

对象设计和集成概述

SteamVR™ Tracking

简介

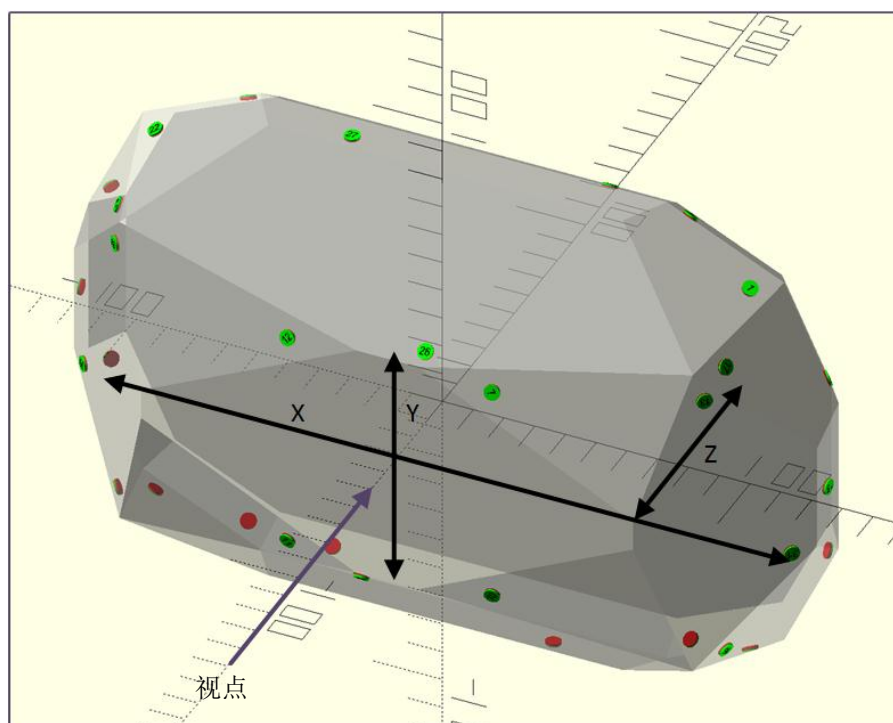
本文档将介绍 SteamVR™ Tracking 定位对象的设计流程。文中参考的各个主题在单独文档中都有详细说明。但是，本文档提供的是一个路线图，以引导设计人员完成从初始方案到与 SteamVR™ 相集成的整个流程。

传感器布置的设计

定义定位对象的形状不仅是流程的第一步，也是定位性能的最重要影响因素之一。开发产品概念时，工业设计师通常会在流程早期就参与进来。这些设计师主要关注产品的人机工学和外观设计，并重视用户体验和美观效果。定位对象的工业设计增添了新的难题，因为对象的大小、形状和表面处理会对定位性能造成直接的影响。

对象的表面用于固定从定位器接收红外参考信号的传感器。务必记住传感器的有效视角为 $\pm 60^\circ$ 。对象的形状和位置必须确保对象满足以下高性能定位条件：

1. 有四个传感器对定位器可见，且不共面。其中一个传感器必须离其他三个组成的平面至少 8mm 以外。
2. 尽可能增加传感器（基线）之间的距离。
3. 所有三个轴中均具有足够的基线。



记住，从定位器的角度来看，传感器可见性和基线会随各个姿势而变。理想情况下，每个方位上都存在四个非共面可见传感器且三个轴均具有足够基线。但是，期望对象在每个姿势下都有理想的可见性和基线是不现实的。成功的对象设计要求作出权衡取舍，最大程度地提高最常用情形中的定位性能。

工业设计师要理解传感器布置的基本规则以融入其设计流程，这一点非常重要。一些常用设计造型，例如平面和直角，会对定位造成不良影响。但是，如果工业设计师将传感器位置的限制视为其设计灵感的种子，他们很可能会想象出完全不同的设计方案，不仅有引人注目的视觉效果，还能实现高性能的定位。有关此主题的更完整的论述，请参阅文档**对象形状和传感器布置**。

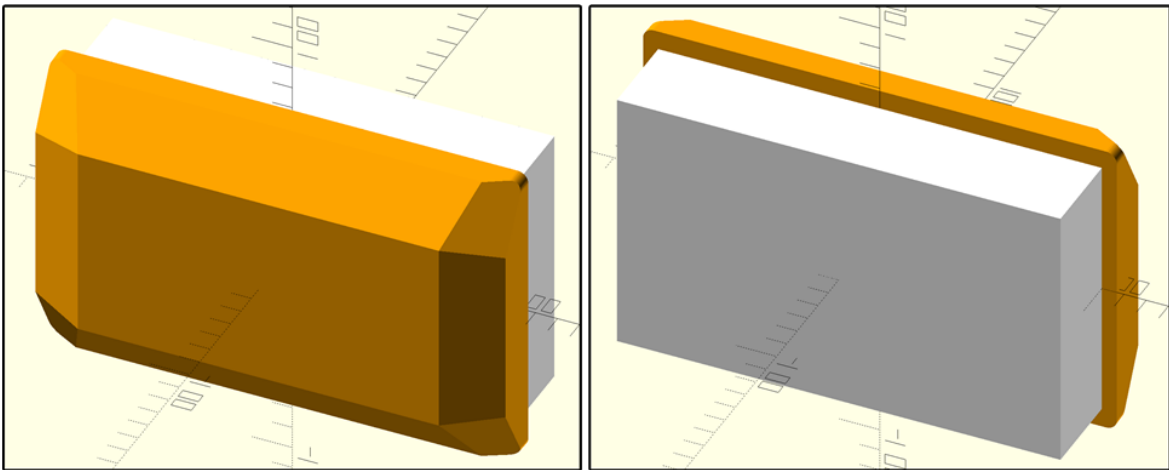
工业设计通常涉及关于材料和表面处理的建议，材料和表面处理会决定对象的外观和感受，同时保护传感器免受损害。但是，覆盖传感器会直接影响定位性能。除了传感器位置，工业设计师和机械工程师还需重视的一点是了解**传感器覆盖**文档中所述的最佳实践。

一旦定义了对对象的形状，务必对可能布置在该表面上的传感器进行此验证，以实现可接受的定位性能。

生成传感器布置

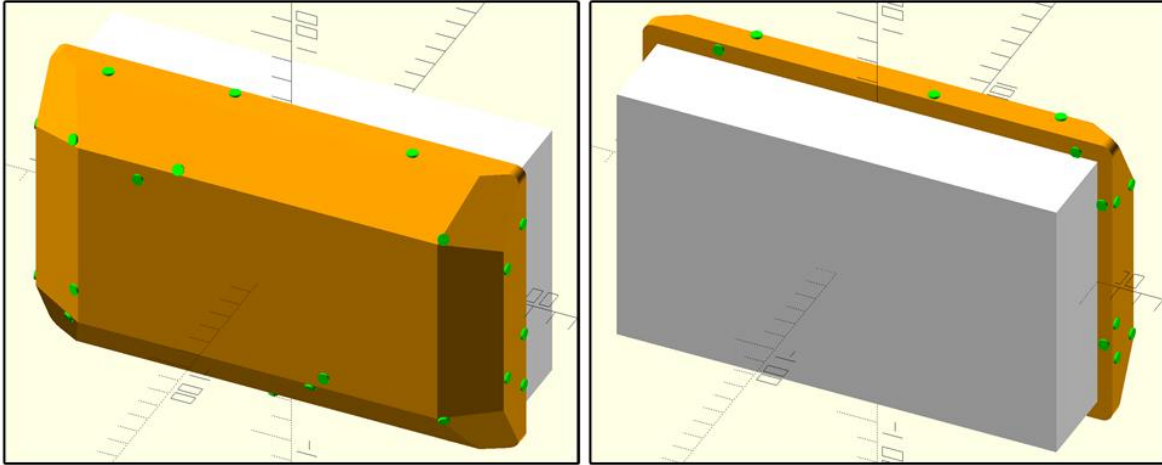
Valve 开发了一种软件工具，可自动将传感器布置在任意形状上，并优化布置以实现最佳定位性能。所有拥有许可证的用户都可以免费使用软件工具 hmd_designer，它包括了很多用于对象设计的有用功能。其中一个最有用的功能是在对象表面上生成传感器布置。Hmd_designer 可接受 STL 格式的 3D 模型。请遵循这些规则来制作 STL 文件，以便帮助 hmd_designer 有效且高效地生成传感器位置：

1. 使用实心形状而非中空壳体。否则，此工具浪费时间地将传感器放到中空形状的内部，仅在它无法找到定位器可见的内部位置时才会把传感器移到外部。
2. 创建 STL 文件来表示无法放置传感器的对象零件，例如把手或安装点。此软件会考虑这些组件所投下的会阻挡定位器所传光线的“阴影”。
3. 创建用于遮盖对象上任何不应放置传感器的表面的 STL 文件。



此软件将传感器布置于表面上，并优化其位置以增强定位性能。首先，用此软件布置全部的 32 个传感器。如果该形状并未通过 32 个传感器产生良好的定位结果，则减少传感器数量绝对无法改善结果。一旦生成了传感器布置，此工具会输出三个文件：

1. 一个 JSON 文件，用于描述各个传感器的位置和方位。此文件是配置的一部分，将上载到设备。
2. 一个 OpenSCAD 格式的 3D 模型文件，用于显示对象以及设备上布置的所有传感器。
3. 一个包含定位模拟数据的文本文件，可显示在二维和三维图上。

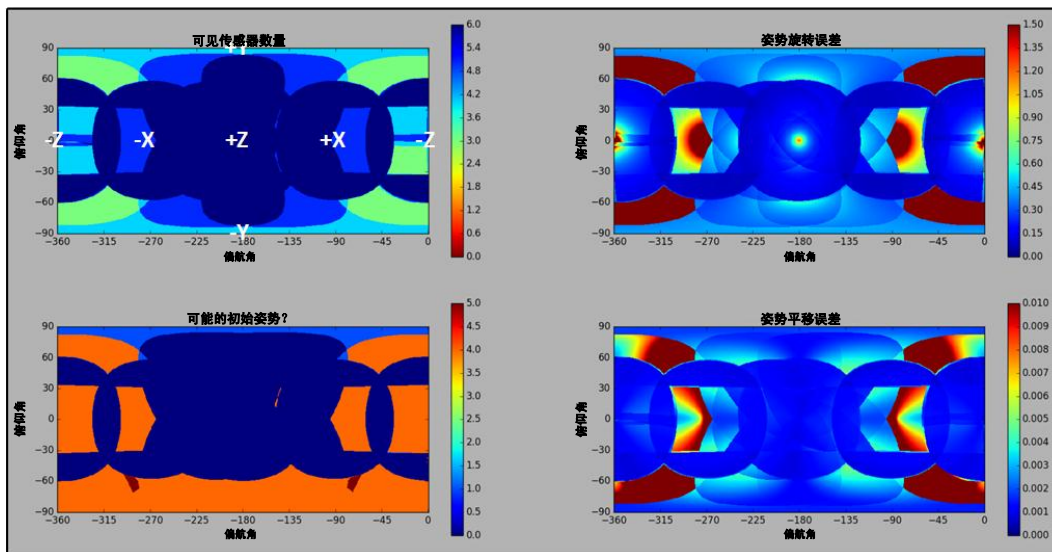


有关详细信息，请参阅**模拟用户手册**中的“生成传感器布置”。一旦生成并模拟了传感器位置，务必查看模拟输出以确定传感器覆盖范围不足的区域，并改进形状以提高定位质量。

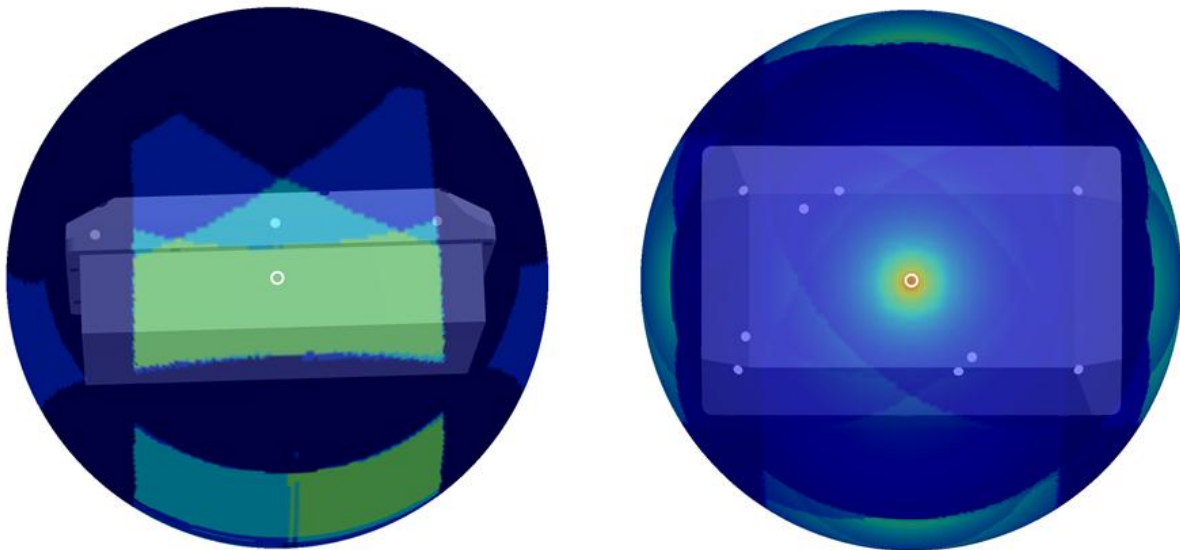
解读模拟输出

使用由 hmd_designer 生成的二维和三维图检查模拟输出，有助于确定要改进传感器布置的区域。尽管二维和三维图显示了相同的数据，但二者用途不尽相同。二维图非常适用于快速查看对象的性能和比较不同设计或设计迭代的相对性能。所有模拟数量和所有方位都一目了然。二维图还可轻松附加到电子邮件和嵌入到演示文稿中，因为它们被渲染为 PNG 文件。

二维图呈现了“可见传感器数量”、“可能的初始姿势？”、“姿势旋转误差”和“姿势平移误差”的模拟数据。这些图直接关系到对象能否满足高性能定位的三个条件。读图时，蓝色表示性能良好，棕色表示性能较差。绿色表示临界性能。二维图均显示为映射到平面上的球体。要了解三个轴如何映射到二维图中需要花费片刻时间。



三维图尤其适用于定位对象上的问题区域。模拟输出形成一个围绕着对象的球体，这样设计师可以调整模拟图方位，以让问题区域始终直接显示于导致该问题的方位之上。借助 3D 查看器，您还可以轻松查看障碍物对定位性能产生的效果。最终，3D 查看器考虑光传感器的 60°视角，并突出显示任何给定姿势下可见的传感器。



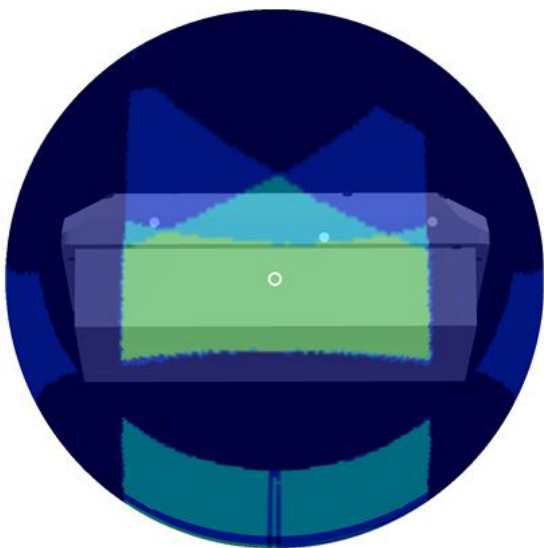
有关详细信息，请参阅文档[解读模拟输出](#)。确定可改进定位性能的区域后，就应该调整对象形状以更好地支持最佳传感器布置了。

改进形状

根据模拟结果来修改定位对象的形状然后再重新模拟来验证性能，这种迭代流程可帮助设计师快速调整对象的设计方案并做权衡取舍。如果模拟输出在特定方位上显示较差或临界结果，应修改形状以改善传感器布置。

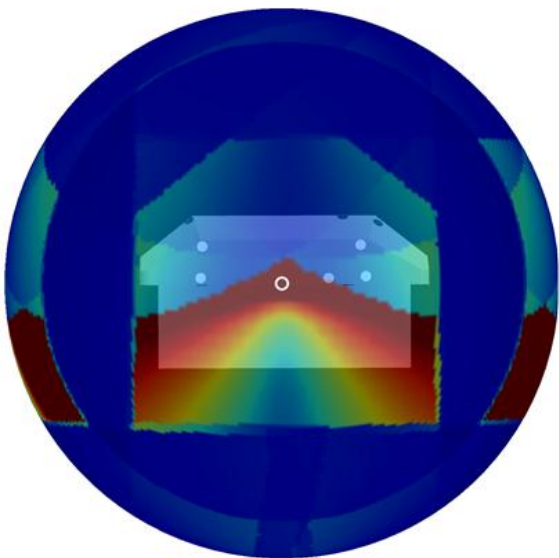
可见传感器

在任意姿势下实现五个可见传感器是一个很好的初始目标。有时，这意味着放开一个在原始设计中被完全遮蔽的表面，以让传感器能够面向其他方向。其他情况下，可能需要向边沿添加小平面，以充分利用传感器的 $\pm 60^\circ$ 视角。



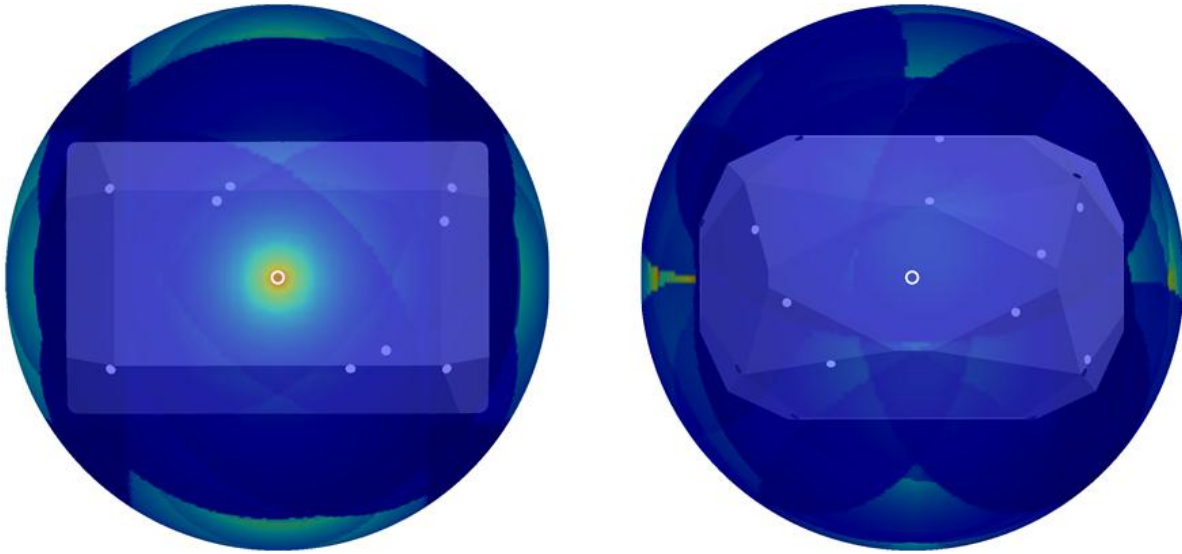
平移误差

如果设计中某些方位上的传感器之间没有足够的基线，可能需要增加对象的宽度。对象越宽，传感器之间的基线越多，从而减少平移误差。请记住，平移误差主要取决于相对与定位器的位移。减少平移误差意味着增大对象相对于定位器的工作距离。



旋转误差

旋转误差是因深度中的基线不足所致。平面之外没有可见传感器是旋转误差的主要导因。可通过添加造型将传感器拉出平面、在平面的后面添加造型、加大各个面的深度或将表面做成曲面以打破表面等做法，来增加各个轴内的基线。如果您正在考虑曲面，记住**传感器覆盖**文档中所述的传感器覆盖规则。



有关调整形状以优化性能的提示和技巧在文档**对象形状和传感器位置**中有详述。一旦对象的基本形状满足定位条件，可减少传感器数量或布置额外的传感器以最大程序减少遮挡。此时还应采用设计软件建议的位置并将它们放在真实机械设计内的实际位置上。

定义传感器布置

优化传感器布置

尽管 hmd_designer 可自动生成最多 32 个传感器的位置，但工程师基本不会直接使用这些位置。实际传感器布置必须考虑到工业设计、整体机械和电气设计以及可制造性。要确定定位对象上的最终传感器数量及其位置，需要执行几个步骤。其中一个主要方法是减少总传感器数量。

如果模拟输出显示有冗余传感器，就可下调传感器计数器的值。尤其是使用较小的对象时，hmd_designer 会将传感器布置的间距极小。尤其是当这些传感器处于同一平面时，也许能够剔除其中一个传感器。可通过减少 hmd_designer 中的传感器计数器值并将模拟结果进行比较，轻松研究能否使用更少的传感器。减少传感器数量是节省成本的一种途径。但是，它也是将传感器重新添加到设计中以改进性能的机会。

将传感器集成到机械设计

请务必记住，用户会与其定位对象互动，可能会遮挡传感器。用户可能会抓住并调整某个对象、在该对象前面挥手，或以设计师意料之外的方式使用对象。要提高设计的稳健性，可将额外的传感器重新添加到设计之中，以应对部分传感器被遮挡的情形。当然，即使选择了最佳的传感器数量，也必须细化其位置以适应一件可加工产品的工业和机械设计。在这里，设计师开始使用 hmd_designer 的输出作为指导方针，将传感器集成到机械设计中。此流程在**定义传感器位置**中有详述。

编写 JSON 文件

实现工业设计意图、适应内部机械特性以及实现经济高效的电气互连，这三个要求是影响传感器在定位对象上最终位置的決定因素。传感器位置不再由 hmd_designer 定义，而是输入到工具中用于可视化和模拟。请记住，在传感器布置生成流程中，hmd_designer 创建的输出文件之一是用于描述各个传感器的位置和方位的 JSON 文件。一旦工程师定义了传感器布置方式，设计师必须创建一个 JSON 文件。JSON 文件包含三个与传感器布置相关的关键数组：

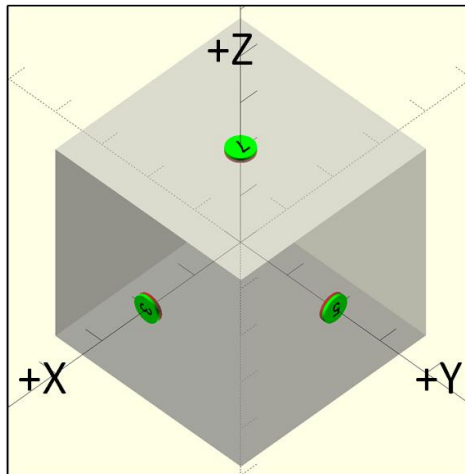
1. “channelMap”- 指示传感器必须连接的 FPGA 端口的编号。
2. “modelNormals”- 包含一个与传感器光电二极管表面相垂直的单位向量。
3. “modelPoints”- 包含光电二极管活动区域中心点的位置向量。

记住，JSON 文件中的所有位置值都以米为单位表示。

```
{
  "channelMap" : [3, 5, 7],
  "modelNormals" : [
    [1, 0, 0],
    [0, 1, 0],
    [0, 0, 1]
  ],
  "modelPoints" : [
    [0.02, 0, 0],
    [0, 0.02, 0],
    [0, 0, 0.02]
  ]
}
```

可视化 JSON 文件

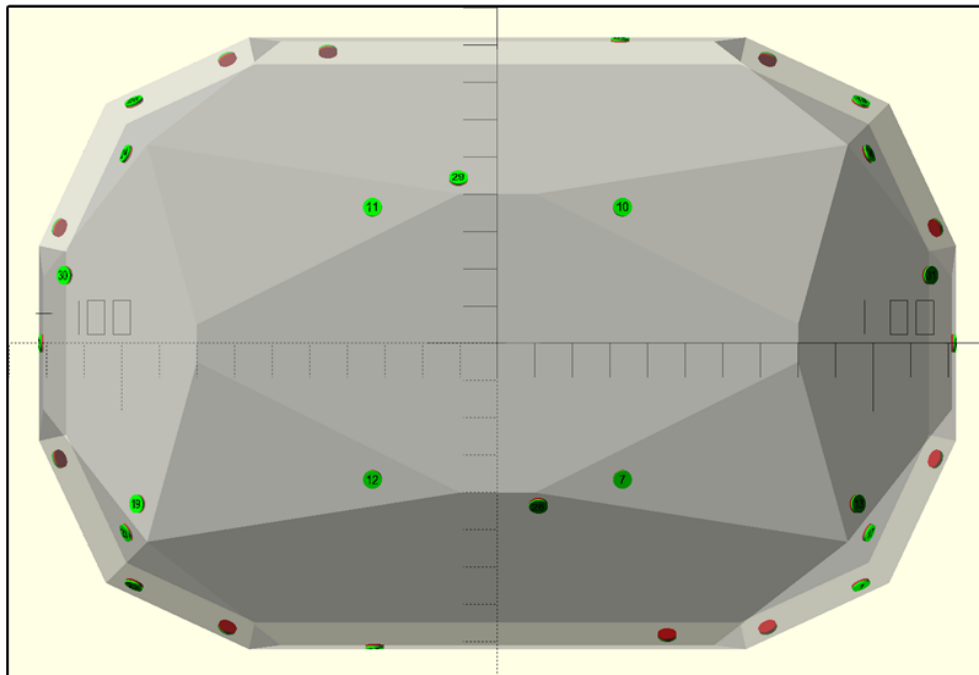
作为一项关键技能，在设计定位对象中要会使用 JSON 文件。幸运的是，借助模拟工具，设计师可以在更新设计后验证其 JSON 文件。模拟工具要用到一个 JSON 文件和 STL 传感器对象，并在 OpenSCAD 中渲染模型。可视化会根据 JSON 文件中指定的位置和方位来显示所布置的传感器，并用通道编号标注各个传感器。

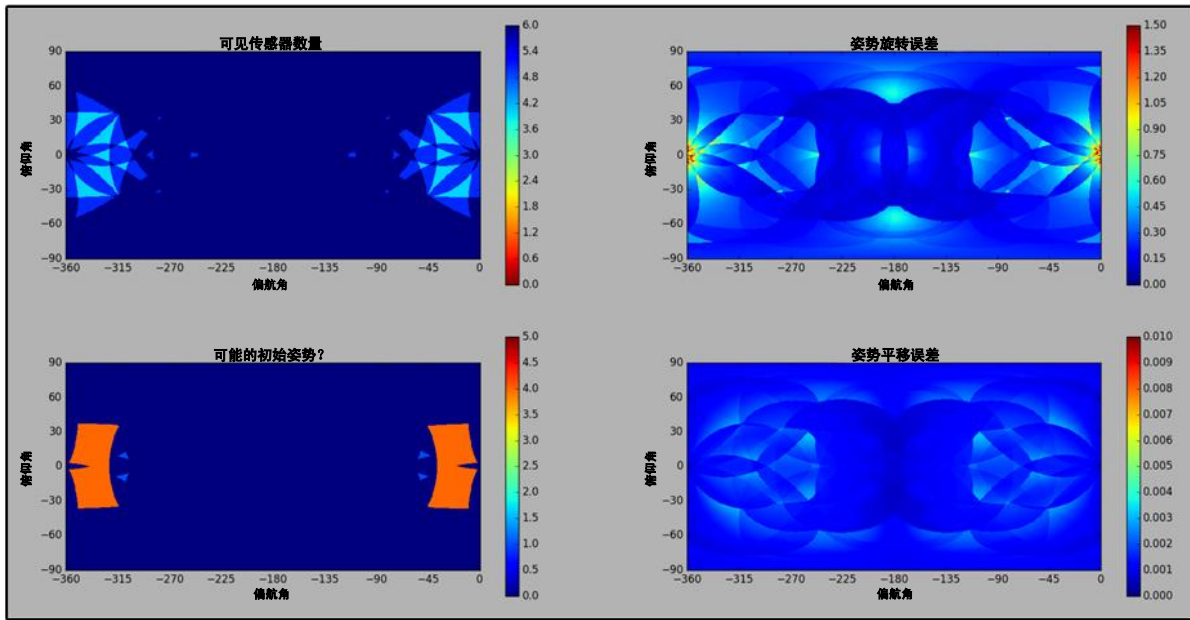


JSON 文件的“imu”变量在文档 **JSON 文件** 中有详述，JSON 文件的可视化流程在**模拟用户手册**中的“可视化现有模型”中有介绍。在 JSON 文件精确反映设计后，务必验证在将传感器位置调整到其实际位置时，对象性能未受影响。

对最终模型进行模拟

在机械设计成熟的过程中验证设计是否仍能达到可接受的定位性能，这是一个迭代过程。在设计成熟的过程中，3D CAD 模型会发生变化，JSON 文件会有更新以反映新位置，同时要对设计进行模拟以验证定位性能。如果出现定位问题，传感器位置会再次发生变化，并重复此流程。将定义对象当前形状的 STL 文件以及包含传感器位置的关联 JSON 文件导入 hmd_designer，使用户能够模拟设计。将模拟输出与之前性能已知的设计迭代和对象进行比较，确保定位质量达到标准。模拟现有 JSON 文件的流程在**模拟用户手册**中的“模拟现有模型”下有介绍。当然，如果每次执行设计迭代时，都要手动将 3D 模型转换为 JSON 文件，我们可以想象会是多么枯燥乏味。





JSON 自动化

尽管可以在 3D 建模软件中测量传感器位置和方位并手动将值输入到 JSON 文件中，但这个过程非常枯燥乏味。脚本作为一种重要工具，可从 3D CAD 软件中提取传感器位置和法线数据。实体建模包和工作流流程实现自动化后，设计迭代流程变得更快也更不易出错。有关建立实体模型以帮助测量传感器位置以及有关传感器数据提取脚本编写方式的提示和技巧，请参阅[从 CAD 中提取传感器数据](#)。

当设计完成时，最好的跟踪测试方法就是创建原型，然后将其整合到 SteamVR™ 之中并体验其第一手性能。

快速成型

快速成型对于验证对象真实定位性能至关重要。要想有效地评估，需要有一个能够将传感器置于正确位置的机械配置，以及用于将传感器连接到 SteamVR™ 的电气系统。3D 打印是最简单的机械原型生成方式。在深化设计以适应注塑工艺之前，创建一个适应 3D 打印流程的初始实心模型是一个关键步骤。可打印 3D 形状并在内部或表面上安装传感器以评估定位。但是，这些传感器需要接往 SteamVR™ 的电气连接。

SteamVR™ Tracking 开发工具包致力于向设计师提供创建快速原型所需的所有电子器件，而无需制造印刷电路板。可在 3D 打印形状上贴装单个的传感器 PCB。这些传感器通过单独的 FPC 跳线（可通过零售经销渠道购买各种长度的跳线）连接至评估板（EVM）。EVM 通过 USB 或无线连接到 SteamVR™ 计算机。



在文档**对象快速成型**中了解更多有关快速成型的方法和建议。一旦构建了原型，设计师可以完成 JSON 文件并开始测试原型。

完成 JSON 文件

每个定位对象都包含一个 JSON 文件，该文件描述了对对象的几何形状以及 SteamVR™ 所需的属性。模拟流程需要设计师指定与传感器布置相关的 JSON 变量，但 JSON 文件中还有更多变量是 SteamVR™ 集成所需的变量。可在文档 **JSON 文件** 中找到 JSON 的完整说明以及如何指定其包含的值。下文提到了一些要点。记住，JSON 文件中的所有位置值都以米为单位表示。

将传感器数组复制到“lighthouse_config”变量中。

```
"lighthouse_config" : {  
  "channelMap" : [3, 5, 7],  
  "modelNormals" : [  
    [1, 0, 0],  
    [0, 1, 0],  
    [0, 0, 1]  
  ],  
  "modelPoints" : [  
    [0.02, 0, 0],  
    [0, 0.02, 0],  
    [0, 0, 0.02]  
  ]  
},
```

使用“imu”变量，通过“plus_x”、“plus_z”和“position”字段设置 IMU 位置和方向。使用 hmd_degineer_gui 可视化对象中 IMU 的位置和方位。有关可视化 JSON 文件的更多信息，请参阅[模拟用户手册](#)。

```
"imu" : {
  "acc_bias" : [ 0, 0, 0 ],
  "acc_scale" : [ 1, 1, 1 ],
  "gyro_bias" : [ 0, 0, 0 ],
  "gyro_scale" : [ 1, 1, 1 ],
  "plus_x" : [ 1, 0, 0 ],
  "plus_z" : [ 0, 0, -1 ],
  "position" : [ 0.002, 0.010, -0.010 ]
},
```

通过将任何定位对象视为一个控制器，可更轻松地对初始定位性能进行故障排除。将“device_class”设置为控制器，并将“render_model”设置为工作渲染模型。

```
"device_class" : "controller",
"render_model" : "generic_hmd",
```

“Head”成员定义定位对象的坐标系与 SteamVR™ 坐标系之间的关系。对于 HMD 和控制器，它的使用方式是不同的。更多详情，请参阅 **JSON 文件**。

```
"head" : {
  "plus_x" : [ -1, 0, 0 ],
  "plus_z" : [ 0, 0, 1 ],
  "position" : [ 0, 0, 0 ]
},
```

在将对象作为控制器完成了初始定位验证后，可通过将“device_class”更改为“hmd”来评估 HMD。要用作 HMD，“head”变量必须让模型的坐标系参考佩戴者双眼之间的点。JSON 文件中还有数个 HMD 特定部分与显示和光学相关，必须按 **JSON 文件**中所述完成这些部分。

测试对象的定位性能时必须用到上述值。但是，在测试对象之前，务必完成 JSON 文件中的所有条目。有关该文件及其内容的完整说明，请参阅 **JSON 文件**。

完成 JSON 文件后，使用 lighthouse_console 实用工具将此文件上载到定位对象。可在 **Lighthouse Console 参考手册**中了解更多关于 lighthouse_console 及其功能的信息。

提示：强烈建议您在使用 uploadconfig 时断开所有其他 SteamVR™ 设备的连接。如果您覆盖了其他对象的配置，它们在渲染后会无法运行。考虑在所有设备上使用 downloadconfig 并将 JSON 文件存档以能够恢复。

```
lh> uploadconfig c:\<folderlocation>\<filename>.json
```

构建对象并上载 JSON 文件后，便可通过 USB 连接到 SteamVR™。尝试在 VR 中使用该对象之前，在 lighthouse_console 中运行几个简单的测试通常可以节省时间。同样，在评估定位性能之前，务必要校准对象。

测试对象的连通性

Lighthouse_console 实用工具提供了一些简单但有效的命令，可用于验证光学传感器、IMU 和通信路径是否正常运行。

- 通过使用 `lighthouse_console` 连接对象，测试 USB 连通性。
- 使用“`dump`”和“`sample`”命令输出传感器信息。查找所有已连接传感器发出的传感器命中数据流。
- 使用“`dump`”和“`imu`”命令，验证 IMU 数据是否在以 1000 Hz 的频率发出合适的数值流。
- 使用“`period`”命令验证对象的各个传感器发出的传感器命中信号。

有关如何运行测试和解读结果的完整说明，请参阅文档**测试对象的连通性**。一旦传感器和 IMU 通过 USB 传输数据流，设备即已准备就绪，可以集成到 SteamVR™ 了。但是，在验证定位性能之前，务必要校准对象。

提示：在对某个对象进行光学校准之前，必须启动该对象。将该对象连接到 SteamVR™，并调试该对象和 JSON 文件，直至设备在 VR 中开始定位。即将进行的光学和 IMU 校准可改善定位性能，但初始定位不需要这两项校准。

光学校准

光学校准是测量传感器在定位对象上的实际位置以及更新 JSON 文件以反映传感器确切布置的流程。制造公差意味着给定设备上的传感器布置绝不会与 JSON 文件中指定的理想值完全匹配。Valve 已编写了一个工具，将理想 JSON 视为起点，并使用定位器的参考信号测量传感器在对象上的确切位置。此工具生成一个特定于已校准设备的已校准 JSON 文件。务必对各个新设备执行此校准流程，无论是在实验室还是在大批量生产中。

定位器处于活动状态时，工具 `vrtrackingcalib` 使用来自定位器的参考信号测量传感器在设备上的实际位置。谨记以下规则以进行正确的校准：

1. 始终从理想的 JSON 文件开始着手。不可从由 `vrtrackingcalib` 生成的 JSON 文件开始校准，否则误差会累积。
2. 对象在校准之前必须上载了一个 JSON 文件。该 JSON 可以是理想 JSON 或者是来自之前校准步骤的 JSON。
3. 确保基准期间仅有一个定位器处于活动状态。如果定位器处于“A”模式下，校准速度会更快。
4. 在空间中四处移动，以得出定位器误差的均值。
5. 使用不同的对象方位以得出其他系统误差的均值。
6. 向定位器显示传感器的多种不同重叠组合。

```
Realigning point cloud to original model:
> RMS: 0.000266
> Ceres Solver Report: Iterations: 14, Initial cost: 1.885440e-007, Final
cost: 1.767502e-007, Termination: CONVERGENCE

> Corrected scale of 0.9999
> Corrected normal shift of 0.0001
> Corrected rotation of 0.0977 deg
> Corrected translation of 0.0636 mm
```

在 `vrtrackingcalib` 生成一个新的 JSON 文件（其中“`modelPoints`”变量中包含了更新后的位置）后，使用 `lighthouse_console` 将新的 JSON 文件上载到定位对象。此 JSON 中的所有其他值都保留自之前上载到设备的 JSON。有关整个光学校准程序的说明，请参阅**光学校准**。

IMU 校准

IMU 会表现出制造流程所产生的偏移和偏差。SteamVR™ 能够校准整个系统中的这些误差。但是，在定位启动后，达到正确的值需要花一些时间。Valve 编写了一个实用程序 `imu_calibrator`，用于测量设备中的 IMU 误差。此校准工具会生成 JSON 片段，可将其添加到 JSON 文件中的“imu”变量中。以测得的偏移和偏差为着手点，可以减少 SteamVR™ 获取最优 IMU 误差值所需的时间。

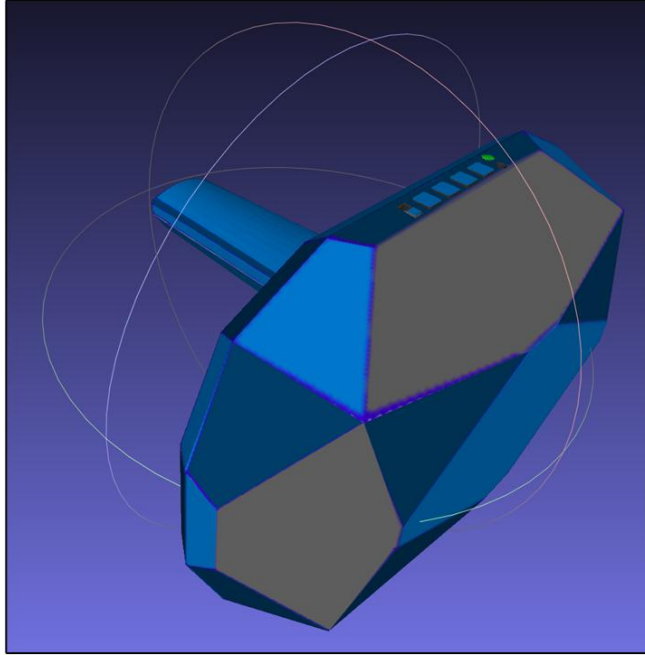
校准流程是指将对象置于六个正交位置上，并由该校准实用程序执行 IMU 测量。校准完成后，可将此 JSON 片段剪切并复制到对象的 JSON 文件中。

```
"acc_scale" : [ 0.998, 0.9983, 0.9915 ],
"acc_bias" : [ 0.05089, -0.03676, -0.2253 ],
"gyro_scale" : [ 1.0, 1.0, 1.0 ],
"gyro_bias" : [ 0.06253, 0.01054, -0.02128 ],
```

现在可对经过测试和校准的对象进行更深入的定位性能评估，并改进对象的设计。但是，要查看 SteamVR™ 中的对象，需要一个渲染模型。首先，渲染模型可以是 SteamVR™ 随附的模型；但如果选用一个与设备形状匹配的渲染模型，可有助于设计师在 VR 中与对象交互。

创建渲染模型

对象的渲染模型是 VR 中显示的用于代表对象的 3D 图像。对象 JSON 文件中参考了默认渲染模型，此模型通常在真实世界中显示为定位对象。如果拥有一个与实际对象匹配的渲染模型，用户在 VR 中查看对象时可轻松选取该对象。一旦用户在 SteamVR™ 中打开一个应用程序，此应用程序会覆盖默认渲染模型以使对象显示为体验所需的手、武器、工具、生物或任何其他实体。



渲染模型至少包含三个文件：

1. 对象文件 (.obj)
2. 材料文件 (.mtl)
3. 纹理文件 (.tga)

这些文件由 3D 设计师使用专业的 3D 图形软件包生成。SteamVR™ Tracking 文档并不提供任何 3D 图形软件包中的完整课程。而是使用一款名为 Blender™ 的开源软件包，介绍了一些基本示例。使用 Blender™ 创建简单渲染模型的流程在**渲染模型**中介绍。

一旦生成了文件，将它们整合到 SteamVR™ 中非常简单：只需将它们放到一个名为“rendermodels”文件夹的子文件夹中，并在对象的 JSON 文件中引用该新文件夹即可。该流程的具体细节在**渲染模型**中有详细说明。准备好渲染模型并通过 USB 连接设备后，启动 SteamVR™ 并在虚拟现实中与新对象进行交互。

评估定位性能

要有效评估定位性能并排除定位问题需要有一定的经验。目前尚无用于量化定位对象性能的方法。随着系统约束条件的增加以及针对不同应用的新对象不断被开发出来，唯一能真正验证性能的方法就是在 VR 中对其进行体验。Valve 强烈建议购买最新的 SteamVR™ 设备，并在 VR 中花时间体会那些性能最佳的设备所给人带来的感受。设计师的感知经过高性能体验的“标定”后，察觉新设备中微小误差的能力就会大大提升。有关评估定位性能的完整指南，请参阅**评估定位性能**。一般而言，评估定位可分为三大类：

1. 对象启动 - 对象是否显示在 VR 中并保持可见？
2. 控制器定位 - 对象是否作为控制器进行定位，且无可见震动或失真？
3. HMD 定位 - 对象定位起来是否具有足够低的振动和失真度从而能够作为头戴式显示器使用，是否无任何可见或致呕效应？

对于被配置为控制器的设备，在启动其评估流程时这一点非常有用。在 VR 中通过已知良好的 HMD 来检查对象是了解新对象定位性能的好办法。可用配置为控制器的对象回答上述问题 1 和 2。对于连接至双目显示屏的对象，在控制器模式下消除任何可见定位问题后，可将对象配置为 HMD 并做进一步的研究。

使用与被测设备匹配的渲染模型可让您更轻松地了解其是否工作正常。要执行此评估，可使用通过 CAD 生成的渲染模型，并使用类似于烘焙环境遮挡图的简单纹理。应在出货 (shipping) 之前，使用 3D 图形网格设计的最佳实践，从无到有创建一个合理的渲染模型。

将对象作 HMD 进行评估可帮助引入 JSON 文件中的“head”变量，同时用户还可以开始评估 VR 中的光学失真和显示特性。这些相互作用的光学、定位和显示系统会使定位问题更难以调试。不过，**评估定位性能**文档还是尝试讲解了一些已知问题及其解决办法。



总结

设计定位对象并将其集成到 SteamVR™ 这一流程是一项跨学科工作，涵盖从初步构想到传感器布置、模拟、原型制作、校准、软件集成、评估和故障排除等一系列流程。请始终牢记，高性能定位所需的条件，并让工业和机械设计师谨记对象形状、材料和表面处理对定位质量的影响。Hmd_designer 作为一款强大的工具，可在深化对象设计的过程中生成传感器布置并模拟结果。一旦定位对象的形状成熟，将传感器放到 3D CAD 模型中，将这些位置和方位导出到 JSON 文件并模拟修订后的设计以验证性能，这是一个重要的迭代过程。为机械工程师的实体建模软件开发一个脚本以提取传感器位置和方位从而生成 JSON 文件，这种做法非常有用。模拟设计很重要，但快速制作对象的原型以测试实际形状这一流程无可替代。要将原型集成到 SteamVR™，需要一个整理好的 JSON 文件以描述对象。JSON 文件均通过 lighthouse_console 将上载到对象。Lighthouse_console 也是一款非常有用的工具，它可在转移至 SteamVR™ 之前用于测试基本连通性。使用 vrtrackingcalib 执行光学校准可在测试前优化定位性能。要在 VR 中观看对象，需要一个渲染模型。工程师可从 SteamVR™ 随附的其中一个渲染模型着手。但是，开发一个能够反映定位对象形状的定制渲染模型对于测试非常有用。最后，评估定位需要经验。通晓现有 SteamVR™ 产品的性能是获取评估新设备定位性能所需经验的最佳途径。对定位问题进行故障排查极具挑战性，但 SteamVR™ 日志和文档可以帮助您识别问题，从而让新的定位对象充分发挥其潜力。