CS-184: Computer Graphics

Lecture #5: 3D Transformations and Rotations

Prof. James O'Brien
University of California, Berkeley

V2013-S-05-1.0

1

Today

- Transformations in 3D
- Rotations
  - Matrices
  - Euler angles
  - Exponential maps
  - Quaternions

2

Tuesday, February 5, 13

3D Transformations

- Generally, the extension from 2D to 3D is straightforward
  - Vectors get longer by one
  - Matrices get extra column and row
  - SVD still works the same way
  - Scale, Translation, and Shear all basically the same
- Rotations get interesting

3

3

Translations

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$ For 2D

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ For 3D

4

4

Scales

$$\tilde{\mathbf{A}} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For 2D

$$\tilde{\mathbf{A}} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For 3D

(Axis-aligned!)

5

5

Shears

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & h_{xy} & 0 \\ h_{yx} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For 2D

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & h_{xy} & h_{xz} & 0 \\ h_{yx} & 1 & h_{yz} & 0 \\ h_{zx} & h_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For 3D

(Axis-aligned!)

6

6

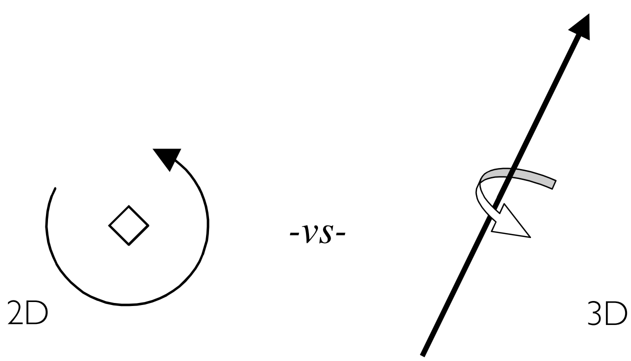
Shears

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & h_{xy} & h_{xz} & 0 \\ h_{yx} & 1 & h_{yz} & 0 \\ h_{zx} & h_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shears y into x

Rotations

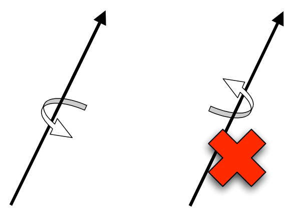
- 3D Rotations fundamentally more complex than in 2D
  - 2D: amount of rotation
  - 3D: amount and axis of rotation



Rotations

- Rotations still orthonormal
- $\mathrm{Det}(\mathbf{R}) = 1 \neq -1$
- Preserve lengths and distance to origin
- 3D rotations DO NOT COMMUTE!
- Right-hand rule
- Unique matrices

DO NOT COMMUTE!

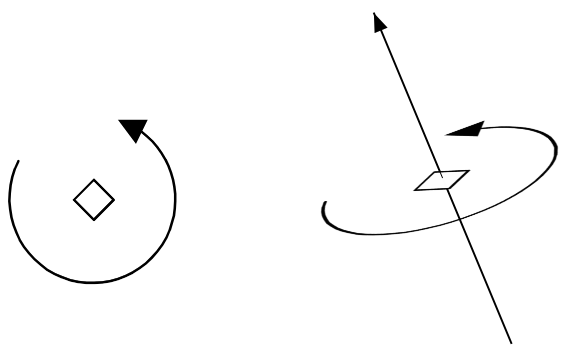


9

9

Axis-aligned 3D Rotations

- 2D rotations implicitly rotate about a third out of plane axis



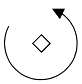
10

10

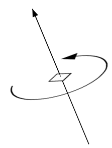
Axis-aligned 3D Rotations

- 2D rotations implicitly rotate about a third out of plane axis

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Note: looks same as $\tilde{\mathbf{R}}$



11

11

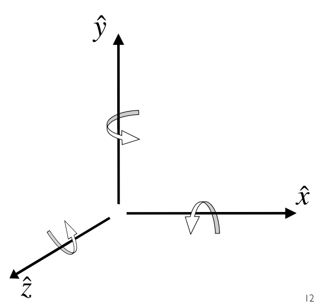
Axis-aligned 3D Rotations

$$\mathbf{R}_{\hat{x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$\mathbf{R}_{\hat{y}} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$\mathbf{R}_{\hat{z}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

"Z is in your face"



12

12

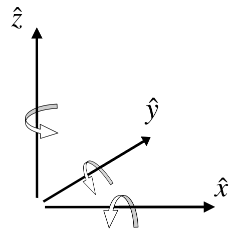
Axis-aligned 3D Rotations

$$\mathbf{R}_{\hat{x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$\mathbf{R}_{\hat{y}} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$\mathbf{R}_{\hat{z}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Also right handed "Zup"



Axis-aligned 3D Rotations

- Also known as "direction-cosine" matrices

$$\mathbf{R}_{\hat{x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \qquad \mathbf{R}_{\hat{y}} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

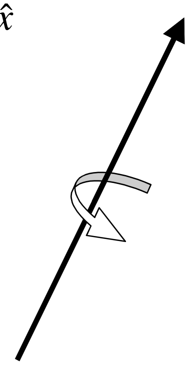
$$\mathbf{R}_{\hat{z}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Arbitrary Rotations

- Can be built from axis-aligned matrices:

$$\mathbf{R} = \mathbf{R}_{\hat{z}} \cdot \mathbf{R}_{\hat{y}} \cdot \mathbf{R}_{\hat{x}}$$

- Result due to Euler... hence called Euler Angles
- Easy to store in vector

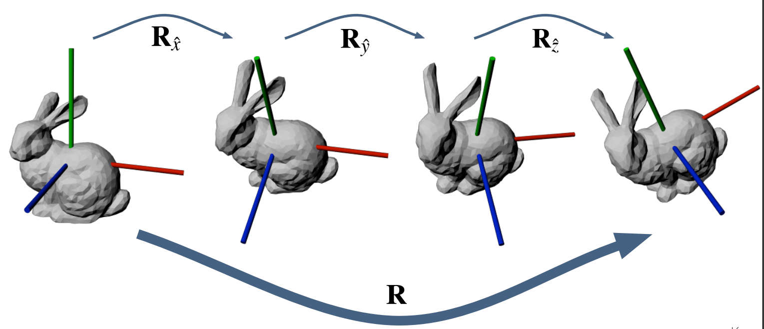


- But NOT a vector.

$$\mathbf{R} = \mathrm{rot}(x, y, z)$$

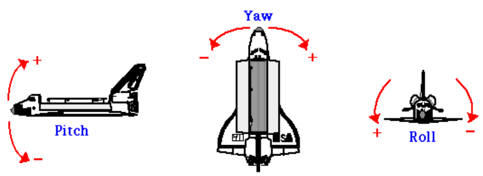
Arbitrary Rotations

$$\mathbf{R} = \mathbf{R}_{\hat{z}} \cdot \mathbf{R}_{\hat{y}} \cdot \mathbf{R}_{\hat{x}}$$



Arbitrary Rotations

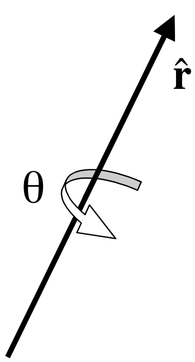
- Allows tumbling
- Euler angles are non-unique
- Gimbal-lock
- Moving -vs- fixed axes
  - Reverse of each other



Exponential Maps

- Direct representation of arbitrary rotation
- AKA: axis-angle, angular displacement vector
- Rotate θ degrees about some axis
- Encode θ by length of vector

$$\theta = |\mathbf{r}|$$



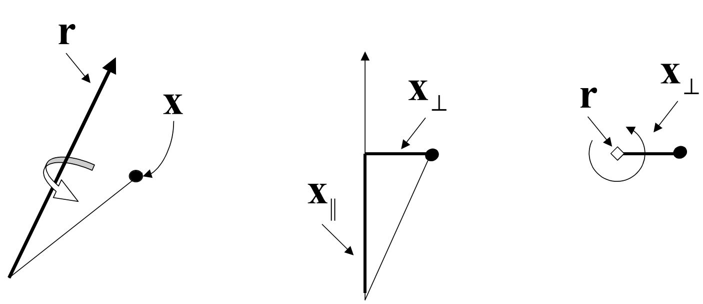
Exponential Maps

- Given vector $\mathbf{r}$, how to get matrix $\mathbf{R}$
- Method from text:
  1. rotate about x axis to put $\mathbf{r}$ into the x-y plane
  2. rotate about z axis align $\mathbf{r}$ with the x axis
  3. rotate θ degrees about x axis
  4. undo #2 and then #1
  5. composite together

19

19

Exponential Maps

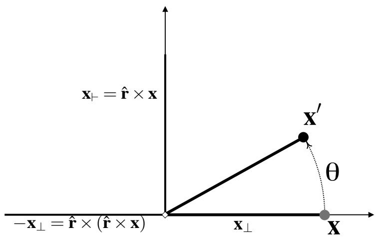


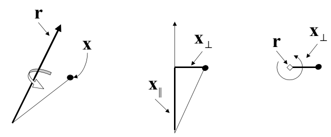
- Vector expressing a point has two parts
  - $\mathbf{x}_\parallel$ does not change
  - $\mathbf{x}_\perp$ rotates like a 2D point

20

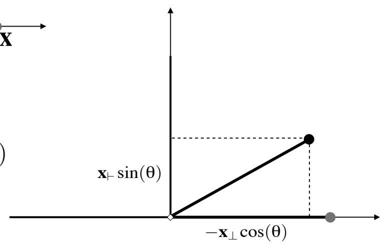
20

Exponential Maps





$$\mathbf{x}' = \mathbf{x}_{||} + \mathbf{x}_{\vdash}\sin(\theta) + \mathbf{x}_\perp \cos(\theta)$$

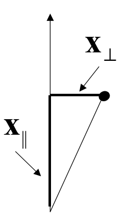


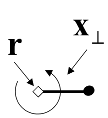
21

Exponential Maps

- Rodriguez Formula

$$\begin{aligned}\mathbf{x}' = {} & \hat{\mathbf{r}}(\hat{\mathbf{r}} \cdot \mathbf{x}) \\ & +\sin(\theta)(\hat{\mathbf{r}} \times \mathbf{x}) \\ & -\cos(\theta)(\hat{\mathbf{r}} \times (\hat{\mathbf{r}} \times \mathbf{x}))\end{aligned}$$





Linear in $\mathbf{x}$

Actually a minor variation ...

22

Tuesday, February 5, 13

Exponential Maps

• Building the matrix

$$\mathbf{x}' = \left((\hat{\mathbf{r}}\hat{\mathbf{r}}^{\mathrm{t}}) + \sin(\theta)(\hat{\mathbf{r}}\times) - \cos(\theta)(\hat{\mathbf{r}}\times)(\hat{\mathbf{r}}\times) \right) \mathbf{x}$$

$$(\hat{\mathbf{r}}\times) = \begin{bmatrix} 0 & -\hat{r}_z & \hat{r}_y \\ \hat{r}_z & 0 & -\hat{r}_x \\ -\hat{r}_y & \hat{r}_x & 0 \end{bmatrix}$$

Antisymmetric matrix

$(\mathbf{a}\times)\mathbf{b} = \mathbf{a} \times \mathbf{b}$

Easy to verify by expansion

Exponential Maps

- Allows tumbling
- No gimbal-lock!
- Orientations are space within π-radius ball
- Nearly unique representation
- Singularities on shells at 2π
- Nice for interpolation

Exponential Maps

- Why exponential?
- Recall series expansion of e^x

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

25

25

Exponential Maps

- Why exponential?
- Recall series expansion of e^x
- Euler: what happens if you put in $i\theta$ for x

$$e^{i\theta} = 1 + \frac{i\theta}{1!} + \frac{-\theta^2}{2!} + \frac{-i\theta^3}{3!} + \frac{\theta^4}{4!} + \cdots$$

$$= \left(1 + \frac{-\theta^2}{2!} + \frac{\theta^4}{4!} + \cdots\right) + i\left(\frac{\theta}{1!} + \frac{-\theta^3}{3!} + \cdots\right)$$

$$= \cos(\theta) + i\sin(\theta)$$

26

26

Exponential Maps

- Why exponential?

$$e^{(\hat{\mathbf{r}}\times)\theta} = \mathbf{I} + \frac{(\hat{\mathbf{r}}\times)\theta}{1!} + \frac{(\hat{\mathbf{r}}\times)^2\theta^2}{2!} + \frac{(\hat{\mathbf{r}}\times)^3\theta^3}{3!} + \frac{(\hat{\mathbf{r}}\times)^4\theta^4}{4!} + \cdots$$

But notice that: $(\hat{\mathbf{r}}\times)^3 = -(\hat{\mathbf{r}}\times)$

$$e^{(\hat{\mathbf{r}}\times)\theta} = \mathbf{I} + \frac{(\hat{\mathbf{r}}\times)\theta}{1!} + \frac{(\hat{\mathbf{r}}\times)^2\theta^2}{2!} + \frac{-(\hat{\mathbf{r}}\times)\theta^3}{3!} + \frac{-(\hat{\mathbf{r}}\times)^2\theta^4}{4!} + \cdots$$

27

27

Exponential Maps

$$e^{(\hat{\mathbf{r}}\times)\theta} = \mathbf{I} + \frac{(\hat{\mathbf{r}}\times)\theta}{1!} + \frac{(\hat{\mathbf{r}}\times)^2\theta^2}{2!} + \frac{-(\hat{\mathbf{r}}\times)\theta^3}{3!} + \frac{-(\hat{\mathbf{r}}\times)^2\theta^4}{4!} + \cdots$$

$$e^{(\hat{\mathbf{r}}\times)\theta} = (\hat{\mathbf{r}}\times)\left(\frac{\theta}{1!} - \frac{\theta^3}{3!} + \cdots\right) + \mathbf{I} + (\hat{\mathbf{r}}\times)^2\left(+\frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \cdots\right)$$

$$e^{(\hat{\mathbf{r}}\times)\theta} = (\hat{\mathbf{r}}\times)\sin(\theta) + \mathbf{I} + (\hat{\mathbf{r}}\times)^2(1-\cos(\theta))$$

28

28

Quaternions

- More popular than exponential maps
- Natural extension of $e^{i\theta} = \cos(\theta) + i\sin(\theta)$
- Due to Hamilton (1843)
  - Interesting history
  - Involves "hermaphroditic monsters"

Quaternions

- Uber-Complex Numbers

$$q = (z_1, z_2, z_3, s) = (\mathbf{z}, s)$$

$$q = iz_1 + jz_2 + kz_3 + s$$

$$i^2 = j^2 = k^2 = -1 \qquad \begin{array}{ll} ij = k & ji = -k \\ jk = i & kj = -i \\ ki = j & ik = -j \end{array}$$

Quaternions

- Multiplication natural consequence of defn.

$$\mathsf{q} \cdot \mathsf{p} = (\mathbf{z}_q s_p + \mathbf{z}_p s_q + \mathbf{z}_p \times \mathbf{z}_q \ \ , \ \ s_p s_q - \mathbf{z}_p \cdot \mathbf{z}_q)$$

- Conjugate

$$\mathsf{q}^* = (-\mathbf{z}, s)$$

- Magnitude

$$||\mathsf{q}||^2 = \mathbf{z} \cdot \mathbf{z} + s^2 = \mathsf{q} \cdot \mathsf{q}^*$$

31

31

Quaternions

- Vectors as quaternions

$$\mathsf{v} = (\mathbf{v}, 0)$$

- Rotations as quaternions

$$\mathsf{r} = (\hat{\mathbf{r}} \sin\frac{\theta}{2}, \cos\frac{\theta}{2})$$

- Rotating a vector

$$\mathsf{x}' = \mathsf{r} \cdot \mathsf{x} \cdot \mathsf{r}^*$$

- Composing rotations

$$\mathsf{r} = \mathsf{r}_1 \cdot \mathsf{r}_2$$

Compare to Exp. Map

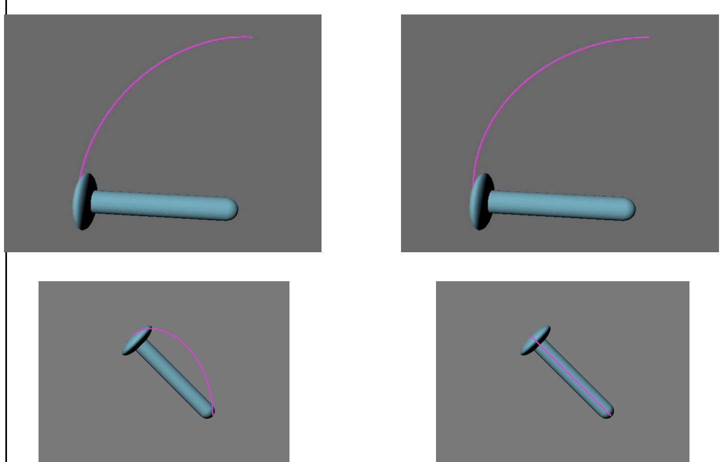
32

32

Quaternions

- No tumbling
- No gimbal-lock
- Orientations are "double unique"
- Surface of a 3-sphere in 4D $||\mathrm{r}|| = 1$
- Nice for interpolation

Interpolation



Rotation Matrices

- Eigen system
  - One real eigenvalue
  - Real axis is axis of rotation
  - Imaginary values are 2D rotation as complex number
- Logarithmic formula

$$(\hat{\mathbf{r}}\times) = \ln(\mathbf{R}) = \frac{\theta}{2\sin\theta}(\mathbf{R} - \mathbf{R}^{\mathsf{T}})$$

$$\theta = \cos^{-1}\left(\frac{\mathrm{Tr}(\mathbf{R}) - 1}{2}\right)$$

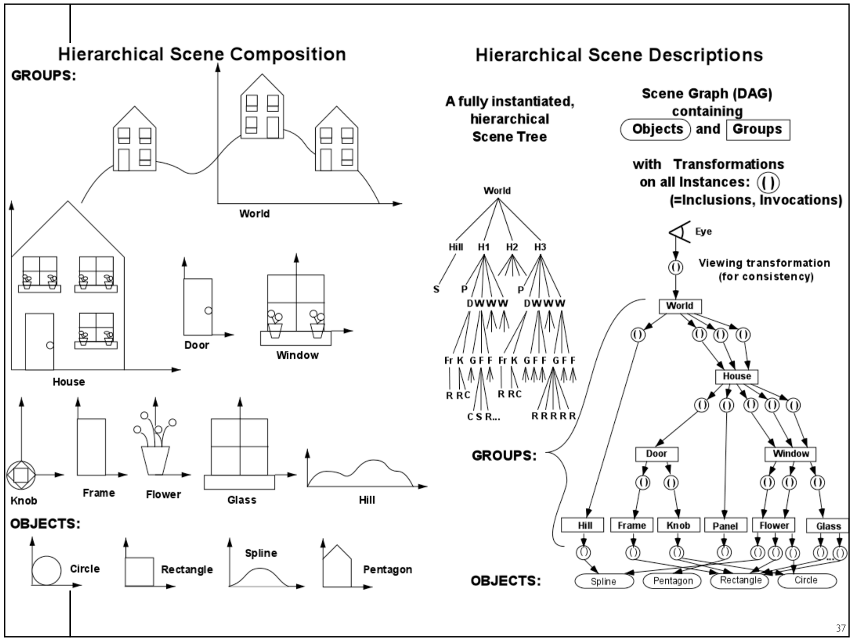
Similar formulae as for exponential...

Rotation Matrices

- Consider:

$$\mathbf{RI} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

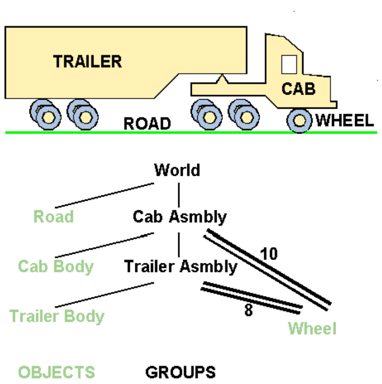
- Columns are coordinate axes after (true for general matrices)
- Rows are original axes in original system (not true for general matrices)



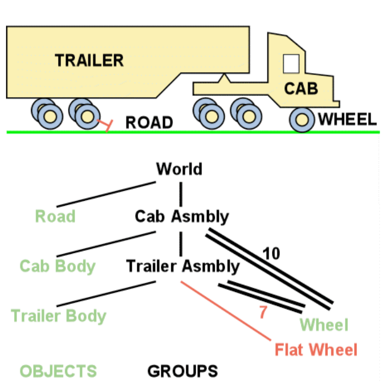
Scene Graphs

- Draw scene with pre-and-post-order traversal
  - Apply node, draw children, undo node if applicable
- Nodes can do pretty much anything
  - Geometry, transformations, groups, color, switch, scripts, *etc.*
  - Node types are application/implementation specific
- Requires a stack to implement "undo" post children
- Nodes can cache their children
- Instances make it a DAG, not strictly a tree
- Will use these trees later for bounding box trees

What is the "Right" Hierarchy for this 18-Wheeler ?



What is the "Right" Hierarchy for this 18-Wheeler ?



39

Note:

- Rotation stuff in the book is a bit weak... luckily you have these nice slides!

40