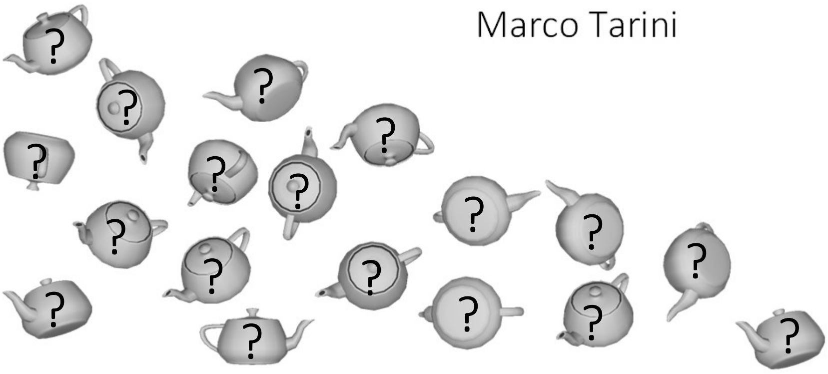
3D Videogames

Rotations: exercises

Marco Tarini



99

Course Plan

lec. 1: Introduction
lec. 2: Mathematics for 3D Games
lec. 3: Scene Graph
lec. 4: Game 3D Physics +
lec. 5: Game Particle Systems
lec. 6: Game 3D Models
lec. 7: Game Materials
lec. 8: Game Textures
lec. 9: Game 3D Animations
lec. 10: Audio for 3D Games
lec. 11: Networking for 3D Games
lec. 12: Interactive Agents for 3D Games
lec. 13: Rendering Techniques for 3D Games

100

2D rotations as 3D rotations

- A 2D rotation (of an angle α , around the origin) can be seen as the *restriction* of a 3D rotation in the X-Y plane (of an angle α , around the... Z axis!)
- Find this 3D rotation in *all* representations:
  - as... a 3x3 Matrix: $\begin{bmatrix} +\cos(\alpha) & -\sin(\alpha) & 0 \\ +\sin(\alpha) & +\cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
  - as... Axis-times-Angle: $[0,0,\alpha]$
  - as... Euler angles (Roll=Z, Pitch=X, Yaw=Y): $[\alpha,0,0]$
  - as... a quaternion: $\left[0,0,\sin\left(\frac{\alpha}{2}\right),\cos\left(\frac{\alpha}{2}\right)\right]$

106

Exercises: *find the rotation that...*

- For all the following exercises:
  we can pick any rotation representation!
  (unless otherwise specified)
  - As long as we have algorithms to translate one representation into another
  - For each task, try to identify which format is the most convenient to use!

107

Find the «from-to» rotation

- Given a pair of versors $\hat{v}$ and $\widehat{w}$, ($\hat{v}$ = *"from"* and $\widehat{w}$ = *"to"*) find the minimal rotation that brings $\hat{v}$ into $\widehat{w}$
  - applications in several contexts
  - i.e., find minimal rotation R such that $R(\hat{v}) = \widehat{w}$
- A solution as axis-and-angle:
  - the axis a is found as $\hat{v} \times \widehat{w}$ (renormalizing it)
  - about the angle α , we know that its cosine is $(\hat{v} \cdot \widehat{w})$ and its sine is $|| \hat{v} \times \widehat{w} ||$, so $\alpha = \text{atan2}(|| \hat{v} \times \widehat{w} || \, , \, \hat{v} \cdot \widehat{w})$

with minimal angle

e.g. NPC aiming a bazooka

109

Find the «look-at» rotation

- Given observer's position $\mathbf{e}$ and observed point $\mathbf{t}$
  find the rotation (i.e., the orientation)
  for a character who must be looking in that direction
- That specification is incomplete:
  we also need another input: a «target up-vector» $\hat{\boldsymbol{u}}$
  - the character wants to keep its up-direction as similar as possible to $\hat{\boldsymbol{u}}$, while looking toward $\mathbf{t}$
  - Usually, the (world) up-vector, e.g. (in Unity) (0,1,0)
- Useful for… characters heads looking at something / facing toward something, setting up the camera…

110

Find the «look-at» rotation

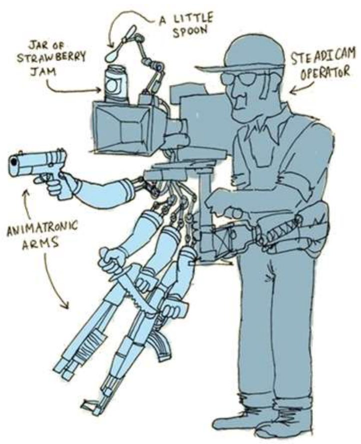
- Solution: as a 3×3 matrix
  - find the $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{z}}$ directions of this local character
  - they must be 3 reciprocally orthogonal versors
  - they are the columns of the sought matrix
- that is (assuming Unity conventional axis names):
  - $\hat{\boldsymbol{z}} = (\mathbf{t} - \mathbf{e}) / ||\mathbf{t} - \mathbf{e}||$
  - $\hat{\boldsymbol{y}} = \hat{\boldsymbol{u}}$? Wrong: it wouldn't be necessarily orthogonal with $\hat{\boldsymbol{z}}$
  - but, $\hat{\boldsymbol{x}} = \hat{\boldsymbol{u}} \times \hat{\boldsymbol{z}} \ / \ || \hat{\boldsymbol{u}} \times \hat{\boldsymbol{z}} ||$ (note the re-normalization)
    because the right direction is orthogonal to both $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{u}}$
  - finally, $\hat{\boldsymbol{y}} = \hat{\boldsymbol{z}} \times \hat{\boldsymbol{x}}$
    (note: re-normalization isn't needed here)
    (note the ordering, in both passages: $\hat{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{z}}$)

111

What about the "look-at" complete transform

- Setting up the complete transform of a camera (from the same data):
  - Camera position: is the translation component
  - the "look-at" rotation : is the rotation component
  - (scale component = 1)

In Computer Vision
the set of these parameters are
defined as the camera
extrinsic parameters



"Camera-man in videogame logic"
unknown artist, circa 2010

113

Update the orientation of a rolling ball *

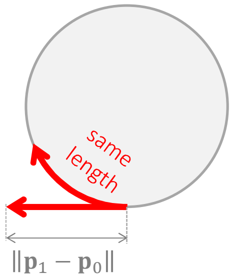
- A ball with radius r stands on a flat plane (with plane normal $\hat{\mathrm{n}}$), and it's currently oriented with rotation R_0 and positioned (center position) in $\mathbf{p}_0$
- It then rolls in position $\mathbf{p}_1$ (staying on the plane)
- Find its new orientation R_1

Marble Madness, Atari, 1986

* a classic of many 3D games! Including early "3D" games

114

Update the orientation of a rolling ball *



Solution (trace): as axis-angle…

- The axis must be:
  - parallel to the ground; therefore, orthogonal to $\hat{\mathrm{n}}$!
  - orthogonal to the direction of motion $(\mathbf{p}_1 - \mathbf{p}_0)$
  - (also, it must be expressed as a unit vector)
- The angle α must satisfy…
  full-circumference : length-of-arch = full-circle : α
  ($2\pi r$; $\|\mathbf{p}_1 - \mathbf{p}_0\|$; 2π radiants, or 360°)

116

Update the orientation of a rolling ball *

- Once we found the new motion R_N (rotation as a motion, a spin), in must be cumulated with the old state R_0 (rotation as a state, an orientation)

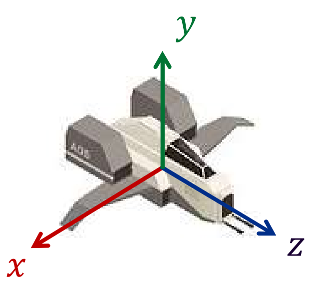
$$\mathrm{R}_0 \longleftarrow \mathrm{R}_N \circ \mathrm{R}_0$$

- To do so, it's more convenient to convert both into a format that allows for easy cumulation
  - Like quaternions
  - How to do so? See exercises at the end

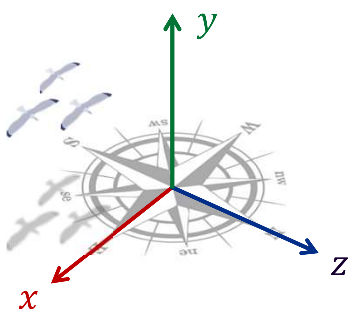
117

Find the orientation of a spaceship/airplane "character"

Local Space



Global Space



118

Find the orientation of a spaceship/airplane "character"

- Find the orientation R_P of an airplane at spawn time
  - The airplane is going NNE, and climbing up at 30° angle.
  - Its wing lines is parallel to the ground.
- Local space of airplane:
  - X-axis: left-right (the direction of the wings)
  - Y-axis: below to above
  - Z-axis: engine-to-propeller
- World space:
  - X-axis: west to east
  - Y-axis: ground to sky
  - Z-axis: south to north

NNE = halfway between North and NE

(which handedness is world and local spaces?)

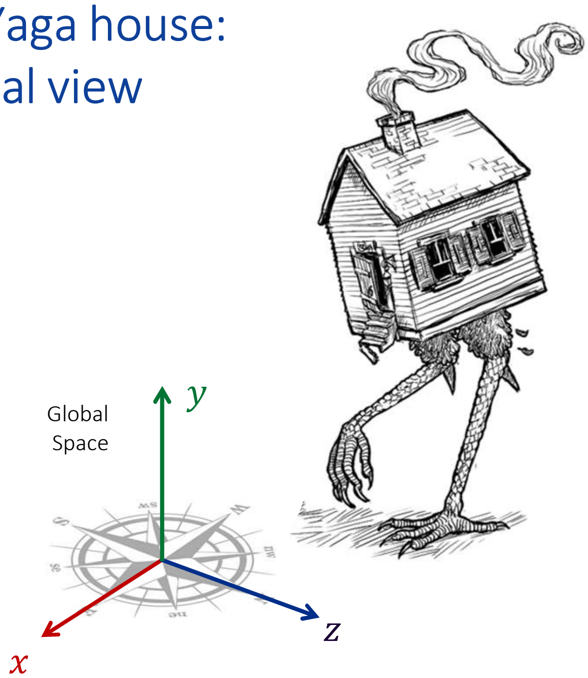
119

Find the orientation of the head of the pilot of previous exercise

- The head of the pilot inside that plane is tilted 20° to the left, and 10° degrees above
- What it is its orientation R_H?
- Local space of the head:
  - X-axis: left-eye to right-eye
  - Y-axis: chin to top of the head
  - Z-axis: view direction

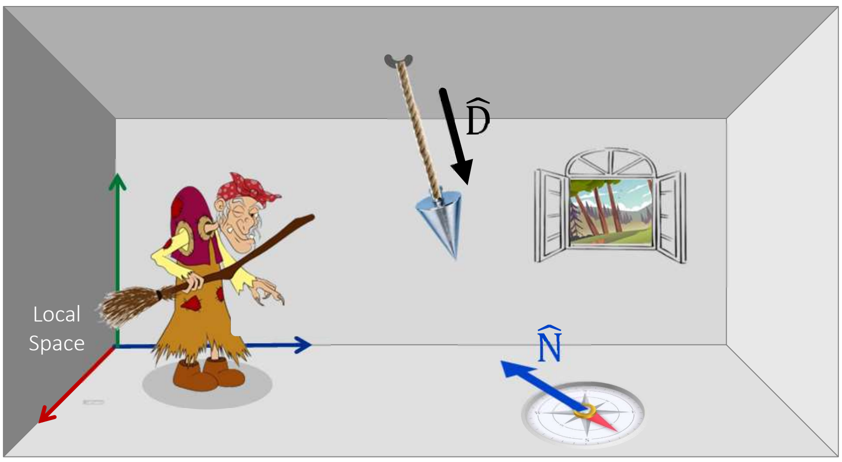
120

Baba Yaga house: external view



122

Baba Yaga house: internal view



123

Baba Yaga house
Which rotation is associated to the house?

- The witch Baba Yaga lives in a moving house on chicken legs
- The house is now still, with some unknown orientation.
- Inside the house, Baba Yaga
  keeps a compass (ita: *"una bussola"*)
  and a plumb line (ita: *"un filo a piombo"*)
- Knowing the current direction $\hat{\mathrm{D}}$ of the plumb line
  (which always points downward)
  and the direction $\hat{\mathrm{N}}$ of the compass needle
  (parallel to the house floor, and always pointing North)
  can you express the orientation of the house? (in any form?)
- Note: Baba Yaga measures $\hat{\mathrm{D}}$ and $\hat{\mathrm{N}}$ as versors in *house space*
  (the local space of the house)
  (where, for example, the axis Y is defined as orthogonal to the floor)

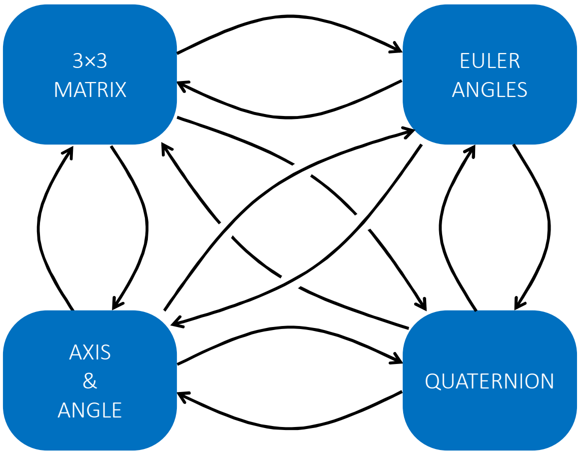
124

Baba Yaga house
Which rotation is associated to the house?

- Solution trace: as a 3×3 matrix
  - find the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}$ directions of GLOBAL space…
    expressed in LOCAL SPACE
  - they are the rows of the sought matrix (understand this!)
- that is:
  - $\hat{\mathbf{y}} = -\hat{\mathrm{D}}$ (the world "up" dir is the opposite of $\hat{\mathrm{D}}$)
  - $\hat{\mathbf{z}} = \hat{\mathbf{N}}$? Wrong: the compass needle is not exactly the world "north": it is constrained to be parallel to the (oblique?) floor!
  - but, the world "east" is orthogonal to both $\hat{\mathbf{N}}$ and $\hat{\mathbf{y}}$
    $\hat{\mathbf{x}} = \hat{\mathbf{y}} \times \hat{\mathbf{N}} \,/\, \| \hat{\mathbf{y}} \times \hat{\mathbf{N}} \|$ (why the re-normalization?)
  - finally, $\hat{\boldsymbol{x}} = \hat{\boldsymbol{y}} \times \hat{\boldsymbol{z}}$ (note: re-normalization isn't needed here)
    (note the ordering, in both passages: $\hat{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{z}}$)

125

Converting between rotation representations: many exercises!



Every representation is useful in some circumstance.

In a game engine or lib, conversions are useful.

Not all arrows need to be implemented!

Just ensure there is a path from anything to anything.

Try as many "arrows" as you can!

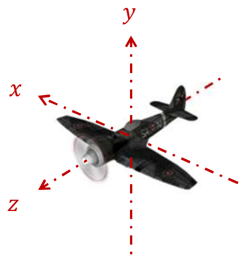
126

From: axis-&-angle
To: quaternion, or viceversa

- Trivial exercise. Observations:
  - When going from an angle-based representation (*Euler angles*, *Axis-&-Angle*) to a non-angle-based representation (*Matrix*, *Quaternion*) you'll need trigonometric functions (sin , cos , ...)
  - When going from a non-angle-based representation (*Euler angles*, *Axis-&-Angle*) to an angle-based representation (*Matrix*, *quaternion*) you'll need inverse trigonometric functions (asin , acos , atan2) ← Remember this convenient one exists!

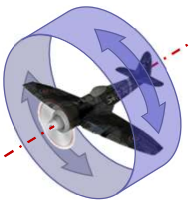
127

from: Euler angles
to: 3x3 matrix

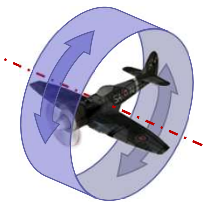


Assume "Unity" axis definitions

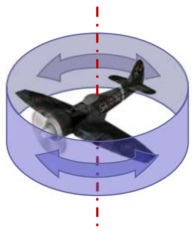
- Question:
  - Which matrix $\mathbf{R}$ does this?



Roll α
(1st)



Pitch β
(2nd)



Yaw γ
(3rd)

128

from: Euler angles
to: 3x3 matrix

the order is prescribed by the choice of Euler Angles

$$\mathrm{R} = \mathrm{R_y}(\gamma) \cdot \mathrm{R_x}(\beta) \cdot \mathrm{R_z}(\alpha)$$

$$\begin{bmatrix} +\cos(\beta) & 0 & +\sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & +\cos(\beta) \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & +\cos(\alpha) & -\sin(\alpha) \\ 0 & +\sin(\alpha) & +\cos(\alpha) \end{bmatrix} \quad \begin{bmatrix} +\cos(\alpha) & -\sin(\alpha) & 0 \\ +\sin(\alpha) & +\cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

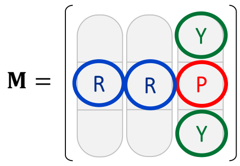
See: rotations in 2D

- What about the vice-versa?
  - a more difficult exercise
  - requires inverse trigonometric functions (of course)

129

from: 3x3 matrix to: Euler Angles

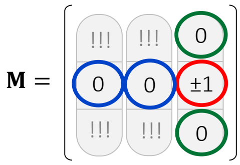
Assuming "Unity" axis definitions



A trace:

- Yaw angle:
  - it's given by the moving direction of the airplane "on the map" (N-S-E-W)
  - i.e. the *(x,z)* projection on the horizontal world flat plane of the forward vector!
  - Remember atan2 exists
  - Special case when it's (0,0)! (gimbal lock!)
- Pitch angle:
  - its sine is the *y* component of the forward vector!
  - It's between -90° and +90°, so that suffices
  - Note: when *y* is ±1 => gimbal lock!
- Roll angle:
  - its sine is proportional to the *y* component of the right vector (see why?)
  - its cosine is proportional to the *y* component of the up vector (see why?)
  - Special case when it's (0,0)! (gimbal lock!)

gimbal lock:



130

from: Euler angles - to: Quaternion

- Which quaternion encodes the rotation by Euler Angles (ROLL, PITCH, YAW) = $(\alpha_R, \alpha_P, \alpha_Y)$?

Rotations:

name	Axis	Order	Axis & Angle	As quaternion		
ROLL	Z	1st	$\begin{pmatrix}0\\0\\1\end{pmatrix}, \alpha_R$	$s_R\ k + c_R$	with	$s_R = \sin\left(\frac{\alpha_R}{2}\right)$, $c_R = \cos\left(\frac{\alpha_R}{2}\right)$
PITCH	X	2nd	$\begin{pmatrix}1\\0\\0\end{pmatrix}, \alpha_P$	$s_P\ i + c_P$	with	$s_P = \sin\left(\frac{\alpha_P}{2}\right)$, $c_P = \cos\left(\frac{\alpha_P}{2}\right)$
YAW	Y	3rd	$\begin{pmatrix}0\\1\\0\end{pmatrix}, \alpha_Y$	$s_Y\ j + c_Y$	with	$s_Y = \sin\left(\frac{\alpha_Y}{2}\right)$, $c_Y = \cos\left(\frac{\alpha_Y}{2}\right)$

Answer: $(s_Y\ j + c_Y)(s_P\ i + c_P)(s_R\ k + c_R) =$

$(s_Y\ j + c_Y)(-s_P\ s_R\ j + s_P c_R\ i + c_P\ s_R\ k + c_P c_R) =$

$\dots$

131

from: 3x3 matrix to: axis-&-angle

- Question:
  - Given a rotation matrix R ,
    find axis $\hat{a}$ and rotation angle α
  - Assumption: R is actually a rotation matrix
- Trace:
  1. Observation: for the given matrix R ,
     $\mathrm{R}\,\hat{a} = \hat{a}$ (why?)
  2. In other words,
     $\hat{a}$ is an eigenvector of R of eigenvalue 1
  3. Find α : remember atan2 exists

132

from: axis-&-angle to: 3x3 matrix

- Question:
  - Which matrix R rotates by α degrees around axis $\hat{a}$?
- Trace:
  1. Find *any* rotation
     matrix R_A mapping $\hat{a}$ the axis into the X axis
     (hint: find three orthogonal versors to use as columns of R_A , one of them being $\hat{a}$)
  2. Define
     a rotation matrix R_x rotating by α around X axis
  3. Then: $\mathrm{R} = \mathrm{R}_A^{-1} \cdot \mathrm{R}_x \cdot \mathrm{R}_A$ (understand why)

133

from: Quaternion to: 3x3 Matrix

- Which matrix $\mathbf{M}$ encodes the same rotation as quaternion $\mathbf{q} = (a\,i + b\,j + c\,k + d)$?
- Trace: let's find out the three columns of $\mathbf{M}$!

$\mathbf{M} =$

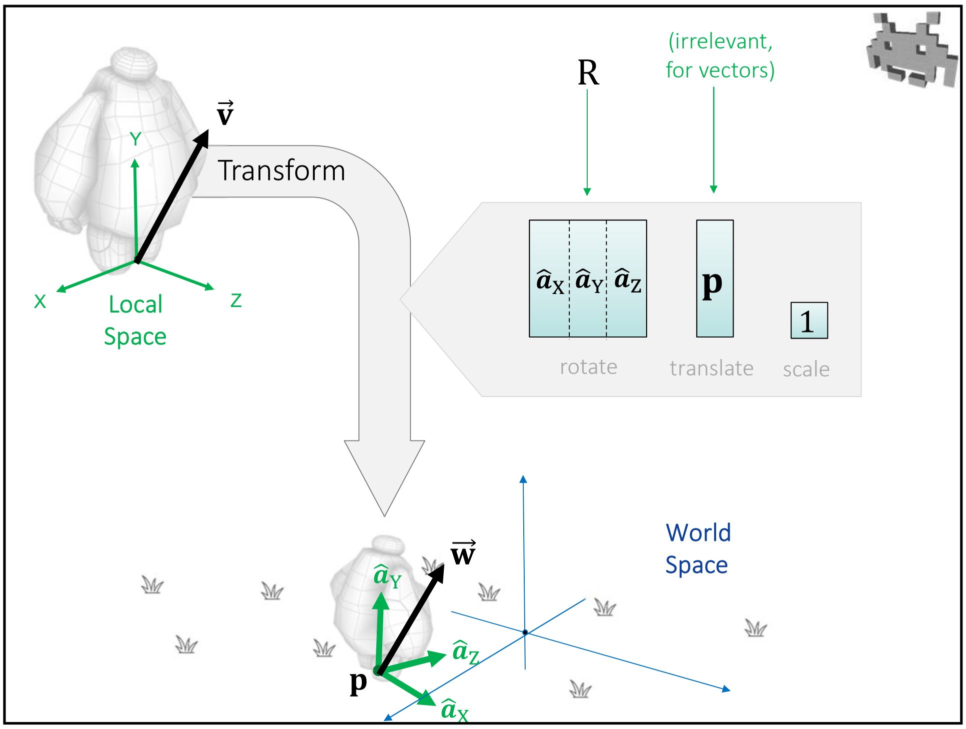
Axis (versors)	In local space		In global Space	
	in cartesian coords	as quaternion	as quaternion	in cartesian coords
X	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	i	$\mathbf{q}\,i\,\bar{\mathbf{q}} = \ldots$	$\begin{pmatrix}?\\?\\?\end{pmatrix}$
Y	$\begin{pmatrix}0\\1\\0\end{pmatrix}$	j	$\mathbf{q}\,j\,\bar{\mathbf{q}} = \ldots$	$\begin{pmatrix}?\\?\\?\end{pmatrix}$
Z	$\begin{pmatrix}0\\0\\1\end{pmatrix}$	k	$\mathbf{q}\,k\,\bar{\mathbf{q}} = \ldots$	$\begin{pmatrix}?\\?\\?\end{pmatrix}$

135

A "trivial" final exercise: space transformation of vectors

- Let $\vec{\mathbf{v}}$ be a vector defined in local space
- Let $\vec{\mathbf{w}}$ be the corresponding vector in world space
- Let $\hat{\boldsymbol{a}}_X\ \hat{\boldsymbol{a}}_Y\ \hat{\boldsymbol{a}}_Z$ be the three versors describing the three object-space axis (expressed in world space)
  - assume the transformation has scaling = 1
- Problem 1: given $\vec{\mathbf{v}}$, find $\vec{\mathbf{w}}$
- Problem 2: given $\vec{\mathbf{w}}$, find $\vec{\mathbf{v}}$
- Solutions: trivial, right?
  - The rotation of the transform is given by the matrix $\mathrm{R} = [\hat{\boldsymbol{a}}_X \mid \hat{\boldsymbol{a}}_Y \mid \hat{\boldsymbol{a}}_Z]$
  - Then, by definition, $\vec{\mathbf{w}} = \mathrm{R}\,\vec{\mathbf{v}}$ and $\vec{\mathbf{v}} = \mathrm{R}^{-1}\,\vec{\mathbf{w}}$
- However, the task is to address both problems using only geometric intuition, and the algebra of point, vector, versors…
  - without any concept of "rotation"

136



137

Solution (trace)

$$\vec{\mathbf{v}} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \qquad \vec{\mathbf{w}} = \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix}$$

- Problem 1: given $\vec{\mathbf{v}}$, find $\vec{\mathbf{w}}$
  - See how $\vec{\mathbf{w}}$ can be found as a linear combination of $\hat{\boldsymbol{a}}_X$, $\hat{\boldsymbol{a}}_Y$, $\hat{\boldsymbol{a}}_Z$
  - ...with linear weights v_x, v_y, v_z (the coordinates of $\vec{\mathbf{v}}$)
  - *Bonus:* rewrite that expression in matrix form...
    (that is, using $\vec{\mathbf{v}}$, not v_x, v_y, v_z)
- Problem 2: given $\vec{\mathbf{w}}$, find $\vec{\mathbf{v}}$
  - See how each of v_x, v_y, v_z (the coordinates of $\vec{\mathbf{v}}$)
    can be found as a dot product (...with $\hat{\boldsymbol{a}}_X$, $\hat{\boldsymbol{a}}_Y$, $\hat{\boldsymbol{a}}_Z$)
  - [note: this only works because $\hat{\boldsymbol{a}}_X$, $\hat{\boldsymbol{a}}_Y$, $\hat{\boldsymbol{a}}_Z$ are unitary and orthogonal]
  - *Bonus:* rewrite that expression in matrix form...
    (that is, using $\vec{\mathbf{v}}$, not v_x, v_y, v_z)
- ...you rediscovered that R and R^T are the inverse of each other!
  - as they are the ways to solve two *inverse* geometrical problems

138