# 遠隔 VR 可視化アプリケーション VR-PBVR

ユーザーマニュアル

Ver. 1.0

国立研究開発法人日本原子力研究開発機構システム計算科学センター 2023/03/15

# 目次

更	新履歴	£		. 3
1.	はじ	こめに	-	. 4
	1.1.	動作	=環境	. 5
2.	依有	ミライ	゚゚゚゙゙ブラリ	. 5
	2.1.	VR-	KVS	. 5
	2.2.	CG]	FormatExt4KVS	. 5
	2.3.	assi	mp-5.0.0	. 5
	2.4.	fbx2	20195_fbxsdk_vs2017_win.exe	. 5
	2.5.	free	glut-MSVC-3.0.0-2.mp	. 5
	2.6.	glev	v-2.1.0-win32	. 6
	2.7.	ovr_	_sdk_win_1.30.0_public	. 6
	2.8.	cma	ake-3.16.3-win64-x64	. 6
	2.9.	img	rui-1.79.zip	. 6
3.	セッ	・トア	<b>'</b> ップ	. 7
	3.1.	Visu	ual C++の設定	. 7
	3.2.	Qt (	Creator の設定	. 9
	3.3.	CM	ake	11
	3.4.	KVS	S2.9mod4oculus と依存ライブラリ	11
	3.4.	1.	FreeGLUT	11
	3.4.	2.	GLEW	11
	3.4.	3.	KVS2.9mod4oculus.	11
	3.5.	CG]	FormatExt4KVS と依存ライブラリ	12
	3.5.	1.	Assimp	12
	3.5.	2.	Autodesk FBX SDK	13
	3.5.	3.	CGFormatExt4KVS	13
	3.6.	Ocu	llusPBVR_JAEA と依存ライブラリ	13
	3.6.	1.	Oculus SDK	13
	3.6.	2.	Dear ImGui	14
	3.6.	3.	OculusPBVR_JAEA	14
	3.7.	サー	-バプログラム	16
4.	VR-	PBV	TR の起動	18
5.	VR		での操作	
	5.1.		<b>『パネル</b>	
	5.1	1	時間ステップの設定 エラー! ブックマークが定義されていません	_

5.2.	オフ	゛ジェクトの操作	. エラー! ブックマークが定義されていません。
5.2	.1.	平行移動	エラー! ブックマークが定義されていません。
5.2	.2.	拡大縮小	エラー! ブックマークが定義されていません。
5.2	.3.	回転	エラー! ブックマークが定義されていません。
6. Oc	ulusF	PBVR_JAEA の終了	
6.1.	クラ	イアントの終了	24
			エラー! ブックマークが定義されていません。

# 更新履歴

版	更新日	更新内容
1.0	2023/03/15	(初版)

# 1. はじめに

本書は、日本原子力研究開発機構システム計算科学センター(CCSE)で開発した粒子ベースの遠隔 VR 可視化アプリケーション VR-PBVR の、インストールおよび使用方法を説明したマニュアルである。CCSE はこれまでに粒子ベースのクライアント・サーバ型遠隔可視化アプリケーション PBVR(CS-PBVR)を開発してきた。VR-PBVR は CS-PBVR を VR 可視化向けに拡張したものである。

CS-PBVR は京都大学小山田研究室で開発された PBVR (Particle Based Volume Rnderring) 法、および、可視化ライブラリ KVS (https://github.com/CCSEPBVR/KVS)をベースにして開発されたものであり、遠隔地にあるサーバ上の大規模ボリュームデータの高速な遠隔可視化を実現している。 CS-PBVR は、遠隔サーバ上で動作する Particle Sampler とユーザ PC 上で動作する Particle Renderer で構成される。Particle Sampler はストレージ上のボリュームデータに対して可視化用粒子データを生成し、ソケット通信を介して接続された Particle Renderer に粒子データを転送する。Particle Renderer は受信した粒子データからボリュームレンダリング画像を生成し、粒子の再生成無しに視点変更が可能である。

VR-PBVR は CS-PBVR の Particle Renderer をヘッドマウントディスプレイ (HMD)Oculus 向けに拡張したものである。VR-PBVR の Particle Renderer は粒子データから両眼視用の画像を生成し、HMD に送出する。VR-PBVR はヘッドトラッキングによる視点変更に加えて、VR 用コントローラーOculus touch による可視化データの座標変換が可能である。また、VR 空間上の GUI によるタイムステップの制御やバウンディングボックスの表示を制御できる。そして、CS-PBVR の可視化パラメータ編集用 GUI が利用可能であり、多変数向けの伝達関数を編集できる。

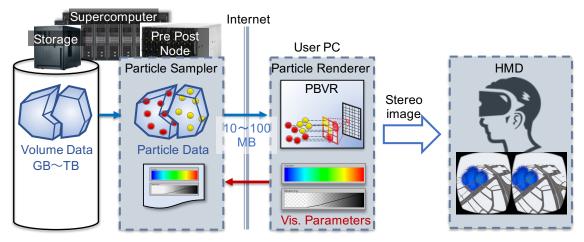


図 1.1-1 VR-PBVR の構成

#### 1.1. 動作環境

VR-PBVR は以下の環境で動作を確認している。

	必要スペック
CPU	Intel i3-6100 / AMD Ryzen 3 1200 / AMC FX4350
GPU	NVIDIA GTX 1050Ti / AMD Radeon RX 470
メモリ	8GB RAM
HMD	Oculus Rift (CV1) / Oculus Rift S / Quest / Quest2
os	Windows 10 (Home/Pro) / Windows 11 Pro
コンパイラ	Visual Studio 2017 / Visual Studio 2019
開発環境	Qt Creator 5.9 / Qt Creator 6.5

# 2. 依存ライブラリ

VR-PBVR のビルド・実行には以下のライブラリが必要である。それらのソースコードは github(https://github.com/kawamuratakuma/VRKVS)

および CCSE の Web ページ(<a href="https://ccse.jaea.go.jp/software/VRKVS/">https://ccse.jaea.go.jp/software/VRKVS/</a>) からダウンロード可能である。

#### 2.1. VR-KVS

VR-KVS は VR-PBVR 向けに修正された KVS であり、HMD Oculus 向けに基本的な可視化機能を提供する。

#### 2.2. CGFormatExt4KVS

CGFormatExt4KVS は 3DS 形式および FBX 形式のポリゴンデータを VR-KVS 形式のポリゴンデータに変換するライブラリである。

#### 2.3. assimp-5.0.0

Assimp は3DS形式のポリゴンデータの読み込み機能を提供するC++ライブラリである。 これは CGFormatExt4KVS のビルド時に静的リンクされるため、実行時には不要である。

# 2.4. fbx20195\_fbxsdk\_vs2017\_win.exe

これは Autodesk 社が提供する FBX 形式のポリゴンデータの読み込み機能を提供するライブラリ FBX SDK のインストーラである。FBX SDK は CGFormatExt4KVS 及び VR-PBVR のビルドと実行に必要である。

#### 2.5. freeglut-MSVC-3.0.0-2.mp

これは Windows 用の GLUT である OpenGLUT のバイナリパッケージであり、VR-KVS 及び VR-PBVR のビルドと実行に必要である。

# 2.6. glew-2.1.0-win32

これは Windows 用の GLEW バイナリパッケージであり、VR-KVS 及び VR-PBVR のビルドと実行に必要である。

# 2.7. ovr\_sdk\_win\_1.30.0\_public

Oculus SDK は VR-PBVR のビルドに必要である。

# 2.8. cmake-3.16.3-win64-x64

Assimp のビルドに cmake 3.16 以上が必要である。

# 2.9. imgui-1.79.zip

Dear imgui は HMD 上で 2D GUI を生成するオープンソース C++ライブラリであり、 VR-PBVR のビルドに必要である。

# 3. セットアップ

本章では、VR-PBVR をコンパイルするための事前設定と手順、及び VR-PBVR の実行に 必要なフィルタ及びサーバプログラムの事前設定につて説明する。これらの操作は最初に 1 回実行するだけでよい。

以下では、作業の起点となるディレクトリを〈BASE\_DIR〉と表記する。

# 3.1. Visual C++の設定

以下では VS2017 での設定方法について記述するが、VS2019 でも同様である。

始めに Visual C++ 2017(Community Edition 以上が必要)を Microsoft の Web サイト (https://visualstudio.microsoft.com/ja/vs/older-downloads/)からダウンロードし、インストールする。次に Visual Studio Installer を起動し、「詳細▼」→「変更」(図 3.1-1)を選択する。



図 3.1-1 Visual Studio Installer の画面(1)「詳細▼」→「変更」

コンポーネントの選択画面で、以下のコンポーネントをインストールする(マイナーバージョンは違ってもよい)。

- (A) 「ワークロード」タブで選択
  - (1) C++ によるデスクトップ開発(図 3.1-2)
- (B) 「個別のコンポーネント」タブで選択
  - (1) Windows 10 SDK (10.0.17763.0) (図 3.1-3)
  - (2) デスクトップ C++ 用 Windows 10 SDK (10.0.16299.0)[x86 および x64] (図

3.1-3

- (3) CMake の Visual C++ ツール(図 3.1-4)
- (4) VC++ 2017 version 15.9 v14.16 latest v141 tools (図 3.1-4)

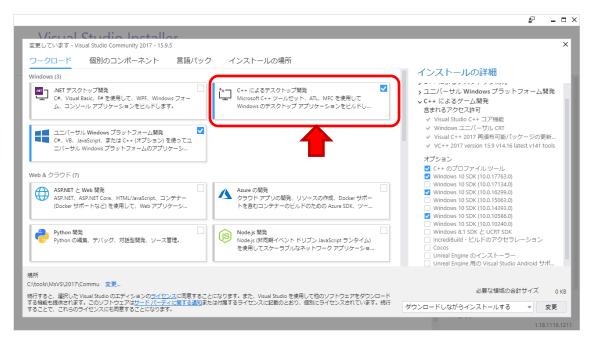


図 3.1-2 Visual Studio Insteller の画面 (2) コンポーネント選択 - C++によるデスクトップ開発



図 3.1-3 Visual Studio Installer の画面 (3) コンポーネント選択 - Windows SDK



図 3.1-4 Visual Studio Installer の画面(3)コンポーネントの選択 - cmake と VC++ tools

コンポーネントの選択後、右下の「変更」ボタンを押下しインストールする。

#### 3.2. Qt Creator の設定

Qt 公式 Web サイトから Qt オープンソース版のインストーラー(Qt Online Installer <a href="https://qt.io/download-open-source">https://qt.io/download-open-source</a>)をダウンロードして実行する。

Select categories において、「latest release」と「Preview」のチェックを外し、「LTS」のみチェックされた状態にする(図 3.2-1)。

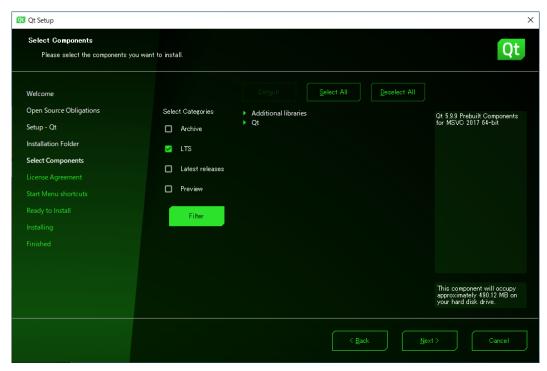


図 3.2-1 Qt インストーラ画面:コンポーネントの選択(その1)

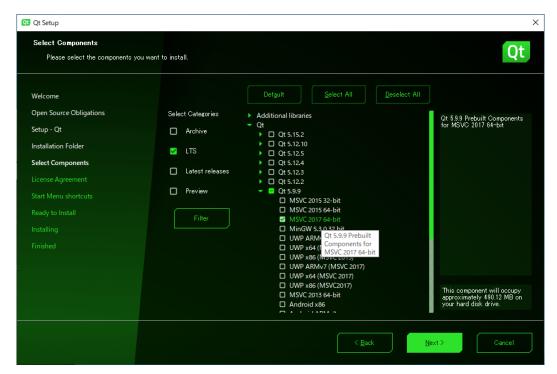


図 3.2-2 Qt インストーラ画面:コンポーネントの選択(その2)

続いて、「► Qt」をクリックし、下表のコンポーネントを選択する(図 3.2-2)。

· Qt > Qt 5.9.9 > MSVC 2017 64-bit

- Qt > Qt 5.9.9 > Sources
- · Qt > Developer and Designer Tools > Qt Creator 4.14.0
- · Qt > Developer and Designer Tools > Qt Creator 4.13.2 CDB Debuger Support
- · Qt > Developer and Designer Tools > Debugger Tools for Windows
- · Qt > Developer and Designer Tools > CMake 3.18.3 64-bit
- · Qt > Developer and Designer Tools > ninja 1.10.0

#### 3.3. CMake

cmake-3.16.3-win64-x64.zip を「cmake-3.16.3-win64-x64」フォルダに解凍し、それを「<BASE\_DIR>¥tools¥」フォルダに移動する。

## 3.4. KVS2.9mod4oculus と依存ライブラリ

#### 3.4.1. FreeGLUT

freeglult-MSVC-3.0.0-2-mp.zip を「freeglut」フォルダに解凍し、それを「<BASE\_DIR>¥lib¥」フォルダに移動する。「freeglut¥lib¥x64¥」フォルダ中の「freeglut.lib」を「<BASE\_DIR>¥lib¥」フォルダにコピーする。

ユーザー環境変数「KVS\_GLUT\_DIR」に値「<BASE\_DIR>¥lib¥freeglut」を設定する。

#### 3.4.2. GLEW

glew-2.1.0-win32.zip を「glew-2.1.0」フォルダに解凍し、それを「<BASE\_DIR>¥lib¥」フォルダに移動する。「glew-2.1.0¥lib¥Release¥x64¥」フォルダ中の「glew32.lib」および「glew32s.lib」を「<BASE\_DIR>¥lib¥」フォルダにコピーする。

ユーザー環境変数「KVS\_GLEW\_DIR」に値「<BASE\_DIR>¥lib¥glew-2.1.0」を設定する。

#### 3.4.3. KVS2.9mod4oculus

ユーザー環境変数「KVS\_DIR」に値「<BASE\_DIR>¥lib¥kvs」を設定する。KVS はこのフォルダにインストールされる。

KVS2.9mod4oculus.zip を「KVS2.9mod4oculus」フォルダに解凍し、それを「<BASE\_DIR>」フォルダに移動する。VS2017用 x64 Native Tools コマンドプロンプト(以下「Tools コマンドプロンプト」)を起動する。Tools コマンドプロンプト上で「<BASE\_DIR>¥KVS2.9mod4oculus」フォルダに移動し、以下のコマンドを実行する。

nmake

nmake install

ユーザー環境変数「Path」に値「<BASE\_DIR>¥lib¥kvs¥bin」を追加する。

# 3.5. CGFormatExt4KVS と依存ライブラリ

#### 3.5.1. Assimp

assimp-5.0.0.zip を「assimp-5.0.0」フォルダに解凍し、それを「<BASE\_DIR>¥lib¥」フォルダに移動する。

Tools コマンドプロンプトで「<BASE\_DIR>¥lib¥assimp-5.0.0」フォルダに移動し、以下のコマンドを実行する。

```
cd assimp-5.0.0

SET CMAKE_BIN=<BASE_DIR>\text{\text{tools}\text{\text{cmake}}}-3.16.3-\text{win64}-x64\text{\text{bin}\text{\text{cmake}}}.exe

SET SOURCE_DIR=.

SET GENERATOR=\text{Visual Studio 15 2017}

SET BINARIES_DIR=\text{"./BINARIES/x64"}

SET CMAKE_GENERATOR=\text{Visual Studio 15 2017}

SET CMAKE_GENERATOR_INSTANCE=C:\text{\text{Program Files (x86)}\text{\text{\text{Microsoft Visual}}}} 

Studio\text{\text{2017}\text{\text{Community}}}

\text{\text{CMAKE_BIN\text{\text{CMake}_GENERATOR_INSTANCE\text{\text{\text{"-D CMAKE}_GENERATOR=\text{\text{"\text{CMAKE}_GENERATOR}}}} 

\text{\text{RINSTANCE=\text{\text{\text{\text{\text{GENERATOR}_INSTANCE\text{\text{\text{\text{\text{\text{\text{CMAKE}_GENERATOR}}}}} 

\text{\text{\text{CMAKE}_BIN\text{\text{\text{\text{\text{\text{CMAKE}_GENERATOR}}}}} 

\text{\text{\text{\text{CMAKE}_BIN\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\t
```

コマンド実行後に、

<BASE\_DIR>¥lib¥assimp-5.0.0¥include¥assimp¥config.h にコピーする。

ビルドが成功すると「assimp-5.0.0\BINARIES\\x64\cde」フォルダ下の「Release」、「Debug」フォルダに、以下の assimp のライブラリファイルが作成される。

- Release
  - > assimp-vc141-mt.lib
  - ➤ assimp-vc141-mt.dll
- Debug
  - ➤ assimp-vc141-mtd.lib
  - > assimp-vc141-mtd.dll
  - > assimp-vc141-mtd.pdb

以下の2つのユーザー環境変数を設定する。

変数名	変数値
ASSIMP_INC_DIR	<pre><base_dir>¥lib¥assimp-5.0.0¥include</base_dir></pre>
ASSIMP_LIB_DIR	<pre><base_dir>¥lib¥assimp-5.0.0¥BINARIES¥x64¥code¥Release</base_dir></pre>

#### 3.5.2. Autodesk FBX SDK

Autodesk\_FBX\_Review\_Win\_64bit.exe を用いて<FBX\_SDK\_DIR>に FBX SDK をインストールする。そして以下の 2 つのユーザー環境変数を設定する。

変数名	変数値
FBX_SDK_INC_DIR	<pre><fbx_sdk_dir>¥include</fbx_sdk_dir></pre>
FBX_SDK_LIB_DIR	<pre><fbx_sdk_dir>¥lib¥vs2017¥x64¥release</fbx_sdk_dir></pre>

## 3.5.3. CGFormatExt4KVS

CGFormatExt4KVS.zip を「CGFormatExt4KVS」フォルダに解凍し、それを「<BASE\_DIR>¥lib」フォルダに移動する。

「CGFormatExt4KVS」 フォルダにある「kvsmake\_libs.vc.conf\_template」を「kvsmake\_libs.vc.conf」にリネームし、「FBX\_SDK\_DIR」と「ASSIMP\_DIR」にそれぞれライブラリのパスを設定する。このパスは半角スペースを含むことがあるため""で括ること。

Tools コマンドプロンプトで「<BASE\_DIR>¥lib¥CGFormatExt4KVS¥Lib」フォルダに 移動し、以下のコマンドを実行する。

kvsmake -g LibCGFormatExt4KVS kvsmake lib

コマンド実行が成功すると、「<BASE\_DIR>\\ Lib\\ CGFormatExt4KVS\\ Lib\\ フォルダに「LibCGFormatExt4KVS.lib\\ が作成される。

環 境 変 数 「 CGFORMAT\_EXT4KVS\_SHADER\_DIR 」 に 、 値 「<BASE DIR>¥lib¥CGFormatExt4KVS¥Lib」を設定する。

# 3.6. OculusPBVR\_JAEA と依存ライブラリ

#### 3.6.1. Oculus SDK

「<BASE\_DIR>¥lib」フォルダ内に「ovr\_sdk\_win\_1.30.0\_public」フォルダを作成する。ovr\_sdk\_win\_1.30.0\_public.zip を解凍して、その中にあるファイルとフォルダを全て「<BASE\_DIR>¥lib¥ovr\_sdk\_win\_1.30.0\_public¥」フォルダ内にコピーする。

以下の2つのユーザー環境変数を設定する。

変数名	変数値
OCULUS_INC_DIR	<pre><base_dir>\frac{\text{Plib\formal}}{\text{cond}} \text{SASE_DIR} \text{SIDE} \text{SIDE} \text{Cond} \text{SIDE} \text{Cond} Co</base_dir></pre>
OCULUS_LIB_DIR	<pre><base_dir>¥lib¥ovr_sdk_win_1.30.0_public¥Lib0VR¥Lb¥Windows</base_dir></pre>
	¥x64¥Release¥VS2017

#### 3.6.2. Dear ImGui

imgui-1.79.zip を「imgui-1.79」フォルダに解凍し、それを「<BASE\_DIR>¥lib」フォルダに移動する。imgui-1.79\_cmakefiles.zip を「imgui-1.79」フォルダに解凍し、それを「<BASE\_DIR>¥lib¥imgui-1.79」フォルダに移動する。

Tools コマンドプロンプトで「<BASE\_DIR>¥lib¥imgui-1.79」フォルダに移動し、以下のコマンドを実行する。

cmake -G "Visual Studio 15 2017" -A x64 .
cmake --build . --config release
copy Release\[ \text{libimgui.lib} \] .
copy examples\[ \text{Release\[ \text{libimgui\_impl\_opengl3.lib} } \] .

コマンド実行後、「<BASE\_DIR>¥lib¥imgui-1.79」フォルダに「libimgui.lib」と「libimgui\_impl\_opengl3.lib」が生成される。

# 3.6.3. OculusPBVR\_JAEA

OculusPBVR\_JAEA.zip を「OculusPBVR\_JAEA」フォルダに解凍し「<BASE\_DIR>¥」フォルダに移動する。

# 3.6.3.1. プロジェクトの設定

プロジェクトファイル「<BASE\_DIR>¥OculusPBVR\_JAEA¥QtClient¥QtClient.pro」を QtCreator で開く。プロジェクトの設定画面において左サイドバーの"Buld & Run"の下に表示されている"Desktop Qt 5.9.9 MSVC2017 64bit" > "Build"をクリックし「ビルド設定」を表示する(図 3.6-1)。

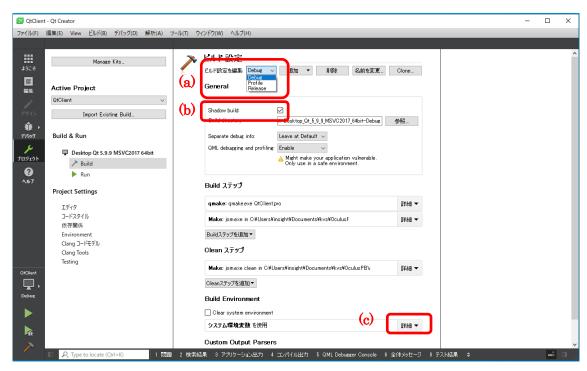


図 3.6-1 「Desktop Qt 5.9.9 MSVC2017 64bit」をクリックした直後の画面

# (1) ビルド設定を編集

"Edit build configuration"(図 3.6-1 (a))において「Release」を選択する。

# (2) General > Shadow Build

「General > Shadow Build」 (図 3.6-1 (b))をチェックする。

# (3) Build Environment

"Build Environment"の「詳細」((図 3.6-1 (c)))をクリックし、以下の環境変数を追加する。

	<del>-</del>
Variable	Value
(変数名)	(変数値)
GLEW_DIR	<base_dir>¥lib¥glew-2.1.0</base_dir>
KVS_SOURCE	<base_dir>¥KVS2.9mod4oculus</base_dir>
IMGUI_DIR	<base_dir>¥lib¥imgui-1.79</base_dir>

# 3.6.3.2. コンフィギュレーション

SETTINGS.pri ファイル内の DEFINES 変数の値で VR-PBVR の構成を設定できる。

#### (1) ミラー表示

VR-PBVR は HMD の画像をディスプレイ上の GUI にミラー表示する。DEFINES のデフォルト値は(ア)である。

- (ア) DEFINES += MIRROR\_IMAGE\_BOTH\_DISTORTED 両目画像(歪みあり)を ミラー表示する。
- (イ) DEFINES += MIRROR\_IMAGE\_BOTH 両目画像(歪みなし)をミラー表示する。
- (ウ) DEFINES += MIRROR\_IMAGE\_LEFT\_ONLY 左目画像のみをミラー表示する。
- (エ) DEFINES += MIRROR\_IMAGE\_RIGHT\_ONLY 右目画像のみをミラー表示する。

#### (2) オブジェクト操作

Touch コントローラは可視化オブジェクトを掴み、その動きでオブジェクトを座標変換する。その掴む動作は Touch コントローラの HandTrigger または IndexTrigger に紐付けられている。DEFINES のデフォルト値は (ア) である。

- (ア) DEFINES += GRAB\_USING\_HAND\_TRIGGER 中指ボタン(HandTrigger)に掴む動作を定義する。
- (イ) DEFINES += GRAB\_USING\_INDEX\_TRIGGER人差し指ボタン(IndexTrigger)に掴む動作を定義する。

# (3) デバッグ出力

標準出力に Scene, Screen, TouchController クラスの処理ログを出力する。

- (ア) Scene クラス: "DEFINES += DEBUG\_SCENE"
- (イ) Screen クラス: "DEFINES += DEBUG\_SCREEN"
- (ウ) TouchController クラス:"DEFINES += DEBUG\_TOUCH\_CONTROLLER"

# 3.6.3.3. ビルド

メニューバー > 「ビルド」 > 「プロジェクト "QtClient"をビルド」

#### 3.7. サーバプログラム

CS-PBVR は大規模データを効率よく並列処理するために領域分割するフィルタプログラムと、サーバ上でボリュームデータを可視化用粒子に変換する Particle Sampler、そしてユーザ PC 上でボリュームレンダリングする Particle Renderer で構成されている。 VR-PBVR は CS-PBVR の Particle Renderer に関する VR 拡張であり、フィルタプログラムやParticle Sampler は CS-PBVR と共通である。詳細は

CS-PBVR のマニュアル (https://ccse.jaea.go.jp/software/PBVR/) を参照すること。

#### 4. VR-PBVR の起動

VR-PBVR は CS-PBVR と共通のサーバを利用する。本章は VR-PBVR の Particle Renderer の起動方法を記述する。サーバの利用方法は CS-PBVR のマニュアルを参照すること。

VR-PBVR はプロジェクトファイルから起動される。"Buld & Run" > "Desktop Qt 5.9.9 MSVC2017 64bit" > "Run" から開かれる「実行時の設定」画面からコマンドライン引数が入力できる。VR-PBVR は左サイドバー下のボタン、もしくはメニューバーから実行される。コマンドライン引数として以下のオプションが利用可能である。

- · -vin (可視化データファイル名を指定する)
- · -tf (伝達関数ファイル名を指定する)
- · -lefty (左利き用設定で実行する)
- -cgmodel <FBX file or 3ds file> (FBX ファイルもしくは 3ds ファイルを 表示する)
- ・ その他、CS-PBVR と同じオプションが利用可能である。

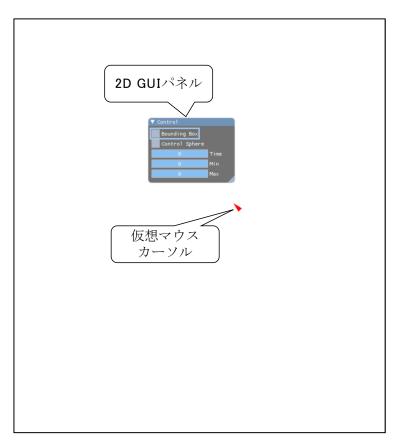


図 3.7-1 起動直後の制御パネルとマウスカーソル

実行直後の3D空間内には、レイオーバーで表示される制御パネルとそれを操作するための仮想マウスカーソルが描画される(図 3.7-1)。

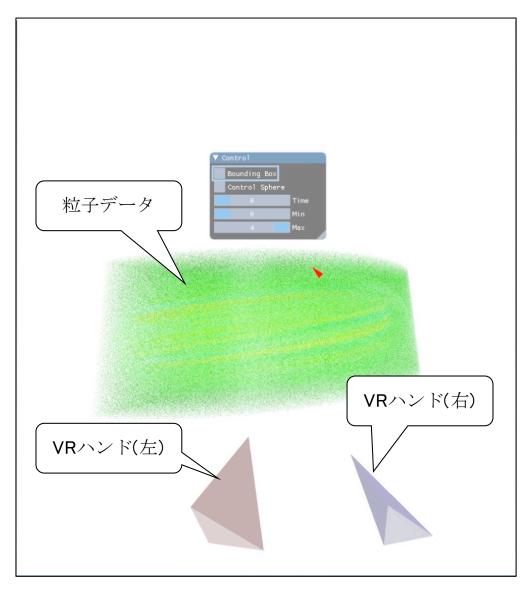


図 3.7-2 可視化データと VR ハンド

レンダラーはサーバから粒子データを受信し、可視化データと VR ハンドを描画する(図 3.7-2)。

# 5. VR 空間での操作

# 5.1. 制御パネル

制御パネルは赤い三角形のマウスカーソルで操作できる。右利きモードの場合、マウスカーソルは Touch Controller の左スティックで移動され、A ボタンでクリックされる(図 5.1-1)。左利きモードの場合、マウスカーソルは右スティックで移動され、X ボタンでクリックされる(図 5.1-2)。



図 5.1-1 右利きモード("-lefy"オプションなし)のマウスカーソルの操作



図 5.1-2 左利きモード(-lefty"オプションあり)のマウスカーソルの操作

制御パネル(図 5.1-3) は以下の機能を提供する。

- (1) バウンディングボックスの表示 On/Off の切り替え(図 5.1-4)
- (2) 制御球の表示 On/Off の切り替え (図 5.1-5)
- (3) 時間ステップの設定
  - (a) 表示ステップの指定
  - (b) 最小ステップの指定
  - (c) 最大ステップの指定

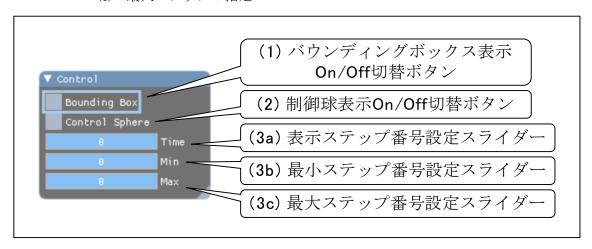


図 5.1-3 GUI パネルの操作項目

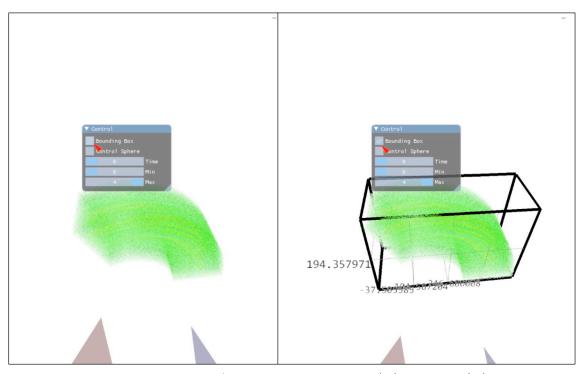


図 5.1-4 バウンディングボックス表示 Off(左)と表示 On(右)

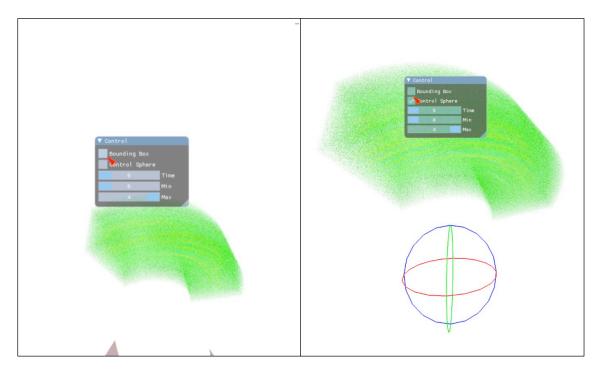


図 5.1-5 制御球表示 Off(左)と表示 On(右)

時間ステップについて、一番上のスライダーが表示ステップの設定、次が表示可能なステップの下限、一番下が表示可能なステップの上限である。制御パネルの時間ステップとディスプレイ上の GUI は連動している(図 5.1-6)。

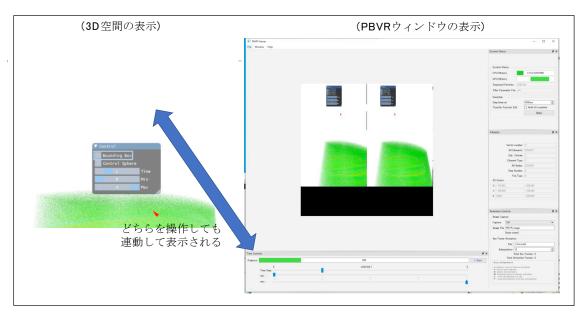


図 5.1-6 VR 空間とディスプレイ上の GUI

# 5.2. 座標変換

オブジェクトは Touch コントローラーのトリガーボタンで掴まれ、ジェスチャーで座標変換される。トリガーボタンは初期状態で HandTrigger (図 5.2-1)である。



図 5.2-1 トリガーボタンの位置

両手を同じ方向に動かすジェスチャーでオブジェクトが並行移動する(図 5.2-2)。



図 5.2-2 平行移動

両手を離す、あるいは近づけるジェスチャーでオブジェクトが拡大あるいは縮小する(図 5.2-3)。

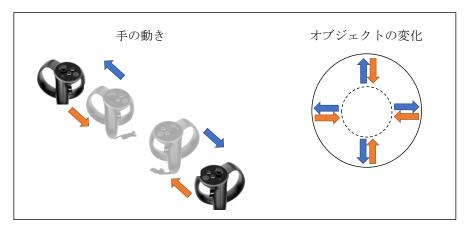


図 5.2-3 拡大縮小

両手を回すジェスチャーでオブジェクトが回転する (図 5.2-4)。

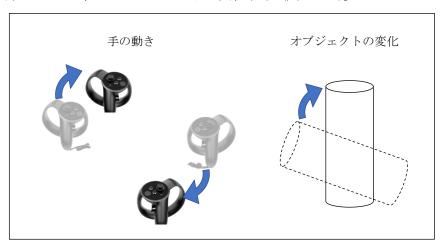


図 5.2-4 回転の操作方法

# 6. VR-PBVR の終了

# 6.1. クライアントの終了

クライアントはディスプレイ上のメインウィンドウの左上の×ボタンで終了する。サーバはターミナル上から Ctrl+C を入力して終了する。