

# Inverse Kinematics

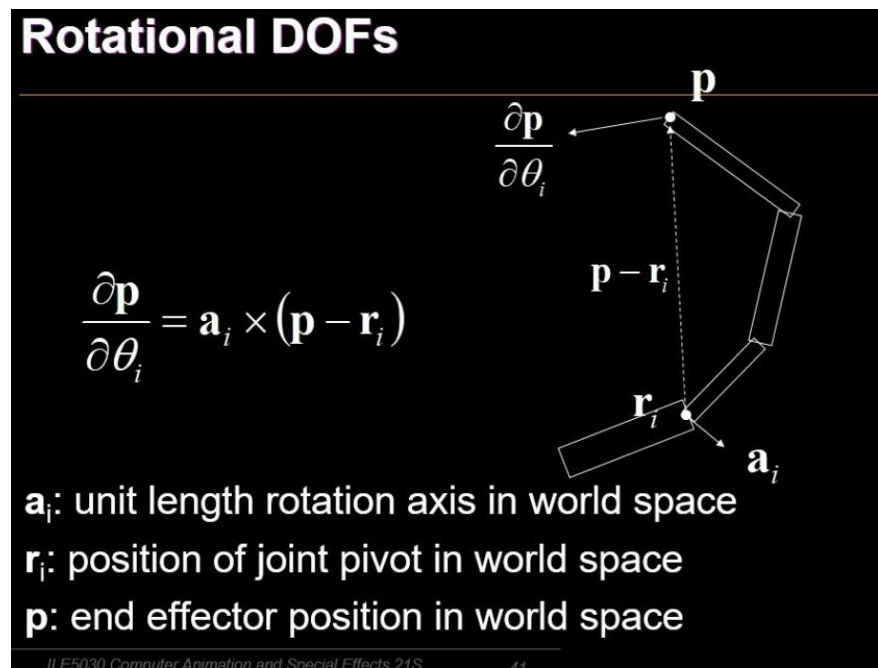
資科工碩 309551007 陳鎮華

## ● Introduction

此份實驗主要在實作反向運動學(Inverse Kinematics, IK)，先前實作的正向運動學為透過骨架從根節點向外一個個關節的在地坐標系推出各關節的世界座標為何，但在某些情況下，我們需要將指定的關節擺放在指定的位置上(如：抓握、爬梯...)。此時我們必須將指定的關節移動到目標位置，並計算出其他連動關節需要調整的適當角度，這樣子透過目標位置反推骨架其他關節位置角度的方法稱為反向運動學。

## ● Fundamentals

我們可以透過先前正向運動學的式子反推出由目標位置回推的式子，原理上近似於所有父子關節的累積結果應該要使最後一個節點的位置能落在目標位置上。我們可以將 Jacobian 內的項視為各關節自由度所造成的影響，每一個 column 放的是骨頭擁有旋轉自由度的旋轉軸，而所有變動關節自由度的影像總和應該會往指定的位置趨近，再將計算過程透過迭代的方式慢慢將指定關節接近目標位置。



(Source: NCTU-CS Computer Animation Course )

上方圖片代表變化的其中一個關節，也代表著 Jacobian 內的其中一項，並可以透過上方式子表示出關節角度變化後所造成的影響，每個關節的所有自由度影響累積即為使指定關節往目標位置趨近。

## Iterative IK Using Inverse Jacobian

$$\theta = f^{-1}(P)$$

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

$$\dot{\theta} = J^{-1}(\theta)V$$

$$\theta_{k+1} = \theta_k + \Delta t J^{-1}(\theta_k)V$$

- Linearize about  $\theta_k$  locally
- Small increments

(Source: NCTU-CS Computer Animation Course )

在反向運動學的計算過程中，我們需要透過迭代的方法慢慢將指定關節向目標位置趨近，我們可以先透過 Pseudoinverse 與最小平方方法解出 theta 的結果再對骨架做更新。

### ● Implementation

實作方面必須先建立 Jacobian Matrix，每個 Column 代表影響關節的各個自由度的旋轉軸，若此自由度無法旋轉則將旋轉軸設為零向量，因此在三維空間下加上 affine transformation 的關係，計算方便我們 Jacobian 的大小為  $(4, 3 * \text{bone\_num})$ 。而 target 的矩陣相差多少的距離向量  $(4, 1)$ ，而解出來的 theta 矩陣則為關節個自由度需要增加的角度  $(3 * \text{bone\_num}, 1)$ 。Pseudoinverse 的部分由於我們要維持 row independent 的特性，因此我們將最後一個 row 先剔除(全為 0)，再進行 J+ 以及 theta 的計算。

### ● Result and Discussion

#### ■ 重建後的骨架模型



## ■ 不同關節指定位置



## ■ 討論

反向運動學的模擬可將指定關節移動到目標的位置，並將過程中可能影響到的關節自動的連帶進行影響，但在迭代的過程中有時候出現不穩定的現象會使骨架出現不自然的晃動與震盪。

## ● Conclusion

反向運動學與正向運動學為動畫領域最基本的傳統演算法，實作方法也有許多種，此兩項技術皆為實用且常用的技巧，只要將數學的過程理解清楚，在實作上將矩陣填寫正確就得到蠻好的效果，但在某些情況下需要檢查解出來的角度是否有維持穩定，否則會使骨架造成無限擺盪的不自然現象。