

Bevezetés a számítógépi grafikába

3 dimenziós felületek (testek) megjelenítése

Troll Ede Mátyás

Matematikai és Informatikai Intézet
Eszterházy Károly Egyetem

Eger, 2019



1 Testek megjelenítése

- Hátsó lapok eltávolítása
- Z-buffer algoritmus

2 Árnyalási technikák

- Konstans árnyalás
- Gouraudárnyalás
- Phong árnyalás

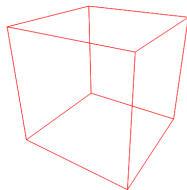
1 Testek megjelenítése

- Hátsó lapok eltávolítása
- Z-buffer algoritmus

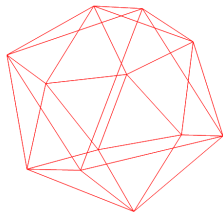
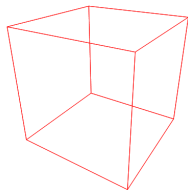
2 Árnyalási technikák

- Konstans árnyalás
- Gouraudárnyalás
- Phong árnyalás

Hátsó lapok eltávolítása



Hátsó lapok eltávolítása



Hátsó lapok eltávolítása

Zárt poliéderek megjelenítését általában a hátsó, nem látható lapok eltávolításával kezdjük.

Hátsó lapok eltávolítása

Zárt poliéderek megjelenítését általában a hátsó, nem látható lapok eltávolításával kezdjük.

Ehhez előállítjuk az adott lap egy normálvektorát egy adott pontban (vektoriális szorzat),

Hátsó lapok eltávolítása

Zárt poliéderek megjelenítését általában a hátsó, nem látható lapok eltávolításával kezdjük.

Ehhez előállítjuk az adott lap egy normálvektorát egy adott pontban (vektoriális szorzat), majd annak az adotttól a kamerába mutató vektor szögének függvényében (90° -nál nagyobb szöget zárnak be) jelenítjük meg a lapot (skaláris szorzat).

Hátsó lapok eltávolítása

Zárt poliéderek megjelenítését általában a hátsó, nem látható lapok eltávolításával kezdjük.

Ehhez előállítjuk az adott lap egy normálvektorát egy adott pontban (vektoriális szorzat), majd annak az adott pól a kamerába mutató vektor szögének függvényében (90° -nál nagyobb szöget zárnak be) jelenítjük meg a lapot (skaláris szorzat).

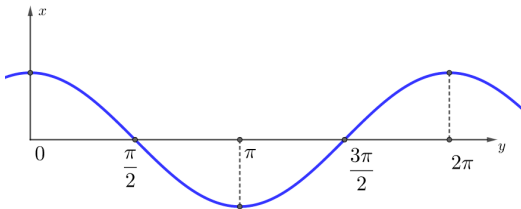
$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = |\mathbf{a}| |\mathbf{b}| \cos \phi$$

Hátsó lapok eltávolítása

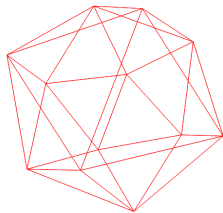
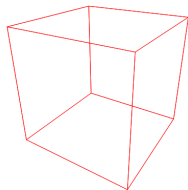
Zárt poliéderek megjelenítését általában a hátsó, nem látható lapok eltávolításával kezdjük.

Ehhez előállítjuk az adott lap egy normálvektorát egy adott pontban (vektoriális szorzat), majd annak az adotpól a kamerába mutató vektor szögének függvényében (90° -nál nagyobb szöget zárnak be) jelenítjük meg a lapot (skaláris szorzat).

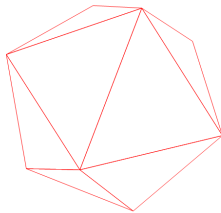
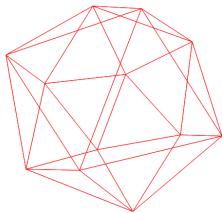
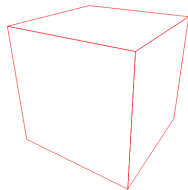
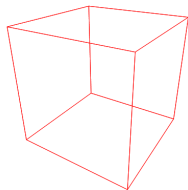
$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = |\mathbf{a}| |\mathbf{b}| \cos \phi$$



Hátsó lapok eltávolítása



Hátsó lapok eltávolítása



Mielőtt megjelenítenénk a lapokat, az átfedések kezelésére rendezzük azokat.

Mielőtt megjelenítenénk a lapokat, az átfedések kezelésére rendezzük azokat. Hogyan rendezzünk háromszögeket?

Mielőtt megjelenítenénk a lapokat, az átfedések kezelésére rendezzük azokat. Hogyan rendezzünk háromszögeket?

A háromszögeket súlypontjuk z koordinátája szerint érdemes rendezni, és az így kapott háromszögeket hátulról kezdjük megjeleníteni. Így az elől levő lapok fogják kitakarni a hátsókat.

Mielőtt megjelenítenénk a lapokat, az átfedések kezelésére rendezzük azokat. Hogyan rendezzünk háromszögeket?

A háromszögeket súlypontjuk z koordinátája szerint érdemes rendezni, és az így kapott háromszögeket hátulról kezdjük megjeleníteni. Így az elől levő lapok fogják kitakarni a hátsókat.

A rendezéshez érdemes a *quick sort* algoritmust módosítani úgy, hogy az a háromszögek fent említett adata alapján rendezzen.

Mielőtt megjelenítenénk a lapokat, az átfedések kezelésére rendezzük azokat. Hogyan rendezzünk háromszögeket?

A háromszögeket súlypontjuk z koordinátája szerint érdemes rendezni, és az így kapott háromszögeket hátulról kezdjük megjeleníteni. Így az elől levő lapok fogják kitakarni a hátsókat.

A rendezéshez érdemes a *quick sort* algoritmust módosítani úgy, hogy az a háromszögek fent említett adata alapján rendezzen. Az algoritmus tulajdonságai alapján az első rendezés lesz a leglassabb, viszont minden más esetben egy már nagyjából rendezett struktúrát kell rendeznünk.

Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus.

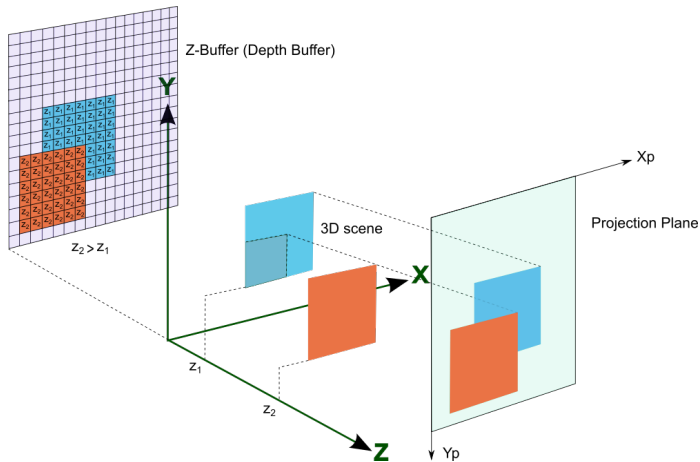
Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta.

Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta. Az algoritmus implementálásakor szükség van a színek tárolása mellett az egyes mélységek tárolására is, innen az algoritmus neve.

Z-buffer algoritmus



Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta. Az algoritmus implementálásakor szükség van a színek tárolása mellett az egyes mélységek tárolására is, innen az algoritmus neve. Az algoritmus élőszavas leírása:

Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta. Az algoritmus implementálásakor szükség van a színek tárolása mellett az egyes mélységek tárolására is, innen az algoritmus neve. Az algoritmus előszavas leírása:

A mélység puffert maximális mélységre, a pixelekhez pedig a háttérszínre rendeljük

Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta. Az algoritmus implementálásakor szükség van a színek tárolása mellett az egyes mélységek tárolására is, innen az algoritmus neve. Az algoritmus előszavas leírása:

A mélység puffert maximális mélységre, a pixelekhez pedig a háttérszínre rendeljük

Bejárjuk az összes objektumot

Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta. Az algoritmus implementálásakor szükség van a színek tárolása mellett az egyes mélységek tárolására is, innen az algoritmus neve. Az algoritmus élőszavas leírása:

A mélység puffert maximális mélységre, a pixelekhez pedig a háttérszínre rendeljük

- Bejárjuk az összes objektumot

- Bejárjuk az objektum vetületének összes pixelét

Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta. Az algoritmus implementálásakor szükség van a színek tárolása mellett az egyes mélységek tárolására is, innen az algoritmus neve. Az algoritmus élőszavas leírása:

A mélység puffert maximális mélységre, a pixelekhez pedig a háttérszínre rendeljük

- Bejárjuk az összes objektumot

 - Bejárjuk az objektum vetületének összes pixelét

 - Ha az adott pixelhez tartozó eredeti pont z koordinátája nagyobb, mint az ebben a pontban tárolt mélység érték

Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta. Az algoritmus implementálásakor szükség van a színek tárolása mellett az egyes mélységek tárolására is, innen az algoritmus neve. Az algoritmus előszavas leírása:

A mélység puffert maximális mélységre, a pixelekhez pedig a háttérszínrt rendeljük

- Bejárjuk az összes objektumot

 - Bejárjuk az objektum vetületének összes pixelét

 - Ha az adott pixelhez tartozó eredeti pont z koordinátája nagyobb, mint az ebben a pontban tárolt mélység érték

 - Kiszámítjuk és tároljuk a pixel színét az adott pontban

Z-buffer algoritmus

A Z-buffer algoritmus poligonokkal határolt alakzatok megjelenítésére szolgáló algoritmus. Az algoritmus az egyes pixeleken azt vizsgálja, hogy melyik alakzat látszik rajta. Az algoritmus implementálásakor szükség van a színek tárolása mellett az egyes mélységek tárolására is, innen az algoritmus neve. Az algoritmus előszavas leírása:

A mélység puffert maximális mélységre, a pixelekhez pedig a háttérszínre rendeljük

- Bejárjuk az összes objektumot

- Bejárjuk az objektum vetületének összes pixelét

- Ha az adott pixelhez tartozó eredeti pont z koordinátája nagyobb, mint az ebben a pontban tárolt mélység érték

- Kiszámítjuk és tároljuk a pixel színét az adott pontban

- A mélység értéket az objektum z koordinátájának értékére állítjuk

1 Testek megjelenítése

- Hátsó lapok eltávolítása
- Z-buffer algoritmus

2 Árnyalási technikák

- Konstans árnyalás
- Gouraudárnyalás
- Phong árnyalás

A megjelenítés során természetes igényként merül föl, hogy az adott testet ne pusztán a háttérszínnel töltsük ki, hanem színezzük is azt ki.

A megjelenítés során természetes igényként merül föl, hogy az adott testet ne pusztán a háttérszínnel töltsük ki, hanem színezzük is azt ki.

Jelen fejezetben a megjelenítésre koncentrálunk, az anyag és a fényforrás tulajdonságainak kezelését később tárgyaljuk.

Konstans árnyalás

A Flat shading (konstans árnyalás) a poligonhálók legegyszerűbb megjelenítési eljárása.

Konstans árnyalás

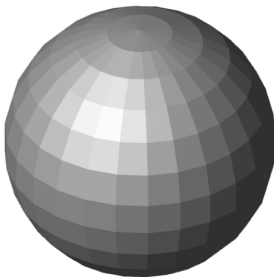
A Flat shading (konstans árnyalás) a poligonhálók legegyszerűbb megjelenítési eljárása.

Az eljárás során minden lap esetén egy pontban kiszámítjuk a színértéket, és ezt alkalmazzuk a teljes lap minden pixelén.

Konstans árnyalás

A Flat shading (konstans árnyalás) a poligonhálók legegyszerűbb megjelenítési eljárása.

Az eljárás során minden lap esetén egy pontban kiszámítjuk a színértéket, és ezt alkalmazzuk a teljes lap minden pixelén.



Az eljárás mint látjuk nem ad kielégítő eredményt.

Az eljárás mint látjuk nem ad kielégítő eredményt. Jelen eljárás az alábbi feltételek esetén lenne megfelelő:

- az alakzatot megvilágító fénysugarak párhuzamosak

Az eljárás mint látjuk nem ad kielégítő eredményt. Jelen eljárás az alábbi feltételek esetén lenne megfelelő:

- az alakzatot megvilágító fénysugarak párhuzamosak
- a nézőpont végtelen távol van (párhuzamos vetítés)

Az eljárás mint látjuk nem ad kielégítő eredményt. Jelen eljárás az alábbi feltételek esetén lenne megfelelő:

- az alakzatot megvilágító fénysugarak párhuzamosak
- a nézőpont végtelen távol van (párhuzamos vetítés)
- az ábrázolandó objektum poliéder, nem pedig egy görbült felület poliéderrel való közelítése

A megjelenítés során általában a felületi normálisok alapján határozzuk meg a színeket.

A megjelenítés során általában a felületi normálisok alapján határozzuk meg a színeket.

Ez azonban a szomszédos lapok egymáshoz való viszonya miatt ugrásszerű változásokat jelenthet az egyes lapok színeiben.

A megjelenítés során általában a felületi normálisok alapján határozzuk meg a színeket.

Ez azonban a szomszédos lapok egymáshoz való viszonya miatt ugrásszerű változásokat jelenthet az egyes lapok színeiben.

A probléma kiküszöbölésére elterjedt általános módszer az, hogy a csatlakozó lapok közös csúcspontjában kiszámítjuk a normálisokat, majd azoknak vesszük azok átlagát, és azt rendeljük a megjelenítendő laphoz.

A megjelenítés során általában a felületi normálisok alapján határozzuk meg a színeket.

Ez azonban a szomszédos lapok egymáshoz való viszonya miatt ugrásszerű változásokat jelenthet az egyes lapok színeiben.

A probléma kiküszöbölésére elterjedt általános módszer az, hogy a csatlakozó lapok közös csúcspontjában kiszámítjuk a normálisokat, majd azoknak vesszük azok átlagát, és azt rendeljük a megjelenítendő laphoz.

Ekkor általában az így kapott vektor egyik lapra sem lesz merőleges.

A megjelenítés során általában a felületi normálisok alapján határozzuk meg a színeket.

Ez azonban a szomszédos lapok egymáshoz való viszonya miatt ugrásszerű változásokat jelenthet az egyes lapok színeiben.

A probléma kiküszöbölésére elterjedt általános módszer az, hogy a csatlakozó lapok közös csúcspontjában kiszámítjuk a normálisokat, majd azoknak vesszük azok átlagát, és azt rendeljük a megjelenítendő laphoz.

Ekkor általában az így kapott vektor egyik lapra sem lesz merőleges.

Görbült felületek esetén konstans árnyalás helyett általában valamilyen folytonos árnyalást (smooth shading) alkalmazunk.

Gouraud árnyalás

Gouraud árnyalás esetén a háromszöglap minden csúcsában kiszámítjuk a színinformációt a normálisok segítségével.

Gouraud árnyalás

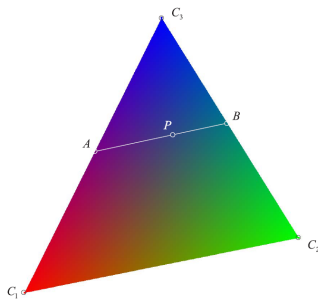
Gouraud árnyalás esetén a háromszöglap minden csúcsában kiszámítjuk a színinformációt a normálisok segítségével.

A lap élein a csúcspontok színének lineáris interpolációjával határozzuk meg a színt,

Gouraud árnyalás

Gouraud árnyalás esetén a háromszög lap minden csúcsában kiszámítjuk a színinformációt a normálisok segítségével.

A lap élein a csúcspontok színének lineáris interpolációjával határozzuk meg a színt, míg belső pontok esetén az alábbi ábra szerinti kettős lineáris interpolációval.



Gouraud árnyalás

Gouraud árnyalás esetén a háromszöglap minden csúcsában kiszámítjuk a színinformációt a normálisok segítségével.

Gouraud árnyalás

Gouraud árnyalás esetén a háromszöglap minden csúcsában kiszámítjuk a színinformációt a normálisok segítségével.

A lap élein a csúcspontok színének lineáris interpolációjával határozzuk meg a színt,

Gouraud árnyalás

Gouraud árnyalás esetén a háromszöglap minden csúcsában kiszámítjuk a színinformációt a normálisok segítségével.

A lap élein a csúcspontok színének lineáris interpolációjával határozzuk meg a színt, míg belső pontok esetén az alábbi ábra szerinti kettős lineáris interpolációval.



Az eljárásnak sajnos vannak hátrányai:

Az eljárásnak sajnos vannak hátrányai:

- A test körvonala nem lesz sima (látszik a poligonnal való közelítés)

Az eljárásnak sajnos vannak hátrányai:

- A test körvonala nem lesz sima (látszik a poligonnal való közelítés)
- Nagy tükröződő felületeknél az emberi szem számára ugrásszerű változás mutatkozik (Mach-sáv hatás).

Az eljárásnak sajnos vannak hátrányai:

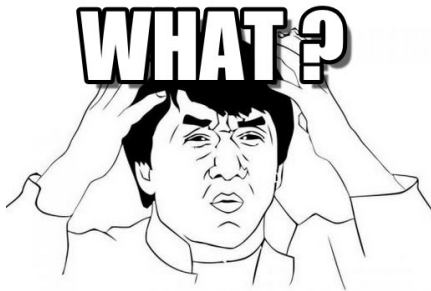
- A test körvonala nem lesz sima (látszik a poligonnal való közelítés)
- Nagy tükröződő felületeknél az emberi szem számára ugrásszerű változás mutatkozik (Mach-sáv hatás).

A fenti problémák csökkenthetők a poligonháló finomításával, de ekkor megnő a tárigény és a feldolgozási idő is.

A Phong árnyalás abban különbözik a Gouraud árnyalástól, hogy a színek helyett a normálisokat interpolálja.

Phong árnyalás

A Phong árnyalás abban különbözik a Gouraud árnyalástól, hogy a színek helyett a normálisokat interpolálja.



Phong árnyalás

A Phong árnyalás abban különbözik a Gouraud árnyalástól, hogy a színek helyett a normálisokat interpolálja.

Phong árnyalás

A Phong árnyalás abban különbözik a Gouraud árnyalástól, hogy a színek helyett a normálisokat interpolálja.

Tehát a lap élei mentén a csúcspontokban levő normálisok lineáris interpolációjával határozzuk meg a normálist,

Phong árnyalás

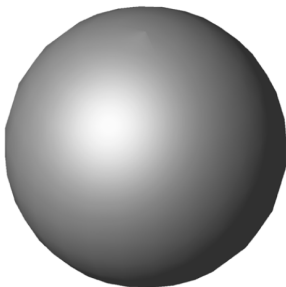
A Phong árnyalás abban különbözik a Gouraud árnyalástól, hogy a színek helyett a normálisokat interpolálja.

Tehát a lap élei mentén a csúcspontokban levő normálisok lineáris interpolációjával határozzuk meg a normálist, míg belső pont esetén kettős interpolációval.

Phong árnyalás

A Phong árnyalás abban különbözik a Gouraud árnyalástól, hogy a színek helyett a normálisokat interpolálja.

Tehát a lap élei mentén a csúcspontokban levő normálisok lineáris interpolációjával határozzuk meg a normálist, míg belső pont esetén kettős interpolációval.



Végeredményben tehát minden pontban ki kell számítani (interpolálni) a felületi normálist, majd annak értéke alapján kiszámítani a pontbeli színértéket.

Végeredményben tehát minden pontban ki kell számítani (interpolálni) a felületi normálist, majd annak értéke alapján kiszámítani a pontbeli színértéket.

Az eljárás természetesen több számítást igényel a Gouraud árnyalásnál, viszont szebb eredményt ad.

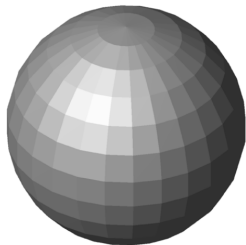
Végeredményben tehát minden pontban ki kell számítani (interpolálni) a felületi normálist, majd annak értéke alapján kiszámítani a pontbeli színértéket.

Az eljárás természetesen több számítást igényel a Gouraud árnyalásnál, viszont szebb eredményt ad.

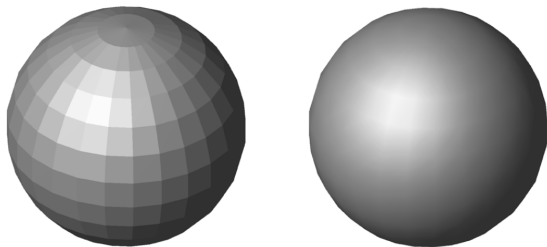
A képhatár szögletessége továbbra is csak a poligonháló finomításával lehetséges.

Árnyalások eredményének összehasonlítása

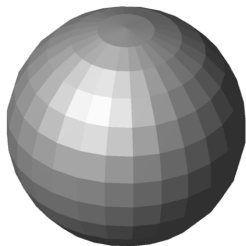
Árnyalások eredményének összehasonlítása



Árnyalások eredményének összehasonlítása



Árnyalások eredményének összehasonlítása



Köszönöm a figyelmet!