Mapping optical motion capture data to skeletal motion using a physical model

Victor B. Zordan
Nicholas C. Van Der Horst

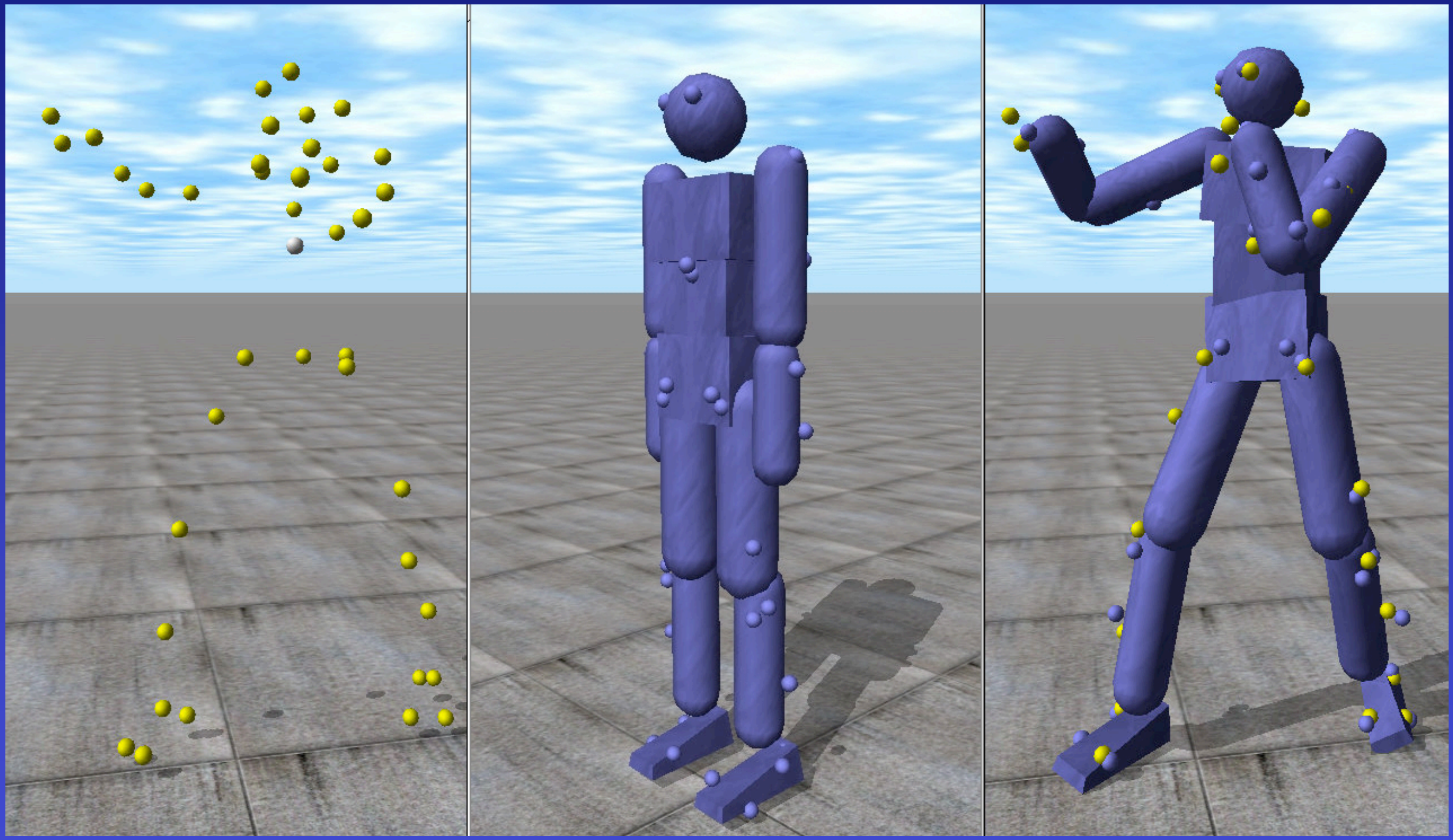
University of California, Riverside

Riverside Graphics Lab

Motivation

Motivation

Optical data + Skeleton ⟶ Posture



Problems: no perfect match, joint-center and rigid-body assumptions, limits on ranges of motion, aesthetic and production requirements

Motivation

Isn't this problem solved by inverse kinematics (IK) in commercial solvers?

- Data is becoming more available (e.g. CMU mocap site)
  - BUT you want to map it to our own character
- Money... Filmbox is expensive!

IK vs. our physical modeling approach

- Direct mapping of data - landmark for landmark
- Whole body solution - root gets no special priority
- Easily avoids singularities - straight limbs not a problem
- Avoids footskate - via ground contact reaction forces

Motivation

Recorded data is becoming more available (e.g. CMU site) but we want to map it to our own character

Commercial packages exist (like Kaydara's filmbox and Vicon's Motionbuilder) but they are expensive

Also, their solution is based on inverse kinematics (IK) which has known problems that lead to noticable flaws:

1) Ill-defined singularities yielding limbs that do not become fully straight
2) Indirect, root-centric mapping leading to errors that propogate, e.g. footskate
3) Redundancies corrected by adhoc heuristics causing various quirk artifacts

Background

Motion capture editing

Too many to mention, see mocap session SIGGRAPH ’02

Mapping to skeletons

Silaghi, Plankers, Fua, Boulic, Fua, Thalmann ’98
Molet, Boulic, Thalmann ’99
Monzani, Baerlocher, Boulic, Thalmann ’00
O`Brien, Bodenheimer, Brostow, Hodgins ’00

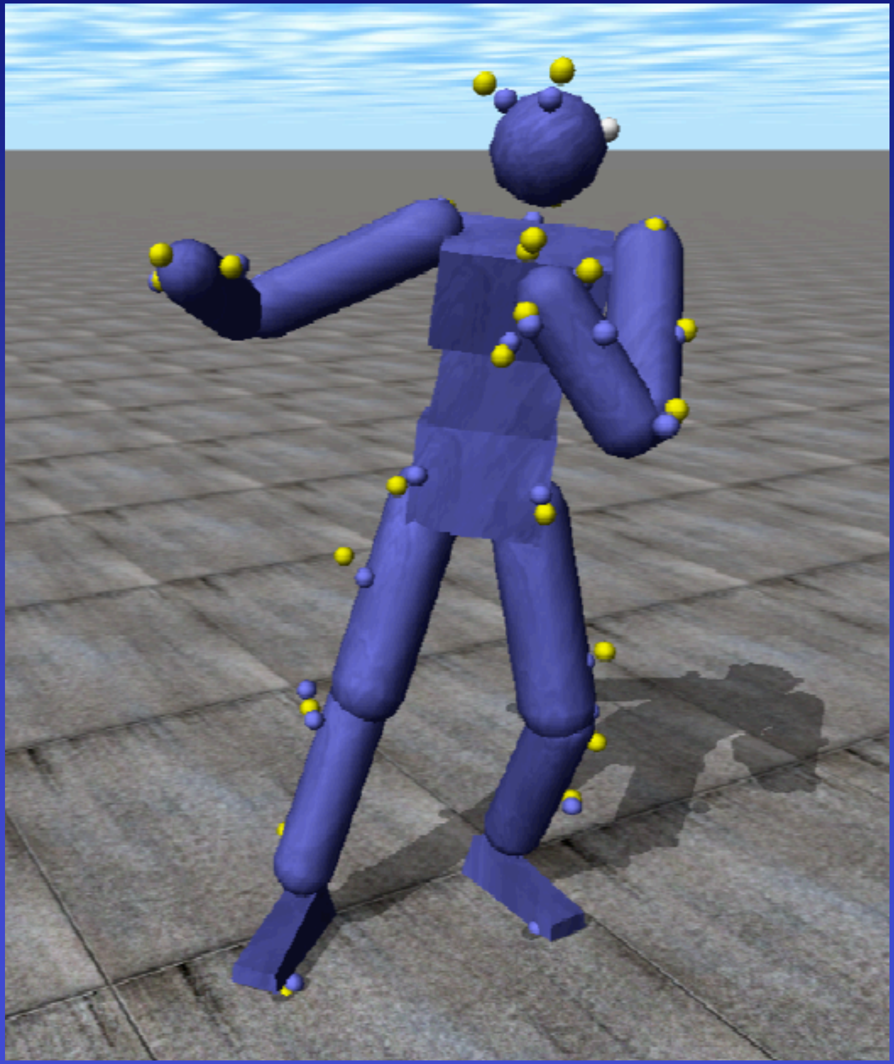
Ude, Mann, Riley, Atkeson ’00
Pollard, Hodgins, Riley, Atkeson ’02

Kovar, Schreiner, Gleicher '02

Physics and motion capture

Rose, Guenter, Bodenheimer, Rose ’96
Popovic & Witkin ’99
Pollard ’99, Pollard & Behmaram-Mosavat ’00
Zordan & Hodgins ’02

Approach overview

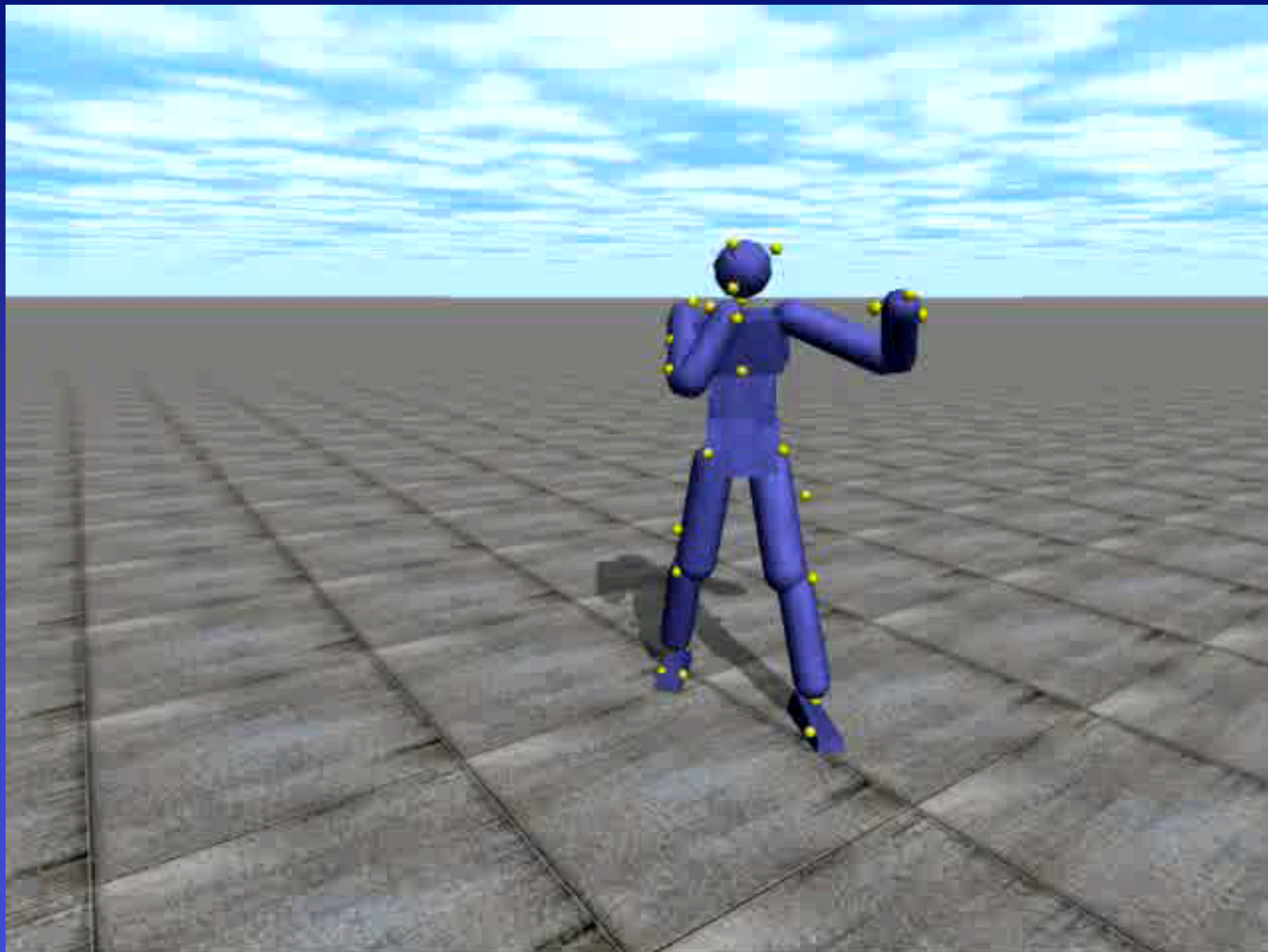


Simulation is used offline to compute postures

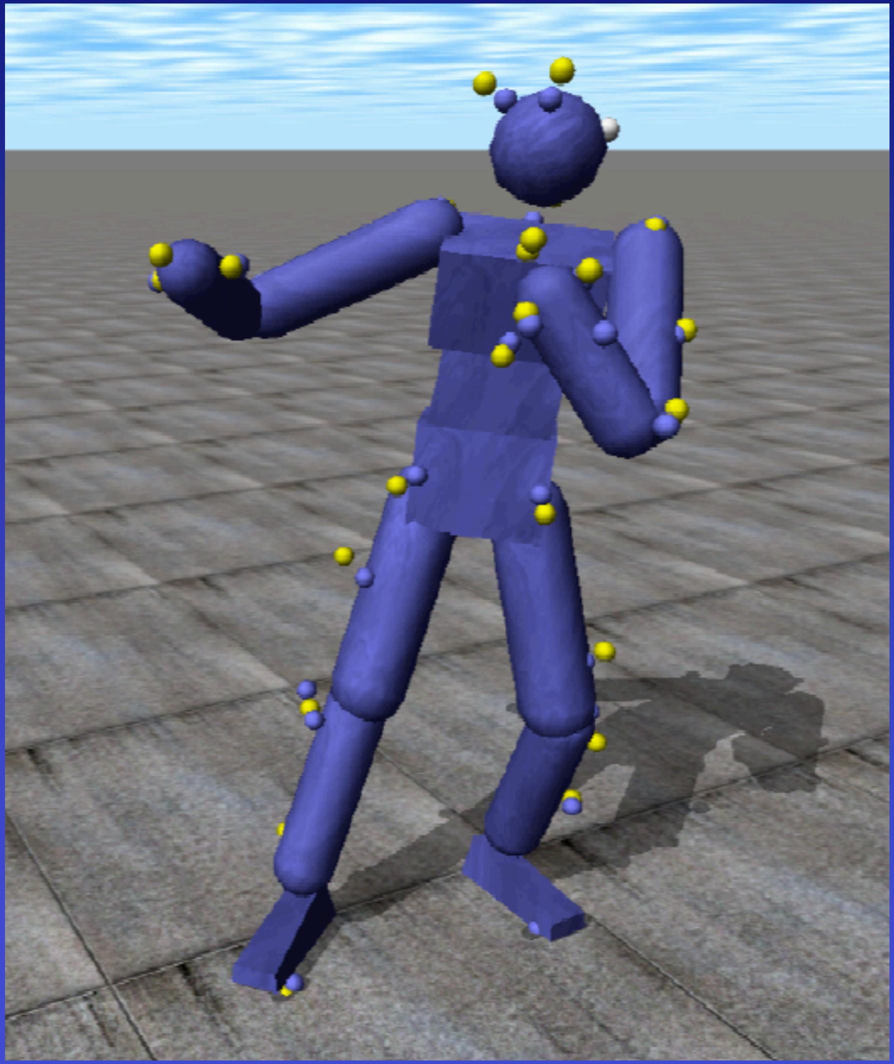
Internal torque actuators allow the simulation to act as a flexible ragdoll

Force springs pull 'ragdoll' to reach the data, marker by marker

Contact (e.g. ground) may be added through force



Approach overview

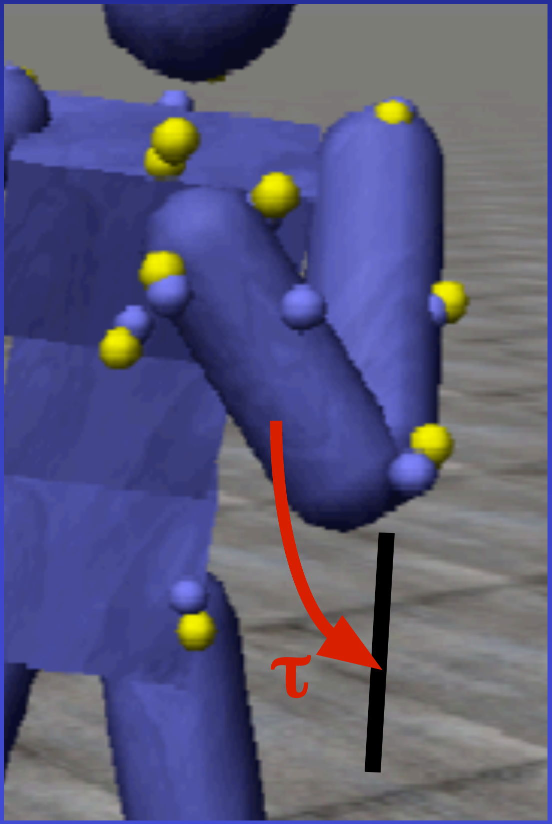


Basic Algorithm

```
foreach (data sample) {
    update [yellow] markers
    while (not still) {
        compute torques
        compute body forces
        if (active)
          compute contact forces
        update simulation
    }//while
    record posture
}//for
```

Internal torque control

PD-servo's control 3D ball joints at each articulation point to resist bending



$$\tau = k(\theta_d - \theta) - b(\dot{\theta})$$

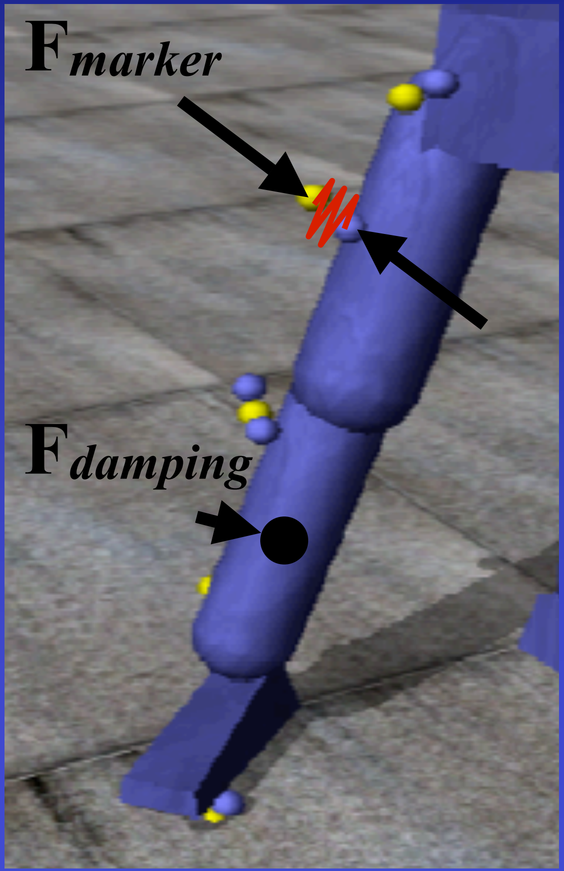
θ_d from rest position

k and b are stiffness and damping, inertial scaled (Zordan & Hodgins '02)

No joint limits

Additional body forces

Force-driven virtual 'landmarks' placed by hand guide the simulated bodies to follow the markers



Springs pull the simulation to the marker data

$$F_{marker} = -k_f X_{error}$$

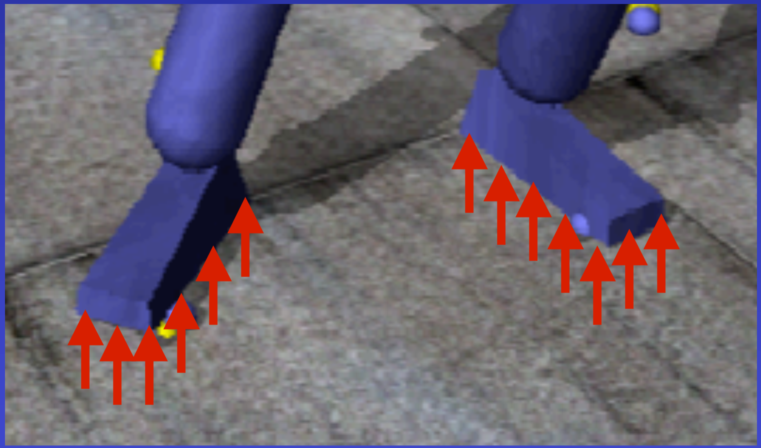
Body motion is damped

$$F_{damping} = -b_f V_{body}$$

Note, markers near joints affect both nearby bodies

Additional constraint forces

Avoiding foot/ground penetration and foot skate



Normal ground forces
flatten the foot on ground
via a penalty method

Marker data is used to
tag when each foot is
sliding or not

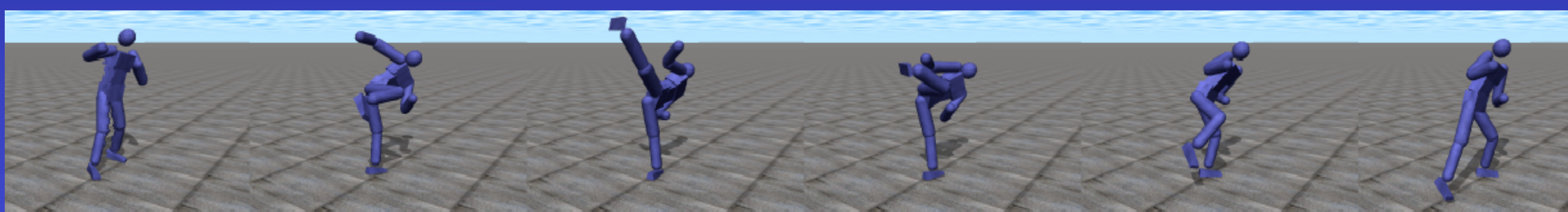
Horizontal friction forces
(not shown) resist in
opposite direction of
the simulated point
velocity when in slip

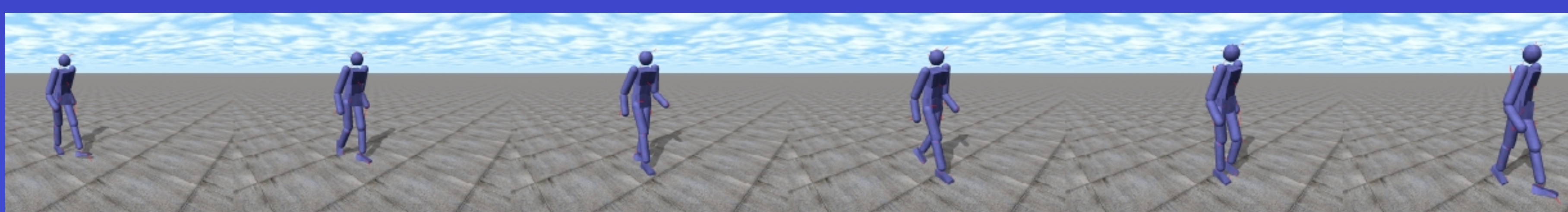
Implementation details & examples

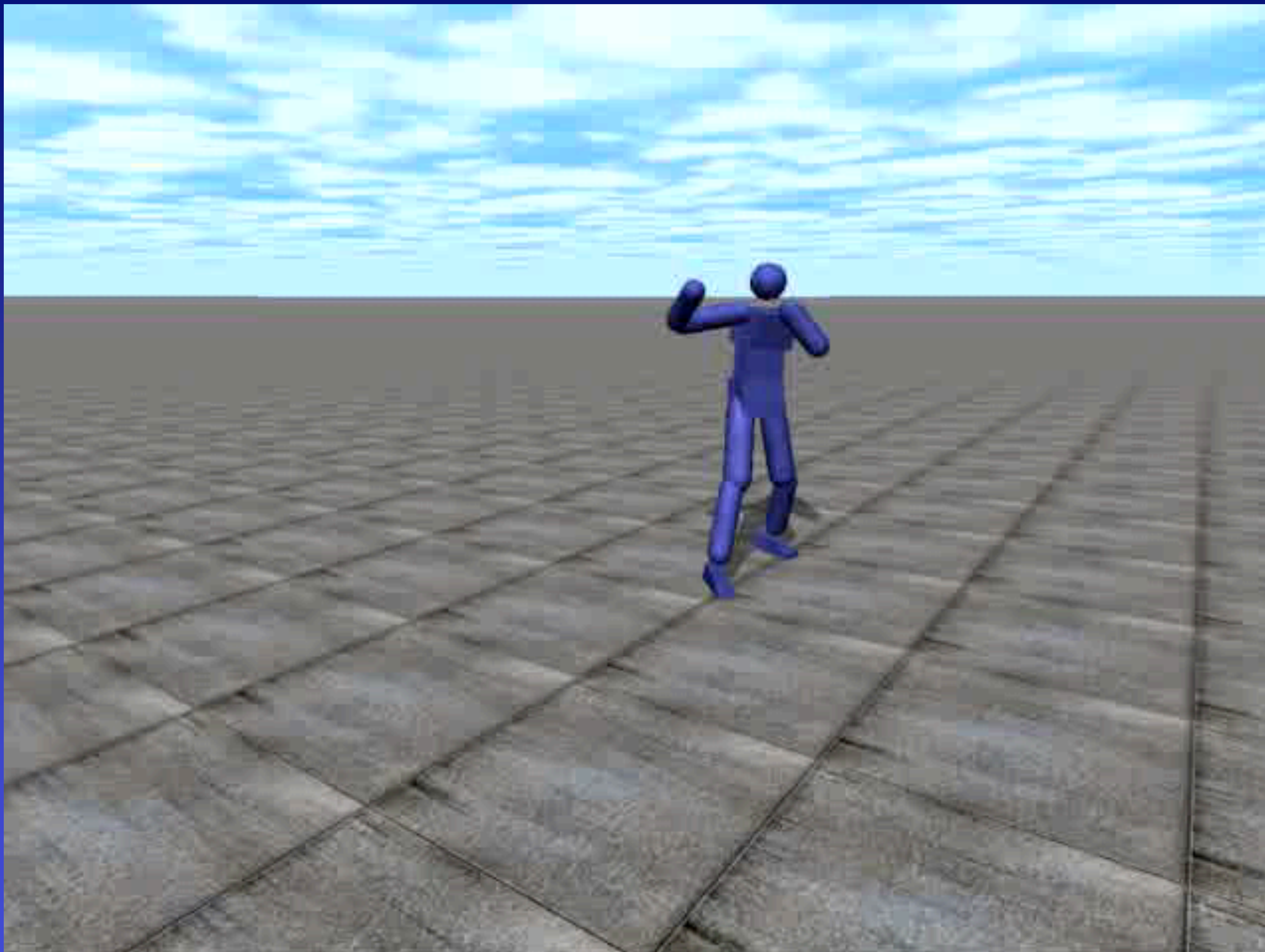
39 Degrees of freedom - simulated in ODE (free!)

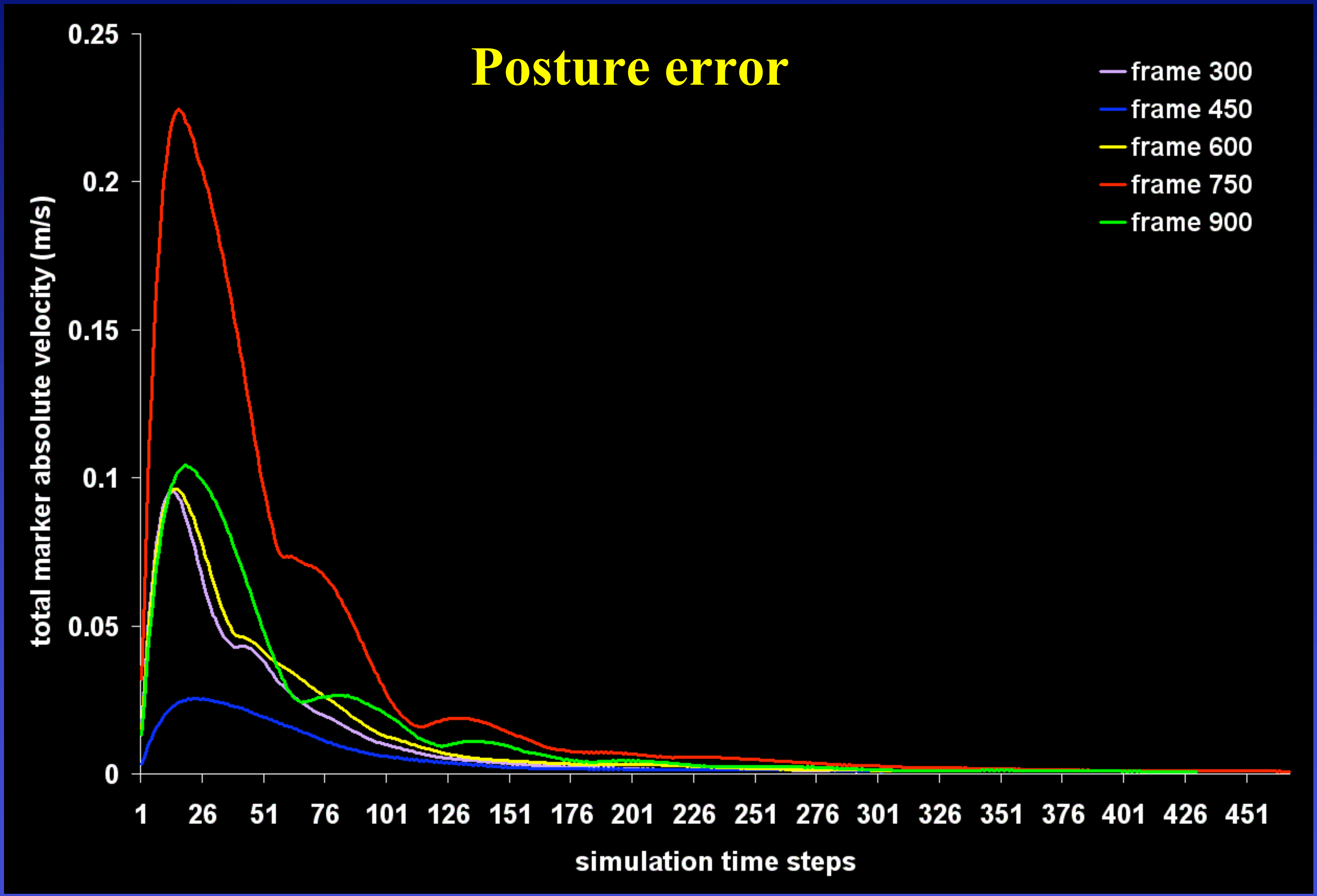
Runs about 2-3 frames/sec on 2.4 GHz Pentium IV

4 tuned parameters - torque stiffness & damping
marker spring stiffness
body force damping
(plus, ground contact model)

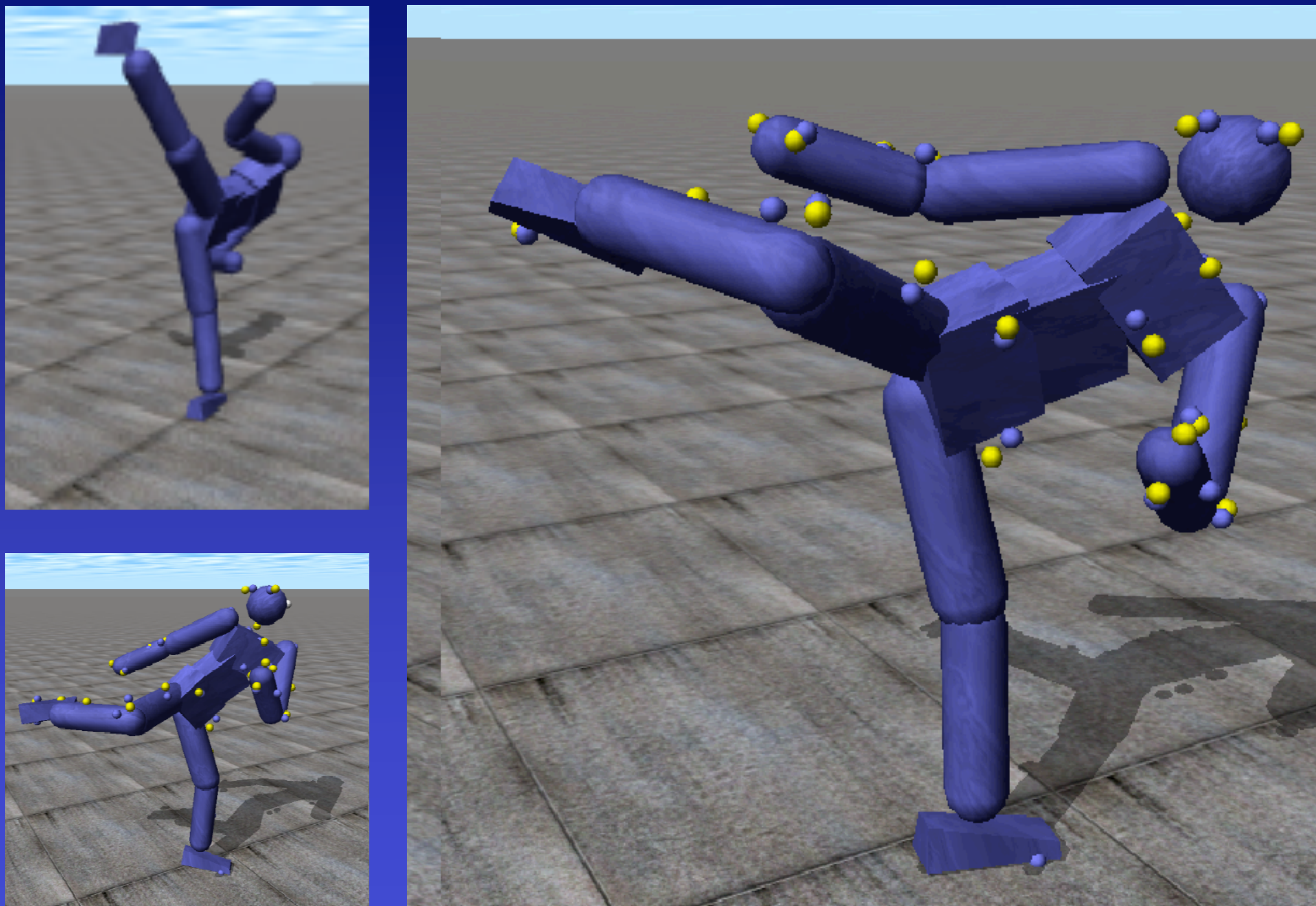


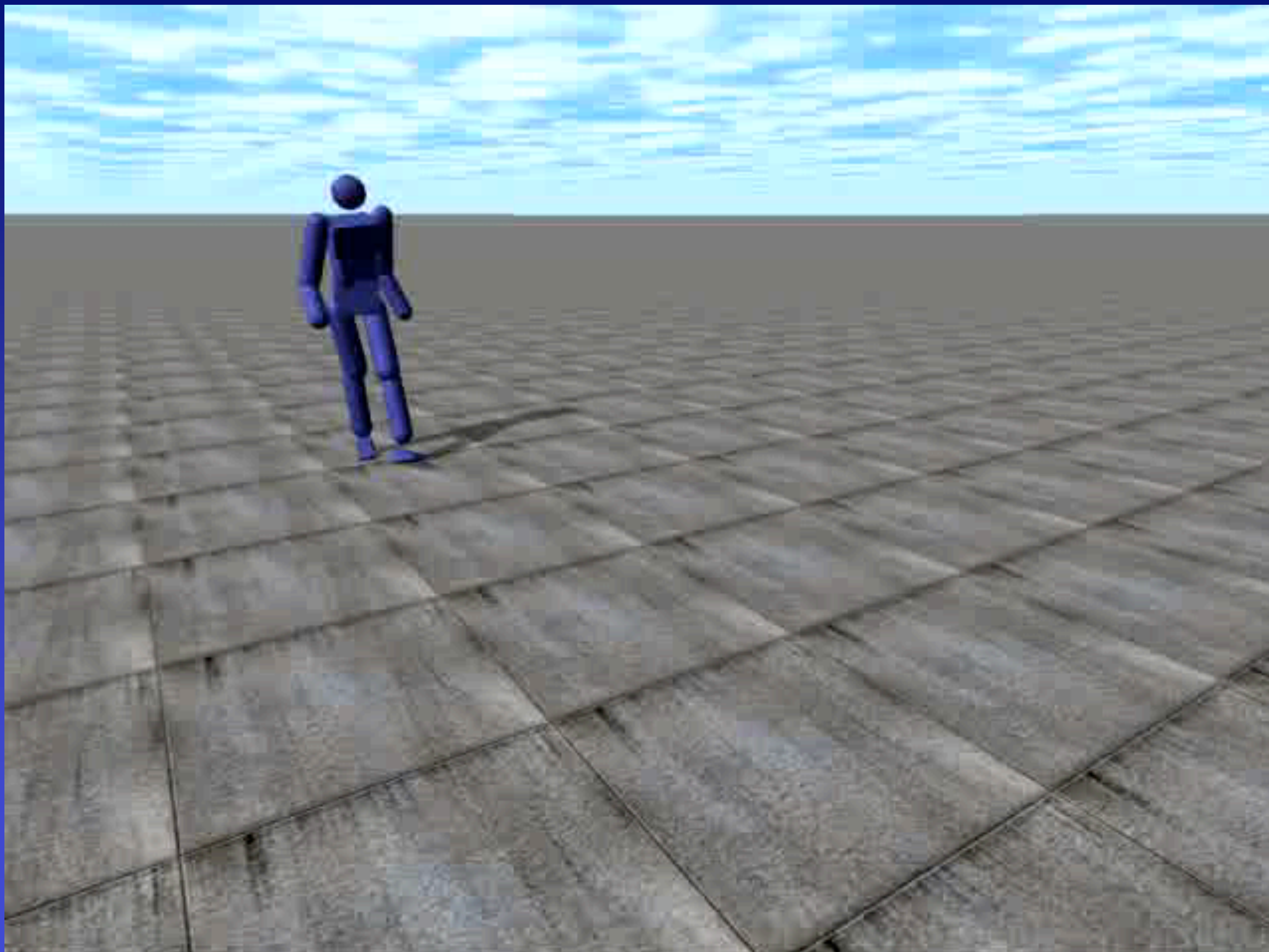


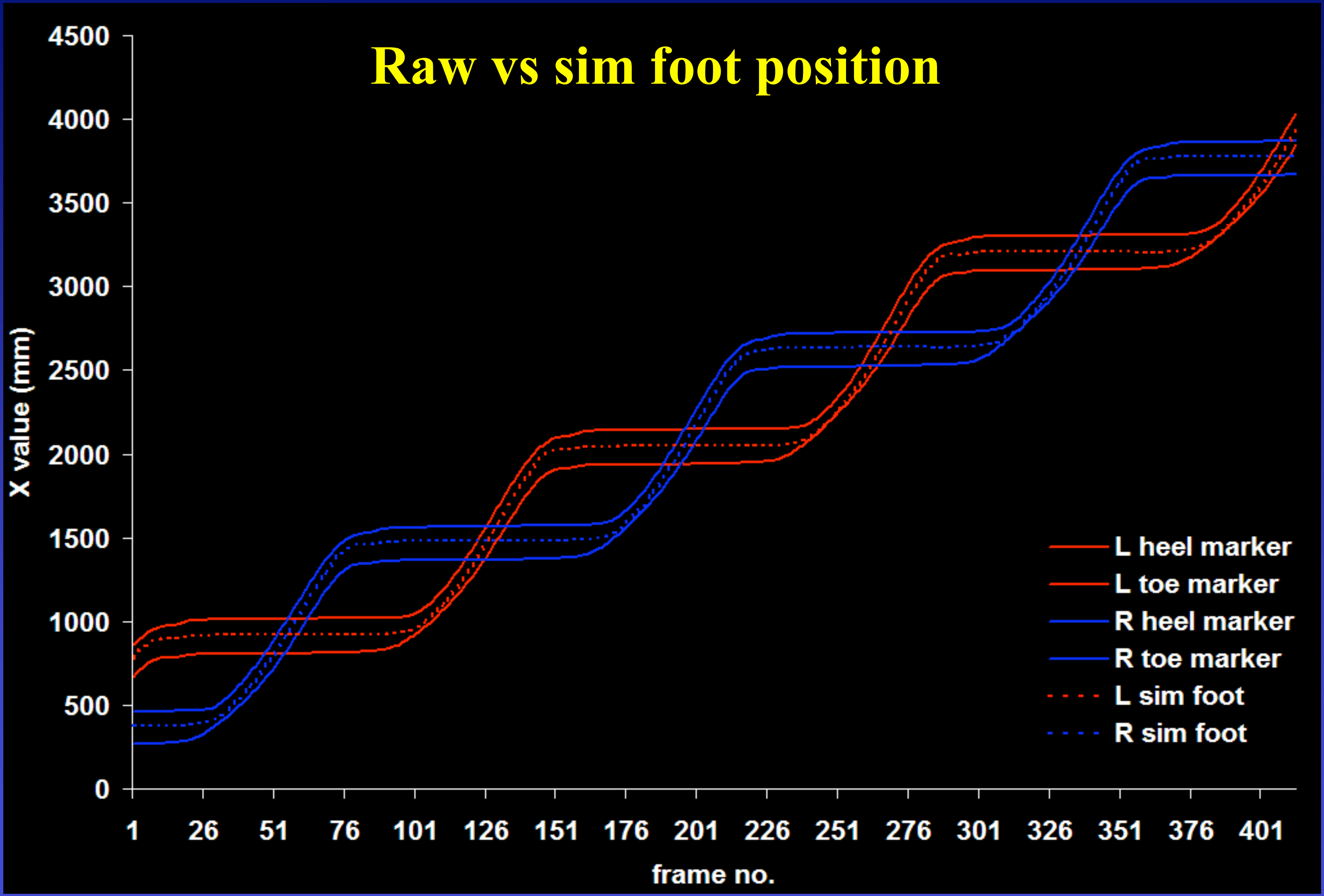




Posture error
frame 300
frame 450
frame 600
frame 750
frame 900
total marker absolute velocity (m/s)
0.25
0.2
0.15
0.1
0.05
0
1
26
51
76
101
126
151
176
201
226
251
276
301
326
351
376
401
426
451
simulation time steps






Raw vs sim foot position
X value (mm)
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1
26
51
76
101
126
151
176
201
226
251
276
301
326
351
376
401
frame no.
L heel marker
L toe marker
R heel marker
R toe marker
L sim foot
R sim foot

Conclusion/future work

Simple, easy to implement, and inexpensive

Would dovetail nicely with a skeleton estimator

Likely requires a two-pass process for motion
severe character retargeting

Would benefit from a specialized marker set
(markers spread over body parts with
highly repeatable landmarks, for example)

Should run interactively, to be used during the
live motion capturee shoot

www.cs.ucr.edu/~rgl

Thank you!