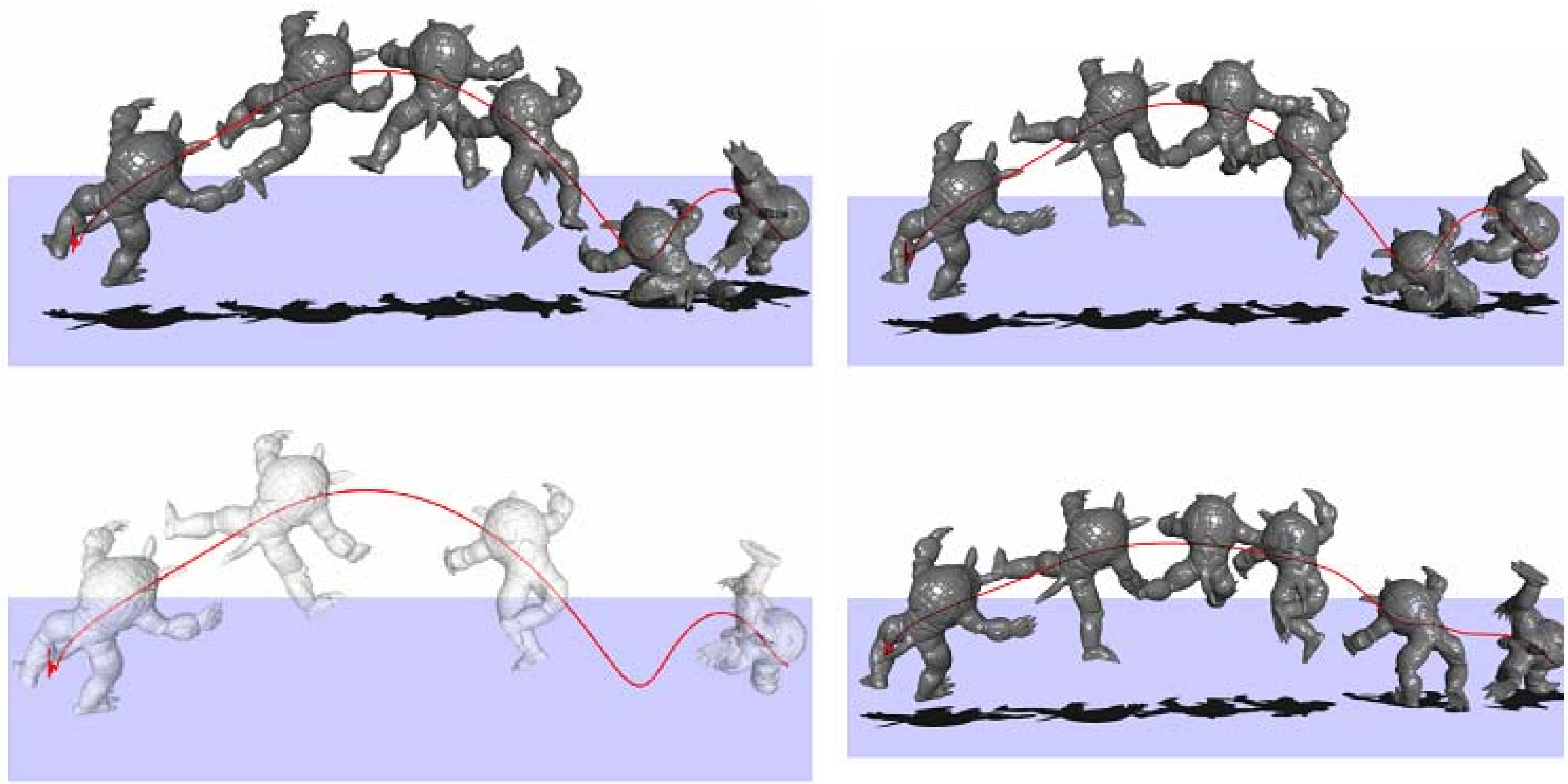
Directable animation of elastic objects

Ryo Kondo, Takashi Kanai, Ken-ichi Anjyo SIGGRAPH 2005



Outline

- Introduction
- Directable animation framework
- Physically-based elastic body animation
- Deformation control
- Trajectory control
- Results
- Discussion

Introduction

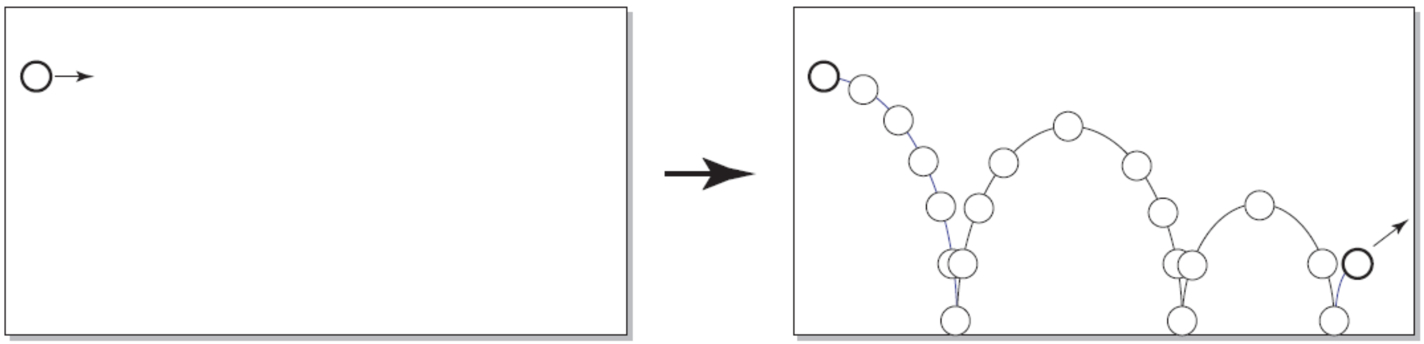
- How to create animations?
  - Keyframe control as the most intuitive method (intentional).
  - Physical simulation has also become widely used (obeys physical laws).
- Goal is to achieve both physics-based realism and user-specified expressive motion.
- Recent research:
  - Keyframing of smoke simulation.
  - Trajectory control of rigid body simulation.

Directable animation framework

- To construct plausible motions for elastic objects we want:
  1. Physical realism.
  2. Edit the local geometry of an object at a given time as the user desires.
  3. Edit the trajectory of an object as the user desires.

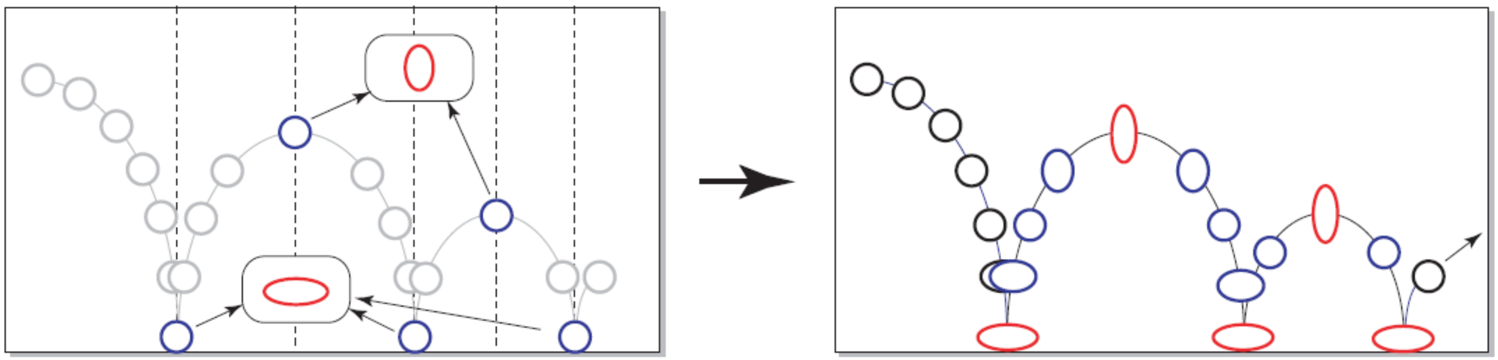
- We need:
  1. Physically-based elastic body animation.
  2. Deformation control.
  3. Trajectory control.

Physically-based elastic body animation



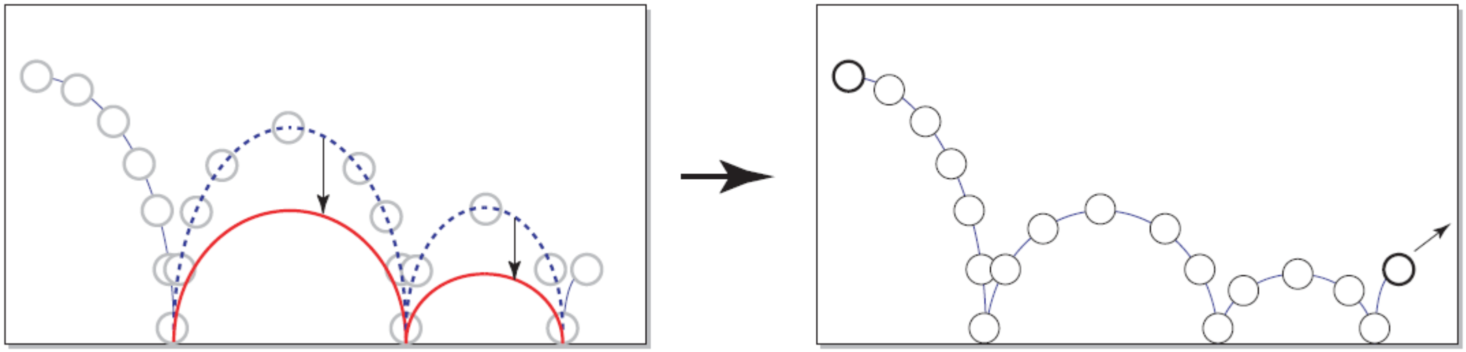
- Simulation with the finite element method.
- Position and velocity are recorded at each timestep.

Deformation control of objects



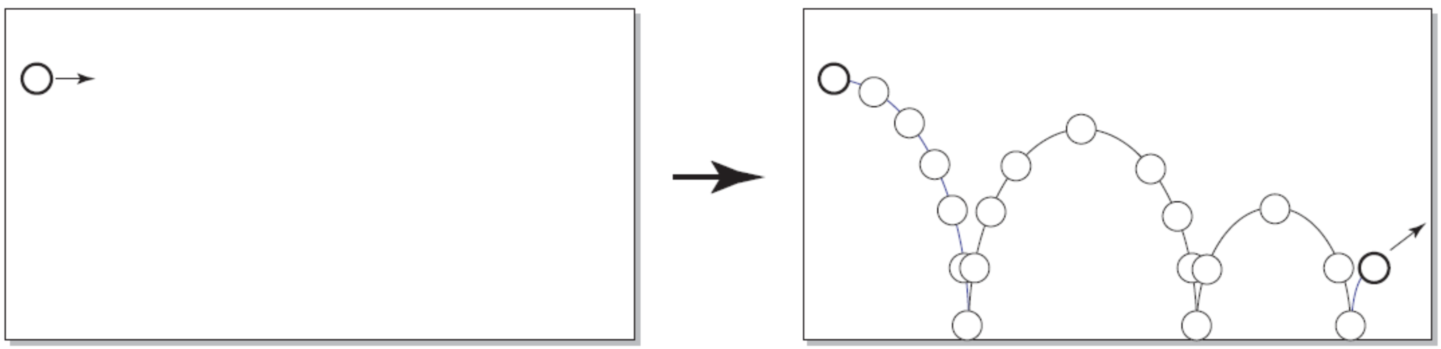
- User can set a keyframe for the shape of an object at a given time.
- User can modify the shape.
- Recalculate motion according to the shapes of the keyframes.

Trajectory control of objects



- User can edit the trajectory of the object (position, velocity and rotation).
- Rearrange animation according to modified trajectory.

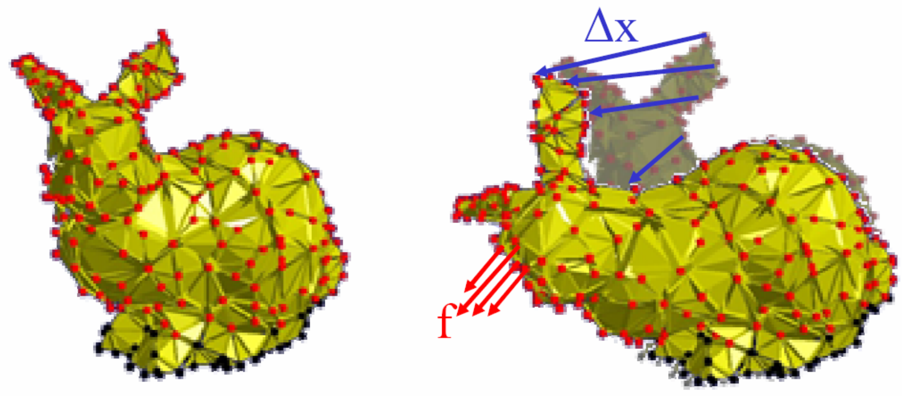
Physically-based elastic body animation



The finite element method

- Elastic forces:

  $\mathbf{f}_{el} = \mathbf{K} \cdot \Delta\mathbf{x}$ (Stiffness Matix $\mathbf{K} \in \mathbb{R}^{3n \times 3n}$)



- Dynamic Deformation:

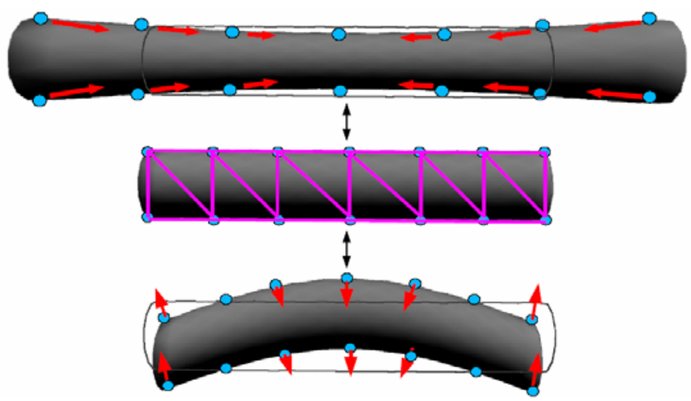
  $$\mathbf{M}\mathbf{x}'' + \mathbf{C}\mathbf{x}' + \mathbf{K}\Delta\mathbf{x} = \mathbf{f}_{ext}$$

- Notation we use: $M\ddot{x} + C\dot{x} + K(x - o) = f_{ext}$

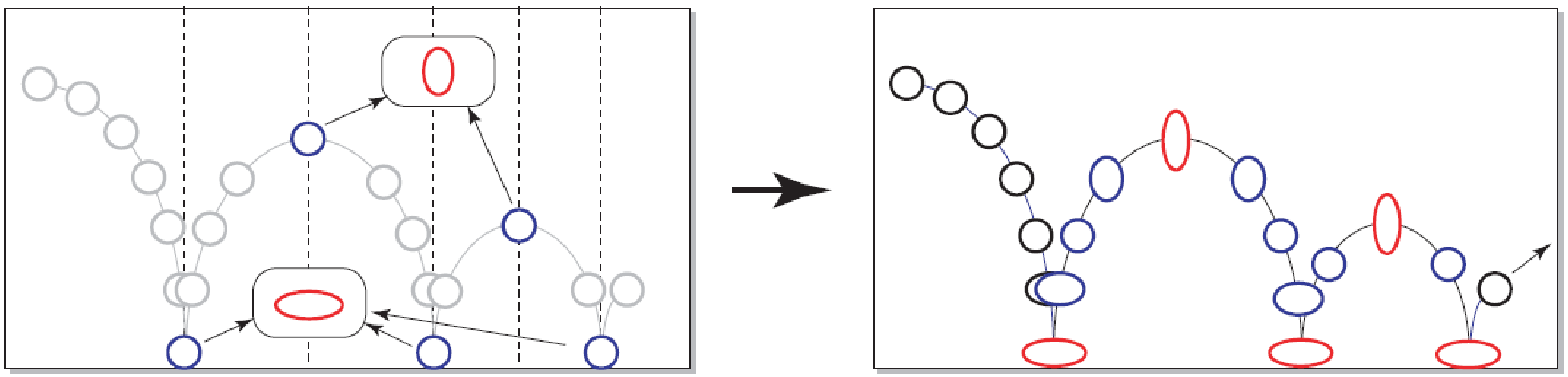
The finite element method

- Important:
  - Stiffness matrix *K* and original position *o* define the resting shape of an elastic object.

$$M\ddot{x} + C\dot{x} + K(x - o) = f_{ext}$$

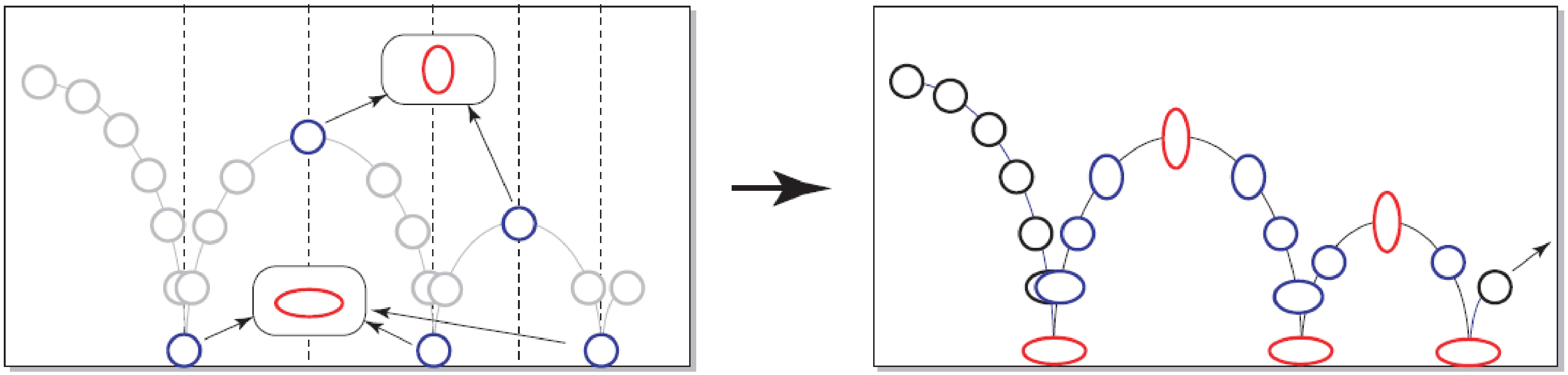


Deformation control of objects



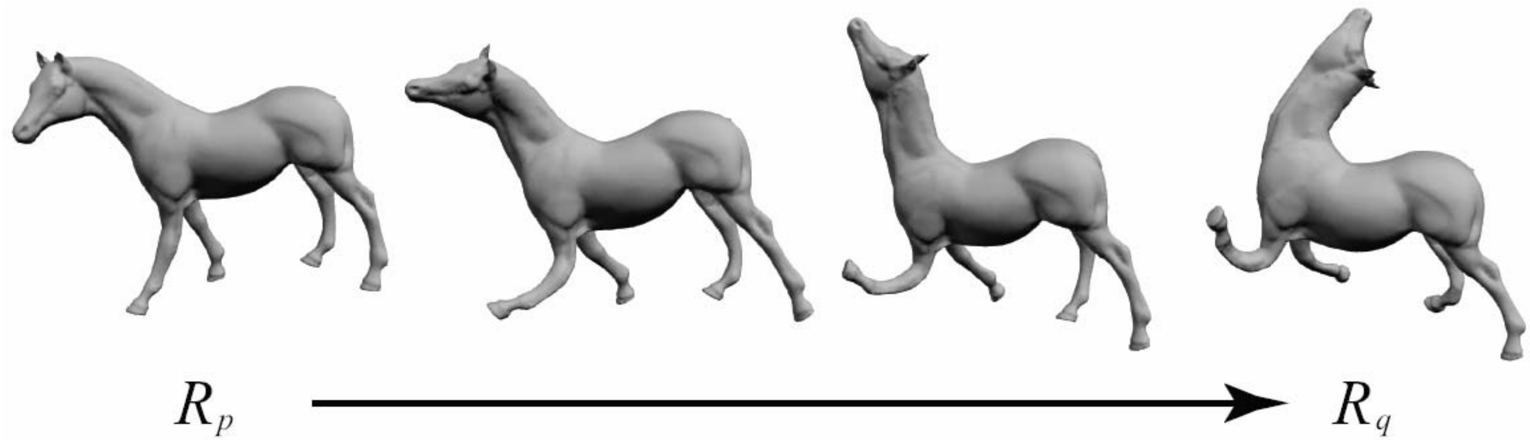
- User-defined set of keyframe shapes.
- Idea: Replace resting shape of the elastic object at a keyframe by the user-defined keyframe shape.

Physics-oriented interpolation



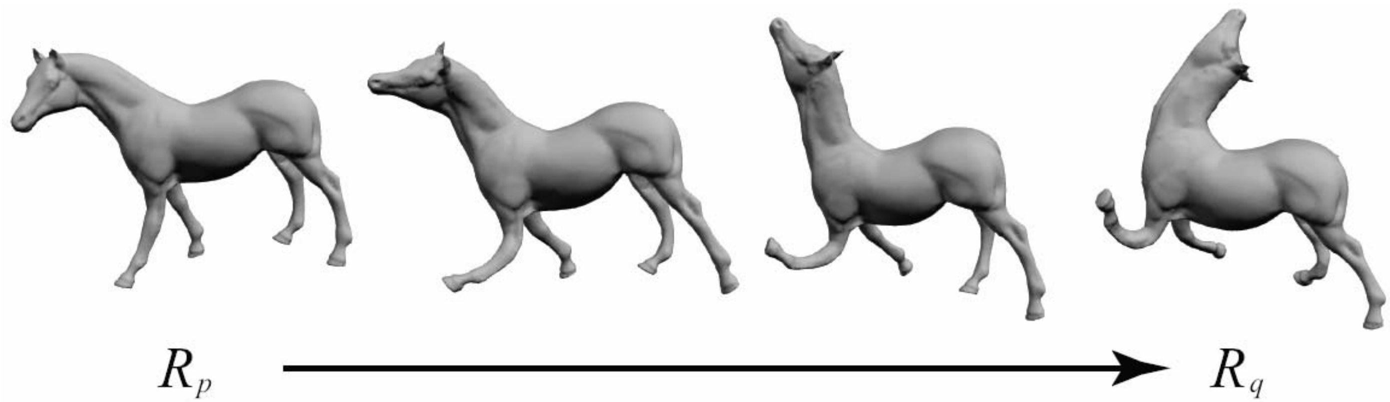
- Problem: Displacements between neighbor keyframe shapes are large.
- Solution: Continuously replace resting shape of the elastic object between keyframes.

Physics-oriented interpolation



- Interpolation function $R_{pq}(t)$ which interpolates two neighbor resting shapes R_p and R_q.

Physics-oriented interpolation

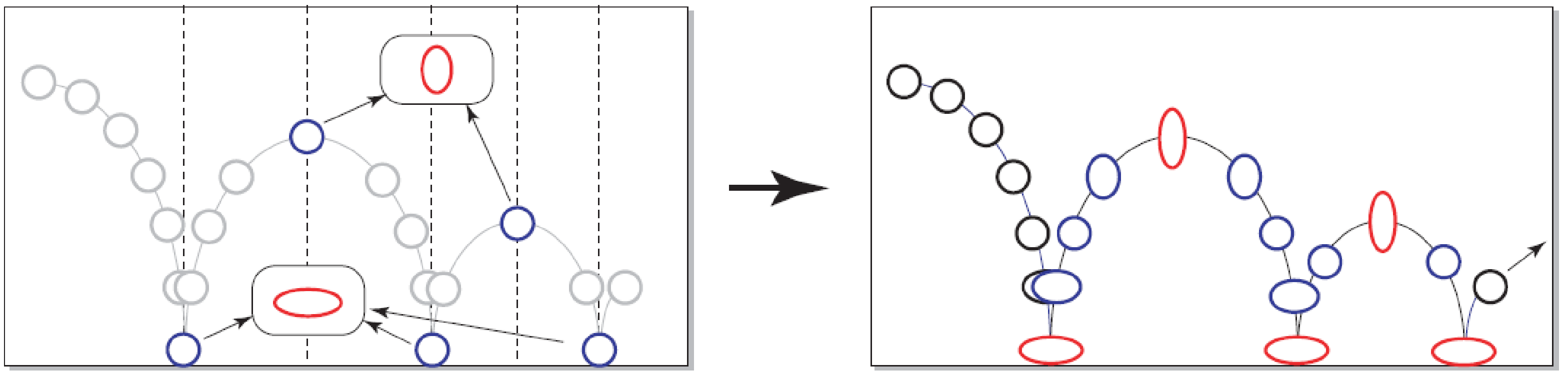


- $R_{pq}(t)$ is found by solving the differential equation from R_p as initial state with K_q, o_q derived from R_q.

$$M\ddot{x} + C\dot{x} + K_q(x - o_q) = 0$$

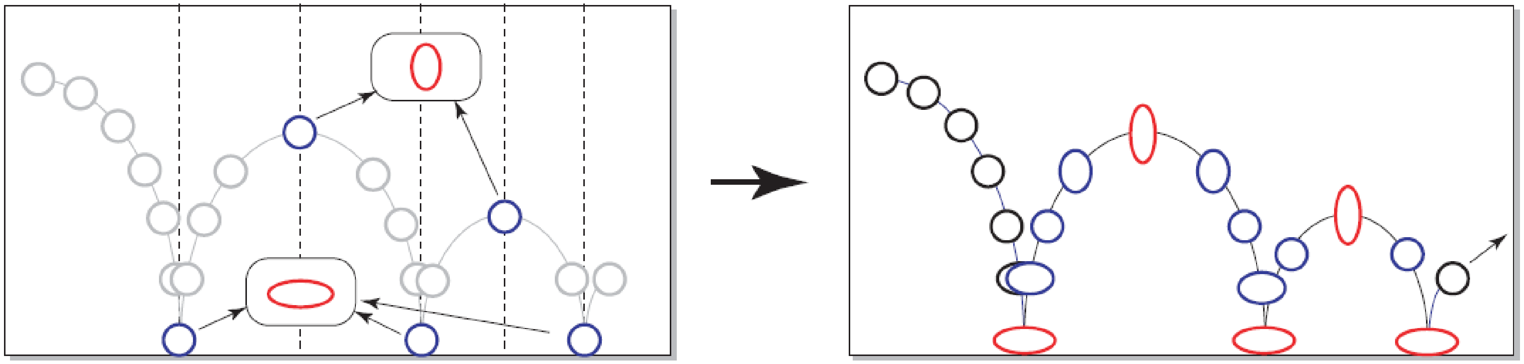
- Restrain restoring forces by extreme damping.

Physics-oriented interpolation



- From our resting shape interpolations we derive the time-varying stiffness matrix *K(t)* and original position *o(t)*.

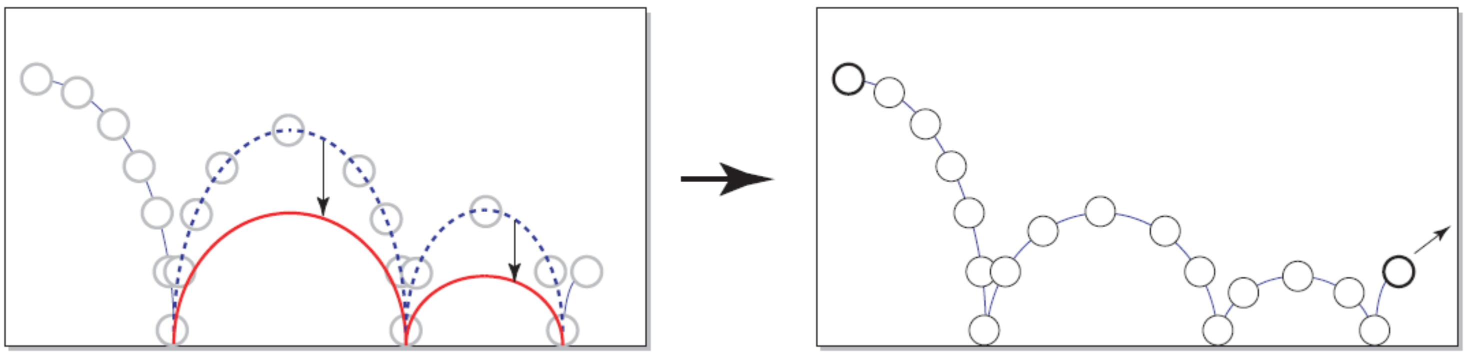
Physics-oriented interpolation



- The final animation is computed with:

$$M\ddot{x} + C\dot{x} + K(t)(x - o(t)) = f_{ext}$$

Trajectory control of objects



Compensation for positions and velocities

- Center of mass position and velocity:

$$v_m(t) = \frac{1}{m}\sum_i (v_i(t)\cdot m_i), \qquad x_m(t) = \frac{1}{m}\sum_i (x_i(t)\cdot m_i),$$

$$m = \sum_i m_i.$$

- Position and velocity given by trajectory: $x'(t), v'(t)$
- Differencies:

$$\Delta v(t) = v_m(t) - v'(t), \qquad \Delta x(t) = x_m(t) - x'(t)$$

Compensation for rotations

- The global rotation matrix is defined as:

$$R(t) \;=\; ortho(\frac{1}{m}\sum_i (R_i(t)\cdot m_i)), \quad m = \sum_i m_i$$

- $R_i(t)$, m_i are the rotation matrix and mass of the tetrahedral element *i*.
- For the given trajectory rotation $R'(t)$ the difference rotation matrix is:

$$\Delta R(t) = R'(t)R(t)^T.$$

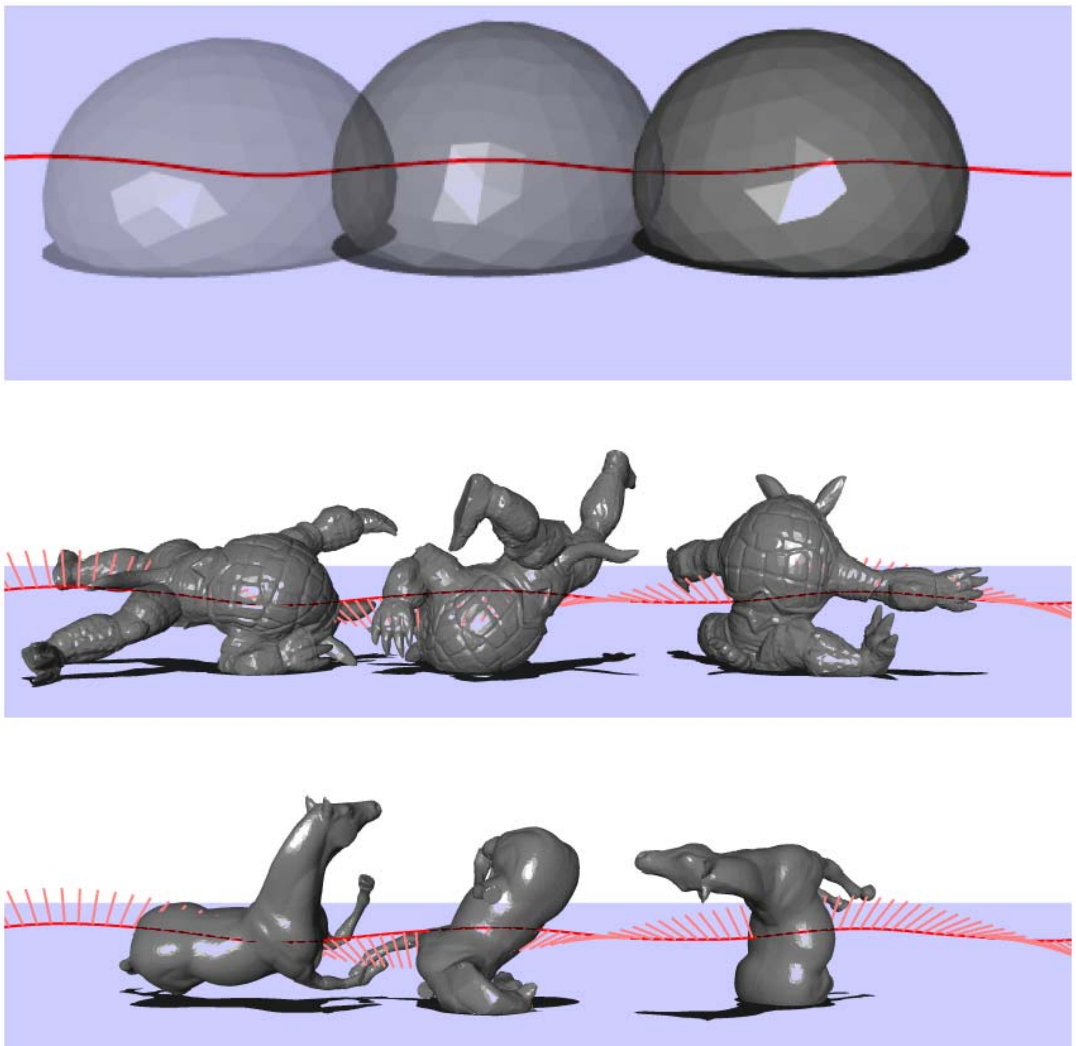
Trajectory control

- While recomputing animation. For each simulation point, at each time step. Correct positions and velocities:

$$\tilde{v}(t_i) = v_m(t_i) + \Delta R(t_i)\big(v(t_i) + \Delta v(t_i) - v_m(t_i)\big)$$
$$\tilde{x}(t_i) = x_m(t_i) + \Delta R(t_i)\big(x(t_i) + \Delta x(t_i) - x_m(t_i)\big)$$

- Are used to compute $x(t_{i+1})$, $v(t_{i+1})$.

Trajectory control



Results

- Video

Future work

- Prototype provides only simple interface to modify shape. Commercial modeling system for more precise deformation control.
- More automatic functions required in keyframing. E.g. adding keyframes before and after collisions.
- Dealing with many deformable objects at the same time.

Advantages

- General method for deformable solids.
- No need for detailed muscle or skeleton structure of the objects.
- Intuitive control.
- Easy to implement.
- Possible application in real-time interactive animation.

Limits

- Not “accurate”. Resting shapes are only guides. Trajectory is only a constraint.
- Keyframe shape delay.
- Keyframes should be set relatively far apart.
- Limited when changing topology. E.g. fracturing.
- Condition of $K(t)$ could become bad.

Discussion

???