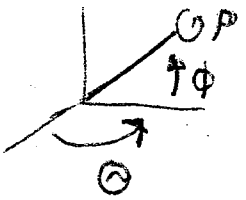
Lecture 8: More Rotations + Hierarchies

Gimbal Lock:

Easiest to show with a picture of a gimbal (thanks to Nick)



$\theta = z$
$\phi = y$
$\rho = x$

if $\phi = 90°$, θ and ρ do the same thing

Quaternions don't have these problems
embed $SO(3) \in S(3)$

Have other problems:
- need unit length (or self normalizing)
- not all sets of 4 numbers ~~are~~ are rotations
  must keep this constraint
- auto normalization isn't a cure-all
  direction of change vanishes
- what user interface?
- interpolation not obvious
- But: can do "right" thing, interpolate

A hack -

Say you know 3 quaternion parts - x, y, z
and $\sqrt{x^2 + y^2 + z^2 + w^2} = 1$
you can compute w (to a sign flip)

This is an idea of where the Exponential Map comes from.

Exponential Maps:

Idea: encode axis/angle by making angle be the magnitude of the axis
(this is what the hack above does)

Clearly - there must be a problem (encoded rotation in $\mathbb{R}^3$)

- zero vector, things get wierd
  Can't compute a direction to not rotate about
  (Sebastian's paper shows how to approximate)
- rotate once (or n times around) things get wierd - all points on the $\|v\| = 2\pi$ sphere are the same!
  so don't get that far away!
- Ball and Sockets Stuff - very useful,
  can't say I've worked through the details

Interpolating Quaternions and Exp Maps

- Exp Maps are in $\mathbb{R}^3$, so can interpolate
  takes some care to make sure you get consistent quaternions
- Recent results show how to spline quaternions - but its not easy!

How to deal w/ Quaternions

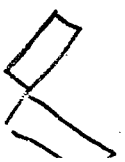
- use them internally, but show Euler Angles to the user (3D Studio MAX)
  gets interpolation right
- get timing by slerp, with control over t

Kinematics -

the study of how things move without regard to what makes them move

CG: articulated (hierarchical figures) without dynamics

Why kinematics?



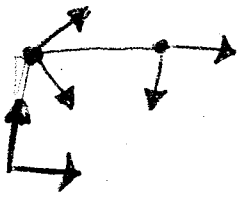
6 d.o.f + 6 d.o.f - constraint

6 d.o.f + relative motion

Can model constraints explicitly

Hierarchy

local vs. parent co-ordinate systems

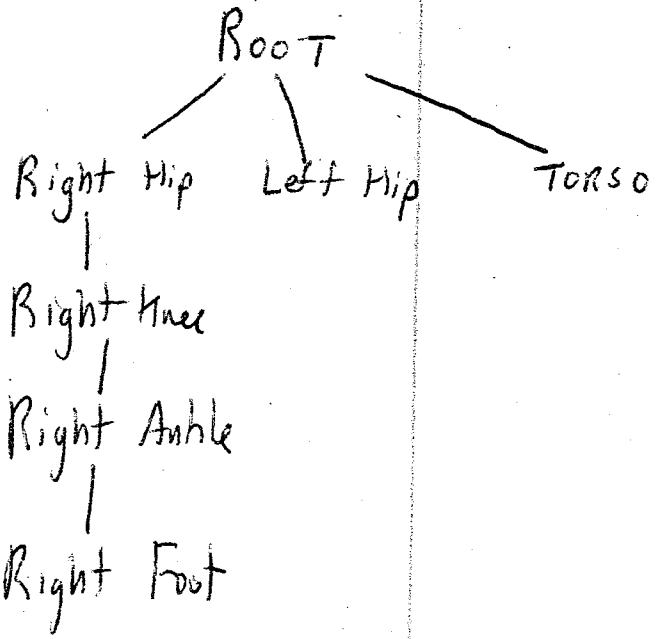


root

joint @ 1,5 $\theta = +45°$

joint @ 4,4 $\theta = +45°$

TREES

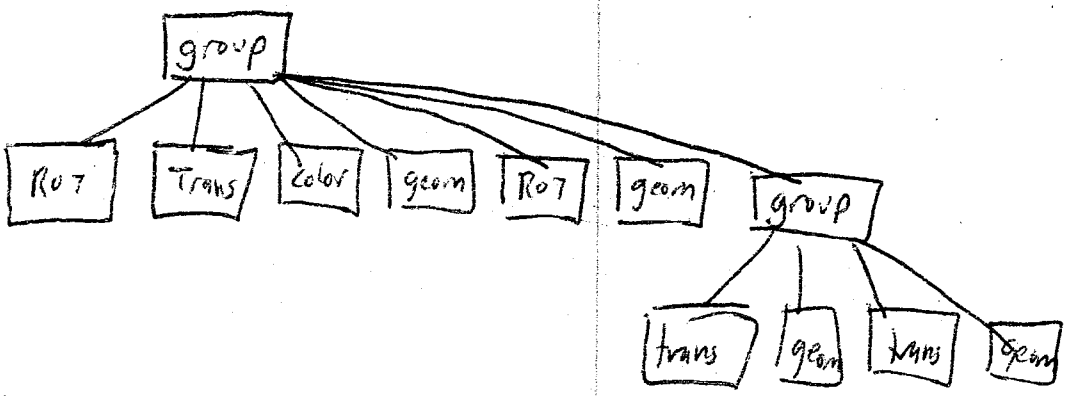


(1) Doesn't have to be just rotations

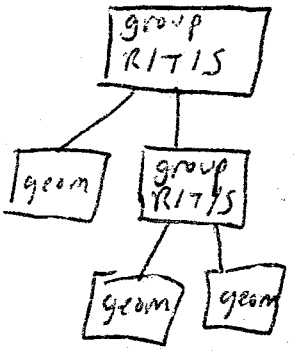
Generalized hierarchies

Inventor - Style (transforms in line)

vs.



vs.



matrix stacks
traversals

forward kinematics -
given parameters ⇒ where are things
$x = f(q)$

inverse kinematics
given goal positions ⇒ what angles
~~$f(q)$~~

"generalized IK"
doesn't just have to be rigid, articulated figure
solve $x = f(q)$ for q given x

forward = function evaluation
inverse = equation solving