

Izofelületek megjelenítése

Tetraéder hálóval definiált modellek megjelenítése modern
grafikus kártyákon

Készítette: Szennai István

Mi az az izofelület?

- Izogörbék 3D-s analógiája
- $F(x, y, z) = c$ összes megoldása (c az izoérték)
- F -nek egy mintavételezése ismert (skalármező)
- Feladat: rekonstruálni és megjeleníteni

1	2	2	2	2	1
2	3	4	4	3	2
2	4	5	5	4	2
2	4	5	5	4	2
2	3	4	4	3	2
1	2	2	2	2	1



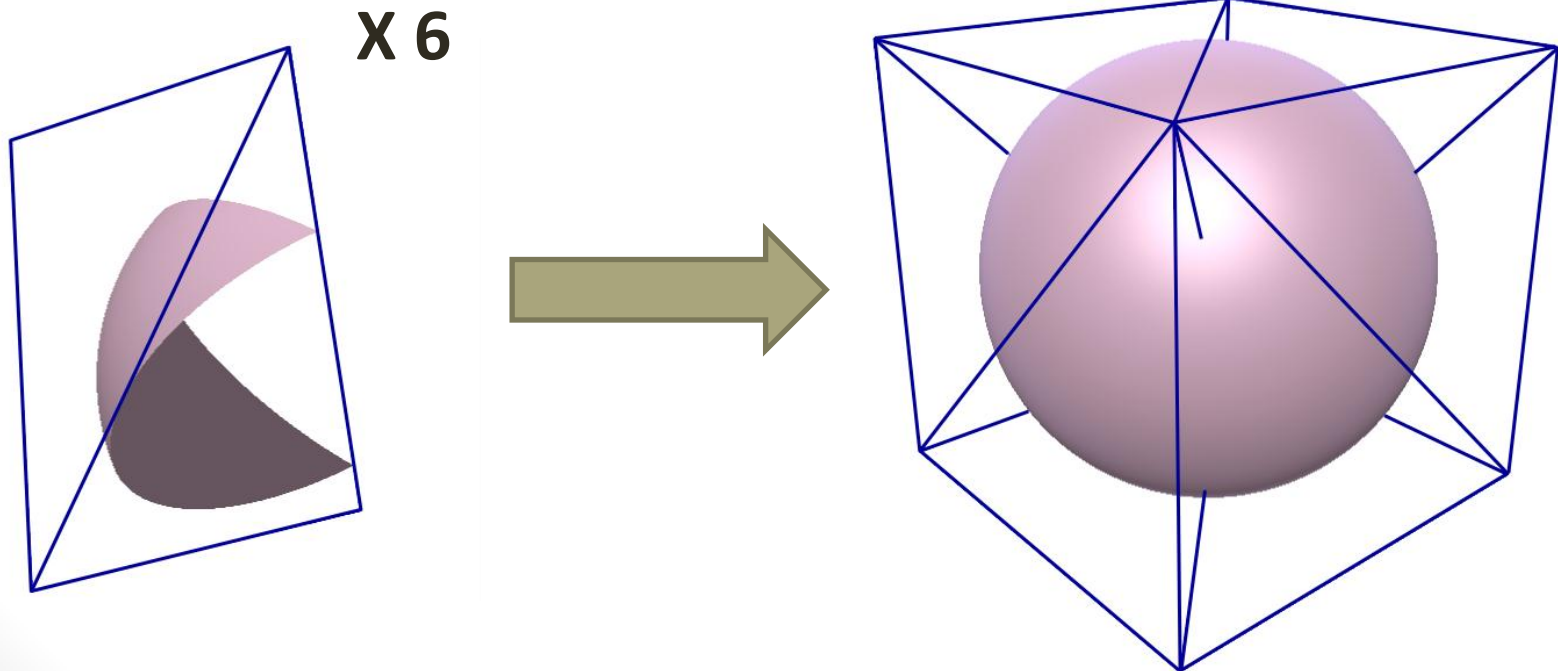
1	2	2	2	2	1
2	3	4	4	3	2
2	4	5	5	4	2
2	4	5	5	4	2
2	3	4	4	3	2
1	2	2	2	2	1

Modellek építése

- Általában marching cubes algoritmussal
 - Háromszöghálót készít
 - Gyors megjelenítés
 - Ugyanis a hardver erre van tervezve
- Én is egy létező, de viszonylag új módszert választottam
 - Algebrai felületdarabok
 - Tetraéderekre megszorítva
 - Ezekből egy spline

Algebrai felületdarabok

- $P(x, y, z) = 0$ polinom összes megoldása
- Egy tetraéderen belül
- Hasonlóan építhetők velük spline-ok, mint \mathbb{R} -ben



Tetraéderre megszorítás

- Bernstein-Bézier alak

$$\sum_{i+j+k+l=n} w_{ijkl} \binom{n}{ijkl} r^i s^j t^k u^l$$

- Ha (r, s, t, u) eleme $[0, 1]^4$, akkor a tetrán belül van
- Ha még a BB alak itt 0, akkor a felületnek is része
- n a polinom foka (nálam 2)
- w_{ijkl} Bézier súlyok (a.k.a. Bernstein polinom együtthatói)

Megjelenítés

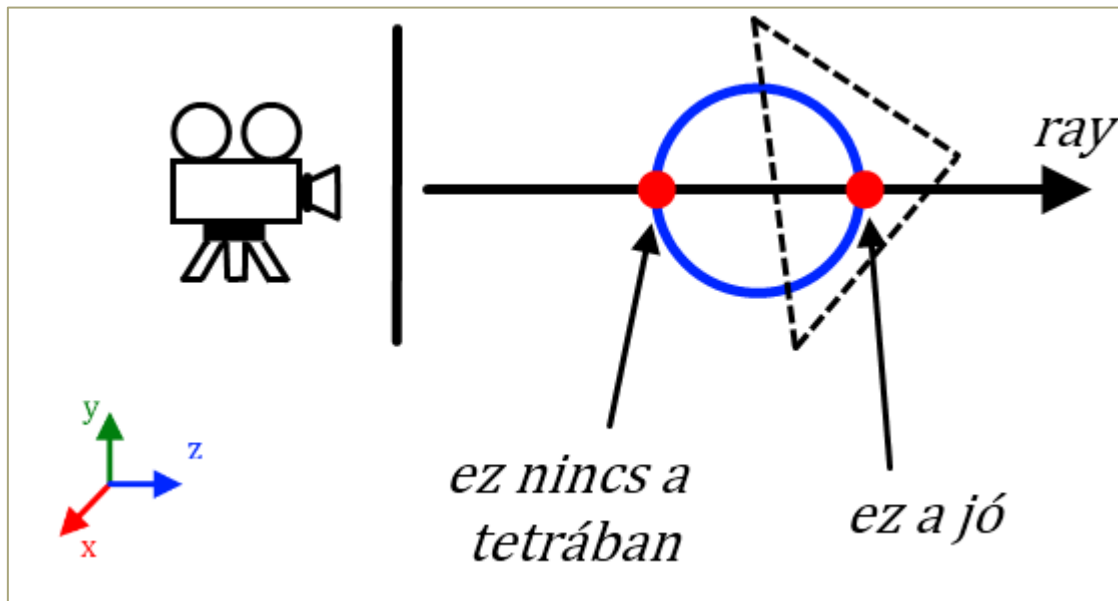
- *Loop & Blinn (2006)*: CPU + pixel shader (DX9)
- *Kloetzli, Olano & Rheinganz (2008)*: geometry shader (DX10)
- *Kalbe, Koch & Goesele (2009)*: OpenGL + CUDA

- *Én*: DX9, de teljesen a GPU-n
 - Utóbbihoz hasonlóan instancing
 - Rekonstrukció (egyelőre) offline
 - Egyszerű view frustum culling
 - LOD kezelés

- Az elsőnél jobb, a többinél nem feltétlenül
- Más perspektíva: gamedev

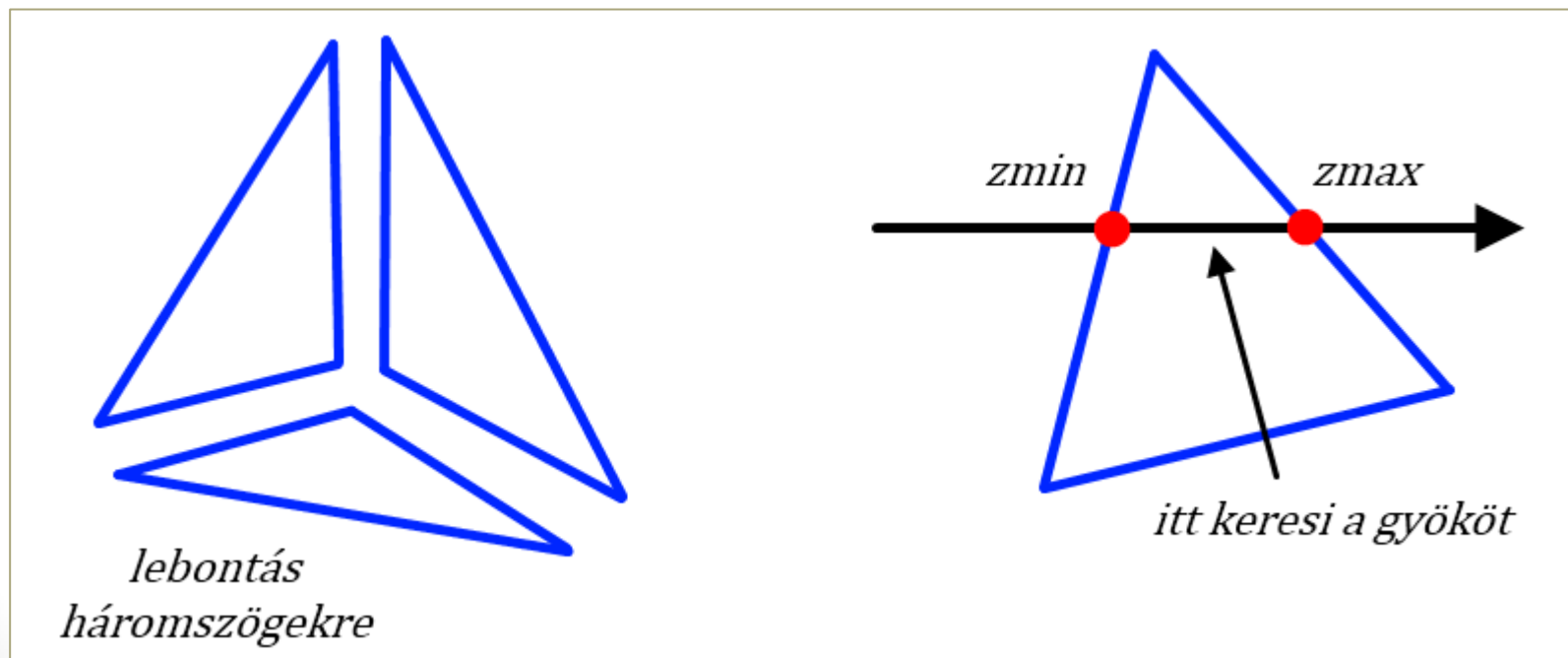
Hogyan?

- *Loop* megközelítése: raycasting
 - Screen space-ben a primary ray-ek párhuzamosak
 - Tehát a polinomban x és y ismert (pixel koordináták)
 - z -t kell megtalálni (mélység)
 - Egyváltozós polinom gyökkeresése



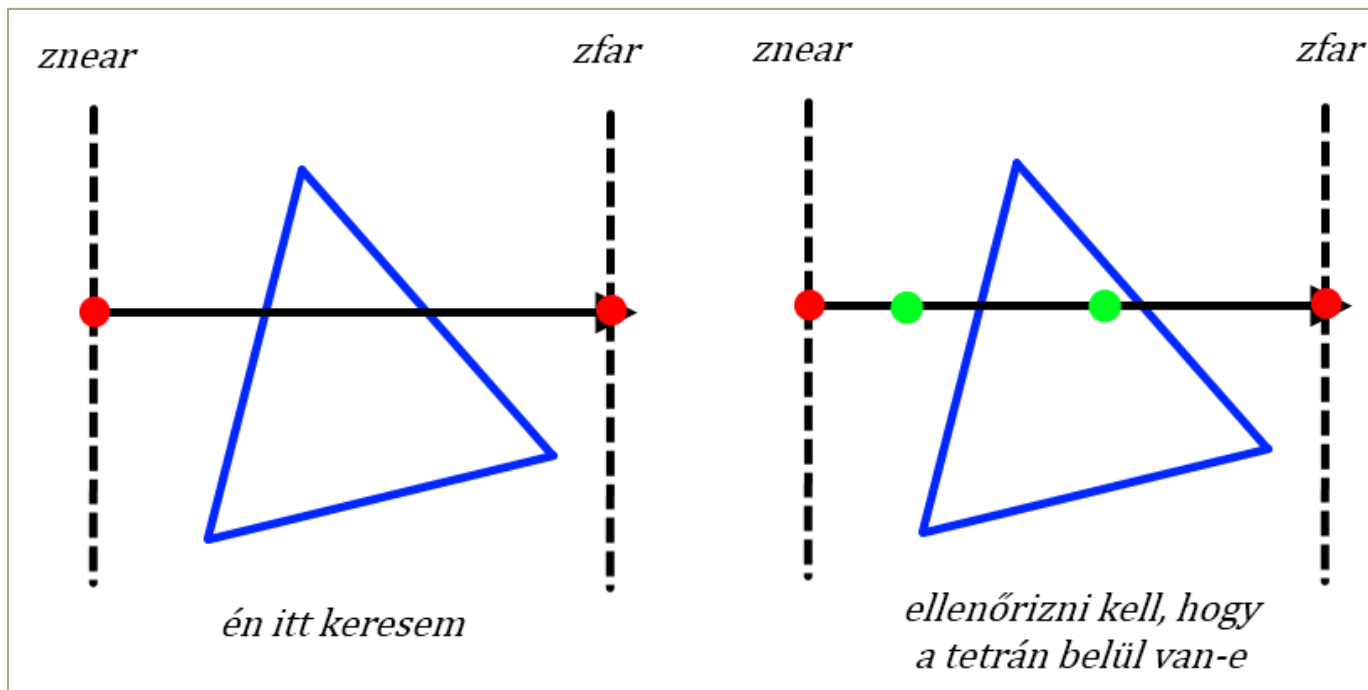
Trükközés

- *Loop* a ray-tetraéder metszetek között keresi a gyököt
- Le kell bontani a tetraédert screen space-beli Δ -ekre
- A CPU-n lassú, a vertex shader-ben meg sok elágazás



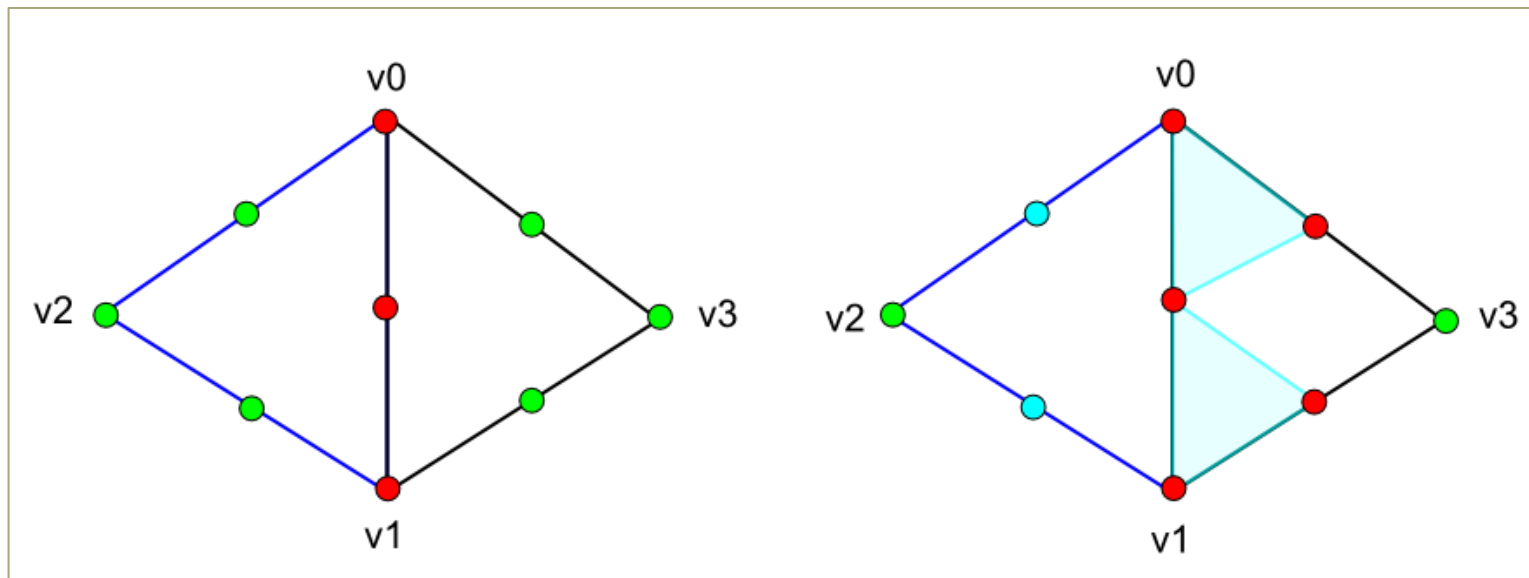
Trükközés reloaded

- Én a két vágósík között keresem a gyököt
- Baricentrikus koordináták döntenek el, hogy jó-e
- Így nem kell háromszögekre bontani
- Cserébe a pixel shader-be kell néhány elágazás



Spline építés

- BB alakban egyszerűen kifejezhető a folytonosság



- De általában ütköznek a feltételek
- Ezért átlagolni kell őket

Összefoglalás

- **Algebrai felületek előnyei:**

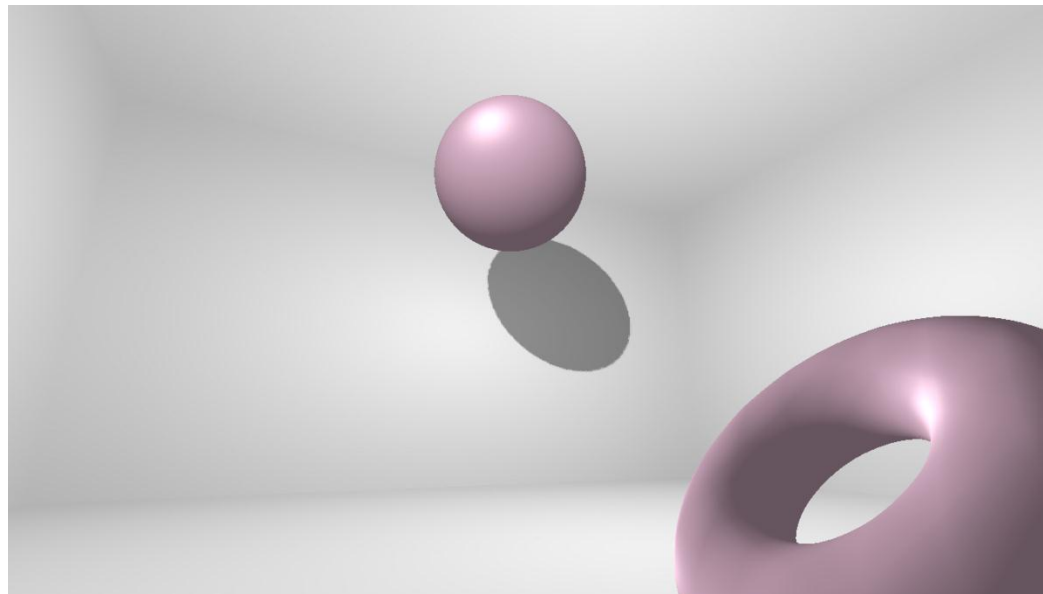
- Egyetlen skalármező több réteget tud tárolni a modellből
- Közel lehet menni, a felületek széle nem lesz „baltával faragott”
- A normálvektorokat közvetlenül a matematikai alakból lehet számolni, így realiztikusabb a megvilágítás
- Morphing (pl. lerp két polinomdarab között)

- **Hátrányok:**

- Sok tetraéderre erőforrásigényes (next-gen hardver?)
- Nem mindig folytonos az illeszkedés (átlagolás miatt)
- Nem trivi a textúrázás

Lehet-e használni játékokban?

- Igen, sőt akár már ma is!
 - LOD kezelés hasonló
 - Animálható
 - Könnyen integrálható grafikus motorokba



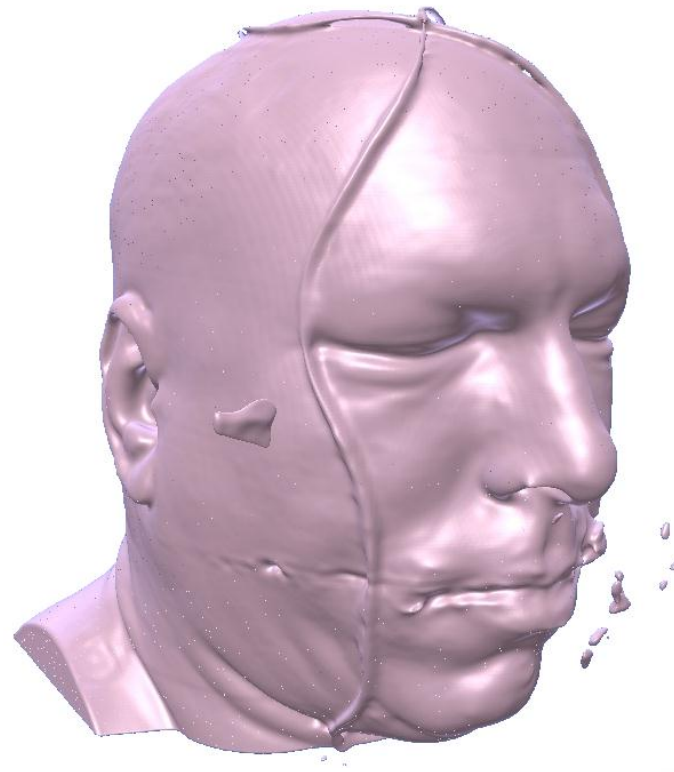
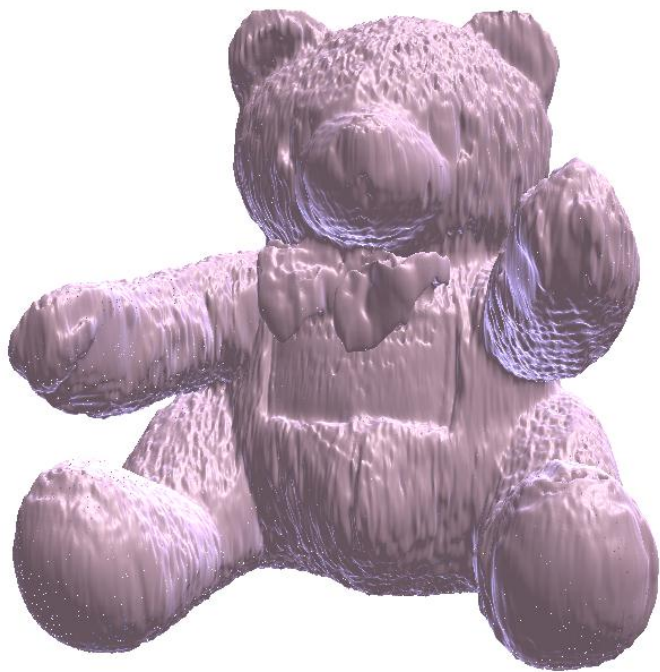
Fizika + real-time shadows izofelületekkel

Néhány futási eredmény

- Frames per second-ban
- Az üres tetraédereket kidobva
- 1360x768-as felbontáson
- ASUS ENGTS 250 videókártyán

Felület neve	Tetraéderek száma	CPU + DX9	GS + DX10	GPU + DX9
Torus	944	130	700	1030
Chmutov	95.000	1	26	38
Heart	109.000	1	25	50
Foot	872.000	0	5	9
Vismale	3.500.000	0	kifagy	3

Demo + kérdések



Köszönöm a figyelmet!

Irodalomjegyzék

- *Charles Loop, Jim Blinn:*
 - Real-Time GPU Rendering of Piecewise Algebraic Surfaces.
- *C. Rössl, F. Zeilfelder, G. Nürnberger, H.-P. Seidel:*
 - Reconstruction of Volume Data with Quadratic Super Splines.
- *John Kloetzli, Marc Olano, Penny Rheingans:*
 - Interactive Volume Isosurface Rendering Using BT Volumes.
- *Thomas Kalbe, Thomas Koch, Michael Goesele:*
 - High-Quality Rendering of Varying Isosurfaces with Cubic Trivariate C1-continuous Splines.