Skeletons and Skin

CMSC425.01 Spring 2019

Still at tables …

Administrivia

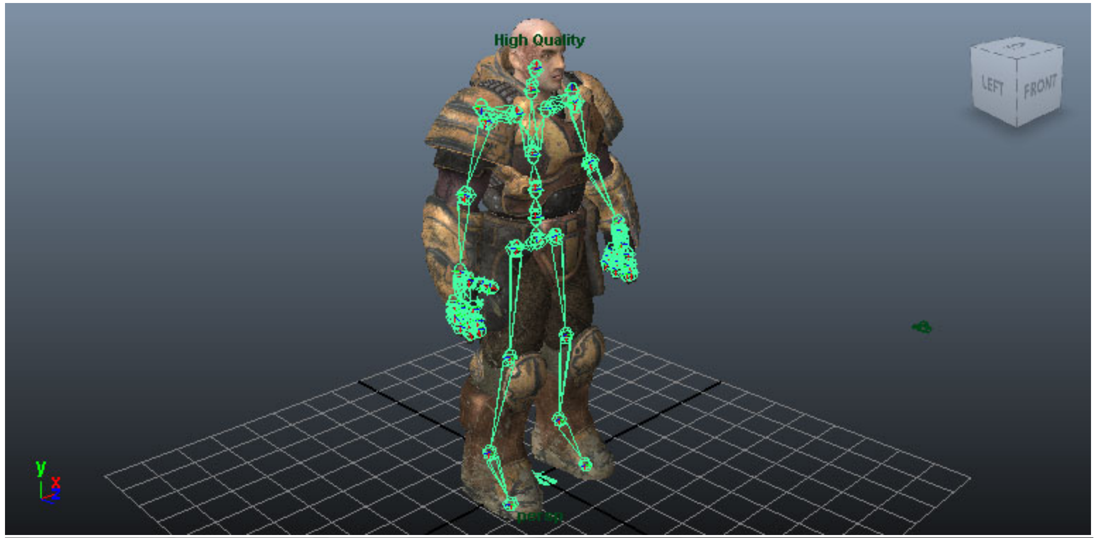
- Next Hw and Project 2 coming still coming …
- Mini-lectures still coming – videos on single topics (Panopto on Elms)
- The M-word – Midterm. Monday April 1st.

Today’s question

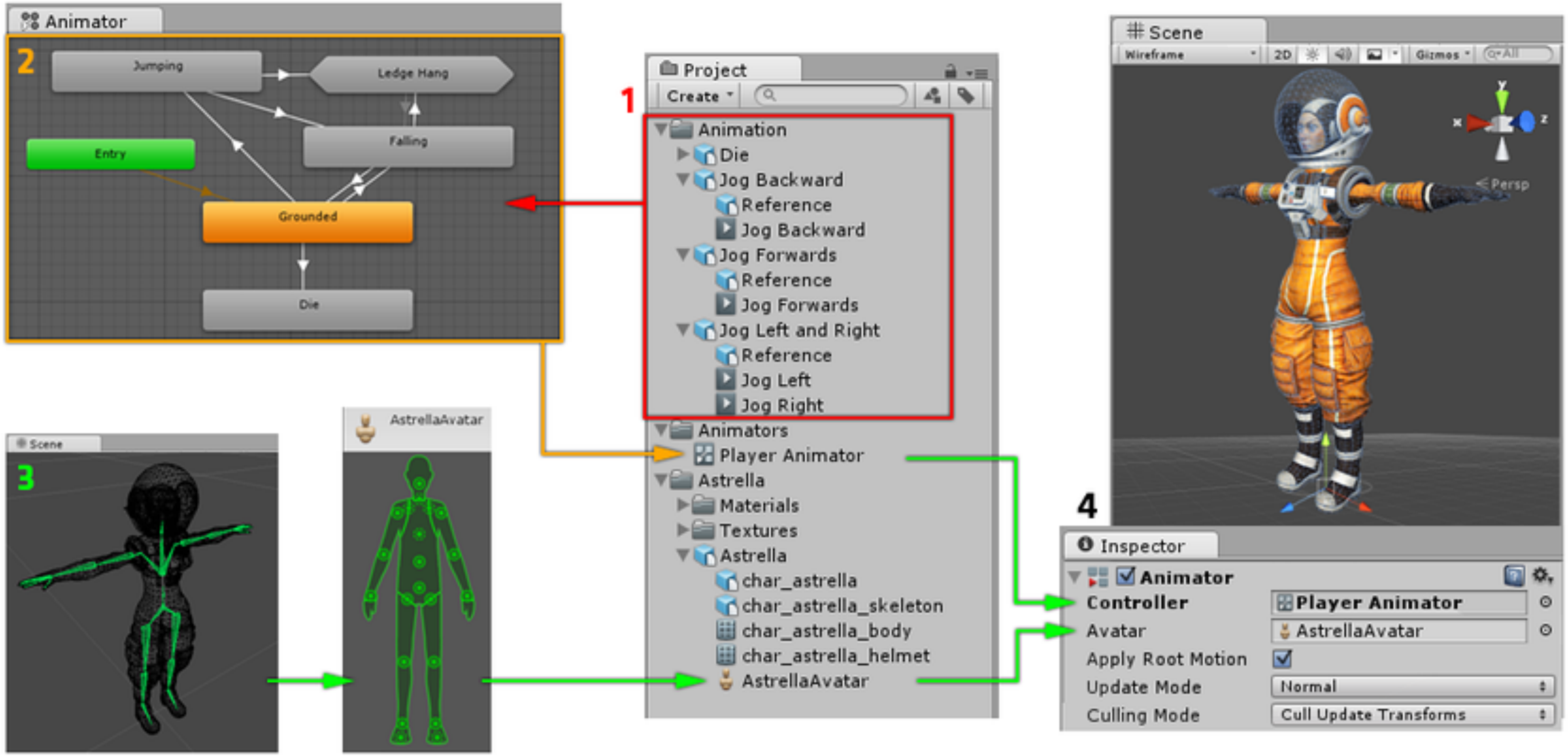
Animating articulated figures: Skeletons and Skin

Skeletons and rigging

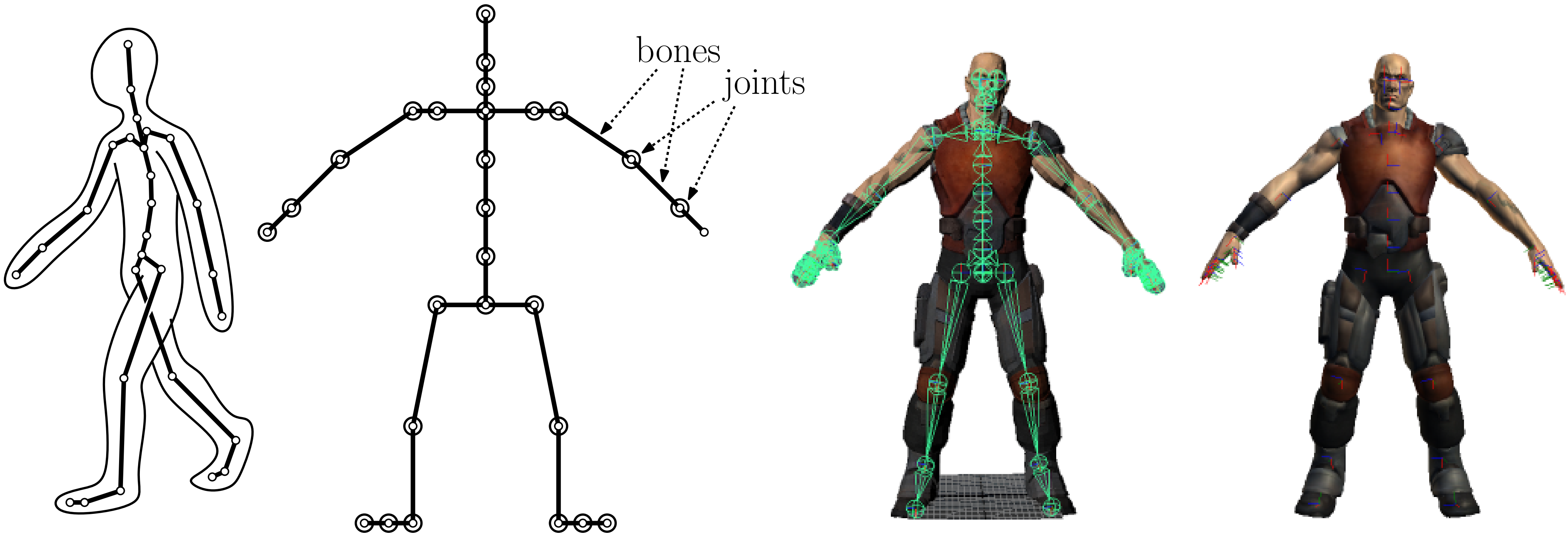
- Character animation
  - Create a skeleton
  - Define transforms between parts
  - Interpolate transforms to move
  - Rig with “flesh”
  - Create behavior animations
  - Blend between animations for smooth actions in game
- Can find as Unity Assets
- Use Mecanim tool
- https://www.youtube.com/watch?v=HPwu7eIwjV8



Unity: Mecanim

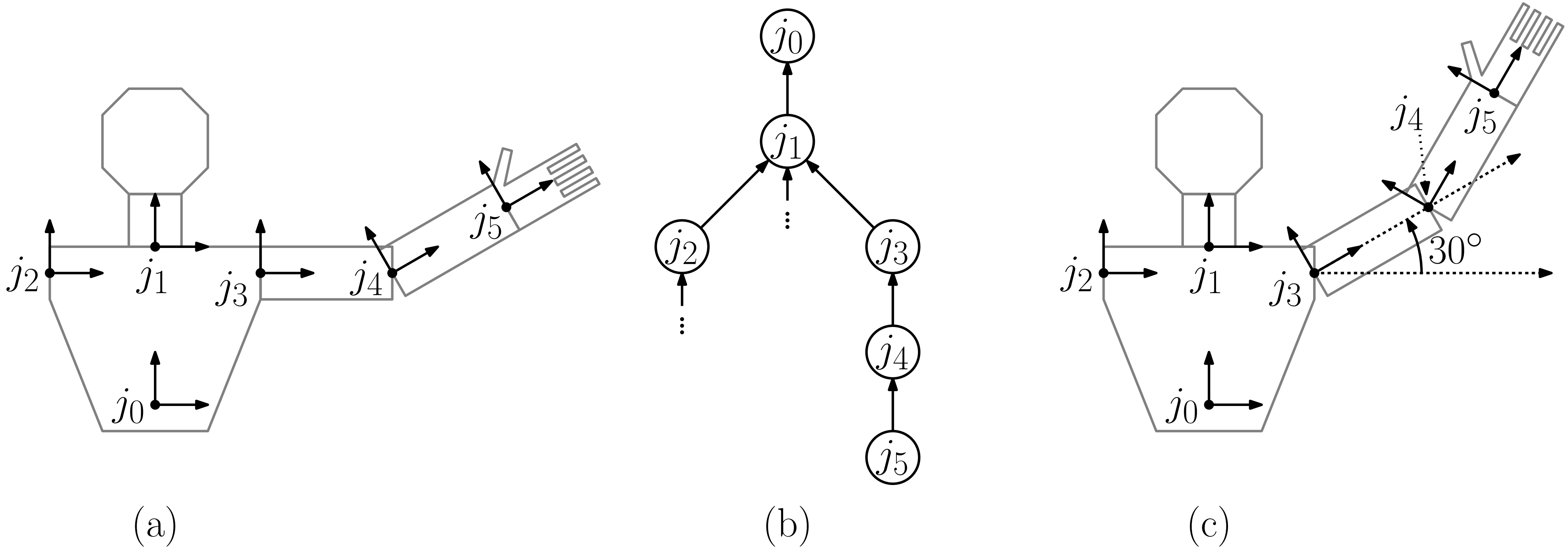


Skeleton: bones and joints, bind pose



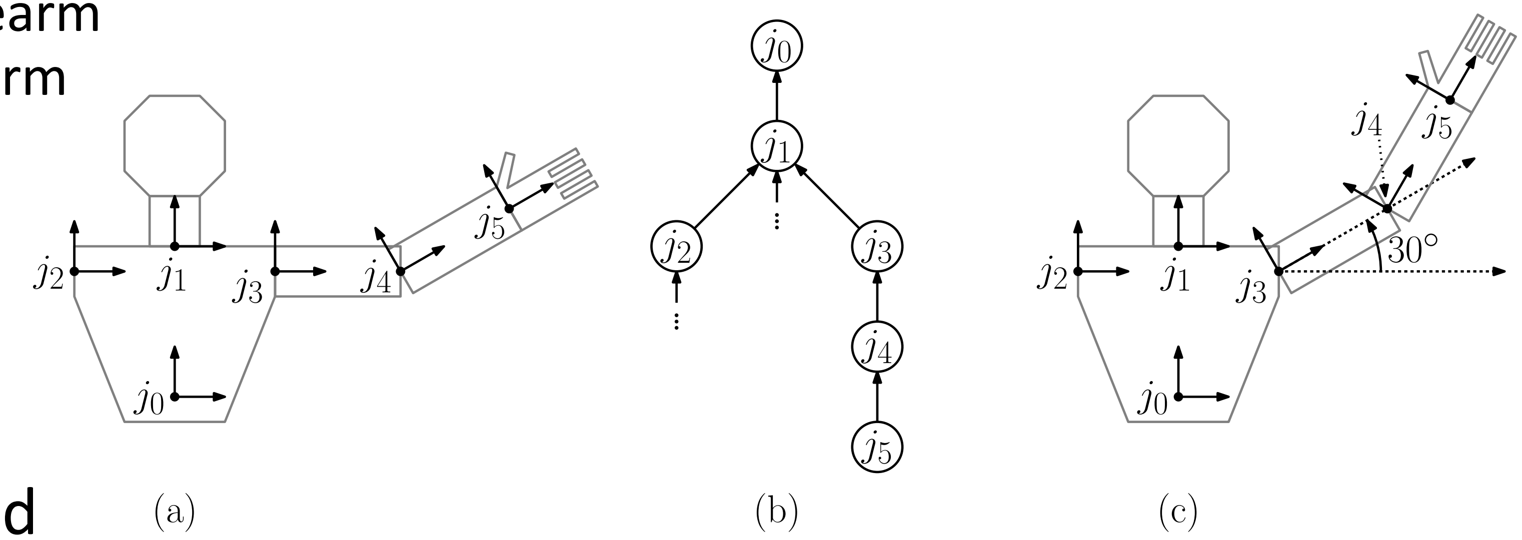
(a) (b) (c)

Skeletal model as joint tree



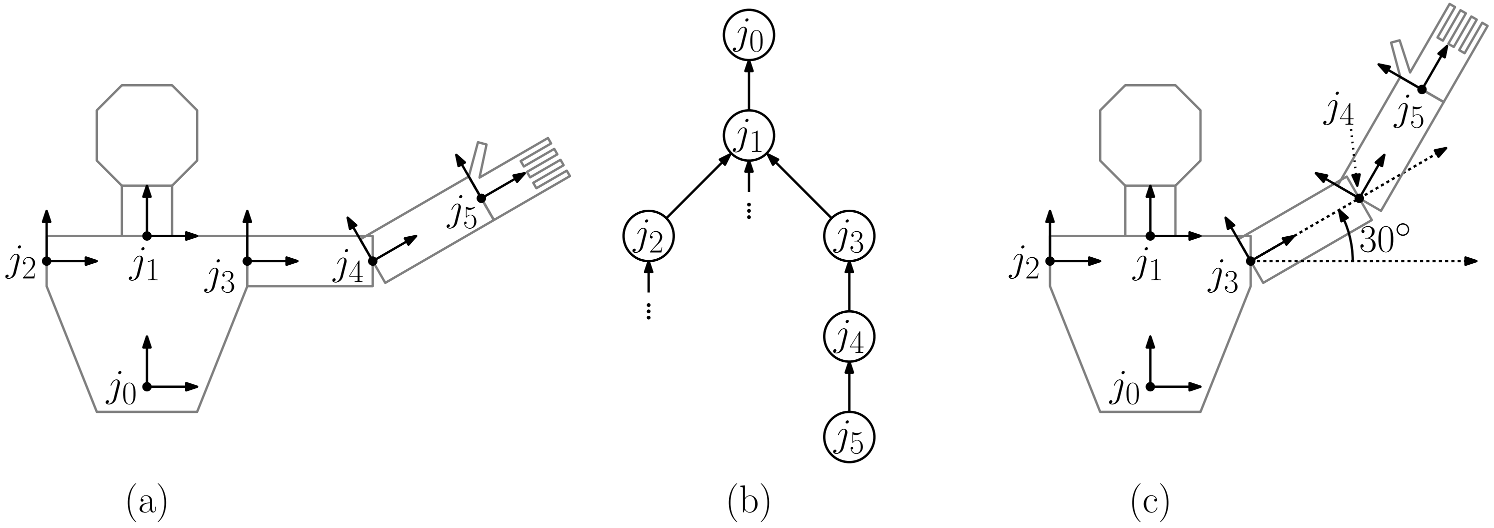
Motion as transform propagation

- Queen's wave
  - Use wrist (j5) to rotate hand
  - Use elbow (j4) to rotate forearm
  - Use shoulder (j3) to rotate arm
- Parent relation
  - p(j) = parent joint
  - p(j5) = j4
- Rotating parent rotates child
- Rotating child does not rotate parent



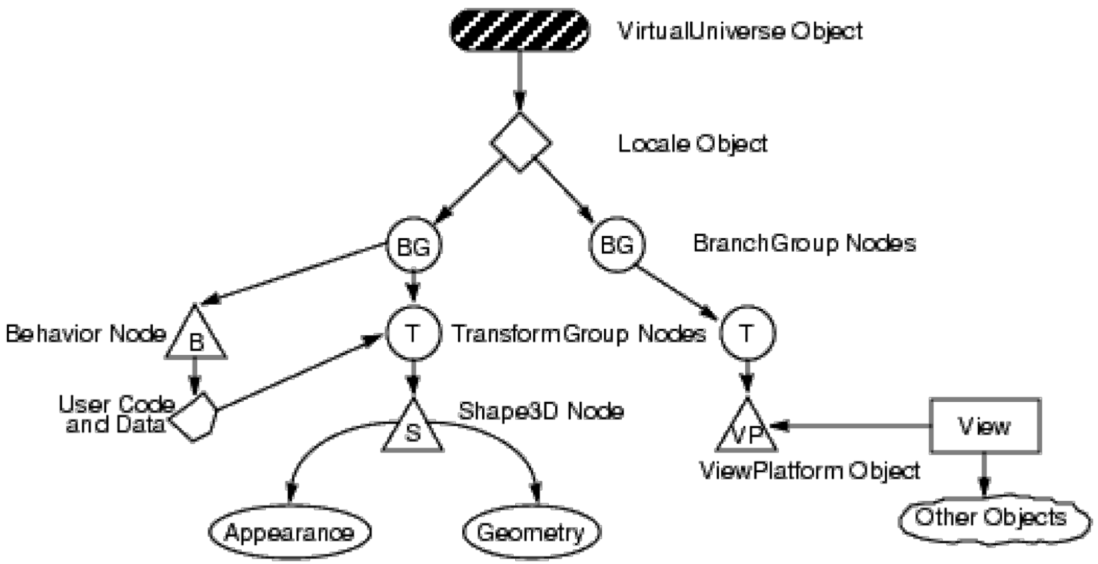
Traversing joint tree with transform stack

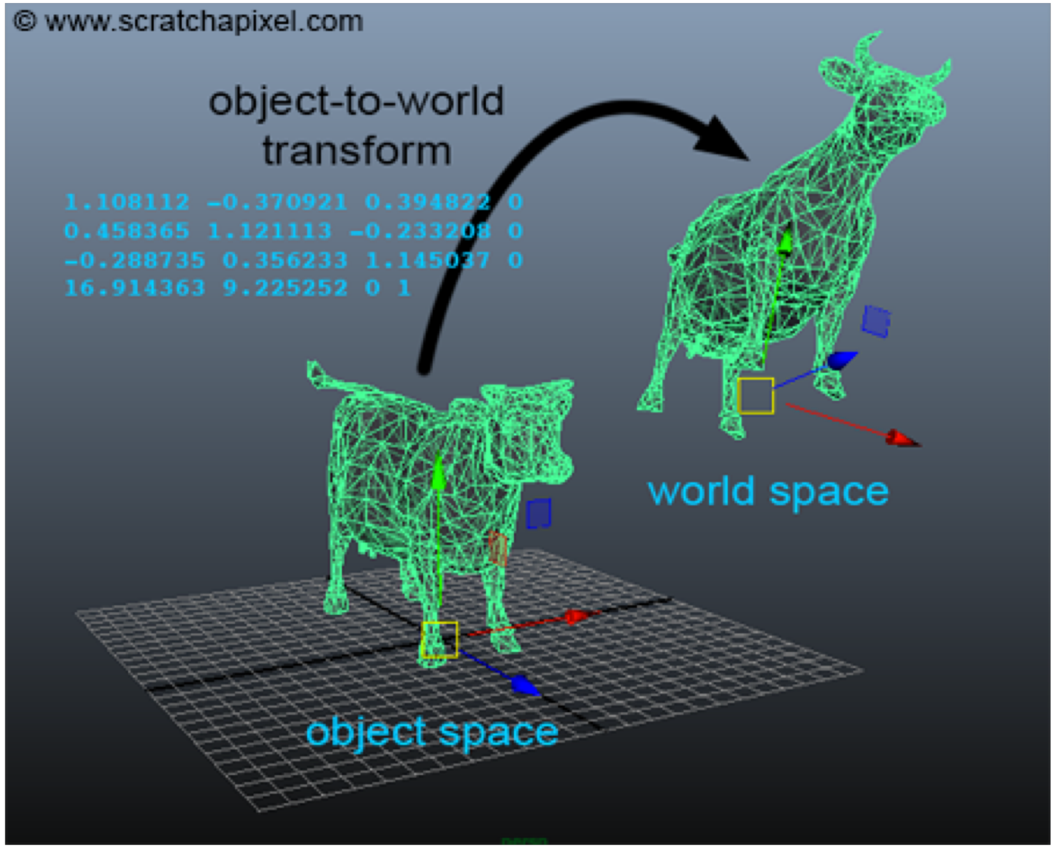
- Start with transform M = I (identity)
- Visit j0
  - M' = M*M_j0
- Visit j1
  - M'' = M*M_j1
  - Push M'' on stack
- Visit j2
  - M''' = M''*M_J2
  - Transform points attached to j2
  - Pop stack
- Visit j3
  - M'''' = M''*M_j3



- Multiply on right down a branch
- Push when need to revisit
- Apply M to points on branch

Scene graph – similar tree for all objects



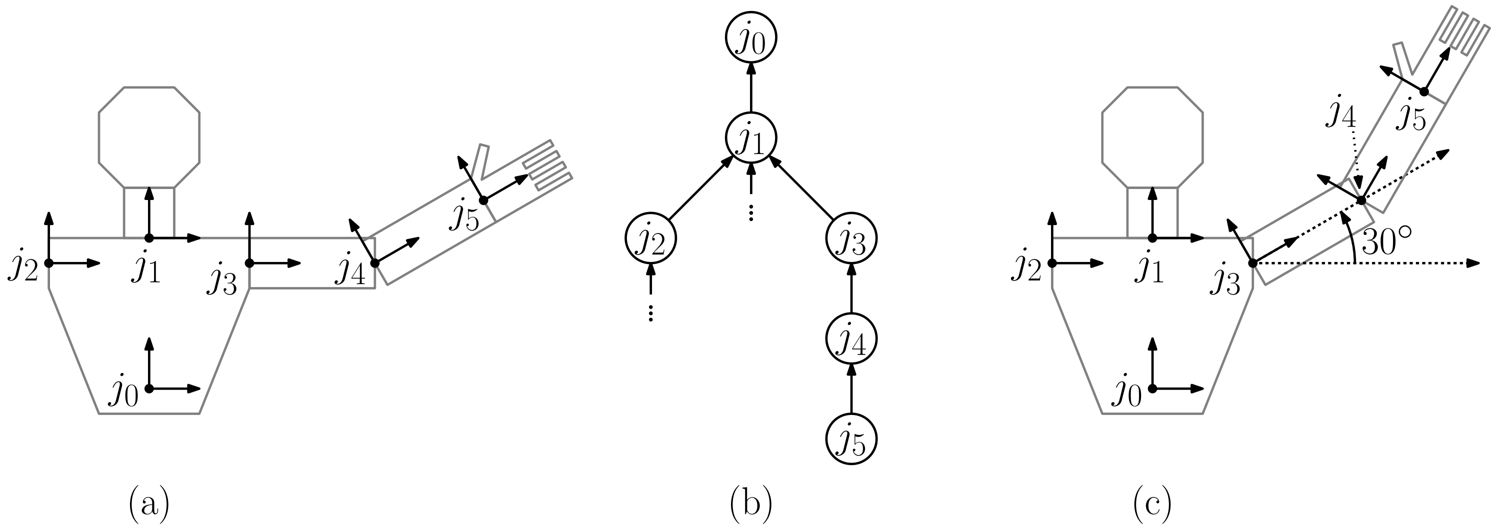


- Directed graph of all objects in scene
- Nodes have shape, appearance, transform, camera, light info

Joint constraints: degrees of freedom (DOF)

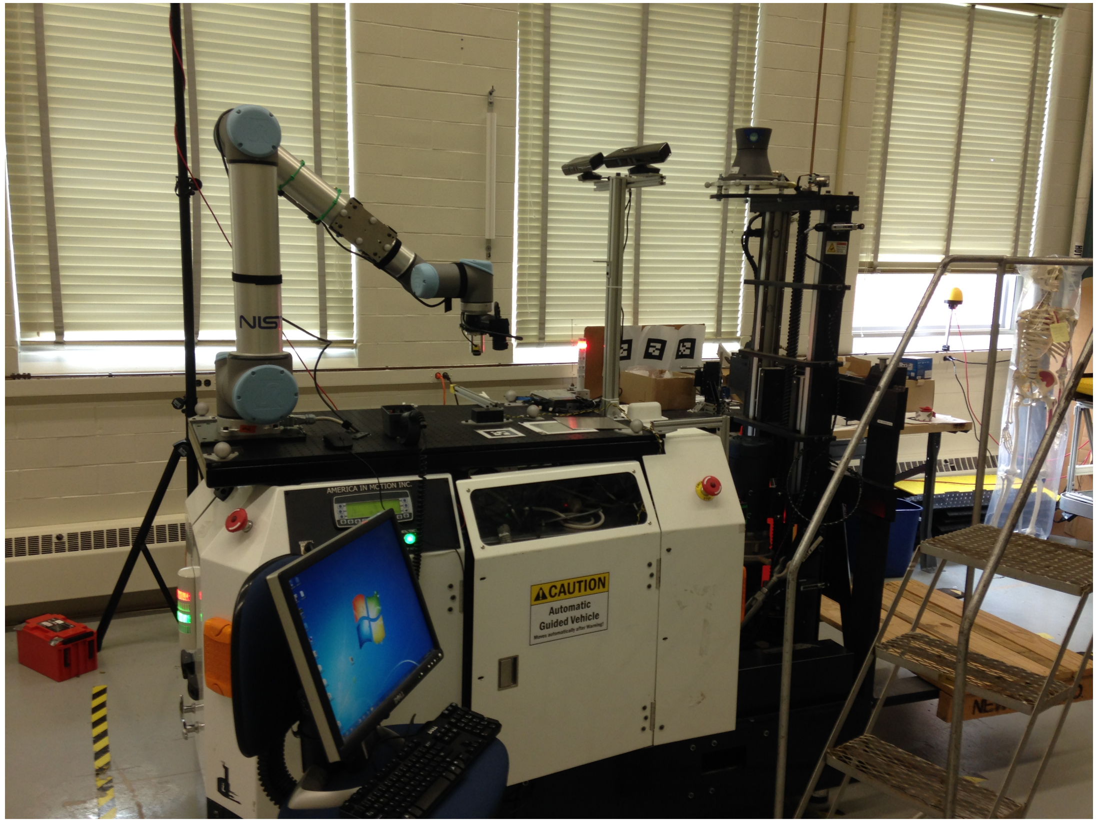
- Number of rotations supported by joint
  - Knee – 1 degree
  - Foot – 2 degrees

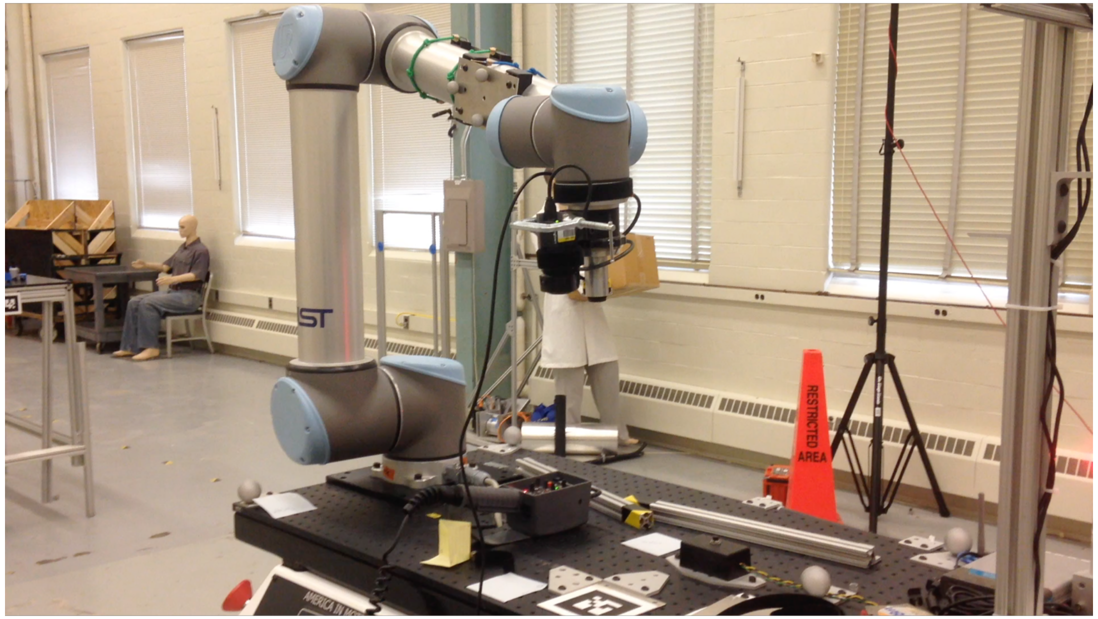
  - Wrist?
  - Elbow?
  - Shoulder?



- Limits on each joint
  - Rotation in range [angle1,angle2]

Aside: Learn figure animation, learn robots

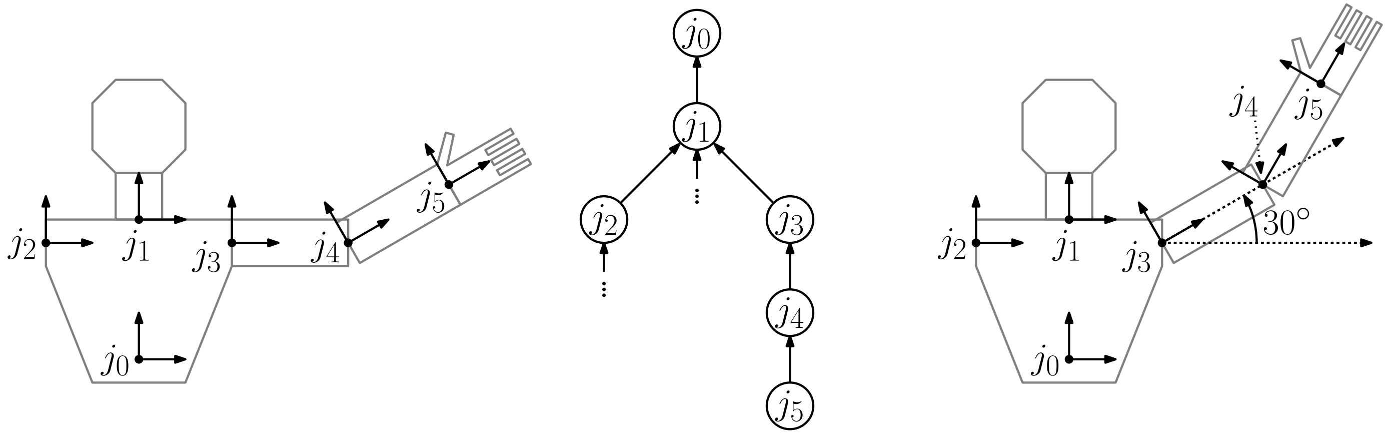




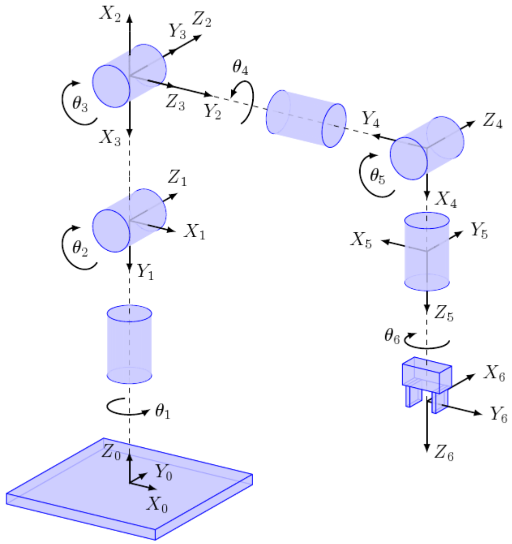
Kinematics – study of motion w/out forces

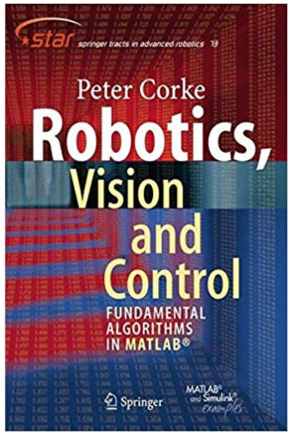
- ***Kinematics***
  - Forward – given joints and transformations, estimate end position
  - Reverse – given end position estimate transformations

- Forward – "easy"
- Reverse – hard!



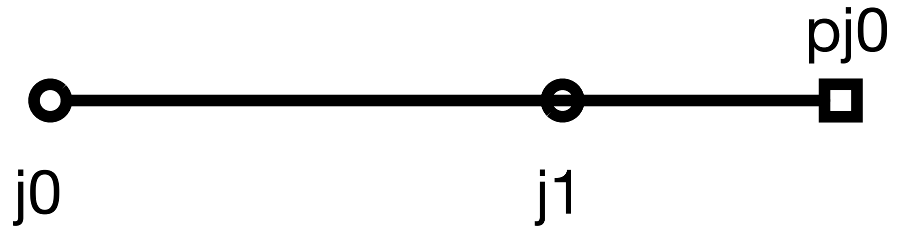
(a) (b) (c)





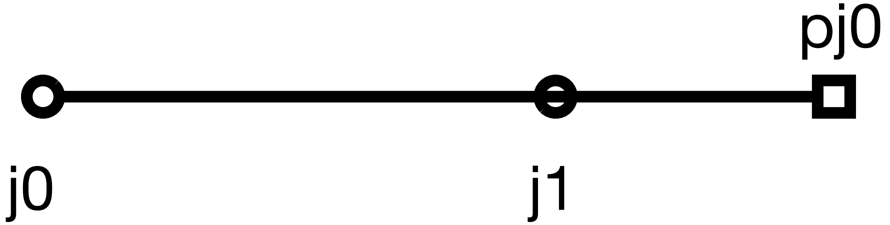
Joint transformations: simple

- Initial: resting pose
- $T_{[j1 \leftarrow j0]} = M_{T(2,0)}$
- $T_{[j1 \leftarrow j0]} * \mathrm{p} = M_{T(2,0)} * (1,0) = (3,0)$

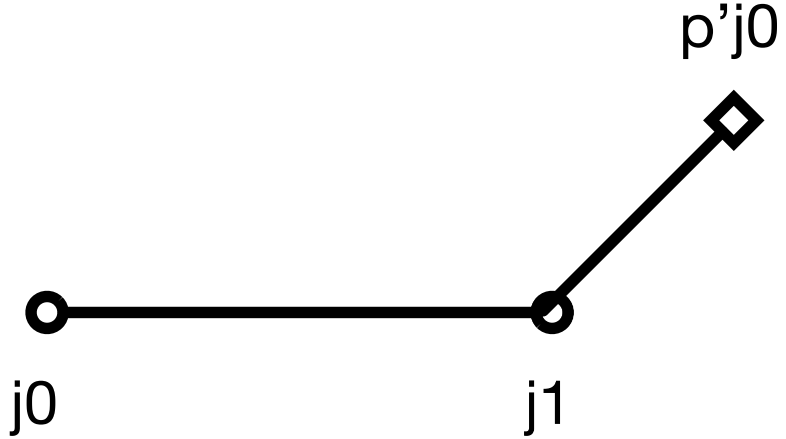


Joint transformations: simple

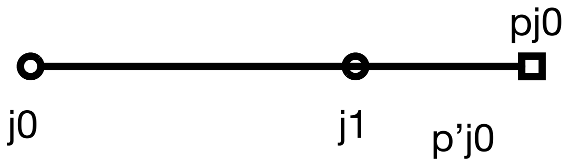
- Initial: resting pose
- $T_{[j1 \leftarrow j0]} = M_{T(2,0)}$
- $T_{[j1 \leftarrow j0]} * \mathrm{p} = M_{T(2,0)} * (1,0,1) = (3,0,1)$



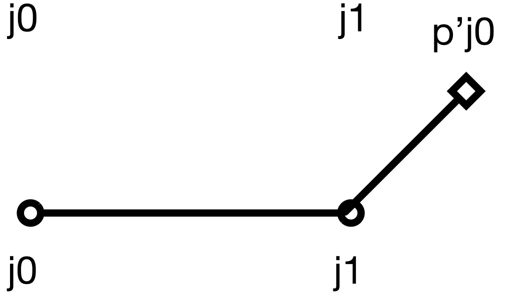
- Rotate wrist 45 degrees in j0 coordinates
- $M_{R(45)} * (1,0,1) = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 1)$



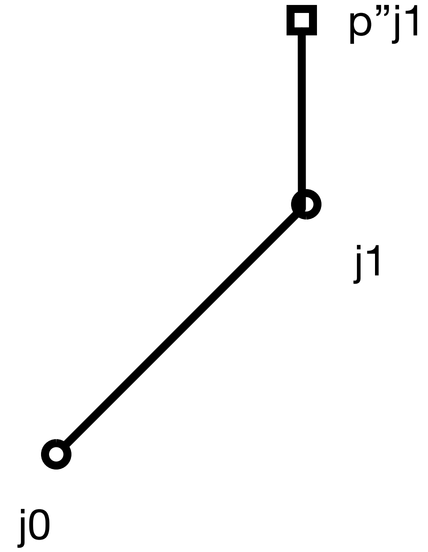
Joint transformations: simple



- Initial: resting pose
- $T_{[j1 \leftarrow j0]} = M_{T(2,0)}$
- $T_{[j1 \leftarrow j0]} * p_{j0} = M_{T(2,0)} * (1,0,1) = (3,0,1)$

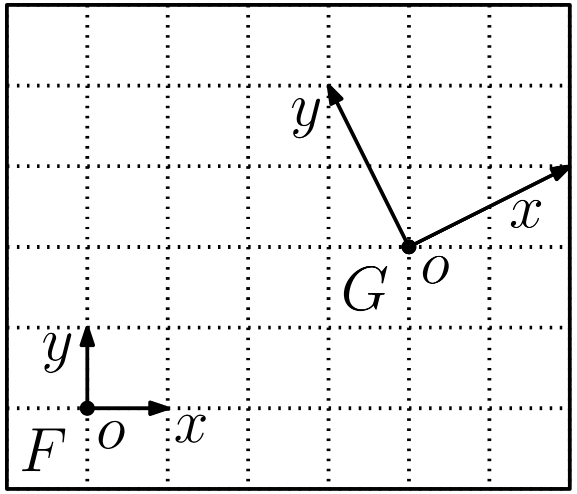


- Rotate wrist 45 degrees in j0 coordinates
- $M_{R(45)} * (1,0,1) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 1\right)$



- Rotate shoulder 45 degrees in j1 coordinates
- $p''_{j1} = M_{R(45)} * T_{[j1 \leftarrow j0]} * p'_{j0} = (\sqrt{2}, 1 + \sqrt{2}, 1)$

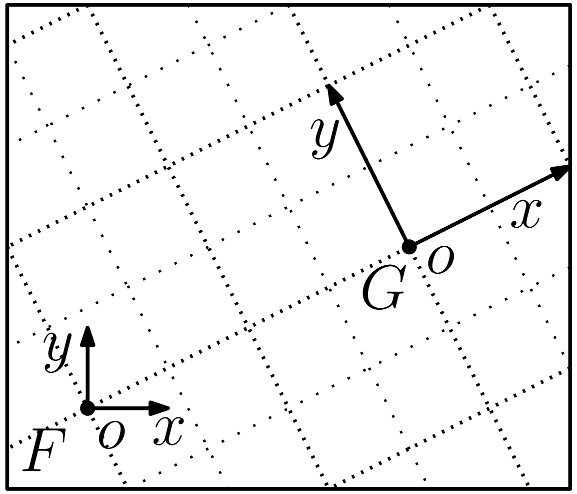
Coordinate transformations – points (location)



$$G.x_{[F]} = (2, 1, 0)$$

$$G.y_{[F]} = (-1, 2, 0)$$

$$G.o_{[F]} = (4, 2, 1)$$

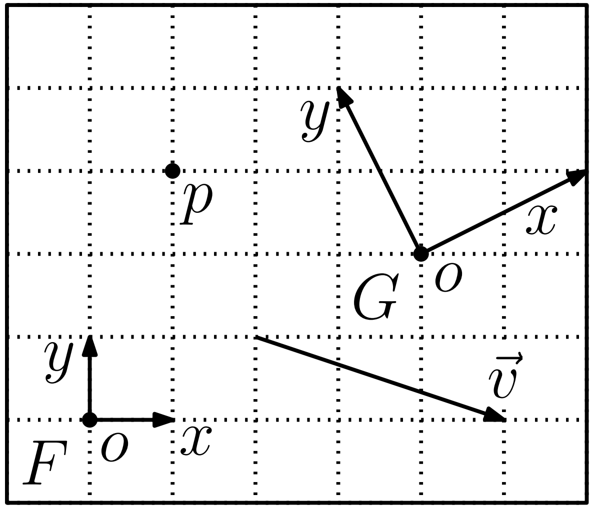


$$F.x_{[G]} = \left(\frac{2}{5}, -\frac{1}{5}, 0\right)$$

$$F.y_{[G]} = \left(\frac{1}{5}, \frac{2}{5}, 0\right)$$

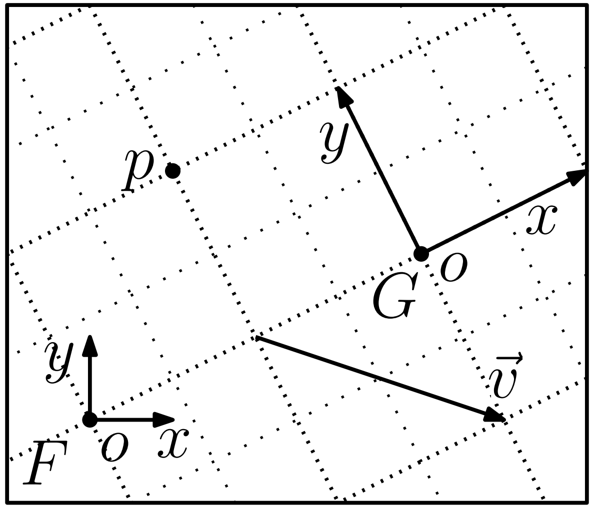
$$F.o_{[G]} = (-2, 0, 1)$$

Coordinate transforms – vectors (orientation)



$$p_{[F]} = (1, 3, 1)$$

$$\vec{v}_{[F]} = (3, -1, 0)$$

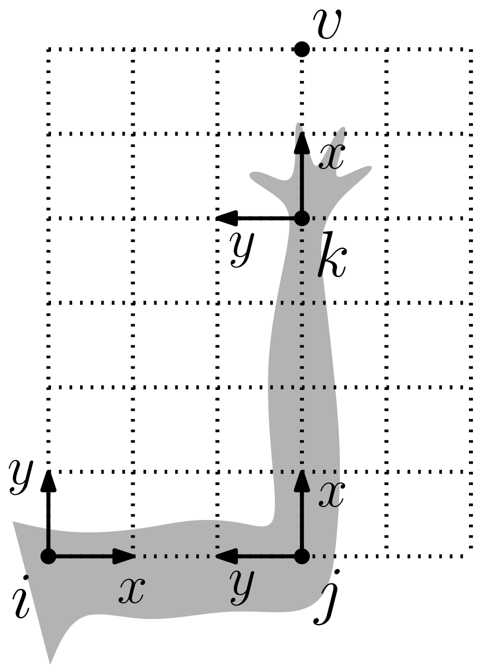


$$p_{[G]} = (-1, 1, 1)$$

$$\vec{v}_{[G]} = (1, -1, 0)$$

Arm example

- Three joints
  - Wrist: k T[k <- j]
  - Elbow: j T[j <- i]
  - Shoulder: I
- Binding pose
  - Translations
  - One reflection



(a)

$$v_{[k]} = (2, 0, 1)$$

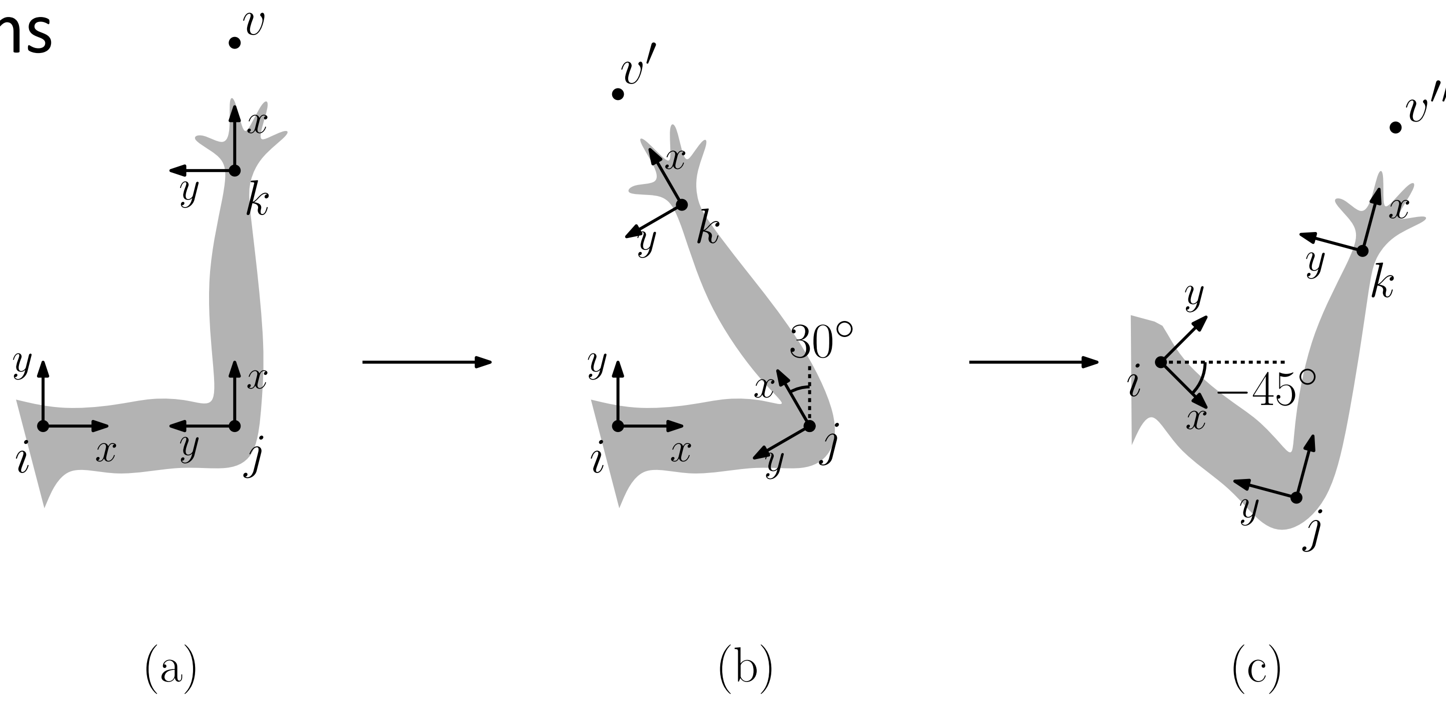
$$v_{[j]} = (6, 0, 1)$$

$$v_{[i]} = (3, 6, 1)$$

(b)

Forward kinematics – rotate elbow, shoulder

- Have binding transforms
  - T[k <- j]
  - T[j <- i]
- Have two rotations
  - M_R(30)
  - M_R(45)
- Apply in what order?



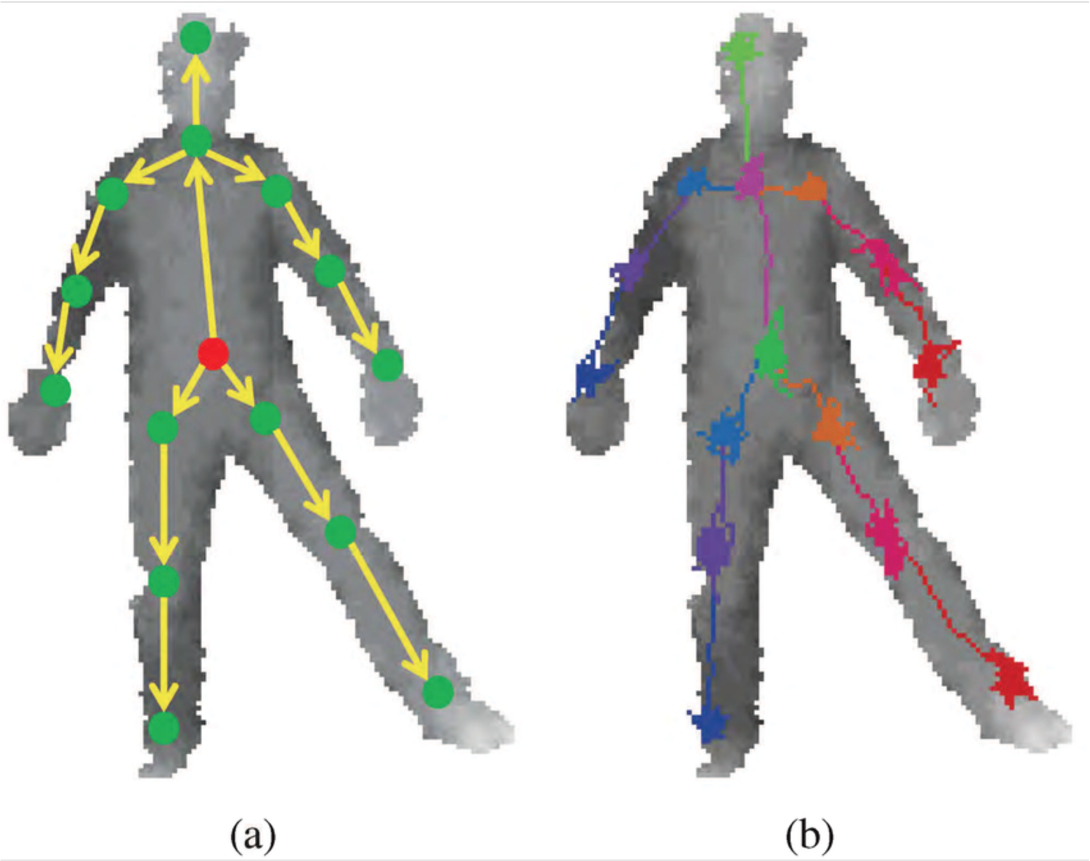
Summary

- Animated character has

  Skeleton which has

  Joints which have

  Transforms

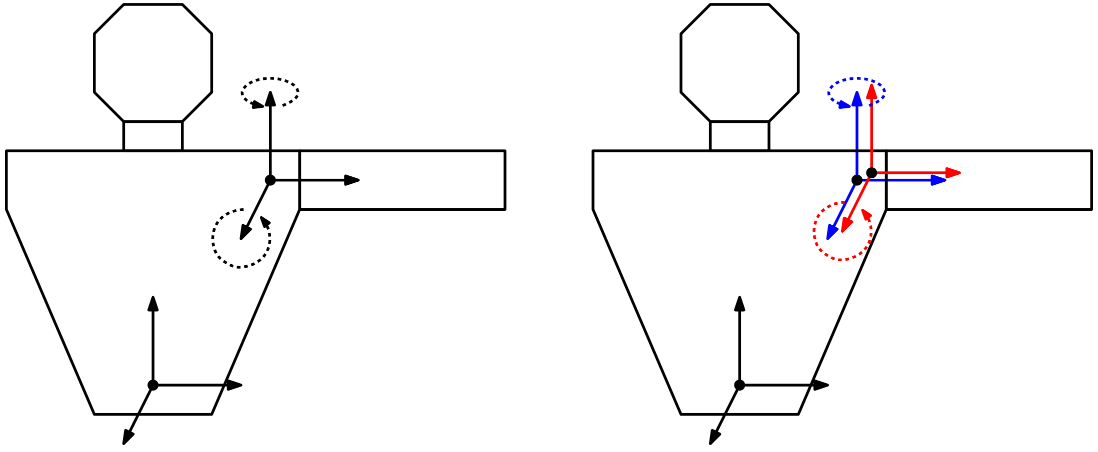


Meta joints

- Collocated joints

- Simplify transforms
- Each joint has rotation around one axis - 1 DOF
- Combine to get multiple DOFs

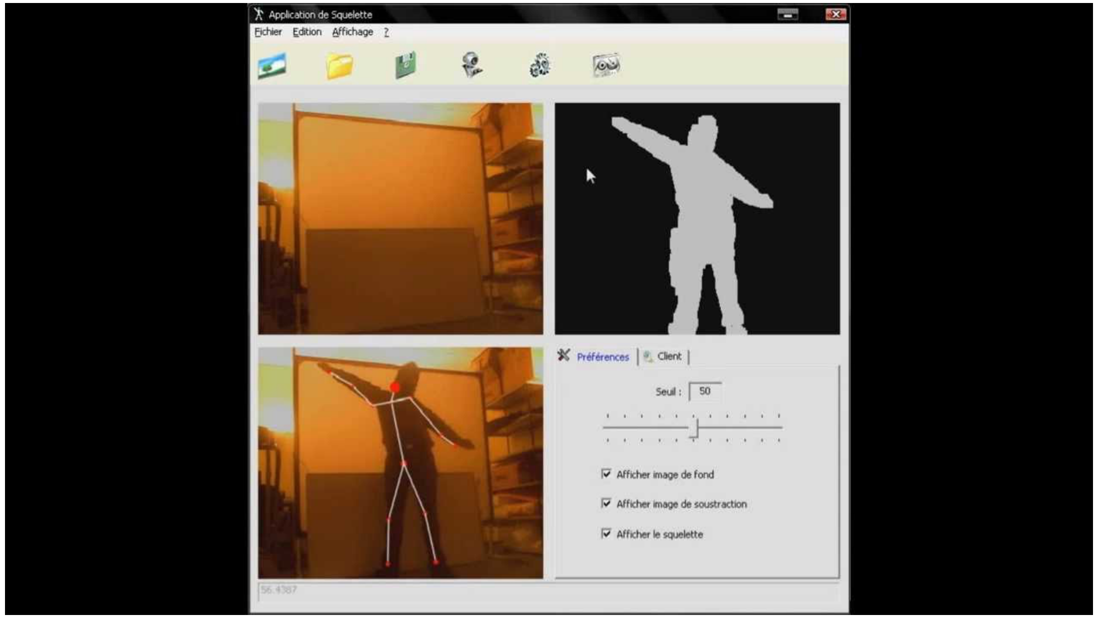
- No translation



(a) (b)

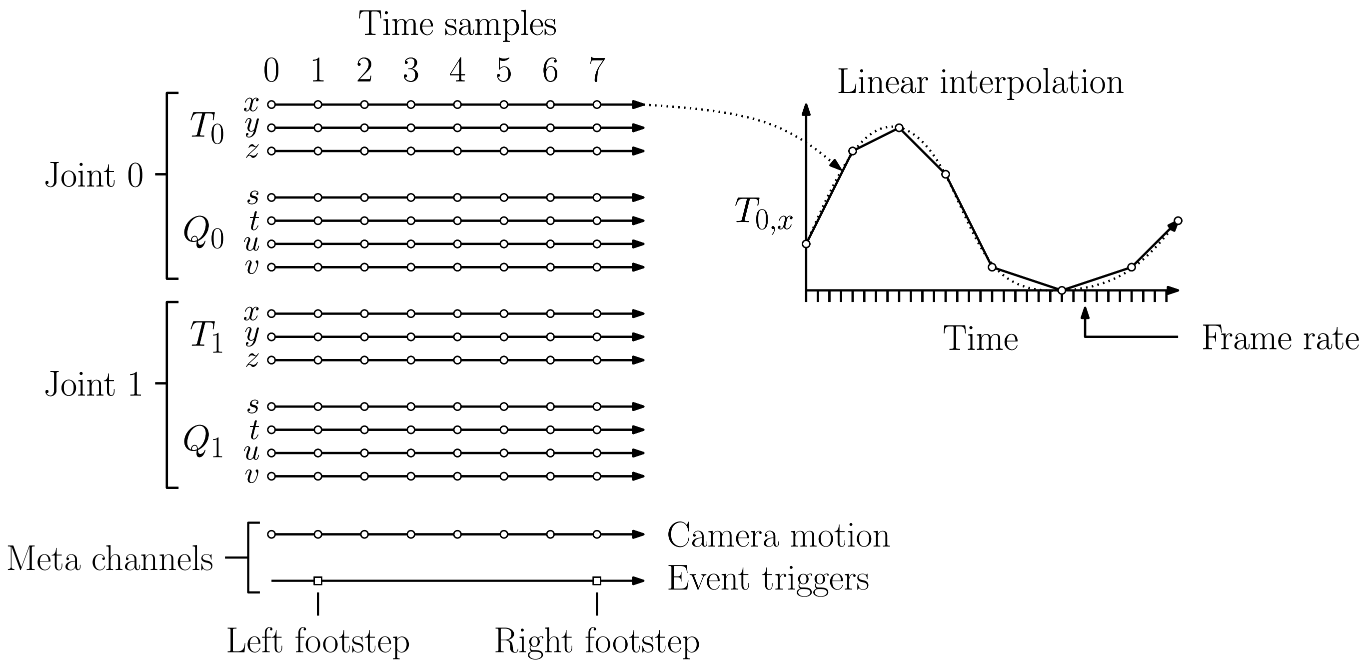
Animating skeletons

- Key framing.
- Motion capture.
  - https://www.youtube.com/watch?v=tNqGT2wnNSM
- Goal oriented.
- Also – parametric equations
  - Pick ups in Project 1b



Data representation of motion/animation

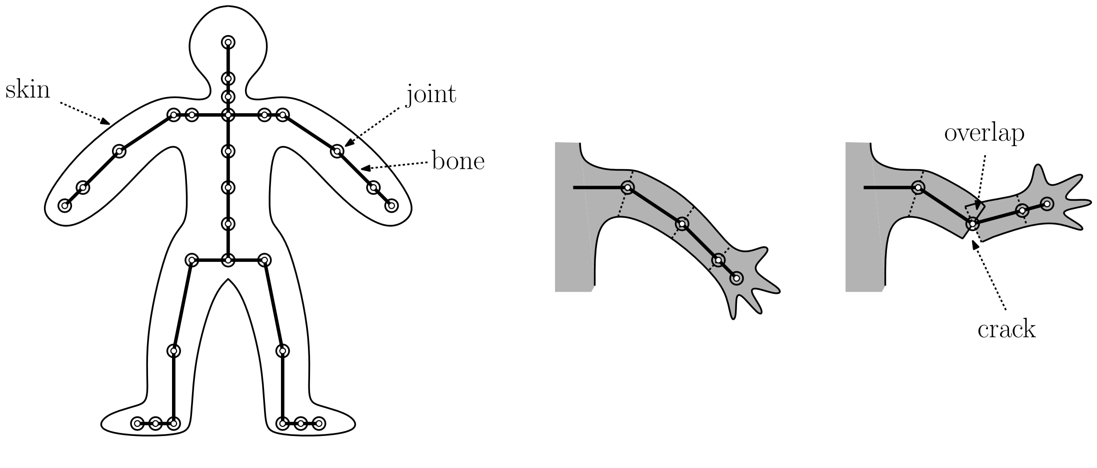
- Joint positions over time
  - T – translation
  - Q – Quaternion
- Interpolation between key frames/samples
  - Cubic for position
  - Spherical for quaternions



Skinning

- Bind mesh to bone between joints
- Moves with parent joint

- Problems
  - Cracking and distortion



(a) (b)

Skinning

- Bind mesh to bone between joints
- Moves with parent joint

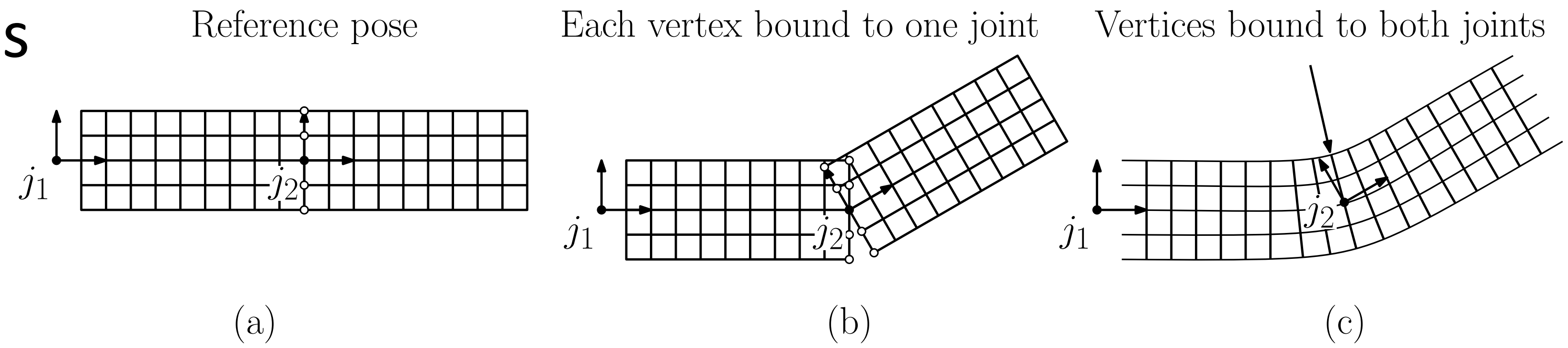
- Problems
  - Cracking and distortion

- Cheats!
  - Use fantasy character with disconnected parts
  - Use robot with mechanical joints that require no skinning

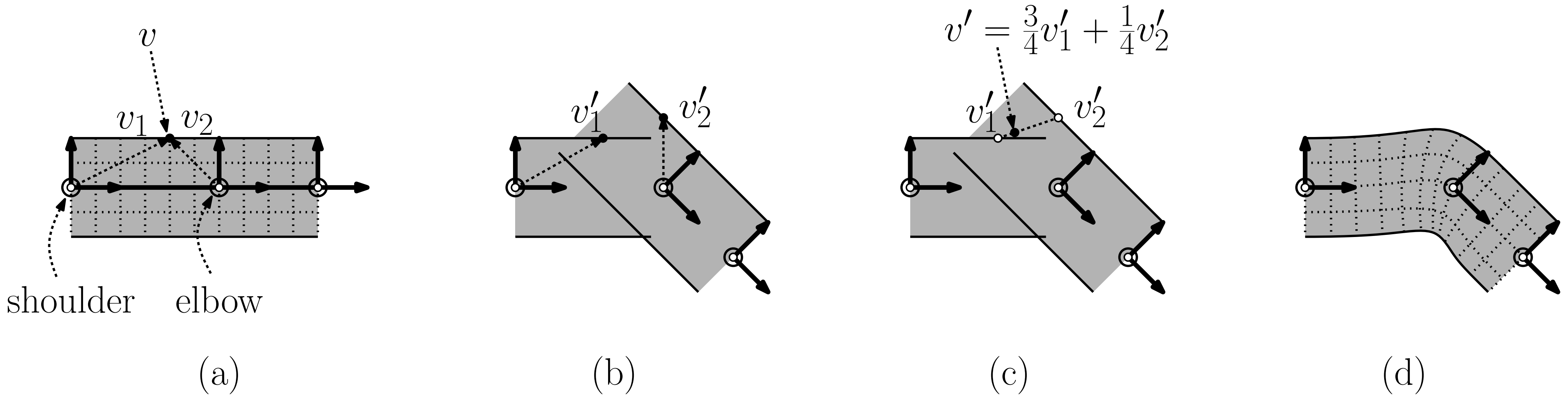
Blending at joints

- Bind mesh vertices to ***one*** joint
- Move with joint (bone between)
- Cracks!

- Bind to ***two*** joints
- Interpolate



Weighted linear blending



State of art rigging – muscles and more