

# Mixed Reality による創薬支援システムの開発と応用

坂本 亘<sup>†1</sup> 関嶋 政和<sup>†1</sup>

**概要**：創薬において創薬標的タンパク質や薬候補化合物の立体構造の可視化は重要な役割を果たしている。しかし、現在多くの分子構造描画システムでは本来 3 次元の立体構造を 2 次元のディスプレイで描画している。本来 3 次元であるタンパク質や薬候補化合物の立体構造は、その相互作用や化合物の最適化を考える上で、3 次元で可視化した方がより有用な知見が得られると考えられる。そこで、本研究ではタンパク質や化合物の立体構造を Mixed Reality(複合現実)を実現するデバイスである HoloLens を用いて複合現実で描画するシステムを開発した。HoloLens の性能に起因する表示上のパフォーマンスに課題はあるものの、今後課題を解決していくことで、創薬において本システムは有用なものになると考えられる。

**キーワード**：Mixed Reality, Virtual Reality, 可視化, Structural Drug Discovery, IT 創薬

## 1. はじめに

現在、一つの新薬を開発するためには約 30 億ドルと 10-12 年間の期間が必要とされている[1]。そのため、これらのコストを削減する為の情報技術を活用した創薬支援(IT 創薬) システムの需要が高まっている[2][3]。一方、タンパク質の立体構造情報は近年急速に充実してきており、タンパク質立体構造情報データベースである PDB の登録データ数は、2006 年からの 10 年間でおよそ 3 倍になった[4]。そのため、近年では立体構造情報を活用した SBDD(Structure-based Drug Discovery) が広く用いられている。SBDD の中には、薬剤標的となるタンパク質に対して効果のある化合物を探すために、標的タンパク質と化合物の立体構造情報を用いて計算機シミュレーションなどを行う手法である Structure-based virtual screening[5] や、一般的な薬剤よりも小さいフラグメント分子から出発し、実験的に決定した結合状態の構造を用いてより薬らしい化合物へと化合物を変化させていく Fragment evolution[6] などが含まれ、いずれも創薬において有用な手法である。これらの手法に共通することは、いずれもタンパク質の立体構造情報の可視化が重要であるという点である。しかし、現在の分子構造描画システムでは、本来 3 次元である分子構造を 2 次元のディスプレイ上に表示させるのが主流である。しかし、技術の向上により、ヘッドマウントディスプレイを用いた Virtual Reality(VR) と呼ばれる仮想現実や、Mixed Reality(MR) と呼ばれる複合現実の表現力が向上している。さらに、近年ではそれらのデバイスを安価で入手し、開発することも可能になってきている。VR や MR の技術を用いることで、現実世界では本来 3 次元であるものを、より現実世界に近い形で見ることが可能になると考えられる。例えば、アメリカの Case Western Reserve University では医学生教育として人体の模型や心臓のホログラムを MR で行っている[7]。分子構造についても同様に、3 次元の構造

は、3 次元で見て、その立体構造や相互作用を確認することが有用であると考えられる。

既存の研究として、VR 向けのヘッドマウントディスプレイである Oculus Rift を対象に開発されたソフトウェアとして Molecular Rift[8] があり、タンパク質や化合物を Oculus Rift 上に表示させたり、Kinect を用いてジェスチャーで操作したりする機能が実現されている。しかし、VR 向けヘッドマウントディスプレイでは視界がふさがるため周囲との議論がしづらく、手元の資料を見るためには一旦ヘッドマウントディスプレイを外さなければならない。また、PC との接続や Kinect が必要で、簡単には移動が出来ないという問題もある。一方で、MR では現実世界にホログラムを重ねるため、VR と違い視界が開けており周囲とコミュニケーションを取りやすく、手元の資料なども同時に見ることができる。また、その中でも HoloLens はホログラムを複数人で共有を可能にするほか、PC と接続することなく単体で動作することができるという特徴がある。本研究では、遠隔での利用、ホログラムの周りを歩きながらタンパク質分子構造の確認などの利用法を想定して HoloLens を対象にした分子構造描画システムを開発した。



図 1 HoloLens

<sup>†1</sup> 東京工業大学  
Department of Computer Science, Tokyo Institute of Technology.

## 2. 開発

### 2.1 開発の概要

本研究では、Microsoft 社の開発している HoloLens(図 1) と呼ばれる MR 用のヘッドマウントディスプレイを対象に開発を進めた。HoloLens に関する詳細なスペックを表 1 に示す。

表 1 HoloLens の基本性能

| 項目        |                                                              |
|-----------|--------------------------------------------------------------|
| スピーカー     | ビルトインスピーカー                                                   |
| 無線 LAN    | Wi-Fi 802.11ac                                               |
| Bluetooth | Bluetooth 4.0 LE                                             |
| OS        | Windows 10                                                   |
| CPU       | Intel 32 bit architecture (1GHz)                             |
| GPU       | Custom built Microsoft Holographic Processing Unit (HPU 1.0) |
| ストレージ     | 64GB Flash                                                   |
| RAM       | 2GB                                                          |
| 電源        | 2.5A / 5.2V                                                  |
| 重量        | 579g                                                         |

開発環境においては、ゲームエンジンには、現状の VR、MR 開発におけるデファクトスタンダードとなっている Unity を用いた。Unity のバージョンは、標準で HoloLens に対応している 5.5 を用いた。

本システムを開発するにあたって、実装した機能は大きくわけてタンパク質の 3 次構造情報ファイルの読み込みと、読み込んだタンパク質の複数の HoloLens での共有である。また、開発にあたって本システムに期待される機能を図 2 にまとめた。実装済みのものは青、今後実装していくものは緑で示した。以下ではそれぞれの実装について述べる。

### 2.2 Universal Windows Platform とそれに付随する開発上の問題について

具体的な実装の記述に先立ち、HoloLens の開発において重要となる Universal Windows Platform(UWP) について述べる。UWP とは、簡単に言えば Windows10 の Windows ストアで配信されているアプリケーションの規格のことである。そして、UWP に対応したアプリケーションは、Windows10 だけでなく、Windows タブレットや Windows Phone、そして Xbox や HoloLens でも動作する。逆に、

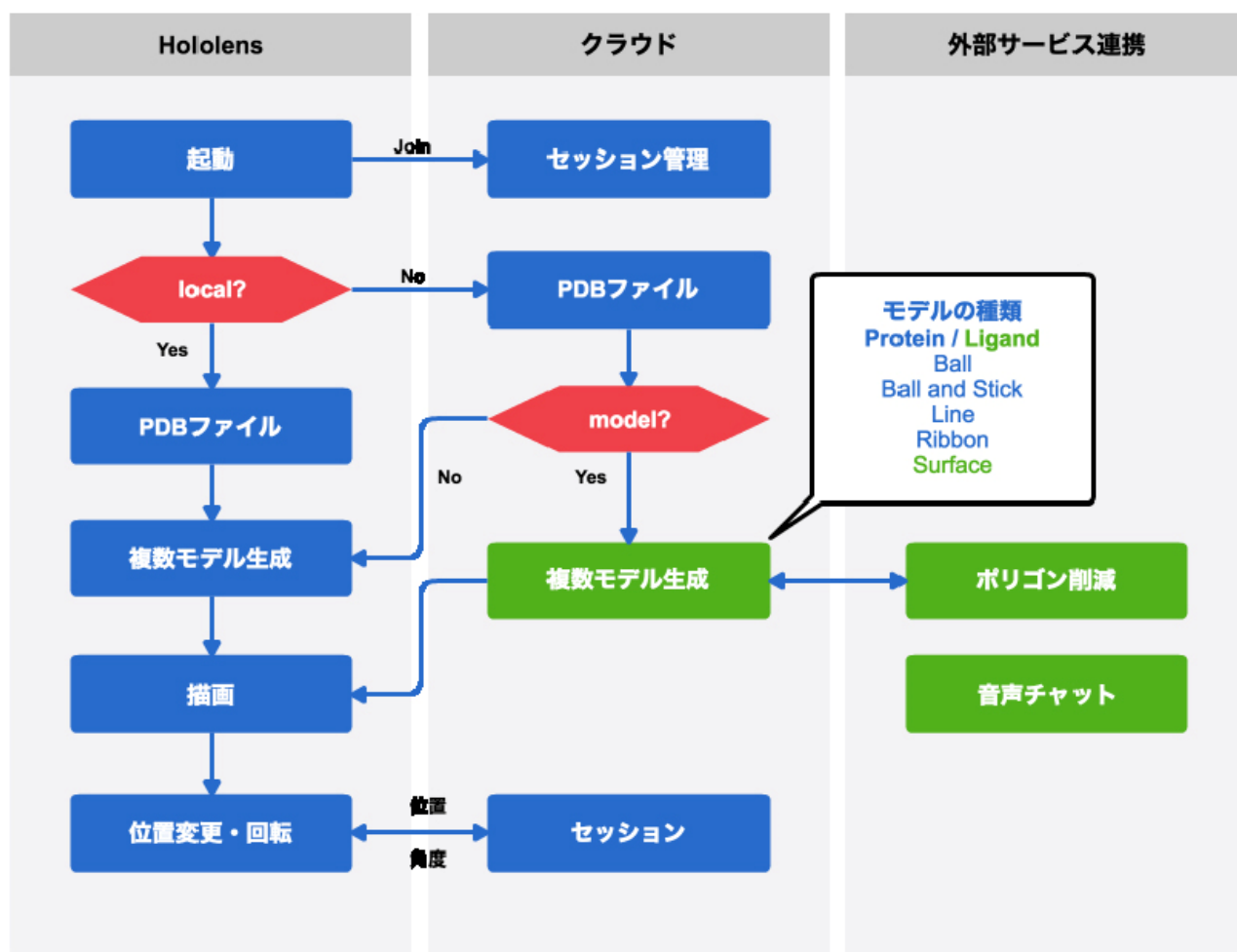


図 2 本システムに期待される機能

HoloLens では従来の Windows 用アプリケーションは動作せず、UWP 向けのアプリケーションしか動作しない。注すべき点としては、UWP は従来の Windows 用アプリケーションとは実行環境が異なることである。UWP では .NET Core Framework をサポートしており、.NET Core Framework ではセキュリティが強化されているため、従来では利用可能であったいくつかの API が使えなくなっている。例えばファイル IO は必ず非同期で書かなければならず、(非同期処理は .NET Framework 4.5 からサポートされたため、.NET Framework 3.5 の Unity 上では動作せず、互換性がない) エンドユーザーの許可も必須になった。また、それぞれのアプリケーションを保護された領域内で動かすため、外部のプロセスを呼べない、レジストリに書き込めないといった問題が存在する。このため、従来では動いていたコードが動かないことが多々あり、本研究においてはこの問題の解決が重要であった。

### 2.3 タンパク質の立体構造情報ファイルの読み込み

本機能を実装するにあたって、(1)PyMol[9] や Jmol[10] などの既存の分子構造描画システムで構造ファイルを読み込み 3 次元モデルに変換してから本システムで読み込む方法と、(2) タンパク質の 3 次元構造ファイルを本システムで直接読み込む方法の 2 つの方法が考えられる。

当初は(1) で実装を進めたが、既存の分子構造描画システムで作成したファイルだと HoloLens で扱うにはポリゴン数が大きくなってしまい、実行速度が大幅に低下することで実用に耐えられなくなってしまった。また、UWP ではセキュリティ上外部プロセスを呼ぶことはできないため、HoloLens から外部プロセスを直接呼び出すことはできない。そのため、本システム実行中に既存の分子構造描画システムを用いて 3 次元モデルを作成するには外部のサーバー上で既存の分子構造描画システムを動作させ、本システムと連携させるためにエンドポイントを用意する必要がある。さらに、3 次元モデル化してから読み込むと、予め 3 次元化したモデルを含めてコンパイルする場合は問題がないが、実行中に読み込むとファイルが大きすぎてしばらく動作が固まったようになってしまった。以上の問題点から、本システムでは(2) のようにタンパク質の 3 次元構造情報記述ファイルを構造解析し、内部で 3 次元モデルを生成する方法を採用した。

タンパク質の 3 次元構造情報の記述のフォーマットとして有名なものとしては PDB や SDF、MOL、MOL2 があるが、本研究では最も一般的に利用される

PDB フォーマットで記述されたファイルを読み込みの対象にした。

#### 2.3.1 PDB ファイルの構造解析

PDB ファイルは基本単位が行(行指向)であり、1 行はヘッダとアノテーションから構成される。ATOM、HETATM、TER、END のヘッダ以外はコメント行であり、ATOM、

HETATM ヘッダは座標に関するデータが記述されており、TER はポリペプチド鎖の終了を表し、END はこの PDB ファイルの終了を表す。その他にも本システムで扱うヘッダとして、HELIX や SHEET、CONECT があり、HELIX は  $\alpha$  ヘリックスをとるアミノ酸の位置が残基番号で示され、SHEET は同様に  $\beta$  シートを表し、CONECT はタンパク質分子ではない原子間の結合が記述されている。

本システムで特に重要なのが ATOM、HETATM で始まる行であり、ATOM はポリペプチドの各原子の座標、HETATM はタンパク質以外(水分子や ATP、金属イオン、結合する低分子など)の座標が記述されている。

本システムでは PDB ファイルを 1 行ずつ END まで読み込み、ヘッダの内容によって場合分けし、それぞれのヘッダに対応したアノテーションの構造を読み込み、ヘッダごとに異なるリストに格納している。また、Unity は左手座標系であるが PDB ファイルは右手座標系なので、X 座標の符号を反転している。あとはその座標に従い、Unity に標準で備わっている基本モデルの 1 つである Cube として原子の種類で色分けして表示している。Sphere ではなく Cube を採用しているのは、Unity の基本モデルの Sphere だと 1 つあたりの三角形ポリゴン数が 768 であるが、Cube を用いることで 12 に抑えることができるためである。これを Chavent[11] らの手法をもちいてシェーダーで加工している。

#### 2.3.2 PDB ファイルの配置場所について

ファイルの配置場所は、予め HoloLens 内に配置した PDB ファイルだけでなく、Microsoft OneDrive 上に配置した PDB ファイルや外部のサーバーに配置した PDB ファイルも読み込みができるようにした。ローカルに配置したファイルの読み込みには従来の .NET Framework で動いていた API が使えないという問題があるが、ランタイム時にリソースを読み込みできる Unity の Resources という機能を使い実現した。

#### 2.3.3 読み込んだタンパク質の共有について

タンパク質の共有では(1)Microsoft がオープンソースで開発している HoloToolkit-Unity[12] 内の Sharing Service を用いるか、(2)Unity5.1 から導入されたマルチプレイヤー用のネットワークシステム UNET を用いるか、(3) 数々の Unity 製ゲームで採用された実績のある外部サービスである Photon Unity Networking(PUN)[13] を用いるかの選択肢がある。

(1) では自分でサーバーを用意しなければならないが、(2) ではそれぞれのデバイスにクライアント、サーバーの役割を持たせることができるのでサーバーレスでも動作し、(3) は SaaS(Software as a Service) 型での利用もサポートしているのでこちらでサーバーを用意する必要がないという特徴がある。今回は実装のしやすさを考え、自前でサーバーを用意する手間はあるものの、(1) の Sharing Service を

採用した。Sharing Service を用いることで、サーバー・クライアント型であることを利用し遠隔でも利用しやすいというメリットがある。例えば、P2P(Peer to Peer) 等で実装する場合、NAT 配下のホスト同士の接続する際に NAT トラバーサル(NAT 超え) について何らかの対処をしなければならない。また、P2P は大学や企業などネットワークによっては制限されているというデメリットもある。サーバーは、Microsoft Azure 上に Windows Server 2016 Datacenter をインストールした仮想マシンを用意し、通信には UDP のポート 20600 番-20604 番を使用した。複数の HoloLens でホログラムを操作できるようにすると、お互いに同時に動かしたりした際にホログラムの座標の更新が頻繁に発生し、ホログラムの位置が正しく定まらない。そこで、本システムでは 1 台のホロレンズを操作用(プライマリ)、他のホロレンズはホログラムの確認用(セカンダリ) に分けている。まず、プライマリのホロレンズで本システムを起動すると、サーバー上にセッションを作成する。次に、セカンダリのホロレンズを起動すると、サーバー上にセッションがあれば加わり、なければ何もしない。このとき、セカンダリのホロレンズではサーバーにセッションを作成することはできない。プライマリのホロレンズでホログラムを動かしたりした際に、サーバーにホログラムの座標と角度を送信し、セカンダリのホロレンズでサーバーからその情報を受け取って描画している。また、HoloLens では毎回アプリケーションを起動した座標が原点の座標となるため、起動するたびに座標系が変わってしまう。当然複数の HoloLens でも異なった座標系となるため、ホログラムを座標で固定したとしても毎回異なった場所にホログラムが配置されてしまう。そこで、HoloLens の空間認識機能(図 3) をもちいて、HoloLens が認識した空間でのホログラムの座標(空間アンカー) も共有することで複数の HoloLens でも同じ場所にホログラムが表示されるようにした。

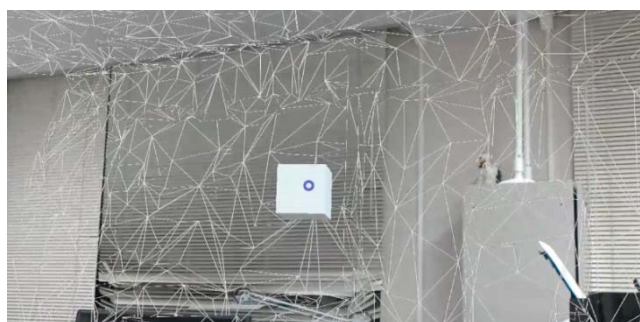


図 3 HoloLens による空間認識

### 3. 本システムの特徴と想定される利用法

#### 3.1 特徴

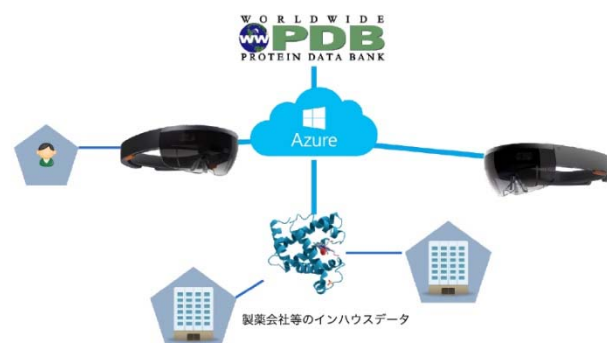


図 4 システムの概念図

本システム(図 4)では、分子を構成する原子の空間配置を示す 3 次元の座標ファイルである PDB ファイルをあらかじめローカル、または Microsoft Azure 上や Microsoft OneDrive 上に配置することで、タンパク質の 3 次元構造の表示を行うことを可能にした。読み込んだタンパク質やリガンドは、自由に動かしたり回転したりすることが可能である。図 5~図 8 は実際のタンパク質(PDB ID: 1FKB) を本システムで表示した様子である。図 5 は Line モデル、図 6 は Ball モデル、図 7 は Ball and Stick モデル、図 8 は Ribbon モデルを表す。Surface モデルは計算に時間がかかる上に、Ball モデルの約 10 倍とポリゴン数が大きくなりすぎることから現時点では実用的な表示に対応できていない。

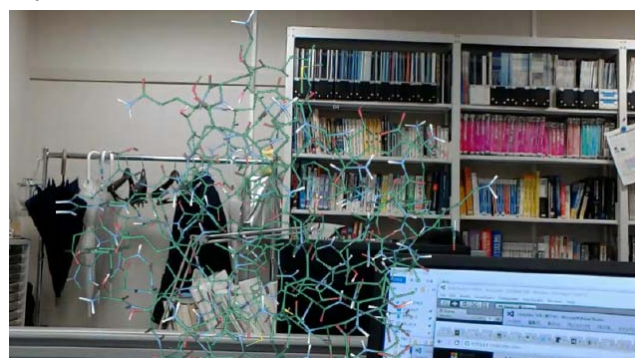


図 5 Line モデル

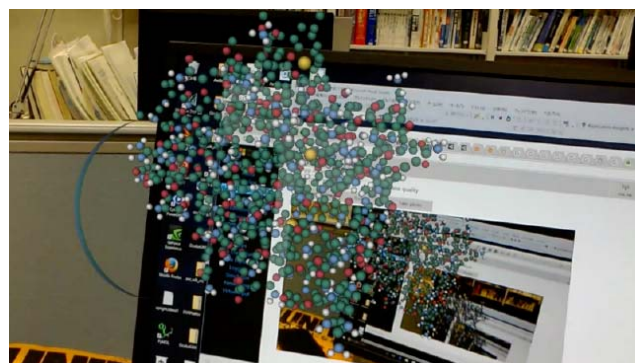


図 6 Ball モデル

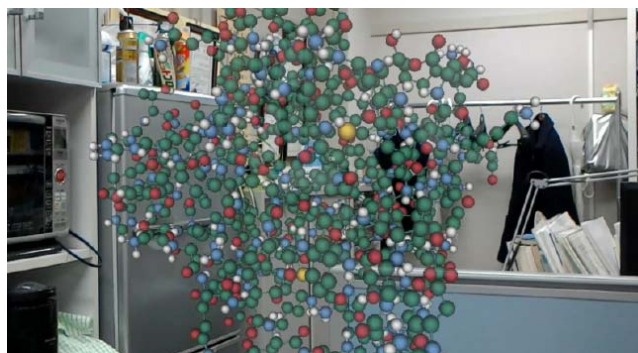


図 7 Ball and Stick モデル



図 8 Ribbon モデル

### 3.2 想定される利用法

#### 3.2.1 対面での利用

本研究では、Mixed Reality を用いてシステムを開発したことで視界がふさがらないために、Virtual Reality を用いた分子構造描画システムでは不可能であったタンパク質やリガンドのホログラムと同時に手元の資料を見ながらの議論が可能である(図9)。また、複数人がホログラムを共有し、周囲から歩き回ることが可能なほか、覗き込みながらタンパク質の立体構造やタンパク質とリガンドの結合様式を理解し、リガンドの最適化についての議論を可能にしている。



図 9 対面での利用

#### 3.2.2 遠隔での利用

製薬企業におけるタンパク質の立体構造やリガンドの情報は社外秘であることが多く、現在は、関係する研究者が1カ所に集まり議論を行うことが多いが、本システムではクラウドと連携する機能を備えており、離れたサイト間でホログラムや他者のホログラムに対しての視点の共有も

行うことが可能である。

## 4. まとめと今後の課題

### 4.1 まとめ

本研究では Mixed Reality を用いた、特に HoloLens を対象とした分子構造描画システムを開発し、3次元のタンパク質の立体構造を3次元のまま可視化することを実現した。さらに、可視化された立体構造を対面もしくは遠隔で共有することを可能にした。

### 4.2 今後の課題

现阶段では、HoloLens のパフォーマンスに課題があるが、タンパク質の Surface を求める計算や、ポリゴン数が大きくなりやすいモデルのポリゴン数を削減するプログラムなどの、HoloLens では処理に時間がかかるような処理はクラウドで処理し、HoloLens の負荷を軽減することができれば創薬において本研究はより有用なものになると考えられる。

HoloLens を快適に動作させるには、1秒あたりに描画されるフレーム数(FPS)を60以上に保つ必要がある[14]。FPSは様々な要因に左右されるが、その中で最も影響が大きいのは描画する3次元モデルのポリゴン数であり、FPSを保つためにはポリゴン数を全体で約1万ポリゴン以下まで落とす必要がある。しかし、今回巨大な分子構造を描画した際に、残基数107と比較的小さいタンパク質やリガンドにおける単純な3次元構造であるBallモデルでもポリゴン数が1万を超える為、動作が重くなっている。また、HoloLens では外部プロセスを呼べないため内部でPDBファイルの読み込みから描画まで行っている。この処理では大量の原子データから巨大な分子モデルを内部で生成、読み込みする必要があるため、同様のデメリットからファイルの読み込みから描画完了までに時間がかかるケースが見られた。この問題も、読み込む3次元モデルのポリゴン数を削減することで改善できると考えられる。

**謝辞** 本研究の一部は、JSPS 科研費 基盤(B) 15H02776 及び JST リサーチコンプレックス推進プログラム「世界に誇る社会システムと技術の革新で新産業を創る Wellbeing Research Campus “Tonomachi”」、AMED 創薬等ライフサイエンス研究支援基盤事業の支援を受けて行われた。

## 参考文献

- [1] Asher Mullard. New drugs cost US\$2.6 billion to develop. Nature Reviews Drug Discovery, Vol. 13, No. 12, p. 877, 2014.
- [2] S. Chiba et al., Identification of potential inhibitors based on compound proposal contest: Tyrosine-protein kinase Yes as a target, Sci Rep, 5, 2015, 17209, <http://doi.org/10.1038/srep17209>.
- [3] 関嶋政和. オープンイノベーションによるIT創薬 コンテスト形式による薬剤候補化合物の探索. Vol. 58, No. 12, 2016 (accessed 2017-05-23).
- [4] RCSB PDB. Yearly growth of total structures, 2017. <http://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content>

=total (accessed 2017-05-23).

- [5] T. Cheng et al., Structure-based virtual screening for drug discovery: a problem-centric review, The AAPS journal, Vol. 14, No. 1, pp. 133-41, 2012.
- [6] D.C. Rees et al., Fragment-based lead discovery, Nature Reviews Drug Discovery, Vol. 3, No. 8, pp. 660-672, 2004.
- [7] Cwru takes the stage at microsoft 's build conference to show how hololens can transform learning. <http://case.edu/hololens/> (accessed 2017-05-23).
- [8] M. Norrby et al., Molecular Rift: Virtual Reality for Drug Designers, Journal of chemical information and modeling, Vol. 55, No. 11, pp. 2475-84, 2015.
- [9] The PyMOL Molecular Graphics System, Version 1.8 Schrödinger, LLC (accessed 2017-05-23).
- [10] Jmol: an open-source Java viewer for chemical structures in 3D. <http://www.jmol.org/> (accessed 2017-05-23).
- [11] M. Chaventet al., GPU-accelerated atom and dynamic bond visualization using hyperballs: A united algorithm for balls, sticks, and hyperboloids. Journal of Computational Chemistry, Vol. 32, No. 13, pp. 2924-2935, 2011.
- [12] Holotoolkitunity, <https://github.com/Microsoft/HoloToolkit-Unity> (accessed 2017-05-23).
- [13] Photon Unity Networking, <https://www.photonengine.com/> (accessed 2017-05-23).
- [14] Performance recommendations for Unity, [https://developer.microsoft.com/ja-jp/windows/mixed-reality/performance\\_recommendations\\_for\\_unity](https://developer.microsoft.com/ja-jp/windows/mixed-reality/performance_recommendations_for_unity) (accessed 2017-05-23).
- [15] Blender Online Community, Blender - a 3D modelling and rendering package, Blender Foundation, Blender Institute, Amsterdam, <http://www.blender.org> (accessed 2017-05-23).

## 付録

### 付録 A ポリゴン数と FPS について

Protein Data Bank[16] からダウンロードした、サイズの小さい PDB ID : 1FKB を用いて、本システムで読み込んだ場合、PyMol で作成したモデルを Unity で読み込んだ場合についてのポリゴン数と、HoloLens で描画した際の FPS についてまとめる。ここでは、PyMol で出力したデータは直接 Unity でうまく読み込めなかったため、PyMol で出力した wrf をオープンソースの 3 次元コンピュータグラフィックスソフトウェアである Blender[15] で fbx 形式に変換してから Unity で読み込んでいる。ball や ball and stick では本システムの方がポリゴン数が少なく、ribbon や surface だと PyMol で描画したモデルの方がポリゴン数が少ない。これは、本システムでは line や ball and stick はポリゴン数 12 の立方体をベースに描画しているのに対して、PyMol のモデルでは一つ一つの ball や stick のポリゴン数が多く、複数のモデルの組み合わせの ball や ball and stick に対しては本システム内で描画したほうがポリゴン数を抑えることが出来る。一方で、ribbon や surface は複数のモデルの組み合わせではなく、それ単体で 1 つのモデルである。こちらに関しては本システムでは無駄なポリゴンを描画してしまっていると考えられる。いずれにせよ、Microsoft が公式に述べている快適な動作環境 (FPS が 60 以上) を満たさないのは課題である。

表 A.2 ポリゴン数

| 種類             | 本システム  | PyMol |
|----------------|--------|-------|
| ball           | 13620  | 69216 |
| ball and stick | 27240  | 95776 |
| ribbon         | 12848  | 8460  |
| surface        | 102176 | 14063 |

表 A.2 FPS

| 種類             | 本システム | PyMol |
|----------------|-------|-------|
| ball           | 25    | 1     |
| ball and stick | 19    | 1     |
| ribbon         | 4     | 59    |
| surface        | 1     | 29    |