



VRTRIX Data Glove Unity SDK Tutorial



Date	Modified by	Comments
2018-10-22	Guo	Init Version.
2019-01-28	Guo	Add Interaction Part
2019-03-10	Guo	Add Motion Capturing Integration Part
2019-06-13	Guo	Update Manual For VRTRIX Data Glove Pro
2020-06-14	Guo	Support multiple gloves

简介

Introduction

VRTRIX™ 数据手套通过遍布全手的高性能 9 轴 MEMS 惯性传感器实时采集各指头关节运动数据，并通过反向动力学还原骨骼运动，可以在虚拟现实的场景中实现对真实手部运动的重现，并进行精细的手部运动还原和交互。每只手套上分布有 6/7/11 个传感器，双手共 12/14/24 个，可以实时高精度低延迟输出全手所有关节的运动姿态。

VRTRIX™ 数据手套提供 Unity SDK 以供开发者在 Unity 平台下接入数据手套硬件进行驱动。支持模型动作实时渲染，手势识别，虚拟现实支持，虚拟现实环境下的交互以及与全身动捕的整合。开发者可以通过 Unity SDK 中开放的 API 接口和场景示例工程，与原有虚拟现实项目进行整合，或者与全身动捕设备进行整合。

VRTRIX™ 数据手套 Unity SDK 目前包含有 5 个示例场景：

- VRTRIXGloveDataStreaming (主要展示了对数据手套硬件数据流的获取与实时渲染，同时展示了一些有用的硬件参数的读取以及对手套硬件的控制，例如电池电量，无线连接信号强度，如何触发震动反馈等。)
- VRTRIXGloveGestureRecognition (主要展示了数据手套手势识别的功能)
- VRTRIXMultipleGloves (主要展示了如何在一台 PC 上同时驱动多副手套，或者通过 tcp 以以太网通信驱动多副手套的功能)
- VRTRIXGloveVR (主要展示了如何在 VR 环境下实现数据手套的数据流传输，以 HTC tracker 为腕部追踪设备情况下。)
- VRTRIXGloveVRInteraction(主要展示了如何在 VR 环境下实现手套与虚拟物体的交互)。

系统要求

System Requirements

- Unity 2017.1.0 及以上版本，目前最新测试兼容 Unity 2019.3。
- Windows 7 SP1 及以上版本。

开发准备

Prepare for Development

1. 请首先确保数据手套服务端软件（VRTRIXGloveServer）已经成功安装，且数据手套已经成功连接，HTC Tracker 腕部追踪器也已经成功设置（如果需要开发 VR 应用）。如果还未进行该操作，请按数据手套操作手册安装软件和操作后再进行下面的步骤。
2. 对于希望在 VR 环境下运行数据手套的用户，需要先配置好 VR 环境，本 Unity SDK 默认采用 HTC Vive 头盔，以及 SteamVR 开发环境。具体请见：<https://www.vive.com/cn/support/>
3. 数据手套 Unity SDK 以 .unitypackage 包形式进行分发，最新版 SDK 会在 https://github.com/VRTRIX/VRTRIXGlove_Unity3D_SDK/releases 上进行更新，请关注以获取最新 SDK。

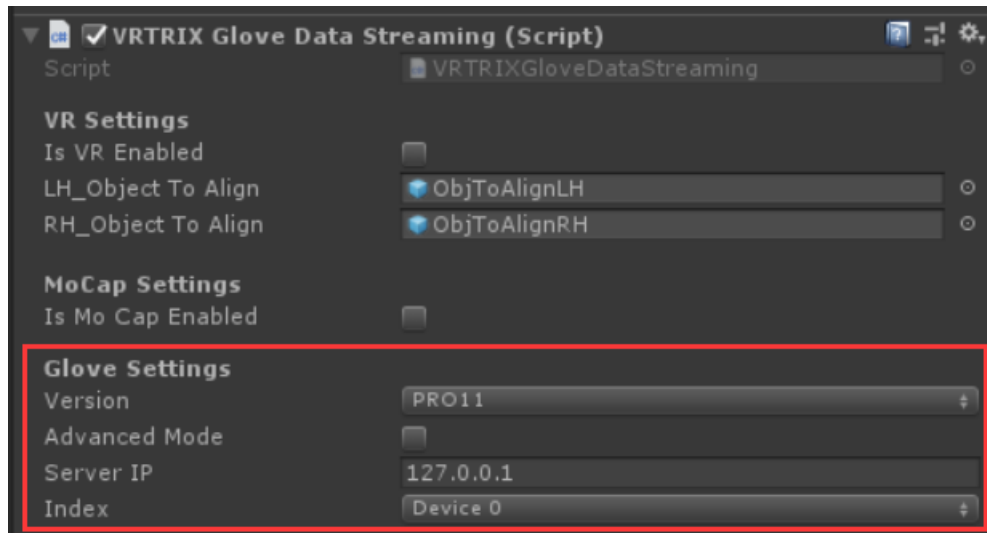
快速开始

Quick Start

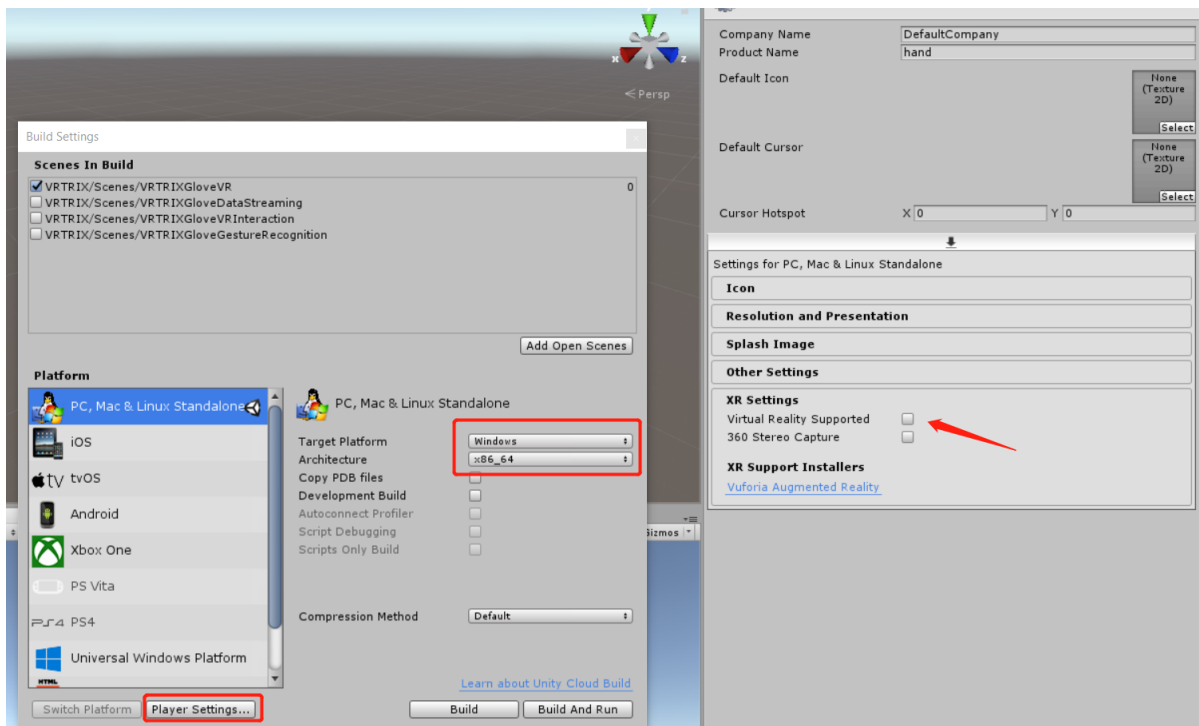
1. 运行数据手套服务端软件：双击运行 VRTRIXGloveServer.exe 服务端软件。
2. 导入数据手套 SDK 包：新建一个 Unity 工程将数据手套 SDK 包导入。
3. 运行 Demo 场景：运行 VRTRIXGloveDataStreaming 场景，点击 UI 左左侧的 Connect 即可和手套进行连接，就能看到 3D 手部模型实时渲染动画了。



4. **预制体导入：**新建一个场景，然后在预制体文件夹 VRTRIX/Prefabs 中找到名为 VRTRIXDataGlove 的预制体，将其拖拽至新的场景中。然后修改脚本参数：数据手套硬件版本，服务端 IP 地址，以及设备 ID。服务端 IP 地址为运行着数据手套服务端软件的 PC 在局域网中的地址，如果是本机运行则无需修改，使用默认的 127.0.0.1 即可。设备 ID 当服务端同时连接多副数据手套的时候需要选择，如果只连接一副则无需修改，使用默认的 Device0 即可。然后运行场景，就能看到 3D 手部模型实时渲染动画了。



5. **VR 环境使用：**在预制体文件夹 VRTRIX/Prefabs 中找到名为 VRTRIXDataGlove_VR 的预制体，将其拖拽至场景中。同样地，如果需要修改服务端 IP 和硬件设备 ID，在脚本属性栏中修改即可。随后修改 Unity 生成配置：File->Building Setting->Player Settings->XR Settings 中 Virtual Reality Supported 勾选，如下图：



随后就可以运行场景，在 HTC Vive 头盔中显示出实时渲染的手部动作。

示例场景

Demo Scenes

1. **VRTRIXGloveDataStreaming:** 主要展示了对数据手套硬件数据流的获取与实时渲染，同时展示了一些有用的硬件参数的读取以及对手套硬件的控制，例如电池电量，无线连接信号强度，如何触发震动反馈等。



该 Demo 程序基于 TCP 协议构建，所以该 Demo 程序不仅可以在本地 PC 连接到手套，也可以在同一局域网内的远端 PC 连接到手套并获取手套的数据流，只需要修改窗口中的 IP 地址为连接着数据手套接收器的 PC 的 IP 地址即可。Hardware Version 代表数据手套硬件版本，可以在 DK1/DK2/Pro/Pro7/Pro11 之间切换。Device ID 代表连接到服务端 PC 的手套 ID，如果该 PC 只连接了 1 副手套，那么使用默认的 Device0 即可，如果有多副手套，则依次选择对应的 ID 号。

上图右侧显示数据手套的状态栏，从上到下分别为数据手套无线通信信号强度示意，校准程度（可理解为磁场干扰状态）示意，电池剩余电量百分比，以及每秒数据手套发送的数据包帧率。

UI 功能介绍如下：

- **Connect/Disconnect:** 连接手套硬件/断开连接手套硬件。
- **Trigger Haptics:** 触发手套掌心震动，持续 500ms。
- **Reset:** 重置手套方向，重置方向由所选择的 LHObjToAlign 和 RHObjToAlign 控制。
- **Advanced Panel:** 将打开高级设置页面，勾选后会出现数据手套算法微调框体（如下图），在该框体中可以对数据手套目前的形态进行微调。



- **Unlock Yaw:** 数据手套航向角解锁，勾选后可以解锁数据手套五指航向张开的功能，但是注意该功能开启后需要手套远离磁性物体，包括但不限于铁质物品，手机，电脑机箱，音箱等。
- **Finger Spacing:** 当无需五指航向张开功能时，可以固定手指之间的夹角，可根据不同模型进行微调，角度设置越大即手指张开程度越大。
- **Curved Spacing:** 代表握拳时五指的间距，通常我们希望五指伸直时有一定的夹角，但是五指握拳状态时基本处于并拢状态，该参数用于调整这个状态。

- **Thumb Proximal/Middle/Distal Offset:** 分别代表拇指三个关节的偏差值，可以通过下拉框进行切换，该偏差值主要由于 3D 模型绑定时骨骼和真实骨骼的偏差决定，一般来说只调整 Thumb Proximal Offset 即可使拇指关节正常。
 - **Reset Value:** 用于使拇指所有关节的偏差值复位为初始值。
 - **Align Fingers:** 由于用户手型不同以及传感器的装配偏差，如果感觉使用时手指弯曲程度不一，可以将五指并拢伸直后，点击该按钮进行校准，该校准类似 T-Pose 校准，只是该功能是分别校准左手和右手，方便单人进行操作。
 - **8-Figure Calibration:** 手套在出厂前已经预先做好了地磁校准的工作，可以无需做此操作，但是如果使用地点有磁场干扰，可以使用如下方法再次进行磁场校准：在 demo 连接上手套的情况下，带着手套在空间中缓慢的划“8”字，硬件将会自适应当前地磁场，持续该动作 10-20s，尽量使得在运动过程中手套朝向覆盖空间中所有方向，观察手部形态，如果手部形态已经正常，而后点击 UI 界面左侧的 8-Figure Calibration 按钮将校准结果存入硬件。
2. **VRTRIXGloveGestureRecognition:** 主要展示了数据手套手势识别的功能。UI 左侧按钮含义与以上场景一致。双手中任意一只手触发 剪刀、石头、布三个手势，对应的手势就会着色。
3. **VRTRIXMultipleGloves:** 主要展示同时获取多副手套数据的示例。获取多副手套数据有两种可能性，一种是多副手套（接收器）直接连接同一 PC，在同一主机上运行。另一种可能性是通过以太网 TCP 协议获取连接在其他主机上的手套的数据。这两者可以混用，例如两副手套连接 1 号 PC，而另外两副手套连接 2 号 PC，则我们可以通过该场景在任意一台在同一局域内的 PC 获取所有 4 副手套的数据。如下图：

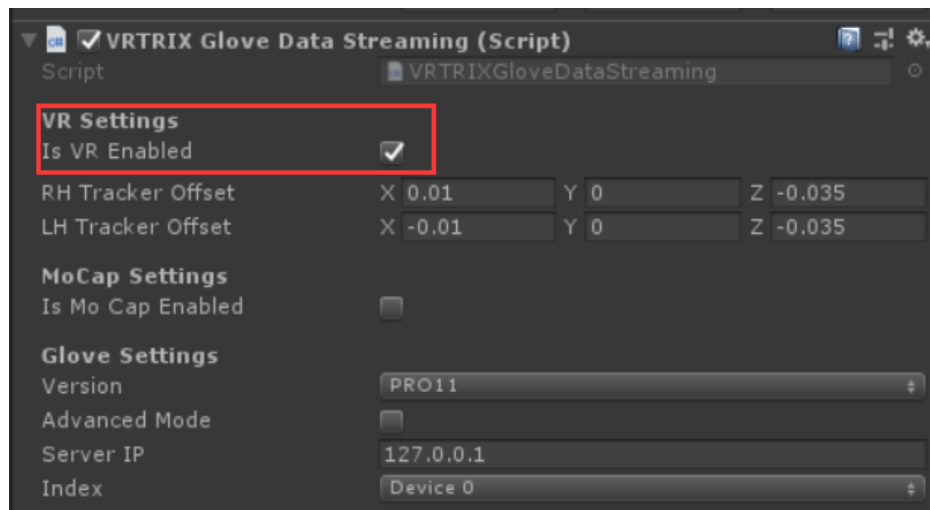


该场景中包含了 6 个 VRTRIXDataGlove 预制体，UI 界面上也有 6 个连接框体，即可以连接 6 副手套，但实际上 6 副手套只是同一 PC 同时连接手套数量的上限，而不是场景可以连接手套数量的上限，所以可以自行添加更多的 VRTRIXDataGlove 预制体，通过其他 PC 服务器获取连接其上的手套数据，同一空间内支持 12 副手套同时工作。

4. **VRTRIXGloveVR:** 主要展示了如何在 VR 环境下实现数据手套的数据流传输，默认以 HTC tracker 为腕部追踪设备。UI 功能基本与 VRTRIXGloveDataStreaming 一致，不再赘述，但是 UI 右上方“Reset”按钮在 VR 环境下功能含义有变化：

- **Reset:** 根据 tracker 方向重置手套方向，VR 环境下进入场景后将手放平前伸，点击 Reset 可以校准 HTC tracker 和手腕的相对位置和手腕的俯仰姿态，手腕姿态校准为平直朝前。

注意在 VR 环境下脚本配置处的 Is VR Enabled 应当勾选，tracker 和手腕的 offset 可以通过参数进行配置。



5. **VRTRIXGloveVRInteraction:** 主要展示了如何在 VR 环境下实现手套与虚拟物体的交互。UI 界面所有功能含义与 VRTRIXGloveVR 场景一致。场景中主要包含以下几大交互部分：

- **传送 (Teleport):** 左右手任意一只手做出传送手势后，会出现一道弧线用于提示传送目的地，选定目的地后松手进行传送。目的地有两种形式：单点或者区域。分别对应预制体：TeleportPoint 和 TeleportArea。单点目的地只能传送到一个特定点，区域目的地可以传送到区域内任意一个点。

- **抓取（Interactable）**：左右手任意一只手触碰到可抓取物体且做出抓取手势后，可以将可抓取物体抓取在手上。松手时默认物体无重力且回归抓取前的位置。对应预制体：Interactable。
- **投掷（Throwable）**：左右手任意一只手触碰到可投掷物体且做出抓取手势后，可以将可投掷物体抓取在手上。松手时默认物体有重力，且以离开手瞬间的速度为初速度进行运动，即投掷行为。对应预制体：ThrowableBall，ThrowableCube。
- **点击（Click）**：左右手任意一只手触碰到 UI 按钮且做出点击手势，UI 按钮的功能就会被触发。

以上所述的手势定义如下：



传送（Teleport）



抓取（Grab）



点击（Click）



OK

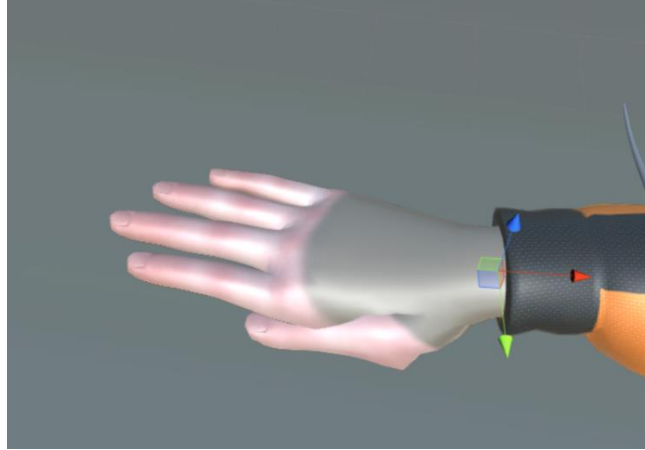
全身动捕整合/模型替换

Integrate with MotionCap Devices/Using Custom Models

1. **全身模型/手部模型要求**：与全身动捕进行整合，第一步就是将数据手套的手部骨骼定义与全身模型的手部骨骼定义相匹配，为了实现最佳效果，全身模型的手部骨骼及模型需满足以下要求：

- 手腕关节及以下所有子关节骨骼轴向一致（即选中每个关节时，坐标系方向相同）。
- 手腕关节及以下所有子关节局部旋转为 0 时，模型姿态应该呈现五指并拢，平伸朝前，且与手背平齐，五指的指甲盖朝上（包含大拇指在内）如下图所示：

注意：如果模型大拇指无法由于蒙皮权重等原因无法满足上述要求，也可以通过调整拇指偏差参数（Thumb Offset）来对数据进行调整，以实现最佳效果，下面会具体阐述操作步骤。



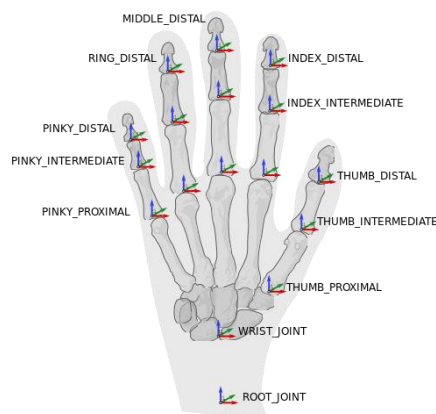
2. 骨骼名称匹配:

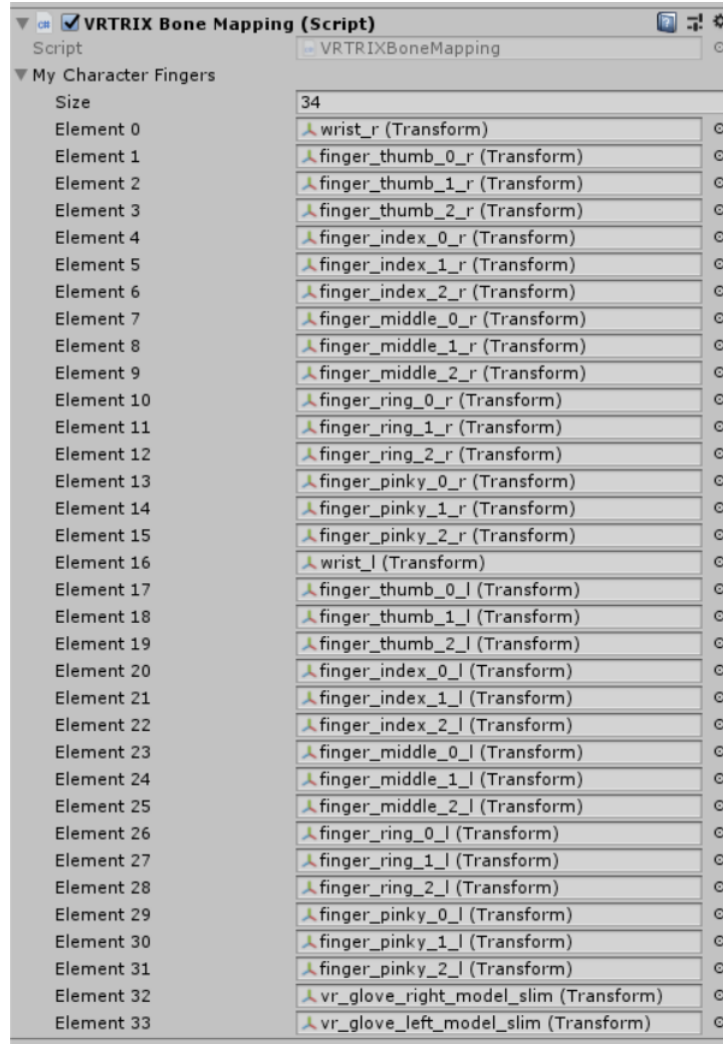
将数据手套定义的手部骨骼名称与全身模型的手部骨骼名称匹配可以通过脚本 **VRTRIXBoneMapping** 提供的接口配置来轻松完成。将模型左右手手部骨骼关节一一拖拽到对应的脚本配置框中，即可完成名称匹配。

匹配时的顺序如下:

右手手腕-右手拇指近端关节-右手拇指中端关节-右手拇指远端关节-右手食指近端关节-右手食指中端关节-右手食指远端关节-右手中指近端关节-右手中指中端关节-右手中指远端关节-右手无名指近端关节-右手无名指中端关节-右手无名指远端关节-右手小指中端关节-右手小指中端关节-右手小指远端关节

左手手腕-左手拇指近端关节-左手拇指中端关节-左手拇指远端关节-左手食指近端关节-左手食指中端关节-左手食指远端关节-左手中指近端关节-左手中指中端关节-左手中指远端关节-左手无名指近端关节-左手无名指中端关节-左手无名指远端关节-左手小指中端关节-左手小指中端关节-左手小指远端关节

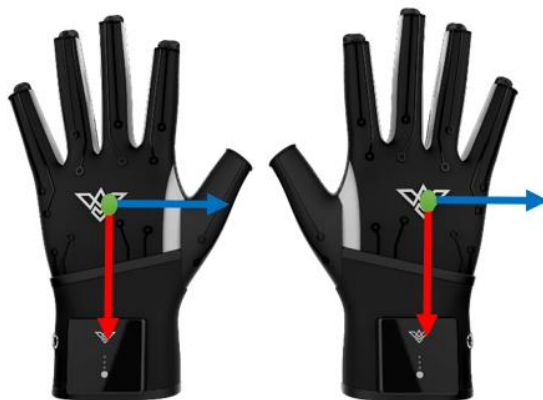




3. 骨骼轴向匹配:

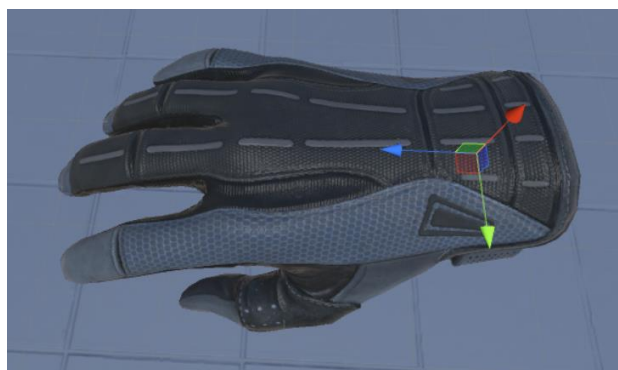
将数据手套定义的手部骨骼轴向与全身模型手部骨骼轴向匹配。手套硬件的坐标系定义如下

图中红色坐标轴为 x 轴，绿色坐标轴为 y 轴，蓝色坐标轴为 z 轴。





模型骨骼的坐标系定义在 Unity 中选中进行查看：



两者对比可以看出模型的 x 轴（红色轴）对应为手套硬件定义下的 y 轴，即 $(0, 1, 0)$ ，模型的 y 轴（绿色轴）对应为手套硬件定义下的 -z 轴，即 $(0, 0, -1)$ ，模型的 z 轴（蓝色轴）对应为手套硬件定义下的 -x 轴，即 $(-1, 0, 0)$ 。所以我们在脚本的配置中填入上述矩阵：

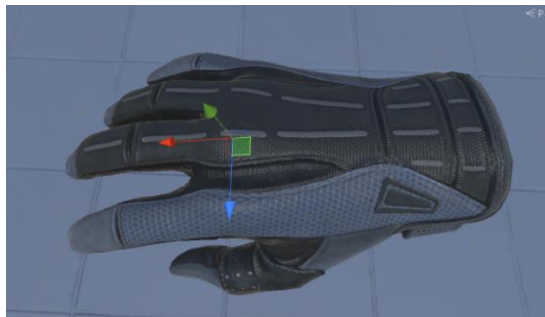
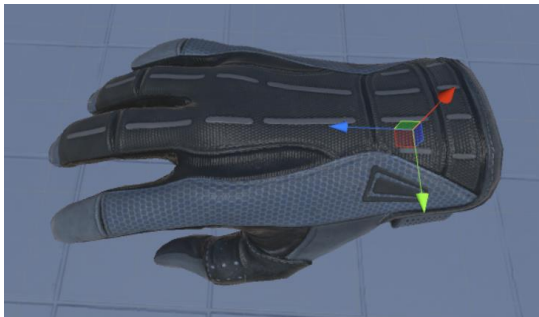
Model Mapping Settings

Ql_modeloffset	X 90	Y 90	Z 0
Qr_modeloffset	X -90	Y -90	Z 0
▼ Ql_axisoffset			
Size	3		
Element 0	X 0	Y -1	Z 0
Element 1	X 0	Y 0	Z 1
Element 2	X -1	Y 0	Z 0
▼ Qr_axisoffset			
Size	3		
Element 0	X 0	Y 1	Z 0
Element 1	X 0	Y 0	Z -1
Element 2	X -1	Y 0	Z 0

同理，左手模型也与硬件定义的坐标系做对比，得到左手的矩阵。

如果遇到有的模型手指的轴向和手腕不一致，可以进一步通过配置里的 `modeloffset` 来调整。

例如该模型手指和手腕坐标系如下：

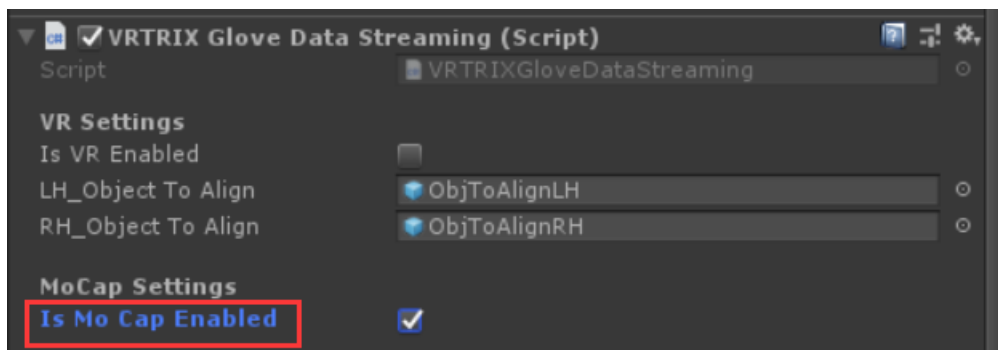


即将手指坐标系统绕 `x` 轴旋转 `-90`，再绕 `y` 轴旋转 `-90` 可以得到腕部坐标系。所以 `modeloffset` 填写 `(-90, -90, 0)`。如果腕部轴向和手指各关节轴向一致，则此处填写 `0` 即可。

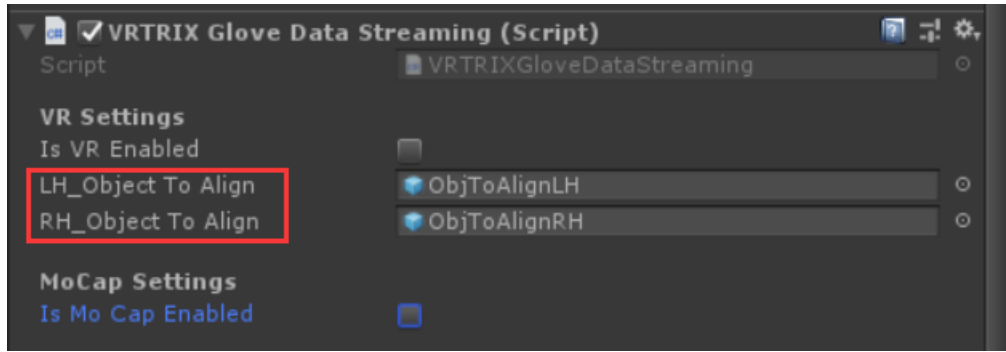
4. 骨骼初始朝向匹配：

与动捕设备结合开始时往往需要做 `T-Pose` 校准，这个匹配步骤目的在于将手套采集的数据与动捕设备采集到的腕部数据对齐，否则手部会感觉和身体动作不一致。

- 如果动捕设备有腕部节点，即可以输出腕部数据的情况下，建议以动捕设备腕部数据为准，此时需要勾选 `IsMocapEnabled`，勾选后手套腕部数据将以动捕设备腕部数据为准，相当于手套仅提供手指数据。



- 如果动捕设备没有腕部节点，即无法输出腕部数据的情况下，则不勾选 `IsMocapEnabled`，而是将脚本中的对齐物体 (`LH_ObjectToAlign` 和 `RH_ObjectToAlign`) 改为全身模型中手腕的父节点（即肘关节）即可。这样手套在连接全身动捕数据后需要穿戴动捕服演员保持 `T-Pose` 然后点击 UI 左侧的 `Reset` 按键，也可以将手腕与动捕数据父关节对齐。该对齐操作只在初始 `T-Pose` 情况下进行一次即可。



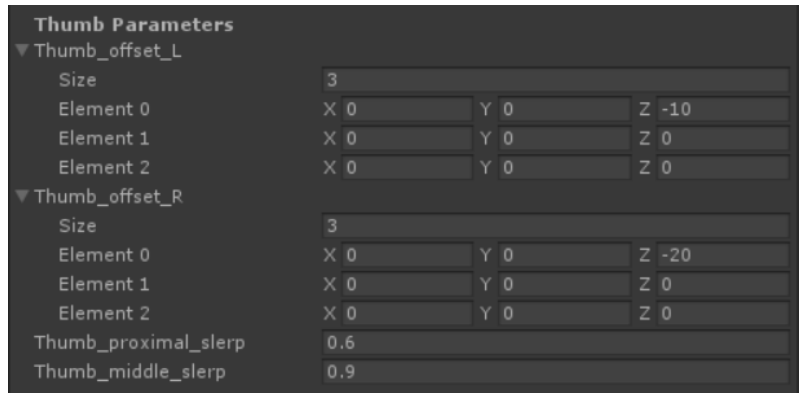
5. 拇指关节调整

由于每个人手指长度和穿戴关节位置不尽相同，手套硬件传感器贴装位置也会有微小差异。

为达到最佳效果，可以在 Unity 插件中脚本的设置框中对拇指数据进行微调。

微调参数包括拇指三个关节的 **offset**，即在原始数据的基础上加上一个旋转偏移量以匹配用户手型和模型。此参数根据实际情况适当调整即可。

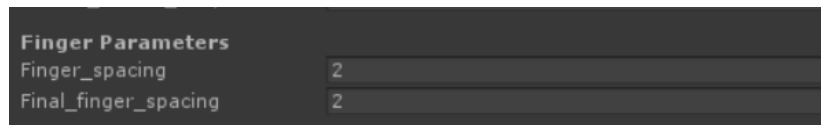
微调参数还包括了拇指插值率这一参数，例如拇指近端关节（**thumb_proximal**）插值率可以在 0 到 1 之间调整，越接近 1 则该关节越接近末端关节的旋转，反之越接近 0 则该关节越远离末端关节的旋转，可以根据模型实际情况调整。



6. 四指间隙调整

在四指航向角没有解锁的情况下，四指之间的角度是固定的，该角度可以通过参数进行调整，原则上在 0-8° 之间进行调整，调整 **Finger_Spacing** 参数即可。

同时如果模型握拳状态下，手指之间仍有一定间隙，那么可以通过调整 **Final_Finger_Spacing** 参数进行调整，该参数越大则握拳时模型手指间隙越小。



至此，所有模型更换的步骤都已经完成，可以运行场景查看追踪效果了。

常见问题

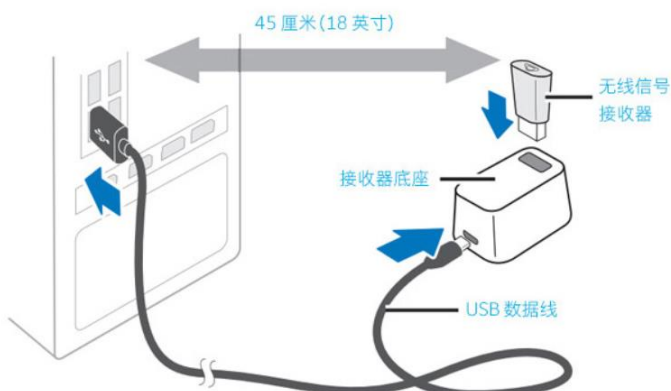
FAQ

1. 为何运行 **Unity Demo** 场景手部动作卡顿或经常断开？

首先确保在运行 Unity SDK 前先安装并运行过服务端软件 VRTRIXGloveServer，如果未安装请参考文档“北京无远弗届数据手套操作手册”先安装客户端软件对硬件驱动进行正确的安装和初始化配置。

如果已经安装和运行服务端软件，依然有卡顿或者断开的情况，将手套 USB 接收器通过 USB2.0 Hub 接出，使得手套 USB 接收器距离电脑 45cm 以上，避免 PC 相邻 USB 口上有大流量的硬件设备例如无线网卡，移动硬盘等，同时避免手套 USB 接收器和手套之间没有过多的遮挡。在正常电磁环境下，手套的正常通信距离应在 10m 以上。

注意：无线信号接收器与电脑之间应保持至少45厘米(18英寸)的距离，并且应放在不会移动的位置。



2. 为何运行 **Unity Demo** 场景点击 **Connect** 后无反应，或者模型不动？

首先确保两个 USB 接收器正常插在 PC 的 USB 接口上，然后确保服务端软件 VRTRIXGloveServer 正常运行，且提示手套处于就绪（Ready）状态。

3. 为何运行 **Unity Demo 3D** 场景正常但是 VR 场景无法连接或者无反应？

首先确保在运行 VR Demo 前运行过客户端配置软件 VRTRIXGloveServer，对 HTC Tracker 进行过配置。未经过配置的 HTC Tracker 无法正确识别手套故而 VR 场景无法连接。未配置请参考文档“北京无远弗届数据手套操作手册”。

如果已经配置过 HTC Tracker，那么请确保 Unity 的虚拟现实支持处于打开状态，具体操作请参考本文档快速开始-VR 环境使用部分。

4. 为何运行 Unity Demo VR 场景时，手部俯仰角度不对？

双手平伸朝前，点击 UI 界面上的 Reset 按钮，就能将手部俯仰角度与 Tracker 对齐。

5. 为何运行 Unity Demo 场景时，手部形态有时出现不正常的跳动或者有所扭曲？

请远离磁性物体，包括电脑系统，铁架桌椅，耳机，手机等磁场辐射源，远离 1m 以上然后戴着手套划八字，八字校准 1 分钟后或看手部形态稳定下来后，点击 UI 左侧的 Hardware Calibrate 对当前磁环境参数进行储存。该操作只需要进行一次，以后在相同环境下无需再进行校准。同时观察 demo 场景中信息栏中是否出现磁铁图标，该图标代表传感器目前所处磁场环境不正常。

6. 为何替换手部模型后，手部形态不正常？

请确认手部骨骼名字是否有对应错误的地方，以及手部骨骼轴向是否有计算错误。如果都没有，请检查模型，是否手部每个骨骼的轴向一致。请认真阅读本文档全身动捕整合部分关于模型替换的说明再试。