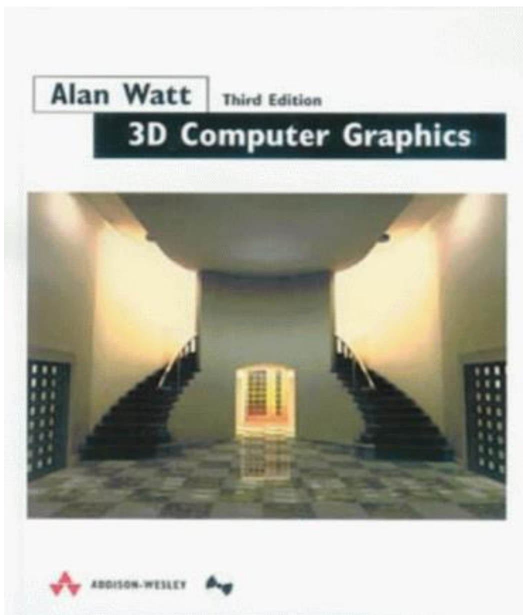
The University Of Sheffield.

COM3503/4503/6503: 3D Computer Graphics

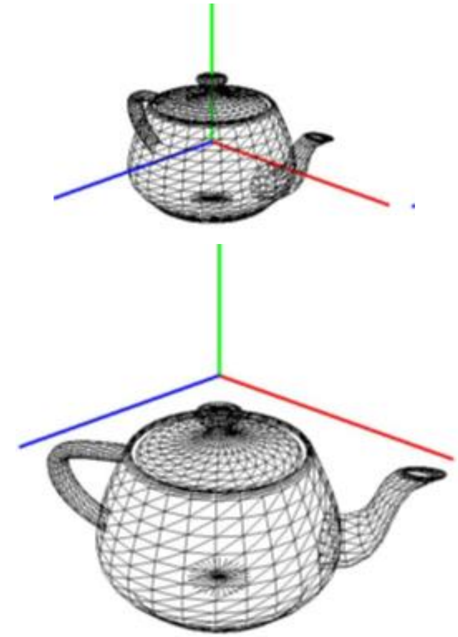
Lecture 3. Hierarchical structures and scene graphs

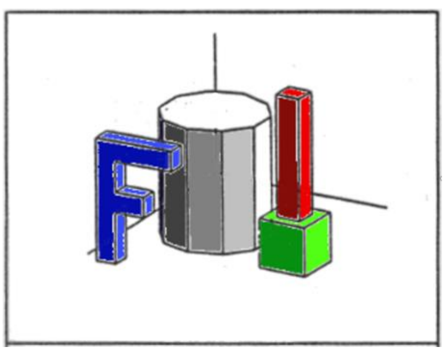


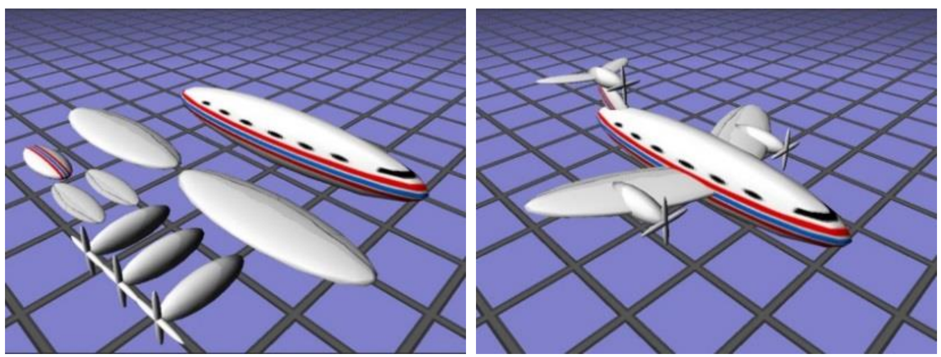
Dr. Steve Maddock
s.maddock@sheffield.ac.uk

1. Introduction

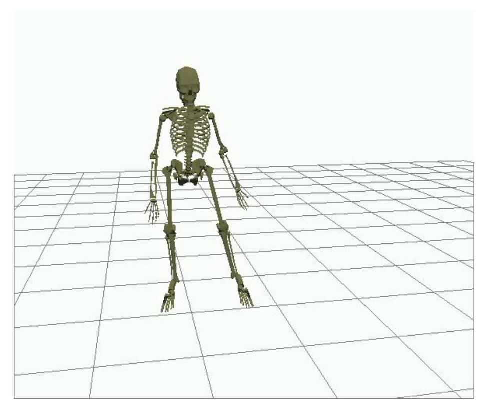
- Use transformations to:
  - Manipulate individual objects
  - Build scenes
  - Build complex objects from pieces
  - Control relationship between parts in complex hierarchical objects
  - Conversion between coordinate systems





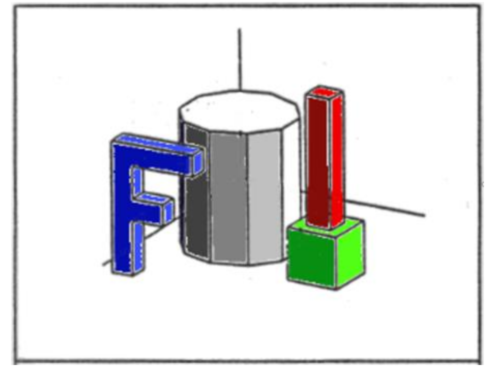


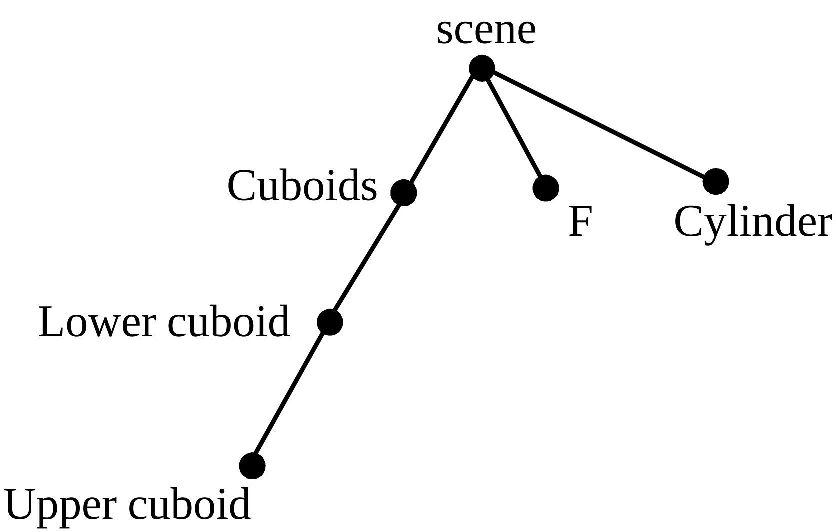
http://viz.aset.psu.edu/jack/java3d/

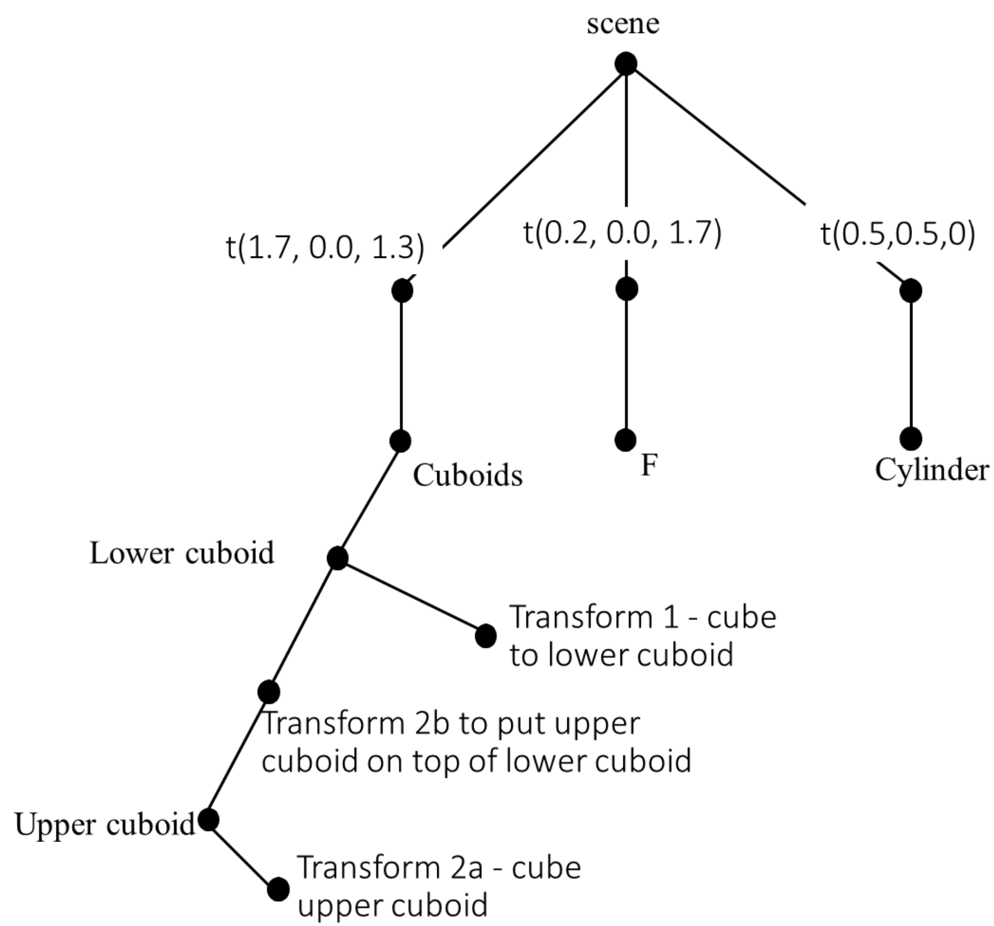


2. Scene graphs

- Multiple ways to represent it
- With or without transformation nodes

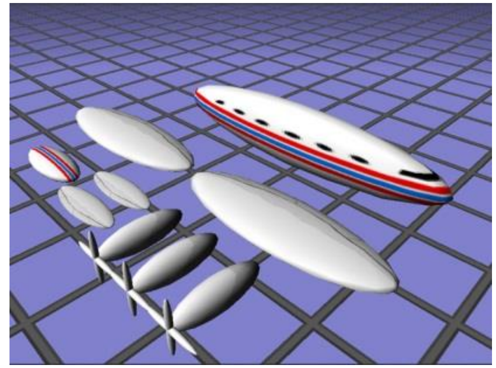


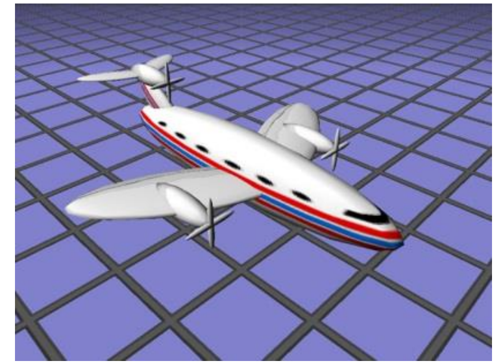




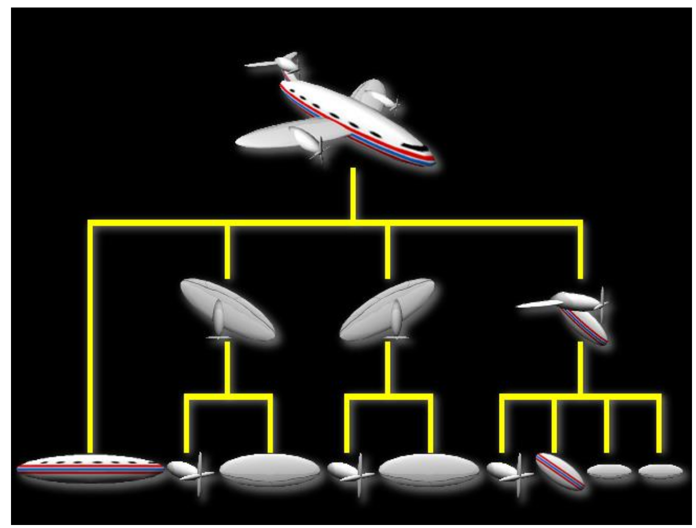
3. Complex objects – hierarchical modelling

- Simple objects can be combined to make more complex objects
- A scene graph represents the hierarchy
  - Transformations to compose each piece into the hierarchy
  - Simple visualisation only shows the pieces, not the transformation nodes
- The whole object can be transformed by transforming the root
- Individual pieces can be transformed
  - E.g. rotate propellers

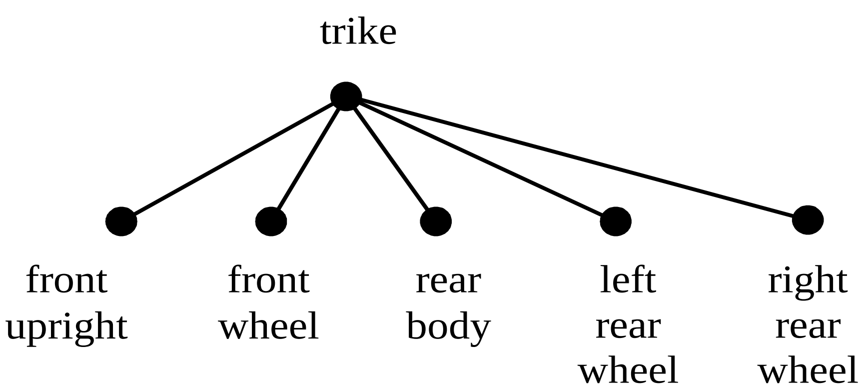


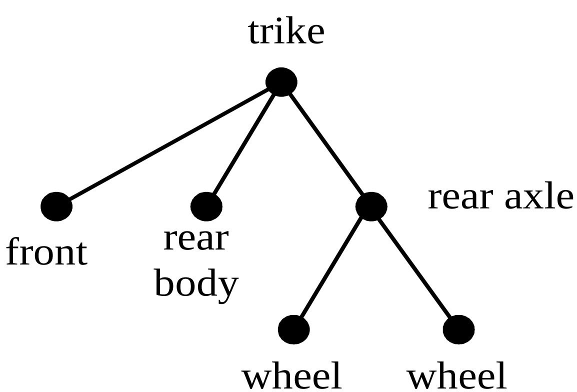


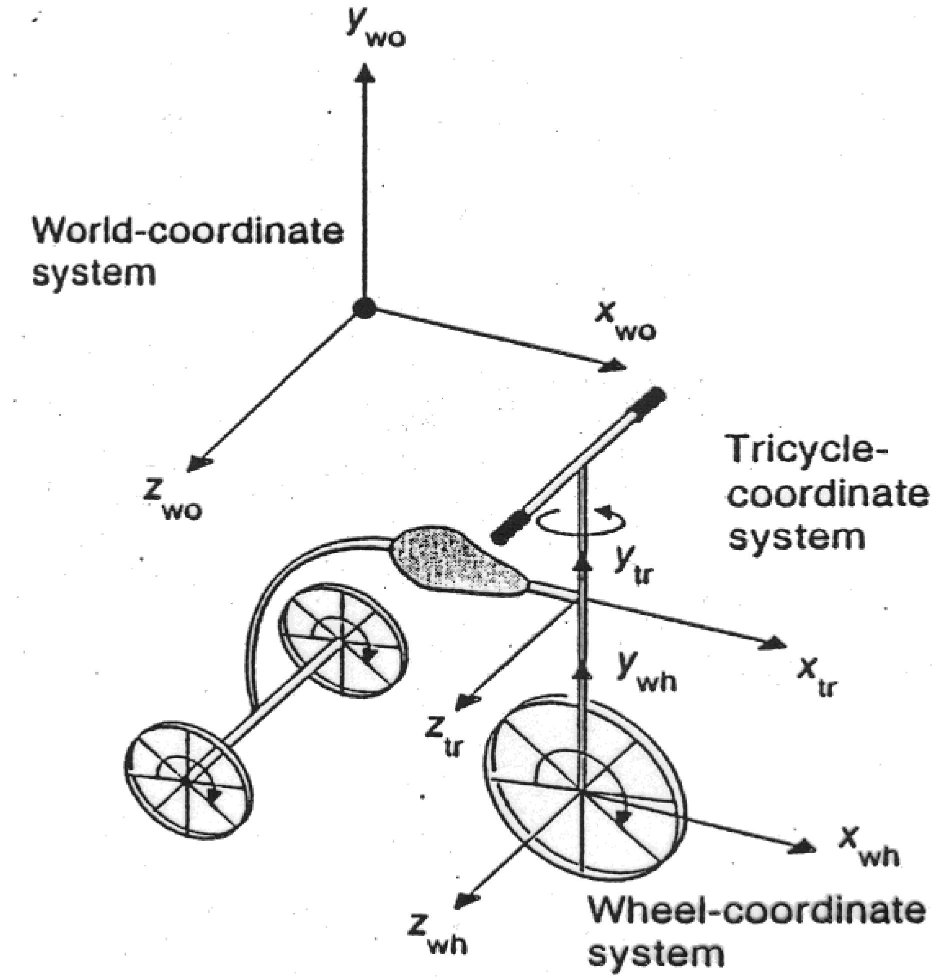
http://viz.aset.psu.edu/jack/java3d/



4. Alternative hierarchies





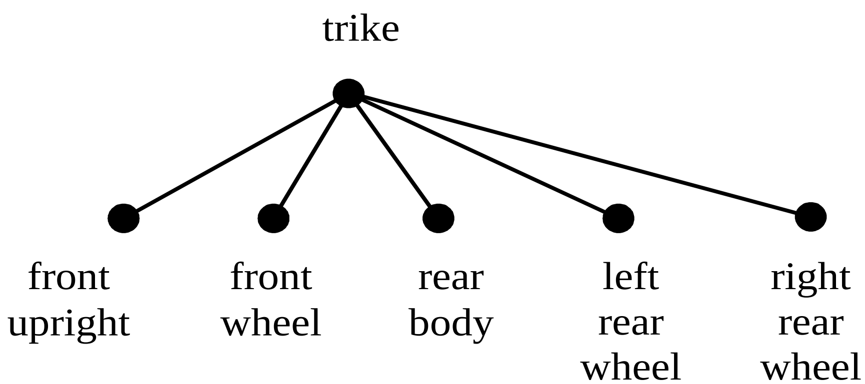


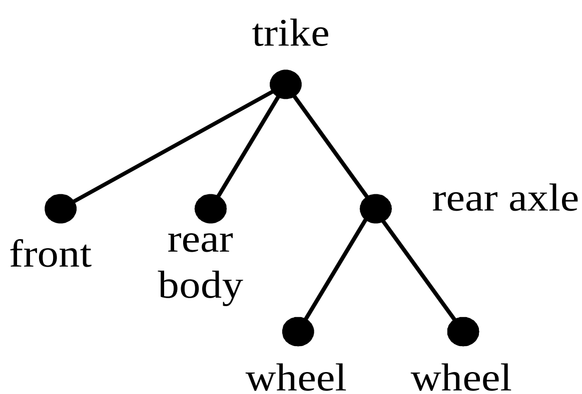
A stylised tricycle (p.190 from (Foley et al, 1994))

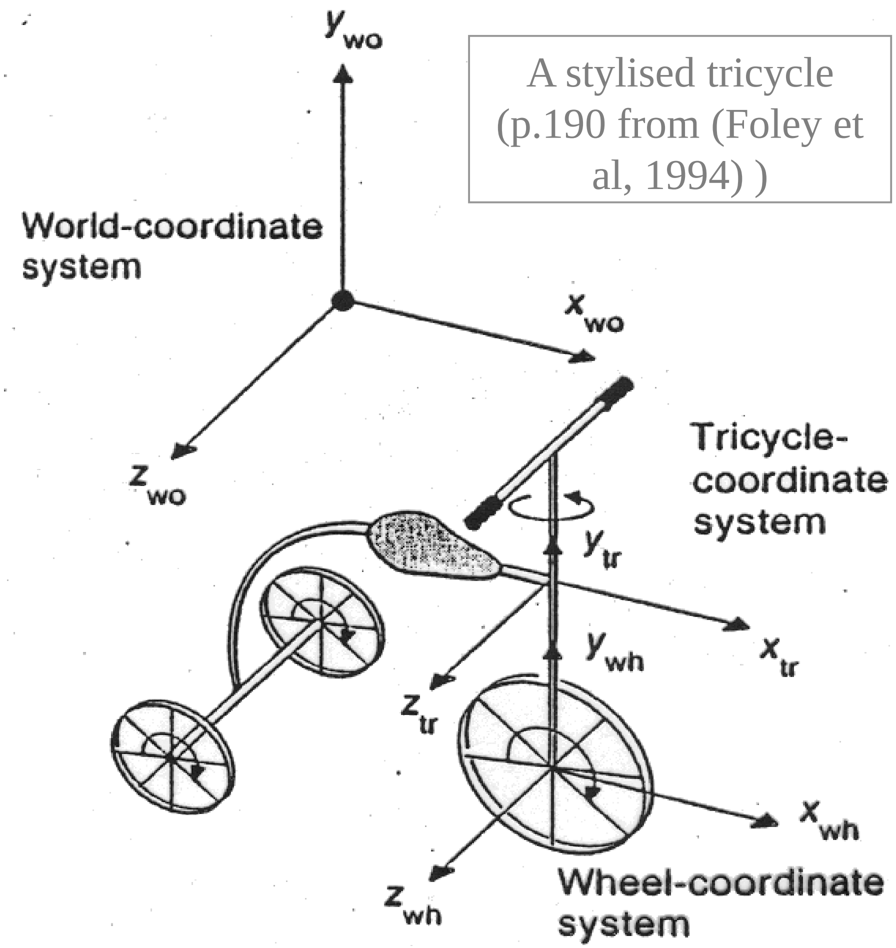
Foley, J.D., A. van Dam, S.K.Feiner, J.F.Hughes, R.L.Philips, “Introduction to Computer Graphics”, Addison-Wesley, 1994

4. Alternative hierarchies

- Consider how to animate the parts
  - Relationships across the hierarchy
  - E.g. wheels must all rotate together







A stylised tricycle (p.190 from (Foley et al, 1994))

Foley, J.D., A. van Dam, S.K.Feiner, J.F.Hughes, R.L.Philips, “Introduction to Computer Graphics”, Addison-Wesley, 1994

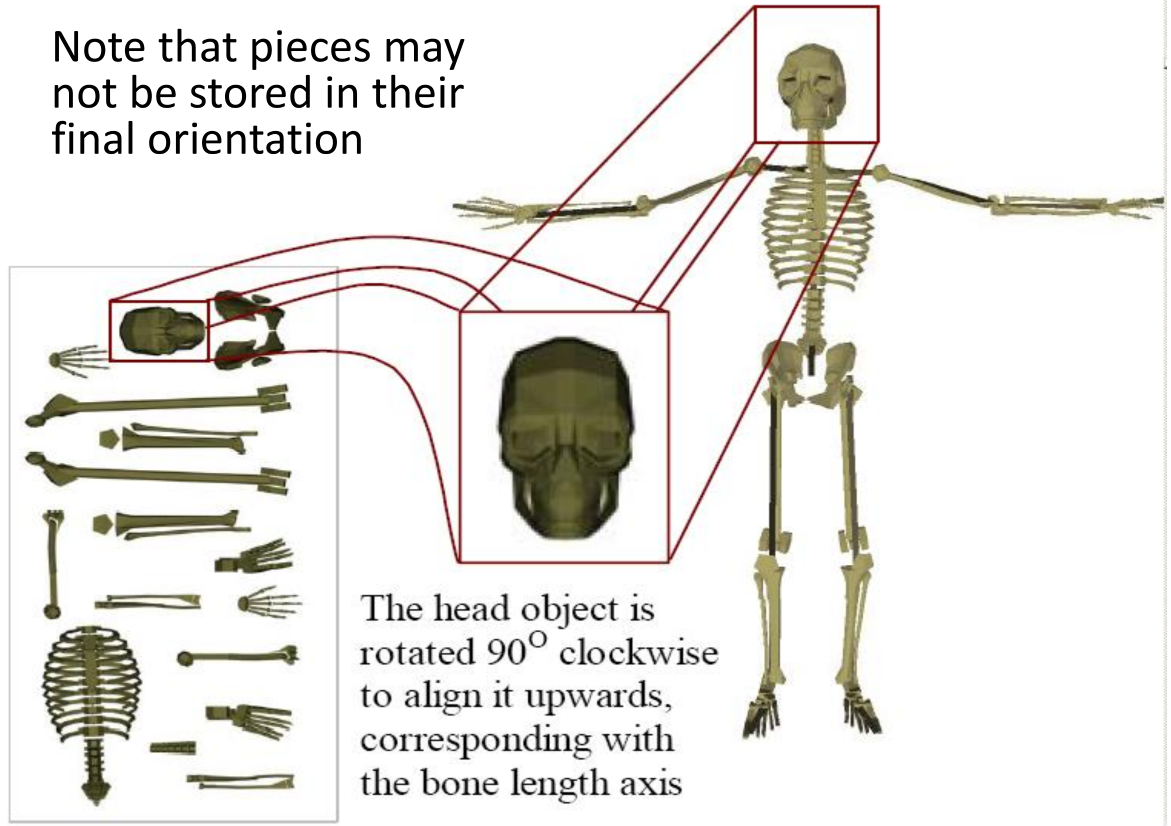
5. An articulated hierarchical object

- Many objects are composed of multiple pieces that move in relation to each other
- A human arm has an upper arm, a lower arm and a hand
- Parent-child relationship
- If the parent moves, so do its children
  - A child is attached to its parent
- If a child moves, e.g. rotates about the connection point, its parent need not
- We can represent this kind of parent-child relationship with a scene graph

5.1 The scene graph

- Root - Hips
  - Chest
    - Neck
      - Head
    - LeftCollar
      - LeftShoulder
        - LeftElbow
          - LeftWrist
    - RightCollar
      - RightShoulder
        - RightElbow
          - RightWrist
  - LeftHip
    - LeftKnee
      - LeftAnkle
  - RightHip
    - RightKnee
      - RightAnkle

Note that pieces may not be stored in their final orientation

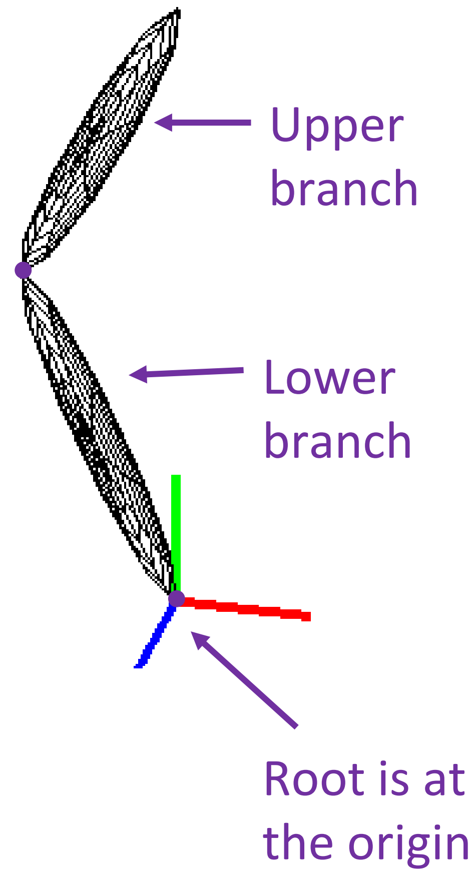


The head object is rotated 90^{o} clockwise to align it upwards, corresponding with the bone length axis

Courtesy of Mike Meredith

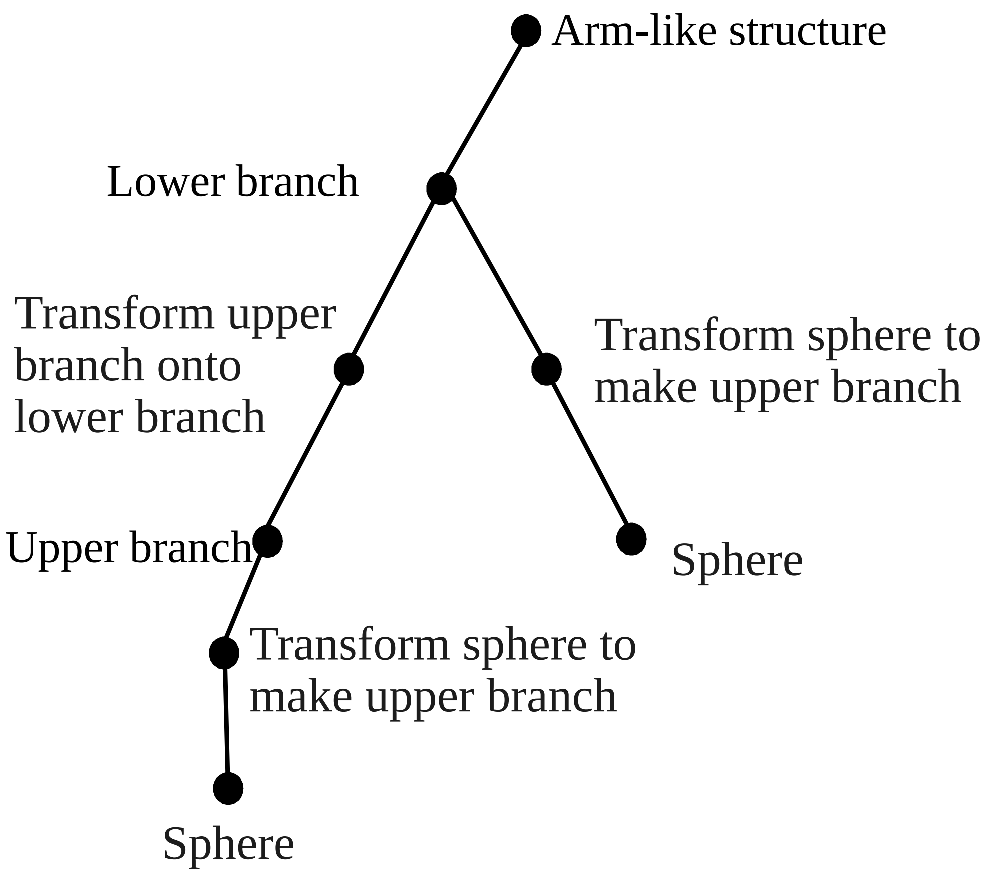
6. Example: A arm-like structure

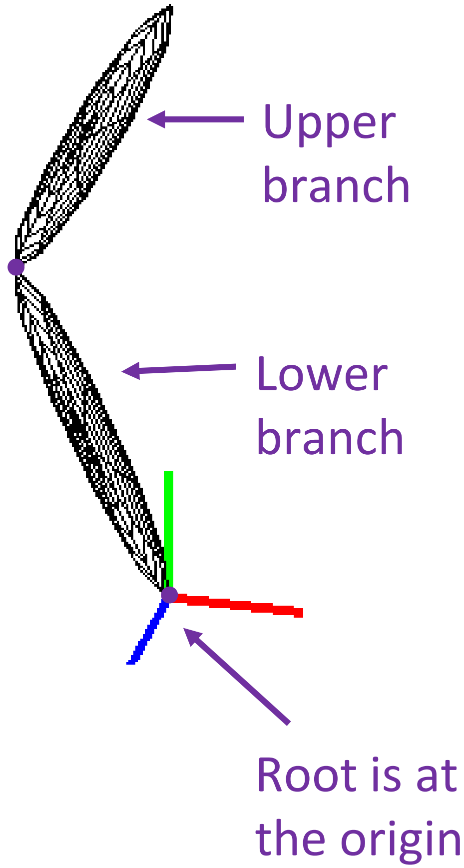
- Lower and upper branch are each made of an ellipse (i.e. a scaled sphere)
- Lower branch is at origin and rotates about the origin
- Upper branch is connected to lower branch and rotates about the connection point

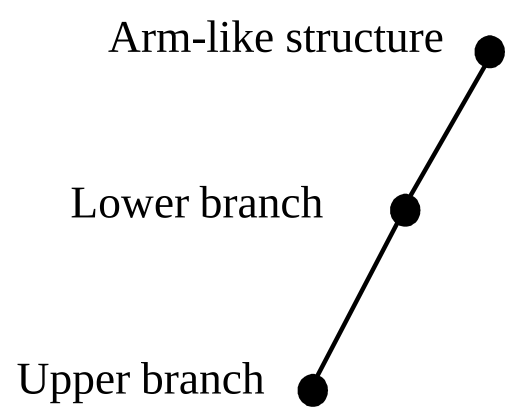


6.1 The scene graph

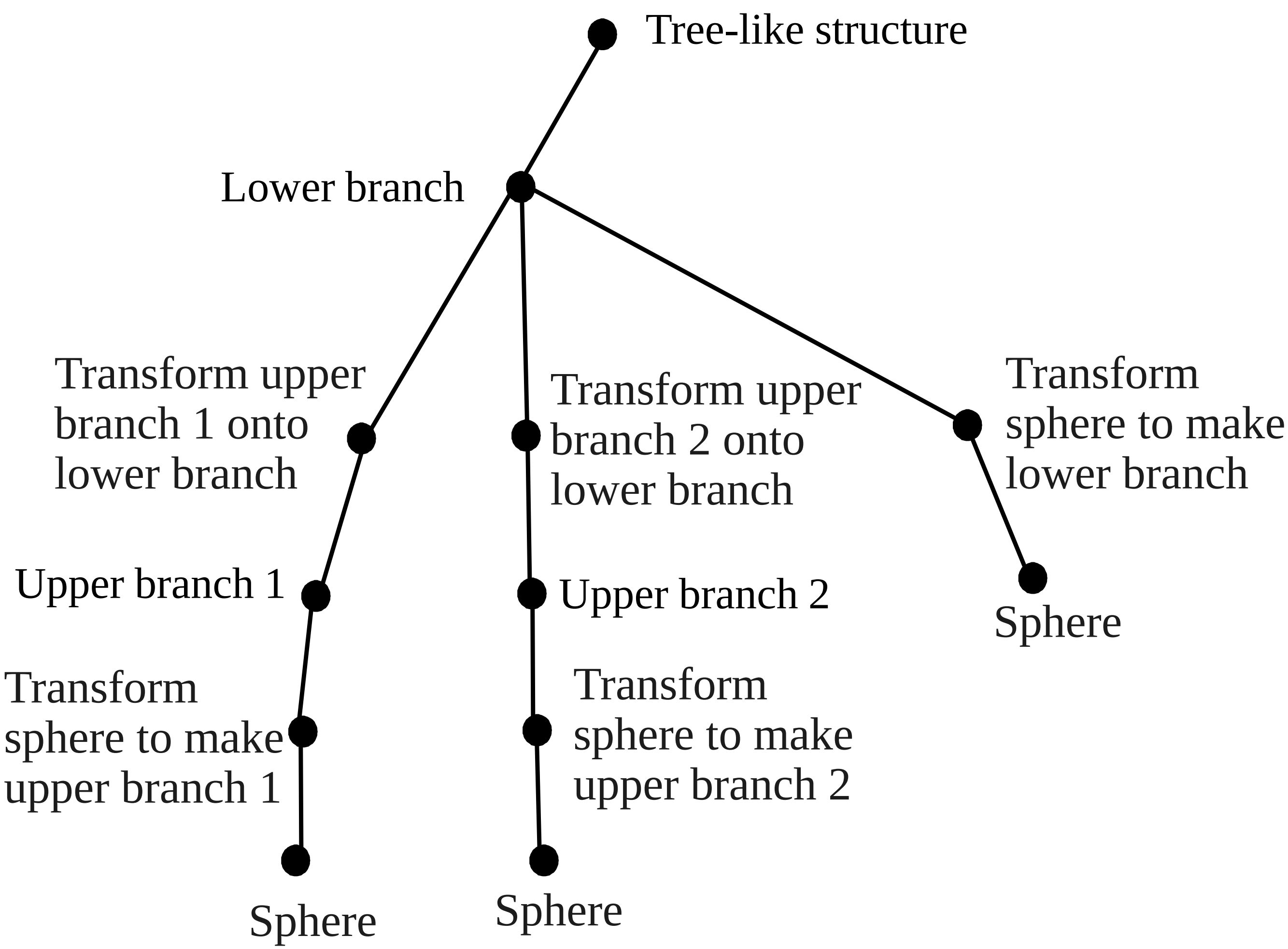
- The scene graph represents the hierarchy (upside down)

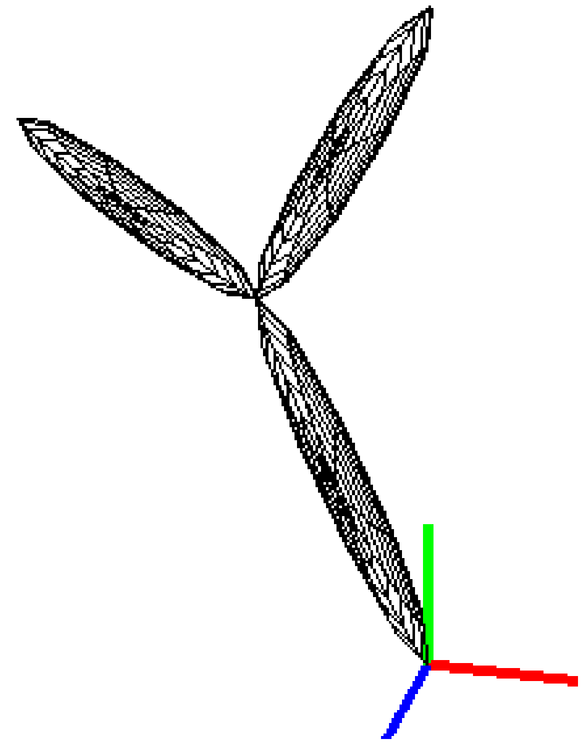


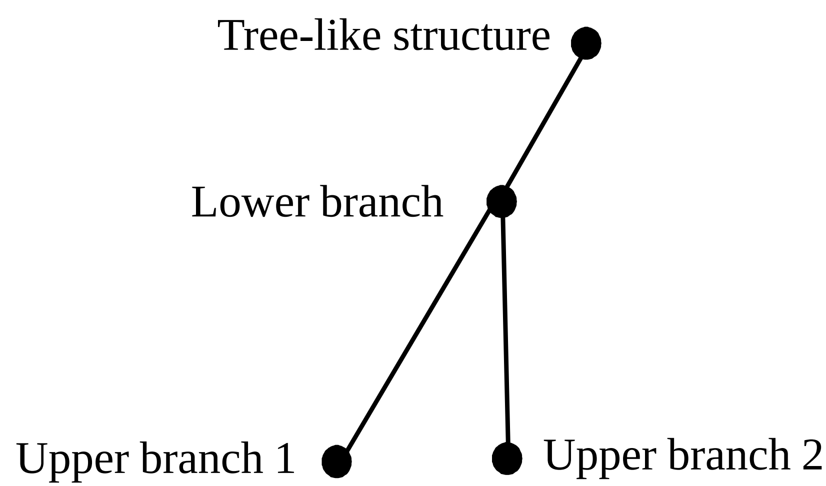




6.2 Two upper branches

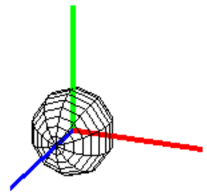


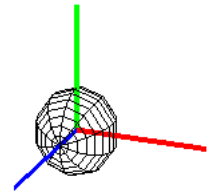




6.3 Drawing a branch: alternatives

1 Sphere of radius 0.5. Diameter of 1 is easier to work with.

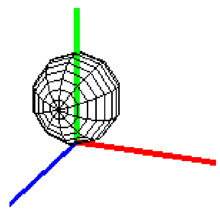


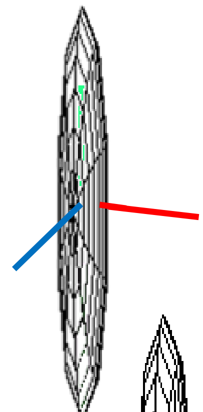


Sphere of radius 1.0.

Diameter is therefore 2.

2 Translate up y axis so lowest point is at origin



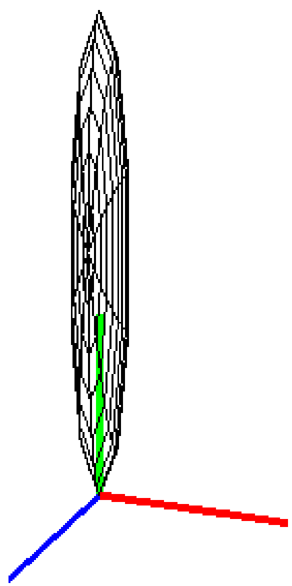


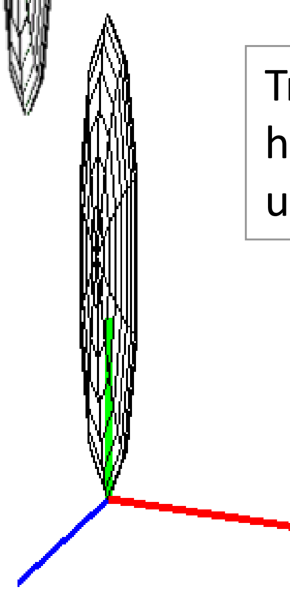
Scale to required height

3 Scale in y direction by h and in x and z directions by 2r.

h is the height of the required branch.

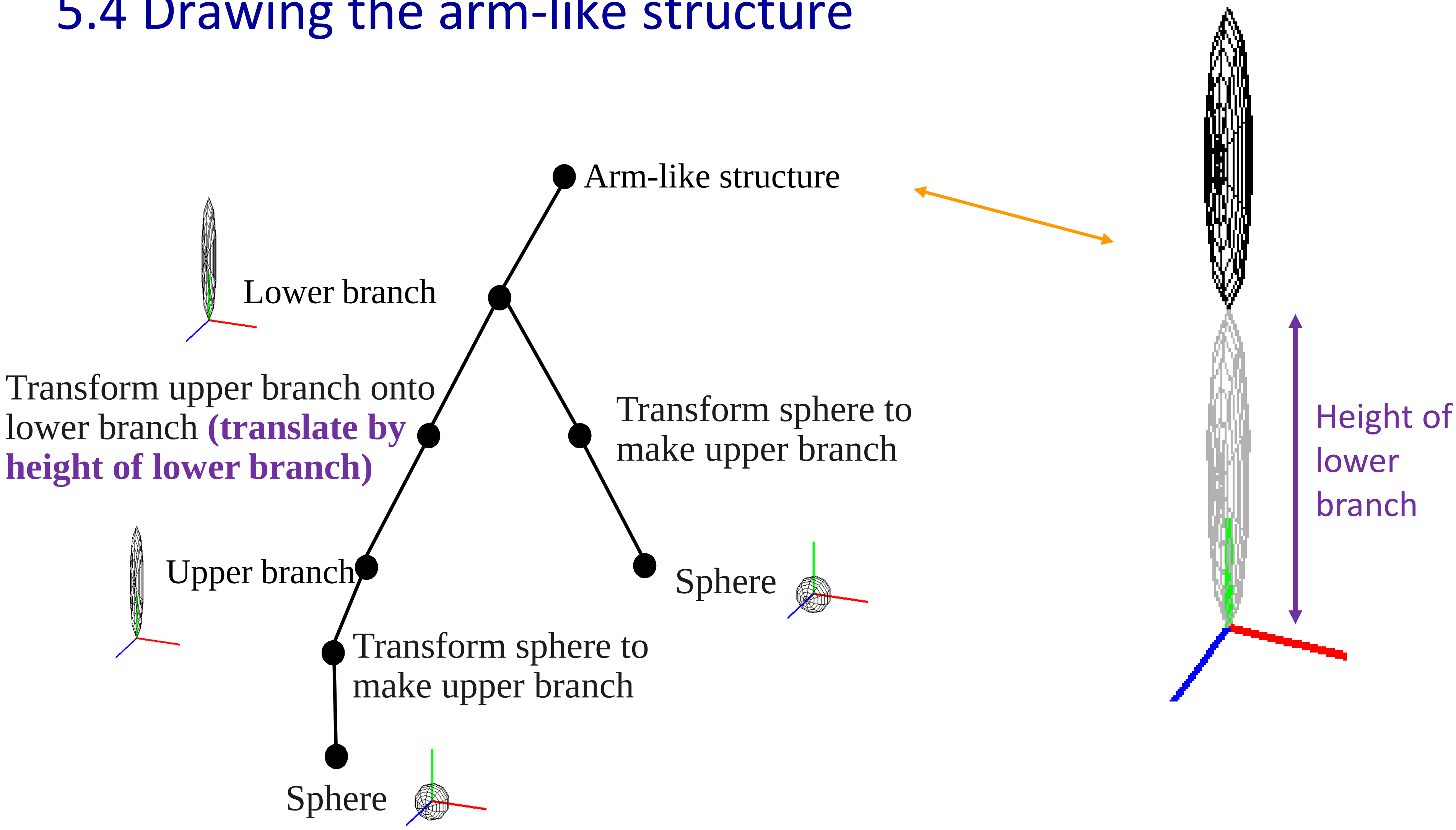
Given current radius of 0.5, need to scale x and z by 2r, where r is new radius



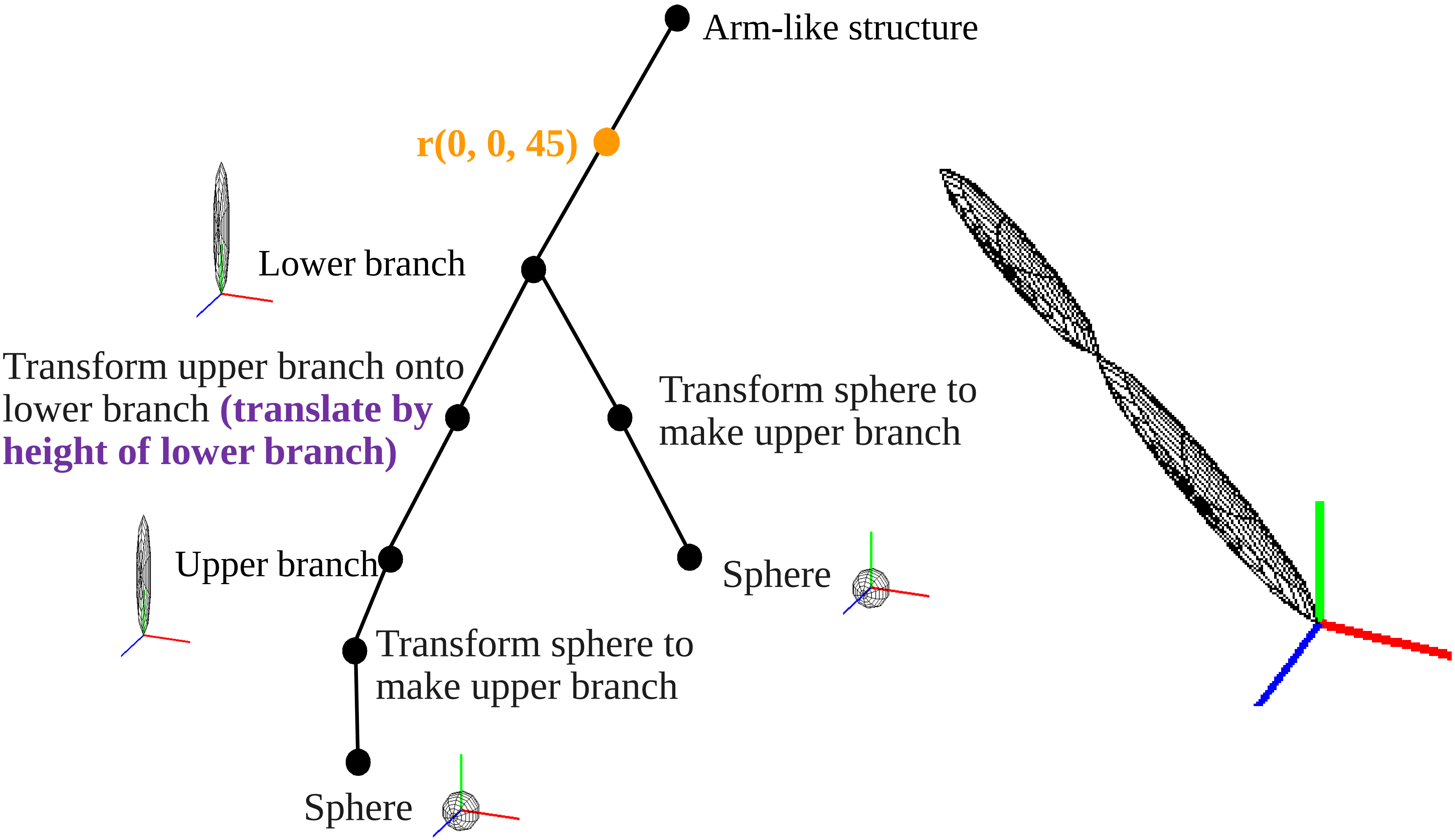


Translate by half the height up the y axis

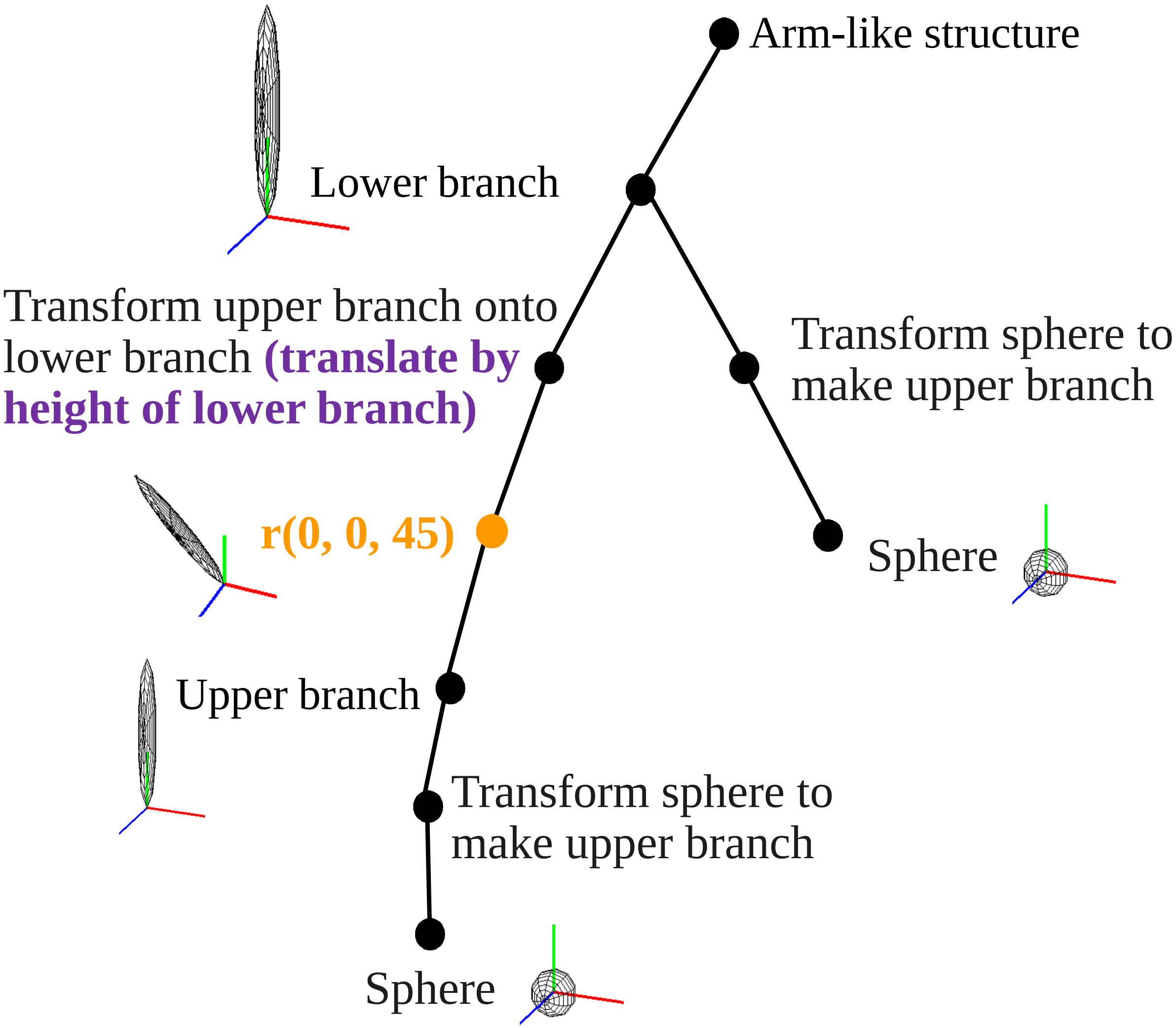
5.4 Drawing the arm-like structure

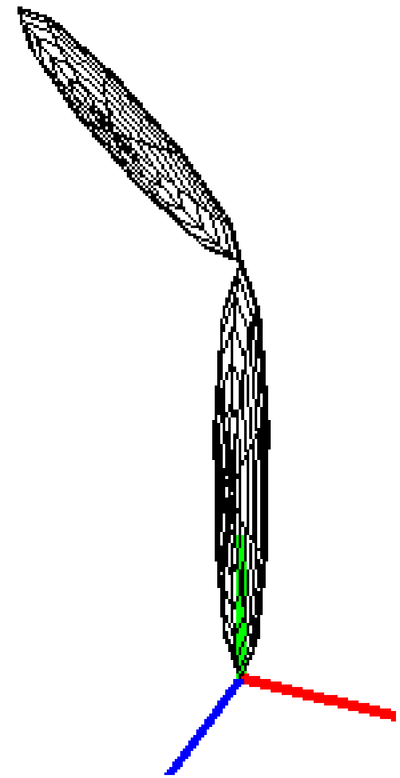


5.5 Rotate the upper and lower branches together

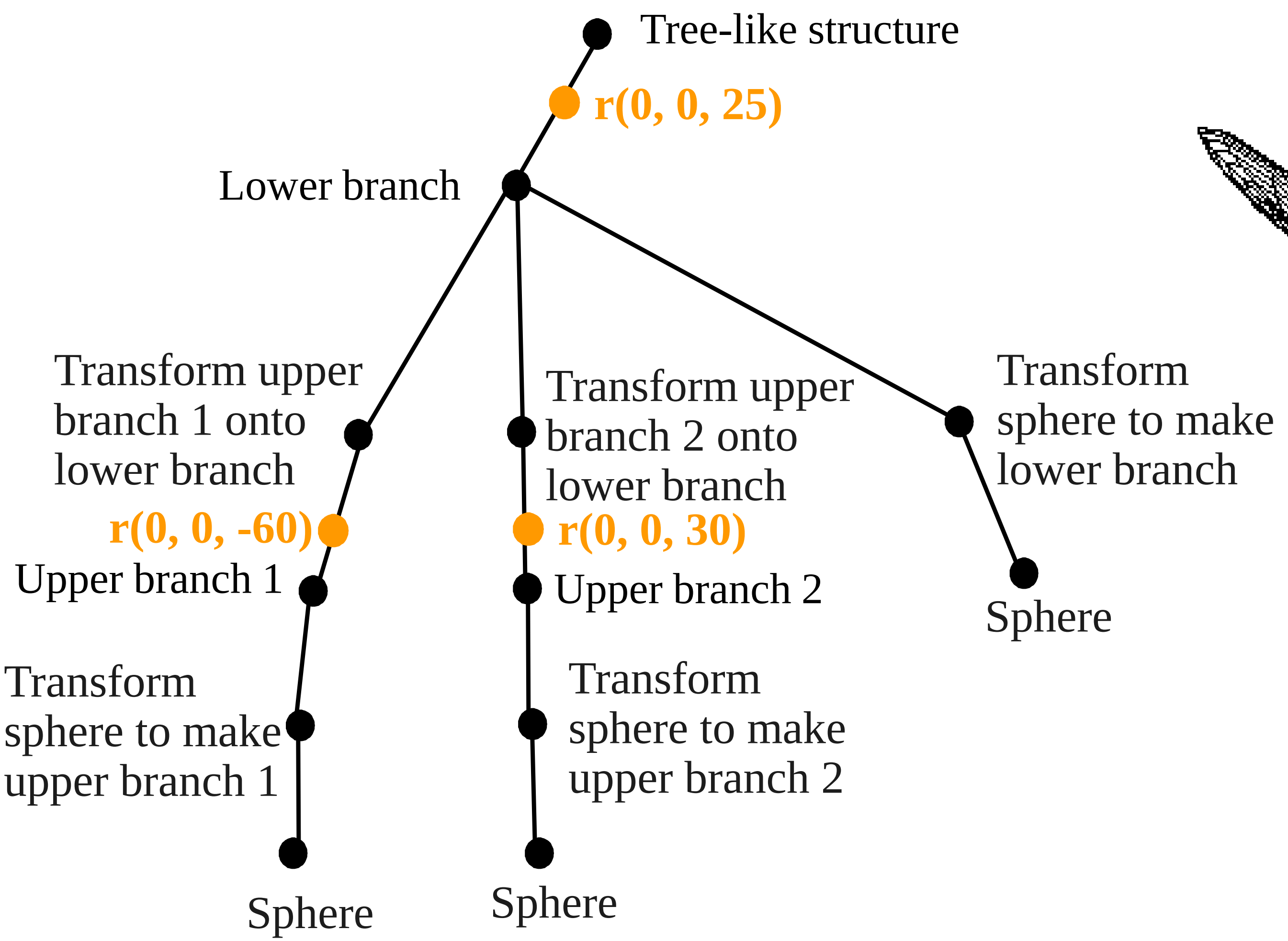


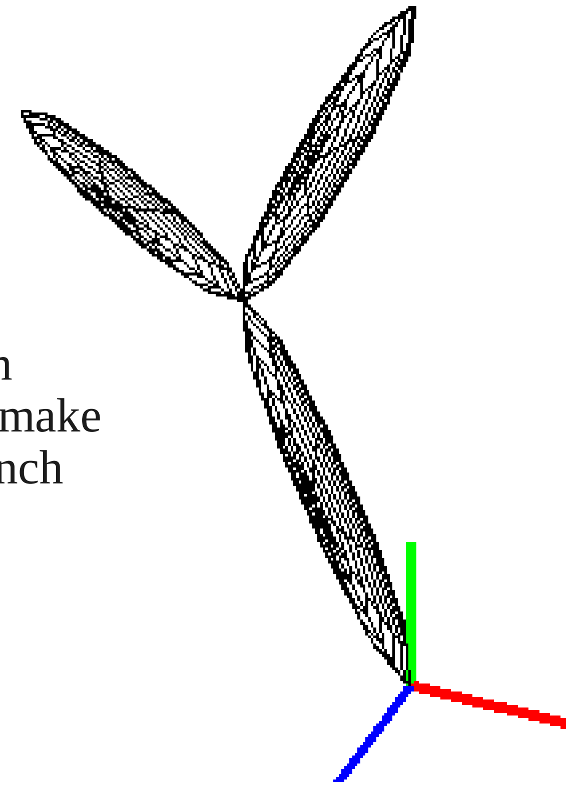
5.6 Rotate the upper branch only



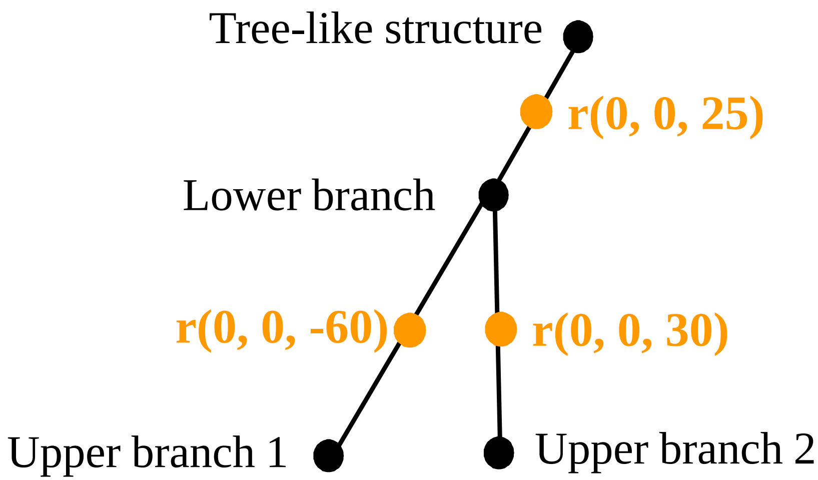


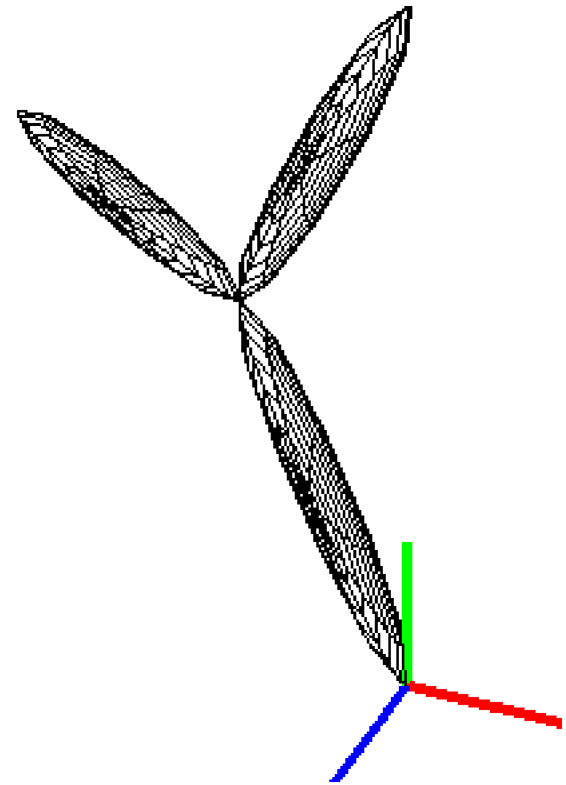
5.7 Rotate branches of tree-like structure





5.7 Rotate branches of tree-like structure





6. Visualising the transformation frames

- The origin of the object is not necessarily the point which the children rotate about
- Different frames of reference can be considered

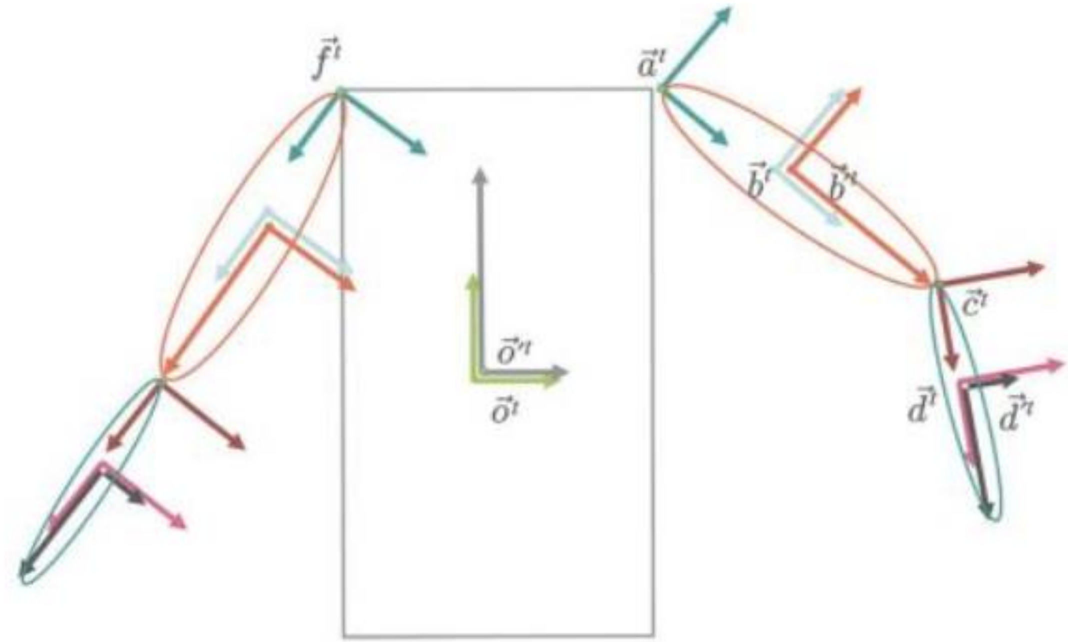


Figure 5.4
In this example, the green frame is the object frame $\vec{\mathbf{o}}^t = \vec{\mathbf{w}}^t O$, and the gray frame is the scaled object frame $\vec{\mathbf{o}}'^t = \vec{\mathbf{o}}^t O'$. The coordinates of a unit cube drawn in $\vec{\mathbf{o}}'^t$ forms a rectangular body. The matrix O can be changed to move the entire robot. The cyan frame $\vec{\mathbf{a}}^t = \vec{\mathbf{o}}^t A$ is the right shoulder frame. The rotational factor of A can be changed to rotate the entire right arm. The red frame $\vec{\mathbf{b}}^t = \vec{\mathbf{a}}^t B$ is the right upper arm frame. The light blue frame $\vec{\mathbf{b}}'^t = \vec{\mathbf{b}}^t B'$ is the scaled right upper arm frame. The coordinates of a unit sphere drawn in $\vec{\mathbf{b}}'^t$ forms the ellipsoidal right upper arm. $\vec{\mathbf{c}}^t = \vec{\mathbf{b}}^t C$ is the right elbow frame. The rotational factor of C can be changed to rotate the right lower arm. $\vec{\mathbf{d}}^t = \vec{\mathbf{c}}^t D$ and $\vec{\mathbf{d}}'^t = \vec{\mathbf{d}}^t D'$ are, respectively, the orthonormal and scaled lower arm frames used to draw the lower arm. The frame $\vec{\mathbf{f}}^t = \vec{\mathbf{o}}^t F$ is the left shoulder frame.

S.J. Gortler, "Foundations of 3D Computer Graphics", MIT Press, 2012

6. Visualising the transformation frames

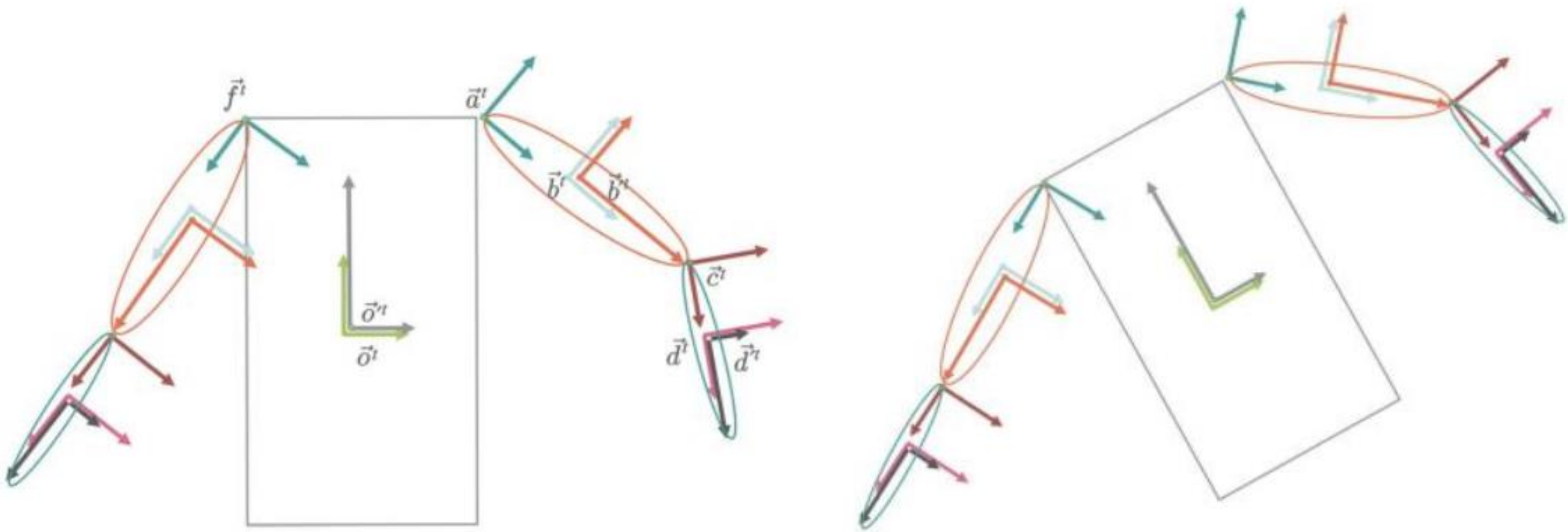


Figure 5.5
To move the entire robot, we update its O matrix.

S.J. Gortler, "Foundations of 3D Computer Graphics", MIT Press, 2012

6. Visualising the transformation frames

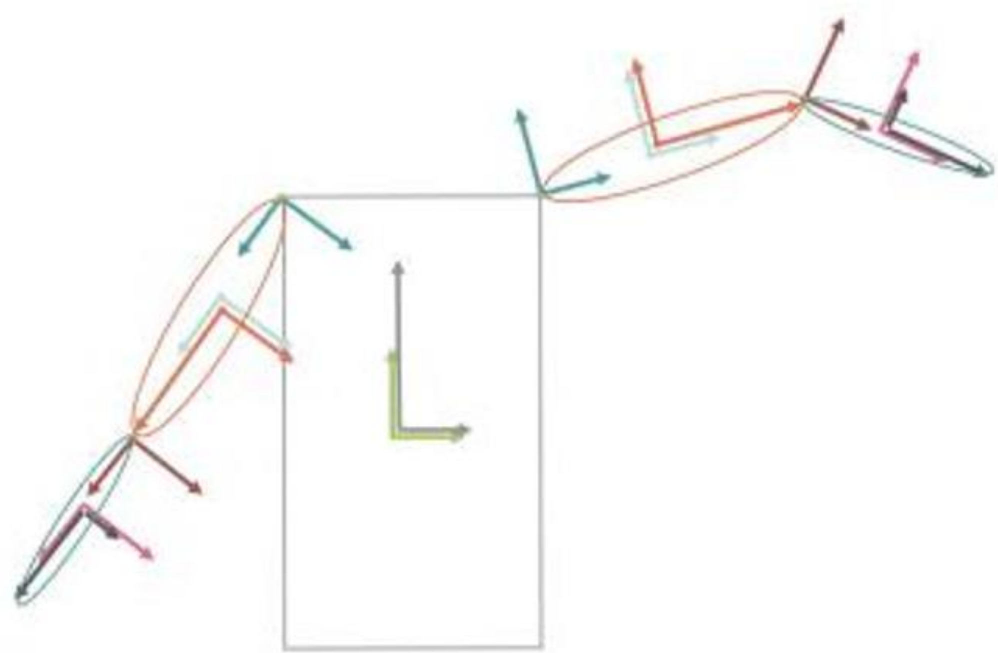


Figure 5.6
To bend the arm at the shoulder, we update the A matrix.

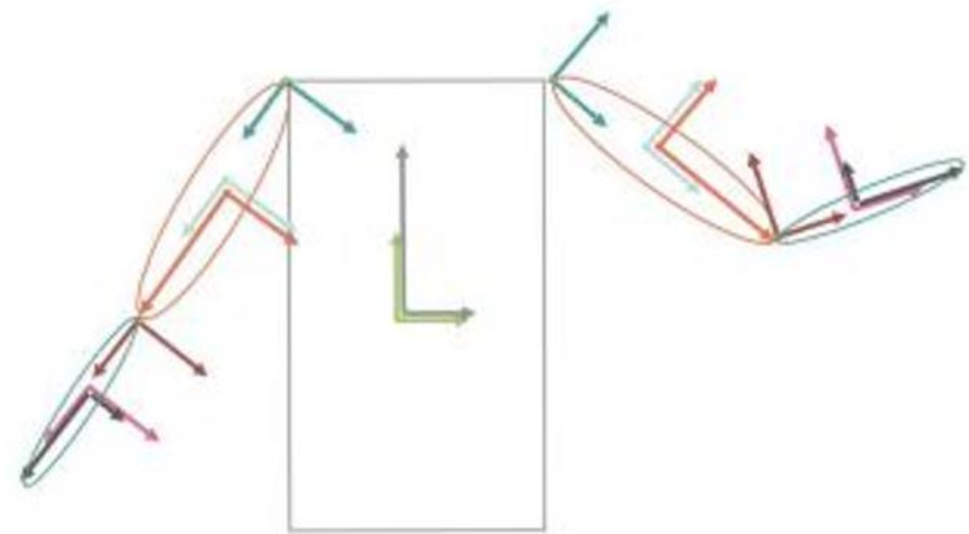
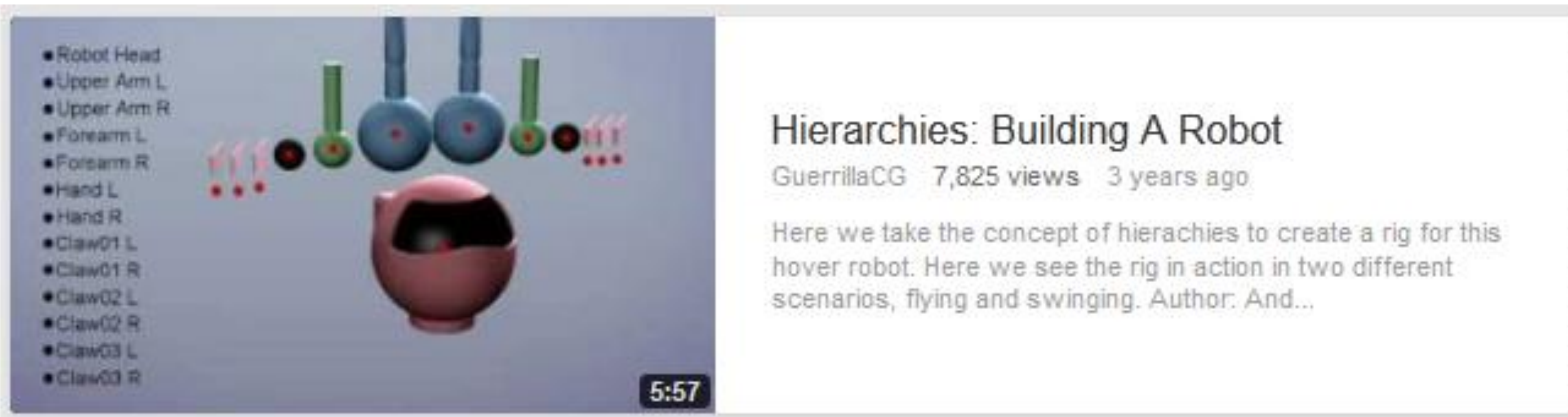


Figure 5.7
To bend the elbow robot, we update its C matrix.

S.J. Gortler, "Foundations of 3D Computer Graphics", MIT Press, 2012

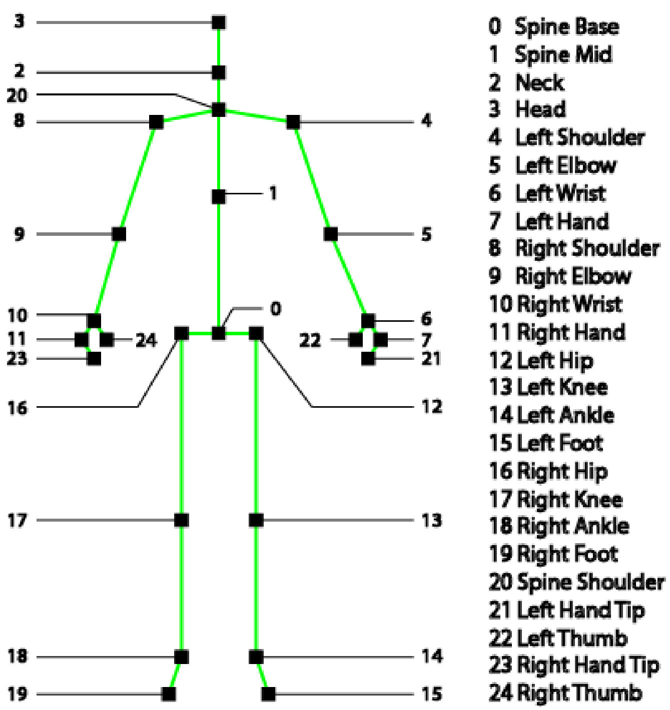
7. Example

- The Guerrilla CG Project: [www.youtube.com/user/**GuerrillaCG**](www.youtube.com/user/GuerrillaCG), [**https://www.youtube.com/watch?v=q54fbMvl3GE**](https://www.youtube.com/watch?v=q54fbMvl3GE)

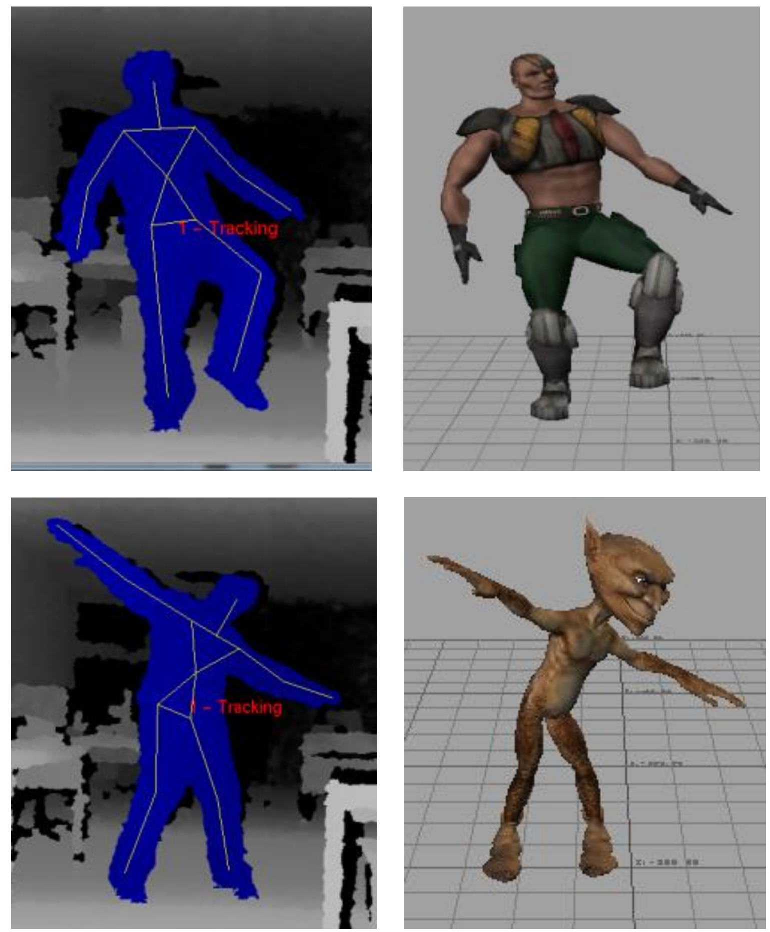


8. Skeletons and skins

- Previous examples have used rigid pieces to make a complex object
- Instead, use a ‘skeleton’ (stick figure) represented as a transformation hierarchy and then add a skin



Based on data at: https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx



8. Skeletons and skins

- To map a vertex, v, in the left hand into the root node, we use

  $v_w = Mv$

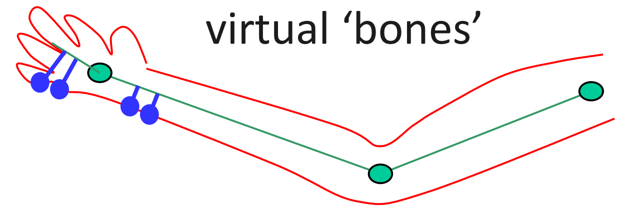
  $M = T_{Hips} R_{Hips} T_{Chest} R_{Chest} T_{LeftCollar}$
  $R_{LeftCollar} T_{LeftShoulder} R_{LeftShoulder}$
  $T_{LeftElbow} R_{LeftElbow} T_{LeftWrist} R_{LeftWrist}$

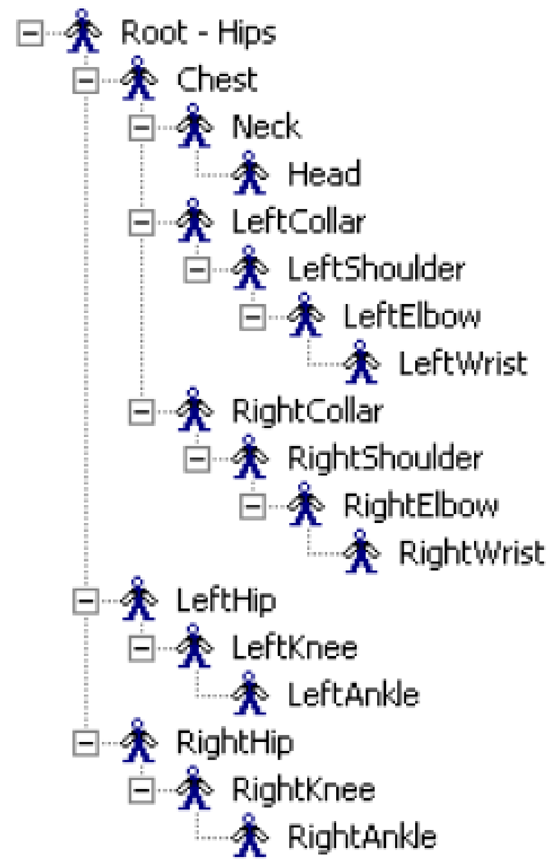
  $$M = \prod_{i=1}^{n} M_i$$

  where $M_i = T_i R_i$, and M_i is the local matrix for a particular node (T = translate, R = Rotate)

- All Ts except T_{Hips} remain constant throughout an animation sequence

Skin vertices are attached to the virtual ‘bones’





9. Summary

- Use transformations to:
  - Manipulate individual objects
  - Build scenes
  - Build complex objects from pieces
  - Control relationship between parts in complex hierarchical objects
  - Conversion between coordinate systems
- A scene graph represents a complex scene or a hierarchical model constructed of pieces
- In commercial systems, a scene graph API can be used
  - Or a matrix stack can be implemented
- Scene graphs have been extended to include other kinds of nodes besides transformations, e.g. switch nodes