

タイトル	仮想現実（VR）を使った初心者のための ラケットスポーツ練習システムの構築
著者	平田， 恵啓; Hirata, Yoshihiro
引用	工学研究：北海学園大学大学院工学研究科紀要(20)：23-28
発行日	2020-12-25

研究論文

仮想現実（VR）を使った初心者のための ラケットスポーツ練習システムの構築

平 田 恵 啓*

A Study on developing a racket sports training VR system for beginners

Yoshihiro Hirata*

要 旨

パソコンおよびグラフィックチップの高速化と比較的安価な VR システムの普及により、個人レベルで仮想現実（VR）を楽しむことができるようになってきた。一方、卓球やバドミントン・テニスといったラケットスポーツは道具と仲間さえいれば簡単に楽しめるものの、ボールを狙ったところに打ち込むための動きを体得するのは容易でない。そこで本論文では、ラケットスポーツのうち卓球・バドミントン・テニスを対象とした初心者が単独でも取り組める練習環境の VR システム構築を試作する過程で明らかになった、実現のためのシステム要件ならびに仕様について報告する。

1. はじめに

仮想現実（Virtual Reality：VR）は現実には存在しない世界や直接目では見ることのできない遠方やマイクロ世界を人工的に作り出し、目前で見ているような錯覚を与えさらには疑似体験までをさせる技術である。パソコンの性能向上と個人で導入可能な VR システムの普及により、VR は身近な存在となりつつあるが、まだゲーム分野以外でのキラーアプリは存在していない。

一方、温泉宿の卓球や公園でのバドミントンのようにちょっとした道具と仲間がいればできるラケットスポーツは身近な存在である。しかし初心者が慣れないラケットを使ってボールやシャトルを狙った場所に打ち込むことはとても難しく、一人では練習も素振り程度しかできない。

そこで、本論文では VR 上で動作するラケットスポーツの練習システムの構築を行う。既存の VR アプリの問題点を挙げて、その解決方法や組み込むべき機能についての考察を行う。

2. VR システムとその変遷

仮想現実感を与えるには少なくとも人工的な視聴覚情報が必要となる。特に映像はきめの細かい自然な景色が頭部の動きに追従して滑らかに変化させるための高速演算能力と高品質表示装置が求められる。なお、滑らかな映像とは、毎秒 30 枚以上の静止画（以後、フレームと呼ぶ）を連続的に計算・生成・表示で得られる動画のことで、フレームレートが高いほど滑らかになる。

2.1 高機能型 VR システム

1980 年代にはロボットの遠隔制御に向けた視覚・聴覚・触覚を提示する装置の開発報告が行われている¹⁾。2007 年発表の国産の透過投影型ウェアラブルディスプレイシステムは高視野角・高解像度で高臨場感と高没入感を体験できるもので、頭部や足元に取り付けたマーカ位置を複数カメラで同定するモーショントラッキングシステムの追加で CG 世界を歩き回る映像を作り出すことができるが、基本システムが 840 万円の価格設定であった²⁾。2020 年 8 月に発売告知された Teslasuit³⁾ はモーションキャプチャ・電気刺激型の触覚フィー

* 北海学園大学大学院工学研究科電子情報生命工学専攻

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

ドバック・心拍数計測を統合したスーツ型デバイスで、視聴覚提示デバイスとの組み合わせで商業分野に加えて医療・スポーツ競技などの新たな方向への発展も期待されるものの、上下で55万円程度である。

2.2 普及型 VR システム

継続的なパソコンやグラフィックチップの性能向上は、リアルタイムCGに要するハードウェアを趣味の範囲で賄えるようにしたが、液晶モニタ越しの体験はVRに程遠く、安価な専用の視覚提示装置が熱望されていた。2013年発表のOculus Rift⁴⁾は加速度センサを内蔵するゴーグル型の画像表示装置(ヘッドマウントディスプレイ:HMD)を採用し、頭部の上下・左右・回転の動き検出と画像表示ができるVRシステムで、2016年に製品版が600ドルで販売された。またこの年にはHTC VIVE、PlayStation VRも発売されたことから手軽に高品質なVR体験ができるVR元年と呼ばれ、VRブームが巻き起こる。

2.3 既存 VR ラケット競技アプリの問題点

すでに卓球やテニスのVRゲームが複数販売されている。これらはゲーム性を優先するあまり、ラケットのフレームに当たったボールが中心部に当たったかのように正しく跳ね返ってしまう、力の入らない姿勢での衝突でもいい感じで飛んでいく、コントローラ内蔵の振動素子による衝突感のフィードバックが物足りないなど、まじめな練習環境としてはとても不十分なものである。

3. VR システム構築に向けた要件の確定

3.1 アプリ環境を構築するシステムの選定

3.1.1 ハードウェアの選定

幾つかの普及型VRシステムが存在するが、独自アプリ開発の容易さと追加の位置検出デバイス



図1 HTC VIVE PRO⁵⁾

の有無を考慮してHTC VIVE PRO(図1, 表1)を開発の基本システムに採用した。さらに、付属のコントローラよりも実物のラケットに類似した把持感覚が得られるように、プラスチック製の汎用ラケットと卓球ラケット(図2)を導入し追加のVIVEトラッカ2.0(図3)を装着した。開発用のPCおよびGPUはHTC VIVEの推奨スペックを参考にし、CPU:AMD Ryzen7 1800X(3.6GHz)、メモリ:16GB、OS:Windows10、GPU:NVIDIA GeForce GTX 1080とした。

3.1.2 開発ソフトウェアの選定

独自アプリの開発環境には、開発開始の時点で広く用いられておりプログラミングについての情報が豊富なユニティ・テクノロジー社のUnity⁶⁾を使用した。Unityはゲーム開発プラットフォームとして知られており、2Dや3Dゲームの作成をプログラミング言語C#で書かれたスクリプトを使うことでアプリ内のオブジェクトをプログラマブルに扱うことができる。アセットと呼ばれるテクスチャや3Dオブジェクト、スクリプトなどを含む事前に用意されたライブラリファイルを組み込むことで機能拡張を可能としている。Steam VR pluginアセットの導入により、VIVE PROのHMDとコントローラおよびVIVE

表1 HTC VIVE PROの主な仕様⁵⁾

ヘッドセット (HMDを含む)	
スクリーン	デュアル AMOLED 3.5 インチ
解像度	片目あたり 1440×1600 ピクセル (合計 2880×1600 ピクセル)
リフレッシュレート	90 Hz
視野角	110 度
オーディオ	Hi-Res ヘッドセット
入力	内蔵マイク
センサ	SteamVR トラッキング, G センサ, ジャイロスコープ, 近接センサ
コントローラ	
センサ	SteamVR トラッキング 2.0
入力	多機能トラックパッド, グリップボタン, 2段階トリガー, システムボタン, メニューボタン
連続使用時間	約 6 時間
VIVE トラッカ 2.0	
センサ	SteamVR トラッキング 2.0
連続使用時間	約 6 時間

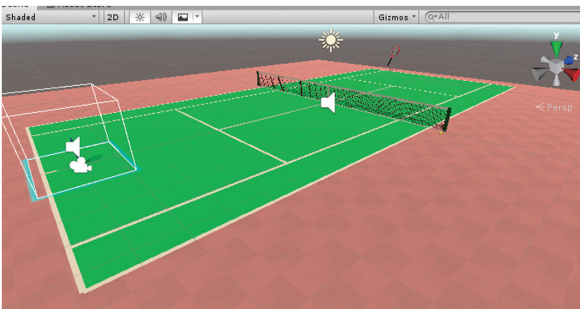
図2 汎用ラケットと卓球ラケット⁵⁾図3 VIVE トラッカ 2.0⁵⁾

図4 Unity 内に配置したテニスコート

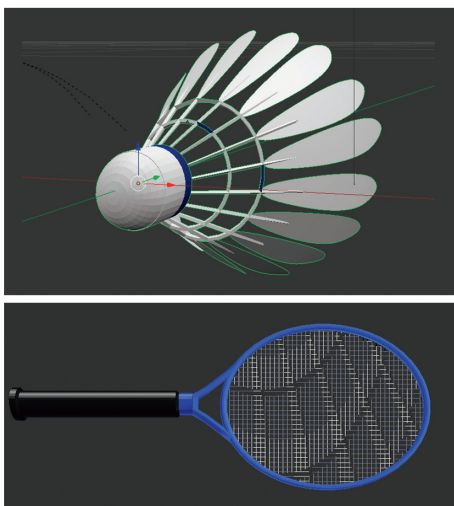


図5 Blender で作成したバドミントンシャトルとテニスラケット

トラッカが利用可能になる。

独自アプリ内で用いる卓球台やテニスコート（図4）などの3DオブジェクトはUnityで提供される直方体などの基本オブジェクトの変形・組み合わせ・着色で作成し、バドミントンの羽や卓球ラケット・ストリングの張られたテニスラケットや支柱間に固定されたネットなどの複雑な形状のオブジェクトはオープンソースで無償提供されている3DCGソフトウェアのBlender⁷⁾を用いて作成した（図5）。

3.2 基本アプリの開発

本研究では独自のVRアプリケーションを開発するので、前節で示したシャトルなどの3Dオブジェクトの作成に加えて、ユーザのラケットを振る動作に起因するオブジェクトの振る舞いを規定する必要がある。

3.2.1 3D オブジェクトの制御方法

Unity上のあらゆる3Dオブジェクトは基本コンポーネント Transform（プロパティ：X, Y, Z 座標の位置, X, Y, Z 軸の回転角, X, Y, Z 軸でのスケール）を有している。フレームの計算・生成前にスクリプトでこれらのプロパティ値を変更すると、それが反映され3D空間内での移動・変形を行うことができる。この方法では、他のオブジェクトとの衝突などの相互作用もすべてスクリプトで判断・制御することになり、物理シミュレーション計算を含めてプログラマがすべてを記述することとなる。

そこでUnityでは、Rigidbody コンポーネント（プロパティ：質量, 空気抵抗, 重力効果, 物理エンジン有無, 衝突検出, 軸に沿った動き制限）と Collider コンポーネント（プロパティ：中心位置, スケール, 摩擦係数・反発係数, 物理エンジン有無）が用意されている。これらを3Dオブジェクトに付与すると、Nvidia PhysX 物理エンジンによる自動計算が用いられ、初期状態を与えるだけでオブジェクトの時間的な推移や他のオブジェクトとの相互作用もコードの記述不要で処理させることができる。この方法でプログラマがオブジェクトの動きに変化を与えたい場合は、オブジェクトの Transform を直接書き換える代わりに AddForce() コマンドで力を加えるよう物理エンジンに指示を与えることが求められる。

3.2.2 貫通問題とその解決策

上記の手続きを理解したうえで、手始めに物理エンジンを利用した卓球シーンをアプリ構築し、その動作を検証した。その結果、とてもゆっくりとラケットとボールが接触するケースでは自然な感じでボールが跳ね返るのに対して、早めの速度で衝突するケースではボールがラケット後方にすり抜ける貫通現象が確認された。この対処法として Rigidbody の衝突検出プロパティを用いるべきとのネット記事を複数確認したが、解決できない。そこで、オブジェクト同士が高速で衝突する場合は図6のように Collider 領域の重なり判定がフレームとフレームの間で衝突が起こるために物理エンジンが衝突を検出できないと考えた。

そして、図7のようにラケットに取り付ける Collider の厚みを進行方向後方に増やす処理を追加した。この処理により卓球での貫通はほぼ解決することとなった。

3.2.3 物理エンジンの仕様に関する考察

同様の手法でバドミントンのシーンを作成し動作検証をしたところ、ラケットと衝突後のシャトルがあらぬ方向へ飛ばぬ現象が時折観察された。原因としてバドミントンではより速くラケットを振ることを考えたが、十分な説明ができない。改めて卓球に戻って種々の条件でボールの挙動を調べ

たところ、次の事実が明らかになった。

図8は卓球台とボールおよびラケットの Collider をすべて有効にし、卓球台上にボールが静止している状態でラケットを卓球台の上からわずかに重なるようにおろした際のスクリーンショットである。卓球台にボールが半分近くめり込んでいるが、さらにラケットを押し下げるとボールは卓球台をすり抜けて床に落ちてしまった。これは VR の画像が物理エンジンで計算されており、卓球台上のボールはラケットと無関係に静止したままであるとの想定に合わない。ネット上でも当該事象の記事は見つからなかったため、これまでの知見をもとに以下の考察を導き出した。

卓球台とラケットに限定すると、現実世界では両者は重ならないので、VR 空間でも Collider が付与された両者は物理エンジンによって重ならないように処理されるはずである。一方で VR 内のラケット位置はユーザが握る現実のラケットに固定したトラックの位置を反映させている。卓球台オブジェクトにラケットオブジェクトつまりコントローラがぶつかると手を動かして矛盾が生じた場合は、ラケットオブジェクトがコントローラから脱落するか、オブジェクトに付けられた Collider に例外的なふるまいを発生させて帳尻を合わせる処理が行われる可能性がある。

ラケットが卓球台に重なることでボールが卓球台にめり込む事象は、卓球台オブジェクトに固定されているはずの Collider の上端が当初の設定位置よりも下がったことを示している（図9）。また、さらにラケットを下げていくとボールが卓球台を貫通したのは、卓球台の Collider の厚みがゼロになったためと推察できる。つまり、使用している物理エンジンは VR のユーザ体験を優先させ

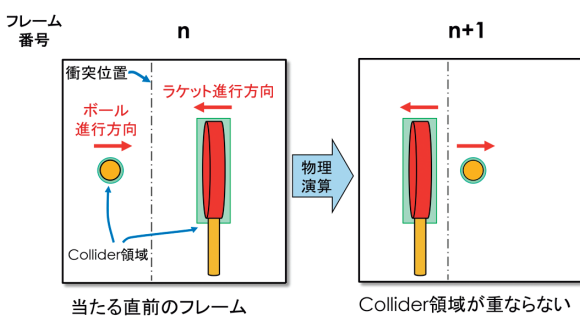


図6 貫通現象の考察

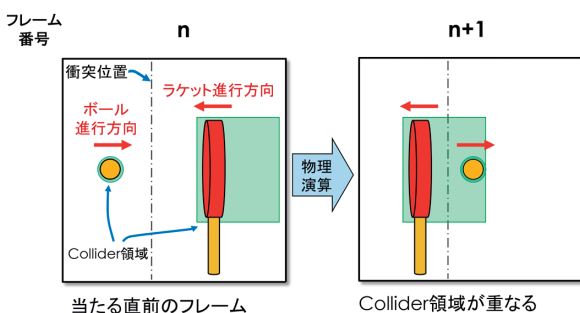


図7 貫通現象への対処法

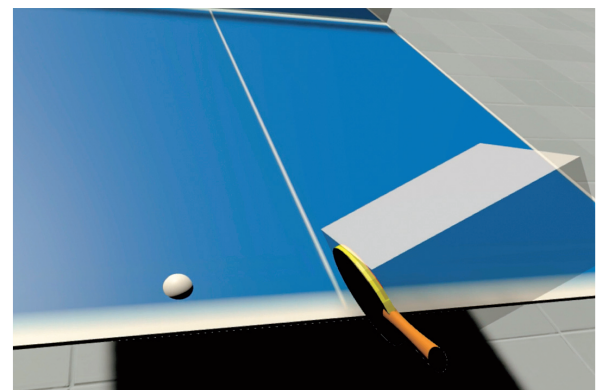


図8 卓球台にめり込むボール

る設計であり、コントローラとラケットオブジェクトを固定してラケットの Collider は設定のままとする一方で、卓球台の Collider 厚みを自動的に変化させて物理シミュレーションの破綻を防ぐようにふるまったと推察される。

以上の考察を踏まえてバドミントンのシーンで衝突直後のフレームを観察したところ、シャトルがラケットの Collider 内部に重なるように表示されており、上述の破綻防止の処理が行われた結果として、正しく跳ね返ることなくあらぬ方向に飛んでしまったと結論付けた。また、3.2.2 で正しく動作して見えたのは、温泉卓球のようにラケットのスイング速度がとても遅い状況ではボールがラケットの Collider 内部に埋まる状態が発生しなかったものと思われる。

3.2.4 物理エンジンの仕様を考慮したラケットの取り扱い方法

卓球台の Collider 厚みがゼロになったり、バドミントンのシャトルがあらぬ方向に飛ぶなどを避けるには、ラケットの Collider の物理エンジンプロパティをオフにして物理計算から外せばよい。しかしラケットとボールやシャトルとの相互作用も自動計算されず、常にすり抜け現象が生じる。

対応策としてラケットとボールの Collider が重なったと判定されたタイミングで、従来の物理エンジンに代えてスクリプトによって衝突前の進行方向と逆方向の力を加える方法をとるようにした。この力の大きさは、ひとつ前の画像と現在の画像でのラケット位置の差分から速度ベクトルを求めて適当な係数をかけたものとした。

3.3 一人で練習できるための設定

初心者のユーザが単独で練習できるようにするためには、相手コートに飛ばしたボールやシャトルが再びユーザに向かって戻ってくるラリーの状態を実現する仕組みが必要となる。また、ユーザに向かってくるボールやシャトルは適度な速度でラケットが振りやすい位置に打ち出されることが

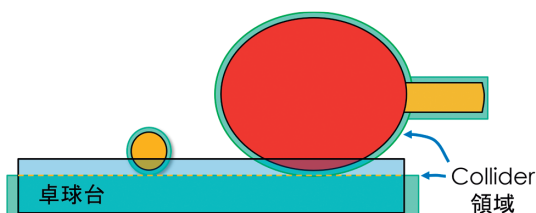


図9 卓球台にボールがめり込む現象の考察

望ましい。今回は相手側コートにラケットを振る相手ロボットなどを作る代わりにラケット単体を配置し、ボールやシャトルに合わせて移動して跳ね返す設定を組み込んだ。

3.3.1 卓球シーンのコード実装

卓球は比較的プレイエリアが狭い。そのため、相手側の卓球台の奥端に置いたラケットが卓球台の幅 $152.5 + 10$ cm、台の上面から高さ 50 cm の四角い領域内で移動し、衝突後のボールが自陣側に戻ってくればラリー感覚が得られる。そこで、フレーム毎にボールとラケットの左右および上下方向の位置の差を計算し、その差に応じた反対向きの力を与えるコードを加えた。

3.3.2 硬式テニスシーンのコード実装

テニスコート半面の広さは幅 10.97 m、奥行 11.89 m と VR で用いられる数 m 角に比べて広いので、ラリーを想定してユーザは右側サイドベースライン付近に立つように設定した。また、相手ラケットはベースラインの 0.4 m 手前に配置し、幅 11 m 高さ 2.7 m のエリアで卓球シーンと同様に移動させる設定とした。ユーザにとって打ちやすいボールが戻ってくるように、返球時点のユーザラケット位置の方向で且つサービスラインの前方 1 m に自由落下する速度になる力をボールに加えてワンバウンドするようにした (図 10)。

3.3.3 バドミントンシーンのコード実装

バドミントンで使用するシャトルは空気抵抗が大きくその落下位置はコートの前後に大きく変化する。そこで、相手ラケットの移動は前後左右の水平面内とし、シャトルの落下地点は図 11 に示すようにフレーム間のシャトル位置の差からラ

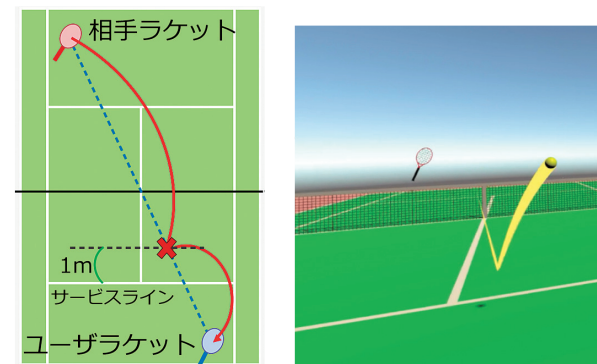


図10 テニスでのボール落下位置設定

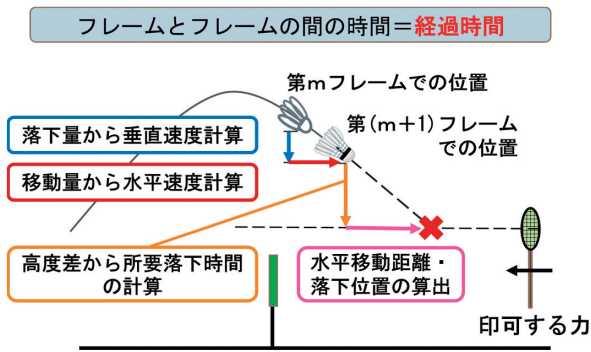


図11 バドミントンでの相手ラケットの前後方向への移動

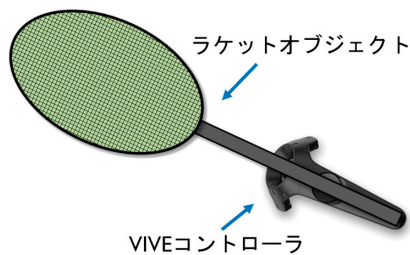


図12 バドミンントンのラケットオブジェクトはコントローラに取り付けた

ケットフェース高さにシャトルが到達する地点を随時計算してそこに進むように力を与えた。

バドミントンラケットの重量は75~95gと非常に軽いので、手首のスナップを利かせてシャトルを打つ、つまり、手首の力を使ってラケットヘッドを高速に動かすことが頻繁に行われる。重量85gのVIVEトラックを装着した汎用ラケットは、その重心が手から20cmほど離れており、これをスナップを利かせて動かすことは不可能であった。そこで、VIVEトラックの利用をあきらめて、代わりにVIVEコントローラにバドミントンラケットのオブジェクトを取り付け、コントローラの座標系で処理するように設定を行った(図12)。VIVEコントローラは約200gと本物のラケットの倍ほど重い、重心位置が手に近くなることで現実的な動きができることを確認した。

4. まとめと今後の展開

本論文では、ラケット競技の初心者が単独でラリー練習を行うためのVRアプリシステムの基礎部分を構築するとともに、満たすべき最小の仕様を決定した。

ラケットとボールやシャトルとの物理作用はスクリプトでのプログラミングとなるため、衝突時の力の印加に加えて、特に卓球や硬式テニスでは摩擦を考慮したボールへの回転印加処理の組み込みが急がれる。

また、効率の良い運動の習得には適切なアドバイスとフィードバックが重要である⁸⁾。そのための方策には、1)各競技の熟練者やユーザ自身のラケットスウィングを外部カメラで撮影しVR内でビデオ再生する機能、2)VR内でボールやシャトルを理想的に打つ手のオブジェクトを表示させ、その動きに合わせてラケットを振る反復練習機能、3)Teslasuitのように、正しくない姿勢で打った際に筋肉に違和感を与えるフィードバック機能を組み込むといった改良が検討課題に挙げられている。

参考文献

- 1) PC Magazine 80年代・90年代のVR <https://www.pcmag.com/news/the-wacky-world-of-vr-in-the-80s-and-90s>
- 2) 西川善司, 古林克臣, 野生の男, izm, 比留間和也: VRコンテンツ開発ガイド, エムディエヌコーポレーション, 2017.
- 3) VR Electronics 社 Teslasuit 公式サイト <https://teslasuit.io/>
- 4) Oculus 社 公式サイト <https://www.oculus.com/>
- 5) HTC 社 VIVE 公式サイト製品情報 <https://www.vive.com/jp/product/>
- 6) Unity technology 社 Unity 公式サイト <https://unity.com/ja>
- 7) Blender Foundation Blender 公式サイト <https://www.blender.org/>
- 8) 日本スポーツ協会: 公認スポーツ指導者養成テキスト 共通科目 III, pp.140-146, 2013.