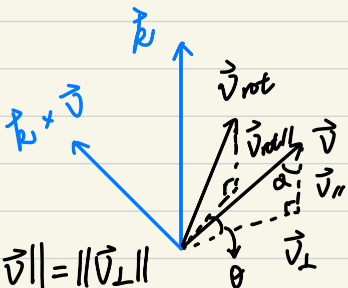
Rodrigues 公式

输入原向量，旋转轴和旋转角
输出旋转后的向量

$\vec{v}$ 是三维空间中的向量，$\vec{k}$ 是与转轴同向的单位向量，θ 是 $\vec{v}$ 绕 $\vec{k}$ 的右手（逆时针）方向转过的角，求旋转后的 $\vec{v}_{rot}$。



step1. 将 $\vec{v}$ 分解为 $\vec{v}_{\perp}$ 和 $\vec{v}_{\parallel}$，则 $\vec{v}=\vec{v}_{\perp}+\vec{v}_{\parallel}$

显然，$\vec{v}_{\parallel}=(\vec{v}\cdot\vec{k})\cdot\vec{k}$

$\vec{v}_{\perp}=\vec{v}-\vec{v}_{\parallel}=(\vec{k}\cdot\vec{k})\cdot\vec{v}-(\vec{k}\cdot\vec{v})\vec{k}$

$\|\vec{k}\times\vec{v}\|=\|\vec{v}_{\perp}\|$

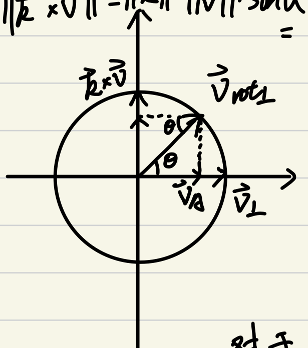
因为 $\|\vec{k}\times\vec{v}\|=\|\vec{k}\|\cdot\|\vec{v}\|\cdot\sin\alpha=\|\vec{v}_{\perp}\|$

由向量三重积展开

$$\vec{a}\times(\vec{b}\times\vec{c})=(\vec{a}\cdot\vec{c})\vec{b}-(\vec{a}\cdot\vec{b})\cdot\vec{c}$$

令 $\vec{a}=\vec{k}, \vec{c}=\vec{k}, \vec{b}=\vec{v}$

则 $\vec{v}_{\perp}=\vec{k}\times(\vec{v}\times\vec{k})=-\vec{k}\times(\vec{k}\times\vec{v})$



从图中几何关系可以看出 $\vec{v}_{rot\parallel}=\vec{v}_{\parallel}$
现在只要求 $\vec{v}_{rot\perp}$，而且 $\|\vec{v}_{rot\perp}\|=\|\vec{v}_{\perp}\|$

对于 $\vec{v}_{rot\perp}$，可以表示为 $\vec{v}_{\perp}$ 和 $\vec{k}\times\vec{v}$ 向量的和。
所以将 $\vec{v}_{rot\perp}$ 表示为这两个向量方向单位向量乘以各自对应的标量后得到的向量和

$\|\vec{v}_A\|=\|\vec{v}_{rot\perp}\|\cos\theta$

$\|\vec{v}_A\|=\|\vec{v}_{\perp}\|\cdot\cos\theta$

$\vec{v}_A=\|\vec{v}_{\perp}\|\cdot\cos\theta\cdot\hat{a}$

$=\vec{v}_{\perp}\cdot\cos\theta$

同理 $\vec{v}_B=(\vec{k}\times\vec{v})\sin\theta$

$$\vec{v}_{rot\perp}=\frac{\vec{v}_{\perp}}{\|\vec{v}_{\perp}\|}\cdot\|\vec{v}_{\perp}\|\cos\theta+\frac{\vec{k}\times\vec{v}}{\|\vec{k}\times\vec{v}\|}\cdot\|\vec{k}\times\vec{v}\|\cdot\sin\theta$$

$$=\vec{v}_{\perp}\cdot\cos\theta+(\vec{k}\times\vec{v})\sin\theta$$

$$\vec{V}_{rot} = \vec{V}_{rot\parallel} + \vec{V}_{rot\perp}$$
$$= \vec{v}_{\parallel} + \vec{v}_{\perp}\cdot\cos\theta + (\vec{k}\times\vec{v})\sin\theta$$
$$= \vec{v}_{\parallel} + (\vec{v} - \vec{v}_{\parallel})\cos\theta + (\vec{k}\times\vec{v})\sin\theta$$
$$= \cos\theta\cdot\vec{v} + (1-\cos\theta)\vec{v}_{\parallel} + (\vec{k}\times\vec{v})\sin\theta$$
$$= \cos\theta\cdot\vec{v} + (1-\cos\theta)\cdot(\vec{k}\cdot\vec{v})\cdot\vec{k} + (\vec{k}\times\vec{v})\sin\theta$$
$$= \vec{v} - (1-\cos\theta)\vec{v} + (1-\cos\theta)(\vec{k}\cdot\vec{v})\cdot\vec{k} + (\vec{k}\times\vec{v})\sin\theta$$
$$= \vec{v} + (1-\cos\theta)\left[(\vec{k}\cdot\vec{v})\vec{k} - (\vec{k}\cdot\vec{k})\cdot\vec{v}\right] + (\vec{k}\times\vec{v})\sin\theta$$

由向量三重积公式展开得

$$\vec{a}\times(\vec{b}\times\vec{c}) = (\vec{a}\cdot\vec{c})\cdot\vec{b} - (\vec{a}\cdot\vec{b})\cdot\vec{c}$$

令 $\vec{c}=\vec{v}$　$\vec{a}=\vec{b}=\vec{k}$

上式 $= \vec{v} + (1-\cos\theta)\left[\vec{k}\times(\vec{k}\times\vec{v})\right] + (\vec{k}\times\vec{v})\sin\theta$

由公式

$$\vec{a}\times\vec{b} = \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} \times \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} 0 & -z_a & -y_a \\ z_a & 0 & -x_a \\ -y_a & x_a & 0 \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = A\cdot\vec{b}$$ 替换上式

令 R_k 替换 $\vec{k}$ 得

$$\vec{V}_{rot} = \vec{v} + (1-\cos\theta)\left[R_k\cdot(R_k\cdot\vec{v})\right] + R_k\cdot\vec{v}\sin\theta$$
$$= \left[I + \sin\theta\cdot R_k + (1-\cos\theta)R_k^2\right]\cdot\vec{v}$$
$$\Downarrow$$
$$\vec{V}_{rot} = M\vec{v}$$

M为所求旋转矩阵

? θ怎么求(空间中两点相对位姿)

$\vec{w}_k \in \mathbb{R}^3$ $\quad \vec{w}_k = (x, y, z)^T = \theta \cdot (\bar{x}, \bar{y}, \bar{z})^T$

$\|\vec{w}_k\| = |\theta| \, \|(\bar{x}, \bar{y}, \bar{z})^T\| = |\theta|$

利用Rodrigues有 R_k，即 $\hat{w} = \begin{bmatrix} 0 & \bar{z} & -\bar{y} \\ \bar{z} & 0 & -\bar{x} \\ -\bar{y} & \bar{x} & 0 \end{bmatrix}$

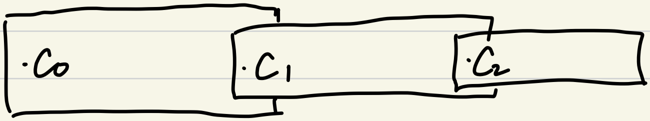
$$\exp(\vec{w}_k) = M = I + \sin(\|\vec{w}_k\|) \cdot \hat{w}_k + \cos(\|\vec{w}_k\|) \cdot \hat{w}_k^2$$

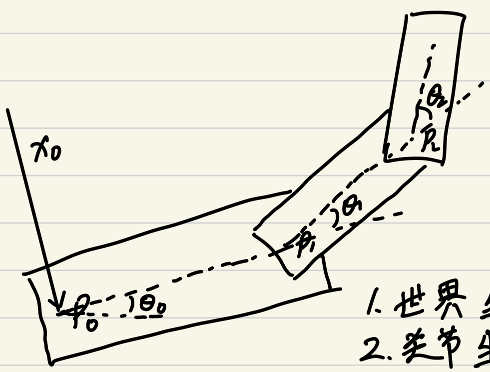
骨骼动画(正向动力学)

控制每个关节的坐标 → 骨骼的姿态 → 角色

实际上，每个关节节点存储的是相对父节点的坐标，更准确的是相对父节点的坐标变换矩阵.

关节模型



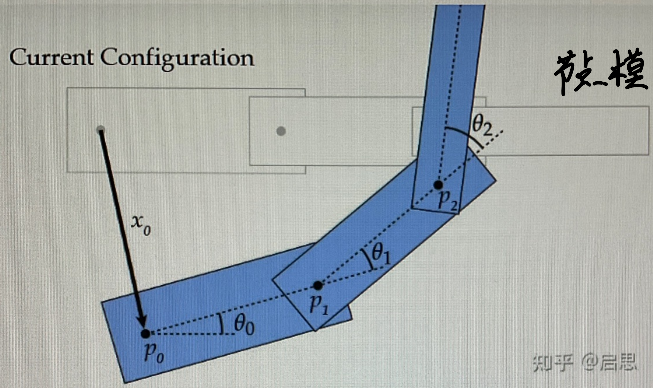


为了定义整个关节模型

1. 每个关节所在骨骼的长度
2. 相对上一关节的转角
3. 根节点的坐标

1. 世界坐标系 world
2. 关节坐标系 (以某个关节为原点的坐标系
3. 骨架(物体)坐标系 树根节点为原点，就是②的特殊情况

关节坐标系 $\xrightarrow{\text{正向运动学}}$ 世界坐标系



节点模型中保存的是子到父的R.T（因为要逐级往上转坐标系）

我们要得到一个节点在世界坐标系下的坐标。

这里的根节点坐标是 x_0 。定义关节的末端影响器为这个关节伸出去的末端。**如何计算末端影响器的世界坐标？**

设第 i 个关节对应的旋转矩阵是 R_i ，平移矩阵是 T_i ，原点坐标 $O=(0,0,0)$ ，那么：

关节坐标指当前节点在自己坐标系下的坐标。

1. p_0 的关节坐标是 O ，这也是在骨架坐标系的坐标，通过根节点的世界坐标进行变换，即对应的世界坐标是 $x_0+O=x_0$
2. p_1 的关节坐标是 O ，首先由 T_1 进行一个平移，再由 R_1 进行旋转，得到它在父关节的关节坐标系坐标 R_1T_1O ，这也是它在骨架坐标系的坐标，所以对应的世界坐标是 $x_0+R_1T_1O$
3. p_2 的关节坐标是 O ，同理可得它的世界坐标是 $x_0+R_1T_1R_2T_2O$
4. p_3 同理，不过没在图上标注，就是最后一块骨骼的末端。世界坐标是 $x_0+R_1T_1R_2T_2R_3T_3O$

如果我们把 x_0 也看做一个平移变换 T_0，再添加一个旋转变换 $R_0=I$，那么很容易发现，从只需要拿原点坐标乘平移、旋转矩阵，再乘父节点，一直乘到根节点，就可以得到关节的末端影响器的世界坐标。即

$$x_i = R_{root}T_{root}\dots R_{fai}T_{fai}R_iT_iO$$

令 $M_{i\ to\ world} = R_{root}T_{root}\dots R_{fai}T_{fai}R_iT_i$

所以任意 $p^{(w)} = M_{i\ to\ world}\cdot p$

关节坐标系中任意点p的坐标。

所以正向动力学解决了给定角度θ可以求解顶点世界坐标的问题，即 $x=f(\theta)$

而逆向动力学解决的是给定坐标，求解旋转角 θ 的问题.

即 $\theta = f^{-1}(x)$

应用：补非关键帧动画

解法一：CCD 循环坐标下降法.

每次选一个关节转一个角度，使末端影响器更接近目标位置.（每次旋转到当前关节与目标点的连线上）

解法二：基于优化的方法. 写出 loss 表达式 梯度↓.

对工业界中常用的 LBS（线性混合蒙皮）

实际上是对每个骨骼的变换矩阵去插值，系数由美工决定.

$$\bar{t}'_i = \sum_{k=1}^{K} W_{k,i}\, G'_k(\vec{\theta}, J)\, \bar{t}_i = \left(\sum_{k=1}^{K} W_{k,i}\, G'_k(\vec{\theta}, J)\right) \cdot \bar{t}_i$$

对节点 i，在T-pose下坐标为 $\bar{t}_i$，有 K 个骨骼控制它，第 k 个骨骼在当前的变换矩阵为 $G'_k(\vec{\theta}, J)$，权重为 $W_{k,i}$，那么最终的世界坐标为 $\bar{t}_i'$

$$\sum_{k=1}^{K} W_{k,i} = 1 \qquad K \le 4$$

$(R_{root} = I \quad T_{root} = x_0)$

在之前，我们已经有 $M_{j\ to\ world} = R_{root} T_{root} \cdots R_j \cdot T_j$

在姿态变换时可以不考虑相对世界的坐标.

$$M_{j\ to\ skeleton} = (R_{root} T_{root})^{-1} \cdot M_{j\ to\ world}.$$

不妨设 T-pose 中关节坐标系到骨架坐标系的变换矩阵为 $M^{(0)}_{j\ to\ skeleton}$，当前时刻关节坐标系到骨架坐标系的变换矩阵为 $M^{(t)}_{j\ to\ skeleton}$.

☆☆☆ $\bar{t}_i$表示的是在初始T-pose下的骨骼坐标.

$$G'_k(\vec{\theta}, J) = G_k(\vec{\theta}, J) \cdot G_k(\vec{\theta}^*, J)^{-1}$$

将$\bar{t}_i$转换到对应的世界坐标系.

t时刻，关节→世界

根据关节模型保存的信息实时计算.

0时刻，骨骼→关节.

0时刻骨骼绑定好后就有的

$$= \left(R_{root} T_{root} \cdots R_j T_j\right)^{-1}$$

$$= R^{(t)}_{root} T^{(t)}_{root} \cdots R^{(t)}_j T^{(t)}_j$$

$$\Downarrow$$

$$G_k(\vec{\theta}, J) = \prod_{j \in A(k)} \begin{bmatrix} \exp(\vec{w}_j) & j_j \\ \vec{0} & 1 \end{bmatrix}$$

递归下去，k的所有父节点.

旋转平移.

smpl 对蒙皮不做改变，但是会对顶点作改变.

$$\bar{t}'_i = \sum_{k=1}^{K} w_{k,i} G'_k(\vec{\theta}, J(\vec{\beta}))(\bar{t}_i + b_{s,i}(\vec{\beta}) + b_{p,i}(\vec{\theta}))$$

体型参数β：拥有10个维度去描述一个人的身体形状，每个维度可以被解释为高矮、胖瘦等.

姿态参数θ：拥有$(23+1)\times3$个维度去描述关节的位姿状态.（3个自由度每节点）

smpl 三过程：基于形状的 blend shape（混合变形）

$$B_s(\beta)=\sum_{n=1}^{|\beta|}\beta_n\cdot S_n$$

S是通过机器学习到的参数，维度为(6890, 3, 10)

10个维度，每一个维度都有一组 6890×3. 6890个顶点的参数.

$\beta\in\mathbb{R}^{10}$ 表示每个维度的权重.

S_n应该是 T-pose 下的骨架坐标.

这里使用直接叠加的方法.

就是往 b 上去靠 $a+w\cdot b$ 去接近 b.

这种方法需要决定 w 的大小，使 $\frac{a}{a+wb}$ 控制在一定范围内.

smpl 三过程：基于姿态的 blend shape

K为23. joint数.

$$B_p(\theta)=\sum_{n=1}^{9K}\left(R_n(\theta)-R_n(\theta^*)\right)P_n$$

因为我们是计算相对 T-pose 状态下的线性叠加偏量，所以人体的位姿应该也是要相对 T-pose 状态下进行变化，因为括号里减去了 T-pose 位姿的影响.

θ 是 23×3 = 69 个数组成的参数

P_n是通过机器学习得到参数，维度是(6890, 3, 207)

在每组pose下全身顶点在骨骼坐标系下的坐标.

23×9. 因为每个pose都用旋转矩阵来表示.

$R_n(\theta)$的作用是通过 Rodrigues 公式来将

θ转换成旋转矩阵 M.

故为 23 个关节. 每个关节对应一个 pose.

这里也使用了 $a+wb$ 的方式去接近 b.

这种方式有一个好处: 就是我有目标的情况下, 我可以以不同程度去接近目标, 而不是直接通过直观的相乘的方式去计算 loss 再优化, 实际上降低了计算和模型复杂度.

transform

zeros (N, 24, 1, 3)   m_poseRot (N, 24, 3, 3)

localTransform

m_joint (N, 24, 3). ← poseRotHomo 齐次坐标.

(N, 24, 4, 3).

< > translations [0,(N,3)]

---

vector<m>[24](3×3)

zeros (24, 1, 3) + poseRot (24, 3, 3) → poseRotHomo (24, 4, 3)

24×3×3

3×3 → 4×3

joints (24, 3) 关节的位置坐标.

祖先位置 ← 相减

得到 (1,3).

3×3

0 0 0 1

kinematic_tree_table

(祖先, 当前的)

joints (24, 3).

加入到.

translations.

unsqueeze.   (24, 3).

translations (24, 3, 1)

+

ones (24, 1, 1)

(24, 4, 4).

localTransformations.

4×3

poseRotHomo + translations (24, 4, 1)

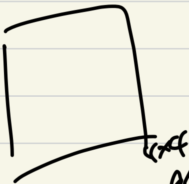
(24, 4, 3)   4×3   4×1

$N \times 24 \times 4 \times 4$ (0 written above the 24)

local Transformations (24, 4, 4)

$[0, (4\times 4)]$

glotransformation [0] = (4×4) 根节点.

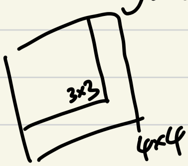


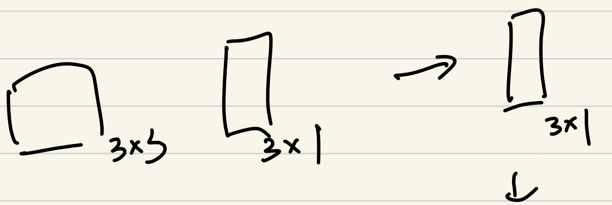
ancestor ↓ glotransformation[ancestor] 4×4 × i ↓ localTransformations. 4×4

↑ 这是一个递归运算的过程

glotransformations (24, 4, 4)

joints (24, 3)





eliminated (24, 3, 1)

zeros (24, 1, 1)

$$\underset{(24,3,1)}{\text{eliminated}} + \underset{(24,1,1)}{\text{zeros}} \longrightarrow \underset{(24,4,1)}{\text{eliminatedHomo}} \quad 24\times \begin{bmatrix} A \\ B \\ C \\ 0 \end{bmatrix}_{4\times 1}$$

$$\underset{(24,4,3)}{\text{zeros}} + \underset{(24,4,1)}{\text{eliminatedHomo}} \rightarrow \underset{24,4,4}{\text{eliminatedHomo}}.$$

$$24\times4\times4 \quad \begin{bmatrix} 0 & 0 & 0 & A \\ 0 & 0 & 0 & B \\ 0 & 0 & 0 & C \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underset{24,4,4}{\text{relativeTransformations}} = \underset{24,4,4}{\text{globalTransformation}} - \underset{24,4,4.}{\text{eliminatedHomo.}}$$

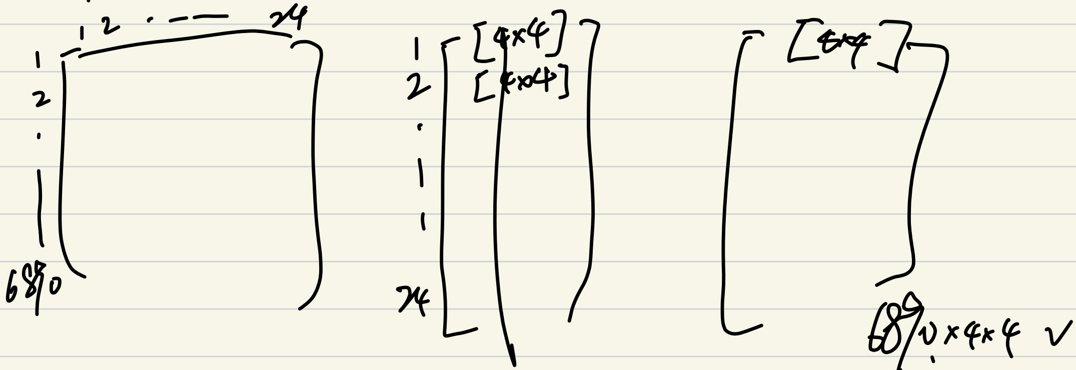
restShape (6890, 3) + ones (6890, 1)

→ restShapehomo (6890, 4) ✓

weights × relativeTransformations → coefficients

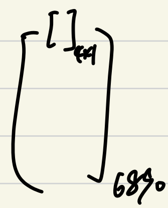
(24, 4, 4) (6890, 4, 4)

6890, 24



restShapeHomo unsqueeze.

6890, 4, 1

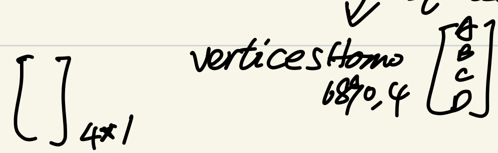


coefficients · restShape Homo. = verticesHomo

6890, 4, 4 6890, 4, 1 6890, 4, 1

↓ squeeze.

verticesHomo
6890, 4



$$\text{verticesHomoW}_{6890,1} \quad \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_{6890} \end{bmatrix}.$$

$$\text{homoUnit} = \text{verticesHomo} / \text{verticesHomoW}. \quad \begin{bmatrix} A/D \\ B/D \\ C/D \\ 1 \end{bmatrix}_{6890,4}$$

$$\text{cart} = \text{homoUnit} \rightarrow \begin{bmatrix} A/D \\ B/D \\ C/D \end{bmatrix}_{6890,3}.$$