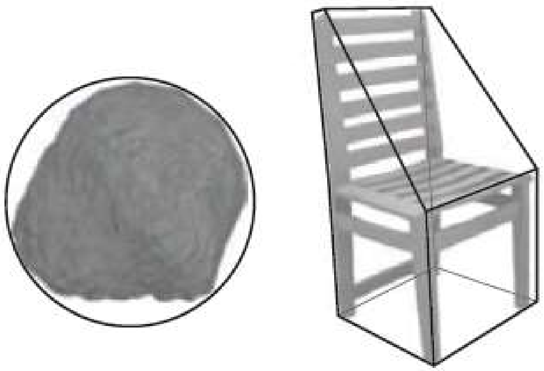
Animation and Simulation

He Wang (王鹤)

Collisions and Body Dynamics

- Collision Detection
  - Contact, interpenetration
  - Many types (support, sliding, etc.)
  - Collidable entities
    - Collision representation vs visual representation
    - Favour simple geometries (spheres, boxes, capsules, convex hulls, etc.)

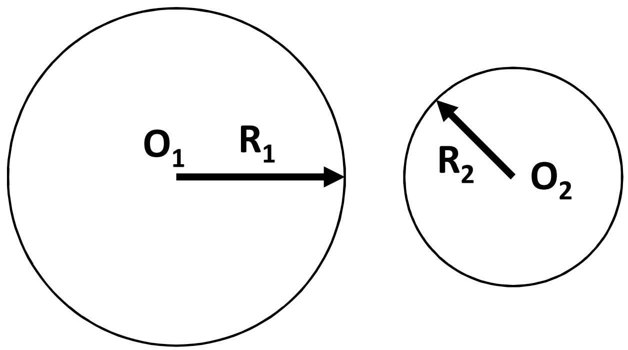


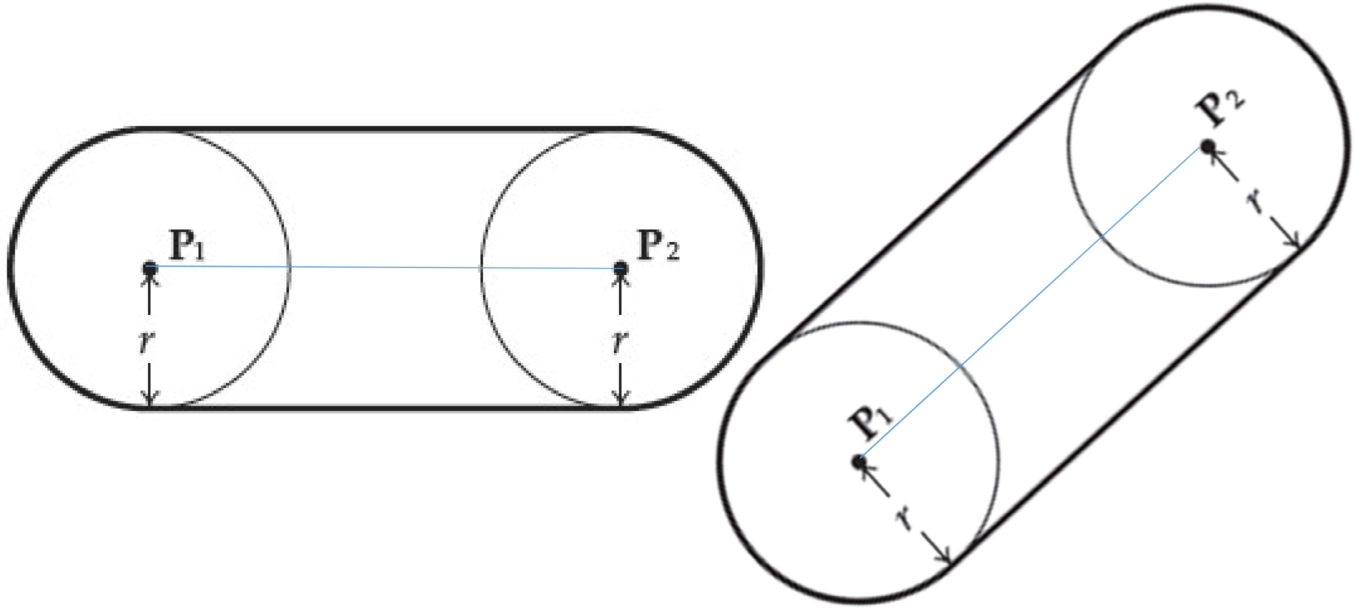
Collisions and Body Dynamics

- Collision Detection
  - Collidable entities
    - Collision representation vs visual representation
    - Favour simple geometries (spheres, boxes, capsules, convex hulls, etc.)
      - Roughly define the form of the object
      - Transformed with the object
      - Can be complex
  - Collision world
    - Group and maintain all collidable objects
    - Usually separate from simulation engine

Collisions and Body Dynamics

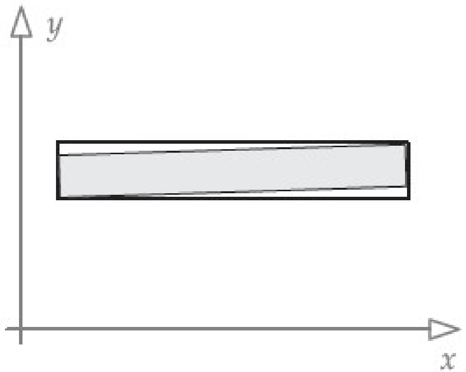
- Collision Detection
  - Collision Primitives
    - Spheres
    - Capsules

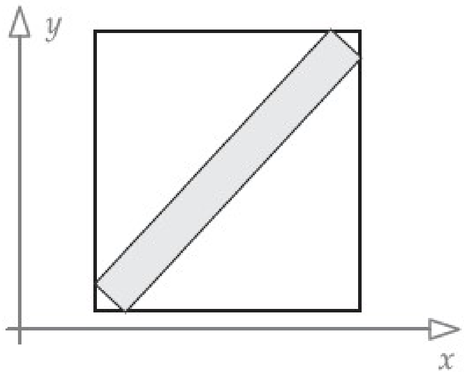


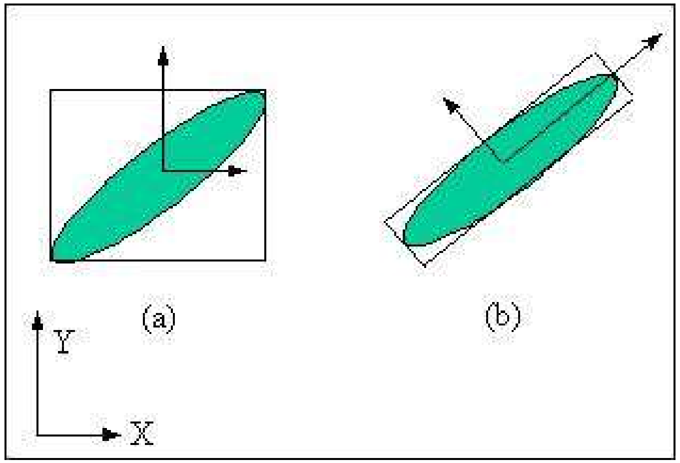


Collisions and Body Dynamics

- Collision Detection
  - Collision Primitives
    - Spheres
    - Capsules
    - Axis-aligned Bounding Boxes (AABBs)
    - Oriented Bounding Boxes (OBBs)

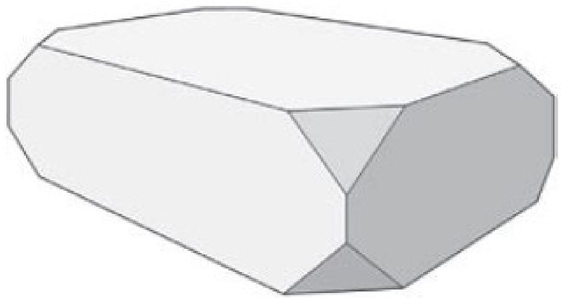






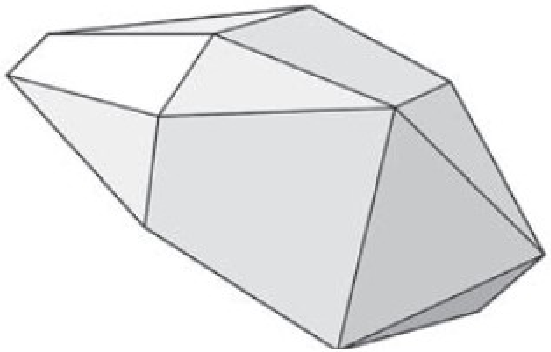
Collisions and Body Dynamics

- Collision Detection
  - Collision Primitives
    - Spheres
    - Capsules
    - Axis-aligned Bounding Boxes (AABBs)
    - Oriented Bounding Boxes (OBBs)
    - Discrete Oriented Polytope (DOP)



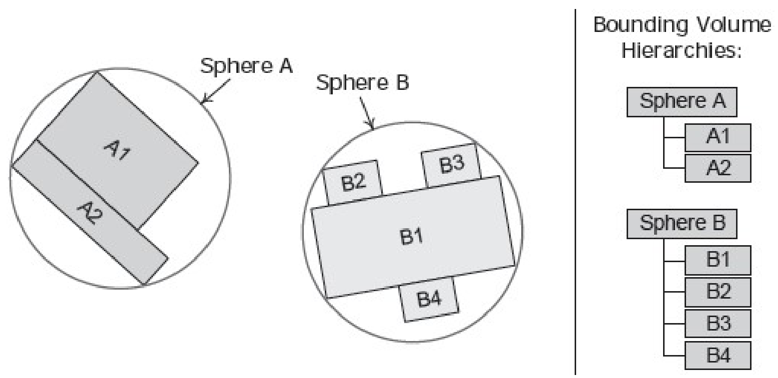
Collisions and Body Dynamics

- Collision Detection
  - Collision Primitives
    - Spheres
    - Capsules
    - Axis-aligned Bounding Boxes (AABBs)
    - Oriented Bounding Boxes (OBBs)
    - Discrete Oriented Polytope (DOP)
    - Arbitrary Convex Volumes



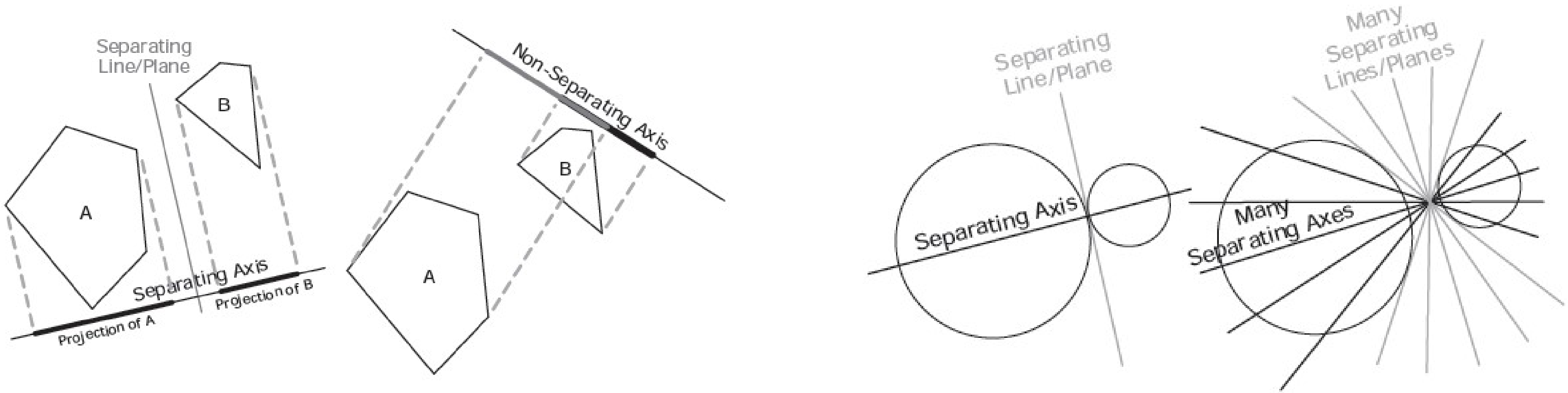
Collisions and Body Dynamics

- Collision Detection
  - Collision Primitives
    - Spheres
    - Capsules
    - Axis-aligned Bounding Boxes (AABBs)
    - Oriented Bounding Boxes (OBBs)
    - Discrete Oriented Polytope (DOP)
    - Arbitrary Convex Volumes
    - Compound shapes



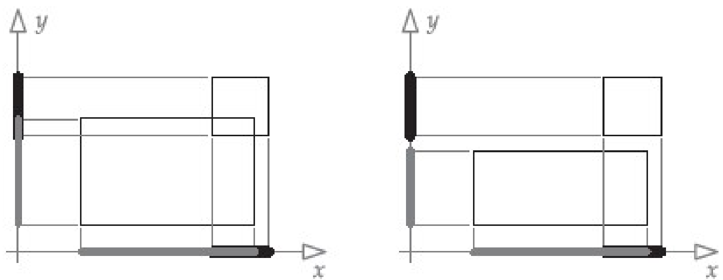
Collisions and Body Dynamics

- Collision Detection
  - Collision Testing
    - Point-sphere
    - Sphere-sphere
    - Separating Axis Theorem
      - If an axis can be found where projections are **convex** shapes do not overlap



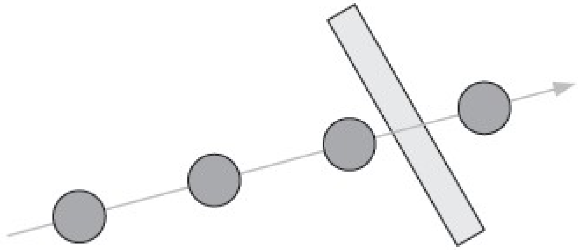
Collisions and Body Dynamics

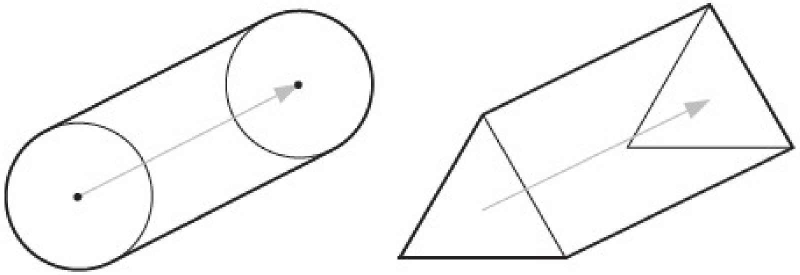
- Collision Detection
  - Collision Testing
    - Point-sphere
    - Sphere-sphere
    - Separating Axis Theorem
    - AABB-AABB

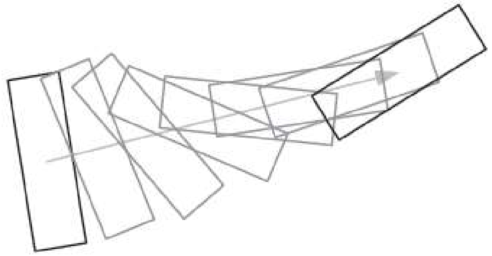


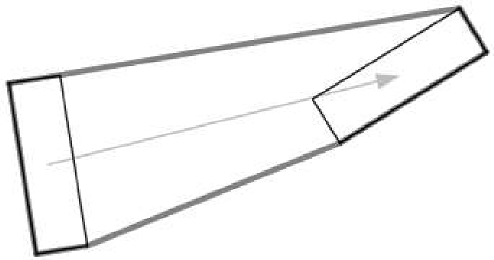
Collisions and Body Dynamics

- Collision Detection
  - Collision Testing
    - Moving objects (swept volume)



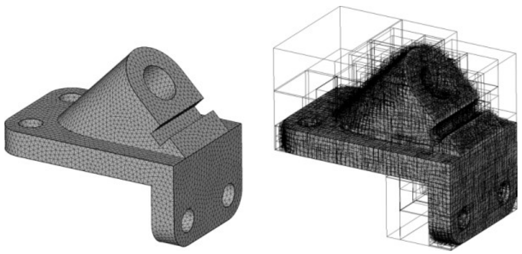


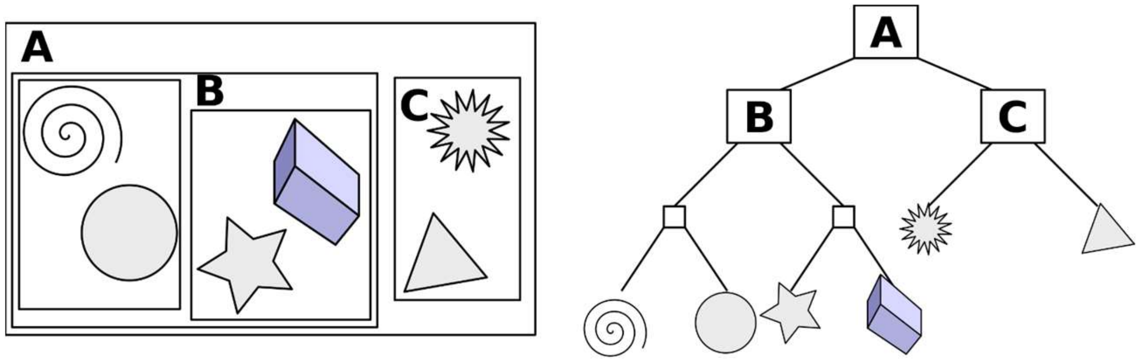




Collisions and Body Dynamics

- Collision Detection
  - Performance Optimisation
    - Non-trivial to determine for two objects
    - Large number of objects
    - Temporal coherency (reuse previous information)
    - Spatial Partitioning (BSP, Kd-tree, sphere-tree, etc.)
    - Phasing (Broad, mid and narrow)
      - First AABB
      - Course bounding volumes
      - Individual primitives





Collisions and Body Dynamics

- Rigid Body Dynamics
  - Collision Response
    - Basics

      $$E = V + T$$ *T* is Kinetic Energy and *V* is Potential Energy

      $$T_{\text{linear}} = \frac{1}{2}\mathbf{p} \cdot \mathbf{v}. \qquad T_{\text{angular}} = \frac{1}{2}\mathbf{L} \cdot \boldsymbol{\omega}.$$

    - Energy conservation

Collisions and Body Dynamics

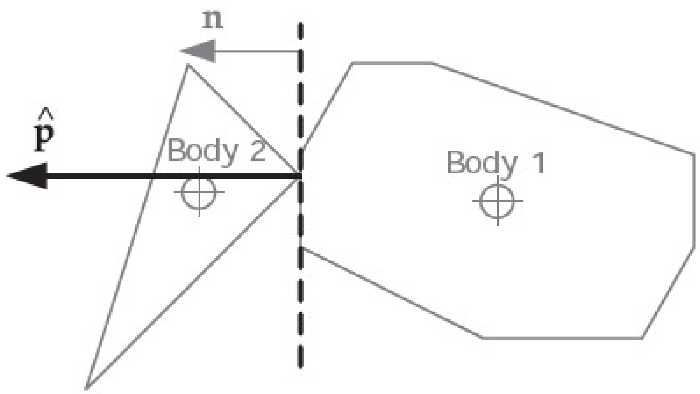
- Rigid Body Dynamics
  - Impulses
    - Collision in instantaneous moment
    - No friction
    - Can be approximated by a coefficient of restitution

$$\mathbf{p}_1 + \mathbf{p}_2 = \mathbf{p}_1' + \mathbf{p}_2', \qquad \text{or}$$
$$m_1 \mathbf{v}_1 + m_2 \mathbf{v}_2 = m_1 \mathbf{v}_1' + m_2 \mathbf{v}_2',$$

$$\longrightarrow \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_1 v_1'^2 + \frac{1}{2} m_2 v_2'^2 + T_{\text{lost}}$$

Collisions and Body Dynamics

- Rigid Body Dynamics
  - Impulses
    - Collision in instantaneous moment
    - No friction
    - Can be approximated by a coefficient of restitution



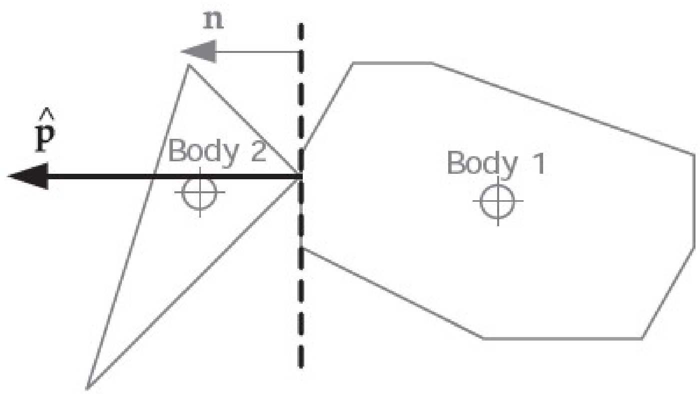
$$\mathbf{p}_1' = \mathbf{p}_1 + \hat{\mathbf{p}}; \qquad \mathbf{p}_2' = \mathbf{p}_2 - \hat{\mathbf{p}};$$

$$m_1\mathbf{v}_1' = m_1\mathbf{v}_1 + \hat{\mathbf{p}}; \qquad m_2\mathbf{v}_2' = m_2\mathbf{v}_2 - \hat{\mathbf{p}};$$

$$\mathbf{v}_1' = \mathbf{v}_1 + \frac{\hat{p}}{m_1}\mathbf{n}; \qquad \mathbf{v}_2' = \mathbf{v}_2 - \frac{\hat{p}}{m_2}\mathbf{n}.$$

Collisions and Body Dynamics

- Rigid Body Dynamics
  - Impulses



$$\mathbf{p}_1' = \mathbf{p}_1 + \hat{\mathbf{p}}; \qquad \mathbf{p}_2' = \mathbf{p}_2 - \hat{\mathbf{p}};$$

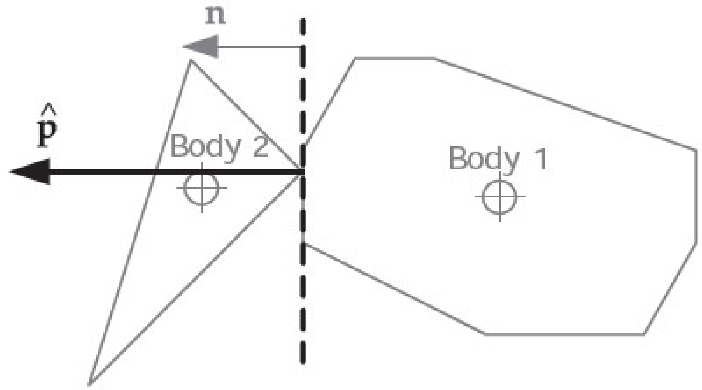
$$m_1\mathbf{v}_1' = m_1\mathbf{v}_1 + \hat{\mathbf{p}}; \qquad m_2\mathbf{v}_2' = m_2\mathbf{v}_2 - \hat{\mathbf{p}};$$

$$\mathbf{v}_1' = \mathbf{v}_1 + \frac{\hat{p}}{m_1}\mathbf{n}; \qquad \mathbf{v}_2' = \mathbf{v}_2 - \frac{\hat{p}}{m_2}\mathbf{n}. \qquad (\mathbf{v}_2' - \mathbf{v}_1') = \varepsilon(\mathbf{v}_2 - \mathbf{v}_1).$$

$$\hat{\mathbf{p}} = \hat{p}\,\mathbf{n} = \frac{(\varepsilon + 1)(\mathbf{v}_2 \cdot \mathbf{n} - \mathbf{v}_1 \cdot \mathbf{n})}{\dfrac{1}{m_1} + \dfrac{1}{m_2}}\,\mathbf{n}.$$

Collisions and Body Dynamics

- Rigid Body Dynamics
  - Impulses



What if hit a wall? Coefficient = 1 for body 2

$$\hat{\mathbf{p}} = \hat{p}\,\mathbf{n} = \frac{(\varepsilon+1)(\mathbf{v}_2\cdot\mathbf{n}-\mathbf{v}_1\cdot\mathbf{n})}{\dfrac{1}{m_1}+\dfrac{1}{m_2}}\ \mathbf{n}.$$

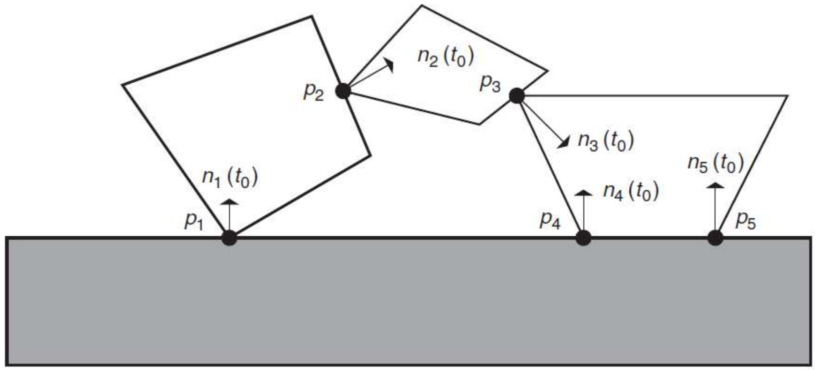
$$\hat{\mathbf{p}} = -2m_1(\mathbf{v}_1\cdot\mathbf{n})\ \mathbf{n};$$

$$\mathbf{v}_1' = \frac{\mathbf{p}_1+\hat{\mathbf{p}}}{m_1} = \frac{m_1\mathbf{v}_1 - 2m_1(\mathbf{v}_1\cdot\mathbf{n})\ \mathbf{n}}{m_1}$$

$$= \mathbf{v}_1 - 2(\mathbf{v}_1\cdot\mathbf{n})\ \mathbf{n}.$$

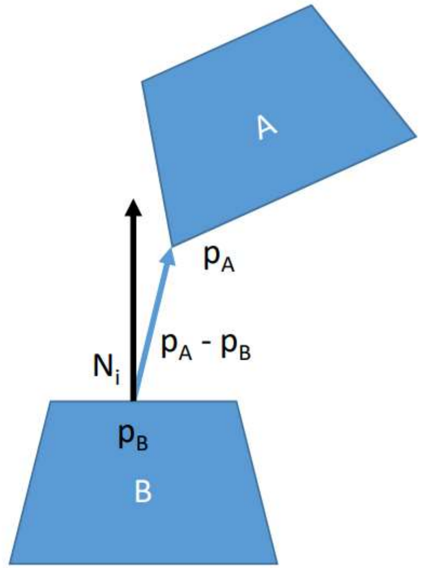
Physically-based Animation

- Rigid body simulation
  - Collision
    - Resting contact



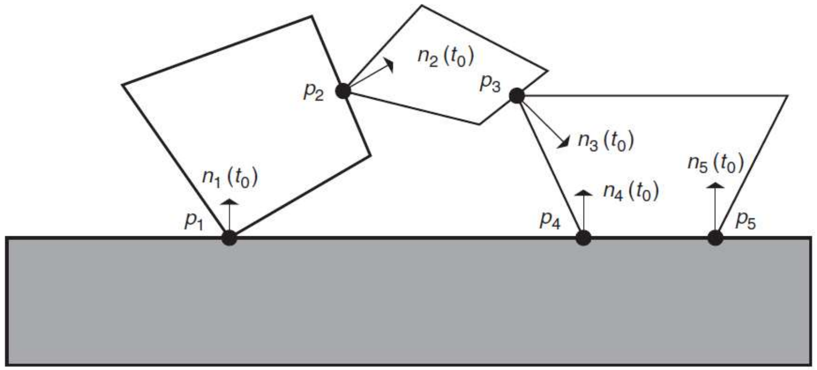
$$d_i(t) = (p_A(t) - p_B(t)) \cdot N_i$$

d > 0, moving
d = 0, stay
d < 0, no!!!!



Physically-based Animation

- Rigid body simulation
  - Collision
    - Resting contact



$$d_i(t_0) = \dot{d}_i(t_0) = 0 \qquad \ddot{d}_i(t) \geq 0$$

$$\dot{d}_i(t) = \dot{\mathrm{N}}_i(t) \cdot (\mathrm{p}_A(t) - \mathrm{p}_B(t)) + \mathrm{N}_i \cdot (\dot{\mathrm{p}}_A(t) - \dot{\mathrm{p}}_B(t))$$

$$\ddot{d}_i(t) = (\mathrm{p}_A(t) - \mathrm{p}_B(t)) \cdot \ddot{\mathrm{N}}_i + 2(\dot{\mathrm{p}}_A(t) - \dot{\mathrm{p}}_B(t)) \cdot \dot{\mathrm{N}}_i + (\ddot{\mathrm{p}}_A(t) - \ddot{\mathrm{p}}_B(t)) \cdot \mathrm{N}_i$$

$$\ddot{d}_i(t) = \boxed{2(\dot{\mathrm{p}}_A(t_0) - \dot{\mathrm{p}}_B(t_0)) \cdot \dot{\mathrm{N}}_i} + \boxed{(\ddot{\mathrm{p}}_A(t_0) - \ddot{\mathrm{p}}_B(t_0)) \cdot \mathrm{N}_i}$$

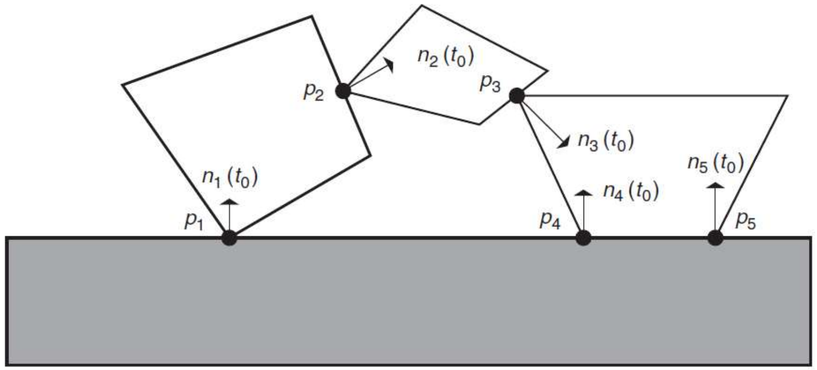
Not dependent on force f

dependent on force f

$d_i(t) \geq 0$	the forces must prevent penetration
$f_i \geq 0$	the forces must push objects apart, not together
$\ddot{d}_i(t) f_i = 0$	either the objects are not separating or, if the objects are separating, then the contact force is zero

Physically-based Animation

- Rigid body simulation
  - Collision
    - Resting contact



$$\ddot{d}_i(t) = (p_A(t) - p_B(t)) \cdot \ddot{N}_i + 2(\dot{p}_A(t) - \dot{p}_B(t)) \cdot \dot{N}_i + (\ddot{p}_A(t) - \ddot{p}_B(t)) \cdot N_i$$

$$\ddot{d}_i(t) = \boxed{2(\dot{p}_A(t_0) - \dot{p}_B(t_0)) \cdot \dot{N}_i} + \boxed{(\ddot{p}_A(t_0) - \ddot{p}_B(t_0)) \cdot N_i}$$

Not dependent on force f

dependent on force f

Relative acceleration $\ddot{d}_i(t) = b_i + \sum_{j=1}^{n}(a_{ij}f_j)$

Unknowns, f

Physically-based Animation

- Rigid body simulation
  - Collision
    - Resting contact

$$\ddot{d}_i(t) = (p_A(t) - p_B(t)) \cdot \ddot{N}_i + 2(\dot{p}_A(t) - \dot{p}_B(t)) \cdot \dot{N}_i + (\ddot{p}_A(t) - \ddot{p}_B(t)) \cdot N_i$$

Relative acceleration

Solve for f $\ddot{d}_i(t) = b_i + \sum_{j=1}^{n}(a_{ij} f_j)$

$$d_i(t) = (p_A(t) - p_B(t)) \cdot N_i$$

Linear velocity

$$\dot{p}_A(t) = v_A(t) + \omega_A(t) \times r_A(t)$$
$$\dot{p}_B(t) = v_B(t) + \omega_B(t) \times r_B(t)$$

$$\omega(t) = I(t)^{-1} L(t)$$

Angular velocity

Vector from the Centre of Mass to the contact point

linear acceleration A force f_j acting in direction $n_j(t_0)$ produces $f_j / m_A \cdot n_j(t_0)$

$$\ddot{p}_A(t) = \boxed{\dot{v}_A} + \dot{\omega}_A(t) \times r_A(t) + \boxed{\omega_A(t) \times (\omega_A(t) \times r_A(t))}$$

$$\dot{\omega}_A(t) = I_A^{-1}(t)\boxed{\tau_A(t)} + \boxed{I^{-1}(t)(L_A}$$

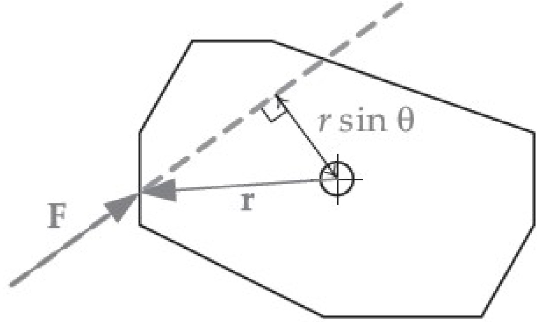
Force related

Fo

rce unrelated

$$(p_j - x_A(t_0)) \times f_j\, n_j(t_0)$$

$$\mathbf{N} = \mathbf{I}\alpha(t) = \mathbf{I}\frac{d\omega(t)}{dt} = \frac{d}{dt}(\mathbf{I}\omega(t)) = \frac{d\mathbf{L}(t)}{dt}$$



$$\mathbf{N} = \mathbf{r} \times \mathbf{F}.$$

Physically-based Animation

- Rigid body simulation
  - Collision
    - Resting contact

$$\ddot{d}_i(t) = (p_A(t) - p_B(t)) \cdot \ddot{N}_i + 2(\dot{p}_A(t) - \dot{p}_B(t)) \cdot \dot{N}_i + (\ddot{p}_A(t) - \ddot{p}_B(t)) \cdot N_i$$

Relative acceleration

Solve for f $\ddot{d}_i(t) = b_i + \sum_{j=1}^{n}(a_{ij}f_j)$

$$d_i(t) = (p_A(t) - p_B(t)) \cdot N_i$$

Linear velocity

$$\dot{p}_A(t) = v_A(t) + \omega_A(t) \times r_A(t)$$
$$\dot{p}_B(t) = v_B(t) + \omega_B(t) \times r_B(t)$$

$$\omega(t) = I(t)^{-1}L(t)$$

Angular velocity

Vector from the Centre of Mass to the contact point

linear acceleration A force f_j acting in direction $n_j(t_0)$ produces $f_j / m_A \cdot n_j(t_0)$

$$\ddot{p}_A(t) = \boxed{\dot{v}_A} + \dot{\omega}_A(t) \times r_A(t) + \boxed{\omega_A(t) \times (\omega_A(t) \times r_A(t))}$$

Force unrelated

$$\dot{\omega}_A(t) = I_A^{-1}(t)\boxed{\tau_A(t)} + \boxed{I^{-1}(t)(L_A(t) \times \omega_A(t))}$$ Angular acceleration

Force related

Force unrelated

$$(p_j - x_A(t_0)) \times f_j\, n_j(t_0)$$

Physically-based Animation

- Rigid body simulation
  - Collision
    - Resting contact

Force independent　　Force dependent

Solve for f $\ddot{d}_i(t) = b_i + \sum_{j=1}^{n}(a_{ij}f_j)$　　$\ddot{d}_i(t) = 2(\dot{p}_A(t_0) - \dot{p}_B(t_0)) \cdot \dot{N}_i + (\ddot{p}_A(t_0) - \ddot{p}_B(t_0)) \cdot N_i$　　$f_i \geq 0$　$\ddot{d}_i(t) f_i = 0$

$$\ddot{p}_A(t) = \dot{v}_A + \dot{\omega}_A(t) \times r_A(t) + \omega_A(t) \times (\omega_A(t) \times r_A(t))$$

Can be decomposed into force dependent/independent terms, the same as $\ddot{p}_B(t)$

Force dependent $$f_j\left(\frac{N_j(t_0)}{m_A} + (I_A^{-1}(t_0)(p_j - x_A(t_0) \times N_j(t_0)) \times r_A\right)$$

Force independent $$\frac{F_A(t_0)}{m_A} + I_A^{-1}(t)\tau_A(t) + \omega_A(t) \times (\omega_A(t) \times r_A) + (I_A^{-1}(t)(L_A(t) \times \omega_A(t))) \times r_A$$

Net external force　　Net external torque

Physically-based Animation

- Rigid body simulation
  - Collision
    - Resting contact

Solve for f $\ddot{d}_i(t) = b_i + \sum_{j=1}^{n}(a_{ij}f_j)$

Subject to
$$f_i \geq 0$$
$$\ddot{d}_i(t)f_i = 0$$

Quadratic Programing to handle $\ddot{d}_i(t) = 0$

$$\text{minimize} \quad \tfrac{1}{2}\mathbf{x}^{\mathrm{T}}Q\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x}$$
$$\text{subject to} \quad A\mathbf{x} \leq \mathbf{b},$$