1 动画原理

在上一节中, 我们主要讨论各种物理现象的模拟方法. 然而对于具有能动性的对象, 如人和动物, 我们往往需要通过动画来表现其运动. 本节将介绍动画的基本原理和技术.

1.1 旋转的四元数表示

1.1.1 四元数的基本概念

我们在**几何变换**一节中介绍了用矩阵表示物体旋转的方法. 现在我们介绍另一种常用的表示旋转的方法: 四元数.

定义 1.1 四元数

定义四元数 $q = a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$, 其中 $a, b, c, d \in \mathbb{R}$, 且 $\mathrm{i}, \mathrm{j}, \mathrm{k}$ 满足以下乘法规则:

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1, \quad \mathrm{ij} = -\mathrm{ji} = \mathrm{k}, \quad \mathrm{jk} = -\mathrm{kj} = \mathrm{i}, \quad \mathrm{ki} = -\mathrm{ik} = \mathrm{j}.$$

由于四元数虚部的运算规则与三维空间中的叉乘类似, 因此我们可以将四元数 q 表示为 $q = [w, \boldsymbol{v}]$, 其中 $\boldsymbol{v} = (b, c, d)^{\mathrm{t}} \in \mathbb{R}^3$.

定理 1.2 四元数的乘法 四元数 $q_1 = [w_1, \boldsymbol{v}_1], q_2 = [w_2, \boldsymbol{v}_2]$ 的乘积为

$$q_1 q_2 = [w_1 w_2 - \boldsymbol{v}_1 \cdot \boldsymbol{v}_2, w_1 \boldsymbol{v}_2 + w_2 \boldsymbol{v}_1 + \boldsymbol{v}_1 \times \boldsymbol{v}_2]$$

证明. 根据四元数的定义和向量叉积的定义即可得出上式. □

并且由此不难得出四元数的运算满足结合律和分配律, 但不满足交换律.

定义 1.3 四元数的模长与单位四元数

定义四元数 q 的**模长**为 $||q|| = \sqrt{q \cdot q} = \sqrt{w^2 + \boldsymbol{v} \cdot \boldsymbol{v}}$.

定义**单位四元数**为模长为 1 的四元数. 单位四元数可以表示为 $q = [\cos\theta, \boldsymbol{u}\sin\theta]$, 其中 $\boldsymbol{u}$ 是三维的单位向量.

定义 1.4 四元数的共轭与逆元

定义四元数 $q = [w, \boldsymbol{v}]$ 的共轭 $q^* = [w, -\boldsymbol{v}]$ 为其虚部取相反数的结果.

根据四元数乘法的定义不难得出

$$qq^* = [w^2 + \boldsymbol{v} \cdot \boldsymbol{v}, \boldsymbol{0}] = ||q||^2$$

于是可以定义四元数 q 的**逆元**为

$$q^{-1} = \frac{q^*}{||q||^2}$$

并且总有 $qq^{-1}=q^{-1}q=[1,\mathbf{0}]$. 对于单位四元数而言总有 $q^*=q^{-1}$.

1.1.2 用四元数表示旋转

定理 1.5 旋转的四元数表示 设 $\boldsymbol{v}$ 是三维空间内的单位向量, $q=[\cos\theta,\boldsymbol{u}\sin\theta]$ 为单位四元数. 令 $v=[0,\boldsymbol{v}]$, 则 qvq^* 的实部为 0, 虚部为 $\boldsymbol{v}$ 绕旋转轴 $\boldsymbol{u}$ 旋转弧度 2θ 后得到的向量.

证明. 由四元数的乘法定义可得

$$\begin{aligned}qvq^* &= [\cos\theta,\boldsymbol{u}\sin\theta][0,\boldsymbol{v}][\cos\theta,-\boldsymbol{u}\sin\theta]\\ &= [-\boldsymbol{u}\cdot\boldsymbol{v}\sin\theta,\cos\theta\boldsymbol{v}+\sin\theta(\boldsymbol{u}\times\boldsymbol{v})][\cos\theta,-\boldsymbol{u}\sin\theta]\\ &= [0,\boldsymbol{v}\cos 2\theta+(\boldsymbol{u}\times\boldsymbol{v})\sin 2\theta+\boldsymbol{u}(\boldsymbol{u}\cdot\boldsymbol{v})(1-\cos 2\theta)]\end{aligned}$$

由此可见 qvq^* 的实部为 0. 而其虚部正是 $\boldsymbol{v}$ 绕旋转轴 $\boldsymbol{u}$ 旋转弧度 2θ 后得到的向量 (根据我们在旋转矩阵的推导中的结论可知). □

1.2 运动学

1.2.1 前向运动学

以人体为例, 我们可以把各个关节视作节点, 各个骨骼视作连接节点的边. 这样, 人体就可以表示为一个带有根节点的树形结构. 我们只需要知道根节点的位置和各个关节的旋转角度, 就可以计算出各个节点的位置.

以刚体铰链模型 $P_0\cdots P_n$ 为例展示前向运动学的过程. 节点 $P_0,\cdots,P_n$ 的初始位置记作 $\boldsymbol{p}_0^{\mathrm{st}},\cdots,\boldsymbol{p}_n^{\mathrm{st}}$, 向下一个关节的位移即为 $\boldsymbol{l}_i=\boldsymbol{p}_i^{\mathrm{st}}-\boldsymbol{p}_{i-1}^{\mathrm{st}}(i=1,\cdots,n)$, 每个节点上都带有一个局部坐标系 $\mathcal{C}_i$.

现在, 我们为除去 P_n 外的每个关节 P_j 指定一个旋转矩阵 $\boldsymbol{R}_j^{\mathrm{loc}}(j=1,\cdots,n-1)$(这里的旋转矩阵是基于该关节关联的局部坐标系 $\mathcal{C}_j$ 所定义的), 每个关节 P_j 处的旋转带动其后的所有关节 (包括其上带的局部坐标系) 和骨骼旋转. 因此, 节点 P_j 的总的旋转矩阵 $\boldsymbol{R}_j^{\mathrm{tot}}$ 即为它的局部坐标系 $\mathcal{C}_j$ 相对于根节点的坐标系 $\mathcal{C}_0$ 的旋转与它本身的旋转的复合. 而 $\mathcal{C}_j$ 的旋转事实上就是由 P_{j-1} 节点的旋转定义的, 因此 $\boldsymbol{R}_j^{\mathrm{tot}}=\boldsymbol{R}_{j-1}^{\mathrm{tot}}\boldsymbol{R}_j^{\mathrm{loc}}$.

现在, 我们就可以从根节点开始计算每个子节点的位置 $\boldsymbol{p}_j^{\mathrm{ed}}(j=1,\cdots,n)$. 根节点的位置 $\boldsymbol{p}_0^{\mathrm{ed}}=\boldsymbol{p}_0^{\mathrm{st}}$ 已知, 而每个节点的位置可以通过其父节点的位置加上旋转后的位移向量得到, 即

$$\boldsymbol{p}_j^{\mathrm{ed}}=\boldsymbol{p}_{j-1}^{\mathrm{ed}}+\boldsymbol{R}_{j-1}^{\mathrm{tot}}\boldsymbol{l}_j$$

1.2.2 逆向运动学

逆向运动学 (Inverse Kinematics, IK) 的任务与前向运动学恰恰相反, 其需要根据给定的部分关节的位置 $\boldsymbol{p}_i$(包括但是不仅限于末关节)，反向求解出一组关节局部旋转 $\boldsymbol{R}_i^{\mathrm{loc}}$ 的解. 逆向运动学可以被用于机械臂或者是虚拟角色的控制, 也可以帮助修复一些质量较差, 没有满足部分接触要求的动作捕捉数据.

对于简单的, 具有两个骨骼的系统, IK 问题有解析的解. 然而, 对于更复杂的系统, 我们只能通过数值方法

求解 IK 问题. 下面介绍两种常用的方法: **循环坐标推演 (Cyclic Coordinate Descent, CCD)** 和**前向后向抵达 (Forward and Backward Reaching, FABR)**.

CCD IK CCD IK 的思想是从末端关节开始, 依次调整每个关节的旋转, 使得末端关节逐渐接近目标位置.

具体地, 假定当前铰接刚体系统共有 $n+1$ 个关节与 n 个连接关节的骨骼, 我们希望末端的关节位置 $\boldsymbol{p}_n$ 距离目标 t 尽可能接近. 于是遍历第 $n-1,\cdots,0$ 个关节, 每次求解关节 i 处的旋转矩阵 $\boldsymbol{R}_i^{\text{loc}}$ 使得调整后 $\boldsymbol{p}_i$, $\boldsymbol{p}_n$ 和 t 三点共线[1], 然后将更新后的 $\boldsymbol{R}_i^{\text{loc}}$ 应用于第 i 个关节及其后续的所有关节, 更新它们的位置. 重复上述过程直到末端关节 $\boldsymbol{p}_n$ 足够接近目标 t 或者达到最大迭代次数为止.

FABR IK FABR IK 的思想是通过两次遍历关节链, 根据骨骼长度调整关节位置, 使得末端关节接近目标位置.

具体地, 假定当前铰接刚体系统共有 $n+1$ 个关节与 n 个连接关节的骨骼, 我们希望末端的关节位置 $\boldsymbol{p}_n$ 距离目标 t 尽可能接近.

首先是 Backward 步骤的计算. 我们将末端关节 $\boldsymbol{p}_n$ 移动到目标位置 t, 记 $\boldsymbol{p}_n^{\text{back}}=\boldsymbol{t}$. 接着, 我们从末端关节向根节点遍历每个关节 $i(i=n-1,\cdots,0)$, 将 $\boldsymbol{p}_i$ 更新到 $\boldsymbol{p}_i^{\text{back}}$ 使得

$$\left|\boldsymbol{p}_{i+1}^{\text{back}}-\boldsymbol{p}_i^{\text{back}}\right|=|\boldsymbol{l}_{i+1}|,\quad \boldsymbol{p}_i,\boldsymbol{p}_i^{\text{back}},\boldsymbol{p}_{i+1}^{\text{back}}\text{共线}$$

于是可得

$$\boldsymbol{p}_i^{\text{back}}=\boldsymbol{p}_{i+1}^{\text{back}}+\frac{\boldsymbol{p}_i-\boldsymbol{p}_{i+1}^{\text{back}}}{\left|\boldsymbol{p}_i-\boldsymbol{p}_{i+1}^{\text{back}}\right|}|\boldsymbol{l}_{i+1}|$$

如此往复直到根节点位置[2]. 现在, 我们得到了一组新的关节位置 $\left\{\boldsymbol{p}_i^{\text{back}}\right\}_{i=1}^n$.

接下来是 Forward 步骤的计算. 记 $\boldsymbol{p}_0^{\text{for}}=\boldsymbol{p}_0^{\text{st}}$. 接着, 我们从根节点向末端遍历每个关节 $i(i=1,\cdots,n)$, 将 $\boldsymbol{p}_i^{\text{back}}$ 更新到 $\boldsymbol{p}_i^{\text{for}}$ 使得

$$\left|\boldsymbol{p}_i^{\text{for}}-\boldsymbol{p}_{i-1}^{\text{for}}\right|=|\boldsymbol{l}_i|,\quad \boldsymbol{p}_i^{\text{back}},\boldsymbol{p}_i^{\text{for}},\boldsymbol{p}_{i-1}^{\text{for}}\text{共线}$$

于是可得

$$\boldsymbol{p}_i^{\text{for}}=\boldsymbol{p}_{i-1}^{\text{for}}+\frac{\boldsymbol{p}_i^{\text{back}}-\boldsymbol{p}_{i-1}^{\text{for}}}{\left|\boldsymbol{p}_i^{\text{back}}-\boldsymbol{p}_{i-1}^{\text{for}}\right|}|\boldsymbol{l}_i|$$

如此往复直到末端节点位置. 现在, 我们得到了一组新的关节位置 $\left\{\boldsymbol{p}_i^{\text{for}}\right\}_{i=1}^n$. 这组新的关节满足骨骼长度约束, 并且 $\boldsymbol{p}_n$ 相比之前更接近目标 $\boldsymbol{t}$.

重复上述的 Backward 和 Forward 步骤直到末端关节 $\boldsymbol{p}_n$ 足够接近目标 t 或者达到最大迭代次数为止.

Jacobian IK 我们先来介绍一些有关 Jacobian 矩阵的知识. 考虑非线性变换 $(x,y)\mapsto(u,v)$, 其微分关系为

$$\mathrm{d}u=\frac{\partial u}{\partial x}\mathrm{d}x+\frac{\partial u}{\partial y}\mathrm{d}y,\quad \mathrm{d}v=\frac{\partial v}{\partial x}\mathrm{d}x+\frac{\partial v}{\partial y}\mathrm{d}y$$

[1] 确切地说, 是令 $\boldsymbol{p}_n-\boldsymbol{p}_i$ 和 $\boldsymbol{t}-\boldsymbol{p}_i$ 同向. 在 `glm` 库中, 求解将向量 `a` 旋转到向量 `b` 对应的四元数可以用 `glm::rotate` 函数实现.

[2] 实际上根节点的位置不用更新.

写成矩阵的形式则有

$$\begin{bmatrix} \mathrm{d}u \\ \mathrm{d}v \end{bmatrix} = \begin{bmatrix} \dfrac{\partial u}{\partial x} & \dfrac{\partial u}{\partial y} \\ \dfrac{\partial v}{\partial x} & \dfrac{\partial v}{\partial y} \end{bmatrix} \begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \end{bmatrix}$$

记 $\mathrm{d}\boldsymbol{a} = \begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \end{bmatrix}, \mathrm{d}\boldsymbol{b} = \begin{bmatrix} \mathrm{d}u \\ \mathrm{d}v \end{bmatrix}, \boldsymbol{J} = \begin{bmatrix} \dfrac{\partial u}{\partial x} & \dfrac{\partial u}{\partial y} \\ \dfrac{\partial v}{\partial x} & \dfrac{\partial v}{\partial y} \end{bmatrix}$, 则有

$$\mathrm{d}\boldsymbol{b} = \boldsymbol{J}\mathrm{d}\boldsymbol{a}$$

这就描述了两个坐标系之间的非线性变换的性质, 矩阵 $\boldsymbol{J}$ 被称作这一变换的 Jacobian 矩阵.

现在我们考虑刚体铰链模型, 末端位置 $\boldsymbol{p} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{t}}$ 仅与各个关节的旋转角度 θ_i 有关. 记 $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \cdots & \theta_n \end{bmatrix}^{\mathrm{t}}$, 类似地就有

$$\mathrm{d}\boldsymbol{p} = \boldsymbol{J}\mathrm{d}\boldsymbol{\theta}$$

我们的目标是为了确定末端位置如何从一个位置 $\boldsymbol{p}_e$ 移动到另一个位置 $\boldsymbol{p}_t$, 这要求系统末端的位移为 $\Delta\boldsymbol{p} = \boldsymbol{p}_e - \boldsymbol{p}_t$. 当位移 $\Delta\boldsymbol{p}$ 取的足够小时, 我们就根据上述微分式近似地得到

$$\Delta\boldsymbol{p} = \boldsymbol{J}\Delta\boldsymbol{\theta}$$

然后对式子两边左乘 Jacobian 矩阵的逆即可得到

$$\Delta\boldsymbol{\theta} = \boldsymbol{J}^{-1}\Delta\boldsymbol{p}$$

根据位置变换 $\Delta\boldsymbol{p}$ 即可求得各关节的角度变化 $\Delta\boldsymbol{\theta}$.

1.3 动作捕捉

动作捕捉 (Motion Capture, Mocap) 是通过设备记录人和物体运动的方法. 就角色来说, 我们需要一些专业的动捕演员, 让其穿戴特定的设备进行表演. 在每一时刻, 演员的肢体运动甚至面部表情会都会被设备记录下来, 转化成骨骼的旋转数据, 记录完毕后将其交给后续的处理流程. 我们接下来介绍一些常用的动作捕捉设备及其原理.

1.3.1 外骨骼

外骨骼是一种基于机械结构的动作捕捉设备. 演员需要穿戴一个带有多个传感器的外骨骼装置, 这些传感器通常安装在关节处, 用于测量关节的旋转角度. 通过读取这些传感器的数据, 我们可以实时获取演员各个关节的旋转信息, 从而重建出演员的动作.

外骨骼的优点是可以提供高精度的关节旋转数据, 且不受外部环境光线和遮挡的影响. 然而, 外骨骼设备通常较为笨重, 限制了演员的动作自由度.

1.3.2 光学动作捕捉

光学动作捕捉系统是目前最常用的动作捕捉技术, 通常使用多个摄像头来跟踪演员身上的发光或反光的标记点. 摄像头捕捉到标记点的位置后, 通过三角测量法计算出标记点在三维空间中的位置. 然后, 通过对这些位置数据进行处理, 可以推断出演员的骨骼姿态和动作.

1.4 动作合成

动作捕捉得到的数据一般需要经过一系列的处理. 本节主要介绍一些动作合成的技术.

1.4.1 动作连接

我们采集到的动捕数据往往只是一个动作片段, 为生成完整, 流畅的动作还需要将多段动捕数据中的动作串联起来.

1.4.2 动作对齐

简单的动作连接不能解决动作朝向不对齐的问题. 例如动作 A 是一段向右的行走, 动作 B 是一段向左的跑步, 那么我们拼接之后会发现虚拟角色在从走切换到跑时突然调转了一个方向, 并且在此过程中脚底还会出现明显的打滑现象. 我们需要对齐 A 与 B 中人物的运动方向.

在进行运动方向对齐前, 我们先定义角色的**朝向坐标系 (Facing Frame)**.

定义 1.6 朝向坐标系 朝向坐标系是固定在角色上的一个局部坐标系, 可以用参数 $(\boldsymbol{R}, \boldsymbol{t})$ 表示; 旋转矩阵 $\boldsymbol{R}$ 表示局部坐标系在世界坐标系下的旋转, 平移向量 $\boldsymbol{t}$ 表示局部坐标系的原点在世界坐标系下的位置.

一般而言, $\boldsymbol{t}$ 取角色的根关节在世界坐标系下的位置, $\boldsymbol{R}$ 则是一个绕 y 轴的旋转, 满足朝向坐标系的 z 轴指向角色面朝的方向.

现在, 记 transition frame 时刻 A, B 序列中的朝向坐标系分别为 $\boldsymbol{R}_{A0}, \boldsymbol{t}_{A0}$ 和 $\boldsymbol{R}_{B0}, \boldsymbol{t}_{B0}$, 我们需要对 B 序列的朝向坐标系进行变换 $\mathcal{T}$ 使得其与动作 A 的朝向坐标系在初始状态下先对齐, 也即让动作 A 和动作 B 在连接开始时具有相同的朝向; 然后考虑 transition frame 后的某一帧 i, 在这一帧时动作序列 A 和 B 中的朝向坐标系分别为 $\boldsymbol{R}_A(i), \boldsymbol{t}_A(i)$ 和 $\boldsymbol{R}_B(i), \boldsymbol{t}_B(i)$. 我们对 $\boldsymbol{R}_B(i), \boldsymbol{t}_B(i)$ 做相同的变换 $\mathcal{T}$, 就得到了对齐后的动作 B', 然后对 A 与 B' 进行动作连接即可.

考虑在帧 i 时刻任取一个 $\boldsymbol{R}_A(i), \boldsymbol{t}_A(i)$ 坐标系下的三维坐标 $\boldsymbol{x}$, 设其转换为 $\boldsymbol{R}_{A0}, \boldsymbol{t}_{A0}$ 坐标系下的坐标为 $\boldsymbol{x}_0$; 取 $\boldsymbol{R}_B(i), \boldsymbol{t}_B(i)$ 坐标系下的同样的坐标 $\boldsymbol{x}$, 设其转换为 $\boldsymbol{R}_{B0}, \boldsymbol{t}_{B0}$ 坐标系下的坐标为 $\boldsymbol{x}_1$, 那么应当有如下关系:

$$\boldsymbol{x}_0 = \boldsymbol{x}_1$$