Collision detection

- Static interference tests
- Exact collision prediction
- Conservative tests and acceleration techniques

"Collision detection is one of the most challenging and most important problems in computer animation!"

Updated September 15, 2011

Types of animations

- Camera moves in static environment
- Rigid motion: Objects are rigid, they only rotate and translate
- Articulated bodies: Parameterized linkage of rigid bodies
  - A child's pose (forearm) is obtained from a parent's pose (upper-arm) by an arbitrary rotation (elbow angle) and a fixed translation (length of upper-arm).
- Deformations: Each shape deforms with time
  - Deformation may be simulated using naïve physics or computed to interpolate between two shapes (3D morphing)
- Reactive: Responds to user actions
  - Shape moves and deforms when approached or poked by the user

All must detect collisions and adjust behavior

The collision detection problem

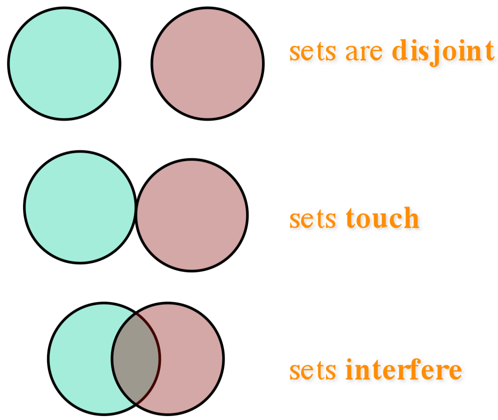
Shapes A and B evolve (move, deform) with time.

Notation for their instance at time $t>0$: $A(t)$ and $B(t)$.

Find the smallest t such that $A(t) \cap B(t) \neq \emptyset$.

Interference

- A and B interfere when they share at least one common point.
- $A \cap B \neq \emptyset$

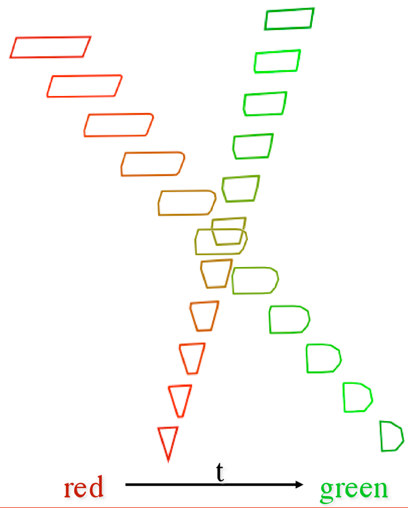


Static interference tests for collision

A and B evolve (move, deform) with time: A(t) and B(t).

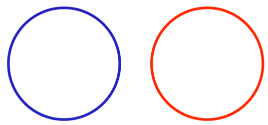
Test and then assume A(0)∩B(0)=∅

```
t=0;
repeat {
if (A(t+dt)∩B(t+dt) ≠∅) {
  repeat 5 times {
   if (A(t+dt/2)∩B(t+dt/2)≠∅) dt=dt/2;
   else t+=dt/2; };
   return(t); };
    }
return(-1);}   // no collision found
```

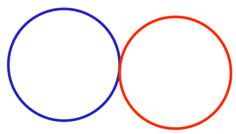


How to test for interference

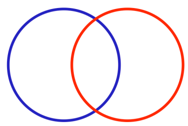
- Often (not always) we are using the bundaries of shapes to test for interference.



bundaries are disjoint



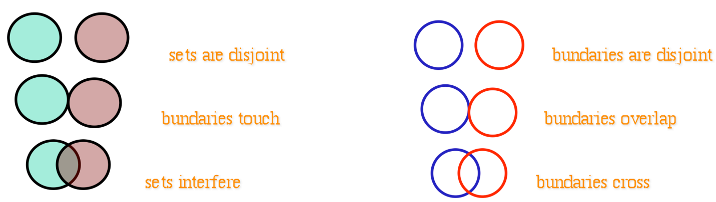
bundaries overlap



bundaries cross

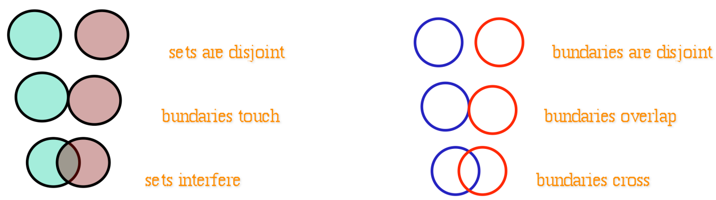
Testing boundaries

- A and B are disjoint if and only if their boundaries are disjoint
  - True or false?
  - If true, prove or justify
  - If false, provide a counterexample
    and propose a correct test using the boundaries:



Distinguish touch and interfere

- How can you use tests on boundries to distinguish
  - disjoint
  - touching
  - interfering



Detecting collisions between balls

- For simplicity, we assume that objects collide when their enclosing balls collide
  - Compute enclosing ball of center c and radius r as follows:
    - $\mathbf{c}$:= $((x_{min}+x_{max})/2, (y_{min}+y_{max})/2, (z_{min}+z_{max})/2)$
    - r := max (||**cv**||) for all vertices **v** of the object
- Interference detection:
  - After each time step, check whether any pair of objects interfere
  - Ball(c_1,r_1) and Ball(c_2,r_2) interfere when
  - Need very small steps not to miss a chock

$$||\mathbf{c_1 c_2}|| < r_1+r_2$$

- Collision prediction:
  - Express the relative motion of object 2 in the CS of object 1
    - For motions with fixed velocities, $\underline{\mathbf{v}}_1$ and $\underline{\mathbf{v}}_2$, you get $\mathbf{v}_1$ and $\mathbf{v}_2$, $\underline{\mathbf{v}} = \underline{\mathbf{v}}_2 - \underline{\mathbf{v}}_1$.
  - Shrink object 2 to a point c_2 and expand object 1 to Ball(c_1,r_1+r_2)
  - Check whether curve swept by c_2 intersects Ball(c_1,r_1+r_2)
    - Constant velocity motion: find smallest time t>0 when $(\mathbf{c_2}+t\underline{\mathbf{v}})^2=(\mathbf{r_1+r_2})^2$

Collision between disks

Assume that disk D_1=disk(C_1,r_1) will travel with constant velocity V_1. Similarly, disk D_2 =disk(C_2,r_2) will travel with constant velocity V_2. Assume that they are initially disjoint.

- How would you compute the **time t** when they will **collide**?

  Solve $((C_2+tV_2)–(C_1+tV_1))^2=(r_1+r_2)^2$ for t

  $(C_1C_2+t(V_2–V_1))^2=(r_1+r_2)^2$

  $(V_2–V_1)^2\, t^2 + 2C_1C_2\bullet(V_2–V_1)\, t + C_1C_2{}^2 –(r_1+r_2)^2 = 0$

  return the smallest positive value of t if one exists

*We can reduce this problem to the one of **line/circle intersection**:*

D_2 is stationary, D_1 moves by $V_1–V_2$

D_2 has radius r_1+r_2 (inflated by r_1), D_1 is a point (deflated by r_1)

when will D_1 (the point) hit the inflated D_2?

Elastic shock between two disks

Assume that disk D_1=disk(C_1,r) traveling with **constant velocity** V_1 and disk D_2 =disk(C_2,r) traveling with constant velocity V_2 have just collided. What should their velocities V_1' and V_2' be?

(We assume that they have the ***same mass****.)*

We must **exchange their normal velocities**

$N = U(C_1C_2)$ (normal direction to both at contact point)

$U_1 = (V_1 \bullet N)\, N$ (normal components of velocities)

$U_2 = (V_2 \bullet N)\, N$

$V_1' = V_1 - U_1 + U_2$ (cancel U_1 and add U_2)

$V_2' = V_2 - U_2 + U_1$ (to exchange their normal velocities)

Elastic shock with a fixed disk

Assume that disk D_1=disk(C_1,r), traveling with **constant velocity** V, has just collided with a **static** disk D_2 =disk(C_2,r). What should its velocity V' after the bounce?

(We assume that D_2 has infinite mass.)

We must negate (**reverse**) the **normal component** of V

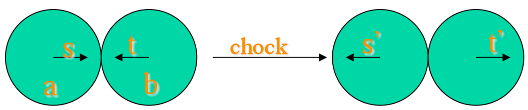
$N = C_1C_2$.unit (normal direction to both at contact point)

$U = (V \bullet N)\, N$ (normal components of V)

$V' = V_1 - 2U$ (swap direction of U)

Elastic collision between balls

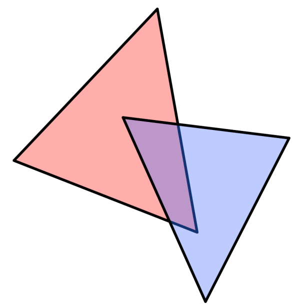
- Tangential velocities are maintained. Masses: a and b
- Velocities in the normal direction: s and t before and s' and t' after the shock
  - Conservation of momentum: as+bt=as'+bt' (1)
  - Conservation of energy: $as^2+bt^2=as'^2+bt'^2$ (2)
  - Regrouping in (1): a(s-s')=b(t'-t) (3)
  - Regrouping in (2) and using $x^2-y^2=(x+y)(x-y)$: a(s-s')(s+s')=b(t'-t)(t'+t) (4)
  - Combining (3) and (4) : s+s'=t+t' (5)
  - Substituting t'=s+s'-t obtained from (5) in (1) yields: as+bt=as'+bs+bs'-bt (6)
  - Reorganizing (6) : (a+b)s'=(a-b)s+2bt (7)
  - Reorganizing and swapping (s,a) and (t,b):
  - s'=s+2b(t-s)/(a+b) and t'=t+2a(s-t)/(a+b) (8)
- Note that when a=b, then s'=t and t'=s (exchange of normal velocities)
- When a>>b: s'=s (not affected) and t'=2s-t (reverse speed of b if a is static)



2D interference between triangles

- Two triangles interfere if a vertex of one lies inside the other
  - True or false?

  - Justify
  - of provide a counterexample
  - and a correct test



Modify your test to distinguish disjoint, interfere, and touch

Test for polygon/polygon intersection in 2D?

- A and B are polygons
  - They are connected sets in 2D, but may have holes
- Write the high-level test for checking whether they interfere

For each connected **component** of **A** do, if the **first vertex** of that component is **in or on B**, return TRUE

For each connected component of B do, if the first vertex of that component is in or on A, return TRUE

For each **edge** Ea of A do, for each **edge** Eb of B do, if Ea intersects Eb, return TRUE

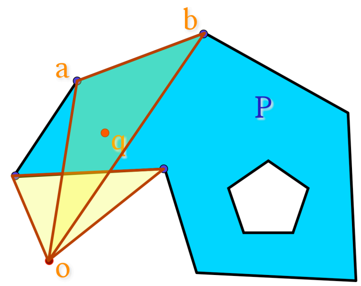
Otherwise return FALSE

Details

- Point-in-polygon
- Edge/edge intersection

Point-in-polygon test

- **Algorithm for testing whether point** q **is inside polygon P**



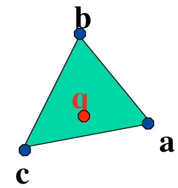
inP:=false;

for each edge (**a**,**b**) of P {if (inTriangle(**q**,**a**,**b**,**o**)) then inP := !inP;

return(inP);

Point-in-triangle test in 2D

- **Is point** q **in tri**(a,b,c) ?



ab×aq

bc×bq

ca×cq

all have the same sign

Edge/edge intersection in 2D

- **Write a geometric expression that returns TRUE when edges** (a,b) **and** (c,d) **intersect**

$$((ab \times ac > 0) \mathrel{!=} (ab \times ad > 0))$$

$$\&\&$$

$$((cd \times ca > 0) \mathrel{!=} (cd \times cb > 0))$$