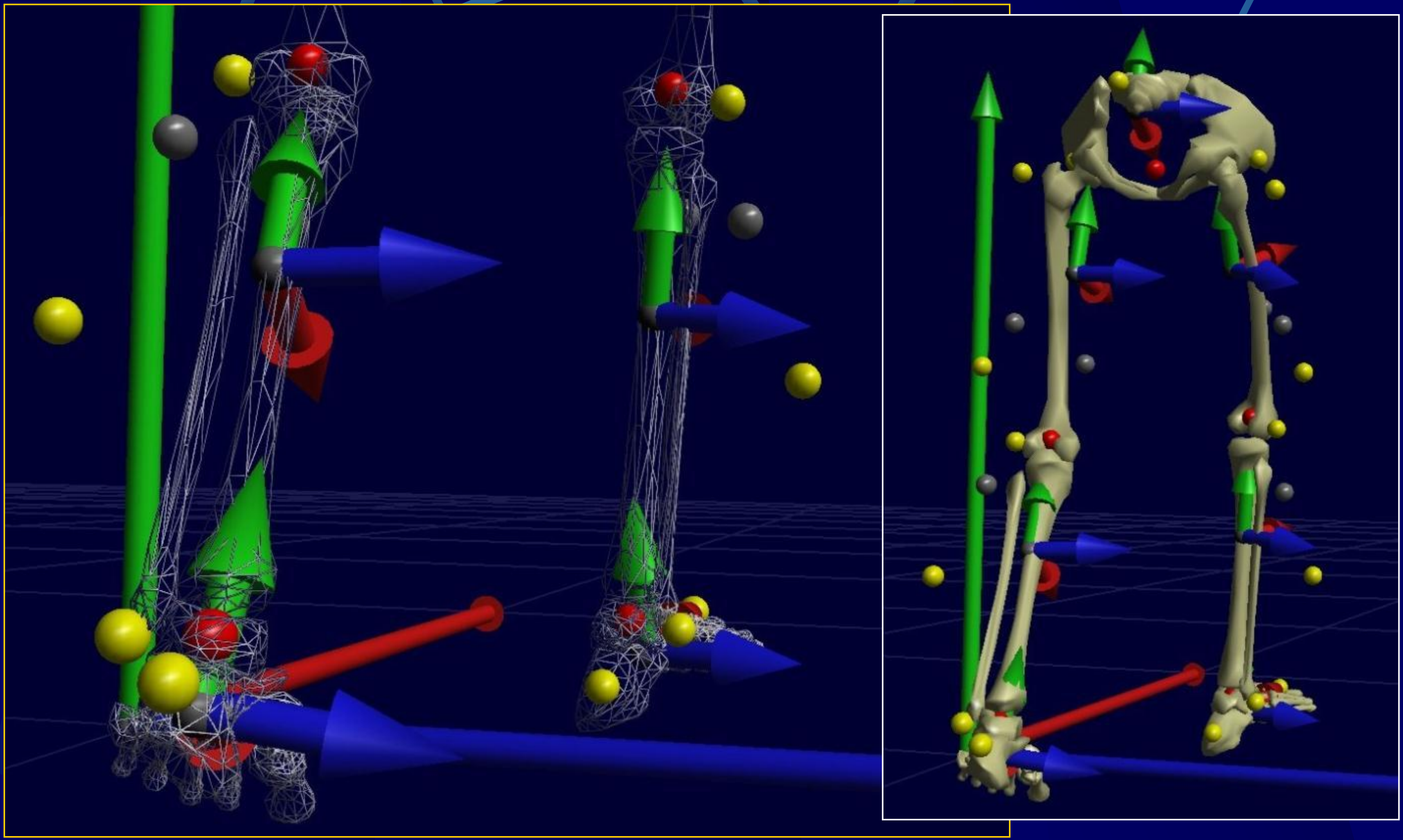
Visualizing Orientation using Quaternions

Gideon Ariel; Rudolf Buijs;
Ann Penny; Sun Chung

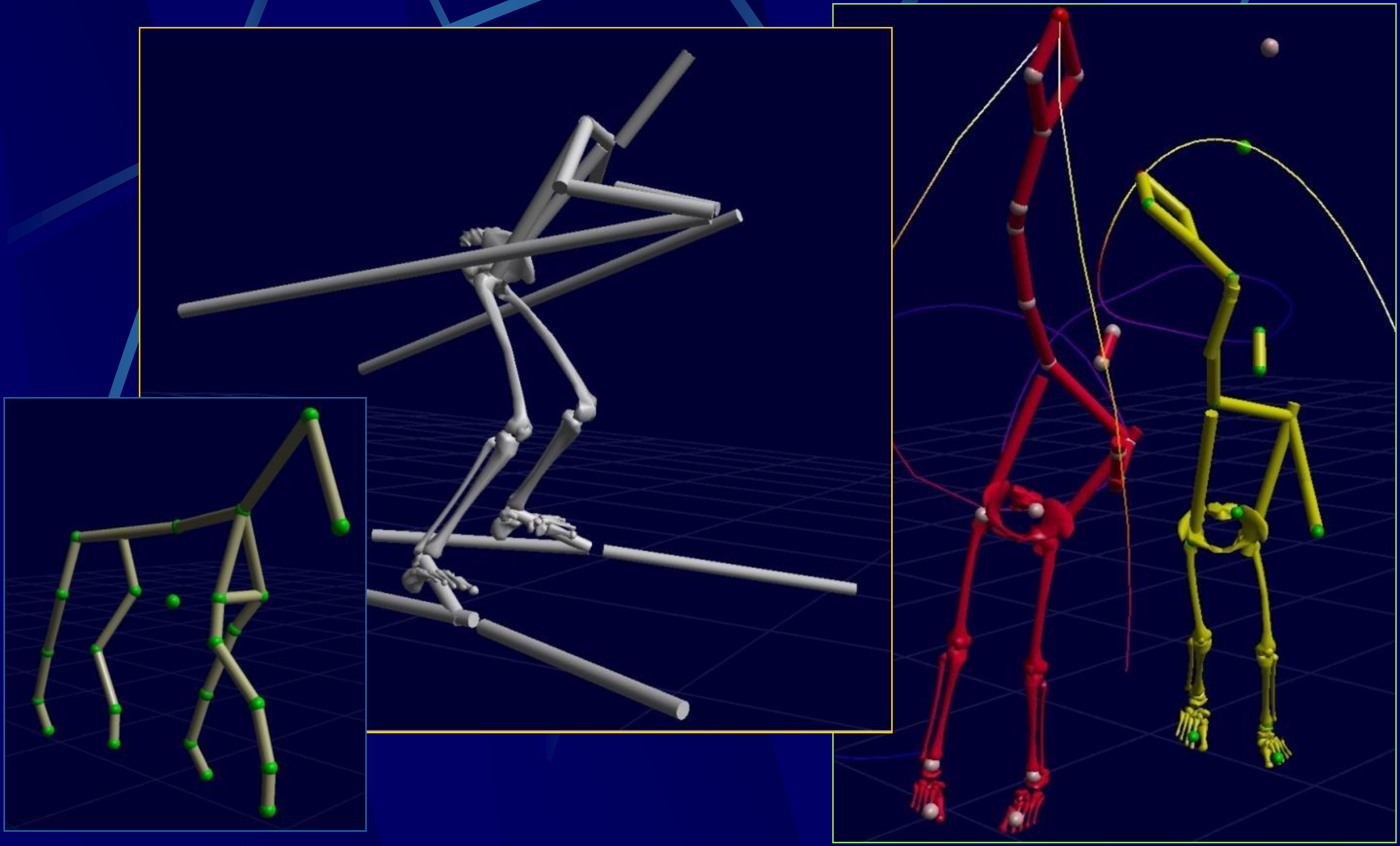
Introduction

- The Issue: representing orientation
  - Not a “religious” discussion about interpretation or standardization of (joint) angles (Woltring – Grood)
  - But rather, a practical solution to visualization issues
- Context: visualization
  - Rendering offers a better 3D “view” than traditional methods
  - Real-time rendering is possible
  - Required hardware has become affordable
- The Math
  - Euler angles and their many “challenges”
  - Alternatives to Euler angles
  - Best alternative: quaternions
- Conclusions
- Q & A

Rendering – a better 3D view



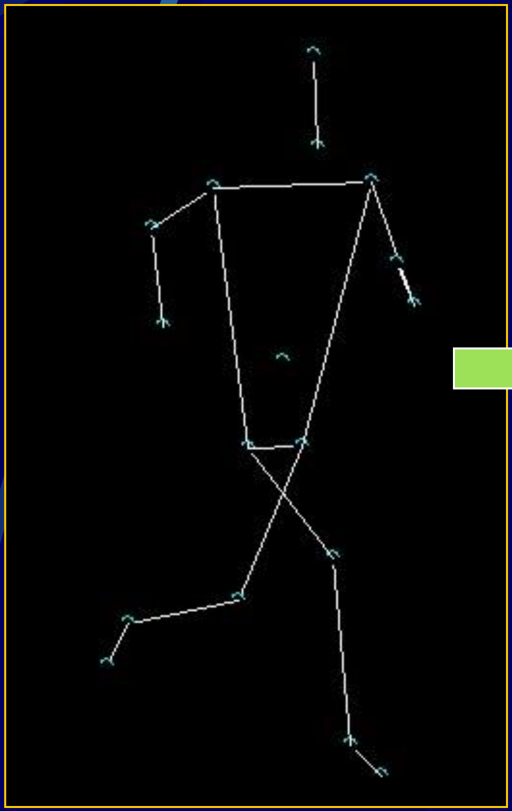
Rendering – a better 3D view

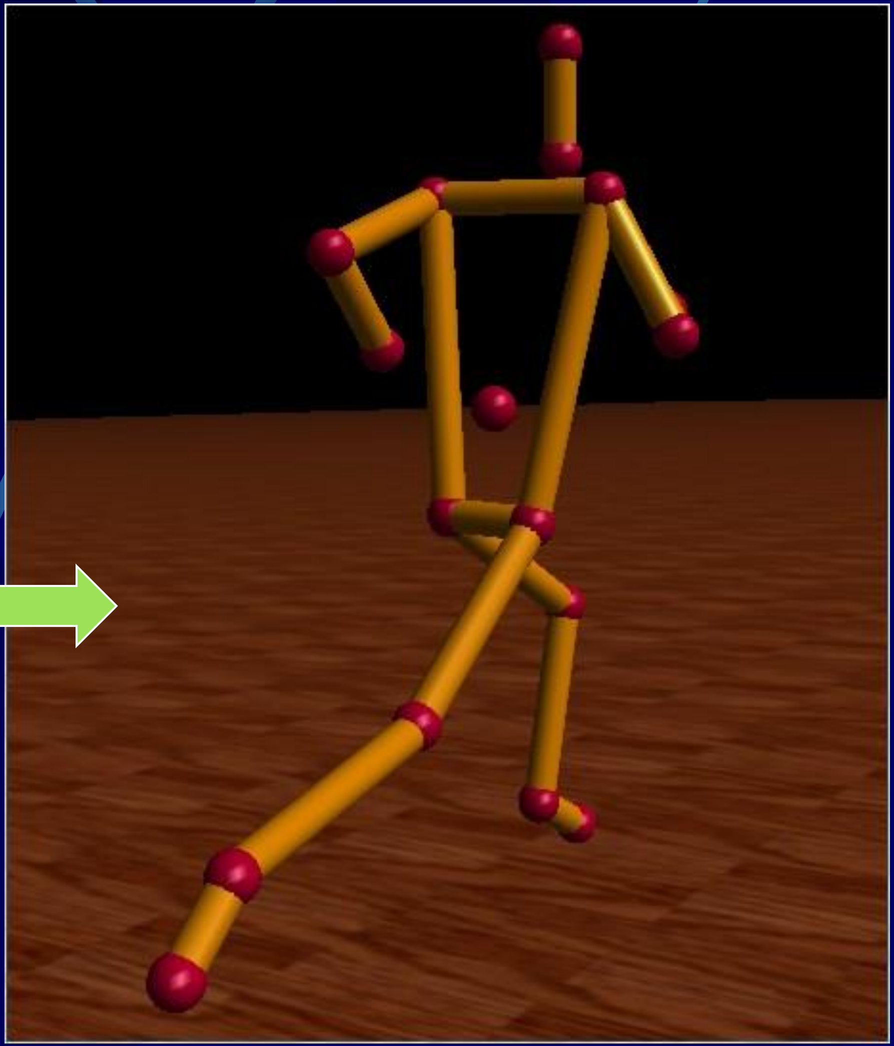


Rendering – in real-time

Q: Why now?

A: Cheap hardware for real-time rendering is available





What is “Rendering” anyway?

High level definition of:

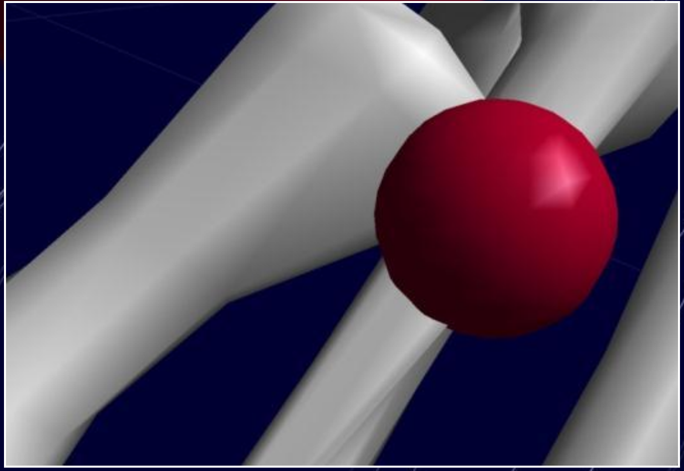
- Graphics objects
- Lighting
- Environmental effects
- Behavior
- Physical characteristics

Calculate views:

- Computer screens
- Immersive workbench
- Head-tracked glasses

Rendering - Performance

- Performance expressed in:
  - Fill Rate (pixels per second)
  - Rendering speed (triangles per second)
  - Smoothness (frames per second)
- Status PC Hardware (mid-2000):
  - Fill Rate > 1G pps
  - Rendering > 10M tps
  - Smoothness > 100 fps
- Prices
  - PC: < $1500
  - Graphics card: $75 - $500



Let's use … 6 degrees of freedom

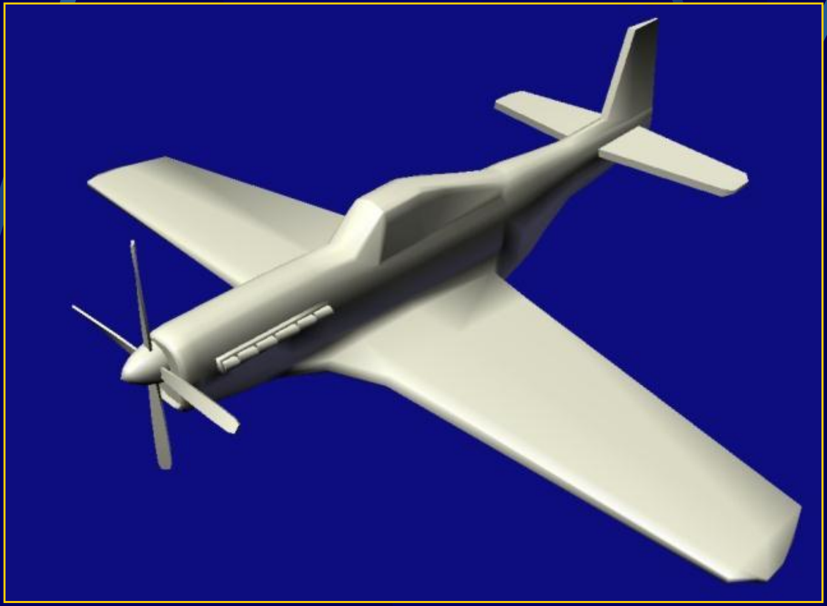
- So, let's just describe "graphic objects" and their behavior (e.g. their movement over time)
- This seems easy enough: we know we can uniquely describe an object's spatial orientation on any given instant with our familiar "6 degrees of freedom"
  - 3 for position: (x, y, z)
  - 3 for attitude, Euler or Cardan angles: (ϕ, θ, ψ)
- Advantages
  - No parameter redundancy
  - This seems intuitive
- **But …**

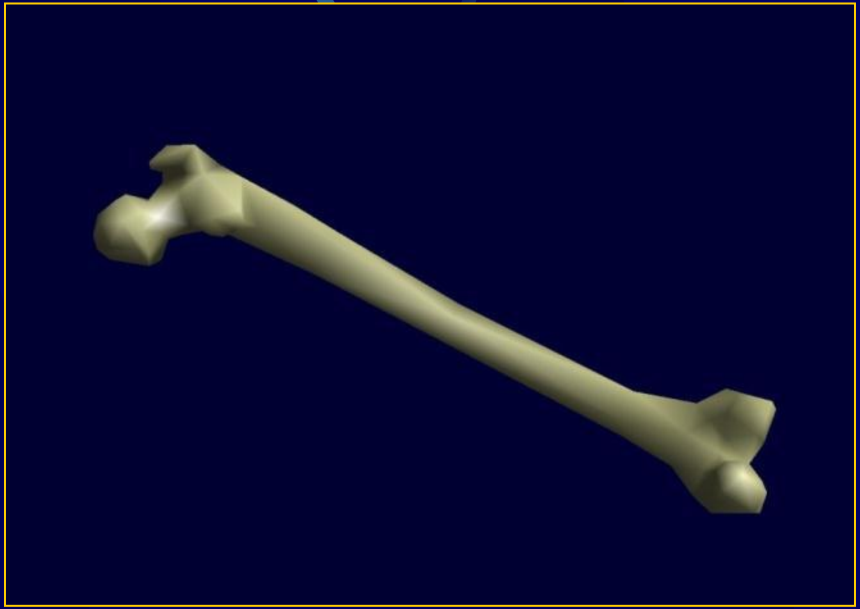
Euler Angles – Challenge 1

- There are 12 different sequences of rotations
  - XYZ; XZY; YXZ; YZX; ZXY; ZYX
  - XYX; XZX; YXY; YZY; ZXZ; ZYZ
- This is confusing
- It gets even worse when actually using them:
  - Operations are computationally expensive (involves many trigonometric calculations)
  - There are numerical instabilities (singularities)
  - Example: linear case velocity → position is easy. But when "integrating" angular velocity → Euler angles: which sequence is the right one???

Euler Angles – Challenge 2

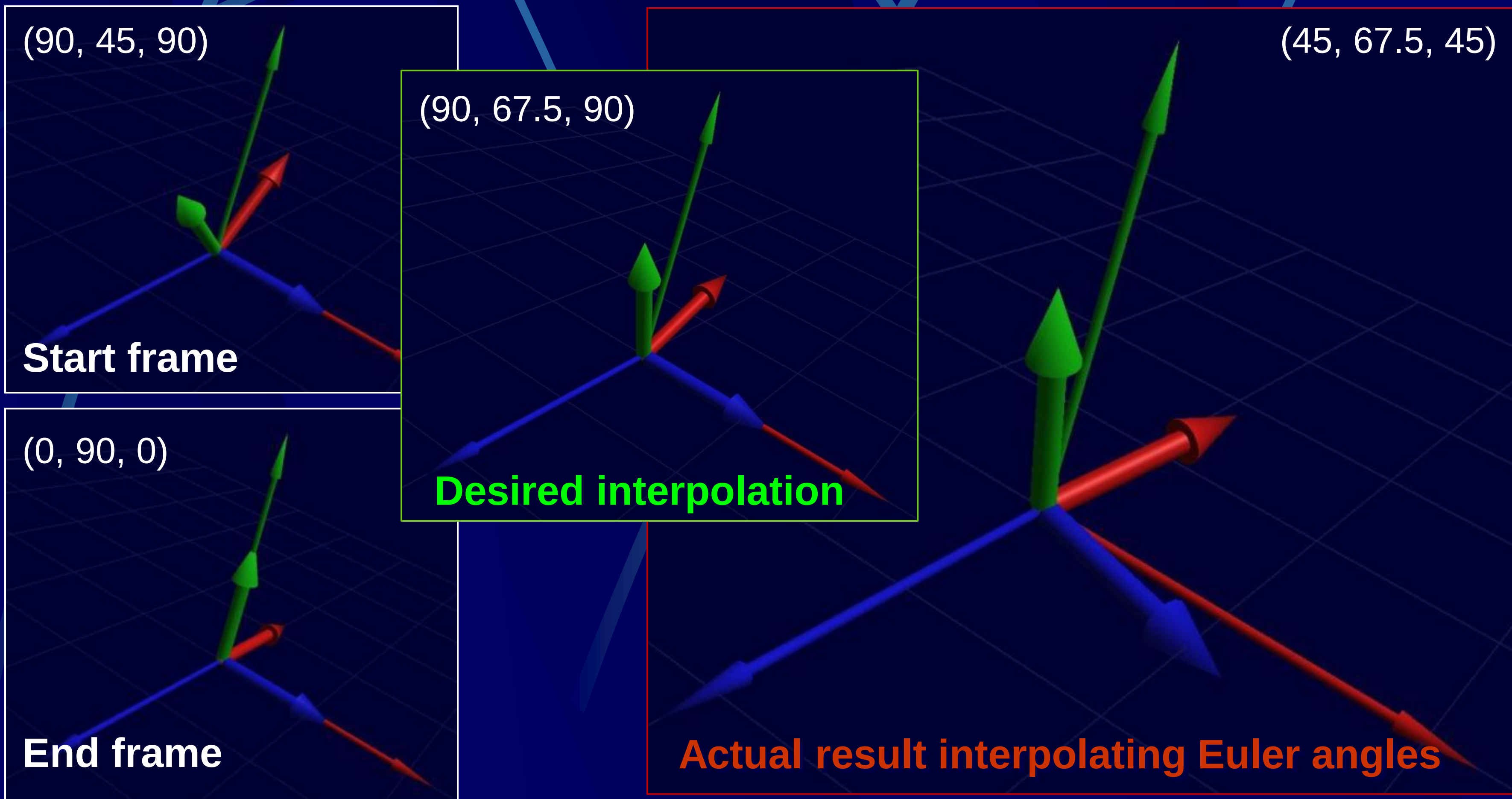
- Gimbal lock:
  - Loss of 1 degree of freedom
  - Axes of rotation seem to “stick”
  - A real-world problem, <u>not</u> a mathematical artifact



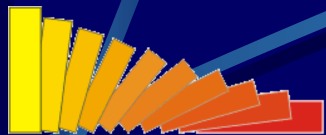


Euler Angles – Challenge 3

- Interpolation causes unnatural movement



Interpolation – why do I care?

1. 

Input: 10 key frames (sampled 1 second at 10 Hz)

2. 

Traditional “play back” is <u>jerky</u>
10 Hz (1 sec); 5 Hz (2 sec)
2 Hz (5 sec); 1 Hz (10 sec)

3. 

4. 

Rendering is always <u>smooth</u>
50 Hz (2 sec); 50 Hz (5 sec)
50 Hz (10 sec)

5. 

6. 

In fact, we get the same smooth results
with fewer key frames (data reduction)

Euler Angles - Alternatives

- Euler angles just have too many problems
- There must surely be better alternatives
  1. Rotation matrices
  2. Quaternions
  3. Euler's Axis – Angle representation
  4. Helical (screw) Axis
  5. Rodriguez-Hamilton parameters
  6. …
- Only the first two alternatives are used extensively in other disciplines

Alternative 1 – Rotation Matrices

- 9 or 16 Parameters
- Advantages
  - No singularities
  - Universally applicable
- Disadvantages
  - Obviously contains redundant information for storing orientation (9 parameters for 3 DOF, or 16 for 6 DOF)
  - Operations are computationally intensive
  - Rotational interpolation is difficult
  - Picks up non-rotational components in calculations: shearing, scaling, etc. → matrix must be renormalized

$$R = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix}$$

Alternative 2 - Quaternions

- Discovered by Sir Hamilton in 1843
- Preferred rotation operator in chemistry, robotics, space shuttle controls, 3D games, VR, etc…
- Advantages
  - Represents "pure" attitude or rotation
  - No mathematical singularities
  - Operations are computationally easy
  - Smooth and easy interpolation → <u>great</u> for animation
  - Impress your friends: rotate anything in 4D!
- Disadvantages
  - Completely unintuitive → fortunately not a real issue

Quaternions – what are they?

- An extension to complex numbers

$$q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$$

- Quaternion multiplication governed by:

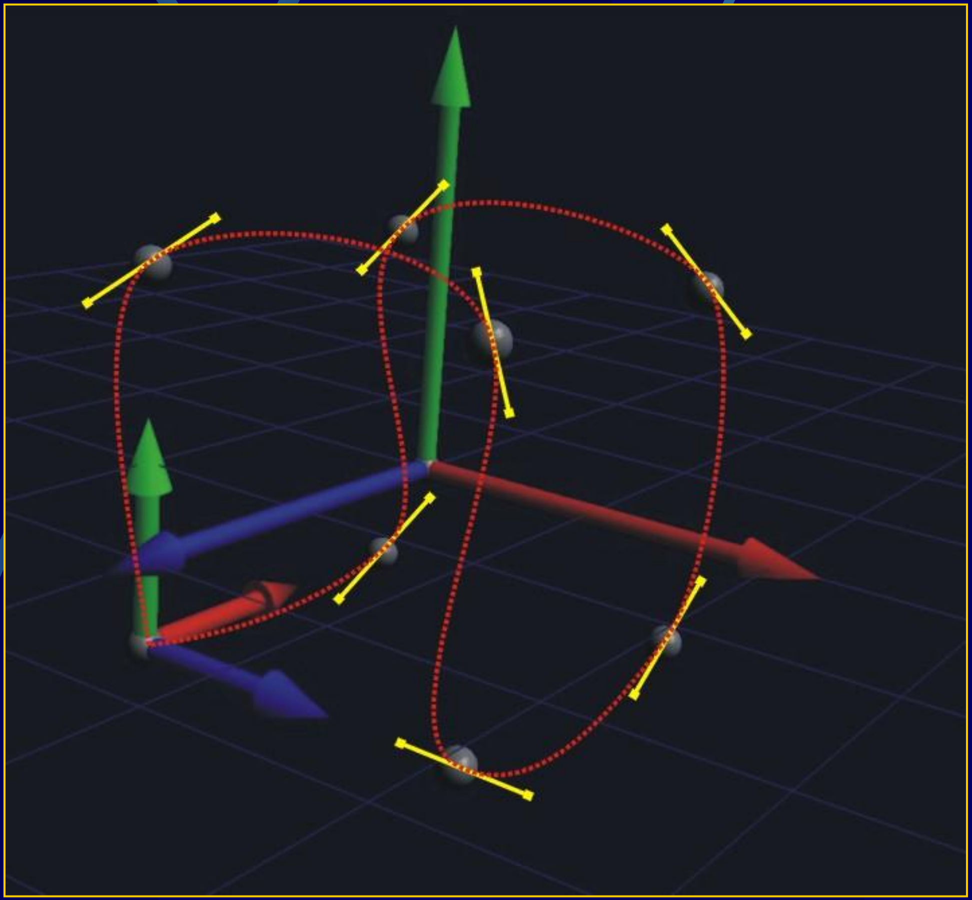
$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

$$\mathbf{ij} = -\mathbf{ji} = \mathbf{k} \qquad \mathbf{jk} = -\mathbf{kj} = \mathbf{i} \qquad \mathbf{ki} = -\mathbf{ik} = \mathbf{j}$$

- Operation $q\mathbf{v}q^{-1}$ rotates a vector $\mathbf{v}$ about the axis of q through twice the angle of q
- q^{-1} is easy to calculate, especially for unit quaternions
- Quaternion multiplication can be used to compose rotations: product q_1q_2 represents rotations q_1 and q_2.
- Quaternion powers can be used to iterate or divide rotations: this allows for smooth interpolation

Quaternions & Interpolation

- Quaternions are <u>very</u> suitable for animating attitude
  - Linear interpolation
  - SLERP interpolation
- <u>Equally</u> suitable for animating a camera (or more camera's)
  - Camera is just another object
- Position
  - Linear interpolation
  - Splines
- Other aspects
  - Scaling
  - Morphing
  - …



Conclusions

- Rendering is a great tool for visualizing 3D data
  - Gives more information than traditional stick-figure displays
  - Real-time interaction
  - Visually appealing
  - Hardware for high-performance rendering is cheap
- Quaternions are suitable for representing orientation
  - Quaternion manipulation is mathematically easy
  - No singularities or exceptions
  - Easy interpolation for smooth animation
- This presentation is online at:

www.sportsci.com/presentations/capetown