

タイトル	仮想現実感用ヘッドマウントディスプレイOculus と手ジェスチャ認識デバイスLeapmotion の併用に関する試み
著者	菊地, 慶仁; Kikuchi, Yoshihito; 開地, 祐仁; Kaichi, Yasuhito
引用	工学研究 : 北海学園大学大学院工学研究科紀要(16): 21-26
発行日	2016-09-30

研究論文

仮想現実感用ヘッドマウントディスプレイ Oculus と 手ジェスチャ認識デバイス Leapmotion の併用に関する試み

菊地 慶仁*・開地 祐仁**

An attempt on combination of virtual reality head mount display Oculus
and hand gesture sensor Leapmotion

Yoshihito Kikuchi* and Yasuhito Kaichi**

要 旨 (Abstract)

近年における VR (Virtual reality・仮想現実感) 向けのヘッドマウントディスプレイの開発は、クラウドファンディングによる開発準備段階から市販製品の販売開始へと進捗している。VR 用ヘッドマウントディスプレイは、ゴーグル状のディスプレイで視界の全域を覆い、ヘッドバンドに装着されたイヤホンを併用することで高い没入感を得ることができる。しかしながら、視界が効かない状況でコンピュータに対してキーボードやマウスなどを用いることは現実的ではなく、目視無しに操作できるインターフェースが必要とされる。本研究は一般的な指の座標や動作を検出する Leapmotion controller を用い、VR 用ヘッドマウントディスプレイを装着したまま手ジェスチャで PC とインターフェースを取る試みについて報告する。

1. VR 用ヘッドマウントディスプレイの概要

1.1 Oculus の構成

ここでは、VR 用ヘッドマウントディスプレイについて、2016 年 3 月に出荷が始まった Oculus 社製 Rift CV (Consumer Version)¹⁾ を中心に紹介する。

図 1 に Oculus Rift の出荷時の構成を示す。中央部が Rift 本体であり、左端に Sensor (商品名、後述)、その手前に Oculus Remote (リモコン)、及び右端に Controller (X-BOX ゲームパッド) が示されている。

本体内部には 1024×1200 ドットサイズの有機 EL ディスプレイ 2 枚が格納されており、左右一対のレンズ系を通して右目画像と左目画像を右目及び左目に別々に見せることで立体視を提供する。この際の視野角は 110 度を確保している²⁾。

本体中には、装着者の左右方向と上下方向の首降り動作を検知するジャイロセンサが備えられており、首降りの動作を PC 本体へと送信している。この際に装着者が外界を見回しているのと等しく見えるように本体内部のディスプレイの描画を首降り動作に対応して変動させる。描画のレスポンスが十分な速度であれば、本体を通して外界を見ているのと同等の視覚を得ることが可能となる。

図 1 左端の Sensor は、Oculus 装着者のポジショントラッキングを行うための赤外線カメラである。Oculus 筐体には目視不可能な赤外線 LED が埋め込まれており、この画像から装着者の位置を測定する。またカメラとの距離測定も可能で、Oculus 装着者がこのセンサに近づいたりその付近を動いた結果を VR 出力へ反映させることができる。具体的には、PC へ接近した場合に見える画像に寄って拡大されるような描画などである。

Oculus のような VR ヘッドマウントディスプ

* 北海学園大学大学院工学研究科電子情報工学専攻

Graduate School of Engineering (Electronics and Information, Eng.), Hokkai-Gakuen University

** 日鉄住金テックスエンジ株式会社 (北海学園大学工学部電子情報工学科卒)

NIPPON STEEL & SUMIKIN TEXENG.CO.,LTD. (Graduated from Hokkai-Gakuen University)



図1 Oculus Rift 初期出荷内容，中央部が Rift 本体，左端が Sensor，その手前 Oculus Remote（リモコン），及び右端が Controller（X-BOX ゲームパッド）¹⁾

レイでは，VR酔いと呼ばれる乗り物酔いに似た症状を感じることもある。これは，首振りの動作と得られる視界の間で時間的な隔りがある場合に起こりやすい。Oculusでは，装着者の首振りに対する画面描画の反応時間が20 msecとされている。一般的なスマートフォンやタブレットなどでは100 msec程度なので非常に高速な対応がなされている。またVR酔いを避けるためにレンドリングされた各フレームを補正して首振り運動と違和感が生じにくいための工夫がなされている。このような対策を行うためにレンドリングを行う画像領域は実際のディスプレイサイズよりもはるかに大きな領域を取っていると考えられ，また画像のリフレッシュレートも一般のディスプレイの30 Hzに対してOculusでは90 Hzと高いレートを維持している。

このような描画を行うためにグラフィックス機構には非常に高い性能が要求され，OculusのサイトではNVIDIA GeForce GTX 970以上が推奨されている。

次に本体とPCの接続について図2に示す。画像及び音声はHDMIを通じてヘッドセットに送られる。ヘッドセットでの首振りの検知，ポジショニングセンサによる装着者の位置，ゲームパッドの操作はUSB3.0を通じて本体に送られる。

Oculusの開発当初にはタッチコントローラが提唱され写真なども公開されていた。これはゲー

ムパッドを二つに切断して両手に持つもので，指によるボタン操作と，Touch controllerを振り回すジェスチャの認識を行うものであったが，初期出荷同梱物には組み入れられなかった。このため現状では，装着者からのリプレイやメニューセレクトなどの入力はゲームパッドを用いて行われる。

またヘッドセットを用いた入力としては，装着者が見ている画面上にマーカーが示され，このマーカーと選択候補を一定時間重ねることで選択を行うメカニズムも用意されている。

1.2 本研究の課題

これまで，Oculusのメカニズム及びPCとのインターフェースについて述べてきた。VRヘッドマウントディスプレイは，視界の全体をカバーして立体視映像を提供することで，装着者に強力な没入感を与えることができる。逆に入力デバイスとしてキーボードやマウスのような視覚と手元の感覚に依存するインターフェースを使用することがほぼ不可能となってしまふ点が，大きな問題となっている。従って本研究では，この問題の解消を目的として，VRヘッドマウントディスプレイを装着したままで指や手などで装着者の意図を直接入力できるインターフェースの開発を目指す。

本報告の構成は，第2章で本研究で用いた指ジェスチャ用センサーであるLeapmotion controllerとOculusのそれぞれの開発環境について

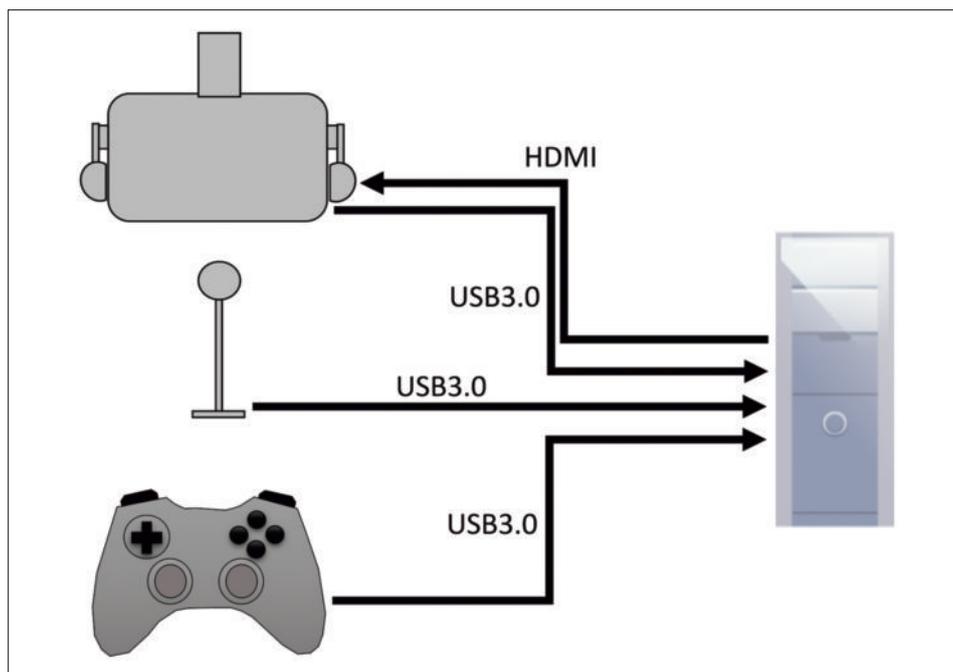


図2 Oculus Rift 本体と PC との接続, 画像及び音声の出力は HDMI にて, 他の入力は USB3.0 を経由して行われる

述べる. 第3章ではこれらを統合して開発する試みについて述べる. 第4章では, 試作したシステムについて述べ, 未達成な点について考察し, 第5章でまとめとする.

2. VR用のヘッドマウントディスプレイ用及び Leapmotion controller 開発環境

2.1 Oculus 用アプリケーション開発環境

Oculus 用アプリケーションの開発のために, Oculus VR SDK が開発元から提供されている. Microsoft Visual C++ にこのライブラリを組み込むことでアプリケーションを開発できる. ライブラリには C++ の DirectX/OpenGL サンプルが付いてくるので対応はかなり容易となる.

また Oculus を用いたゲームコンテンツを開発環境としてゲームエンジンを用いる方法もある. Unity⁴⁾ と Unreal Engine⁵⁾ がそれぞれサポートしており, 他の PC やスマートフォン, タブレットなど向けのゲームと同じように, Oculus 向けのゲームコンテンツとしてリリースできる. プログラミングはオブジェクト指向のフレームワークで行われ, 予めシステムが用意したクラス向けに c# スクリプトプログラムを作成することによって行う. また Unity では Asserts と呼ばれる拡張

モジュールを組み込むことで後述の Leapmotion controller などの外部の機器接続を行うこともできる.

2.2 Leapmotion controller

図3に Leapmotion controller の外観図を示す³⁾. 本体サイズは 80 (W) × 30 (D) × 10.5 (H) mm で, 重量は約 45 g である. PC 本体とは USB で接続する. 測定原理は, 赤外線 LED によって投射される不可視のスポットを本体に2つある赤外線カメラで撮影し, 近距離に存在する棒状の形状を指として抽出するものである. 当初は, 指先が離れた片手単独の状態であれば認識率が極端に落ちることがあり, 指を握りしめた状態, ハンドサインや, また印結びのように指先が合わさった状態では認識できない問題点があったが次第に解消が進んでいる.

Leapmotion controller の開発用ライブラリは開発元のウェブサイトからサンプルコードと伴に入手することができる.

2.3 本研究における課題

Leapmotion controller は Oculus 専用として開発されているものではなく, 広く PC とのイン



図3 Leapmotion controller 装置外観図³⁾



図4 Oculus DK2の正面に Leapmotion をマウントした状態³⁾

ターフェースを取ることを目的としている。このため Oculus の開発環境と合わせて開発環境を構築し、また動作においても互いに影響を与えないように開発を行う必要がある。

3. Oculus 向け Leapmotion Controller 連携アプリケーションの開発

3.1 使用器材

本研究で使用した器材は、CPU Corei7, Memory 8 Gbyte, 256 Gbyte SSD, GPU として Nvidia GTX 980 を装着した PC を用いた。開発

時に用いた Oculus は DK2 と呼ばれるバージョンで、現行の市販品版 CV (Consumer version) の一世代前のものである。この DK2 の正面部分に Leapmotion controller をマウントして用いた (図4)。

3.2 C++ アプリケーションとしての開発と 問題点

Microsoft Visual C++ を用い、Oculus SDK 及び Leapmotion SDK を組み込んで、コンパイラにライブラリが所在するフォルダの箇所を指定した。指定に関しては別個のフォルダに格納しそれ

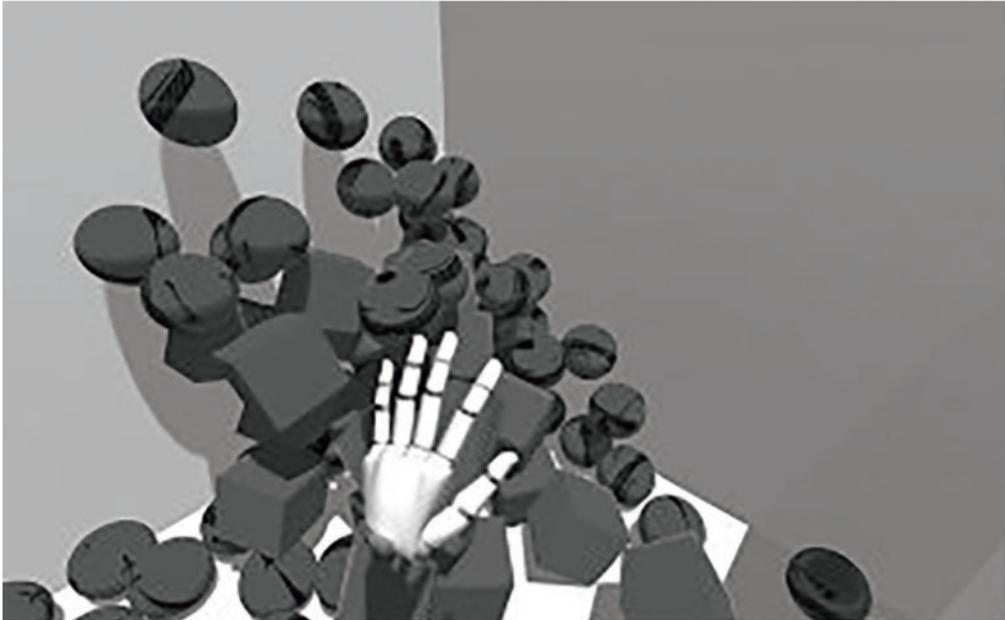


図5 ジェスチャ認識に成功し立方体と球のオブジェクトを出現させた状況

それを登録するだけなので問題は生じなかった。

次に実際のコンパイル作業に入ったところリンク時に問題が発生した。理由は、Leapmotion SDK 中でグラフィック描画を行う Cynder と呼ばれるライブラリ中に OpenGL を用いるライブラリがあり、これが Oculus SDK で用いられている OpenGL と競合してしまうためであった。リンクの順番の変更など調整を行ったが解決は難しかった。Cynder もしくは Oculus SDK から競合する OpenGL を削除する方法も考えられたが、それぞれのライブラリに初めから組み込まれて提供されているため、結果的に C++ のプログラムとしてコンパイルを行うことは不可能と判断した。

3.3 Unity によるゲームコンテンツとしての開発

3.2 で述べたように C++ アプリケーションとして開発を行うことが難しかったので、ゲームエンジンの Unity を用いることとした。そのために Leapmotion controller を併用目的の Leapmotion Unity Core Asserts を Unity に組み込んで用いた。

Unity の Virtual Reality Supported をオンにし Oculus Rift に対応させた。手のモデルを VR 内に表示し、親指と人差し指を接近させると引力を発生させてオブジェクトを吸着し、保持している状況を可能にした。

Unity のシステム中では以下の 4 種類のジェスチャが用意されており、検知した際に特定のイベントを発生させることができる。

- ・円を描く（サークル）
- ・キーをタップ（下に指を素早く振る）
- ・スクリーンをタップ（前に指を素早く振る）
- ・スワイプ

これら 4 つのジェスチャを検知可能にし、スワイプ認識時に立方体を、サークル認識時に球をシーン内に発生させるスクリプトをプログラムした。

4. 動作実験

4.1 実験結果

Oculus 及び Leapmotion を PC に接続し、システムの動作確認を行った。

特にエラーは発生せず、Oculus Rift のヘッドトラッキングに対応し、ジェスチャでオブジェクトを発生させ、発生させたオブジェクトを把持することが可能であることを確認した。

4.2 問題点

実行時には以下のような問題点が発生し、今後の解決の必要性が認められる。

- 1) Leapmotion を PC モニタへ向けると手の誤

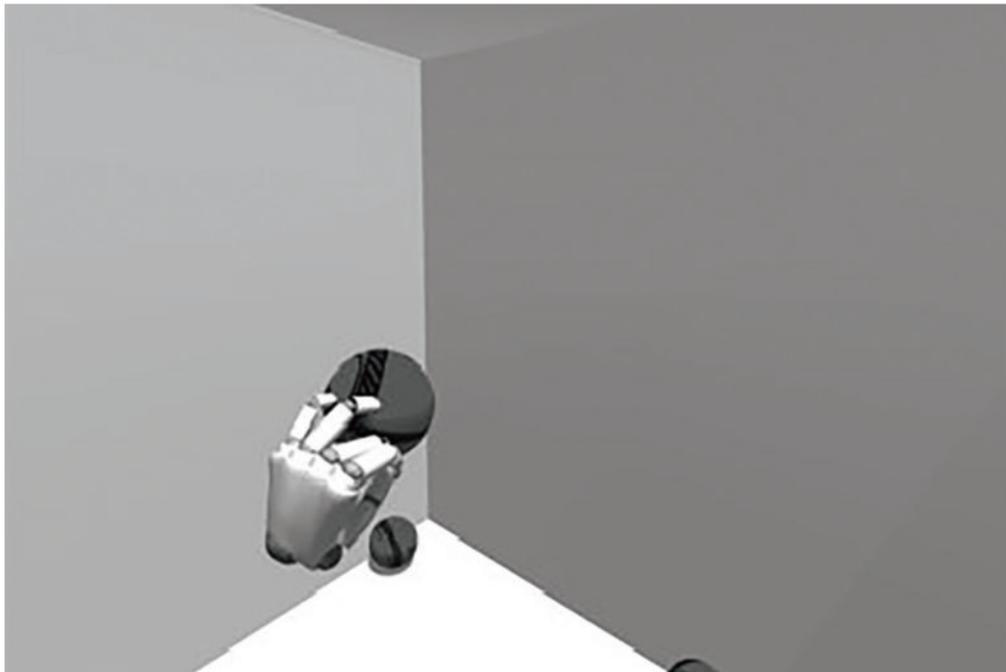


図6 発生させた球オブジェクトを把持している状況

認識が発生し、存在しないはずの手がシーンに表示される。原因としては Leapmotion をモニタ画面に向けてしまうことで、赤外 LED 発光がモニタ画面で反射されてしまい誤認識が発生していると考えられる。

- 2) ジェスチャを一回行ったつもりでも複数回のジェスチャとしての検知や、別のジェスチャとしての認識が発生した。これは、人間が行っているジェスチャ動作に対するシステム側での認識速度が速すぎるためと考えられる。ジェスチャ認識を行う際にジェスチャの持続時間や一度ジェスチャを行ったあとの認識再開までの時間等を調整する必要があると考えられる。

5. 結論

本報告では以下の報告を行った。

- 1) VR 用ヘッドマウントディスプレイ Oculus を紹介し、マウスやキーボードによらないインターフェース方式が必要である点を認識した。
- 2) 指の位置を検出してジェスチャ認識を行うためのセンサ Leapmotion controller を紹介し、これらを組み合わせたシステムとして用いる方式を提案した。
- 3) C++ ライブラリを用いた開発形式ではコン

パイルに問題が発生し、Unity を用いる形式での開発を行った。

- 4) ゲームエンジン Unity 上のスクリプトプログラムとして Leapmotion の使用を可能としてシステムを試作した。開発したシステムを用いて Oculus 装着者がジェスチャで PC とのインターフェースを取れることを確認し、幾つかの問題点があることも確認できた。

今後の展開としては、より実際の利用を想定して、ジェスチャ認識の誤動作を排除し安定した認識を実現することが課題である。その後、具体的にジェスチャを用いたインターフェースの併用について試みる必要がある。

参考文献

- 1) Oculus 公式サイト
<https://www.oculus.com/>
- 2) Unity + Oculus Rift 開発メモ
<https://framesynthesis.jp/tech/unity/oculusrift/>
- 3) Leapmotion 公式サイト
<https://www.leapmotion.com/>
- 4) Unity 公式サイト
<http://japan.unity3d.com/>
- 5) Unreal Engine 公式サイト