More Rotations; Visualization, Simulation
Videos 4: Virtual & Augmented Reality, Viz-Sim

William H. Hsu
Department of Computing and Information Sciences, KSU

Public mirror web site: http://www.ksolresearch.org/Courses/CIS536
Instructor home page: http://www.cis.ksu.edu/~bhsu

Readings:
Today: Chapter 10, 13, §17.3 – 17.5, Eberly
Next class: §2.4.3, 8.1, Eberly 2e, GL handout

Wikipedia, Visualization: http://bit.ly/gVxfRf

Where We Are

Review [1]: Representing 3 Rotational DOFs

3x3 Matrix (9 DOFs)
- Rows of matrix define orthogonal axes
Euler Angles (3 DOFs)
- Rot x + Rot y + Rot z
Axis-angle (4 DOFs)
- Axis of rotation + Rotation amount
Quaternion (4 DOFs)
- 4 dimensional complex numbers

Rotation about z axis
\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\theta) & -\sin(\theta) \\
0 & \sin(\theta) & \cos(\theta)
\end{bmatrix}
\]

Rotation about y axis
\[
\begin{bmatrix}
\cos(\theta) & 0 & \sin(\theta) \\
0 & 1 & 0 \\
-\sin(\theta) & 0 & \cos(\theta)
\end{bmatrix}
\]

Rotation about x axis
\[
\begin{bmatrix}
\cos(\theta) & -\sin(\theta) & 0 \\
\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Acknowledgements:
CGA Rotations, Dynamics & Kinematics

David C. Brogan
Visiting Assistant Professor, Computer Science Department, University of Virginia
http://www.cs.virginia.edu/~dbrogan/

Steve Rotenberg
Visiting Lecturer
University of California – San Diego
http://graphics.ucsd.edu

Rick Parent
Professor
Department of Computer Science and Engineering
Ohio State University
http://www.cse.ohio-state.edu/~parent/

Adapted from slides © 2000 – 2004 D. Brogan, University of Virginia

Adapted from slides © 2007 – 2011 R. Parent, Ohio State University

Review [2]: Method 1
Rotation Matrices – Roll, Pitch, & Yaw

Review [3]: Method 2
Rotation Matrices – Quaternions & Euler Angles
Review [3]: Method 2
Fixed Angles & Gimbal Lock

\[ P' = R_z(\beta)R_y(\gamma)R_x(\alpha)P \]

Fixed order: e.g., \( x, y, z \), also could be \( x, y, z \)

Global axes

Gimbal Lock Illustrated [1]
- Gimbal Lock: Loss of DOF when 2 of 3 Gimbals Driven until Parallel
- Animated Examples
  - \( \theta, \phi, \psi \) & \( \phi \) (left), \( \psi \) & \( \psi \) (right)
  - Caution: Seefeld (right) refers to these as \( \text{"x"} \) (red) & \( \text{"z"} \) (blue)
  - \( y \) (Pitch) = "x", \( x \) (Roll) = "y", \( z \) (Yaw) = "z" ("zed")

Review [4]: Method 3
Euler Angles & Order Independence

\( (\alpha, \beta, \gamma) = R_x(\alpha)R_y(\beta)R_z(\gamma) \)
- Rotate \( \alpha \) degrees about x-axis
- Rotate \( \beta \) degrees about y-axis
- Rotate \( \gamma \) degrees about z-axis

Axis order is not defined
- \( (y, z, x), (x, z, y), (z, y, x) \) are all legal
- Pick one

Review [5]: Euler Angle Sequences
- This means that we can represent an orientation with 3 numbers
- A sequence of rotations around principal axes is called an Euler Angle Sequence
- Assuming we limit ourselves to 3 rotations without successive rotations about the same axis, we could use any of the following 12 sequences:
  - XYZ
  - XZ
  - XZ
  - XYZ
  - ZXY
  - ZY
  - ZY
  - XYZ
  - YX
  - YX
  - YX

Using Euler Angles [1]: Representing Orientations
- This gives us \( 3! \times 3(2) \times 2 \times 3 \times 2 = 12 \) redundant ways to store an orientation using Euler angles
- Different industries use different conventions for handling Euler angles (or no conventions)
Using Euler Angles [2]: Conversion: Euler Angle to RM

- To build a matrix from a set of Euler angles, we just multiply a sequence of rotation matrices together:

\[
R_x R_y R_z = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & c_y & -s_y & 0 \\
0 & s_y & c_y & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{align*}
&c_x c_y c_z - s_x s_z \\
&c_x s_y c_z + s_x s_z \\
&-c_z s_y \\
&c_y \\
&c_y c_z s_x - s_y c_x \\
&c_y s_z s_x + c_x s_y \\
&-s_y s_z \\
&c_y
\end{align*}
\]

Review [6]: Method 4
Axis-Angle: Specification

- Given
  - \( r \) – vector in space to rotate
  - \( \mathbf{n} \) – unit-length axis in space about which to rotate
  - \( \theta \) – amount about \( \mathbf{n} \) to rotate

- Solve
  - \( r' \) – rotated vector

Review [7]: Method 5
Quaternions to RM, Axis-Angle

Quaternions [1]: Basic Idea

- Remember complex numbers: \( a + ib \)
  - Where \( i^2 = -1 \)
  - Invented by Sir William Hamilton (1843)
  - Remember Hamiltonian path from Discrete Math?

Quaternions:

- \( Q = a + bi + cj + dk \)
  - Where \( i^2 = j^2 = k^2 = -1 \) and \( ij = k \) and \( ji = -k \)
  - Represented as: \( q = (a, v) = s + v_i + v_j + v_k \)

Quaternions [2]: Definition

- A quaternion is a 4-D unit vector \( q = [x \ y \ z \ w] \)
  - It lies on the unit hypersphere \( x^2 + y^2 + z^2 + w^2 = 1 \)

For rotation about (unit) axis \( \mathbf{v} \) by angle \( \theta \)

- vector part = \( (\sin(\theta/2)) \mathbf{v} = [x \ y] \)
- scalar part = \( (\cos(\theta/2)) = w \)
- \( \sin(\theta/2) \) \( n_x \), \( \sin(\theta/2) \) \( n_y \), \( \sin(\theta/2) \) \( n_z \), \( \cos(\theta/2) \)

Only a unit quaternion encodes a rotation - normalize

Quaternions [3]: Equivalent RM & Composition

Rotation matrix corresponding to a quaternion:

\[
\begin{align*}
&[x \ y \ z] = \begin{bmatrix}
1 & -2yz - 2z^2 & 2xy - 2xz & 2xz - 2yz \\
2xy + 2xz & 1 - 2x^2 - 2z^2 & 2y^2 - 2xz \\
2xz + 2yz & 2y^2 - 2xz & 1 - 2x^2 - 2y^2
\end{bmatrix}
\end{align*}
\]

Quaternion Multiplication:

- \( q_1^{-1} q_2 = \{ [x_1 \ y_1] \} \{ [x_2 \ y_2] \} = \{ [x_1 x_2 - y_1 y_2] \} \}
- quaternion \( q_1 \) \( q_2 \) = quaternion
- this satisfies requirements for mathematical group
- Rotating object twice according to two different quaternions is equivalent to one rotation according to product of two quaternions
Quaternions [4]:
Examples

X-roll (roll) of π

- (cos(π/2), sin(π/2) (1, 0, 0)) = (0, 1, 0, 0)

Y-roll (pitch) of π

- (0, 0, 1, 0)

Z-roll (yaw) of π

- (0, 0, 0, 1)

R_z (π) followed by R_y (π)

- (0, 0, 1, 0) times (0, 0, 1, 0) = (0, 1, 0, 0)

= (0, 1, 0, 0)

Quaternions [5]:
Interpolation

Biggest advantage of quaternions

- Interpolation
- Cannot linearly interpolate between two quaternions because it would speed up in middle
- Instead, Spherical Linear Interpolation, slerp()
- Used by modern video games for third-person perspective
- Why?

Hint: see http://youtu.be/-jBKKV2V8eU

Quaternions [6]:
Spherical Linear Interpolation (SLERP)

Quaternion is a point on the 4-D unit sphere

- Interpolating rotations requires a unit quaternion at each step
- Another point on the 4-D unit sphere
- Move with constant angular velocity along the great circle between two points

Any rotation is defined by 2 quaternions, so pick the shortest SLERP

To interpolate more than two points, solve a non-linear variational constrained optimization

- Ken Shoemake in SIGGRAPH 85 (www.acm.org/dli)

Quaternions [7]:
Comparison with Euler Interpolation

Quaternion (white) vs.
Euler (black) interpolation

Left images are linear interpolation
Right images are cubic interpolation

Quaternions [8]:
Code

- Nate Robins’ Implementation: http://bit.ly/1 mQdev

- File glib.c
- glibMatrix
- glibMotion

Spherical Interpolation [1]:
Spheres

Think of a person standing on the surface of a big sphere (like a planet)

- From the person’s point of view, they can move in along two orthogonal axes (front/back) and (left/right)
- There is no perception of any fixed poles or longitude/latitude, because no matter which direction they face, they always have two orthogonal ways to go
- From their point of view, they might as well be moving on an infinite 2D plane, however if they go too far in one direction, they will come back to where they started!
Spherical Interpolation [2]: Hyperspheres

- Now extend concept to moving in hypersphere of unit quaternions
- Now have three orthogonal directions to go
- No matter how oriented in this space, can always go some combination of forward/backward, left/right and up/down
- Go too far in any direction: back to start point
- Location on unit hypersphere: orientation
- Moving in arbitrary direction corresponds to rotating around some arbitrary axis

Visualization [1]: Animating Simulations

- Virtual Reality: Computer-Simulated Environments
  - Physical Presence: Real & Imaginary
  - Hardware: User Interface
    - Head-mounted display (HMD), gloves – see PopOptics goggles (left)
    - VR glasses, wand, etc. – see NCSA CAVE (right)

Visualization [2]: Virtual Reality (VR)

- Dynamics & Kinematics
  - Kinematics: Study of Motion without Regard to Causative Forces
    - Forward: from angles to position
    - Inverse: finding angles given desired position
  - Dynamics: Study of Motion & Changes in Motion
    - Forward: model forces over time to find state
    - Inverse: given state and constraints, calculate forces
  - Modeling systems – e.g., articulated figure
  - VR glasses, wand, mounted
  - Forward: from angles to position
  - Inverse: finding angles given desired position

Visualization [3]: Virtual Environments (VE)

- Virtual Environment: Part of Virtual Reality Experience
  - Other Parts

Visualization [4]: Augmented Reality (AR)

- Augmented Reality: Computer-Generated (CG) Sensory Overlay
  - Added to Physical, Real-World Environment

Dynamics: Study of Motion & Changes in Motion
- Given: initial position \( p_0 \), velocity \( v_0 \), gravitational constants
- Calculate: position \( p \) at time \( t \)
- Inverse: given state and constraints, calculate forces
- Given: desired position \( p \), at time \( t \), gravitational constants
- Calculate: position \( p_0 \), velocity \( v_0 \), read

- For non-particle objects: rigid-body dynamics (http://bit.ly/yf3e3g)

Kinematics: Study of Motion without Regard to Causative Forces
- Modeling systems – e.g., articulated figure
Summary

- Reading for Last Class: §17.1 – 17.2, Eberly 2nd
- Reading for Today: Chapter 10, 13, §17.3 – 17.5, Eberly 2nd
- Reading for Next Class: §2.4.3, 8.1, Eberly 2nd, GL handout
- Last Time: Rotations in Animation
  - Matrix, fixed angles, Euler angles, axis
  - Quaternions & how they work – properties, arithmetic operations
  - Gimbal lock defined & illustrated
- Quaternions Concluded
  - Incremental rotation: spherical linear interpolation (slerping)
  - Advantages of slerping vs. cubic interpolation between Euler angles
  - Uses: character animation, camera control (rotating Look vector)
- Dynamics & Kinematics (Preview of Lectures 28 – 30)
- Today: Modeling & Simulation
  - Virtual / augmented reality (VR/AR) & virtual environments (VE)
  - Visualization & simulation (Viz-Sim) preview

Terminology

- Last Time: Rotation using Matrices, Fixed Angles, Euler Angles
- Gimbal Lock
  - Loss of DOF
- Axis-Angle – Rotate Reference Vector \( r \) about Arbitrary Axis \( A/n \)
- Quaternions
  - Quaternions – different representation of arbitrary rotation
  - Exponential maps – 3-D representation related to quaternions
  - Visualization – Communicating with Images, Diagrams, Animations
  - Simulation – Artificial Model of Real Process for Answering Questions
  - VR, VE, VA, AR
    - Virtual Reality: computer-simulated environments, objects
    - Virtual Environment: part of VR dealing with surroundings
    - Virtual Artifacts: part of VR dealing with simulated objects
    - Augmented Reality: CG sensory overlay on real-world images