

## 第 8 章

### 真实世界和虚拟世界中的运动

到目前为止，关于运动的讨论仅限于一些特定的话题。5.3 节讲述了眼球运动，6.2 节涵盖了运动的感受，第 3 章则说明了注视点是如何改变的，但未对运动进行精确的数学描述。本章中我们则想更准确地对运动进行建模，因为真实世界和虚拟世界的物理效果都会影响 VR 体验。运动物体的加速度和速度会影响 VWG 中的模拟以及用于捕捉真实世界中用户运动的跟踪方法。因此，本章提供了运动方面的一些基础知识，更深层次的内容将在第 9 章和第 10 章进行介绍。

8.1 节介绍数学和物理学的基本概念，包括速度，加速度和刚体运动。8.2 节介绍了人类前庭系统的生理学和感知问题，它可以感知速度和加速度。8.3 节描述了如何在 VWG 中描述和制作动作，其中包括数值积分和碰撞检测。8.4 节着重介绍视觉，这是因为视觉和前庭系统之间的感觉冲突是 VR 疾病产生的根源，即在前庭系统未被欺骗时，眼睛可能会感知到运动。这可以被认为是真实和虚拟世界物理现象之间的竞争。

#### 8.1 速度和加速度

##### 8.1.1 一维世界

我们从最简单的情况开始，如图 8.1 所示。设想一个只能在垂直方向上运动的 1D 世界。设  $y$  为移动点的坐标。它在任何时间  $t$  的位置由  $y(t)$  表示，这意味着  $y$  是时间的函数，同时  $y$  又是一个动点，每秒都会产生许多数值。

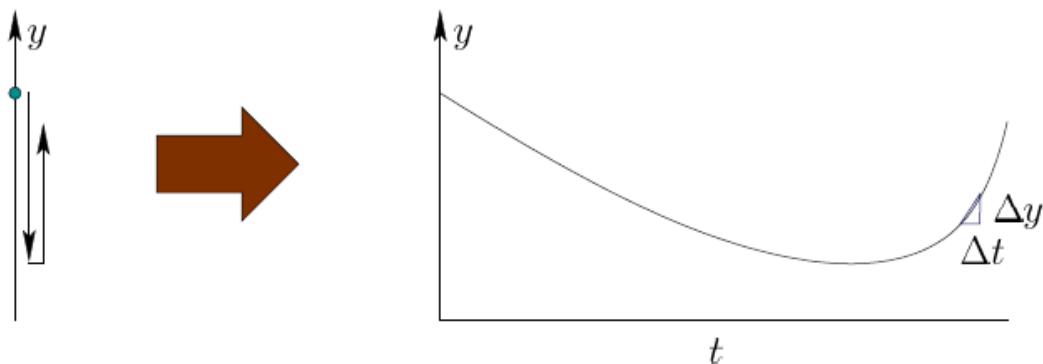


图 8.1：在一维世界中移动的一个点。

#### 速度

那个这个点的移动有多快呢？通过微积分，它的速度  $v$  被定义为  $y$  关于时间的导数：

$$v = \frac{dy(t)}{dt} \quad (8.1)$$

使用数值计算， $v$  近似等于  $\Delta y / \Delta t$ ，其中  $\Delta t$  表示时间上的微小变化：

$$\Delta y = y(t + \Delta t) - y(t) \quad (8.2)$$

换句话说， $\Delta y$  是从时间开始变化到结束时  $y$  的变化。速度  $v$  可以用来估计  $y$  的变化：

$$\Delta y \approx v \Delta t \quad (8.3)$$

这种近似估计会随着  $t$  到  $t + \Delta t$  时间内  $\Delta t$  的减小和  $v$  本身变化的减小而改善。

当然，速度可能随时间的推移而改变。而任意时刻的位置可以通过积分计算得到：

$$y(t) = y(0) + \int_0^t v(s) ds \quad (8.4)$$

其中，假设  $y$  开始移动的时刻为  $t = 0$ 。如果  $v(t)$  对于任意时刻是恒定的，则用  $v$  表示，此时  $y(t) = y(0) + vt$ 。(8.4) 中的积分考虑到了  $v(t)$  变化的情况。

## 加速度

下一步是通过数学方式描述速度的变化，即加速度  $a$ 。它被定义为：

$$a = \frac{dv(t)}{dt} \quad (8.5)$$

该形式与 (8.1) 相同，只是  $y$  被  $v$  替换。可以近似地生成近似值。近似计算的表达式也相似，即  $\Delta v \approx a \Delta t$ 。加速度本身也可以随时间变化，变为时间函数  $a(t)$ 。以下等式则为加速度和速度的积分关系：

$$v(t) = v(0) + \int_0^t a(s) ds. \quad (8.6)$$

由于加速度会变化，因此你可能会怀疑这样的命名过程是否会一直继续下去。然而在大多数情况下，高阶衍生单位是没有必要的。其中一个主要原因是经典物理学的运动通过力和加速度便能充分表示。例如，牛顿第二定律指出  $F = ma$ ，其中  $F$  是作用在一个点上的力， $m$  是物体质量， $a$  是加速度。

举一个简单例子，考虑重力加速度  $g = 9.8 \text{ m/s}^2$ ，某一点相对于地球以加速度  $g$  向下加速。使用 (8.6) 对其进行积分，则速度  $v(t) = v(0) - gt$ 。使用 (8.4) 的速度积分并假设  $v(0) = 0$ ，可以得到：

$$y(t) = y(0) - \frac{1}{2}gt^2 \quad (8.7)$$

### 8.1.2 3D 世界中的运动

#### 点的移动

现在考虑 3D 世界中点的运动。想象一下 3.1 节中定义的几何模型随时间而移动。这会导致模型上每个点  $(x, y, z)$  的移动，从而每个点的坐标会是一个时间函数：

$$(x(t), y(t), z(t)). \quad (8.8)$$

因此 8.1.1 节中的速度  $v$  和加速度  $a$  须展开为三个方向的坐标。速度  $v$  由  $(v_x, v_y, v_z)$  代替以分别指示相对于  $x$ 、 $y$  和  $z$  轴的速度。该情况下， $v$  的大小称为速率：

$$\sqrt{v_x^2 + v_y^2 + v_z^2} \quad (8.9)$$

更进一步，加速度也会扩展为三个分量： $(a_x, a_y, a_z)$ 。

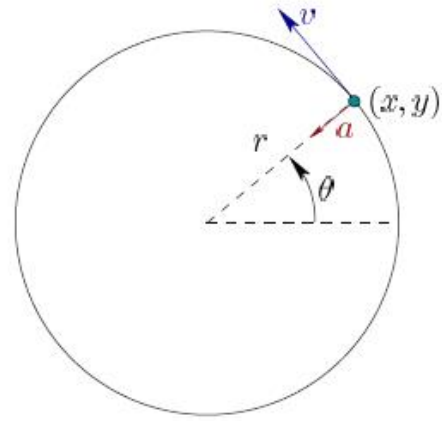
## 刚体运动

现在假设刚体在三维世界中移动。在这种情况下，所有点会一起移动。那么，我们怎样才能简洁地形容这一运动呢？回想一下 3.2 节的平移或旋转。首先，考虑一个简单的例子，假设旋转是禁止的，只允许刚体在空间中平移。在这种设定下，知道刚体上一点的位置便足以确定刚体上所有点随时间的位置。如果有一点已经改变其位置至  $(x_t, y_t, z_t)$ ，则所有点都进行了相同的位移。更重要的是，每个点的速度和加速度也都是相同的。

一旦允许旋转，这种简单的行为不会总是发生。当刚体旋转时，各点不再保持相同的速度和加速度。这对于理解 8.4 节中的 VR 疾病以及追踪方法如何估算位置和方向来说至关重要，具体细节将在第 9 章中讨论。



(a)



(b)

图 8.2: (a) 以恒定角速度  $\omega$  旋转的“木马”。(图片来自西雅图公园和娱乐中心) (b) 从上向下的视图中，旋转木马上某一点的速度矢量  $\mathbf{v}$  与包含它的圆相切；该圆以旋转轴为中心，加速度矢量  $\mathbf{a}$  指向中心。

## 角速度

为了理解这些问题，考虑旋转木马的简单情况，如图 8.2 (a) 所示。它的方向在任何时候都可以用  $\theta(t)$  来描述；见图 8.2 (b)。令  $\omega$  表示其角速度：

$$\omega = \frac{d\theta(t)}{dt} \quad (8.10)$$

默认情况下， $\omega$  的单位是弧度/秒。如果  $\omega = 2\pi$ ，则刚体在一秒钟后返回至相同的位置。假设  $\theta(0) = 0$ ，并且  $\omega$  是常数，则时间  $t$  处的方向由  $\theta = \omega t$  确定。为描述刚体上一个点的运动，使用极坐标  $r$  和  $\theta$  会非常方便：

$$x = r \cos \theta \text{ and } y = r \sin \theta. \quad (8.11)$$

$\theta = \omega t$  代入得到：

$$x = r \cos \omega t \text{ and } y = r \sin \omega t \quad (8.12)$$

再对时间进行求导，得：

$$v_x = -r\omega \sin \omega t \text{ and } v_y = r\omega \cos \omega t \quad (8.13)$$

这里的速度是一个二维矢量，且该矢量与包含点  $(x, y)$  的圆相切，见图 8.2 (b)。

这是非常直观的，因为这一点正是朝着这个方向前进；然而方向也很快就会改变，并始终沿着圆移动。这种速度变化意味着产生了非零加速。点  $(x, y)$  的加速度通过再次求导获得：

$$a_x = -r\omega^2 \cos \omega t \text{ and } a_y = -r\omega^2 \sin \omega t. \quad (8.14)$$

结果是一个指向圆心的二维加速度矢量（图 8.2 (b)），中心即是圆的旋转轴。这就是所谓的向心加速度。如果你站在那一点，你会感觉到一个相反的方向拉力，好像是“木马”试图把你从中心扔掉。这也正是旋转的空间站产生非自然重力的原因。

### 3D 角速度

现在考虑 3D 刚体的旋转。回第 3.3 节，欧拉旋转定理意味着每个 3D 旋转都可以描述为围绕一个轴  $\mathbf{v} = (v_1, v_2, v_3)$  旋转了角度  $\theta$ 。由于刚体的方向在短时间内就发生变化，所以想象一下与旋转变化相对应的轴。在旋转木马的情况下，该轴向量将是  $\mathbf{v} = (0, 1, 0)$ 。对于更一般的情况， $\mathbf{v}$  可以是任何单位向量。

因此 3D 角速度表示为 3D 矢量：

$$(\omega_x, \omega_y, \omega_z). \quad (8.15)$$

这可以从 2D 情况中延伸而来，即取出 2D 情况下的  $\omega$  并将其乘以矢量  $\mathbf{v}$ ，并根据坐标轴位置对各分量进行加权。因此，这些分量可以被认为是  $\omega_x = \omega_{v_1}$ ， $\omega_y = \omega_{v_2}$  和  $\omega_z = \omega_{v_3}$ 。

$\omega_x$ ， $\omega_y$  和  $\omega_z$  分量分别对应于俯仰角，偏航角和翻滚角的旋转速率。由于非交叉性和运动奇点，我们避免了 3.3 节中的这类表示。但事实证明，对于速度而言这些问题并不存在[303]。因此，我们可以在这个阶段暂时不使用四元数。

### 角加速度

如果允许  $\omega$  随时间变化，那么我们必须考虑角加速度。在 2D 情况下，这被定义为

$$\alpha = \frac{d\omega(t)}{dt} \quad (8.16)$$

对于三维情况，有三个分量，其表达为：

$$(\alpha_x, \alpha_y, \alpha_z). \quad (8.17)$$

三个分量可以分别解释为俯仰角，偏航角和翻滚角的加速度。

## 8.2 前庭系统

如 2.3 节所述，平衡感（或前庭感）向大脑提供有关头部如何定向或如何移动的信息。这是通过测量头部线性和角加速度的前庭器官完成的。这些器官连同其相关的神经通路，共同被称为前庭系统。该系统对涉及运动的身体功能起着至关重要的作用，从步行或跑步等普通活动到需要大量训练的活动（如体操或芭蕾舞）。回忆 5.3 节，它还可以通过 VOR 使眼球运动独立于头部运动。

前庭系统对于 VR 来说很重要，因为它常会被忽视，并导致感受的不匹配（请回忆 6.4 节中的这个问题）。当前 VR 系统中，没有任何工程设备可以将前庭信号呈现给显示器，从而精确地刺激前庭器官所应达到的值。电流式前庭刺激或许存在一些可行性，这种刺激为器

官提供电刺激[80,79]。然而，这种技术可能需要更多年的研究才足够准确以及舒适，最终被批准以供大众安全使用。另一种可能性是通过低频振动来刺激前庭系统，但这或多或少会分散人们的注意力。

生理学

图 8.4 显示了人体头部前庭器官的位置。其像眼睛和耳朵那样，有两个对称的器官，分别位于右侧和左侧。图 8.3 显示了每个前庭器官的生理机能。耳蜗处理之后 11.2 节所述的听力部分，其余部分属于前庭系统。椭圆囊和球囊测量线性加速度；它们共同形成耳石系统。当头部未倾斜时，椭圆囊的感测表面大部分位于水平平面（或我们常见坐标系中的  $xz$  平面），而球囊的相应表面位于垂直平面中，被称为矢状面，或  $yz$  平面）。该情况下，椭圆囊感知加速度分量  $a_x$  和  $a_z$ ，球囊感知  $a_y$  和  $a_z$ 。

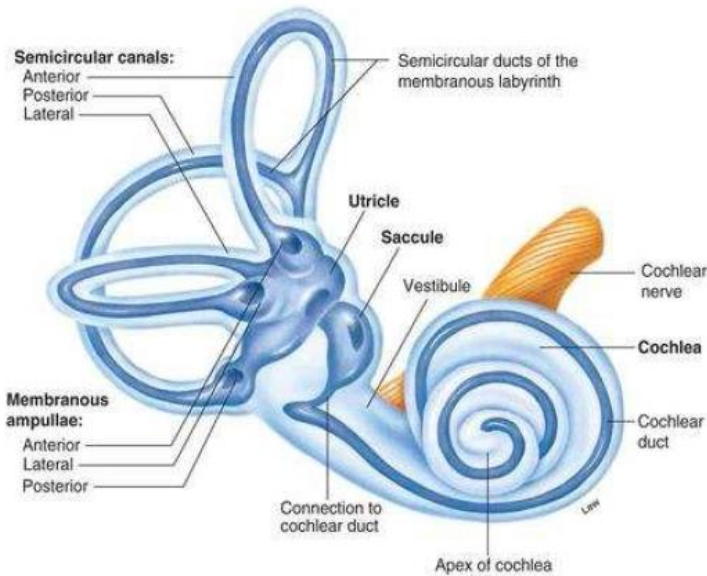


图 8.3：前庭器官

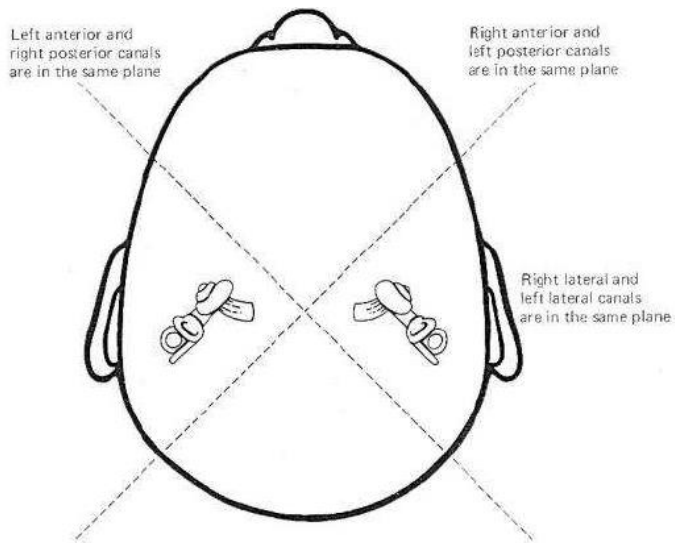


图 8.4：前庭器官位于耳后。（图片来自 CNS Clinic Jordan）



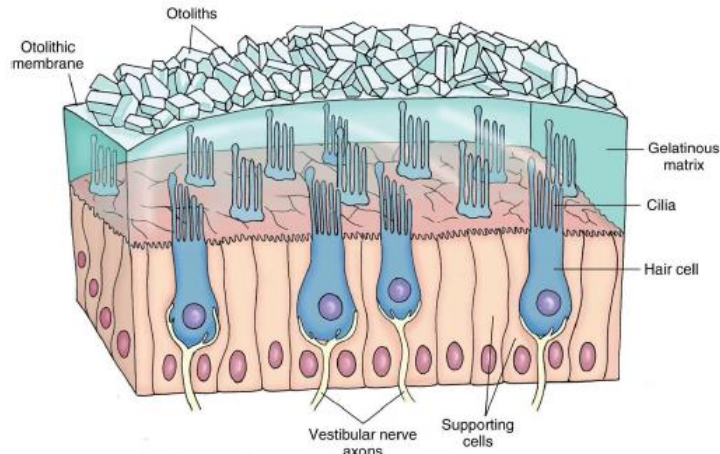


图 8.5: 感知线性加速度的耳石器官（椭圆囊或球囊）

半规管测量角加速度。每个通道具有约 0.2 至 0.3mm 的直径，并且沿着直径约为 2 至 3mm 的圆弧弯曲。令人惊讶的是，这三条半规管基本上两两垂直的，因此它们可以分别测量角速度的三个分量。三个半规管的特定名称分别为前管，后管和侧管。它们与我们通常默认的 3D 坐标轴不完全一致。从图 8.4 中注意到，每组半规管相对于垂直轴旋转了 45 度。因此，左耳的前管与右耳的后管对齐。同样，左耳的后管与右耳的前管对齐。虽然在图中没有显示，但横向的半规管也偏离水平面 30 度左右。即使这样，角加速度的所有三个分量仍可以被感知到，因为这些半规管之间是互相垂直的。

## 线性加速度的感知

为了理解加速度如何被感知，我们从耳石系统的构造开始理解。图 8.5 为耳石器官的示意图，该器官可以是椭圆囊或球囊。机械性感受器以毛细胞的形式将加速度转化为神经信号。每个毛细胞都含有嵌入凝胶状基质的纤毛。重量集中于纤毛顶部，因此当产生横向加速度时，移位的重量会产生剪切力，导致纤毛弯曲。加速度越高，弯曲越大，同时神经脉冲的传输速率越高。横向偏转在两个维度均是有可能发生的。例如在椭圆囊的情况下， $xz$  平面内任何方向的线性加速度都会导致纤毛弯曲。为了区分该平面内的特定方向，纤毛被极化，使得每个细胞对一个特定方向敏感。这是通过较厚的动纤毛来实现的，而细胞的所有其他纤毛都通过一种带状物穿过尖端实现连接，从而它们可以全部弯曲在一起。

一种主要的感知限制是由于物理学的一个基本规律而来：爱因斯坦等价原理。除了前庭系统外，它还会影响 VR 追踪系统（见第 9.2 节）。该问题的关键在于重力。如果我们在太空深处，远离任何重力，那么由传感器测量的线性加速度将是相对于固定坐标系的纯加速度。在地球上，我们则受到重力，就好像我们在一艘火箭上以大约 9.8 米/秒的速度向上加速。等价原理指出，重力和真正的线性加速度对身体的影响是难以区分的。图 8.6 显示了耳石器官的感知结果，无论头部是倾斜还是线性加速，都会以相同的信号发送到大脑。如果闭上眼睛或戴上 VR 耳机，则无法区分倾斜和加速。在大多数情况下，我们并不会被这种现象所欺骗，因为前庭信号在加速时会伴随着其他刺激，如视觉和转动信号。

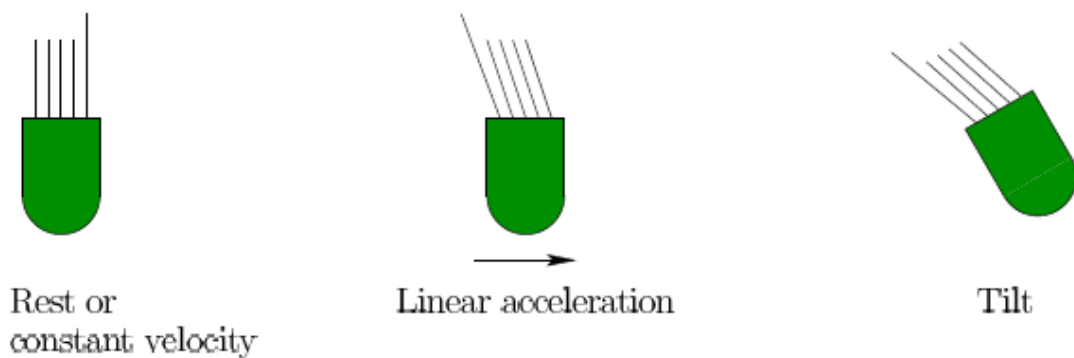


图 8.6: 由于爱因斯坦等价原理，耳石器官无法区分头部相对于重力的线性加速度与倾斜加速度。无论哪种情况，纤毛都以相同的方式偏转，向神经结构发送相同的信号。

### 感应角加速度

半规管的感知原理与耳石器官相同。它们通过在毛细胞末端弯曲纤毛来测量加速度。粘性流体在每个通道内移动。一个叫做壶腹帽的弹性结构会阻塞半规管的一小部分，并包含毛细胞；见图 8.7。比较半规管与木马的旋转。如果我们在旋转木马的周围放置充满液体的管子，那么流体将以恒定的角速度保持相当稳定的状态。然而，如果施加角加速度，由于流体与管道之间的摩擦（以及内部流体粘度），流体将开始在管内进行相对移动。在半规管中，移动的流体向壶腹帽施加压力，使其变形并弯曲其内部毛细胞上的纤毛。请注意，恒定的角速度原则上不会对圆筒造成压力；因此，半规管测量的是角加速度而不是角速度。每个半规管极化的，它主要响应围绕垂直于包含整个半规管平面的轴的旋转。

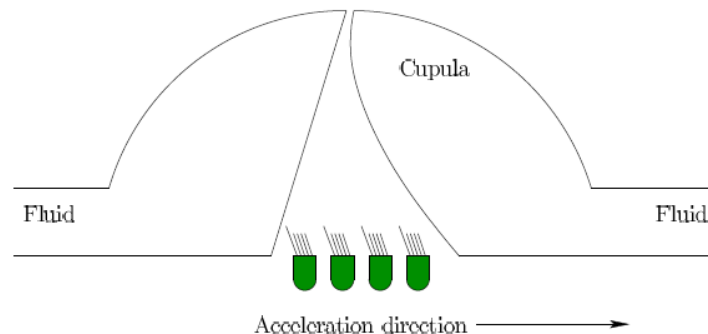


图 8.7: 壶腹帽包含一个容纳纤毛的中心膜。如果角加速度出现与导管方向一致，则压力施加到杯状物上，这导致纤毛弯曲并发送神经信号。

### 对感知的影响

来自前庭系统的感知响应与其他感觉相比通常较弱，尤其是视觉。同时，前庭系统在提供与其他系统（包括视觉和本体感觉）一致的线索时会产生良好的反应。不匹配的感知则会产生问题。例如，当前庭系统运行不正常时，有些人可能会感到眩晕。在这种情况下，他们会觉得周围的世界正在旋转或摇摆。常见的症状是恶心，呕吐，出汗和行走困难。由于 VOR，这甚至可能会影响眼球运动。后续的 8.4 节则会解释 VR 中前庭和视觉线索不匹配时所造成的不良反应。

## 8.3 虚拟世界中的物理

### 8.3.1 根据经验调整物理

如果我们想让我们的大脑相信我们生活在虚拟世界中，那么我们对现实世界的许多期望应该在虚拟世界中相匹配。我们已经看到了光的物理学（第 4 章）适用于虚拟世界的视觉渲染（第 7 章）。虚拟世界中的动作也应该以熟悉的方式表现。

这意味着 VWG 包含一个物理引擎，通过遵循物理世界的原则来管理虚拟世界中的物体的运动。作用于身体的力量，重力，流体流动和身体之间的碰撞应该以感性令人信服的方式处理。物理学引擎在整个工程和物理学的任何模拟环境中都会出现。在视频游戏，计算机图形和电影中，这些引擎执行的操作非常接近我们对 VR 的需求，这就是为什么 Unity 3D 和虚幻引擎等流行游戏引擎已经很快适应 VR 使用的原因。如第 2.2 节所述，我们还没有到达一个广泛采用 VR 引擎的时代。因此，现代游戏引擎目前还需要好好理解和运用。

为了确定什么样的物理引擎需要借鉴，改编或者从头构建，应该考虑期望的 VR 体验并确定将出现的运动类型。一些常见的一般问题是：

- 匹配区域是否保持固定，还是需要通过移动来移动用户？如果需要运动，那么用户是否会步行，跑步，游泳，驾驶汽车或飞行太空船？
- 用户是否会与对象进行交互？如果是这样，那么需要什么样的交互？可能性包括携带武器，打开门，投掷物体，倒饮料，操作机械，绘画和组装结构。
- 多个用户将共享相同的虚拟空间吗？如果是这样，那么他们的议案将如何协调或约束呢？
- 虚拟世界是否包含自行移动的实体，例如机器人，动物或人类？
- 用户是否会沉浸在熟悉或陌生的环境中？熟悉的环境可能是家庭，教室，公园或城市街道。一个充满异国情调的环境可能是水肺潜水，月球探索或穿过血管。

除了物理引擎之外，这些问题也将指导接口的设计，这将在第 10 章中介绍。

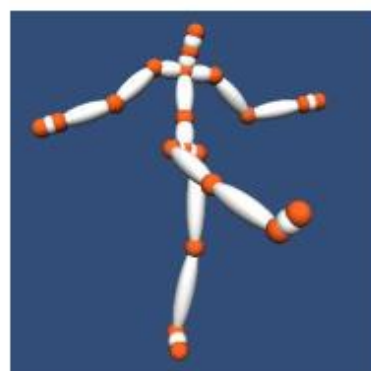
基于对上述问题的回答，物理引擎设计可能简单而高效，或者完全压倒性的。正如第 7.4 节所述，一个关键的挑战是保持经常更新的虚拟世界，以使用户和对象之间的交互很好地同步，并且为显示器提供低延迟投影。

请注意，目标可能并不总是完美匹配在现实世界中会发生的事情。在熟悉的环境中，我们可能会期望重要的匹配；然而，在陌生的环境中，做出舒适体验往往变得更加重要，而不是完美匹配现实。即使在模拟自己在世界上散步的情况下，我们也经常希望偏离现实世界的物理学，因为它会导致 VR 晕眩（见 8.4 节）。

本节的其余部分涵盖了一些常见的基本方面：1）物理系统的数值模拟，2）使用人类输入的系统的控制，3）碰撞检测，它确定身体是否相互干扰。



(a)



(b)



图 8.8: (a) 在侠盗猎车手游戏中出现的虚拟车 (猎豹); 它应该拥有多少自由度? (b) 通过关节连接刚体的人体骨骼通常是化身运动的基础。(图来自于 SoftKinetic)

### 8.3.2 数值模拟

#### 虚拟世界的状态

想象一个包含许多移动刚体的虚拟世界。对于每个物体, 请考虑其自由度 (DOF), 这对应于唯一确定其位置和方向所需的独立参数数量。我们想知道在一次性瞬间将每个身体置于适当位置所需的所有参数列表。所有这些参数的值的规范被定义为虚拟世界的状态。然后可以将物理引擎的工作描述为计算渲染系统所需要的每个时刻或虚拟世界的“快照”的虚拟世界状态。一旦确定了状态, 第 3 章的数学变换就被用来将身体正确地放置在世界中并计算它们应该如何在显示器上出现。

#### 自由度

虚拟世界模型中有多少个参数? 如第 3.2 节所述, 不受约束的物体具有 6 个自由度, 这意味着 6 个参数可将其放在任何位置。在很多情况下, 自由度因约束而丢失。例如, 在地面上滚动的球只有 5 个自由度, 因为它可以实现沿地面的任何 2D 位置, 并且也具有任何 3D 方向。通过指定沿着地面的位置 (两个参数) 和它面对的方向 (一个参数) 来描述具有 3 个自由度的汽车可能是足够的; 见图 8.8 (a)。但是, 如果赛车在空中飞行时可以在空中飞行或撞击, 那么所有 6 个自由度都是需要的。

对于许多型号, 刚体以允许相对运动的方式连接在一起, 这被称为多体运动学[163,303]。例如, 一辆汽车通常有 4 个车轮, 每个车轮可以滚动以提供一个旋转的自由度。此外, 前轮可以转向提供额外的自由度。转向通常会使得前轮一致, 这意味着一个自由度足以描述两个车轮。如果汽车有一个复杂的悬挂系统, 那么它不能被视为一个单一的刚体, 这将增加更多的自由度。

同样, 动画角色可以通过附加刚体形成骨架来制作; 见图 8.8 (b)。骨架中的每个刚体通过关节连接到一个或多个其他物体。例如, 可以通过将胳膊, 腿和脖子连接到刚性躯干来形成简单的人物角色。左上臂通过肩关节连接到躯干, 手臂的下部通过肘关节连接, 依此类推。有些关节允许比其他关节更多的自由度。例如, 肩关节有 3 个自由度, 因为它可以相对于躯干偏航, 俯仰和滚动, 但肘关节只有一个自由度。

为了充分模拟人体的灵活性, 需要 244 个自由度, 由 630 块肌肉控制[366]。在很多情况下, 这有着太多的细节, 可能导致高计算复杂性而难以实现。此外, 人们应该时刻提防意外出现的未知因素 (在 1.1 节中提到), 其中更多的现实主义可能会导致人物感知的蠕变增加。因此, 拥有更多自由度并不是明显更好, 而是由 VR 内容创建者决定需要多少机动性才能使角色变为现实, 并且以一种有针对性的目的引人注目。

在极端情况下, 刚体不足以塑造世界。我们可能希望看到波浪在湖面上真实地荡漾起来, 或者在微风中微微流淌的头发。在这些通用设置中, 使用非刚性模型, 在这种情况下, 可以将状态视为连续函数。例如, 形式为  $y=f(x, z)$  的函数可以描述水的表面。没有进行一些限制简化, 结果可能会有无限多个自由度。这种设置中的运动通常使用偏微分方程 (PDE) 来描述, 其在数值上被整合以获得期望时间的状态。通常, 计算成本足够高, 以至于禁止将它们用于交互式 VR 体验, 除非通过预计算运动或快速简化模型来实现快捷方式。

#### 微分方程

我们现在介绍一些基本的微分方程来模拟运动。结果描述通常被称为动态系统。第一步

是根据状态描述刚体速度。回到涉及一个或多个刚体的模型，该状态对应于有限数量的参数。让

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \quad (8.18)$$

该式表示  $n$  维状态向量。如果每个  $x_i$  对应于刚体的位置或方向参数，则状态向量将所有物体置于其位置。让

$$\dot{x}_i = \frac{dx_i}{dt} \quad (8.19)$$

表示每个参数的时间导数或速度。

为了在任何时间  $t$  获得状态，速度需要随时间积分。在 (8.4) 之后，每个状态变量的积分决定了时间  $t$  的值：

$$x_i(t) = x_i(0) + \int_0^t \dot{x}_i(s) ds, \quad (8.20)$$

其中  $x_i(0)$  是时间  $t=0$  时  $x_i$  的值。

式 (8.20) 存在两个主要问题：

1. 积分几乎总是必须用数字来评估。
2. 速度  $\dot{x}_i(t)$  必须在每个时间  $t$  被指定。

采样率对于第一个问题，时间被离散化为步骤，其中  $\Delta t$  是步长或采样率。例如， $\Delta t$  可能是 1ms，在这种情况下，可以计算时间  $t=0, 0.001, 0.002, \dots$  的状态，以秒为单位。这可以被认为物理引擎的一种帧频。每个  $\Delta t$  对应于新帧的产生。

如第 7.4 节所述，VWG 应将虚拟世界框架的生成与渲染过程同步，以便世界不会陷入中间状态，其中一些变量更新为新时间，而其他变量则停留在前一时间。这是虚拟世界中的一种撕裂。然而，这并不意味着渲染器和物理引擎之间的帧速率是相同的。通常情况下，物理引擎的帧速率要高得多以提高数值精度。使用采样率  $\Delta t$ ，(8.20) 近似为

$$x_i((k+1)\Delta t) \approx x_i(k\Delta t) + \sum_{j=1}^k \dot{x}_i(j\Delta t)\Delta t, \quad (8.21)$$

对于每个状态变量  $x_i$ ，(8.21) 一次一个步骤更简单。令  $x_i[k]$  表示  $x_i(k\Delta t)$ ，它是时间  $t=k\Delta t$  时的状态。以下是根据旧状态  $x_i[k]$  表示新状态  $x_i[k+1]$  的更新定律：

$$x_i[k+1] \approx x_i[k] + \dot{x}_i(k\Delta t)\Delta t, \quad (8.22)$$

从  $x_i[0] = x_i(0)$  开始。

## 龙格-库塔数值积分法

(8.21) 中使用的近似称为欧拉积分。这是最简单的近似，但在许多实际环境中表现不佳。最常见的改进之一是第四代 Runge-Kutta 积分方法，它将新状态表示为

$$x_i[k+1] \approx x_i[k] + \frac{\Delta t}{6}(w_1 + 2w_2 + 2w_3 + w_4), \quad (8.23)$$

其中

$$\begin{aligned}
w_1 &= f(\dot{x}_i(k\Delta t)) \\
w_2 &= f(\dot{x}_i(k\Delta t + \frac{1}{2}\Delta t) + \frac{1}{2}\Delta t w_1) \\
w_3 &= f(\dot{x}_i(k\Delta t + \frac{1}{2}\Delta t) + \frac{1}{2}\Delta t w_2) \\
w_4 &= f(\dot{x}_i(k\Delta t + \Delta t) + \Delta t w_3).
\end{aligned} \tag{8.24}$$

虽然这比欧拉积分代价更高,但实际上提高的准确度通常是值得的。存在许多其他方法,其性能取决于  $\dot{x}$  表达的特定方式并随时间而变化[133]。

## 时不变的动态系统

(8.20) 的第二个问题是确定  $\dot{x}(t)$  的表达式。这就是物理定律,例如由于施加的力和重力引起的刚体加速。最常见的情况是时不变的动态系统,其中  $\dot{x}$  仅取决于当前状态而不取决于特定时间。这意味着每个组件  $x_i$  被表示为

$$\dot{x}_i = f_i(x_1, x_2, \dots, x_n), \tag{8.25}$$

对于某个给定的向量值函数  $f = (f_1, \dots, f_n)$ 。这可以通过使用  $x$  和  $\dot{x}$  来表示  $n$  维向量以压缩形式来表示:

$$\dot{x} = f(x). \tag{8.26}$$

上面的表达式通常称为状态转换方程,因为它表示状态的变化率。

这是一个简单的一维状态转换方程的例子:

$$\dot{x} = 2x - 1. \tag{8.27}$$

这被称为线性微分方程。速度  $x$  大致与  $x$  的值相加。幸运的是,线性问题可以在“纸上”完全解决。(8.27) 的解决方案是一般形式

$$x(t) = \frac{1}{2} + ce^{2t}, \tag{8.28}$$

其中  $c$  是一个常数,取决于  $x(0)$  的给定值。

## 相空间

不幸的是,运动通常用加速度(有时候是高阶导数)来描述,需要两次积分。这会产生高阶微分方程,难以处理。出于这个原因,相空间表示是在物理学和工程学中发展起来的。在这种情况下,状态变量的速度本身被视为状态变量。这样,加速度就成为速度变量的速度。

例如,假设位置  $x_1$  受重力作用,产生加速度  $a = -9.8\text{m/s}^2$ 。第二个变量  $x_2$ , 它被定义为  $x_1$  的速度。因此,根据定义,  $\dot{x}_1 = x_2$ 。此外,  $\dot{x}_2 = a$ , 因为速度的导数是加速度。这两个方程都符合 (8.25) 的形式。通常情况下,状态的数量会随着加速度的增加而增加(或者甚至是更高阶的导数),但是结果的动态性以 (8.25) 的形式表示,这更容易处理。

## 处理用户输入

现在考虑用户命令对象移动的情况。例子包括驾驶汽车,飞行太空船,或者漫步一个化身。这引入了一些新的参数,称为控制,动作或动态系统的输入。包含这些新参数的微分方程称为控制系统[14]。

令  $u = (u_1, u_2, \dots, u_m)$  为控件的向量。(8.26) 中的状态转移方程简单地扩展为包含  $u$

的方程：

$$\dot{x} = f(x, u). \quad (8.29)$$

图 8.9 显示了一个有用的例子，它举了驾驶汽车的例子，来说明如何决定汽车的速度。例如， $u_s = 1$  正向驱动， $u_s = -1$  反向驱动。为了得到更快的速度，一共设置有 10 个驱动器。控制器确定前轮如何转向。状态向量是  $(x, z, \theta)$ ，它对应于汽车在水平  $xz$  平面中的位置和方向。

状态转换方程是：

$$\begin{aligned} \dot{x} &= u_s \cos \theta \\ \dot{z} &= u_s \sin \theta \\ \dot{\theta} &= \frac{u_s}{L} \tan u_\phi. \end{aligned} \quad (8.30)$$

使用 Runge-Kutta 积分或类似的数值方法，考虑到随着时间的推移控制  $U_s$  和  $u_\phi$ ，可以计算汽车的未来状态。

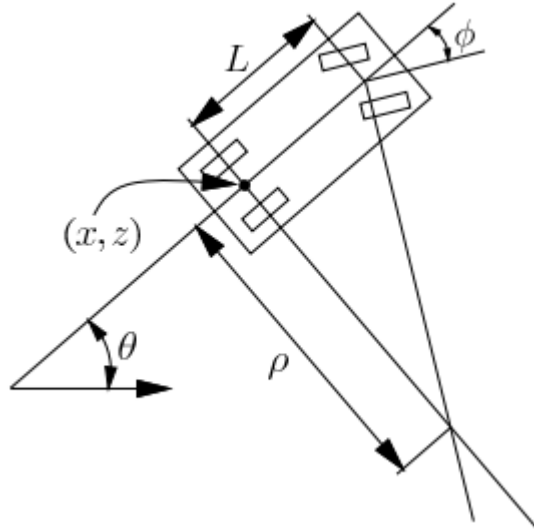


图 8.9：简单的可操纵汽车的俯视图。它的位置和方向由  $(x, y, \theta)$  给出。参数  $\rho$  是最小转弯半径，其取决于最大允许转向角  $\phi$ 。通过将视角放置在后轴中心的上方，该模型也可用于“引导”人物化身。

该模型也可用于从第一人称视角引导 VR 用户的虚拟步行。然后视点根据  $(x, z, \theta)$  改变，而高度  $y$  保持固定。对于 (8.30) 中的模型，汽车必须前进或后退才能改变其方向。通过将第三个分量更改为  $\theta = u\omega$ ，用户可以直接指定角速度。这会导致用户旋转到位，就像旋转木马一样。这些更多的例子出现在 [163] 的第 13 章中，包括通过加速控制的物体。

根据离散时间点来定义运动，这在概念上有时很有用，称为阶段。使用 (8.29) 的数值积分，我们可以考虑在时间  $\Delta t$  上应用控制  $u$  以获得新的状态  $x[k+1]$ ：

$$x[k+1] = F(x[k], u[k]). \quad (8.31)$$

函数  $F$  通过在  $\Delta t$  上积分 (8.29) 获得。因此，如果状态是  $x[k]$ ，并且  $u[k]$  被应用，则  $F$  计算  $x[k+1]$  作为下一阶段的状态。

### 8.3.3 碰撞检测

构建物理引擎的最大挑战之一是处理机体之间的碰撞。来自物理学或工程学的标准运动定律通常不考虑这种相互作用。因此，专门的算法用于检测何时发生这种碰撞并作出适当的响应。由于计算机图形仿真和视频游戏的广泛需求以及机器人的运动规划，碰撞检测方法和相应的软件非常丰富。

### 实体或边界模型？

图 8.10 显示了一个 2D 世界中两个三角形碰撞检测的第一个困难。前两种情况（图 8.10 (a) 和 8.10 (b)）显示明显的情况；然而，第三种情况，图 8.10 (c) 可能不明确。如果一个三角形完全位于另一个三角形内部，那么这是一次碰撞吗？如果我们将外三角解释为实体模型，那么是的。如果外三角形只是边界边缘，并且意味着内部是空的，那么答案是否定的。这就是为什么强调 3.1 节中有一个连贯的模型；否则，边界可能建立得不够好，无法区分内部和外部。

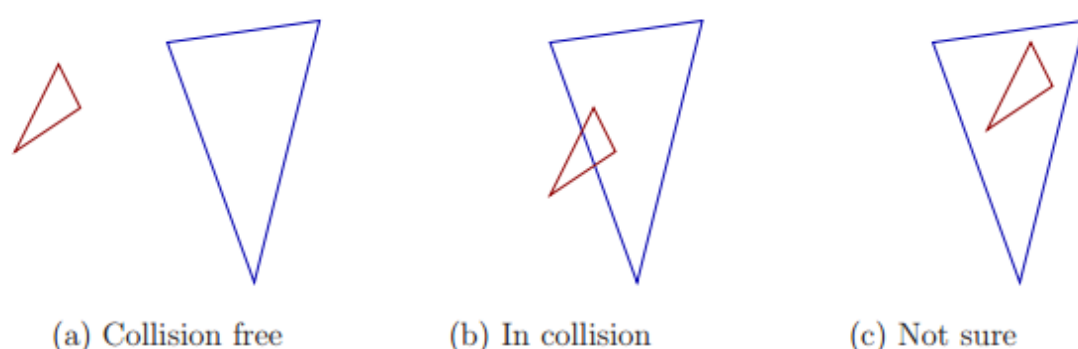


图 8.10: 碰撞检测的三个有趣案例（这些是 2D 示例）。最后一种情况可能会或不会导致碰撞，具体取决于型号。

### 距离函数

许多碰撞检测方法受益于维持一个距离函数，该距离函数可以跟踪物体距离碰撞的距离。例如，让  $A$  和  $B$  表示由两个不同模型在  $R^3$  中占用的所有点的集合。如果它们处于碰撞中，那么它们的交集  $A \cap B$  不是空的。如果它们没有发生碰撞，那么  $A$  和  $B$  之间的 Hausdorff 距离是最接近的一对点之间的欧几里德距离，从  $A$  和  $B$  中取一个。让  $d(A, B)$  表示这个距离。如果  $A$  和  $B$  相交，那么  $d(A, B) = 0$ ，因为  $A \cap B$  中的任何点将产生零距离。如果  $A$  和  $B$  不相交，那么  $d(A, B) > 0$ ，这意味着它们没有发生碰撞（换句话说，没有碰撞）。如果  $d(A, B)$  很大，那么即使其中一个或两个正在移动， $A$  和  $B$  在不久的将来也很可能无碰撞。这导致了一系列称为增量距离计算的碰撞检测方法，该方法假定在对算法的连续调用之间，身体只移动一小部分。在这个假设下，算法对于凸多面体[181,214]实现“几乎恒定的时间”性能。非凸体可以分解为凸面组件。

与距离有关的概念是穿透深度，这表明一个模型在另一个模型中的位置有多远[182]。这对设置允许两个物体之间有多少干扰的阈值很有用。例如，用户可能能够将他的头部戳到墙壁两厘米处，但再靠近点的话就应该采取避让措施。

### 简单的碰撞测试

在最低级别，碰撞检测通常需要测试一对模型基元来确定它们是否相交。在 3D 模型形成模型的情况下，我们需要一种方法来确定两个三角形是否相交。这与第 7.1 节中视觉渲染所需的光线三角形相交测试类似，并且涉及解析几何中的基本工具，例如交叉积和平面方程。有效的方法在[106,216]中给出。



## 广义狭义两阶段

假设一个虚拟世界已经被定义为数百万个三角形。如果两个复杂的非凸面体被检查碰撞，那么计算成本可能很高。对于这种情况，碰撞检测往往成为一个两阶段过程：

1. 广义阶段：在广义阶段中，任务是避免为彼此远离的物体执行昂贵的计算。可以在每个物体周围放置简单的边界框，并且可以执行简单的测试以避免昂贵的碰撞检查，除非这些边框相交。在某些情况下，散列方案可以用来大大减少必须测试相交的盒子对的数量[215]。

2. 狭义阶段：在狭义阶段，单个模型部件对各自仔细检查碰撞。这涉及代价较高的测试，例如三角形 - 三角形交叉。

在广义阶段中，分层方法通常将每个身体分解成一棵树。树中的每个顶点都代表一个包含身体某个子集的边界区域。根顶点的边界区域包含整个主体。指导选择边界区域类型的两个相反的标准：

1. 该区域应尽可能紧密地符合预期的模型要点。
2. 两个区域的相交测试应尽可能高效。

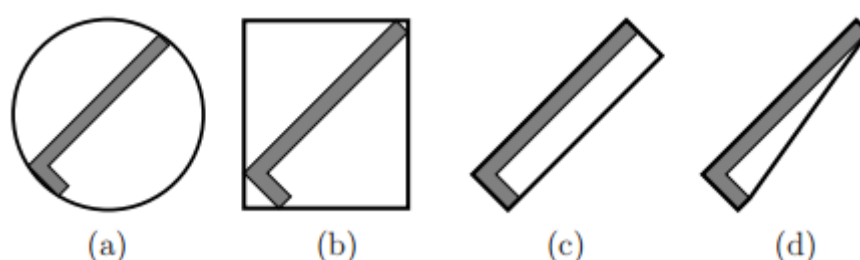


图 8.11：四种不同的边界区域：(a) 球体，(b) 轴对齐边界框（AABB），(c) 定向边界框（OBB）和 (d) 凸包。 每种方法通常比前一种方法提供更严格的近似值，但测试与其他方法的交叉代价更高。

图 8.11 显示了几种流行的选择，例如 L 形机身。如 7.1 节所述，分层方法对于快速消除视觉渲染中考虑的许多三角形也很有用。

该树是为一个物体构建的，A（或者，B）递归地如下构建。对于每个顶点，考虑包含在边界区域中的 A 中所有点的集合 X。通过定义两个较小的边界区域来构造两个子顶点，其结合覆盖 X。分割是为了使每个孩子覆盖的部分具有相似的尺寸。如果几何模型由诸如三角形的图元组成，则可以进行分割以将三角形分成两组大致相同数量的三角形。然后为每个孩子计算边界区域。图 8.12 显示了一个 L 形主体情况下的分割示例。通过分割直到获得非常简单的组，递归地生成孩子。例如，对于空间中的三角形，除非顶点表示单个三角形，否则将进行拆分。在这种情况下，很容易测试两个三角形的交点。

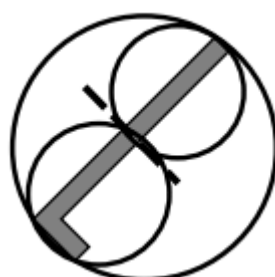


图 8.12：大圆圈显示了覆盖 L 形体的顶点的边界区域。在沿着虚线进行分割之后，使用两个较小的圆来覆盖身体的两个半部分。每个圆圈对应一个子顶点。

考虑确定身体 A 和 B 是否碰撞的问题。假设  $T_a$  和  $T_b$  树分别为 A 和 B 构建。如果  $T_a$  和  $T_b$  的根顶点的边界区域不相交，则知道  $T_a$  和  $T_b$  由于没有进行任何附加计算，则不会相互碰

撞。如果边界区域确实相交，则将  $T_a$  的子区域的边界区域与  $T_b$  的边界区域进行比较。如果其中任何一个相交，则  $T_b$  的边界区域被其子区域的边界区域替换，并且该过程继续递归。只要边界区域相交，遍历树的较低层次，直到最终达到叶子。算法在叶子上测试单个三角形进行碰撞，而不是边界区域。请注意，随着树被遍历，如果  $T_a$  的顶点  $v_1$  的边界区域不与  $T_b$  的顶点  $v_2$  的边界区域相交，则  $v_1$  的子节点不必与  $v_2$  的子节点进行比较。通常，相对于测试所有交叉的三角形的方法，这大大减少了比较次数。

## 虚拟现实中的不匹配的障碍

虽然碰撞检测是一个标准的，好解决的问题，但虚拟现实再一次带来了非同寻常的挑战。主要困难之一是匹配区域，其中真实世界和虚拟世界共享相同的空间。这造成了三个有趣的情况：

**真实障碍：**在这种情况下，真实世界存在障碍，但不存在于虚拟世界中。这是潜在的危险！例如，你可以移动你的手臂，碰一下真实的热咖啡，这是虚拟世界中没有的。如果你正在用 VR 耳机走路，那么想象一下如果一组真正的下楼梯没有出现，会发生什么。至少，如果用户接近匹配区域的边界，则应呈现边界。这种不匹配促使 HTC Vive 耳机中引入了 Chaperone 系统，使用朝外的相机来检测和渲染可能阻碍用户运动的真实物体。

**仅虚拟障碍：**这种情况并不危险，但可能非常影响体验。用户可以在虚拟现实环境中穿过墙壁，而不会在现实世界中感受到任何反应。这在大多数情况下不应该被允许。随着虚拟墙被联系，VWG 可以简单地停止移动虚拟世界中的观点；然而，这会在真实和虚拟运动之间产生不匹配，这对用户来说可能会感到不舒服。在试图引导他们摆脱对虚拟障碍的干扰时，保持用户的舒适性仍然是一项艰巨的挑战。

**真实和虚拟障碍：**如果障碍物在真实世界和虚拟世界中都相匹配，那么这种效果在感觉上是强大的。例如，您可能站在现实世界中略微升高的平台上，而虚拟世界显示您站在建筑物屋顶上。如果屋顶和平台边缘完美对齐，那么您可以用脚感受边缘。你会害怕走出边缘吗？更简单的情况是呈现与用户可能坐在的真正椅子相匹配的虚拟椅子。

## 8.4 不匹配的运动和观察

第 2.3 节提到的运动是一种由不同的视觉刺激引起的自我运动幻觉。换句话说，即使实际上没有发生运动，大脑也被欺骗地认为头部正在移动。图 2.20 显示了闹鬼的摇摆幻觉，它使上下颠倒的人们信服。然而，人们静止的时候房子在移动。在真实世界中没有相应运动的情况下，通过移动用户的视点，通常也会在 VR 中感知到运动。



图 8.13：由于世界上的运动而引起的图像中的特征光流。 这些是使用图像处理算法自动计算的。（图来自于图宾根的马克斯普朗克智能系统研究所的 Andreas Geiger）

虚拟现实带来的移动的认知因素是不匹配线索的一个主要例子，这在第 6.4 节中讨论过。尽管 McGurk 效应没有任何有害的副作用，但不幸的是，这个因素会让很多人都会遇到晕眩，恶心，偶尔甚至呕吐等疾病症状。因此，它应该非常谨慎地应用于 VR 体验。此外，如果使用它，应该尝试改变内容，以使副作用最小化。业界巨头经常宣称他们最新的 VR 设备已经击败了 VR 疾病问题；然而，这忽略了以下违反直觉的行为：

如果设备在空间分辨率，帧速率，跟踪精度，视场和延迟方面更好，那么由虚拟现实带来的移动认知因素和其他不匹配线索让人们感到不舒服的潜在性更高。

简单而直观地说，如果头盔能够更准确地模拟真实情况，那么感官线索就会更强大，而我们的感知系统会对错误的线索更加自信。它甚至可以模仿较差的设备，从而可以比较评估早期 VR 系统的副作用。在某些情况下，线索的不匹配可能是无害的（尽管可能导致降低存在感）。在其他情况下，由于大脑难以解决小的冲突，错配可能导致更大的疲劳。在最糟糕的情况下，VR 疾病会出现，因为虚拟现实带来的移动的认知因素是当今 VR 体验的最大罪魁祸首。最糟糕的情况之一是第一人称射击游戏直接适用于虚拟现实，在虚拟形象探索敌对环境时，几乎一直会有虚拟现实中的移动。

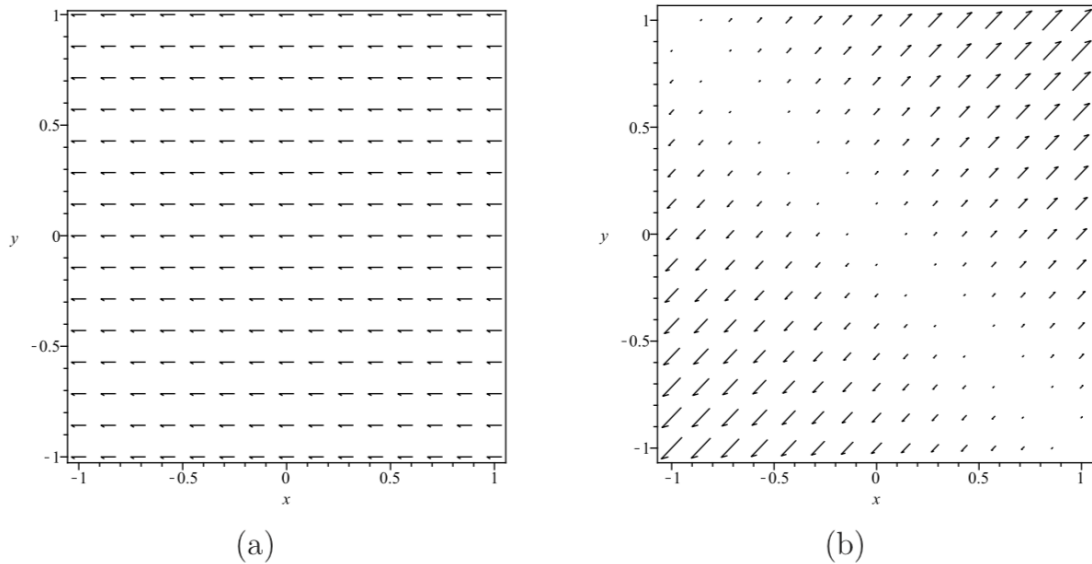


图 8.14: 矢量场示例: (a) 一个常数矢量场，对于每个矢量  $(-1, 0)$ ，无论位置在哪里。 (b) 在这个矢量场中， $(x, y) \rightarrow (x+y, x+y)$ ，矢量指向远离  $(-1, 1)$  到  $(1, -1)$  的对角线，长度与距离成正比。

## 光流

从第 6.2 节回忆，人类视觉系统具有专门用于检测视场中视觉特征运动的神经结构；见图 8.13。它实际上是穿过视网膜的这些特征的图像。因此，有一个描述表面上移动点速度的数学概念是有用的。因此，我们定义了一个矢量场，它在表面上的每个点处分配一个速度矢量。如果曲面是  $xy$  平面，则是速度矢量

$$(v_x, v_y) = \left( \frac{dx}{dt}, \frac{dy}{dt} \right)$$

在每个点  $(x, y)$  被分配。例如，

$$(x, y) \mapsto (-1, 0)$$

是一个常量向量场，它每个地方都分配  $v_x = -1$  和  $v_y = 0$ ；见图 8.14 (a)。 向量场

$$(x, y) \mapsto (x + y, x + y)$$

不是常数，并且在每个点  $(x, y)$  处分配  $v_x = v_y = x + y$ ；见图 8.14 (b)。 对于这个向量场，速度方向总是对角线的，但矢量的长度（速度）取决于  $x + y$ 。

为了最准确地描述沿着视网膜的特征的运动，应该在与光感受器的位置相对应的球形表面上定义向量场。相反，我们将在一个正方形区域描述向量场，并且理解它应该被转换到一个球体上以获得更高的精度。

## 虚拟现实中的移动的类型

虚拟现实中的移动的认知因素可以由虚拟世界中视角的角速度和线速度的任意组合引起。为了有效地表征不同类型运动的影响，将视点速度分解为三个线性分量  $v_x$ ， $v_y$  和  $v_z$  以及三个角分量  $\omega_x$ ， $\omega_y$  和  $\omega_z$  是很方便的。 因此，我们考虑这六种情况下的光流量（见图 8.15）：

**yaw 向移动：**如果视点绕  $y$  轴逆时针旋转（正  $\omega_y$ ），则所有的视觉特征以相同的速度从右向左移动，如图 8.15 (a) 所示。 等同地，虚拟世界围绕用户顺时针旋转；然而，相反方向的自我运动被感知。 这会让用户感觉自己仿佛置身于旋转木马（回忆图 8.2）。

**pitch 向移动：**通过绕  $x$  轴（正的  $\omega_x$ ）逆时针旋转视点，所有特征以相同的速度向下移动，如图 8.15 (b) 所示。

**roll 向移动：**逆时针旋转视角约  $z$ （光轴（正向  $\omega_z$ ）），使特征绕图像中心顺时针旋转，如图 8.15 (c) 所示。 每个特征的速度与包含它的圆相切，速度与从特征到图像中心的距离成正比。

**横向移动：**在这种情况下，视点被转换到右侧，对应于正向  $v_x$ 。 结果，这些特征水平移动；然而，在偏航方面有一个重要的区别：与较近物体相对应的特征比远距离物体更快。图 8.15 (d) 通过假设该特征的垂直位置对应于其深度（深度域中较低）来描绘该场。 这是视差的再现，在这种情况下，会产生侧向运动的错觉，并将其与偏航运动区分开来。

**垂直移动：**视点向上平移，对应于正向  $v_x$ ，导致向下移动，如图 8.15 (e) 所示。 视差再次导致特征的速度取决于相应对象的距离。这样可以将纵向切割与俯仰切割区分开来。

**向前/向后的移动：**如果视点沿着光轴从场景转移（正向  $v_z$ ），那么特征向内流向图像中心，如图 8.15 (f) 所示。它们的速度取决于它们距图像中心的距离以及它们在虚拟世界中相应对象的距离。 由此产生的错觉是后退运动。在负  $z$  方向上的平移会导致向前的运动（例如在“星球大战”电影中跳转到超空间之后的千年猎鹰飞船）。

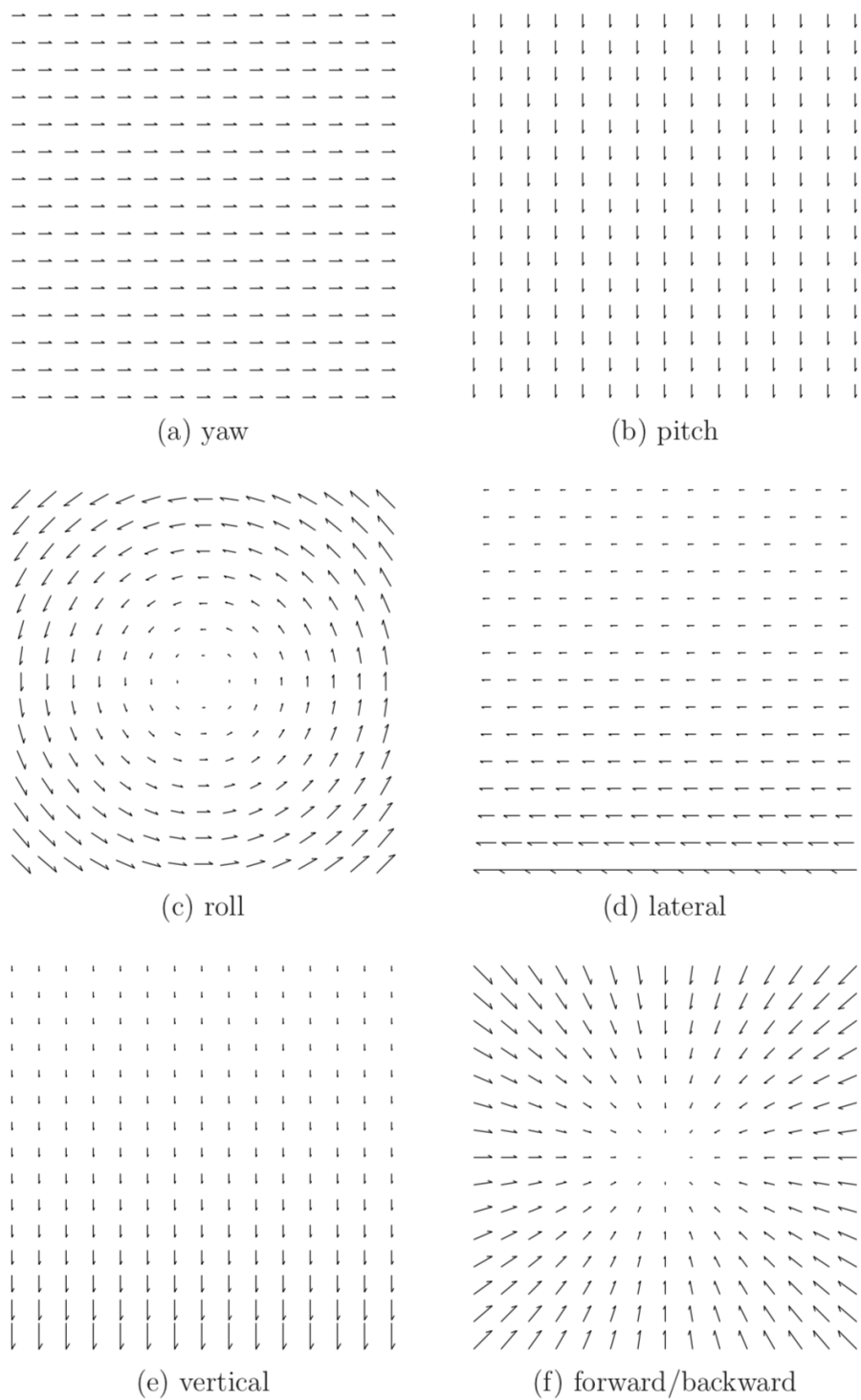


图 8.15: 根据刚体运动的六个自由度, 六种不同类型的光流。 这些中的每一个都是 **vection** 的一个贡献组成部分

前两种有时被称为圆形切面, 后三种被称为线性切割。由于我们的眼睛被吸引到移动的特征上, 所以改变视点可能会引发顺利追踪眼球运动 (请参阅第 5.3 节)。在这种情况下, 图 8.15 所示的光流与视网膜上的特征不一致。因此, 我们迄今为止的表征忽略了眼动, 其经常被设计用于抵消光流并在视网膜上提供稳定的图像。尽管如此, 由于本体感受, 大脑知道这些眼球旋转, 这导致了对自身运动的相同感知。



所有形式的动力都会引起感知速度，但加速度的感觉更为复杂。首先考虑视点的纯粹旋转。如果偏航角度，俯仰角度和侧倾角度的旋转速度变化，则会感知到角加速度。即使在角速度恒定的情况下，偏航，俯仰或侧倾也可以感知线性加速度。这是由于图 8.2 (b) 所示的旋转木马效应。

现在考虑纯粹的线性运动（不旋转）。视点的任何线性加速度都将被视为加速度。但是，如果视点以恒定速度移动，那么这是唯一没有感知加速度的视觉形式。在 VR 头戴式设备中，用户可能会感觉到由于光学失真或渲染和显示中的其他缺陷而导致的加速度。

## 前庭不匹配

我们尚未考虑这六种情况中每一种与前庭线索不匹配的效果。如果使用者没有相对于地球移动，那么只有重力应该被前庭器官（特别是耳石器官）感知。假设用户面向前而没有任何倾斜。在这种情况下，任何感知的加速度都会导致不匹配。例如，偏航俯仰应该引起朝向旋转中心的可感知的恒定加速度（回忆图 8.2 (b)），这与前庭重力线索不匹配。作为另一个例子，向下的垂直视觉应该使得使用者感觉他正在跌倒，但是前庭提示会另外指示。

对于在恒定角速度下的移动，俯仰和翻滚运动的情况，由于半圆形运河测量角加速度，所以与前庭器官感测到的旋转可能没有冲突。因此，视点的角速度必须改变以引起前庭系统的这部分不匹配。由于耳石器官感知到的线性加速度不匹配，可能会产生疾病。

如果头部实际在移动，那么前庭器官就会受到刺激。这种情况更难理解，因为与现实世界中线性和角度加速度相对应的前庭线索与表征不同加速度的视觉线索相结合了。在某些情况下，这些线索可能更一致，而在其他情况下，它们可能会进一步分歧。

## 影响灵敏度的因素

虚拟移动的强度受很多因素影响：

视场角百分比：如果只有一小部分视野在移动，那么人们倾向于认为它是由运动物体引起的。然而，如果大部分视野正在移动，那么他们就认为自己正在移动。人类视觉系统实际上包括具有感受野的神经元，其覆盖大部分视网膜以检测自身运动[35]。随着 VR 头戴式设备增加了视野，他们投射到视网膜的更大区域，从而加强了视觉线索。

曝光时间：由于暴露在光流中的时间而导致的对于自身运动的感知增加。如果曝光时间非常短暂，如几毫秒，则不会发生任何动作。

空间频率：如果虚拟世界很复杂，具有许多小的结构或纹理，那么视觉特征的数量将大大增加，并且光流变成更强的信号。随着 VR 设备显示分辨率的提高，可以生成更高的空间频率。

对比度：对比度越高，光流信号越强，因为更容易检测到这些特征。因此，典型的情况是强度更大。

其他感官线索：回顾 6.4 节，感知现象取决于许多线索的组合。可以通过提供更多一致的线索来增强观察力。例如，向前移动可能伴随着用户脸上的风扇，隆隆的引擎以及虚拟世界中的静止物体的声音。同样地，可以通过提供与没有发生相应的运动的现实世界一致的提示来减弱视觉效果。

先验知识：只要事先知道应该感知什么样的动作会影响移动的进行。这导致了先前的偏差，如果偏差与自身运动相反，则可能需要更长的时间才能克服，但如果与自身运动一致，则需要更少的时间来克服。先前的偏见可能来自某人告诉用户将要发生的事情，或者可能仅仅是通过用户一生中类似视觉体验的积累。此外，用户可能期望该动作是由于所采取的动作而产生的，例如转动虚拟车辆的方向盘。

注意力：如果用户被其他活动分散注意力，例如瞄准虚拟武器或选择菜单选项，则可能

会减轻虚拟移动及其副作用。

事先训练或适应：在足够的暴露下，身体可以学会将真实运动与真实运动区分开来，以使运动变得舒适。因此，许多用户可以接受培训，通过反复，长时间的接触来克服 VR 疾病。

由于所有这些因素以及现代 VR 头戴式耳机的不完善性，很难表征虚拟移动的效能及其对用户舒适性的副作用。

## 进一步阅读

有关矢量场，速度和动力学系统的基本概念，请参见[12]。机械动力系统的建模和分析见[275]。[364,365]介绍了人体运动的具体问题。有关游戏引擎的概述，请参见[103]，包括模拟物理和碰撞检测等问题。有关特定碰撞检测算法的介绍，请参见[100，182]。

前庭系统的一个很好的介绍，包括它作为一个动态系统的反应[150]。在文献[145]中对透视和视觉晕动病进行了深入调查，其中包括大量的参考文献供进一步阅读。一些重要的文章阐述了对动机的敏感性[6,13,69,178,179,279,340]。