

## 第九章 追踪

追踪物理世界的运动对于任何 VR 系统都是关键的部分。追踪是将 VR 头显引入消费型电子产品的最大障碍之一，并且由于我们希望扩展和改进 VR 体验，它仍然会是一个重大的挑战。高精度的追踪方法主要由商用硬件组件支持，例如惯性测量单元(IMU)和由于智能手机产业发展，规格和成本急剧下降的相机。

根据目前被追踪的物体情况，VR 系统中可能会出现三类追踪：

### 1. 使用者的感官：

回想 2.1 节，人的感觉器官，比如眼睛和耳朵，都有由身体控制的自由度。如果一个显示器连接到一个感应器，它应该被视为在 VR 中被附着到周围的世界，那么器官的位置和方向就需要被追踪。我们可以将追踪变换的反方向应用于刺激来正确地“抵消”这些自由度。大部分的焦点在于头部追踪，这对 VR 的视觉和听觉成分来说已经足够了。然而，如果渲染和显示技术需要补偿 5.3 节中讨论的眼动，视觉系统可能还需要人眼追踪。

### 2. 用户的其他身体部位：

如果用户想要在虚拟世界中看到自己身体被正确地表示，那么身体的运动理应被追踪，以便它可以在匹配区域中再现。也许面部表情或手势是互动所必需的。虽然完美匹配是追踪感觉器官的理想选择，但追踪其他身体部位不是必需的。现实世界中的小动作可以转换成更大的虚拟世界动作，以使用户更不费力。在极限情况下，用户只需按下一个按钮即可更改主体配置。例如，她可以通过一次点击就能在她的虚拟手中抓住一个物体。

### 3. 其余环境：

在用户周围的现实世界中，可以追踪物理对象。对于存在于真实世界但不存在于虚拟世界中的物体，出于安全原因，系统需要提醒用户它们的存在。想象一下，用户即将撞墙，或绊倒一名幼童。在一些 VR 应用中，被追踪的物理对象可以在 VR 中匹配，以使用户在与它们交互时接收触碰反馈。在其他应用中，例如远程监控，物理世界的很大一部分可以通过实时捕捉被“引入”虚拟世界。

第 9.1 节介绍了追踪围绕单个轴的旋转的简单情况从而为将框架扩展到追踪 3D 刚体的 3-自由度方向的 9.2 节做铺垫。这主要依赖于 IMU 的角速度读数。最常见的用途是追踪穿戴 VR 头显设备的头部运动，但也可能适用于追踪手持控制器或其他设备。第 9.3 节讨论了位置和方向的追踪，这在大多数系统中要求在物理世界的固定部分和被追踪物体之间的视线可见性。9.4 节讨论追踪通过节点连接在一起的多个物体的情况。最后，第 9.5 节介绍了使用传感器构建物理世界的表示以便将其引入虚拟世界的情况。

## 9.1 2D 方向追踪

本节介绍如何使用惯性测量单元(IMU)估算刚体的方向。主要应用是确定视点方向，即第 3.4 节介绍的当用户佩戴 VR 头显设备时的  $R_{eye}$ 。另一个应用是估计手持控制器的方向。例如，假设我们想根据用户指示的方向制作一个适用于虚拟世界的激光笔。场景中亮红色点的位置将由控制器的估计方位决定。更一般地说，如果它具有附加的 IMU，则可以确定物理世界中任何人体部位或移动物体的方位。

为了估计方向，我们首先仔细考虑第 8.1.2 节的旋转木马模型来考虑 2D 情况。在这种情况下，技术问题很容易可视化，并扩展到更重要的 3D 旋转情况。因此，假设我们在旋转的旋转木马上安装陀螺仪。它的工作就是像旋转木马旋转一样测量角速度。在本章中通过区分真实参数值和估计值将会很方便。为了实现这个目标，一个“帽子”将被放置在估计值上。因此，令  $\hat{\omega}$  对应估计或测量的角速度，其可能与真实值  $\omega$  不同。

$\hat{\omega}$ 和 $\omega$ 如何相关？如果陀螺仪功能完好，则 $\hat{\omega}$ 等于 $\omega$ ；然而，在现实世界中，这是无法实现的。 $\hat{\omega}$ 和 $\omega$ 之间的差异的主要原因是标定误差。标定质量从昂贵的IMU(数千美元)到廉价(一美元)之间有很大区别。

我们现在定义一个简单的校准误差模型。以下传感器映射表明传感器输出如何与真实角速度相关：

$$\hat{\omega} = a + b\omega. \quad (9.1)$$

上面， $a$ 和 $b$ 分别被称为偏移和比例。它们是干扰测量的未知常数。如果 $\omega$ 被完美测量，将有 $a=0$ 和 $b=1$ 。

考虑校准误差的影响。比较测得的角速度和真实的角速度得出：

$$\hat{\omega} - \omega = a + b\omega - \omega = a + \omega(b - 1). \quad (9.2)$$

现在想象使用传感器来估计旋转木马的方向。我们想了解真实方向 $\theta$ 与使用传感器输出计算的估计值 $\hat{\theta}$ 之间的差异。令 $d(t)$ 表示称为漂移误差的时间函数：

$$d(t) = \theta(t) - \hat{\theta}(t). \quad (9.3)$$

请注意， $d(t)$ 可能是负值，可通过应用绝对值来获得 $|d(t)|$ ，从而强制其为正值。这将被避免以简化讨论。

假设初始给定 $\theta(0)=0$ ，为了保持简单，角速度 $\omega$ 是恒定的。通过随时间积分(9.2)，漂移误差是

$$d(t) = (\hat{\omega} - \omega)t = (a + b\omega - \omega)t = (a + \omega(b - 1))t. \quad (9.4)$$

当然，随着 $a$ 偏离0或随着 $b$ 偏离1，漂移误差会增加(正向或负向)；然而，请注意第二个分量与 $\omega$ 成比例。忽略 $a$ ，这意味着漂移误差与旋转木马的速度成正比。在使用陀螺仪追踪VR头显设备方面，这意味着随着头部旋转更快，追踪误差以更快的速度增加[168]。

在这一点上，即使对于这种情况，也必须解决四个常见问题才能建立一个有效的追踪系统：

**1.校准：**如果有更好的传感器可用，那么两者可以紧密配对，以便更差传感器的输出进行转换以尽可能接近更好的传感器。

**2.集成：**传感器提供离散时间点的测量结果，产生一个采样率。方向通过聚合或综合测量来估计。

**3.配准：**初始定位必须以某种方式确定，或者通过额外的传感器，或者巧妙的默认假设或启动程序来确定。

**4.漂移误差：**随着误差随时间的增加，需要其他传感器来直接估算并补偿。

所有这些问题在本章中都会涉及到更复杂的设置。组合来自多个传感器读数的信息的过程通常称为传感器融合或过滤。

我们在2D情况下讨论其中的每一个问题，然后再将这些想法扩展到第9.2节中的3D情况。

## 校准

你可以购买传感器，并开始使用它，并假定它已经很好地校准。然而，对于更便宜的传感器，校准常常不可靠。假设我们有一个昂贵的，校准良好的传感器，可以获取误差很小的角速度。设 $\hat{\omega}$ 表示其输出，以将其与永远未知的真值 $\omega$ 区分开来。现在假设我们要校准一堆

便宜的传感器，以便它们尽可能地接近昂贵的传感器。这可以通过将它们安装在可移动表面上并比较它们的输出来完成。为了获得更高的精度和控制，最昂贵的传感器可能是完整的机械系统的一部分，例如昂贵的转盘，校准台或机器人。设 $\hat{\omega}$ 表示待校准的廉价传感器的输出；每个便宜的传感器都必须单独校准。

校准涉及多次采样，有时需要数千次，并将 $\hat{\omega}'$ 与 $\hat{\omega}$ 进行比较。一个通用的标准是平方和的误差，由下式给出

$$\sum_{i=1}^n (\hat{\omega}_i - \hat{\omega}'_i)^2 \quad (9.5)$$

对于角速度的  $n$  个样本。其任务是确定一个适用于廉价传感器输出 $\hat{\omega}$ 的变换，以便它尽可能接近昂贵的传感器输出 $\hat{\omega}'$ 。

使用(9.1)中的误差模型，我们可以选择优化误差的常量  $c_1$  和  $c_2$ ：

$$\sum_{i=1}^n (c_1 + c_2 \hat{\omega} - \hat{\omega}')^2. \quad (9.6)$$

这是一个经典的回归问题，称为线性最小二乘。它通常通过计算包含采样数据的非方阵矩阵的 Moore-Penrose 伪逆求解[341]。

一旦计算出  $c_1$  和  $c_2$ ，每一个未来的传感器读数都被转换为

$$\hat{\omega}_{cal} = c_1 + c_2 \hat{\omega}, \quad (9.7)$$

其中 $\hat{\omega}$ 是原始的原始传感器输出， $\hat{\omega}_{cal}$ 是校准输出。因此，校准会在便宜的传感器输出周围产生一种不可见的外壳，以便模拟昂贵的传感器。外部流程不再显示原始的廉价传感器输出。因此，本章其余部分将校准后的输出简称为 $\hat{\omega}$ 。

## 集成

传感器输出通常以常规采样率达到。例如，Oculus Rift 陀螺仪每 1ms 提供一次测量(产生 1000Hz 的采样率)。令 $\hat{\omega}[k]$ 指第  $k$  个样本，它在时间  $k \Delta t$  到达。

时刻  $t = k \Delta t$  时的方向  $\theta(t)$  可以通过积分来估计为：

$$\hat{\theta}[k] = \theta(0) + \sum_{i=1}^k \hat{\omega}[i] \Delta t. \quad (9.8)$$

每个输出 $\hat{\omega}[i]$ 引起 $\Delta \theta[i] = \hat{\omega}[i] \Delta t$ 的旋转。有时以增量形式写入(9.8)更方便，这表示在每个新传感器输出到达后更新为 $\hat{\theta}$ ：

$$\hat{\theta}[k] = \hat{\omega}[k] \Delta t + \hat{\theta}[k-1]. \quad (9.9)$$

对于第一种情况， $\hat{\theta}[0] = \theta(0)$ 。

如果 $\omega(t)$ 在 $\theta(k \Delta t)$ 和 $\theta((k+1) \Delta t)$ 之间变化很大，那么知道 $\hat{\omega}[k]$ 对应什么是有帮助的。它可以是间隔 $\Delta t$ 开始时的角速度，间隔的结束时间或者间隔内的平均值。如果是开始或结束，那么积分的梯形近似可能会随着时间的推移产生更少的误差[133]。

## 配准

在式(9.8)中，假定初始方向 $\theta(0)$ 是已知的。实际上，这与配准问题相对应，这是真实和虚拟世界之间的初始对齐。为了理解这个问题，假设 $\theta$ 表示VR头盔的偏航方向。一种可能性是分配 $\theta(0) = 0$ ，其对应于当追踪系统打开时头显设备面对的任何方向。这可能是系统启

动时。如果头显设备具有“头戴式”传感器，则可以在用户将头显设备戴上时开始。很多时候，前进的方向可能会以不好的方式无意中设置。例如，如果一个人开始一个 VR 演示，并将头显设备交给面对另一个方向的其他人，那么在 VR 中，用户不会朝向预期的前进方向。这可以通过导致“向前”( $\theta(t)$ )被重新定义为用户目前面对的任何方向的简单选项来修复。

这个整体问题的一个替代方法是声明  $\theta(0) = 0$  以对应于物理世界中固定的方向。例如，如果用户坐在计算机监视器前面的桌子上，则向前方向可以被定义为用户和头盔面向监视器的偏航角。实施这个解决方案需要一个能够测量相对于周围物理世界的偏航方向的传感器。例如，在 Oculus Rift 中，用户面对一个固定的摄像头，它对应于前进方向。

## 漂移校正

为了建立一个有用的追踪系统，不能允许漂移误差(式 9.3)累积。即使陀螺仪被完美校准，由于诸如量化输出值，采样率限制和未建模噪声等其他因素，漂移误差仍会增加。第一个问题是估计漂移误差，通常用附加传感器完成漂移误差。这个实例将在 9.2 节中给出。对于简单的旋转木马的示例，想象一下架空摄像机偶尔会拍摄一张照片来测量方向。令  $\hat{\theta}_d[k]$  表示从该单个传感器测量结果到达阶段  $k$  的估计方位。

由于漂移误差，现在有两个相互矛盾的信息来源：1)通过积分陀螺仪估计的方向  $\hat{\theta}[k]$ ，以及 2)由相机(或其他一些独立的传感器)瞬时估计的方向  $\hat{\theta}_d[k]$ 。混合这两种来源的经典方法是互补过滤器，它在两种估计之间进行数学插值：

$$\hat{\theta}_c[k] = \alpha \hat{\theta}_d[k] + (1 - \alpha) \hat{\theta}[k], \quad (9.10)$$

其中  $\alpha$  是必须满足  $0 < \alpha < 1$  的增益参数。以上， $\hat{\theta}_c[k]$  表示阶段  $k$  处的校正估计。由于陀螺仪在短时间内通常是准确的，但会逐渐漂移，所以选择  $\alpha$  接近于零(例如  $\alpha = 0.0001$ )。这导致瞬时估计值  $\hat{\theta}_d[k]$  逐渐产生影响。另一方面，如果  $\alpha$  接近于 1，那么由于每个阶段  $\hat{\theta}_d[k]$  的误差，估计的方向会大幅波动。另一个考虑因素是，如果传感器输出  $\hat{\theta}_d[k]$  的速率比陀螺仪采样速率低得多，则使用最近记录的输出。例如，相机图像可能产生 60Hz 的方位估计，而陀螺仪产生 1000Hz 的输出。在这种情况下， $\hat{\theta}_d[k]$  将保持 16 或 17 个阶段的相同值，直到新的相机图像变为可用。

选择增益  $\alpha$  足够高以便漂移得到校正是非常重要的，但要足够低以使用户不会感觉到校正。增益可以通过使用卡尔曼滤波器“最优地选择”[45,140,158];然而，如果我们有一个线性随机系统，最优性只能保持不变，而这种情况不适用于人体追踪。卡尔曼和补充滤波器之间的关系，对于本章中使用的确切模型，见[119]。

使用简单的代数，(9.10)中的互补过滤器公式可以重新生成以产生以下等效表达式：

$$\hat{\theta}_c[k] = \hat{\theta}[k] - \alpha \hat{d}[k] \quad (9.11)$$

其中，

$$\hat{d}[k] = \hat{\theta}_d[k] - \hat{\theta}[k]. \quad (9.12)$$

以上， $\hat{d}[k]$  只是阶段  $k$  中漂移误差的估计值。因此，互补滤波器可以被想象为以一个小的比例量  $\alpha$  来施加有符号的误差，以试图逐渐将其强制为零。

## 9.2 3D 方向追踪

### IMU

回忆 2.1 节(图 2.9)，IMU 最近已经从大型机械系统转向廉价微型 MEMS 电路。这种进

展是高质量定位追踪的关键推动力。陀螺仪测量沿着三个正交轴的角速度，以获得  $\omega_x$ ， $\omega_y$  和  $\omega_z$ 。对于每个轴，感应元件位于垂直平面内，非常类似于前庭器官的半规管(8.2 节)。每种情况下的传感元件都是微机械的机械元件，像音叉一样振动和操作。如果传感器以其灵敏度方向旋转，则元件会受到科里奥利力，被转换成电信号。这些信号被校准以产生每秒度数或弧度的输出;见图 9.1。

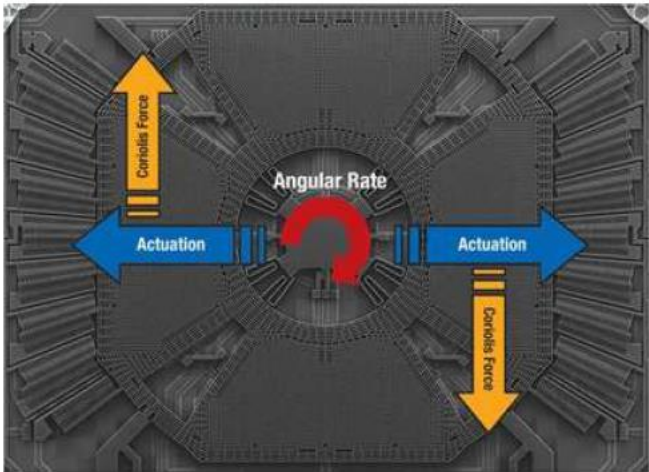


图 9.1: 振动 MEMS 元件在旋转过程中对科里奥利力做出响应，将其转换为电信号。(图源自 Fabio Pasolini)

IMU 通常包含用于检测漂移误差的附加传感器。最常见的情况是，加速度计沿着三个轴测量线性加速度以获得  $a_x$ ， $a_y$  和  $a_z$ 。其操作原理如图 9.2 所示。MEMS 磁力计也出现在许多现代 IMU 上，这些 IMU 沿着三个垂直轴测量磁场强度。这通常通过 MEMS 结构的机械运动来完成，所述 MEMS 结构在磁场内传导时受到洛伦兹力的作用。

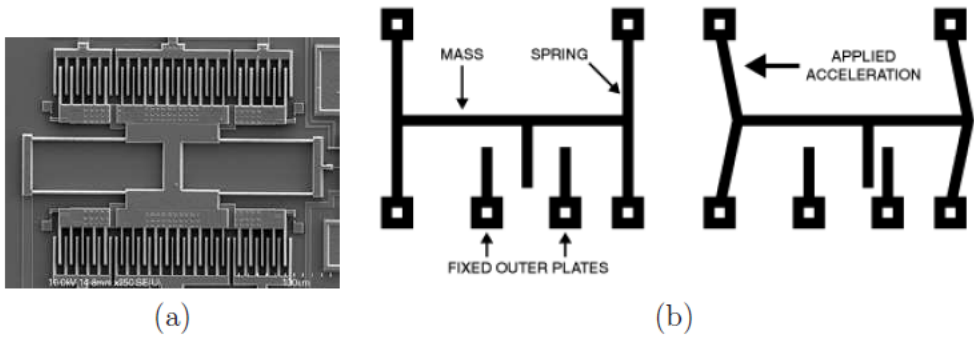


图 9.2: (a)用于感测线性加速度的 MEMS 元件。(b)由于在一个方向上的线性加速度，这些极板移动并导致在外板之间测量的电容变化。(图源自 David Askew)

### 校准

回顾第 9.1 节传感器输出由于校准问题而失真。在一维角速度情况下，只有两个参数，出现在(9.1)中的比例尺和偏移量。在 3D 设置中，这自然会扩展到 3 个比例尺和 3 个偏移量参数;然而，情况由于 MEMS 元件的非正交性带来的误差会更糟。所有这些都可以由排列在齐次变换矩阵中的 12 个参数来解释：



$$\begin{bmatrix} \hat{\omega}_x \\ \hat{\omega}_y \\ \hat{\omega}_z \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & j \\ d & e & f & k \\ g & h & i & \ell \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ 1 \end{bmatrix} \quad (9.13)$$

共有 12 个而不是 6 个自由度，因为左上角的  $3 \times 3$  矩阵不被限制为旋转矩阵。 $j, k$  和  $\ell$  参数对应于偏移量，而其他所有参数都处理比例和非正交性。采用与 9.1 节相同的方法，可以通过最小平方误差相对于更好的传感器的输出来估计该变换的逆，其提供了真实情况。MEMS 传感器的输出通过应用估计的齐次变换被调整以提高性能(这是(9.7)到 12 参数情况的延伸)。这种通用方法适用于校准陀螺仪和加速度计。磁力计也可以用这种方式进行校准，但还会有其他问题，如软铁偏压。

MEMS 传感器面临的另一个挑战是处理其他微妙的依赖关系。例如，输出对 MEMS 元件的特定温度敏感。如果 VR 头显设备在使用过程中变热，那么在实践中可能出现的每个温度都需要校准参数。幸运的是，IMU 通常包含一个温度传感器，可用于将校准参数与相应的温度相关联。最后，MEMS 元件可能对作用在电路板上的力敏感，这可能会被改变，例如，由悬挂连接器改变。必须注意将 MEMS 电路的外部电路板力量隔离开来。

## 集成

现在考虑将陀螺仪输出序列转换为 3D 定向估计的问题。在每个阶段  $k$ ，一个向量

$$\hat{\omega}[k] = (\hat{\omega}_x[k], \hat{\omega}_y[k], \hat{\omega}_z[k]) \quad (9.14)$$

从传感器到达。在 9.1 节中，传感器输出  $\hat{\omega}[k]$  在方向上被转换为  $\Delta \theta[k]$ 。对于 3D 情况，方向的变化表示为四元数。

令  $q(v, \theta)$  为由轴角转换公式(3.30)得到的四元数。回顾第 8.1.2 节，瞬时旋转轴是角速度的大小。因此，如果  $\hat{\omega}[k]$  是阶段  $k$  中的传感器输出，则估计的旋转轴是

$$\hat{v}[k] = \hat{\omega}[k] / \|\hat{\omega}[k]\|. \quad (9.15)$$

此外，在时间  $\Delta t$  期间发生的估计旋转量是

$$\Delta \hat{\theta}[k] = \|\hat{\omega}[k]\| \Delta t. \quad (9.16)$$

使用估计的旋转轴(9.15)和数量(9.16)，估计方向随时间的变化  $\Delta t$

$$\Delta \hat{q}[k] = q(\hat{v}[k], \Delta \hat{\theta}[k]). \quad (9.17)$$

在每个阶段使用(9.17)，获得最新传感器输出后的估计方向  $\hat{q}[k]$  从  $\hat{q}[k-1]$  递增计算为

$$\hat{q}[k] = \Delta \hat{q}[k] * \hat{q}[k-1], \quad (9.18)$$

其中  $*$  表示四元数乘法。这是(9.9)的 3D 泛化，其中可以使用简单的加法来结合 2D 情况下的旋转。在(9.18)中，需要四元数乘法来聚合方向的变化(简单加法是可交换的，这对于 3D 旋转是不合适的)。

## 配准

偏航组件的配准问题与第 9.2 节中的相同。正向可以从刚体的初始方向选择，或者可以相对于世界中的固定方向来确定。俯仰和滚转方向应该被确定以便它们与重力对齐。虚拟世

界不应该出现相对于现实世界的倾斜(除非这是所期望的效果, 这种情况很少)。

## 倾斜校正

(9.3)中的漂移误差  $d(t)$  是单个角度, 可能是正值或负值。如果增加到估计值  $\hat{\theta}(t)$ , 则将获得真正的方向  $\theta(t)$ 。它与 3D 情况类似, 但是具有四元数代数。三维漂移误差表示为

$$d(t) = q(t) * \hat{q}^{-1}(t), \quad (9.19)$$

如果  $q(t) = \hat{q}(t)$ , 它等于旋转。此外, 请注意, 将漂移误差应用于估计产生  $q(t) = d(t) * \hat{q}(t)$ 。

由于漂移误差是 3D 旋转, 因此可以将其构造为偏航, 俯仰和滚转的结果。让倾斜误差表示俯仰和滚转相对应的漂移误差部分。这将使用“向上”传感器进行检测。让偏航误差指的是漂移误差的剩余部分, 将使用“指南针”进行检测。实际上, 并不存在完美的“上”和“罗盘”传感器, 这些传感器将在后面进行介绍。

假设连接到刚体的传感器总是返回与固定世界坐标系中的  $y$  轴平行的“向上”向量。换句话说, 它将与重力平行。由于传感器安装在身体上, 它会在身体的坐标框架中报告其值。例如, 如果身体滚动 90 度, 以使其  $x$  轴向上指向, 则“向上”向量将被报告为  $(0,0,1)$ , 而不是  $(0,1,0)$ 。为了解决这个问题, 将传感器输出转换为世界范围会很方便。这涉及通过  $q(t)$  (身体方向) 旋转它。对于我们的例子, 这个滚动旋转会将  $(0,0,1)$  变换为  $(0,1,0)$ 。图 9.3 显示了一个 2D 例子。

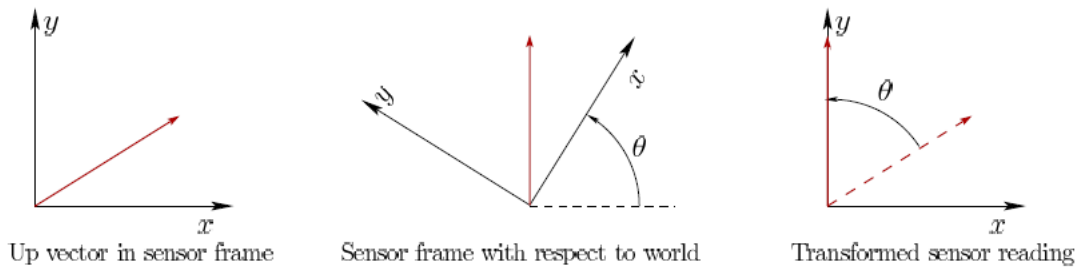


图 9.3: 如果通过旋转  $\theta$  的加速度计完全检测到“向上”, 则其输出需要旋转  $\theta$  以从世界框中查看它。

现在假设发生了漂移误差, 并且  $\hat{q}[k]$  是估计的方向。如果将这个变换应用于“向上”向量, 那么由于漂移误差, 它可能不会与  $y$  轴对齐, 如图 9.4 所示。向上向量  $u$  被投影到  $xz$  平面中以获得  $(u_x, 0, u_z)$ 。倾斜轴位于  $xz$  平面内, 并构造为投影向量的法向:  $t = (u_z, 0, -u_x)$ 。围绕轴线  $t$  执行  $\phi$  的旋转将使向上向量与  $y$  轴对齐。因此, 漂移误差的倾斜误差部分是四元数  $q(t, \phi)$ 。

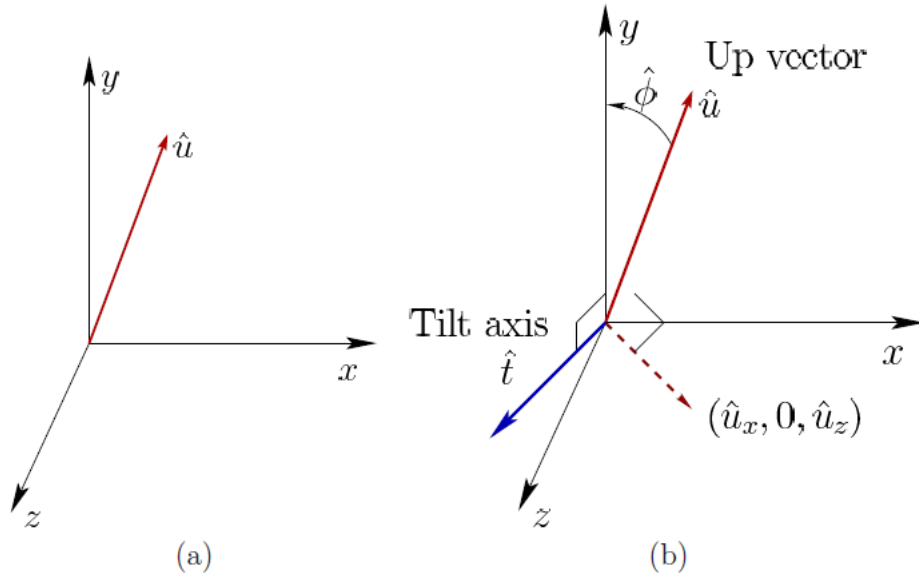


图 9.4: (a)倾斜误差导致 y 轴与使用估计的 $\hat{q}[k]$ 旋转感知测得的向上矢量之间的差异以获得 $\hat{u}$ 。 (b)倾斜轴与 $\hat{u}$ 垂直; 围绕倾斜轴的 $-\hat{\phi}$ 旋转将使它们对齐, 从而消除了倾斜误差。

不幸的是, 没有传感器直接测量“向上”。实际上, 由于重力作用在传感器上, 所以加速计用于测量“向上”方向, 导致向上加速度的感觉大约为  $9.8\text{m/s}^2$ 。问题在于它也会响应刚体真正的线性加速度, 并且由于爱因斯坦等价原理的原因, 这不能与重力分开。它测量重力和真正的线性加速度的矢量和, 如图 9.5 所示。一个简单的启发式方法就是只有当它的幅度接近  $9.8\text{m}^2$  时[75], 才将加速度计输出信任为“向上”方向的估计值。这可以对应于刚体静止的常见情况。然而, 这种假设是不可靠的, 因为向下和横向线性加速度可以组合以提供接近  $9.8\text{m}^2$  的输出量值, 但是具有远离“向上”的方向。可以同时考虑其他传感器的输出或“上”似乎改变的速率来构建更好的启发式方法。

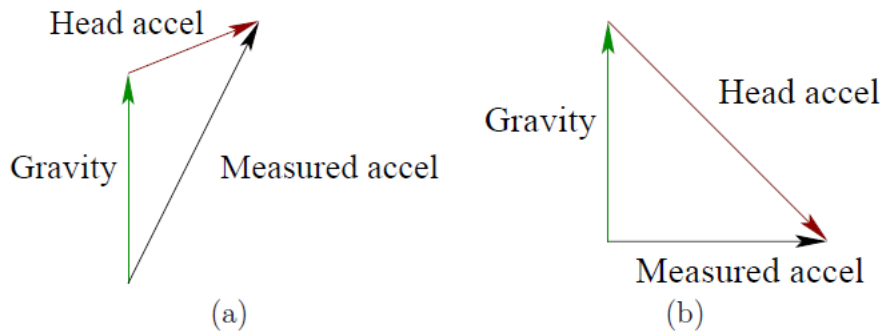


图 9.5: (a)没有重力传感器; 加速度计测量由重力引起的视在加速度和身体的真实加速度的矢量和。 (b)仅在幅度约为  $9.8\text{m}^2$  时接受读数作为重力的简单启发式算法在某些情况下会失败。

假定加速度计产生重力方向的可靠估计, 则通过使用(3.34)从加速度计输出 $\hat{a}$ 计算向上矢量 $\hat{u}$ 以获得

$$\hat{u} = \hat{q}[k] * \hat{a} * \hat{q}[k]^{-1}. \quad (9.20)$$



## 偏航校正

剩余漂移误差分量由“罗盘”检测，该“罗盘”输出位于世界  $xz$  平面的矢量，并始终指向“北”。假设这是  $\hat{n} = (0, 0, -1)$ 。传感器输出再一次出现在身体坐标系中，需要通过  $q$  进行变换。 $\hat{n}$  和  $-z$  轴之间的差异是导致的偏航漂移误差。

就像“up”传感器一样，现实世界中没有“指南针”。相反，有一个磁力计，它可以测量一个三维磁场矢量： $(m_x, m_y, m_z)$ 。假设这是用来测量地球的磁场。事实证明，场向量不会“指向”北极。地球的磁场产生的 3D 矢量通常不在水平面上，从而导致倾斜的角度。因此，第一个问题是传感器输出必须投影到  $xz$  平面中。厄瓜多尔居民可以享受近乎水平的磁场矢量；然而，在芬兰，它们更接近于垂直方向；见图 9.6。如果磁场矢量接近垂直，那么水平分量可能变得太小而无用。

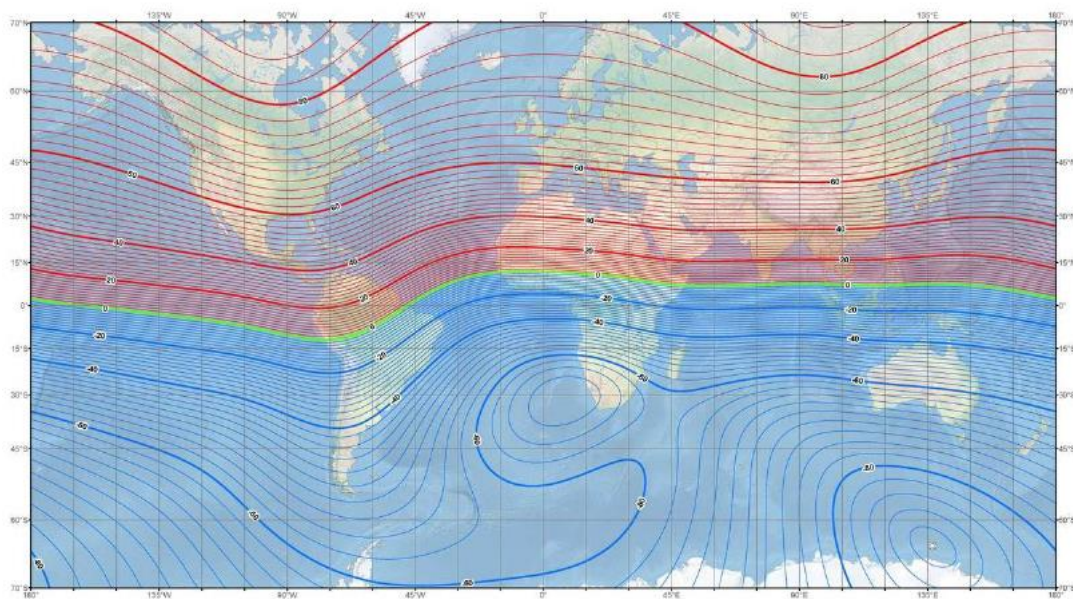


图 9.6：地球磁场矢量的倾角在地球上变化很大。（地图由 NOAA / NGDC 和 CIRES 开发）

另一个问题是水平面上的投影矢量不指向北，导致偏角；这是偏离北方。幸运的是，真正的北方的参考并不重要。只需要配准传感器输出记录在登记阶段提供一个固定的偏航基准。

最重要的问题是磁力计测量所有磁场源的矢量和。除了地球之外，建筑物由于铁磁金属而产生自己的场。此外，这种材料通常存在于包含传感器的电路板上。对于这种情况，场与传感器一起移动，产生恒定的矢量偏移。作为磁场源的材料被称为硬铁。其他材料扭曲穿过它们的磁场；这些被称为软铁。磁力计校准方法主要考虑由于硬铁偏置和由于软铁偏置引起的偏心造成的偏移[92,155]。

在这些磁力仪校准完成之后，偏航漂移误差可以从大多数位置以几个精度进行估算，这足以保持偏航误差逐渐积累。但仍然会存在问题。如果在传感器附近放置强场，则读数将取决于小的位置变化。这可能导致测量的方向随着刚体来回平移而改变。另一个问题是，在一些建筑物位置，地球磁场的矢量和由建筑物产生的场可能近似为零(如果它们具有相似的大小并指向相反的方向)。在这种不幸的情况下，磁强计不能产生用于偏航漂移误差检测的有用输出。

## 滤波

使用检测到的漂移误差，滤波的工作方式与第 9.1 节中所述的相同。补充过滤器(9.10)升级为与四元数一起使用。用  $\alpha$  来表示插值变得稍微复杂一些。令  $(v, \theta)$  表示方向  $\hat{d}[k]$  的轴角表示，其是估计的漂移误差(四元数值)。令  $q(v, \alpha \theta)$  表示由  $v$  轴和角度  $\alpha \theta$  给出的四元数。对于较小的  $\alpha$  值，这可以被认为是“向”  $\hat{d}[k]$  的一小步。

四元数互补滤波器变成

$$\hat{q}_c[k] = q(v, -\alpha \theta) * \hat{q}[k], \quad (9.21)$$

其形式与(9.12)相似。2D 情况下的简单减法已经在上面通过乘以左侧的反四元数来取代。估计的漂移误差  $\hat{d}[k]$  通过将估计的倾斜和偏航误差相乘而获得。或者，他们可以分别对互补滤波器做出贡献，每种都有不同的增益，甚至与来自更多来源的漂移误差估计相结合 [197]。

## 设置视点

为了预测或图像变形的目的，虽然可能需要调整以考虑替代时间，但使用估计方位  $\hat{q}[k]$  设置视点，如 7.4 节所述。令  $\hat{q}(t)$  表示时间  $t$  的估计方向。就 3.4 节的转换而言，我们刚估计了  $R_{eye}$ 。为了计算正确的观点，需要进行反变换。因此， $\hat{q}^{-1}(t)$  将正确转换模型以考虑估计的视点。

## 调试提示

将追踪的方向连接到视点时，程序员经常犯错误。图 9.7 显示了一个常见错误的表格。

Yaw	Pitch	Roll	Error
+	+	+	None
-	+	+	L/R mix, flipped $x$
+	-	+	L/R mix, flipped $y$
+	+	-	L/R mix, flipped $z$
+	-	-	Inverse and L/R mix, flipped $x$
-	+	-	Inverse and L/R mix, flipped $y$
-	-	+	Inverse and L/R mix, flipped $z$
-	-	-	Inverse

图 9.7：帮助调试常见视点转换错误的一览图。每个+表示当执行偏航，俯仰或滚动时，虚拟世界移动正确的方式。每个 - 意味着它以相反的方式移动。如图所示，第一种情况是正确的，其他的都是错误。“L/R 混合”意味着左手和右手坐标系混合。

为了确定变换是否被正确应用，人们应该戴上头戴式设备并尝试围绕三个典型轴旋转：纯偏航，纯俯仰和纯翻滚。让+表示世界正在相对于头部旋转正确移动。让 - 表示它似乎向相反的方向移动。图 9.7 显示了八个可能的结果以及每个问题最可能的原因。

## 头部模型

头部运动的转换部分尚未得到解决。理想情况下，虚拟世界中的头部应与现实世界中的高度相同。这可以通过转换  $T_{eye}$  矩阵来处理(3.36)。

我们还必须说明，随着头部转动，眼睛也会改变其位置。例如，在偏航头部运动中(点头“否”)，瞳孔在  $x$  方向上移动几厘米。更准确地说，它们沿水平面上的圆弧移动。为了更接近模仿现实世界，可以通过根据虚拟头部模型改变旋转中心来模拟眼睛在空间中的移动 [3]。这个技术到 9.3 节将被使用，其中位置是由更多的传感器估算的。

回想 3.5 节，首先考虑了 cyclopean 的观点，然后通过插入  $T_{left}$ (3.50)和  $T_{right}$ (3.52)来应用

水平偏移来修改处理左眼和右眼。以类似的方式，可以增加  $y$  和  $z$  方向上的偏移来考虑来自旋转头的位移。结果是在  $T_{\text{right}}$  和  $T_{\text{left}}$  之前或之后插入以下内容：

$$T_{\text{head}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & p \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9.22)$$

其中  $h$  是高度参数， $p$  是突出参数。见图 9.8。这个想法是选择对应于头部旋转中心的  $h$  和  $p$ 。参数  $h$  是沿着  $y$  轴从旋转中心到眼睛高度的距离。典型值是  $h=0.15\text{m}$ 。突起  $p$  是从旋转中心到环眼的距离。一个典型值是  $p=-0.10\text{m}$ ，这是负值，因为它与  $z$  轴相反。当用户旋转他的头部时，使用假头部模型来近似眼睛的位置，然而这并不是完美的。如果躯干移动，那么这个模型完全失效，导致真实和虚拟世界头部运动之间的不匹配。尽管如此，这款头戴机型目前在三星 Gear VR 等流行设备中使用。

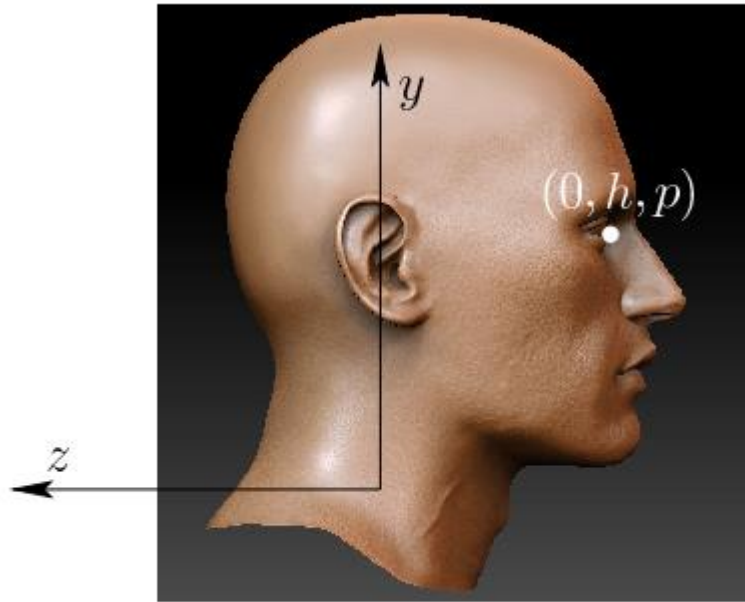


图 9.8：为了获得头部模型，移动旋转中心使得方向改变引起眼睛的合理平移。高度  $h$  沿着  $y$  轴，并且点  $p$  沿着  $z$  轴(这将导致结果为负值)

头部中心的高度也存在问题。用户可能坐在现实世界中，但站在虚拟世界中。这种不匹配可能会让人体验不佳。不管 VR 提供的视觉刺激，大脑知道身体是由于本体感觉而坐着的。如果用户站立，则可以设定头部中心高度，以使眼睛处于与现实世界中相同的高度。对于全六自由度追踪的情况，甚至也会存在这个问题，接下来将介绍这个问题，那就是用户可能会坐着，可以添加一个垂直偏移量以使他看起来像站在 VR 中。

### 9.3 追踪位置和方向

本部分介绍追踪运动刚体的所有 6 个自由度，其中最重要的情况是头部追踪。为了方便起见，我们将把身体的位置和方向称为姿势。如(9.22)所述，六自由度追踪使得  $T_{\text{eye}}$  从 3.4 可以完全从传感器数据中获得，而不是从头部模型发现位置。通过估计位置，随着用户将头部左右移动，强大的视差深度提示变得更强烈。她甚至可以接近一个小物体，并从任何角度观察它，例如从上面，下面或侧面看。本节中的方法也可用于追踪空间中的手或在 VR 体验中操



作的对象。

## 为什么不整合加速度计？

试图用 IMU 单独完成 6 自由度追踪似乎非常的自然。回想图 9.5，加速度计测量真实线性加速度和重力加速度的矢量和。如果将重力分量从输出中减去，如启发式地完成倾斜校正，则看起来剩余部分是纯粹的身体加速度。为什么不简单地整合这个加速两次以获得位置估计？问题在于，漂移误差率比陀螺仪大得多。简单的校准误差会导致陀螺仪外壳中的漂移误差线性增长，因为它是一次积分的结果。经过双重积分后，校准误差会导致二次增长的漂移误差。在几分之一秒后，这个误差在实践中变得难以忍受。此外，真实的身体加速度无法准确提取，特别是当身体快速旋转时。最后，随着漂移的累积，可以使用什么传感器来估计位置漂移误差？单靠 IMU 无法帮助。请注意，它甚至不能以恒定速度区分运动，包括零运动，这与我们的前庭器官相同。尽管存在缺陷，现代 IMU 仍然是六自由度追踪系统的重要组成部分，因为它们具有高采样率和精确处理旋转部件的能力。

## 制造你自己的浪潮

基于 IMU 的追踪方法是被动的，因为它依赖于环境中已经存在的信息来源。相反，可以通过将波传播到环境中的方法。由于人类在相同的环境中工作，所以可以感知的波如光和声音都不是优选的。相反，主动追踪系统中的常见能源包括红外线，超声波和电磁场。

考虑从扬声器发射超声波脉冲(高于 20,000 赫兹)并使用麦克风收听其到达。这是一个发射探测器对的例子：扬声器是发射器，而麦克风是探测器。如果时间测量在源和目的地之间同步，则可以计算到达时间(TOA 或飞行时间)。这是脉冲通过发射器和检测器之间的距离  $d$  所花费的时间。根据介质中的已知传播速度(超声波 330 m / s)，估计距离  $d$ 。超声系统的一个不好的限制是表面之间的混响，导致在每个检测器处多次接收脉冲。

在正确运行时，探测器的位置可以缩小到以发射机为中心的半径为  $d$  的球体;见图 9.9(a)。

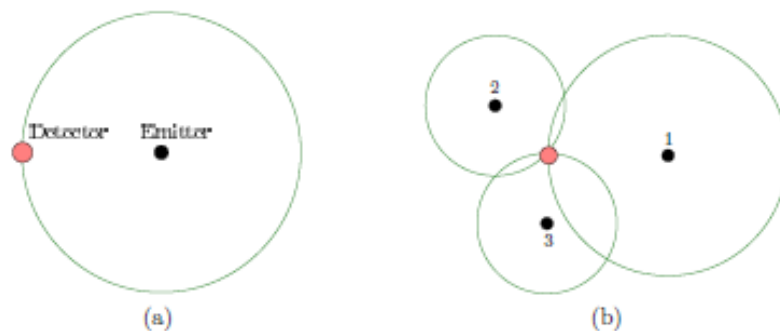


图 9.9: 三边测量的原理使得探测器的位置可以通过估计到已知发射器的距离来确定。图示为一个 2D 示例: (a)来自单个发射器的探测器可以在沿着圆的任何地方; (b)使用三个发射器，该位置是唯一确定的。

通过使用两个发射器和一个麦克风，位置被缩小到两个球体的交点，产生一个圆圈(假设发射器位置已知)。使用三台变送器时，位置被缩小到两点，并且有四个或更多变送器，位置是唯一确定的。发射器和探测器的角色可以很容易地颠倒过来，以便被追踪的物体携带发射器，并将几个接收器放置在其周围。将这些测量结合起来以确定位置的方法称为三边测量。如果使用无线电，光线或红外线等电磁波代替超声波，则即使不可能直接测量传播时间，仍可以应用三边测量法。如果发射机幅度是已知的，则可以基于功率退化来估计距离，而不是 TOA。或者，可以发射时变信号，并且当接收到的信号叠加到发射信号上时可以估计其反射相移。

如果检测器不知道脉冲开始的准确时间，那么他们可以比较它们之间到达时间的差异；这被称为到达时间差(TDOA)。这组可能的位置是双曲面而不是球面。然而，双曲面片可以与多个发射器——检测器交叉以获得多点定位的方法。这在第二次世界大战的迪卡导航系统中用于定位船只和飞机。我们的耳朵也使用这个原理来定位声音的来源，这将在第 11.3 节中介绍。



图 9.10: (a)磁偶极子提供一个随位置改变而改变其大小和方向的场。(b)Razer Hydra 游戏控制器系统，使用基站产生弱磁场，使其能够追踪控制器的位置。

最后，一些方法可以通过发射在追踪区域上变化的复杂场来追踪位置。例如，通过创建一个磁偶极子(可能用信号编码以区别于背景场)，可以在野外估计场中的身体位置和方向;见图 9.10(a)。 Razer Hydra 追踪系统中的这种原理被用于产生磁场的基站中的视频游戏;见图 9.10(b)。一个缺点是该场可能在每个环境中变得不可预测地扭曲，导致直线运动被估计为弯曲。请注意，这些要求与 9.2 节中使用磁强计进行偏航校正所需要的相反;在这种情况下，场地需要在追踪区域保持不变。为了估算位置，该领域应该在不同地点有很大差异。

## 能见度

6 自由度追踪最强大的范例是可视性。这个想法是识别称为特征的物理世界的特殊部分，并计算它们沿着视线射线到达已知位置的位置。图 9.11 显示了一个受相机启发的例子，但也可以使用其他硬件。追踪的一个关键方面是区别。如果所有功能看起来都是相同的，那么在追踪过程中可能难以确定和维护“哪个是哪个”。随着时间的推移，每个特征都应该被分配一个独一无二的标签，因为世界上的刚体移动。相互混淆的特征可能会对身体姿势造成灾难性的糟糕估计。

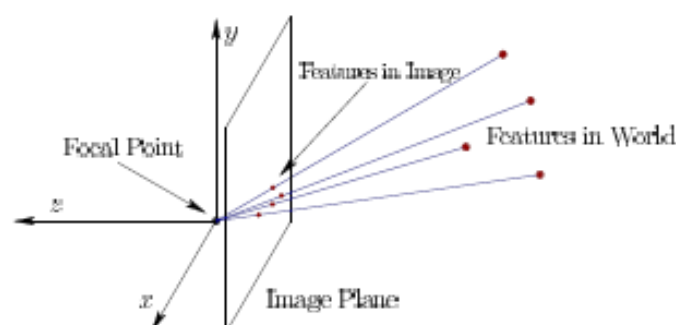


图 9.11: 真实世界包含特殊的特征，这些特征被确定为位于通过透视投影连接到焦点的线段上。

用于检测功能的最常见的传感器是数码相机。检测，标记和追踪功能是计算机视觉或图像处理中的常见任务。有两个功能选项：

1.自然特征：在追踪过程中会自动发现这些功能，分配标签并进行维护。

2.人工特征：将功能设计并放置到环境中，以便可以轻松检测到这些功能，与预先分配的标签进行匹配并进行追踪。

自然特征是有利的，因为没有设置成本。环境不需要设计。不幸的是，它们也更加不可靠。使用相机，这被认为是一个硬计算机视觉问题，因为它可能像人类视觉系统一样具有挑战性。对于某些物体，纹理和光照条件，它可以很好地工作，但要使其可靠地用于所有可能的设置是非常困难的。想象一下，试图在空白的墙上找到并追踪功能。因此，人工特征在产品中更为常见。

对于人造特征，最简单的解决方案之一是将特殊标签打印到要追踪的物体上。例如，可以在对象上打印明亮的红色点，然后在图像中以红色斑点的形式扫描它们的外观。为了解决可区分性问题，可能需要多种颜色，例如红色，绿色，蓝色和黄色点。如果这些颜色自然存在于图像的其他部分，可能会出现問題。更可靠的方法是设计一个与图像其余部分明显不同的特定标签。这样的标签可以被编码以包含大量的信息，包括唯一的识别号码。最常见的编码标签之一是 QR 码，图 9.12 显示了一个例子。



图 9.12：示例 QR 码，可以打印并用作人工特征。(来自维基百科的图片)

到目前为止描述的特征被称为被动，因为它们不释放能量。希望在这个世界上有足够的射线，以便足够反射该特征并进入相机传感器。更可靠的替代方案是设计发出自己的光的主动功能。例如，彩色 LED 可以安装在头盔或控制器的表面上。这是以需要电源并增加整体物体成本和重量为代价的。此外，其工业设计可能会因为它可能像圣诞树一样点亮而受损。



图 9.13：Oculus Rift 包含隐藏在 IR 透明塑料后面的 IR LED。(由 iFixit 拍摄)

## 红外线遮掩

幸运的是，所有这些技巧都可以转移到光谱的红外(IR)部分，以便相机可以看到这些特征，但对人不可见。图案可以绘制到高度反射红外能量的物体上，或者，IR LED 可以安装到设备上。Oculus Rift 就是这种情况，红外 LED 甚至隐藏在透明的红外能量塑料后面，但对人类来说却是黑色的，见图 9.13。



在某些情况下，可能很难将 LED 安装在物体上，例如追踪整个人体的细微运动。这称为 MOCAP 或动作捕捉，在第 9.4 节中进行了描述。在 MOCAP 系统中，强大的红外线发光二极管位于相机周围，以便它们照亮放置在场景中的反光标记。每个标记都可以被想象成光谱 IR 部分的球面镜。一个缺点是它的范围受到限制，因为红外能量必须从摄像机位置移动到目标并再次返回。由于能量作为距离的函数以二次方式消散，所以将距离加倍到达相机的四分之一的能量水平。

在这一点上，我们很自然地想知道，既然最终的图像处理问题微不足道，为什么要捕捉整个图像。主要原因是低成本数码相机和图像处理软件的普及。为什么不简单设计一个产生二进制读数的发射器-检测器对，指示可见光束是否被遮挡？这正是探测光束在自动车库门系统中的工作原理，以确保门不会关闭时夹到人：红外 LED 向检测光电二极管发射能量，该光电二极管本质上是一个开关，当它接收到足够的能量水平其目标波长(在这种情况下是 IR)。为了减少能量消耗的量，镜子或镜头可用于聚焦能量。

更好的是，红外激光可以直接瞄准探测器。接下来的任务是使用镜头和移动镜子，以便从某个固定位置可见的每个检测器都会在某个点照亮。光束可以使用透镜从一个点扩展到一条线，然后使用旋转镜在线移动。这是 HTC Vive 的灯塔追踪系统的基础，本节稍后会对此进行介绍。

## N 点透视(PnP)问题

使用  $n$  个观察特征需要“移动”刚体。这被称为  $n$  点透视(或 PnP)问题。我们可以借用第 3 章的大部分数学原理，然而，在这里我们考虑的是真实世界中的身体位置，而不是虚拟世界。此外，我们有一个反向问题，即根据图像中的点确定身体位置。到目前为止，已经考虑了相反的问题。对于第 7 章中的视觉渲染，图像是基于(虚拟)世界中已知的身体位置生成的。

这些功能可以放置在身体上或周围的世界中，具体取决于传感方法。现在假设他们在身体上。每个特征对应于具有在身体的框架中定义的坐标的点  $p = (x, y, z)$ 。设  $T_b$  是包含姿势参数的齐次变换矩阵，其假定是未知的。如(3.22)中所示，将变换  $T_b$  应用于点  $p$  可以将其置于真实意义上的任何位置。回想一下变换链公式(3.41)，它进一步确定身体上的每个点在图像中出现的位置。矩阵  $T_{eye}$  保持相机姿态，而  $T_{vp}$  和  $T_{can}$  包含透视投影并将投影点转换为图像坐标。

现在假设已经观察到一个特征位于图像坐标中的位置  $(i, j)$ 。如果  $T_b$  未知，但给出了所有其他变换，则将有 6 个独立参数进行估计，对应于 6 个 DOF。观察  $(i, j)$  为变换链(3.41)提供了两个独立的约束条件，其中一个为  $i$ ，另一个为  $j$ 。因此刚体失去 2 个自由度，如图 9.14 所示。这是 P1P 问题，因为  $n$  是特征的数量。



图 9.14：每个可见的特征消除了 2 个 DOF。在左侧，可以看到单个特征，而刚刚生成的刚体只剩下 4 个自由度。在右侧，两个特征是可见的，导致只有 2 个自由度。这可以如下所示。如果还施加了一些旋转，则接触两个分段的边缘可以在保持其长度的同时来回移动。围绕边缘共有的轴旋转提供第二个 DOF

P2P 问题对应于观察图像中的两个特征并导致四个约束。在这种情况下，每个约束消除了两个自由度，导致只剩下两个自由度，见图 9.14。继续下去，如果观察到三个特征，则对于 P3P 问题，保留零自由度(除了在身体上选择共线特征的情况)。看起来问题完全解决了；然

而，零自由度允许多种解决方案(它们是解决方案空间中的孤立点)。P3P 问题对应于试图将给定的三角形放入由光线形成的金字塔中，以便每个三角形顶点触及不同的光线。这通常可以通过四种方式完成，难以形象化。想象一下，试图将由奶酪制成的高而薄的金字塔(单面)切片，以使四个不同的切片具有完全相同的三角形尺寸和形状。P4P 和 P5P 的情况也会导致模糊的解决方案。最后，在 P6P 的情况下，如果没有四个特征共面，总会得到独特的解决方案。所有的数学细节都在[354]中计算出来。

PnP 问题已经被描述为理想的情况，即对身体上的特征点进行完美的坐标分配，并通过成像过程完美地观察这些特征点。在实践中，由于诸如传感器噪声，图像量化和制造容差等因素而产生小误差。这会导致估计姿态的模糊和错误，这可能与正确答案有很大差异[281]。因此，在实践中可以使用更多的特征来提高准确性。此外，可以在使用该设备之前应用诸如光束法平差[111,280,325]之类的校准过程，使得可以在执行姿态估计之前更精确地分配特征点位置。使用 RANSAC 可以提高鲁棒性[77]。

## 基于相机的实现

可见性问题可以使用相机，通过两种常规方式来解决，如图 9.15 所示。相机镜头是类似于第 3 章中图 3.14 的眼镜框。世界固定的相机通常是静止的，这意味着相机镜头相对于世界不会移动。可以使用单个变换来将从相机框估计的对象姿态转换为简单的世界框。例如，对于 Oculus Rift 头戴式设备，头部姿态可以转换为世界坐标系，其中-z 方向指向相机，y 为“上”，并且位置位于相机的中心追踪区域或基于用户初始头部位置的合适默认值。对于固定对象的摄像机，估计的姿态是从相机画面转换到世界画面的，是从世界上仍然固定的特征得出的。例如，如果将 QR 码放在墙上，就可以获得这种情况。

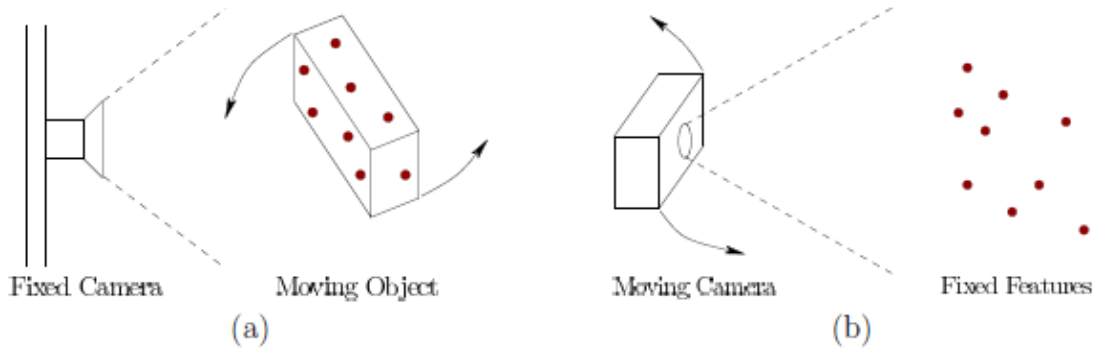


图 9.15: 摄像机放置的两种情况: (a)世界固定的摄像机是静止的，并且物体相对于它的运动使用物体上的特征进行估计。(b)物体固定摄像机经常处于运动状态，理想状态下的特征固定在世界坐标系中。

就像 IMU 一样，校准对于提高传感精度非常重要。以下齐次变换矩阵可以应用于相机产生的图像：

$$\begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9.23)$$

矩阵中出现的五个变量称为相机的内部参数。 $\alpha_x$  和  $\alpha_y$  参数处理缩放， $\gamma$  处理剪切， $u_0$  和  $v_0$  处理光轴偏移。这些参数通常是通过拍摄所有尺寸和距离都经过仔细测量的物体的图像并进行最小二乘估计来选择减少平方和误差的参数(如 9.1 节所述)来估算的。对于广角镜头，可能需要进一步校准以克服光学失真(回忆第 7.3 节)。

现在假设在图像中观察到一个特征,也许使用某种形式的斑点检测从图像的其余部分提取与其对应的像素[280,318]。这对全局快门相机来说是最简单的,因为所有像素都将对应相同的时间。在卷帘快门的情况下,可能需要变换图像以消除运动的影响(回忆图 4.33)。观察到的特征的位置被计算为斑点像素位置的统计量。最常见的情况是,使用所有斑点像素的平均值,导致非整数图像坐标。许多问题会影响性能: 1)由于每个像素的图像坐标都是整数,所以会出现量化误差;2)如果该特征没有覆盖足够的像素,则量化误差更差;3)照明条件的变化可能使得难以提取特征,特别是在自然特征的情况下;4)在某些角度,图像中两个或多个特征可能变得接近,使得难以分离它们相应的斑点;5)由于各种特征进入或离开摄像机视图,所得到的估计姿态可能会跳跃。此外,误差往往沿光轴方向变大。

## 基于激光的实现

通过设计一个特殊的发射器-探测器对,能够在很远的距离上准确地解决可见性问题。这是通过 2016 年 HTC Vive 的灯塔追踪系统和 1989 年的明尼苏达扫描仪[301]完成的。图 9.16 显示了 HTC Vive 的灯塔追踪硬件,图 9.17(a)有效地模拟了摄像机的操作。



图 9.16: HTC Vive 头戴式设备中使用的基于激光的追踪方法: (a)基站包含旋转鼓,可发出水平和垂直的红外光片。左上方出现一排红外 LED,提供同步闪光。(b)头戴式头盔前面的口袋中的光电二极管检测入射的 IR 光。

如果基站是摄像机,则扫描垂直条纹将对应于估计对应于该特征的像素的行,见图 9.17(a)。同样,扫描的水平条纹对应于像素列。旋转鼓的旋转速率是已知的,并且与相机帧率类似。精确的时间记录为光束击中每个光电二极管。

考虑相对于基站的极坐标(距离和角度)。使用扫描的角速度和相对定时差,可以容易地估计从基站“观察”的特征之间的角度。尽管特征之间的角度很容易确定,但它们相对于来自基站的某个固定方向的角度必须确定。这是通过一系列同时脉冲的红外 LED 实现的,以便所有光电二极管都能检测到闪光(如图 9.16(a)所示)。例如,这可以对应于每个光束处于 0 方向的时刻。基于从闪光直到光束撞击光电二极管的时间以及已知的角速度,确定所观察特征的角度。为了减少时间漂移误差,可以在操作期间周期性地使用闪光灯。

与照相机一样,从基站到特征的距离也是未知的,但可以通过解决 PnP 问题来确定。也可以使用多个基站,其方式与使用多个摄像头或多个眼睛来推断深度相当。结果是在大面积上进行精确追踪,如图 9.17(b)所示。

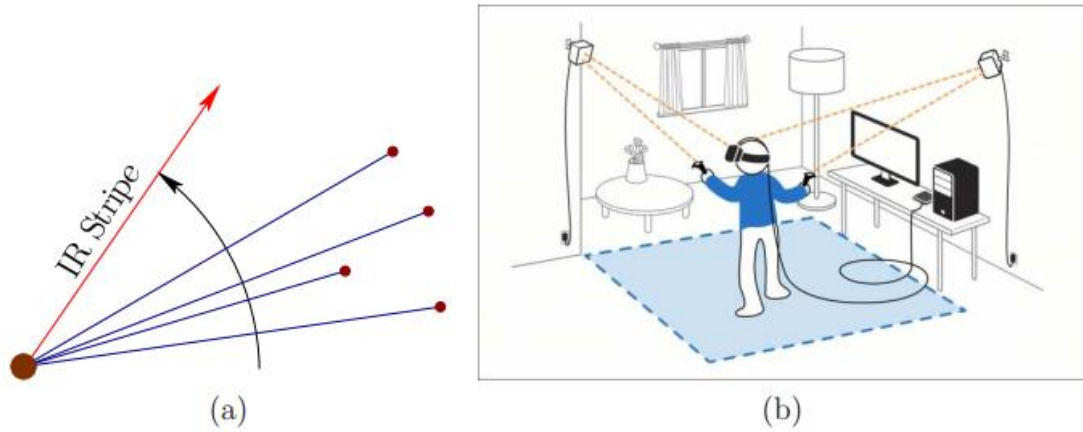


图 9.17: 这是基于激光追踪方法中红外条纹角度扫描的 2D 视图(如 HTC Vive)。对应于顶视图, 其中垂直条纹围绕底座旋转偏航旋转。在这种情况下, 观察水平方向上的角度位置, 类似于相机图像的列坐标。这也可以对应一个方面在这种情况下, 观察垂直条纹以俯仰旋转和垂直方向上的角度位置自旋。当光束击中光电二极管的特征时, 由于自同步闪光以来的旋转速率和时间, 所以方向是已知的。(b)通过将两个基站放置在追踪区域的角落处的极点之上, 可以精确地追踪头盔和控制器的区域。(由 Chris Stobing 所画)

## 滤波

如第 9.2 节所述, 来自传感器的输出通过滤波方法随时间合并以维持预测。在当前的设置中, 可以通过组合可见性信息和 IMU 的输出来维持姿态。对于姿态的方向分量, 可以使用(9.10)中的互补滤波器。相机为检测方向漂移误差提供了额外的来源。摄像机光轴是偏航误差估计检测的直接参考, 这使得它成为磁力仪的一个合适的替代品。如果相机倾斜度已知, 则相机还可以提供精确的倾斜误差估计。

由于陀螺仪提供的角速度的准确, 高频估计, IMU 对获得高度精确的定向追踪至关重要。如果摄像机或灯塔系统的帧速率非常高, 则可能存在足够的传感器数据以进行精确的位置追踪; 但是, 最好直接测量衍生工具。不幸的是, IMU 不能测量线速度。但是, 线性加速度计的输出可以按照本节开始部分的方法进行使用。假设加速度计估计出的世界框架内的主体加速度为(假定重力分量已从加速度计输出中减去)。

$$\hat{\mathbf{a}}[k] = (\hat{a}_x[k], \hat{a}_y[k], \hat{a}_z[k]) \quad (9.24)$$

通过数值积分, 可以通过  $\mathbf{a}[k]$  估计速度  $\mathbf{v}[k]$ 。通过对速度估计进行积分来估计位置  $\mathbf{p}[k]$ 。使用简单欧拉积分的更新方程如下

$$\begin{aligned} \hat{\mathbf{v}}[k] &= \hat{\mathbf{a}}[k]\Delta t + \hat{\mathbf{v}}[k-1] \\ \hat{\mathbf{p}}[k] &= \hat{\mathbf{v}}[k]\Delta t + \hat{\mathbf{p}}[k-1]. \end{aligned} \quad (9.25)$$

请注意, 每个方程实际上同时处理三个分量  $x$ ,  $y$  和  $z$ 。通过在右侧添加  $\frac{1}{2}\hat{\mathbf{a}}[k]\Delta t^2$ , 可以进一步提高第二个方程的精度。

如前所述, 加速度的双重积分导致快速增长的位置漂移误差, 用  $\mathbf{d}_p[k]$  表示。PnP 解决方案检测到的误差提供了  $\mathbf{d}_p[k]$  的估计值, 但可能比 IMU 产生的观测值低很多。例如, 相机可能以 60 FPS 拍摄照片, 而 IMU 可能会以 1000 FPS 记录加速度。

从(9.10)的互补滤波器可以扩展到双重积分的情况下, 我们得到



$$\begin{aligned} p_c[k] &= \hat{p}[k] - \alpha_p \hat{d}_p[k] \\ v_c[k] &= \hat{v}[k] - \alpha_v \hat{d}_v[k]. \end{aligned} \quad (9.26)$$

上式中,  $p_c[k]$ 和  $v_c[k]$ 分别是校正的位置和速度, 它们分别由互补滤波器计算。估计值  $p[k]$ 和  $v[k]$ 使用(9.25)计算。与 IMU 更新相比, 参数  $\alpha_p$  和  $\alpha_v$  控制给予漂移误差估计的重要程度。

方程(9.26)实际上等价于卡尔曼滤波器, 它是具有高斯噪声的线性动力系统的情况以及也受高斯噪声影响的传感器的最佳滤波器(提供可能的最精确的估计)。设  $\omega_d^2$  表示动态系统中估计的高斯噪声的方差, 并设  $\omega_s^2$  表示传感器噪声方差。如果参数选择为

$\alpha_p = \sqrt{2\omega_d/\omega_s}$  和  $\alpha_v = \omega_d/\omega_s$  [119], 互补滤波器(9.26)等效于卡尔曼滤波器。存在各种各样的替代过滤方法; 然而, 相对于校准, 传感器误差模型和特定于设置的动态系统模型, 使用不同滤波方法的影响通常较小。此外, 性能要求主要是基于感知的, 这可能与设计滤波方法的经典标准不同[167]。

一旦过滤器运行, 其姿态估计可用于帮助 PnP 问题。PnP 问题可以通过使用最新的加速度计输出扰动滤波器估计的姿态来逐步解决, 以便观察到的特征完美匹配。可以对姿势进行小的调整, 以便将平方和误差降低到可接受的水平。在大多数情况下, 当 PnP 问题具有模糊的解决方案时, 可见的特征很少, 这可以提高可靠性。如果没有逐步地确定姿态, 可能会灾难性的跳转到另一个 PnP 解决方案。

## 9.4 追踪附属体

许多追踪问题涉及估计一个物体相对于另一个附着的运动物体的运动, 例如眼睛在眼窝内旋转。虽然眼睛在空间被视为一个刚体时可能有六个自由度, 但是一旦给出头部姿态, 它的位置和方向就可以用两个或三个参数完全表征。其他例子包括相对于躯干的头部, 相对于手腕的手部, 以及手指相对于其中间骨头的尖端。基于骨架, 整个人体甚至可以排列成树形的附属体。此外, 也可以通过类似的方式发现其他生物体上的附着体, 例如狗或猴子, 以及机器, 例如机器人或汽车。在汽车的情况下, 车轮相对于车身旋转。在所有这些情况下, 结果都是多体系统。物体相对于彼此姿态的数学表征被称为多体运动学, 并且完全确定它们的速度和加速度被称为多体动力学。

### 人眼追踪

一个多世纪以来, 视觉科学家一直在使用眼动追踪系统来研究眼球运动。VR 的三个主要用途是: 1)如第 5.4 节所述, 完成中心渲染, 以便高分辨率渲染只需要在中心凹上的图像部分执行。2)通过记录追踪数据来研究人类行为, 从而帮助治疗 VR 疾病, 提高注意力和体验的有效性等。3)在 VR 中渲染视线的方向, 以便通过提供眼神接触和指示某人关注的焦点来改善社交互动; 参见 10.4 节。

现如今已经开发了三种通用类型的眼动追踪方法[63,338]。第一种是眼电图(EOG), 它从位于每只眼睛周围的面部皮肤上的多个电极获得测量结果, 电位对应于眼肌活动, 通过滤波来确定相对于头部的眼睛方向。第二种方法使用隐形眼镜, 隐形眼镜包含一个微小的电磁线圈, 导致周围电磁场的潜在变化。第三种方法称为视频眼动(VOG), 它将红外光照射到眼睛上, 并使用相机或光电二极管感测其角膜反射。反射基于 Purkinje 图, 如图 9.18 所示。由于其成本低, 侵入性小, 这是目前最常用的方法。隐形眼镜方法是最准确的; 然而, 也是最不舒服的。

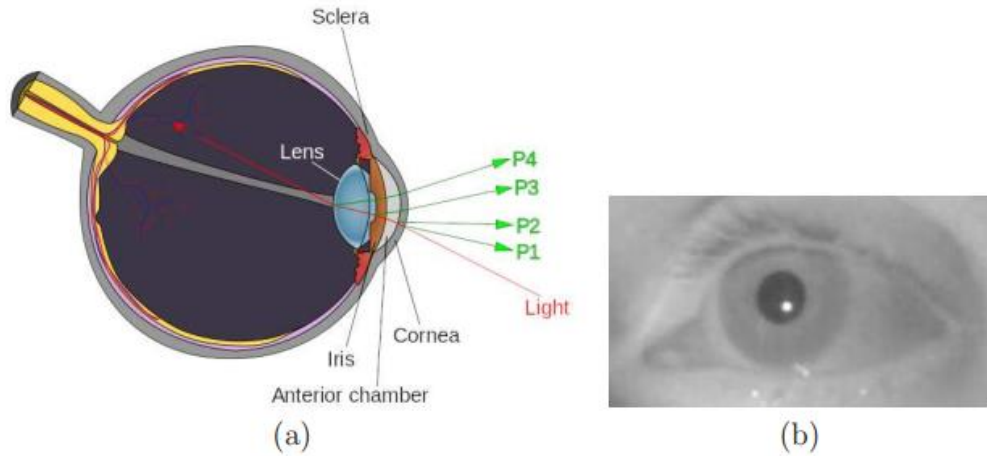


图 9.18: (a)红外光源的第一幅, 有时是第四幅 Purkinje 图像用于人眼追踪(图源于 Wikipedia)(b)第一幅 Purkinje 图像如图所示产生明亮的反射(图源于 Massimo Gneo, Maurizio Schmid, Silvia Conforto, Tomasso D'Alessio)

## 正向运动学

假设眼睛追踪方法估计了相对于人类头骨的眼睛方向, 然后将相应地放置在虚拟世界中。这种转变必须涉及头部和眼部变换的组合。对于一个更复杂的问题, 考虑每个关节处使用躯干的姿势以及骨骼之间形成的所有角度, 在虚拟世界中将手指的位置放准确。要理解这些和其他相关问题如何解决, 我们先介绍一些 2D 的例子。

多体系统的每个主体称为**连杆**, 并且一个主体**连杆**在一个关节处, 这允许它们之间存在一个或多个运动自由度。图 9.19 显示了一个平面物体在附着到另一个平面物体时常见的两种运动方式。**旋转关节**最为常见, 并表示人体肘部所允许的运动。

考虑定义  $m$  条**连杆**,  $B_1$  到  $B_m$ , 并确定两个**连杆**相交点的位置。每个**连杆**上的点是使用其自身的**主体框架**的坐标定义的。在这个框架中, 主体如图 9.20 中的  $B_{i-1}$  所示, 原点位于连接  $B_{i-1}$  到  $B_{i-2}$  的连接点, 而  $x_{i-1}$  轴指向连接  $B_{i-1}$  到  $B_i$  的连接点。将  $B_i$  上的点移动到  $B_{i-1}$  的主体框架中的适当位置, 应用齐次变换

$$T_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & a_{i-1} \\ \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (9.27)$$

这将  $B_i$  旋转  $\theta_i$ , 然后沿  $x$  轴将其平移  $a_{i-1}$ 。对于旋转关节,  $\theta_i$  是一个变量,  $a_{i-1}$  是一个常数。对于棱形关节,  $\theta_i$  是常数,  $a_{i-1}$  是一个变量。

通过应用乘积  $T_2 \cdots T_i$  将  $B_i$  上的点移入  $B_1$  的主体框架中。图 9.21 显示了一个三连杆的例子。移动第一条连杆  $B_1$  进入世界框架, 可以应用一般的 2D 齐次变换:

$$T_1 = \begin{pmatrix} \cos \theta_1 & -\sin \theta_1 & x_t \\ \sin \theta_1 & \cos \theta_1 & y_t \\ 0 & 0 & 1 \end{pmatrix}. \quad (9.28)$$

这种变换只是通过将  $T_1 T_2 \cdots T_i$  添加到矩阵乘积中来移动每个  $B_i$ 。

3D 连杆链在概念上以相同的方式处理, 但代数运算变得更加复杂。有关更多详细信息, 请参见[163]的第 3.3 节。图 9.22 显示了通过允许一对 3D 连杆相互滑动而获得的六种不同



类型的关节。每个连杆都根据关节分配了一个方便的坐标系。每个齐次变换  $T_i$  包含常量和变量的混合，其中变量对应于关节允许的自由度。最常见的分配方案称为 Denavit-Hartenberg 参数[110]。在某些设置中，可能最好将每个  $T_i$  替换为旋转主体的参数化四元数，然后添加一个简单的加法对主体做平移。

连杆树也可以考虑；一个常见的例子是人体躯干作为根部，头部，两条手臂和两条腿是从它延伸出来的连杆。人的手是另一个例子。这种情况下的坐标系通常使用 Kleinfinger-Khalil 参数[148]分配。

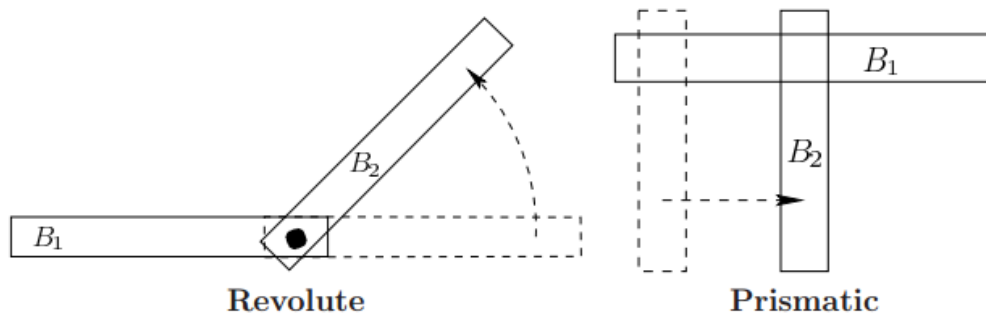


图 9.19: 两种类型的 2D 关节：旋转关节允许一个连杆相对于另一个旋转，并且棱形关节允许一个连杆相对于另一个进行平移。

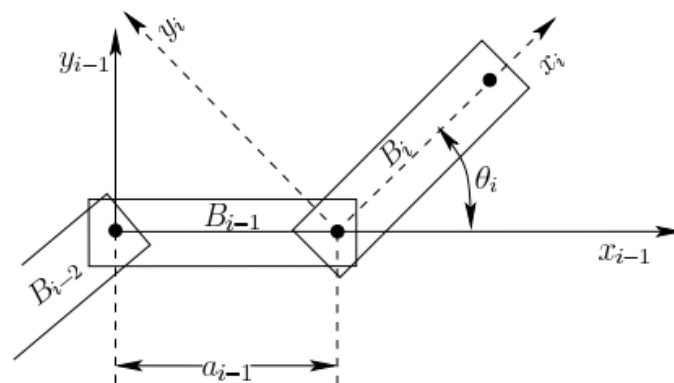


图 9.20: 对于  $1 < i < m$ ，主体框架的每个  $B_i$  基于这些关节连接到  $B_{i-1}$  和  $B_{i+1}$ 。

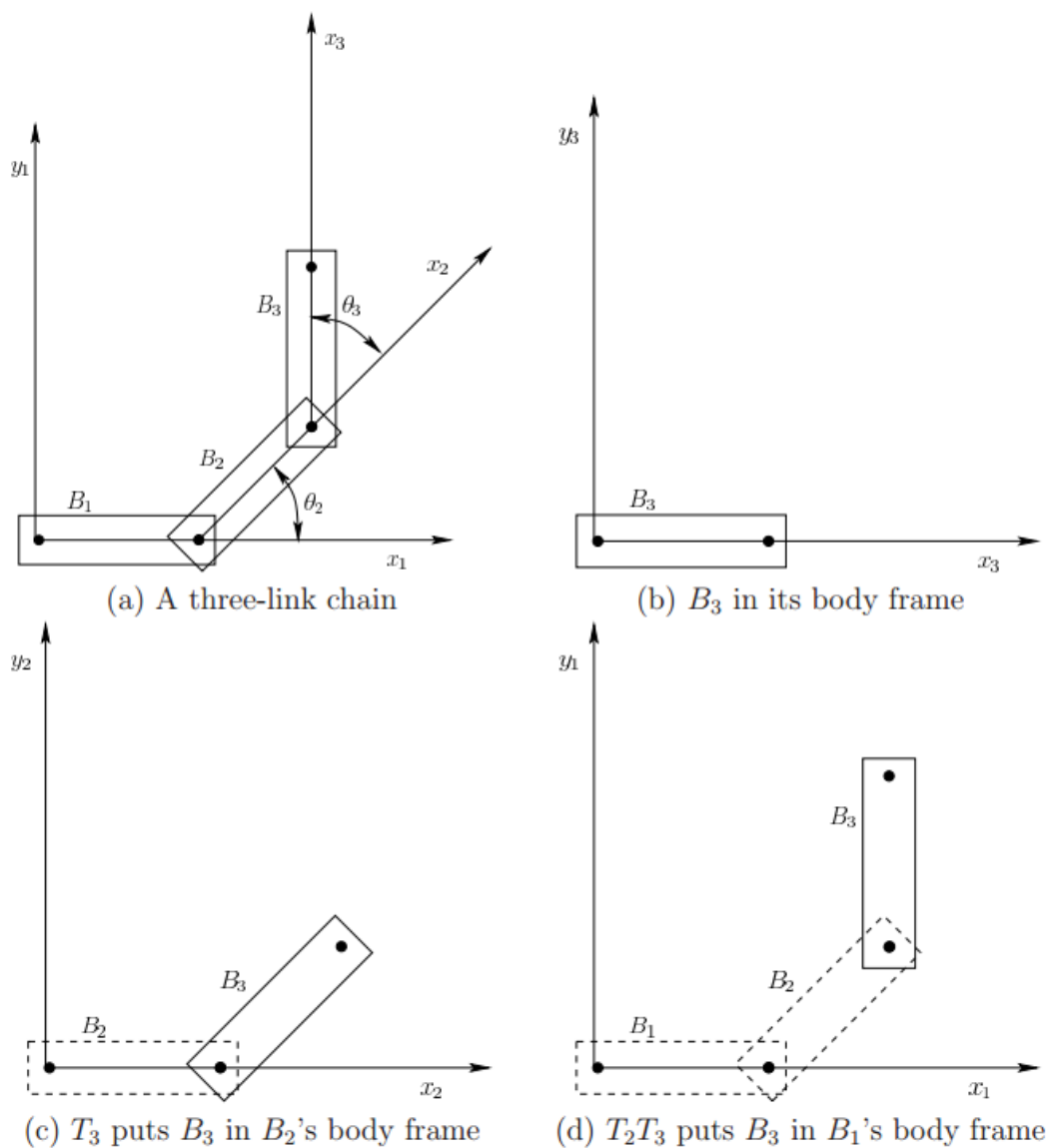


图 9.21: 将变换  $T_2T_3$  应用于  $B_3$  的模型上。如果  $T_1$  是单位矩阵, 那么这产生了  $B_3$  中虚拟世界中的点的位置。

## 约束及逆向运动学

回顾第 9.3 节中的 **PnP** 问题, 该问题涉及到通过观察到的约束来计算物体的姿态。一个类似的问题是通过考虑对主体的约束来确定一连串主体的关节参数。一个常见的例子是仅使用手的姿势来计算手臂连杆的姿势, 我们称之为**逆向运动学问题**(参见[8]和[163]的第 4.4 节)。就 **PnP** 而言, 解的数量可以是无限的, 有限的, 一个或零个。一些 2D 例子如图 9.23 所示。通常, 如果最后一个链路受到限制, 则中间链路的运动自由度随着链路数量的增加而增加。假设连杆不是在一些特殊的单一配置中的情况, *Chebyshev-Grübler-Kutzbach 准则*可以给出自由度的数量[9]。动画视频游戏角色的一个常见问题是保持运动学约束, 例如手抓住门把手, 即使躯干或门正在移动。在这种情况下, *迭代优化*通常用于执行每个联合参数, 直到误差被充分减小。在我们的例子中, 通过测量手和门把手之间的距离来计算误差。

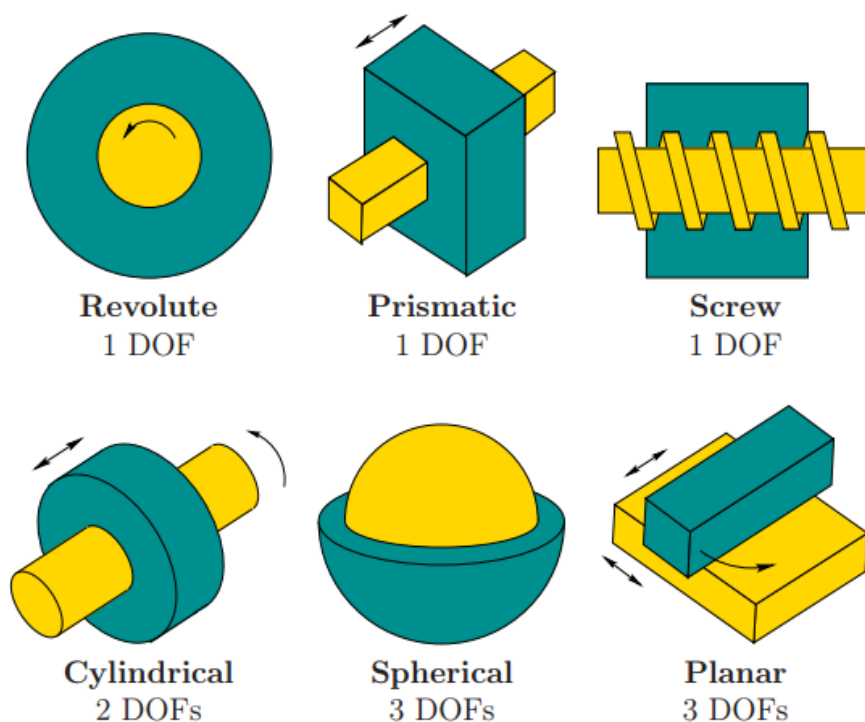


图 9.22: 由两个物体之间的 2D 表面接触生成 3D 关节的类型。

## 运动捕捉系统

附属体的追踪系统使用运动学约束来提高准确性。最常见的应用是追踪人体，对于该人体而言，骨骼在连杆和关节方面是很好理解的[366]。这种动作捕捉系统一直是电影行业的重要技术，因为真人演员的动作被带入虚拟动画世界。操作如图 9.24 所示。第 9.3 节介绍的特征与人体相同，并可在捕捉工作室周围安装摄像头。对于可见度也存在同样的选择，过去十年中最常用的方法是使用带有周围红外 LED 的摄像头并将反光标记放置在演员身上。

为了获得每个主体部位的独特姿势，似乎需要六个特征(回顾第 9.3 节中的 P6P);然而，由于运动学约束，并不需要这么多。如果我们是为了在沿着骨架移动时捕捉皮肤运动，则可以使用额外的特征。这对于面部运动尤其重要。目前许多新的 MOCAP 技术正在开发中。例如，Noitom 开发的系统仅通过将 IMU 放置在身体上来捕捉人体运动。某些系统仅通过摄像头捕捉运动，如 Leap Motion(见图 9.25)用于手部追踪的情况，以及由微软和 8i 用于通过在绿色屏幕上提取轮廓来进行全身追踪的系统。基于现代深度传感器的解决方案在不久的将来也可能会流行起来。一个挑战是降低成本和安装高度准确可靠的系统。

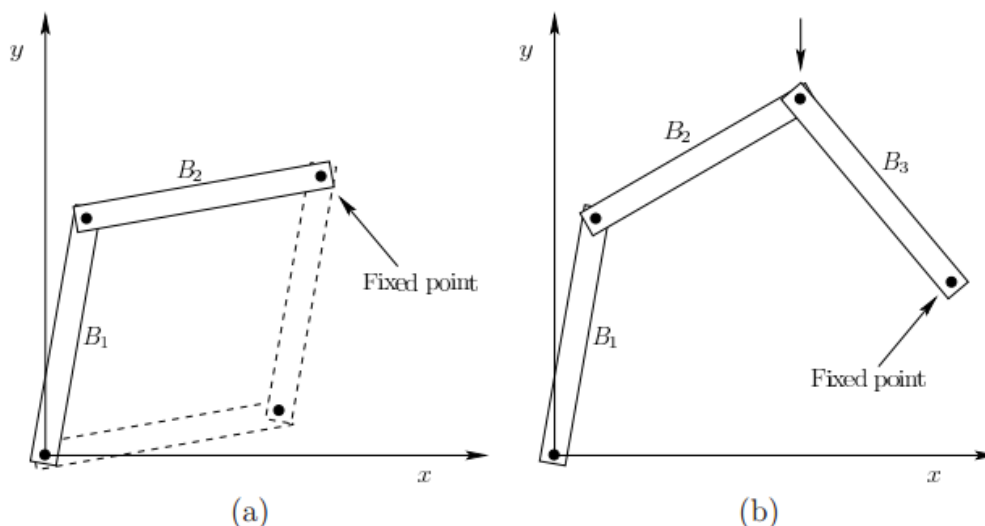


图 9.23: (a)两条连杆的方向可以从固定点的位置推断;然而,如果角度不受限制,还有第二种解决方案。(b)在三条连杆的情况下,当一端固定时,存在一维解决方案族。可以通过向下按顶部接头来观察,这会导致  $B_1$  逆时针旋转。这相当于传统的四杆机构,用于驱动蒸汽机的车轮(第四个“连杆”仅仅是固定的背景)。

## 9.5 环境三维扫描

目前为止,本章介绍了如何使用传感器来追踪一个或多个刚体的运动。在本节中,我们讲述如何使用传感器来构建刚体的几何模型,可能是可移动的,也可能是固定的,如 3.1 节所介绍的。可移动模型通常对应于由用户操控的物体,如剑、锤子或咖啡杯等。这些模型通常由 3D 扫描器构建,以受控的方式从不同角度对物体成像。物体放置在被摄像机或其他传感器包围的平面上,或者放置在转台上,以便从不同的视点角度来观察物体。或者,也可以采取物体保持静止不动而传感器在周围移动,如图 9.26(a)。

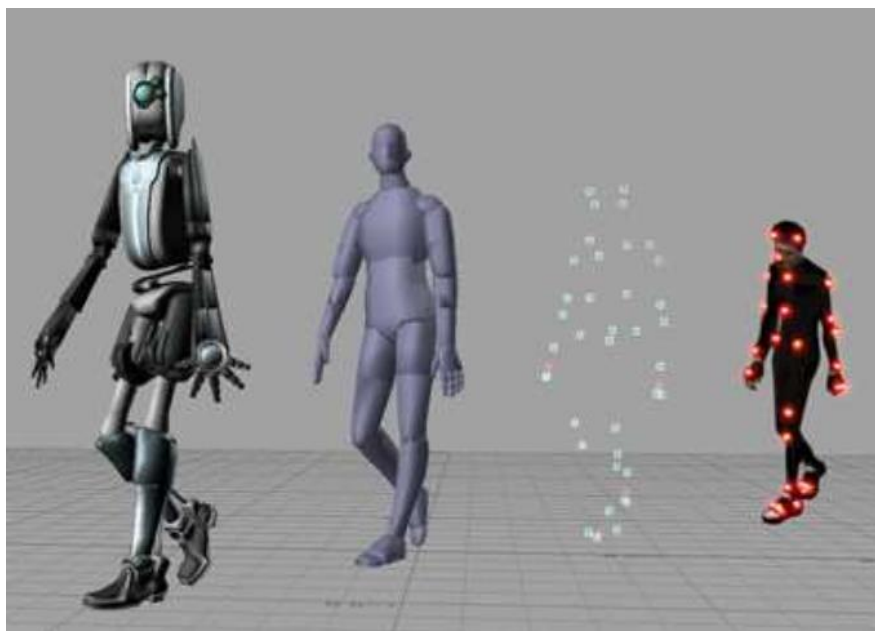


图 9.24: 通过运动捕捉系统(MOCAP),将特征点放置在演员的身体周围。提取其运动并与运动学模型进行匹配。模型中的每个刚体都有一个相关的几何模型,将该模型渲染成最终的动画角色。(图源于 Wikipedia 用户 Hipocrite)

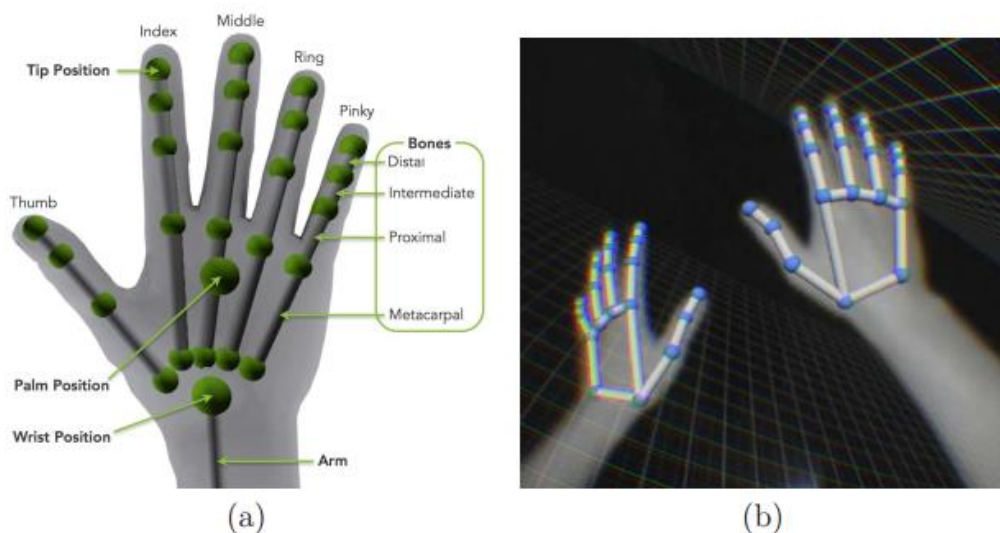


图 9.25: (a)使用 Leap Motion 追踪得到的手部模型 (b) 将追踪模型叠加在实际手部的图像上

## SLAM

3D 扫描器适用于较小的物体，里面的传感器是面朝内部的。对于较大的物体和固定模型，传感器通常是面朝外部的，如图 9.26(b)。静态模型的一个常见例子是建筑物的内部。扫描这些模型对于调查和取证变得越来越重要。这也是经典的机器人领域中的地图构建问题，携带传感器的机器人以导航和避免碰撞为目的建立其环境的 2D 或 3D 表示。机器人通常需要根据传感器估算其位置，这就是所谓的定位问题。VR 的机器人定位和追踪机构基本上是同样的问题，主要的区别在于将已知的动作命令给予机器人，但并没有直接给出其对应的人类意图。机器人经常需要同时完成地图构建和定位的任务，这会导致即时定位与地图构建问题；常用缩写 SLAM 表示。由于定位，地图构建和 VR 追踪问题的相似性，机器人和 VR 之间存在着深层次的联系。因此，许多数学模型，算法和传感技术也有所重叠。考虑到 VR 的大型固定模型的可能用途。可以将其捕获然后给予用户在当前时间或稍后时间的虚拟世界。图像数据可以与模型的三维坐标相结合，生成照逼真的模型(回忆 2.2 节的图 2.14)。这是通过将图像块贴图到模型三角形上来实现的。



图 9.26: (a)Afinia ES360 扫描仪，物体在转台上旋转时可以生成物体的 3D 模型 (b)FARO Focus3D X 330 时一部朝外的扫描仪，用于构建大型环境的精准 3D 模型；它包含一个 GPS 接收器，用于将独立的扫描结果

## 当前场景的实时捕捉

传感器可以用来捕捉用户当前正在体验 VR 的物理世界，而不用去捕捉要传到用户眼中的世界。这允许匹配区域中的障碍物在虚拟世界中呈现，这对于安全性或改善交互性可能是有用的。为了安全起见，可以对匹配区域的边界进行渲染，以提醒用户。像是一杯咖啡洒了或宠物穿过匹配区之类的危险情形也应当指示出来。通过将来自物理世界的固定物体带入虚拟世界可以提高交互性。例如，如果用户坐在桌子前面，则可以在虚拟世界中绘制一张桌子。如果她推一下虚拟桌子，她会感觉到真正的桌子在向后退。这是在 VR 中提供触碰反馈的相对简单的方法。

## 全景体验已经可以满足了吗？

在着手创建周围 3D 世界的大型详细地图之前，我们要考虑是否有必要。如第 7.5 节所述，现如今的全景图像和视频越来越容易捕捉。在一些应用中，建立用户在从彼此接近的多个位置捕捉的全景图之间进行传送的体验可能就足够了。

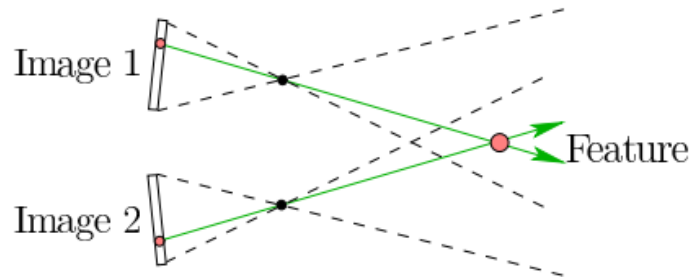


图 9.27：通过使用两个摄像头，立体视觉使得 3D 世界中的特征的位置能够通过来自每个摄像头的相应可见光线相交来确定。为此，必须知道摄像机的校准参数和相对姿态。同样，一个照相机可以被照亮该特征的激光器替代，以使其对于其余照相机可见。在任何一种情况下，原理都是通过交叉两条可见光线来得到结果。

## 主要的要素

利用传感器数据建立 3D 模型的步骤如下：

1. 从固定位置提取 3D 点云
2. 将多个位置的点云结合起来
3. 将点云转换为三角形网格

第一步，在提取 3D 点的同时将传感器放置在固定的位置和方向，这可以通过多种方式来完成。理论上，6.1 节中的任何深度线索都可应用于相机图像以提取 3D 点。焦点，纹理和阴影的变化通常在计算机视觉中被用作单目线索。如果两台摄像机面对相同的场景并且它们的相对位置和方向已知，则使用双目线索来确定深度。通过识别两幅图像中相同的自然特征，可以通过每个图像相应的射线相交的位置来识别空间中的点，如图 9.27。

如第 9.3 节所述，存在自然和人为特征之间的选择。可以组合使用单台摄像机和红外投影仪或激光扫描仪，通过识别图像中亮点出现的位置来提取深度。这是 Microsoft Kinect 传感器的基础(请回顾第 2.1 节中的图 2.10)。由此产生的 3D 点集合通常称为点云。

在第二步中，问题在于如何合并来自多个位置的扫描数据。如果扫描的时候扫描器之间的相对位置和方向已知，那么问题很好解决。在图 9.26(a)所示的物体扫描仪的情况下，这是通过在转台上旋转物体来实现的，以便位置保持固定，并且每次扫描的方向都是精确已知的。



假设传感器是由安装中机器人上的，例如无人机。机器人通常保持自己的姿态估计，以避免冲突并确定其任务是否达到。这对于确定与执行扫描的时间相对应的姿势也是有用的。通常情况下，姿态估计不够精确，所以要解决估计姿态变化的优化问题，直到重叠扫描之间的数据很好地对齐。在这种情况下，通常使用 *估计最大化(EM)* 算法，该算法在统计学意义上以产生数据的最大似然解释的方式递增调整姿态。如果传感器由人携带，则扫描设备可能包含额外的传感器，如图 9.26(b) 中扫描仪的 GPS 情况；否则，融合来自多个扫描的数据的问题会变得非常困难。

第三阶段，我们已经获得了大量的点云，问题是产生一个纯粹的几何模型。这个过程存在着很多的困难。首先，点密度可能差异很大，尤其是在进行两次或多次重叠扫描的情况下。在这种情况下，有些点可能会被丢弃。另一个问题是可能存在异常值，它们对应着远离其正确位置的孤立点。我们需要一些方法来检测和删除异常值。还有一个问题是数据中可能存在大的空洞或间隙。一旦数据已被充分清理，通常使用表面来拟合数据，从而形成三角网格。每一个问题本身都可以成为一个研究方向。想要更加熟悉整个过程，可以考虑试用开源的 *Point Cloud Library*，它可以处理第二和第三阶段出现的操作。一旦获得三角形网格，如果图像数据也可用，就可以执行纹理贴图操作。VR 面临的最大的挑战之一是，相较于在计算机屏幕上显示，模型的许多缺陷通常在 VR 显示中观察的更为明显。

## 进一步阅读

除了[82]等学术论文外，IMU 校准的一些最有用的报道出现在公司白皮书中，例如[244]。有关磁力计校准，请参见[92,155,165,329]。[166, 164, 165, 168]中介绍了 Oculus Rift 3D 方向追踪。要充分理解基于视觉的追踪方法，请参阅视觉书[111,191,318]。许多研究文献中的 PnP 方法，如[354,367]。对于 VR/AR 的追踪方法的一个很好但较为远古的方法是[344]。被引用最多的作品之一是[141]。关于集成 IMU 和可视化数据进行追踪可以参见[235]。

眼动追踪在[63,338]中进行了调查。人体追踪在[368]中介绍。要充分理解运动学约束和逆向运动问题的解决方案，请参阅[8,10,48]。从机器人的角度来看 SLAM 在[322]中有详细介绍。最近一项基于计算机视觉的 SLAM 调查出现在[86]中。在更大范围内的过滤或传感器融合可以用信息空间来表示(见[163]的第 11 章)。