

Osnove virtualnih okruženja

Igor S. Pandžić

Detekcija presjeka i sudara

- ◆ Metode testiranja da li se dva geometrijska elementa sijeku (i gdje se sijeku)
 - Npr. zraka-trokut, zraka-kugla, trokut-trokut
- ◆ Široka primjena
 - Detekcija sudara
 - Odabir (engl. picking)
 - Iscrtavanje praćenjem zrake
 - Odbacivanje po projekcionom volumenu

- ◆ Da li se (složeni) predmeti u sceni sudaraju
 - Gdje se sudaraju?
 - Kako na to reagirati?
- ◆ Metode zasnovane na detekciji presjeka
- ◆ Primjene
 - Navigacija
 - Simulacija vozila, leta
 - Simulacije zasnovane na fizici (npr. simulacija odjeće)
 - Igre
 - Robotika, planiranje putanje itd.



- ◆ Detekcija presjeka
 - Opća pravila detekcije presjeka
 - Presjek zrake sa kuglom, kvadrom, trokutom, poligonom
 - Presjeci obujmica
 - Kugla-kugla, kugla-kvadar, kvadar-kvadar
 - Presjeci obujmica s projekcionim volumenom:
 - Kugla, kvadar
- ◆ Detekcija sudara
 - Aproksimacija predmeta zrakama
 - Hijerarhijska detekcija sudara

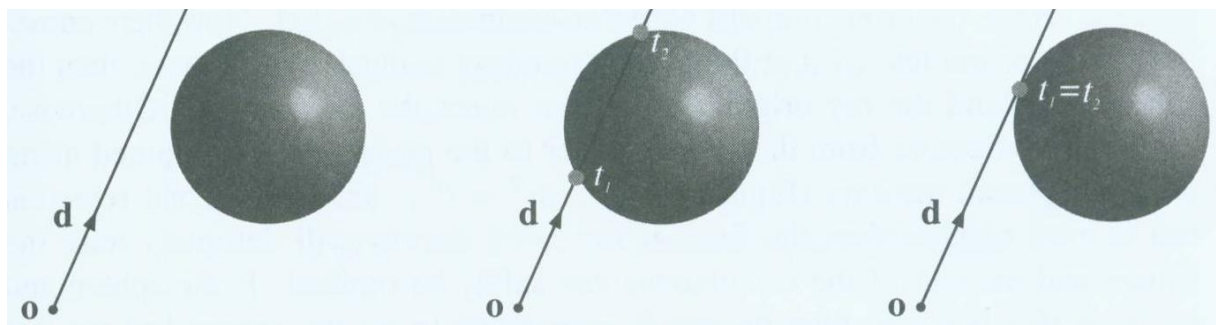
- ◆ Prvo odbaciti trivijalne slučajeve
- ◆ Ukoliko se koristi više testova/faza, prvo jednostavnije
 - Koristiti rezultate iz prethodne faze
- ◆ Ako se testira jedan element u odnosu na veći broj drugih elemenata, sve što je moguće izračunati u pripremnoj fazi

Presjek zraka-kugla: matematičko rješenje

- ◆ Kugla polumjera r sa središtem u C : $\|P - C\| - r = 0$
- ◆ Zraka smjera D s ishodištem O : $P(t) = O + tD, t \geq 0$
- ◆ Uvrštavanjem: $\|P(t) - C\| - r = 0 \Rightarrow t^2 + 2tb + c = 0$

$$b = D \cdot (O - C), c = (O - C) \cdot (O - C) - r^2$$

$$t = -b \pm \sqrt{b^2 - c}$$



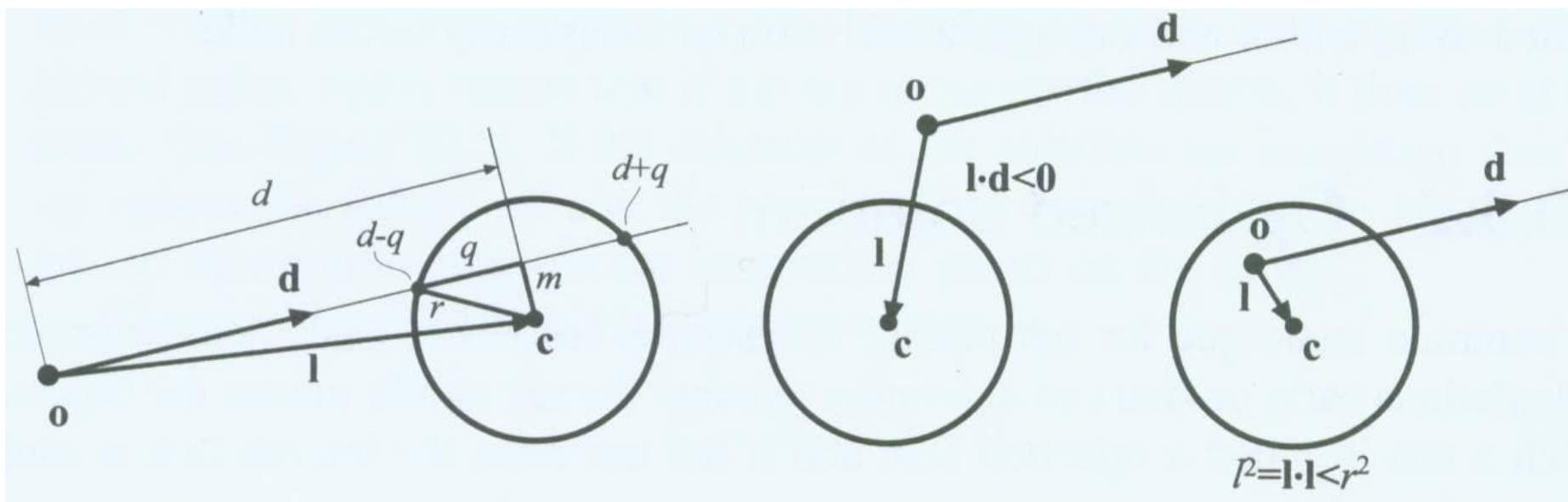
$$b^2 - c < 0$$

$$b^2 - c > 0$$

$$b^2 - c = 0$$

Zraka-kugla: optimalno rješenje (1/2)

- ◆ Koriste se jednostavni testovi za odbacivanje



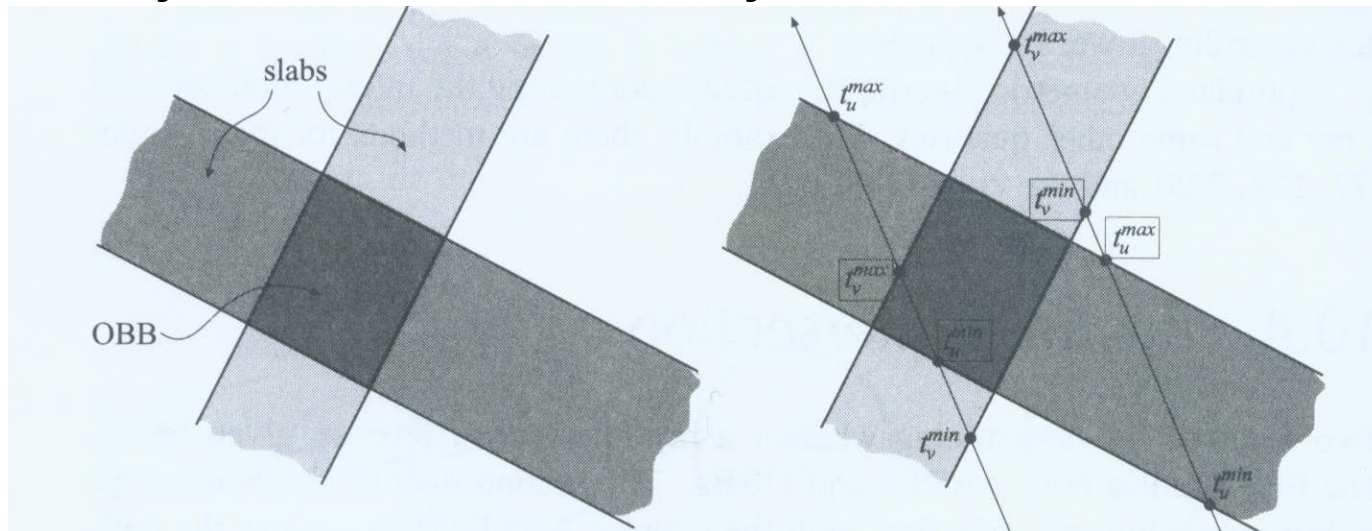
Zraka-kugla: optimalno rješenje (2/2)

```
RaySphereIntersect(o, d, c, r)
returns ({REJECT, INTERSECT}, t, p)

1:  l = c - o
2:  d = l · d
3:  l2 = l · l
4:  if(d < 0 and l2 > r2) return (REJECT, 0, 0);
5:  m2 = l2 - d2
6:  if(m2 > r2) return (REJECT, 0, 0);
7:  q = √(r2 - m2)
8:  if(l2 > r2) t = d - q
9:  else t = d + q
10: return (INTERSECT, t, o + td);
```


♦ Metoda krišaka

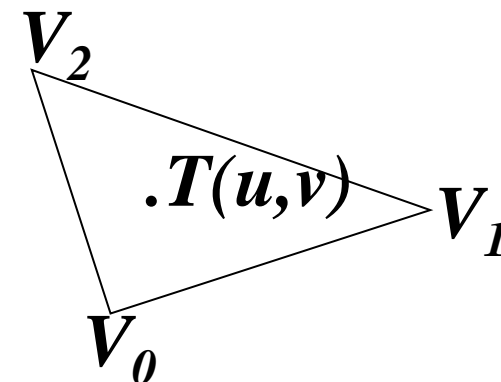
- Kvadar je presjek 3 kriške (u, v, w)
- Zraka siječe krišku u dvije točke $t_i^{\min}, t_i^{\max}, i \in \{u, v, w\}$
- $t^{\min} = \max(t_u^{\min}, t_v^{\min}, t_w^{\min}), t^{\max} = \min(t_u^{\max}, t_v^{\max}, t_w^{\max})$
- Ako je $t^{\min} \leq t^{\max}$ zraka siječe kvadar



- ◆ Baricentrične koordinate
 - Korisne u raznim interpolacijama unutar trokuta

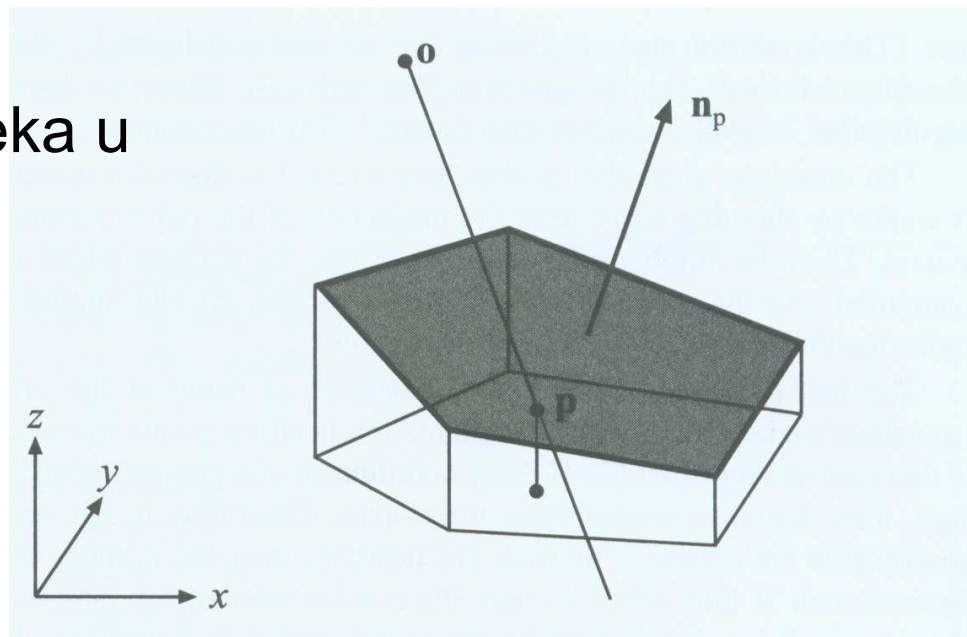
$$T(u, v) = (1 - u - v)V_0 + uV_1 + vV_2$$

$$u \geq 0, v \geq 0, u + v \leq 1$$

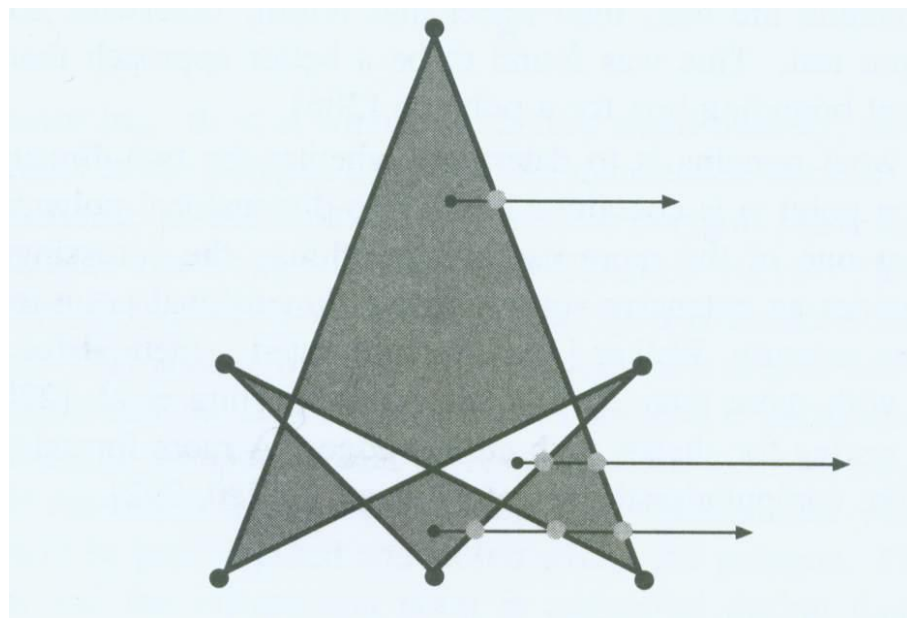


- ◆ Zraka $P(t) = O + tD, t \geq 0$
- ◆ Uvrštavanjem $P(t)=T(u, v)$ - sustav s 3 nepoznanice

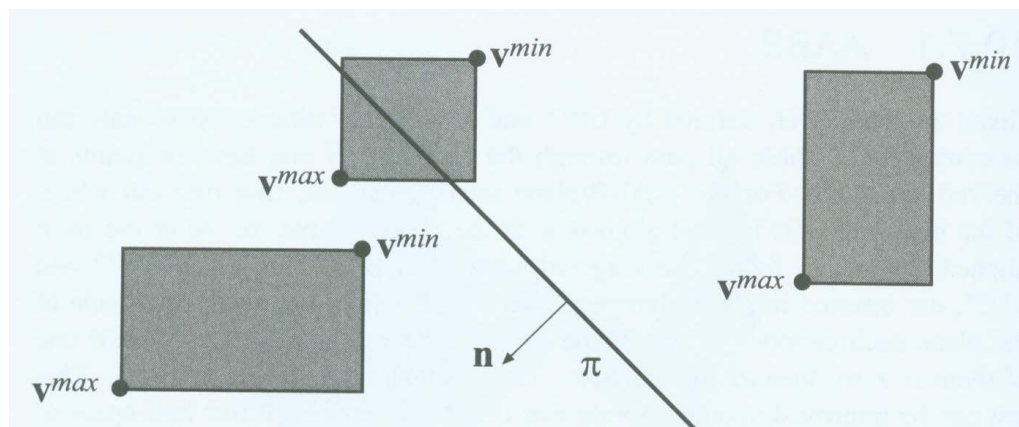
- ◆ Poligon u ravnini π
 1. Presjek zrake s ravninom
 2. Projekcija poligona i točke presjeka u koordinatnu plohu s najvećom površinom projekcije
 3. Da li je točka presjeka u 2D poligonu?



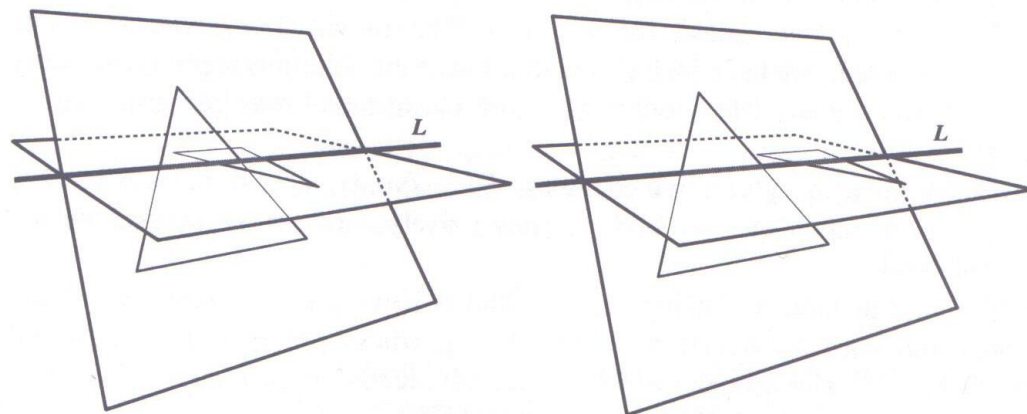
- ♦ Zraka iz točke u proizvoljnom smjeru presijeca poligon n puta
 - Ako je n neparan broj, točka je u poligonu
 - Ovo važi za sve poligone, uključujući konkavne i samo-presjecajuće



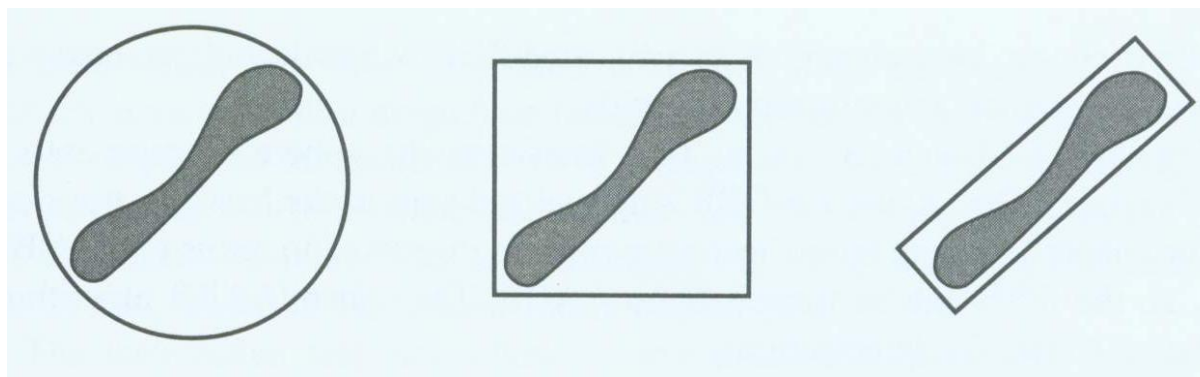
- ◆ Ako su bilo koja dva vrha kvadra na suprotnim stranama ravnine, presjek
 - Uvrstiti svaki vrh u jednačbu ravnine, dobiva se udaljenost d ; usporediti predznake
- ◆ Jednostavnije: dovoljno je testirati dva vrha na dijagonali najbližoj normalni na ravninu



- ◆ Trokut T_1 u ravnini π_1 , T_2 u π_2
 1. Izračunati udaljenost svih vrhova T_1 do π_2 ; ako su istog predznaka nema presjeka
 2. Isti test za T_2 do π_1
 3. Pravac L je presjek π_1 i π_2 ; pronaći presjeke T_1 i T_2 sa L ; to su intervali I_1 i I_2 ; ako se preklapaju, imamo presjek



- ◆ Obujmice:
 - Kugla
 - Kvadar paralelan s osima (engl. Axis-aligned bounding box, AABB)
 - Općeniti kvadar (engl. Oriented bounding box, OBB)

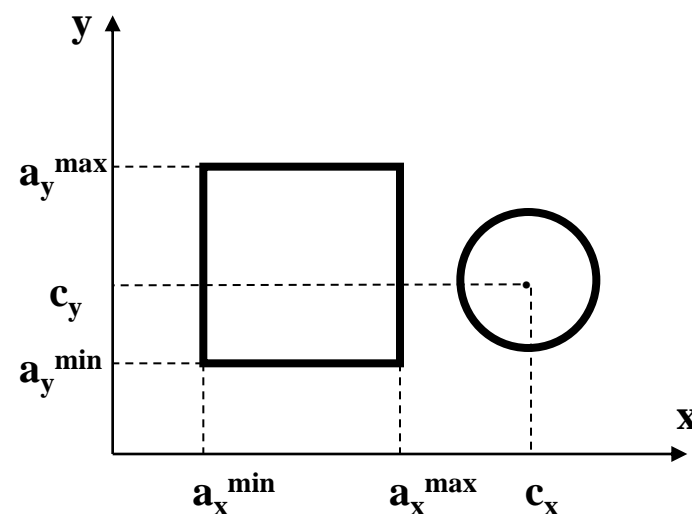


- ◆ Trivijalan test
 - Izračunati udaljenost dvaju središta
 - Usporediti sa zbrojem polumjera

Presjek kugla-kvadar paralelan s osima

- ◆ Ideja: usporediti udaljenost središta kugle od kvadra s polumjerom kugle

```
bool SphereAABB_intersect(c, r, A)  
returns({OVERLAP, DISJOINT});  
1:  $d = 0$   
2: for each  $i \in \{x, y, z\}$   
3:   if( $c_i < a_i^{min}$ )  
4:      $d = d + (c_i - a_i^{min})^2$ ;  
5:   else if( $c_i > a_i^{max}$ )  
6:      $d = d + (c_i - a_i^{max})^2$ ;  
7: if( $d > r^2$ )  
8:   return (DISJOINT);  
9: return (OVERLAP);
```



- ◆ Za opći kvadar:
 - prvo transformirati centar kugle u koordinatni sustav kvadra

- ♦ Vrlo brz, jednostavan i često korišten test
- ♦ Gleda se preklapanje po svakoj osi posebno

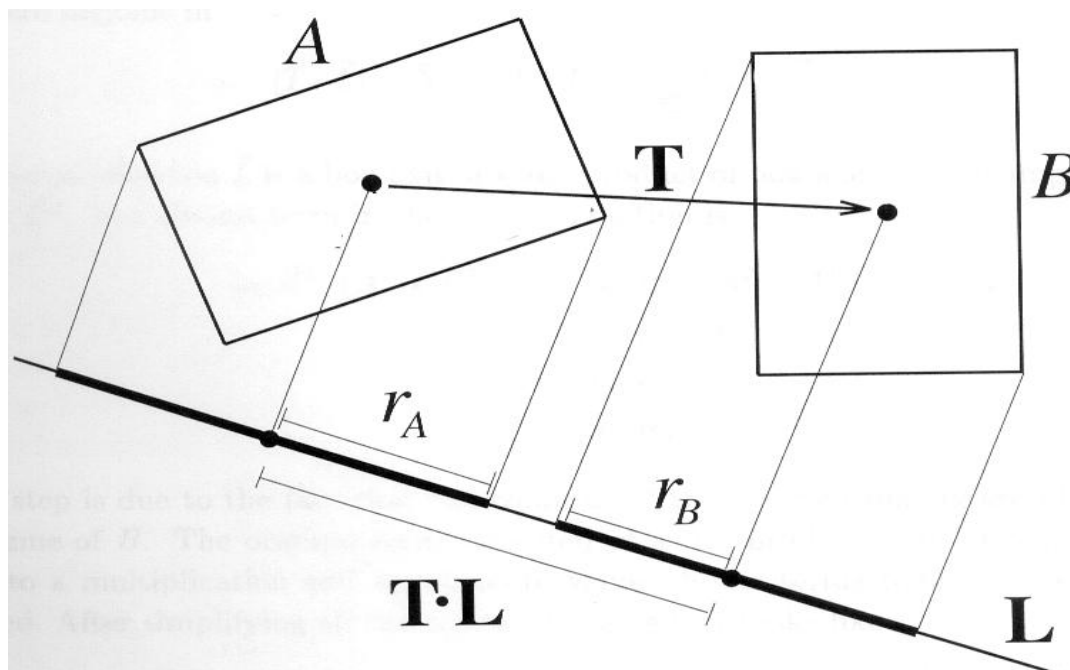
```
bool AABBB_intersect( $A, B$ )  
returns({OVERLAP, DISJOINT});  
1: for each  $i \in \{x, y, z\}$   
2:   if( $a_i^{min} > b_i^{max}$  or  $b_i^{min} > a_i^{max}$ )  
3:     return (DISJOINT);  
4: return (OVERLAP);
```

- ◆ Engl. separating axis theorem
- ◆ Za dva konveksna poliedra koji se ne sijeku, postoji ploha koja ih razdvaja i koja je paralelna s jednom od stranica jednoga od poliedara, ili s plohom koju opisuju dva brida, po jedan iz svakog poliedra
- ◆ Projekcije poliedara na normalu razdvajajuće plohe se ne preklapaju - to je razdvajajuća os

- ◆ Testiraju se potencijalne razdvajajuće osi po teoremu
- ◆ Kandidati su:
 - Osi kvadra A (3)
 - Osi kvadra B (3)
 - Kros-produkti parova osi A i B (9)
- ◆ Testira se ovim redoslijedom
 - Ukoliko nema presjeka, najčešće se ustanovi vrlo brzo
 - Najveći broj operacija ako presjek postoji

Presjek kvadar-kvadar (2/2)

- ◆ Kako testirati svaku os?



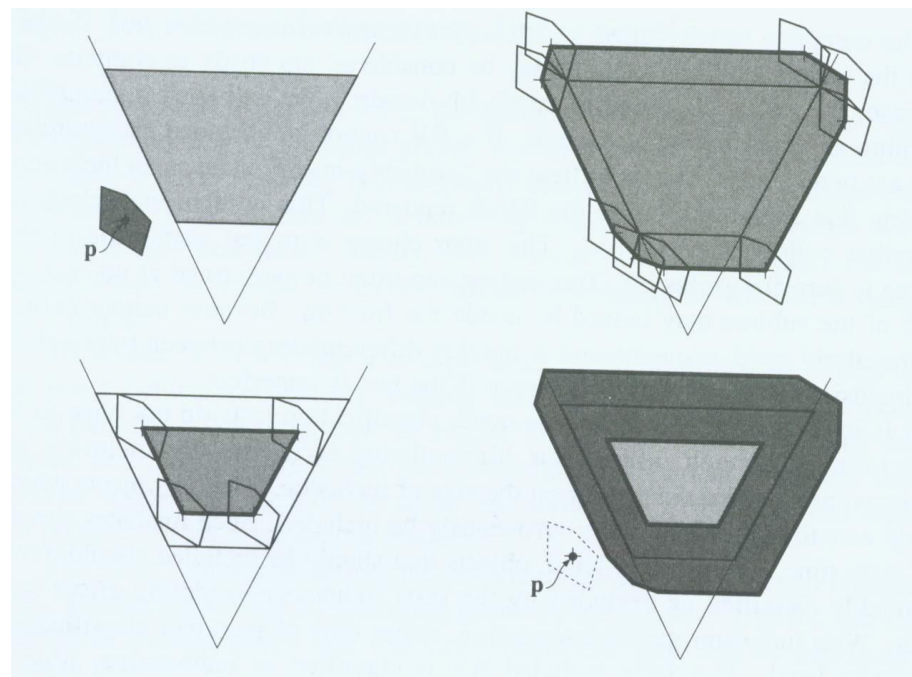
Nema sudara ako je

$$T \cdot L > r_A + r_B$$

- ◆ Koristi se za odbacivanje po projekcionom volumenu
- ◆ Puni test
 - Vani - odbaci sve
 - Unutra - crtaj sve
 - Presjek - testiraj sljedeću razinu
- ◆ Jednostavni test
 - Vani - odbaci sve
 - Vjerojatno unutra
 - A) crtaj sve (možda previše) ILI
 - B) testiraj sljedeću razinu (možda nepotrebno)

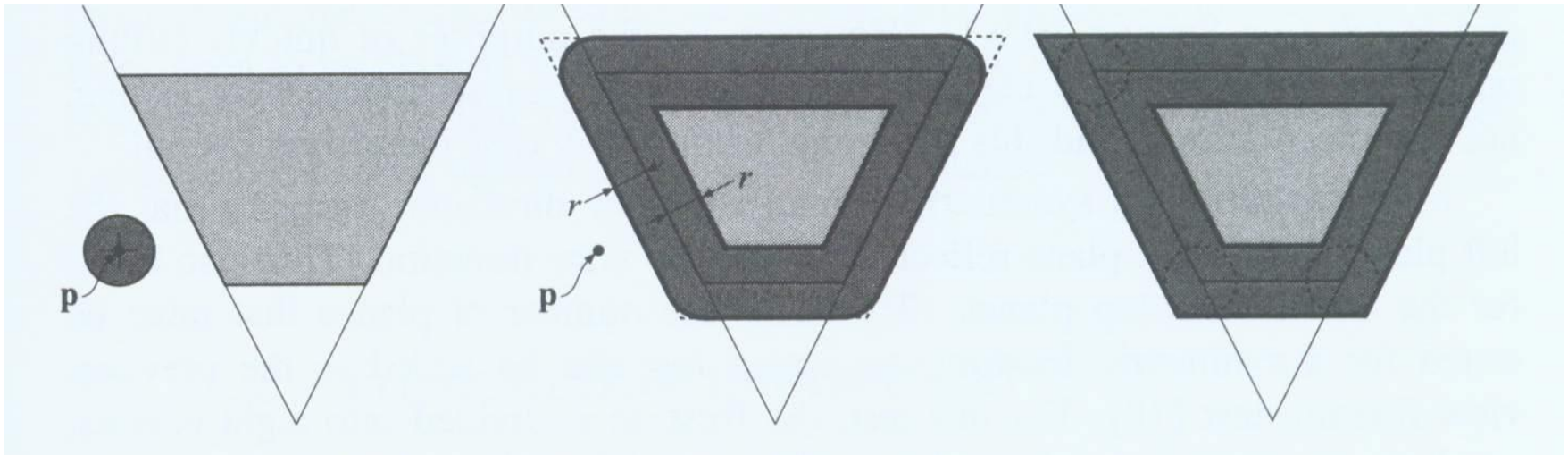
◆ Pojednostavljenje testa

- Povlačeći obujmicu (općeniti oblik) po vanjskom i unutrašnjem rubu volumena, centar obujmice opisuje nove granice
- Test se svodi na točku u volumenu



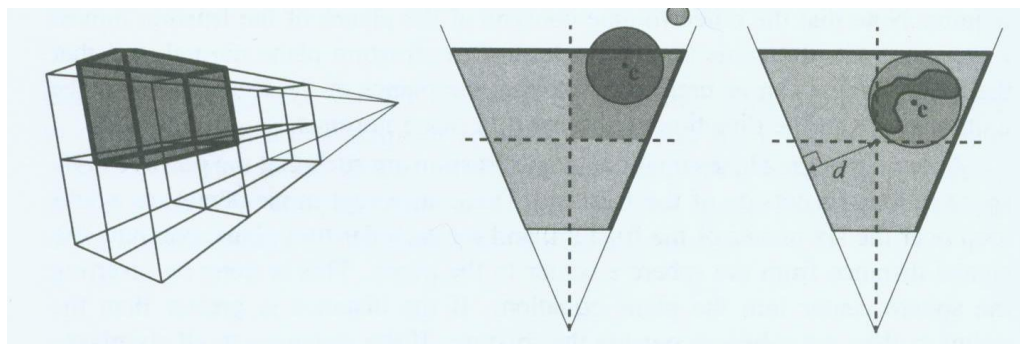
Presjek kugla-projekcioni volumen (1/2)

- ♦ Stranice volumena pomaknute za $\pm r$



Presjek kugla-projekcioni volumen (2/2)

- ◆ Postupak (normale stranica prema van)
 - Za svaku stranicu volumena, izračunati udaljenost d od središta kugle do stranice
 - Ako je $d > r$ za bilo koju stranicu, kugla je vani
 - Ako je $d < -r$ za sve stranice, kugla je unutra
 - U ostalim slučajevima, presijecanje
- ◆ Broj stranica za testiranje može se smanjiti na pola prethodnim testiranjem oktanta

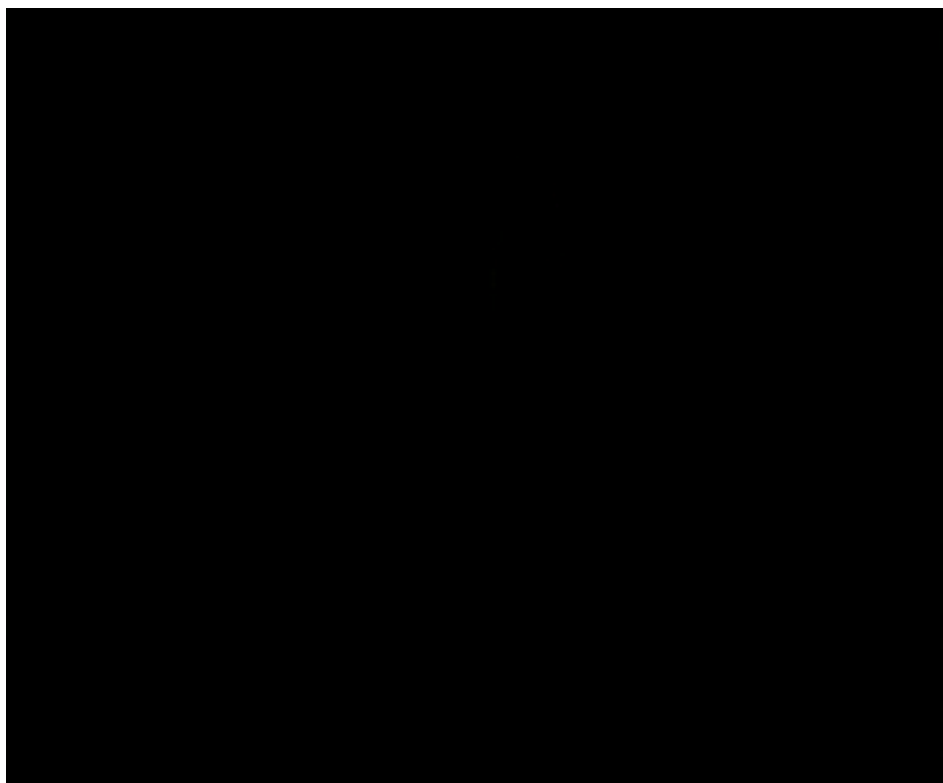


- ◆ Slično kao za kuglu, testirati svaku stranicu volumena posebno
 - Koristiti test ravnina-kvadar
- ◆ Druga mogućnost: teorem razdvajajuće osi

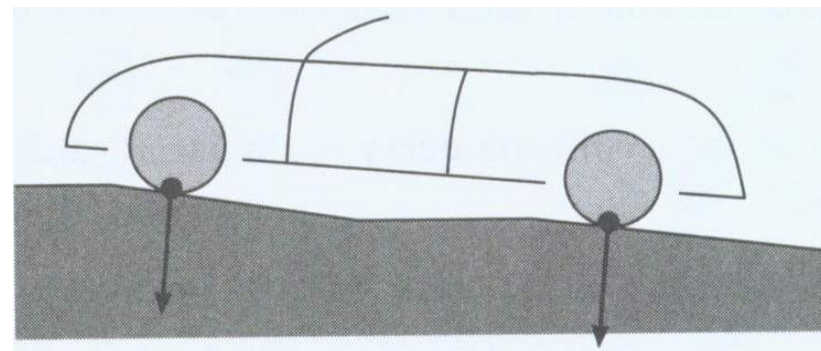
- ◆ Engl. collision detection
- ◆ Do sada obrađene metode koriste se za detekciju sudara među predmetima u sceni
- ◆ Točni zahtjevi na test ovise o primjeni
 - Preciznost
 - Vrsta i količina podataka
 - Statička ili dinamička detekcija
 - Kvantni efekt



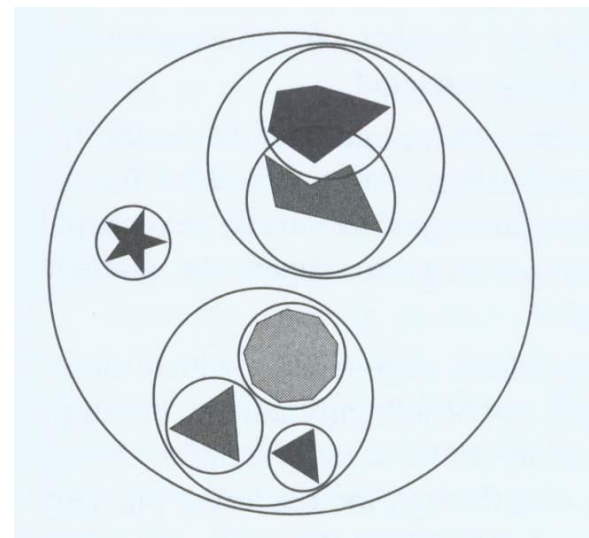
- ◆ Aproksimacija predmeta zrakama
- ◆ Hijerarhijska detekcija sudara



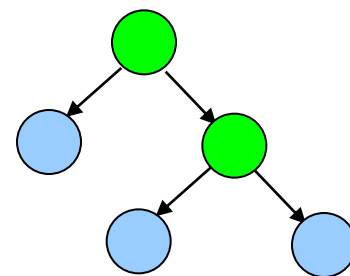
- ♦ Jednostavna i brza metoda
- ♦ Prilično gruba aproksimacija, ali dobri rezultati u praksi
- ♦ Zrake se postavljaju na strateška mjesta na predmetu
- ♦ Traži se presjek zrake sa scenom
 - $d=0$ - predmet dotiče scenu
 - $d>0$ - predmet je iznad scene
 - $d<0$ - predmet ulazi u scenu



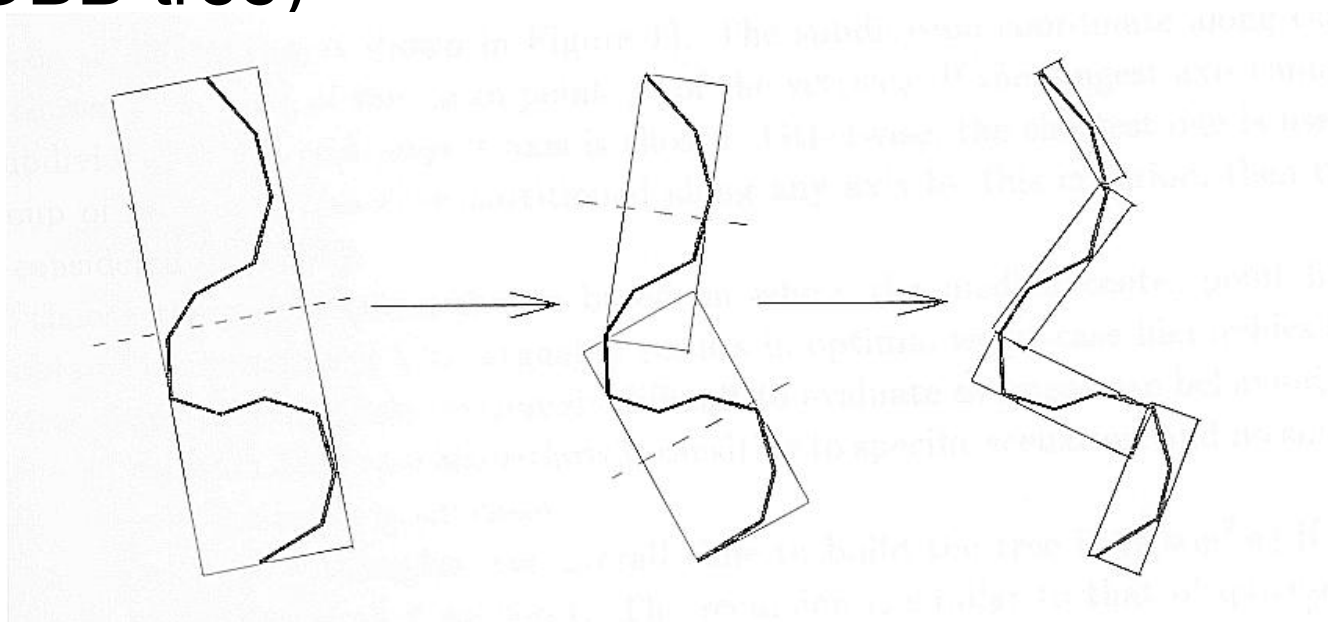
- ♦ Svaki predmet je skup poligona
- ♦ Svaki predmet se pretvara u hijerarhiju (stablo) obujmica
- ♦ Testiranje sudara između dva takva predmeta vrši se od vrha hijerarhije
 - Ako se već najviša razina obujmica ne siječe, vrlo brzo ustanovimo da nema sudara



- ◆ Hijerarhija se zapisuje u obliku stabla
 - Grane sadrže obujmice
 - Listovi sadrže geometriju
- ◆ Automatska gradnja hijerarhije
 - Pristup s vrha (engl. top-down approach)
 - Naći obujmicu za sve poligone, to je vrh stabla
 - Podijeliti na dva dijela, za svaki naći obujmicu i dodati u stablo
 - Ponoviti
 - Pristup s dna (engl. bottom-up approach)
 - Poligoni se ubacuju u stablo jedan po jedan, tako da minimalno uvećaju ukupnu obujmicu

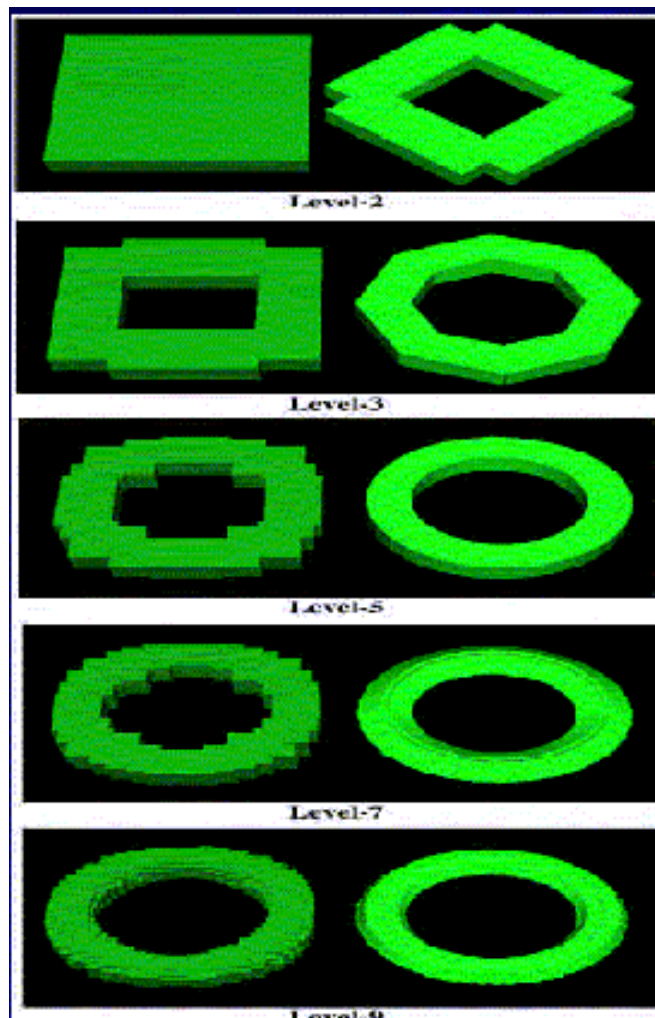


- ♦ Mogu se koristiti razne vrste obujmica
- ♦ Primjer: hijerarhija orijentiranih kvadara (engl. OBB-tree)



Hijerarhija kvadara - primjer torusa

Kvadar
paralelan s
osima
(AABB)



Orijentirani
kvadar
(OBB)

Hijerarhija kugli



Test sudara dviju hijerarhija

```
FindFirstHitCD( $A, B$ )
returns ( $\{TRUE, FALSE\}$ );
1:  if(not overlap( $A_{BV}, B_{BV}$ ) return FALSE;
2:  else if(isLeaf( $A$ ))
3:    if(isLeaf( $B$ ))
4:      for each triangle pair  $T_A \in A_c$  and  $T_B \in B_c$ 
5:        if(overlap( $T_A, T_B$ )) return TRUE;
6:    else
7:      for each child  $C_B \in B_c$ 
8:        FindFirstHitCD( $A, C_B$ )
9:  else
10:    for each child  $C_A \in A_c$ 
11:      FindFirstHitCD( $C_A, B$ )
12:  return FALSE;
```

Za više detalja

- ♦ ***<http://www.cs.unc.edu/~geom/collide/>***
- ♦ ***Real-Time Rendering***, T. Möller, E. Haines, A K Peters
- ♦ ***www.realtimerendering.com***
- ♦ ***www.magic-software.com***
- ♦ ***Graphics Gems, www.graphicsgems.org***