

konvex ponthalmaz: ha az alakzat bármely 2 pontjával együtt a 2 pontot összekötő szakasz valamennyi pontját is tartalmazza.

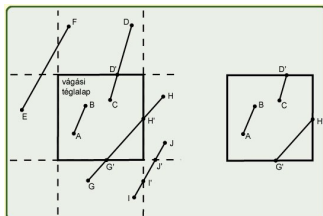
konkáv ponthalmaz: ha van olyan az alakzat 2 pontját összekötő szakasz, amelyik nem tartozik teljes egészében az alakzathoz.

konvex halmaz burka: a legszűkebb konvex poligon, mely vagy belső pontként vagy csúcspontként tartalmazza a halmaz összes pontját.

egyszerű sokszög: nem keresztezheti önmagát. (a benne szereplő szakaszoknak az előírt csatlakozási pontokon kívül nincs más közös pontjuk.)

Chonon-Sutherland szakaszvágó algoritmus:

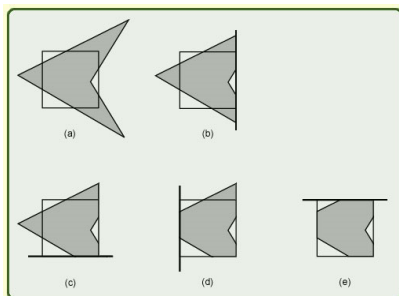
1001 1000 1010 A síkot 9 részre osztja,
0001 0000 0010 a középső kilenced (0000) a képernyő.
0101 0100 0110



A szakasz kezdő- és végpontjának a kódja: a,b.

1. Ha $a|b = 0$ a szakasz vágás nélkül megrajzolható.
2. Ha $a \& b \neq 0$ a szakasz elhagyható, mert egyetlen pontja sem látszik.
3. Megkeressük az első olyan oldalegyenest, melyik metszi a szakaszt, a metszésponttal helyettesítjük a megfelelő végpontot, és az eljárást az 1.lépéstől újrakezdjük.

Sutherland – Hodgman poligonvágó algoritmus



Végigmegy a képmező oldalain, és az oldalegyenesekkel elvágja a poligont úgy, hogy minden egyes vágás után lezárja a csonkolt poligont az oldalegyenes megfelelő szakaszával.

Hátsó lapok eltávolítása (back-face culling)

Poliéderek ábrázolásánál célszerű használni.

- ha a normálissal bezárt szög 90° ében látjuk
- ha 90° -nál nagyobb: biztosan nem látható
- ha 90° -nál kisebb: van esély hogy látható, de takarásban is lehet.

Konvex poliéder esetén biztosan látható. Konkáv poliéder esetén lehet nem.

Zárt felület esetén is csak akkor eredményez korrekt, láthatóság szerinti képet, ha egyetlen konvex poliédert ábrázolunk. Konkáv vagy több egymást részben átfedő poliéder esetén további vizsgálat szükséges. Ne használjuk nyílt poligonháló esetén.

Festő algoritmus

Alapfoglata, hogy a megjelenítendő objektumokat a nézőponttól való távolság függvényében sorba kell állítani. Az algoritmus a helyes takarási viszonyokat úgy alakítja ki, hogy a nézőhöz közelebb eső objektum képe felülírja a távolabbi objektum képét. A *képfestés* technikájához hasonlóan a távolabbi objektumokra ráfesti a közelebbieket.

Lépések:

- rendezzük a poligonokat mélység szerint
- hátulról előre haladva fessük ki a poligonokat

Problémák: ciklikus átfedés (megoldás: az egyik poligon kettévágása),

lapok kölcsönös átfedése
(megoldás: poligon kettévágása)

Mélység-puffer (z-puffer) algoritmus

Megközelítése: Mi látszik 1-1 pixelen?! Pixelenként tárolandó: szín, mélység (z-puffer)

Lépések:

A mélység puffert a maximális mélységre állítjuk, vagyis arra a mélységre, amelytől a távolabbi pontokat nem veszünk figyelembe, a pixelekhez pedig a háttér színét rendeljük. Az objektumnak a határoló felületén végighaladva a határoló felületet pixelekké konvertálva, a felület képére eső minden pixelre:

- meghatározzuk a pixelhez tartozó felületi pont mélységét
- ha ez az érték kisebb a pixeleknek a mélység-pufferben tárolt értékénél, vagyis az új pont látszik, akkor:
 - a pixel mélységét a pont mélységére állítjuk
 - meghatározzuk a pont színét és ezt rendeljük a pixelhez

előnye: egyszerű,
gyakorlatilag minden modellel
alkalmazható (eredendően poliédere fejlesztették ki)
gyorsítható párhuzamos feldolgozással
hardver támogatottsága jó
hátránya: memóriai igényes

Scan-line algoritmus (pixelsor algoritmus)

A scan-line algoritmusok pixelsorokként készítik a képet. A pixelsorra illeszkedő vetítő síkkal elmetsszük az objektumokat és a síkhoz legközelebb eső metsző pontokat rajzoljuk ki.

A pixelsor és a háromszögek közös részét szegmensnek nevezzük. Ha a pixelsor egy szegmense csak egy háromszöghöz tartozik, akkor ezt egyszerűen meg kell csak jeleníteni. Ha egy pixel több szegmenshez tartozik, akkor a megfelelő háromszögek mélységi vizsgálatával kell eldönteni, hogy melyik háromszög felületi pontját kell kirajzolni.

előnye: felesleges lépések kiszűrhetőek
gyorsítás: párhuzamos feldolgozás; hardver implementáció

Fénysugár követő algoritmus (ray tracing, ray casting)

A fénysugárkövető módszer is azt használja ki, hogy a kép pontokból áll. Alapötlete az, hogy meghatározza az ábrázolandó objektumnak azt a pontját, amely egy adott pixelen látszik.

Lépések:

A nézőpontot összekötjük a kép pixeleivel, azaz egy sugarat bocsátunk ki a nézőpontból a pixeleknek megfelelő téglalap szimmetriaközéppontján át a modellterbe. Ezekkel a félegyenesekkel elmetsszük az ábrázolandó objektumokat. Meghatározzuk a nézőponthoz legközelebbi metszéspontot, majd ennek a pontnak a színét, és a pixelhez ezt a színt rendeljük.

előnye: egyszerű
jó minőségű kép (optikai hatások jól modellezhető)
hátránya: sok számítást igényel, így időigényes.
gyorsítás: párhuzamos feldolgozás

Terület-felosztásos algoritmus

A megjelenítési területet rekurzívan addig bontjuk téglalapokra, míg csak egy poligon esik bele, azaz csak egy színt kell kirajzolnunk (egyszerű eset).

Warnock-féle algoritmus:

Feltételezi, hogy poligonokat akarunk megjeleníteni. A megjelenítendő poligonoknak a vizsgált területtel a kapcsolata lehet: körülveszi; metszi; benne van; nincs közös részük.

Lépések:

Ha a megjelenítendő poligonoknak nincs közös része a területtel, fessük ki a területet a háttérszínnel.
Ha csak egy poligon metszi a területet, vagy benne van a területben:

- fessük ki a területet a háttérszínnel, majd jelenítsük meg a poligont.

Ha csak egyetlen poligon veszi körül a területet fessük ki a területet a poligon színével.

Ha több olyan poligon is van, amelyek körülveszik, metszik vagy benne vannak a területben, de egy körülölelő poligon a többi előtt van: fessük ki a területet a poligon színével.
Ha a fentiek egyike sem teljesül: osszuk fel a területet 4 részre, és minden új területre:

ha elértük a kép felbontását rendezzük a poligonokat, a legközelebbi színével fessük ki a pixel.
rekurzív folytatassuk az eljárást az 1. lépéstől.

Szín, színrendszer

A színeket legcélszerűbb egy háromdimenziós tér pontjaiként felfogni. A tér egy bázisának rögzítése után minden színt egy számhármassal segítségével lehet azonosítani. Kétféle színmegadási (színkeverési) mód:

- additív (hozzáadó) színkeverés,
- szubtraktív (kivonó).

Additív színkeverés esetén azt írjuk elő, hogy mennyi vörös (Red), zöld (Green) és kék (Blue) színkomponenst adjunk a fekete színhez. A színkomponensek értéke a [0., 1.] intervallumon változtatható. Ezt a színelőállítási módot RGB színrendszernek szokás nevezni. Ezzel összhangban van:

- a színes raszteres grafikus display-k működése (a képernyő minden egyes pixele különböző intenzitású vörös, zöld és kék színt tud kibocsátani)
- az emberi szem működése (világosságát – pálcikák; színlátás – csapok (vörös, zöld és kék érzékelők))

Szubtraktív színkeverés esetén azt írjuk elő, hogy mennyi türkiz (Cyan), bíbor (Magenta) és sárga (Yellow) színt kell kivonnunk a legösszetettebb színből (fehér) a kívánt szín előállításához. Ezt CMY színrendszernek nevezzük.

CMYK színtér: megegyezik a CMY színtérrel, csak még hozzáadódik a fekete szín. A CMY alapszínekből csak szürkét tudunk előállítani, a nyomdatechnikában viszont szükség van fekete színre is.

Egyéb színkeverési módszerek: pl. a képzőművészek által használt színárnyalat, fényesség és telítettség (**HLS** – Hue, Lightness, Saturation) színmeghatározás.

RGBA mód: az RGB komponensek mellett egy negyedik, ún. **alfa** (A) komponenst is megadunk. Az alfa komponenssel az átlátszóságot (színek fedését) adhatjuk meg (nem színkomponens). Alfa 1. esetén a szín nem átlátszó, 0. esetén teljesen átlátszó.

Az ember által egyenletesen változó érzékelt intenzitásváltozás nem úgy érhető el, hogy az RGB értékeket lineárisan változtatjuk. $(0,1 \rightarrow 0,11$ jelentősebbnek érzékeljük, mint a $0,5 \rightarrow 0,55$).

A fényerő érzékelésénél is hasonlóan $(50W \rightarrow 100W$ nagyobb ugrásnak érzékeljük, mint $100W \rightarrow 150W$). Monitorok esetén ennek a problémának a megoldását gamma-korrekciónak nevezik.

A grafikus kártyán a képernyő minden pixeléhez ugyanannyi memória tartozik, melyben a pixelek színét tároljuk. A puffer méretét azzal jellemezzük, hogy pixelenként hány bit áll rendelkezésre a színek tárolására (bit/pixel).

$32 \text{ bit/pixel} - \text{RGBA azaz } 2^{32} - 2^8 = 16,77 \text{ millió szín és átlátszóság; true color.}$

$64 \text{ bit/pixel} - \text{animáció esetén (a memória lehetővé tesz két } 32 \text{ bit/pixel színpuffert)}$

Ezeket a paramétereket illetve hogy melyik színindexhez milyen RGB érték tartozik, lekérdezhetjük.

A számítógépek grafikus alaprendszereiben a színek megadása általában kétféleképpen történhet.

1. közvetlenül megadjuk a kívánt szín RGB komponenseit,
2. egy indexszel hivatkozunk egy színtáblázatra (színpaletta) valamely elemére. Természetesen a paletta színei a felhasználó által megváltoztathatók.

A felhasználói programban döntünk kell a színmegadás módjáról, és ez a továbbiakban nem változtatható meg. Egy alkalmazásban vagy végig RGBA módban, vagy színindex módban kell dolgoznunk. A rendszer a színt a csúcspontokban számolja ki. A végső színt befolyásolják: megvilágítás, árnyalás, textúra.

Mikor használunk RGBA módot? Legtöbb rendszerben RGBA módban több szín jeleníthető meg, továbbá számos speciális effektus, mint pl. az árnyalás, megvilágítás, textúra-leképezés, atmoszférikus hatások, antialiasing és átlátszóság RGBA módban sokkal eredményesebben valósítható meg.

A színindex mód előnyei. Célszerű színindex módot választanunk,

- ha egy olyan programot kell átírunk OpenGL-be, amely már ezt a módot használta;
- ha kevesebb színt akarunk használni, mint a színpuffer által kínált maximum;
- ha a bit/pixel érték alacsony, azaz kevés szín jeleníthető meg;
- ha olyan speciális trükköket akarunk megvalósítani, mint a színanimáció, vagy a rétegekbe rajzolás;
- ha pixelenkénti logikai műveleteket akarunk a színeken végrehajtani.

Megvilágítás, fényforrások, árnyalás:

Az OpenGL csak a csúcspontokban számítja ki a szint. Egy csúcspont színe nem más, mint a pontból a nézőpontba (a szembe) eljutó fény színe, amit az objektum anyagának tulajdonságai, az uralkodó fényviszonyok, továbbá az ábrázolt alakzatok optikai kölcsönhatásai határoznak meg.

Fényösszetevők:

• **A környezeti fény** (ambient light) az ábrázolandó térrészben mindenütt jelen lévő, állandó intenzitású fény, melynek forrása, iránya nem ismert. A környezeti fény két részből tevődik össze:

- a térben a fényforrásoktól függetlenül jelen lévő környezeti fényből – globális környezeti fény
- a fényforrásokból származó (pl. többszörös tükröződések) környezeti fényből - fényforrások környezeti fénykomponense

• **A szórt fénynek** (diffuse light) van iránya, mindig valamelyik fényforrásból jön. Fő jellemzője, hogy az objektumokkal ütközve minden irányba azonos módon és mértékben verődik vissza. Hatása nézőponttól független, csak a fényforrástól, az anyagtulajdonságoktól és a csúcspontbeli normálistól függ.

• **A tükrözött fénynek** (specular light) is van iránya és forrása, és hatása nemcsak az anyagtulajdonságoktól és a csúcspontbeli normálistól, hanem a nézőponttól is függ.

Fényforrás megadása

- **fényforrás helye** Lehet végesben lévő, vagy végtelen távoli pont is.

- A végtelen távolban lévő fényforrásból kibocsátott sugarak párhuzamosak (napsugarak).
- A végesben lévő fényforrások (pontoszerű fényforrások) azonos intenzitású fényt bocsátanak ki minden irányba. Megadható, hogy a fényforrástól távolodva milyen mértékben csökken a fény erőssége, azaz hogyan tompul a fény (attenuation).

Reflektor (spot-light): Végesben lévő fényforrásból reflektort is létrehozhatunk. Reflektor létrehozásához meg kell adnunk a fénykúp tengelyének irányát és fél nyílásszögét, továbbá azt, hogy a fény intenzitása hogyan csökken a kúp tengelyétől a palást felé haladva.

- **kibocsátott fény** A fényforrás által kibocsátott fény 3 összetevőből áll: környezeti fény (ambient light), szórt fény (diffuse light) és tükrözött fény (specular light). Mindhárom színét az RGBA komponenseivel kell megadni.

- **hozzájárulása a környezeti fényhez** A környezeti fényösszetevő azt adja meg, hogy az adott fényforrás milyen mértékben járul hozzá a térrész környezeti fényéhez.

Anyagtulajdonságok

Az objektumok anyagának optikai tulajdonságai:

- az objektum által kibocsátott fényt;
- az anyag milyen mértékben veri vissza a környezeti, a szórt és a tükrözött fényt;
- a ragyogást.

Fényvisszaverési tulajdonságok megadása: (tapasztalati mennyiségek)

- környezeti fény visszaverődési együttható (ambient reflection);
- szórt fény visszaverődési együttható (diffuse reflection);
- tükrözött fény visszaverődési együttható (specular reflection).

Árnyalás:

Konstans árnyalás (flat shading) esetén minden lapnak csak egyetlen pontjában számítjuk ki a szint, és az egész lapra ezt alkalmazzuk. Csak akkor megoldás, ha a fénysugarak párhuzamosak; a nézőpont végtelen távoli pont és az ábrázolt objektum poliéder.

Folytonos árnyalás (smooth shading) esetén a csúcspontok színét –RGBA módban a szint komponensenként, színindex módban az indexeket – lineárisan interpolálja a rendszer a szakaszok belső pontjai színének kiszámításához. Folytonos árnyalás alkalmazásával el tudjuk tüntetni a képről a görbült felületeket közelítő poligonháló éleit.

Gouroud-féle árnyalás: a csúcspontokban a csúcspontbeli normálisok segítségével kiszámítjuk a szint. A lap élei mentén a csúcspontok színének lineáris interpolációjával határozzuk meg a szint, a lap belső pontjaiban pedig kettős lineáris interpolációval. Hátránya, hogy a test körvonala nem lesz sima; Mach-sáv hatás (megoldás: modell finomítása).

Phong-féle árnyalás: nem a fény intenzitását interpolálja, hanem a normálisokat. Hátránya a képhatár szögletessége.

Fényképszerű képek létrehozása

- optikai kölcsönhatások modellezése (átlátszóság; árnyék; tükröződés)
- testek által kibocsátott fény
- felületi érdesség

Átlátszóság:

- fénytörés nélkül (röntgenszerű): a fénysugár irányváltoztatás nélkül halad át a testen.
- szórásmentes fénytöréssel: a fénysugár a testbe való be- és kilépéskor megváltoztatja irányát.
- átlátszósági együttható: valós szám 0..1 között. Minden szinkomponensre ezt alkalmazzák.

Tükröződés: az adott felületi pontban valamely más test látszik-e.

Árnyék: a fényforrás és a vizsgált alakzat között van-e valamilyen tárgy. (éles és elmosódott határ)

Textúra: A felületre „ráhúzott” mintázat. A felszín leképzése a síkra (topologikus transzformáció). Egyes rendszerek többretegű, az egyes rétegek átlátszóságának mértékét is megadó un. 3D-s textúrákat is megengednek. A textúrát előállíthatjuk valamilyen algoritmussal, de egy fénykép raszterizálásával kapott pixeltömbbel is. A textúrát a modelltérből rendeljük hozzá a geometriai alakzathoz, így a felületi mintán is érvényesül a centrális vetítésből eredő, un. perspektív torzítás. A textúra sorokba és oszlopokba rendezett texelekből (texture element) épül fel. Mipmapping (sok dolog egy helyen) -> textúra részletek.

Felületi érdesség (bump mapping)

A számítógéppel létrehozott képek általában sima, fényes, fémesen csillogó felületeket ábrázolnak. A felületi érdesség azt jelenti, hogy a határoló lap nem sík, hanem a síkból kisebb-nagyobb besüllyedéssel, kitüremkedéssel létrehozott görbült felület. A felületi normálisok megváltoztatásával érijük el. (véletlenszerűen, algoritmus szerint, bittérkép szerint)

Radiosity módszer:

Azon a feltételezésen alapszik, hogy egy zárt környezetben a fényenergia állandó. Nem különböztet meg fényforrásokat és ábrázolandó testeket, hanem azt feltételezi, hogy minden test kibocsáthat, átengedhet, elnyelhet és visszaverhet fényt. Iránytól független hatások modellezése (környezeti fény, szórt visszaverődés). Azonos megvilágítású foltokat keres. Kombinálás a fénysugár követő módszerrel.

Transzformációk

- **nézőpontba transzformálás:** A nézőpont és a nézési irány kiválasztásával azt határozzuk meg, hogy az alakzatot honnan és milyen irányból nézzük, vagyis azt, hogy az alakzat mely részét látjuk. Alapértelmezés szerint az origóból nézünk a negatív z tengely irányába. Ezzel un. nézőpontkoordináta-rendszert hozzuk létre.

- **modelltranszformáció:** Az ábrázolandó alakzatot elmozgathatjuk a koordináta-rendszerhez képest, vagy akár alakjukat és méretarányaikat is megváltoztathatjuk.

Modelltranszformációk:

Egybevágósági transzformációk: bármely 2 pont és a neki megfelelő 2 pont távolsága egyenlő.

eltolás, elforgatás, tükrözés, mozgatás (eltolás és forgatás)

Hasonlósági transzformációk: bármely 2 pont és a neki megfelelő 2 pont távolságának aránya állandó. kicsinyítés-nagyítás

Affin transzformáció: osztóviszony állandó. skálázás, nyírás

Projektív transzformáció: kettősviszony állandó.

- **vetítési transzformáció:** transzformáció, mely a teret síkra képezi le. A rendszer erre két lehetőséget kínál: a centrális és a merőleges vetítést. A vetítés módjának kiválasztásához a nézőpontkoordináta- rendszerben meg kell adnunk azt a térrészt, amit ábrázolni szeretnénk. Ez centrális vetítés esetén csonka gúla, merőleges vetítésnél téglalestet. A vetítési transzformáció nem síkbeli képet eredményez, hanem a vetítendő térrészt (csonka gúlát, téglalestet) egy kockára képezi le. Centrális (perspektív) vetítést akkor alkalmazunk, ha valószerű képet szeretnénk létrehozni (az emberi látás is így működik), merőleges vetítést pedig méret- vagy méretarány-helyes képek létrehozásakor.

- **képmezőtranszformáció:** A képmező a képernyő ablakának az a téglalap alakú pixeltömbje, amire rajzolunk. A képmező-transzformáció a definiált kockát képezi le arra a téglalatra, melynek egyik lapja a képmező. Az (x, y) koordinátapárral a képmező bal alsó sarkát, a width paraméterrel a szélességét, a height paraméterrel pedig a magasságát kell megadnunk pixelekből.