,” Proc. of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry (VRCAI'08), Article No.7, 2008.
- [4] T. Igarashi and J. Mitani, “Apparent Layer Operations for the Manipulation of Deformable Objects,” ACM Transactions on Graphics, Volume 29, Issue 4, Article No.110, SIGGRAPH 2010, July 2010.
- [5] 三谷 純, 五十嵐 健夫, “Apparent Layer Operations を活用した結び目と水引のCGモデル構築,” 日本図学会秋期大会, pp.145-148, 2011.
- [6] Y. Yamakawa, A. Namiki, M. Ishikawa, and M. Shimojo, “One-handed Knotting of a Linear Flexible Object based on Reconfigurable Skill Synthesis Strategy,” Proc. of ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots (ReMAR 2009), pp.486-493, 2009.
- [7] M. Saha and P. Isto, “Manipulation Planning for Deformable Linear Objects,” IEEE Transactions on Robotics, Vol.23, No.6, pp.1141-1150, December 2007.
- [8] 羽根田治, 図解 ひも&ロープの結び方日本文芸社 2013.
- [9] M. Toyoura, T. Shono, and X. Mao, “BioMetal glove,” Proc. of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST'10), pp.135-138, 2010.
- [10] K. Minamizawa, S. Fukamachi, H. Kajimoto, N. Kawakami, and S. Tachi, “Gravity grabber: wearable haptic display to present virtual mass sensation,” Proc. of ACM SIGGRAPH 2007, Article No.8, 2007.