

Title	Unity と HTC VIVE を利用したシミュレーション結果 の3D-VR 表示入門
Author(s)	萩田, 克美; 松本, 茂紀
Citation	サイバーメディアHPCジャーナル. 2018, 8, p. 21-28
Version Type	VoR
URL	https://doi.org/10.18910/70827
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

# Unity と HTC VIVE を利用したシミュレーション結果の 3D-VR 表示入門

萩田 克美<sup>1</sup>、松本 茂紀<sup>2</sup> <sup>1</sup>防衛大学校 応用物理学科、<sup>2</sup>日立製作所 研究開発グループ

#### 1. はじめに

VR ゲームの普及を受けて、エンターテイメント や建築や医療分野の利用[1-3]のみならず、計算科学 研究での VR 活用が急速に進んでいる[4-8]。特に、 HTC VIVE[9]をはじめ VR 装置が低価格化し、グラ フィックデバイスの性能も向上し、更には Unity[10] や Unreal Engine[11]等のソフトウェア開発環境が発 展したことが、強い牽引力となっている。VR 空間 内で自由に探索し、シミュレーション結果、特に、 複雑な 3 次元構造の空間認知や挙動の把握が促され ると期待している。

大阪大学サイバーメディアセンターは、可視化サ ービス[12]として、多人数でのディスカッション利 用が可能な「24 面大型立体表示システム(大阪大学 吹田キャンパス)」と「15 面シリンドリカル立体表 示システム(グランフロント大阪 9F うめきた拠 点)」を一般向けに供用している(図1)。

それらの有効活用としても、手元の環境でのコン テンツ検討は有意義である。本稿では、手軽な 3D-VR 表示の入門として、動きのないコンテンツに 限定し、3D プリンタでも用いられる 3D データを VR 空間中に表示する簡単な方法を明らかにするこ とで、没入型 3D-VR 可視化装置の活用の機運を高め ることを目的とする。

なお、動きのある分子動力学シミュレーションの 結果を表示させる場合は、現時点では専門的なプロ グラムの知識が必要である。例えば、汎用の可視化 アプリ VMD[13]や AVS/Express の表示をそのまま 3D-VR に変換するミドルウエア(EasyVR や AVR)を 利用する方が、独自に開発するよりも、実務上にお いては効率的であると考えられる。

本稿では、HTC VIVE と Unity により、3D データ を VR 空間中に表示する簡単な手順を紹介する。 (a) 24 面大型立体表示システム



(b) 15 面シリンドリカル立体表示システム



図1 大阪大学の没入型 3D-VR 可視化装置

# 2. HTC-VIVE と Unity について

HTC-VIVE は、PC 用の VR ゲームデバイスとして 最も普及しているデバイスである。測位方法の工夫 により"ルームスケール"での VR が可能となった 家庭ゲーム用 VR デバイスである (図 2)。



図 2 HTC-VIVE と、GPU 搭載パソコン

Unity はゲームエンジンと呼ばれ、3 次元空間内の 物体について、描画や物理演算などのパーツやツー ルが用意されたゲーム開発のための統合環境であ る。現在、ゲームエンジンとして主流なものは、Unity と Unreal Engine である。Unity は主に個人向けのソ フトウェアとして作成されたものである。一方、 Unreal Engine はゲーム会社が開発したものである。 Unity や Unreal Engine は、個人やアカデミック利用 は無料であり、導入が容易である。Unity を使うメリ ットは、Unreal Engine に比べ Assets という開発素材 が充実していることや、インターネット上での情報 が多いことが挙げられる。

### 3. HTC-VIVE のデバイスの準備設定

3.1 VR デバイスの関連ソフトウェアのインストール

HTC VIVE を利用する場合、NVIDIA の最新ドラ イバをインストール後、ViveSetup を実行して、 SteamVR を導入する。手順は以下の通りである。 (1) 最初に、Windows アップデートを実行するなど して、NVIDIA の新しいドライバを設定する。

(2) 次に、ViveSetup を通じて、PC ゲーム配信ツー
 ル「Steam」をインストールする。(Steam のアカウント作成が必要である。)

(3) 引き続き、SteamVR をインストールする。時間 短縮で ViveSetup を用いない場合は、Steam の中のラ イブラリで、SteamVR を検索しインストールする。 執筆時点でのバージョンは 1515522829 であった。

#### 3.2 VR デバイスの利用準備

次に、HTC VIVE の初期設定を行った(図 3)。 (1) ケーブル配線を行い、センサーを設置後、 SteamVR を起動する。(デスクトップの SteamVR の アイコンをクリックするか、Steam 上で SteamVR を プレイ実行する。)

(2) ハンドデバイスの電源を ON にして、ヘッドマウントディスプレイとともに、センサーの正面に置いて、デバイスを認識させる。

(3) ルームセットアップを実行し、プレイエリアを 設定した。



図3 デバイスの認識とルームセットアップ

#### 4. VMD での OBJ ファイルの作成

分子動力学シミュレーションの可視化で良く用い られる VMD では、Wavefront OBJ ファイルを出力す ることができる。ここでは、表示試験用のデータ作 成の手順を示す。

VMD では、Nanotube Builder という機能で、カー ボンナノチューブの原子配置を作成することができ る(図 4)。作成したモデルを、図 5 のように Ball-stcik 表現に変更し、図 6 の手順で Wavefront OBJ 形式の ファイルを出力する。Wavefront OBJ 形式では、「.obj」 と「.mtl」の 2 つの形式のファイルが出力される。



図 4 VMD の Nanotube builder での CNT の作成



図5 VMD での Ball-Stick 表示の設定



図6 VMD での OBJ ファイルのエクスポート

# 5. Unity を利用したアプリの表示手順

Unity 2017.3 をインストールした後に、Wavefront OBJのデータを、HTC VIVE で表示させるために必 要な操作手順を示す。「VR 用のカメラのセットアッ プ」と「オブジェクトの配置」の2つのステップを 実施する。

## 5.1 VR 用カメラのセットアップ

(1) Unity を起動しプロジェクトを作成した(図7)。



図7 Unity での新規プロジェクトの作成

(2) [Window]-[Asset Store]で、Asset Store に接続し、
 「SteamVR」を検索した。用いたアセットのバージョンは 1.2.3 である。「Download」を実行した(図 8)。



図8 Unity アセット「SteamVR」のダウンロード

(3)「Import Unity Package」で、「SteamVR plugin」を インポートした。「SteamVR Setting」では「Accept All」 を選択し、下側の画面中の Assets に SteamVR のフォ ルダが作成されていることを確認した(図 9)。



図9 アセット「SteamVR」のインポート

(4) 次に、カメラの設定を行った。具体的には、図
10 の通り、左側の「Hierarchy」の中にある Main
Camera を削除して、[Assets]-[SteamVR]-[Prefabs]の中
にある「Camera Rig」を「Hierarchy」にドラッグ&
ドロップした。



図 10 VR 用のカメラの設定

# 5.2 オブジェクトの配置

(1) 3D-VR データを下側の「Assets」にドラッグ&ド ロップした。図 11 には、3D-VR データとして、 CNT.obj と CNT.mtl のペアを取り込んだ状態を例 示している。 Assets の中の「CNT.obj」のアイコン を、左側の「Hierarchy」にドラッグ&ドロップした。 この後、中央の「Scene」のウィンドウに、3D オブ ジェクトが表示された。



図 11 CNT.obj と CNT.mtl のインポートと、表示設定

(2) オブジェクトの位置や角度、スケールの変更は、 左側の「Hierarchy」で対象のオブジェクトを選択し た後に、値を変更する。このとき、Play モードが停 止状態でなければならない。中央上部の Play ボダン を押し、Play モードを実行した(図 12)。これによ り、ヘッドマウント中に、CNT.objの内容が配置さ れた空間が表示されることを確認した。



図 12 OBJ の位置調整と VR 表示 (Play モード)

### 6. 断面カット機能の設定

Unity では「Cross Section Shader」というアセット を用いることで、任意位置での断面カットが実現で きる。また「VRTK」というアセットを用いること で、断面位置を VIVE のハンドデバイスにより操作 可能となる。Cross Section Shader はバージョン 1.0 を、VRTK のバージョンは 3.2.1 をそれぞれ用いた。

# 6.1 断面カット機能の追加

 Assets Store で、「Cross Section Shader」と「VRTK」 をインポートした(図 13)。



図 13 断面カット機能用のアセットのインポート

(2) 左側の「Hierarchy」の中で、[GameObject]-[3D
 Object]-[Quad]を選択し、「quad object」を作成した。
 名前を「Quad」から「Cutting\_plane」に変更した。
 右側の「Inspector」で、「X」と「Y」の値を「3」に

設定した。「Mesh Collider」の項目中の「Convex」の チェックボックスを on にした(図 14)。



図 14 Cutting\_plane の設定

(3) [Assets]-[VRTK]-[Scripts]-[Interactions]の中にある 「VRTK\_InteractableObject」を、左側の「Hierarchy」 の「Cutting\_plane」にドラッグ&ドロップした。右側 の「Inspector」で、「VRTK\_Interactable Object (Script)」 の項目中の「Is Grabbable」のチェックボックスを on にした(図 15)。



図 15 Cutting\_plane を Grab 可能とする設定

(4) さらに、右側の「Inspector」の「Add Component」で[Physics]-[Rigidbody]を追加した(図 16)。新たに追加された「Rigidbody」の項目中の「Is Kinematic」のチェックボックスを on にした。「Use Gravity」を off にした。[Assets]-[Cross Section Shader]-[Examples]-[Materials]の中の「DoubleFaceTransparent」を、「Cutting\_plane」の「Inspector」にドラッグ&ドロップした(図 17)。



図 16 Cutting\_plane の Rigidboty 設定の追加



図 17 Cutting\_plane の Rigidboty の属性設定

(5) Assets/Cross Section Shader/Scriptsの中にあるスク リプト「OnePlaneCuttingController.cs」を、メモ帳で 変更した(図 18)。

	ts:	~0	Scripts(7)te(	. <i>P</i>
ArtBall.cs ArtBall.cs ArtBall.cs Gener.TheraBenes.CutingControl Coment TheraPerfusion Coment TheraPerfusion Coment TheraPerfusion Coment TheraPerfusion ThereAARTanesCutingController.cs ThereAARTanesCutingController.cs Coment Com	oller : N m rer>(); Transform m.positio (_PlaneP erd.mater 'lanePosit 'lanePosit	AnnoBehaviou Wector(new Wector(new Sorral", not Sorral", not Sorral", not Sorral", not Sorral", not Sorral", not Sorral", not Sorral", not Sorral", not Sorral	Vector3(0 Vector3(0 rms]); position); tion);	.0,-1)

図 18 「OnePlaneCuttingController.cs」の変更

(6)「Assets」中の OBJ のアイコンを選択して、右側の「Inspector」にある Import Setting の Materials の中で、Import Materials の Location の項目を、「Use External Materials (Legacy)」を選択し、オブジェクトの Shader 設定を変更した(図 19)。



図19 オブジェクトの Shader 設定の変更

(7) 左側の「Hierarchy」中で OBJ データに関連した 子オブジェクトのすべてを選択し、[Assets]-[Cross Section Shader]-[Scripts]フォルダの中にあるスクリ プト「OnePlaneCuttingController」を右側の「Inspector」 にドラッグ&ドロップした。うまくいかない場合は、 個別にドラッグ&ドロップした(図 20)。



図 20 子オブジェクトへの断面スクリプトの設定

(8)子オブジェクトの全てを選択した状態で、左側の「Hierarchy」中の「Cutting\_plane」を右側の「Inspector」の項目「One Plane Cuttting Controller (Script)」の「Plane」に、ドラッグ&ドロップした(図 21)。



図 21 子オブジェクトの Cuting\_plane の設定

(9) 子オブジェクトのすべてを選択した状態で右側の「Inspector」に表示されるシェーダについて、[CrossSection]-[OnePlaneBSP]を設定した(図 22)。



図 22 オブジェクト shader への OnePlaneBSP の設定

# 6.2 VIVE コントローラーのセットアップ

VIVE のハンドデバイスを利用するために、VIVE コントローラーの設定を行った。

 左側の「Hierarchy」で、[GameObject]-[Create Empty]を選択し、2つのGameObjectを作成した。名 前を、「VRTK\_SDKManager」と「SteamVR\_SDK」 に変更した(図23)。



図 23 VRTK\_SDK 関連の空のオブジェクトの作成

(2) [Assets]-[VRTK]-[Scripts]-[Utilities]-[SDK]の中の 「VRTK\_SDKManager」と「VRTK\_SDKSetup」を、 作成した「VRTK\_SDKManager」と「SteamVR\_SDK」 に各々ドラッグ&ドロップした(図 24)。



図 24 VRTK\_SDK 関連の設定

 (3) 左側の「Hierarchy」にある「[CameraRig]」を 「SteamVR\_SDK」の下に移動した(図 25)。(もし くは、いったん、「[CameraRig]」を削除し、 「SteamVR\_SDK」の下に追加した。)



図 25 「CaemraRig」の設定(移動)

(4) 左側の「Hierarchy」で「SteamVR\_SDK」を選択し、右側の「Inspector」で「VRTK\_SDK Setup (Script)」中の「Quick Selection」のプルダウンを「Fallback」から「SteamVR」に変更した(図 26)。



図 26 VMD での OBJ ファイルのエクスポート

(5) [Assets]-[VRTK]-[Scripts]-[Interactions]の中にある 「VRTK\_ControllerEvents」、「VRTK\_InteractGrab」と 「VRTK\_InteractTouch」を、「[CameraRig]」の下にあ る「Controller(left)」と「Controller(right)」にドラッ グ&ドロップした(図 27)。



図 27 コントローラーの機能の設定

(6) 「Controller(left)」と「Controller(right)」の「Inspector」の項目「VRTK\_Intaract Grab (Script)」の中のプルダウン「Grab Button」を、「Grip Press」から「Trigger Press」に変更した(図 28)。



図 28 コントローラーのトリガーの設定

(7) 左側の「Hierarchy」にある「VRTK\_SDKManager」 を選択し、右側の「Inspector」の項目「VRTK\_SDK Manager (Script)」で、「+」のマークを押して、項目 「Setups」を追加した。そのとき、左側の「Hierarchy」 の「SteamVR\_SDK」を、右側の「Inspector」の項目 「Setups」の「None (VRTK\_SDK Setup)」にドラッグ &ドロップし「SteamVR\_SDK (VRTK\_SDKsteup)」 に変わることを確認した(図 28)。

# 7. おわりに

本稿では、手軽な 3D-VR 入門として、3D プリン タでも用いられる 3D データ(Wavefront OBJ 形式) を VR 空間中に表示する簡単な方法を明らかにし た。HTC VIVE のようなゲーム用 VR 装置を、シミ ュレーションやデータ解析の手軽な結果観察用途に 利用し、本格的な結果観察や多人数議論で大阪大学 の没入型 3D-VR 可視化装置の活用を期待している。

本稿で紹介した方法で、例えば、図 29 のように、 電子顕微鏡(FIB-SEM)で観察したゴム中に分散し たナノ粒子の3次元凝集構造を、自由自在な位置か ら、自由自在な方向の断面を観察することができる。



図 29 電子顕微鏡(FIB-SEM)で観察してゴム中のナノ 粒子の 3 次元構造[14]の断面カット VR 視聴画面の例

今回の検討では、手軽な入門のために、動きのな いコンテンツを対象としたが、本来的に Unity はゲ ーム構築環境であり、動きやゲーム性のあるコンテ ンツの作成が可能である。今後は、動きのあるコン テンツを扱い、如何に見やすく可視化表現するかの 検討も進めていく予定である。

#### 謝辞

本研究は JHPCN 公募課題(課題番号:10-MD01, 11-MD02, 12-MD03, jh130028-NA19, jh140026-MD02, 14-NA28, jh150002-NA01, jh160036-NAH) や、HPCI 公募課題(課題番号:hp130062, hp130122, hp140191) などの実施を通じて得られた知見の一部を活用して いる。また、一連の VR 可視化技術の検討にあたり、 名古屋大学 荻野 正雄准教授、高橋 一郎技官、大阪 大学 下條 真司教授、伊達 進准教授、安福 健祐講 師、海洋研究開発機構 川原 慎太郎博士、東京都市 大学 宮地 英生教授、サイバネットシステム 黒木 勇博士、日立化成 太田 浩司博士には、有益な議論 や助言を得ている。ここに感謝の意を表する。

# 参考文献

- Stein, J: Why Virtual Reality Is About to Change the World. Time http://time.com/3987022/ (2015).
- (2) 杉本真樹: VR/AR 医療の衝撃, ボーンデジタル (2017).
- (3) 日本バーチャルリアリティ学会編: バーチャル リアリティ学,(株)工業調査会 (2010).
- (4) Balo, A. R., Wang, M., Ernst, O. P: Accessible virtual reality of biomolecular structural models using the Autodesk Molecule Viewer. Nature Methods, Vol.14 (2017) pp.1122–1123.
- (5) Norrby, M., Grebner, C., Eriksson, J., Boström, J.: Molecular Rift: Virtual Reality for Drug Designers. J. Chem. Inf. Model, Vol.55 (2015) pp.2475–2484.
- (6) García-Hernández, R. J., Kranzlmüller, D.: Virtual Reality toolset for Material Science: NOMAD VR tools. in Lecture Notes on Computer Science Vol.10324 (2017) pp.309-319.
- (7) Hagita, K., Matsumoto, S., Otha, K.: A study of commodity 3D-VR for computational material sciences. 投稿準備中.
- (8) 萩田克美、松本茂紀、高橋一郎、荻野正雄: Unity を利用したシミュレーション結果の VR 表示に おける可視化処理性能の簡易定量評価法の検 討,投稿準備中.
- (9) https://unity3d.com/
- (10) https://www.unrealengine.com/
- (11) https://www.vive.com/.
- (12) http://vis.cmc.osaka-u.ac.jp/.
- (13) Humphrey, W., Dalke, A. and Schulten, K.: VMD -Visual Molecular Dynamics. J. Molec. Graphics, Vol. 14 (1996), pp. 33-38.
- (14) Hagita, K., Higuchi, T., Jinnai, H.: Super-resolution for asymmetric resolution of FIB-SEM 3D imaging using AI with deep learning. Sci.Rep. 8 (2018) 5877.