Topics

1. Introduction: What is Computer Graphics?
2. Raster Images (image input/output devices and representation)
3. Scan conversion (pixels, lines, triangles)
4. Ray Casting (camera, visibility, normals, lighting, Phong illumination)
5. Ray Tracing (shadows, supersampling, global illumination)
6. Spatial Data Structures (AABB trees, OBB, bounding spheres, octree)
7. Meshes (connectivity, smooth interpolation, uv-textures, subdivision, Laplacian smoothing)
8. 2D/3D Transformations (Translate, Rotate, Scale, Affine, Homography, Homogeneous coordinates)
9. Viewing and Projection (matrix composition, perspective, Z-buffer)
10. Shader Pipeline (Graphics Processing Unit)
11. Animation (kinematics, keyframing, Catmull-Romm interpolation, physical simulation)
12. 3D curves and objects (Hermite, Bezier, cubic curves, curve continuity, extrusion/revolve surfaces)
13. Advanced topics overview

Topic 11.

Animation

Animation Timeline

1908: Emile Cohl (1857-1938) France, makes his first film, FANTASMAGORIE, arguably the first animated film.

1911: Winsor McCay (1867-1934) makes his first film, LITTLE NEMO. McCay, already famous for comic strips, used the film in his vaudeville act. His advice on animation:

Any idiot that wants to make a couple of thousand drawings for a hundred feet of film is welcome to join the club.

1928: Walter Disney (1901-1966) working at the Kansas City Slide Company creates Mickey Mouse.

1974: First Computer animated film “Faim” from NFB nominated for an Oscar.

Animation Principles

Squash & Stretch

Timing

Ease-In & Ease-Out

Arcs

Anticipation

Follow-through & Secondary Motion

Overlapping Action & Asymmetry

Exaggeration

Staging

Appeal

Straight-Ahead vs. Pose-to-Pose

Squash and Stretch

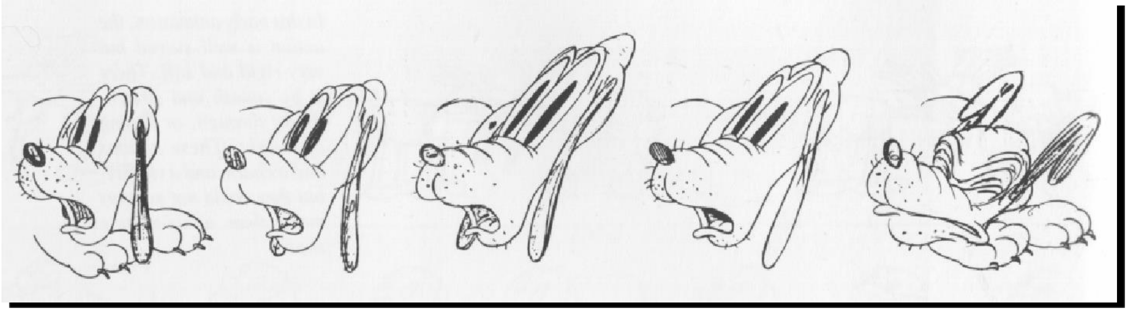
Rigid objects look robotic: deformations make motion natural

Accounts for physics of deformation

- Think squishy ball…
- Communicates to viewer what the object is made of, how heavy it is, …
- Usually large deformations conserve volume: if you squash one dimension, stretch in another to keep mass constant

Also accounts for persistence of vision

- Fast moving objects leave an elongated streak on our retinas



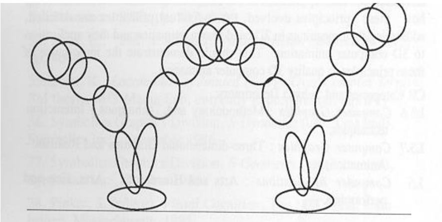


FIGURE 2. Squash & stretch in bouncing ball.

Anticipation

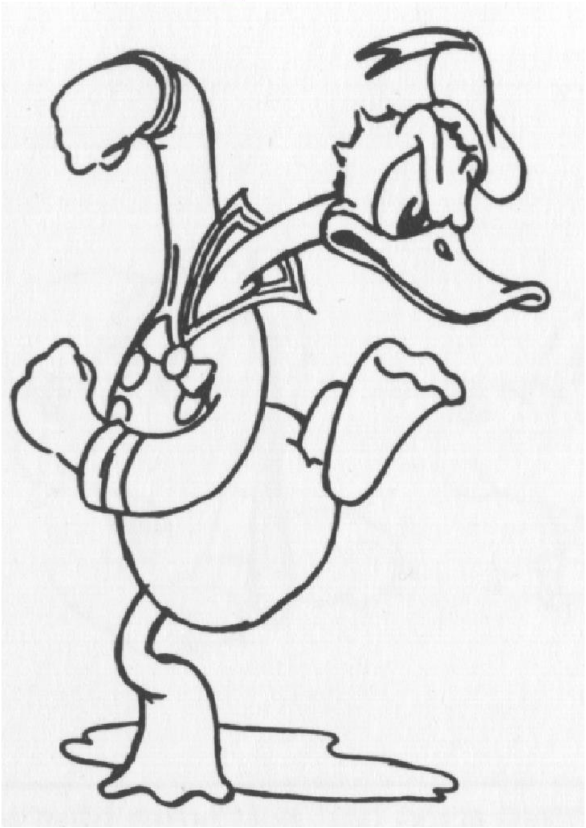
The preparation before a motion

- E.g. crouching before jumping, pitcher winding up to throw a ball

Often physically necessary, and indicates how much effort a character is making

Also essential for controlling the audience’s attention, to make sure they don’t miss the action

- Signals something is about to happen, and where it is going to happen.



Animation Principles

Squash & Stretch

Timing

Ease-In & Ease-Out

Arcs

Anticipation

Follow-through & Secondary Motion

Overlapping Action & Asymmetry

Exaggeration

Staging

Appeal

Straight-Ahead vs. Pose-to-Pose

What can be animated?

Lights

Camera

Jointed figures

Deformable objects

Clothing

Skin/muscles

Wind/water/fire/smoke

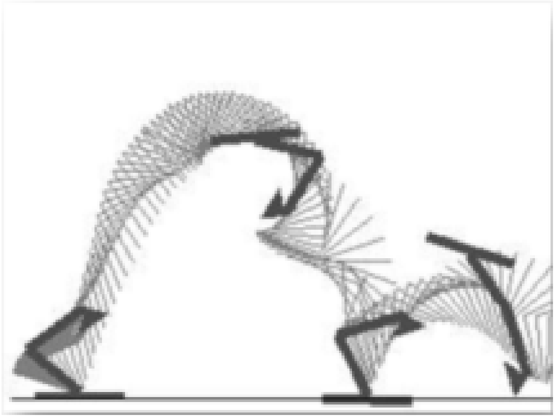
Hair

…any variable, Given the right time scale, almost anything…

Elements of CG (animation)

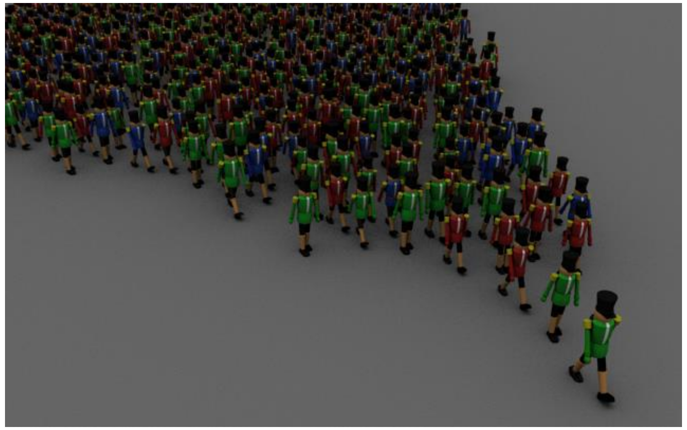
How does one make digital models move?

Keyframing



Physical simulation

Motion capture



Behavior rules

Keyframes

Keyframes, also called extremes, define important poses of a character:

Jump example:

- the start
- the lowest crouch
- the lift-off
- the highest part
- the touch-down
- the lowest follow-through

- Frames in between (“inbetweens”) introduce nothing new to the motion.
- May add additional keyframes to add some interest, better control the interpolated motion.

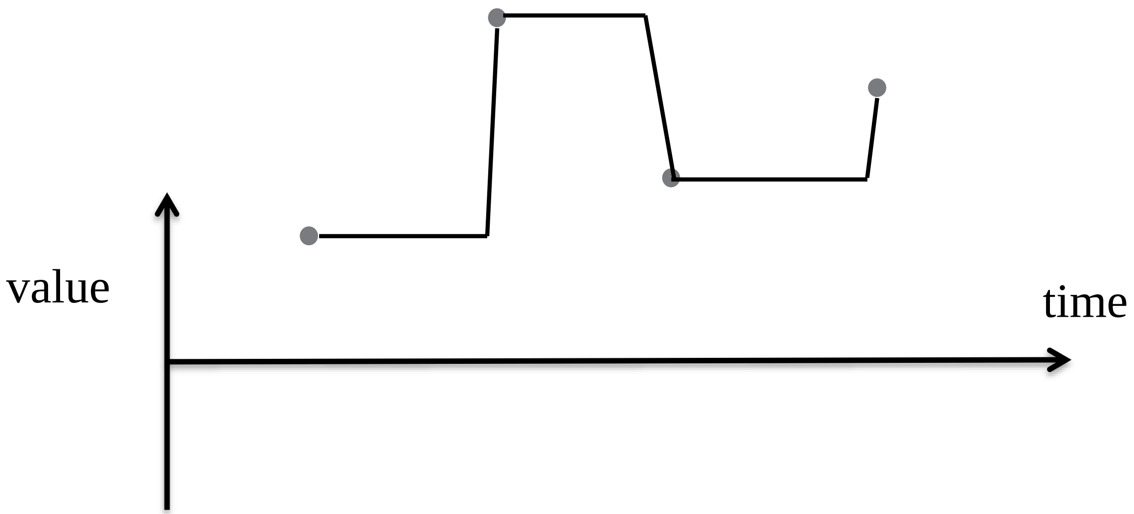
Keyframe Animation

The task boils down to setting animated variables (e.g. positions, angles, sizes, …) at each frame.

Straight-ahead: set variables in frame 0, then frame 1, frame 2, … forward in time.

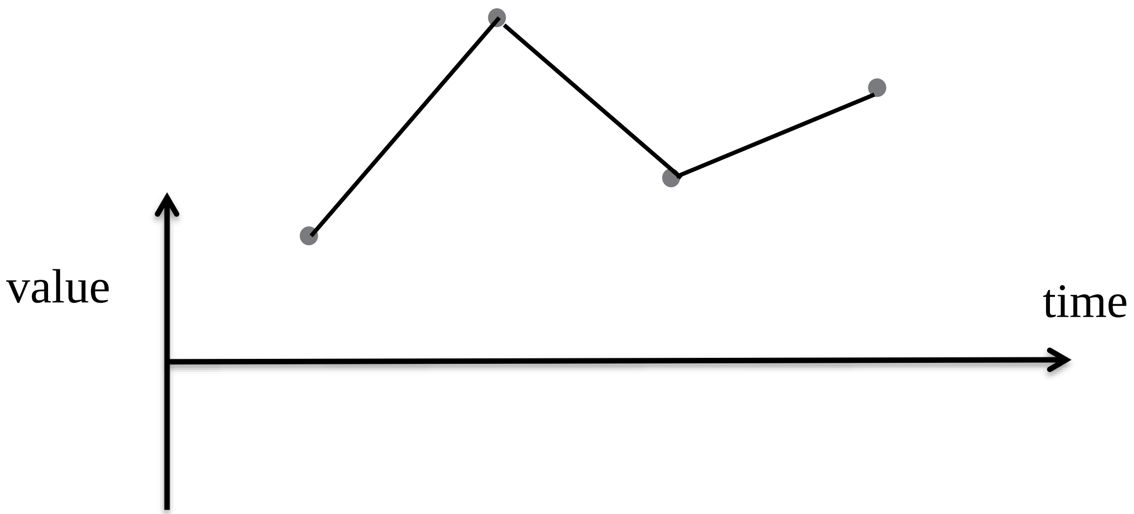
Pose-to-pose: set the variables at keyframes, let the computer smoothly interpolate values for frames in between.

Keyframe: Interpolation



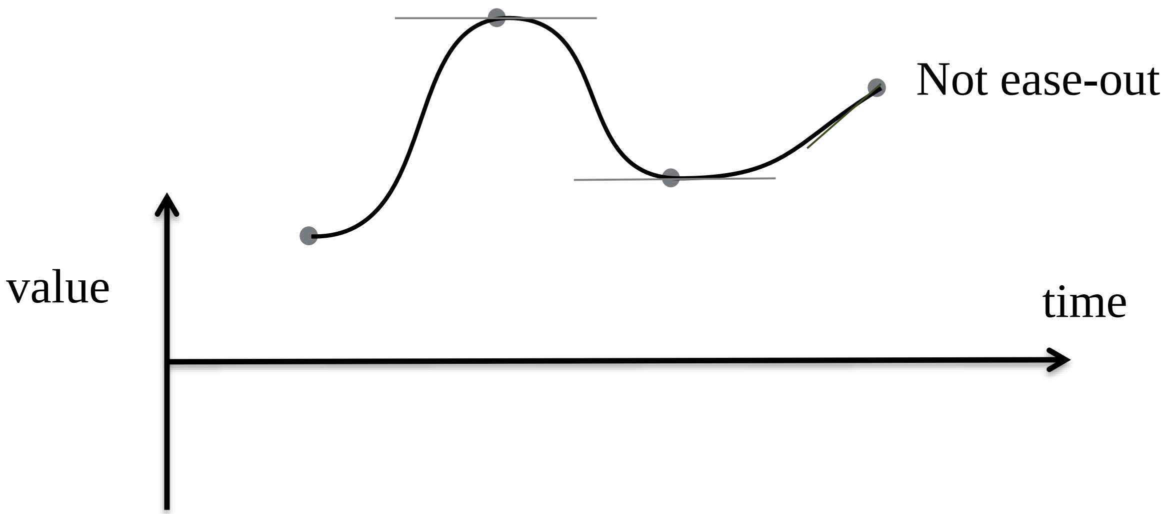
Hold

Keyframe: Interpolation



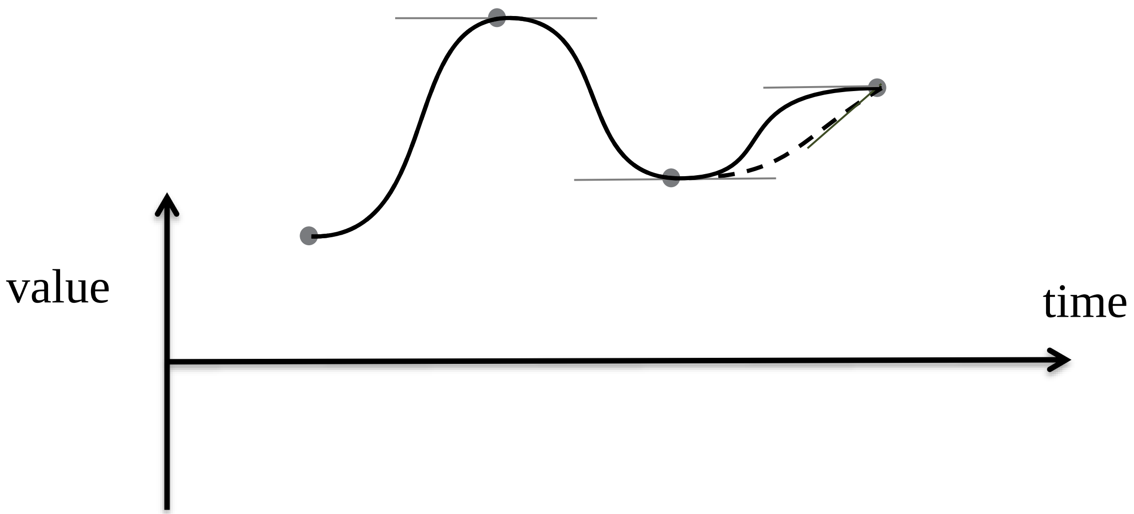
Linear

Keyframe: Interpolation



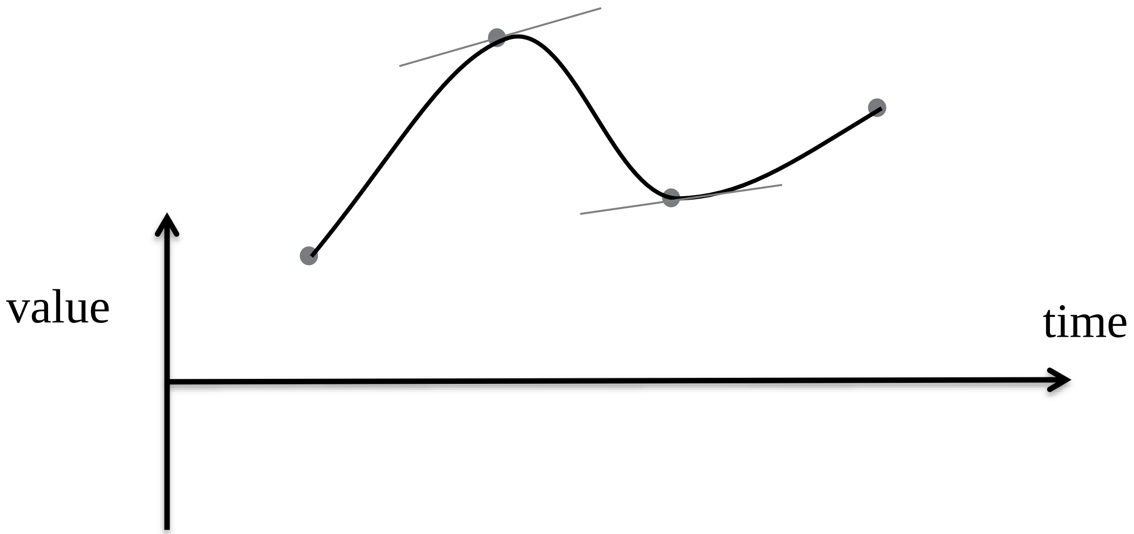
Ease-in Ease-out

Keyframe: Interpolation



Ease-in Ease-out

Keyframe: Interpolation



Spline

Designing Polynomial Curves from constraints

$p(t) = TA$, where T is powers of t. for a cubic $T=[t^3\ t^2\ t^1\ 1]$.

Written with geometric constraints $p(t) = TMG$, where M is the **Basis matrix** of a design curve and G the specific design constraints.

An example of constraints for a cubic Hermite for eg. are end points and end tangents. i.e. p_0, p'_0 at t=0 and p_1, p'_1 at t=1. Plugging these constraints into $p(t) = TA$ we get.

$$
\begin{array}{ll}
 & \text{B} \\
p(0) = p_0 = & [\ 0\ 0\ 0\ 1\ ]\ A_h \\
p(1) = p_1 = & [\ 1\ 1\ 1\ 1\ ]\ A_h \\
p'(0) = p'_0 = & [\ 0\ 0\ 1\ 0\ ]\ A_h \\
p'(1) = p'_1 = & [\ 3\ 2\ 1\ 0\ ]\ A_h
\end{array}
\quad \Rightarrow \quad G=BA,\ A=MG \Rightarrow M=B^{-1}
$$

Hermite Basis Matrix

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1}$$

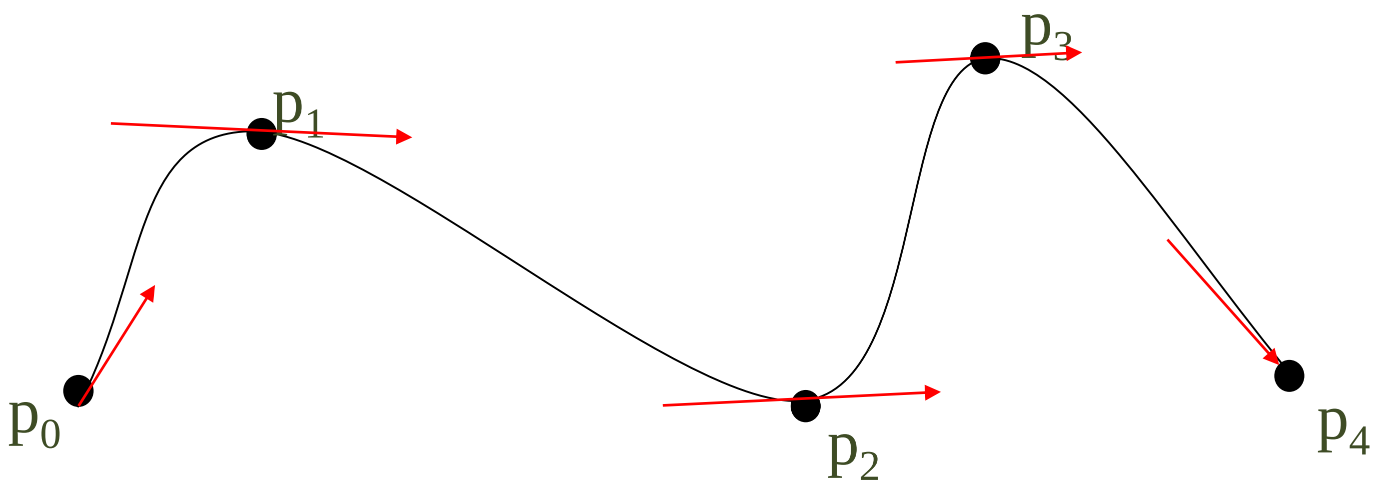
$$= M_{hermite}$$

$$\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The columns of the Basis Matrix form Basis Functions such that:

$p(t)= (2t^3 -3t^2 +1)p_0 + (-2t^2 +3t^2)p_1 + (t^3 -2t^2 +t)p'_0 + (t^3 -t^2)p'_{1.}$

Interpolation: Catmull-Romm Splines



Pick tangents based on a factor k (1/2 for eg.) of the vector between neighbor points.
$p'_i = k*(p_{i+1} - p_{i-1})$.

For the end-points there is only one neighbor:

$p'_0 = k*(p_1 - p_0)$.
$p'_n = k*(p_n – p_{n-1})$.

Physical Simulation

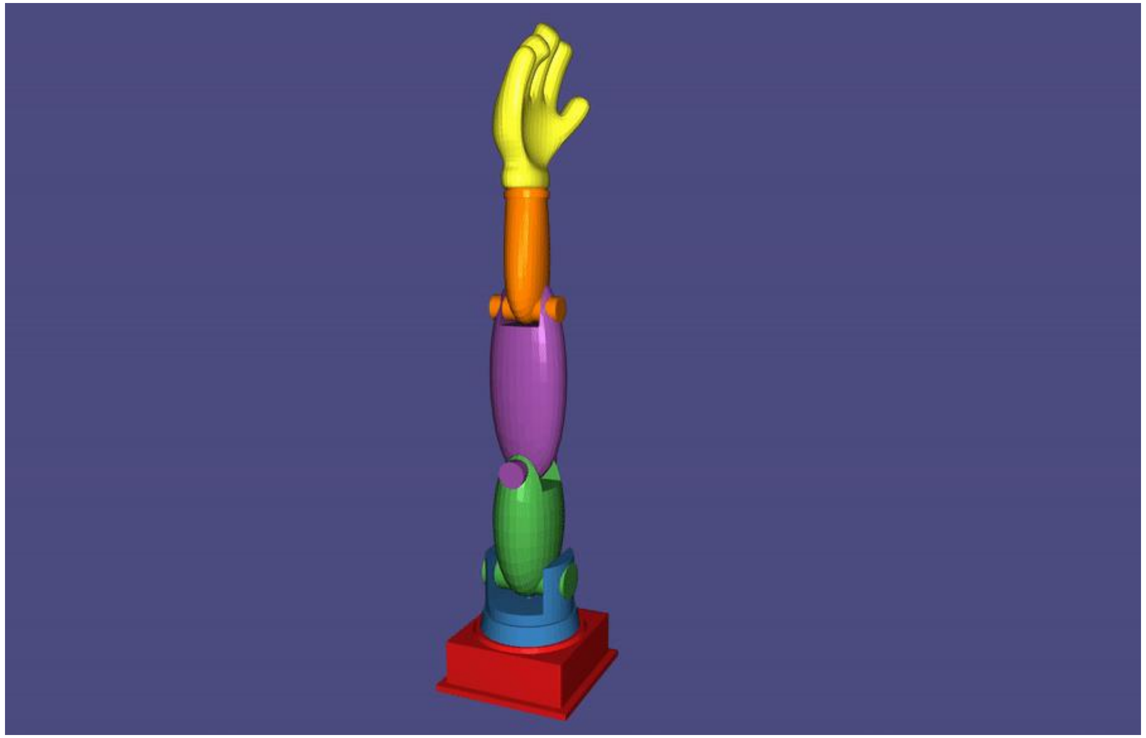
Particles

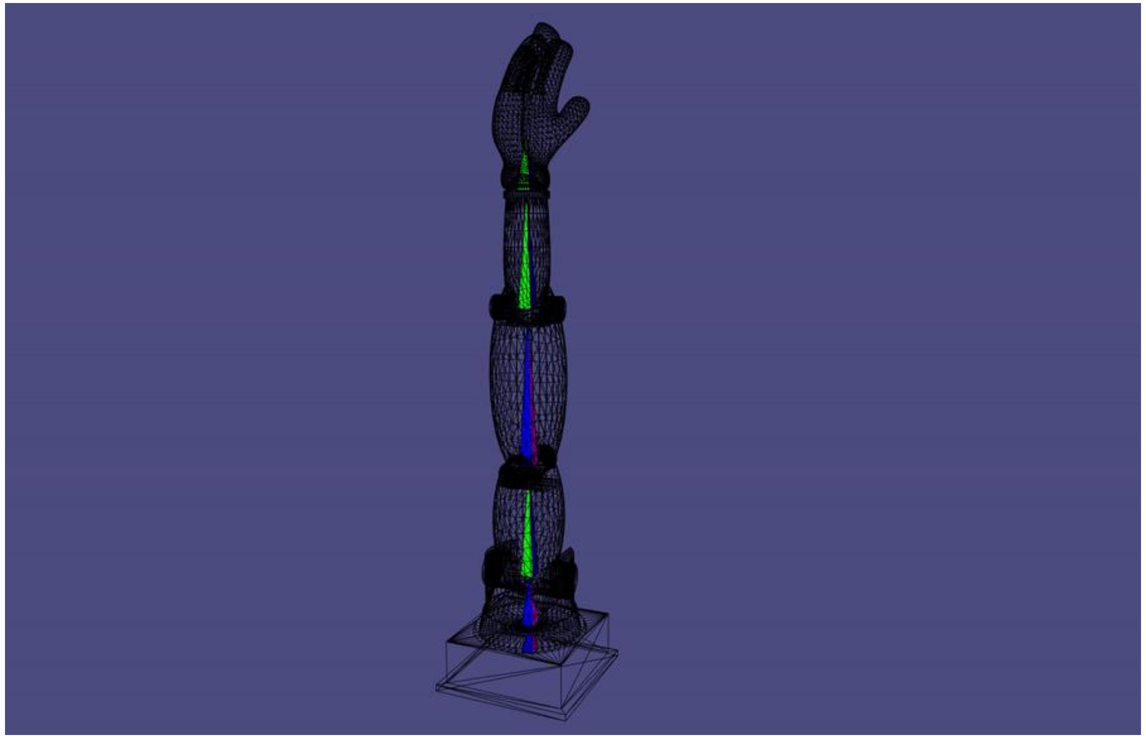
Position x
Velocity $v = dx/dt$
Acceleration $a = dv/dt = d^2x/dt^2$

Forces

Gravity $f=mg$
Spring-damper $f=-kx-cv$
…
Simulation: x,v,a used to compute forces yeilding total force F.
F=ma used to update a, a used to update v, to update x…

Skeletons

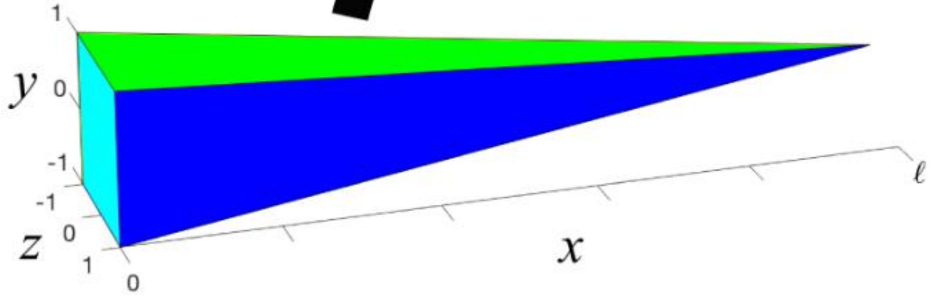


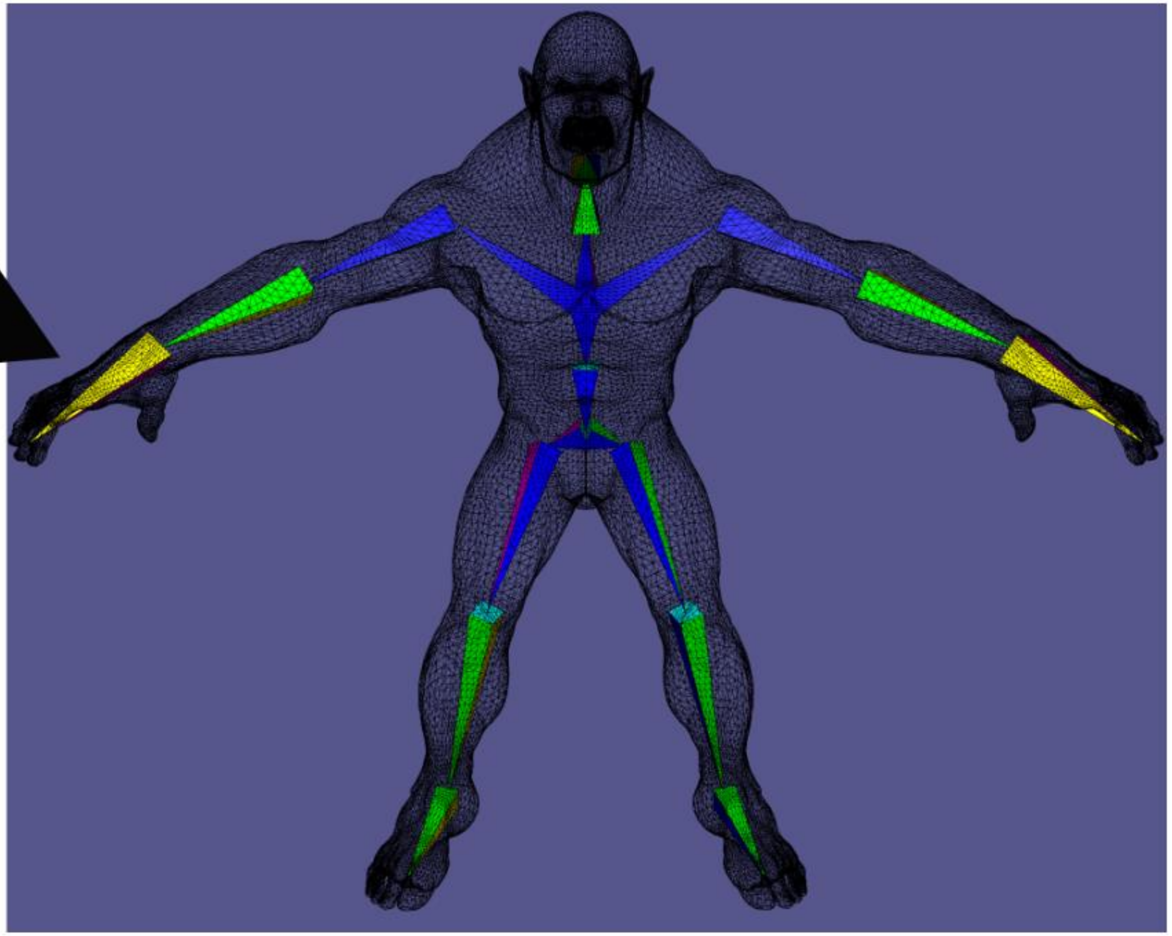


Skeletons: Rest Bone

$$\widehat{\mathbf{T}} = (\widehat{\mathbf{R}} \quad \hat{\mathbf{t}}) \in \mathbb{R}^{3\times 4}$$

Bone of length ℓ :

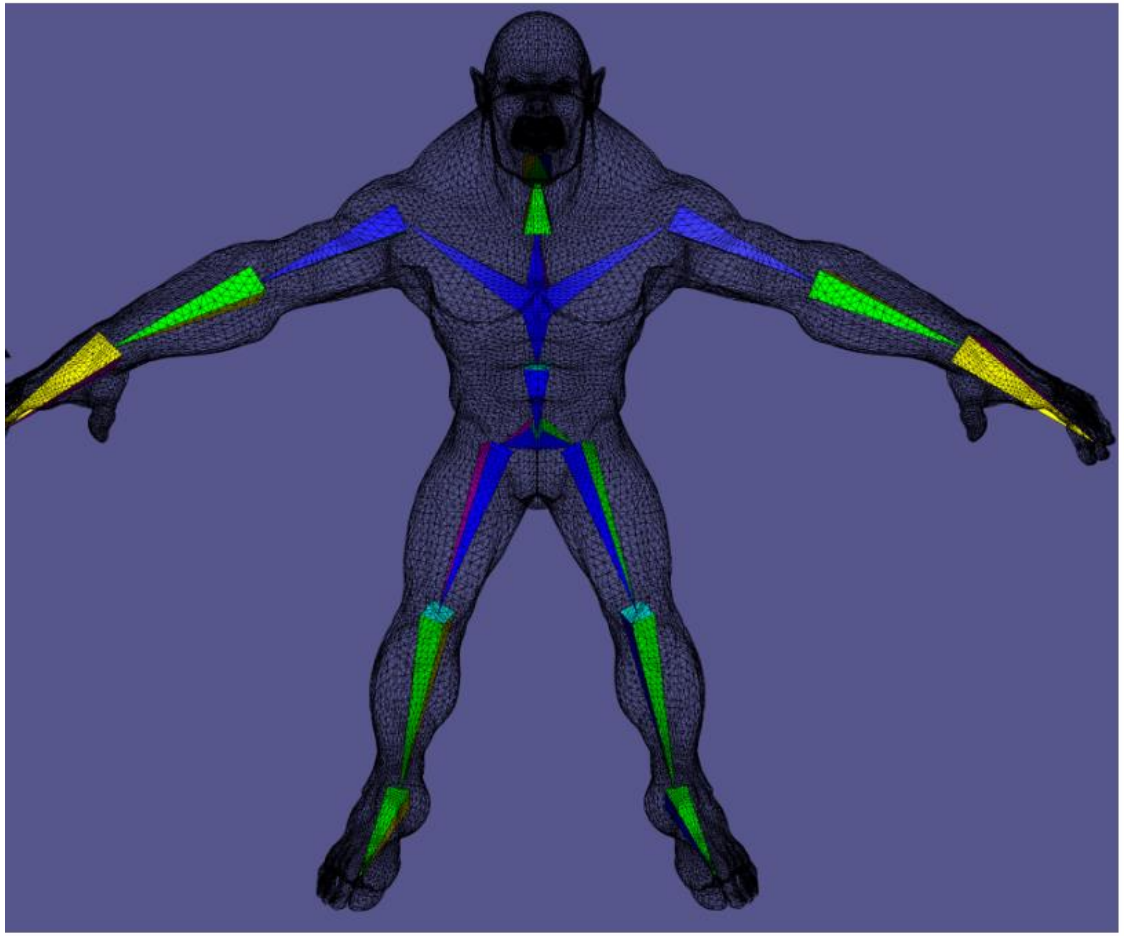




Skeletons: Rest Bone

rest position of *tail*= *tip* of parent bone *p*.

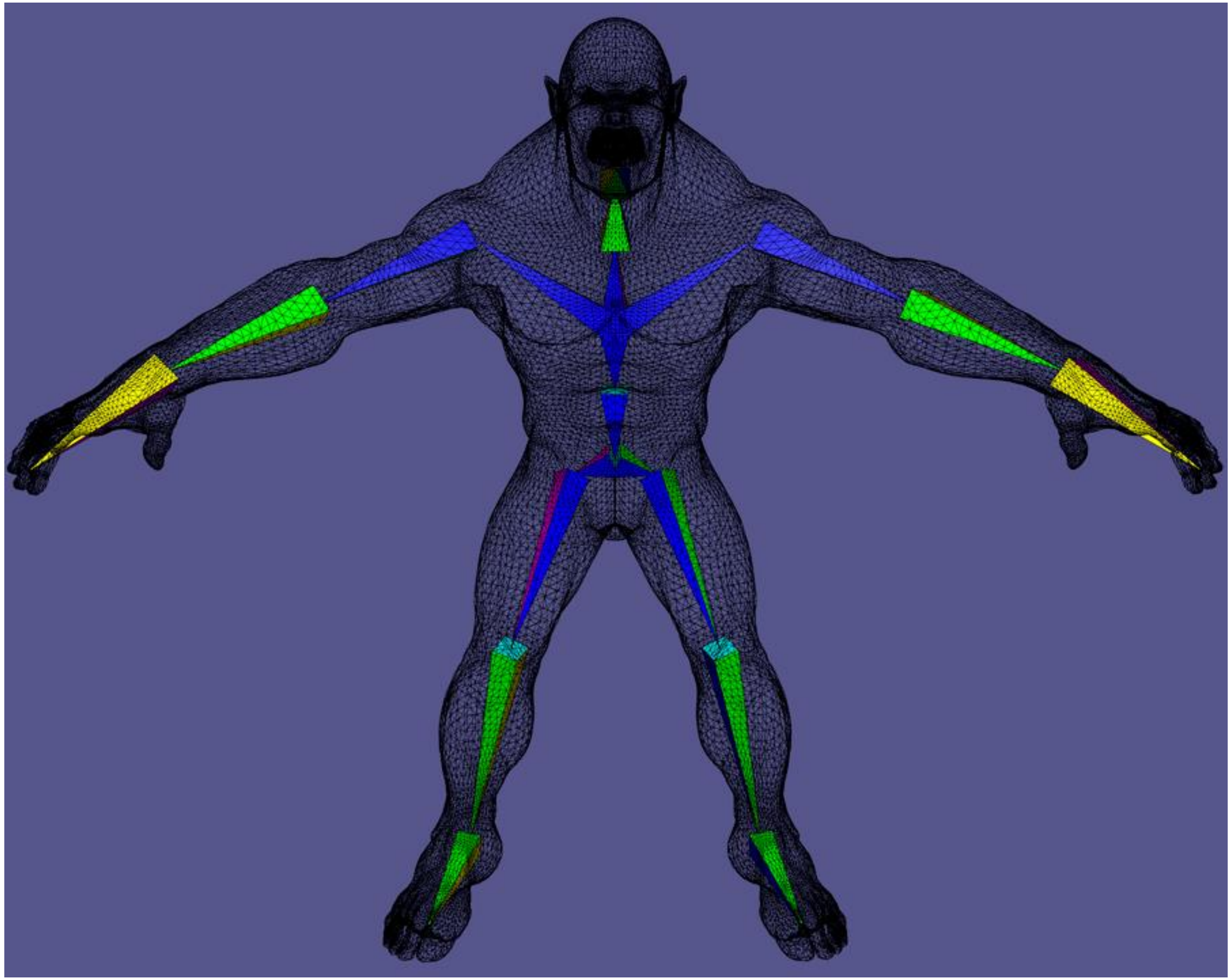
$$\hat{t} = \hat{T}_p\,[l,0,0]^T$$



Skeletons: Pose Bone

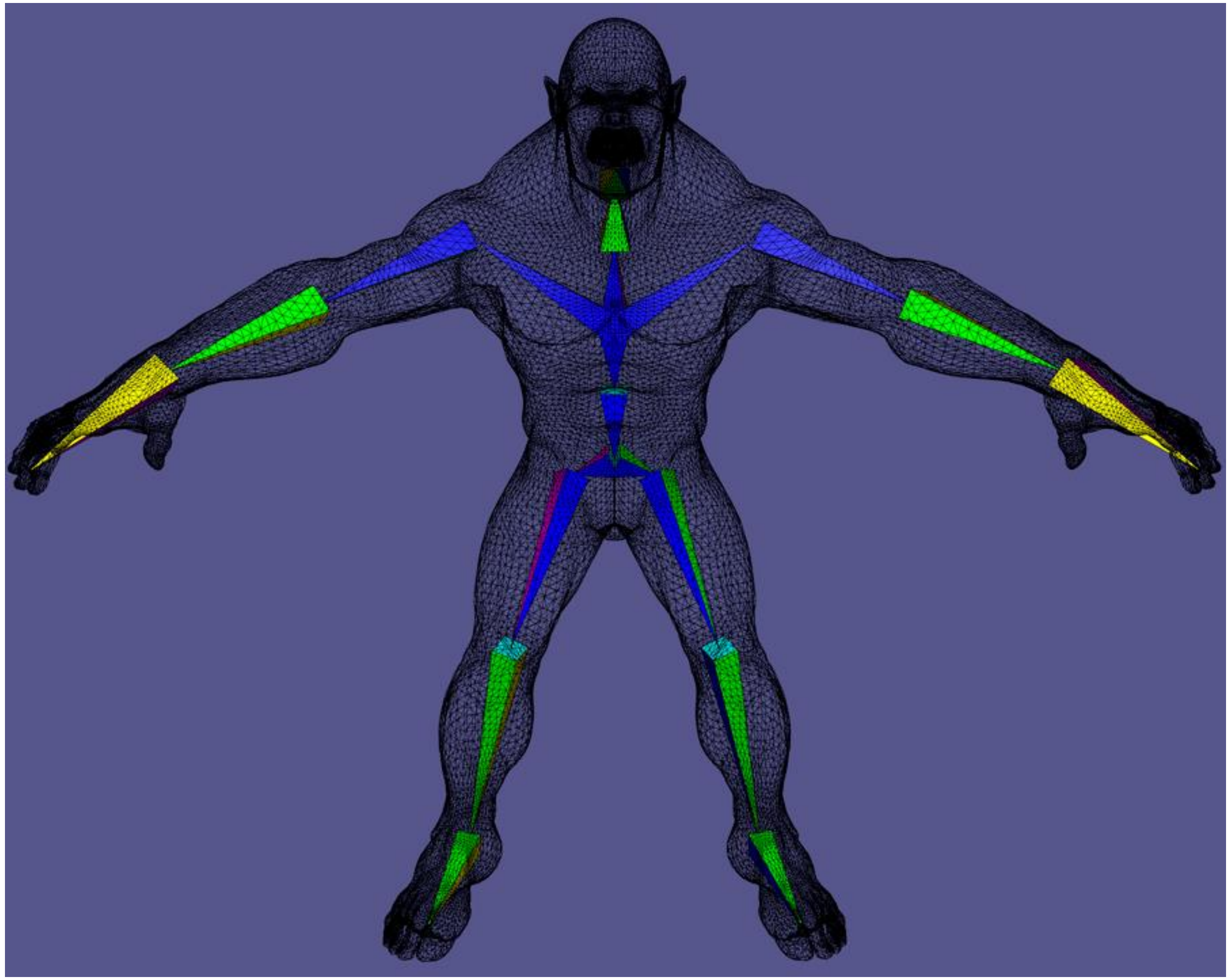
Move points to x from rest $\hat{x}$ based on bone rotation transform T:

$$x = T\hat{x}$$



Skeletons: Forward Kinematics

Let the relative rotation at a bone joint *i* be R_i:



$$\mathbf{T}_i = \mathbf{T}_{p_i} \begin{pmatrix} \widehat{\mathbf{T}}_i \\ 0\,0\,0\,1 \end{pmatrix} \begin{pmatrix} & 0 \\ \overline{\mathbf{R}}_i & 0 \\ & 0 \\ 0\,0\,0 & 1 \end{pmatrix} \begin{pmatrix} \widehat{\mathbf{T}}_i \\ 0\,0\,0\,1 \end{pmatrix}^{-1}$$

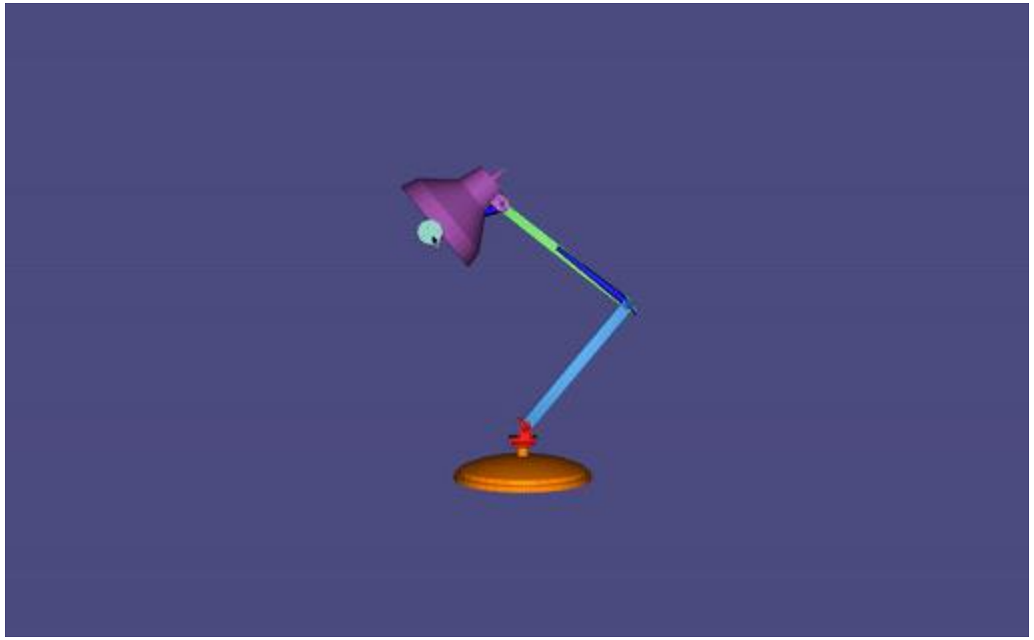
$$\mathbf{T}_i = \mathbf{T}_{p_i} \widehat{\mathbf{T}}_i \begin{pmatrix} & 0 \\ \mathbf{R}_x(\theta_{i3}) & 0 \\ & 0 \\ 0\,0\,0 & 1 \end{pmatrix} \begin{pmatrix} & 0 \\ \mathbf{R}_z(\theta_{i2}) & 0 \\ & 0 \\ 0\,0\,0 & 1 \end{pmatrix} \begin{pmatrix} & 0 \\ \mathbf{R}_x(\theta_{i1}) & 0 \\ & 0 \\ 0\,0\,0 & 1 \end{pmatrix} \widehat{\mathbf{T}}_i^{-1}$$

utilize canonical twist-bend-twist rotations (three Euler angles, $\theta_1, \theta_2, \theta_3$).

Skeletons: Inverse Kinematics

What is the pose (set of joint angles *a*) that lets us reach a given point (end-effector position).

$$\mathbf{a} = \begin{pmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \\ \theta_{21} \\ \theta_{22} \\ \theta_{23} \\ \vdots \\ \theta_{m1} \\ \theta_{m2} \\ \theta_{m3} \end{pmatrix}$$



What does it mean to reach (get as close as possible) to a point?

Skeletons: Inverse Kinematics

Closeness energy can be measured the squared distance between the pose tip x_b of some bone b and a desired goal location $q \in R^3$.

$$E(\mathbf{x}_b) = \|\mathbf{x}_b - \mathbf{q}\|^2.$$

Given pose vector a, the bone tip x_b is:

$$\mathbf{x}_b(\mathbf{a}) = \mathbf{T}_b \widehat{\mathbf{d}}_b$$

Now given any number of end-effectors $b_1, .. b_k$:

$$\min_{\mathbf{a}} \underbrace{\sum_{i=1}^{k} \|\mathbf{x}_{b_i}(\mathbf{a}) - \widehat{\mathbf{x}}_{b_i}\|^2}_{E(\mathbf{x}_b(\mathbf{a}))}$$

And we impose some joint angle limits: $\min_{\mathbf{a}^{\min} \leq \mathbf{a} \leq \mathbf{a}^{\max}} E(\mathbf{x}_b(\mathbf{a}))$

Minimizing this energy is a non-linear least squares problem.

Skeletons: Inverse Kinematics Minimization

Projected Gradient Descent:

Start with an initial pose *a*, and move in direction of decrease in *E*, project the pose to stay within limits and iterate towards solution.

$$\mathbf{a} \leftarrow \mathbf{a} - \sigma\left(\frac{dE(\mathbf{x}(\mathbf{a}))}{d\mathbf{a}}\right)^T$$

$$\mathbf{a} \leftarrow \mathbf{a} - \sigma\left(\frac{d\mathbf{x}(\mathbf{a})}{d\mathbf{a}}\right)^T\left(\frac{dE(\mathbf{x})}{d\mathbf{x}}\right)$$ chain rule

$\frac{dE}{d\mathbf{a}} \in \mathbb{R}^{|\mathbf{a}|}$, $\frac{dE}{d\mathbf{x}} \in \mathbb{R}^{|\mathbf{x}|}$, and $\frac{d\mathbf{x}}{d\mathbf{a}} \in \mathbb{R}^{|\mathbf{x}|\times|\mathbf{a}|}$

$$\mathbf{J} = \frac{d\mathbf{x}}{d\mathbf{a}}.$$ also known as Jacobian measures the change in *x* for changes in joint angles *a*,

J can be computed using Finite Differences: $\mathbf{J}_{i,j} \approx \frac{\mathbf{x}_i(\mathbf{a} + h\delta_j)}{h}.$ $h = 10^{-7}$

$\left(\frac{dE(\mathbf{x})}{d\mathbf{x}}\right)$ is gradient of $\sum_{i=1}^{k} \|\mathbf{x}_{b_i}(\mathbf{a}) - \widehat{\mathbf{x}}_{b_i}\|^2$

Project to within limits: $\mathbf{a}_i \leftarrow \max[\mathbf{a}_i^{\min}, \min[\mathbf{a}_i^{\max}, \mathbf{a}_i]].$

Find a good step that lowers energy: $E(\mathrm{proj}(\mathbf{a} + \sigma\,\Delta\,\mathbf{a})) < E(\mathbf{a}).$

Linear Blend Skinning

Rigid Skinning: objects/points are controlled by a single bone.

Smooth Skinning: weights for multiple bones influence a point *v*.

Where do the weights *w* come from?

$$\mathbf{v}_i = \sum_{j=1}^{m} w_{i,j} \mathbf{T}_j \begin{pmatrix} \widehat{\mathbf{v}}_i \\ 1 \end{pmatrix}.$$