情報処理北海道シンポジウム 2023

VR コントローラに握力検出機能を拡張するセンサの開発

岡 秀哉　湯村 翼
(北海道情報大学)†

1 はじめに

頭部に装着し，没入するデバイスとして，VRヘッドセットが存在する．VRヘッドセットは通常，頭部に装着するヘッドマウントディスプレイ(head-mounted display:HMD)とVRコントローラの2点で構成され，VRコントローラは手に持って操作を行う．VRコントローラにはボタンを押すことや位置トラッキングを行う機能が搭載されている．しかし，握力を利用して操作する機能は一般的には搭載されていない．握力をコントローラの操作手段に加えることができれば，VRでの新たな表現に繋がる可能性がある．先行研究として，圧力センサまたは握力を使用したVRコントローラは存在する [1] [2] [3] [4]．しかし触覚フィードバックを目的として開発されたものや，親指と人差し指でつまむ動作の圧力を利用したものである．そのためそれらは，コントローラ把持時に加える握力を使用して，反映することを目的としたコントローラではない．また，Dexcontroller [5]は握力，手の形を利用したVRコントローラである．把持時の動作により，テストアプリの挙動が変化する．しかし，Dexcontrollerは，既存のVRコントローラを拡張する設計にはなっていない．既存のVRコントローラに握力検知の機能を拡張できれば，あらゆる既存のVRデバイスにおいて握力による操作が可能となる．そこで本研究では，既存のVRコントローラの持ち手に巻き付けるだけで握力による操作が可能となるベルト状センサ，GRAB(Grip Recognizable Attachment Belt)を開発する．

2 GRAB

2.1 設計

GRABは，既存のVRコントローラの持ち手に巻き付けるだけで握力による操作が可能となる．GRABはVRコントローラとともにHMDにデータを送信する(図 1 構成図)．本研究では，巻き付ける対象のVRコントローラに，Meta Quest 2コントローラを使用する(図 2 Meta Quest 2 コントローラ図 2 左)．また，GRAB(図 2 右)には圧力を感知するセンサであるVelostat [6]を使用する．Velostatは薄く曲げ伸ばししやすいため，巻き付けることに適している．

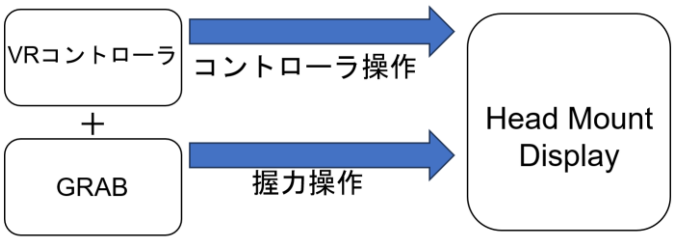

図 1 構成図

図 2 Meta Quest 2 コントローラ(左)と GRAB(右)

GRABの制御にはM5Stack [7]を用いる．M5Stackは制御と同時にMeta Quest 2のHMDに握力情報を送信する．その際，Meta Quest 2に握力情報を送信する方法は無線通信と有線通信の2種類がある．送信方法として，有線でMeta Quest 2と接続した場合，コントローラを振り回したり，頭部を動かす際に，操作を阻害したり，体勢によっては事故が生じる可能性がある．実際に，VRヘッドセットを使用した際の事故も多い [8] [9]．対して無線通信では，有線接続に用いるM5StackとMeta Quest 2間の通信ケーブルは存在しない．そのため，GRABを使用するユーザは物理的な干渉を受けず，有線接続の際に起きうる事故を防止することができる．加えて，本研究では巻き付けるだけで使用可能な握力対応コントローラの開発を目指していることから，ユーザのセットアップの手間を削減したい．そのため，本研究は無線での通信で開発を行う．通信方法は，M5Stackが行うことのできるBluetooth Serialを使用する．

GRABでの圧力検知に用いるVelostatは，圧力を加える

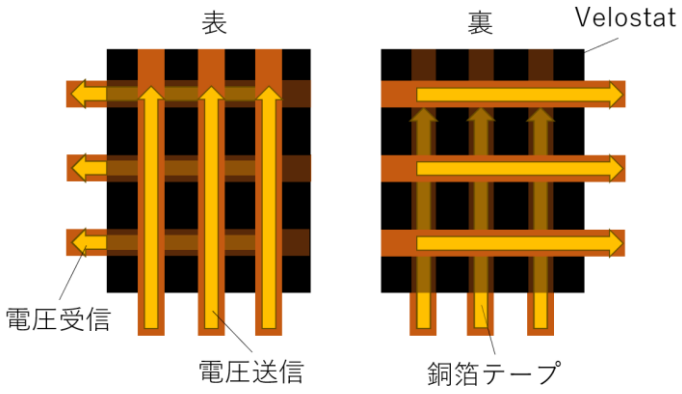

図 3 GRAB の構造

と電気抵抗が低下する特性を持つ．この特性を使用して，表面から電圧を送信し，裏面で電圧を受信することで，電圧の変化を握力情報とする(図 3)．その握力情報を取得する地点を作成するため，銅箔テープを格子状に配置し，その間にVelostatを挟み込む．

続いて，挟んだ銅箔テープを電圧送信側と電圧受信側の，2点でM5Stackと接続する．これで電気の送受信がM5Stackによって制御可能な状態となる（図 3)．テープが重なる地点に対して圧力が加われば，電圧に変化が生じる．そのため，テープが重なる地点を圧力感知地点とする．Meta Quest 2コントローラにGRABを巻き付けて，コントローラの使用者が握ることにより電気抵抗が低下する．圧力感知地点からM5Stackが受信した電気を握力として，VRテストアプリに反映する．M5Stackが受信した電圧は数値として0から4095の値に変換される．この数値をVRテストアプリに送信し，握力として扱う．

2.2 プロトタイプ実装

GRABの動作を確認するため，3点の握力検知が可能なプロトタイプを実装した(図 4)．プロトタイプでは，プログラミング言語Processingを用いて握力の可視化機能を実装した．GRABの圧力感知地点に加えられている握力が，円の大きさによって表現される．GRABが検知した圧力の数値を円の直径として利用し，データの受信ごとに円の拡大と縮小を繰り返す．プロトタイプでは，GRABとPCは有線でシリアル通信を行っているが，HMDへ送信する場合にはBluetooth Serialを用いる．また，GRABを把持した際，手に付着した汗などが誤動作を発生させるが，絶縁体である紙で覆うことにより誤動作を防いだ．

プロトタイプ実装により，GRABで握力が検知されることを確認した．絶縁体として用いた紙は耐久性に劣るため，今後，紙ではなくプラスチックやゴムなどのシート型の絶縁体でGRABを覆い，手とGRABが接しないように設計する必要がある．

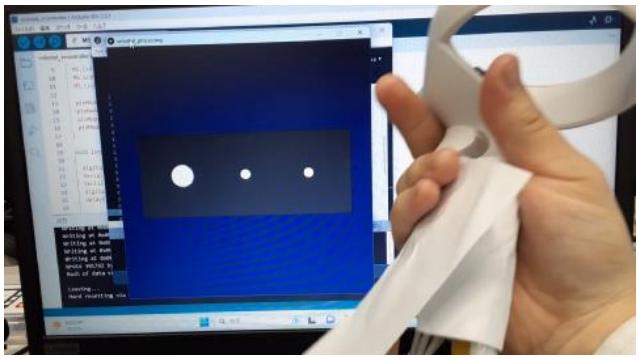

図 4 GRAB プロトタイプ

3 VR テストアプリ

3.1 概要

本研究はVRデバイスで使用することを目的としており，GRAB把持時の握力を反映するVRテストアプリをUnityで実装する．VRテストアプリでは，オブジェクトを握り潰すような動作や，オブジェクト破壊後のエフェクトが力加減により変化する仕組みの実装を予定している．

テストアプリにおける重要な機能は，握力のキャリブレーションである．握力には個人差があり，計測した数値を画一的に扱うだけでは意図通りの操作はできない．キャリブレーションを行うため，限界まで握る動作と力を加えず把持する動作をそれぞれ数秒間促すシーンを用意する．その際の値をそれぞれ上限値および下限値に設定し，GRABでの操作の範囲に利用する．

3.2 Bluetooth プラグインの実装

VR テストアプリの実装には Unity を用いる．本研究で対象とする VR デバイス Meta Quest 2 は Android OS が稼働し，Unity でアプリを実装する際には apk ファイルとしてビルドする．apk ファイルとは，Android application package の略称であり，主に Android OS 上でインストール時に使用するファイル形式である [10]．Android OS 上では外部との通信には Bluetooth の使用が想定される．GRAB で用いるマイコン M5Stack も，Bluetooth Serial を用いて Meta Quest 2 と通信する．

Unity がビルドする apk ファイルは，初期設定では Bluetooth を使用できない．そのため，Bluetooth を使用するためのプラグインを作成する必要がある．プラグイン

とは，この場合 Unity の外で作成したコードのことであり，Unity で本来使用できない機能を使用するものである [11]．Bluetooth プラグインは，Android Studio を用いて aar ファイルとして実装を行う．aar ファイルとは，Android ARchive の略称であり，アプリのビルドに必要なソースコード，リソースファイル，マニフェストが含まれたものである [12]．Android Studio で Unity のクラスを使用可能にするため，Unity に付属している classes.jar ファイルを使用する [13]．classes.jar ファイルには Unity で使用するクラスが含まれており，これを利用して aar ファイルで Unity クラスを使用可能にする．しかし，classes.jar ファイルを使用して Android Studio により aar ファイルをビルドする際は，コンパイル時のみに classes.jar ファイルを使用するよう設定する必要がある．理由としては，Unity で apk ファイルをビルドする際にエラーが発生するためである．原因は元々Unity 側に存在する classes.jar ファイルと，Android Studio でビルドした aar ファイルとしてのプラグインに含まれる classes.jar ファイルのクラス等が共に重複し，エラーが発生し，Unity でのビルドが不可能になる．

本研究では，Bluetooth プラグインの実装は未完了である．Unity プラグインを aar ファイルとして実装できることを確認しており,今後 Bluetooth の受信機能を追加する.Bluetooth プラグインの実装後,これを用いて GRAB のマイコン M5Stack から送信される握力データを受信する Meta Quest 2 アプリを実装する．

4 おわりに

4.1 まとめ

本研究では，既存のVRコントローラに巻きつけるだけで握力による操作機能を追加するベルト状センサGRABを提案した．プロトタイプ実装を行い，GRABで握力が検知されることを確認した．プロトタイプ実装ではProcessingによる可視化を行ったが，本来はVRヘッドセットMeta Quest 2と連携して動作させる想定である．そのためには，3節に示した通りUnityのBluetoothプラグインの実装が必要であり，今後実装を完了させる．

プロトタイプでは，握力の検知点が3点に限定される．コントローラの性能を高めるため，計測が可能な点を増やし，検出の空間解像度を上げる．本論文のプロトタイプでは同一列上の3点の検出を行ったが，将来的には3×3の9点を検出する．検出点が9点になれば，小指と薬指による力の違いを検出でき，VR空間内での操作方法の幅を広げる．

GRABの有効性を検証するため，複数人にGRABコントローラを使用したテストアプリを体験させる被験者実験を行う予定である．体験後に5段階評価でのアンケートを実施してGRABの評価を行い，改善点を明らかにする．アンケートでは，GRABの握りやすさ，有効性，現実での動作を制限していたか，装着性の問題など，VRテストアプリ内での動作と，現実の動作では違和感が発生するかの評価を取得する予定である．具体的なアンケート項目の内容は今後検討する．

4.2 課題

プロトタイプ実装で明らかとなった課題が2点ある．1点は，M5Stackの固定についての問題である．プロトタイプでは，GRABの配線やM5StackをVRコントローラに固定する機構は実装しなかった．そのため，操作の妨げとなった．M5Stackを手首に固定し，配線をまとめるためのアダプタを設計し，3Dプリンタで出力する予定である．

もう1点は，シート部分の固定についての問題である．プロトタイプでは，ユーザが把持するVelostatをVRコントローラと接着していない．そのため，意図せずGRABが外れてしまう場合があり，操作の中断や再装着の手間につながる．シート両端に磁石を配置して固定できるような設計で，GRABのシート部分を固定することができると考える．

4.3 今後の展望

GRABでは，握力の値の他，把持を行っている指の種類，把持時の指の順番も操作に用いることが可能である．これら効果的に用いたVR空間での操作方法を検討する．

プロトタイプで開発したGRABでは2，3本の指で握力を計測するが，複数枚を同時に用いることで検出できる指の数を増やすことができる．VRコントローラは左右両手に把持することが多く，両手を使用したGRABの開発も検討する．

本研究ではMeta Quest 2ヘッドセットでの動作を対象として開発を行った．ただし，さまざまなVRヘッドセットを拡張可能なことがGRABの特徴であり，対象デバイスはMeta Quest 2に限らない．HTC VIVEヘッドセットなど，他VRヘッドセットを対象としたテストアプリの実装を行い，デバイスの差異による影響も検証したい．