

# 关节(角色)动画

金小刚

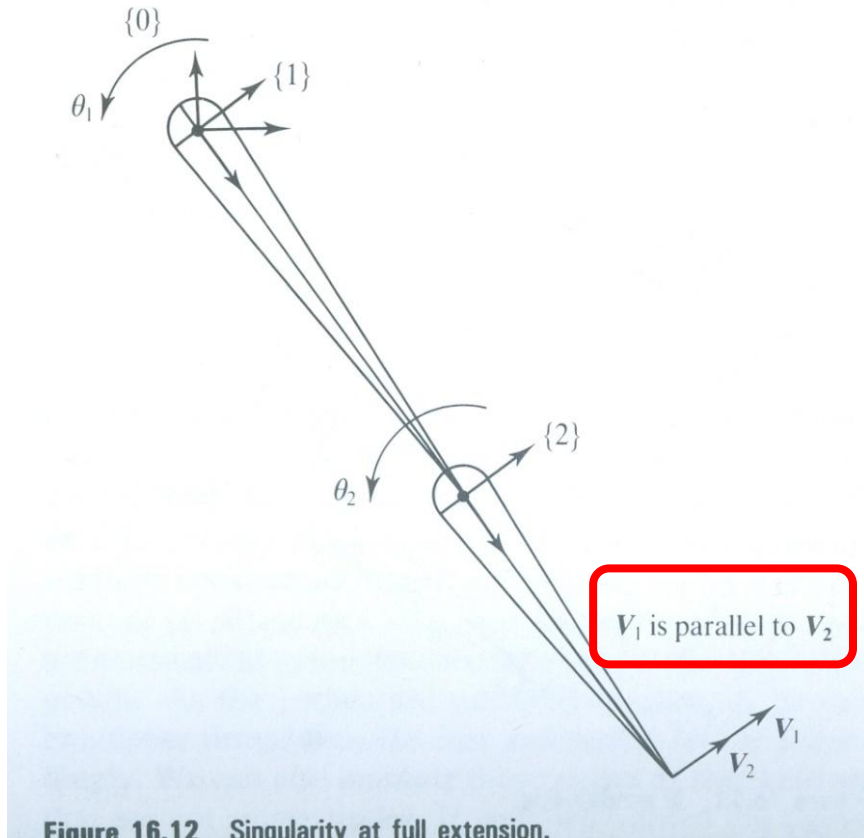
Email: [jin@cad.zju.edu.cn](mailto:jin@cad.zju.edu.cn)

浙江大学CAD&CG国家重点实验室

紫金港校区蒙民伟楼512

# 逆向雅克比方法——奇异情况的处理

- 物理上，当连杆**完全伸展**或**不同连杆的轴重合**时，会发生奇异情况：



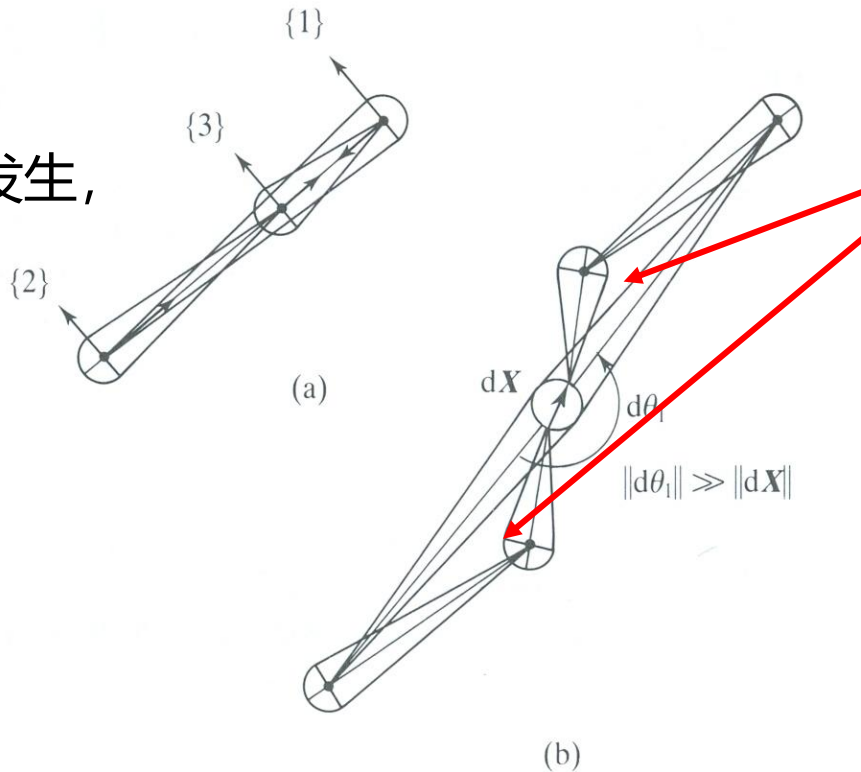
- 有两种处理方法：
  - 简单地不允许完全伸展
  - 在奇异区域引入额外的迭代操作

**雅克比矩阵的行之间线性相关!**

# 逆向雅克比方法——奇异情况的处理

- $V$  微小变化时引起  $\dot{\theta}$  大的变化:

当三个连杆重合时奇异发生,  
末端影响器放在base处



两种奇异情况

$V$  很小的变化导致  $\dot{\theta}$   
很大的变化

Figure 16.13 Ill conditioning.

# 逆向雅克比方法——奇异情况的处理

- 阻尼最小二乘法

- 引入一减少**伪逆敏感程度**的附加项 ( $\lambda$ 为用户可控的参数)

原来为:  $\dot{\theta} = J^T (JJ^T)^{-1} V$

改造为:  $\dot{\theta} = J^T (JJ^T + \lambda I)^{-1} V$

# IK中引入更多的控制

- 伪逆实际上是一种在最小二乘意义上的求解方法，它求解许多解中的一个，使得**关节角速度**的变化最小化
- 采用伪逆求解雅克比矩阵的IK 方法有可能得到不自然的结果
- 在求解逆向雅克比矩阵时，可加入**额外的控制项**，以得到更自然的结果：

$$J\dot{\theta} = V$$

**控制项**

$$\dot{\theta} = J^+V + \boxed{(J^+J - I)z}$$

# IK中引入更多的控制——控制项中加入零线速度

- 但是：控制项不应在**线速度**上增加任何变化，即引入的线速度为0

引入的变化为  $\longrightarrow \dot{\theta} = (J^+J - I)z$

代入该公式时  $\longrightarrow V = J\dot{\theta}$

引入的线速度为  $\longrightarrow V = J(J^+J - I)z$

经过简单的矩阵操作，  
可以发现...

$$\left\{ \begin{array}{l} V = (JJ^+J - J)z \\ V = (J - J)z \\ V = 0z \end{array} \right.$$

...引入**零线速度**  $\longrightarrow V = 0$

但是可以用来**偏移**关节角解向量！

# IK中引入更多的控制

- 为了朝特定的关节角偏移，例如关节限制之间的中间角， $z$  定义为

$$z_i = \alpha_i (\theta_i - \theta_{ci})^2$$

其中  $\theta_i$ : 当前关节角,

$\theta_{ci}$ : 所需的关节角,

$\alpha$ : 关节增益(Joint Gain)。

这不是硬约束(hard constraints);

# IK中引入更多的控制

- 关节增益(Joint gain)
  - 指定达到所需角的相对重要性
  - 增益越大，则关节的刚性越强
    - 增益大时: 关节角迅速逼近所需要的关节角
    - 增益为0时: 退化为传统的雅克比伪逆

# IK中引入更多的控制——如何求解系统？

$$\dot{\theta} = J^+V + (J^+J - I)z$$

$$\dot{\theta} = J^+V + J^+Jz - Iz$$

$$\dot{\theta} = J^T (JJ^T)^{-1} (V + Jz) - z$$

$$\dot{\theta} = J^T [(JJ^T)^{-1} (V + Jz)] - z$$

# 雅克比J的零空间(Null Space)

- 控制项 $\dot{\theta}$ 在J的零空间(null space):

$$\dot{\theta} = (J^+J - I)z$$

- J的零空间为对末端影响器线速度没有影响的向量集:

$$\dot{\theta} \in \text{nullspace}(J) \Leftrightarrow J\dot{\theta} = 0$$

- 零空间可用来达到第二个目的:

$$\dot{\theta} = J^+V + (J^+J - I)z$$

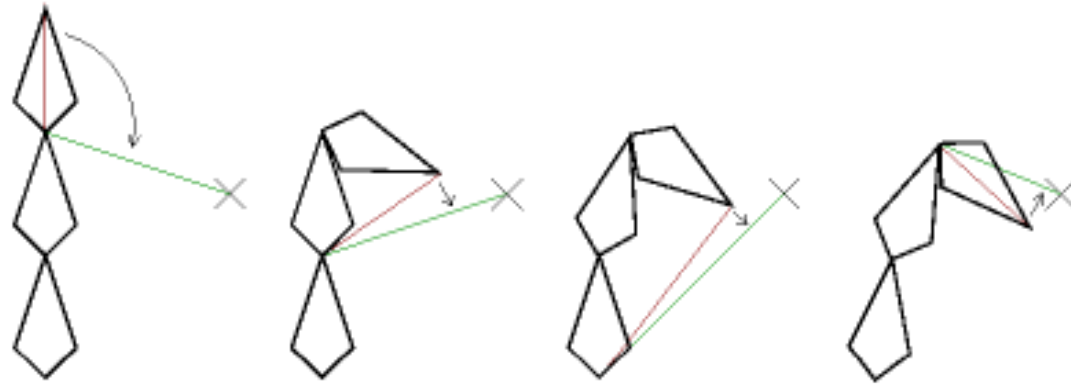
$\min_{\theta} G(\theta)$

求得**自然的**关节角姿势:

$$G(\theta) = \sum_i \alpha_i (\theta_{\text{natural}}(i) - \theta(i))^2$$

# 循环坐标下降法

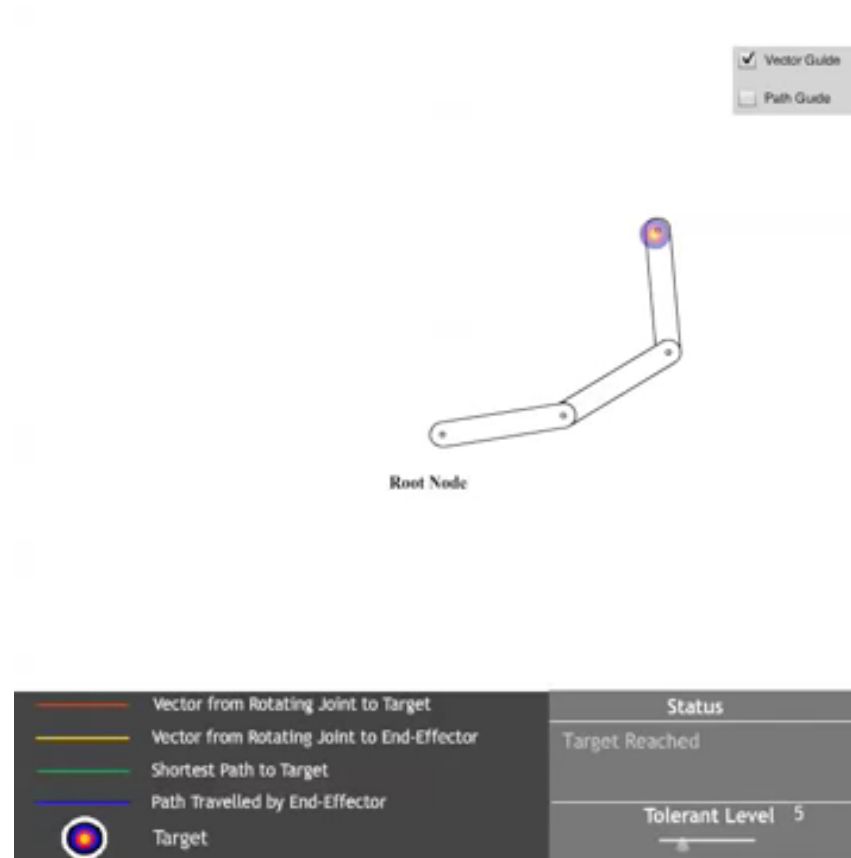
## (Cyclic Coordinate Descent (CCD))



- 对所有受IK影响的骨骼，按从最远侧子骨骼到父骨骼的顺序执行迭代操作：旋转当前骨骼，使当前骨骼位置到目标骨骼的连线指向IK目标位置。
- 由于所有骨骼是从一个特定状态出发开始IK计算，所得到的结果也会比较稳定。通常5 ~ 10次迭代之后就能得到很好的结果。

Wang and Chen. "A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators." IEEE Transactions on Robotics and Automation. Vol. 7, No. 4, August 1991, pp. 489-499.

# 循环坐标下降法 (Cyclic Coordinate Descent (CCD))

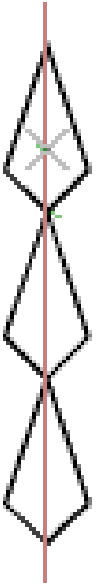


<https://github.com/Germanunkol/CCD-IK-Panda3D>

# 循环坐标下降法

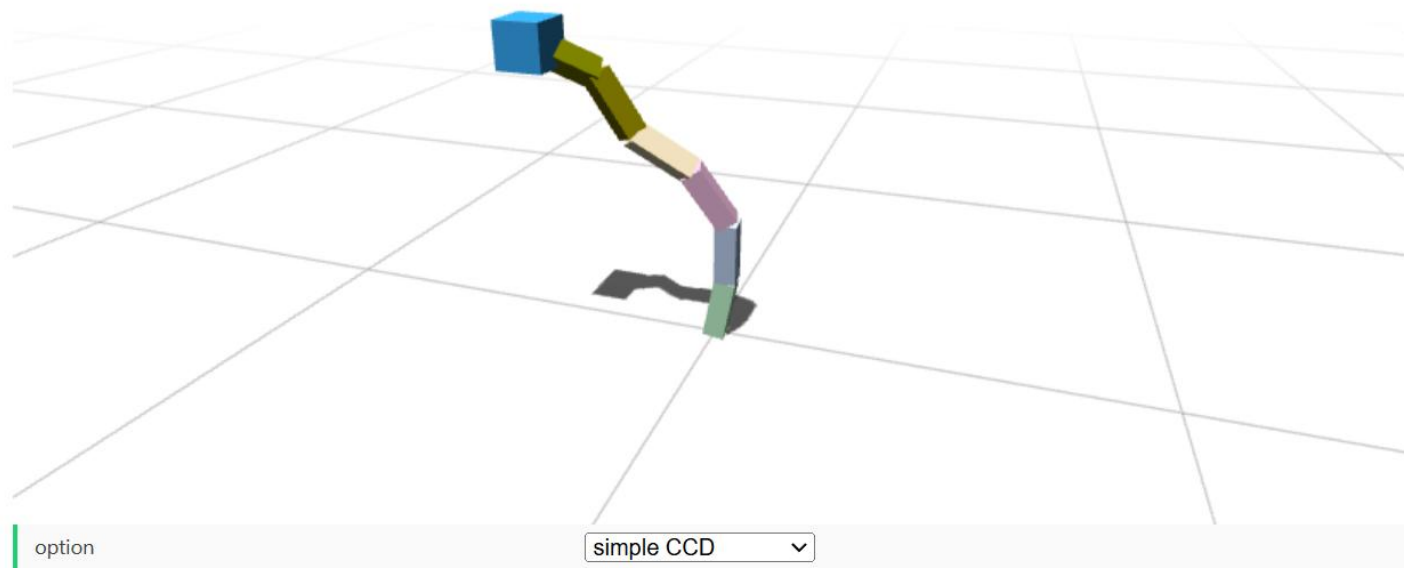
## (Cyclic Coordinate Descent (CCD))

- 如果是人体骨骼，不是所有的关节都可以向任意方向旋转，所以我们必须对骨骼的旋转加以约束，比如肘关节实际上只有一个轴的自由度，而且不能向后弯曲。
- **奇异情况：**当所有需要IK控制的骨骼正好在一条直线上，而IK目标位置正好在也落在这条直线上时（如下图），算法就会失败，因为不论迭代多少次，每一个骨骼都会认为自己不需要旋转。
- **解决技巧：**如果发现骨骼链“很直”，就向骨骼允许的任意方向加一些细微的旋转；或者干脆在骨骼的限制角度数据中就禁止完全“伸直”。



# 循环坐标下降法的实现

<http://mobile.rodolphe-vaillant.fr/entry/114/cyclic-coordonate-descent-inverse-kynematic-ccd-ik>



# CCD方法的优缺点

- **优点:**

- **简单:** 实现和理解相对简单
- **实时:** 计算成本较低, 因此适合实时应用

- **缺点:**

- **会陷入局部极小:** CCD有时会陷入局部极小值, 无法找到复杂结构的全局最优解
- **不真实的姿势:** production of unrealistic poses

# 正向和反向到达IK法(FABRIK)

(Forward And Backward Reaching Inverse Kinematics)

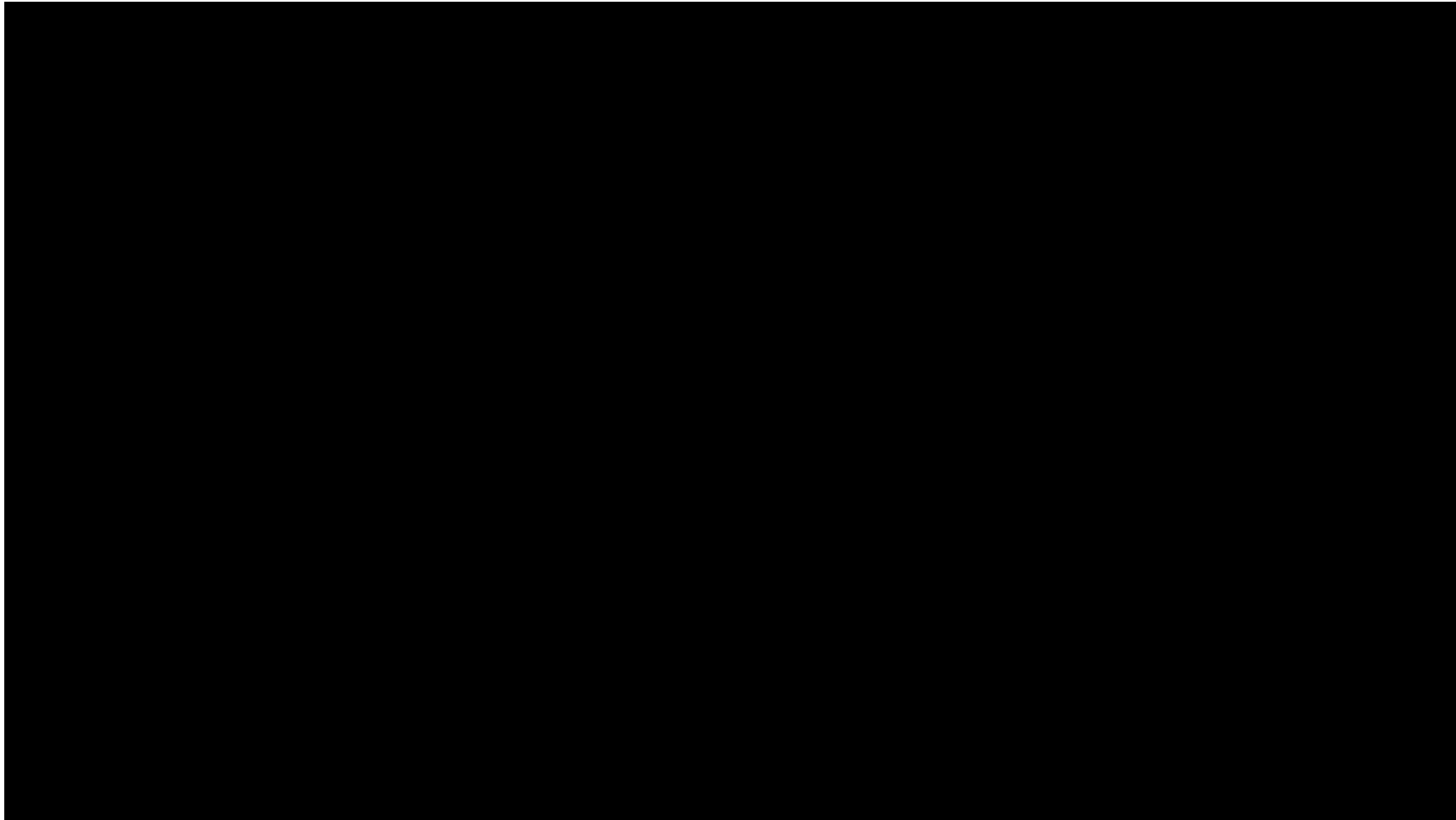
- **思想：**是一种**启发式迭代IK**求解方法。它通过之前的关节位置，以**向前迭代**和**向后迭代**的方式更新关节位置，从而使得末端影响器达到目标位置。
- 可**避免使用旋转角度或矩阵**，而是通过定位**线上的点**来找到每个关节位置。
- FABRIK 已经被整合到**Unreal Engine**, **Unity3D**和**Panda**(由迪士尼开发的 3D 开源游戏引擎)等游戏引擎中。
- 论文主页: <http://www.andreasaristidou.com/FABRIK.html>
- 开源实现: <https://github.com/henriksod/Fabrik2DArduino>
- <https://zhuatlan.zhihu.com/p/361520144>

Aristidou A, Lasenby J. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models*, 2011, 73(5):243-60.

# IK Solver (FABRIK) for Unity

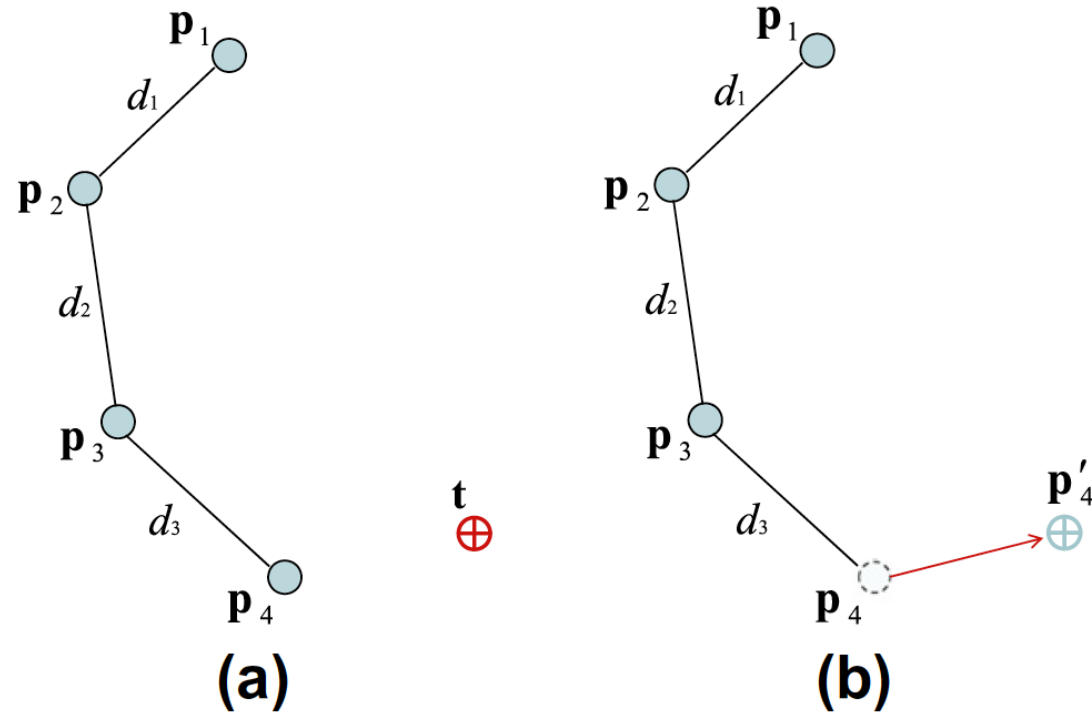
<https://github.com/joaen/EasyIK>

---



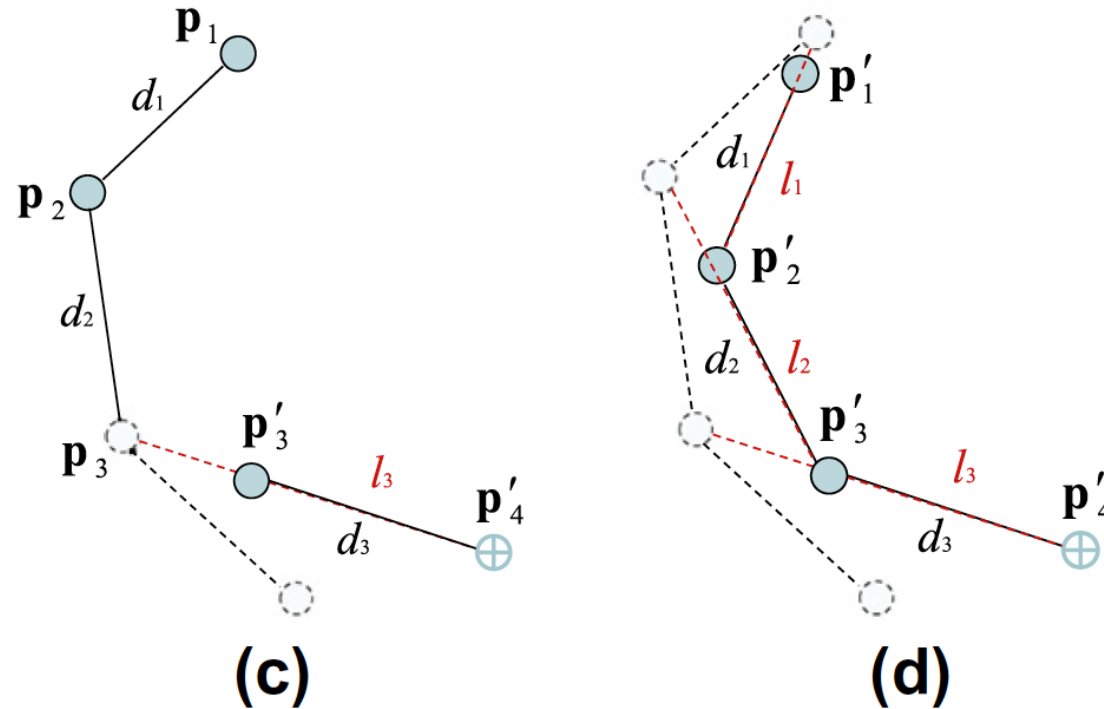
# An example of a full iteration of FABRIK for the case of a single target and 4 manipulator joints

原理



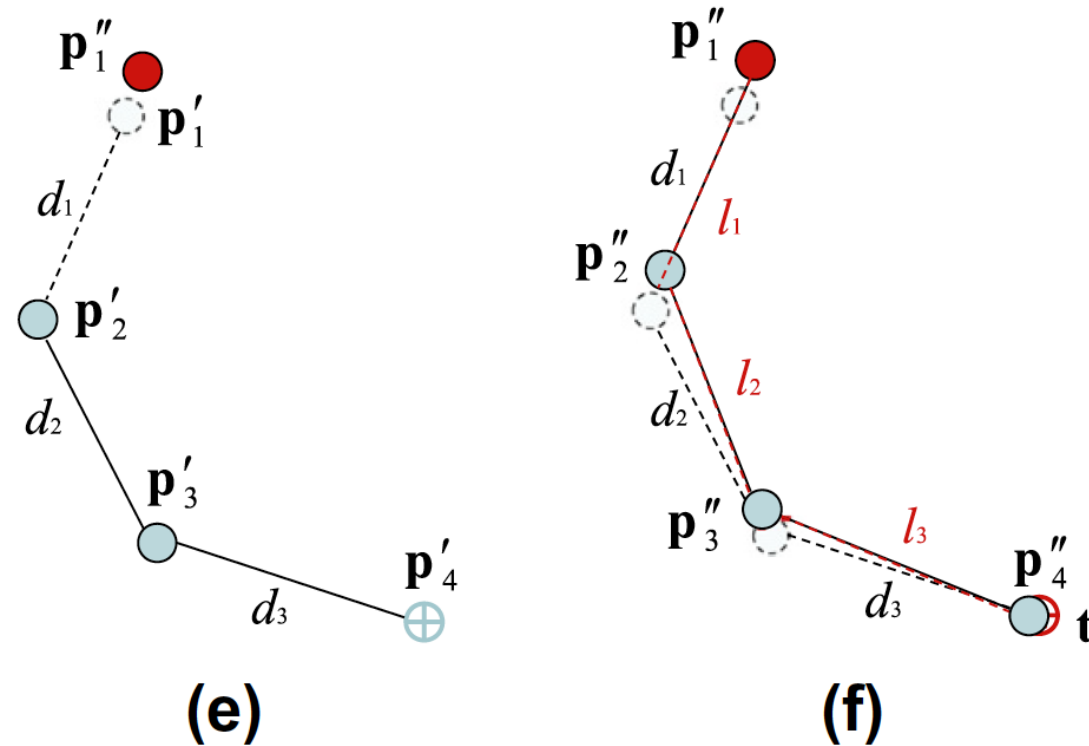
(a) The initial position of the manipulator and the target, (b) move the end effector  $p_4$  to the target,

# An example of a full iteration of FABRIK for the case of a single target and 4 manipulator joints



(c) find the joint  $\mathbf{p}'_3$  which lies on the line  $l_3$  that passes through the points  $\mathbf{p}'_4$  and  $\mathbf{p}_3$ , and has distance  $d_3$  from the joint  $\mathbf{p}'_4$ , (d) continue the algorithm for the rest of the joints,

# An example of a full iteration of FABRIK for the case of a single target and 4 manipulator joints

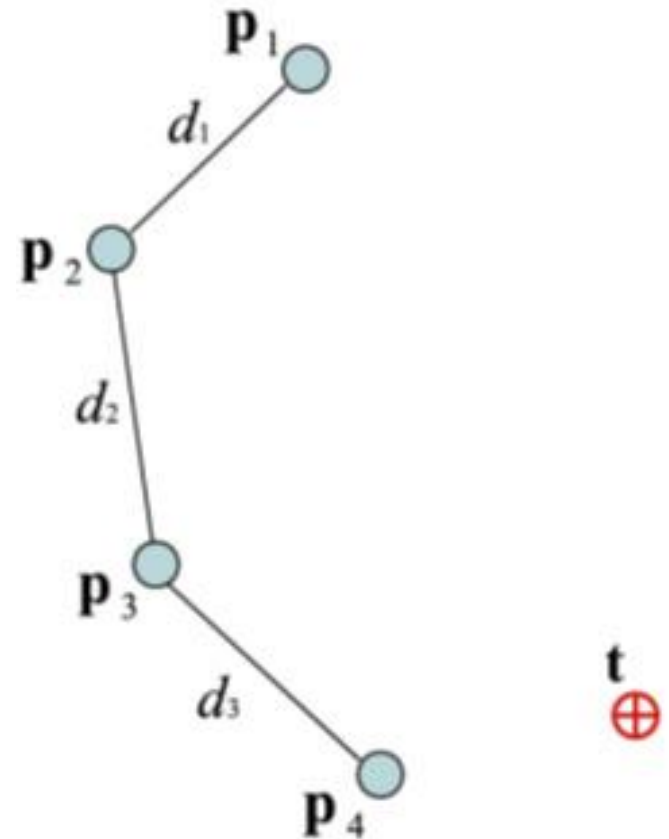


(e) the second stage of the algorithm: move the root joint  $p_1'$  to its initial position, (f) repeat the same procedure but this time start from the base and move outwards to the end effector.

**The algorithm is repeated until the position of the end effector reaches the target or gets sufficiently close.**

# FABRIK: 算法

- **输入:** 关节位置  $\mathbf{p}_i (i = 1, \dots, n)$ , 目标位置  $\mathbf{t}$ ,  
两个关节之间的距离  $d_i = |\mathbf{p}_{i+1} - \mathbf{p}_i| (i = 1, \dots, n-1)$
- if  $|\mathbf{p}_1 - \mathbf{t}| > (d_1 + d_2 + \dots + d_{n-1})$  then // **目标无法达到**  
计算向量  $(\mathbf{t} - \mathbf{p}_1)$ , 找到沿**该向量**的每个关节的相对位置(  
想象一下沿着目标方向延伸整个关节链条)
- else // **目标是可以达到的**
  - 将  $\mathbf{b} = \mathbf{p}_1$  设置为初始 (根) 位置 (注意: 因为  $\mathbf{p}_1$  的位置会被更新)
  - 计算末端影器  $\mathbf{p}_n$  距目标位置  $\mathbf{t}$  的距离:  $\text{diff} = |\mathbf{p}_n - \mathbf{t}|$
  - While ( $\text{diff} > \text{tolerance}$ )
    - 执行向前和向后传递 (Forward and Backward pass)



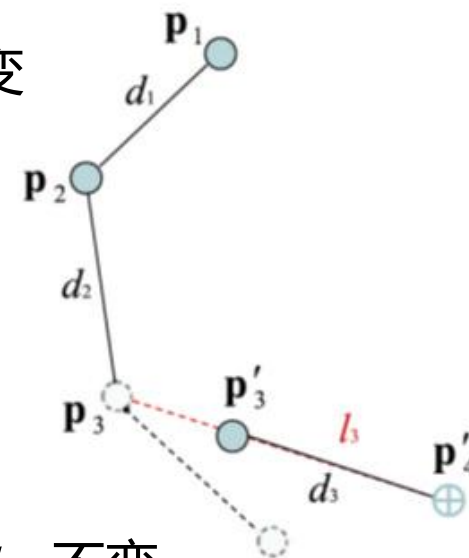
# FABRIK: Forward and Backward pass

- 把末端影响器移到位置 $t$ , 即 $\mathbf{p}_n = t$
- 对于前面的每一关节点 $\mathbf{p}_i$ ,  $i = n - 1$  to  $1$ : (Forward Pass)  
计算向量 $(\mathbf{p}_{i+1} - \mathbf{p}_i)$ , 并更新关节  $i$  沿该向量的新位置 $\mathbf{p}'_i$ , 并保持相对距离 $d_i$ 不变

- 到目前为止, 我们的根节点会稍微偏离其初始位置 $\mathbf{b}$ , 设置 $\mathbf{p}_1 = \mathbf{b}$
- 对于每个关节点 $\mathbf{p}_i$ ,  $i = 1$  to  $n - 1$ : (Backward Pass)

计算向量 $(\mathbf{p}_i - \mathbf{p}_{i+1})$ , 并更新关节  $i+1$  沿该向量的新位置 $\mathbf{p}'_{i+1}$ , 并保持相对距离 $d_{i+1}$ 不变

- 更新 $\text{diff} = |\mathbf{p}_n - t|$



# FABRIK伪代码

**Algorithm 1.** A full iteration of the FABRIK algorithm

---

**Input:** The joint positions  $\mathbf{p}_i$  for  $i = 1, \dots, n$ , the target position  $\mathbf{t}$  and the distances between each joint

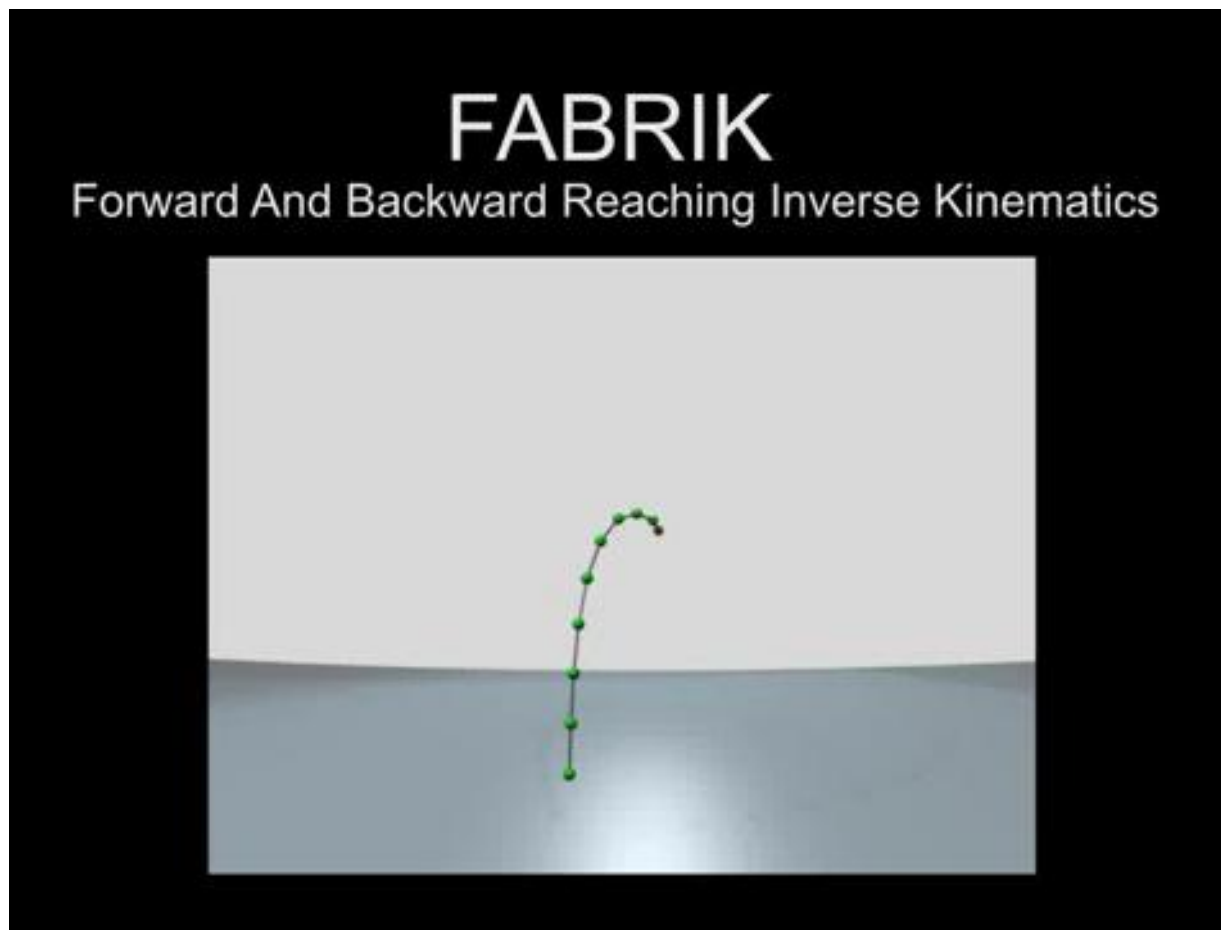
$$d_i = |\mathbf{p}_{i+1} - \mathbf{p}_i| \text{ for } i = 1, \dots, n - 1.$$

**Output:** The new joint positions  $\mathbf{p}_i$  for  $i = 1, \dots, n$ .

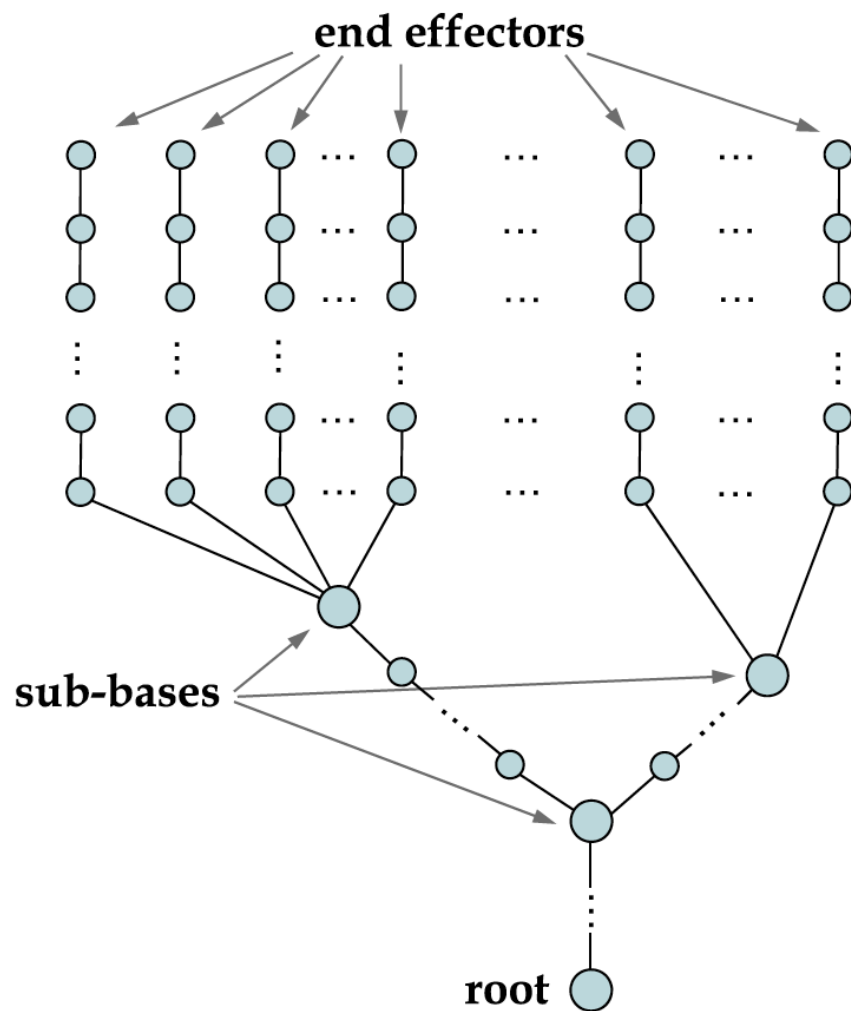
```
1.1 % The distance between root and target
1.2  $dist = |\mathbf{p}_1 - \mathbf{t}|$ 
1.3 % Check whether the target is within reach
1.4 if  $dist > d_1 + d_2 + \dots + d_{n-1}$  then
1.5     % The target is unreachable
1.6     for  $i = 1, \dots, n - 1$  do
1.7         % Find the distance  $r_i$  between the target  $\mathbf{t}$  and
the joint
position  $\mathbf{p}_i$ 
1.8          $r_i = |\mathbf{t} - \mathbf{p}_i|$ 
1.9          $\lambda_i = d_i / r_i$ 
```

```
1.10     % Find the new joint positions  $\mathbf{p}_i$ .
1.11      $\mathbf{p}_{i+1} = (1 - \lambda_i) \mathbf{p}_i + \lambda_i \mathbf{t}$ 
1.12 end
1.13 else
1.14     % The target is reachable; thus, set as  $\mathbf{b}$  the
initial position of the
joint  $\mathbf{p}_1$ 
1.15      $\mathbf{b} = \mathbf{p}_1$ 
1.16     % Check whether the distance between the end
effector  $\mathbf{p}_n$ 
and the target  $\mathbf{t}$  is greater than a tolerance.
1.17      $dif_A = |\mathbf{p}_n - \mathbf{t}|$ 
1.18     while  $dif_A > tol$  do
1.19         % STAGE 1: FORWARD REACHING
1.20         % Set the end effector  $\mathbf{p}_n$  as target  $\mathbf{t}$ 
1.21          $\mathbf{p}_n = \mathbf{t}$ 
1.22         for  $i = n - 1, \dots, 1$  do
1.23             % Find the distance  $r_i$  between the new joint
position
 $\mathbf{p}_{i+1}$  and the joint  $\mathbf{p}_i$ 
1.24              $r_i = |\mathbf{p}_{i+1} - \mathbf{p}_i|$ 
1.25              $\lambda_i = d_i / r_i$ 
1.26             % Find the new joint positions  $\mathbf{p}_i$ .
1.27              $\mathbf{p}_i = (1 - \lambda_i) \mathbf{p}_{i+1} + \lambda_i \mathbf{p}_i$ 
1.28         end
1.29         % STAGE 2: BACKWARD REACHING
1.30         % Set the root  $\mathbf{p}_1$  its initial position.
1.31          $\mathbf{p}_1 = \mathbf{b}$ 
1.32         for  $i = 1, \dots, n - 1$  do
1.33             % Find the distance  $r_i$  between the new joint
position  $\mathbf{p}_i$ 
and the joint  $\mathbf{p}_{i+1}$ 
1.34              $r_i = |\mathbf{p}_{i+1} - \mathbf{p}_i|$ 
1.35              $\lambda_i = d_i / r_i$ 
1.36             % Find the new joint positions  $\mathbf{p}_i$ .
1.37              $\mathbf{p}_{i+1} = (1 - \lambda_i) \mathbf{p}_i + \lambda_i \mathbf{p}_{i+1}$ 
1.38         end
1.39          $dif_A = |\mathbf{p}_n - \mathbf{t}|$ 
1.40     end
1.41 end
```

# FABRIK法——演示



# 多个末端影响器(multiple end effectors)



# FABRIK方法的优缺点

- **优点:**

- **高效:** 可用更少的处理时间和更少的迭代来达到目标
- **简单:** 它使得对任何逆向运动学问题的配置变得简单，能处理各种约束条件（如关节角度限制），复杂的链式结构
- **全局优化:** FABRIK倾向于通过**全链迭代**调整，提供更好的全局解决方案

- **缺点:**

- **依赖于初始姿态:** FABRIK的求解结果可能依赖于初始姿态。不同的初始姿态可能导致不同的解决方案，这在某些情况下可能不是理想的。
- **奇异情况:** 当所有骨骼在一条直线上，而IK目标位置正好在落在这条直线的两个关节之间时，算法会失败。**解决方法:** 允许关节链在用户限制范围内弯曲一个非常小的角度



# 基于优化的IK计算方法

- 把IK转化为一非线性优化问题

- 例如

$$\text{minimize } x^2(y + 1) + \sin(x + y)$$

$$\text{subject to } x \geq 0, y \geq 0$$

- 目标函数(Objective function)

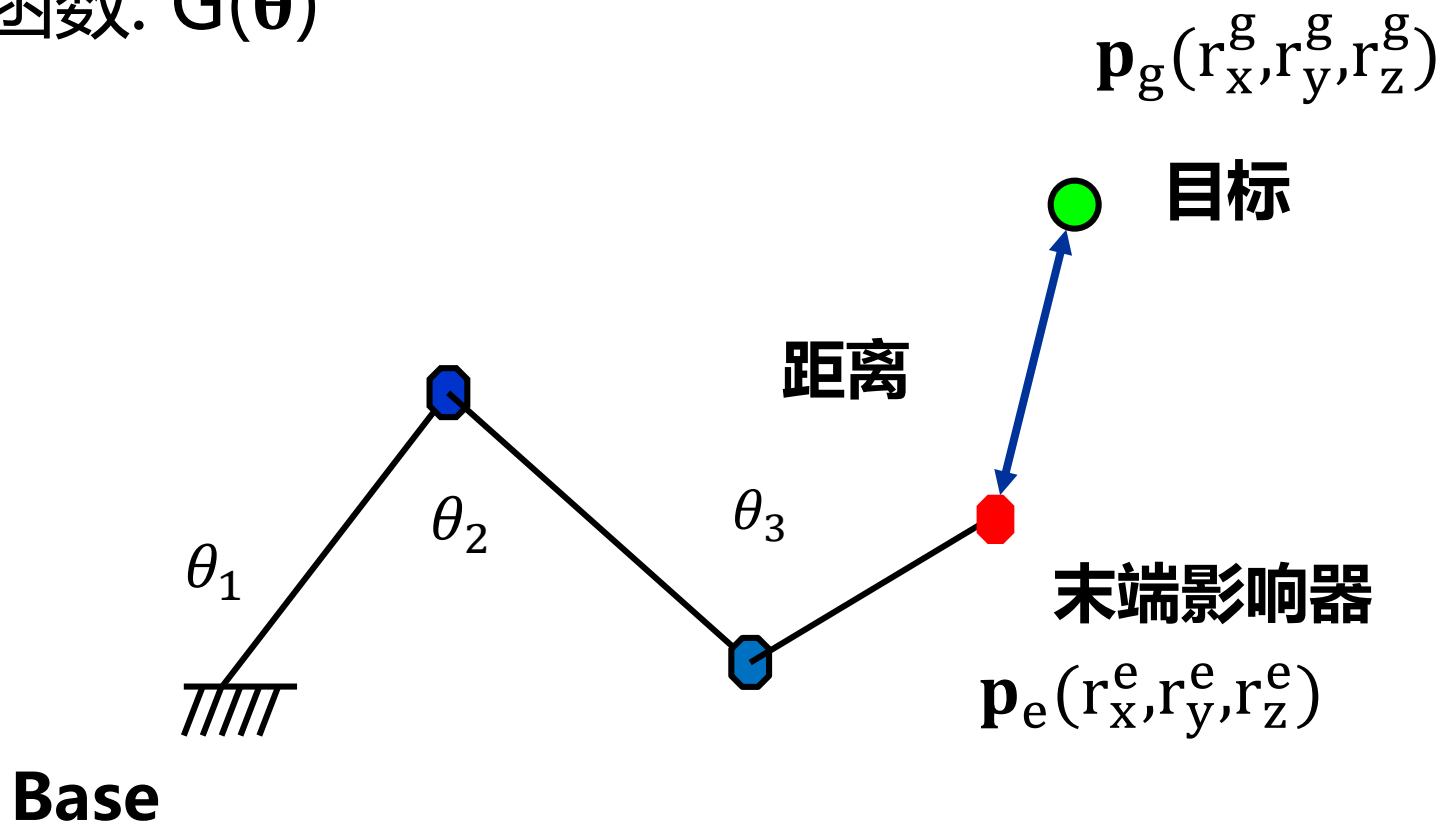
- 约束(Constraint)

- 迭代算法(Iterative algorithm)

Ref: Jianmin Zhao, Norman I. Badler, “Inverse kinematics positioning using nonlinear programming for highly articulated figures”, ACM Transactions on Graphics, 1994, 13(4)

# 目标函数(Objective Function)

- 末端影响器到目标位置/方向的“距离”
- 关节角的函数:  $G(\theta)$



# 目标函数

位置目标函数      方向目标函数(方向由**一对**正交向量来定义)

$$\|\mathbf{p}_g - \mathbf{p}_e\|^2$$

$$\|\mathbf{r}_x^g - \mathbf{r}_x^e\|^2 + \|\mathbf{r}_y^g - \mathbf{r}_y^e\|^2$$

**总目标函数为位置/方向目标的加权和:**

$$G(\theta) = \omega \|\mathbf{p}_g - \mathbf{p}_e\|^2 + (1 - \omega) (\|\mathbf{r}_x^g - \mathbf{r}_x^e\|^2 + \|\mathbf{r}_y^g - \mathbf{r}_y^e\|^2)$$

# 非线性优化

- 约束的非线性优化问题：

$$\begin{cases} \text{minimize} & G(\boldsymbol{\theta}) \\ \text{subject to} & \begin{cases} \mathbf{a}^T \boldsymbol{\theta} = \mathbf{b}_1 & \text{limb coordination} \\ \mathbf{a}^T \boldsymbol{\theta} \leq \mathbf{b}_2 & \text{joint limits} \end{cases} \end{cases}$$

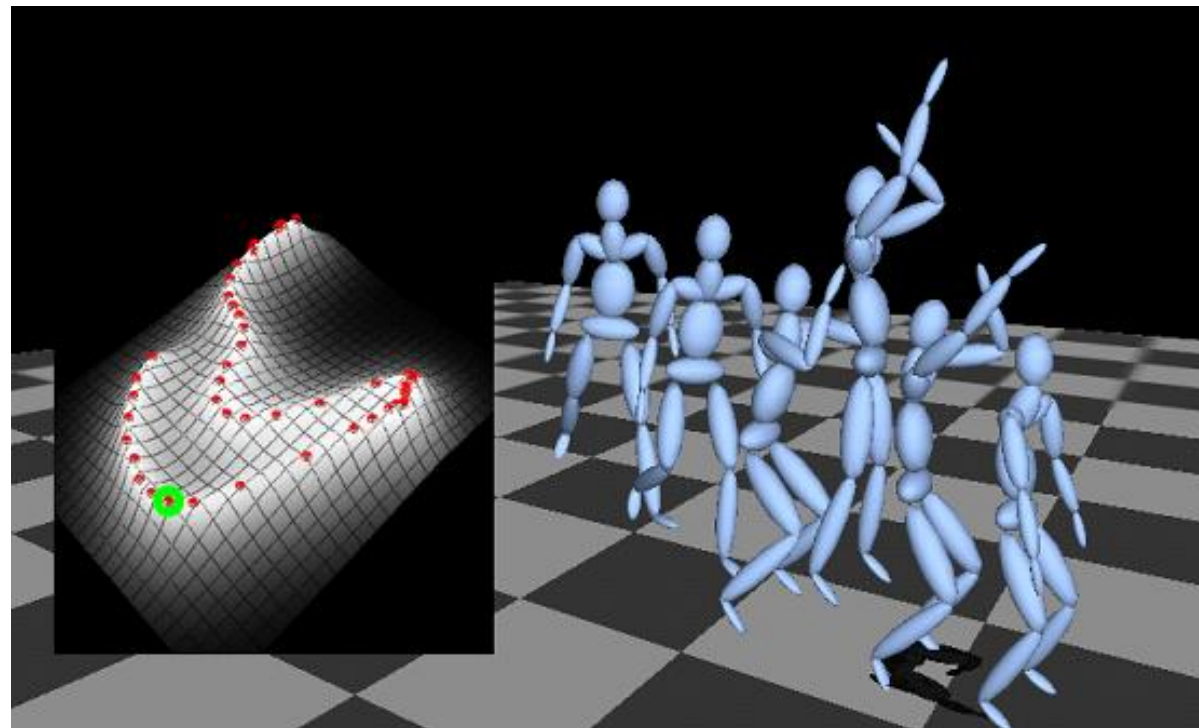
- 求解
  - 标准的数值方法
  - MATLAB或其它优化软件包
    - 求得的通常是局部极小
    - 依赖于初始条件

# 基于样例的方法(Example-based Method)

- 基于构造的统计模型的IK

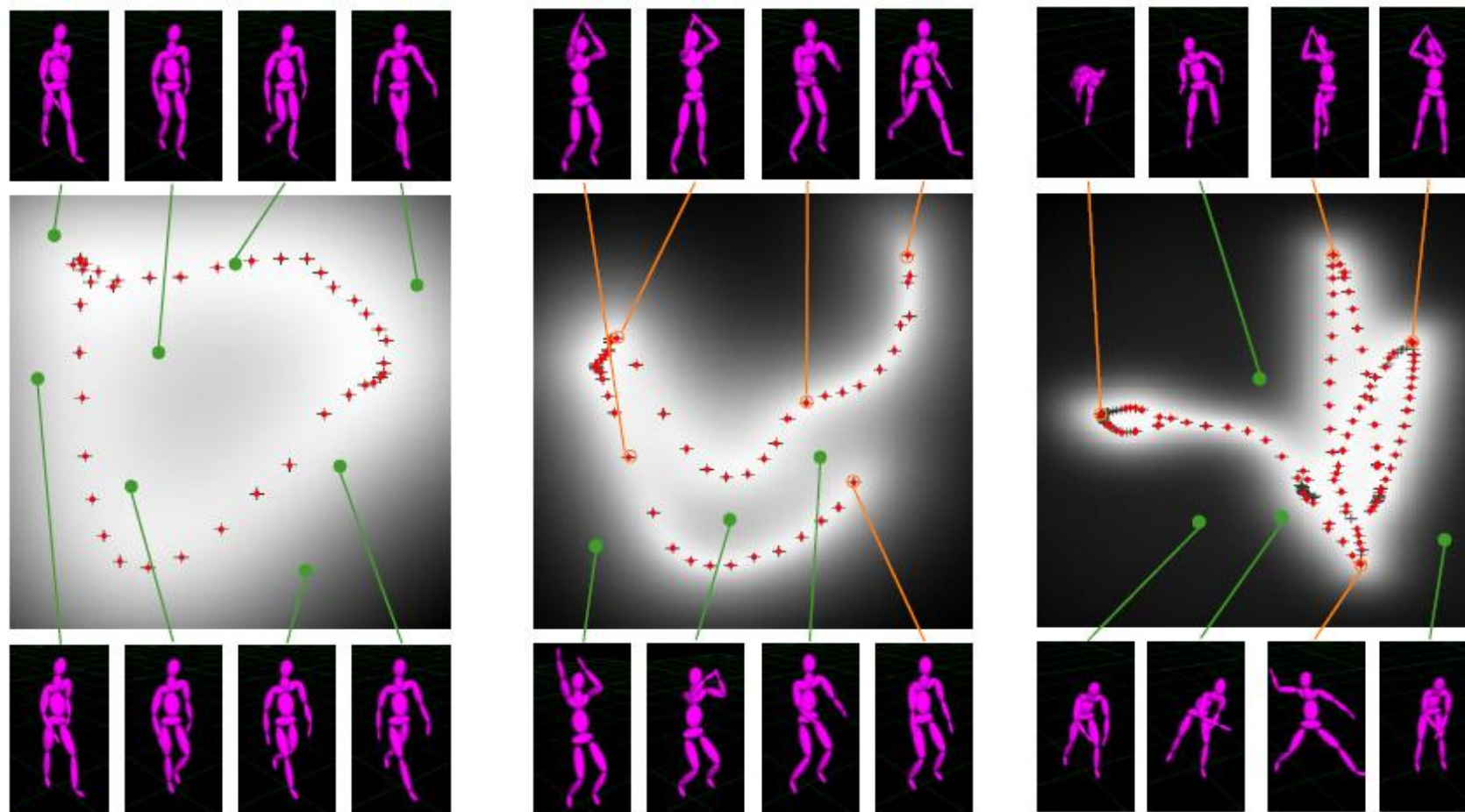
Grochow et al., "Style-based inverse kinematics," SIGGRAPH' 04

- 提供了基于给定约束的最优可能的姿态



# 基于样例的方法(cont.)

- 构造姿态空间 (**training**, extrapolated)



# Videos

## **Style-Based Inverse Kinematics**

**Keith Grochow   Steven L. Martin  
Aaron Hertzmann   Zoran Popovic**

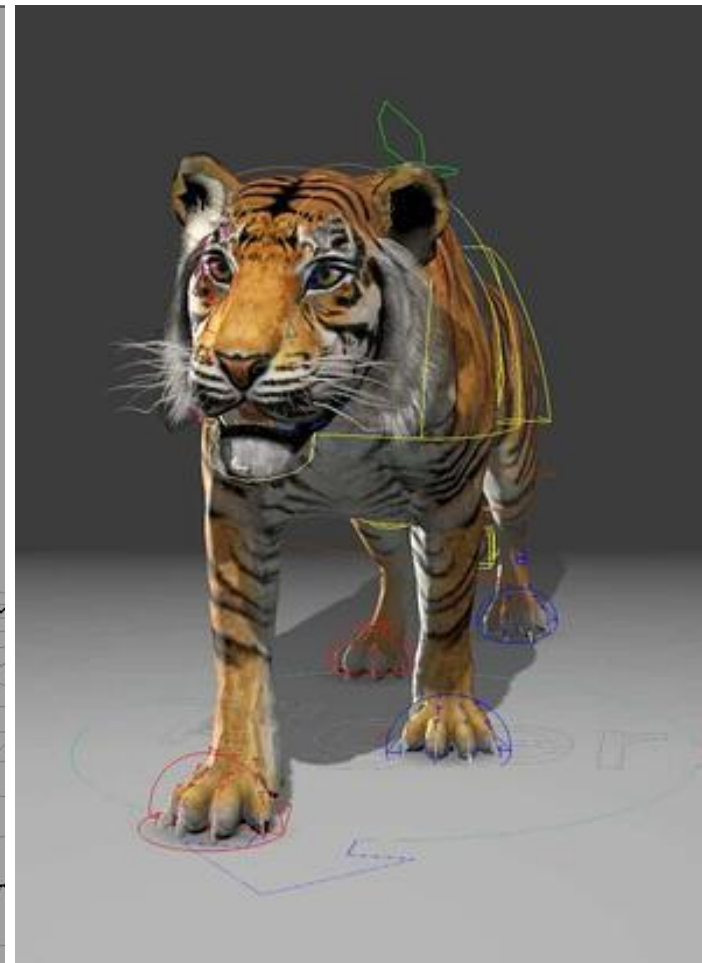
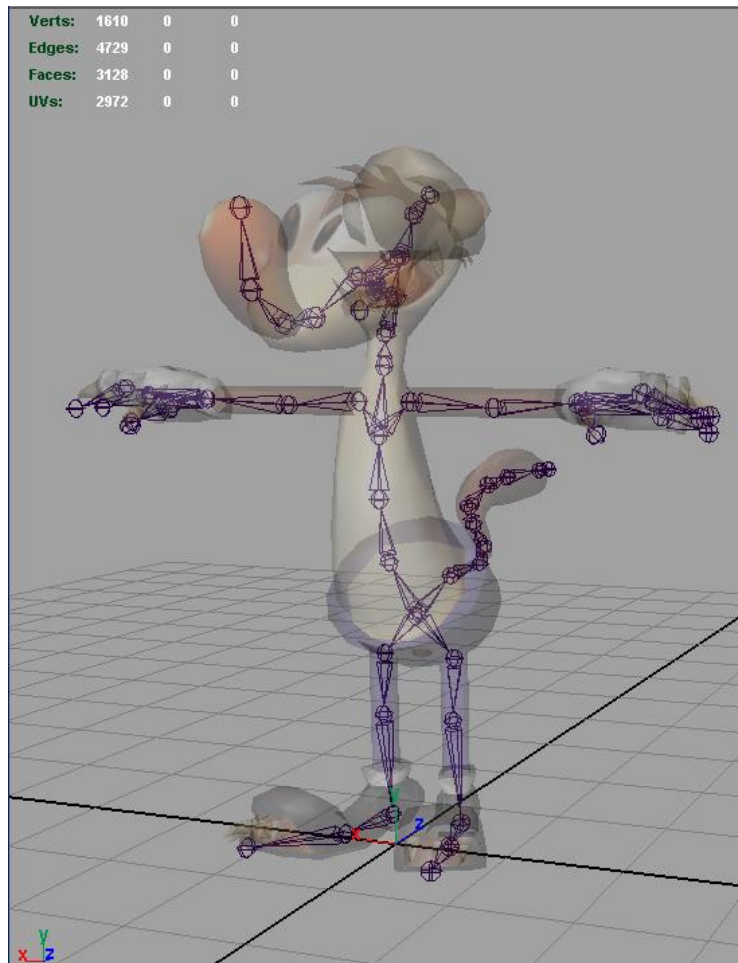
# Neural Inverse Kinematics

- Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, Cewu Lu: HybrIK: A Hybrid Analytical-Neural Inverse Kinematics Solution for 3D Human Pose and Shape Estimation. CVPR 2021: 3383-3393
- Jiefeng Li, Siyuan Bian, Qi Liu, Jiasheng Tang, Fan Wang, Cewu Lu, NIKI: Neural Inverse Kinematics with Invertible Neural Networks for 3D Human Pose and Shape Estimation. In CVPR 2023.
- Raphael Bensadoun, Shir Gur, Nitsan Blau, Tom Shenkar, Lior Wolf: Neural Inverse Kinematics. ICML 2023.



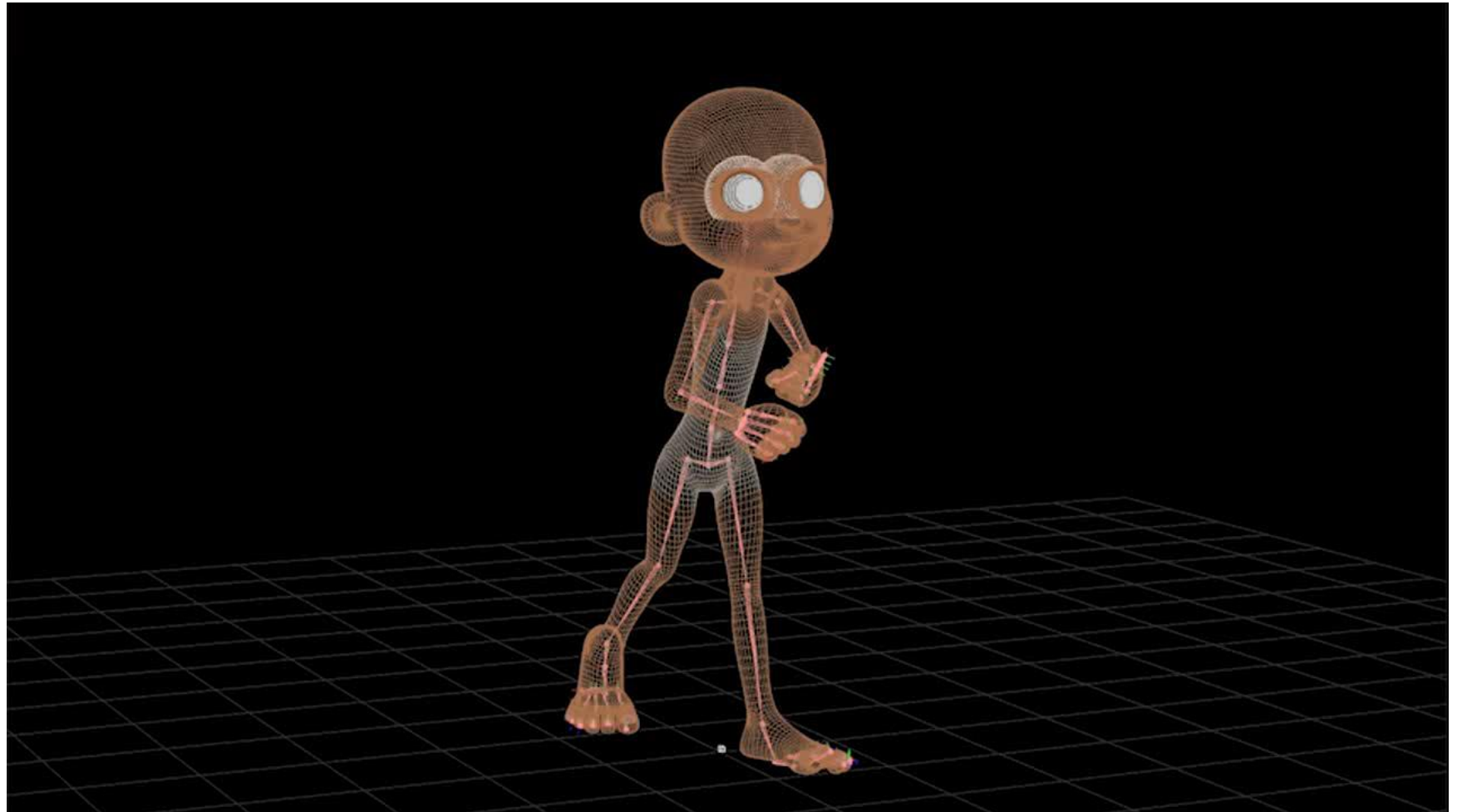
# 骨架与角色模型的绑定

- 到目前为止，我们讨论了骨架的运动
- 角色的表面(外皮)必须随着骨架的运动而运动(变形)
- 这个过程称为 **Rigging** (Skinning, Enveloping, **Vertex Blending**, **SSD** (Skeleton Subspace Deformation))



# 骨架与角色模型的绑定

The Science Behind  
**PIXAR**

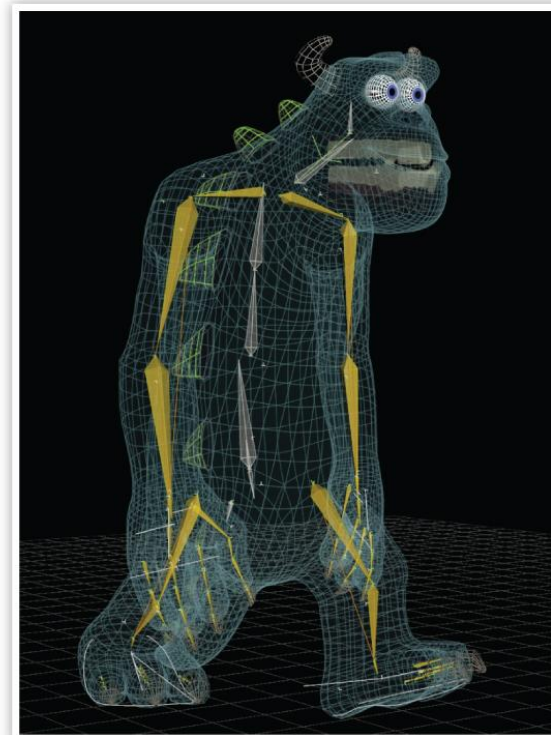
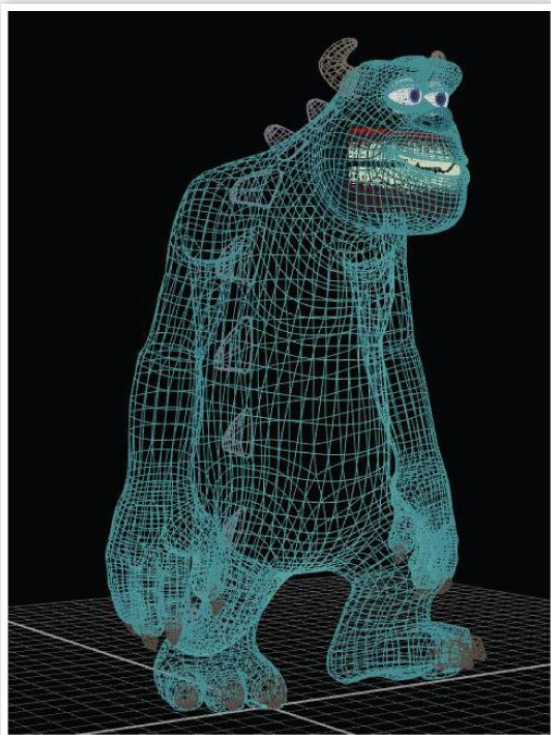


# What is a rig?

- 在计算机动画中，数字角色通常是动的。如何让角色动起来？那需要用到一种称为**Rigging**，中文称为角色绑定的技术。
- 什么叫Rig？我们看一下皮克斯动画工作室的定义。Rig的中文意思是**索具**。数字**索具**(Digital Rig)是允许数字模型移动的虚拟骨骼、关节和肌肉。这有点像牵线木偶上的绳子。**一个好的索具具有恰到好处的灵活性**。如果没有正确的控制，动画师就无法创建他们需要的姿势。但是，太多的灵活性又会使模型摆姿势太耗时。
- 索具在游戏和电影的动画角色中最为常见。这种技术简化了动画过程并提高了生产效率。一旦与骨骼绑定，任何三维对象都可以根据需要进行控制和形变。在数字娱乐行业，骨骼绑定几乎是对角色设置动画的标准方式，也是极其为重要的一步。实现流畅和复杂的动画完全取决于动画流水线中索具阶段的质量。

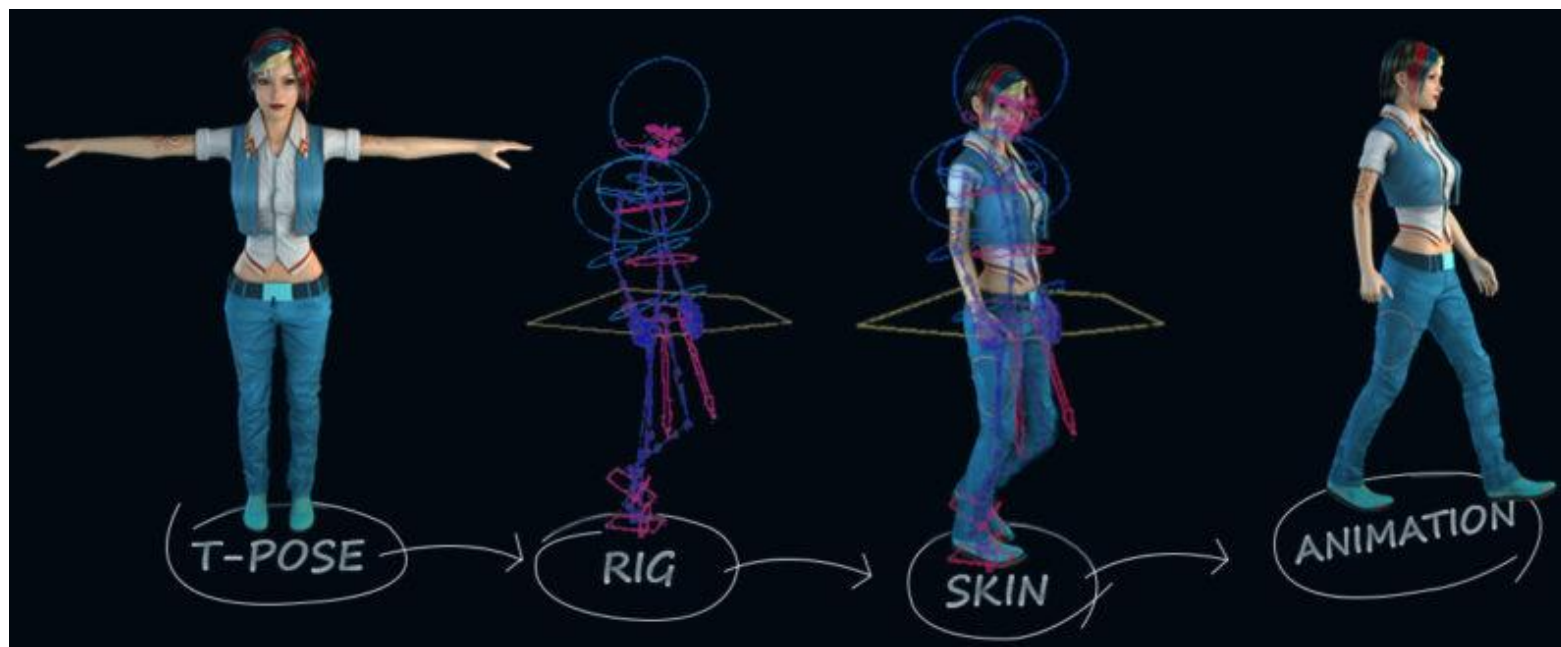
# What does a rigger do?

Rigger, 也就是角色绑定师。角色绑定师从角色的虚拟三维模型开始。他们根据故事研究角色需要如何移动。

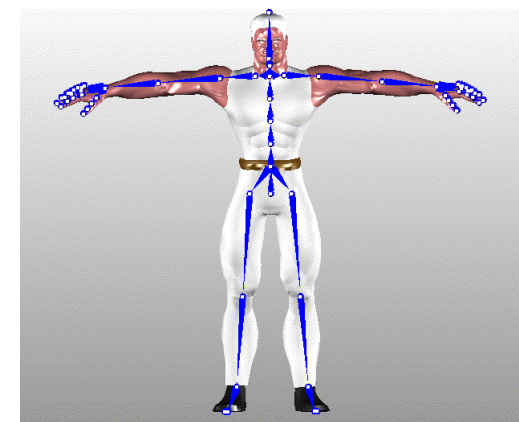


Riggers start with a wireframe model (on the left) and add the virtual bones (on the right) so Sulley from Monsters, Inc. can be posed.

# 中性姿势或静止姿势



- 在骨架绑定中，皮肤的运动定义为对应控制骨架的函数
- 很多骨架绑定系统采用一个称为中性姿势或**静止姿势**(T-Pose、Rest pose)的几何信息



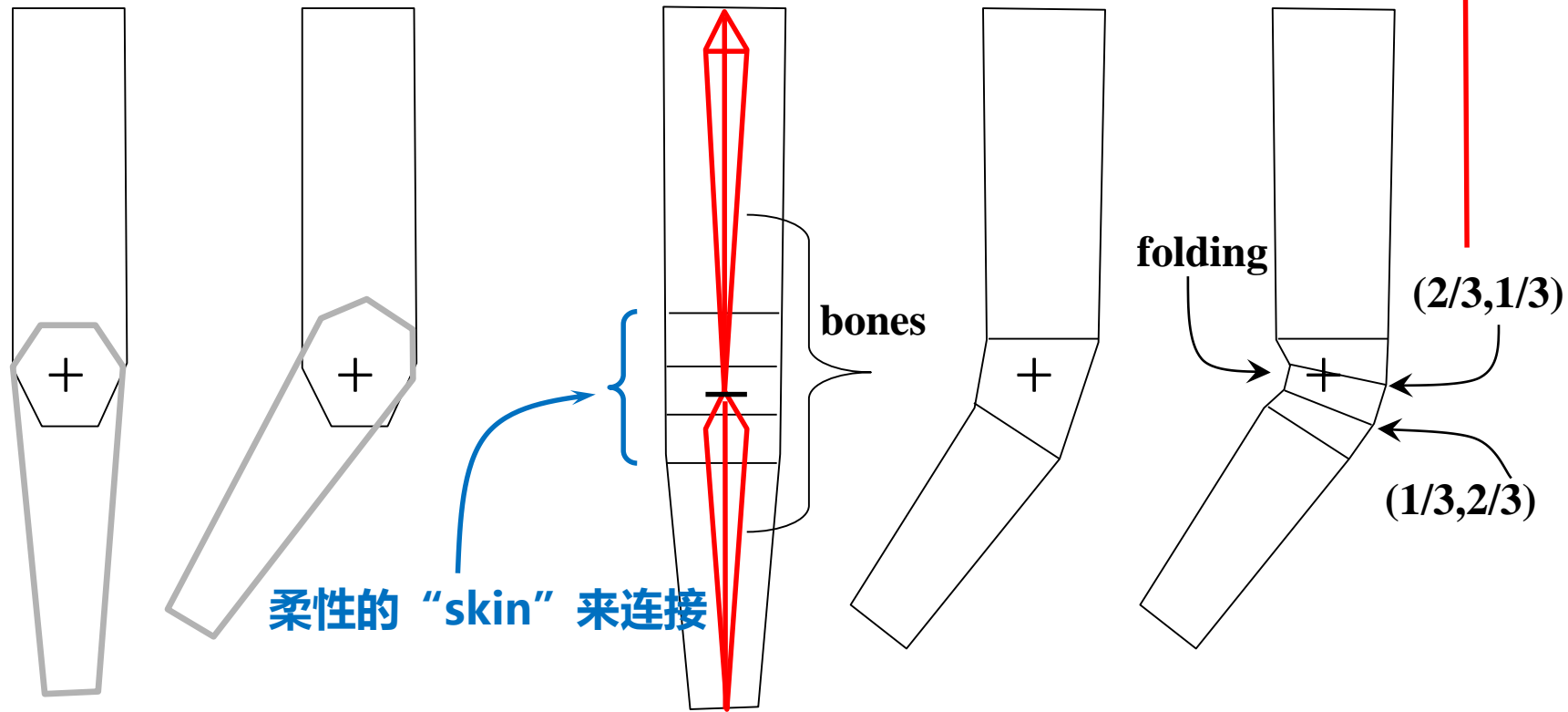
# 问题的引出

- 但是，对于某些点，我们不知道它们属于身体的哪一段？
- 对于肘关节附近的点，可能属于前臂、后臂、或者两者
- 我们希望前臂和后臂都影响这些点的运动！

# 顶点混合(Vertex Blending)

- 假设一个数字角色的手臂用前臂和后臂来模拟，我们要对手臂设置动画。
- 若采用刚体变换设置动画，则肘关节处不像真实手臂。因为我们采用了两个分离的物体来模拟手臂，在肘关节处有重叠部分。
- 解决方法：Vertex Blending!
- 采用一个物体模拟，关节处应是柔性的。

前臂2/3, 后臂1/3

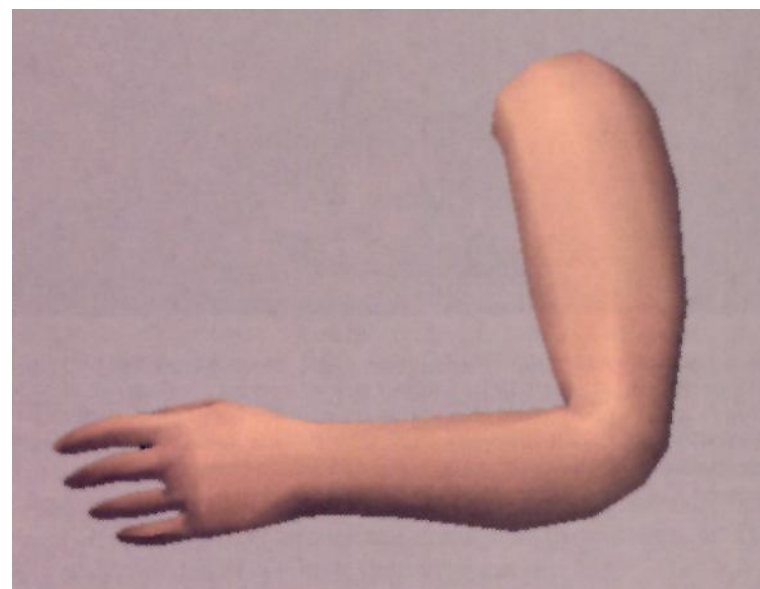
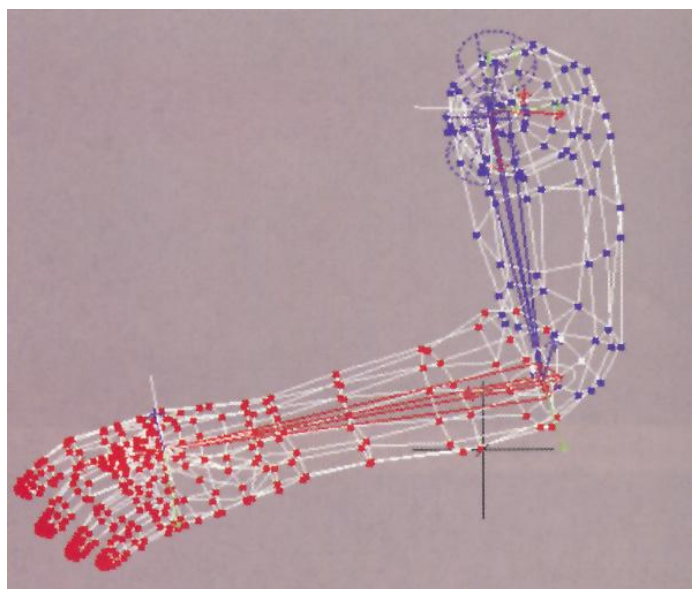


# 顶点混合(Vertex Blending)

- **最简单方法**：前臂和后臂仍单独设置动画，但关节处用一柔性的“skin”来连接。
- 柔性部分的一部分顶点由前臂的矩阵来变换，另一部分由后臂的矩阵来变换。即：**一个三角形**的顶点可以由**不同的矩阵**来变换，**而不是一个矩阵**。这种基本技术有时也称为“Stitching”。
- 进一步推广：一个顶点可以由**几个不同的矩阵进行加权变换**
- **实施方法**：在物体上放置关节骨架，每个骨架按用户给定的权因子影响顶点。

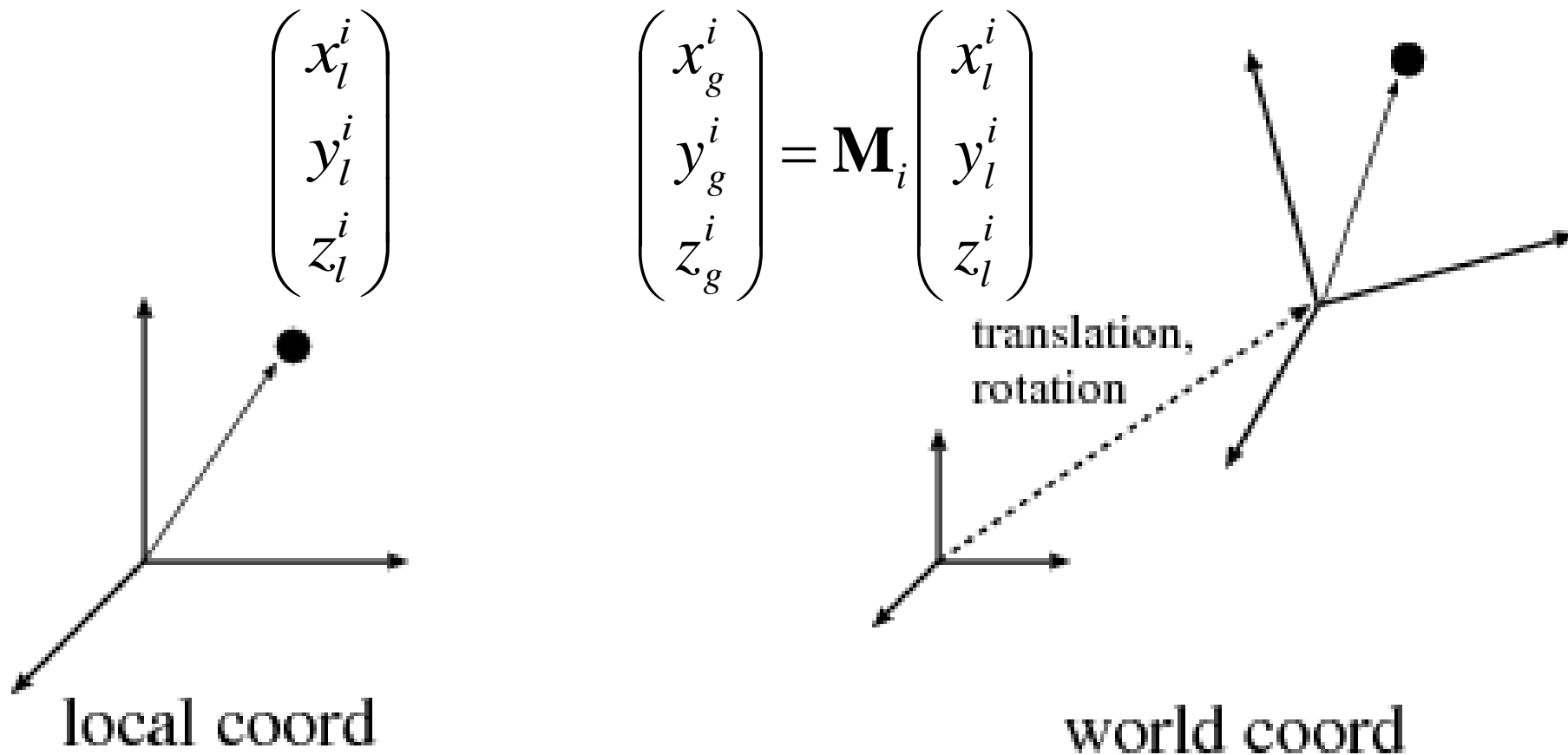
# 顶点混合(Vertex Blending)

- 整只手臂可以是“柔性”的。所有的顶点可以由多个矩阵影响。整个多边形网格常称为骨架上的“skin” (外皮)



Vertex blending 例子

# 顶点混合的数学表示



# 顶点混合的数学表示

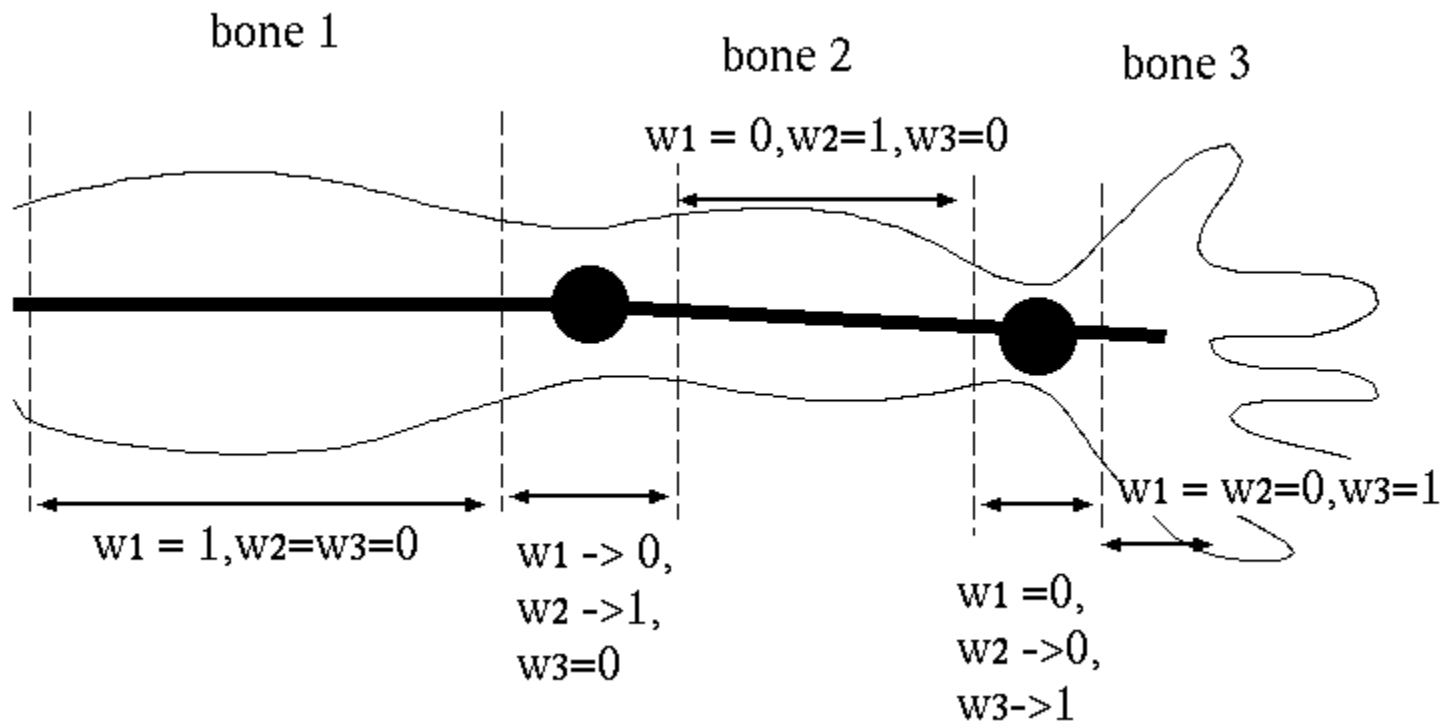
$$\mathbf{u}(t) = \sum_{i=0}^{n-1} \omega_i \mathbf{B}_i(t) \mathbf{M}_i^{-1} \mathbf{P}, \quad \text{其中} \quad \sum_{i=0}^{n-1} \omega_i = 1, \quad \omega_i \geq 0$$

- $\mathbf{P}$ 为变换前的顶点,  $\mathbf{u}(t)$ 为变换后的顶点,  $n$ 为影响 $\mathbf{P}$ 的关节数目
- $\mathbf{M}_i$ : 把第 $i$ 个关节骨架的**局部坐标系**变换到世界坐标系
- $\mathbf{B}_i(t)$ : 第 $i$ 个关节随时间变化的世界变换, 通常是一系列矩阵的连乘
- $w_i$ : 第 $i$ 个关节骨架作用于 $\mathbf{P}$ 的权因子

# 如何来决定权因子?

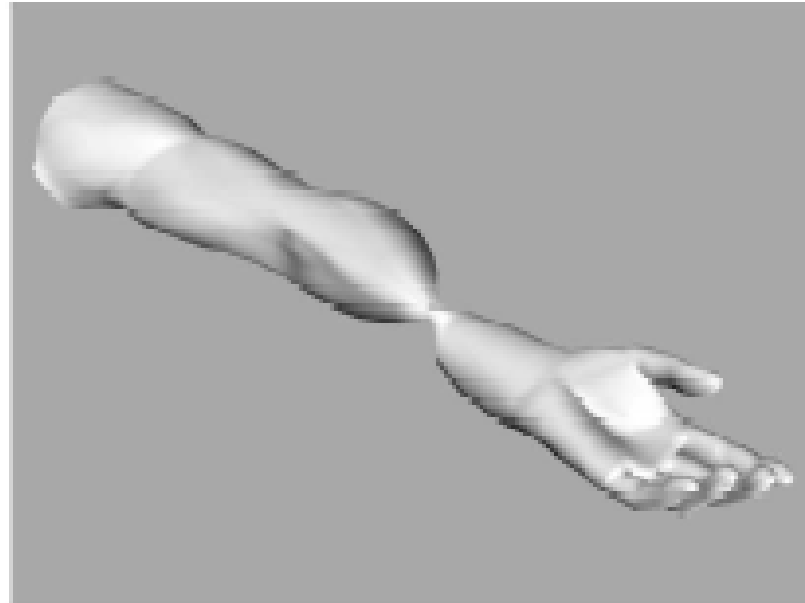
- 决定顶点到骨骼的映射

- 如果顶点 $v$ 在骨骼 $i$ 的中间, 则 $w_i=1$ , 且 $w_j (j \neq i)=0$ ;
- 如果顶点靠近骨骼 $i$ 和骨骼 $i+1$ 的边缘, 则 $w_i$ 逐渐减少到0,  $w_{i+1}$ 逐渐增加到1;
- 如果顶点受到 $\geq 3$ 个骨骼的影响, 其权因子 $w_i$ 可根据顶点到骨骼的距离来决定。



# 线性顶点混合存在的问题

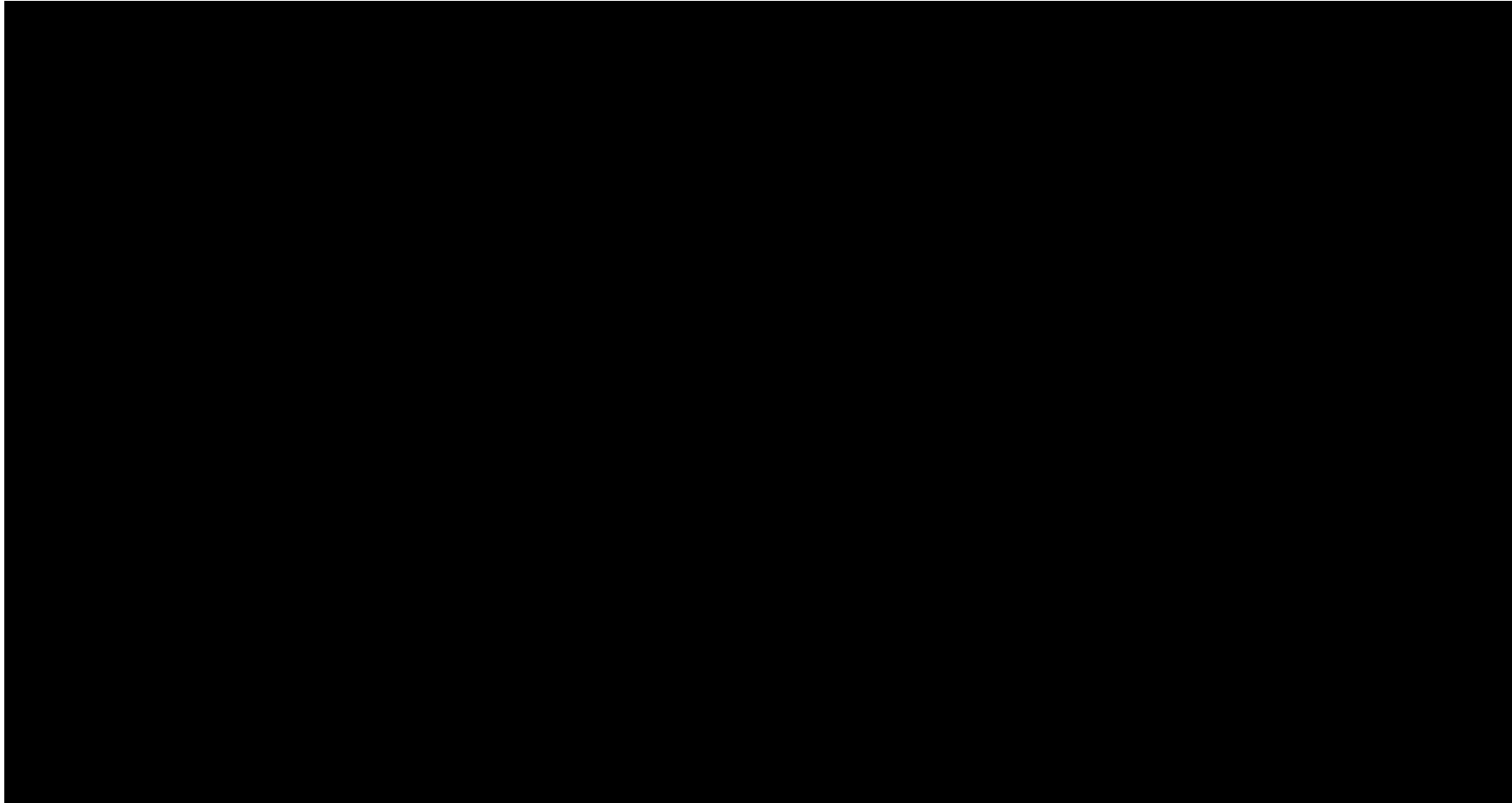
- 当关节旋转到一些极端角时，网格模型的体积会损失；
- 这种现象称为“关节塌陷”或“糖果包裹”；



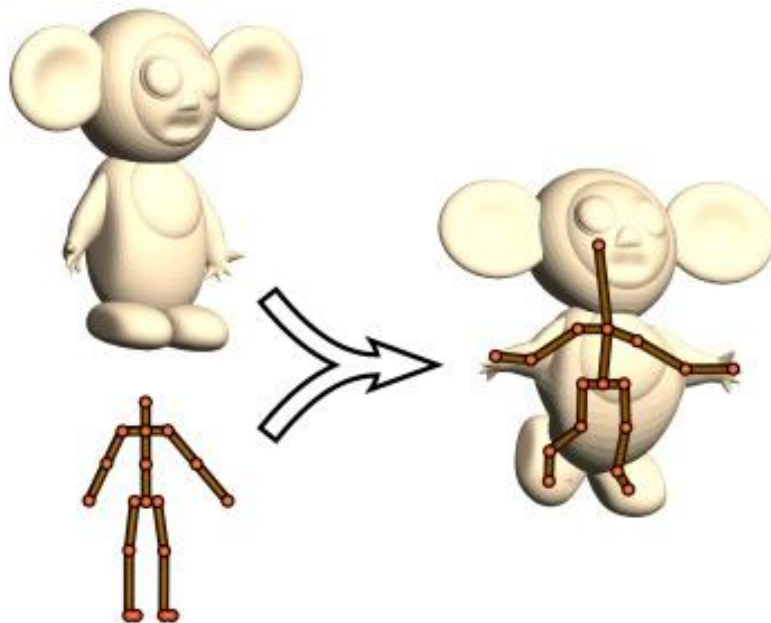


# Spherical blend skinning

---



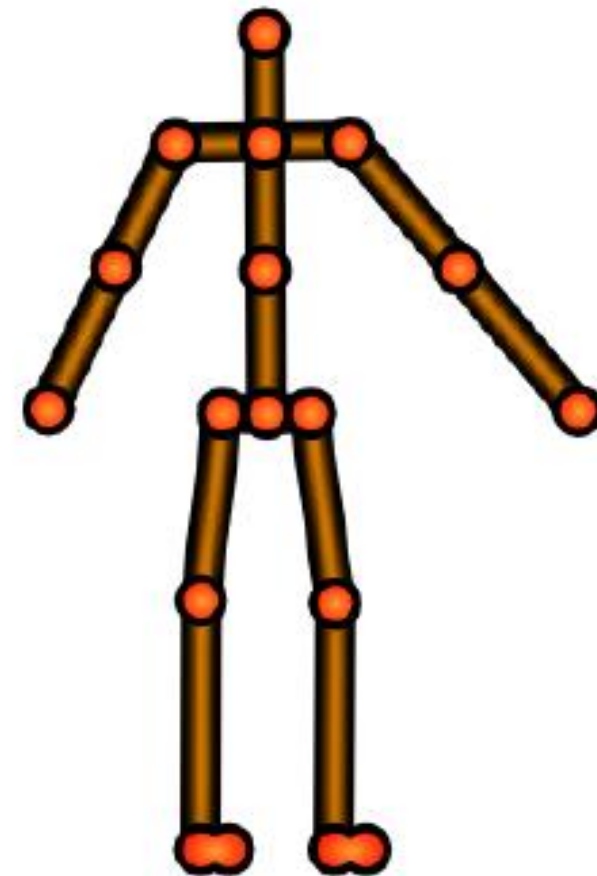
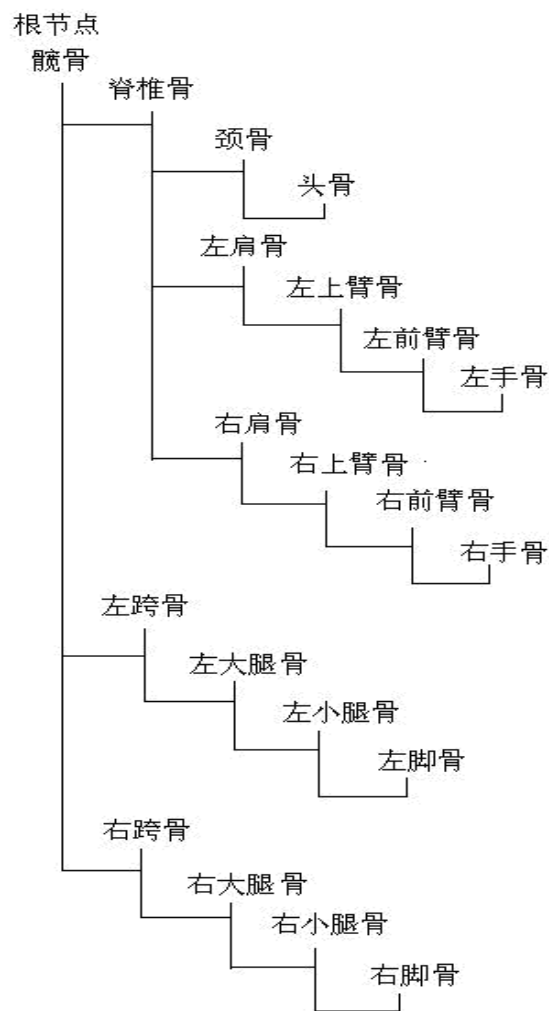
# 骨架与三维角色模型的自动绑定



- 参考文献: Ilya Baran and Jovan Popović. 2007. Automatic rigging and animation of 3D characters. In *ACM SIGGRAPH 2007 papers* (SIGGRAPH '07). ACM, New York, NY, USA, , Article 72 .
- C++ source and a Windows binary: <http://www.mit.edu/~ibaran/autorig/>

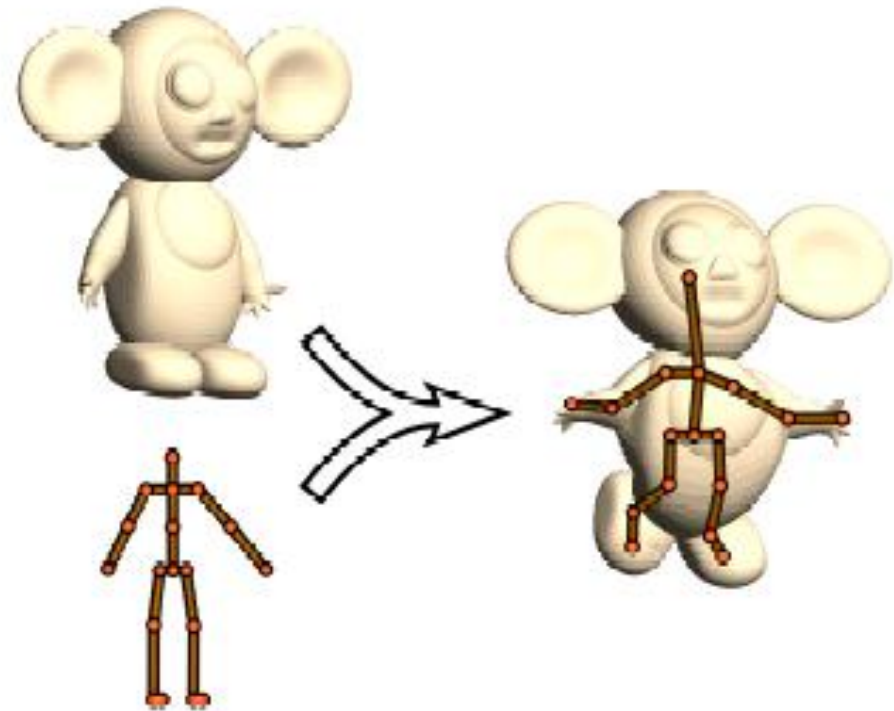
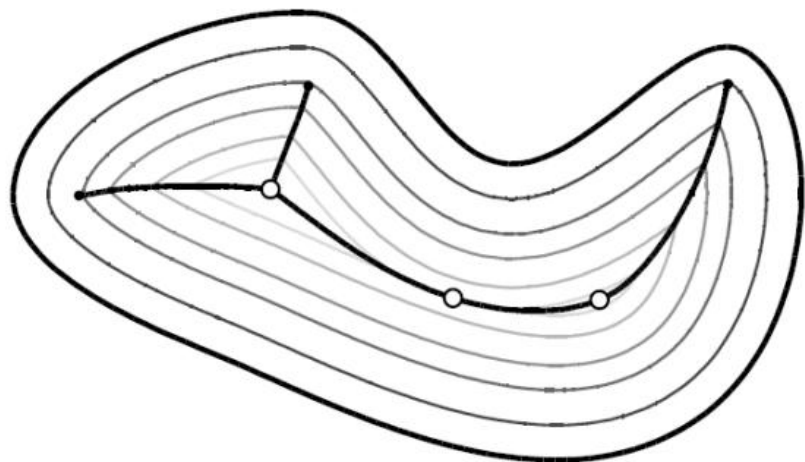
# 主要思想

- **Pinocchio**主要包括两个过程
  - 首先在未知的三维模型中嵌入骨架
  - 然后计算骨骼对表面网格上每个顶点的影响权值，并将表面皮肤依附在骨骼上。



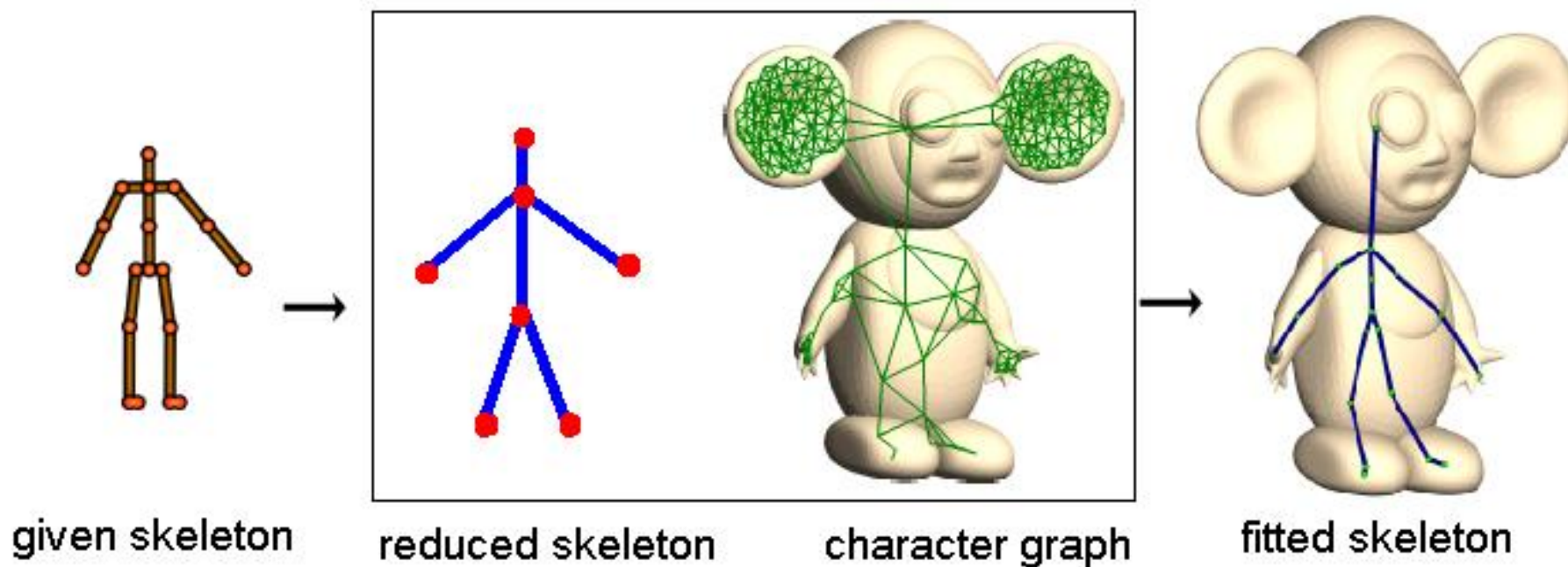
# 骨架计算

- 我们怎样才能**自动**从多边形模型自动算出骨架?
- 可利用物体的中轴(medial axis)
- 可通过园/球体拟合来计算

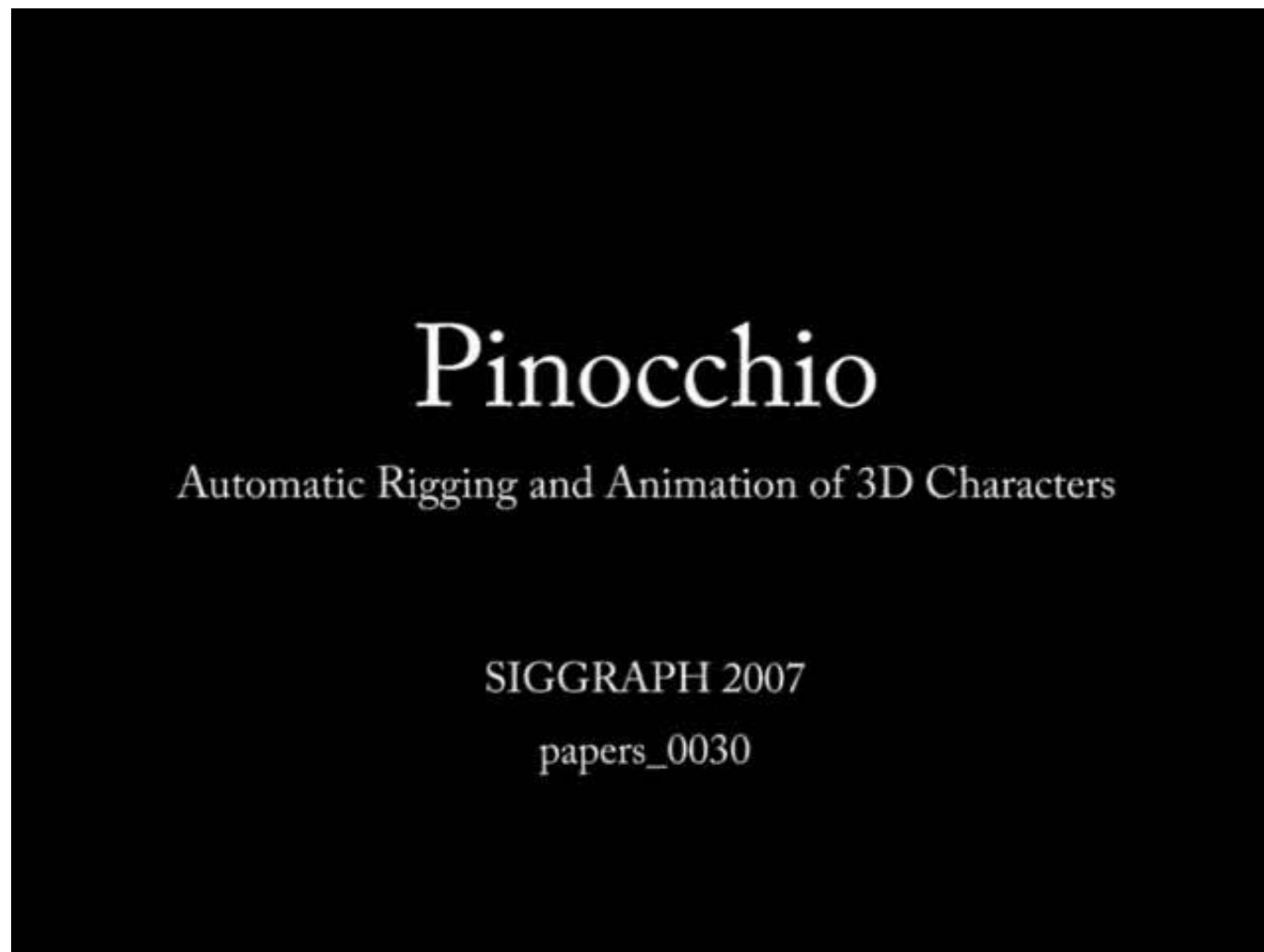


# 离散骨架嵌入

- 从给定的骨架模型建立一个简化的骨架，该骨架包含原骨架的端点和分叉关节点
- 把简化的骨架拟合到角色图(Character graph)中
- 使离散惩罚函数最小化，得到一个嵌入的骨架，再进行优化



# Pinocchio系统演示



# HeterSkinNet: 基于异质图神经网络的自动蒙皮系统

---

## HeterSkinNet: A Heterogeneous Network for Skin Weights Prediction

---

Xiaoyu Pan, Jiancong Huang, Jiaming Mai, He Wang, Honglin Li, Tongkui Su, Wenjun Wang, Xiaogang Jin: HeterSkinNet: A Heterogeneous Network for Skin Weights Prediction. Proc. ACM Comput. Graph. Interact. Tech. (Special Issue of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games) 4(1): 10:1-10:19 (2021)

# 运动捕获 (Mocap)



**猩球崛起3: 终极之战**

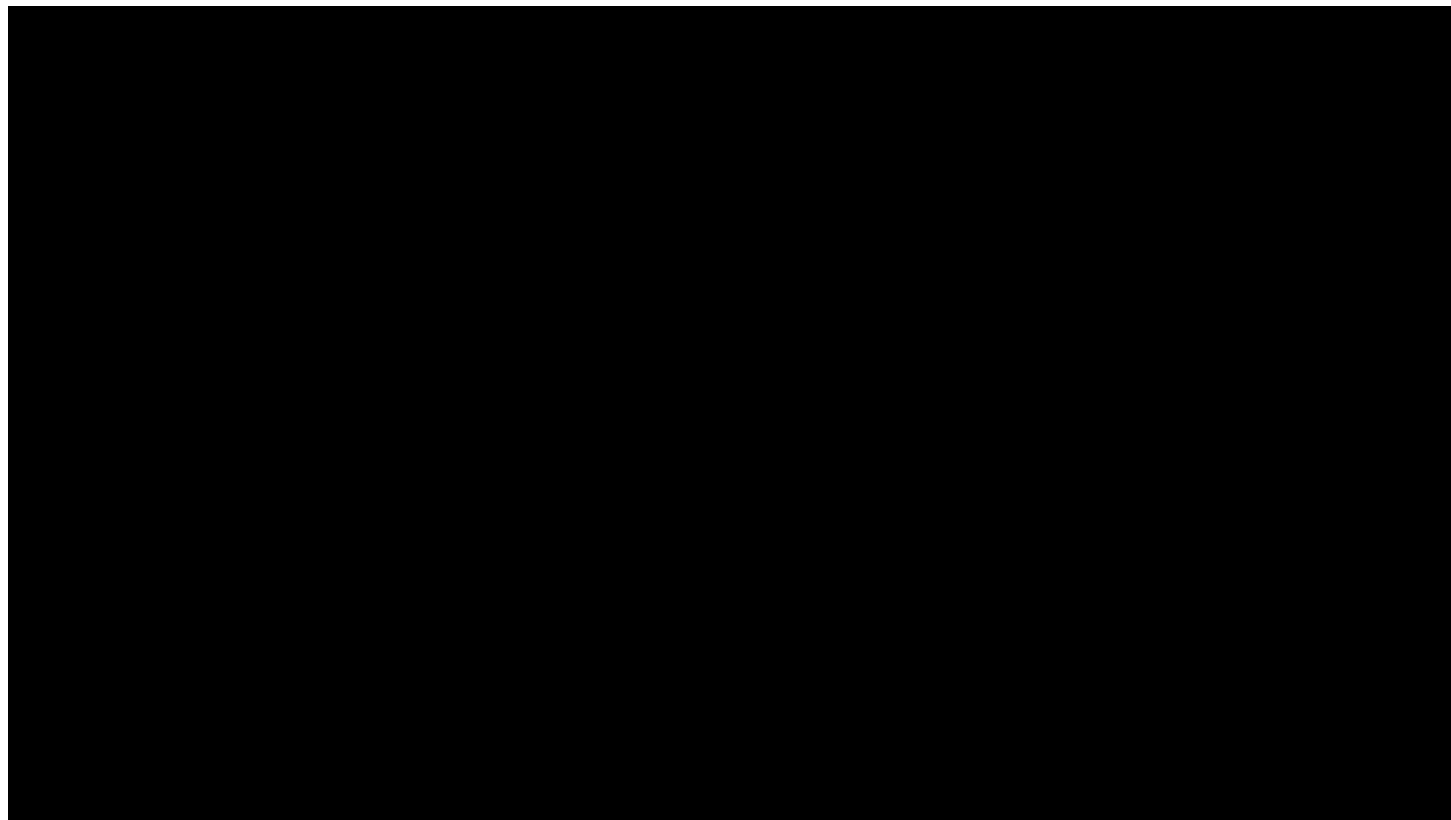
Photo Source: Courtesy of 20th Century Fox – Pictured: Andy Serkis in War for the Planet of the Apes

# 运动捕获定义(Wiki百科)

- **Motion capture is the process of recording the movement of objects or people.**
- In filmmaking and video game development, it refers to recording actions of **human actors**, and using that information to animate digital character models in 2D or 3D computer animation
- When it includes face and fingers or captures subtle expressions, it is often referred to as **performance capture**.
- In many fields, motion capture is sometimes called **motion tracking**, but in filmmaking and games, motion tracking usually refers more to match moving.

# 运动捕获

- 运动捕获(MoCAP, Motion Capture)采用软硬件系统记录表演者的真实运动信息, 并把动作过程复制到一个虚拟的人或动物上。
- 运动捕获可以获取表演者动作的个性和运动细节, 是生成逼真人体动画最实用、最有效的方法, 因而在影视特技、游戏、动画、机器人控制中被广泛采用。



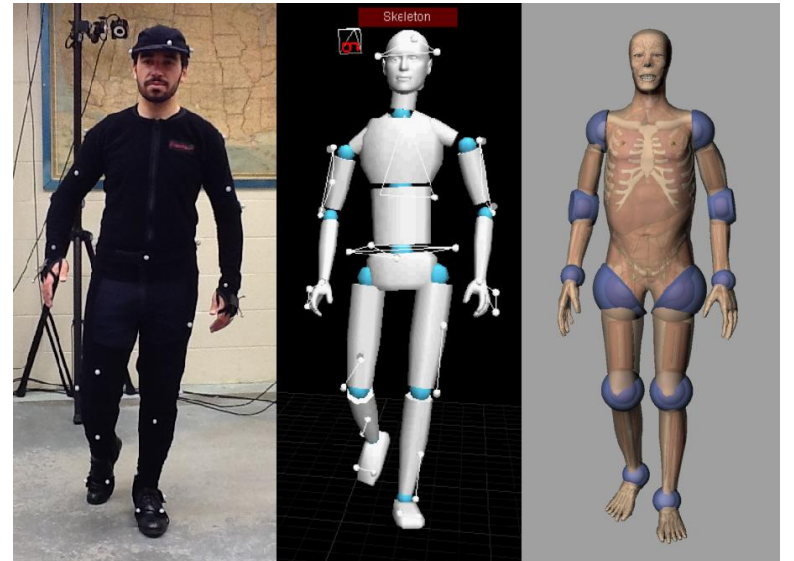
# 采用运动捕获的游戏

- NBA live
- PGA tour (PGA巡回赛)
- NHL hockey (NHL冰球大赛)
- Legends of Wrestling 2 (摔跤传说2)
- Lords of Everquest (无尽的任务战略版)
- Lord of the Rings
- Tony Hawk Pro Skater (托尼霍克滑板)
- Batman: Dark Tomorrow (蝙蝠侠:黑暗未来)
- Grand Theft Auto (侠盗猎车手)
- Soul Caliber (刀魂)

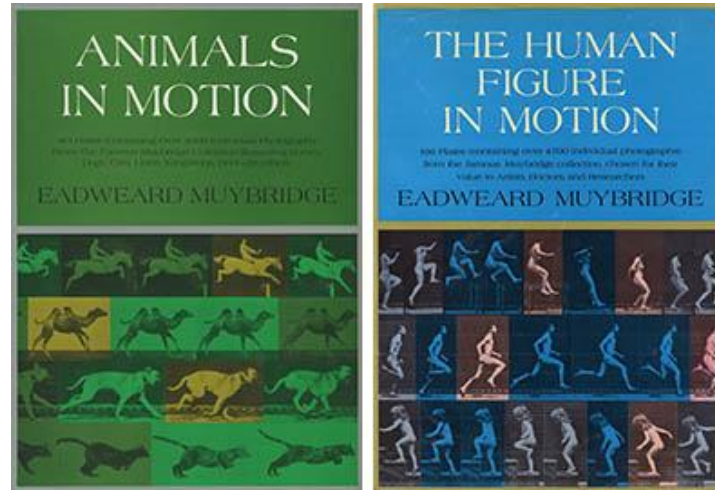
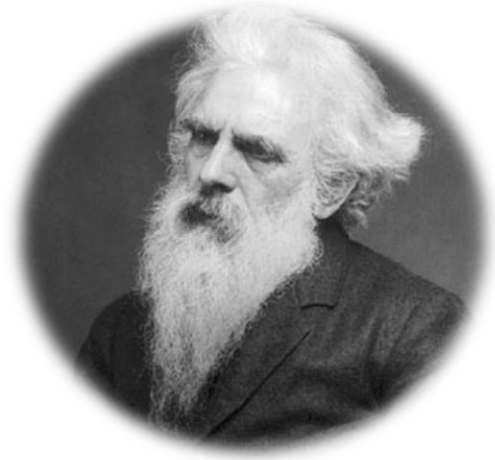


# 为什么需要运动捕获？

- 正向运动学、逆向运动学
  - 制作物理真实的运动是件非常艰巨的任务
  - 需要大量的专业动画人才
- 把运动进行记录并把结果映射到虚拟角色
  - 更加容易
  - 真实运动

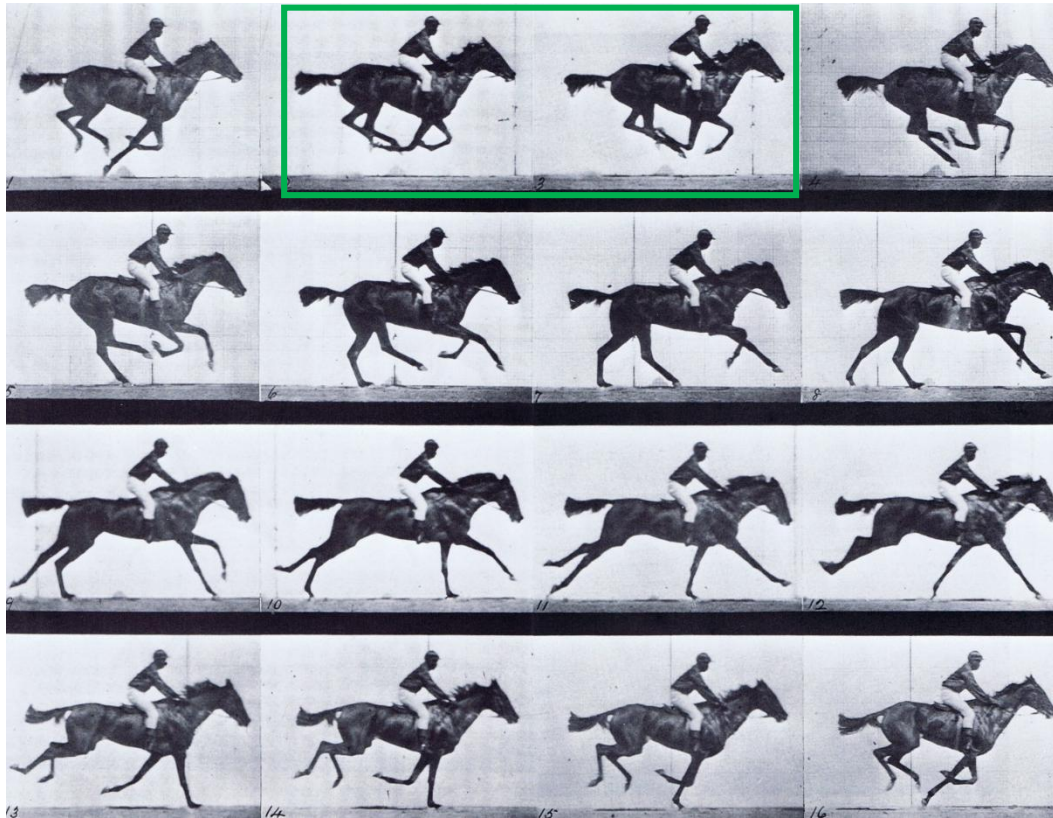


# 运动捕获历史



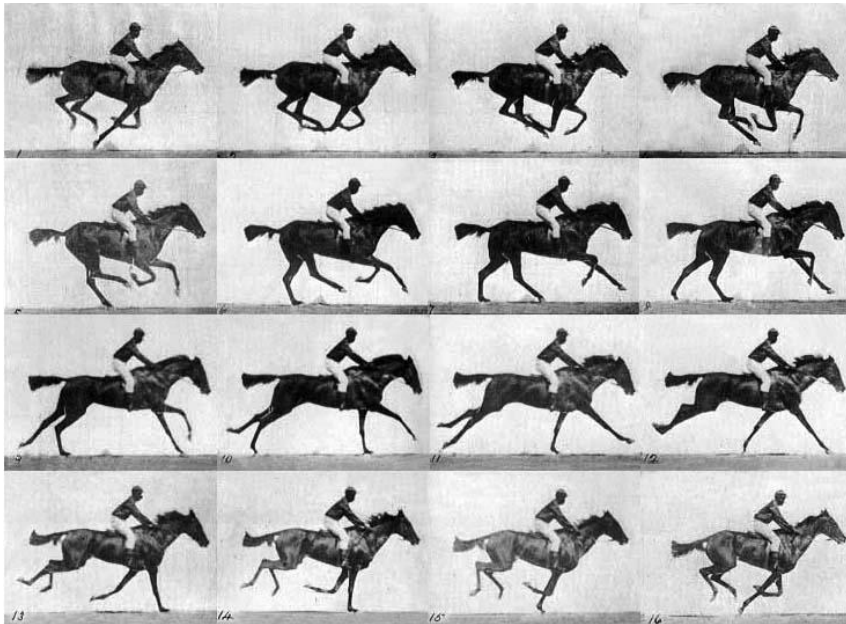
- **Eadweard Muybridge (1830 - 1904)**
- English photographer, pioneered photographic studies of motion and motion-picture projection.
- In 1872 former governor of California Stanford hired Muybridge to prove all four feet of **a horse were off the ground at the same time while trotting (小跑)**.

# 《Animals in Motion》

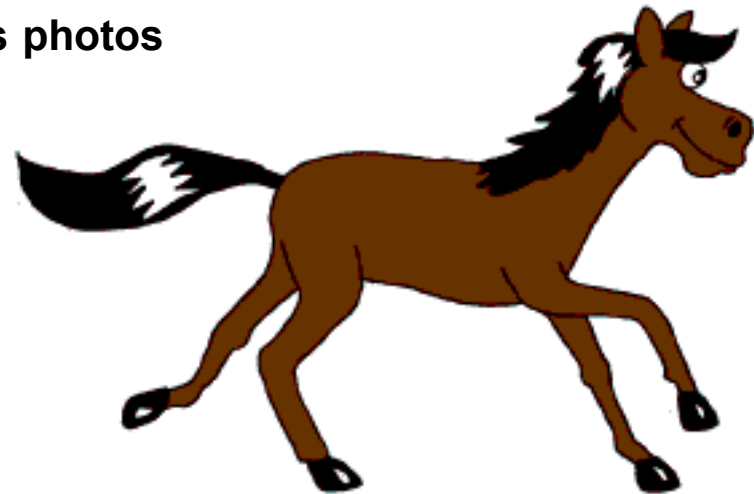


# 影像描摹(Rotoscoping)

- Allow animators to trace **cartoon character** over photographed frames of live performances
  - Invented by **Max Fleischer in 1915**
  - 2D manual motion capture



A horse animated by rotoscoping from Muybridge's photos



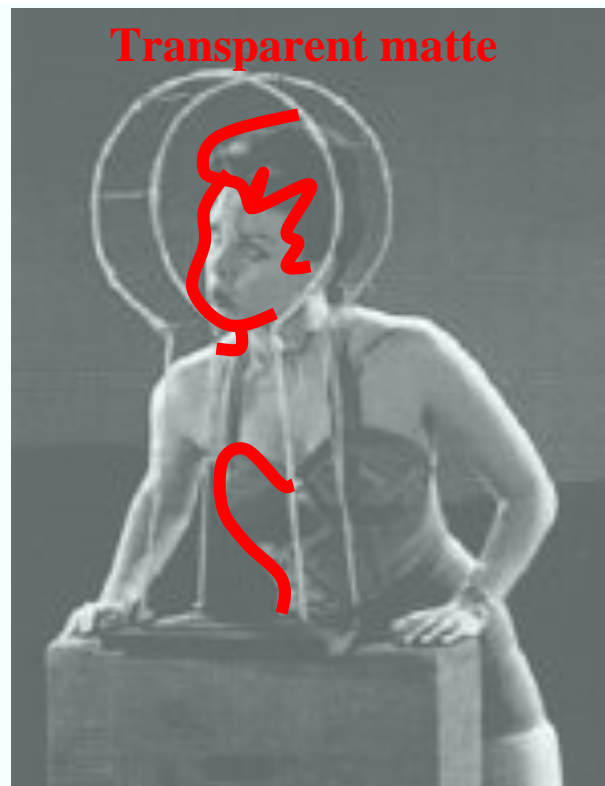
# 影像描摹(Rotoscoping)

运动捕获方法类似于早期Disney公司制作卡通片《白雪公主》时使用的

“rotoscoping”技术，即动画师根据素材画面，采用手工交互的方式跟踪获取画面主体的运动信息。



Captured Video



Tracing



Post Processing

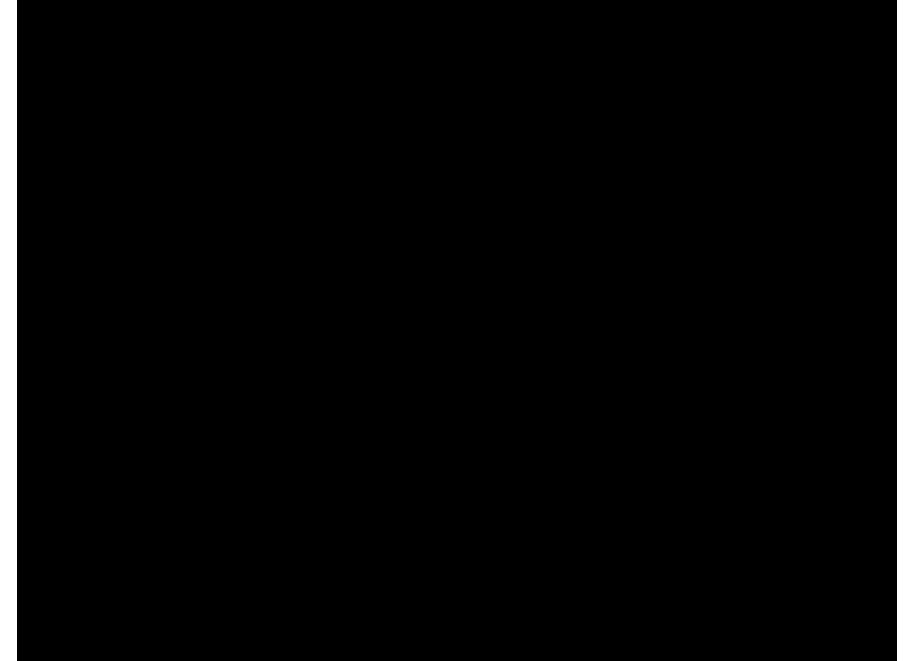
# 运动捕获系统分类

可看作“3D Rotoscoping”：measuring 3D positions, orientations, velocities or accelerations *automatically*

- **电磁**捕获系统(Electromagnetic)
- **机电**捕获系统(Electromechanical)
- **光纤**捕获系统(Fiber optic)
- **光学**捕获系统(Optical)
- **基于视频**的捕获系统(Video-based Mocap)

# 电磁捕获系统(Electromagnetic)

- Each sensor record 3D position and orientation
- Each sensor placed on joints of moving object
- Full-body motion capture needs **at least 15 sensors**
- Popular system: <http://www.ascension-tech.com/>



# 电磁捕获系统(Electromagnetic)

- **Pros**

- measure 3D positions and orientations
- no occlusion problems
- can capture multiple subjects simultaneously

- **Cons**

- magnetic perturbations (metal)
- small capture volume
- cannot capture deformation (facial expression)
- hard to capture small bone movement (finger movement)
- not as accurate as optical mocap systems



# 机电捕获系统(Electromechanical)

- Each sensor measures 3D orientations
  - including 3D accelerometers(加速度传感器), 3D gyros(陀螺), and 3D magnetometers(磁传感器)
- Each sensor placed on joints of moving object
- Full-body motion capture needs **at least 15 sensors**
- Popular systems: <http://www.xsens.com/>



# 机电捕获系统(Electromechanical)

- **Pros**

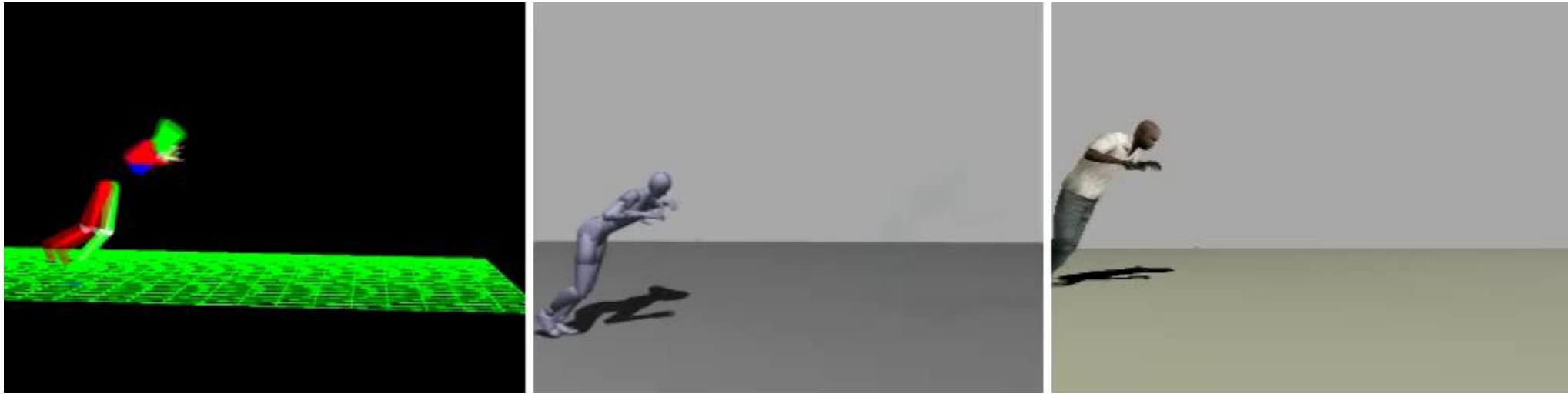
- measure 3D orientations
- no occlusion problems
- can capture multiple subjects simultaneously
- large capture volume
- portable and outdoors capture (e.g. skiing)

- **Cons**

- getting 3D position info is not easy
- the root positions is often measured with ultrasonic position sensors
- cannot capture deformation (facial expression)
- hard to capture small bone movement (finger motion)
- not as accurate as optical mocap system



# 光纤捕获系统(Fiber optic)



- Measures 3D position and orientation of entire tape
- Binding the tape to the body
- Popular systems: <http://www.measurand.com/>

# 光纤捕获系统(Fiber optic)

- **Pros**

- measure 3D positions and orientations
- no occlusion problems
- can capture multiple subjects simultaneously

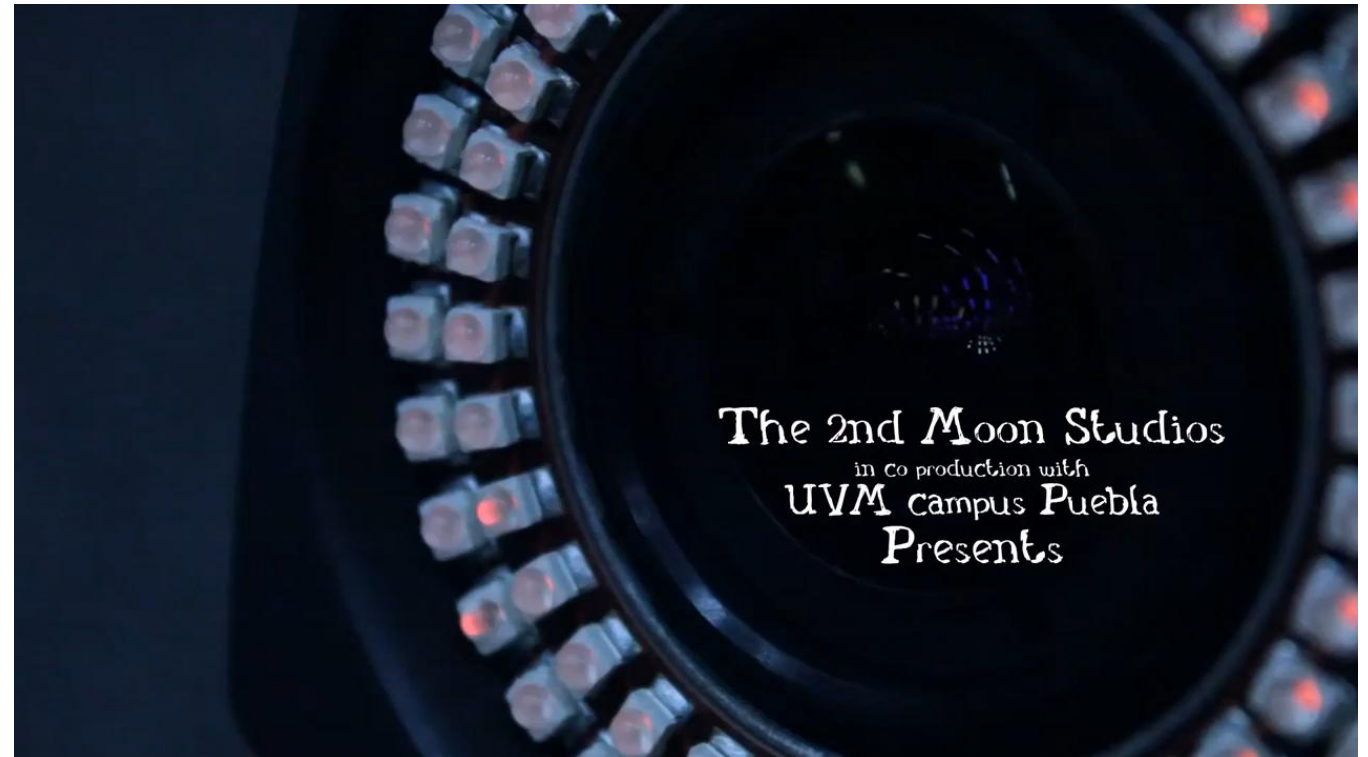
- **Cons**

- magnetic perturbations (metal)
- small capture volume
- cannot capture deformation (facial expression)
- hard to capture small bone movement (finger movement)
- not as accurate as optical mocap systems



# 光学捕获系统(Optical)

- Multiple **calibrated cameras** ( $\geq 8$ ) digitize different views of performance
- Wears retro-reflective markers(反光标记)
- Accurately measures 3D positions of markers
- Vicon mocap system:  
<http://www.vicon.com>



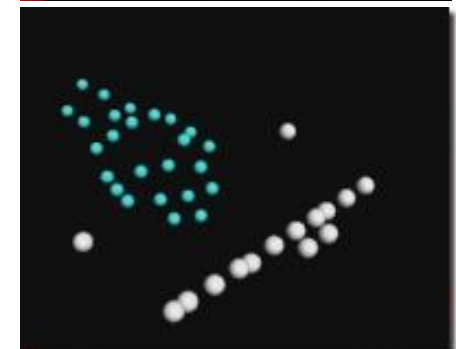
# 光学捕获系统(Optical)

- **Pros**

- measure 3D positions and orientations
- the most accurate capture method
- very high frame rate
- can capture very detailed motion (body, finger, facial deformation, etc.)

- **Cons**

- has occlusion problems
- hard to capture interactions among multiple actors
- limited capture volume
- expensive



# 基于视频的捕获系统(Video-based Mocap)

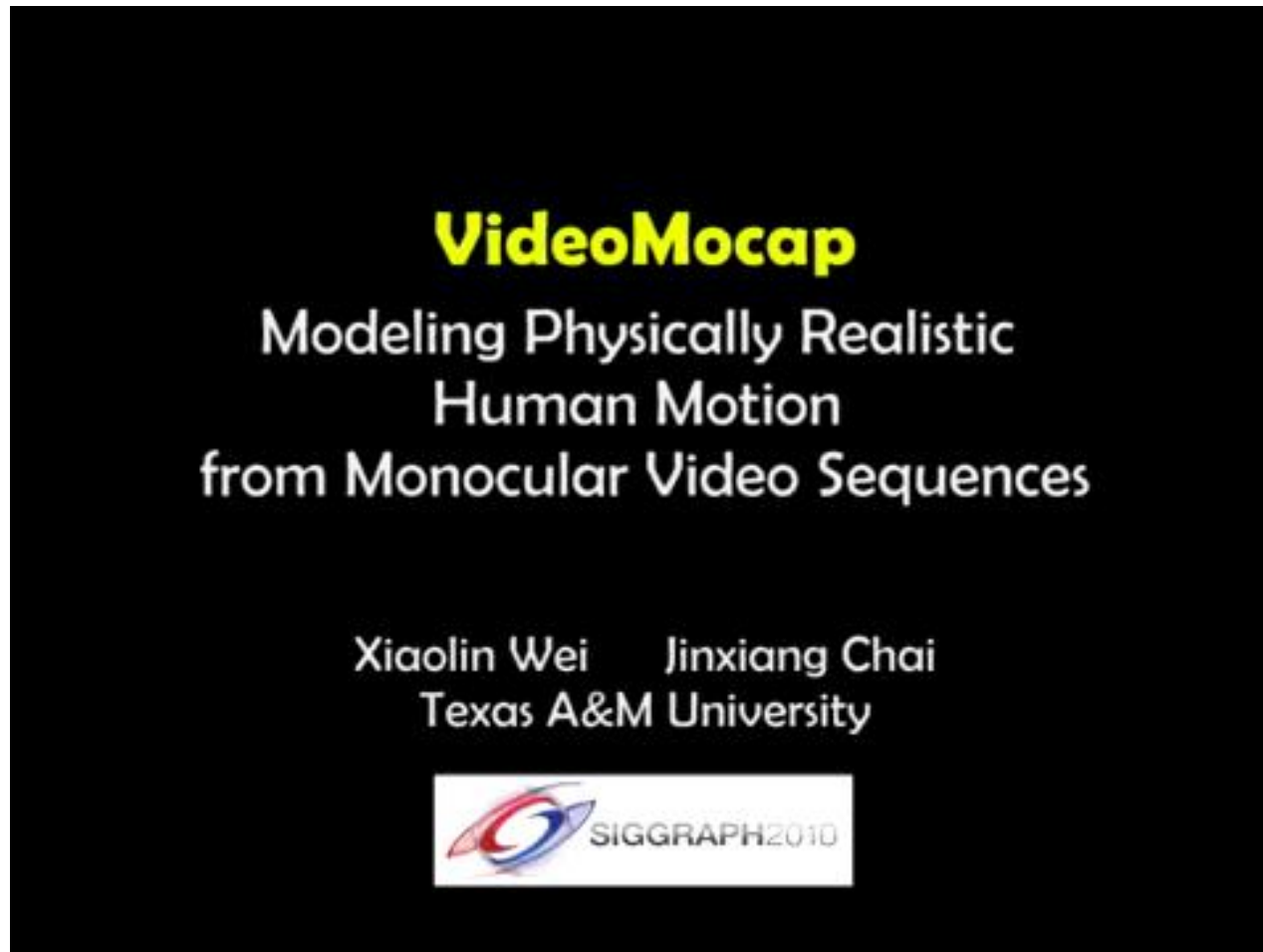
- Capturing 3D performance from single-camera video streams

## Pros:

- capturing human motion anytime, anywhere
- very cheap
- zillions of films, sports footage, and internet videos.


## Cons:

- not a mature technology
- quality is not as good as other capturing technologies.



**VideoMocap**  
Modeling Physically Realistic  
Human Motion  
from Monocular Video Sequences

Xiaolin Wei   Jinxiang Chai  
Texas A&M University

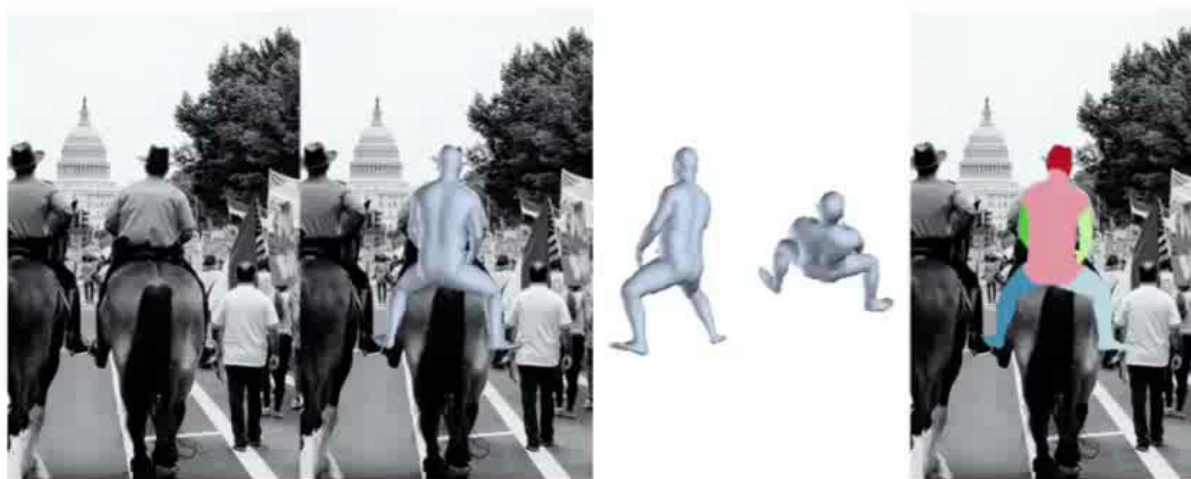


# 近几年活跃方向：基于深度学习的姿态恢复

## End-to-end Recovery of Human Shape and Pose

Angjoo Kanazawa, Michael J Black, David W. Jacobs, Jitendra Malik

Supplementary Materials



# 运动捕获主要思路 (关节)

- 跟踪**参考点**的运动
- 把参考点的信息转化为**关节角度**
- 用关节角驱动关节模型
- 结合运动轨迹，进行角色的控制



# MOCAP流水线 (关节)

- 规划(Planning)
- 标定(Calibration)
- 捕获(Capture)
- 三维位置重建(3D Position Reconstruction)
- 拟合到骨架(Fitting to the Skeleton)
- 后处理(Post Processing)

# Motion Capture的利弊

- 优点

- 只要能被捕获，可记录人体运动的所有细节，运动真实

- 缺点

- 不容易进行
  - 编辑
  - 控制
- 较昂贵



New animation controls make the characters' lips stick together a bit before opening. Farquaad's forehead wrinkles automatically.



Animators could use more than 750 controls to create Shrek's performance. Some controlled one joint or muscle, others controlled groups of several.

Online Motion Capture Database: <http://mocap.cs.cmu.edu/>

# 艺术家对MOCAP的评述

Eric Darnell, co-director of Antz



- **The main problem** with motion capture associated with characters has to do with **mass distribution, weight, and exaggeration**. He says that it is impossible for a performer to produce the kind of motion exaggeration that a cartoon character needs, and the mass and weight of the performer almost never looks good when **applied to a character of different proportions**.

# 艺术家对MOCAP的评述

Richard Chuang, VP at PDI



- The mapping of human motion to a character **with non-human proportions doesn't work**, because the most important things you get out motion capture are the **weight shifts and the subtleties and that balancing act of the human body**. If the proportions change, you throw all that out the door, so you might as well animate it.

# 艺术家对MOCAP的评述



## Godzilla: Karen Goulekas



- The reason that we pulled the plug (终止) on using the motion capture was, very simply, because **the motion we captured from the human actor could not give us the lizard-like motion we were seeking**. The mocap could also not reflect the huge mass of Godzilla either. During our keyframe tests, we found that the Godzilla motion we wanted was one that maintained the sense of huge mass and weight while still moving in a graceful and agile manner. No human actor could give us this result.

# 总结

---

- IK求解技术
- 骨骼绑定技术
- 运动捕获技术

**The End**