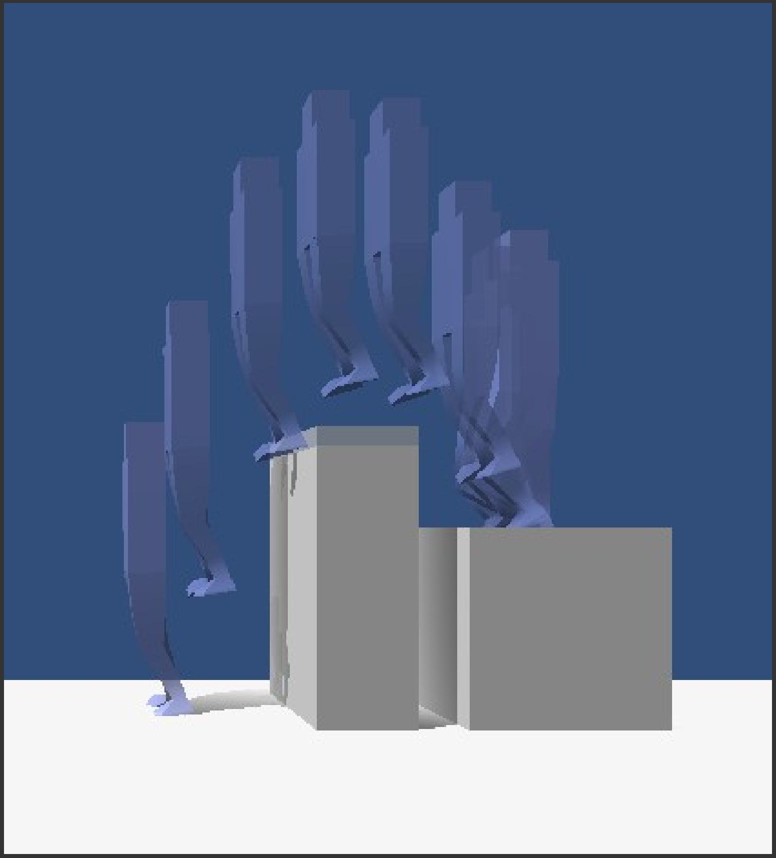
# Controller for Jumping Animations to Achieve Target Positions



Ian Ooi

# Agenda

- **Introduction**
- Skeleton and Muscle Setup
- Torque-based Simulation
- Energy-based Simulation
- Limitations
- Visualization
- Future Work

# Overview

- Animation is hard
- Work and time intensive process
- Use physics simulation to reduce work in part of the process

# Simulating for Animation

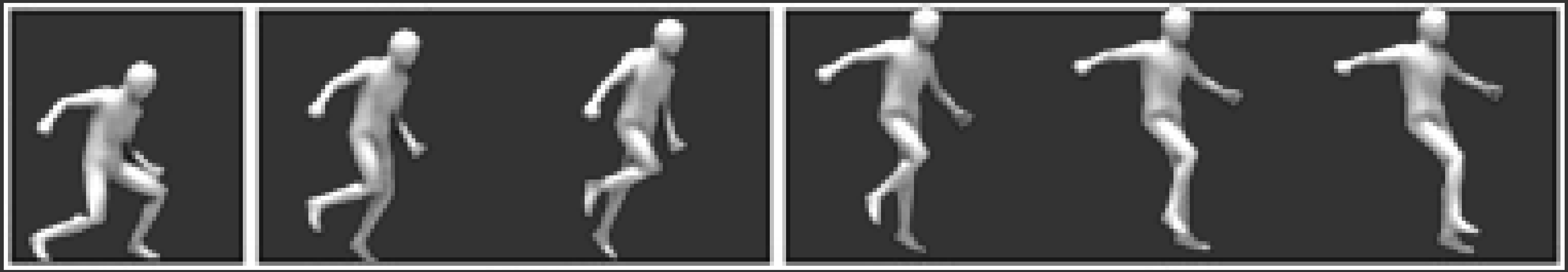
- Simulate muscles to produce a motion/pose for a character's body
- Muscle → Force →    ?    → Position/Velocity
- First need a representation of the character and of a jump

# Jumping

- Acceleration of *center of mass* (CoM) against gravity
- Five stages
- Controller handles stages through in-air

# Jump Stages

- Path Estimation (before jump)
- Windup (a)
- Thrust (b)
- In-Air (c)
- Landing (not pictured)



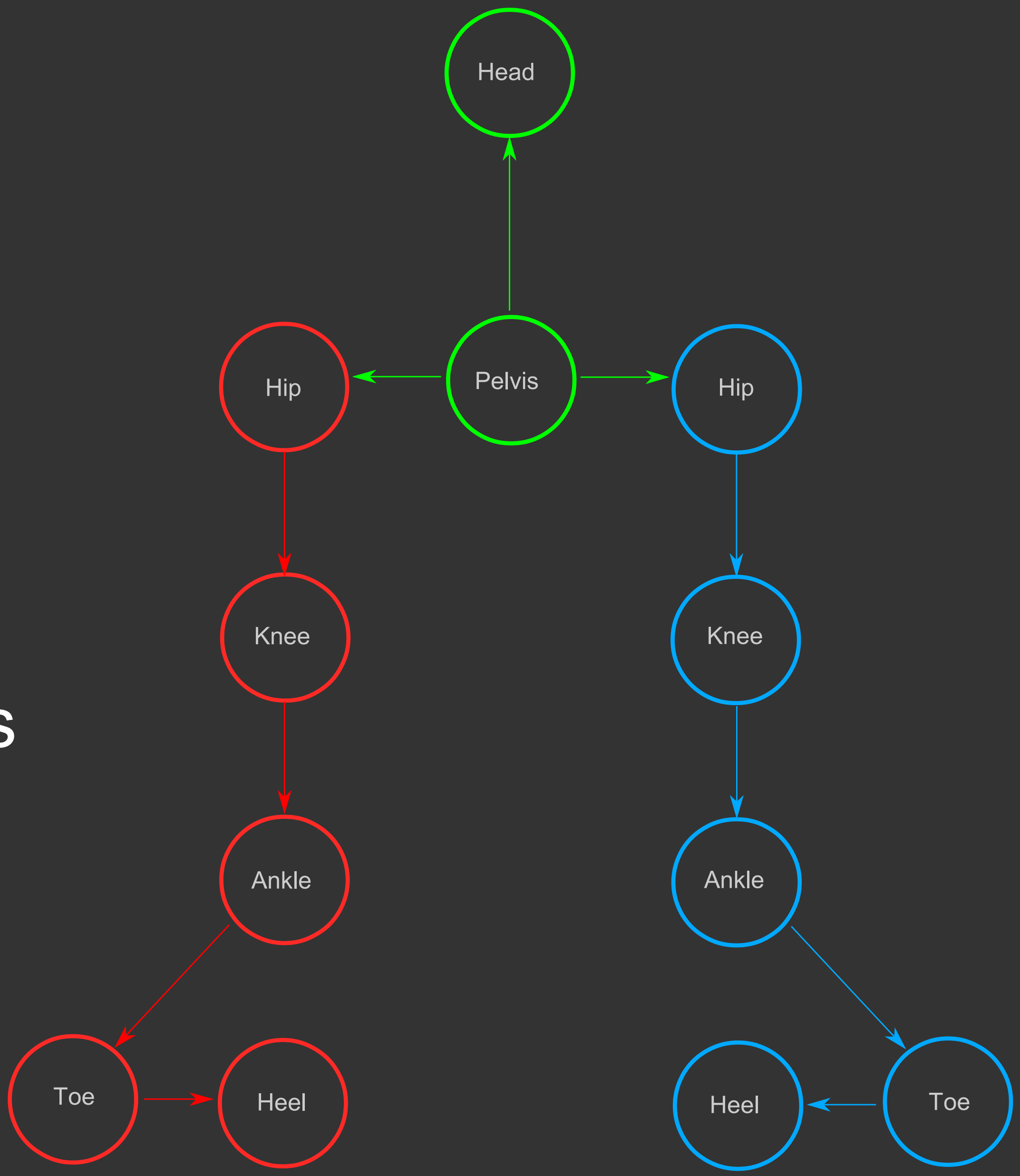
(a) (b) (c)

# Agenda

- Introduction
- **Skeleton and Muscle Setup**
- Torque-based Simulation
- Energy-based Simulation
- Limitations
- Visualization
- Future Work

# Skeleton Setup

- Use rig (skeleton) with mesh created by artist
- Joint has position, rotation, scale
  - Specific to this simulation, also stores the mass of the limb
- Tree of joints



# Muscle Force

- Muscle == Spring (approximately)
- Linear springs (Hooke's Law):

$$F = -k\,s$$

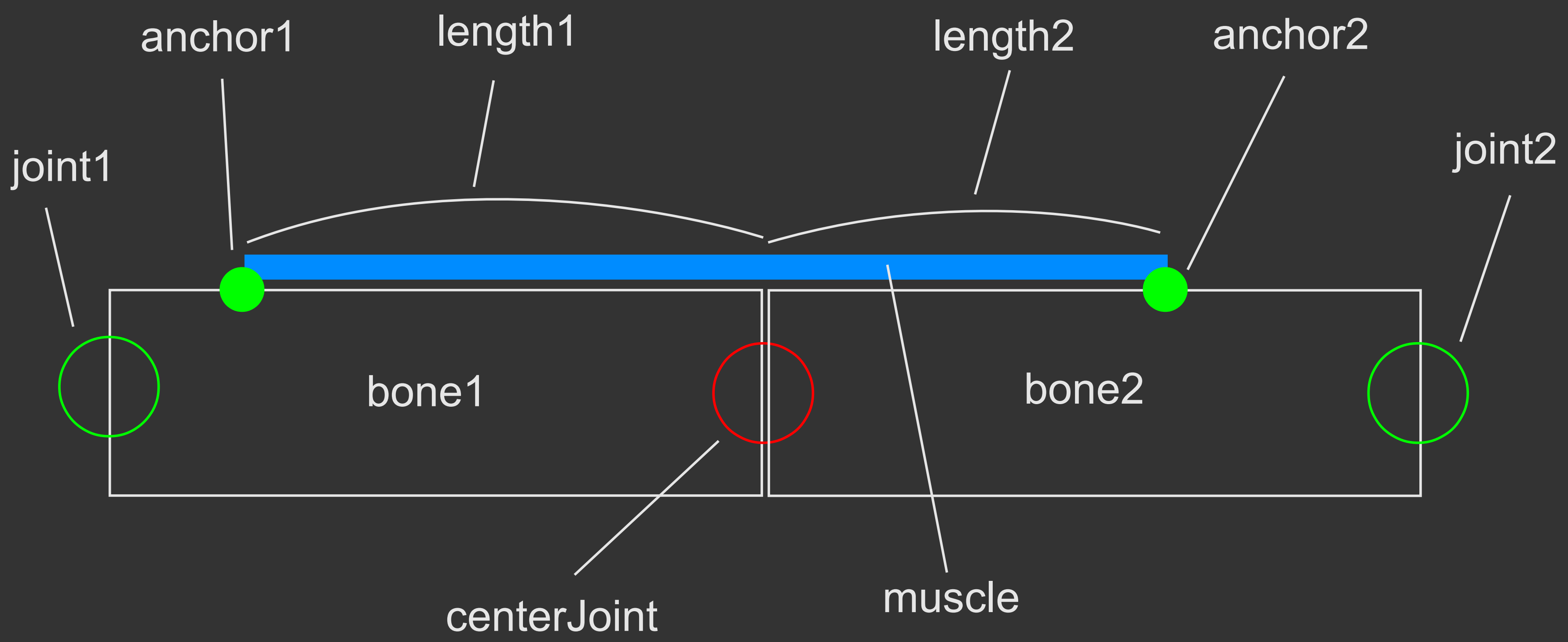
Force (F)

Spring Constant (k)

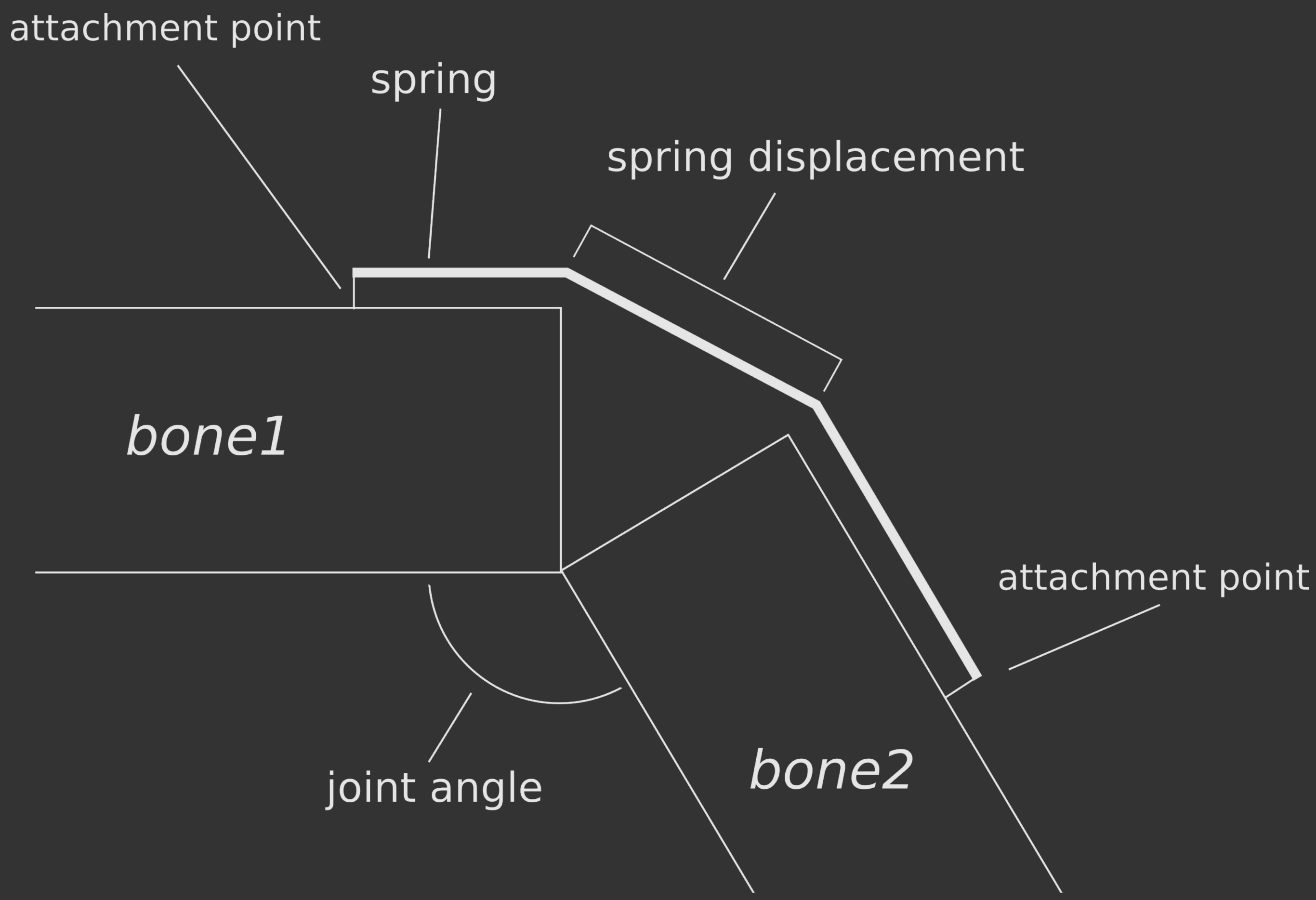
Spring Displacement (s)

# Muscle Setup

- 3 Joints: crosses center, anchored near other two

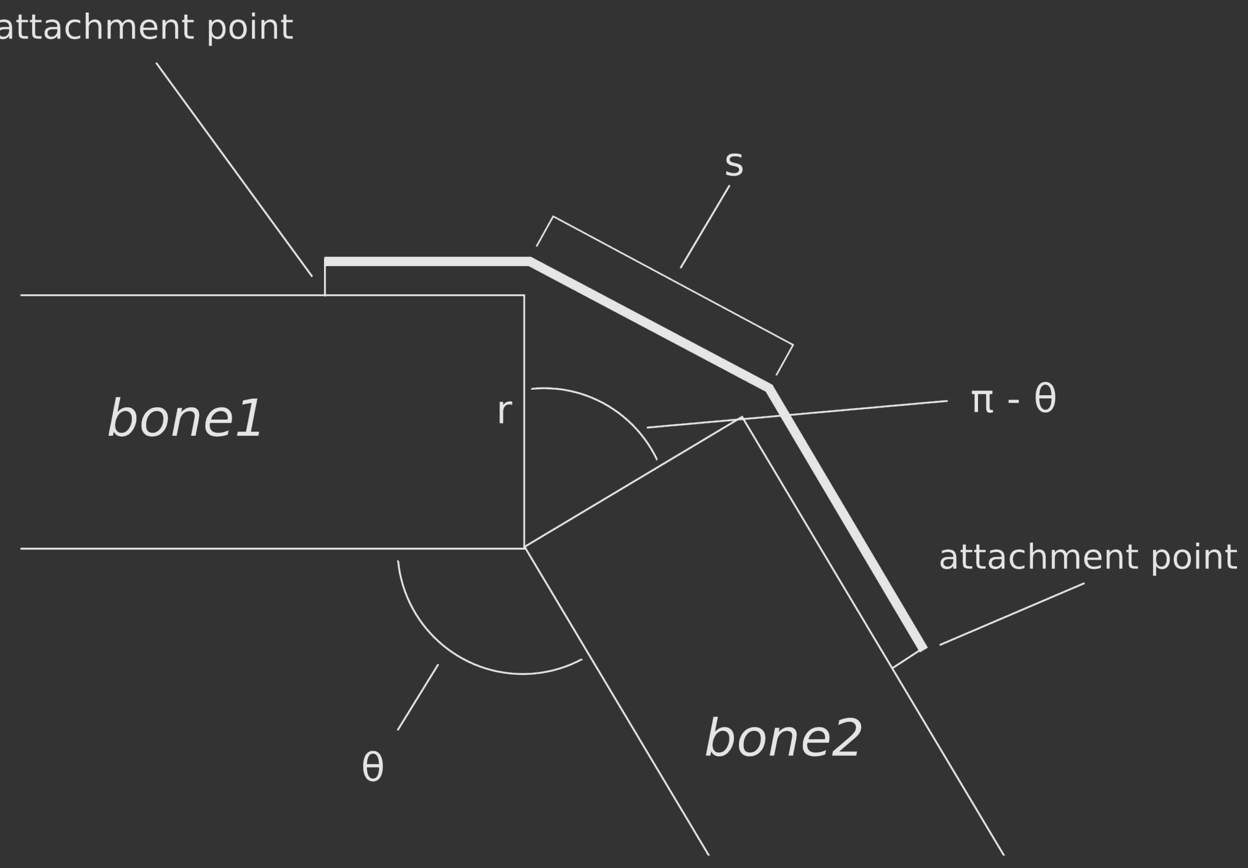


# Muscle Setup



# Calculating Spring Displacement

$$s = \left( \frac{r \sin(\pi - \theta)}{\sin\left(\frac{\theta}{2}\right)} \right)$$



# Selecting k

- Use displacement (s) calculated previously, calculate force
- Need k value for $F = -ks$
- Found empirically to reach 1-1.5m long jump and 0.5-0.75m vertical jump

# Agenda

- Introduction
- Skeleton and Muscle Setup
- **Torque-based Simulation**
- Energy-based Simulation
- Limitations
- Visualization
- Future Work

# Muscle Force and the Skeleton

- Can't just sum the forces of each muscle
- Muscles produce a torque on the joint

$$\tau = rF$$

# Torque and Motion

- Torque is related to angular momentum

$$\tau = \frac{d\vec{L}}{dt} \rightarrow L = \tau\ dt$$

- Angular momentum → linear momentum (direction is tangent to circle)

$$\|\vec{p}\ \| = \frac{L}{r}$$

# Torque and Motion

- Linear momentum relates to acceleration
- Find the direction for *p* by finding the tangent to the traced by the arm at timestep, constantly changing

$$\vec{a} = \frac{1}{m}\left(\frac{d\,\vec{p}}{dt}\right)$$

- Perform for all muscles to find acceleration for body

# Torque and Body Pose

- Have an estimate of what the skeleton can do at a particular body pose
- How to pick a pose?

# Path Estimation

- Estimate desired initial velocity and acceleration
- Classical mechanics
- Forward kinematic “path” through the air

# Path Estimation – Calculating Velocity

- Estimate initial velocity to travel over distance given constant acceleration due to gravity

$$\vec{x} = \vec{x}_0 + \vec{v}_0 t_{air} + \frac{1}{2}\vec{a}\, t_{air}^2$$

$$\downarrow$$

$$\vec{v}_0 = \frac{\vec{x} - \vec{x}_0}{t_{air}} - \frac{\vec{g}\, t}{2}$$

# Path Estimation – Calculating Acceleration

- Assuming we start at rest, acceleration is:

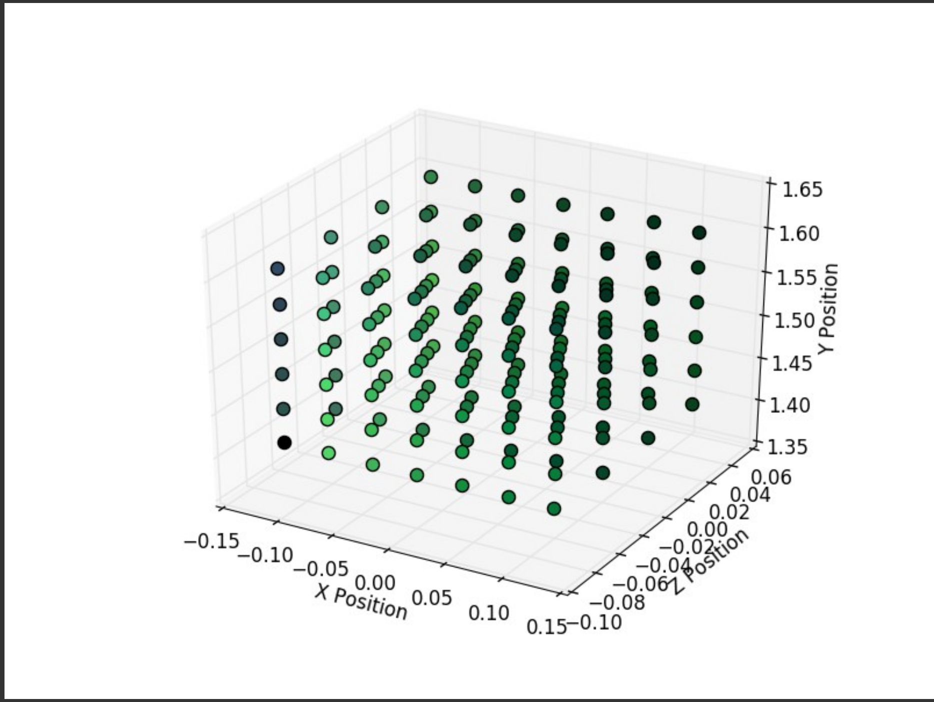
$$\vec{a} = \frac{\vec{v}_0}{t_{windup}}$$

- To accommodate initial velocity:

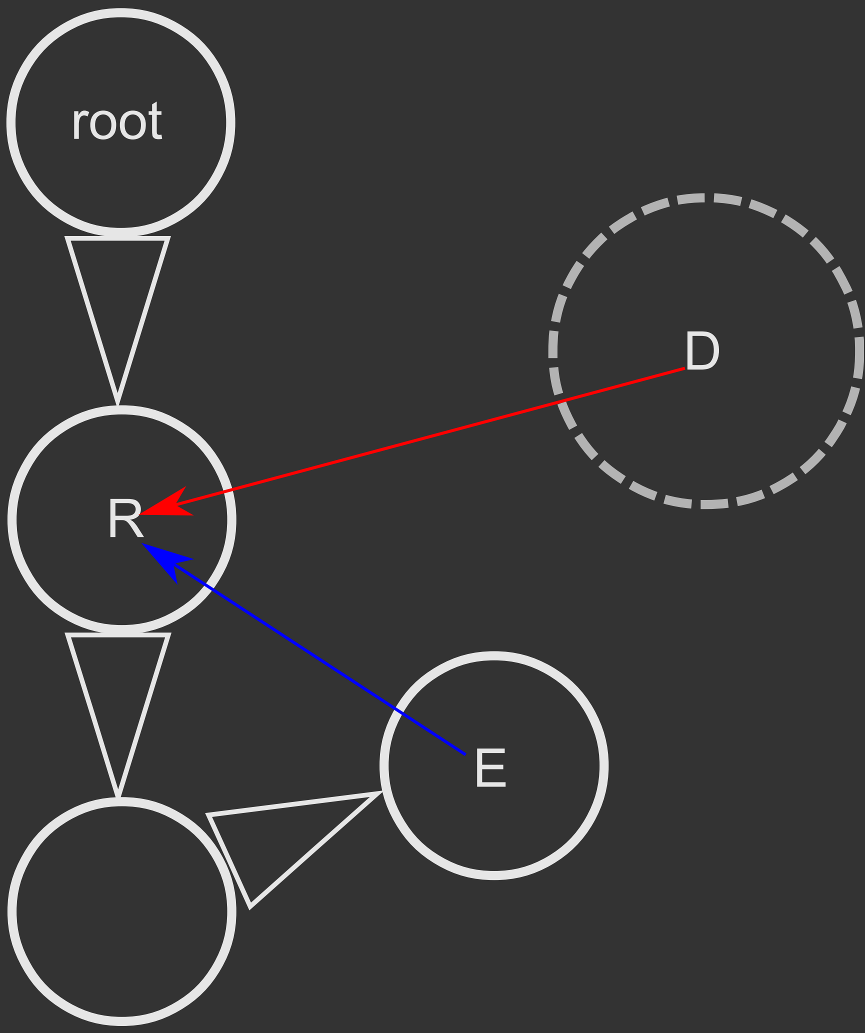
$$\vec{a} = \frac{\vec{v}_0 - \vec{v}_{initial}}{t_{windup}}$$

# Windup - Finding the Pose

- Sample poses of the skeleton
- Given varying pelvis position, solve IK problem to find pose of legs and calculate output acceleration
- How to solve IK?



# Windup – IK Solving



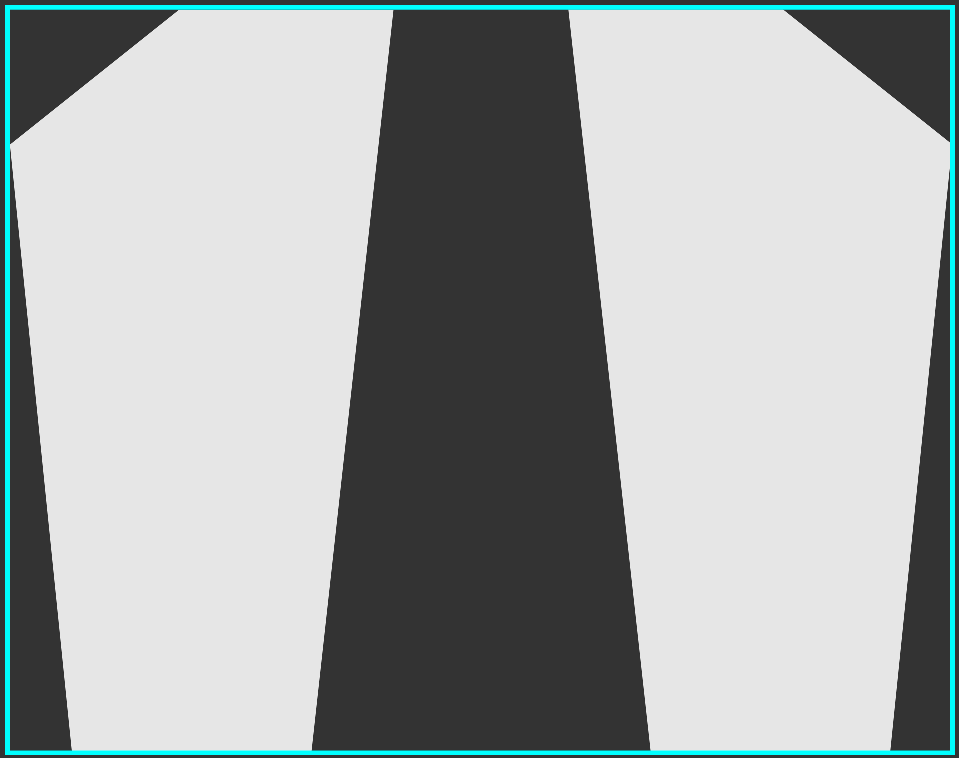

- “Kine” by Jeff Landers
- Cyclic Coordinate Descent
- Simple, greedy method
- Use dot product to find angle between vectors, rotate by that angle

# Windup – Restricting the Samples

- Many different positions for the skeleton, most of which are “invalid” due to balance issues
- Should restrict the sample region to the area where the character is balanced

# Windup – Balance and CoM

- Character balanced when CoM is over the *supporting polygon* (rectangle around contacts with ground)
- Error calculated by projecting CoM onto plane of supporting polygon
- 2D distance from polygon centroid



# Windup – Sampling and Finding Poses

- Sample uniformly within prism in which the character is balanced (supporting polygon up to pelvis height when extended)
- Position pelvis and collect torques and acceleration
- Order by acceleration error, collect lowest, choose among those by balance.

# Windup – PD Control

- Proportional Derivative control used for positional adjustments and for upper body balance adjustments
- Feedback control based on proportion of error and change in error
- Output *u* at time *t* based on the error function *e*
- Constants $k_p$ *and* $k_d$ adjust rate of change

$$u(t) = k_p e(t) + k_d\left(e(t) - e(t-1)\right)$$

# Thrust – Angular Acceleration

- Torque gives angular acceleration through moment of inertia *I*.
- Where *r* is the length of the limb: $I = mr^2$

$$\tau = \frac{dL}{dt} = I\,\frac{dv_\theta}{dt} = I\,a_\theta$$

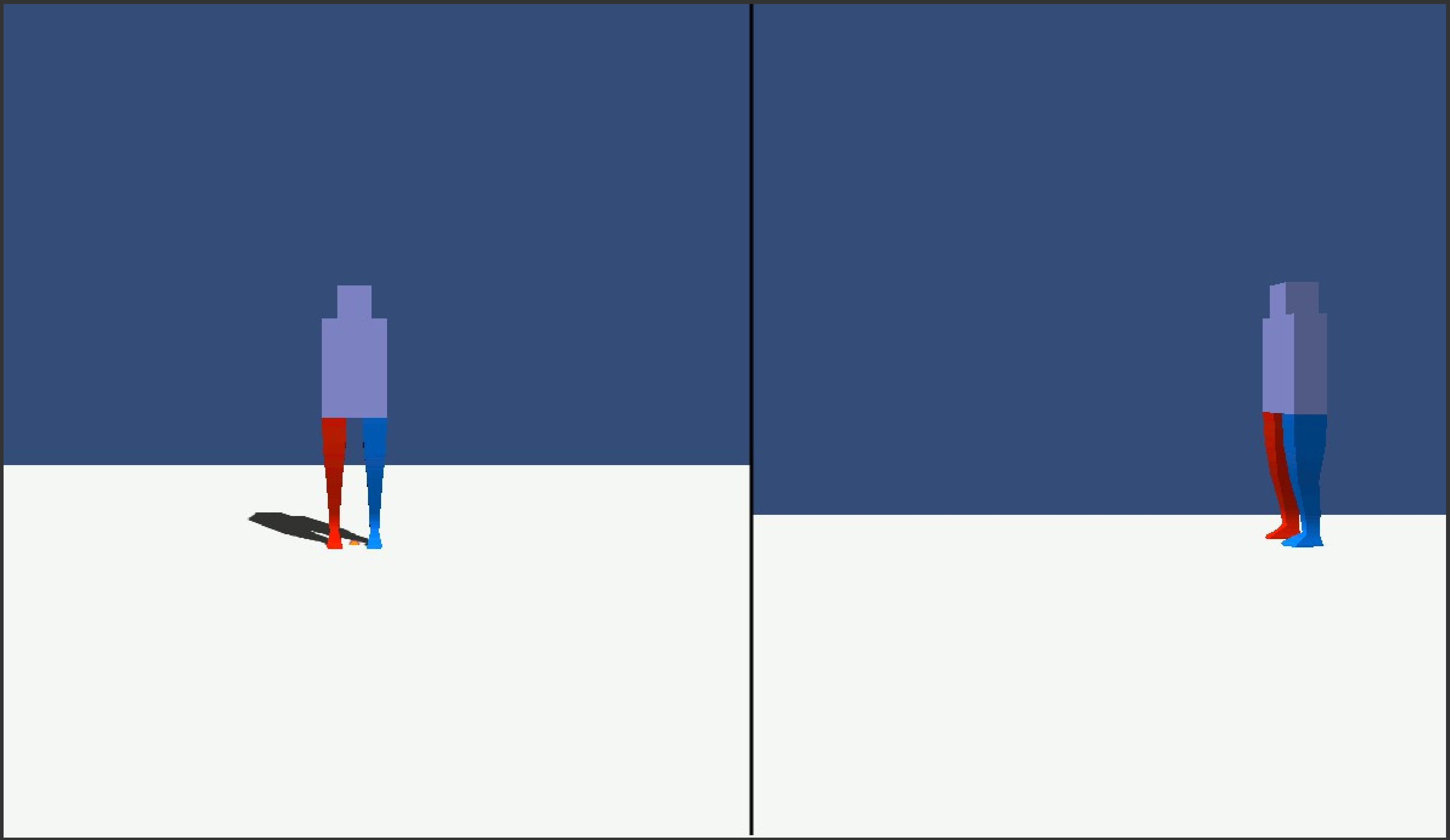
$$\downarrow$$

$$a_\theta = \frac{\tau}{mr^2}$$

# In-Air

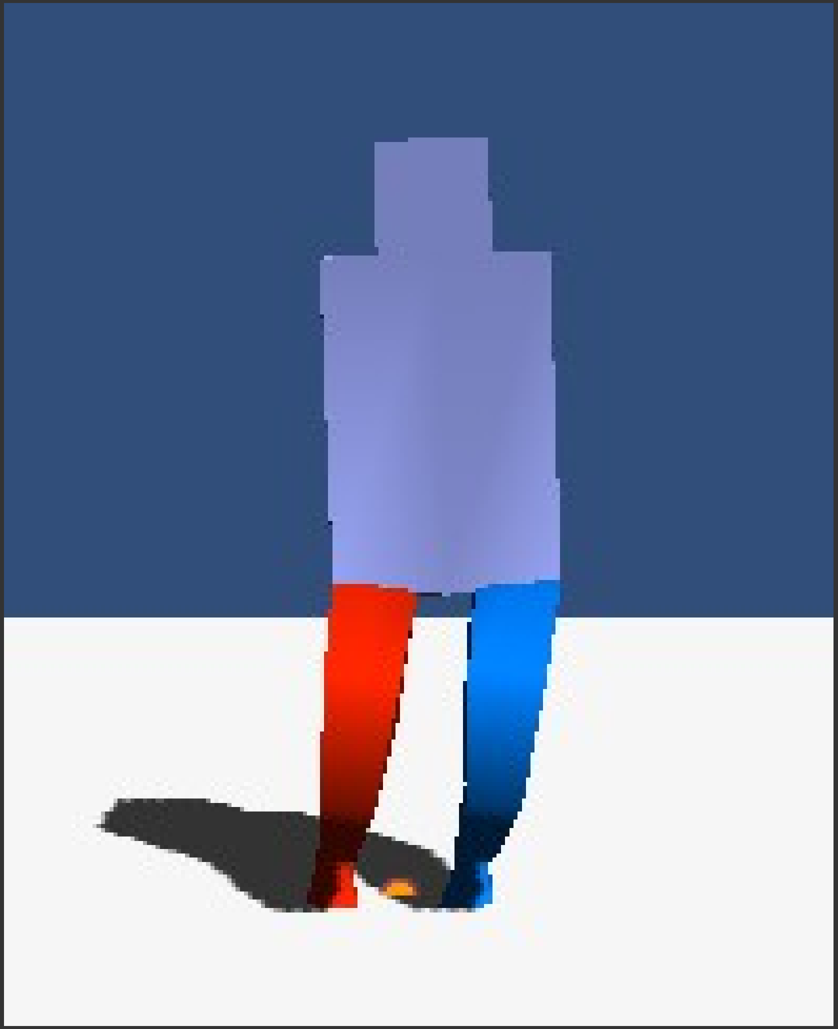
- Rigid body physics simulation
- Move the character as a whole based on the velocity and acceleration from gravity

# Torque-based Simulation Result



# Torque-based Simulation Result

- Issues, some unknown bug
- Inconsistent convergence
- Favors one side for a straight long jump
- After fixing some bugs, more consistent but still unstable



# Agenda

- Introduction
- Skeleton and Muscle Setup
- Torque-based Simulation
- **Energy-based Simulation**
- Limitations
- Visualization
- Future Work

# Energy-based Simulation

- Alternative method to torque
  - Attempts to avoid repeated work
- Consider the total elastic energy of the muscles of the character
- Modify windup and thrust stages

# Energy-based Muscle Calculation

- Elastic potential energy for a muscle
- Energy sums simply, no need to calculate resultant as with torque

$$E_{elastic} = \frac{1}{2} k s^2$$

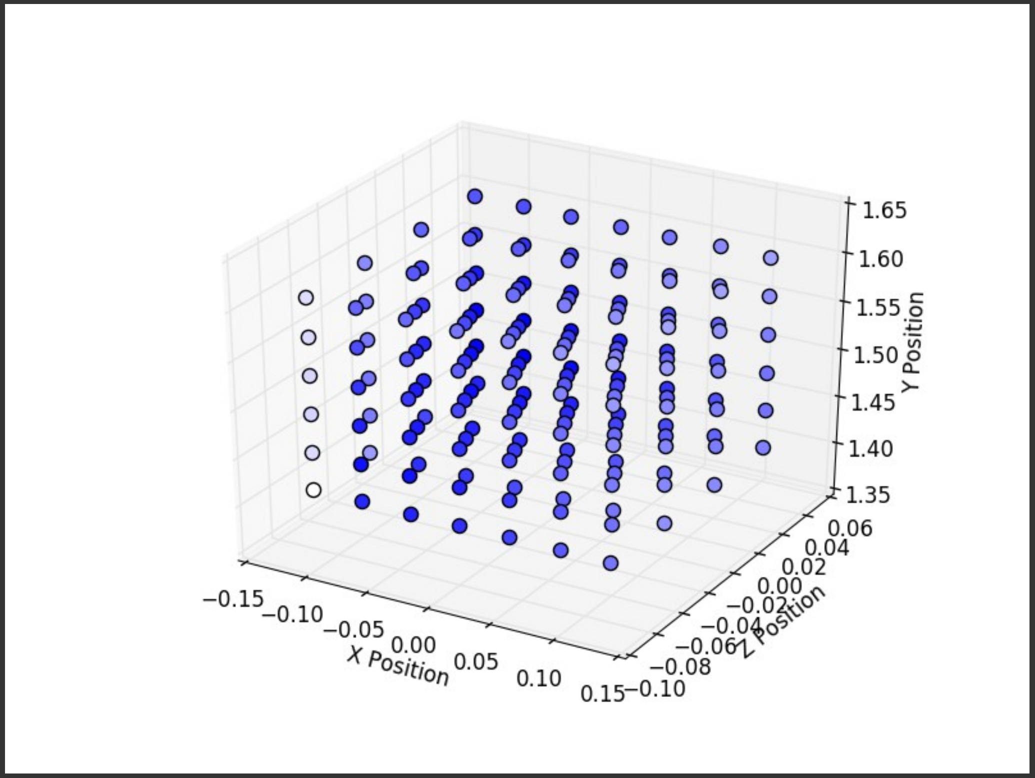
# Energy-based Path Estimate

- Instead of comparing to acceleration, compare to required kinetic energy
- Calculate velocity as before: $\vec{v} = \frac{\vec{x} - \vec{x}_0}{t_{air}} - \frac{\vec{g}\,t}{2}$
- Kinetic energy:

$$E_{kinetic} = \frac{1}{2} m v^2$$

# Energy-based windup

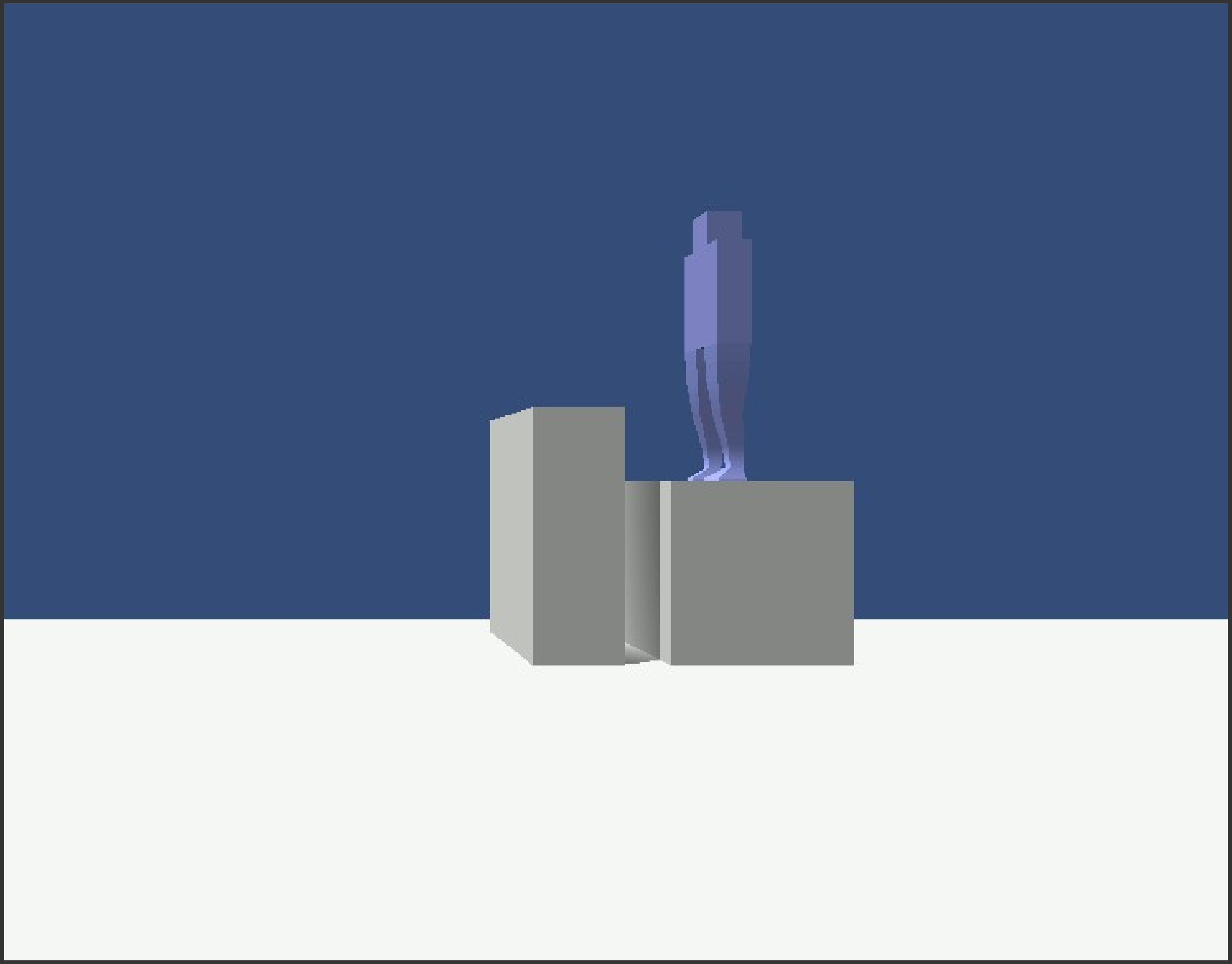
- Sampling used again with PD control
- Looking for an assignment of bends of the joints that produce the desired total energy (or greater)



# Energy-based Thrust and In-Air

- Direction of motion assigned to the desired direction
- Continually extend, moving pelvis in desired direction accelerating to reach target velocity at full extension
- Extension performed within given windup time
- In-Air proceeds as before

# Energy-based Results



# Agenda

- Introduction
- Skeleton and Muscle Setup
- Torque-based Simulation
- Energy-based Simulation
- **Limitations**
- Visualization
- Future Work

# Limitations

- Issue in torque-based simulation
- Less flexible than specifying per frame
- Requires specification of muscle and world constants
- Lack of upper body definition in current simulation

# Limitations - Lack of Patella

Image retrieved from: http://imgur.com/KSMB2tI [Accessed March 15, 2016]

# Agenda

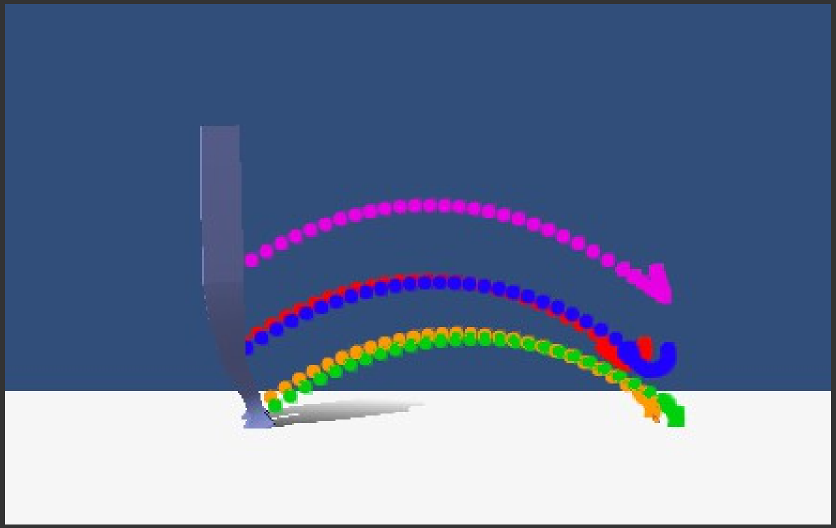
- Introduction
- Skeleton and Muscle Setup
- Torque-based Simulation
- Energy-based Simulation
- Limitations
- **Visualization**
- Future Work

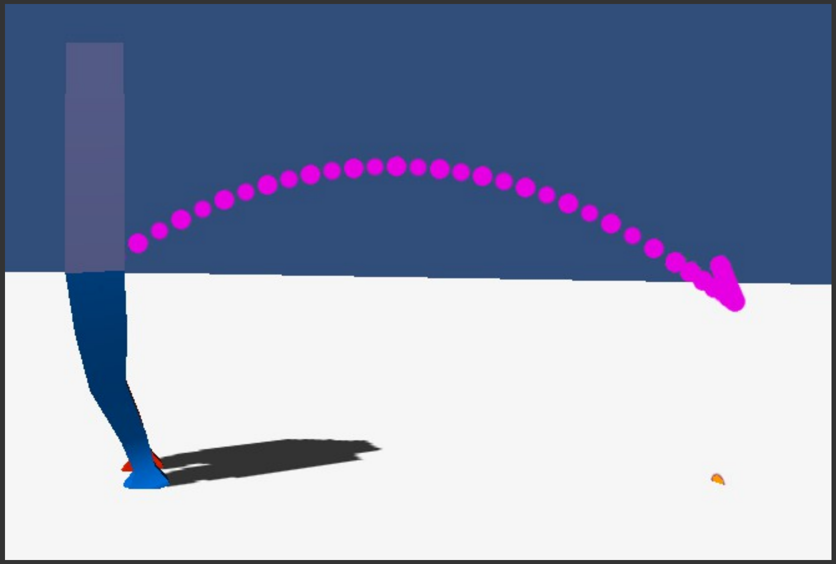
# Visualization of Motion

- Used for debugging, viewing, and presentation
- Types used:
  - Trails following parts of the character
  - Ghost images of the character
  - Composites of frames
  - Animated sequences/Sequences of frames
  - Gizmos/Handles (Unity3D features)

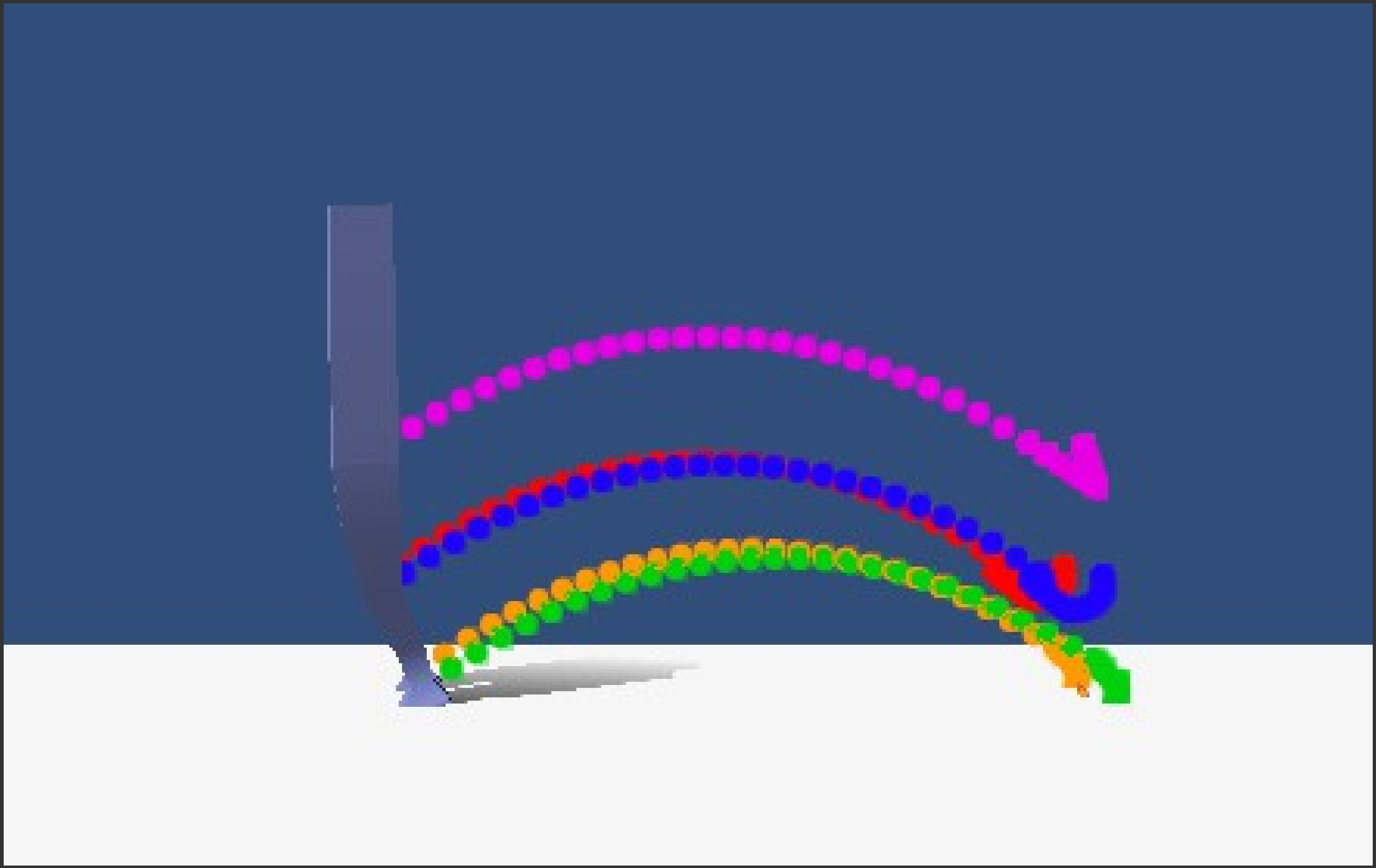
# Trail Visualization

- Marker particles placed at position of joint at certain rate
- Questionably useful for windup/thrust phases
- Good for tracing in-air path
- Shows some IK twisting issues

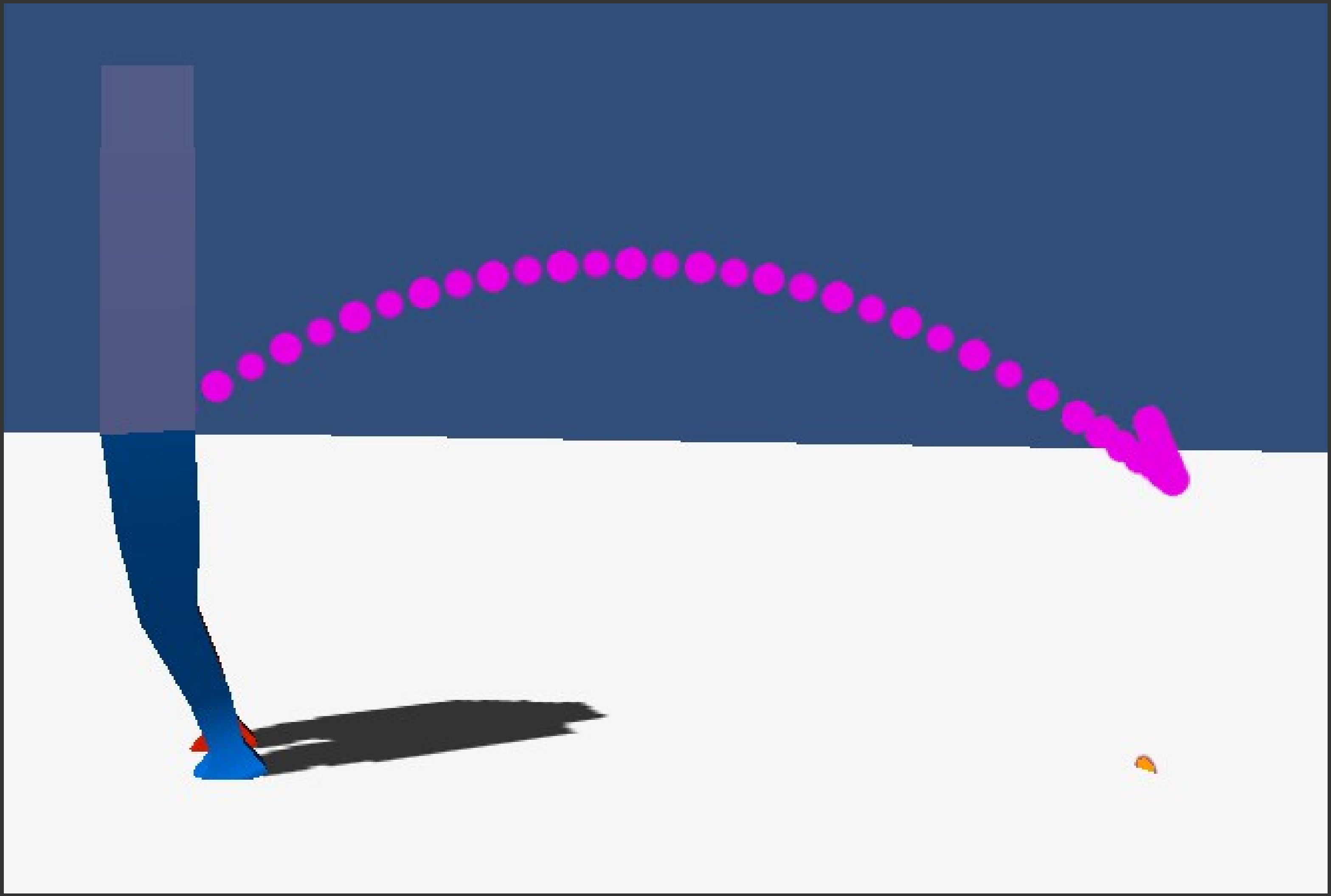




# Trail Visualization

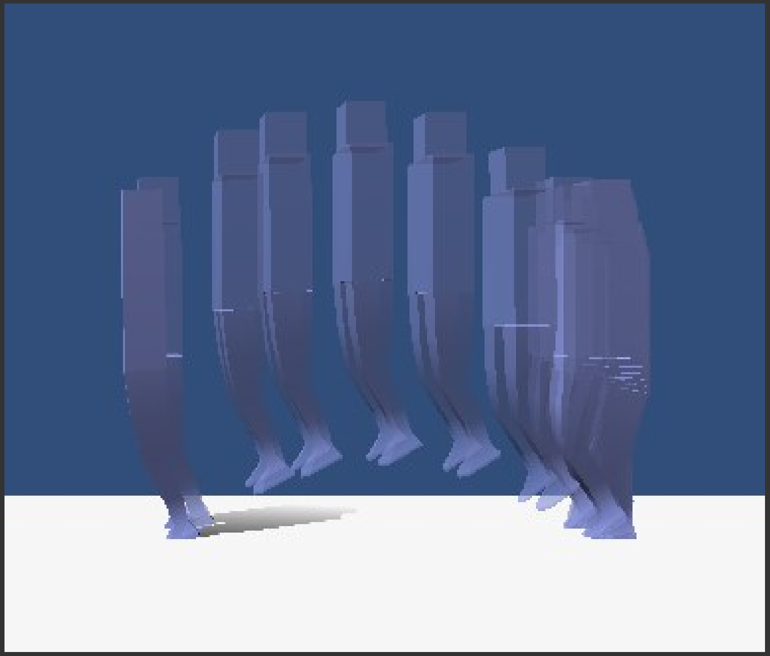


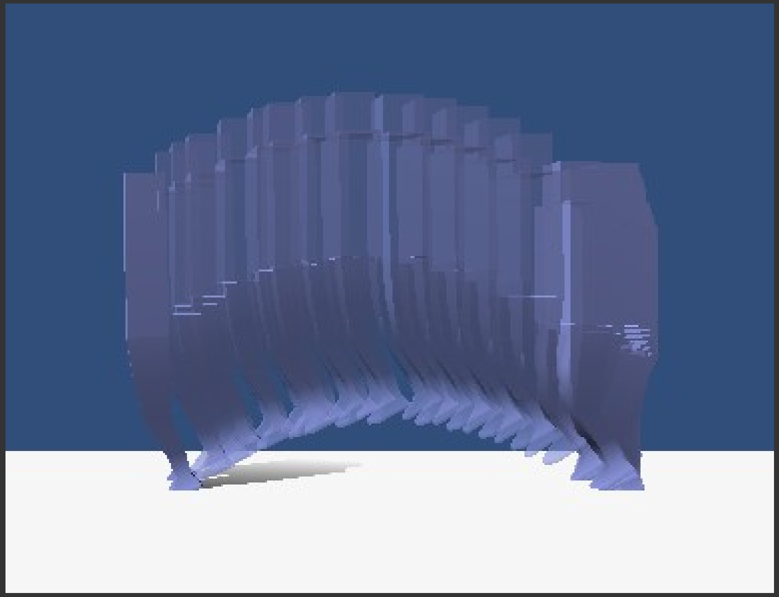
# Trail Visualization



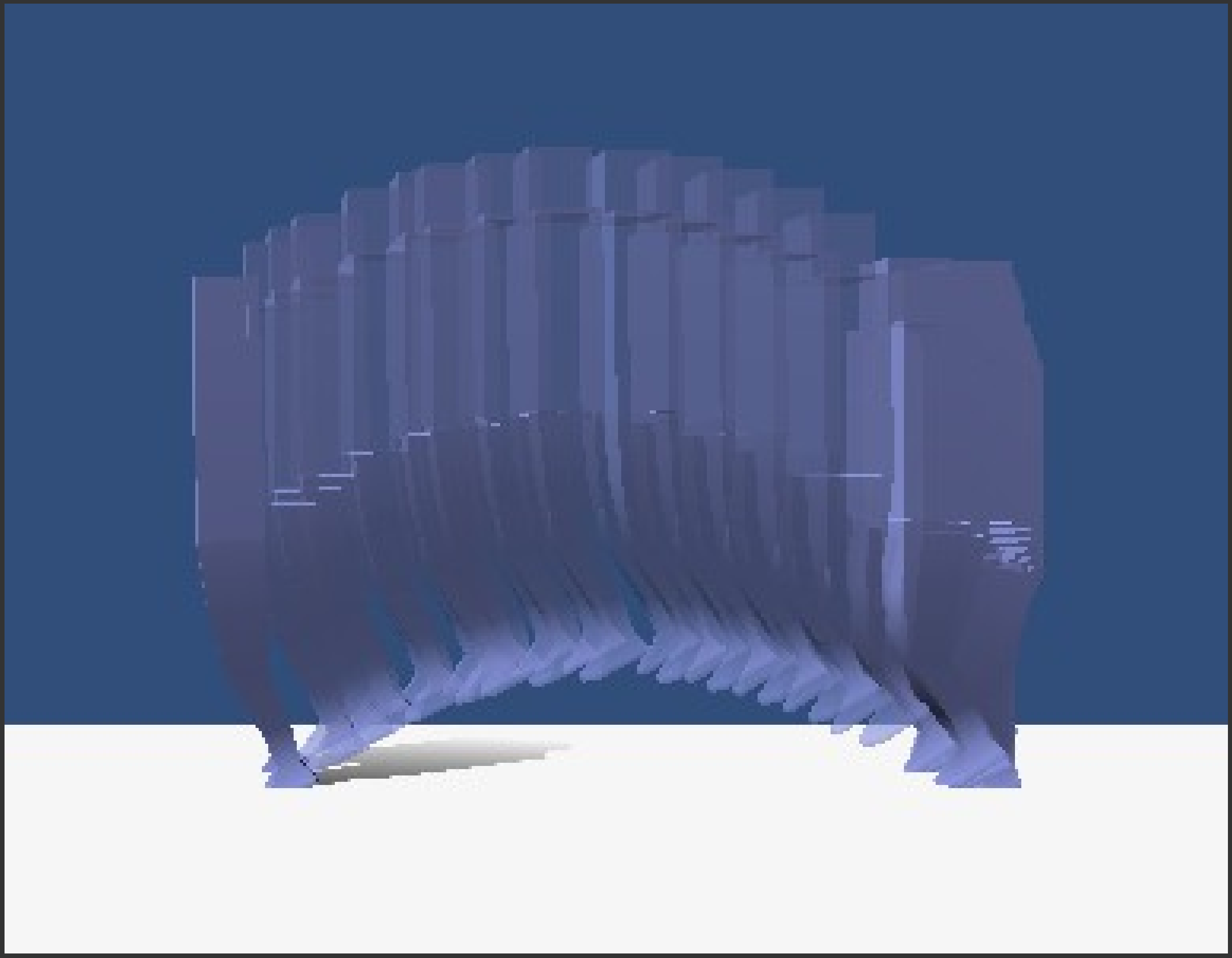
# Ghost Image Visualization

- Instead of leaving a dot behind, leave a copy of the character mesh
- Messy and busy visual, but can show issues in timing and positioning

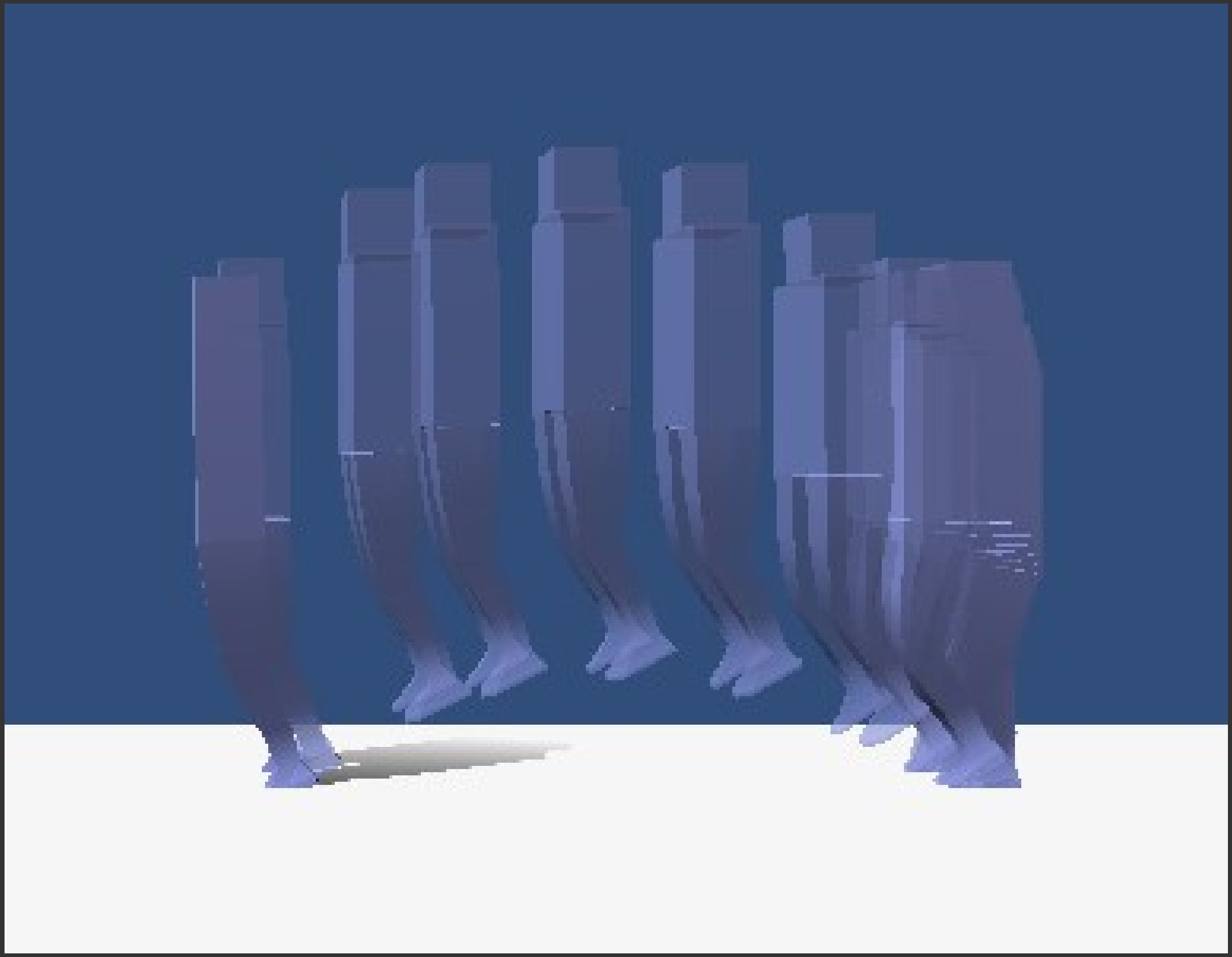




# Ghost Image Visualization

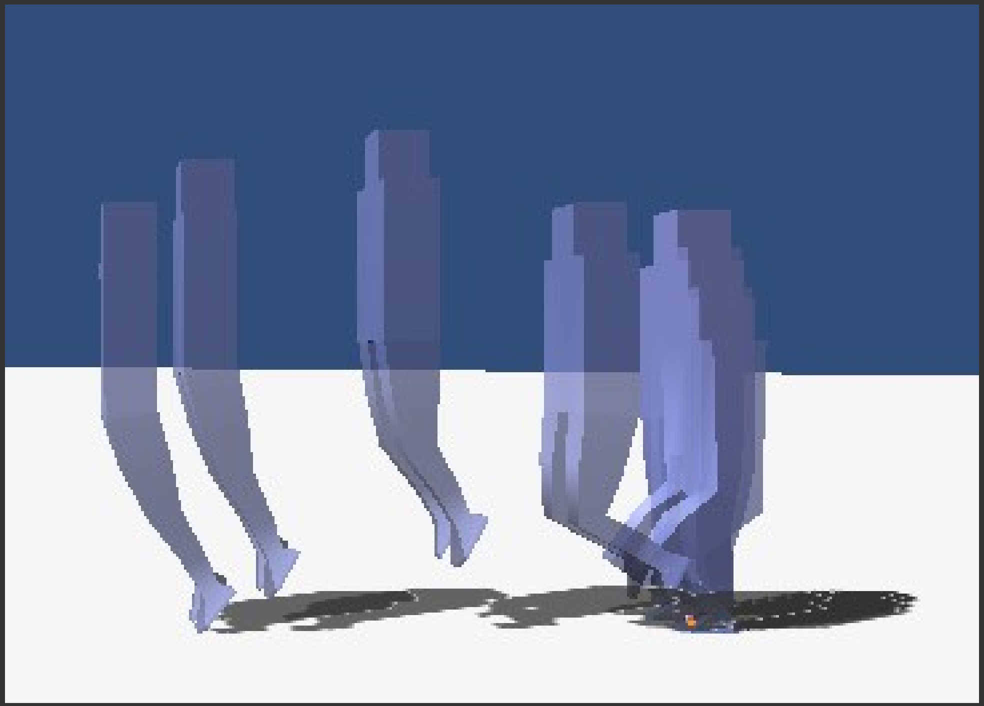


# Ghost Image Visualization



# Frame Composite Visualization

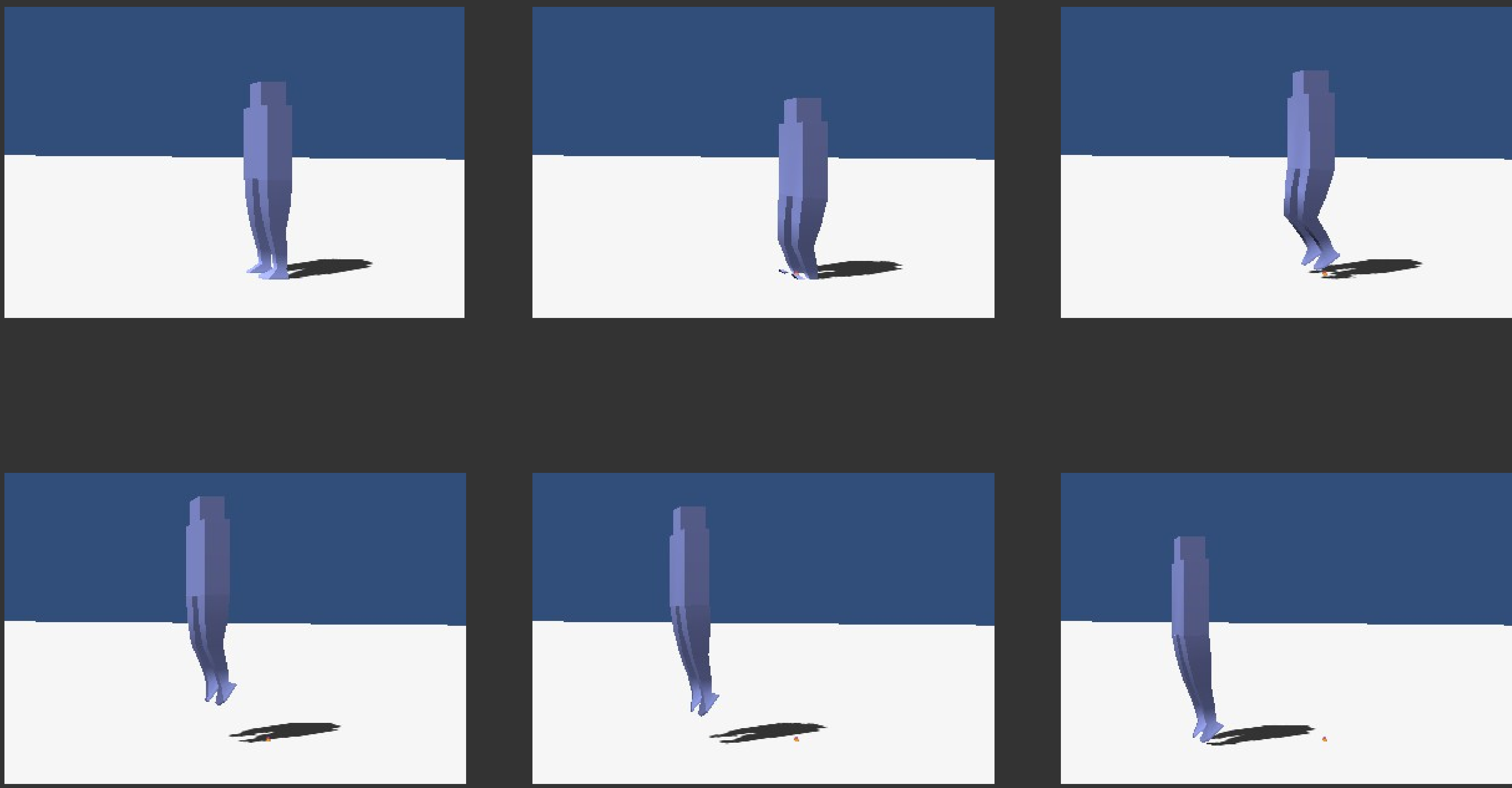
- Select few frames
- Matte image
- Composite mattes
- Similar issues to the ghost image visualization



# Animated Sequences

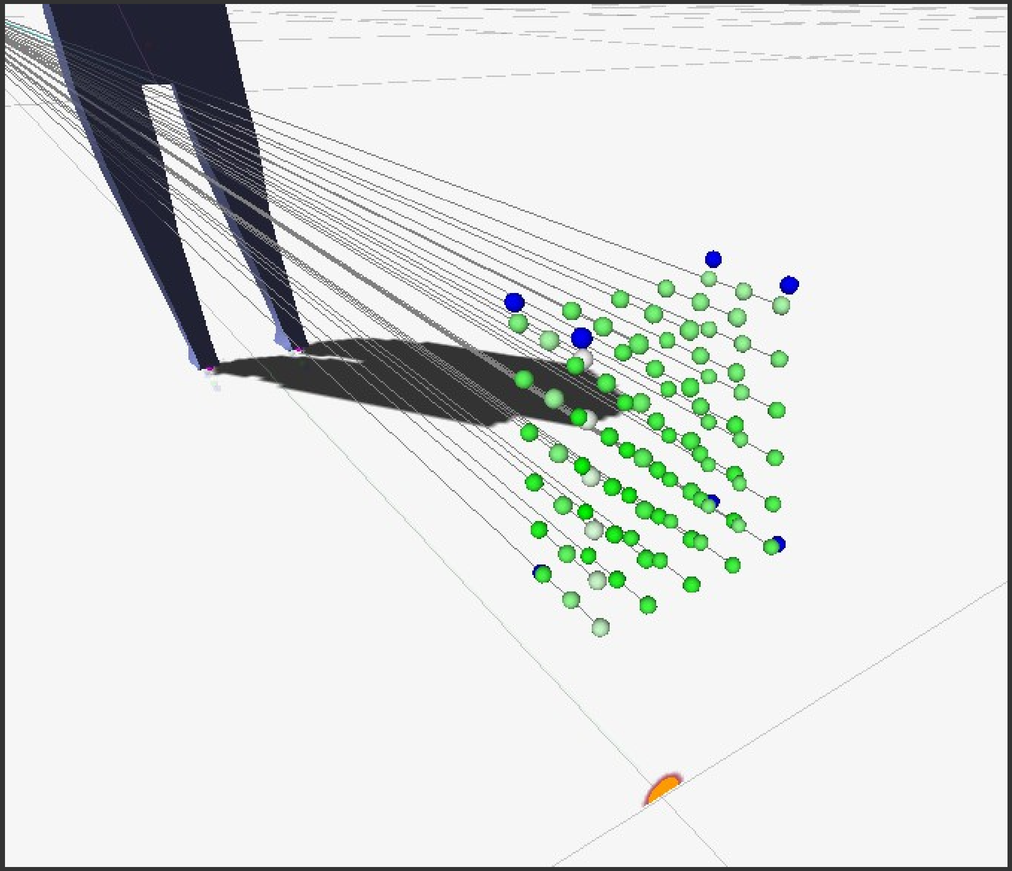
- GIFs as seen in results slides
  - Only usable in digital formats
- Sequences of frames (for print formats)
  - Long sequences of frames
  - space consuming
  - not very clear (user must glance between many frames across several images or over a long image)

# Frame Sequence



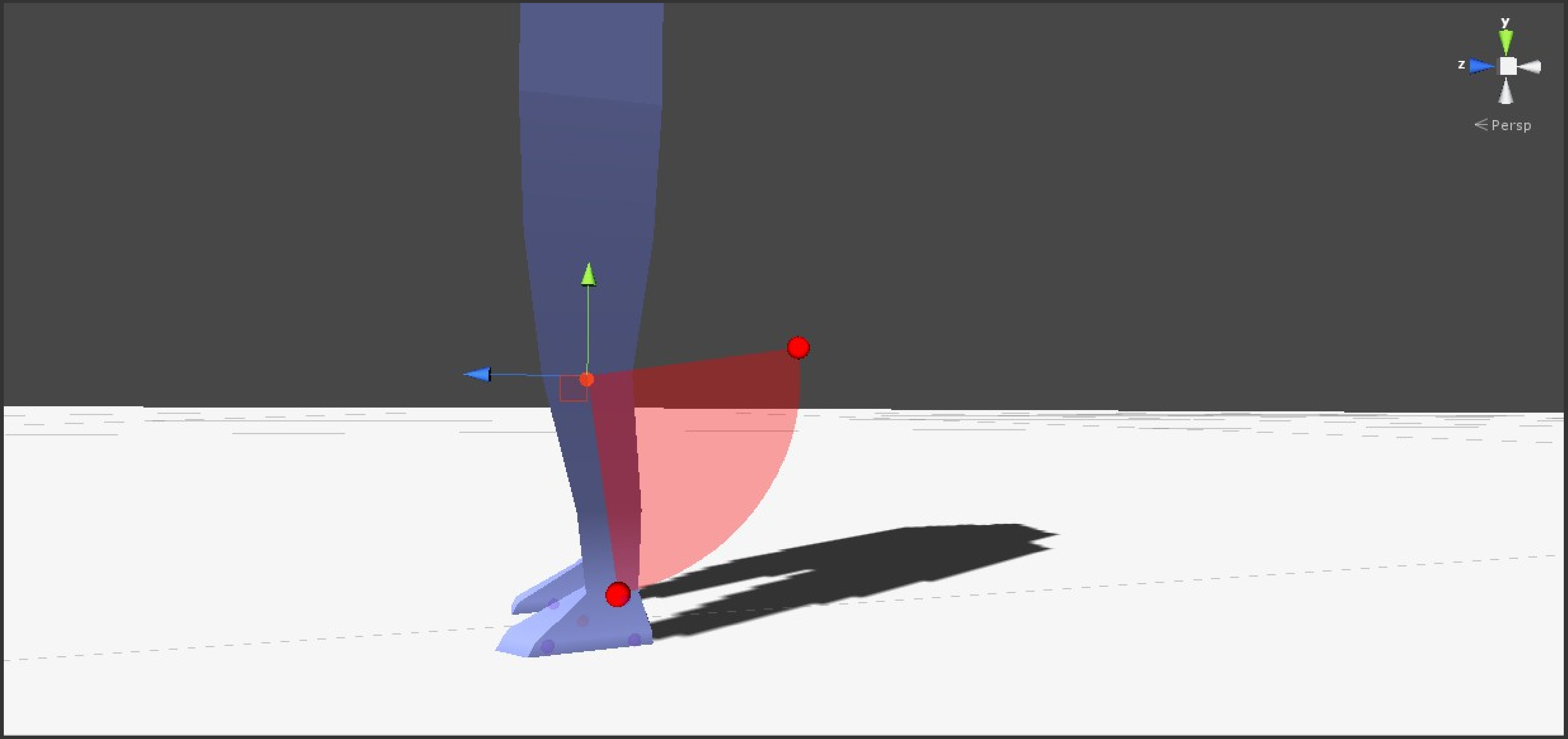
# Gizmos

- Debugging visuals provided by Unity3D
- Used for visualizing vectors and samples

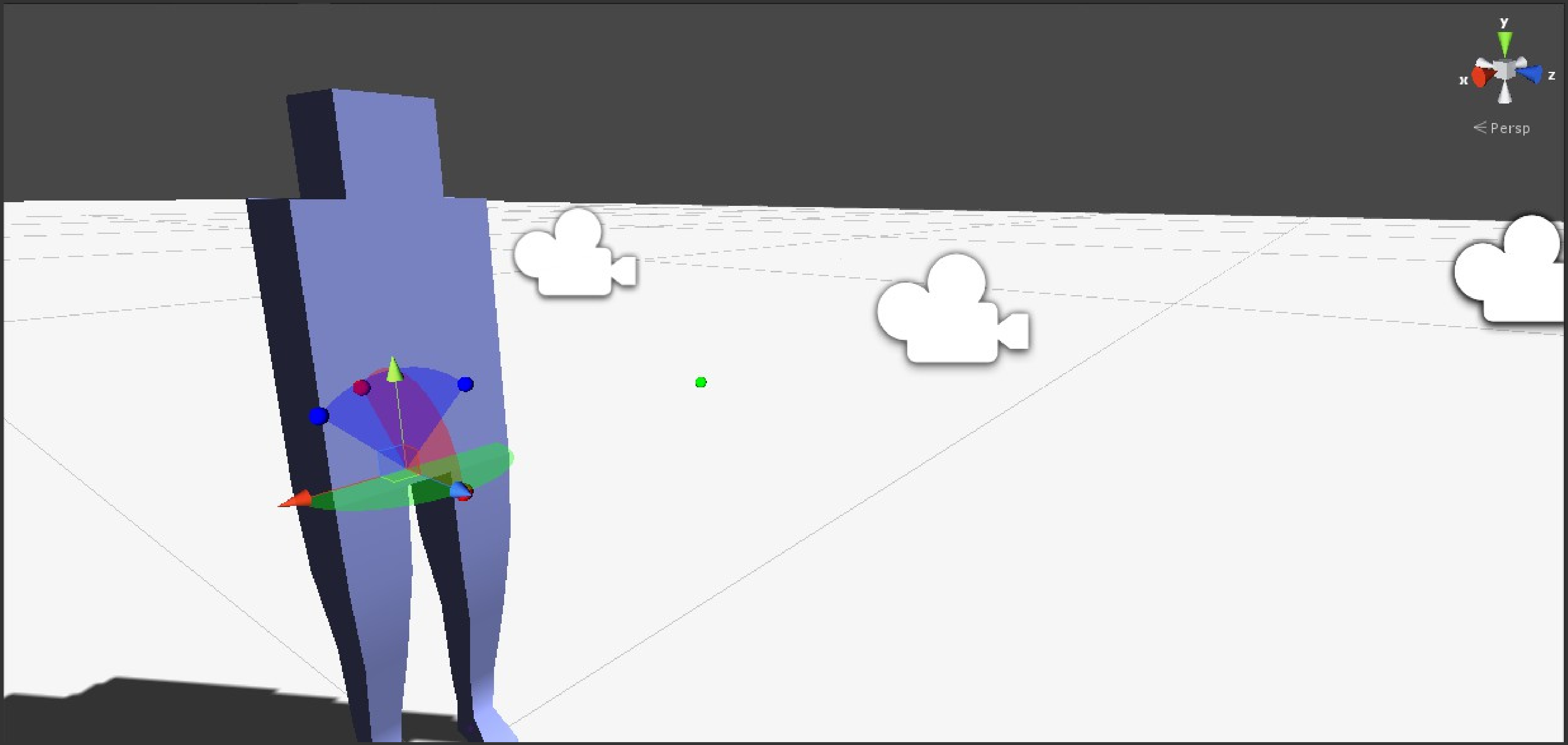


# Handles

- Used for visualizing joint constraints
- Allows visual editing of joint constraints



# Handles



# Agenda

- Introduction
- Skeleton and Muscle Setup
- Torque-based Simulation
- Energy-based Simulation
- Limitations
- Visualization
- **Future Work**

# Future Work

- Allow specification by desired height
  - Jump over achieved by changing times, can instead detect required height and modify time accordingly
- Attempt a learning approach
- Complex in-air control
  - Flips/Acrobatics
  - Vaults

# Thank you for your time