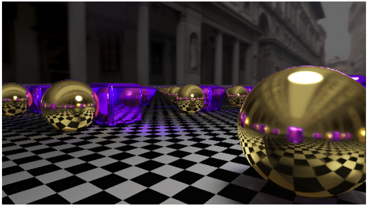
Távolságfüggvényekkel definiált felületek és vizualizációjuk - Quadric tracing

Kiglics Mátyás

Eötvös Loránd University

2021. november 10.

Tartalom



Bevezetés

Motiváció

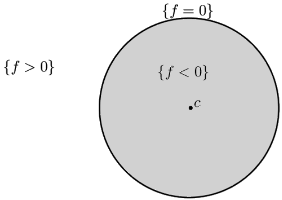
- Célok:
  - Tetszőleges színtér valósághű előállítása
  - Valós időben: gyorsan megjeleníteni
  - Nem valós időben: a lehető legszebb képet kapni
- Eszköz: sugárkövetés

Forrás: DIGIC Pictures

Előjeles távolságfüggvények (SDF)

- Euklideszi távolság: $d(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\|_2$
- Távolságfüggvény: $f(\boldsymbol{p}) = d(\boldsymbol{p}, f^{-1}(0))$
- Előjeles távolságfüggvény (**S**igned **D**istance **F**unction): $f \in C$, és $|f|$ távolságfüggvény
- A felület pontjainak halmaza: $\{\boldsymbol{p} \in \mathbb{R}^3 \mid f(\boldsymbol{p}) = 0\} = \{f = 0\}$

- Gömb SDF: $f(\boldsymbol{p}) = \|\boldsymbol{p} - \boldsymbol{c}\|_2 - r$

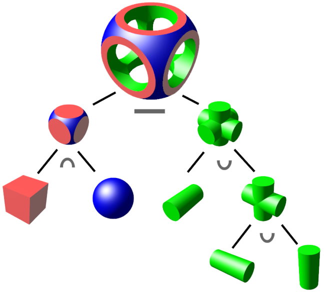


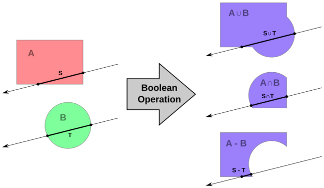
További példák

- Síklap: $f(\boldsymbol{p}) = \langle \boldsymbol{p}, \boldsymbol{n} \rangle + h$
- Téglatest (AAB): $f(\boldsymbol{p}) = \|max(\boldsymbol{q}, 0)\|_2 + min(max(\boldsymbol{q}_x, max(\boldsymbol{q}_y, \boldsymbol{q}_z)), 0)$ ahol $\boldsymbol{q} = |\boldsymbol{p}| - \boldsymbol{b}$
- Tórusz: $\left\| \left[\|\boldsymbol{p}_{xz}\|_2 - R, \boldsymbol{p}_y \right]^T \right\|_2 - r$

Összetett felületek

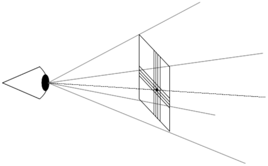
- Nem mindent tudunk egyszerű primitívekkel definiálni
- Használhatók pl. halmazműveletek
- $(f_1 \cup f_2)(\mathbf{p}) := min(f_1(\mathbf{p}), f_2(\mathbf{p}))$
- $(f_1 \cap f_2)(\mathbf{p}) := max(f_1(\mathbf{p}), f_2(\mathbf{p}))$
- $(f_1 \setminus f_2)(\mathbf{p}) := max(f_1(\mathbf{p}), -f_2(\mathbf{p}))$
- Számos további műveletet lehet definiálni

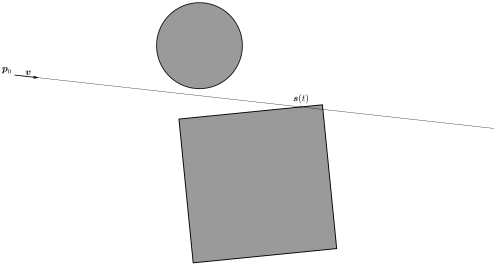




Ray tracing

- Minden pixel $\rightarrow$ szín
- Pixelenként egy sugár
- Sugár: $\boldsymbol{p}_0 + t \cdot \boldsymbol{v} \qquad (\|\boldsymbol{v}\|_2 = 1, t \geq 0)$
- Feladat: keressük azt a legkisebb $t \geq 0$-t, amelyre $f(\boldsymbol{p}_0 + t \cdot \boldsymbol{v}) = 0$
- Egyenletmegoldás





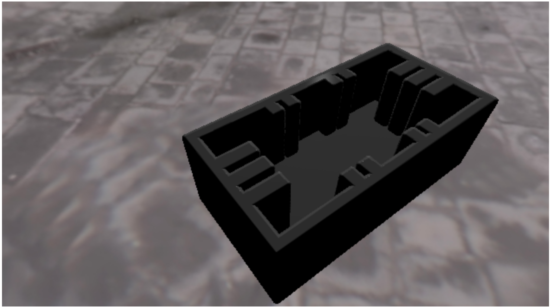
Ray tracing

```
float SDF(vec3 p)
{
    float r0 = dot(p, vec3(0,1,0));
    return r0;
}
```

Ray tracing

```
float SDF(vec3 p)
{
    float r0 = sdfBox(p, vec3(0.5,0.03,0.25));
    float r1 = sdfBox(p - vec3(0.4875,0.15,0), vec3(0.0125,0.15,0.25));
    float r2 = sdfBox(p - vec3(-0.4875,0.15,0), vec3(0.0125,0.15,0.25));
    r1 = min(r1,r2); //union 1
    float r3 = sdfBox(p - vec3(0,0.15,0.2375), vec3(0.5,0.15,0.0125));
    r1 = min(r1,r3); //union 2
    float r4 = sdfBox(p - vec3(0,0.15,-0.2375), vec3(0.5,0.15,0.0125));
    r1 = min(r1,r4); //union 3
    r0 = min(r0,r1); //union 1
    float r5 = sdfBox(p - vec3(-0.181818,0.15,0.2125), vec3(0.0181818,0.15,0.0375));
    float r6 = sdfBox(p - vec3(-0.181818,0.15,-0.2125), vec3(0.0181818,0.15,0.0375));
    r5 = min(r5,r6); //union 1
    float r7 = sdfBox(p - vec3(-0.109091,0.15,0.2125), vec3(0.0181818,0.15,0.0375));
    r5 = min(r5,r7); //union 1
    float r8 = sdfBox(p - vec3(-0.109091,0.15,-0.2125), vec3(0.0181818,0.15,0.0375));
    r5 = min(r5,r8); //union 2
    float r9 = sdfBox(p - vec3(0.109091,0.15,0.2125), vec3(0.0181818,0.15,0.0375));
    r5 = min(r5,r9); //union 1
    float r10 = sdfBox(p - vec3(0.109091,0.15,-0.2125), vec3(0.0181818,0.15,0.0375));
    r5 = min(r5,r10); //union 2
    float r11 = sdfBox(p - vec3(0.181818,0.15,0.2125), vec3(0.0181818,0.15,0.0375));
    r5 = min(r5,r11); //union 1
    float r12 = sdfBox(p - vec3(0.181818,0.15,-0.2125), vec3(0.0181818,0.15,0.0375));
    r5 = min(r5,r12); //union 2
    r0 = min(r0,r5); //union 2
    float r13 = sdfBox(p - vec3(0.425,0.15,-0.08), vec3(0.075,0.15,0.02));
    float r14 = sdfBox(p - vec3(-0.425,0.15,-0.08), vec3(0.075,0.15,0.02));
    r13 = min(r13,r14); //union 1
    float r15 = sdfBox(p - vec3(0.425,0.15,0), vec3(0.075,0.15,0.02));
    r13 = min(r13,r15); //union 1
    float r16 = sdfBox(p - vec3(-0.425,0.15,0), vec3(0.075,0.15,0.02));
    r13 = min(r13,r16); //union 2
    float r17 = sdfBox(p - vec3(0.425,0.15,0.08), vec3(0.075,0.15,0.02));
    r13 = min(r13,r17); //union 1
    float r18 = sdfBox(p - vec3(-0.425,0.15,0.08), vec3(0.075,0.15,0.02));
    r13 = min(r13,r18); //union 2
    r0 = min(r0,r13); //union 3
return r0;
}
```

Ray tracing

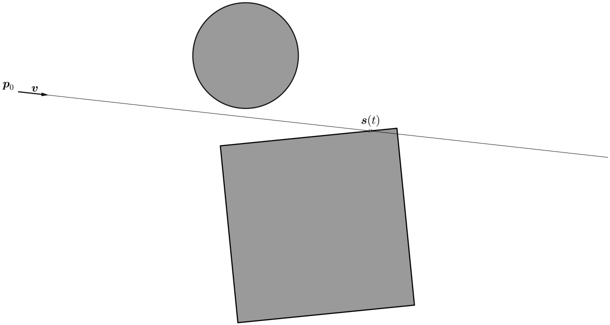


Sphere tracing

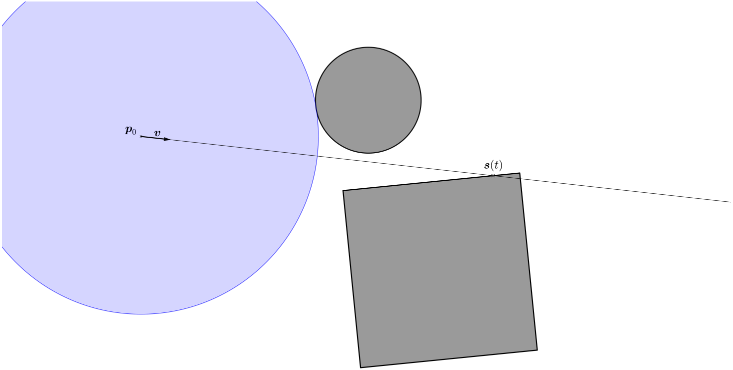
Sphere tracing

- Sugárkövető algoritmus
- John C. Hart 1995 [**hart**]
- A gyökkeresési problémára ad közelítő megoldást
- Általános, minden SDF-re működik

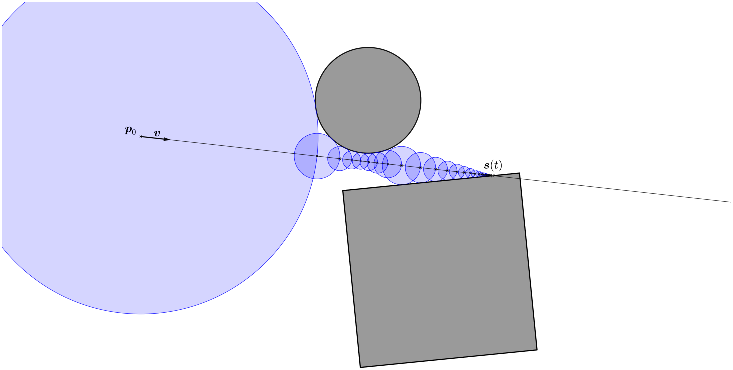
Sphere tracing



Sphere tracing



Sphere tracing



Sphere tracing

- Felület közelében kis lépések

Forrás: `http://shadertoy.com/user/iq`

Forrás: Wikipedia

Sphere tracing

- Felület közelében kis lépések
- Minden lépésben SDF kiértékelés

Forrás: `http://shadertoy.com/user/iq`

Forrás: Wikipedia

Sphere tracing

- Felület közelében kis lépések
- Minden lépésben SDF kiértékelés
- 800x600 pixel és 60 maximális lépés esetén akár $\sim$ 30 millió

Forrás: `http://shadertoy.com/user/iq`

Forrás: Wikipedia

Sphere tracing

- Felület közelében kis lépések
- Minden lépésben SDF kiértékelés
- 800x600 pixel és 60 maximális lépés esetén akár $\sim$ 30 millió
- Mindez 1 képkocka

Forrás: `http://shadertoy.com/user/iq`

Forrás: Wikipedia

Quadric tracing

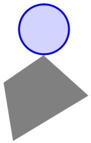
Sphere tracing gyorsítása

- Probléma: túl sok SDF kiértékelés
- Megoldás: növeljük a lépések nagyságát
- Korábbi eredmények: B. Keinert [**keinert**], Cs. Bálint, G. Valasek [**EG2018**]
- Heurisztikák

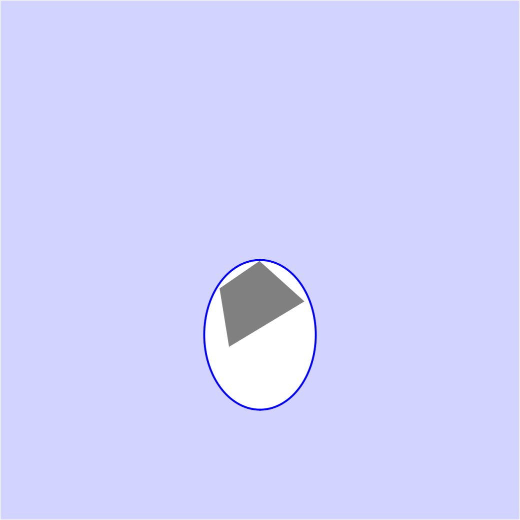
Quadric tracing [quadric]

- Általánosítási kísérlet
- Cél: nagyobb térfogatú térrészt keresni
- Unbounding sphere helyett unbounding quadric
- Akár végtelen nagyságú is lehet

Unbounding quadric példa

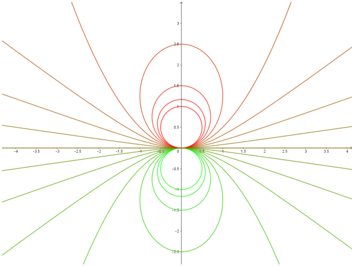


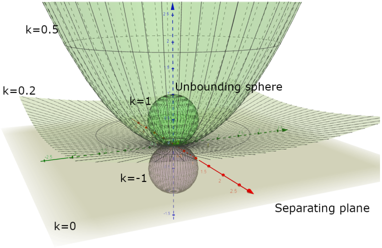
Unbounding quadric példa



Paraméteres másodrendű forgásfelület

- Követelmények:
  - Egyetlen véges paraméterrel leírható ($k \in [-1, 1]$)
  - $\forall k \in [-1, 1]$ esetén a felület tartalmazza az origót
  - $k = 0$: sík, $|k| = 1$: gömb, $|k| = \frac{1}{2}$: paraboloid
  - k függvényében folytonos





Implicit egyenlet

- Legegyszerűbb implicit egyenlettel kifejezni, először síkban
- Kúpszelet általános alakja: $Ax^2 + By^2 + Cy + Dxy + Ex + F = 0$

Implicit egyenlet

- Legegyszerűbb implicit egyenlettel kifejezni, először síkban
- Kúpszelet általános alakja: $Ax^2 + By^2 + Cy + Dxy + Ex + F = 0$
- Elég három együttható, a többi legyen 0

Implicit egyenlet

- Legegyszerűbb implicit egyenlettel kifejezni, először síkban
- Kúpszelet általános alakja: $Ax^2 + By^2 + Cy + Dxy + Ex + F = 0$
- Elég három együttható, a többi legyen 0
- Egyszerűsített forma: $Ax^2 + By^2 + Cy = 0$

Implicit egyenlet

- Legegyszerűbb implicit egyenlettel kifejezni, először síkban
- Kúpszelet általános alakja: $Ax^2 + By^2 + Cy + Dxy + Ex + F = 0$
- Elég három együttható, a többi legyen 0
- Egyszerűsített forma: $Ax^2 + By^2 + Cy = 0$
- $A := A(k), B := B(k), C := C(k)$

Implicit egyenlet

- Legegyszerűbb implicit egyenlettel kifejezni, először síkban
- Kúpszelet általános alakja: $Ax^2 + By^2 + Cy + Dxy + Ex + F = 0$
- Elég három együttható, a többi legyen 0
- Egyszerűsített forma: $Ax^2 + By^2 + Cy = 0$
- $A := A(k), B := B(k), C := C(k)$
- Egy lehetséges megoldás:
  - $A(k) = k^2$
  - $B(k) = 2\left(|k| - \frac{1}{2}\right)$
  - $C(k) = -k$

Implicit egyenlet

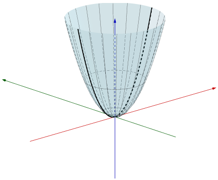
- Legegyszerűbb implicit egyenlettel kifejezni, először síkban
- Kúpszelet általános alakja: $Ax^2 + By^2 + Cy + Dxy + Ex + F = 0$
- Elég három együttható, a többi legyen 0
- Egyszerűsített forma: $Ax^2 + By^2 + Cy = 0$
- $A := A(k), B := B(k), C := C(k)$
- Egy lehetséges megoldás:
  - $A(k) = k^2$
  - $B(k) = 2\left(|k| - \frac{1}{2}\right)$
  - $C(k) = -k$
- $k^2 \cdot x^2 + 2\left(|k| - \frac{1}{2}\right) \cdot y^2 - k \cdot y = 0$

Térbeli kiterjesztés

- $k^2 \cdot x^2 + 2\left(|k| - \frac{1}{2}\right) \cdot y^2 - k \cdot y = 0$

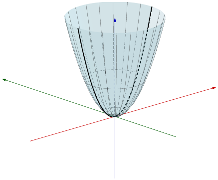
Térbeli kiterjesztés

- $k^2 \cdot x^2 + 2\left(|k| - \frac{1}{2}\right) \cdot y^2 - k \cdot y = 0$
- Térben ugyanez a görbe, függőleges tengely körül elforgatva



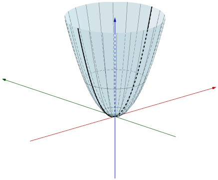
Térbeli kiterjesztés

- $k^2 \cdot x^2 + 2\left(|k| - \frac{1}{2}\right) \cdot y^2 - k \cdot y = 0$
- Térben ugyanez a görbe, függőleges tengely körül elforgatva
- Vízszintes koordináta $\rightarrow$ vízszintes körív
- Kör egyenlete: $(x^2 + y^2) - r^2 = 0$



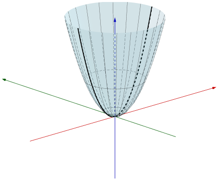
Térbeli kiterjesztés

- $k^2 \cdot x^2 + 2\left(|k| - \frac{1}{2}\right) \cdot y^2 - k \cdot y = 0$
- Térben ugyanez a görbe, függőleges tengely körül elforgatva
- Vízszintes koordináta $\rightarrow$ vízszintes körív
- Kör egyenlete: $(x^2 + y^2) - r^2 = 0$
- $k^2 \cdot (x^2 + y^2) + 2\left(|k| - \frac{1}{2}\right) \cdot z^2 - k \cdot z = 0$



Térbeli kiterjesztés

- $k^2 \cdot x^2 + 2\left(|k| - \frac{1}{2}\right) \cdot y^2 - k \cdot y = 0$
- Térben ugyanez a görbe, függőleges tengely körül elforgatva
- Vízszintes koordináta $\rightarrow$ vízszintes körív
- Kör egyenlete: $(x^2 + y^2) - r^2 = 0$
- $k^2 \cdot (x^2 + y^2) + 2\left(|k| - \frac{1}{2}\right) \cdot z^2 - k \cdot z = 0$
- Sugár-forgásfelület metszéshez behelyettesítés



Sugár-quadric metszés

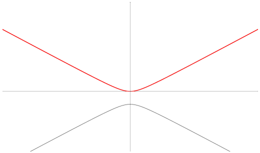
- Emlékeztető: $\boldsymbol{p}_0 + t\boldsymbol{v}$
- Legyen $\boldsymbol{p}_0 = \left[x_0, y_0, z_0\right]^T$
- Ekkor az egyenletbe sugarat behelyettesítve:
  $A \cdot ((x_0 + t \cdot \boldsymbol{v}_x)^2 + (y_0 + t \cdot \boldsymbol{v}_y)^2) + B \cdot (z_0 + t \cdot \boldsymbol{v}_z)^2 + C \cdot (z_0 + t \cdot \boldsymbol{v}_z) = 0$
- t-re rendezve másodfokú egyenlet:

$$a = A \cdot (\boldsymbol{v}_x^2 + \boldsymbol{v}_y^2) + B \cdot \boldsymbol{v}_z^2$$
$$b = 2A \cdot (x_0 \boldsymbol{v}_x + y_0 \boldsymbol{v}_y) + 2B \cdot z_0 \boldsymbol{v}_z + C \cdot \boldsymbol{v}_z$$
$$c = A \cdot (x_0^2 + y_0^2) + B \cdot z_0^2 + C \cdot z_0$$

- 2 megoldás (t_1, t_2), legyen $D < 0$ esetén $t_1 = -\infty, t_2 = \infty$

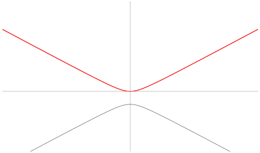
Hiperboloid kérdése

- Ha $0 < |k| < \frac{1}{2}$, a felület hiperboloid alakú



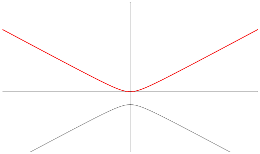
Hiperboloid kérdése

- Ha $0 < |k| < \frac{1}{2}$, a felület hiperboloid alakú
- Csak az egyik ágával foglalkozunk



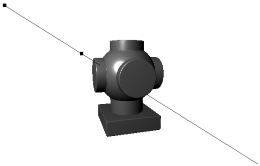
Hiperboloid kérdése

- Ha $0 < |k| < \frac{1}{2}$, a felület hiperboloid alakú
- Csak az egyik ágával foglalkozunk
- $(\boldsymbol{p}_0 + t_1 \boldsymbol{v})_z \cdot k < 0 \Rightarrow t_1 := -\infty$
- $(\boldsymbol{p}_0 + t_2 \boldsymbol{v})_z \cdot k < 0 \Rightarrow t_2 := \infty$



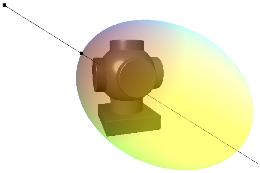
Quadric előállítása

- Origó és függőleges tengely nem praktikus
- Keressük a $\boldsymbol{p}$ pontban az optimális quadricot
- Innen $\boldsymbol{w}$ irányban van a felület
- Legyen az alappont $\boldsymbol{p} + (1-\epsilon) f(\boldsymbol{p}) \cdot \boldsymbol{w}$



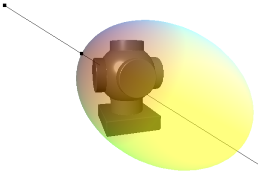
Quadric előállítása

- Origó és függőleges tengely nem praktikus
- Keressük a $\boldsymbol{p}$ pontban az optimális quadricot
- Innen $\boldsymbol{w}$ irányban van a felület
- Legyen az alappont $\boldsymbol{p} + (1 - \epsilon) f(\boldsymbol{p}) \cdot \boldsymbol{w}$
- Számoljuk itt ki a legjobb k paramétert



Quadric előállítása

- Origó és függőleges tengely nem praktikus
- Keressük a $\boldsymbol{p}$ pontban az optimális quadricot
- Innen $\boldsymbol{w}$ irányban van a felület
- Legyen az alappont $\boldsymbol{p} + (1 - \epsilon) f(\boldsymbol{p}) \cdot \boldsymbol{w}$
- Számoljuk itt ki a legjobb k paramétert
- Számításigényes!

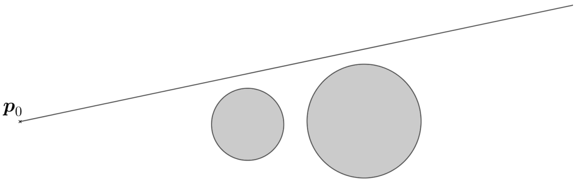


Előfeldolgozás

- Renderelés előtt 1x
- 3D-s diszkrét rács
- Rácspontokban $\boldsymbol{w}, f(\boldsymbol{p})$ és k
- 5 db lebegőpontos szám $\rightarrow$ rácsmérettől függő memóriaigény

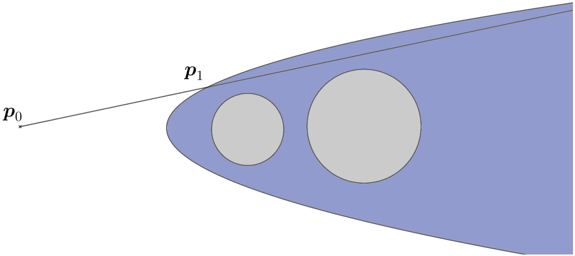
Quadric tracing

- Minden lépésben:
  1. Megkeressük a legközelebbi rácspontot
  2. Kiolvassuk a quardic paramétereit
  3. Kiszámoljuk a metszést
  4. Előrelépünk
- Ha nem tudunk lépni $\rightarrow$ enhanced sphere tracing



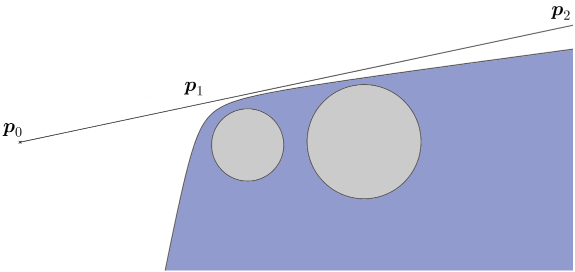
Quadric tracing

- Minden lépésben:
  1. Megkeressük a legközelebbi rácspontot
  2. Kiolvassuk a quardic paramétereit
  3. Kiszámoljuk a metszést
  4. Előrelépünk
- Ha nem tudunk lépni $\rightarrow$ enhanced sphere tracing



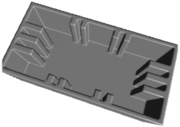
Quadric tracing

- Minden lépésben:
  1. Megkeressük a legközelebbi rácspontot
  2. Kiolvassuk a quardic paramétereit
  3. Kiszámoljuk a metszést
  4. Előrelépünk
- Ha nem tudunk lépni $\rightarrow$ enhanced sphere tracing

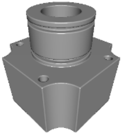


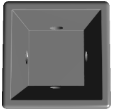
Eredmények

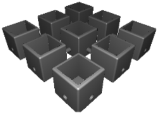

Model 0

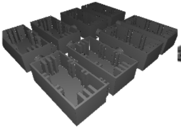

Model 1


Model 2


Model 3

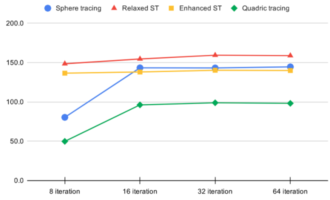

Model 4


Model 5

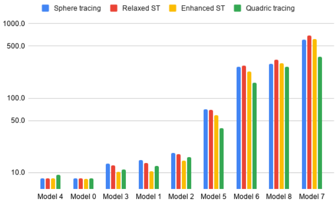

Model 6


Model 7

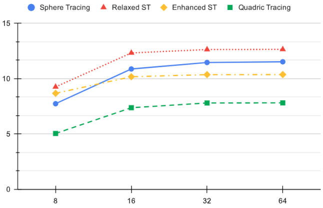
Eredmények



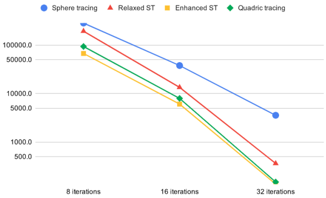
Átlagos relatív rajzolási idő (ms/frame)



Átlagos rajzolási idő (ms)



Átlagos SDF kiértékelések száma



Nem konvergált sugarak száma

Eredmények - Összefoglalás

- Iterációszám csökkent a többi algoritmushoz képest
- Bonyolultabb függvényeken hatékonyabb
- Véges térfogatú felületek megjelenítésére használható

További kutatások

Quadric tracing

- Teszteltük nem véges felületeken, path tracerbe implementálva

Quadric tracing

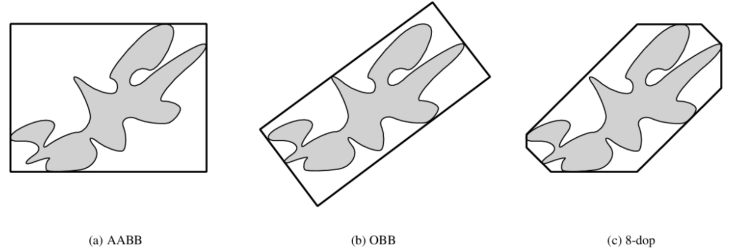
- Teszteltük nem véges felületeken, path tracerbe implementálva
- Enhanced sphere tracing hatékonyabb

Quadric tracing

- Teszteltük nem véges felületeken, path tracerbe implementálva
- Enhanced sphere tracing hatékonyabb
- További optimalizációkat tervezünk

k-DOP

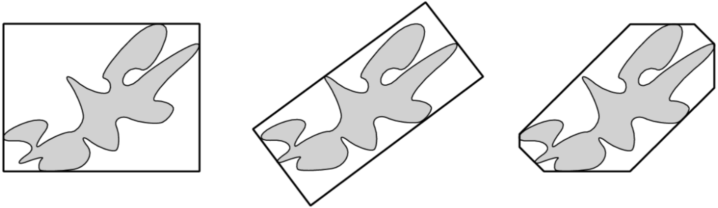
- Quadric helyett konvex politóp
- k-DOP (**D**iscrete **O**rientation **P**olytope) [**kdop**]



Forrás: **kdop**

k-DOP

- Quadric helyett konvex politóp
- k-DOP (**D**iscrete **O**rientation **P**olytope) [**kdop**]
- Eredetileg ütközésvizsgálatra használják

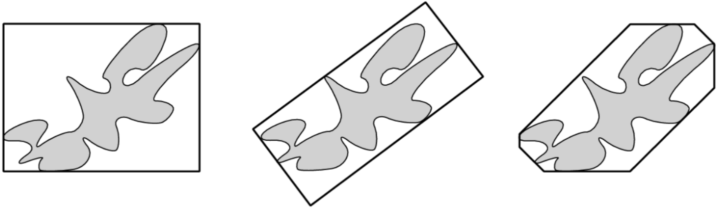


(a) AABB (b) OBB (c) 8-dop

Forrás: **kdop**

k-DOP

- Quadric helyett konvex politóp
- k-DOP (**D**iscrete **O**rientation **P**olytope) [**kdop**]
- Eredetileg ütközésvizsgálatra használják
- Bounding k-DOP triviális

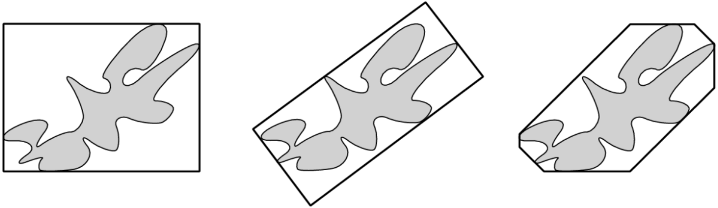


(a) AABB (b) OBB (c) 8-dop

Forrás: **kdop**

k-DOP

- Quadric helyett konvex politóp
- k-DOP (**D**iscrete **O**rientation **P**olytope) [**kdop**]
- Eredetileg ütközésvizsgálatra használják
- Bounding k-DOP triviális
- Unbounding egyáltalán nem (ráadásul 3D kell)



(a) AABB (b) OBB (c) 8-dop

Forrás: **kdop**

Köszönöm a figyelmet!