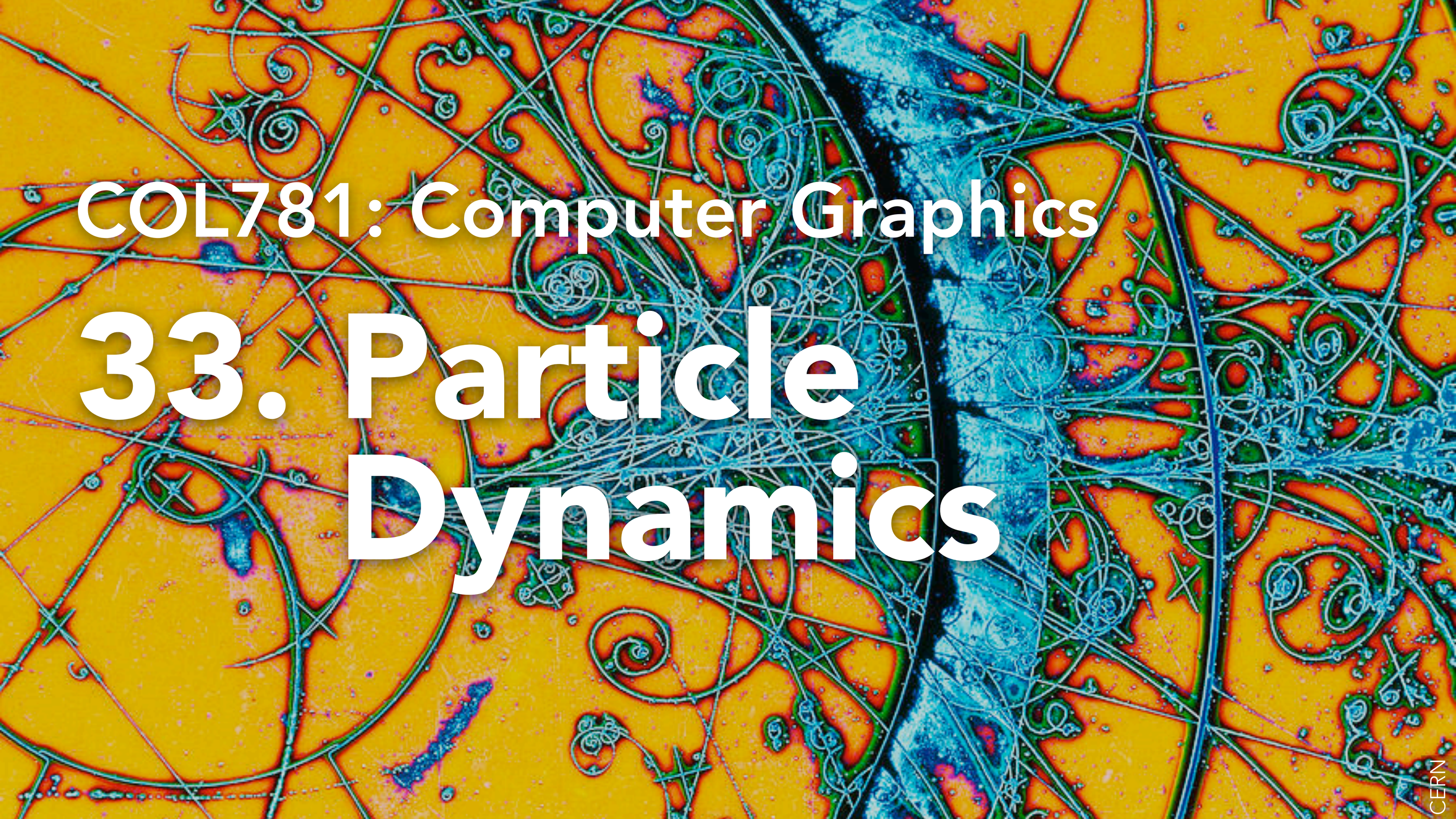


COL781: Computer Graphics

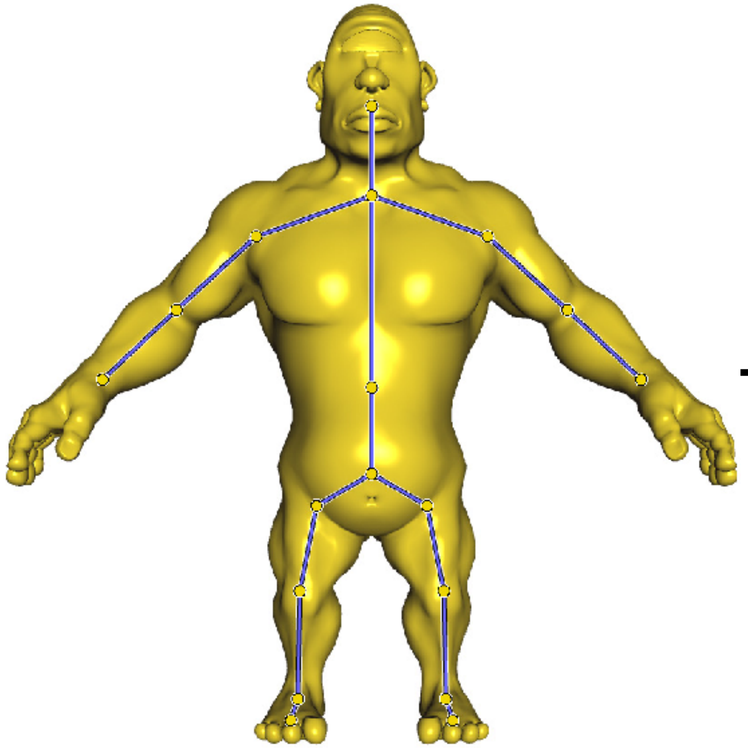
33. Particle Dynamics

CERN

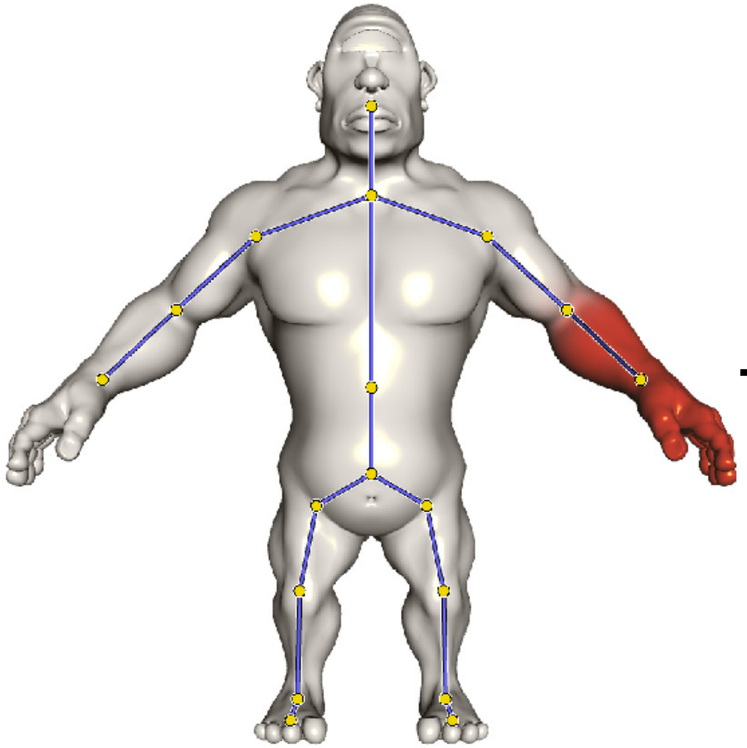
Assignment 3 due today

Assignment 4 coming soon

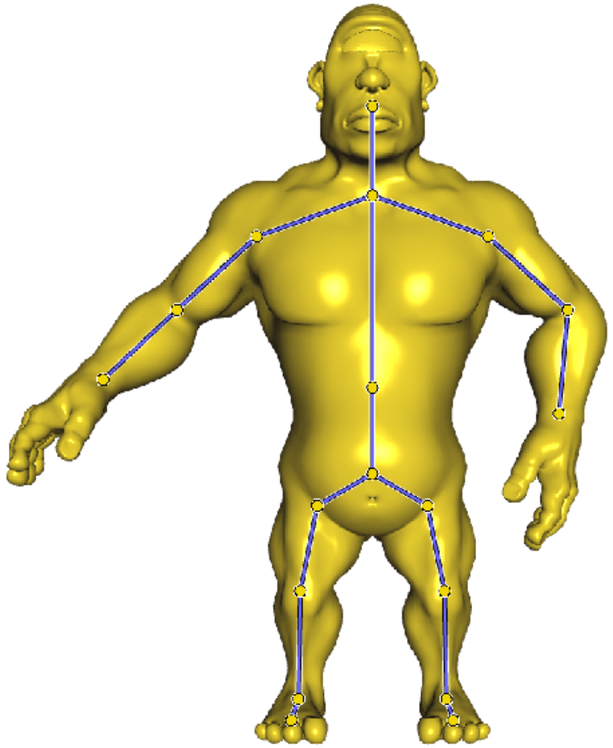
Character animation wrap-up



Define skeleton



Attach skin



Move controls (joints and/or end effectors) to animate

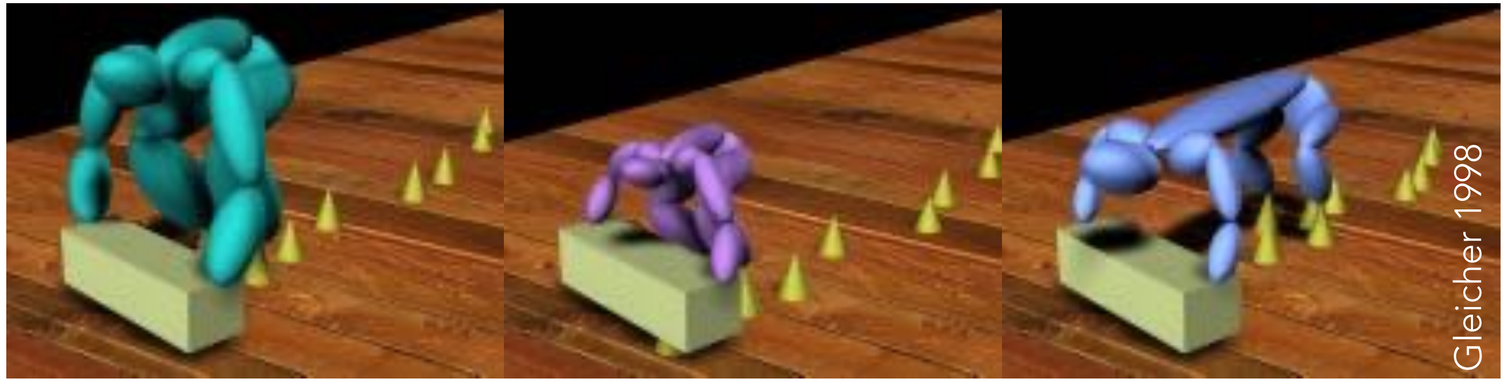
Alec Jacobson

Where does the motion data come from?

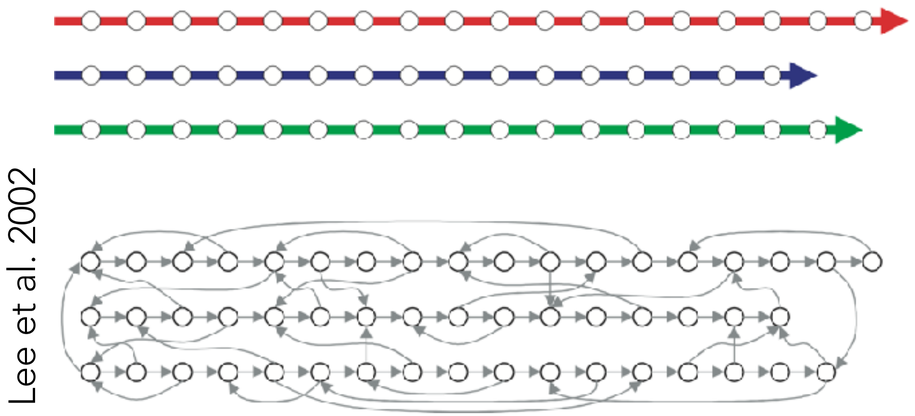
Keyframe animation

Motion capture (“mocap”)

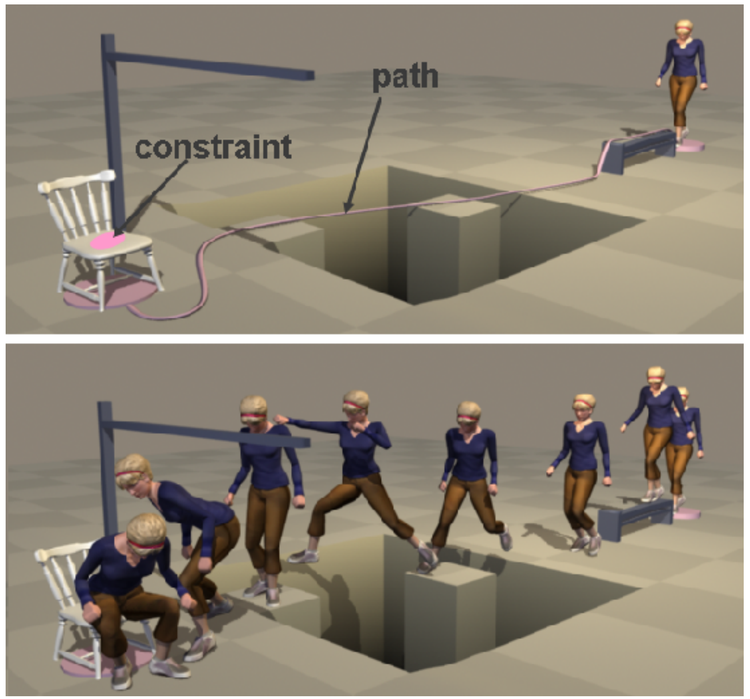
Mocap editing



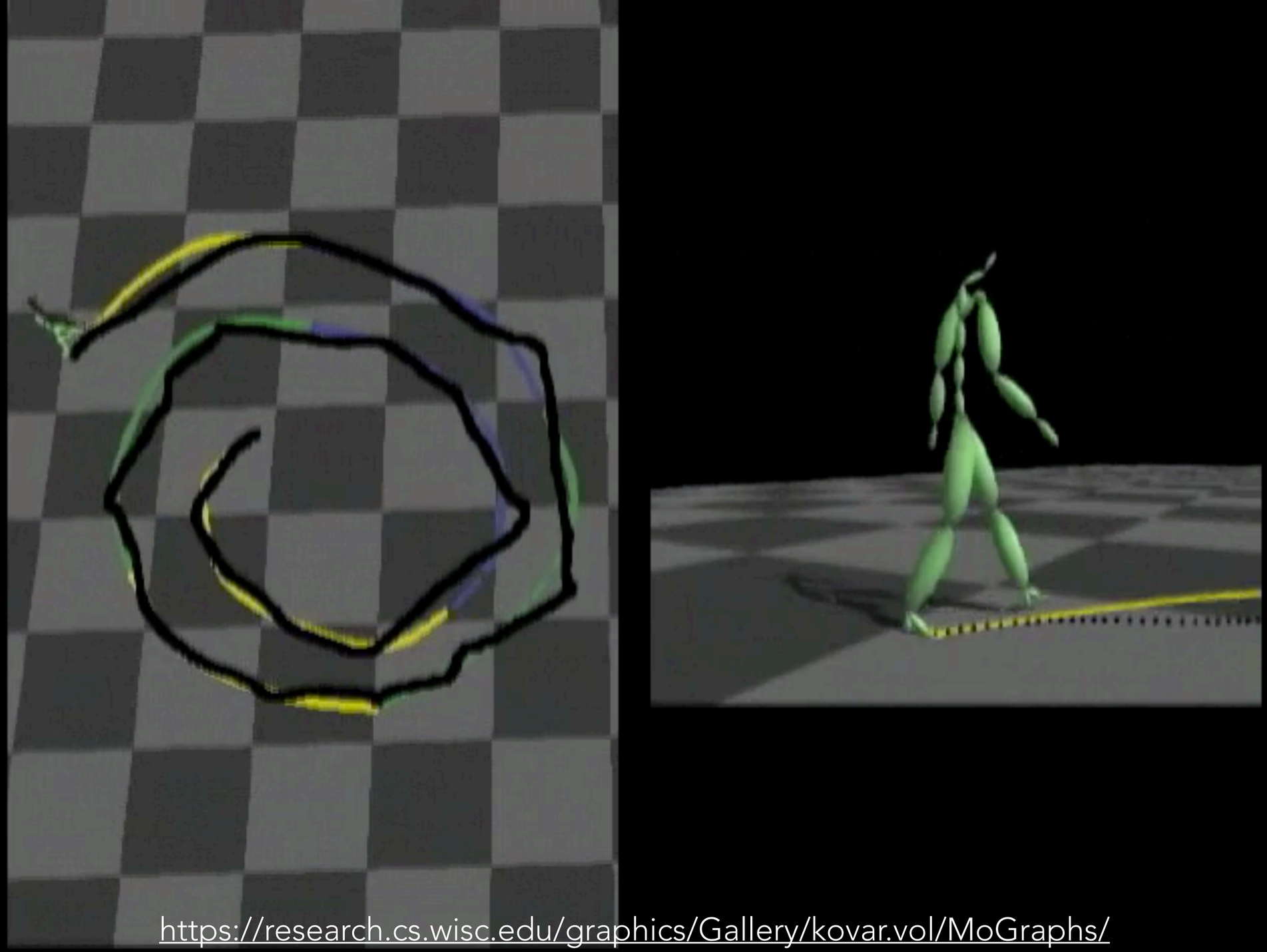
Retargeting



Motion graphs

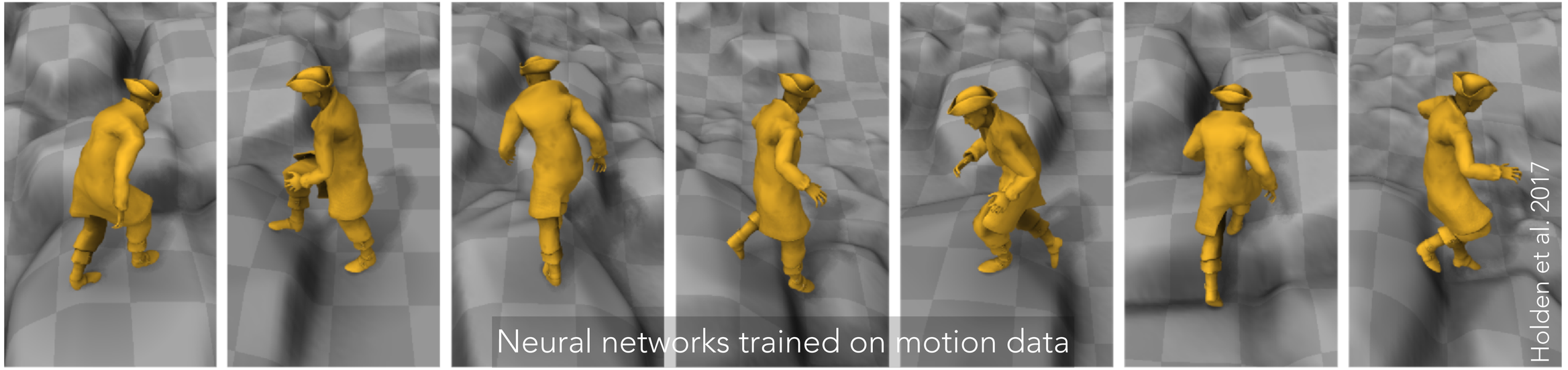


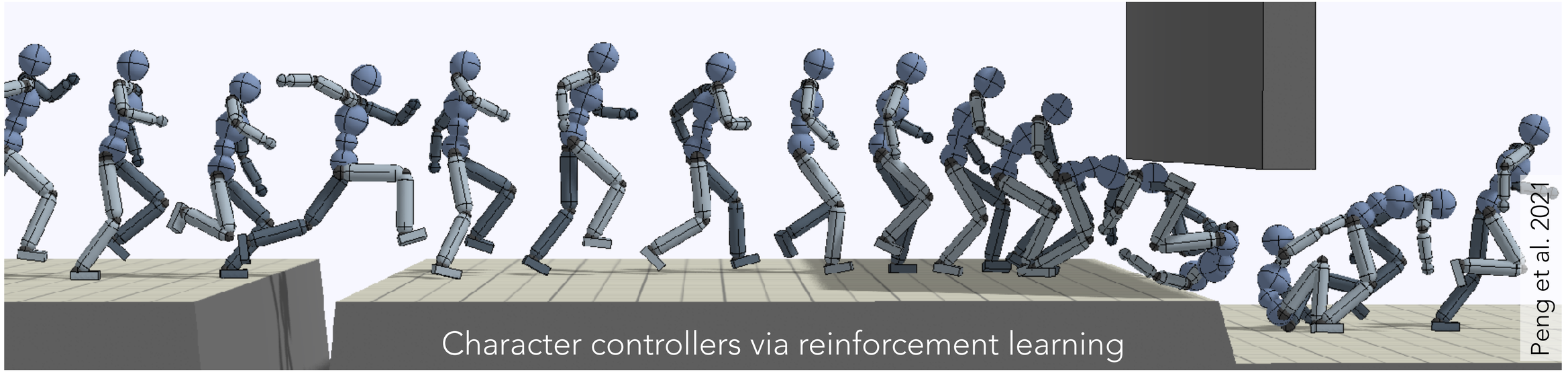
Safonova & Hodgins 2007

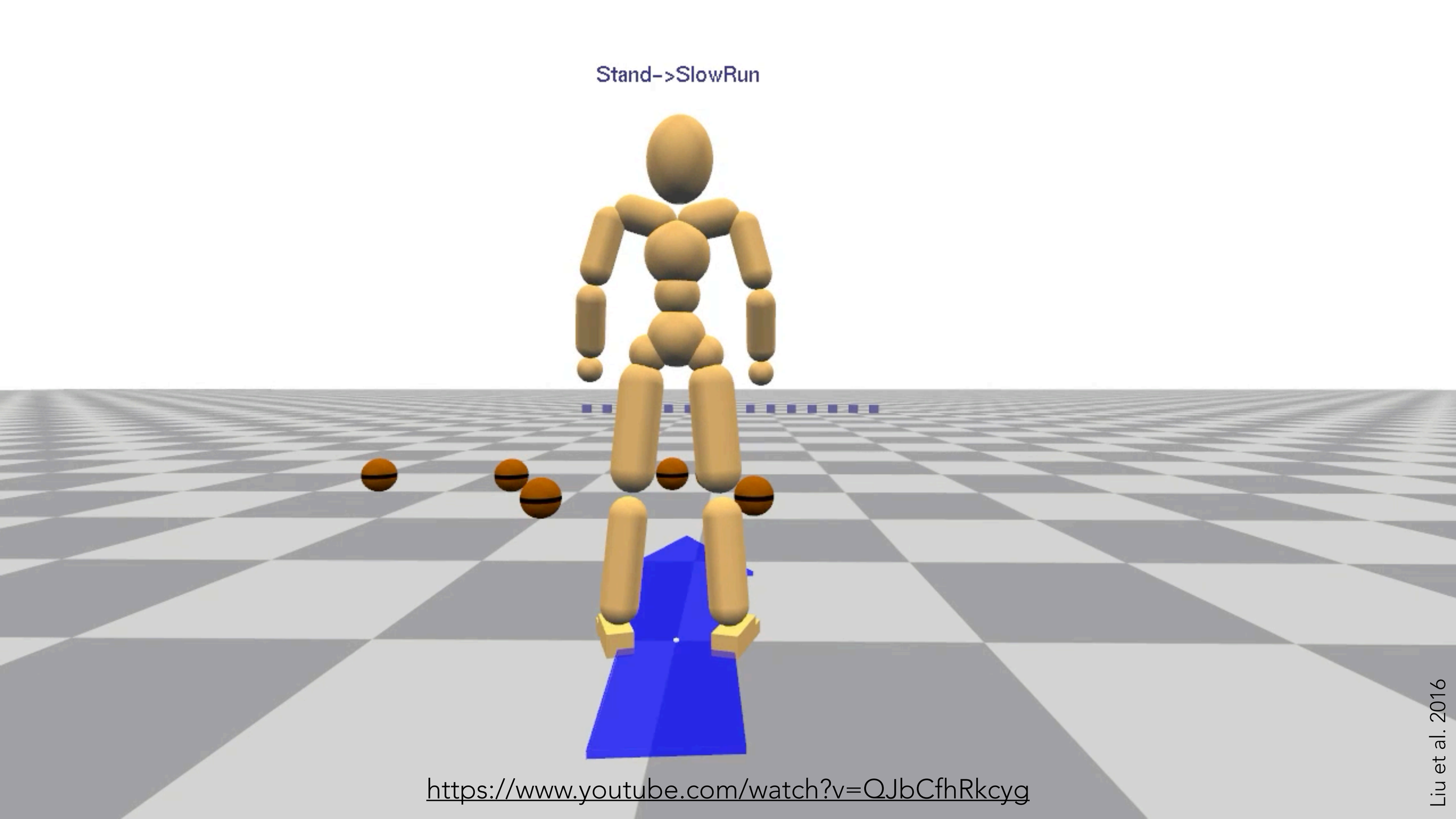


https://research.cs.wisc.edu/graphics/Gallery/kovar.vol/MoGraphs/

Kovar et al. 2002


Neural networks trained on motion data
Holden et al. 2017


Character controllers via reinforcement learning
Peng et al. 2021



https://www.youtube.com/watch?v=QJbCfhRkcyg

Liu et al. 2016

Some things you wouldn’t want to animate using keyframing or motion capture…

Fluids

http://physbam.stanford.edu/~fedkiw/

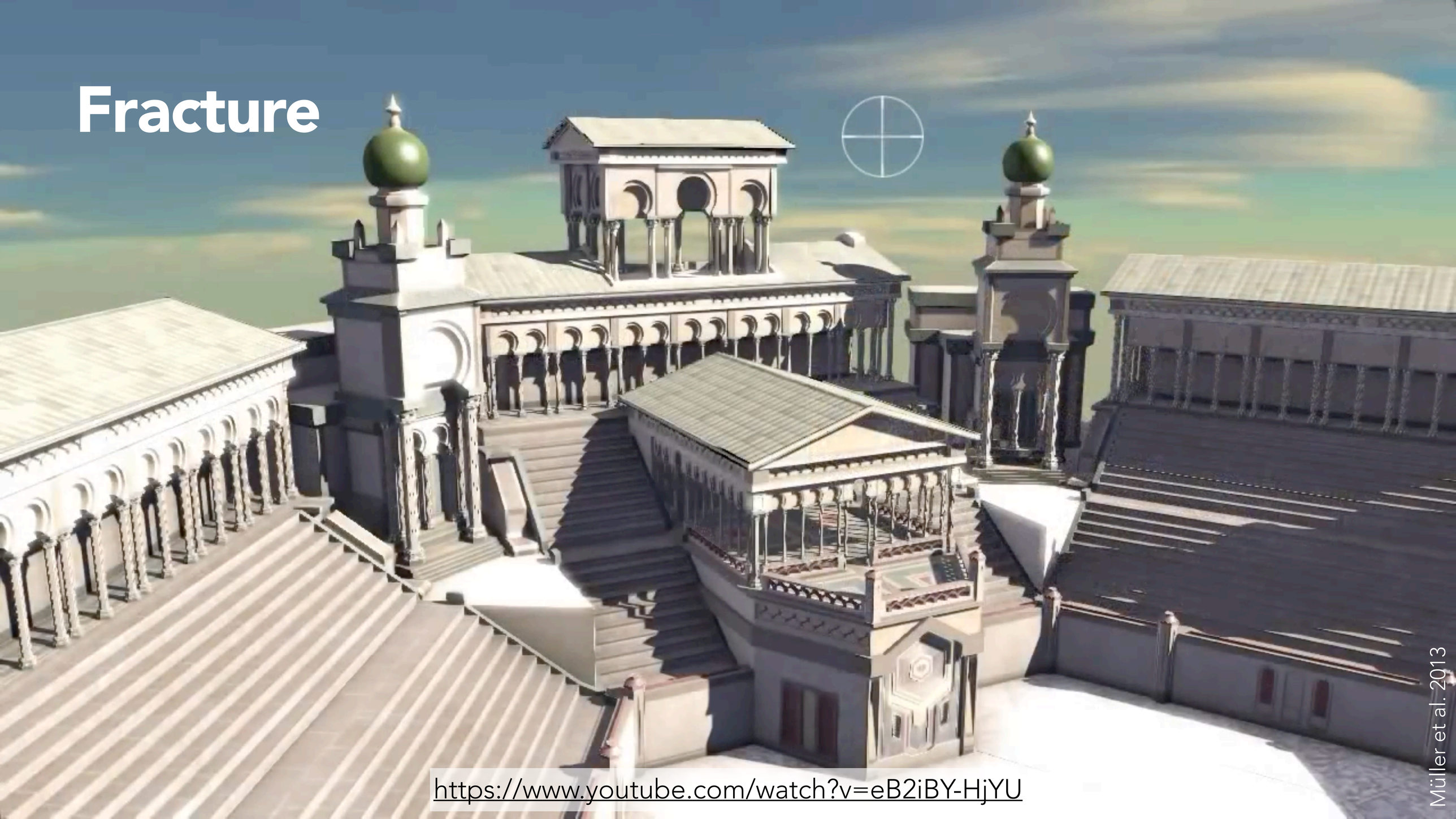
Losasso et al. 2008

Cloth

https://www.cs.columbia.edu/cg/ESIC/esic.html

Goldenthal et al. 2007

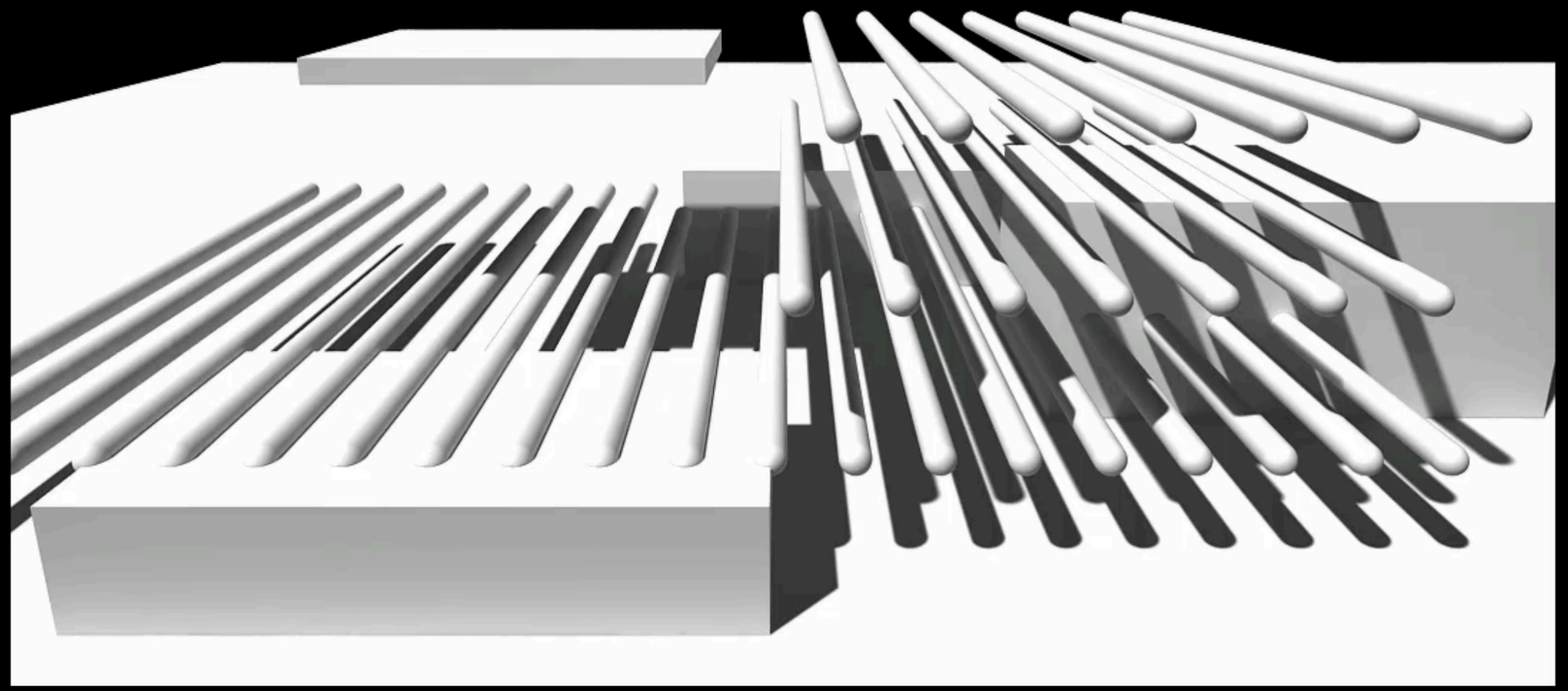
Fracture



https://www.youtube.com/watch?v=eB2iBY-HjYU

Müller et al. 2013

The simplest physics-based characters :)

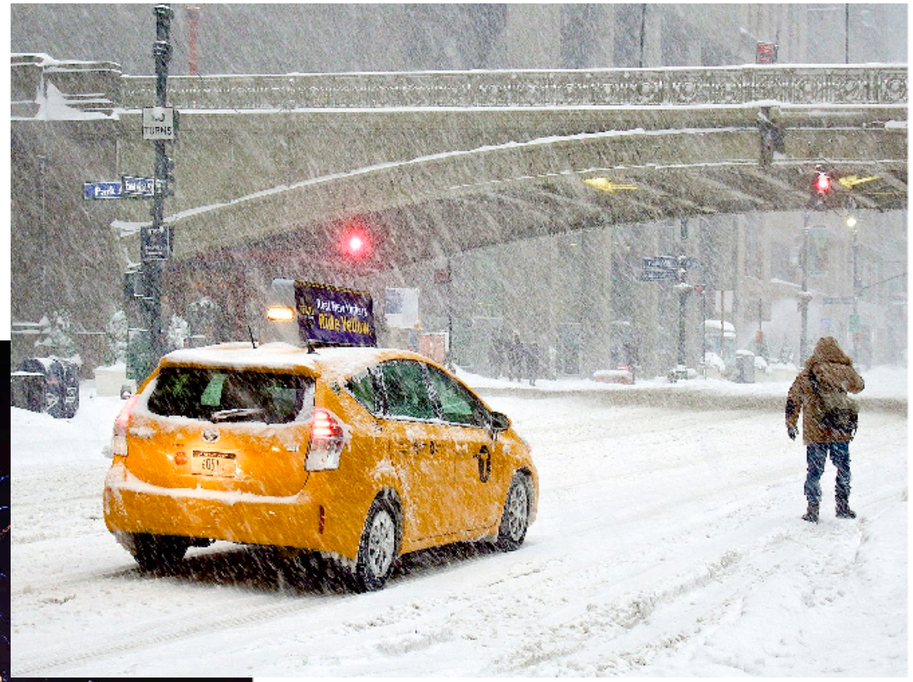


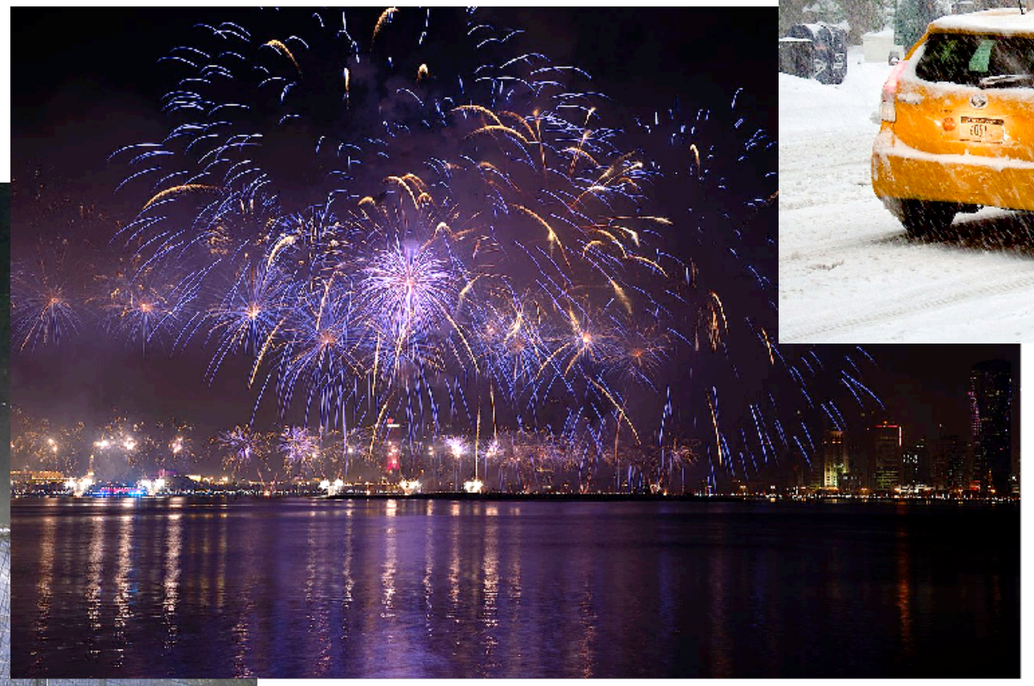
Lewin et al. 2013

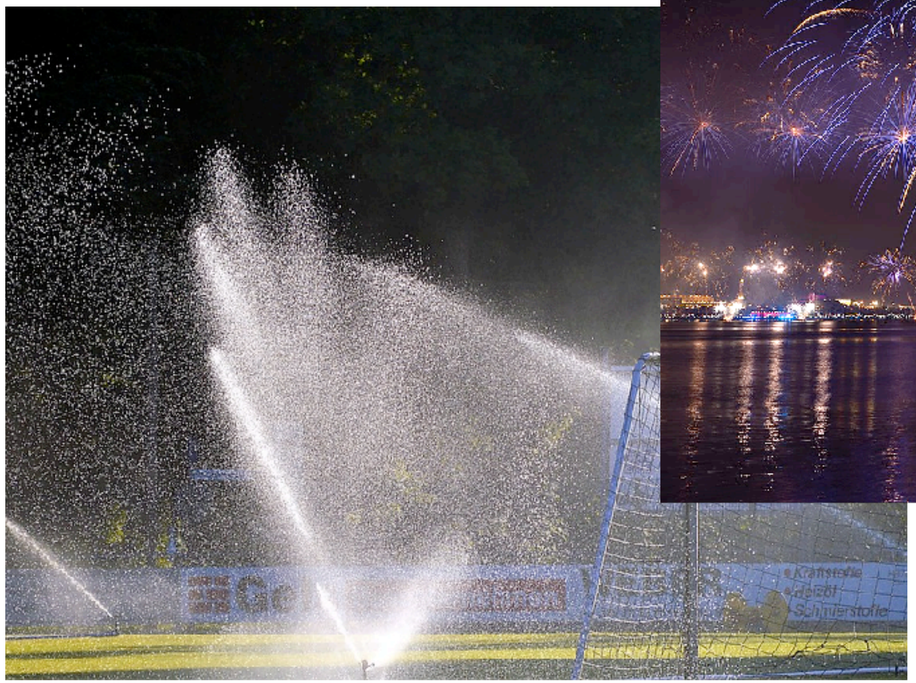
https://researchportal.bath.ac.uk/en/publications/rod-constraints-for-simplified-ragdolls

Let’s start simple…

Particle system = collection of (usually non-interacting) particles in motion



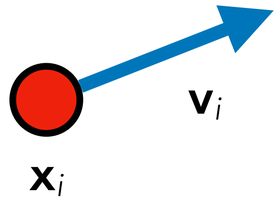




Particle systems

Each particle is a point mass

- Fixed: mass m_i
- Variable state: position $\mathbf{x}_i$, velocity $\mathbf{v}_i$



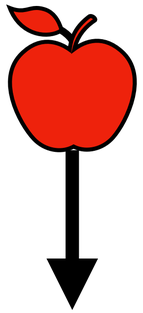
Created by some user-specified emitter

Affected by some user-specified forces

- Usually independent of other particles' states: $\mathbf{f} = \mathbf{f}(t, \mathbf{x}_i, \mathbf{v}_i)$

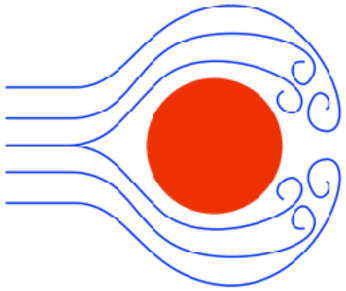
Optionally, deleted after some user-specified lifetime

What kinds of forces?



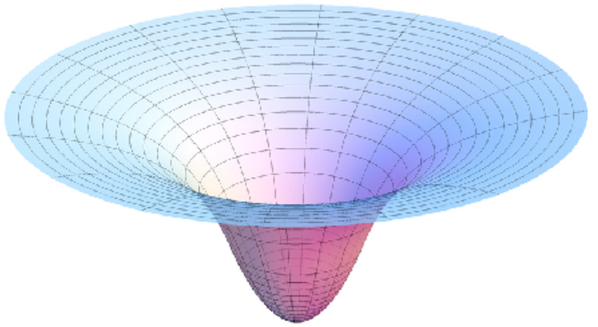
Gravity

$\mathbf{f} = m\mathbf{g}$



Wind / drag

$\mathbf{f} = -k_d(\mathbf{v} - \mathbf{v}_{\text{air}})$

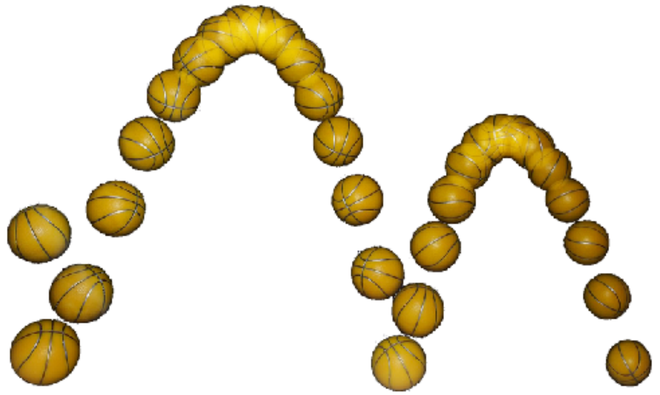


Spatial fields

$\mathbf{f} = \mathbf{f}(\mathbf{x})$

Collisions

$\mathbf{f}$ = …TBD



etc.

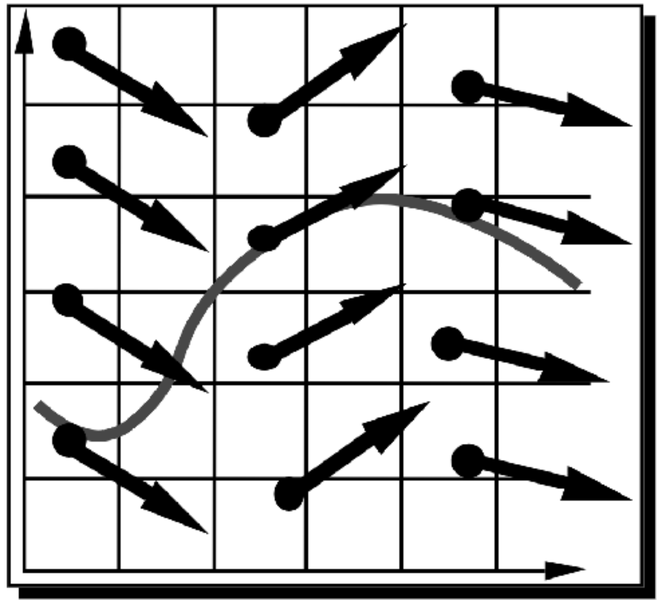
Equations of motion: **f** = *m***a** (where **f** is total force) so…

$$\frac{\mathrm{d}^2\mathbf{x}(t)}{\mathrm{d}t^2} = m^{-1}\,\mathbf{f}(t, \mathbf{x}(t), \mathbf{v}(t))$$

For each emitted particle, we know initial position **x**(0) and velocity **v**(0). How to find **x**(*t*), **v**(*t*) at any future time *t*?

In general, no closed form unless **f** is very simple!

Like with rendering, we need a numerical method…



Witkin & Baraff 1997

Time stepping

Idea: Given a known state ($\mathbf{x}(t)$, $\mathbf{v}(t)$), estimate a near future state ($\mathbf{x}(t+\Delta t)$, $\mathbf{v}(t+\Delta t)$).

Then we can iterate: ($\mathbf{x}(0)$, $\mathbf{v}(0)$) → ($\mathbf{x}(\Delta t)$, $\mathbf{v}(\Delta t)$) → ($\mathbf{x}(2\Delta t)$, $\mathbf{v}(2\Delta t)$) → ($\mathbf{x}(3\Delta t)$, $\mathbf{v}(3\Delta t)$) → ⋯

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}(t)$$

$$\frac{\mathrm{d}\mathbf{v}(t)}{\mathrm{d}t} = m^{-1}\,\mathbf{f}(t, \mathbf{x}(t), \mathbf{v}(t))$$

Note that e.g. $\dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \lim\limits_{h\to 0} \dfrac{\mathbf{x}(t+h) - \mathbf{x}(t)}{h} \approx \dfrac{\mathbf{x}(t+\Delta t) - \mathbf{x}(t)}{\Delta t}$, so

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\,\Delta t$$

$$\frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t)}{\Delta t} \approx \mathbf{v}$$

$$\frac{\mathbf{v}(t + \Delta t) - \mathbf{v}(t)}{\Delta t} \approx m^{-1}\,\mathbf{f}(t, \mathbf{x}, \mathbf{v})$$

Simplest strategy:

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + m^{-1}\,\mathbf{f}(t, \mathbf{x}(t), \mathbf{v}(t))\,\Delta t$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t + \Delta t)\,\Delta t$$

because we already have it from the previous step

Tradeoff! Smaller $\Delta t \Rightarrow$ better estimate,

but more computation to reach any desired t.

PARTICLE DREAMS

Karl Sims

Optomystic

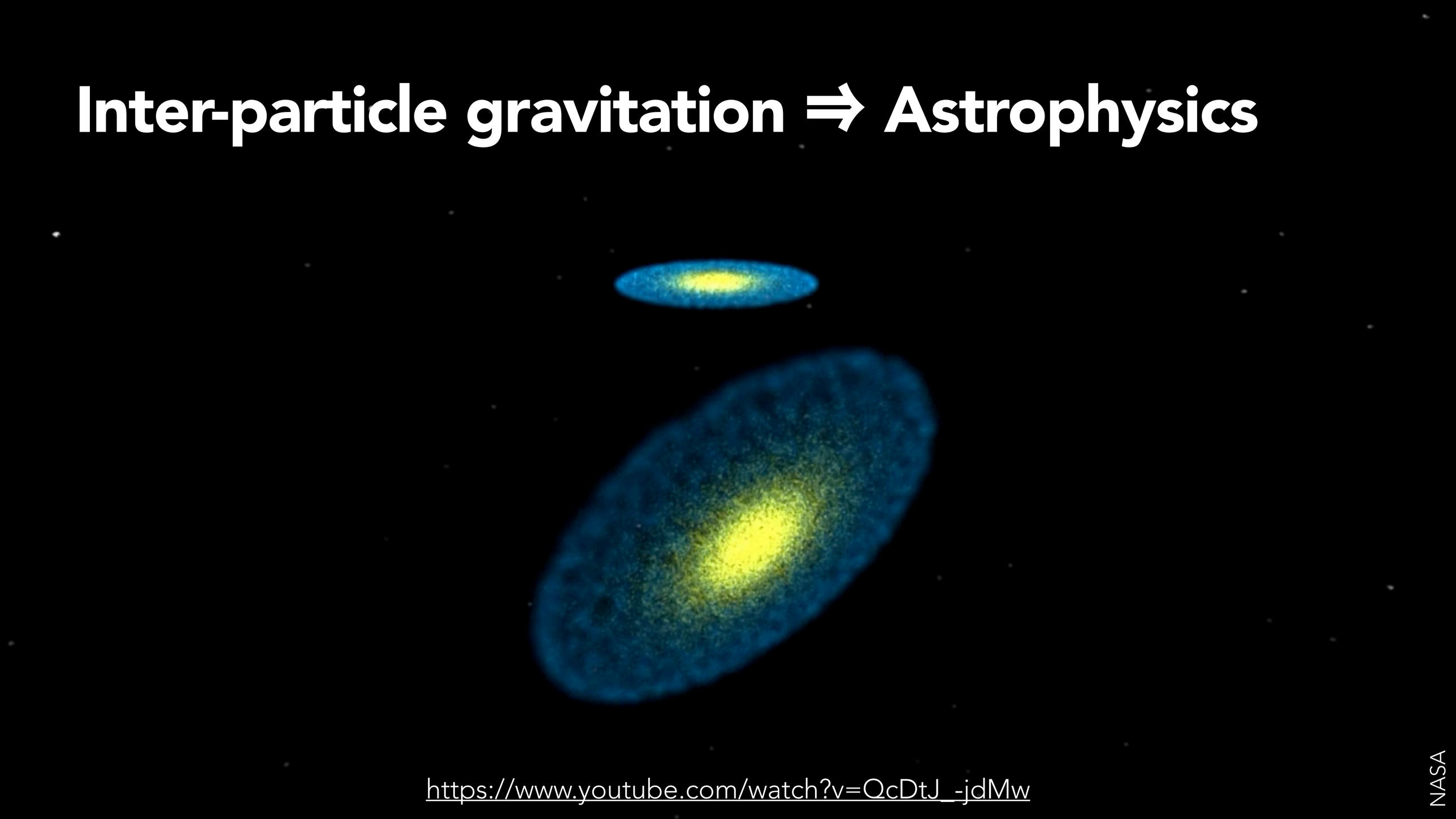
https://www.youtube.com/watch?v=5QEp-oPaQto

What if we add forces between particles?

Total force on *i*th particle may depend on other particles' states $\mathbf{x}_1$, $\mathbf{v}_1$, $\mathbf{x}_2$, $\mathbf{v}_2$, …

Much more interesting behaviour!

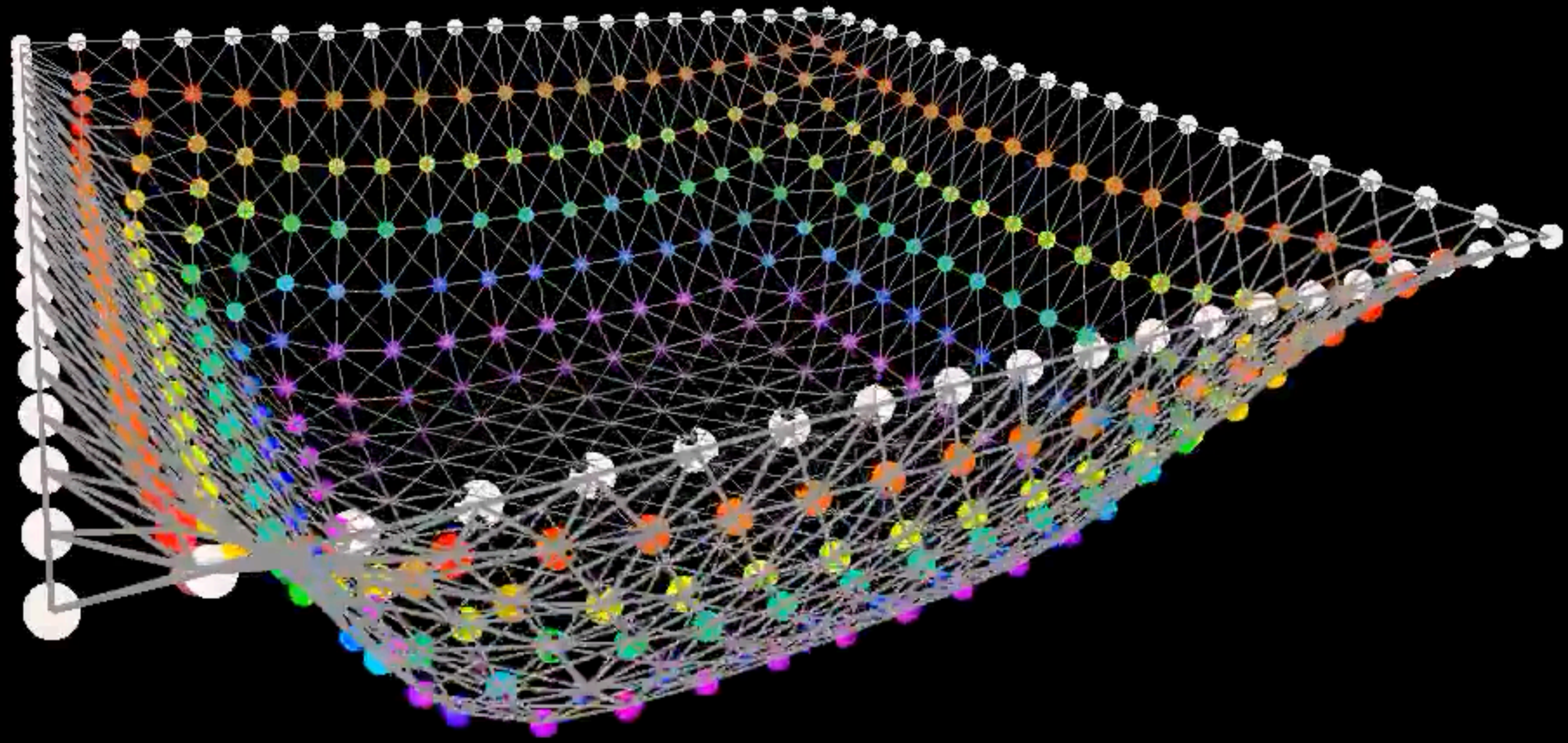
Inter-particle gravitation ⇒ Astrophysics



https://www.youtube.com/watch?v=QcDtJ_-jdMw

NASA

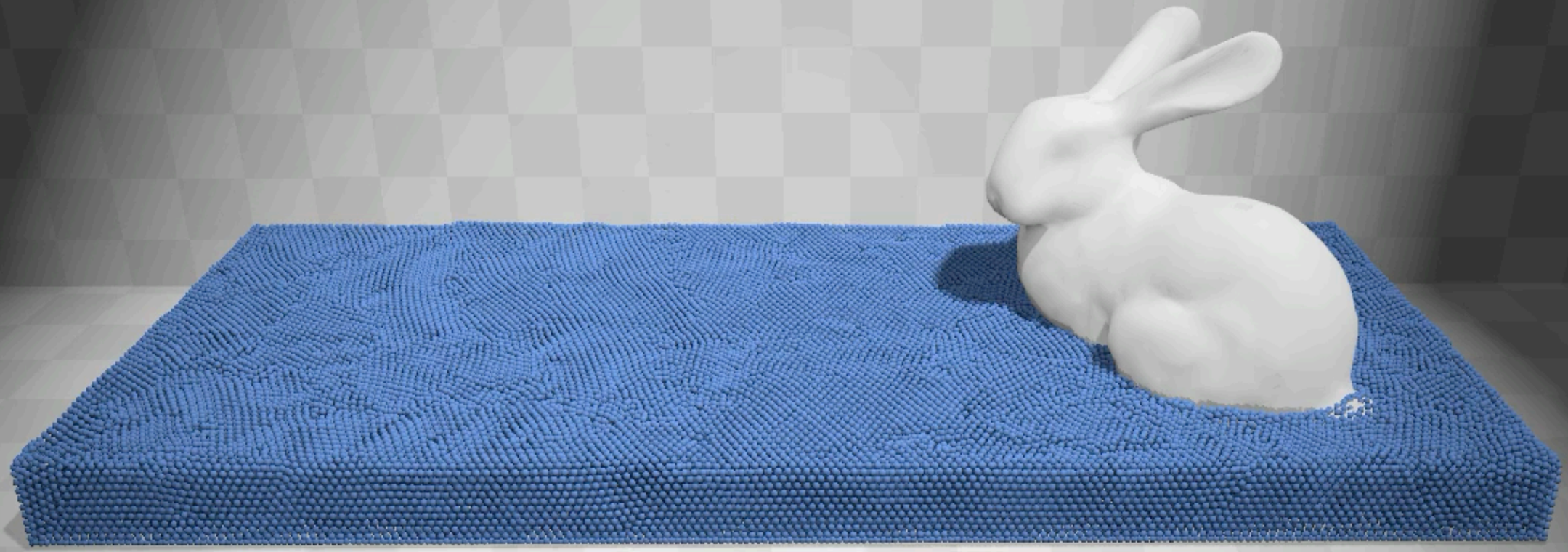
(Damped) springs ⇒ Cloth, hair, etc.



https://www.youtube.com/watch?v=ib1vmRDs8Vw

Paul Nathan

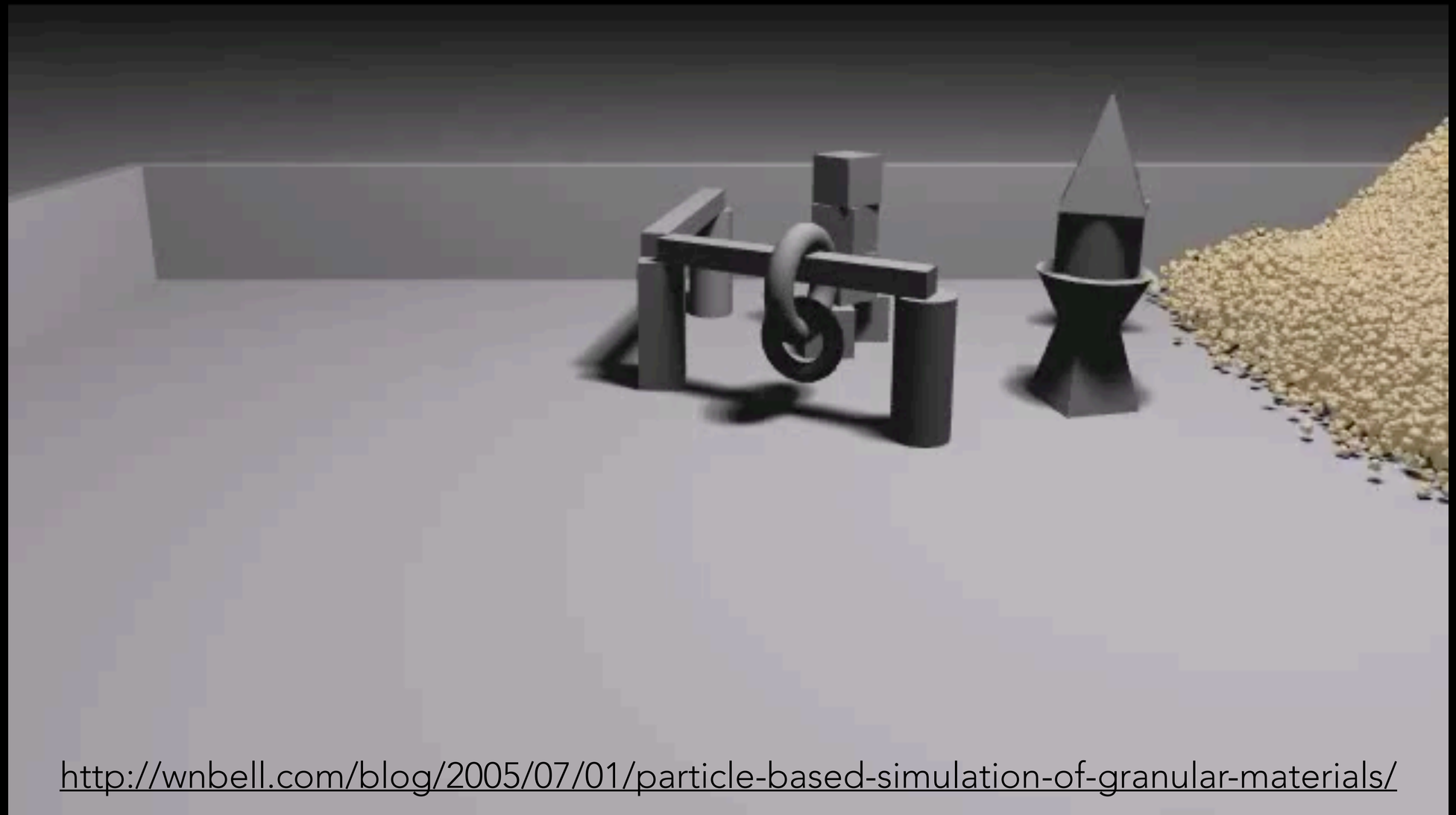
Density-based repulsion ⇒ Fluids



https://www.youtube.com/watch?v=F5KuP6qEuew

Macklin & Müller 2013

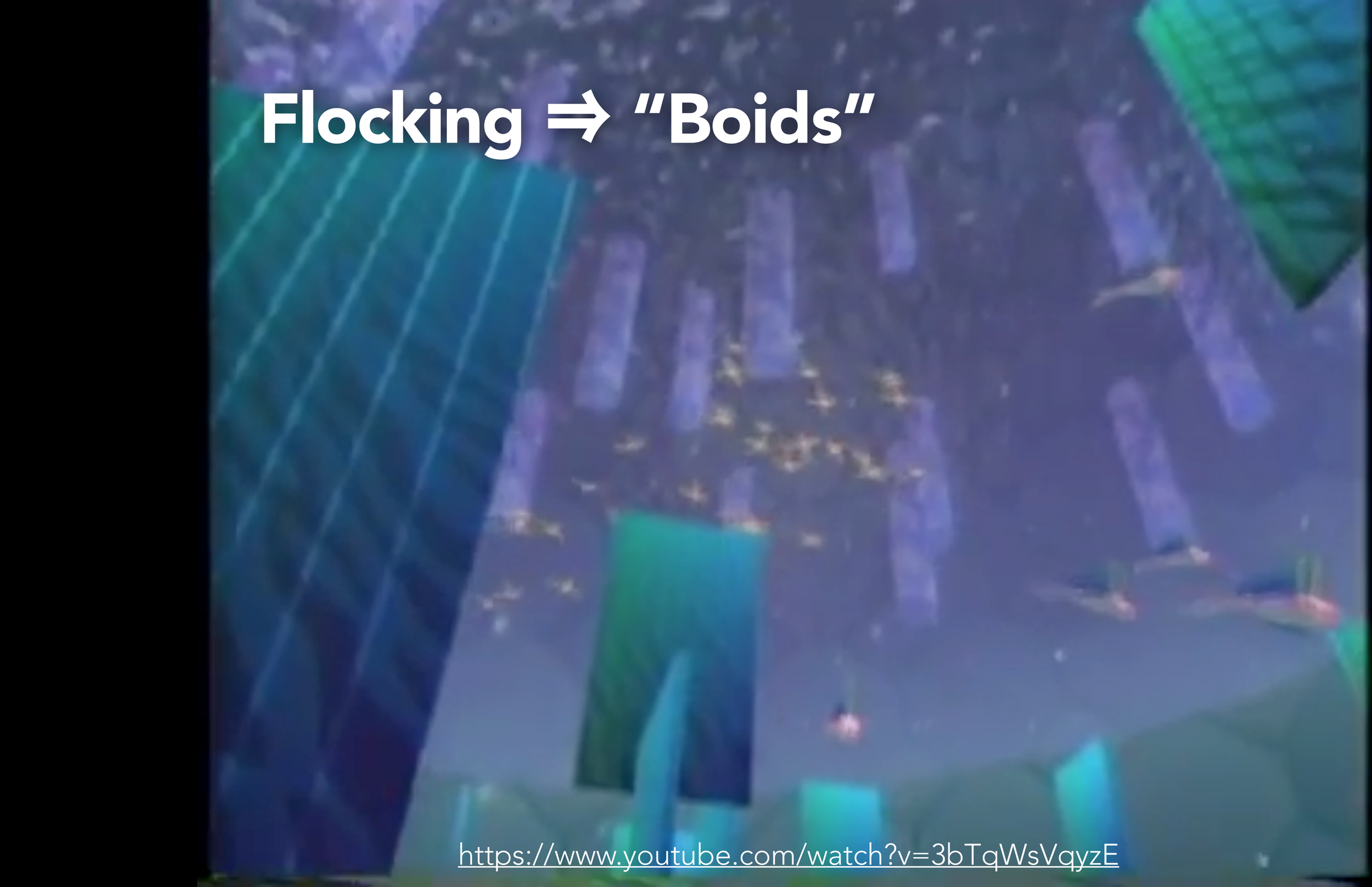
Frictional contact ⇒ granular materials



http://wnbell.com/blog/2005/07/01/particle-based-simulation-of-granular-materials/

Bell et al. 2005

Flocking ⇒ “Boids”



https://www.youtube.com/watch?v=3bTqWsVqyzE

Craig Reynolds

Collision avoidance ⇒ Crowds

https://www.youtube.com/watch?v=NltNl7_1uHs

Weiss et al. 2017

May depend on $\mathbf{x}_1(t)$, $\mathbf{v}_1(t)$, $\mathbf{x}_2(t)$, $\mathbf{v}_2(t)$, …

How to compute? Same strategy:

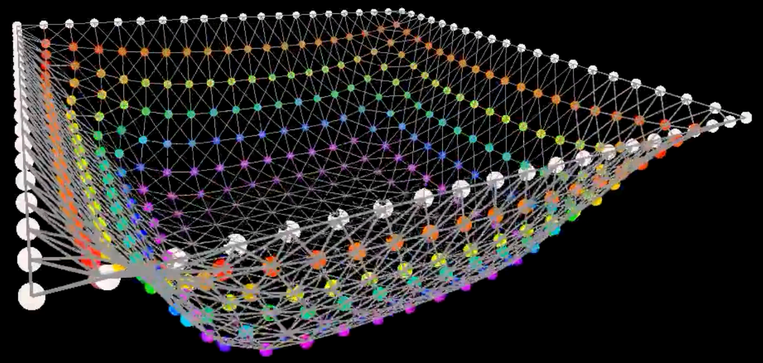
$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + m_i^{-1}\,\mathbf{f}_i(t)\,\Delta t$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t + \Delta t)\,\Delta t$$

Pseudocode:

```
for each particle p:
   p.f = 0
for each force object F:
   for each particle p affected by F:
      p.f += force on p due to F
for each particle p:
   p.v += p.f/p.m * dt
   p.x += p.v * dt
```

Next class: more about mass-spring systems and smarter time stepping schemes



Homework:

Implement a simple particle system (whether in your own graphics code or using a plotting tool e.g. Matplotlib) to simulate something realistic e.g. fireworks, sprinklers, etc.