Számítógépes Grafika

Hajder Levente
hajder@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2019/2020. I. félév

Előadás

- IP-eSZG1E (órarendi kód:IP-eSZG1E/1)
- Tavaszi félév : keresztfélév
- Vizsga: Vizsgaidőszakban
  - Beugró: minimunkérdések: $\geq 70\%$
  - 2 db kidolgozott tétel
- Honlap: `http://cg.elte.hu/~hajder`

Előadás

- IP-eSZG1E (órarendi kód:IP-eSZG1E/1)
- Tavaszi félév : keresztfélév
- Vizsga: Vizsgaidőszakban
  - Beugró: minimunkérdések: $\geq 70\%$
  - 2 db kidolgozott tétel
- Honlap: `http://cg.elte.hu/~hajder`

Előadás

- IP-eSZG1E (órarendi kód:IP-eSZG1E/1)
- Tavaszi félév : keresztfélév
- Vizsga: Vizsgaidőszakban
  - Beugró: minimunkérdések: $\geq 70\%$
  - 2 db kidolgozott tétel
- Honlap: `http://cg.elte.hu/~hajder`

Előadás

- IP-eSZG1E (órarendi kód:IP-eSZG1E/1)
- Tavaszi félév : keresztfélév
- Vizsga: Vizsgaidőszakban
  - Beugró: minimunkérdések: $\geq$ 70%
  - 2 db kidolgozott tétel
- Honlap: `http://cg.elte.hu/~hajder`

Előadás

- IP-eSZG1E (órarendi kód:IP-eSZG1E/1)
- Tavaszi félév : keresztfélév
- Vizsga: Vizsgaidőszakban
  - Beugró: minimunkérdések: $\geq$ 70%
  - 2 db kidolgozott tétel
- Honlap: `http://cg.elte.hu/~hajder`

Előadás

- IP-eSZG1E (órarendi kód:IP-eSZG1E/1)
- Tavaszi félév : keresztfélév
- Vizsga: Vizsgaidőszakban
  - Beugró: minimunkérdések: $\geq$ 70%
  - 2 db kidolgozott tétel
- Honlap: `http://cg.elte.hu/~hajder`

Gyakorlatok

- IP-08ESZGG (órarendi kód: IP-eSZG1G/x)
- Az előadás és a gyakorlat nem előfeltételei egymásnak.
- Követelmény:
  - Órák látogatása
  - 2 db. házi feladat bemutatása (Processing-ben)
    - 2D-s
    - 3D-s

Tartalom

- Előadás
- Gyakorlat

1. Számítógépes grafika
   - Célja, feladata, területei
   - Az emberi szem működése
   - Megjelenítő eszközök
   - Raszter- és vektorgrafika
   - 3D grafika
2. Ajánlás
   - Programok
   - Honlapok
   - Könyvek

Számítógépes grafika

- Az informatika tudomány egy ága
- *Feladata:* Vizuális anyagok előállítása, elemzése és manipulálása (feldolgozás)

Számítógépes grafika

- Az informatika tudomány egy ága
- *Feladata:* Vizuális anyagok előállítása, elemzése és manipulálása (feldolgozás)

Képmanipulálás

Brandon Christopher Warren, flickr

Képmanipulálás

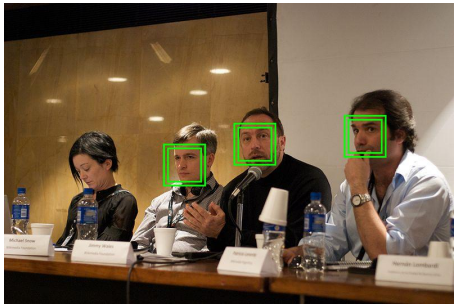
Brandon Christopher Warren, flickr

Nem része a félév anyagának

Képfeldolgozás és képelemzés

- Képfeldolgozás

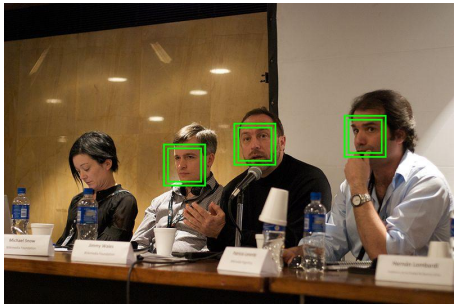
- Képelemzés



Képfeldolgozás és képelemzés

- Képfeldolgozás

- Képelemzés

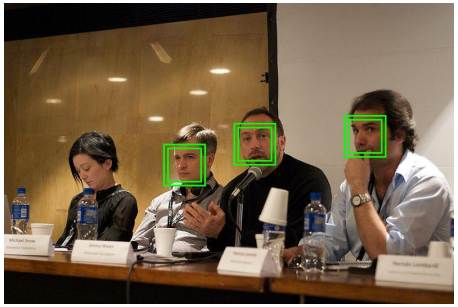


Beatrice Murch and OpenCV

Képfeldolgozás és képelemzés

- Képfeldolgozás

- Képelemzés



Beatrice Murch and OpenCV

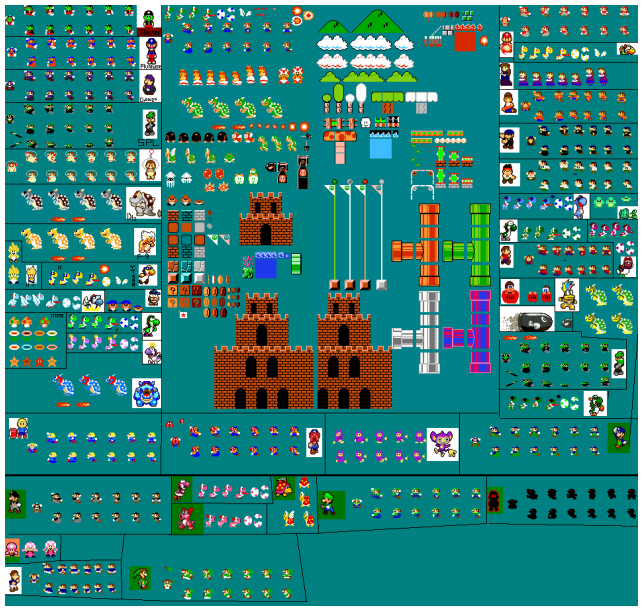
Képszintézis

- Cél: egy két- vagy háromdimenziós ”világról” képet előállítani
- Bemenet: a virtuális világ modellje → Kimenet: kép

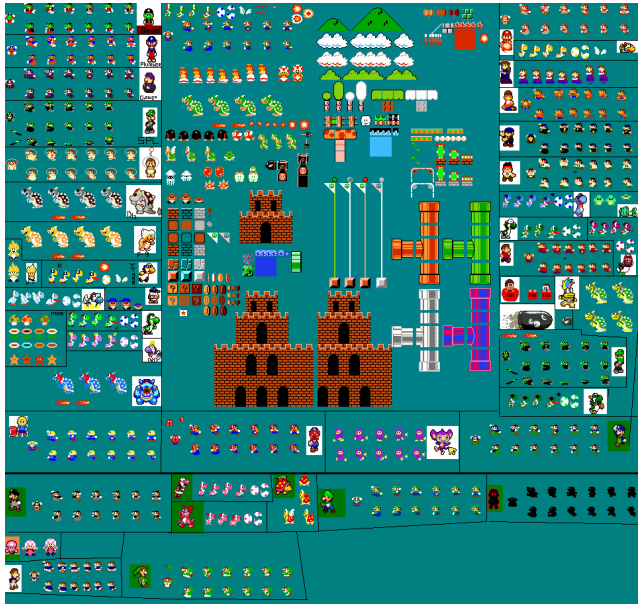
Képszintézis

- Cél: egy két- vagy háromdimenziós ”világról” képet előállítani
- Bemenet: a virtuális világ modellje $\rightarrow$ Kimenet: kép

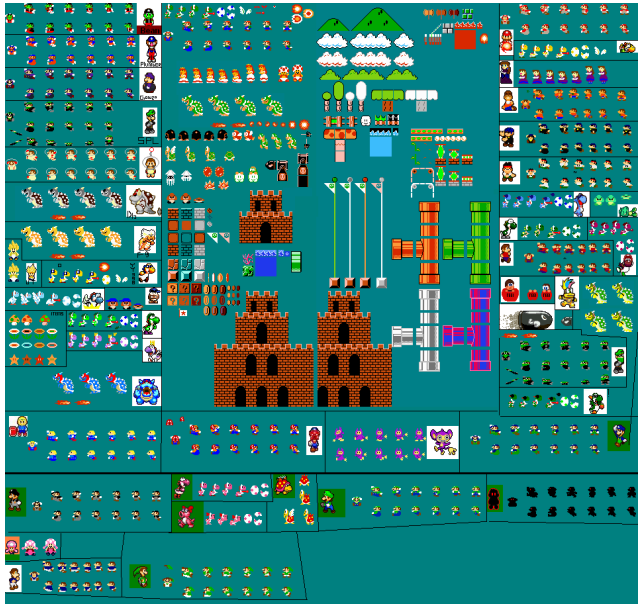
2D: Modell → Kép

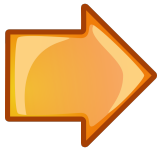


2D: Modell → Kép

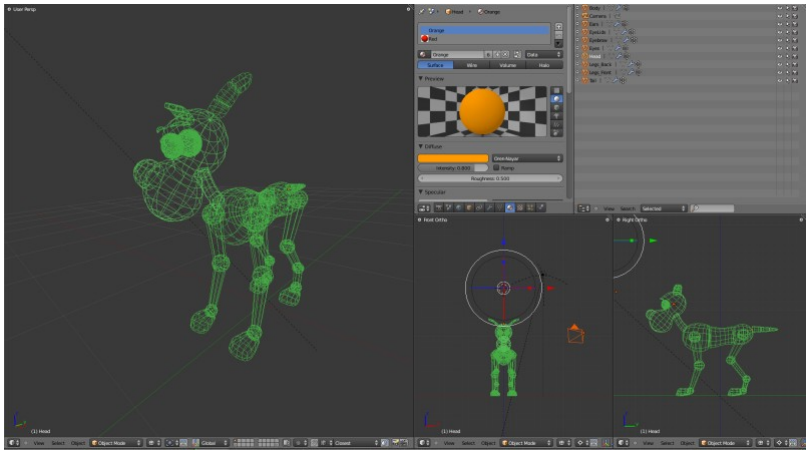


2D: Modell → Kép

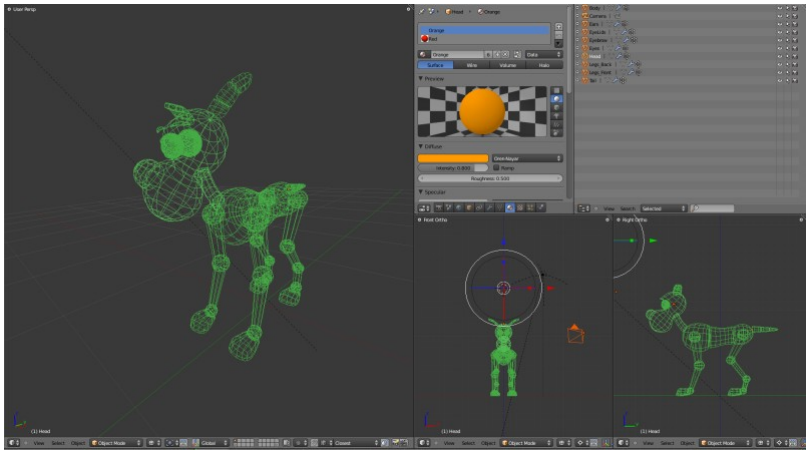




3D: Modell → Kép



3D: Modell → Kép



Képszintézis

Kérdések:

- Hogyan írjuk le a világot? - Modellezés
- Hogyan számítjuk ki a képet? - Algoritmusok
- Hogyan jelenítjük meg azt? - Eszközök

Képszintézis

Kérdések:

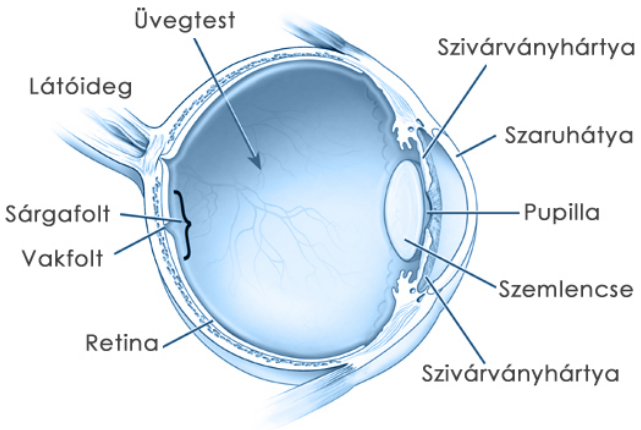
- Hogyan írjuk le a világot? - Modellezés
- Hogyan számítjuk ki a képet? - Algoritmusok
- Hogyan jelenítjük meg azt? - Eszközök

Képszintézis

Kérdések:

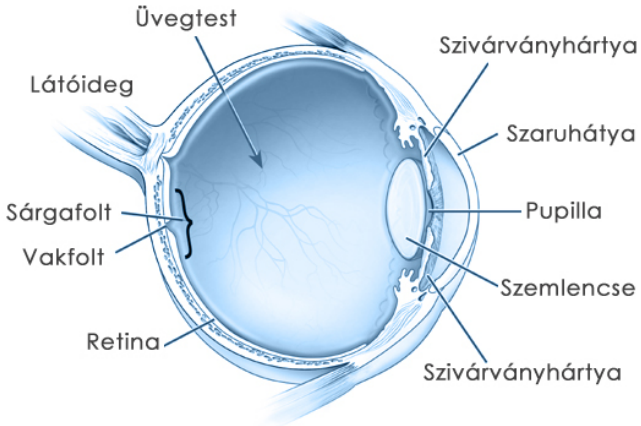
- Hogyan írjuk le a világot? - Modellezés
- Hogyan számítjuk ki a képet? - Algoritmusok
- Hogyan jelenítjük meg azt? - Eszközök

Emberi látás



- Szemgolyó elején az ”optika” van, hátul az érzékelők.
- Pupilla: a bejutó fény mennyiségét szabályozza. (Sötétben tágabb, világosban szükebb rést biztosít)
- Szemlencse: a fény optikai leképzését biztosítja a retinára

Emberi látás



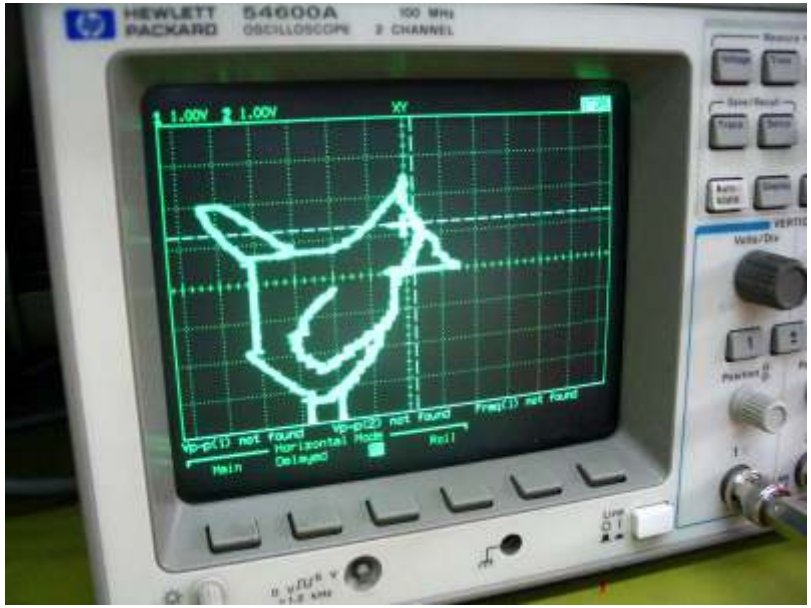
- Idegsejtek felosztása
  - Pálcika: gyenge fényben is lát, színeket nem érzékel
  - Csapok: színeket lát.
    - R csap: rövidhullám-érzékeny (420 nm, ibolya szín)
    - K csap: középhullám-érzékeny (530 nm, zöld szín)
    - H csap: hosszúhullám-érzékeny (560 nm, sárga szín)

A szem becsapása

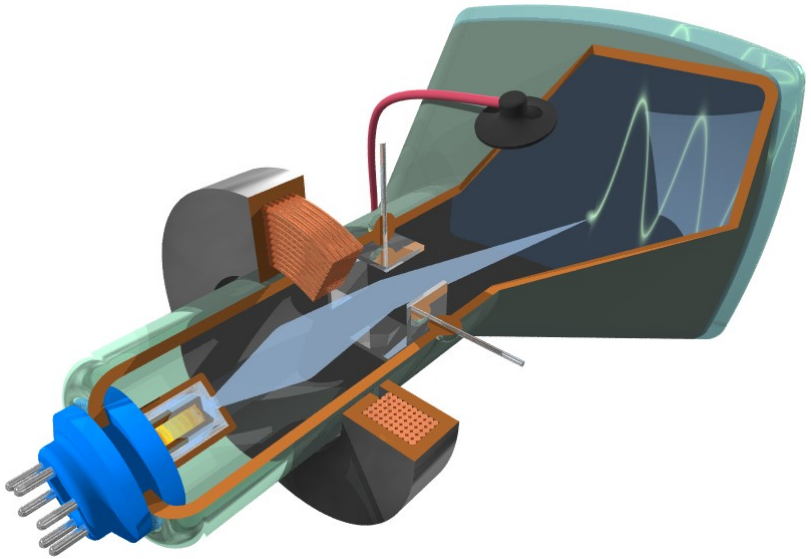
- A szem csapjai három különböző színt képesek látni: ibolya, zöld, sárga
- Számítógépes képmegjelenítők technikai okokból három színt jelenítenek meg: vörös, zöld, kék
- A szem ezzel jól becsapható.
  - A becsapás nem tökéletes, nem minden természetes szín jeleníthető meg a monitorokon!

Eszközök

Oszciloszkóp



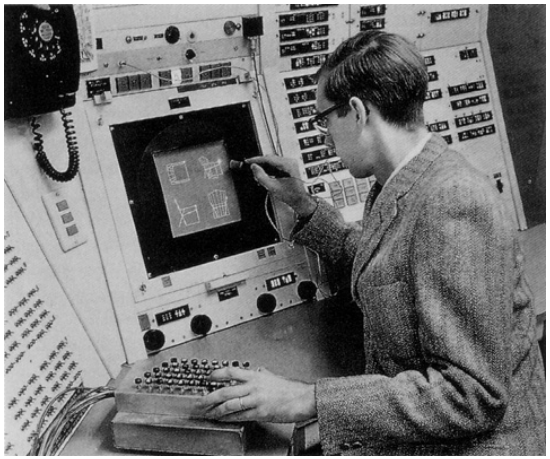
Felix E. Guerrero, flickr

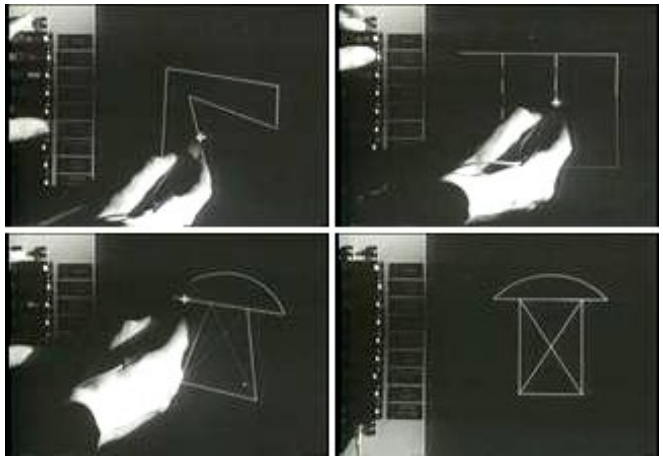


Soren Peo Pedersen, Wikipedia

Eszközök

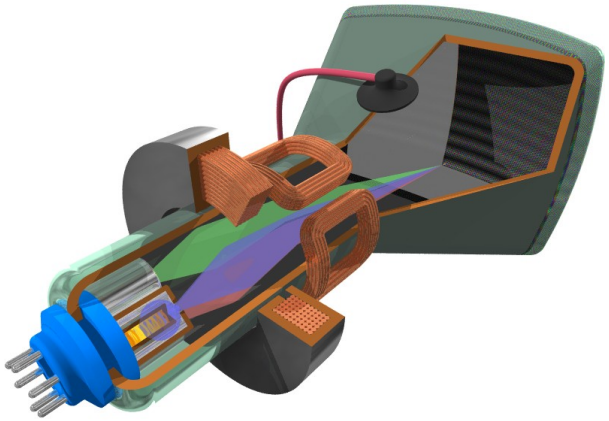
Sutherland - Sketchpad, 1963





- a CAD alkalmazások őse
- 1024x1024-es kijelző
- fényceruzával + 40 nyomógombbal volt vezérelhető
- bevezette a kényszer alapú rajzolást: vízszintes, függőleges, merőleges stb.

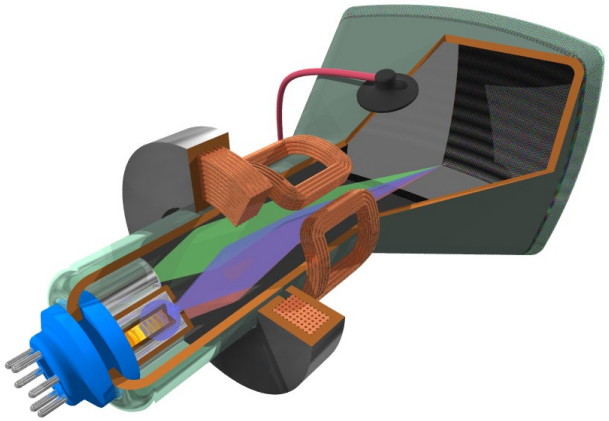
CRT (Cathod Ray Tube) monitor



Soren Peo Pedersen, Wikipedia

- A monitor felülete alatt fényre érzékeny foszforréteg található. (Színenként más fajta)
- Elektronok ”lövik” a foszfort, mire az fényt ad ki magából. A ”lövedék” irányát mégneses mező irányítja.

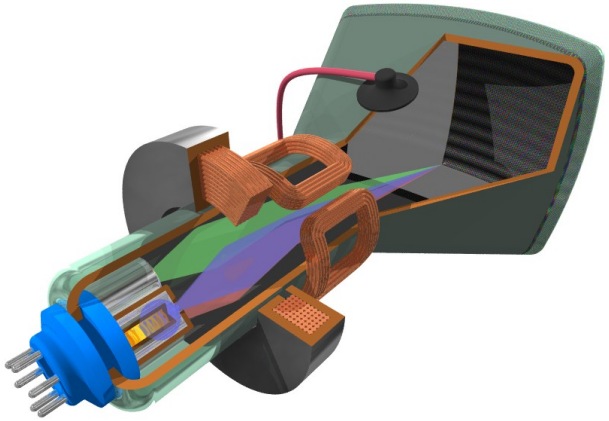
CRT (Cathod Ray Tube) monitor



Soren Peo Pedersen, Wikipedia

- A monitor felülete alatt fényre érzékeny foszforréteg található. (Színenként más fajta)
- Elektronok ”lövik” a foszfort, mire az fényt ad ki magából. A ”lövedék” irányát mégneses mező irányítja.

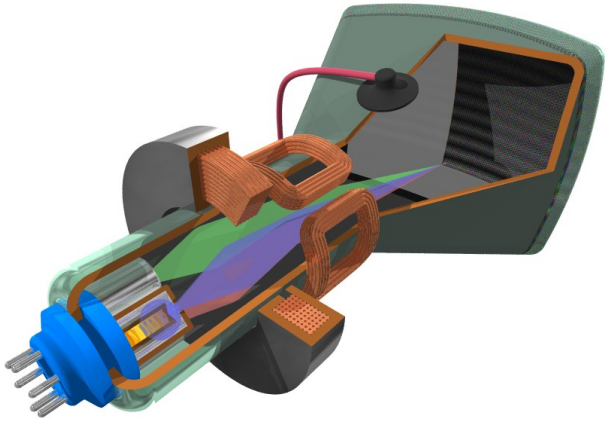
CRT (Cathod Ray Tube) monitor



Soren Peo Pedersen, Wikipedia

- Három színhez (vörös, zöld, kék) három elektronágyú tartozik.
  - Soronként rajzolja ki a képet, soron belül balról jobbra halad.
  - A világítás hamar elhalványul, ezért a pixelek villognak → szemkímélő, ha nagy frekvenciával villog.

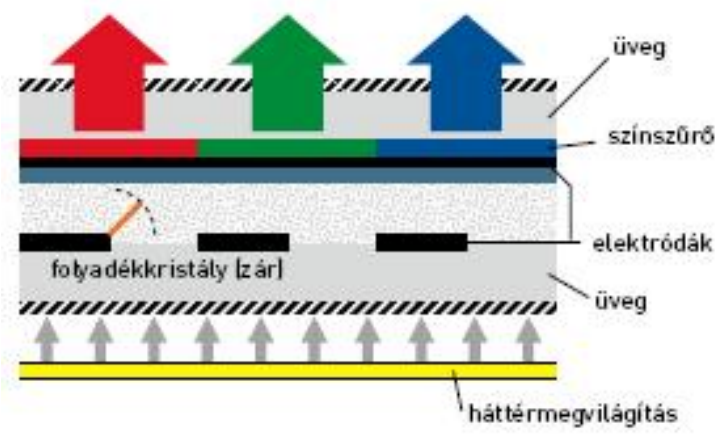
CRT (Cathod Ray Tube) monitor



Soren Peo Pedersen, Wikipedia

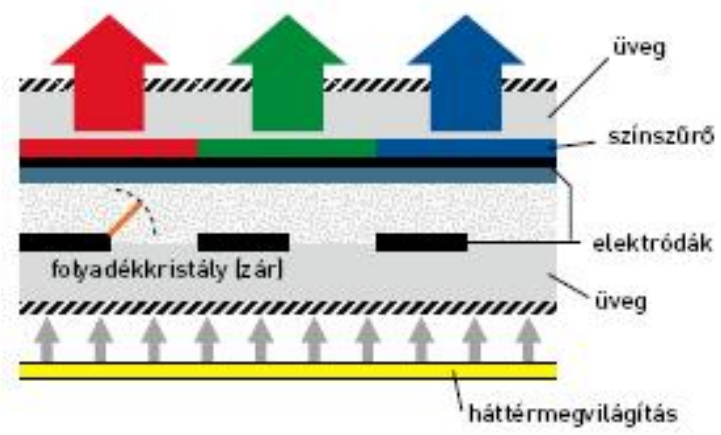
- Három színhez (vörös, zöld, kék) három elektronágyú tartozik.
  - Soronként rajzolja ki a képet, soron belül balról jobbra halad.
  - A világítás hamar elhalványul, ezért a pixelek villognak → szemkímélő, ha nagy frekvenciával villog.

LCD (folyadékkristályos) monitor



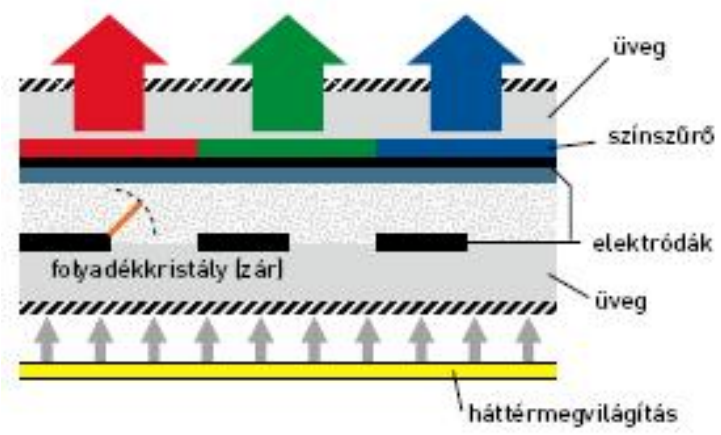
- Működése hasonló a fekete-fehér LCD kijelzőhöz (pl. óra kijelzője).
- Színek előállítása beépített színszűrőkkel
- Háttérvilágítás szükséges (passzív monitor), fény polarizálásával dolgozik.
  - LED-es monitor: háttérvilágítás LED-ekkel.

LCD (folyadékkristályos) monitor



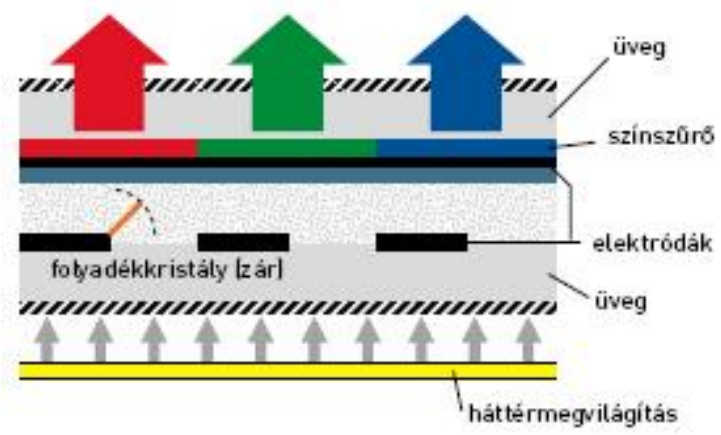
- Működése hasonló a fekete-fehér LCD kijelzőhöz (pl. óra kijelzője).
- Színek előállítása beépített színszűrőkkel
- Háttérvilágítás szükséges (passzív monitor), fény polarizálásával dolgozik.
  - LED-es monitor: háttérvilágítás LED-ekkel.

LCD (folyadékkristályos) monitor



- Működése hasonló a fekete-fehér LCD kijelzőhöz (pl. óra kijelzője).
- Színek előállítása beépített színszűrőkkel
- Háttérvilágítás szükséges (passzív monitor), fény polarizálásával dolgozik.
  - LED-es monitor: háttérvilágítás LED-ekkel.

LCD (folyadékkristályos) monitor



- Működése hasonló a fekete-fehér LCD kijelzőhöz (pl. óra kijelzője).
- Színek előállítása beépített színszűrőkkel
- Háttérvilágítás szükséges (passzív monitor), fény polarizálásával dolgozik.
  - LED-es monitor: háttérvilágítás LED-ekkel.

TFT (Thin Film Transistor) monitor

- Minden képpont egy önálló transzisztor. (aktív mátrix)
- Erős fénnyel (pl. nap) szemben gyenge a fényereje
- Laptopokban előszeretettel használják

TFT (Thin Film Transistor) monitor

- Minden képpont egy önálló transzisztor. (aktív mátrix)
- Erős fénnyel (pl. nap) szemben gyenge a fényereje
- Laptopokban előszeretettel használják

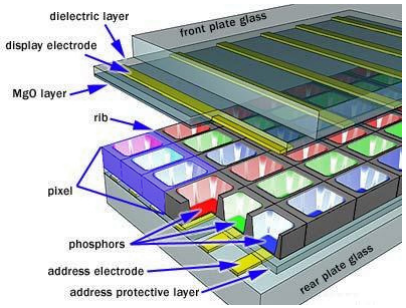
TFT (Thin Film Transistor) monitor

- Minden képpont egy önálló transzisztor. (aktív mátrix)
- Erős fénnyel (pl. nap) szemben gyenge a fényereje
- Laptopokban előszeretettel használják

TFT (Thin Film Transistor) monitor

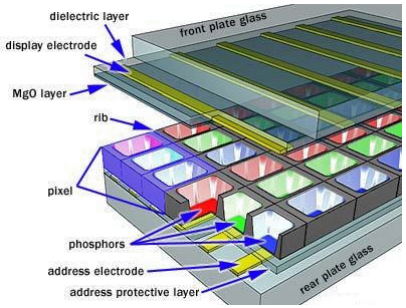
- Minden képpont egy önálló transzisztor. (aktív mátrix)
- Erős fénnyel (pl. nap) szemben gyenge a fényereje
- Laptopokban előszeretettel használják

PDP (Plazma Display Panel) monitor



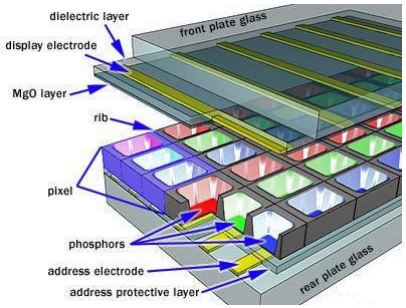
- Működési elve hasonló a CRT monitorokhoz.
- Gázok keverékének nagy UV-sugárzással kísért ionizációs kisülése készteti a képpont anyagát színes fény sugárzására
- Rendkívüli fényereje és kontrasztos képe van. Gyors a frissítési frekvenciája, ezért szép folyamatosak (és élesek) a mozgások.
- Hátránya a relatíve magas fogyasztás.

PDP (Plazma Display Panel) monitor



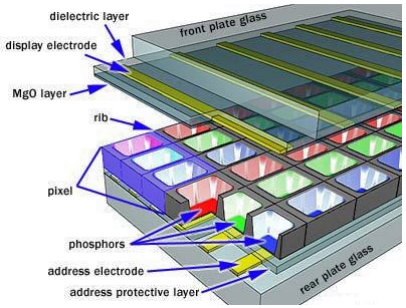
- Működési elve hasonló a CRT monitorokhoz.
- Gázok keverékének nagy UV-sugárzással kísért ionizációs kisülése készteti a képpont anyagát színes fény sugárzására
- Rendkívüli fényereje és kontrasztos képe van. Gyors a frissítési frekvenciája, ezért szép folyamatosak (és élesek) a mozgások.
- Hátránya a relatíve magas fogyasztás.

PDP (Plazma Display Panel) monitor



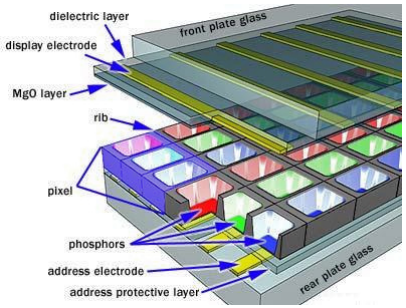
- Működési elve hasonló a CRT monitorokhoz.
- Gázok keverékének nagy UV-sugárzással kísért ionizációs kisülése készteti a képpont anyagát színes fény sugárzására
- Rendkívüli fényereje és kontrasztos képe van. Gyors a frissítési frekvenciája, ezért szép folyamatosak (és élesek) a mozgások.
- Hátránya a relatíve magas fogyasztás.

PDP (Plazma Display Panel) monitor



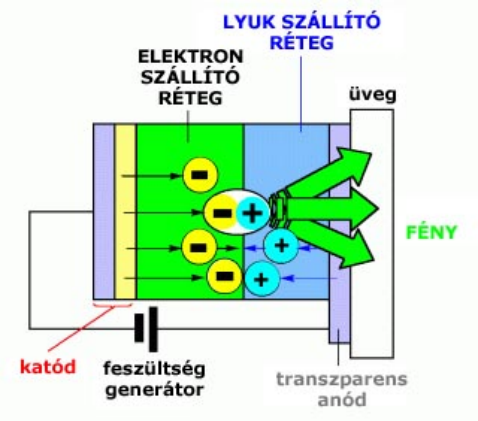
- Működési elve hasonló a CRT monitorokhoz.
- Gázok keverékének nagy UV-sugárzással kísért ionizációs kisülése készteti a képpont anyagát színes fény sugárzására
- Rendkívüli fényereje és kontrasztos képe van. Gyors a frissítési frekvenciája, ezért szép folyamatosak (és élesek) a mozgások.
- Hátránya a relatíve magas fogyasztás.

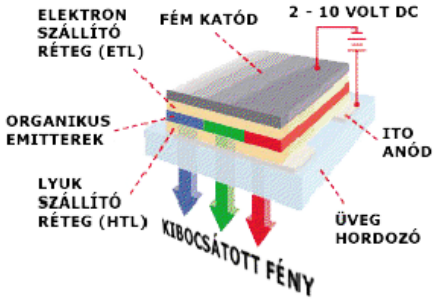
PDP (Plazma Display Panel) monitor



- Működési elve hasonló a CRT monitorokhoz.
- Gázok keverékének nagy UV-sugárzással kísért ionizációs kisülése készteti a képpont anyagát színes fény sugárzására
- Rendkívüli fényereje és kontrasztos képe van. Gyors a frissítési frekvenciája, ezért szép folyamatosak (és élesek) a mozgások.
- Hátránya a relatíve magas fogyasztás.

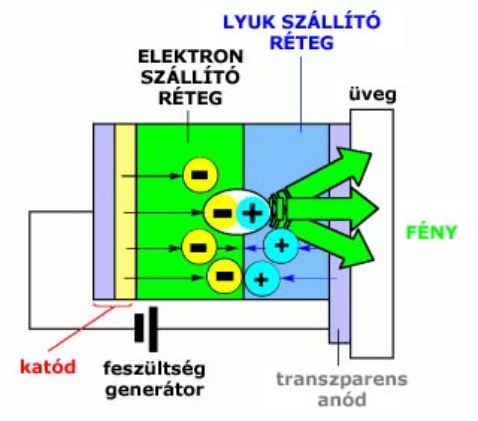
OLED (Organic Light-Emitting Diode) monitor

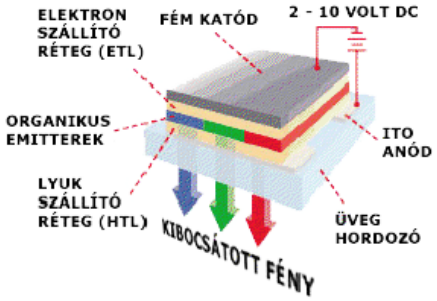




- Ha feszültséget kötünk a szerves anyagra, elektronok és ”lyukak” mozognak szemben egymással
- A találkozáskor energia szabadul fel. Ebből látható hullám lesz. A foton f frekvenciája (színe) a $E = hf$ összefüggéssel számítható, ahol E a foton energiája, h az ún. Planck állandó.

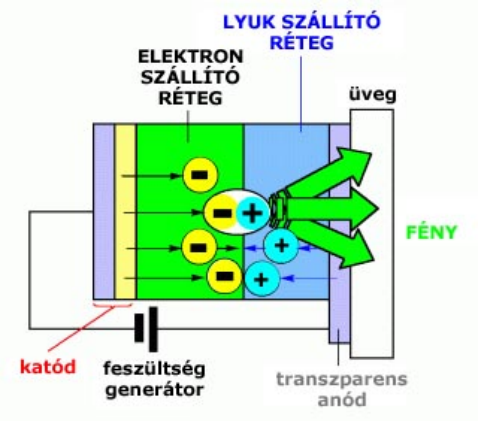
OLED (Organic Light-Emitting Diode) monitor

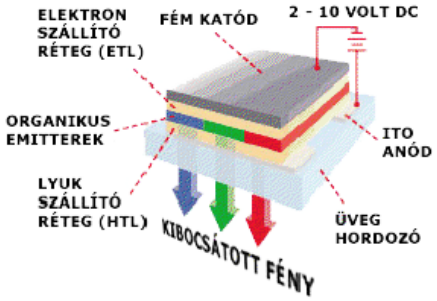




- Ha feszültséget kötünk a szerves anyagra, elektronok és ”lyukak” mozognak szemben egymással
- A találkozáskor energia szabadul fel. Ebből látható hullám lesz. A foton f frekvenciája (színe) a $E = hf$ összefüggéssel számítható, ahol E a foton energiája, h az ún. Planck állandó.

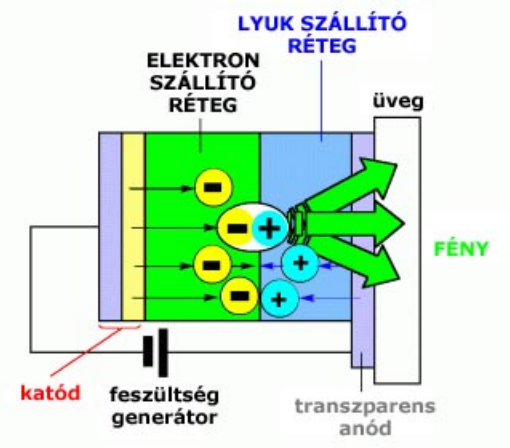
OLED (Organic Light-Emitting Diode) monitor

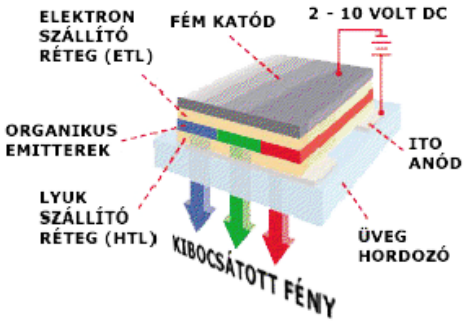




- Ha feszültséget kötünk a szerves anyagra, elektronok és ”lyukak” mozognak szemben egymással
- A találkozáskor energia szabadul fel. Ebből látható hullám lesz. A foton f frekvenciája (színe) a $E = hf$ összefüggéssel számítható, ahol E a foton energiája, h az ún. Planck állandó.

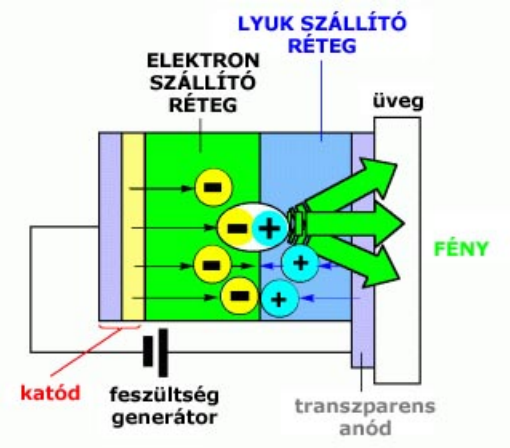
OLED (Organic Light-Emitting Diode) monitor

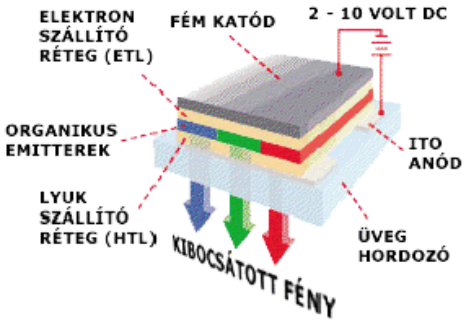




- Háttérvilágítás nincsen, az aktív (szerves) elemek adják a fényt.
- Képe szép és kontrasztos. Nagy a betekintési szöge.
- Az ára borsosabb az LCD-hez képest.

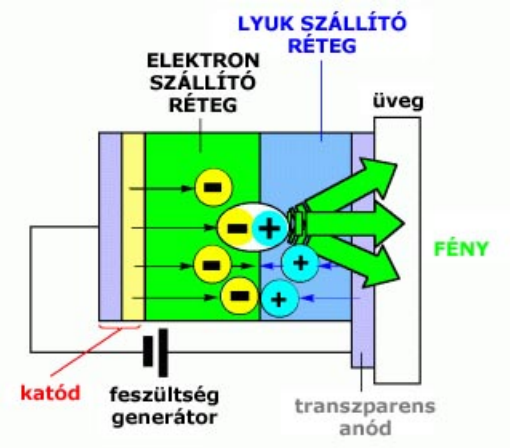
OLED (Organic Light-Emitting Diode) monitor

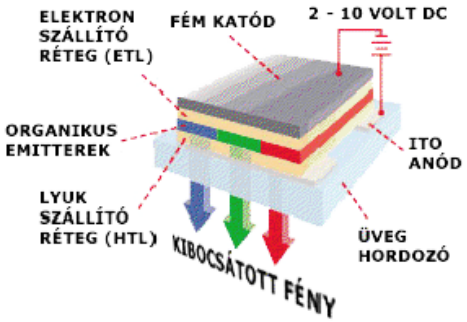




- Háttérvilágítás nincsen, az aktív (szerves) elemek adják a fényt.
- Képe szép és kontrasztos. Nagy a betekintési szöge.
- Az ára borsosabb az LCD-hez képest.

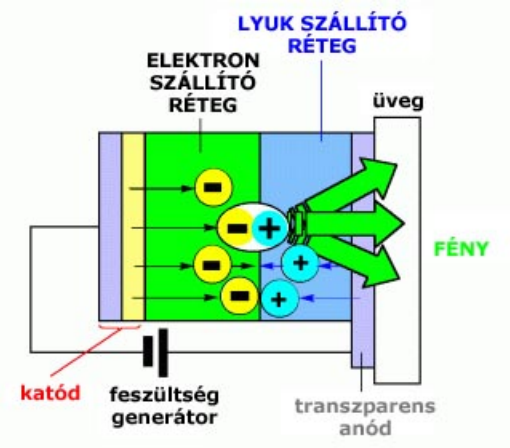
OLED (Organic Light-Emitting Diode) monitor

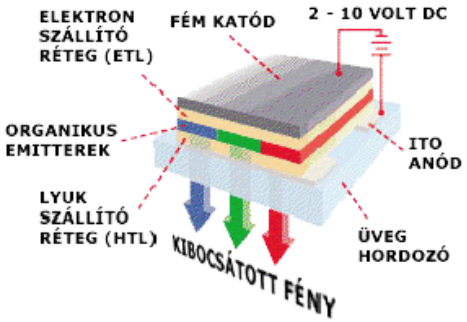




- Háttérvilágítás nincsen, az aktív (szerves) elemek adják a fényt.
- Képe szép és kontrasztos. Nagy a betekintési szöge.
- Az ára borsosabb az LCD-hez képest.

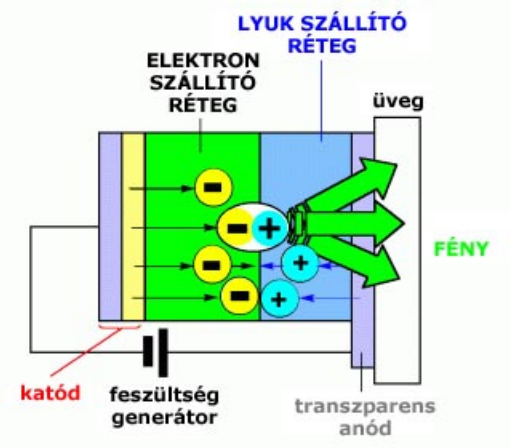
OLED (Organic Light-Emitting Diode) monitor

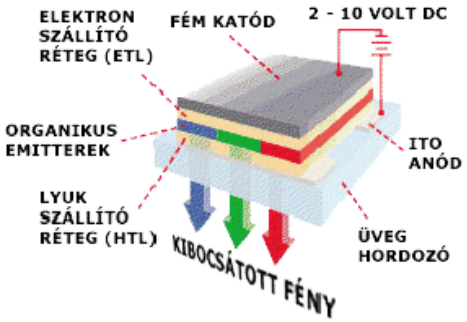




- Háttérvilágítás nincsen, az aktív (szerves) elemek adják a fényt.
- Képe szép és kontrasztos. Nagy a betekintési szöge.
- Az ára borsosabb az LCD-hez képest.

OLED (Organic Light-Emitting Diode) monitor





- Háttérvilágítás nincsen, az aktív (szerves) elemek adják a fényt.
- Képe szép és kontrasztos. Nagy a betekintési szöge.
- Az ára borsosabb az LCD-hez képest.

3D monitorok

- *Stereoscopy*
  Mást lát a két szem, nincs mozgás parallaxis
  - *Head Mounted Display*
  - *Shutter glasses*
  - Polarizált lencséjű szemüveg
  - 3D holografikus elven
- *Autostereoscopy*
  Nem kell hozzá külön eszköz a felhasználó részéről
  - *Parallax barrier*
  - *Lenticular lens*

3D monitorok

- *Stereoscopy*
  Mást lát a két szem, nincs mozgás parallaxis
  - *Head Mounted Display*
  - *Shutter glasses*
  - Polarizált lencséjű szemüveg
  - 3D holografikus elven
- *Autostereoscopy*
  Nem kell hozzá külön eszköz a felhasználó részéről
  - *Parallax barrier*
  - *Lenticular lens*

3D monitorok

- *Stereoscopy*
  Mást lát a két szem, nincs mozgás parallaxis
  - *Head Mounted Display*
  - *Shutter glasses*
  - Polarizált lencséjű szemüveg
  - 3D holografikus elven
- *Autostereoscopy*
  Nem kell hozzá külön eszköz a felhasználó részéről
  - *Parallax barrier*
  - *Lenticular lens*

3D monitorok

- *Stereoscopy*
  Mást lát a két szem, nincs mozgás parallaxis
  - *Head Mounted Display*
  - *Shutter glasses*
  - Polarizált lencséjű szemüveg
  - 3D holografikus elven
- *Autostereoscopy*
  Nem kell hozzá külön eszköz a felhasználó részéről
  - *Parallax barrier*
  - *Lenticular lens*

3D monitorok

- *Stereoscopy*
  Mást lát a két szem, nincs mozgás parallaxis
  - *Head Mounted Display*
  - *Shutter glasses*
  - Polarizált lencséjű szemüveg
  - 3D holografikus elven
- *Autostereoscopy*
  Nem kell hozzá külön eszköz a felhasználó részéről
  - *Parallax barrier*
  - *Lenticular lens*

3D monitorok

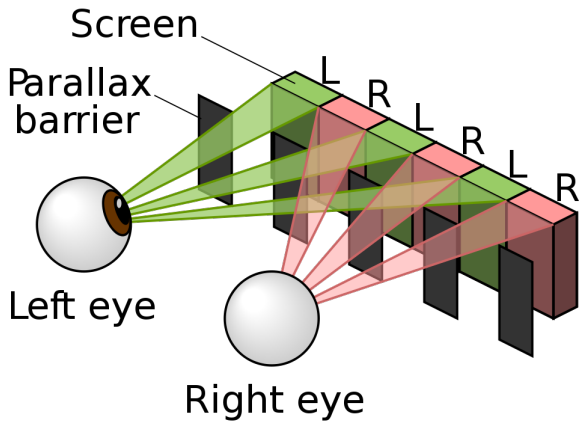
- *Stereoscopy*
  Mást lát a két szem, nincs mozgás parallaxis
  - *Head Mounted Display*
  - *Shutter glasses*
  - Polarizált lencséjű szemüveg
  - 3D holografikus elven
- *Autostereoscopy*
  Nem kell hozzá külön eszköz a felhasználó részéről
  - *Parallax barrier*
  - *Lenticular lens*

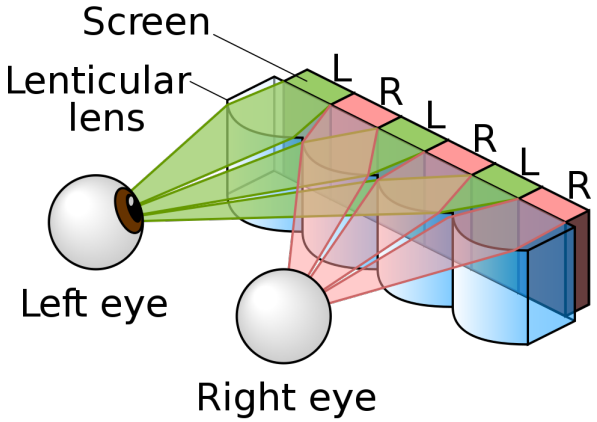
3D monitorok

- *Stereoscopy*
  Mást lát a két szem, nincs mozgás parallaxis
  - *Head Mounted Display*
  - *Shutter glasses*
  - Polarizált lencséjű szemüveg
  - 3D holografikus elven
- *Autostereoscopy*
  Nem kell hozzá külön eszköz a felhasználó részéről
  - *Parallax barrier*
  - *Lenticular lens*

Eszközök

Parallax és Lenticular





Marvin Raaijmakers, Wikipedia

és még...

- nyomtatók
- 3D nyomtatók
- plotterek
- projektorok

Rasztergrafika

Doink, vecteezy.com

Rasztergrafika

Doink, vecteezy.com

Vektorgrafika

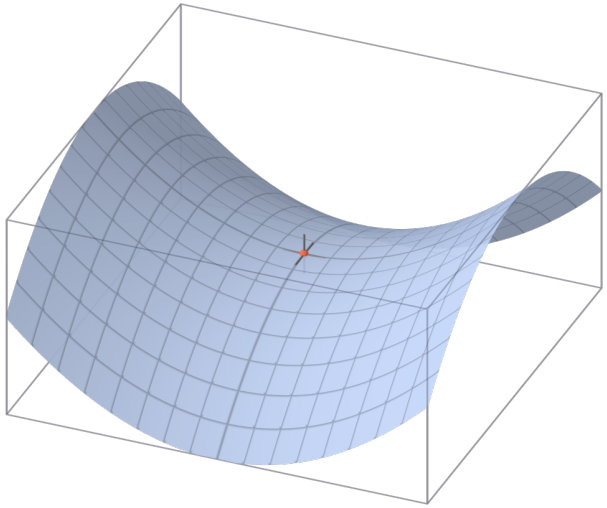
Doink, vecteezy.com

Vektorgrafika

Doink, vecteezy.com

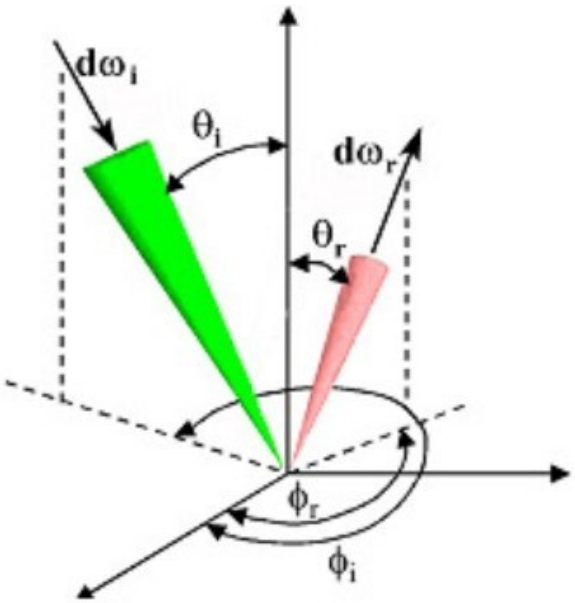
Modellezés

- Geometriai modellek
- Optikai paraméterek
- Textúrák
- *mind lehet generált, mért, fényképezet stb.*



Modellezés

- Geometriai modellek
- Optikai paraméterek
- Textúrák
- *mind lehet generált, mért, fényképezet stb.*



Modellezés

- Geometriai modellek
- Optikai paraméterek
- Textúrák
- *mind lehet generált, mért, fényképezet stb.*

Modellezés

- Geometriai modellek
- Optikai paraméterek
- Textúrák
- *mind lehet generált, mért, fényképezet stb.*

Algoritmusok

Megközelítési módok

- Sugárkövetés
- Inkrementális képszintézis

Fényjelenségek

- Tükröződés, fénytörés
- Vetett árnyékok
- Globális illuminació
- Térfogati jelenségek

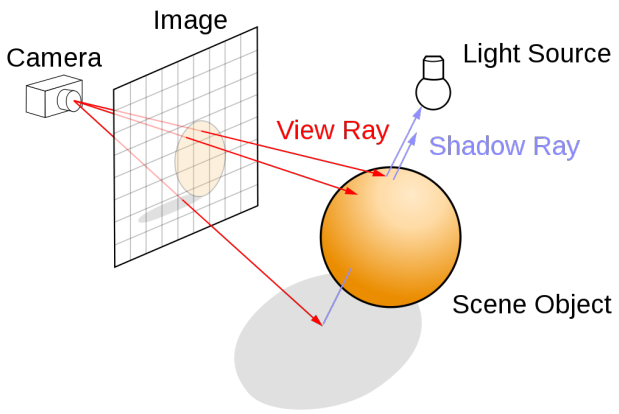
Algoritmusok

Megközelítési módok

- Sugárkövetés
- Inkrementális képszintézis

Fényjelenségek

- Tükröződés, fénytörés
- Vetett árnyékok
- Globális illuminació
- Térfogati jelenségek



Henrik, Wikipedia

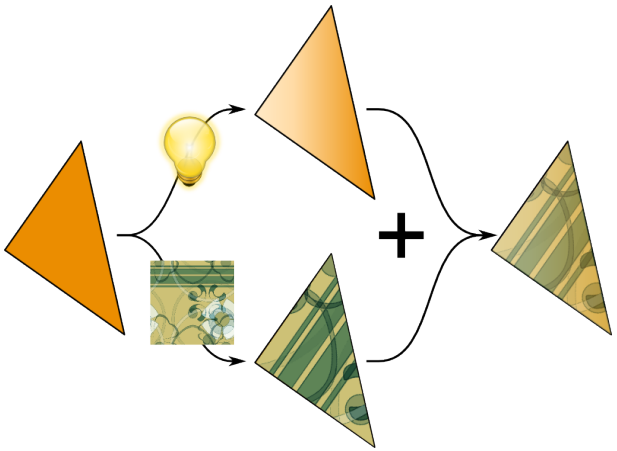
Algoritmusok

Megközelítési módok

- Sugárkövetés
- Inkrementális képszintézis

Fényjelenségek

- Tükröződés, fénytörés
- Vetett árnyékok
- Globális illuminació
- Térfogati jelenségek



Algoritmusok

Megközelítési módok

- Sugárkövetés
- Inkrementális képszintézis

Fényjelenségek

- Tükröződés, fénytörés
- Vetett árnyékok
- Globális illuminació
- Térfogati jelenségek

Algoritmusok

Megközelítési módok

- Sugárkövetés
- Inkrementális képszintézis

Fényjelenségek

- Tükröződés, fénytörés
- Vetett árnyékok
- Globális illuminació
- Térfogati jelenségek

Gilles Tran, Oyonale.com

Algoritmusok

Megközelítési módok

- Sugárkövetés
- Inkrementális képszintézis

Fényjelenségek

- Tükröződés, fénytörés
- Vetett árnyékok
- Globális illuminació
- Térfogati jelenségek

CryEngine2

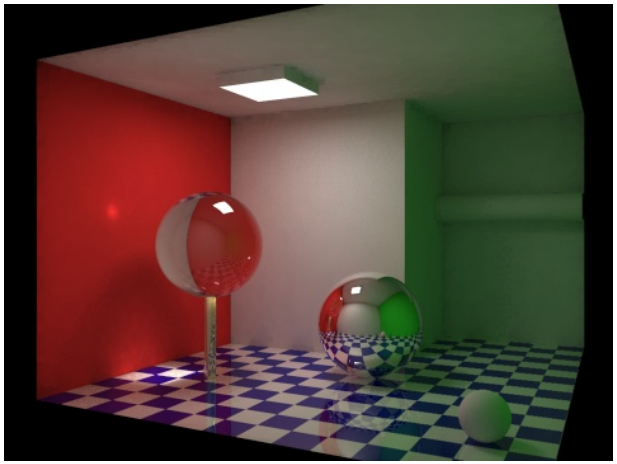
Algoritmusok

Megközelítési módok

- Sugárkövetés
- Inkrementális képszintézis

Fényjelenségek

- Tükröződés, fénytörés
- Vetett árnyékok
- Globális illumináció
- Térfogati jelenségek



Algoritmusok

Megközelítési módok

- Sugárkövetés
- Inkrementális képszintézis

Fényjelenségek

- Tükröződés, fénytörés
- Vetett árnyékok
- Globális illuminació
- Térfogati jelenségek

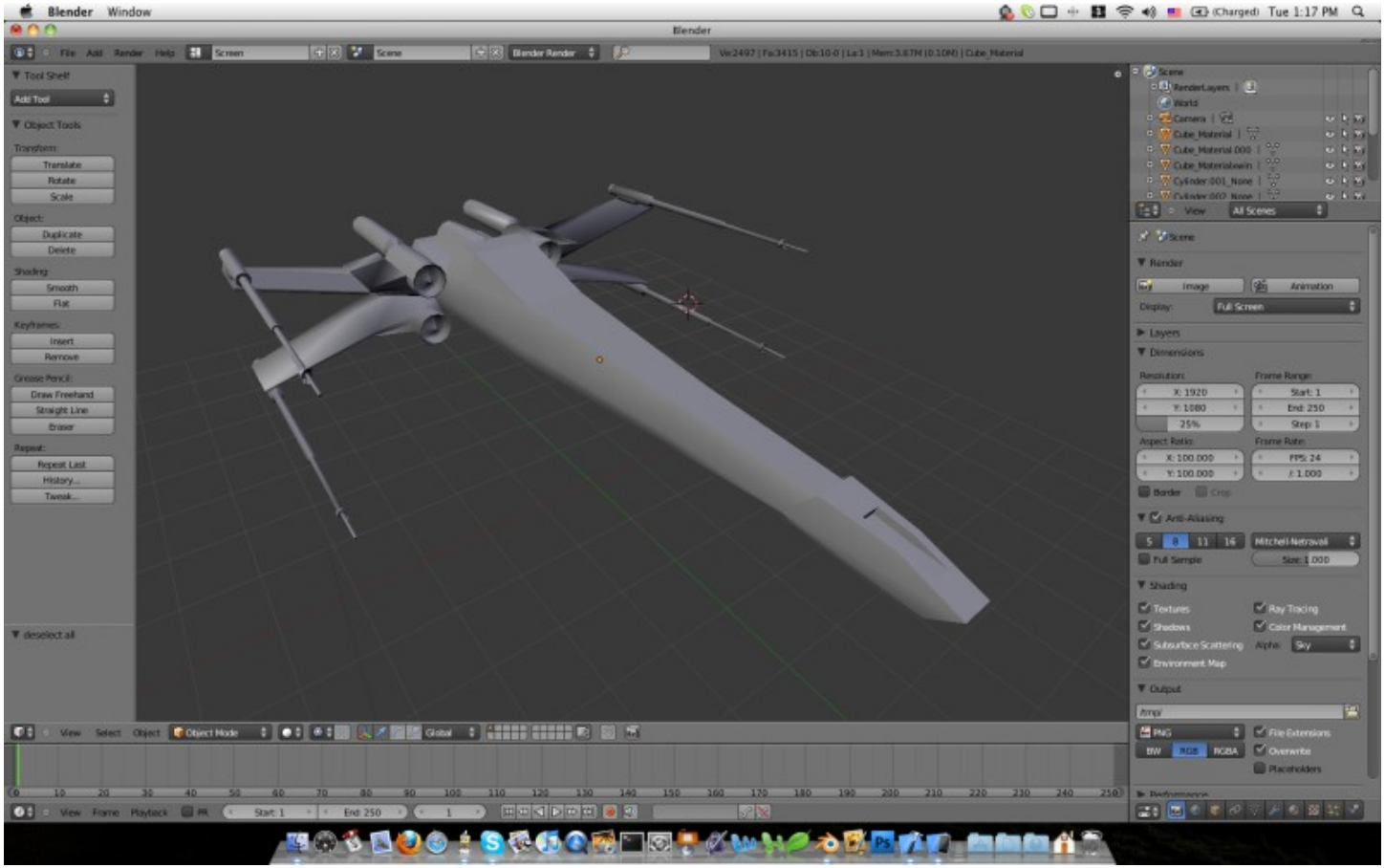
BlendELF.com

Tartalom

- Előadás
- Gyakorlat

1. Számítógépes grafika
   - Célja, feladata, területei
   - Az emberi szem működése
   - Megjelenítő eszközök
   - Raszter- és vektorgrafika
   - 3D grafika
2. Ajánlás
   - Programok
   - Honlapok
   - Könyvek

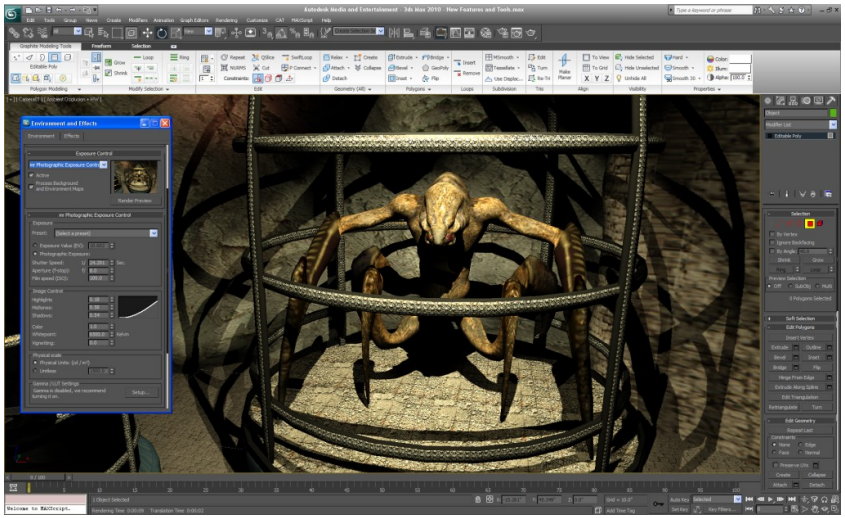
blender



http://www.blender.org

Autodesk Maya & 3D Studio Max

Maya



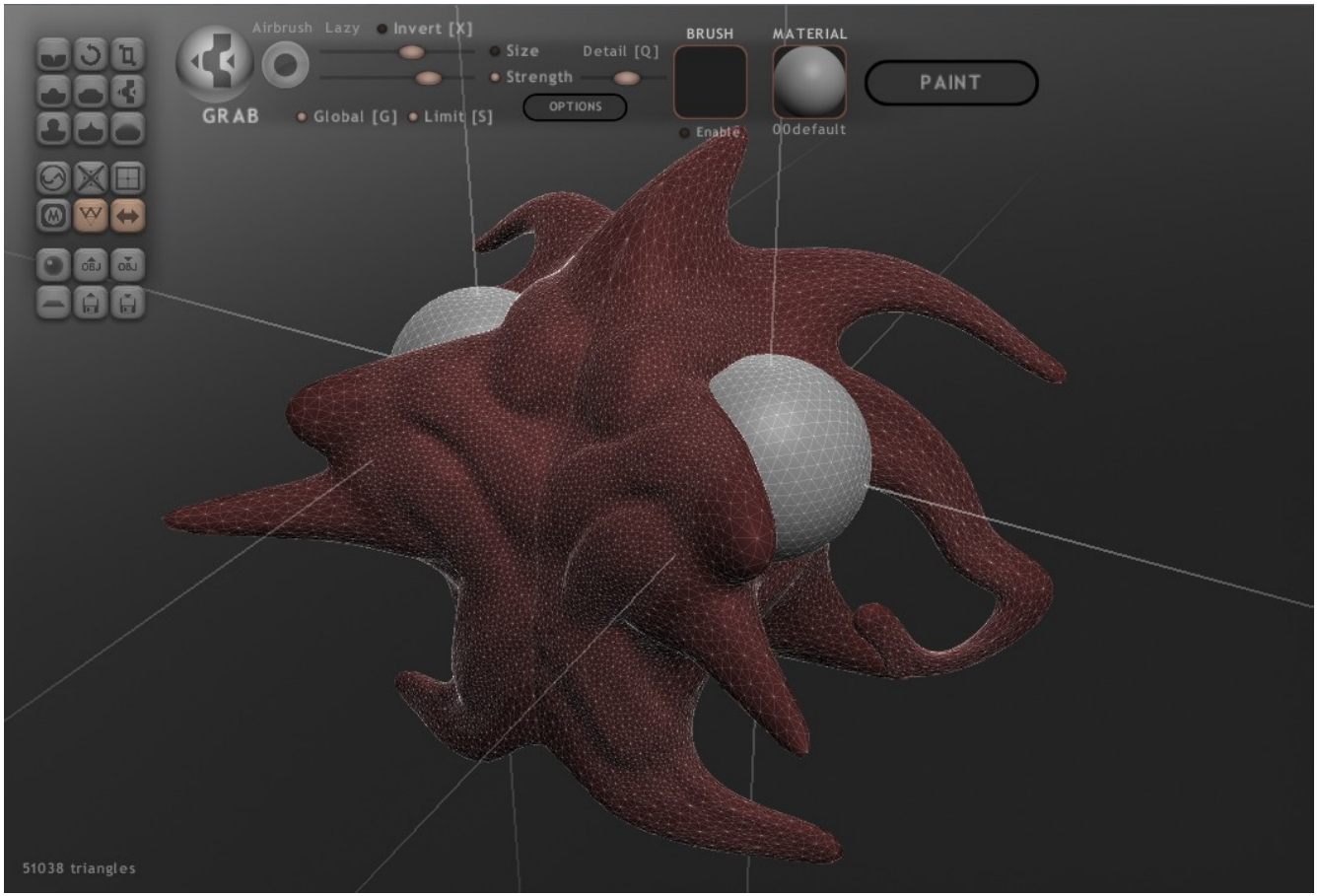
3D Studio Max

PovRay

Jaime Vives Piqueres, www.ignorancia.org

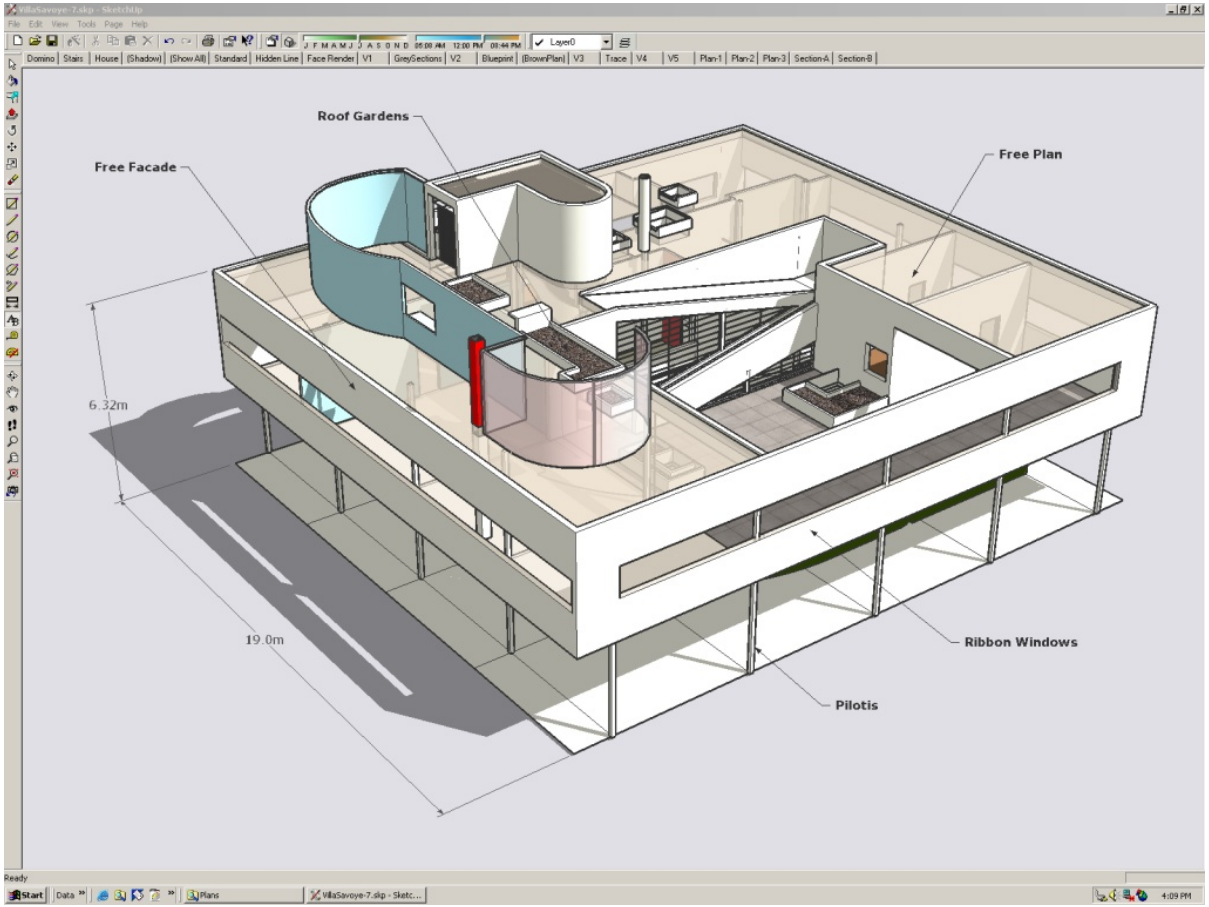
http://www.povray.org/

sculptris



http://www.sculptris.com

Trimble/Google SketchUp



http://sketchup.google.com/

Honlapok

- http://processing.org/
  Processing alkalmazás + dokumentáció
- http://www.gamedev.net/
  Cikkek, fórumok
- http://nehe.gamedev.net/
  OpenGL *tutorial*-ok
- http://portal.acm.org/portal.cfm
  Tudományos cikk adatbázis, nem csak grafikáról
  (A cikkek csak egyetemi hálózatból tölthetők le)

Honlapok

- http://processing.org/
  Processing alkalmazás + dokumentáció
- http://www.gamedev.net/
  Cikkek, fórumok
- http://nehe.gamedev.net/
  OpenGL *tutorial*-ok
- http://portal.acm.org/portal.cfm
  Tudományos cikk adatbázis, nem csak grafikáról
  (A cikkek csak egyetemi hálózatból tölthetők le)

Honlapok

- http://processing.org/
  Processing alkalmazás + dokumentáció
- http://www.gamedev.net/
  Cikkek, fórumok
- http://nehe.gamedev.net/
  OpenGL *tutorial*-ok
- http://portal.acm.org/portal.cfm
  Tudományos cikk adatbázis, nem csak grafikáról
  (A cikkek csak egyetemi hálózatból tölthetők le)

Honlapok

- http://processing.org/
  Processing alkalmazás + dokumentáció
- http://www.gamedev.net/
  Cikkek, fórumok
- http://nehe.gamedev.net/
  OpenGL *tutorial*-ok
- http://portal.acm.org/portal.cfm
  Tudományos cikk adatbázis, nem csak grafikáról
  (A cikkek csak egyetemi hálózatból tölthetők le)

Ajánlott irodalom

- Szirmay-Kalos L., Antal Gy., Csonka F.
  Háromdimenziós grafika, animáció és játékfejlesztés.
  ComputerBooks, 2003.
- Szirmay-Kalos L.
  Számítógépes grafika.
  ComputerBooks, 1999.
- Nyisztor K.
  Shaderprogramozás - Grafika és játékprogramozás DirectX-szel
  Szak Kiadó, 2009
- P. Martz.
  OpenGL röviden
  Kiskapu Kiadó, 2007

Számítógépes Grafika

Hajder Levente és Baráth Dániel
hajder@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2019/2020. I. félév

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Modellezés feladata

- Geometriai modellezés feladata
  - A világunkat modellezni kell a térben.
    - Valamilyen koordinátarendszer szükséges
  - Mozgó (dinamikus) tárgyak esetén a térbeli mozgást le kell írni.
    - Mozognak
    - Pörögnek-forognak
    - Kicsinyednek-nagyobbodnak
- Vetítés feladata
  - A háromdimenziós világot levetíteni a kétdimenziósképsíkra.
  - Kameraszerű képet adni.

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Descartes-koordinátarendszer

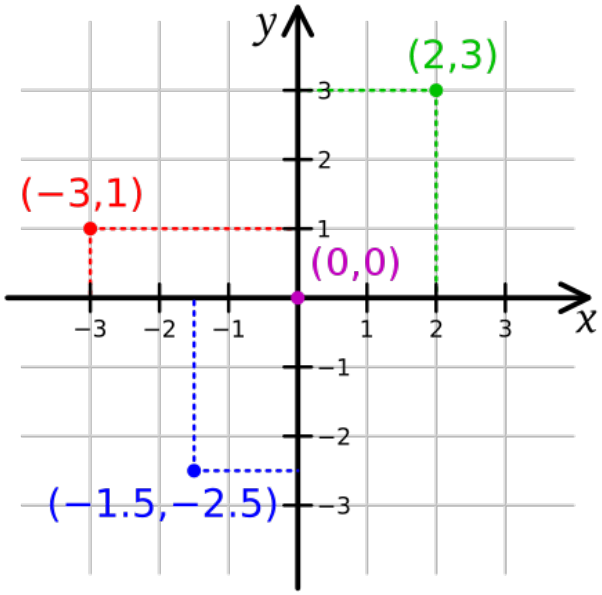
- Descartes, 1637.: *Értekezés a módszerről* (Discours de la méthode pour bien conduire sa raison et chercher la vérité dans les sciences)
- Legtöbbször ezzel találkozunk, ez a legegyszerűbb és legelterjettebb megadási mód
- A dimenziónak megfelelő számú koordinátával írjuk le a pontokat

Descartes-koordinátarendszer

- Az euklidészi sík minden véges pontjához egyértelműen hozzárendel egy rendezett, valós (x, y) számpárt, ahol az x az abszcissza és y az ordináta
- Ehhez válasszunk két, egymásra merőleges, irányított egyenest, az OX (x) és OY (y) koordinátatengelyeket, amelyek egymást az O pontban, az origóban metszik. Ekkor a P pont x és y koordinátái a P pont OX és OY irányított egyenesektől vett előjeles távolsága
- Három dimenzióban hasonló: 3 koordináta, 3 egymásra merőleges koordináta tengely

Geometriai értelmezés

- *Szemléletesebben*: A P(a,b,c) az a pont, amit az origóból az x tengely mentén a egységet lépve, majd az y tengely mentén b egységet lépve, végül a z tengely mentén c egységet lépve kapunk



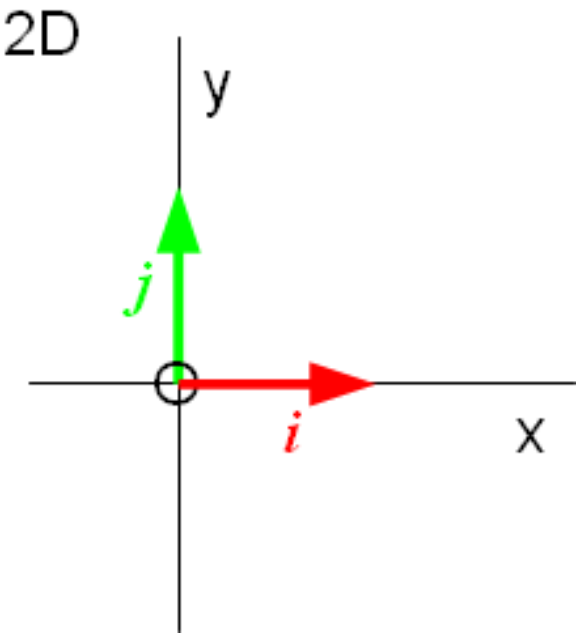
Geometriai értelmezés

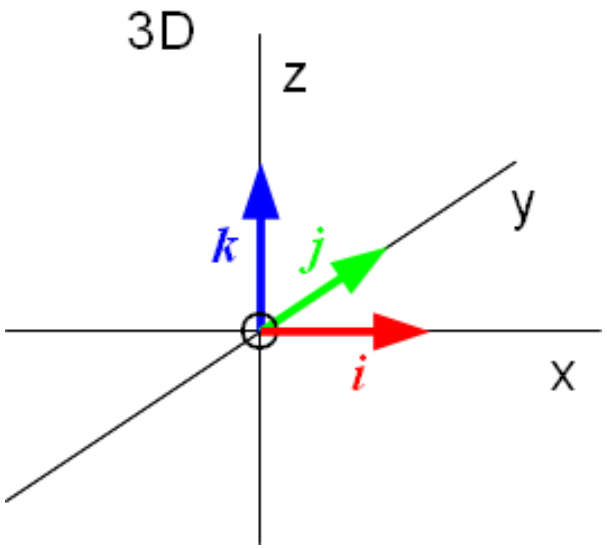
- Vagyis a fenti értelmezés szerint, felhasználva az egységnyi hosszú, koordinátatengelyek irányába mutató **i**, **j**, **k** bázisvektorokat, az $[a, b, c]^T$ koordináták a következő pontot azonosítják:

$$P = O + a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$$

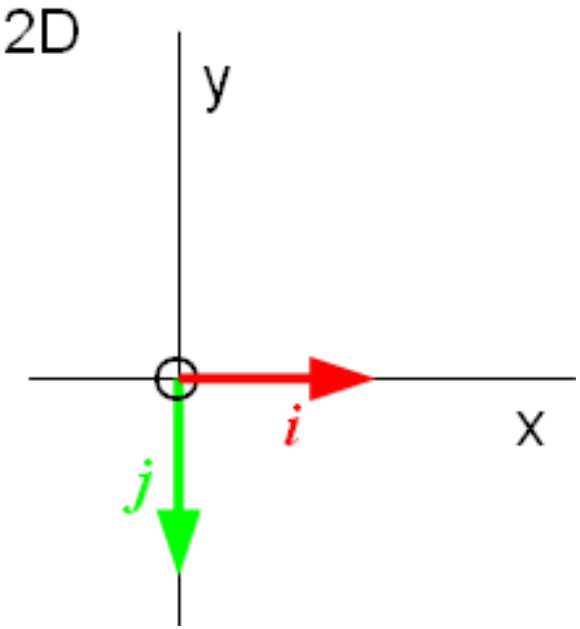
$$= O + a\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + c\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

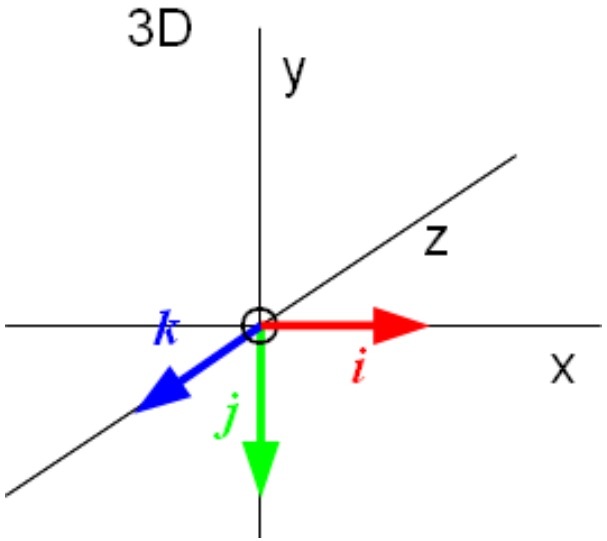
Sodrásirány - jobbsodrású rendszer





Sodrásirány - balsodrású rendszer



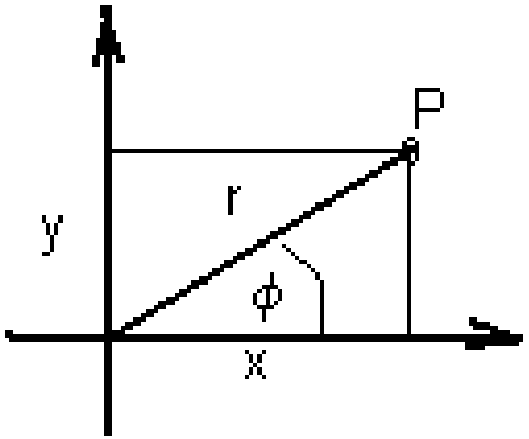


Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Síkbeli polákoordináta-rendszer

- Egy O kezdőpontból (referenciapontból) induló félegyenes (polártengely) határozza meg.
- Egy P pont helyét két adat azonosítja: (r, ϕ)
  - $r \geq 0$: a P pont O-tól vett távolsága
  - $\phi \in [0, 2\pi)$: az O-n és P-n átmenő egyenes polártengellyel bezárt szöge
- $r = 0$ esetén a felírás nem egyértelmű



Konverziók

- Polár $\rightarrow$ Descartes:

$$x = r\cos\phi,\ \ y = r\sin\phi$$

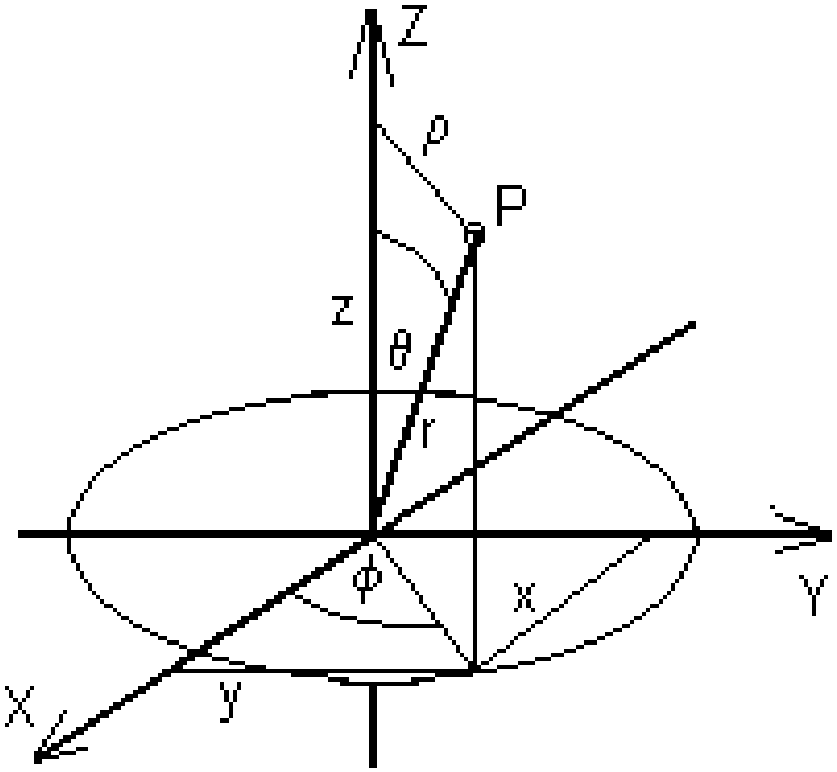
- Descartes $\rightarrow$ polár:

$$r = \sqrt{x^2 + y^2}$$

$$\phi = \begin{cases} \arctan\frac{y}{x}, & x > 0 \wedge y \geq 0 \\ \arctan\frac{y}{x} + 2\pi, & x > 0 \wedge y < 0 \\ \arctan\frac{y}{x} + \pi, & x < 0 \\ \frac{\pi}{2}, & x = 0 \wedge y > 0 \\ \frac{3\pi}{2}, & x = 0 \wedge y < 0 \\ ???, & x = 0 \wedge y = 0 \end{cases}$$

- arctan() nem egyértelmű ($\pi - periodikus$) ezért programozásban az $\phi = atan2\,(y, x)$ függvényt használjuk

Gömbi koordináták



Gömbi koordináták

- Térbeli polár-koordináták; egy alapsík (és annak PKR-e) illetve egy arra merőleges "Z tengely"
- Egy térbeli P pontot három adat reprezentál: (r, ϕ, θ)
  - r, ϕ: a P pont alapsíkra vett vetületének polárkoordinátái
  - $\theta \in [0, \pi]$: az O és P-t összekötő egyenes Z tengellyel bezárt szöge

Konverziók

- A síkban látott esethez hasonló feltételek mellett:
- Gömbi $\rightarrow$ Descartes:

$$x = r\cos\phi\sin\theta,\ \ y = r\sin\phi\sin\theta,\ \ z = r\cos\theta$$

- Descartes $\rightarrow$ gömbi:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \phi &= atan2(y, x) \\ \theta &= \arccos\frac{z}{r}, \qquad\qquad r \neq 0 \end{aligned}$$

- Az x=y=z=0 eset itt is többértelmű!

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Baricentrikus koordináták

- August Ferdinand Möbius [1827]
- Motiváció: sokszor egy konkrét, véges része érdekes csak számunkra a térnek. Ennek a Descartes-félénél egy "kiegyensúlyozottabb" reprezentációját keressük.
- A baricentrikus koordináták nem függnek egy pont önkényes origónak választásától

Tömegközéppont

- Mechanikai analógia: pontrendszer tömegközéppontja
- Legyen $n+1$ pontunk és helyezzünk minden P_i pontba $m_i \in \mathbb{R}$ súlyt. Ekkor a tömegközéppont:

$$M = \frac{1}{\sum_{i=0}^{n} m_i} \sum_{i=0}^{n} m_i P_i$$

Baricentrikus koordináták

- Ha $\mathbb{E}^n$ -ben az $A_0, ..., A_n$ pontok kifeszítik a teret (nem egy $n-1$ dimenziós altérbe esnek), akkor a tér bármely X pontjához találhatóak $\lambda_0, ..., \lambda_n$ valós számok úgy, hogy

$$X = \sum_{i=0}^{n} \lambda_i A_i,$$

ahol a λ_i baricentrikus koordinátákra teljesül, hogy

$$\sum_{i=0}^{n} \lambda_i = 1.$$

Megjegyzés

- Ha $\forall i$-re $\lambda_i \geq 0$, akkor konvex kombinációról beszélünk és a pontok konvex burkába esnek a kombináció eredményei.
- Az affin transzformációk nem változtatják meg a baricentrikus koordinátákat
- *Homogén jellegű* koordináták: egy $h \neq 0$ számmal megszorozva a koordinátákat ugyanazt a pontot kapjuk.

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Homogén koordináták

- Az euklideszi tér minden pontjához hozzárendelünk egy számnégyest, *homogén koordinátákat*: $P_{hom} = [x, y, z, w]^T$
- Átváltás $P_{real} = [x', y', z']$ Descartes-i és $P_{hom} = [x, y, z, w]^T$ homogén koordinátákkal megadott pontok között:
  - "Odafelé:" $x' = x/w$, $y' = y/w$, és $z' = z/w$
  - "Visszafelé:" $x = hx'$, $y = hy'$, $z = hz'$, és $w = h$, ahol $h \in \Re$
- A megfeleltetés többértelmű, hiszen h értéke tetszőleges lehet (zérust kivéve): egy valós koordinátához végtelen sok homogén koordinátás alak tartozik.

Affin transzformációk

- Lineáris transzformációk 4×4-es mátrixszal leírhatóak
  - A transzformációkat $P = [x, y, z, 1]$ alakú pontokon végezzük
  - $P' = TP$ alakban kapjuk az eredményt
- Transformációk típusai:
  - Egybevágósági transzformációk
    - Eltolás (T_{elt})
    - Elforgatás (T_{forg})
    - Tükrözés (T_{tukor})
  - Hasonlósági transzformációk
    - Skálázás (T_s)
  - Egyéb transzformációk
    - Nyírás (T_{ny})
  - A transzformációk egymás után elvégezhetőek, ebben az esetben (jobbról balra haladva) szorzatot kell felírni. Például: $T = T_{forg1} T_s T_{forg2} T_{elt1} T_{ny} T_{s2}$

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Eltolás

- $d = [d_x, d_y, d_z]$ vektorral való eltolást valósít meg
- Alakja $T_{elt} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- Az eredmény vektor $P' = T_{elt}P$ hossza megegyezik az eredeti P vektor hosszával $\rightarrow$ T_{elt} egybevágósági transzformáció

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Elforgatás

- X, Y, Z tengely körüli elforgatás szögét jelöljük rendre α-val, β-val és γ-val.
- A forgatási mátrixok alakjai:

$$T_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_\beta = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ és}$$

Elforgatás

- Továbbá:

$$T_\gamma = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- A teljes 3D-s forgatás a három forgatás szorzataként adható meg: $T_{forg} = T_\alpha T_\beta T_\gamma$
- Az eredmény vektor $P' = T_{forg} P$ hossza megegyezik az eredeti P vektor hosszával $\rightarrow$ T_{forg} egybevágósági transzformáció

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Tükrözés

- $d = [d_x, d_y, d_z]$ vektorral való eltolást valósít meg
- Alakja (tükrözés az X tengelyre) $T_{tukor} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- Az eredmény vektor $P' = T_{tukor}P$ hossza megegyezik az eredeti P vektor hosszával, ezért T_{tukor} egybevágósági transzformáció
- A tükrözés a többi tengely esetében is lehetséges, de az X tengelyre tükrözéssel + forgatással tetszőleges forgatás + tükrözés helyettesíthető.

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Skálázás

- A skálázás a P vektor hosszát nyújtja meg s-szeresére
- Alakja $T_s = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- A transzformáció a méretet megváltoztatja, de a szögeket nem $\rightarrow$ hasonlósági transzformáció

Skálázás

- Koordinátánkánt lehet saját skálát is alkalmazni:
- $T_s = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- A hasonlósági tulajdonságot ezzel elveszítjük

Tartalom

1. Modellezés feladata
2. Koordinátarendszerek
   - Descartes-koordinátarendszer
   - Polárkoordináták
   - Baricentrikus koordináták
   - Homogén koordináta
3. Affin transzformációk
   - Eltolás
   - Elforgatás
   - Tükrözés
   - Skálázás
   - Nyírás

Nyírás

- A nyírás a tárgy szögeit is megváltoztatja
- Alakja $T_{ny} = \begin{bmatrix} 1 & a & b & 0 \\ 0 & 1 & c & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Számítógépes Grafika

Hajder L. és Baráth D.
hajdere@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2019/2020. I. félév

Tartalom

1. Vetítési transzformációk
   - Párhuzamos (merőleges) vetítés
   - Skálázottan párhuzamos vetítés
   - Perspektív vetítés
     - Homogén osztás
     - Perspektív transzformáció alakja
     - Homogén osztás hatása
     - Speciális esetek
2. Transzformációk összehasonlítása

Tartalom

1. Vetítési transzformációk
   - Párhuzamos (merőleges) vetítés
   - Skálázottan párhuzamos vetítés
   - Perspektív vetítés
     - Homogén osztás
     - Perspektív transzformáció alakja
     - Homogén osztás hatása
     - Speciális esetek
2. Transzformációk összehasonlítása

Motiváció

- A színterünk képét akarjuk előállítani: vetíteni egy síkra
- Az ember által látott képet nem lehet előállítani affin transzformációk segítségével. A "távolodó" párhuzamosok összetartanak, nem maradnak párhuzamosak.
- Ez a látvány előállítható *központi vetítéssel*. Ez a transzformáció a *homogén térben* lineáris transzformáció.
- Az affin transzformációk nem "bántották" az ideális elemeket, a fentiekhez azonban ez "kell"

Általános eset

Ha egy *homogén* transzformációs mátrix utolsó sora nem $[0, 0, 0, 1]$, akkor az olyan *homogén lineáris transzformáció*, ami az eukleidészi térnek nem lineáris transzformációja.

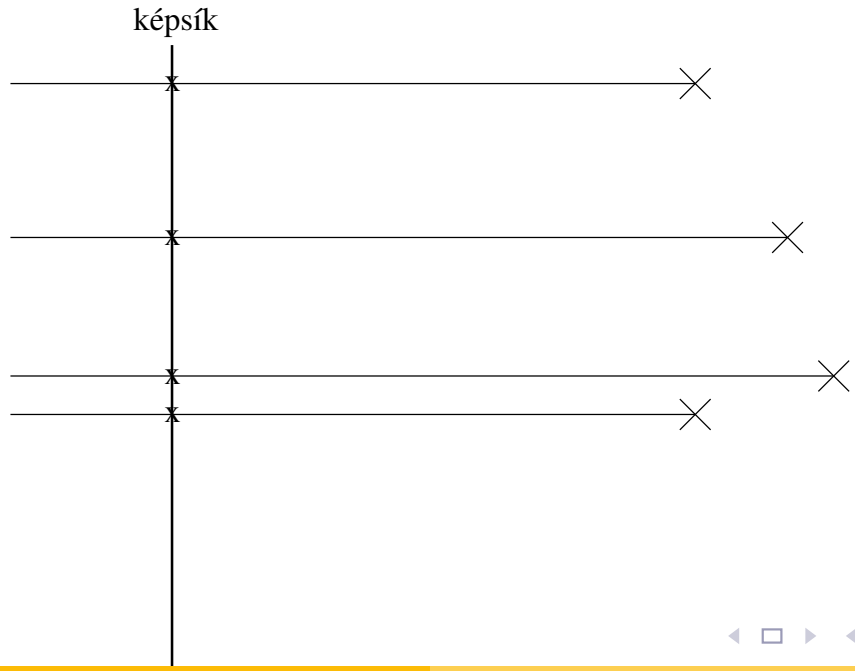
Tartalom

1. Vetítési transzformációk
   - Párhuzamos (merőleges) vetítés
   - Skálázottan párhuzamos vetítés
   - Perspektív vetítés
     - Homogén osztás
     - Perspektív transzformáció alakja
     - Homogén osztás hatása
     - Speciális esetek
2. Transzformációk összehasonlítása

Párhuzamos (merőleges) vetítés

- A mátrix ami megadja egyszerű, például az XY síkra való vetítés

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



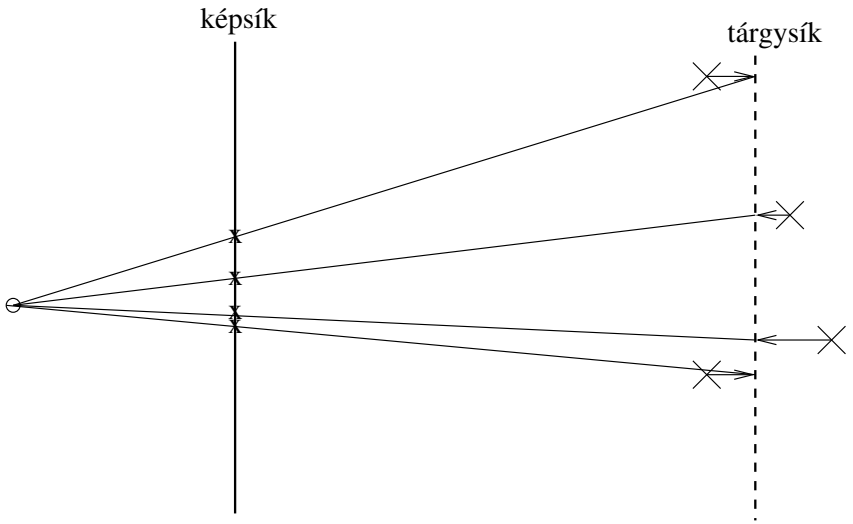
Tartalom

1. Vetítési transzformációk
   - Párhuzamos (merőleges) vetítés
   - Skálázottan párhuzamos vetítés
   - Perspektív vetítés
     - Homogén osztás
     - Perspektív transzformáció alakja
     - Homogén osztás hatása
     - Speciális esetek
2. Transzformációk összehasonlítása

Skálázottan párhuzamos vetítés

- A párhuzamos vetítéssel majdnem megegyezik.
- A mátrix ami megadja egyszerű, például az XY síkra való vetítés

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



Tartalom

1. Vetítési transzformációk
   - Párhuzamos (merőleges) vetítés
   - Skálázottan párhuzamos vetítés
   - Perspektív vetítés
     - Homogén osztás
     - Perspektív transzformáció alakja
     - Homogén osztás hatása
     - Speciális esetek
2. Transzformációk összehasonlítása

Ismétlés: homogén osztás

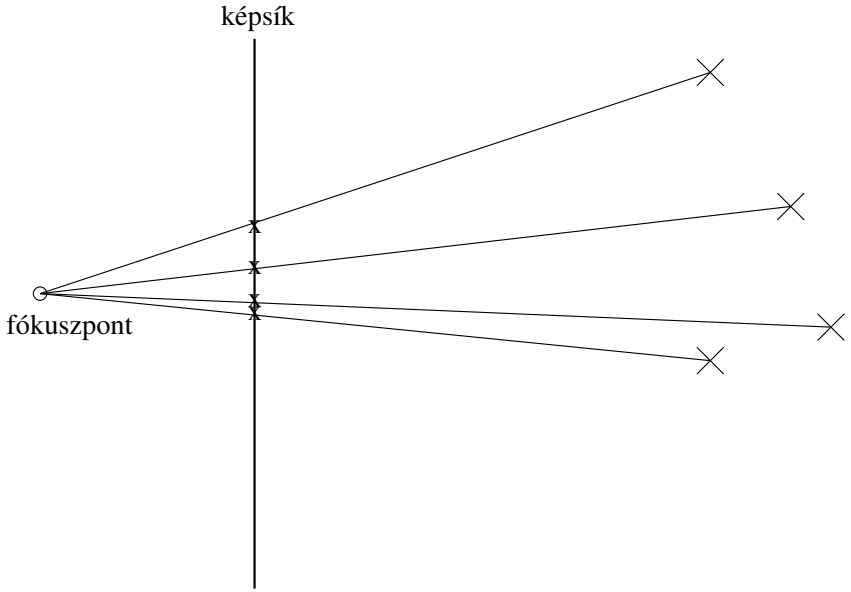
- Mivel egy $\mathbf{T}$ "valódi" projektív transzformáció utolsó sora nem $[0, 0, 0, 1]^T$, ezért

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \mathbf{T} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

  transzformáció után, $w \neq 1$ általános esetben.
- Ha ezt a pontot az eukleidészi térbe szeretnénk átvinni (mert pl. meg akarjuk jeleníteni), akkor végig kell osztanunk w-vel.
- (Persze csak akkor, ha $w \neq 0$)
- Ezt nevezzünk homogén osztásnak.
- Értelmezhető alacsonyabb dimenzióban is.

Perspektív transzformáció

- Fókuszpontba gyűjti a sugarakat.
- Fókusztávolság: fókuszpont és a képsík távolsága
- Hívják projektív (vetítő) transzformációnak és középpontos vetítésnek is.

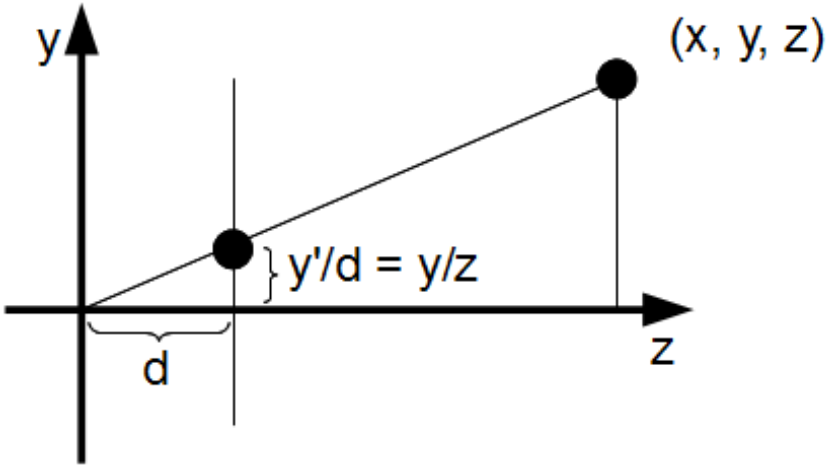


Középpontos vetítés

- Perspektív transzformáció képlete:

$$x' = d\frac{x}{z}$$
$$y' = d\frac{y}{z}$$

- d neve: fókusztávolság



Perspektív transzformáció 2D-s homogén transzformációval

- Perspektív transzformáció alakja:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ \frac{Z}{d} \end{bmatrix}$$

- Homogén osztást elvégezve:

$$\begin{bmatrix} X \\ Y \\ \frac{Z}{d} \end{bmatrix} \sim \begin{bmatrix} d\frac{X}{Z} \\ d\frac{Y}{Z} \\ 1 \end{bmatrix}$$

Perspektív transzformáció 3D-s homogén transzformációval

- Perspektív transzformáció módosul:

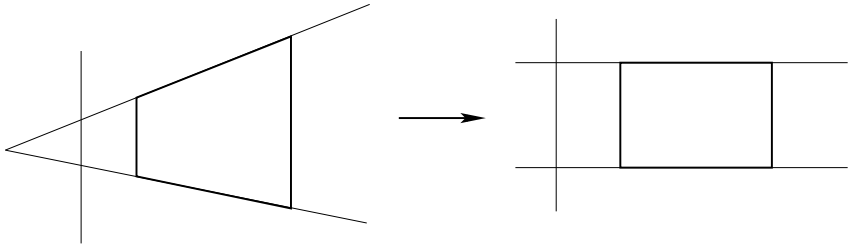
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ \frac{Z}{d} \end{bmatrix}$$

- Homogén osztást elvégezve:

$$\begin{bmatrix} X \\ Y \\ Z \\ \frac{Z}{d} \end{bmatrix} \sim \begin{bmatrix} d\frac{X}{Z} \\ d\frac{Y}{Z} \\ d \\ 1 \end{bmatrix}$$

Homogén osztás hatása

- A homogén osztás a vetítő egyeneseken levő pontokat ugyanabba a képpontba viszi.
- Felfogható úgy, hogy a perspektív transzformációból párhuzamos vetítést csinál.



Speciális eset: pontok a $Z = 0$ síkon

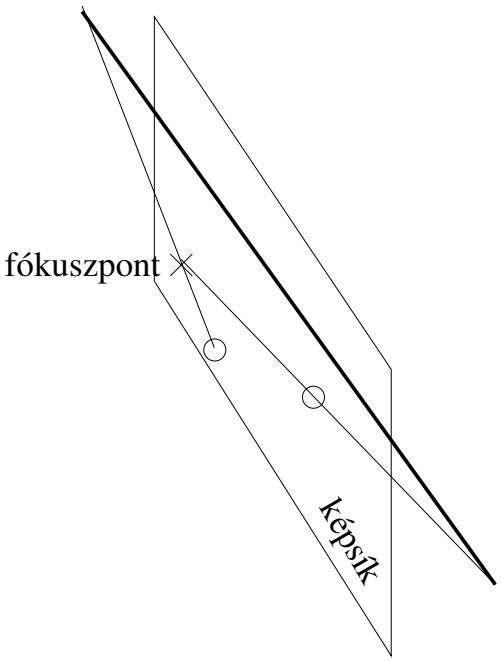
- Homogén osztás:

$$\begin{bmatrix} X \\ Y \\ Z/d \end{bmatrix} \sim \begin{bmatrix} d\frac{X}{Z} \\ d\frac{Y}{Z} \\ 1 \end{bmatrix}$$

- $Z = 0$ esetén nullával osztunk: pont a végtelenben
- A végtelennek iránya van: (X, Y).

Speciális eset: $Z = 0$ síkot metsző egyenes

- Eredmény: egyenes (szakasz) kifordul



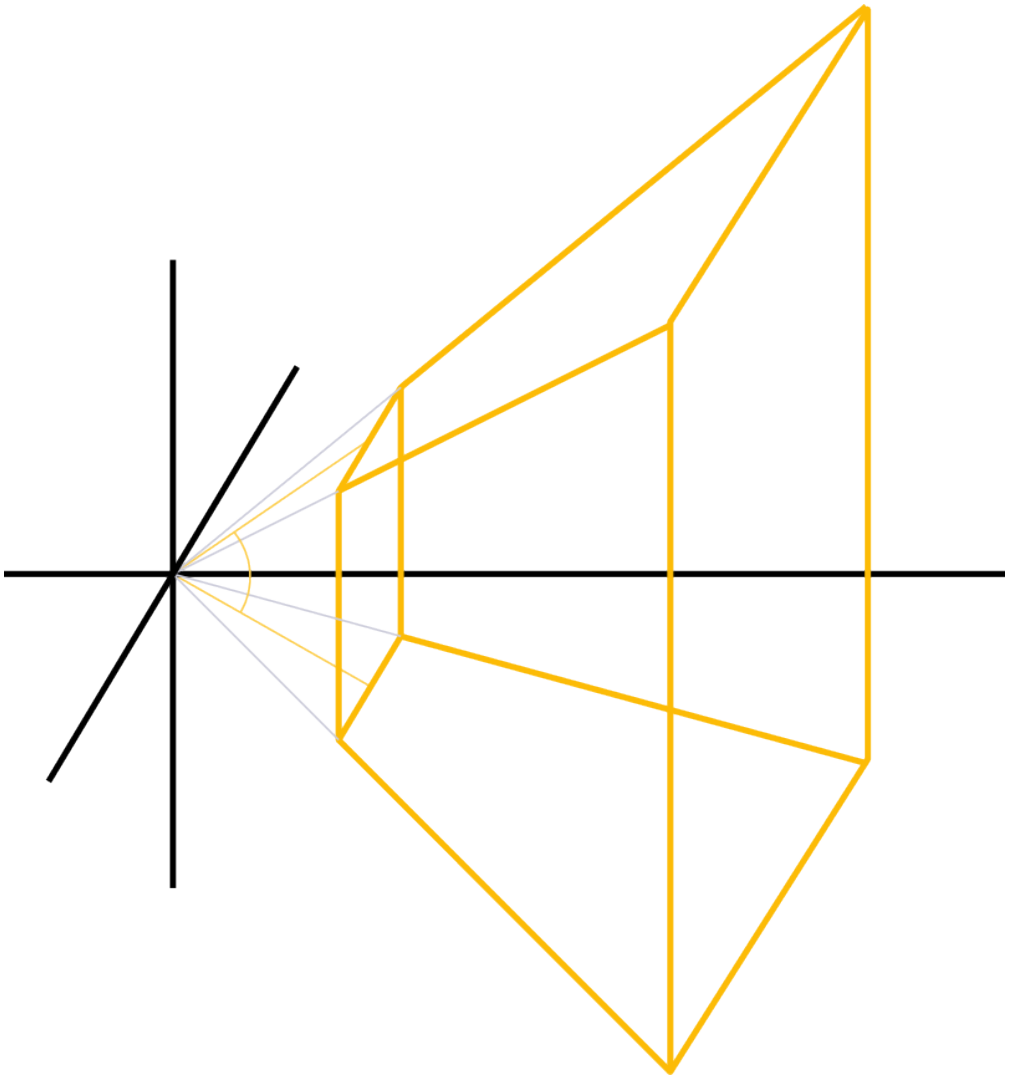
Kép a vetítés után:

- Következmény:
  - $Z = 0$ síkkal metszést figyelni kell vagy
  - Ki kell dobni ezeket az éleket.

Perspektív transzformáció

- Központi vetítést valósít meg.
- Az origóbol a z tengely mentén "nézünk" a térre.
- A látótérnek egy csonkagúla felel meg.
- A transzformáció a szem pozícióban találkozó vetítő egyenesekből párhuzamosokat csinál.
- Paraméterei:
  - a gúla függőleges nyílásszöge (d fókusztávolság szabályozza)
  - a gúla alapjának az oldalainak az aránya,
  - a közeli vágósík távolsága
  - a távoli vágósík távolsága
    - A távoli vágósíkra a sok tárgy miatt van szükség (számítások csökkentése).

Láthatósági gúla



Összegzés: transzformációs mátrixok

$$\left[\begin{array}{c|c} \text{3x3 lineáris rész} & \text{eltolás} \\ \hline \text{projektív rész} & 1 \end{array}\right] \quad \left[\begin{array}{c} x \\ y \\ z \\ \hline 1 \end{array}\right]$$

Megjegyzések

- Mi történik, ha a vektorunk negyedik koordinátája nulla (vagyis ha vektort azonosít a számnégyes)?
- Az eltolás rész nem hat rá!

Megjegyzések

- Figyeljünk: nem mindenhol szoroznak jobbról a vektorokkal, balról is lehet:

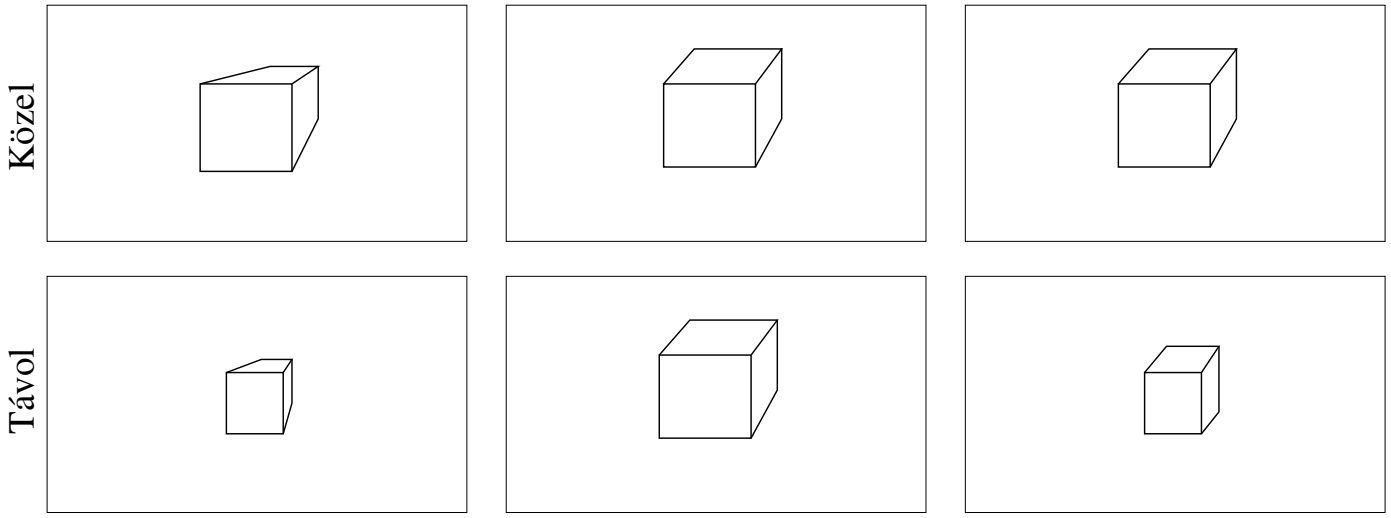
$$\left[\begin{array}{c|c} x, y, z & 1 \end{array}\right] \left[\begin{array}{c|c} \mathbf{A}^T\text{: 3x3 lineáris rész} & \text{projektív} \\ \hline \text{eltolás rész} & 1 \end{array}\right]$$

Tartalom

1. Vetítési transzformációk
   - Párhuzamos (merőleges) vetítés
   - Skálázottan párhuzamos vetítés
   - Perspektív vetítés
     - Homogén osztás
     - Perspektív transzformáció alakja
     - Homogén osztás hatása
     - Speciális esetek
2. Transzformációk összehasonlítása

Transzformációk összehasonlítása

- Végkövetkeztetés: a perspektív transzformáció "verhetetlen".



Perspektív vetítés | Merőleges vetítés | Gyengén perspektív vetítés

- Megjegyzés: valódi kameráknak vannak torzításai is

Számítógépes Grafika

Hajder L. és Baráth D.
hajder@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2019/2020. I. félév

Tartalom

1. Geometria modellezés
2. Egyszerű görbék és felületek
   - Általános leírás
   - Görbék
   - Felületek

Geometria modellezés feladata

- Geometriai alakzatok egzakt leírása
  - Pontok
  - Görbék
  - Testek
  - Felületek
  - stb.
- Kérdések
  - Hogyan adjuk meg ezeket a halmazokat?
  - Hogyan lehet őket kirajzolni?

Esettanulmány

- Keressük egy $P_0 = [x_y, y_0]$ pont és egy egyenes metszéspontját.
- Egyenes megadható többféleképpen, például:
  - $y = ax + b$
  - $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix} + \lambda \begin{bmatrix} 1 \\ a \end{bmatrix}$
- Tanulság:
  - Feladattól és
  - Dimenziótol is függhet a módszer kiválasztása

Tartalom

1 Geometria modellezés

2 Egyszerű görbék és felületek
- Általános leírás
- Görbék
- Felületek

Görbék, felületek leírása

- A görbéket, felületeket (amik közé az egyenes és a sík is tartozik) egy-egy ponthalmaznak tekintjük.
- Hogyan adjuk meg ezeket a halmazokat?
  - explicit: $y = f(x) \rightarrow$ mi van ha vissza akarjuk "fordítani"?
  - implicit: $f(x, y) = 0$
  - parametrikus: $\mathbf{p}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, t \in \mathbb{R}$

Példa # 1: függőleges tengelyű parabola 2D-ben

- A függőleges tengelyű (tehát nem általános) parabola leírása:
  - explicit: $y = ax^2 + bx + c$
  - implicit: $ax^2 - y + bx + c = 0$
  - parametrikus: $\mathbf{p}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} t \\ at^2 + bt + c \end{bmatrix}, t \in \mathbb{R}$

Példa # 2: általános helyzetű (térbeli) gömb

- Gömb paraméterei: középpont ($\begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T$) és sugár ($r$).
- Leírási módok:
  - implicit: $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0$
  - explicit: $z = \pm\sqrt{r^2 - (x - x_0)^2 - (y - y_0)^2} + z_0$
  - Parametrikus:

$$\mathbf{p}(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ y(u, v) \end{bmatrix} = \begin{bmatrix} r\sin(u)\cos(v) \\ r\sin(u)\sin(v) \\ r\ cos(u) \end{bmatrix}$$

$$v \in [-\pi, \pi), u \in [0, \pi]$$

Tartalom

1. Geometria modellezés
2. Egyszerű görbék és felületek
   - Általános leírás
   - Görbék
   - Felületek

Speciális parabola

- Az y tengelyű, $(0, p)$ fókuszpontú parabola
  - Implicit egyenlete: $x^2 - 4py = 0$
  - Explicit egyenlete: $y = \frac{x^2}{4p}$
  - Parametrikus egyenlete: $\mathbf{p}(t) = [t, \frac{t^2}{4p}]^T$, $t \in \mathbb{R}$

Kevésbé speciális parabola

- Mi van, ha a $\mathbf{c} = \begin{bmatrix} c_x & c_y \end{bmatrix}$ ponttal akarjuk eltolni az origóból a parabolát?
- Az implicit és explicit alakban be kell vinni a (c_x, c_y) koordinátákat (pl. implicitből $(x - c_x)^2 - 4p(y - c_y) = 0$ lesz)
- Parametrikus alakban egyszerűen $\mathbf{p}(t) + \mathbf{c}$ lesz az új alak.

Általános helyzetű parabola

- Implicit egyenlet: $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$, feltéve, hogy $B^2 = 4AC$.
  - Túl bonyolult, ritkán alkalmazzák
- Explicit: Még bonyolultabb, teljes négyzetté alakítással
- Parametrikus: függőleges tengelyű parabola elforgatással

$$\mathbf{p}(t) = \left[\cos(\phi)t + \sin(\phi)\frac{t^2}{4p} + c_x, \cos(\phi)\frac{t^2}{4p} - \sin(\phi)t + c_y\right]$$

$$c_x, c_y, t \in \mathbb{R}, \phi \in [-\pi, \pi)$$

Kör

- A $\mathbf{c} \in \mathbb{R}^2$ középpontú, r sugarú kör egy
  - Implicit egyenlete: $(x - c_x)^2 + (y - c_y)^2 - r^2 = 0$
  - Explicit alakban nem tudjuk az egész kört leírni egy függvénnyel (DE két darabban menne, pl. $\mathbf{c} = \mathbf{0}, r = 1$ mellett $y = \pm\sqrt{1 - x^2}$, ahol $x \in [-1, 1]$)
  - Parametrikus egyenlete: $\mathbf{p}(t) = r[\cos t, \sin t]^T + \mathbf{c}$, ahol $t \in [0, 2\pi)$

Ellipszis

- A $\mathbf{c} \in \mathbb{R}^2$ középpontú, nagytengelyével az x tengellyel párhuzamos, $2a$ nagytengelyű és $2b$ kistengelyű ellipszis egy
  - Implicit egyenlete: $\frac{(x-c_x)^2}{a^2} + \frac{(y-c_y)^2}{b^2} - 1 = 0$
  - Explicit alakban: lásd előbb
  - Parametrikus egyenlete: $\mathbf{p}(t) = [a\cos t, b\sin t]^T + \mathbf{c}$, ahol $t \in [0, 2\pi)$

Ellipszis

- De mi van, ha nem akarjuk, hogy x, y tengellyel párhuzamosak legyenek a tengelyeink?
  - Implicit egyenlet: $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$, feltéve, hogy $B^2 < 4AC$.
    - Elég bonyolult...
  - Parametrikus egyenlete: speciális ellipszisből báziscsere segítségével kapjuk: ha az új tengelyek $\mathbf{k}, \mathbf{l}$, akkor $\mathbf{p}(t) = a \cos t\mathbf{k} + b \sin t\mathbf{l} + \mathbf{c}$, ahol $t \in [0, 2\pi)$
    - $\mathbf{k}$ és $\mathbf{l}$ tengelyek egy szög (elforgatás) segítségével kifejezhetőek.

Szakasz

- Legyen adott két pont, $\mathbf{a}, \mathbf{b} \in \mathbb{E}^3$. A két ponton átmenő egyenes parametrikus egyenlete:

$$\mathbf{p}(t) = (1 - t)\mathbf{a} + t\mathbf{b},$$

  ahol $t \in \mathbb{R}$.
- Ha $t \in [0, 1]$, akkor az $\mathbf{a}, \mathbf{b}$ pontokat összekötő egyenes szakaszt kapjuk.

Egyenes

- Paraméteres egyenlet: $p_0 + \lambda d$
- Explicit egyenlete: $y = a'x + b'$
- Implicit egyenlete: $ax + by + c = 0$
  - Skálázásra érzéketlen: $\nu ax + \nu by + \nu c = 0$
- Implicit egyenlet vektoros alakban is írható: $A^T X = 0$ ahol

$$A = \begin{bmatrix} a & b & c \end{bmatrix}^T$$
$$X = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$$

  - A és X skálázásra érzéketlen.
  - X felfogható homogén koordinátaként. Valós koordinátára áttérés esetén a harmadik koordinátával osztani kell!

Két egyenes metszéspontja

- Implicit egyenletekkel:
  - Első egyenes: $A_1^T X = 0$
  - Második egyenes: $A_2^T X = 0$
  - Metszéspont: két egyenlet együttes megoldása $BX = 0$, ahol

$$B = \begin{bmatrix} A_1^T \\ A_2^T \end{bmatrix}$$

  - B mérete: 2×3, B nulltere (pontosabban: nullvektora) adja a megoldást
  - A nullvektor homogén koordinátákban adja az eredményt!

Két egyenes metszéspontja

- Paraméteres egyenletekkel:
  - Első egyenes: $p_0^{(1)} + \lambda^{(1)} d^{(1)}$
  - Második egyenes: $p_0^{(2)} + \lambda^{(2)} d^{(2)}$
- Metszéspont: a két egyenlet közös megoldása:

$$p_0^{(1)} + \lambda^{(1)} d^{(1)} = p_0^{(2)} + \lambda^{(2)} d^{(2)}$$

- Mivel 2D-ben vagyunk, ez valójában 2 egyenlet. Metszéspont a megfelelő lineáris egyenlet megoldásából jön:

$$\begin{bmatrix} \lambda^{(1)} \\ \lambda^{(2)} \end{bmatrix} = \begin{bmatrix} d^{(1)} & -d^{(2)} \end{bmatrix}^{-1} \left(p_0^{(2)} - p_0^{(1)} \right)$$

Spline-ok

Olyan fontos görbeleíró elem, hogy külön előadás lesz róla.

Görbék parametrikus alakja

- Deriváltak: $\mathbf{p}^{(i)}(t) = [x^{(i)}(t), y^{(i)}(t)]^T$, $t \in [...]$, $i = 0, 1, 2, ...$
- Ha a görbét egy mozgó pont pályájának tekintjük, akkor az első derivált a sebességnek tekinthető, a második a gyorsulásnak stb.
  - Valódi sebesség, gyorsulás esetén $t \in \mathbb{R}$ az időt jelöli.

Tartalom

1 Geometria modellezés

2 Egyszerű görbék és felületek
- Általános leírás
- Görbék
- Felületek

Megadás

- Explicit: $z = f(x, y)$
- Implicit: $f(x, y, z) = 0$
- Parametrikus: $\mathbf{p}(u, v) = [x(u, v), y(u, v), z(u, v)]^T$, $(u, v) \in [a, b] \times [c, d]$

Sík

- Paraméteres egyenlet: $p_0 + \lambda_1 d_1 + \lambda_2 d_2$
- Implicit egyenlete: $ax + by + cz + d = 0$
  - Skálázásra érzéketlen: $\nu ax + \nu by + \nu cz + \nu d = 0$
- Implicit egyenlet vektoros alakban is írható: $A^T X = 0$ ahol

$$A = \begin{bmatrix} a & b & c & d \end{bmatrix}^T$$
$$X = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$$

  - A és X skálázásra érzéketlen.
  - X felfogható homogén koordinátaként. Valós koordinátára áttérés esetén a negyedik koordinátával osztani kell!

Két sík metszésvonala

- Implicit egyenletekkel:
  - Első sík: $A_1^T X = 0$
  - Második sík: $A_2^T X = 0$
  - Metszésvonal: két sík együttes megoldása $BX = 0$, ahol

$$B = \begin{bmatrix} A_1^T \\ A_2^T \end{bmatrix}$$

  - B mérete: 2×4, B nulltere 2 vektor: v_1 és v_2
  - Egyenes pontjai: $\alpha v_1 + \beta v_2$
  - Pontok homogén koordinátában vannak, osztani kell az utolsó koordinátával

Két sík metszésvonala

- Paraméteres egyenletekkel:
  - Első sík: $p_0^{(1)} + \lambda_1^{(1)} d_1^{(1)} + \lambda_2^{(1)} d_2^{(1)}$
  - Második sík: $p_0^{(2)} + \lambda_1^{(2)} d_1^{(2)} + \lambda_2^{(2)} d_2^{(2)}$
- Metszésvonal: a két egyenlet közös megoldása:

$$p_0^{(1)} + \lambda_1^{(1)} d_1^{(1)} + \lambda_2^{(1)} d_2^{(1)} = p_0^{(2)} + \lambda_1^{(2)} d_1^{(2)} + \lambda_2^{(2)} d_2^{(2)}$$

- Mivel 3D-ben vagyunk, ez valójában 3 egyenlet, 4 ismeretlennel ($\lambda_1^{(1)}, \lambda_2^{(1)}, \lambda_1^{(2)}, \lambda_2^{(2)}$).
- Megoldás: egy paramétert szabadon megválaszthatunk, a másik három a lineáris egyenletrendszerből számítható.

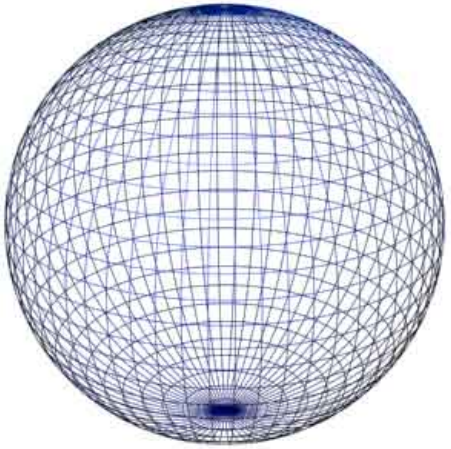
Felületek felületi normálisa

- Definíció szerint a felület normálisa egy adott pontjában az érintősík normálisa.
- A parametrikus alakban adott a felület: $\mathbf{n}(u,v) = \partial_u \mathbf{p}(u,v) \times \partial_v \mathbf{p}(u,v)$
  - a ∂_u az u paraméter szerinti deriválást jelenti, ∂_v a v szerintit.
- Implicit alakban ($f(x,y,z) = 0$) adott felületnél $\mathbf{n}(x,y,z) = \nabla f$, ahol $\nabla f = [f_x, f_y, f_z]^T$
  - ahol $f_x = \frac{\partial f}{\partial x}$, $f_y = \frac{\partial f}{\partial y}$ és $f_z = \frac{\partial f}{\partial z}$
  - Megjegyzés: mindez igaz más dimenzióban. Pl. 2D-ben érintősík helyett érintőegyenes, 4D-ben érintő alsík...stb.

Gömb

- Implicit: $(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 = r^2$
- Parametrikus:

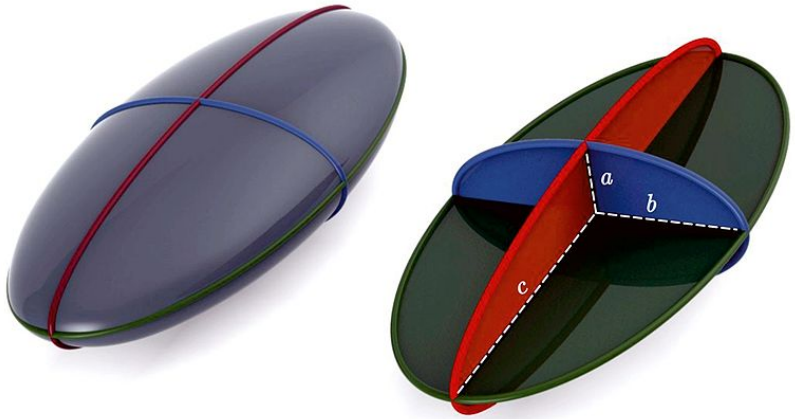
$$\mathbf{p}(u, v) = r[\cos u \sin v, \sin u \sin v, \cos v]^T + \mathbf{c}$$
$$(u, v) \in [0, 2\pi) \times [0, \pi]$$



Speciális ellipszoid

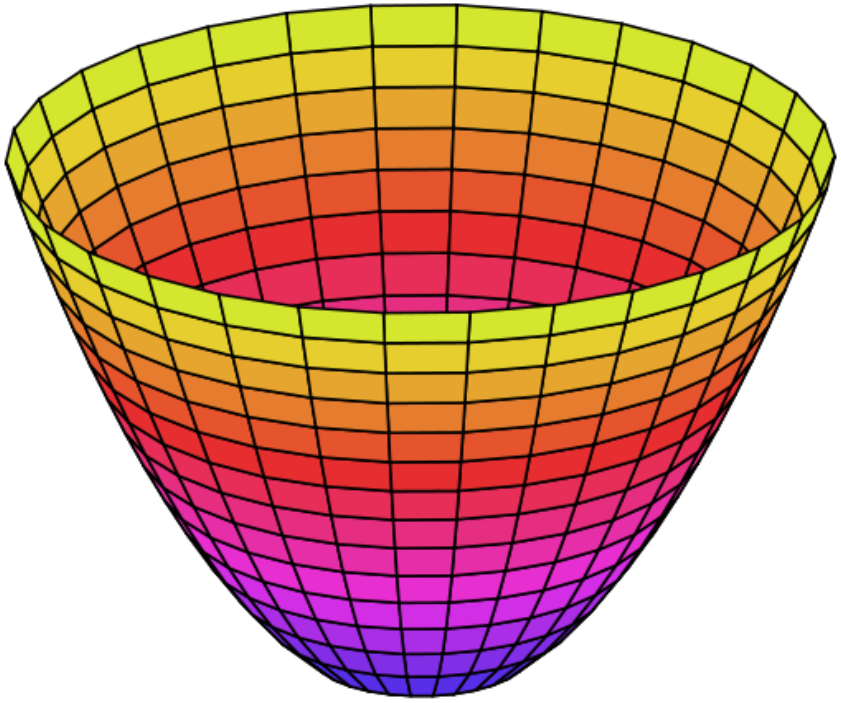
Speciális (tengelyei párhuzamosak a koordinátatengellyel) ellipszoid egyenlete:

- Implicit: $\frac{(x-c_x)^2}{a^2} + \frac{(y-c_y)^2}{b^2} + \frac{(z-c_z)^2}{c^2} - 1 = 0$
- Parametrikus:
  $\mathbf{p}(u,v) = [a\cos u \sin v, b\sin u \sin v, c\cos v]^T + \mathbf{c}$,
  $(u,v) \in [0, 2\pi) \times [0, \pi]$



Egyszerű paraboloid

- Explicit: $\frac{z}{c} = \frac{x^2}{a^2} + \frac{y^2}{b^2}$
- Parametrikus: $\mathbf{p}(u, v) = [u, v, c\frac{u^2}{a^2} + c\frac{v^2}{b^2}]^T$
- A paraboloid tengelye a z tengellyel párhuzamos



Amire figyelni érdemes

- Matematikában általában a felfelé mutató tengelynek a z tengelyt tekintik
- A fenti képletek is ennek megfelelően adják a ”várt” képet
- Grafikában viszont sokszor az y mutat felfelé!

Számítógépes Grafika

Hajder Levente és Baráth Dániel
hajder@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2019/2020. I. félév

Tartalom

1. Bevezetés
2. Görbék
   - Törtvonal
   - Felületi folytonosságok
   - Bezier-görbe
   - B-spline
   - Spline variánsok
   - Felosztott (subdivision) görbék
   - Felosztott (subdivision) felületek

Geometriai modellezés

- Görbéket, felületeket számatalan módszerrel le tudunk írni
  - Explicit egyenlet
  - Implicit egyenlet
  - Parametrikus megadás
- A leírható görbék felületek köre korlátos, leírás sokszor nehézkes.
- Cél: általános görbét felületet matematikailag leírni.

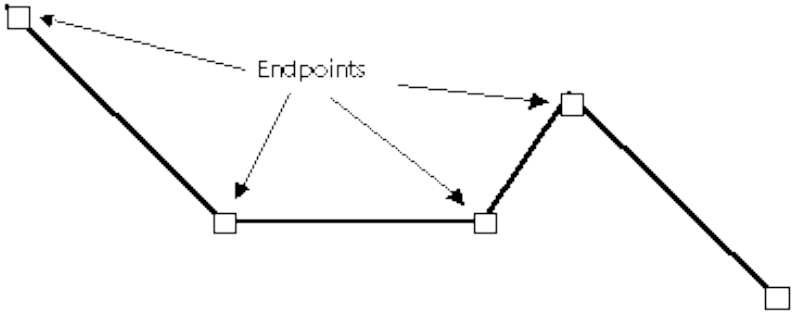
Tartalom

1 Bevezetés

2 Görbék
- Törtvonal
- Felületi folytonosságok
- Bezier-görbe
- B-spline
- Spline variánsok
- Felosztott (subdivision) görbék
- Felosztott (subdivision) felületek

Törtvonal (polyline)

- Az elv egyszerű:
  - Adottak a görbe pontjai
  - Egyenes szakaszokkal összekötjük
- Meglehetősen ”szögletes” érzést kelt



Törtvonal (polyline)

- A görbe matematikai összefüggésekkel is megadható.
- t paraméter $[0, 1]$ intervallumba esik.
- Pontok számát N-nel jelöljük.

$$\mathbf{poly}(\mathbf{t}) = \sum_{i=1}^{N-1} a_i \mathbf{p_i} + b_i \mathbf{p_{i+1}}$$

$$a_i = \frac{t - t_i}{t_{i+1} - t_i} \quad ha \quad t_i < t < t_{i+1}$$

$$b_i = \frac{t_{i+1} - t}{t_{i+1} - t_i} \quad ha \quad t_i < t < t_{i+1}$$

- ahol például $t_i = i/N$

Tartalom

1 Bevezetés

2 Görbék
- Törtvonal
- Felületi folytonosságok
- Bezier-görbe
- B-spline
- Spline variánsok
- Felosztott (subdivision) görbék
- Felosztott (subdivision) felületek

Felületi folytonosságok

- Törtvonal szögletes, esztétikai érzékünket bántja a sok törés.
- A görbékre különböző folytonossági kategóriákat definiálhatunk:
  - C^0: folytonos a görbe, a pontjai kapcsolódnak egymáshoz
  - C^1: folytonos az első deriváltja a görbének: nincsen ugrás a deriváltakban
  - C^2: folytonos a második deriváltja a görbének: nincsen ugrás a deriváltakban
    - Emberi szem számára kellemes látványt nyújt.
    - Meghajlított ág C^2-folytonos.

Tartalom

1 Bevezetés

2 Görbék
- Törtvonal
- Felületi folytonosságok
- Bezier-görbe
- B-spline
- Spline variánsok
- Felosztott (subdivision) görbék
- Felosztott (subdivision) felületek

Bezier-görbe

- Pierre Bézier (Renault-mérnök) találta ki.
- Folytonos felületet ad
- Bezier görbe alapja a Bernstein polinom.

$$(t + (1 - t))^m = \sum_{i=0}^{m} \binom{m}{i} t^i (1 - t)^{m-i}$$

- Ennek pedig a binomiális tétel:

$$(a + b)^m = \sum_{i=0}^{m} \binom{m}{i} a^i b^{m-i}$$

Bezier-görbe

- A Bernstein polinomok adják az egyes kontrollpontokhoz tartozó súlyokat
- Mintha baricentrikus koordináták lennének
- Teljes görbe (parametrikus)

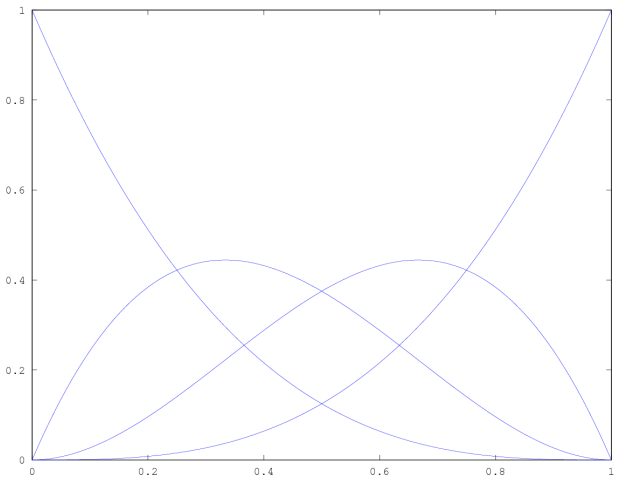
$$\mathbf{p}(t) = \sum_{i=0}^{m} B_{i,m}^{Bezier} \mathbf{p_i}$$

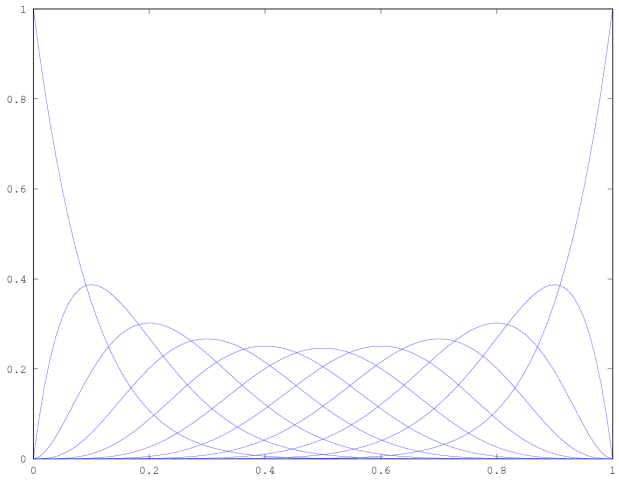
- ahol

$$B_{i,m}^{Bezier} = \binom{m}{i} t^i (1-t)^{(m-i)}$$

Bezier-görbe tulajdonságai

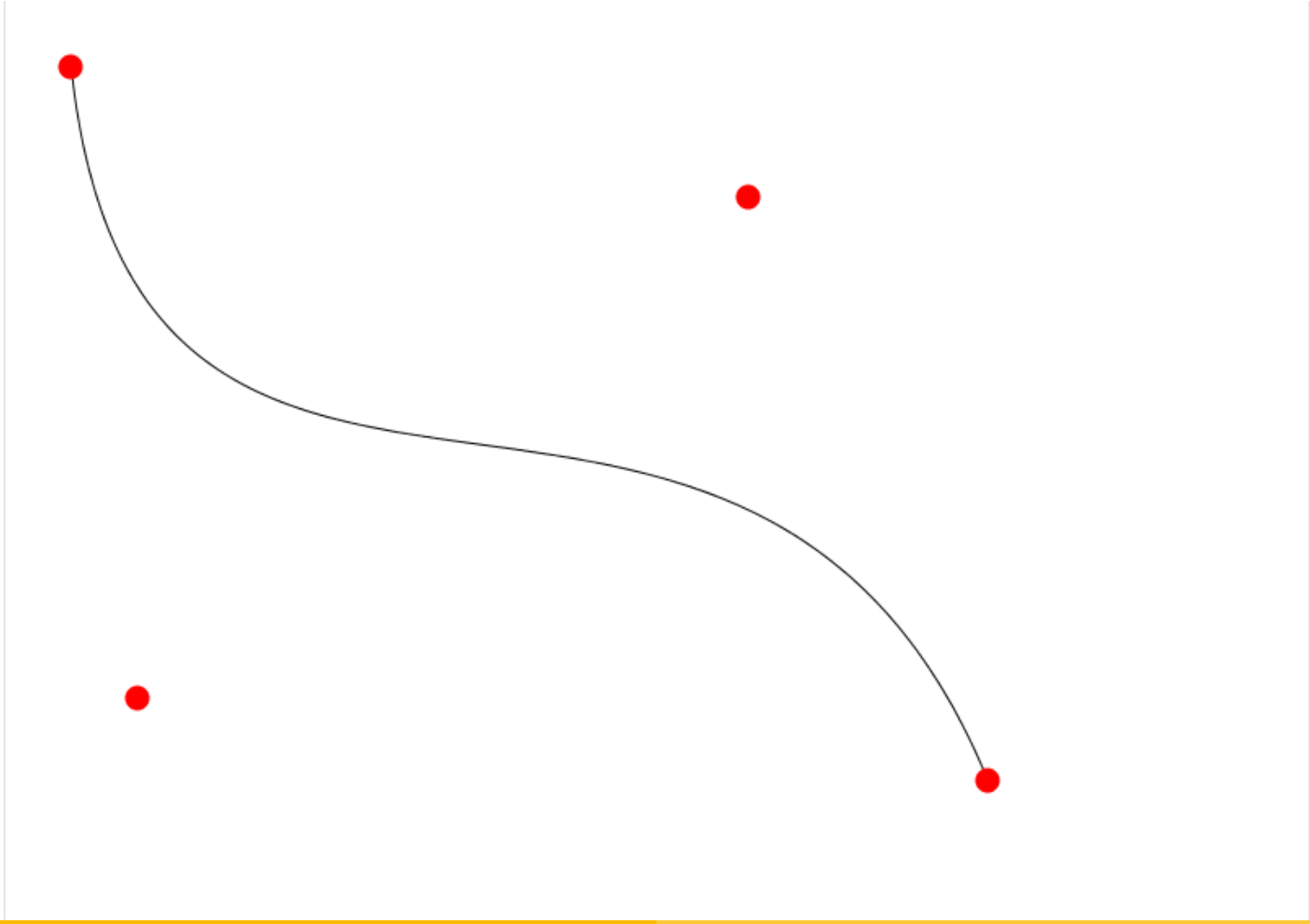
- ”Végtelenül” folytonos
  - Hiszen polinomokból áll, $\rightarrow$ bárhányszor differenciálható
- Mindegyik kontrollpont súlya pozitív $t \in (0, 1)$ tartományban
  - Végpontot leszámítva mindegyik kontrollpont ”beleszól” a görbe alakulásába





Bezier-görbe tulajdonságai

- Végponton átmegy a görbe



Bezier-görbe tulajdonságai

- Bázisfüggvény fontos tulajdonsága

$$B_{i+1,m}^{Bezier} = tB_{i,m-1}^{Bezier} + (1-t)B_{i+1,m-1}^{Bezier}$$

- Tehát az m-edfokú polinom a két szomszédos $(m-1)$-edfokú polinomból kiszámítható.
- Mindez a de Casteljau algoritmus alapja.
  - Adott t paraméternél a Bezier-görbe pontja megszerkeszthető
  - Fokszámnak megfelelő iterációt használunk
  - Első lépés: kontollpontokból törtvonalat készítve, t / $(1-t)$ aránynak megfelelően a törtvonal éleit osztjuk.
  - Iteráció: Előző szakaszpontokból újra törtvonalat készítve, az új törtvonalakat újra arányosan felosztjuk.
  - Utolsó lépés: ha a törtvonal már csak egy szakaszból áll, megkapjuk a Bezier-görbe adott pontját.

Tartalom

1 Bevezetés

2 Görbék
- Törtvonal
- Felületi folytonosságok
- Bezier-görbe
- B-spline
- Spline variánsok
- Felosztott (subdivision) görbék
- Felosztott (subdivision) felületek

B-spline

- A spline C^2 folytonos felület
- Előállítása: törtvonalból kiindulva, súlyozással
- Bezier-görbéhez hasonlóan a kontrollpontok súlyozásával állítjuk elő
- Súlyok előállítás rekurzióval:

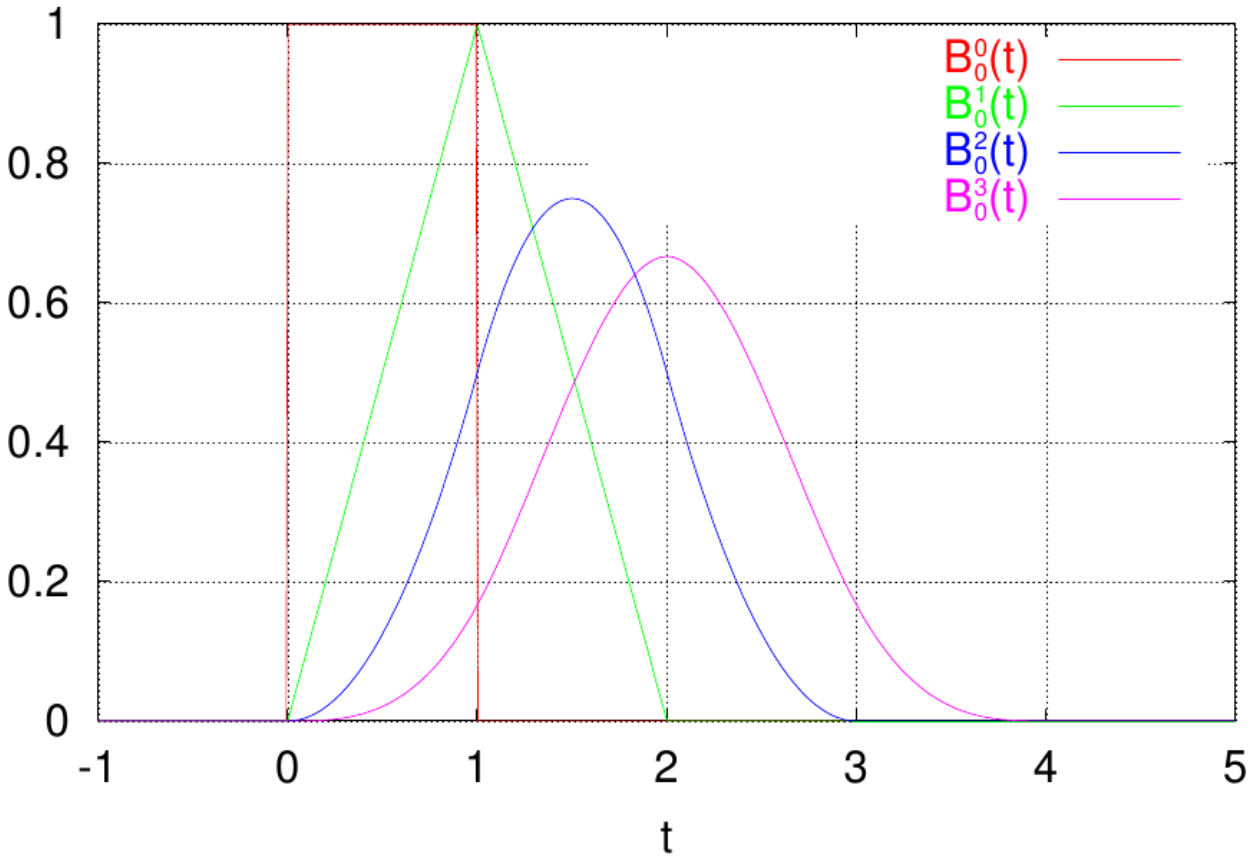
$$B_i^k(t) = \frac{t - t_i}{t_{i+k-1} - t_i} B_i^{k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1}^{k-1}(t)$$

$$B_i^0(t) = \begin{cases} 1 & \textit{if} \quad t_i < t < t_{i+1} \\ 0 & \textit{else} \end{cases}$$

- $t_1, \ldots, t_m$ a kontrollpontokhoz tartozó paraméterek.

B-spline

- A rekurzió alatt a súlyok ”lekerekednek”



Tartalom

1 Bevezetés

2 Görbék
- Törtvonal
- Felületi folytonosságok
- Bezier-görbe
- B-spline
- Spline variánsok
- Felosztott (subdivision) görbék
- Felosztott (subdivision) felületek

Spline variánsok

- Ha a t_i paraméterek egymástól azonos távolságra vannak: uniform B-Spline
- Ha a t_i paraméterek változó távolságban vannak: non-uniform B-Spline (NUBS)
- Ha két súlyhoz tartozó paraméter megegyezik , azaz $t_i = t_{i+1}$ non-uniform rational B-Spline (NURBS)
  - Két kontrollpont egymásra helyezésével a ponthoz közelebb lehet húzni a spline-t.

Spline variánsok

- A spline pontjai nem mennek át a kontrollpontokon.
- Megoldás: válasszuk úgy ki a vezérlő pontokat, hogy a megadott pontokon átmenjen a spline.
- Lineáris egyenlet, ismeretlenek a $\mathbf{p_i}$ vezérlő pontok.

$$\mathbf{q_j} = \sum_i a_i(t)\mathbf{p_i}$$

- 2D-ben két egyenlet pontonként.
  - Végpontra irányt (első derivált t szerint) is megköthetünk.
  - Irány meghatározása egyszerű: $\mathbf{v}(t) = \sum_i a_i^{'}(t)\mathbf{p_i}$
  - Több független spline is összeköthető, ha a végpontoknál az irány megegyezik.

Tartalom

1 Bevezetés

2 Görbék
- Törtvonal
- Felületi folytonosságok
- Bezier-görbe
- B-spline
- Spline variánsok
- Felosztott (subdivision) görbék
- Felosztott (subdivision) felületek

Felosztott (subdivision) görbék

- Cél: Törtvonalas közelítésből szép ”sima” görbe előállítása
- Ötlet: iteratív algoritmus
  - Törtvonalból kiindulva, új pontokat adjunk hozzá
  - Pontok helyét finomítsuk

Felosztott (subdivision) görbe: algoritmus

- Bementet: N pontot tartalmazó törtvonal $\mathbf{p_i}$ koordinátákkal
- Subdivision algoritmus két lépésből áll
  1. Felezőpontok hozzáadása. Új pont: $\mathbf{q_i} = (\mathbf{p_i} + \mathbf{p_{i+1}})/2$
  2. Pontok finomítása: $\mathbf{p_i} = \mathbf{p_i}/2 + (\mathbf{q_{i-1}} + \mathbf{q_i})/4$

Tartalom

1 Bevezetés

2 Görbék
- Törtvonal
- Felületi folytonosságok
- Bezier-görbe
- B-spline
- Spline variánsok
- Felosztott (subdivision) görbék
- Felosztott (subdivision) felületek

Felosztott (subdivision) felületek

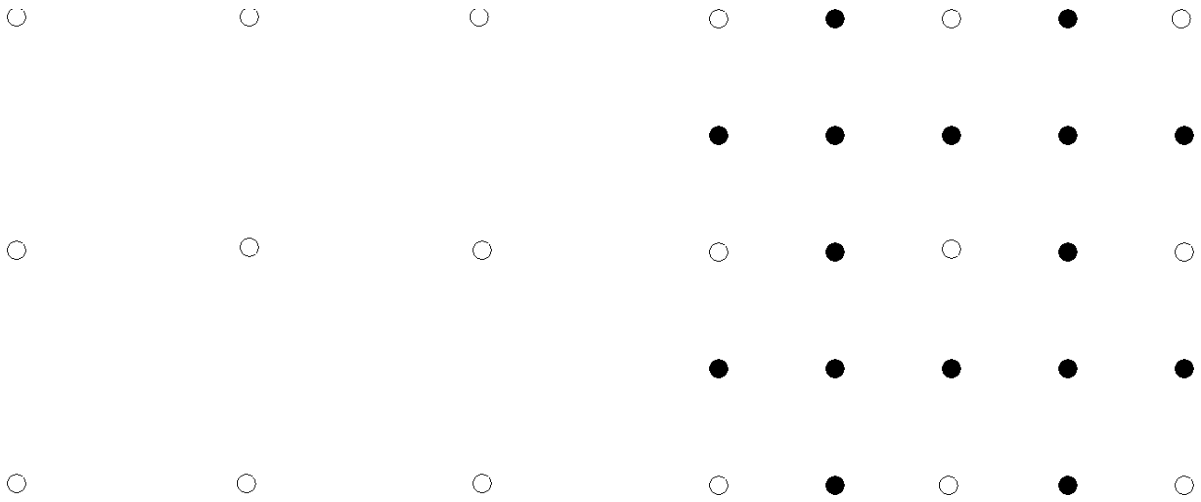
- Elve megegyezik a felosztott görbék előállításával.
- Iteratív az algoritmus
  - Térbeli pontokat sokszorozza
  - Pontokat helyét finomítja

Catmull-Clarc felosztott felület

- Pontokat négyzetrácsként képzeljük el (dupla index)
- Algoritmus lépései:
  1. Térbeli pontokat sokszorozza
  2. Új pontok kiszámítása
     1. Határpontok
     2. Belső pontok #1
     3. Belső pontok #2
  3. Régi pontok pontosítása

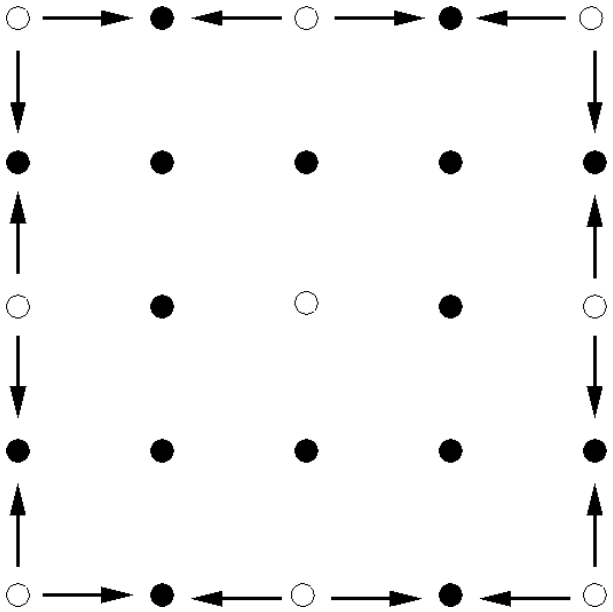
Catmull-Clarc felosztott felület

- $N \times N$ rácsból $(2N-1) \times (2N-1)$ rácsot készítünk.
- Oszlopokat/sorokat kell beszúrni.



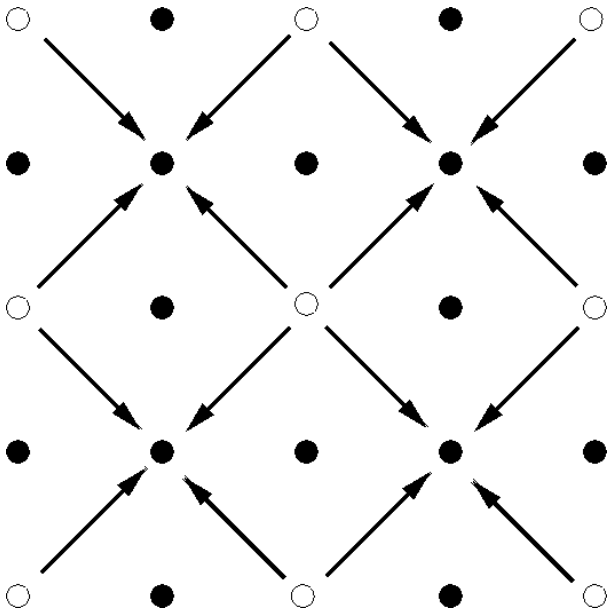
Catmull-Clarc felosztott felület

- Beszúrt pontoknak még nincsen értéke.
- Új szélső (határ) pontok: szomszéd határpontok átlaga



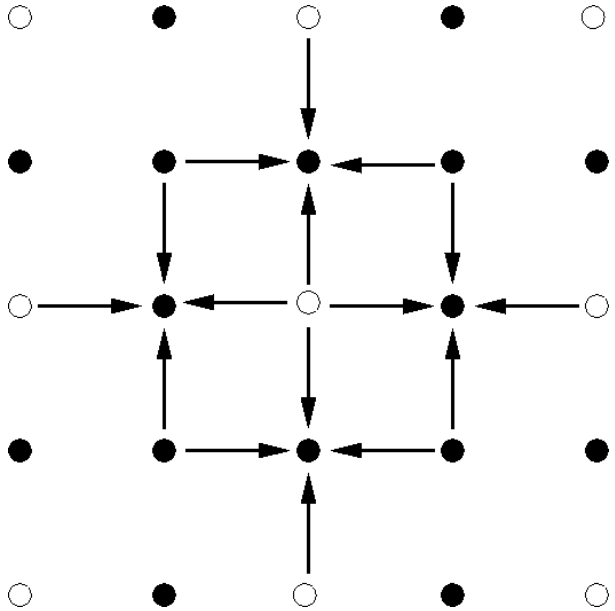
Catmull-Clarc felosztott felület

- Belső beszúrt pontoknak még nincsen értéke
- Interpoláljuk a négy (előző iteráció óta meglévő) szomszédos pont átlagával, ahol az átlóban vannak az ismert pontok



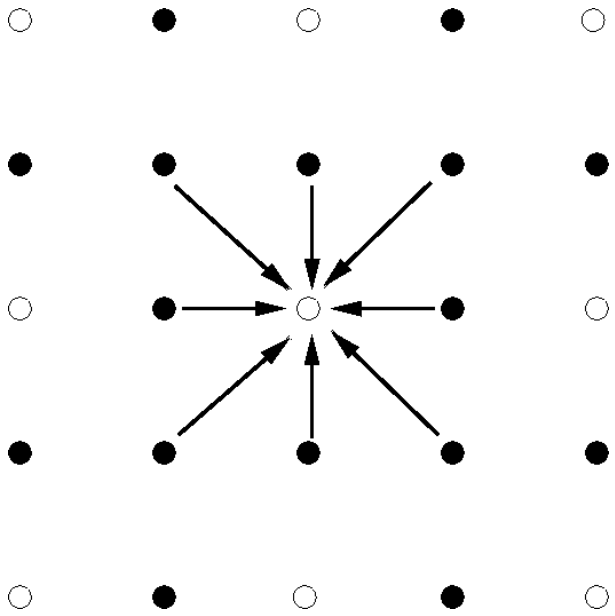
Catmull-Clarc felosztott felület

- Pontoknak még mindig nincsen értéke
- Interpoláljuk a négy szomszédos pont átlagával
  - Kettő szomszédot előző iterációban számoltuk.
  - Másik két szomszédot a jelenlegi iterációban.



Catmull-Clarc felosztott felület

- Régi pont: nem ebben az iterációban szúrtuk be.
- Új érték itt is két szám átlaga
  - Első szám: A pontban meglevő (régi) érték
  - Második szám: nyolc szomszédjának átlaga



Számítógépes Grafika

Hajder Levente és Baráth Dániel
hajder@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2019/2020. I. félév

Tartalom

1 A fény és anyagok
- Anyagok
- Fényforrás modellek
- Fény-felület kölcsönhatás
- Fényvisszaverési modellek
- Buckatérkép
- Színmodellek

A fény és anyagok

- A fény elektromágneses hullám
- Az anyagokat olyan színűnek látjuk, amilyen színű fényt visszavernek
  - A visszaverés egyaránt függ az anyag és a megvilágítás ”színétől”
- Különböző anyagok különböző módon viselkednek a fénnyel szemben

Tartalom

1 A fény és anyagok
- Anyagok
- Fényforrás modellek
- Fény-felület kölcsönhatás
- Fényvisszaverési modellek
- Buckatérkép
- Színmodellek

Felületek osztályozása

- Emittáló felületek
  - Fénykibocsátó felületek emittáló anyagnak hívjuk
  - Ezeket hívjuk fényforrásoknak, ilyen a Nap, a lámpa stb.
- Diffúz felületek
  - A diffúz vagy matt felületeket minden irányból nézve ugyanolyan színűnek látjuk
  - Ilyen például a frissen meszelt fal vagy a homok stb.
  - A diffúz felület a beérkező fénysugár energiáját minden irányban azonos intenzitással veri vissza

Felületek osztályozása

- Spekuláris felületek
  - Tükröző felületek, az ideális fénytörés irányába verik vissza nagyrészt a beérkező fényt
- Átlátszó felületek
  - Ezeken a felületeken áthalad a fény, a beérkező fénysugár energiájának java részét áteresztik

Felületek osztályozása

- Áttetsző felületek
  - Ezek a beérkező fény nagy részét magukba engedik, de csak kis része lép ki az anyagból
  - Pl. tej, bőr
- Anizotróp felületek
  - A felületet a tengelye körül forgatva, a beeső és visszaverődő szögeket tartva is változik a színe
  - Mint például a CD

Tartalom

1 A fény és anyagok
- Anyagok
- Fényforrás modellek
- Fény-felület kölcsönhatás
- Fényvisszaverési modellek
- Buckatérkép
- Színmodellek

Fényforrás modellek

Fény

A fény elektromágneses hullám

Absztrakt fényforrások

- Ambiens fény
- Irány fényforrás
- Pont fényforrás
- Reflektorfény (spotlight)

Fényforrás modellek

Fény

A fény elektromágneses hullám

Absztrakt fényforrások

- Ambiens fény
- Irány fényforrás
- Pont fényforrás
- Reflektorfény (spotlight)

Fényforrás modellek

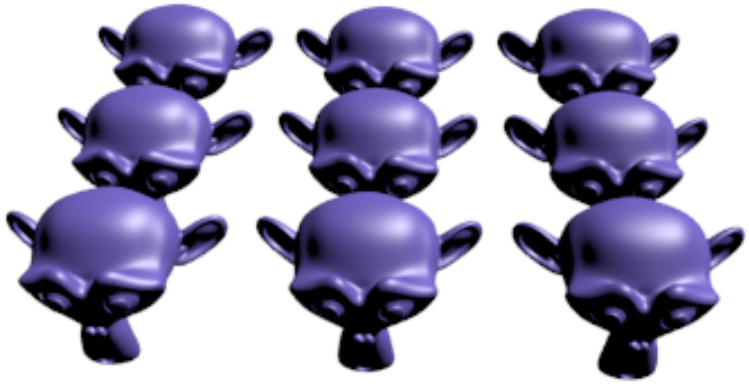
Ambiens fény

- Fénysugarak minden irányba egyenlő mértékben világítanak
- Távolság az intenzitást nem befolyásolja

Fényforrás modellek

Irány fényforás

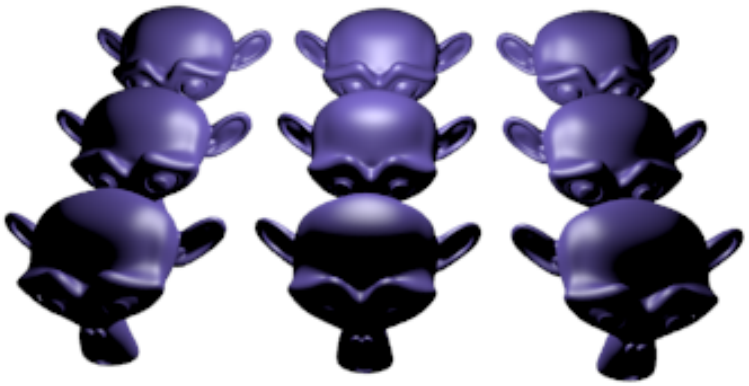
- Fénysugarak párhuzamosak
- Távolsággal a fény intenzitása nem csökken



Fényforrás modellek

Pont fényforrás

- Egy adott pontból indulnak ki a fénysugarak
- Fizika: a távolság négyzetével fordítottan arányos a fény intenzitás
- Szimulációkban a fakulást (falloff) meg lehet adni skaláris, lineáris, kvadratik tagokkal



Fényforrás modellek

Reflektorfény (spotlight)

- Egy adott pontból indulnak ki a fénysugarak
- A fénnyalábot egy kör alapú végtelen gúla határozza meg
- A távolság négyzetével fordítottan arányos a fény intenzitás (valóságban)

Tartalom

1 A fény és anyagok
- Anyagok
- Fényforrás modellek
- Fény-felület kölcsönhatás
- Fényvisszaverési modellek
- Buckatérkép
- Színmodellek

BRDF

- Legyen L^{in} egy adott irányból a felület egy pontjára beérkező, L pedig az onnan visszavert fény intenzitása
- Jelölje **l** a fényforrás felé mutató egységvektort, **v** a nézőpont felé mutató egységvektort, **n** pedig a felületi normálist az adott pontban. A θ legyen az **n** és **l** által bezárt szög
- Ekkor a *kétirányú visszaverődéses eloszlási függvény*, BRDF (bi-directional reflection distribution function) a következő:

$$f_r(\mathbf{l}, \mathbf{v}) = \frac{L}{L^{in} \cos\theta}$$

Jelölések

- $\mathbf{v} := \omega$ a nézeti irány, azaz a szem/kamera fele mutató vektor
- $\mathbf{l} := -\omega'$ a megvilágító, a fényt "adó" pont fele mutató vektor, ekkor a beesési irány $-\mathbf{l}$ $(= \omega')$
- $\mathbf{n}$ a felületi normális
- $\mathbf{v}, \mathbf{l}, \mathbf{n}$ egységvektorok
- θ a $\mathbf{l}$ és a $\mathbf{n}$ által bezárt szög

Helmholtz-törvény

- Helmholtz-féle szimmetria: a fénysugár megfordítható
- Azaz: $f_r(\omega', \omega) = f_r(\omega, \omega')$
- Ez két dologért is jó:
  - Garantálja, hogy végsősoron a radiancia csökken.
  - Nézhetjük ”visszafelé” a sugarakat.

Ideális visszaverődés

Visszaverődési törvény

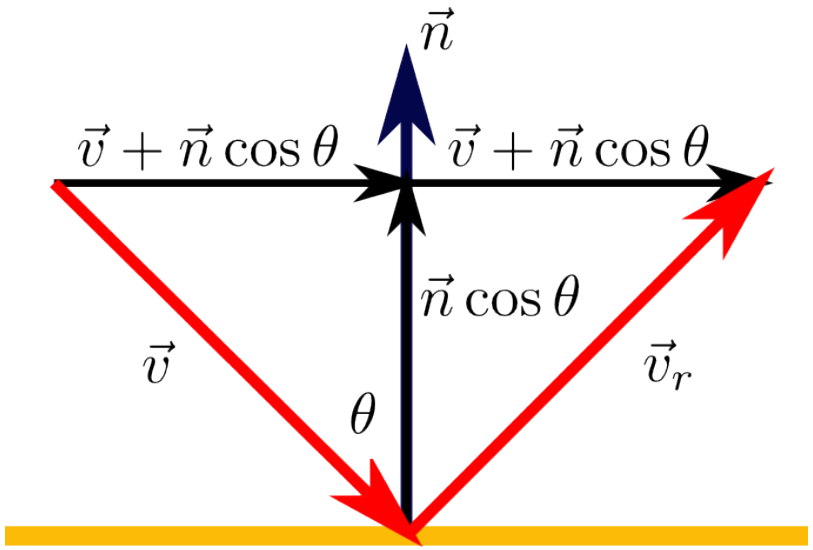
A beesési irány ($-\mathbf{l}$), a felületi normális ($\mathbf{n}$), és a kilépési irány ($\mathbf{r}$) egy síkban van, valamint a beesési szög (θ) megegyezik a visszaverődési szöggel (θ').

Ideális visszaverődés

- Az ideális tükör csak az **r** tükörirányba ver vissza.
- $$f_r(\mathbf{x}, \mathbf{v}, \mathbf{l}) = k_r \frac{\delta(\mathbf{r} - \mathbf{v})}{\cos\theta}$$
- δ a *Dirac-delta* függvény, ami egy általánosított függvény, amely minden nemnulla paraméterre nullát ad, de a valós számok felett vett integrálja 1.
- A k_r visszaverődési együttható a *Fresnel-együttható*. Ez függ az anyag törésmutatójából, és az elektromos vezetési képességéből származik.
- A *Fresnel-együttható* a visszavert és beeső energia hányadát fejezi ki.

Visszaverődési irány

- Általános esetben, egy **v** beeső vektorból a visszaverődési- vagy tükörirány:
- $\mathbf{v}_r = \mathbf{v} - 2\mathbf{n}(\mathbf{n} \cdot \mathbf{v})$
- Mivel $\cos\theta = -\mathbf{n} \cdot \mathbf{v}$
- Mindez csak akkor igaz, ha **n** és **v** vektorok egységnyi hosszúak!
- Általános esetben: $\mathbf{v}_r = \mathbf{v} + 2\mathbf{n}\cos\theta$



Ideális törés

Snellius-Descartes törvény

A beesési irány ($-\mathbf{l}$), a felületi normális ($\mathbf{n}$), és a törési irány ($\mathbf{t}$) egy síkban van, valamint $\eta = \frac{\sin\theta}{\sin\theta'}$, ahol η az anyagok relatív törésmutatója.

Néhány törésmutató

- Vákuum 1.0
- Levegő 1.0003
- Víz 1.3333
- Üveg 1.5
- Gyémánt 2.417

Ideális törés

Snellius-Descartes törvény

A beesési irány ($-\mathbf{l}$), a felületi normális ($\mathbf{n}$), és a törési irány ($\mathbf{t}$) egy síkban van, valamint $\eta = \frac{\sin\theta}{\sin\theta'}$, ahol η az anyagok relatív törésmutatója.

Néhány törésmutató

- Vákuum 1.0
- Levegő 1.0003
- Víz 1.3333
- Üveg 1.5
- Gyémánt 2.417

Ideális törés

- Jelölje **t** az ideális törési irányt.
- Az ideális tükörhöz hasonlóan kapjuk:

$$f_r(\mathbf{x}, \mathbf{v}, \mathbf{l}) = k_t \frac{\delta(\mathbf{t} - \mathbf{v})}{\cos\theta}$$

Ideális törés

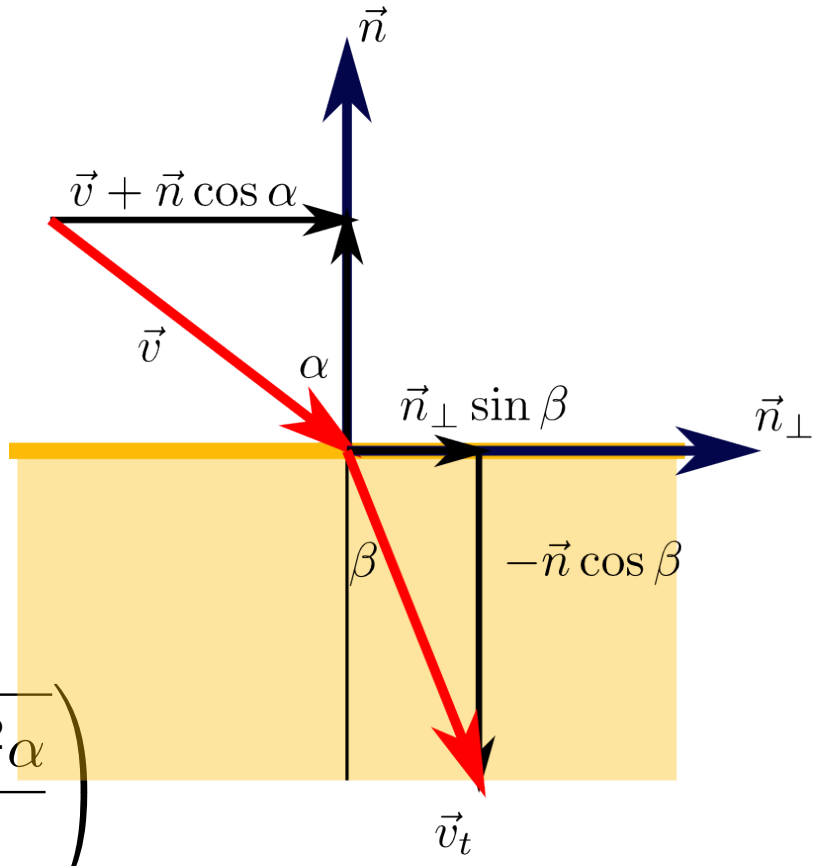
- Jelölje **t** az ideális törési irányt.
- Az ideális tükörhöz hasonlóan kapjuk:

$$f_r(\mathbf{x}, \mathbf{v}, \mathbf{l}) = k_t \frac{\delta(\mathbf{t} - \mathbf{v})}{\cos\theta}$$

Törési irány

- Snellius-Descartes törvény: $\eta = \frac{\sin\alpha}{\sin\beta}$
- $\mathbf{v}_t = \mathbf{n}_\perp \sin\beta - \mathbf{n}\cos\beta$
- $\mathbf{n}_\perp = \frac{\mathbf{v}+\mathbf{n}\cos\alpha}{\sin\alpha}$
- $\mathbf{v}_t = \frac{\mathbf{v}}{\eta} + \mathbf{n}\left(\frac{\cos\alpha}{\eta} - \cos\beta\right)$
- $\cos\beta = \sqrt{1-\sin^2\beta} = \sqrt{1-\frac{\sin^2\alpha}{\eta^2}}$

$$\mathbf{v}_t = \frac{\mathbf{v}}{\eta} + \mathbf{n}\left(\frac{\cos\alpha}{\eta} - \sqrt{1-\frac{sin^2\alpha}{\eta^2}}\right)$$



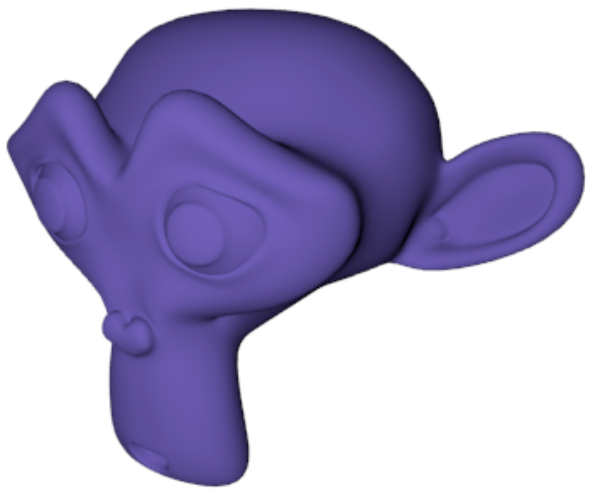
Tartalom

1. A fény és anyagok
   - Anyagok
   - Fényforrás modellek
   - Fény-felület kölcsönhatás
   - Fényvisszaverési modellek
   - Buckatérkép
   - Színmodellek

Lambert-törvény

- Optikailag durva, *diffúz* felületek leírására jó.
- Feltételezés: a visszavert fénymennyiség nem függ a nézeti iránytól.
- Helmholtz-törvényt miatt akkor a bejövő iránytól sem függhet, azaz konstans:

$$f_r(\mathbf{x}, \mathbf{v}, \mathbf{l}) = k_d$$



Spekuláris visszaverődés - Phong modell

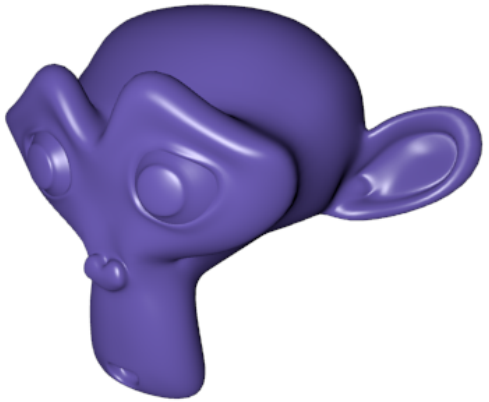
- A tükörirányban intenzíven visszaverő, de attól távolodva gyorsan elhaló ”csillanás” adható meg vele.
- Legyen ϕ az $\mathbf{r}$ tükörirány és a $\mathbf{v}$ nézeti irány által bezárt szög.
- Ekkor $\cos\phi = \mathbf{r} \cdot \mathbf{v}$
- Olyan függvényt keresünk, ami $\phi = 0$-ra nagy, de gyorsan elhal.
- $$f_r(\mathbf{x}, \mathbf{v}, \mathbf{l}) = k_s \frac{\cos^n \phi}{\cos\theta}$$

  Nem szimmetrikus!

Spekuláris visszaverődés - Phong modell

$n = 5$

$n = 25$



$n = 50$

Spekuláris visszaverődés - Phong-Blinn modell

- Legyen **h** a nézeti irány és a megvilágító pont fele mutató vektorok felezővektora.
- $$\mathbf{h} = \frac{\mathbf{v} + \mathbf{l}}{\|\mathbf{v} + \mathbf{l}\|}$$
- Legyen δ a **h** és az **n** normálvektor által bezárt szög.
- Ekkor $\cos\delta = \mathbf{h} \cdot \mathbf{n}$
- $$f_r(\mathbf{x}, \mathbf{v}, \mathbf{l}) = k_s \frac{\cos^n \delta}{\cos\theta}$$
- Nagyon hasonló az egyszerű Phong modellhez, kicsit gyorsabban számítható.

Spektrális képszintézis

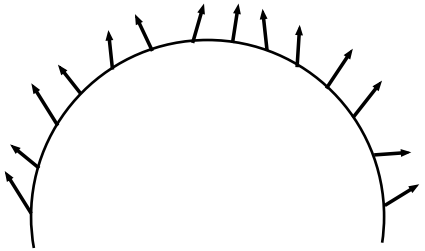
- Különböző hullámhosszú fény máshogy viselkedik a felületeken.
- Színérzet a látható tartományban levő elektromágneses hullámok integrálja a három érzékelőnek megfelelően.
- Fényjelenségeket minden hullámhosszon külön kellene nézni.
  - Rendkívül számításigényes.
  - R,G,B komonensekkel jól közelíthetjük.

Tartalom

1 A fény és anyagok
- Anyagok
- Fényforrás modellek
- Fény-felület kölcsönhatás
- Fényvisszaverési modellek
- Buckatérkép
- Színmodellek

Érdes felületek képzése

- Érdes felületek rengeteg poligonnal (háromszöghálóval) képezhetők.
  - Modell bonyolult, nehéz módosítani.
  - Renderelést lassítja.
- Trükk: egyszerű modellhez finom sűrű normálvektormezőt adunk meg

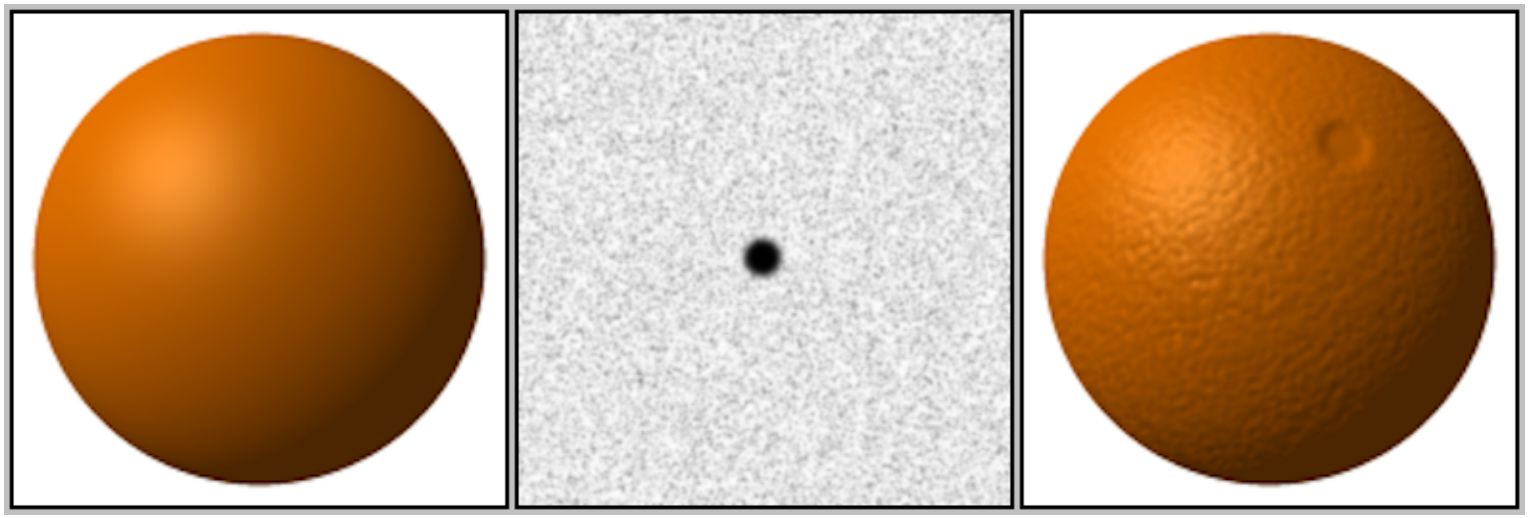


Érdes felületek képzése

- Durva pozíció + finom normálvektorok jó közelítés, ha
  - a felület nagyjából folytonos
  - mélységingadozás kicsi a felületen
- Mélység megadás képként: buckatérkép (bump map)
  - Textúraleképzéshez hasonlóan, képként szokás megadni
  - Buckatérkép leírhatja a mélységváltozást vagy a normálvektorokat (3D: 3 színkomponens).

Esettanulmány: narancs

- Példa (eredeti modell, buckatérkép, új modell):

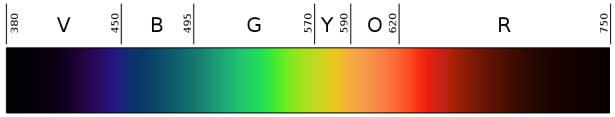


Tartalom

1 A fény és anyagok
- Anyagok
- Fényforrás modellek
- Fény-felület kölcsönhatás
- Fényvisszaverési modellek
- Buckatérkép
- Színmodellek

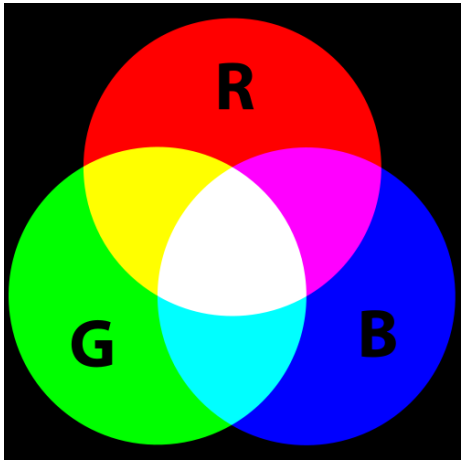
Színmodellek

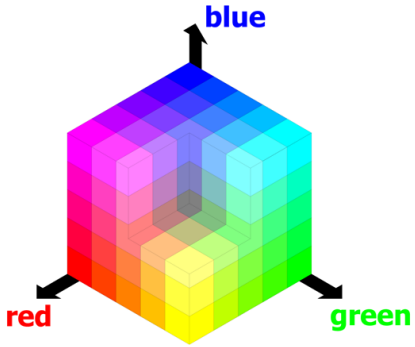
- Fény: elektromágneses hullám
- Emberi szem által látható fény: alapszínek keveréséből
- Alapszínek: szivárvány színei



Színmodellek

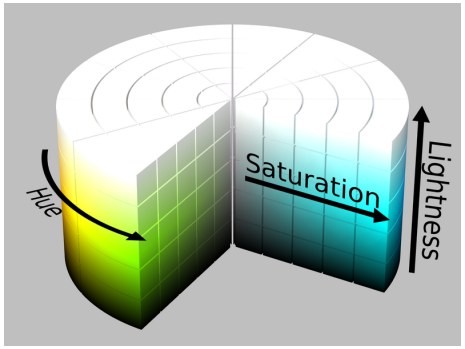
- Monitor színmodellje: RGB
- Majdnem az összes látható szín kikeverhető

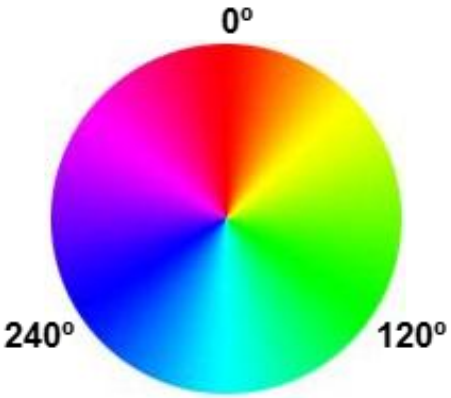




Színmodellek

- ”Emberibb” színmodell: HSL (HSB,HSV)
- Három komponens: hue (színárnyalat), saturation (telítettség), lightness (fényesség)





Számítógépes Grafika

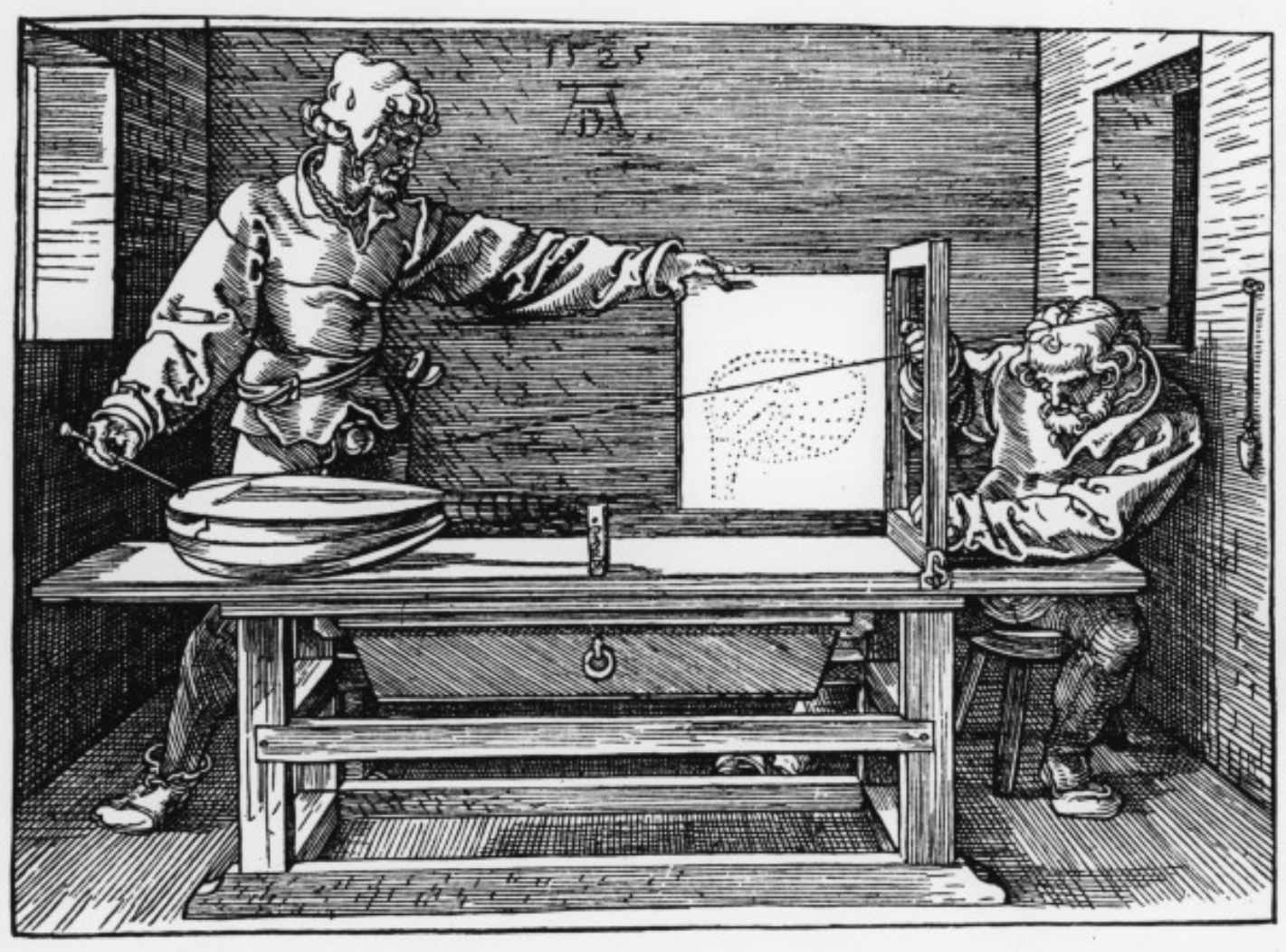
Hajder Levente és Baráth Dániel
hajder@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

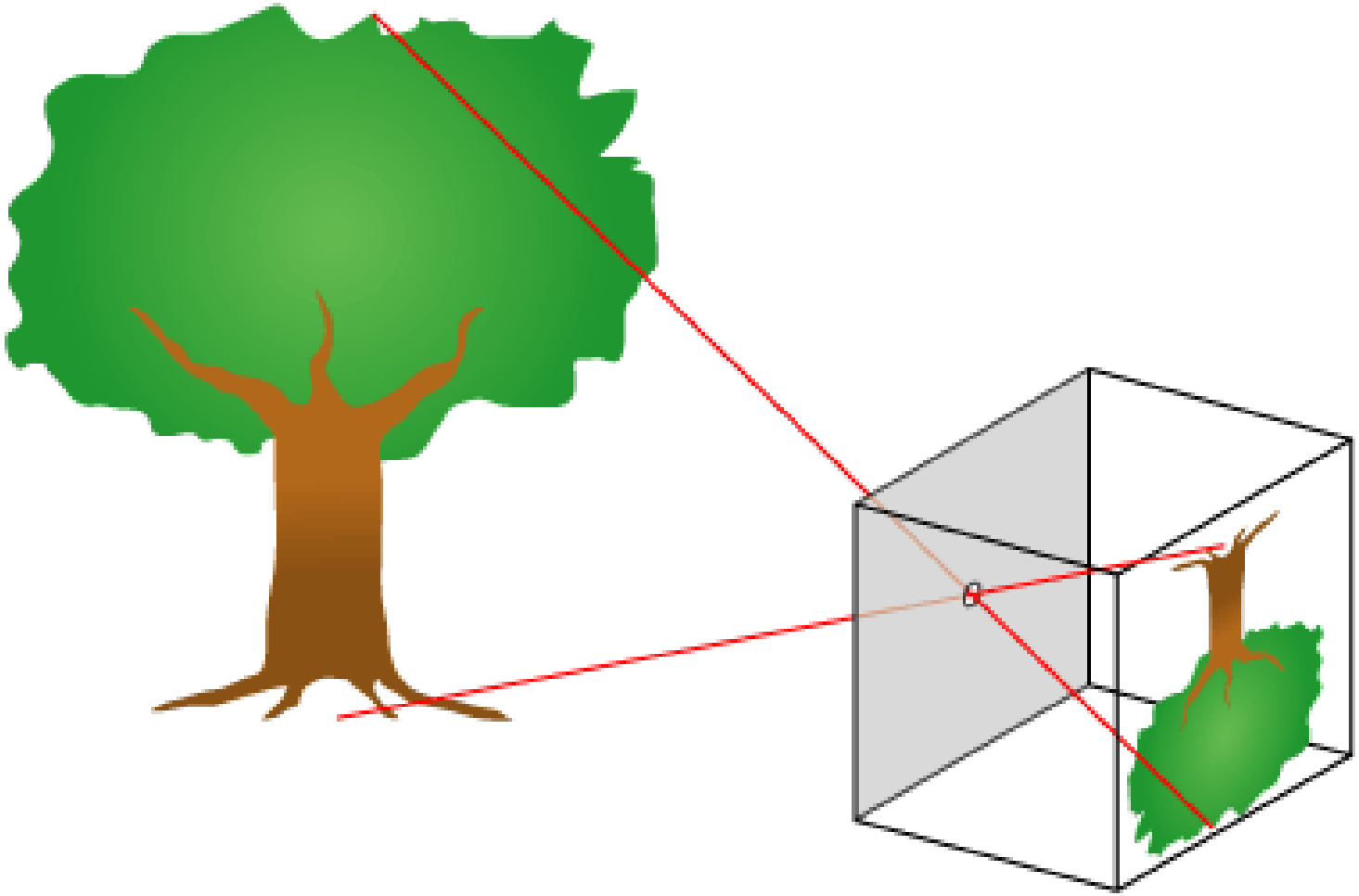
2019/2020. I. félév

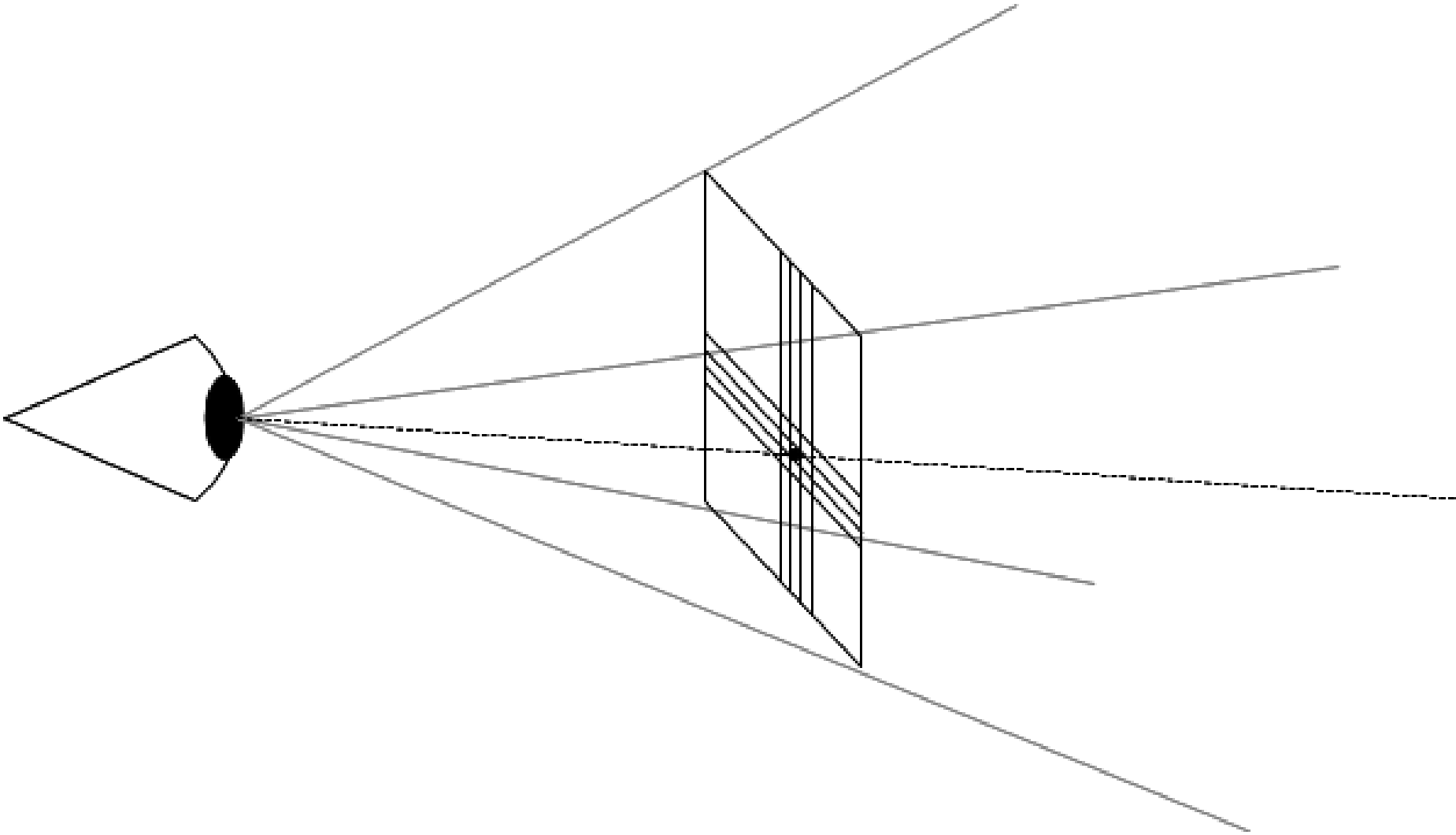
Tartalom

1. Reycasting: motiváció
2. Raycasting
3. Sugarak indítása
4. Metszések
   - Sugár és sík metszéspontja
   - Sugár és háromszög metszéspontja
   - Sugár és poligon metszéspontja
   - Sugár és gömb metszéspontja
   - Transzformált objektumok
5. RayCasting versus RayTracing



Albrecht Dürer, 1525





Motiváció

- Tekintsünk minden pixelre úgy, mint egy kis ablakra a világra
- Milyen színértéket vegyen fel ez a pixel? $\rightarrow$ Nézzük meg, mi látszik onnan a világból és az alapján rendeljünk hozzá a pixelhez egy színt!

Raycasting

```
Minden pixelre:
  Indítsunk egy sugarat a színtérbe

  Minden objektumra a színtérben:
   Nézzük meg, hogy metszi-e a sugár az objektumo

  A legközelebbi metszett objektum
  színével színezzük ki a pixelt
```

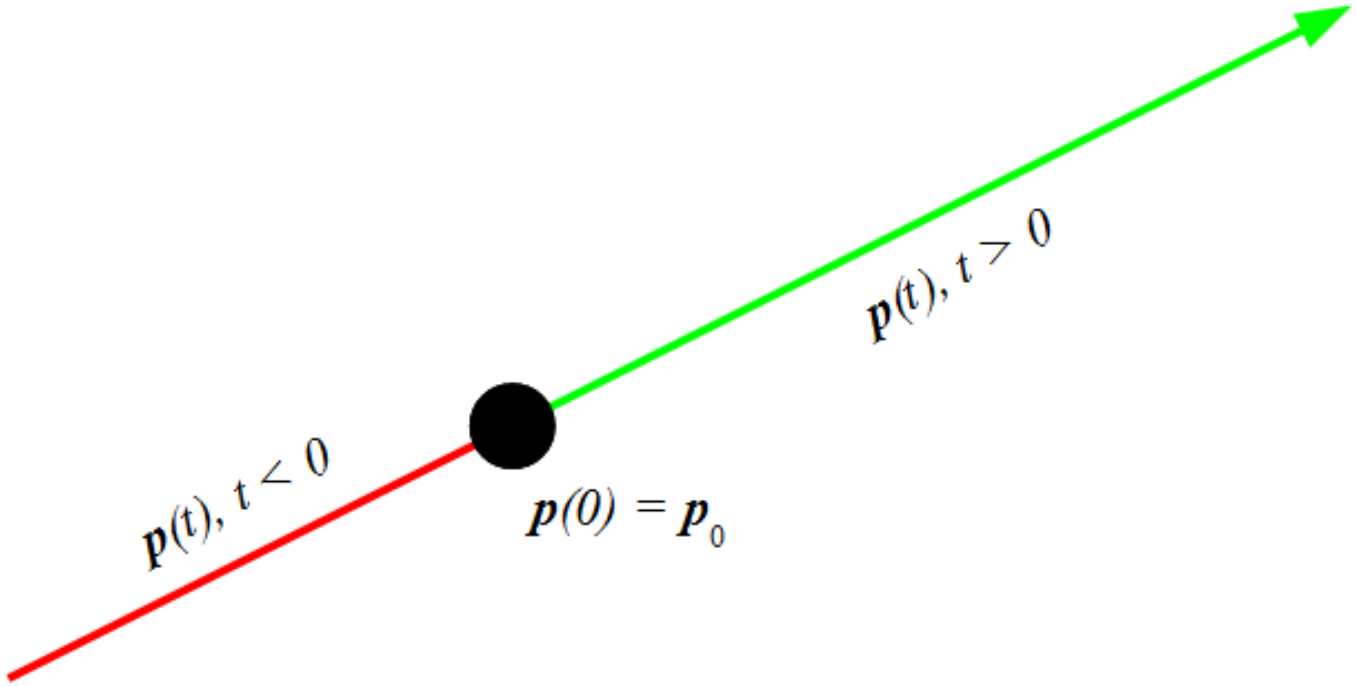
Sugár

- A sugárnak van
  - egy $\mathbf{p}_0$ kiindulási pontja
  - és egy $\mathbf{v}$ iránya
- A parametrikus sugár:

$$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{v},$$

  ahol $t > 0$ (félegyenes!).
- $t = 0$?, $t < 0$? sugár kezdőpontja, sugár mögötti részek

Sugár

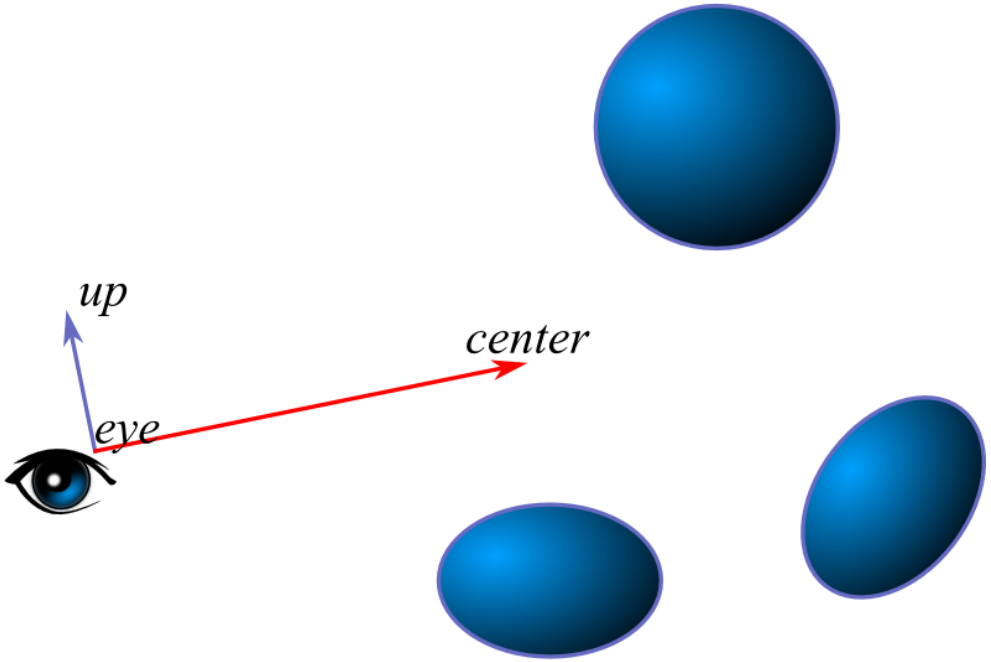


Kérdés

- Honnan indítsuk a sugarat?
- Milyen irányba küldjük a sugarat?
- Hogyan metszük el a sugarat akármivel?

Sugarak indítása

- A szempozicióból indítunk sugarakat minden pixel középpontján keresztül
- Most: középpontosan szeretnénk vetíteni egy szembe, a vetítési sík egy négyszögletes részét megfeleltetve a képernyőnek
- Szem/kamera tulajdonságok:
  - szempozíció (**eye**),
  - egy pont amire néz (**center**),
  - felfele irányt megadó vektor a világban (**up**),
  - nyílásszög, amekkora szögtartományt lát (*fovx, fovy*).
  - (vetítővászon mérete. Most legyen adott: $2\tan\left(\frac{fovx}{2}\right) \times 2\tan\left(\frac{fovy}{2}\right)$ nagyságú)
- Ezek segítségvel fogjuk megadni az (i, j) pixel világbeli koordinátáit


up
center
eye

Sugarak indítása

Keressük a kamera saját $\mathbf{u}, \mathbf{v}, \mathbf{w}$ (jobbkezes!) koordinátarendszerét!

- Nézzen a kamera $-Z$ irányba!

$$\mathbf{w} = \frac{\mathbf{eye} - \mathbf{center}}{|\mathbf{eye} - \mathbf{center}|}$$

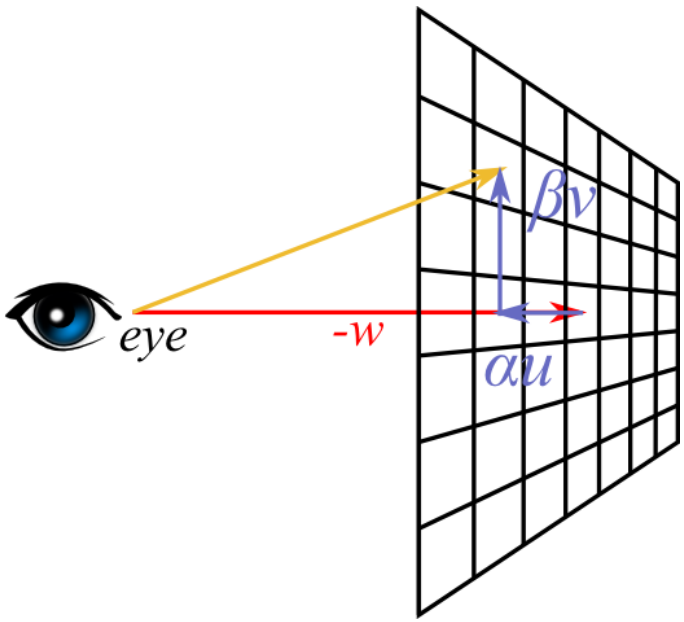
- Az X tengely legyen merőleges mind **w**-re, mind az **up** irányra!

$$\mathbf{u} = \frac{\mathbf{up} \times \mathbf{w}}{|\mathbf{up} \times \mathbf{w}|}$$

- Az Y tengely merőleges **u**-ra és **w**-re is:

$$\mathbf{v} = \mathbf{w} \times \mathbf{u}$$

(i, j) pixel koordinátái



- Legyen **p** az i, j pixel középpontja, a vetítősík egységnyi távolságra a nézőponttól! Ekkor

$$\mathbf{p}(i, j) = \mathbf{eye} + (\alpha\mathbf{u} + \beta\mathbf{v} - \mathbf{w}).$$

- Ahol

$$\alpha = \tan\left(\frac{fovx}{2}\right) \cdot \frac{i - width/2}{width/2},$$

$$\beta = \tan\left(\frac{fovy}{2}\right) \cdot \frac{height/2 - j}{hight/2}.$$

A sugár egyenlete

- A sugár egy félegyenes, amit kezdőpontjával és irányvektorával adhatunk meg.
- Legyen $\mathbf{p}_0$ a sugár kezdőpontja, $\mathbf{v}$ pedig az irányvektora, ekkor

$$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{v}, \quad t \geq 0$$

  megadja a sugár összes pontját.
- Most a sugarak kezdőpontját az előbbieknek megfelelően számoljuk, azaz $\mathbf{p}_0 = \mathbf{p}(i, j)$
- A sugár irányvektora pedig $\mathbf{v} = \frac{\mathbf{p}(i,j) - \mathbf{eye}}{|\mathbf{p}(i,j) - \mathbf{eye}|}$

Metszések

- A sugárkövető programok futásidejük döntő részében metszéseket fognak végezni
- Nézzük meg néhány egyszerű geometriai elemmel vett metszetét a sugárnak
- A sugarunk mindig a fent is látott $\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{v}$ alakú, ahol feltesszük a továbbiakban, hogy $|\mathbf{v}| = 1$

Metszések: parametrikus sugár-implicit felület

- Legyen adva egy $f(\mathbf{x}) = 0$ implicit egyenlet, ami meghatározza a metszeni kívánt felületünket ($\mathbf{x} \in \mathbb{R}^3$)
- A sugarunk egyenlete $\forall t \in [0, \infty)$-re meghatároz egy pontot a térben $\rightarrow$ tegyük be ezt a képletet az implicit egyenletbe!
- Tehát a következő egyenletet kell megoldanunk t-re:

$$f(\mathbf{p}(t)) = 0$$

- A kapott t-től függően a következő esetek állhatnak fenn:
  - Ha $t > 0$, akkor a sugarunk előtt van a felület és metszi
  - Ha $t = 0$ a sugár a felületről indul
  - Ha $t < 0$, akkor a sugár ”mögött” van a felület és metszi a sugár egyenese a felületet (de nekünk $t > 0$ kell!)

Metszések: parametrikus sugár-parametrikus felület

- Legyen adva egy $\mathbf{r}(u, v) = [r_x(u, v), r_y(u, v), r_z(u, v)]^T$ parametrikus felület
- Kell: találni egy olyan t sugárparamétert, amihez létezik (u, v), hogy
$$\mathbf{r}(t) = \mathbf{r}(u, v)$$
- Ez három ismeretlenes (t, u, v), három egyenletes (x, y, z koordinátánként egy) egyenletrendszer
- A t ugyanúgy ellenőrizendő, mint előbb, de most az (u, v)-re is figyeljünk, hogy a felületünk paramétertartományának megengedett részén van-e (általában $(u, v) \in [0, 1]^2$ kell)!

Egyenes és implicit sík metszéspontja

- Síkot megadhatunk implicit alakban: $Ax + By + Cz + D = 0$
- A

$$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{v} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + t \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

sugár egyenese metszi a síkot, ha

$$A(x_0 + tx) + B(y_0 + ty) + C(z_0 + tz) + D = 0$$

Egyenes és implicit sík metszéspontja

- Ezt t-re átrendezve adódik

$$t(Ax + By + Cz) + Ax_0 + By_0 + Cz_0 + D = 0$$
$$t = -\frac{Ax_0 + By_0 + Cz_0 + D}{Ax + By + Cz}$$

- Látható a sík a nézőpontunkból, ha $t > 0$

Egyenes és normálvektoros sík metszéspontja

- Legyen $\mathbf{q}_0$ a sík egy pontja, $\mathbf{n}$ a normálvektora,
- Legyen $\mathbf{p}_0$ ez egyenes egy pontja, $\mathbf{v}$ az irányvektora.
- Az egyenes egyenlete:

$$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{v}$$

- A sík egyenlete:

$$\langle \mathbf{n}, \mathbf{q} - \mathbf{q}_0 \rangle = 0$$

- minden $\mathbf{q}$ pontja a síknak kielégíti ezt az egyenletet

Egyenes és normálvektoros sík metszéspontja

- Behelyettesítve $\mathbf{p}(t)$-t a $\mathbf{q}$ helyére:

$$\langle \mathbf{n}, \mathbf{p}_0 + t\mathbf{v} - \mathbf{q}_0 \rangle = 0,$$

$$\langle \mathbf{n}, \mathbf{p}_0 \rangle + t\langle \mathbf{n}, \mathbf{v} \rangle - \langle \mathbf{n}, \mathbf{q}_0 \rangle = 0,$$

$$t = \frac{\langle \mathbf{n}, \mathbf{q}_0 \rangle - \langle \mathbf{n}, \mathbf{p}_0 \rangle}{\langle \mathbf{n}, \mathbf{v} \rangle} = \frac{\langle \mathbf{n}, \mathbf{q}_0 - \mathbf{p}_0 \rangle}{\langle \mathbf{n}, \mathbf{v} \rangle},$$

  ha $\langle \mathbf{n}, \mathbf{v} \rangle \neq 0$.
- A sugár metszi a síkot, ha: $t > 0$.
- Ha $\langle \mathbf{n}, \mathbf{v} \rangle = 0$, akkor az egyenes párhuzamos a síkkal, és így vagy nincs metszéspontjuk, vagy az egyenes a síkon fut

Egyenes és parametrikus sík metszéspontja

- Síkot megadhatunk egy $\mathbf{q}$ pontjával és $\mathbf{i}, \mathbf{j}$ kifeszítő vektorokkal is: $\mathbf{s}(u, v) = \mathbf{q} + u\mathbf{i} + v\mathbf{j}$
- Metszéspont a $\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{v}$ sugár egyenesével: keressük t és u, v-t úgy, hogy

$$\mathbf{p}(t) = \mathbf{s}(u, v)$$

- Beírva a képleteket adódik

$$\mathbf{p}_0 + t\mathbf{v} = \mathbf{q} + u\mathbf{i} + v\mathbf{j}$$

- Átrendezve kapjuk, hogy

$$\mathbf{p}_0 - \mathbf{q} = -t\mathbf{v} + u\mathbf{i} + v\mathbf{j}$$

Egyenes és parametrikus sík metszéspontja

- Ez három ismeretlenes, három lineáris egyenletből álló egyenletrendszer, ami megoldható, ha $\mathbf{v}, \mathbf{i}, \mathbf{j}$ lineárisan nem összefüggő
- Mátrix alakban:

$$\begin{bmatrix} p_{0x} - q_x \\ p_{0y} - q_y \\ p_{0z} - q_z \end{bmatrix} = \begin{bmatrix} -v_x & i_x & j_x \\ -v_y & i_y & j_y \\ -v_z & i_z & j_z \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix}$$

- Látjuk a síkot, ha $t > 0$ (most $u, v \in \mathbb{R}$ a felület paramétertartománya, ez teljesülni fog)

Háromszög megadása

- Egyértelműen megadható három csúcsával.
- Ha $\mathbf{a}, \mathbf{b}, \mathbf{c}$ a háromszög csúcsai, akkor a hozzátartozó sík pont-normálvektoros implicit megadásához a sík
  - egy pontja $\mathbf{a}, \mathbf{b}, \mathbf{c}$ bármelyike
  - normálvektora

$$\mathbf{n} = \frac{(\mathbf{c} - \mathbf{a}) \times (\mathbf{b} - \mathbf{a})}{\|(\mathbf{c} - \mathbf{a}) \times (\mathbf{b} - \mathbf{a})\|},$$

ahol $\times$ a vektoriális szorzást jelöli, és ekkor $\mathbf{n}$ egységnyi hosszúságú.

Háromszög és egyenes metszéspontja

Két lépés alapján meghatározhatjuk

1. Először számítsuk ki az egyenes és háromszög síkjának metszéspontját
   - Egyenes egy pontja ismert: pl. háromszög **a** csúcsa.
   - Egyenes két irányvektora megadható: pl. $\mathbf{b} - \mathbf{a}$ és $\mathbf{c} - \mathbf{a}$
   - Metszéspontot jelöljük **p**-vel.
2. Majd nézzük meg, hogy a metszéspont a háromszögön belül van-e.

Pont a háromszögön vizsgálat: egyszerűbb módszer

- A háromszög három oldala és a normálvektor alapján két félsík adódik.
- Minden oldal esetén a háromszöget csak az egyik félsík tartalmazza.

$$\langle(\mathbf{p}-\mathbf{a})\times(\mathbf{b}-\mathbf{a}),\mathbf{n}\rangle \geq 0$$
$$\langle(\mathbf{p}-\mathbf{b})\times(\mathbf{c}-\mathbf{b}),\mathbf{n}\rangle \geq 0$$
$$\langle(\mathbf{p}-\mathbf{c})\times(\mathbf{a}-\mathbf{c}),\mathbf{n}\rangle \geq 0$$

- A három frappáns képlet viszonylag sok számítást tartalmaz.

Pont a háromszögön vizsgálat: gyorsabb módszer

- Legyenek $\lambda_1, \lambda_2, \lambda_3$ a $\mathbf{p}$ pont háromszögön belüli baricentrikus koordinátái, úgy hogy

$$\mathbf{p} = \lambda_1 \mathbf{a} + \lambda_2 \mathbf{b} + \lambda_3 \mathbf{c}$$

- p Akkor, és csak akkor van a $\triangle$-ön belül, ha

$$0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1.$$

Pont a háromszögön vizsgálat: gyorsabb módszer

- Először számítsuk ki az egyenes és a háromszög síkjának metszéspontját, ez legyen **p** (már ha létezik).
- Legyenek $\lambda_1, \lambda_2, \lambda_3$ a **p** pont háromszögön belüli baricentrikus koordinátái, úgy hogy

$$\mathbf{p} = \lambda_1 \mathbf{a} + \lambda_2 \mathbf{b} + \lambda_3 \mathbf{c}$$

- *p* Akkor, és csak akkor van a $\triangle$-ön belül, ha

$$0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1.$$

Pont a háromszögön vizsgálat: gyorsabb módszer

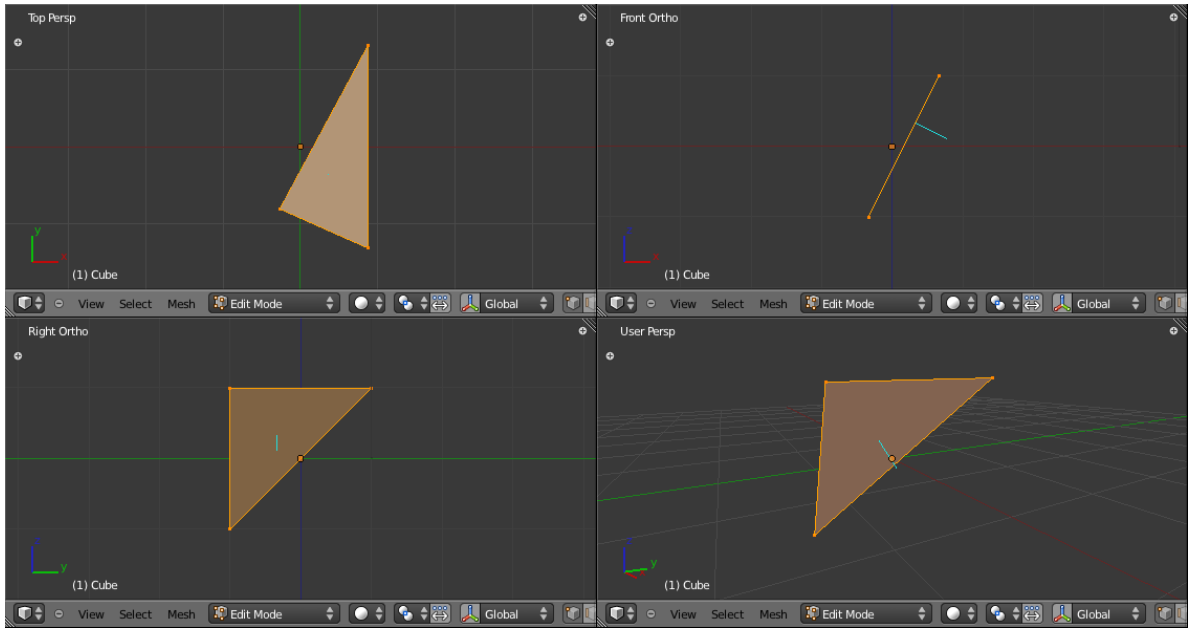
- Tudjuk, hogy $\mathbf{p} = [x, y, z]^T = \lambda_1 \mathbf{a} + \lambda_2 \mathbf{b} + \lambda_3 \mathbf{c}$. Ekkor

$$\begin{aligned} x &= \lambda_1 a_x + \lambda_2 b_x + \lambda_3 c_x \\ y &= \lambda_1 a_y + \lambda_2 b_y + \lambda_3 c_y \\ z &= \lambda_1 a_z + \lambda_2 b_z + \lambda_3 c_z, \end{aligned}$$

  ill. $\lambda_1 + \lambda_2 + \lambda_3 = 1 \Rightarrow \lambda_3 = 1 - \lambda_1 - \lambda_2$
- A gyorsabb számolásért vegyük a fentinek egy síkra vett vetületét
- A koordinátasíkok közül (XY, XZ vagy YZ) arra vegyük a háromszög 2D vetületét, amelyre a háromszög vetületének területe a legnagyobb! $\rightarrow$ a háromszög és a sík normálisa leginkább "egyállású"
- A vetülethez egyszerűen elhagyjuk z, y vagy x egyenletét, megfelelően

Pont a háromszögön vizsgálat: gyorsabb módszer

Azt tengely kell választani, amelyik mentén a legnagyobb a háromszög normálvektorának abszolút értéke.
(Így biztos nem fordulhat elő, hogy a háromszög merőleges a síkra, és csak egy szakasz marad belőle!)



Pont a háromszögön vizsgálat: gyorsabb módszer

- Pl. legyen a z a választott tengely. Ekkor

$$\begin{aligned} x &= \lambda_1 a_x + \lambda_2 b_x + \lambda_3 c_x \\ y &= \lambda_1 a_y + \lambda_2 b_y + \lambda_3 c_y \end{aligned}$$

- Behelyettesítve $\lambda_3 = 1 - \lambda_1 - \lambda_2$-t, és rendezve:

$$\begin{aligned} x &= \lambda_1 (a_x - c_x) + \lambda_2 (b_x - c_x) + c_x \\ y &= \lambda_1 (a_y - c_y) + \lambda_2 (b_y - c_y) + c_y \end{aligned}$$

Pont a háromszögön vizsgálat: gyorsabb módszer

- Rendezve λ_1, λ_2-re kapjuk:

$$\lambda_1 = \frac{(b_y - c_y)(x - c_x) - (b_x - c_x)(y - c_y)}{(a_x - c_x)(b_y - c_y) - (b_x - c_x)(a_y - c_y)}$$
$$\lambda_2 = \frac{-(a_y - c_y)(x - c_x) + (a_x - c_x)(y - c_y)}{(a_x - c_x)(b_y - c_y) - (b_x - c_x)(a_y - c_y)}$$

- A nevező csak degenerált háromszög esetén lehet nulla.
- **p** akkor és csak akkor van a háromszögön belül, ha

$$0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1.$$

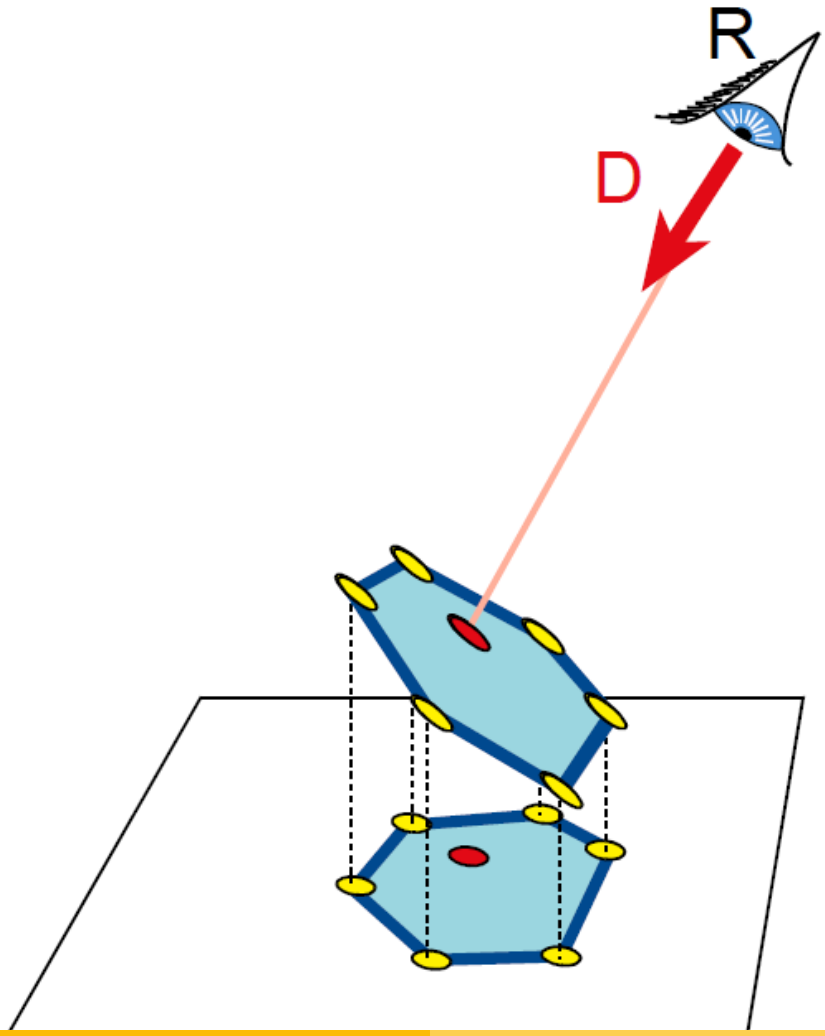
Sugár metszése poligonnal

- Egyszerű, de lassú módszer: osszuk fel háromszögekre a poligont.
- Visszavezettük az előző problémára (sugár metszése háromszögekkel).

Sugár metszése poligonnal

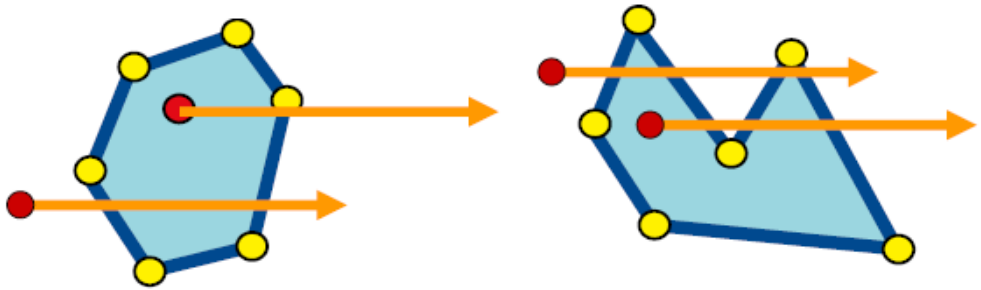
- Gyors módszer adható, ha a poligonunk csúcsai egy síkban vannak.
- Ekkor a metszés két lépésben végezhető:
  - A sugarunkat metszük el a poligon síkjával.
  - Döntsük el, hogy a metszéspont a poligonon belül van-e-
- A másodikat egy síkban érdemes csinálni (vagy a poligon síkjában, vagy a poligon valamely koordinátatengelyre vett vetületének síkjában)

Sugár metszése poligonnal



Pont-poligon tartalmazás teszt síkban

- A pont a poligonon belül van, ha tetszőleges irányú, belőle indított sugárnak páratlan számú metszéspontja van a poligon oldalaival (azaz a sugarat a poligon összes oldalszakaszával el kell metszeni)
- Konkáv és csillag alagú poligonra is működik



Sugár metszése szakasszal

- A poligon $\mathbf{d}_i = (x_i, y_i), \mathbf{d}_{i+1} = (x_{i+1}, y_{i+1})$ csúcspontjai közötti szakasz parametrikus alakja:
  $\mathbf{d}_{i,i+1}(s) = (1-s)\mathbf{d}_i + s\mathbf{d}_{i+1} = \mathbf{d}_i + s(\mathbf{d}_{i+1} - \mathbf{d}_i), s \in [0,1]$
- Ezt kell metszeni a $\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{v}$ alakú sugárral
- Most: a $\mathbf{p}_0 = (x_0, y_0)$ pont az a pont, amiről el akarjuk dönteni, hogy a poligonon belül van-e, $\mathbf{v}$ tetszőleges
- Legyen $\mathbf{v} = (1, 0)$!
- Így a $\mathbf{p}(t) = \mathbf{d}_{i,i+1}(s)$ egyenletet csak y koordinátára kell megoldani

Sugár metszése szakasszal

- Keressük meg, hogy hol metszi a $\mathbf{d}_{i,i+1}(s)$ oldal egyenese a sugarat (=melyik s-re lesz $d_{i,i+1}(s)_y = y_0$?)
- Azaz $y_0 = y_i + s(y_{i+1} - y_i)$
- s-t kifejezve: $s = \frac{y_0 - y_i}{y_{i+1} - y_i}$
- Innen megkapjuk azt az x koordinátát $\mathbf{d}_{i,i+1}(s)$-be behelyettesítve, ahol a sugár metszi a szakaszt (HF: kifejezni t-t DE: kell ez most nekünk?)
- Ha $s \notin [0, 1]$: a sugár nem metszi a szakaszt (csak az egyenesét)
- Ha $t \leq 0$: a sugár egybeesik a szakasszal, vagy mögötte van a metszéspont

A gömb egyenlete

- Az r sugarú, $\mathbf{c} = (c_x, c_y, c_z)$ középpontú gömb implicit egyenlete:

$$(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - r^2 = 0$$

- Ugyanez skalárszorzattal felírva:

$$\langle \mathbf{p} - \mathbf{c}, \mathbf{p} - \mathbf{c} \rangle - r^2 = 0,$$

ahol $\mathbf{p} = (x, y, z)$.

Gömb és egyenes metszéspontja

- Legyen $\mathbf{p}_0$ ez egyenes egy pontja, $\mathbf{v}$ az irányvektora.
- Ekkor az egyenes egyenlete:

$$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{v}$$

- Behelyettesítve a gömb egyenletébe, kapjuk:

$$\langle \mathbf{p}_0 + t\mathbf{v} - \mathbf{c}, \mathbf{p}_0 + t\mathbf{v} - \mathbf{c} \rangle - r^2 = 0$$

- Kifejtve:

$$t^2 \langle \mathbf{v}, \mathbf{v} \rangle + 2t \langle \mathbf{v}, \mathbf{p}_0 - \mathbf{c} \rangle + \langle \mathbf{p}_0 - \mathbf{c}, \mathbf{p}_0 - \mathbf{c} \rangle - r^2 = 0$$

$$t^2\langle \mathbf{v}, \mathbf{v}\rangle + 2t\langle \mathbf{v}, \mathbf{p}_0 - \mathbf{c}\rangle + \langle \mathbf{p}_0 - \mathbf{c}, \mathbf{p}_0 - \mathbf{c}\rangle - r^2 = 0$$

- Ez másodfokú egyenlet t-re (minden más ismert).
- Legyen $D = (2\langle \mathbf{v}, \mathbf{p}_0 - \mathbf{c}\rangle)^2 - 4\langle \mathbf{v}, \mathbf{v}\rangle(\langle \mathbf{p}_0 - \mathbf{c}, \mathbf{p}_0 - \mathbf{c}\rangle - r^2)$
- Ha $D > 0$: két megoldás van, az egyenes metszi a gömböt.
- Ha $D = 0$: egy megoldás van, az egyenes érinti a gömböt.
- Ha $D < 0$: nincs valós megoldás, az egyenes nem metszi a gömböt.

Transzformált objektumok

- Legyen **M** egy adott objektum transzformációs mátrixa.
- Feladat: Keressük **r** sugár és az **M**-mel transzformált objektum metszéspontját!
- Probléma: Hogyan transzformálunk egy gömböt? Pontonként? Képletet írjuk át? ...
- Megoldás: Transzformáljuk inkább a sugarat!

Tétel

Az r sugár és az $\mathbf{M}$-mel transzformált objektum metszéspontja $\equiv$ az $\mathbf{M}^{-1}$-zel transzformált $\mathbf{r}$ sugár és az objektum metszéspontja.

- $\mathbf{M} \in \mathbb{R}^{4\times 4}$, homogén transzformáció
- Sugár kezdőpontja: $\mathbf{p}_0 = (p_x, p_y, p_z) \rightarrow [p_x, p_y, p_z, 1]$
- Sugár iránya: $\mathbf{v} = (v_x, v_y, v_z) \rightarrow [v_x, v_y, v_z, 0]$. Így nem hat rá az eltolás.
- Transzformált sugár $\mathbf{r}'(t) : \mathbf{M}^{-1}\mathbf{p} + t \cdot \mathbf{M}^{-1}\mathbf{v}$

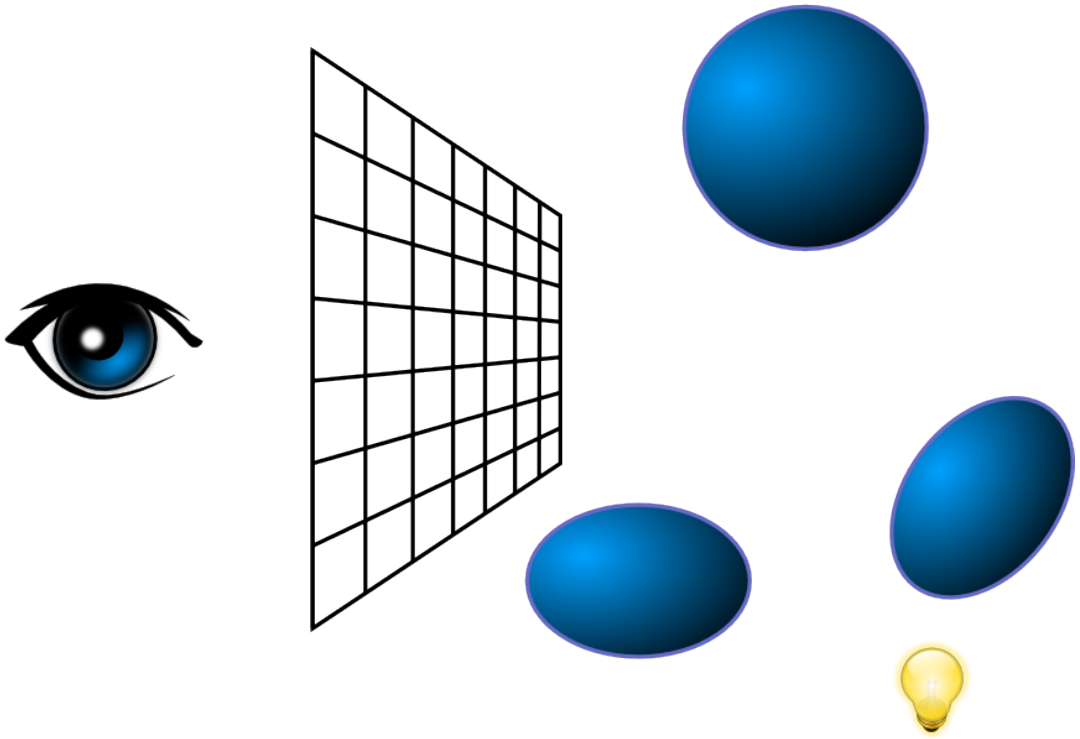
Metszésvizsgálat

- Metszésvizsgálat: használjuk $\mathbf{r}'(t)$-t!
- Metszéspont: $\mathbf{q}$, akkor az eredeti térben $\mathbf{M} \cdot \mathbf{q}$.
- Távolságokat újra kell számolni az eredeti térben!
- Normálvektorok: $\mathbf{n}$ helyett $\mathbf{M}^{-T} \cdot \mathbf{n}$ (inverz-transzponált).

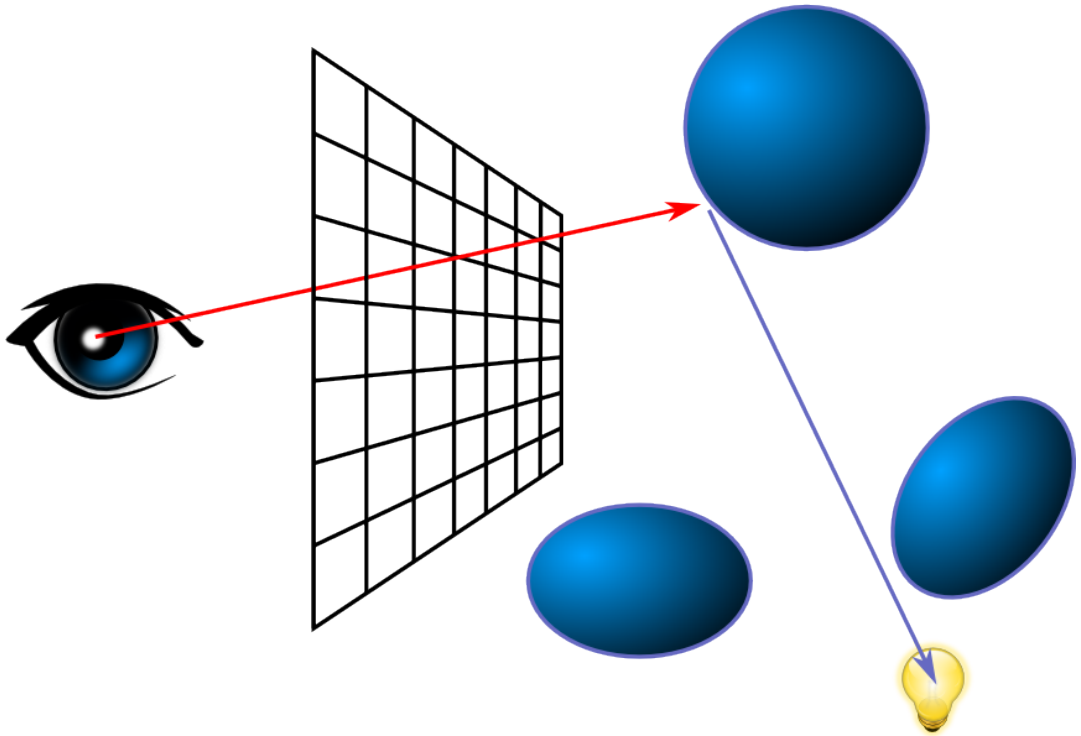
RayCasting versus RayTracing

- Raycasting : sugárvetés
- RayTracing : (rekurzív) sugárkövetés
- Sugárkövetés összetevői: felület pontjait meghatározza a saját színe megvilágítva a
  - fényforrásokkal és a
  - egyéb felületről érkező fénysugarakkal.
- ”egyéb felületeket” két kitüntetett irányban nézzük meg
  - tökéletes tükröződés iránya
  - fénytörés (áttetszés esetén)
- Rekurzivitás: felületekről érkező sugarakat tetszőleges mélységben vizsgáljuk
  - Rekurió mélységének a futási idő korlátot ad.

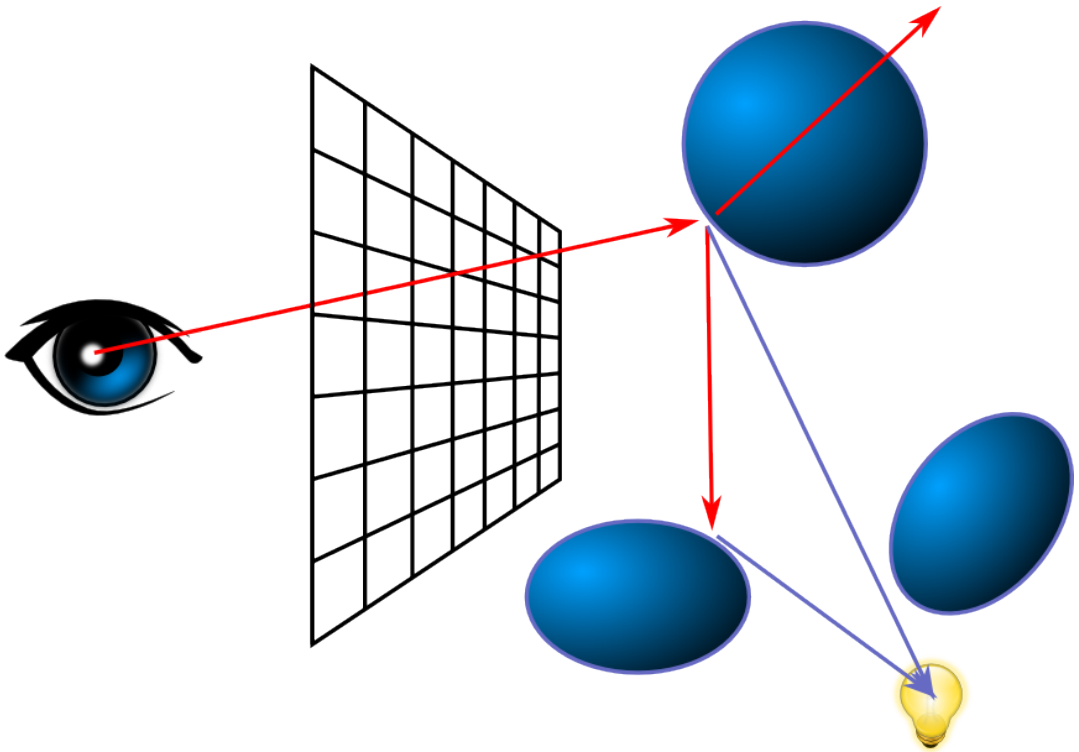
Rekurzív sugárkövetés



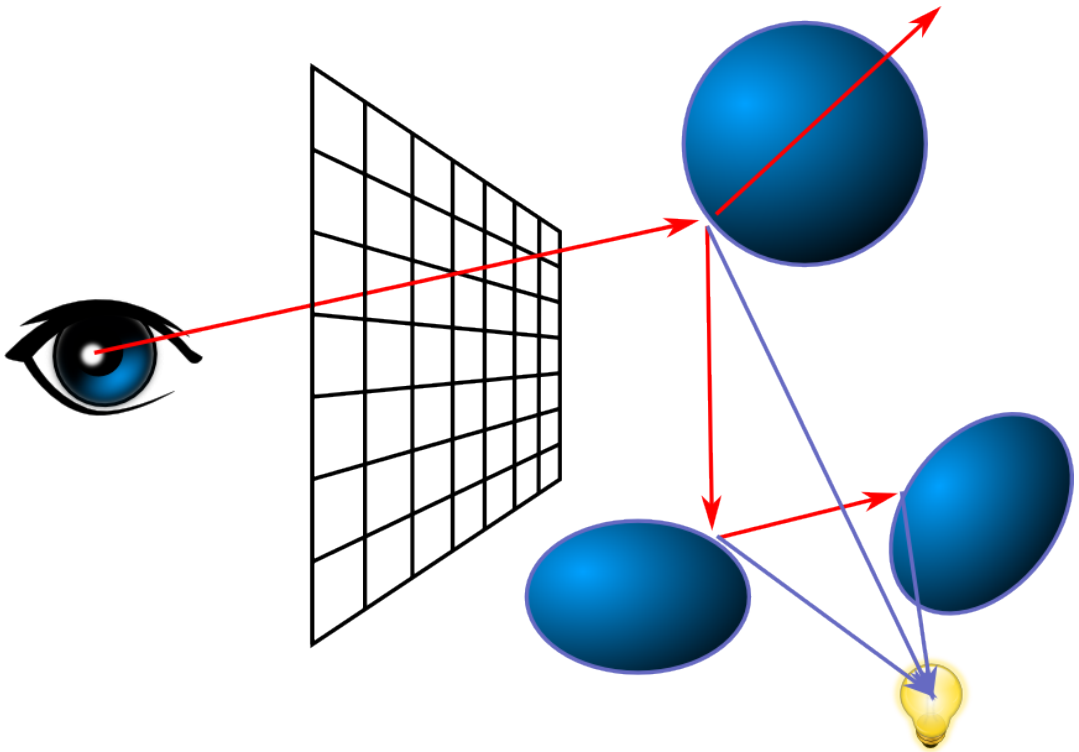
Rekurzív sugárkövetés



Rekurzív sugárkövetés



Rekurzív sugárkövetés



Számítógépes Grafika

Hajder Levente és Baráth Dániel
hajder@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2019/2020. I. félév

Tartalom

1. Tartalom
   - Motiváció
2. Grafikus szerelőszalag
   - Áttekintés
   - Modellezési transzformácó
   - Nézeti transzformácó
   - Perspektív transzformácó
   - Vágás
   - Raszterizáció
   - Megjelenítés
     - Triviális hátlapeldobás
     - Festő algoritmus
     - Z-buffer
3. Lokális illumináció

Tartalom

1 Tartalom
- Motiváció

2 Grafikus szerelőszalag
- Áttekintés
- Modellezési transzformácó
- Nézeti transzformácó
- Perspektív transzformácó
- Vágás
- Raszterizáció
- Megjelenítés
  - Triviális hátlapeldobás
  - Festő algoritmus
  - Z-buffer

3 Lokális illumináció

Grafikus szerelőszalag

- A színterünkről készített kép elkészítésének műveletsorozatát nevezik grafikus szerelőszalagnak (angolul *graphics pipeline*)
- A valósidejű alkalmazásoknak lényegében a színterünk leírását kell csak átadnia, a képszintézis lépéseit a grafikus szerelőszalag végzi
- A szerelőszalagban több koordináta-rendszer váltás is történik - mindent feladatot a hozzá legjobban illeszkedő rendszerben próbálunk elvégezni

Grafikus szerelőszalag

- Lépései főbb műveletek szerint:
  - Modellezési transzformáció
  - Nézeti transzformáció
  - Perspektív transzformáció
  - Vágás
  - Homogén osztás
  - Raszterizáció
  - Megjelenítés
- A grafikus szerelőszalag eredménye egy kép (egy kétdimenziós pixeltömb, aminek minden elemében egy színérték található)

Transzformációk

- A szerelőszalag transzformációinak feladata: a modelltérben adott objektumot "eljuttatni" a képernyő-térbe
- Lépései:

modell k.r.
$\rightarrow$ világ k.r.
$\rightarrow$ kamera k.r.
$\rightarrow$ normalizált eszköz k.r.
$\rightarrow$ képernyő k.r.

Szerelőszalag bemeneti adatai

- Ábrázolandó tárgyak *geometriai modellje*
- *Virtuális kamera* adatai (nézőpont és látógúla)
- A képkeret megadása (az a pixeltömb, amire a színterünk síkvetületét leképezzük)
- A színtérben található fényforrásokhoz és anyagokhoz tartozó *megvilágítási adatok*

Pipeline

- Minden egyes primitív végigmegy az összes lépésen
- Az egyes lépések az eredményüket a következőnek továbbítják
- Többféleképpen megadható és csoportosítható - mi csak egy példát írtunk fel (pl. hol "színezünk"?)

Koordináta-rendszerek

- Modell KR:
  Az objektumok saját koordináta-rendszerükben adottak.
- Világ KR:
  Az objektumok egymáshoz viszonyított helyzete itt adott, ahogy a kamera/szem pozicó és az absztrakt fényforrások pozicója is.
- Kamera KR:
  A koordináták a kamera pozicója és orientációjához relatíven adottak.

Koordináta-rendszerek

- Normalizált eszköz KR:
  A hardverre jellemző, $[-1, 1] \times [-1, 1] \times [0, 1]$ kiterjedésű KR.
- Képernyő KR:
  A megjelenítendő képnek (képernyő/ablak) megfelelő KR (balkezes, bal-felső "sarok" az origó).

Tartalom

1 Tartalom
- Motiváció

2 Grafikus szerelőszalag
- Áttekintés
- Modellezési transzformácó
- Nézeti transzformácó
- Perspektív transzformácó
- Vágás
- Raszterizáció
- Megjelenítés
  - Triviális hátlapeldobás
  - Festő algoritmus
  - Z-buffer

3 Lokális illumináció

Modellezési transzformácó

- A saját (modell) koordináta-rendszerben adott modelleket a világ-koordináta rendszerben helyezi el
- Tipikusan minden modellre különböző (lehet a színterünk két eleme csak a világtrafóban különbözik!)
- Jellemzően affin transzformációk
- Gyakorlatban: ez a *Model* (vagy *World*) mátrix a kódjainkban

Tartalom

1 Tartalom
- Motiváció

2 Grafikus szerelőszalag
- Áttekintés
- Modellezési transzformácó
- Nézeti transzformácó
- Perspektív transzformácó
- Vágás
- Raszterizáció
- Megjelenítés
  - Triviális hátlapeldobás
  - Festő algoritmus
  - Z-buffer

3 Lokális illumináció

Nézeti transzformáció

- A világ koordináta-rendszert a kamerához rögzített koordináta-rendszerbe viszi át.
- A transzformáció a kamera tulajdonságaiból adódik.
- Gyakorlatban: ez a *View* mátrix.

Kamera transzformácó

- Tulajdonságok, mint sugárkövetés esetén: **eye**, **center**, **up**
- Ebből kapjuk a nézeti koordináta-rendszer tengelyeit:

$$\mathbf{w} = \frac{\mathbf{eye} - \mathbf{center}}{|\mathbf{eye} - \mathbf{center}|}$$

$$\mathbf{u} = \frac{\mathbf{up} \times \mathbf{w}}{|\mathbf{up} \times \mathbf{w}|}$$

$$\mathbf{v} = \mathbf{w} \times \mathbf{u}$$

Kamera transzformácós mátrix

- Mátrixot kapjuk: áttérés az $-\mathbf{eye}$ origójú, $\mathbf{u}, \mathbf{v}, \mathbf{w}$ koordinátarendszerbe:

$$T_{View} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -eye_x \\ 0 & 1 & 0 & -eye_y \\ 0 & 0 & 1 & -eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Tartalom

1 Tartalom
- Motiváció

2 Grafikus szerelőszalag
- Áttekintés
- Modellezési transzformácó
- Nézeti transzformácó
- Perspektív transzformácó
- Vágás
- Raszterizáció
- Megjelenítés
  - Triviális hátlapeldobás
  - Festő algoritmus
  - Z-buffer

3 Lokális illumináció

Párhuzamos vetítés

- A mátrix ami megadja egyszerű, például az XY síkra való vetítés

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

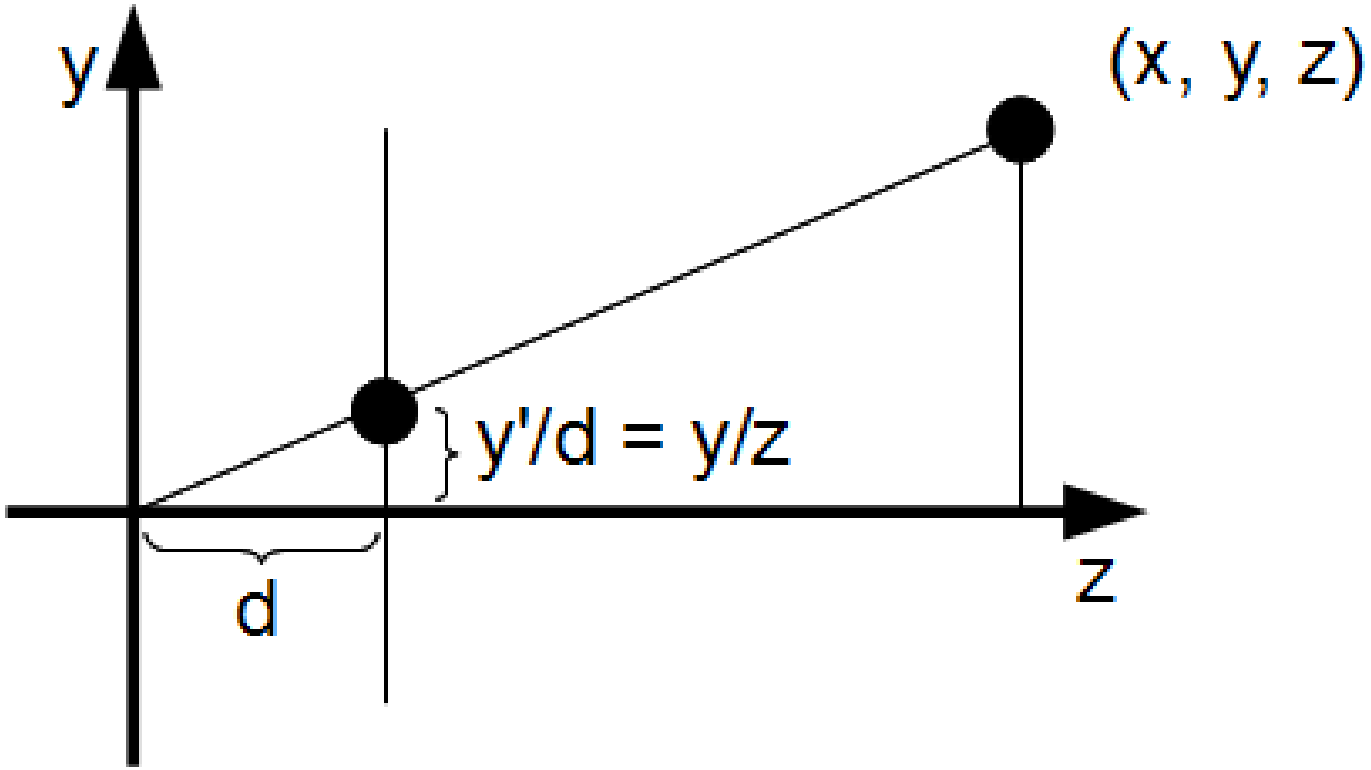
Perspektív transzformácó

- Emlékeztető: 3. EA
- A nézeti csonkagúla által határolt térrészt normalizált eszköz KR-be viszi át
- Ami benne volt a csonkagúlában, az lesz benne a $[-1,1] \times [-1,1] \times [0,1]$ (vagy $[-1,1] \times [-1,1] \times [-1,1]$) tartományban
- A transzformáció a kamerán átmenő *vetítő sugarakból* párhuzamosokat csinál
- A transzformáció a kamerapozíciót a végtelenbe tolja
- Gyakorlatban: ez a *Projection* mátrix

Perspektív transzformácó

- Emlékeztető: tulajdonságok
  - függőleges és vízszintes nyílásszög *(fovx, fovy)* vagy az alap oldalainak az aránya és a függőleges nyílásszög *(fovy, aspect)*,
  - a közeli vágósík távolsága *(near)*,
  - a távoli vágósík távolsága *(far)*

Középpontos vetítés



Középpontos vetítés

- Vagyis:

$$x' = \frac{x}{z}d$$
$$y' = \frac{y}{z}d$$
$$z' = \frac{z}{z}d = d$$

Középpontos vetítés

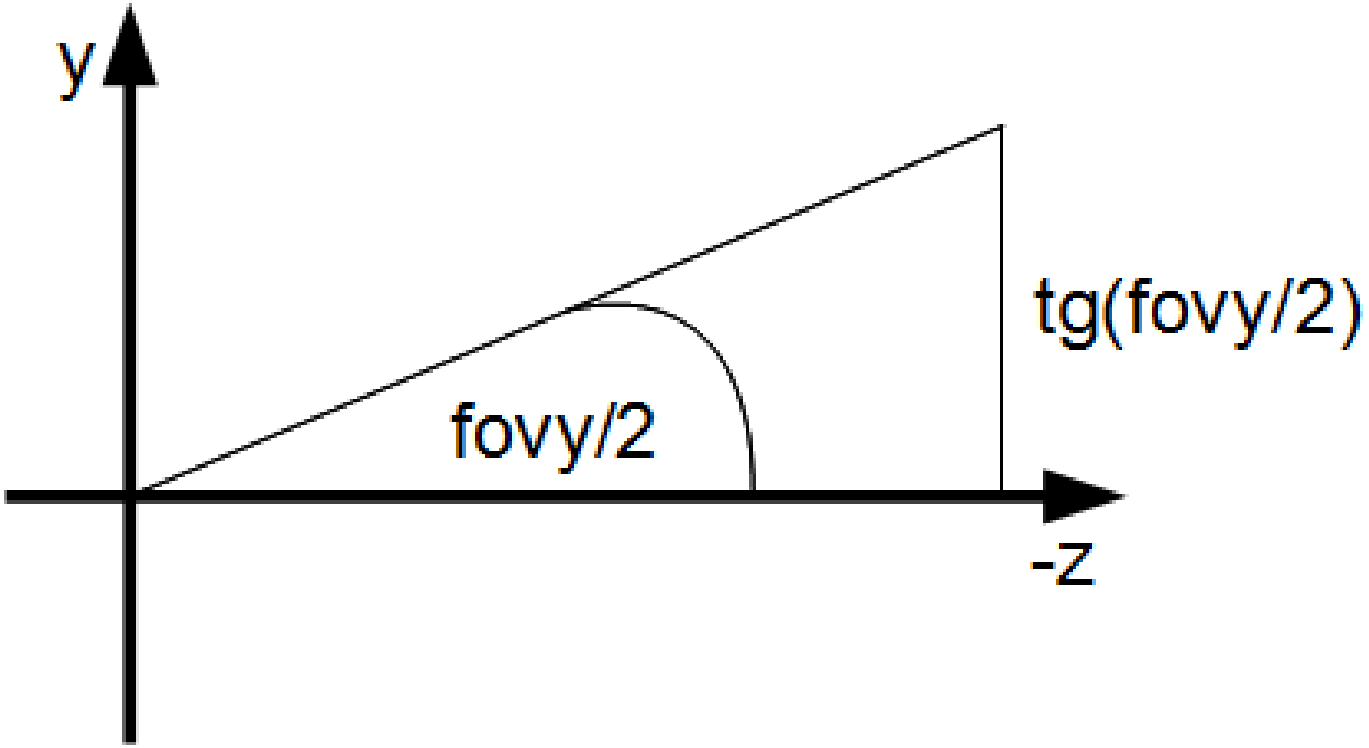
- Az origó, mint vetítési középpont és egy, attól a Z tengely mentén d egységre található, XY síkkal párhuzamos vetítősíkra való vetítés mátrixa:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix}$$

- Homogén osztás után ($\frac{z}{d}$-vel) a fentit kapjuk

Normalizált látógúla

- Figyeljünk: a szerelőszalagunk ezen pontján a kamera $-Z$ felé néz és az origóba van
- A fenti térből térjünk át egy "normalizáltabb" gúlába - aminek nyílásszöge x és y mentén is 90 fokos!



Normalizált látógúla

- Mátrix alakban:

$$\begin{bmatrix} 1/\tan\frac{fovx}{2} & 0 & 0 & 0 \\ 0 & 1/\tan\frac{fovy}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Normalizált látógúla

- Ezután már csak a közeli és a távoli vágosík z koordinátáit kell a normalizálásnak megfelelően átképezni ($-1, 1$ vagy $0, 1$-re):

$$T_{Projection} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{far}{far-near} & \frac{near*far}{near-far} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Tartalom

1 Tartalom
- Motiváció

2 Grafikus szerelőszalag
- Áttekintés
- Modellezési transzformácó
- Nézeti transzformácó
- Perspektív transzformácó
- Vágás
- Raszterizáció
- Megjelenítés
  - Triviális hátlapeldobás
  - Festő algoritmus
  - Z-buffer

3 Lokális illumináció

Vágás

- Cél: ne dolgozzunk feleslegesen azokkal az elemekkel, amikből nem lesz pixel (nem jelennek majd meg).
- Megoldás (kísérlet):
  1. Végezzük el a homogén osztást!
  2. Vágjunk le, ami kilóg a $[-1, 1] \times [-1, 1] \times [0, 1]$-ből!
- Probléma: vegyünk egy olyan szakaszt, ami átlóg a kamera mögé!
- Ez a szakasz átmegy egy ideális ponton $\Rightarrow$ a szakasz képe nem egyezik meg a transzformált pontokat összekötő szakasszal!
- Ez az *átfordulási probléma*.

Vágás homogén koordinátákban

- Megoldás (valóban): Vágás homogén koordinátákban
- Legyen: $[x_h, y_h, z_h, h] = \mathbf{M}_{\text{Proj}}[x_c, y_c, z_c, 1]$
- Cél: $(x, y, z) := (x_h/h, y_h/h, z_h/h) \in [-1, 1] \times [-1, 1] \times [0, 1]$, azaz

Legyen $h > 0$, és

$$-1 < x < 1$$
$$-1 < y < 1$$
$$0 < z < 1$$

Ebből kapjuk:

$$-h < x_h < h$$
$$-h < y_h < h$$
$$0 < z_h < h$$

Tartalom

1 Tartalom
- Motiváció

2 Grafikus szerelőszalag
- Áttekintés
- Modellezési transzformácó
- Nézeti transzformácó
- Perspektív transzformácó
- Vágás
- Raszterizáció
- Megjelenítés
  - Triviális hátlapeldobás
  - Festő algoritmus
  - Z-buffer

3 Lokális illumináció

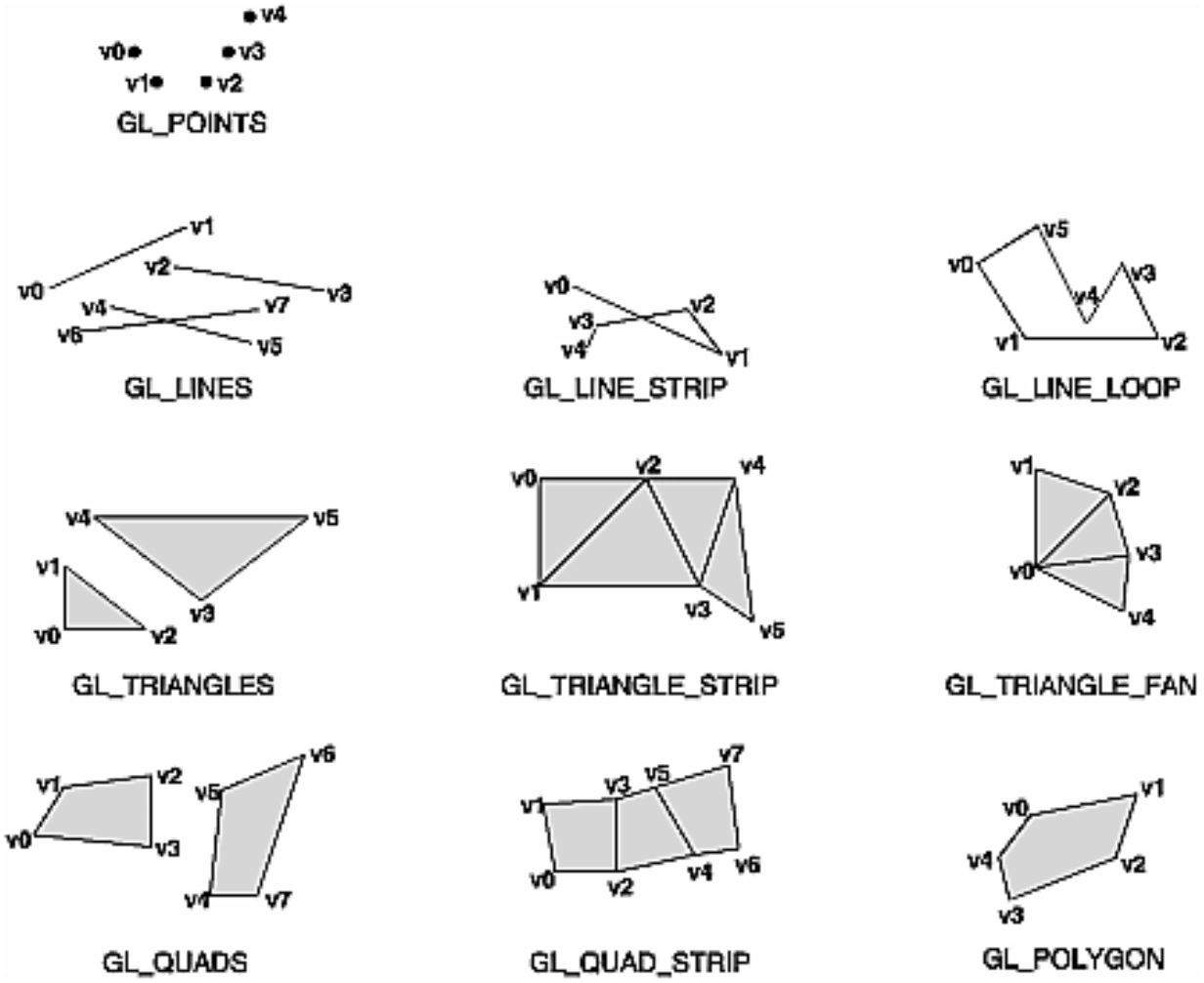
Raszterizáció

- Ne feledjük: eddig minden primitív, amiről beszéltünk folytonos objektum volt
- Azonban nekünk egy diszkrét térben, a képernyő képpontjain kell dolgoznunk
- A primitívek folytonos teréből át kell térni ebbe a diszkrét térbe, ezt hívják *raszterizációnak*

Raszterizáció

- Olyan geometriai primitíveket kell választanunk, amelyeket gyorsan tudunk raszterizálni
- Mi lehet ilyen? Jó lenne pl. ha egyik pixelhez tartozó felületi pontjának koordinátái alapján könnyen számítható lenne a szomszédos pixelekhez tartozó pontok koordinátái, illetve ha síkbeli is lenne...
- A háromszög ilyen!
- Minden egyéb felületet ilyen primitívekkel (lényegében: síklapokkal) közelítünk $\leftarrow$ *tesszeláció*

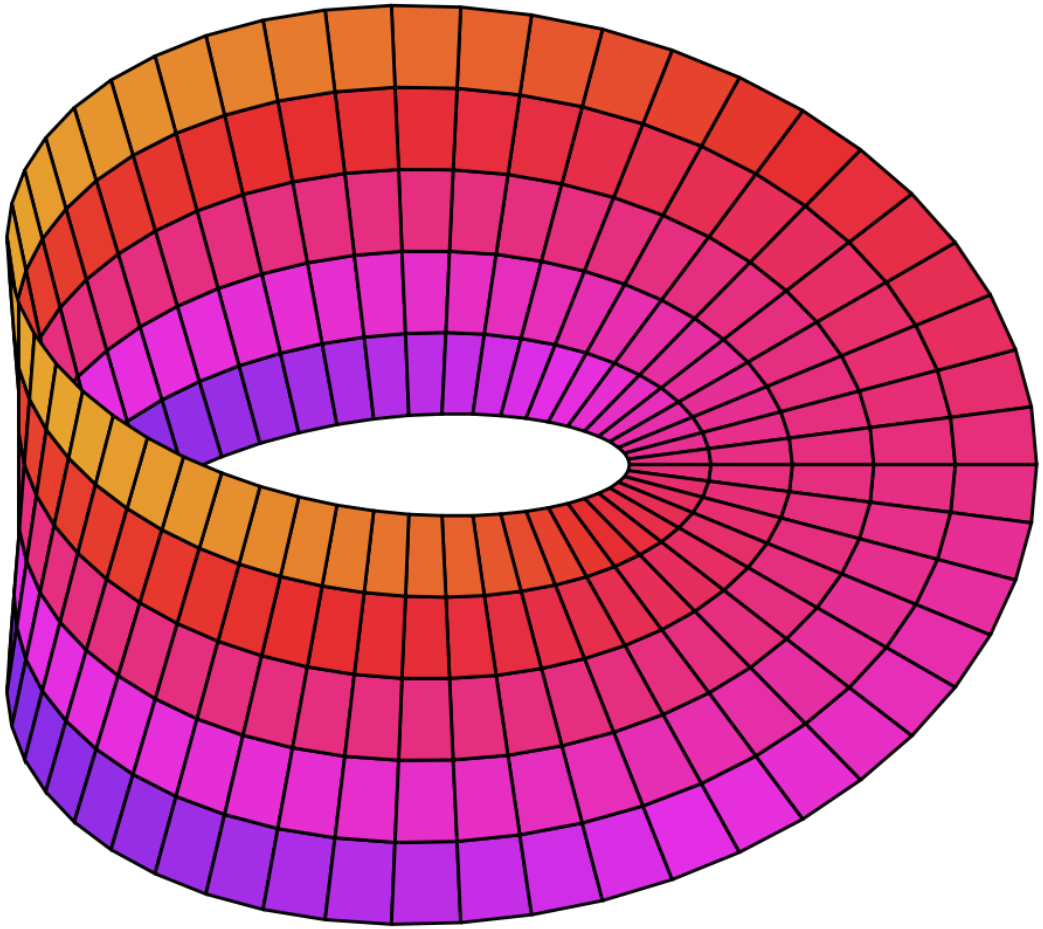
Raszterizáció



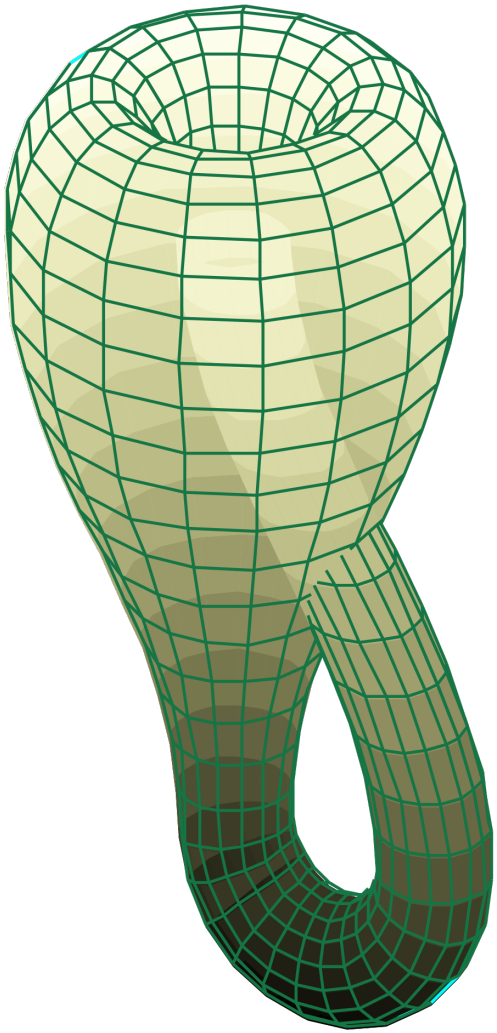
Tesszeláció

- Vigyázzunk: "szép" (teljes oldalakban illeszkedő), 6-reguláris háromszög vagy "szép", 4-reguláris négyszöghálóval nem lehet bármit lefedni degenerált esetek nélkül!
- A fenti reguláris topológiákkal a végtelen síklap, vagy a végtelen hengerpalást, vagy pedig a tórusz topológiájának megfelelő felületek írhatóak le.

Tesszeláció



Tesszeláció



Tartalom

1 Tartalom
- Motiváció

2 Grafikus szerelőszalag
- Áttekintés
- Modellezési transzformácó
- Nézeti transzformácó
- Perspektív transzformácó
- Vágás
- Raszterizáció
- Megjelenítés
  - Triviális hátlapeldobás
  - Festő algoritmus
  - Z-buffer

3 Lokális illumináció

Takarási feladat

- Feladat: eldönteni, hogy a kép egyes részein milyen felületdarab látszik.
- Objektum tér algoritmusok:
  - Logikai egységenként dolgozunk, nem függ a képernyő felbontásától.
  - Rossz hír: nem fog menni.
- Képtér algoritmusok:
  - Pixelenként döntjük el, hogy mi látszik.
  - Ilyen a sugárkövetés is.

Triviális hátlapeldobás, *Back-face culling*

- Feltételezés: Az objektumaink "zártak", azaz ha nem vagyunk benne az objektumban, akkor sehonnan sem láthatjuk a felületét belülről.
- Körüljárási irány: rögzítsük, hogy a poligonok csúcsait milyen sorrendben *kell* megadni:
  - óramutató járásával megegyező (*clockwise, CW*)
  - óramutató járásával ellentétes (*counter clockwise, CCW*)
- Ha a transzformációk után a csúcsok sorrendje nem egyezik meg a megadással, akkor a lapot *hátulról* látjuk $\Rightarrow$ nem kell kirajzolni, *eldobható*.

Festő algoritmus

- Rajzoljuk ki hátulró előre haladva a poligonokat!
- Ami közelebb van, azt később a rajzoljuk ⇒ ami takarva van, takarva lesz.
- Probléma: hogyan rakjuk sorrendbe a poligonokat?
- Már háromszögeknél is van olyan eset, amikor nem lehet sorrendet megadni.

Festő algoritmus

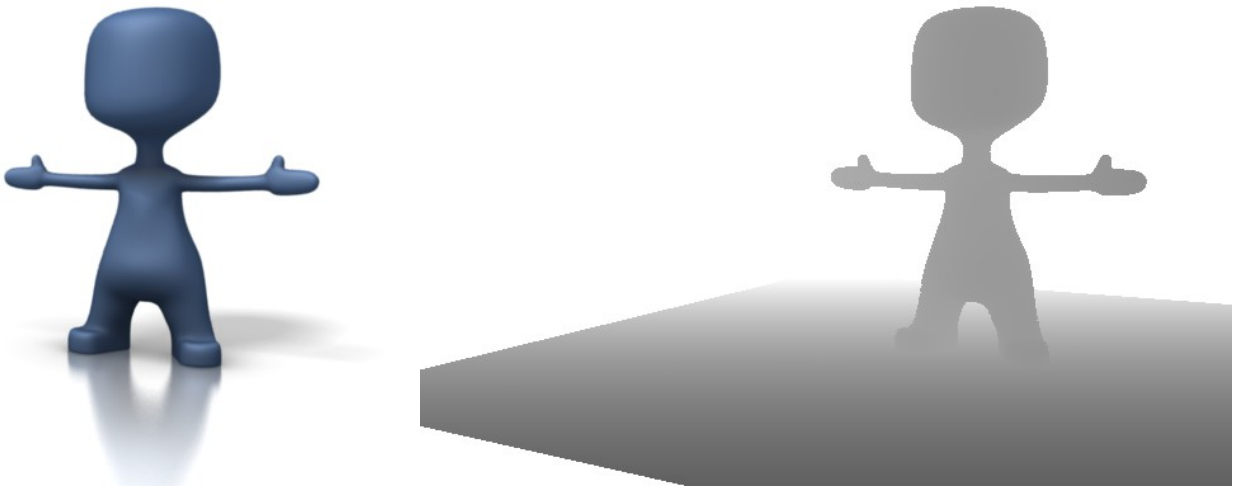
Z-buffer algoritmus

- Képtérbeli algoritmus
- Minden pixelre nyílvántartjuk, hogy ahhoz milyen mélységérték tartozott.
- Ha megint újra erre a pixelre rajzolnánk (*Z-test*):
  - Ha az új Z érték mélyebben van, akkor ez e pont takarva van $\Rightarrow$ nem rajzolunk
  - Ha a régi Z érték mélyebben van, akkor az új pont kitakarja azt $\Rightarrow$ rajzolunk, eltároljuk az új Z értéket.

Z-buffer

- Z-buffer vagy *depth buffer*: külön memóriaterület.
- Képernyő/ablak méretével megegyező méretű tömb.
- Pontosság: a közeli és távoli vágósík közti távolságtól függ.
- Minden pixelhez tartozik egy érték a bufferből.
- Ezzel kell összehasonlítani, és ide kell írni, ha a pixel *átment a Z-teszten*.
- Gyakorlatban:
  - 16-32 bites elemek
  - Hardveres gyorsítás
  - Pl: közeli vágósík: t, távoli: $1000t$, akkor a Z-buffer 98%-a a tartomány első 2%-át írja le.

Z-buffer



by macouno, macouno.com

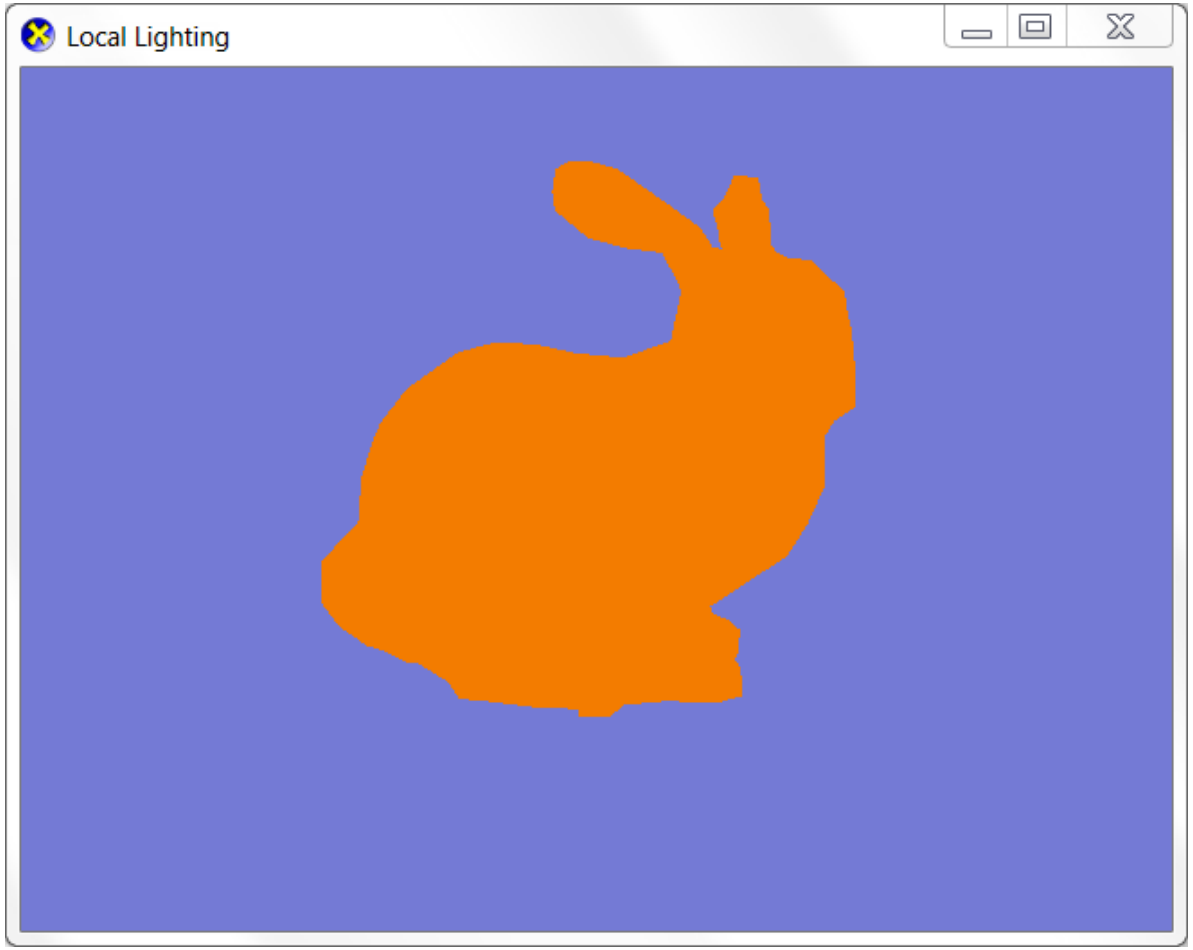
Lokális illumináció

- Ha már megvannak a primitívjeink pixelekre való leképezései, valahogyan számítsunk színeket

Saját színnel árnyalás

- Minden objektumhoz/primitívhez egy színt rendelenünk, és kirajzoláskor el lesz a pixelek értéke.
- Leggyorsabb: az ilumináció gyakorlatilag egyetlen értékadás.
- Borzasztó: se nem valósághű, se nem szép.

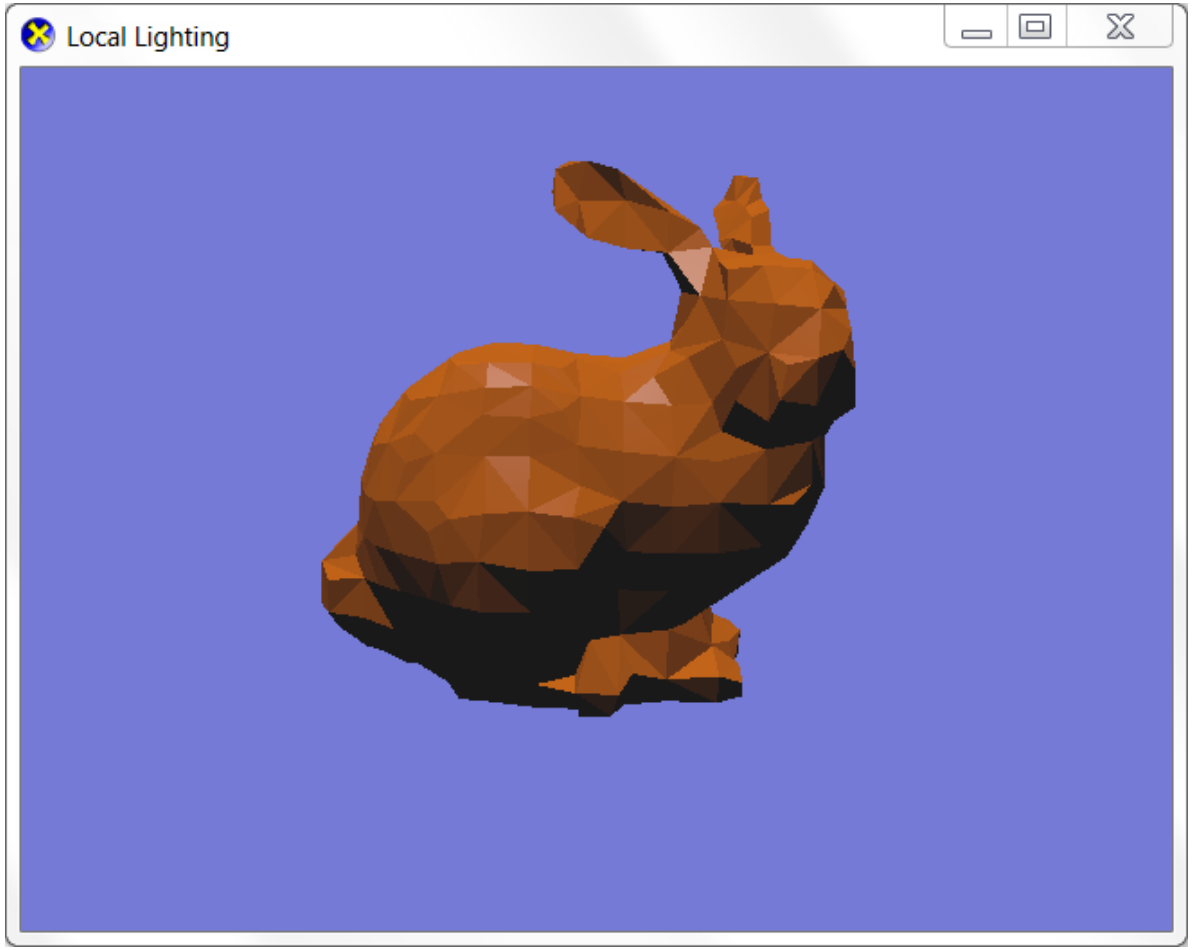
Saját színnel árnyalás



Konstans árnyalás, *Flat shading*

- A megvilágítást poligononként egyszer számítjuk ki, a szín homogén a lapon belül.
- Gyors: a műveletek száma a poligonok számától függ, a pixelek számától független.
- Van hogy használható: íves részeket nem tartalmazó, diffúz, egyszinű objektumokra

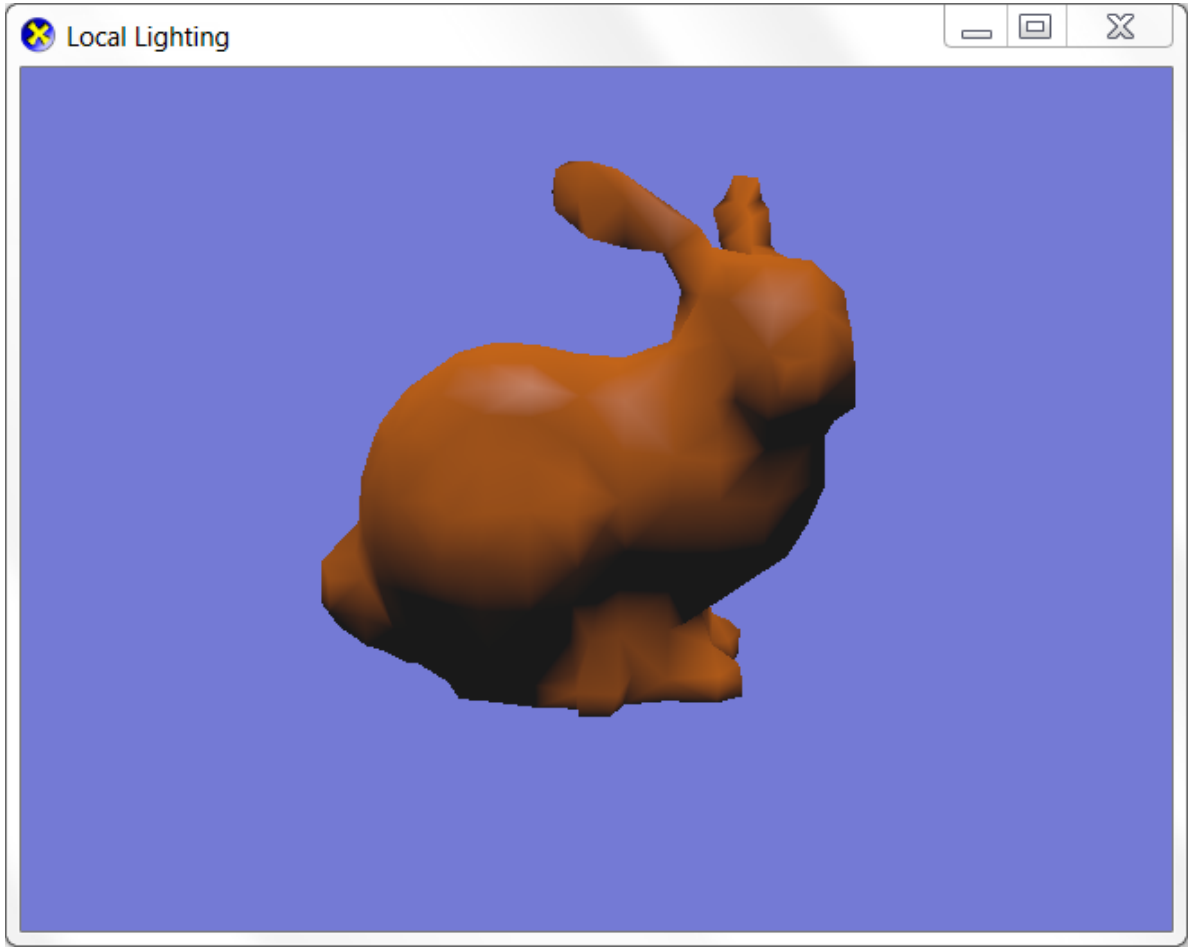
Konstans árnyalás



Gouraud árnyalás

- A megvilágítást csúcspontonként számítjuk ki, a lapon lineáris interpolációval számítjuk a színeket.
- Lassabb: N db megvilágítás számítás + minden pixelre interpoláció.
- Szebb: az árnyalás minősége nagyban függ a poligonok számától. Nagy lapokon nem tud megjelenni a csillanás.

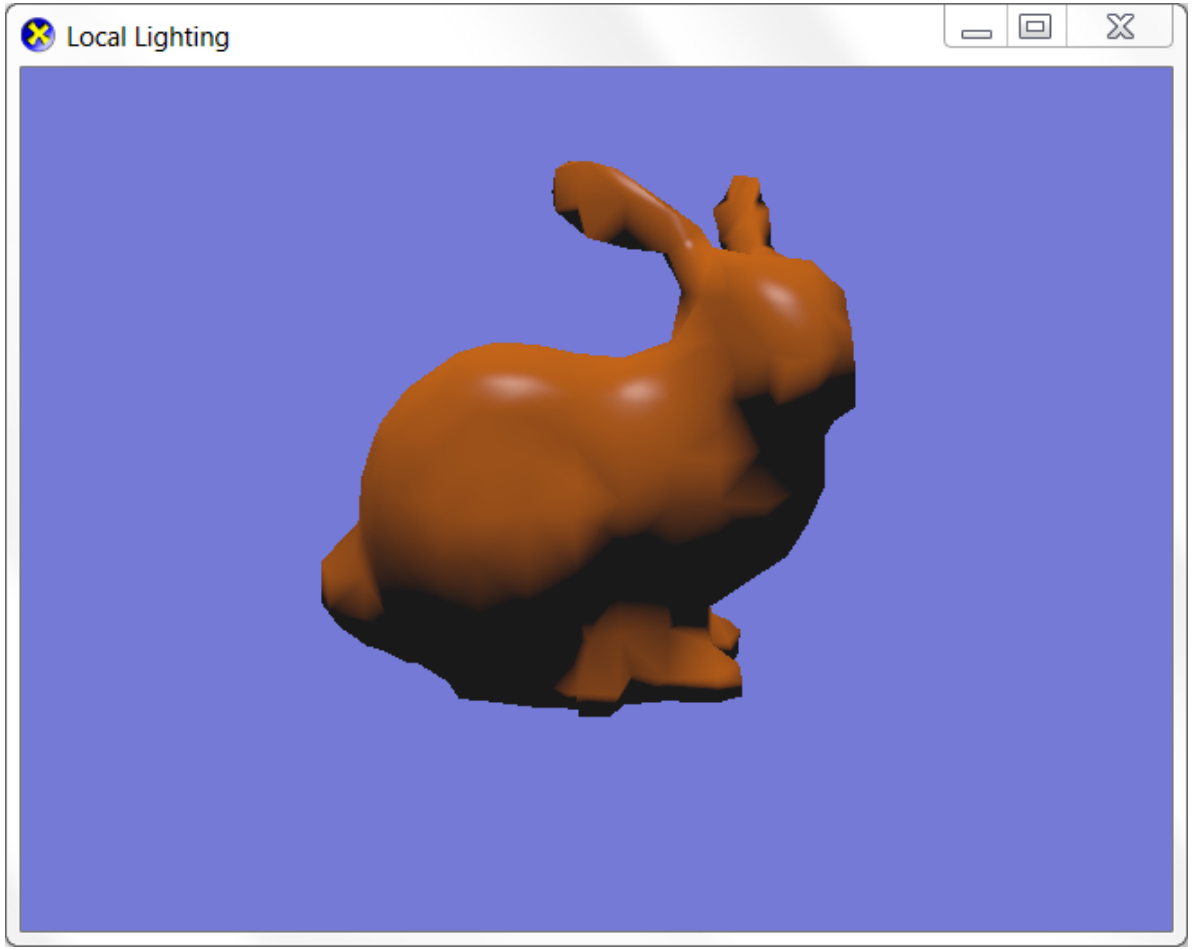
Gouraud árnyalás



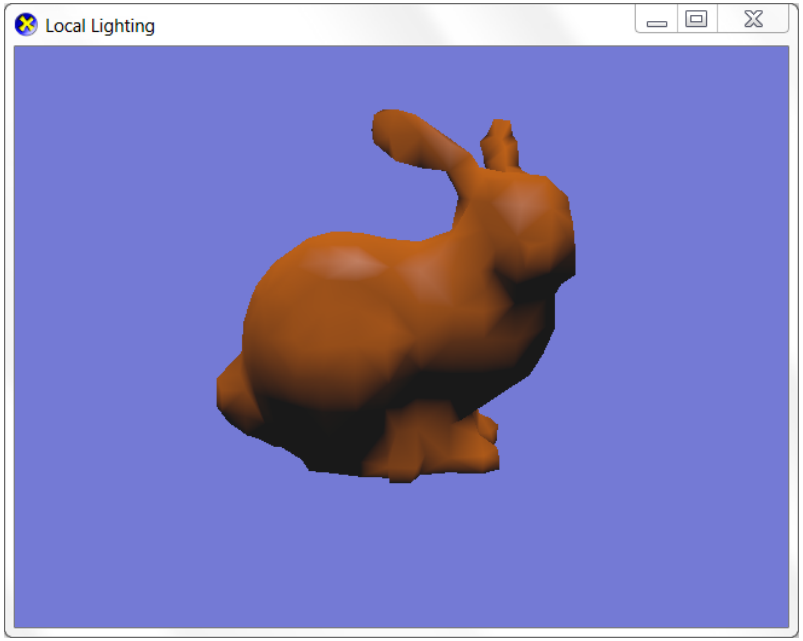
Phong árnyalás

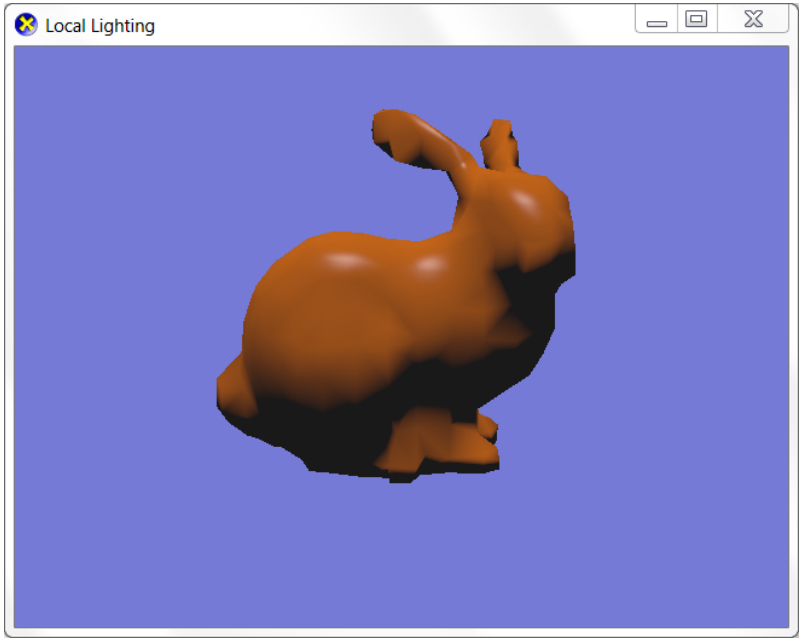
- Csak a normálvektorokat interpoláljuk, a megvilágítást minden pixelre kiszámítjuk.
- Leglassabb: *pixelek száma* db megvilágítás számítás.
- Legszebb: az árnyalás minősége nem függ a poligonok számától. Csillanás akár poligon közepén is meg tud jelenni.

Phong árnyalás



Gouraud vs Phong árnyalás





Számítógépes Grafika

Hajder Levente és Baráth Dániel
hajder@inf.elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

2019/2020. I . félév

Tartalom

1. Animáció
2. Animálható tulajdonságok
3. Animáció típusok
4. Hierachikus rendszerek

Animáció

- Állókép helyett képsorozat
- Objektumok/kamera/világ tulajdonságait változatjuk
- Egy kép $\equiv$ egy idő pillanat
- Képsorozat elég gyors $\rightarrow$ folyamatos mozgást érzékel az emberi szem

Mi az, amit változtatunk?

- Lehet: pozícó, orientáció, szín, normál, BRDF, stb.
- ”Értelme” van: pozíció, orientáció, kamera
- Animációt alkalmazunk:
  - a modellezési transzformációra
  - a kamera transzformációra

Animáció szintézis

- Legyen minden o objetumra, a modellezési trafó: $M_o \in \mathbb{R}^{4\times4}$
- Legyen a kamera trafó: $V \in \mathbb{R}^{4\times4}$
- *Megjegyzés: ha V a View mátrix, akkor csak a kamera pozicióját, orientációját tartalmazza; ha V a View és a Projection együtt, akkor tudjuk vele a látószöget is állítani*
- Legyen mindkettő idő függő!
- $M_o \in \mathbb{R} \to \mathbb{R}^{4\times4}$
- $V \in \mathbb{R} \to \mathbb{R}^{4\times4}$

Általános animációs program váza

```
while keep_running:
  t = get_time()
  for o in objects:
    M_o = M_o(t)
  V = V(t)
  render_scene()
```

Animáció szintézis

- Valósidejű/interaktív:
  - rögtön meg is jelenítjük a képkockákat
  - a számításnak elég gyorsnak kell lennie a folytonosság látszatához
  - a felhasználói eseményekre reagálni kell
  - $\Rightarrow$ olyan részletgazdag lehet a színtér, amit még így meg lehet jeleníteni
  - $\Rightarrow$ inkrementális képszintézist használunk

Valósidejű animációs program váza

```
def update():
  # vagy idle, vagy onFrameMove
  t = get_time()
  for o in objects:
    M_o = M_o(t)
  V = V(t)

def render():
  # vagy display, vagy onFrameRender
  for o in objects:
    render_object(o)
```

Animáció szintézis

- Nem valós idejű/*offline*:
  - ”Nem számít”, hogy mennyi ideig tart kiszámítani egy képkockát
  - Elkülönöl a szintézis és a visszajátszás
    - Először elmentjük a képkockákat
    - Aztán majd videóként lehet visszanézni
  - $\Rightarrow$ a felhasználó nem tud belenyúlni az animációba
  - $\Rightarrow$ olyan részletes színteret, és olyan bonyolult algoritmust használunk, amit ki tudunk várni

Offline rendering

```
Δt = 1/FPS
for (t=t_start; t<t_end; t+=Δt):
  for o in objects:
    M_o = M_o(t)
  V = V(t)
  render_scene_to_disk()
```

Valósidejű animáció – rosszul

- Hogyan lehet a legkönnyebben elrontani az animáció számítást?
- Azzal, ha nem vesszük figyelembe, hogy mennyi idő telt el két képkocka között.
- Pl.: A középpontja körül akarjuk forgatni az objektumot állandó szögsebességgel.

```
model = rotation(phi, 0,1,0); phi +=
phi_step;
```

- Az objektum olyan gyorsan fog forogni, amilyen gyakorisággal ez a kód részlet meghívódik.
- Gyorsabb gépen többször, lassabb gépen kevesebbszer.

Valósidejű animáció – jól

- Hogyal lehet a legkönnyebben ezt megelőzni?
- Sose azt tároljuk, hogy mennyivel kell változtatni, hanem, hogy mi a változás *sebessége*.
- Minden számítás előtt kérjük le, hogy mennyi idő telt el az előző képkocka óta, és ezel szorozzuk a sebességet.
- Pl.: `phi += phi_step;` helyett `phi += get_time_since_last_frame() * phi_speed;`
- Gyorsabb gépen kevesebb, lassabb gépen több idő telik el két képkocka között ⇒ gyors gépen kissebbeket lépünk, lassabban nagyobbakat.

Tartalom

1 Animáció

2 Animálható tulajdonságok
- Kamera animáció
- Pozíció és orientáció
- Roll, Pitch, Yaw

3 Animáció típusok

4 Hierachikus rendszerek

Kamera animáció

- A kamera tulajdonságai:
  - szempozíció (*eye*),
  - egy pont amire néz (*center*),
  - felfele irányt megadó vektor (*up*),
  - a képernyő/ablak oldal aránya (*aspect*),
  - nyílásszög (*fovy*).
- Ezek mind külön-külön változtathatók az animáció létrehozásához.

Pozíció és orientáció

- Pozícó: *"Hol van az objektum?"*
- Orientáció: *"Hogy áll, merrefele néz az objektum?"*
- Elsősorban ezt a kettőt szerenénk változtatni.
- $M_o(t)$ megadja mindkettőt.
- Normális esetben

$$M_o(t) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 \\ A_{21} & A_{22} & A_{23} & 0 \\ A_{31} & A_{32} & A_{33} & 0 \\ p_x & p_y & p_z & 1 \end{bmatrix},$$

- $\vec{p} = (p_x, p_y, p_z)$ a pozíció.
- Az **A** mátrix tartalmazza az orientációt.

Orientációs paraméterek

- Tegyük $\vec{p}$-t és **A**-t is időfüggővé!
- $\vec{p}$ tagjait leírhatjuk külün-külön függvénnyel.
- *Pl.: Valami esik, elég p_y-t változtatni.*
- **A** tagjai összefüggenek, csak az orentáció érdekel belőle minket (nem érdekel: méretezés, nyírás).
- Az orientáció megadható három tengely menti forgatással $\rightarrow$ három független függvénnyel.

Yaw, pitch, roll

- Egy objektum függőleges- (*yaw*), kereszt- (*pitch*) és hossztengelye (*roll*) menti elfordulásait egyszerre adjuk meg.
- Már találkoztunk vele, 3×3-as mátrixszal megadható, három forgatás szorzata.
- Három tengely menti elfordulás szögével megadható $\Rightarrow$ megadja az orientációt is.

Tartalom

1. Animáció
2. Animálható tulajdonságok
3. Animáció típusok
   - Képlet animáció
   - Kulcskocka animáció
   - Pálya animáció
4. Hierachikus rendszerek

Képlet animáció

- Egy adott tulajdonság változását egy megfelelő függvénnyel írjuk le.
- Pl: Óra mutatói
  - Nagymutató: $yaw(t) = t/10$
  - Kismutató: $yaw(t) = t/120$
  - (ha t-t mp-ben adjuk meg, a forgatásokat pedig fokban)
- Pl: Pattogó labda
  - $p_y(t) = |\sin(\omega t + \theta_0)| \cdot e^{-kt}$

Kulcskocka (*key frame*) animáció

- Egy bonyolult mozgást nehézkes lenne képlettel megadni.
- Inkább adjuk csak bizonyos időközönként, hogy *akkor* mit szeretnénk látni.
- Ezek a kulcskockák.
- Egy tulajdonságot két kulcskocka között *interpolációval* számolunk ki.

Kulcskocka (*key frame*) animáció

- Az interpolációval az objektum egyes paramétereire folytonos görbét illesztünk.
- Az animáció lejátszása/elmentése során a program minden képkockában a hozzá tartozó t értékkel kiértékeli az objektum paraméter-függvényeit.
- Ezekből számítja a transzformációs mátrixokat.
- A mátrixok felhasználásval előállítja a képet.

Lineáris interpoláció

- Legyen a két kulcskockánk időpontja t_0 és t_1.
- Legyen az interpolálandó tulajdonság g.
- Lineráis interpolációval $\forall t \in [t_0, t_1]$-re kapjuk

$$g(t) = \left(1 - \frac{t - t_0}{t_1 - t_0}\right) g(t_0) + \frac{t - t_0}{t_1 - t_0} g(t_1)$$

Interpoláció két kulcskocka között

K: Mi a baj a lineáris interpolációval?

V: Ritkán néz ki természetesen. Animáció során a sebesség konstans, előtte, utána nulla.

- Elgurított labda: folyamatosan lassul.
- Zuhanó zongora: folyamatosan gyorsul.
- Rakéta a Földről a Marsra: gyorul, halad, lassít.

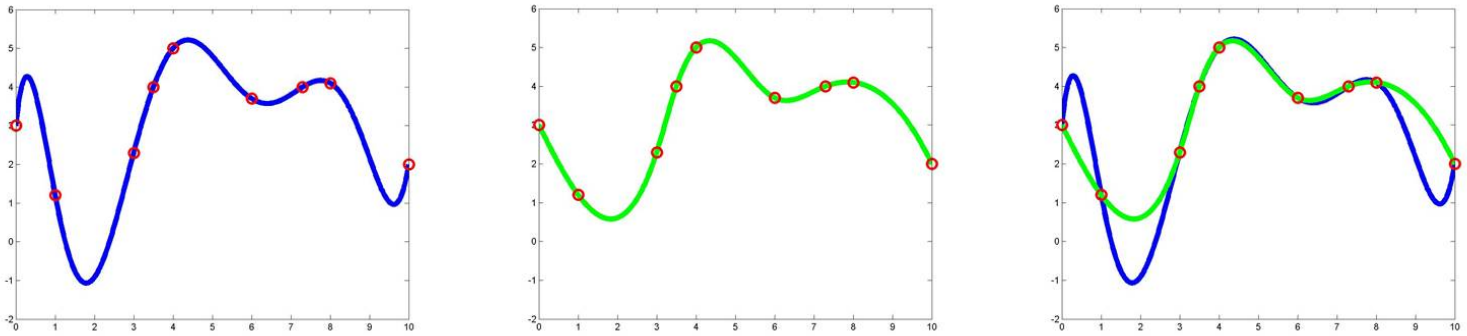
- Ilyen interpolációra használhatunk:
  - Gyök függvényt.
  - Másodfokú függvényt.
  - Logisztikus függvényt/görbét.
- Gyakorlatban: Bézier görbe

Polinom interpoláció

- n kulcspontra fel tudunk írni $n-1$-ed fokú polinomot.
- Interpolációs polinom: minden kulcskockában az előírt értéket veszi fel.
- Együtthatók számíthatók Lagrange intelpolációval.
- A lineáris interpoláció a Lagrange intelpoláció speciális esete $n = 2$-re.

Spline interpoláció

- A polinom interpolációval kapott fv. magas fokszám esetén a szomszédos pontok között ”hullámzik”, így elrontja az animációt.
- Spline interpoláció: használjunk több, egymáshoz kapcsolódó, alacsony fokszámú polinomot az interpolációhoz!



Pálya animáció

- Egy objektum mozgását megadhatjuk a bejárandó pálya megadásával is.
- A pályát egy 3D görbével adjuk meg.
- A model ezen a görbén halad végig.

Orientáció megadása

- Hogyan adjuk meg az objetumunk orentációját?
- Egy *előre*, és egy *felfele* irány egyértelműen meghatározza ezt.
- *Megjegyzés: v.ö. kamera esetén center-eye ill. up vektorok*
- Ha a pályagörbe differenciálható, akkor az megadja a sebességvektort minden időpillanatban.
- A sebességvektor mindig előre fele mutat.

Orientáció megadása

- A *felfele* irány megadására két lehetőségünk is van.
- Ha van egy természetes *felfele*, akkor használjuk azt. (Mindennél, ami nem ből be a kanyarban.)
- Ha ez az irány is változik, akkor ez megegyezik a gyorsulás irányával, azaz a pályagörbe második deriváltjának irányával.

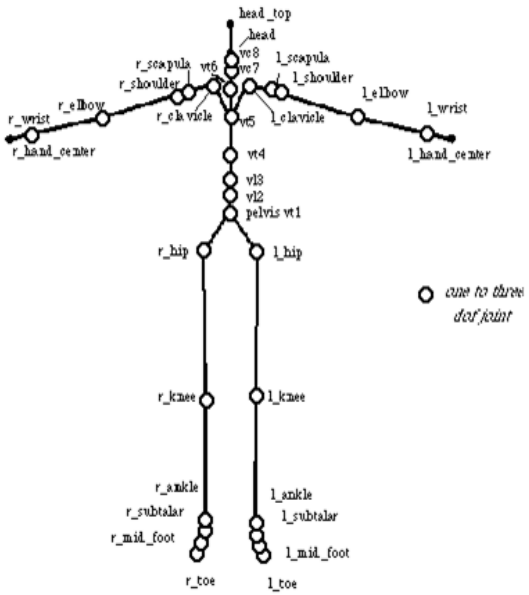
Tartalom

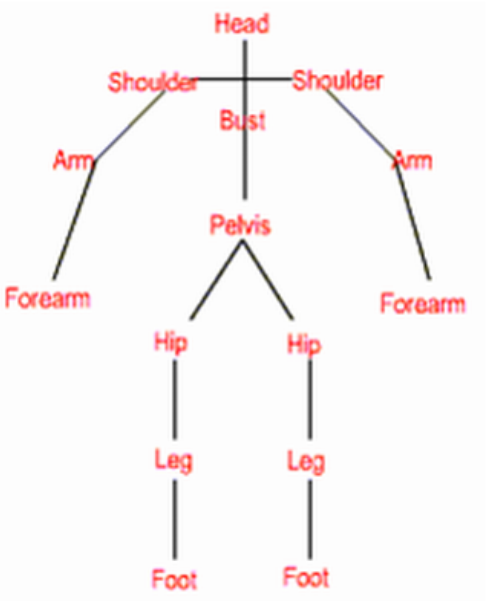
1. Animáció
2. Animálható tulajdonságok
3. Animáció típusok
4. Hierachikus rendszerek
   - Előrehaladó kinematika
   - Inverz kinematika

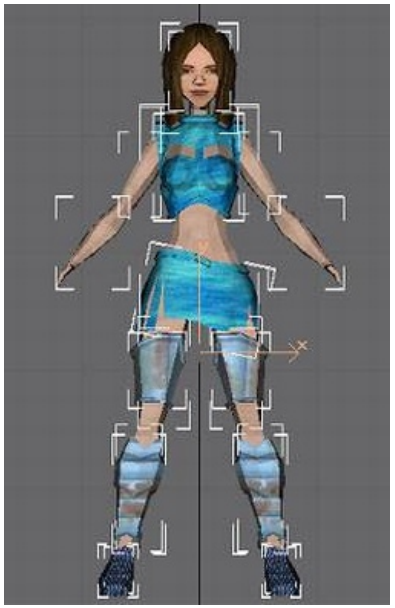
Hierachikus rendszerek

- Egy gyerek objektum mozgását a szülőhöz viszonyítva adjuk meg.
- Gyerekeknek lehetnek újabb gyerekei, s.i.t.
- Hierachikus rendszert – fát – kapunk.

Példa: Emberi test







Kényszerek (*constraints*)

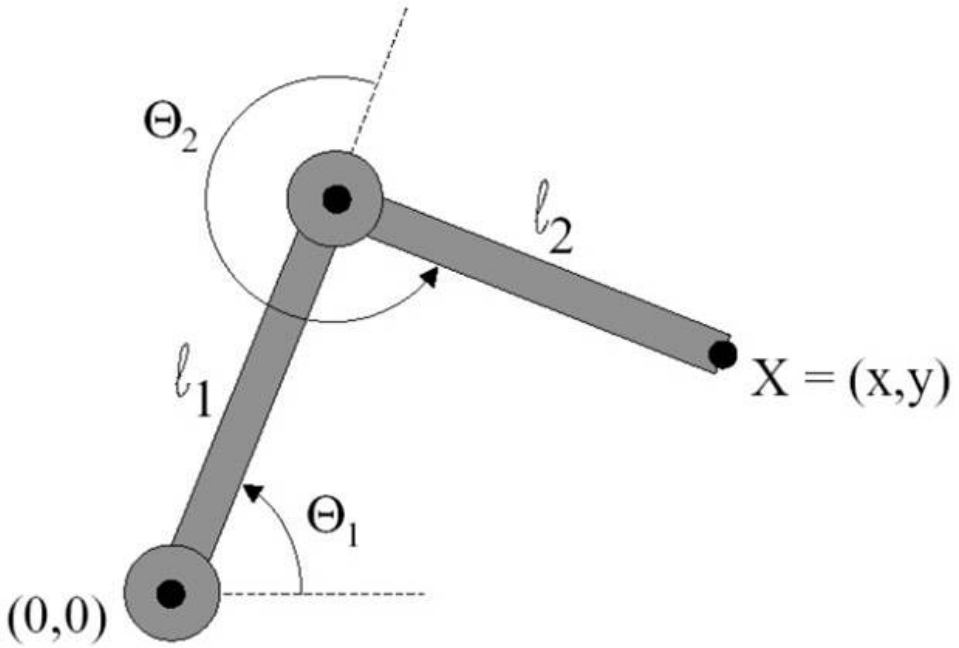
- Nem minden mozgást szeretnénk megengedni a szülőhöz képest.
- Ezeket a megszorításokat írhatjuk le *kényszerekkel*.
- Korlátozhatjuk a szabadságfokokat: pl. könyök csak egy tengely mentén tud forogni, de a csukló kettő
- Vagy a tartományokat: kevesen bírják, ha a fejük 90°-nél többet fordul.

Előrehaladó kinematika

- Végállapotot határozunk meg az állapotváltozók függvényében.
- Szimulációkhoz jól használható.
- Minden elemre megadjuk, a hozzá tartozó transzformációt.
- Ezeket a hierarchiában felülről lefele haladva értékeljük ki.
- Az adott elemhez tartozó transzformáció az összes ős és a saját transzformáció szorzata.

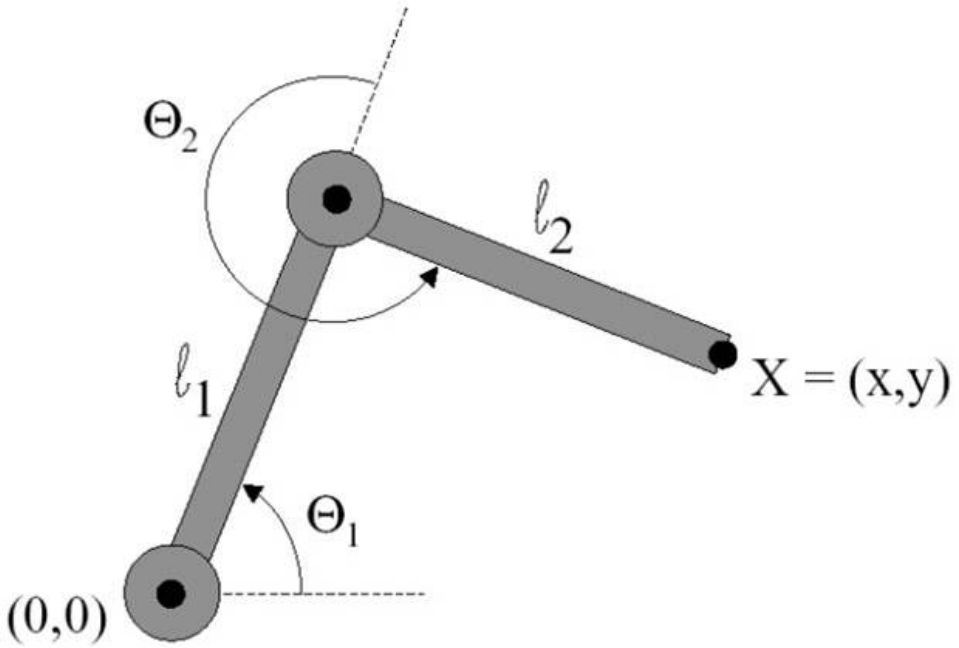
Példa

- Kétszabadságfokú, rotációs csuklókat tartalmazó rendszer.
- A csuklók csak a Z tengely körül fordulnak.



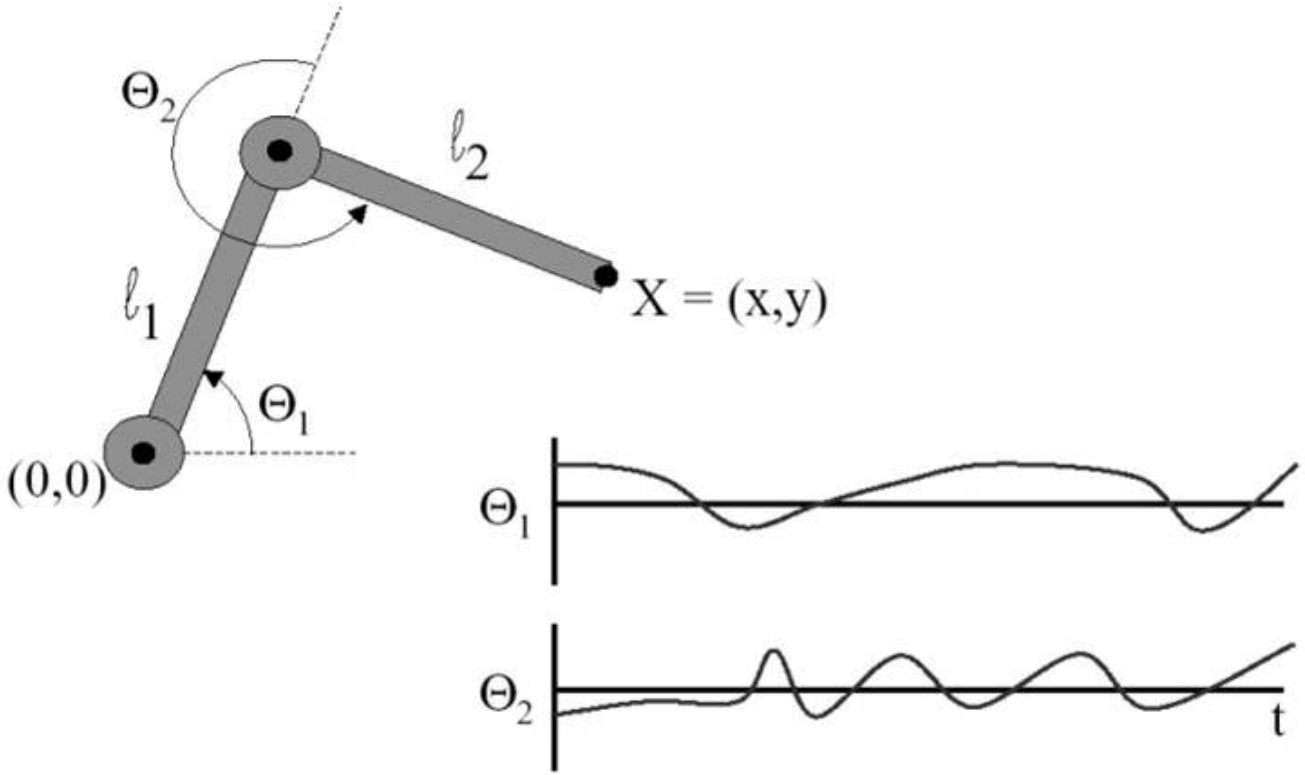
Példa – folyt.

- Állapotváltozók: Θ_1, Θ_2
- A végberendezés (X) pozícióját a gép számolja.
- $X = (l_1 \cos\Theta_1 + l_2 \cos(\Theta_1 + \Theta_2), l_1 \sin\Theta_1 + l_2 \sin(\Theta_1 + \Theta_2))$



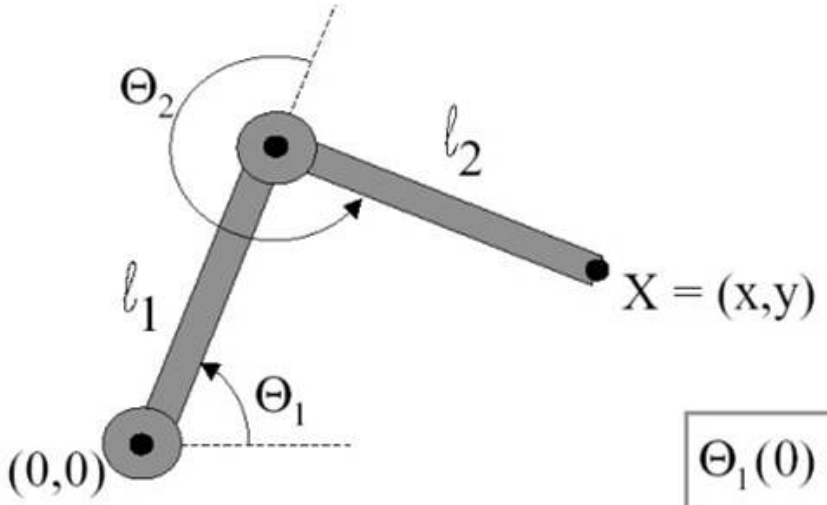
Példa – folyt.

- Az állapotváltozókat megadhatjuk (pl. spline) függvénnyel.



Példa – folyt.

- Az állapotváltozókat megadhatjuk kezdeti értékkel és sebességgel.



$$\Theta_1(0) = 60^\circ \qquad \Theta_2(0) = 250^\circ$$

$$\frac{d\Theta_1}{dt} = 1.2 \qquad \frac{d\Theta_2}{dt} = -0.1$$

Mit nem tud az előrehaladó kinematika?

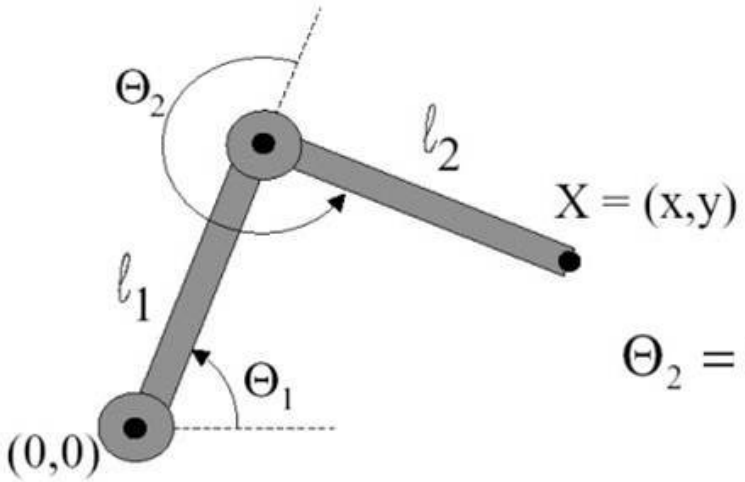
- Az előrehaladó kinematika nem használható, ha a strukturális összefüggés erősen nemlineáris
- Hiába interpolálunk egyenletesen az állapottérben, a végberendezés vadul kalimpálhat a kulcspontok között
- Probélmás esetek:
  - Láb mozgása a talajon
  - Végállapot jó, de menet közben a berendezés részei átmehetnek egymáson.

Inverz kinematika

- Az inverz kinematika a kritikus végberendezés helyzetét interpolálja, majd az állapotváltozók értékét végberendezés interpolált helyzetéből számítja vissza.
- Az inverz kinematika másik neve a cél-orientált animáció.
- *”Ezt szeretném megfogni, hogyan forgassam az izületeimet?”*

Példa

- A végberendezés helyzetéből visszaszámoljuk az állapotváltozók értékét.



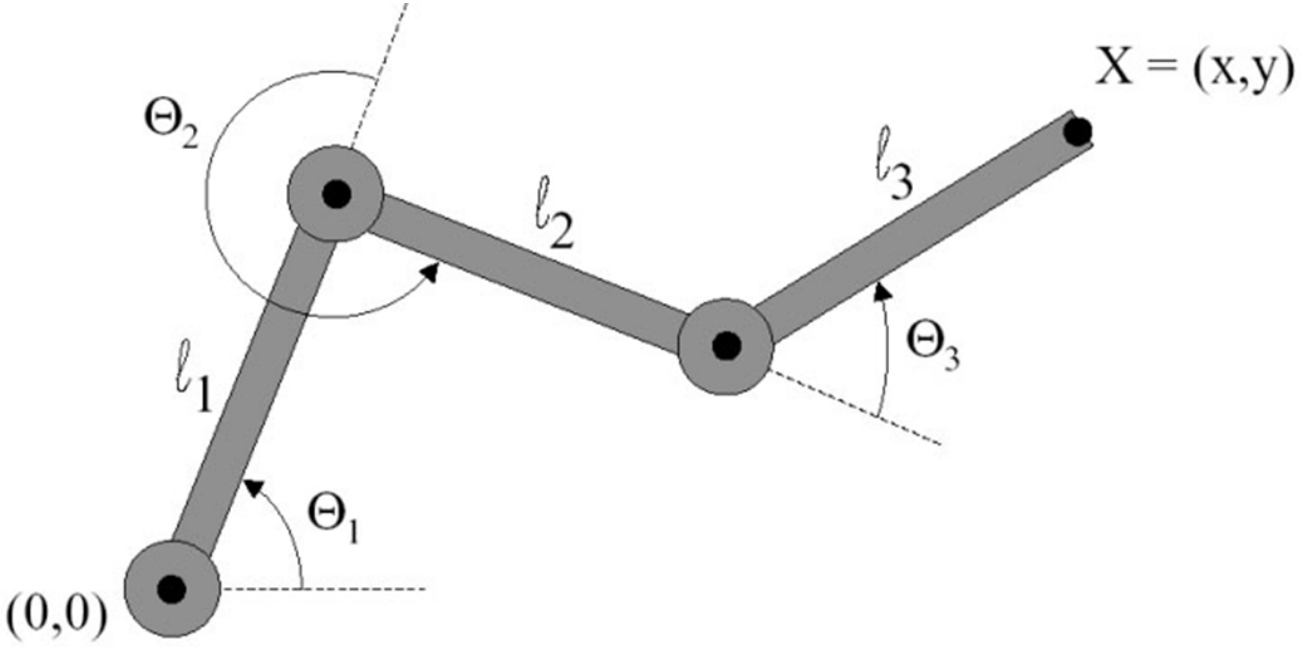
$$\Theta_2 = \cos^{-1}\left(\frac{x^2 + y^2 - l_1^{\,2} - l_2^{\,2}}{2 l_1 l_2}\right)$$

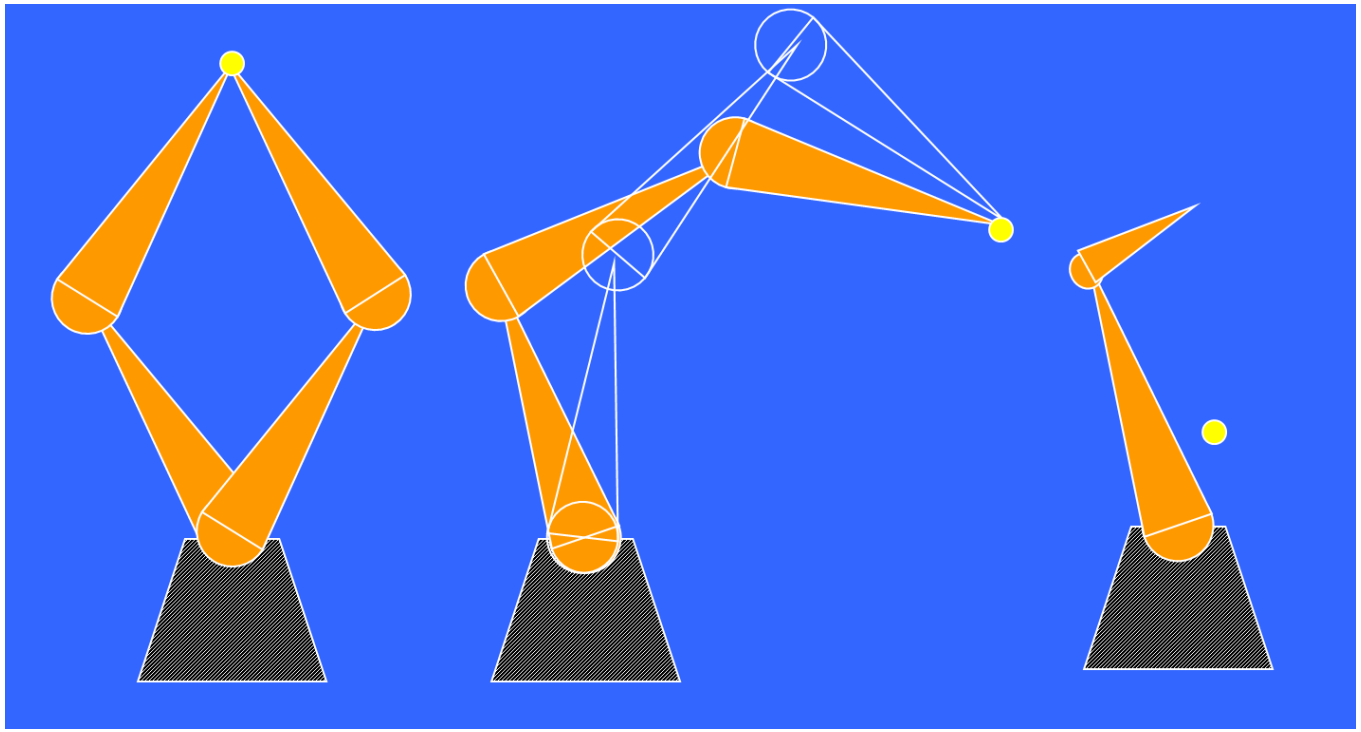
Problémák

- Nehéz ”természetesnek látszó” mozgást leírni vele.
- Az inverz függvény kiszámítása nem triviális,
- és nem is egyértelmű (redundancia).

Példa

- Egyenletek száma: 2, ismeretlen változók száma: 3 $\Rightarrow$ Végtelen sok megoldás!
- Rendszer DOF $>$ végberendezés DOF
- Az emberi csontváz kb 70 DOF!



Példa

Nem egyértelmű, ill. nem létező megoldás