



MITISZK
Miskolc-Térségi Integrált
Szakképző Központ

Magyarország célba ér



Számítógépi grafika

Juhász Imre

Lajos Sándor

**Miskolci Egyetem
Gépészmérnöki és Informatikai Kar**

Készült a HEFOP-3.2.2-P.-2004-10-0011-/1.0 sz.
projekt támogatásával

Miskolc, 2007.

Tartalomjegyzék

Előszó	1
1. Grafikai hardver	2
1.1. Output-eszközök	2
1.1.1. Grafikus display	2
1.1.2. Plotter	6
1.1.3. Hardcopy eszközök	8
1.2. Input-eszközök	10
2. Színek a számítógépi grafikában	16
3. Néhány grafikai algoritmus	19
3.1. Szakasz rajzolása raszteres megjelenítőn	19
3.2. Vágás a képmező határára	21
4. Grafikus alapelemek	25
5. Koordináta- és ponttranszformációk	28
5.1. Koordinátatranszformációk	28
5.1.1. Koordinátatranszformáció a síkon	28
5.1.2. Koordinátatranszformáció a térben	29
5.2. Ponttranszformációk	30
5.2.1. Síkbeli ponttranszformációk	31
5.2.2. Összetett transzformációk	35
5.2.3. Térbeli ponttranszformációk	37
6. Térbeli alakzatok megjelenítése	41
6.1. A tér leképezése a síkra	42
6.2. Láthatósági vizsgálatok	44
6.2.1. Hátsó lapok eltávolítása	44
6.2.2. Mélység-puffer (z-puffer) algoritmus	45
6.2.3. Fénysugárkövető algoritmus	46
7. Fényforrások, megvilágítás, árnyalás	47
7.1. Fényösszetevők	47
7.2. Fényforrások	47
7.3. Anyagtulajdonságok	51
7.4. Fényképszerű képek létrehozása	55

7.4.1.	Átlátszóság	55
7.4.2.	Árnyék és tükröződés	58
7.4.3.	Textúra	62
7.4.4.	Felületi érdekesség	64
8.	Görbék leírása és megjelenítése	65
8.1.	Görbék leírása	65
8.1.1.	Ívhossz	66
8.1.2.	Kísérő triéder	67
8.1.3.	Görbület, simulókör	68
8.1.4.	Torzió	69
8.2.	Görbék megjelenítése	70
9.	Felületek leírása és szemléltetése	74
9.1.	Felületek leírása	74
9.1.1.	Felületi görbe, érintősíkc, normális	76
9.2.	Felületek szemléltetése	77
9.2.1.	Poligonhálók árnyalt megjelenítése	83
10.	Animáció	87
A.	Vektorműveletek geometriai jelentése	88
A.1.	Skaláris szorzat	88
A.2.	Vektoriális szorzat	88
A.3.	Vegyes szorzat	91
B.	Egyenes és sík leírása	93
B.1.	Egyenes leírása	93
B.1.1.	Egyenes explicit leírása	93
B.1.2.	Egyenes implicit leírása	94
B.1.3.	Egyenes paraméteres leírása	95
B.1.4.	Egyenessel kapcsolatos feladatok	96
B.2.	Sík leírása	99
B.2.1.	Sík explicit leírása	99
B.2.2.	Sík implicit leírása	99
B.2.3.	Sík paraméteres leírása	100
B.2.4.	Síkkal kapcsolatos feladatok	100
	Irodalomjegyzék	102

Előszó

A számítógépi grafika az informatika rendkívül dinamikusan fejlődő részterülete. Kezdetben a grafika, az információ grafikus megjelenítése csak néhány kivételesen szerencsés helyzetű felhasználó privilégiuma volt. Mára ez teljesen megváltozott, gyakorlatilag minden kategóriájú számítógép rendelkezik jó minőségű grafikus képességekkel. Legtöbb esetben már az operációs rendszer is grafikus felületen keresztül használható.

Ennek a jegyzetnek az a célja, hogy megismertesse az érdeklődő olvasót a számítógépi grafika alapjaival. Azt is szem előtt tartottuk, hogy a leendő olvasók gépészeti tanulmányokat folytatnak, ezért azokat az ismereteket helyeztük előtérbe, amelyek a gépészeti CAD rendszerek értelmes használatához, a rendszerek grafikai szolgáltatásainak megértéséhez szükségesek. A mai gépészeti CAD rendszerek széleskörű, igényes grafikai szolgáltatásokkal rendelkeznek, nem pusztán vonalak rajzolására szorítkoznak. Segítségükkel a termékről fénykép minőségű képeket hozhatunk létre, melyek a megvilágítást, árnyalást, textúrázást, tükröződést és árnyékot is figyelembe veszik. A jegyzet ábráit is vagy teljes egészében CAD rendszerrel készítettük, vagy az ábra készítésének valamelyik fázisában használtunk CAD rendszert - természetesen a fényképek kivételt képeznek.

A jegyzet végén található irodalomjegyzékben szereplő könyvekből, jegyzetekből a téma iránt mélyebben érdeklődők további információhoz juthatnak.

1. fejezet

Grafikai hardver

A számítógépi grafikához használt hardvert lehet az információ áramlásának iránya szerint is osztályozni. Ilyen alapon grafikus input- (bemeneti) és output (kimeneti) eszközöket különböztethetünk meg. Az input-eszközökkel a felhasználó grafikus információt vihet be a számítógépbe, az output-eszközökkel pedig a számítógép jelenít meg grafikus információt. A mai számítógépeknél a grafika fontos szerepet játszik (már az operációs rendszer felhasználói felülete is grafikus), ezért gyakorlatilag minden számítógéphez tartozik grafikus-input és output-eszköz. Korábban ez csak az úgynevezett interaktív grafikus munkahelyekre volt jellemző.

Az alábbiakban ismertetünk néhány eszközt, de ez semmiképp nem tekinthető teljes listának, ilyen irreális célt nem tűztünk magunk elé.

1.1. Output-eszközök

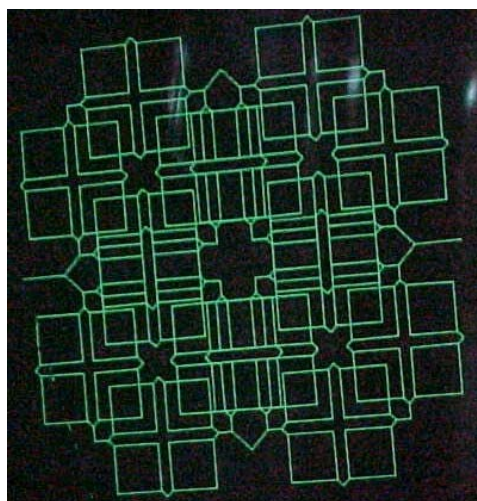
Az output-eszközök fő csoportjai:

- grafikus display,
- plotter,
- hardcopy.

A fenti listában fontossági sorrendben szerepelnek az eszközök, időrendi sorrendben a plotter megelőzi a grafikus display-t.

1.1.1. Grafikus display

A grafikus display alapvető működési elvét tekintve lehet vektoros (vonalarajzoló) vagy raszteres. Vektoros display (1.1. ábra) esetén az elektronsugár közvetlenül rajzol a képernyőre. Ilyen elven működő grafikus display manapság legfeljebb múzeumban található, a raszteres megjelenítők váltak egyeduralkodóvá.



1.1. ábra. Vektorgrafikus monitor és egy rajta megjelenített kép

A raszteres display-k a képet sorokba és oszlopokba rendezett képpontokból (pixelekből – picture element) állítják elő, mint a TV készülékek. Az egyes pixeleken megjelenítendő színt memóriában kell tárolni, tehát a raszteres display-k elterjedéséhez a memória árának csökkenése kellett. Manapság ez a memória legtöbb esetben a videokártyán van, és ideális esetben a mérete olyan, hogy minden pixelhez 64 biten tud információt tárolni. 8+8+8 biten az RGB színtkomponenseket, 8 biten az átlátszóságot, és az animáció (a mozgó kép) érdekében meg kell duplázni ezt a kapacitást.



1.2. ábra. Katódsugárcsöves (CRT) monitor

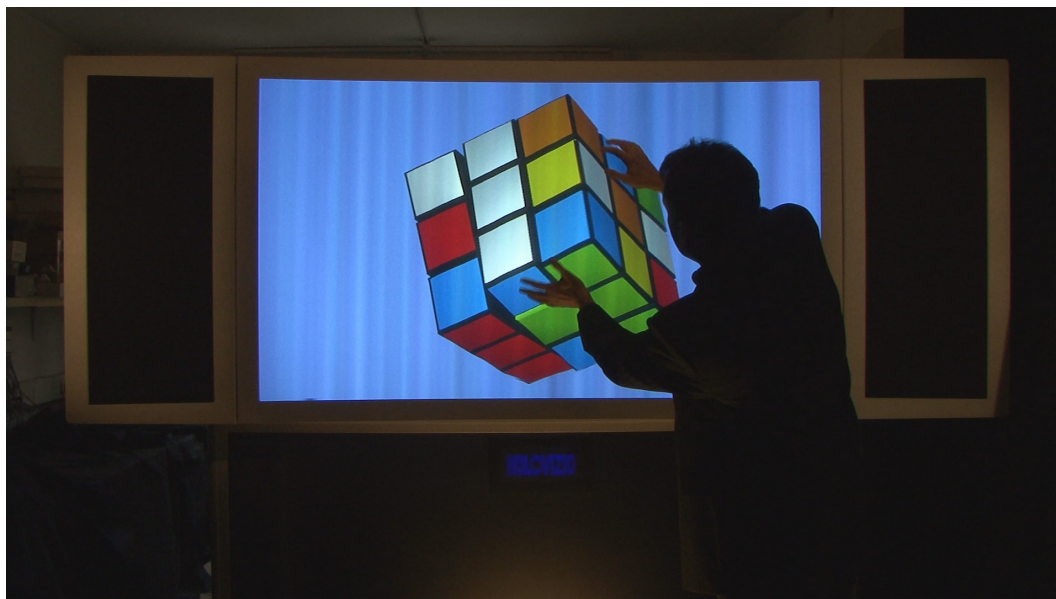


1.3. ábra. Folyadékkristályos (LCD) monitor

A raszteres megjelenítési elv leggyakoribb fizikai megvalósításai:

- katódsugárcső (CRT) (1.2. ábra),
- folyadékkristály (LCD) (1.3. ábra) - passzív mátrix, aktív mátrix (TFT),
- plazma.

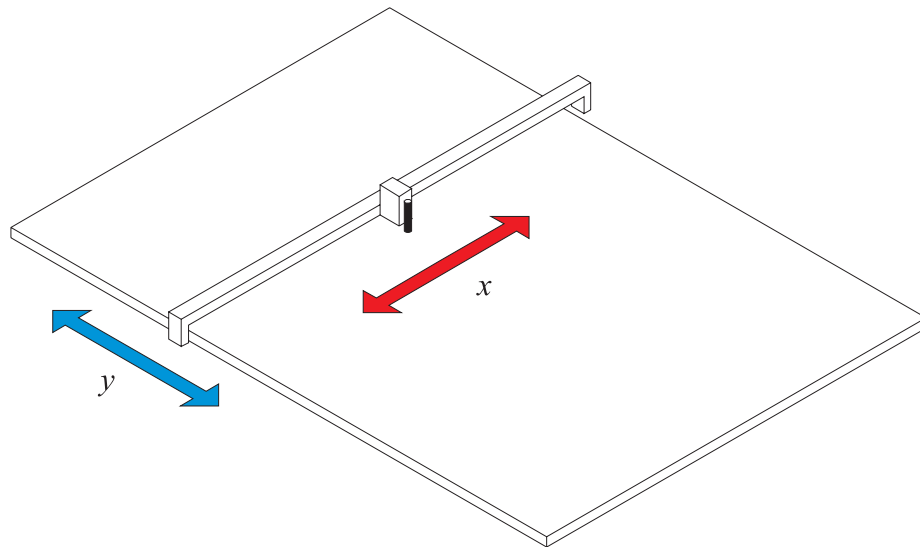
Kezdetben a katódsugárcsöves monitorok voltak az egyeduralkodók, mára azonban megváltozott a helyzet, az eladott monitorok többsége a "lapos" kategóriába tartozik, tehát LCD vagy plazma. Legelterjedtebb a TFT. A változás nem divat kérdése, ésszerű megfontolások állnak mögötte. A CRT monitorokon nagyon szép színes képek jeleníthetők meg, azonban komoly hátrányai vannak ennek a technológiának. A hasznos képterület növelése az eszköz mélységének növelésével jár (nagy hely kell a monitornak), a monitor előtt ülő felhasználót káros sugárzás éri (bár ennek a mértéke az idők során egyre kisebb lett), a kép gyakran torzul (pl. hordósság, trapéz alakú kép).



1.4. ábra. A Holovizio 3D-s megjelenítő (a képet a Holografika Kft. bocsátotta rendelkezésünkre)

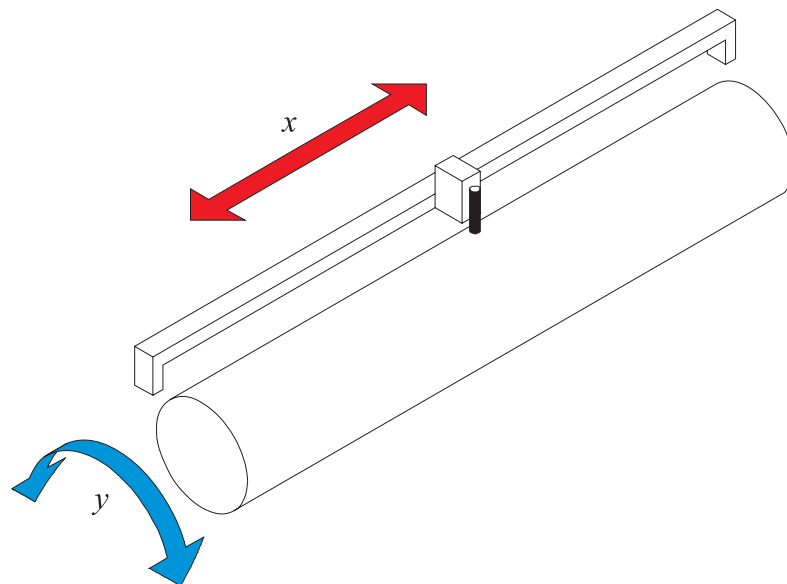
Intenzív kutatások folynak olyan display-k kifejlesztésére, amelyekkel a térben lehet megjeleníteni a számítógépben adathalmazként (számhalmazként) rendelkezésre álló objektumokat. Ilyen 3D-s megjelenítő többféle létezik már, de áruk még viszonylag magas, ezért nem terjedtek még el. Az 1.4. ábrán a holográfia elvén működő 3D-s display látható.

1.1.2. Plotter



1.5. ábra. Síkasztalos plotter sematikus rajza

A plotterek számítógéppel vezérelt rajzoló berendezések, egyfajta CNC (Computer Numerical Control) gépnek tekinthetők (a vezérlő kódjuk is hasonlít a CNC gépekéhez). Felépítésüket tekintve megkülönböztetünk síkasztalos (lásd az 1.5. ábrát) és dobplottert (lásd az 1.6. ábrát). A plotterek rajzolhatnak tollal (lásd az 1.7. ábrát), de működhetnek a tintasugaras printer elvén is (lásd az 1.8. ábrát). Az utóbbi időben a az NC és CNC megmunkáló berendezések elterjedése következtében a plotterek háttérbe szorultak, de nem tűntek teljesen el, mivel nagyméretű rajzok esetén (pl. összeállítási rajz) továbbra is nélkülözhetetlenek.



1.6. ábra. Dobplotter sematikus rajza



1.7. ábra. Tollal rajzoló síkasztalos plotter



1.8. ábra. Tintasugaras dobplotter

1.1.3. Hardcopy eszközök

Hardcopy eszközökön azokat a berendezéseket értjük, melyekkel a képernyőn látható kép papírra, fóliára vagy filmre vihető át. Hardcopy eszközként gyakorlatilag minden mostanság használatos nyomtató igénybe vehető, így pl. a mátrix-, a tintasugaras, a hő-, vagy a lézernyomtató is. Ezen eszközök közös jellemzője, hogy a képet apró pöttyökből (dot-okból) állítják össze. A képpontok sűrűségét a dot/inch értékkel mérjük, vagyis azzal, hogy egy hüvelykre hány képpont kerül. A kép minősége természetesen nő a dot/inch érték növekedésével, de legalább ilyen szerepe van a színezési eljárásnak is. A különböző elven működő nyomtatók különböző technikai megoldással viszik rá a papírra a képpontokat.



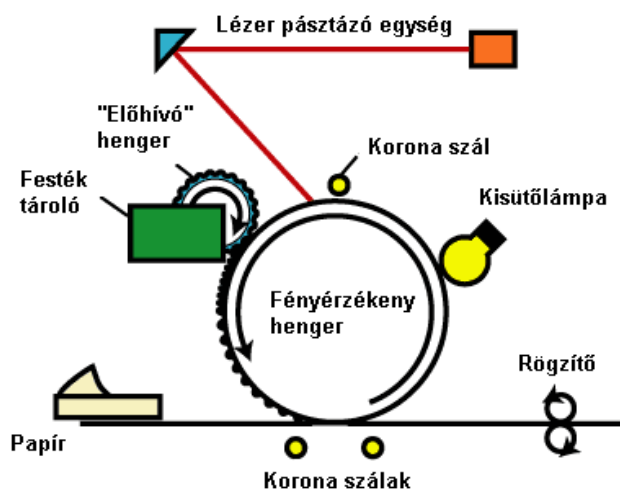
1.9. ábra. Mátrixnyomtató

Mátrixnyomtató (1.9. ábra) esetén apró tűk vannak egy vagy több oszlopba elhelyezve. Ezek előtt festékszalag van, ami előtt a papír helyezkedik el. Ha valamelyik tű ráüt a festékszalagra, a papíron egy képpont keletkezik. A tűk mozgásának vezérlésével érhető el a kívánt hatás. A mátrixnyomtatót manapság már nem használjuk hardcopy eszközként, mert van jobb minőségű képet előállító eszköz is.



1.10. ábra. Tintasugaras nyomtató

Tintasugaras nyomtató (1.10. ábra) esetén a képpontot fúvókákból kilövellő tinta (ink jet) hozza létre. Nagyon szép, fénykép minőségű képek nyomtathatók velük, ehhez azonban speciális papír szükséges (az itatóspapír nem felel meg a célnak).



1.11. ábra. A lézernyomtató működésének sematikus rajza

Lézernyomtató esetén a koronaszál pozitív töltést visz fel a fényérzékeny hengerre. A lézersugár kisüti azokat a pontokat, ahol képpontnak kell lennie. A festék szintén pozitív töltésű, ezért csak a henger kisütött részére tapad rá. A papír az első koronaszáltól erős negatív töltést kap, ezért a festék áttapad a papírra, a második koronaszál a papírt kisüti, hogy az ne ragadjon a dobhoz. A lézernyomtató működésének sematikus rajza látható az 1.11. ábrán. A lézernyomtatókkal rendkívül jó minőségű színes képek hozhatók létre, gazdaságosan üzemeltethetők.

1.2. Input-eszközök

A grafikus input-eszközök nélkülözhetetlenek az interaktív programok használatához, így a CAD rendszerekhez is. A legfontosabb input-eszköz az egér, többféle fizikai megvalósítása van. Kezdetben mechanikus egereket használtak, az ezek aljában elhelyezett gömb mozgásával összhangban mozgott a kurzor. Manapság optikai egereket használunk, ezek nagy előnye, hogy nincs benne mozgó alkatrész, így kisebb a meghibásodás esélye.

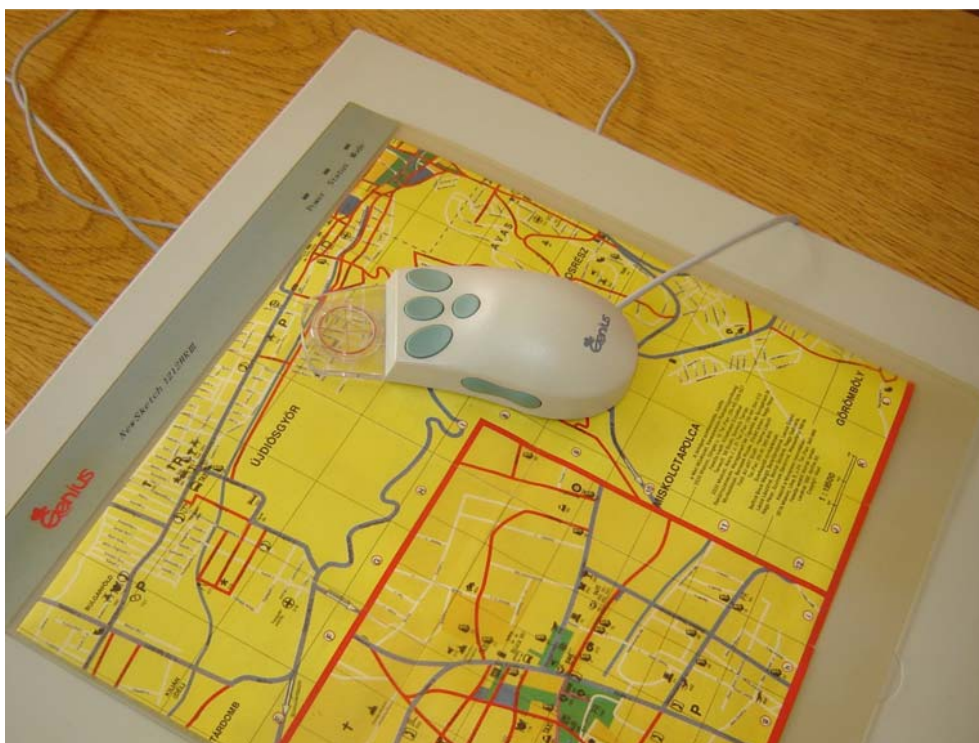
Minden számítógéphez tartozik billentyűzet, ami szintén alkalmas grafikus információ bevitelére, legfeljebb nem kényelmes (bár valószínűleg sok számítógépes-játék függő nem ért ezzel egyet).



1.12. ábra. Tablet

Elterjedt input eszközök a tabletek és a digitalizálók is. Ezek tulajdonképpen csak méretükben és felhasználási területüket illetően különböznek, működési elvük azonos. Segítségükkel egy síklapra helyezett rajzról koordináta-információt lehet bevinni a számítógépbe. Tablet esetén a síklap mérete általában A4 vagy A3, digitalizáló esetén ez akár A0 is lehet. A hely megjelölésére szolgáló eszköz lehet toll, amely végén mikrokapcsoló van, vagy lehet nagyító szállkeresztrel. A tollal viszonylag pontatlanul lehet a helyet megjelölni, ezért ez leginkább tablettal összekötve (1.12. ábra) menüből való választásra alkalmas, pl. CAD rendszerek használatakor. A szállkereszt sokkal pontosabb lokalizálást tesz lehetővé (1.13. ábra), ezért ezt a nagy pontosságot igénylő alkalmazásoknál, mint pl. hagyományos módon készített térképek, rajzok digitalizálására használjuk.

A lapbeolvasóval (scanner) képeket raszterizálhatunk, azaz sorokba és oszlopokba rendezett képpontokká alakíthatjuk át az analóg képet, természetesen megfelelő szoftver alkalmazásával. Az 1.14. ábrán lapbeolvasó képét láthatjuk.



1.13. ábra. Digitalizáló



1.14. ábra. Lapbeolvasó (scanner)



1.15. ábra. Botkormány

A botkormány (1.15. ábra) fő, de nem kizárólagos felhasználási területe a játékprogramok vezérlése.



1.16. ábra. Pozicionáló gömb

Kevésbé elterjedt, de néhány CAD rendszer esetén használatos a pozicionáló gömb (tracking ball) is (1.16. ábra). Ez gyakorlatilag egy hátára fektetett mechanikus egér, vagyis egy gömböt forgatva mozgathatjuk a grafikus kurzort.



1.17. ábra. Fénytoll

A fénytoll (light pen) (1.17. ábra) a vonalrajzoló (vektorgrafikus) megjelenítők legfontosabb input-eszköze volt, de raszteres megjelenítőkhez is kapható. Ez gyakorlatilag olyan ceruza, amelynek a végében fényérzékelő van, aminek segítségével a rámutatással kiválasztott pixel koordinátáját és színét kaphatjuk meg. Ezzel az eszközzel ritkán találkozhatunk.



1.18. ábra. A monitor elé szerelhető érintő képernyő és tartozékai

Az érintő képernyő (touch panel) (1.18. ábra) eléggé elterjedt. Információs pultoknál kiválóan alkalmas navigálásra, menüből való választásra a kiválasztott képernyőterület megérintésével.



1.19. ábra. Space ball (balra) és space mouse (jobbra)

Az eddigi eszközökkel gyakorlatilag csak síkbeli koordinátákat lehetett bevinni. Azonban nagyon sok alkalmazásnál, pl. tervezőrendszerek, de a játékprogramok esetén is szükség van térbeli koordináták, irányok bevitelére. Erre alkalmas eszköz a space ball (szokták 3D-s egérnek is nevezni) (1.19. ábra), és az adatkesztyű (data glove).



1.20. ábra. Tervezői környezet kamerákkal és a kézre tapasztott fényvisszaverő foltokkal

Napjainkban intenzív kutatás folyik, hogy a speciális kesztyűt kiváltsák kamerákkal és a kézre tapasztott fényvisszaverő foltokkal (1.20. ábra).



1.21. ábra. VR sisak

A CAD-ben is egyre inkább teret hódító virtuális valósághoz pedig az adatkesztyű és a VR sisak (1.21. ábra) (Virtual Reality Helmet) eszközpáros használatos. Az utóbbit játékprogramokhoz már régóta használják.



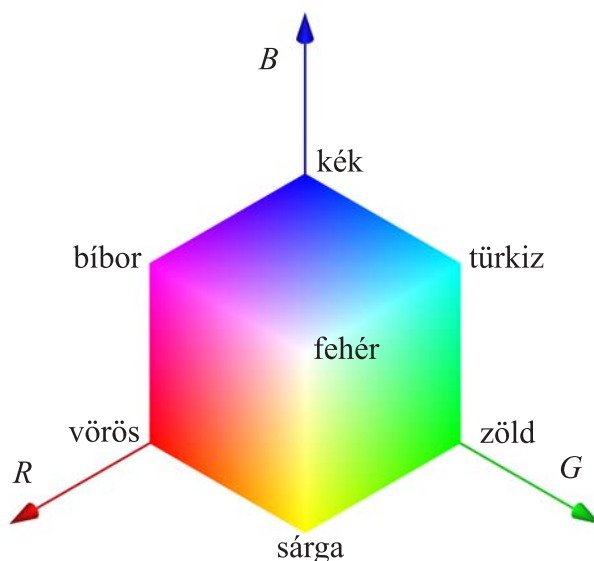
1.22. ábra. Kézmozdulatokkal való tervezés adatkesztyű és 3D-s szemüveg használatával

A viszonylag nagyméretű sisak kiváltására speciális szemüvegeket fejlesztettek ki (1.22. ábra).

2. fejezet

Színek a számítógépi grafikában

A színeket legcélszerűbb egy háromdimenziós tér pontjaiként felfogni. A tér egy bázisának (koordináta-rendszerének) rögzítése után minden színt egy számhármassal lehet azonosítani. A számítógépi grafikában két színelőállítási (színkeverési) mód, azaz kétféle bázis terjedt el. Az egyik a színes grafikus megjelenítőknél használt additív (hozzáadó) színkeverés, a másik a színes nyomtatóknál használt szubtraktív (kivonó).

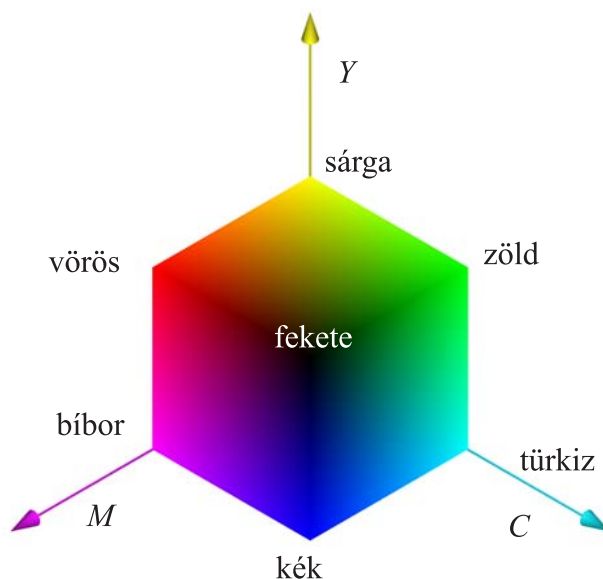


2.1. ábra. Az additív színkeverés (RGB) koordináta-rendszere

Additív színkeverés esetén azt írjuk elő, hogy mennyi vörös (Red), zöld (Green) és kék (Blue) színkomponenst kell adni a fekete színhez, a fekete képernyőhöz. A színteret az egységkockával szemléltetjük, mivel a színkomponensek értéke általában a $[0., 1.]$ intervallumon változtatható a különböző grafikus rendszerekben. Ezt a színelőállítási módot RGB színrendszernek szokás nevezni. A $(0., 0., 0.)$ számhármassal a fekete, az $(1., 1., 1.)$ a fehér színnek, a legösszetettebb színnek felel meg (lásd a 2.1. ábrát). A színes raszteres grafikus megjelenítők működése ezzel összhangban van, hiszen a képernyő minden egyes pixele gyakorlatilag három részből áll, melyek különböző intenzitású vörös, zöld illetve kék színt tudnak kibocsátani. Az RGB színmegadás az emberi szem működésével is összhangban van. Az emberi szem retinájában a világosságlátást lehetővé tevő érzékelő sejtek, a pálcikák, továbbá a színlátást lehetővé

tevő érzékelő sejtek, a csapok találhatók. A csapok három fajtája különböztethető meg aszerint, hogy milyen hullámhosszú fény ingerli őket leginkább. Ez a három hullámhossz éppen a vörös, zöld és kék színé, vagyis leegyszerűsítve azt mondhatjuk, hogy az emberi szemben RGB érzékelők vannak, és a különböző arányban érzékelt színek komponensek összegzése útján áll elő az általunk látott szín.

Szubtraktív színkeverés esetén azt írjuk elő, hogy mennyi türkiz (Cyan), bíbor (Magenta) és sárga (Yellow) színt kell kivonnunk a legösszetettebb színből, a fehérből (a papír színe) a kívánt szín előállítására érdekében (lásd a 2.2. ábrát). Ezt röviden CMY színrendszernek nevezzük.



2.2. ábra. A szubtraktív színkeverés (CMY) koordináta-rendszere

A grafikai hardver által használt színkeverésen kívül más módok is lehetségesek, más bázis is felvehető a színtérben. Ilyen például a képzőművészek által használt színmeghatározás, mely a színárnyalat (Hue), fényesség (Lightness) és telítettség (Saturation) összetevőkon alapul. Ez a HLS színkeverési mód.

A számítógépek grafikus alaprendszereiben a színek megadása általában kétféleképpen történhet. Az egyik színmegadási mód esetén közvetlenül megadjuk a kívánt szín RGB komponenseit, a másik esetén csupán egy indexszel hivatkozunk egy színtáblázatra (színpaletta) valamely elemére. Természetesen a paletta színei a felhasználó által megváltoztathatók, vagyis mi dönthetjük el, hogy az egyes színindexekhez milyen szín tartozzon és ezt bármikor megváltoztathatjuk.

Több rendszerben az RGB komponensek mellett egy negyedik, úgynevezett alfa (A) komponenst is megadhatunk, ezért ilyen esetben RGBA módról beszélünk. Az alfa komponens természetesen nem a szín megadásához kell, ahhoz pontosan három komponens szükséges, vele az átlátszóságot adhatjuk meg, vagyis azt kontrollálhatjuk, hogy az adott színnel kifestendő pixelen csak az új szín látszon, vagy a pixel régi és új színéből együttesen álljon elő a pixel végső színe. Alfa 1 értéke esetén a szín nem átlátszó, 0 esetén teljesen átlátszó, a közbülső értékek esetén pedig a pixel régi és új színét keveri a rendszer az előírt arányban.

A színes grafikus hardver egyrészt felbontásban – a képernyő pixeleinek számában –, másrészt a pixeleken megjeleníthető színek számában különbözik. E kettőt a monitor és a grafikus

kártya minősége, és azok összhangja határozza meg. Jó minőségű, valószerű képek előállításához nem elegendő a felbontást növelni, ugyanis viszonylag alacsony felbontás mellett nagyszámú szín esetén szinte tökéletes képet hozhatunk létre. Gondoljunk pl. a TV készülékekre, melyek felbontása elég kicsi (a PAL rendszerű készülékek 625 soron jelenítik meg a képet) a mai monitorok felbontásához képest, de a megjeleníthető színek száma – ezt szokás színmélységnek is nevezni – gyakorlatilag korlátlan.

A számítógépekben, pontosabban azok grafikus kártyáján, a képernyő minden egyes pixeléhez ugyanannyi memória tartozik, melyben a pixelek színét tárolja a gép. A pixelek színének tárolására használt memóriát színpuffernek nevezzük. A színpuffer méretét azzal jellemezzük, hogy pixelenként hány bit áll rendelkezésre a színek tárolására. 8 bit/pixel esetén a megjeleníthető színek száma $2^8 = 256$. A manapság használatos grafikus kártyák sokkal több színt tudnak megjeleníteni, a 24 bit/pixel szabványossá vált. A legjobbnak mondható kártyák esetén ez az érték 32 bit/pixel (az RGBA összetevők mindegyikére 8 bit, vagyis 16.77 millió szín plusz az átlátszóság, az ilyen grafikus kártyákat szokás true colornak is nevezni) és akkora a memória, hogy ilyen színpufferből két darabot tud használni (erre például animáció esetén van szükségünk), tehát ezt felfoghatjuk 64 bit/pixelként is. Vannak olyan grafikus kártyák is, amelyek három színpuffer használatát is támogatják.

3. fejezet

Néhány grafikai algoritmus

Az 1. fejezetben láthattuk, hogy nagyon sokféle grafikai hardver van. Ezek az eszközök alap-tudásukat illetően igen egyszerűek, pl. egy raszteres megjelenítőnek az egyes pixelek színét lehet megadni a pixelekhez tartozó memóriaterület felülírásával. Ahhoz, hogy a felhasználók kényelmesen tudják használni ezeket az eszközöket, rengeteg alapfeladatot kell megoldani nagyon hatékonyan. Ilyen alapfeladat pl. az egyenes szakasz rajzolása raszteres megjelenítőn. Az ilyen jellegű feladatok megoldása a grafikus alapszoftverek feladata, sőt manapság már a grafikus kártyák szintjén oldják meg őket, tehát egy átlagos felhasználó nem találkozik velük.

Mi itt két alapfeladatra mutatunk megoldást. Az egyik a szakasz rajzolása raszteres megjelenítőn, a másik a képmezőre való vágás.

3.1. Szakasz rajzolása raszteres megjelenítőn

A szakasz képétől azt várjuk el, hogy a képernyő egy téglalap alakú területe legyen, sima határokkal. Ezt a hatást a pixelekből álló képen nem egyszerű megvalósítani, hiszen a határ pl. mindenképpen lépcsős lesz, miután kis téglalapokból rakjuk össze a szakasz képét.

Azt feltételezzük, hogy a szakasz végpontjait a képmezőkoordináta-rendszerben (pixeleken) ismerjük, vagyis a koordináták egész számok. A továbbiakban a kezdőpontra az (x_0, y_0) , a végpontra az (x_1, y_1) koordinátapárral hivatkozunk.

Nagyon sok algoritmus létezik szakasz rajzolásra, mi a DDA (Digital Differential Analyzer) algoritmust ismertetjük. Ez esetszétválasztással kezdődik. Vízszintes és függőleges helyzetű szakaszok esetén ugyanis nincs probléma, a kezdőpontból kiindulva a végpont felé haladva kiszínezzük a soron következő pixelt. Egyéb esetben tekintjük az egyenes

$$y = mx + b$$

alakját, ahol m az egyenes iránytangense, azaz

$$m = \frac{y_1 - y_0}{x_1 - x_0}$$

(m mindenképpen lebegőpontos szám, nem egész!), b pedig az y tengellyel alkotott metszéspont koordinátája (ez biztosan létezik, mivel az y tengellyel párhuzamos szakaszokat már leválasztottuk). Feltételezzük, hogy $x_0 < x_1$. (Ha nem így lenne, a kezdő- és végpont felcserélésével ezt elérhetjük.)

Ha az (x_i, y_i) koordinátájú pixel a szakaszra illeszkedik, azaz

$$y_i = mx_i + b,$$

akkor a szakaszon x irányban Δx távolságra lévő pixel (x_{i+1}, y_{i+1}) koordinátáira

$$x_{i+1} = x_i + \Delta x$$

$$y_{i+1} = mx_{i+1} + b = m(x_i + \Delta x) + b = y_i + m\Delta x$$

teljesül. Ha $\Delta x = 1$, akkor

$$y_{i+1} = y_i + m.$$

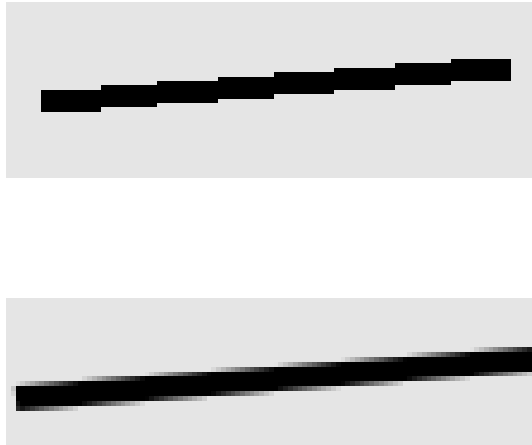
Feltételezve, hogy $x_0 < x_1$ és $|m| \leq 1$, az algoritmust megvalósító C nyelvű függvény a következő:

```
szakasz(int x0,int y0,int x1,int y1)
{
    int x;
    float dx,dy,y,m;
    dx = x1 - x0;
    dy = y1 - y0;
    m = dy / dx;
    for(x = x0,y = y0;x <= x1;x++)
    {
        display_pixel(x, Round(y));
        y += m;
    }
}
```

A `display_pixel()` függvény az adott koordinátájú pixelt kirajzolja, a `Round()` függvény pedig a valós számot egészre kerekíti.

Mint már említettük, a raszteres megjelenítőkön a nem függőleges vagy vízszintes szakaszok képeinek határa töredezett, lépcsős lesz. Kiváltképp így van ez a majdnem vízszintes és majdnem függőleges szakaszok esetén. Ezt a jelenséget angolul aliasingnak, a jelenség csökkentésére, a kép határának kisimítására irányuló technikákat antialiasingnak nevezik. A hardver működési elve miatt, ez a lépcsősség fizikailag nem tüntethető el, de a sima határ látványa elérhető. Egyik lehetőség a pixelek méretének csökkentése lehetne, ennek azonban fizikai határai vannak. Másik lehetőség lehetne, hogy a képet távolabbról nézzük, ennek azonban ésszerűségi határai vannak. Ezek helyett vizuális "trükkhöz" folyamodunk, a sima határ illúziójának keltése érdekében.

Elterjedt technika a pixelek fedettségének számításán alapuló eljárás. A szakaszok képe minden esetben téglalap lesz. A rendszer kiszámítja ezen téglalapot metsző pixelekre, hogy hányad részük esik a téglalap képére. Ezt a számot nevezzük a pixel fedettségének. A szakasz képét metsző pixelek színét a szakasz és a háttér színének, a pixelek fedettségének szerinti vegyítésével határozzuk meg. A kép határának kisimítása tehát azt jelenti, hogy az alaphelyzethez képest több pixelt fest ki a rendszer, de ezeket már nem a szakasz színével, hanem a fedettségtől függően a szakasz és a háttér színének kombinációjával. Vagyis a szakasz és a háttér között fokozatos átmenet lesz, nem pedig ugrásszerű. A 3.1. ábrán egy szakasz simítás nélküli (felső kép) és simított (alsó kép) megjelenítését láthatjuk. Mindkét esetben jelentősen felnagyítottuk a képernyőről levett képet.

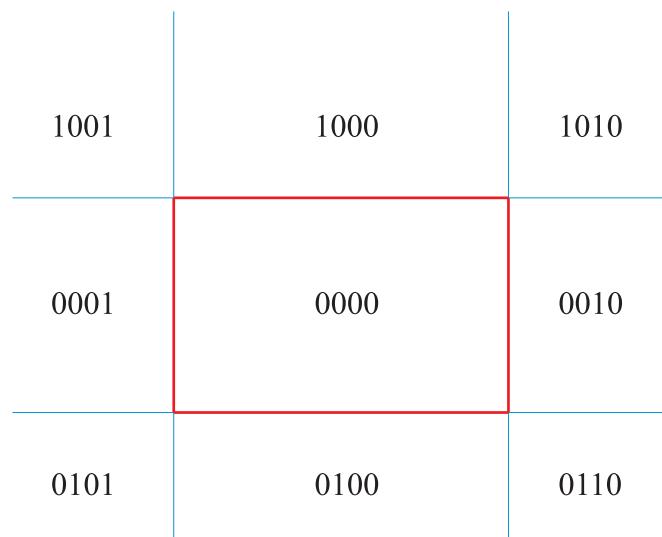


3.1. ábra. Szakaszok határának kisimítása

3.2. Vágás a képmező határára

A síkgeometriában többnyire a végtelen síkot tekintjük kiindulási alapként, ezen helyezzük el alakzatainkat. Amikor azonban szemléltetni kell ezeket akár papírra rajzolva, akár a számítógép monitorán megjelenítve, a végtelen síknak csak igen korlátozott részét használhatjuk. A számítógépi grafikában ez a korlátozott terület a képernyőnek, a képernyő oldalaival párhuzamos oldalú téglalappal határolt területe. Ezt a téglalap alakú területet képmezőnek nevezzük.

Az alakzatok képe tehát csak ezen a téglalapon belül lehet, aminek az a következménye, hogy a képmezőből kilógó alakzatrészeket el kell távolítani. Ezt a műveletet nevezzük vágásnak (clipping). A pont, szakasz és poligon grafikus képelemek vágására mindenképpen kell megoldást biztosítani.

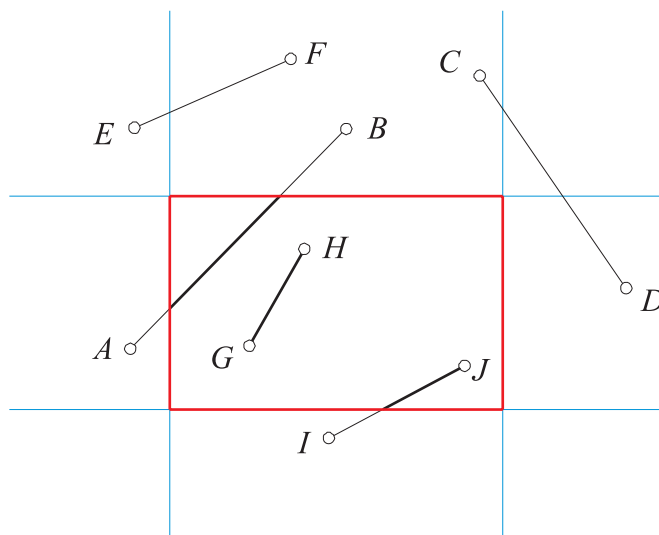


3.2. ábra. A tartományok kódolása a Cohen-Sutherland szakaszvágó algoritmusban

A pont vágásával nincs sok gond, koordináták összehasonlításával könnyű eldönteni, hogy a vizsgált pont a képmezőn belül van-e.

Szakasz vágása esetén felmerülhet persze, hogy nincs is szükség ezzel külön foglalkozni, hiszen a szakasz képe úgyis pixelekből áll, tehát állítsuk elő a szakasz képére eső pixeleket, és azokról könnyen el tudjuk dönteni, hogy a képmezőbe esnek-e. Azonban csekély gondolkodás után rájöhettünk, hogy ez nagyon pazarló (nem hatékony) megoldás, hiszen olyan szakaszokra eső pixeleket is ki kell számolni, mely szakaszokra eső egyetlen pixel sincs a képmezőben. A vágási eljárás akkor hatékony, ha gyorsan kiderül, hogy a szakasz egyetlen pontja sem látszik, és az is, ha a teljes szakasz látszik. Szakaszok vágására több eljárás is létezik, mi itt az egyik legelterjedtebbet, a Cohen-Sutherland algoritmust ismertetjük.

A Cohen-Sutherland algoritmus a képsíkot a képmező oldalegyeneseivel 9 tartományra osztja. Az egyes tartományokhoz kódot rendel, aminek tárolásához tehát 4 bit elegendő. A képmező kódja 0000, a többi tartományhoz úgy rendel kódot, hogy minden oldalegyenest hozzárendel egy bithoz, és azt a bitet 0-ra állítja, ha a tartomány az oldalegyenesnek a képmező felé eső oldalán van, egyébként 1-re állítja (lásd a 3.2. ábrát). Az algoritmus lépései a következők:



3.3. ábra. Különböző helyzetű szakaszok

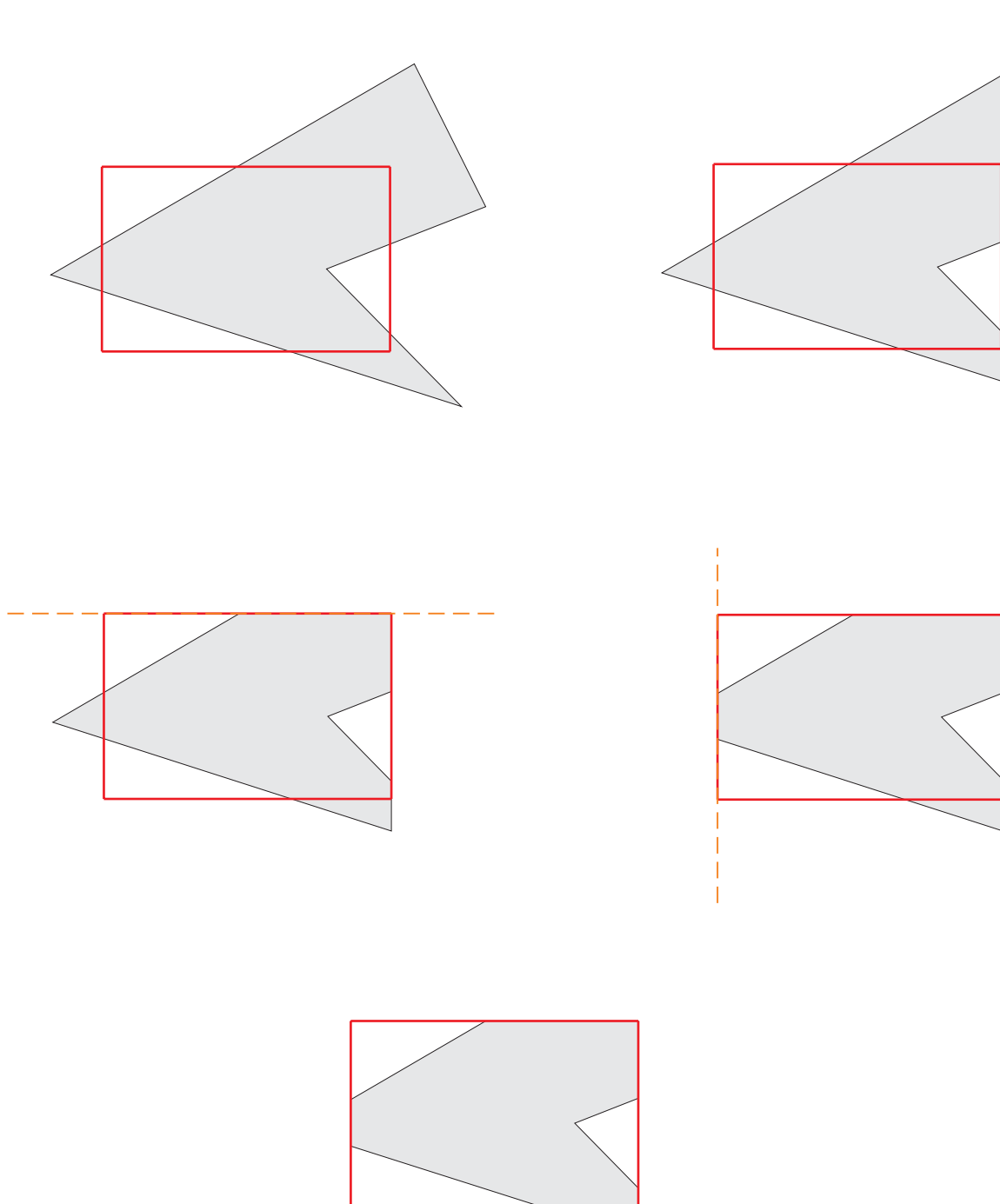
1. A vizsgált szakasz kezdő- és végpontjának meghatározzuk a kódját, és ezeket az a és b változókba tesszük.
2. Ha $a \mid b = 0$ (\mid a bitenkénti "vagy" műveletet jelöli), akkor a szakasz vágás nélkül megrajzolható.
3. Ha $a \& b \neq 0$ ($\&$ a bitenkénti "és" műveletet jelöli), akkor a szakasz elhagyható, mert egyetlen pontja sem látszik, mivel a kezdő- és végpontja is valamelyik oldalegyenesnek a képmezővel ellentétes oldalán van.
4. Megkeressük az első olyan oldalegyenest, amelyik metszi a szakaszt, a metszésponttal helyettesítjük a megfelelő végpontot, az új végpontnak meghatározzuk a kódját és az eljárást a 2. lépéstől újakezdjük.

A fenti algoritmussal legfeljebb két vágás után a 2. vagy 3. lépésbeli feltétel teljesül. Két vágás a 3.3. ábra szerinti AB és CD szakaszok esetén szükséges. Előfordulhat, hogy csak a

második vágás után derül ki a szakaszcól (az ábrán a CD szakasz), hogy egyetlen pontja sem látszik. A GH szakaszcól azonnal kiderül, hogy nem kell vágni, az EF -ről pedig az, hogy elhagyható. Az IJ szakaszt egyszer kell vágni.

A poligonok vágását nem lehet leegyszerűsíteni a határoló szakaszok vágására. A poligon ugyanis zárt síkrészt képvisel, belseje is van, tehát a vágás után is zárt síkidomot kell kapnunk. (Egyébként nem tudnánk a vágás után a belsejét kifesteni, szétfolyna a festék.)

A Sutherland-Hodgman poligonvágó algoritmus végigmegy a képező oldalain, és az oldalegyenesekkel elvágja a az ábrázolandó poligont úgy, hogy minden egyes vágás után lezárja a csonkolt poligont az oldalegyenes megfelelő szakaszával (szakaszaival). Ezt illusztrálja a 3.4. ábra.



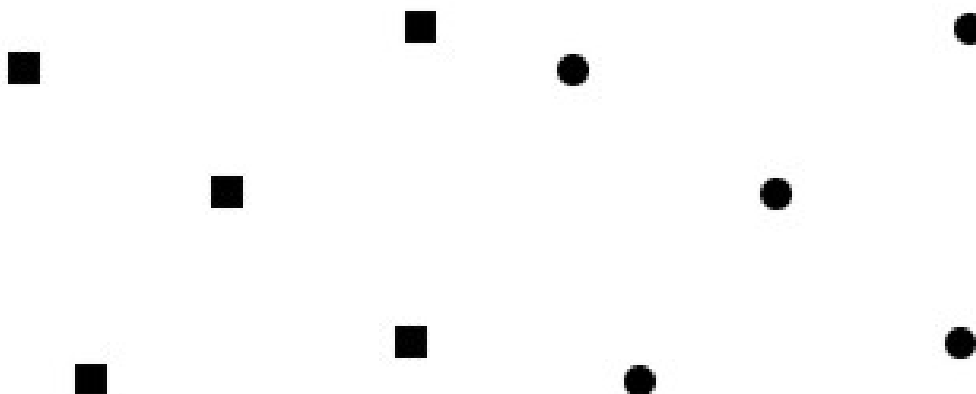
3.4. ábra. A poligonvágás lépései a Sutherland-Hodgman algoritmussal (balról jobbra és felülről lefelé haladunk)

4. fejezet

Grafikus alapelemek

Napjaink grafikus rendszereiben alapvetően háromféle képelemet (építőkövet) használhatunk. Ezek a pont, a szakasz és a poligon. Bármely megjelenítendő alakzat képét, legyen az akármi-lyen bonyolult is, ezekből az egyszerű elemekből kell felépíteni.

A pont alapelem legtöbbször egy négyzetes pixeltömböt jelent ($n \times n$ pixel, $n \geq 1$), de némely grafikus rendszer megenged ettől eltérő alakot is, többnyire körlapot (4.1. ábra). Termé-zetesen a körlapok is a téglalap alakú pixelekből épülnek fel. A pont alapelem színét, méretét, esetleg alakját adhatjuk meg.



4.1. ábra. Pont képelemek

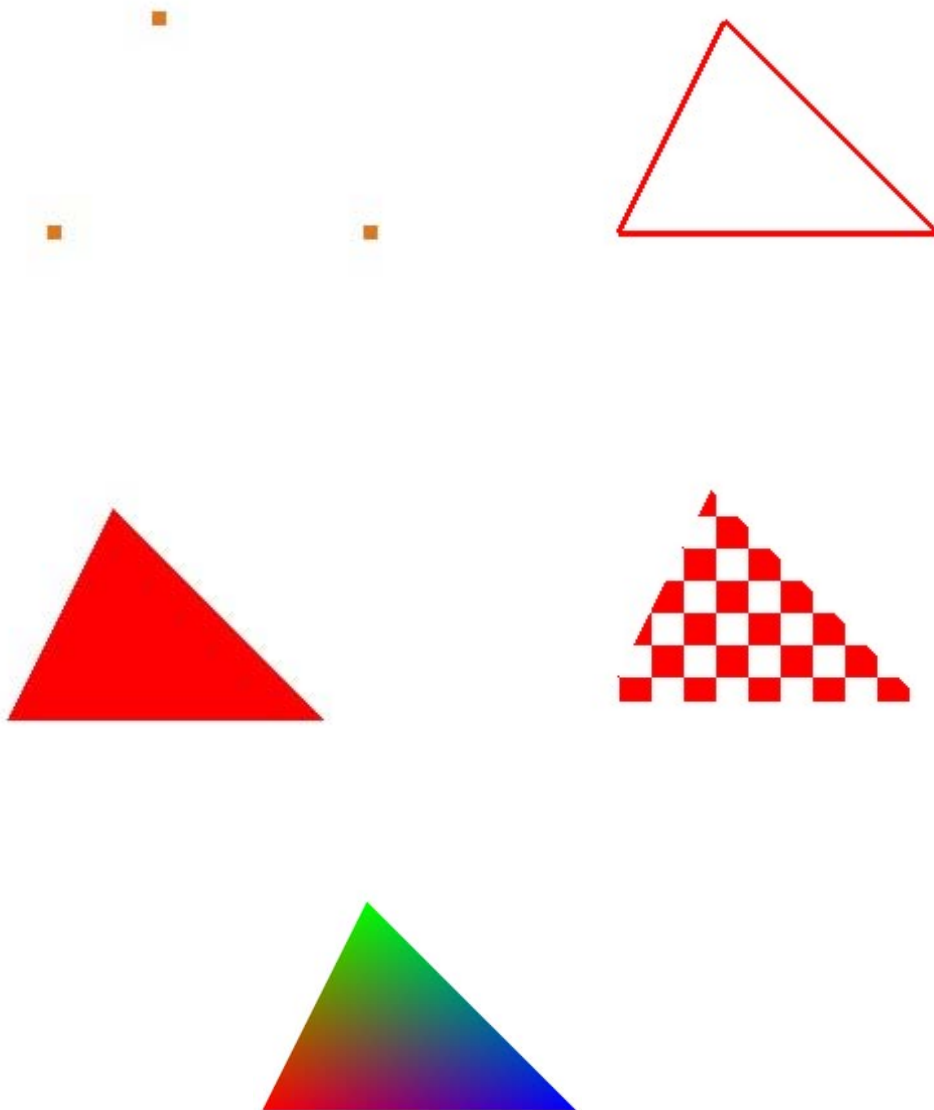
A szakasz alapelem, és a belőle származtatott töröttvonal sok célra használható. Töröttvonal alatt adott pontokat a megadás sorrendjében szakaszokkal összekötő alakzatot értünk, úgy mint ahogy az a geometriában megszokott. A görbéket is töröttvonallal közelítjük, erről a 8. feje-zetben lesz részletesebben szó. A szakasz megjelenését befolyásoló megadható tulajdonságok: szín, vonalvastagság és vonaltípus (pl. folytonos, szaggatott, pontvonal). Árnyalás használata esetén a szakasz végpontjaihoz különböző szín is rendelhető, és a grafikus rendszer a szakasz belső pontjaiban ezek interpolálásával határozza meg a színt.

A számítógépi grafikában poligon alatt szakaszokkal határolt zárt síkrészt értünk. A poli-gont tehát zárt töröttvonal határolja, azonban ez nem jelenti azt, hogy helyettesíthető lenne vele.

A töröttvonal és a poligon között a legfontosabb különbség az, hogy a poligonnak nemcsak határa hanem belseje is van.

A poligon alapelem megjelenését befolyásoló tulajdonságok az alábbiak:

- rendelkezhetünk arról, hogy a határa megjelenjen-e;
- előírhatjuk, hogy a belseje hogyan jelenjen meg
 - ne jelenjen meg,
 - adott színnel legyen kitöltve,
 - a csúcspontokhoz rendelt szín alapján legyen árnyalva (ha van lehetőség árnyalásra),
 - mintával legyen kitöltve.



4.2. ábra. Poligon különböző megjelenítése

Egymáshoz kapcsolódó poligonokból poligonhálót hozhatunk létre. A síklapok által határolt testek, a poliéderek (pl. kocka, gúla, hasáb) felülete poligonhálóval pontosan leírható, tehát ezen testek ábrázolásakor poligonhálót kell megjeleníteni. A nem síklapokkal határolt testeket, felületeket (pl. gömb, henger) többnyire poligonhálóval közelítjük és ezt a közelítő poligonhálót jelenítjük meg. Ilyen esetekben legtöbbször háromszögekből álló poligonhálót alkalmazunk. Görbült felületek ábrázolásával a 8. fejezetben foglalkozunk részletesebben.

A poligonoknak, valamint a belőlük létrehozott poligonhálónak megkülönböztethetjük az oldalait, és a különböző oldalakhoz különböző megjelenési tulajdonságot rendelhetünk. (4.3. ábra)



4.3. ábra. A poligonok két oldala különböző színű lehet

5. fejezet

Koordináta- és ponttranszformációk

A grafikai és geometriai problémák megoldásához az objektumokat koordináta-rendszerben kell leírni, leggyakrabban a derékszögű Descartes-féle koordináta-rendszereket használjuk. Általánosnak tekinthető az a helyzet, hogy az elemi alakzatok saját koordináta-rendszerükben vannak leírva (amiben könnyű őket leírni), és ezekhez az elemi alakzatokat közös térbe elhelyezve hozunk létre új, összetett alakzatokat. Ez tehát azt jelenti, hogy koordináta-rendszerek közötti áttéréseket és az alakzatok transzformációját is le kell tudnunk írni.

A transzformációkat mátrixokkal írjuk le, amihez úgynevezett homogén koordinátákat használunk. A homogén koordináták tárgyalására nem térünk ki részletesen, csak formálisan vezetjük be őket. A homogén koordináták nem kerülhetők meg, hiszen a mai grafikus rendszerek (pl. az OpenGL) is ezeket használják.

A Descartes-féle koordinátákat egy újabb komponenssel egészítjük ki. A helyvektorokat az 1 komponenssel, az irányvektorokat pedig a 0 komponenssel. Az így kapott koordinátákat nevezzük homogén koordinátáknak. A homogén koordináták egy fontos tulajdonsága, hogy arányosság erejéig meghatározottak, ami azt jelenti, hogy pl. a pont koordinátáit egy nullától különböző számmal megszorozva, ugyanannak a pontnak a koordinátáit kapjuk. A síkbeli homogén koordinátáknak tehát három komponense van, és mindig elérhető az, hogy az utolsó komponens 1 vagy 0 legyen. Az utóbbi esetben a koordináták egy irányt (végtelen távoli pontot) határoznak meg.

A homogén koordináták alkalmazása teszi lehetővé, hogy a síkbeli pont- és koordinátatranszformációkat egyetlen 3×3 -as, a térbelieket pedig egyetlen 4×4 -es mátrixszal le tudjuk írni.

5.1. Koordinátatranszformációk

Grafikai, geometriai feladatok során gyakran van szükség arra, hogy valamely alakzatot egy új koordináta-rendszerben írjunk le, azaz koordináta-rendszer transzformációt hajtsunk végre.

5.1.1. Koordinátatranszformáció a síkon

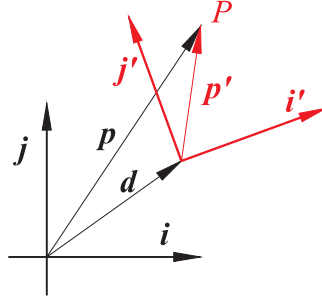
Adottak a síkon az i, j és i', j' egységvektorú derékszögű Descartes-féle koordináta-rendszerek. Ismerjük a P pontnak az i', j' új koordináta-rendszerbeli p' helyvektorát. Határozzuk meg a P pont i, j régi koordináta-rendszerbeli p helyvektorát (lásd az 5.1. ábrát).

$$p = d + p' = x'i' + y'j' + d,$$

amit mátrixokkal

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{i}' & \mathbf{j}' & \mathbf{d} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} i'_x & j'_x & d_x \\ i'_y & j'_y & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

alakban írhatunk fel, a $\mathbf{p} = (x, y)$, $\mathbf{p}' = (x', y')$ jelölést használva, ahol (x', y') az új rendszerbeli komponensek, (i'_x, i'_y) az \mathbf{i}' , (j'_x, j'_y) pedig a \mathbf{j}' vektor komponensei a régi koordináta-rendszerben.



5.1. ábra. Síkbeli koordinátatranszformáció

Tekintsük az előző feladat inverzét!

Adottak a síkon az \mathbf{i}, \mathbf{j} és \mathbf{i}', \mathbf{j}' egységvektorú derékszögű Descartes-féle koordináta-rendszerek. Ismerjük továbbá a P pontnak az \mathbf{i}, \mathbf{j} koordináta-rendszerbeli \mathbf{p} helyvektorát. Határozzuk meg a P pont \mathbf{i}', \mathbf{j}' új koordináta-rendszerbeli \mathbf{p}' helyvektorát (lásd az 5.1. ábrát).

A régi koordináta-rendszerben a $\mathbf{p}' = \mathbf{p} - \mathbf{d}$ teljesül. \mathbf{p}' -nek az új koordináta-rendszerbeli komponenseit a $\mathbf{p}' \cdot \mathbf{i}'$, illetve a $\mathbf{p}' \cdot \mathbf{j}'$ skaláris szorzatokkal kapjuk meg. Ezek felhasználásával az

$$\begin{aligned} x' &= \mathbf{p}' \cdot \mathbf{i}' = (\mathbf{p} - \mathbf{d}) \cdot \mathbf{i}' = \mathbf{p} \cdot \mathbf{i}' - \mathbf{d} \cdot \mathbf{i}', \\ y' &= \mathbf{p}' \cdot \mathbf{j}' = (\mathbf{p} - \mathbf{d}) \cdot \mathbf{j}' = \mathbf{p} \cdot \mathbf{j}' - \mathbf{d} \cdot \mathbf{j}' \end{aligned}$$

kifejezéseket kapjuk. Ennek mátrix alakja

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} i'_x & i'_y & -\mathbf{d} \cdot \mathbf{i}' \\ j'_x & j'_y & -\mathbf{d} \cdot \mathbf{j}' \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Az inverz transzformáció mátrixa tehát az eredeti transzformáció mátrixának az inverze.

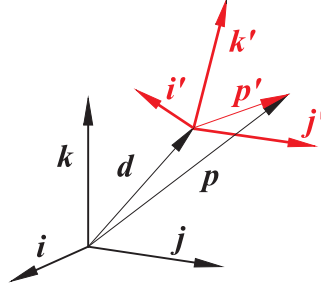
5.1.2. Koordinátatranszformáció a térben

A térbeli koordinátatranszformáció analóg a síkbelivel. Adottak a térben az $\mathbf{i}, \mathbf{j}, \mathbf{k}$ és $\mathbf{i}', \mathbf{j}', \mathbf{k}'$ egységvektorú derékszögű Descartes-féle koordináta-rendszerek. Ismerjük a P pontnak az $\mathbf{i}', \mathbf{j}', \mathbf{k}'$ új koordináta-rendszerbeli $\mathbf{p}' = (x', y', z')$ helyvektorát. Határozzuk meg a P pont $\mathbf{i}, \mathbf{j}, \mathbf{k}$ régi koordináta-rendszerbeli $\mathbf{p} = (x, y, z)$ helyvektorát (lásd az 5.2. ábrát).

$$\mathbf{p} = \mathbf{d} + \mathbf{p}' = x'\mathbf{i}' + y'\mathbf{j}' + z'\mathbf{k}' + \mathbf{d},$$

aminek mátrix alakja

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{i}' & \mathbf{j}' & \mathbf{k}' & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} i'_x & j'_x & k'_x & d_x \\ i'_y & j'_y & k'_y & d_y \\ i'_z & j'_z & k'_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}.$$



5.2. ábra. Térbeli koordinátatranszformáció

Tekintsük a feladat inverzét!

Adottak térben az $\mathbf{i}, \mathbf{j}, \mathbf{k}$ és az $\mathbf{i}', \mathbf{j}', \mathbf{k}'$ egységvektorú derékszögű Descartes-féle koordináta-rendszerek. Ismerjük továbbá a P pontnak a régi koordináta-rendszerbeli $\mathbf{p} = (x, y, z)$ helyvektorát. Határozzuk meg a P pont $\mathbf{i}', \mathbf{j}', \mathbf{k}'$ új koordináta-rendszerbeli \mathbf{p}' helyvektorának (x', y', z') komponenseit. A régi koordináta-rendszerben $\mathbf{p}' = \mathbf{p} - \mathbf{d}$. \mathbf{p}' -nek az új koordináta-rendszerbeli komponenseire

$$\begin{aligned} x' &= \mathbf{p}' \cdot \mathbf{i}' = (\mathbf{p} - \mathbf{d}) \cdot \mathbf{i}' = \mathbf{p} \cdot \mathbf{i}' - \mathbf{d} \cdot \mathbf{i}', \\ y' &= \mathbf{p}' \cdot \mathbf{j}' = (\mathbf{p} - \mathbf{d}) \cdot \mathbf{j}' = \mathbf{p} \cdot \mathbf{j}' - \mathbf{d} \cdot \mathbf{j}', \\ z' &= \mathbf{p}' \cdot \mathbf{k}' = (\mathbf{p} - \mathbf{d}) \cdot \mathbf{k}' = \mathbf{p} \cdot \mathbf{k}' - \mathbf{d} \cdot \mathbf{k}'. \end{aligned}$$

A tranzformáció mátrix alakja

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} i'_x & i'_y & i'_z & -\mathbf{d} \cdot \mathbf{i}' \\ j'_x & j'_y & j'_z & -\mathbf{d} \cdot \mathbf{j}' \\ k'_x & k'_y & k'_z & -\mathbf{d} \cdot \mathbf{k}' \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

5.2. Ponttranszformációk

Ponttranszformáción olyan megfeleltetést értünk, amely során egy pontthalmaz minden pontjához hozzárendelünk egy-egy pontot. Mi itt csak olyan ponttranszformációkat vizsgálunk, amelyek kölcsönösen egyértelműek, illeszkedés és egyenes tartók.

A tranzformációk általános formája

$$\mathbf{p}' = \mathbf{M}\mathbf{p}$$

ahol \mathbf{p} a tranzformálandó, \mathbf{p}' a tranzformált pont helyvektora, \mathbf{M} pedig a tranzformációt leíró mátrix, a továbbiakban a tranzformáció mátrixa. Síkbeli tranzformáció esetén \mathbf{M} 3×3 -as, térbeli tranzformációnál pedig 4×4 -es mátrix, és mindkét esetben $|\mathbf{M}| \neq 0$.

Mind a síkbeli, mind a térbeli transzformációknál az úgynevezett elemi transzformációkat tárgyaljuk. Ezekből az alapvető, legegyszerűbb transzformációkból lehet összetett transzformációkat létrehozni a transzformációk egymás utáni végrehajtásával. Az elemi ponttranszformációk mátrixainak meghatározásához felhasználjuk a koordináta-rendszer transzformációknál szerzett ismereteket. A transzformálandó alakzathoz rögzítünk egy koordináta-rendszert, amely az alakzattal együtt transzformálódik, és így a ponttranszformációt a koordináta-rendszer transzformációra vezetjük vissza.

A transzformációkat a középiskolában megszokott módon csoportosítjuk, vagyis aszerint, hogy mi marad változatlan a transzformáció során.

5.2.1. Síkbeli ponttranszformációk

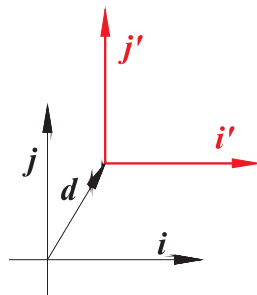
Egybevágósági transzformáció

Az egybevágósági transzformációk és a koordináta-rendszer transzformációk között az az alapvető különbség, hogy a koordináta-rendszer transzformáció esetén az objektum helyben marad és a "nézőpont" változik, egybevágósági ponttranszformációk esetén viszont rögzített vonatkoztatási rendszer mellett az objektum "mozog".

Eltolás. A $d = (d_x, d_y)$ vektorral való eltolást (5.3. ábra) az

$$M = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

mátrix írja le.

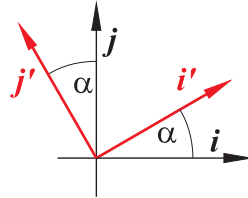


5.3. ábra. Síkbeli eltolás

Elforgatás. Az origó körül α szöggel való elforgatás (5.4. ábra) mátrixa

$$\mathbf{M} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Az óramutató járásával megegyező forgatási irányhoz negatív, az ezzel ellentéteshez pedig pozitív forgatási szög tartozik.



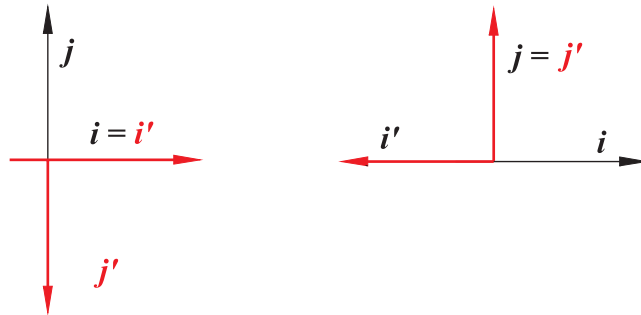
5.4. ábra. Origó körüli forgatás a síkon

Tükrözés. Az x tengelyre vonatkozó tükrözés (5.5. ábra bal oldal) mátrixa

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

az y tengelyre (5.5. ábra jobb oldal) vonatkozó pedig

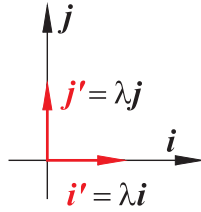
$$\mathbf{M} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$



5.5. ábra. Tükrözés az x tengelyre (bal oldali ábra) és az y tengelyre (jobb oldali ábra)

Az eltolást és elforgatást összefoglalóan mozgásnak szokás nevezni, ugyanis ilyen esetben van olyan síkbeli "mozgás" amivel az egybevágó alakzatok egymással "fedésbe" hozhatók. Az egybevágósági transzformációk közös jellemzője, hogy mátrixuk bal felső 2×2 -es minor mátrixa ortonormált mátrix – azaz sorai és oszlopai páronként merőleges egységvektorok – és determinánsa ± 1 . Mozgás esetén 1, tükrözésnél pedig -1 . Egyenesre vonatkozó tükrözés során az alakzatok körüljárási iránya megváltozik, ezért nincs a síkon olyan mozgás amellyel fedésbe lehetne hozni az alakzatot és a tükörképét, ehhez ki kellene lépni a térbe.

Hasonlósági transzformáció



5.6. ábra. Origó középpontú kicsinyítés

Kicsinyítés, nagyítás. Az origó középpontú kicsinyítés, nagyítás (5.6. ábra) mátrixa

$$\mathbf{M} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

ahol $0 < \lambda \in \mathbb{R}$ a hasonlósági tényező. Ugyanezt a hasonlósági transzformációt írja le a

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

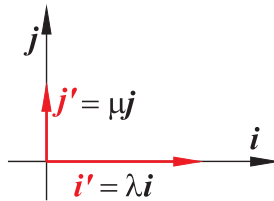
mátrix is, mivel

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/\lambda \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1/\lambda \end{bmatrix} = \begin{bmatrix} \lambda x \\ \lambda y \\ 1 \end{bmatrix}.$$

Affin transzformáció

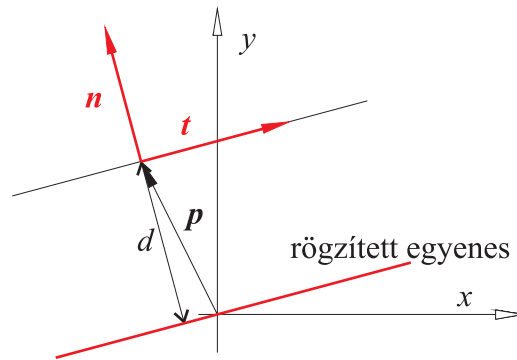
Skálázás. Skálázáson az x és y tengelyek mentén alkalmazott különböző mértékű kicsinyítést (nagyítást) értjük. Az x , illetve y tengely mentén $0 < \lambda \in \mathbb{R}$, illetve $0 < \mu \in \mathbb{R}$ mértékű kicsinyítés (nagyítás) (5.7. ábra) mátrixa

$$\mathbf{M} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$



5.7. ábra. Skálázás

A skálázási transzformációval gyakran találkozunk a számítógépi grafikában. Végezzük el a következő kísérletet! Számítsuk ki egy körre illeszkedő pixelek koordinátáit, és rajzoljuk meg azokat! Azt tapasztaljuk, hogy kör helyett ellipszist látunk a képernyőn. Ez annak a következménye, hogy a pixel nem négyzet, hanem téglalap alakú, aminek következtében a megjelenítés során a kört egy skálázási transzformációnak vetjük alá. Ha tehát kör képeként kört szeretnénk látni a képernyőn, akkor ellipszist kell rajzolnunk, vagyis a körre a pixelek szélesség / magasság arányából eredő skálázási transzformáció inverzét kell alkalmaznunk. A kirajzolandó ellipszis kis- és nagytengelyének aránya a pixel szélesség / magasság arányából számolható ki.



5.8. ábra. Síkbeli nyírás

Nyírás. A nyírás a sík pontjainak olyan eltolása, ahol az eltolás mértéke arányos a pontnak, egy az eltolás irányával párhuzamosan rögzített egyenestől mért előjeles távolságával. Ezt szemlélteti az 5.8. ábra. Az ábrán a rögzített egyenes áthalad az origón, t az eltolás irányába mutató egységvektor, n pedig ennek pozitív normál egységvektora. A λ mértékű nyírás

$$\mathbf{p}' = \mathbf{p} + \lambda d \mathbf{t} = \mathbf{p} + (\lambda \mathbf{n} \cdot \mathbf{p}) \mathbf{t} \quad (5.1)$$

alakban írható fel. Ennek a transzformációnak a mátrixát a következőképpen kaphatjuk meg. Felhasználjuk, hogy

$$(\mathbf{n} \cdot \mathbf{p}) \mathbf{t} = \mathbf{n}^T \mathbf{p} \mathbf{t}$$

(a jobb oldal mátrixok szorzata), valamint

$$\mathbf{n}^T \mathbf{p} \mathbf{t} = \mathbf{t} \mathbf{n}^T \mathbf{p}$$

($\mathbf{t} \mathbf{n}^T$ diadikus szorzat). Ezt behelyettesítve (5.1)-be

$$\mathbf{p}' = \mathbf{p} + \lambda (\mathbf{t} \mathbf{n}^T) \mathbf{p} = (\mathbf{I} + \lambda (\mathbf{t} \mathbf{n}^T)) \mathbf{p},$$

ahol \mathbf{I} az egységmátrix.

$$\lambda (\mathbf{t} \mathbf{n}^T) = \lambda \begin{bmatrix} t_x \\ t_y \\ 0 \end{bmatrix} \begin{bmatrix} n_x & n_y & 0 \end{bmatrix} = \lambda \begin{bmatrix} t_x n_x & t_x n_y & 0 \\ t_y n_x & t_y n_y & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{I} + \lambda(\mathbf{tn}^T) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} t_x n_x & t_x n_y & 0 \\ t_y n_x & t_y n_y & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 + \lambda t_x n_x & \lambda t_x n_y & 0 \\ \lambda t_y n_x & 1 + \lambda t_y n_y & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A nyírás fontos speciális esetei, amikor a rögzített egyenes valamelyik koordinátatengely. Ha a rögzített egyenes az y tengely, akkor

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ \lambda & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\begin{aligned} x' &= x, \\ y' &= y + \lambda x; \end{aligned}$$

ha pedig az x tengely, akkor

$$\mathbf{M} = \begin{bmatrix} 1 & \lambda & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\begin{aligned} x' &= x + \lambda y, \\ y' &= y. \end{aligned}$$

Általános affin transzformáció. Síkbeli affin transzformációt ír le minden olyan 3×3 -as nonszinguláris mátrix, amelynek harmadik sora $(0, 0, c)$, $c \neq 0$. Az affin transzformációk egy fontos jellemzője, hogy a párhuzamosságot megőrzik, tehát a négyzet affin megfelelője legáltalánosabb esetben paralelogramma lehet, a köré pedig ellipszis.

Projektív transzformáció

A projektív transzformáció mátrixára csak az a kikötés, hogy determinánsa legyen nullától különböző, azaz létezzen inverze. Tehát bármely nonszinguláris 3×3 -as mátrix síkbeli projektív transzformációt ír le. A projektív transzformációk során az egyenes képe egyenes lesz, de a párhuzamosság nem marad változatlan. Ezért a négyzet képe tetszőleges négyszög lehet, a körből pedig lehet parabola vagy hiperbola is.

5.2.2. Összetett transzformációk

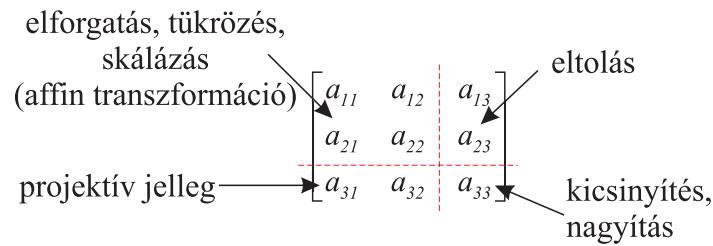
A síkbeli alakzatokon több transzformációt hajthatunk végre egymás után. Ha a \mathbf{p} pontból az \mathbf{M}_1 transzformációval kapott $\mathbf{p}' = \mathbf{M}_1 \mathbf{p}$ pontot az \mathbf{M}_2 transzformációval a $\mathbf{p}'' = \mathbf{M}_2 \mathbf{p}'$ pontba visszük át, akkor

$$\mathbf{p}'' = \mathbf{M}_2 \mathbf{p}' = \mathbf{M}_2 (\mathbf{M}_1 \mathbf{p}) = (\mathbf{M}_2 \mathbf{M}_1) \mathbf{p}$$

miatt a \mathbf{p} pontból az $\mathbf{M} = \mathbf{M}_2 \mathbf{M}_1$ transzformációval a \mathbf{p}'' -t kapjuk. Ennek a gyakorlatban igen nagy haszna van, mivel ha több transzformációt akarunk egymás után végrehajtani több ponton

(pl. egy összetett objektumot meghatározó pontokon), akkor összeszorozzuk az egyes transzformációkat leíró mátrixokat a megfelelő sorrendben, és a pontokat csak ezzel a szorzatmátrixszal (az eredő transzformáció mátrixával) kell megszorozni. Így igen sok szorzást és összeadást, azaz időt takaríthatunk meg.

Összetett transzformációt elemi transzformációk egymás utáni végrehajtásával állíthatunk elő. Így a tetszőleges mozgás eltolásokból és elforgatásokból; az általános egybevágósági transzformáció eltolásokból, elforgatásokból és tükrözésekből; az általános hasonlósági transzformáció egybevágósági transzformációkból és origó középpontú kicsinyítésekből vagy nagyításokból; az általános affin transzformáció pedig skálázásokból és egybevágósági transzformációkból állítható elő.



5.9. ábra. A síkbeli ponttranszformációk mátrixának particionálása

Például a sík tetszőleges $C(c_x, c_y)$ pontja körüli α szögű elforgatását három elemi transzformációra bonthatjuk.

1. A C pontot betoljuk az origóba, azaz $(-c_x, -c_y)$ komponensű eltolás (M_1);
2. origó körüli elforgatás α szöggel (M_2);
3. a C pontot visszatoljuk eredeti helyére, azaz (c_x, c_y) komponensű eltolás (M_3).

Ezek után a transzformáció felírható a $\mathbf{p}' = \mathbf{M}\mathbf{p}$ alakban, ahol $\mathbf{M} = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1$.

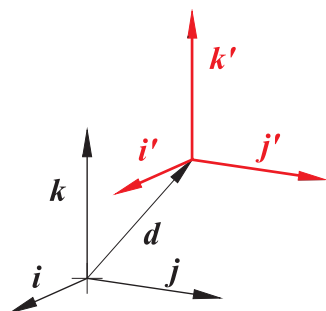
A síkbeli ponttranszformációkat leíró 3×3 -as mátrixok az 5.9. ábra szerint particionálhatók. Az egyes partíciók geometriai hatását is mutatja az ábra.

5.2.3. Térbeli ponttranszformációk

Egybevágósági transzformáció

Eltolás. A $\mathbf{d} = (d_x, d_y, d_z)$ vektorral való eltolás (5.10. ábra) mátrixa

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

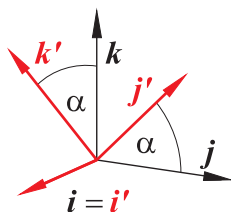


5.10. ábra. Térbeli eltolás

Elforgatás. Térbeli ponttranszformációknál a koordinátatengely körüli elforgatásokat tekintjük elemi forgatási transzformációnak. α szöggel forgatunk a tengelyek körül, az óramutató járásával megegyező forgatási irányhoz negatív, az ezzel ellentéteshez pedig pozitív forgatási szög tartozik.

Az x tengely körüli forgatás (5.11. ábra) mátrixa

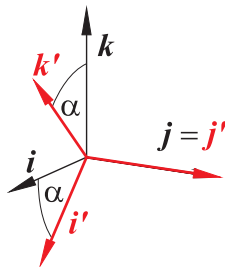
$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$



5.11. ábra. Forgatás az x tengely körül

az y tengely körüli forgatásé (5.12. ábra)

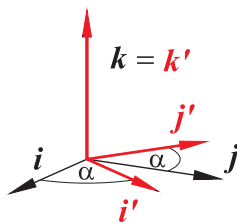
$$\mathbf{M} = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$



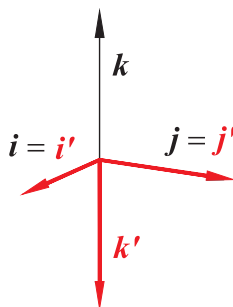
5.12. ábra. Forgatás az y tengely körül

az z tengely körüli forgatásé pedig (5.2.3. ábra)

$$\mathbf{M} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$



Tükrözés. A koordinátasíkokra vonatkozó tükrözéseket tekintjük elemi tükrözési transzformációnak.



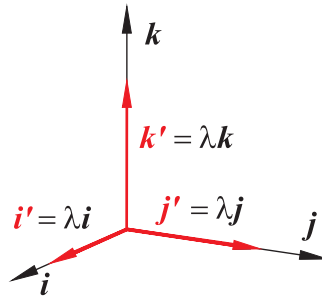
5.13. ábra. Tükrözés az (x, y) koordinátasíkra

Az (x, y) síkra vonatkozó tükrözés (5.13. ábra) mátrixa

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Az (y, z) , illetve (z, x) síkokra vonatkozó tükrözés mátrixa a fentivel analóg, azaz az egység-mátrix első, illetve második oszlopát kell -1 -gyel megszorozni.

Hasonlósági transzformáció



5.14. ábra. Origó középpontú kicsinyítés

Kicsinyítés, nagyítás. Az origó középpontú kicsinyítés (nagyítás) (5.14. ábra) mátrixa

$$\mathbf{M} = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

ahol $0 < \lambda \in \mathbb{R}$ a hasonlósági tényező. Ugyanezt a hasonlósági transzformációt írja le a

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

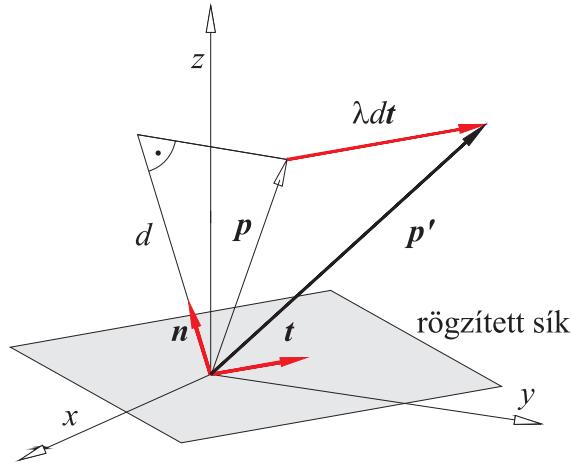
mátrix is.

Affin transzformációk

Skálázás. Az x, y, z tengelyek mentén $\lambda > 0, \mu > 0, \nu > 0$ mértékű kicsinyítés, vagy nagyítás (1.20 ábra) mátrixa

$$\mathbf{M} = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \nu & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Nyírás. A térbeli nyírás a síkbeli általánosítása, tehát a tér pontjainak egy síkkal párhuzamos eltolása mégpedig úgy, hogy az eltolás arányos a pontnak egy az eltolás irányával párhuzamosan rögzített síktól mért előjeles távolságával. Mi olyan nyírást vizsgálunk ahol a rögzített sík áthalad az origón. A síkállás megadható annak \mathbf{n} normál egységvektorával, az eltolás pedig az irány \mathbf{t} egységvektorával, ami tehát merőleges \mathbf{n} -re (lásd az 5.15. ábrát).



5.15. ábra. Térbeli nyírás

A λ mértékű nyírás

$$\mathbf{p}' = \mathbf{p} + \lambda d \mathbf{t} = \mathbf{p} + (\lambda \mathbf{n} \cdot \mathbf{p}) \mathbf{t}$$

alakban írható. Ennek mátrixát a síkbeli esetről követett eljárással írhatjuk fel

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + \lambda t_x n_x & \lambda t_x n_y & \lambda t_x n_z & 0 \\ \lambda t_y n_x & 1 + \lambda t_y n_y & \lambda t_y n_z & 0 \\ \lambda t_z n_x & \lambda t_z n_y & 1 + \lambda t_z n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

alakban. Ebből könnyen megkaphatjuk azokat a speciális eseteket, amikor \mathbf{n} párhuzamos valamelyik koordinátatengellyel.

Általános affin transzformáció. Minden olyan 4×4 -es nonszinguláris mátrix, amelynek negyedik sora $(0, 0, 0, c)$, $c \neq 0$ térbeli affin transzformációt ír le.

Projektív transzformáció

Minden nonszinguláris 4×4 -es mátrix térbeli projektív transzformációt ír le. A térbeli ponttranszformációkon is értelmezhetjük a szorzás műveletét, mint a transzformációk egymás utáni végrehajtását. Erre is igaz minden, a síkbeli transzformációknál tett megállapítás. A 3×3 -as mátrixokat értelemszerűen 4×4 -es mátrixok váltják fel. A projektív transzformáció tehát a legáltalánosabb, 4×4 -es mátrixszal leírható térbeli lineáris transzformáció. Az ennél általánosabb transzformációk - pl. a csak a folytonosságot megőrző topológikus transzformációk - sajnos nem írhatók le egyetlen mátrixszal.

6. fejezet

Térbeli alakzatok megjelenítése

Az ember-számítógép kapcsolat kezdetén az információcsere kizárólag alfanumerikus adatok (számok, szöveges üzenetek) segítségével történt. A grafikai hardver fejlődése azonban lehetővé tette, hogy előbb csak az eredmények megjelenítése, később már az adatok bevitele is grafikus formában menjen végbe. Napjainkban a grafikus információcserének döntő szerepe van a számítástechnika minden területén.

A grafikus információ kimenetet legáltalánosabb értelemben grafikus szemléltetésnek (rendering) nevezzük. A grafikus szemléltetés a tervezői tevékenység minden területén szerephez jut, így a hagyományos értelemben vett műszaki rajz készítésénél, testek és felületek tervezésekor, alkatrészek fizikai jellemzőinek vizsgálatakor (pl. végeelem-analízis), vagy folyamatok, szerkezetek működésének szimulációjakor. A mérnöki tevékenység támogatásához szükséges grafikai eszköztár tehát igen széles, az egyszerű síkbeli rajzok készítésétől, a felületek és testek színes, árnyalt megjelenítésén át, az animációig terjed.

A CAD rendszerekben alkalmazott szemléltetési módszerekkel szemben követelmény, hogy elemi alkatrészek, részegységek és összeállítások modelljeire egyaránt alkalmazhatók legyenek; az alkalmazott hardver képességeit kihasználva megfelelő hatékonyságú feldolgozást biztosítsanak; és a szemléltetés minősége alkalmazkodjon az adott alkalmazás igényéhez.

A számítógéppel segített tervezés különböző szakaszaiban a modell megjelenítéssel szemben eltérő követelményeket támasztunk. A modellek létrehozása során a gyors, a geometriai alapegységeket teljes körűen szemléltető, ugyanakkor könnyen értelmezhető vizuális visszacsatolás a követelmény, a kész modell, vagyis a termék megjelenítésekor pedig a minél nagyobb valószerűség az elérendő cél. Az utóbbi esetben minimális igény a láthatóság szerinti ábrázolás, de a mai CAD rendszerek lehetővé teszik a felhasználó által megadott fényviszonyoknak és anyagtulajdonságoknak megfelelő árnyalt képek előállítását is. A szemléltetésnek ezt az igényes formáját az angol nyelvterületen renderingnek nevezik.

Az objektumok szemléltetésére használt módszerek elsősorban attól függenek, hogy mit - milyen típusú modellt -, és min - milyen megjelenítő eszközön - akarunk megjeleníteni. Az ismertetendő algoritmusok a jelenleg leginkább elterjedt színes raszteres grafikus display használatát feltételezik. A kereskedelemben jelenleg kapható CAD rendszerek túlnyomó többsége - még akkor is, ha a felhasználói interfész nem egyértelműen erre utal - poliédermodellt használ, ezért a poliédermodellek megjelenítése áll vizsgálatunk középpontjában.

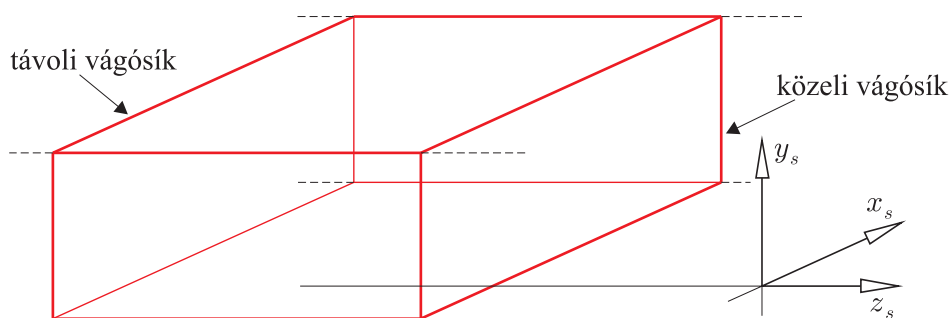
Síkbeli alakzatok megjelenítése nem okoz nehézséget. Az igazi kihívás a testek, felületek, térgörbék, vagyis a térbeli objektumok szemléltetése. Ebben az esetben ugyanis a háromdimenziós tér alakzatait a kétdimenziós síkon (képsíkon, képernyőn) kell ábrázolni. A térnek síkra való leképezésével, illetve síkbeli képek rekonstrukciójával foglalkozó szűkebb tudományterü-

let az ábrázoló geometria, amivel hazánkban minden műszaki tanulmányokat folytató hallgató kapcsolatba kerül.

Az alábbiakban összefoglaljuk a grafikus szemléltetésnek a CAD rendszerek szempontjából legfontosabb elméleti alapjait.

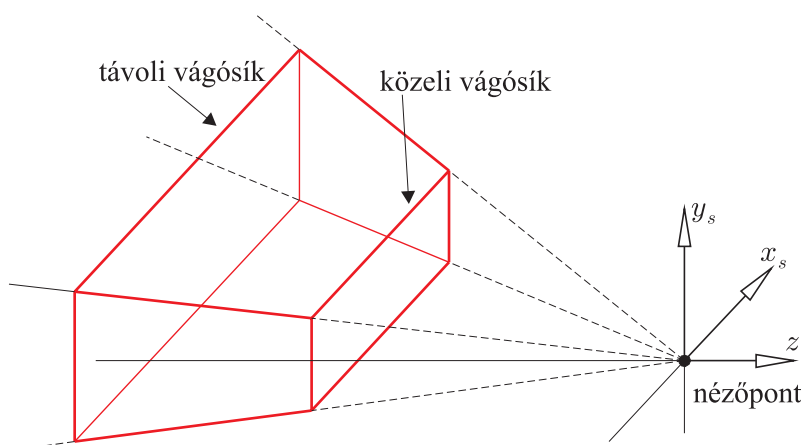
6.1. A tér leképezése a síkra

Térnek síkra való leképezéséhez merőleges vetítést használunk, ha mérethelyes, vagy legalább a méretek arányait tükröző képet, például műszaki rajzot akarunk létrehozni (elől-, felül- és oldalnézet). Szemléletes képek készítéséhez legtöbbször valamilyen axonometriát, esetleg centrális vetítést (perspektívát) alkalmazunk.



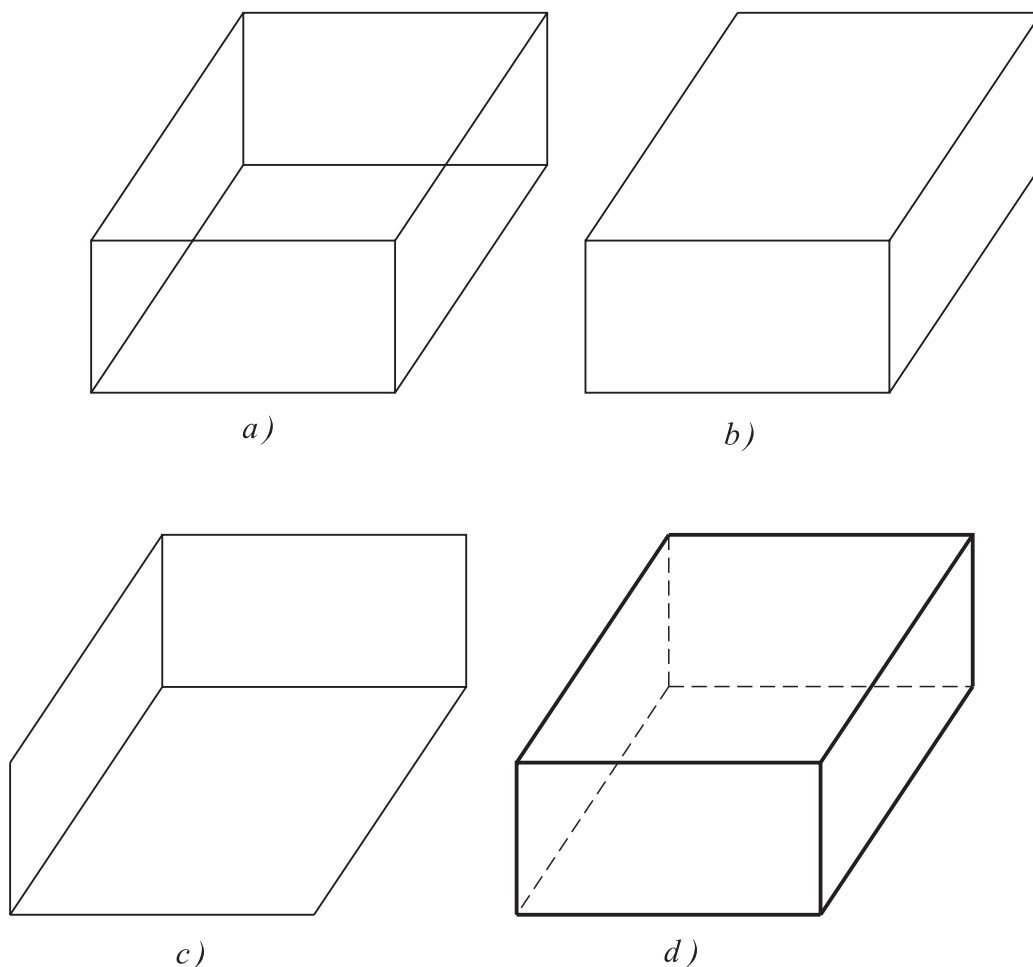
6.1. ábra. A merőleges vetítéshez megadandó téglatest

A térnek síkra való fenti leképezései matematikailag elfajult ponttranszformációt jelentenek, amit le lehet írni 4×4 -es mátrixszal. A grafikus rendszerekben azonban nem ezt a mátrixot kell megadni, hanem az ábrázolandó véges térrészt. Merőleges vetítés esetén az ábrázolandó térrész a koordináta-rendszer tengelyeivel párhuzamos oldalú téglatest (lásd a 6.1. ábrát), centrális vetítés esetén pedig csonka gúla (lásd a 6.2. ábrát).



6.2. ábra. A felhasználó által megadandó csonka gúla centrális vetítés esetén

A szemléletességhez azonban nem elegendő pusztán levetíteni az alakzatot a síkra, mert egy ilyen kép alapján nem mindig egyértelmű az alakzat térbeli elhelyezkedése. Ezt illusztrálja



6.3. ábra. Hasáb ábrázolása éleivel

a 6.3. ábra. A 6.3. a) ábra hasábját ugyanis mind a 6.3. b), mind a 6.3. c) ábra szerint értelmezhetjük. A probléma gyökere abban van, hogy ezek a leképezések dimenzióvesztők, ezért annak érdekében, hogy az egyébként kiváló emberi agy a mélységet, térbeli elhelyezkedést is érzékelje, az elvesztett információt valamilyen grafikai trükkal pótolni kell.

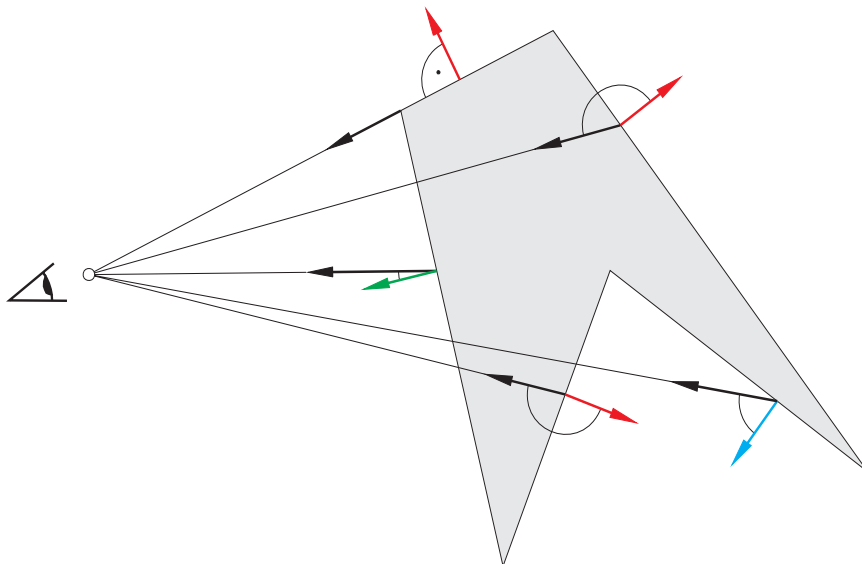
A 6.3. b) és 6.3. c) ábrák sem teljesen egyértelműek, hiszen mindkét ábrát tudjuk úgy is látni, mint három egymáshoz kapcsolódó síklapot, bár ha tudjuk, hogy testről van szó, egy átlagos térlátású embernek nem okoz gondot a hasáb felismerése. Megkönnyítjük azonban a szemlélő dolgát, ha a nem látható éleket szaggatott, a látható éleket pedig vastagabb vonallal rajzoljuk (6.3. d). Ugyancsak növeli a szemléletességet, ha valamilyen megvilágítás mellett árnyaljuk a látható lapokat.

A továbbiakban olyan módszereket ismertetünk, melyek segítségével igényes képeket hozhatunk létre olyan alakzatokról, melyek esetleg csak a tervező képzeletében - és a tervező által feltöltött adatbázisban - léteznek. Számos ilyen algoritmus létezik, mi itt azonban csak a CAD rendszerekben legelterjedtebb, azok használatához és működésének megértéséhez szükséges algoritmusokat mutatjuk be. Az algoritmusok ismertetésénél azt feltételezzük, hogy az ábrázolandó objektum egy olyan speciális koordináta-rendszerben van leírva, ahol a képsík az (x, y) koordinátasík.

6.2. Láthatósági vizsgálatok

A láthatóság szerinti ábrázolásnál két esetet szokás megkülönböztetni. Az egyik az, amikor a takart éleket távolítjuk el (hidden line elimination), a másik pedig amikor a takart felületeket (hidden surface elimination). Vonalas megjelenítők (tollal rajzoló plotter, vektoros display) esetén az első lehetséges, illetve célszerű, a raszteres megjelenítők (pl. raszteres grafikus display, tintasugaras vagy lézer nyomtató) esetén azonban a nagyobb valószerűséget biztosító második.

A láthatóság meghatározásának alapproblémája annak eldöntése, hogy a közös vetítősugárra illeszkedő pontok közül melyik látszik, vagyis a nézőponthoz melyik van legközelebb. Sok láthatósági algoritmus arra épül, hogy a kép pixelekből (képpontokból) áll, és ennek megfelelően egy objektum ábrázolása azt jelenti, hogy a képet alkotó pixelek színét kell meghatározni. Ezért ezeket az algoritmusokat kép-pontosságú algoritmusoknak is nevezik, szemben az úgynevezett objektum-pontosságú algoritmusokkal, melyek az objektumok látható részeit határozzák meg a modell térben. Az utóbbi megközelítés elsősorban a takart-vonalas algoritmusokra jellemző.



6.4. ábra. Hátsó lapok kiszűrése

6.2.1. Hátsó lapok eltávolítása

Poliédrommodellek esetén a láthatóság szerinti ábrázolás első lépéseként célszerű a hátsó lapokat eltávolítani (backface culling), vagyis figyelmen kívül hagyni azokat a lapokat, amelyek normálisa a 6.4. ábra jelölése szerint a nézőpontba mutató fénysugárral 90° -nál nagyobb vagy egyenlő szöget zárnak be (az ábrán a piros színű normálisok). Itt, és a továbbiakban azt feltételezzük, hogy a poliéderek, poligonhálók lapjainak normálisai rendelkezésünkre állnak, és ezen normálisok irányítása konzisztens, vagyis mindegyik vagy kifelé, vagy befelé mutat. A 6.4. ábra perspektívában ábrázolandó, alapsíkra merőleges oldalélű hasáb felülnézetét - az alapsíkra merőleges vetületét - szemlélteti. Az irányított lapok normálisai a felületről kifelé mutatnak. A számítások elvégzése gyors és egyszerű, hiszen csak a megfelelő vektorok skaláris szorzatának előjelét kell vizsgálni.

Ha csak egyetlen konvex poliédert kell ábrázolni, akkor a láthatósági probléma a hátsó lapok eltávolításával megoldott. Konkáv vagy több, egymást részben fedő poliéder esetén azonban további vizsgálatokra van szükség.

Nem alkalmazhatjuk a hátsó lapok kiszűrésének módszerét abban az esetben, ha a lapok (poligonok) nem egy zárt térrészt határolnak. Felületeket, héjszerű szerkezeteket modellező poligonháló esetén ugyanis előfordulhat, hogy egyes poligonok hátoldalát látjuk, vagyis, ha pusztán a felületi normálisok irányítása alapján elhagyunk egy lapot a képen lyuk keletkezhet, mely nem a térbeli állapotot tükrözi.

6.2.2. Mélység-puffer (z-puffer) algoritmus

A mélység-puffer (depth buffer, z-buffer) algoritmus segítségével poligonokkal határolt alakzatokat ábrázolhatunk színes raszteres megjelenítőkön. Az algoritmus azt vizsgálja, hogy egy-egy pixelen mi látszik, de nem kell rendezést végrehajtani pixelenként sem, csupán mindig két értéket kell összehasonlítani. Az algoritmushoz a pixelek színének tárolása (frame buffer) mellett meg kell oldani a mélységek (z értékek) tárolását is. Erre szolgál az úgynevezett mélység-puffer vagy z-puffer, ami egyben névadója is az algoritmusnak.

Az algoritmus főbb lépései a következők:

- a mélység-puffert a maximális mélységre állítjuk, vagyis arra a mélységre, amelytől távolabbi pontokat már nem veszünk figyelembe, a pixelekhez pedig a háttér színét rendeljük;
- az objektumok 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 pixelnek a mélység-pufferben tárolt értékénél, vagyis az új pont látszik, akkor:
 - * a pixel mélységét a pont mélységére állítjuk be;
 - * meghatározzuk a pont színét, és ezt rendeljük a pixelhez.

Bár ezt az algoritmust eredetileg poligonok, poligonhálók láthatóság szerinti ábrázolására fejlesztették ki - ugyanis ezeket lehet legegyszerűbben pixellé konvertálni -, minden olyan objektum láthatóság szerinti ábrázolására alkalmas, mely képének pontjaihoz (a képre eső pixelekhez) meghatározható a mélység és a szín. Poliédermodell esetén a felületi pontok meghatározása, a felület pixelekké konvertálása nem nehéz feladat.

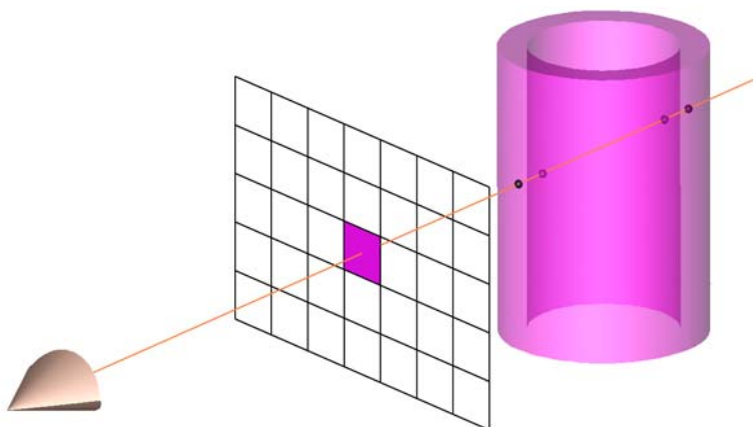
Az algoritmus egyszerű, létezik hardver implementációja is, amit napjainkban egyre több grafikus kártya tartalmaz. Az algoritmus hátránya a mélység-pufferhez szükséges viszonylag nagy memóriaterület.

6.2.3. Fénysugárkövető algoritmus

A fénysugárkövető módszer (ray tracing, ray casting) is azt használja ki, hogy a kép pontokból (pixelekből) épül fel. Alapötlete az, hogy meghatározza az ábrázolandó objektumnak azt a pontját, amely egy adott pixelen látszik.

Az algoritmus a következő lépésekből áll:

- a nézőpontot (a vetítés középpontját) összekötjük a kép pixeleivel (párhuzamos vetítés esetén a vetítési iránnyal párhuzamos egyenest húzunk a pixeleken át), azaz egy sugarat bocsátunk ki a nézőpontból a pixeleken - pontosabban a képsíkon a pixeleknek megfelelő téglalap szimmetria-középpontján - át a modelltérbe (6.5. ábra);
- ezekkel a félegyenesekkel (sugarakkal) elmetsszük az ábrázolandó objektumokat;
- meghatározzuk a nézőponthoz legközelebbi metszéspontot, majd ennek a pontnak a színét, és a pixelhez ezt a színt rendeljük.



6.5. ábra. A láthatóság meghatározása fénysugárkövető algoritmussal

Ez az algoritmus, bár alapötletét tekintve egyszerű, sok számítást kíván, ezért időigényes. Az egyes pixelek színének meghatározása független egymástól ezért az algoritmus alkalmas a párhuzamos feldolgozásra, amivel jelentősen felgyorsítható.

7. fejezet

Fényforrások, megvilágítás, árnyalás

A fényforrások, megvilágítás és árnyalás kezelése a számítógépi grafikában legtöbbször nem a valóságos fizikai jelenségek leírásán, modellezésén alapul, hanem tapasztalati úton nyert összefüggéseken.

7.1. Fényösszetevők

Az ábrázolandó térrészben uralkodó fényviszonyok leírására az alábbi fényösszetevőket szokás figyelembe venni:

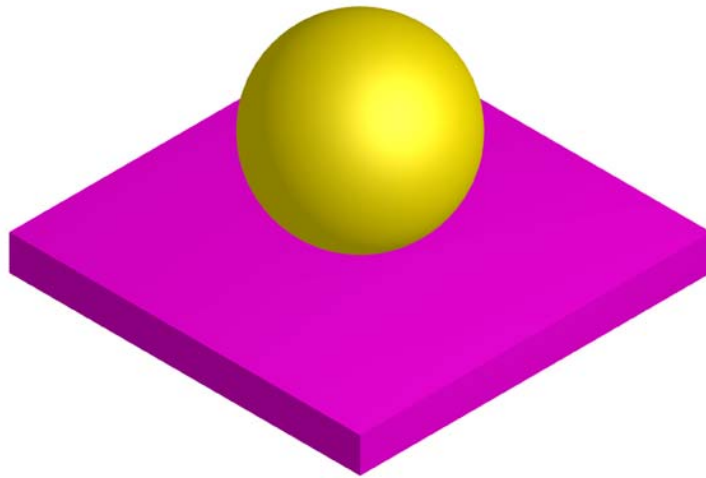
- A környezeti fény (ambient light) az ábrázolandó térrészben mindenütt jelen lévő, állandó intenzitású fény, amelynek forrása, iránya nem ismert (gondoljunk olyan nappali fényre, amikor a nap a felhők mögött van).
- A szórt fénynek (diffuse light) van iránya, mindig valamelyik fényforrásból jön. Fő jellemzője, hogy az objektumokkal ütközve minden irányba azonos módon és mértékben verődik vissza, tehát teljesen mindegy, hogy milyen irányból nézzük az objektumokat, a hatás csak a fényforrástól, az anyagtulajdonságoktól és a pontbeli normálistól függ.
- A tükröző fénynek (specular light) is van iránya és forrása, és hatása nemcsak az anyagtulajdonságoktól és a pontbeli normálistól, hanem a nézőponttól is függ. Gondoljunk egy sima felületű fémgömbre, amit erős, koncentrált fénnel világítunk meg. Megfelelő szögből nézve egy fényes (többnyire fehér) foltot látunk, amelynek mérete és helye a nézőponttól függ, fejünket mozgatva a folt is mozog, mígnem eltűnik.

7.2. Fényforrások

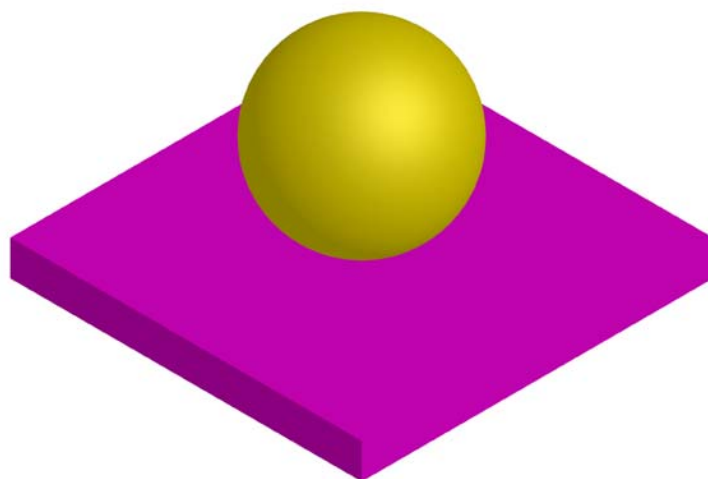
Az ábrázolandó térrészben jelen lévő környezeti fény két részből tevődik össze. Egyrészt a térben a fényforrásoktól függetlenül (azokra vissza nem vezethető) jelen lévő környezeti fényből - ezt globális környezeti fénynek is nevezik -, másrészt a fényforrásokból származó (pl. többszörös tükröződések útján keletkező) környezeti fényből, amit a fényforrások környezeti fénykomponensének nevezünk. A globális környezeti fény tehát speciális fényforrásnak tekinthető.

Bármely más fényforrás létrehozásához meg kell adni a fényforrás által kibocsátott fény RGB komponenseit, valamint a fényforrás helyét. A fényforrás által kibocsátott fényt a környezeti, szórt és tükröző fénykomponenseivel adhatjuk meg.

A fényforrás helyét tekintve lehet végesben lévő (pontoszerű fényforrás, 7.1. ábra) és végtelen távolban lévő. A végtelen távolban lévő fényforrásból kibocsátott sugarak párhuzamosak (ilyenek tekinthetők a valóságban a napsugarak, 7.2. ábra), és a fényforrás pozíciója ezek irányával adható meg. A pontoszerű fényforrás azonos intenzitású fényt bocsát ki minden irányban.



7.1. ábra. Megvilágítás pontoszerű fényforrással



7.2. ábra. Megvilágítás párhuzamos fénysugarakkal

Pontszerű fényforrás esetén megadható, hogy a fényforrástól távolodva milyen mértékben csökken a fény erőssége, azaz hogyan tompul a fény (attenuation). A fény tompulását az

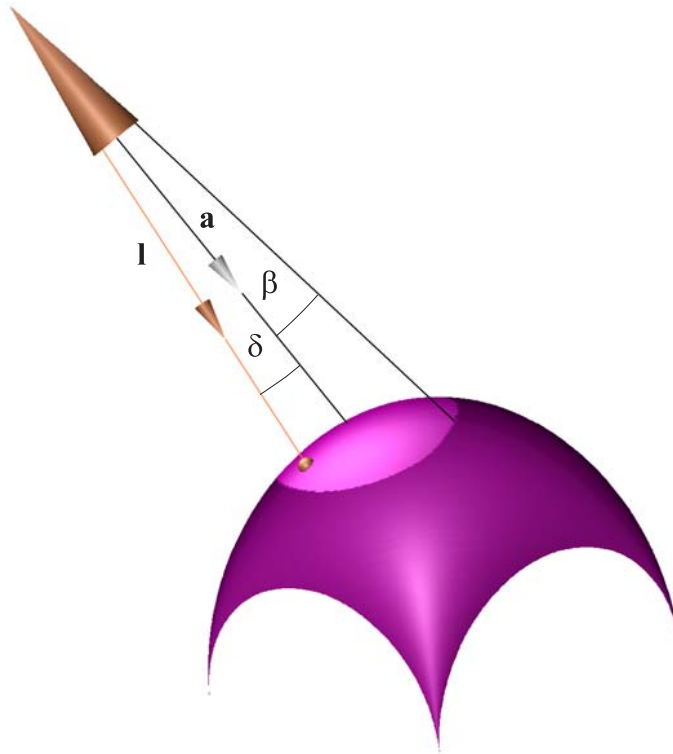
$$f_{att} = \min \left(\frac{1}{c_0 + c_1 d + c_2 d^2}, 1 \right)$$

kifejezéssel szokták leírni, ahol d a fényforrástól mért távolság, c_0 a távolságtól független, c_1 a távolsággal arányos, c_2 pedig a távolság négyzetével arányos együtthatója a fény tompulásának. A fény tompulását tehát a c_i együtthatókkal adhatjuk meg.

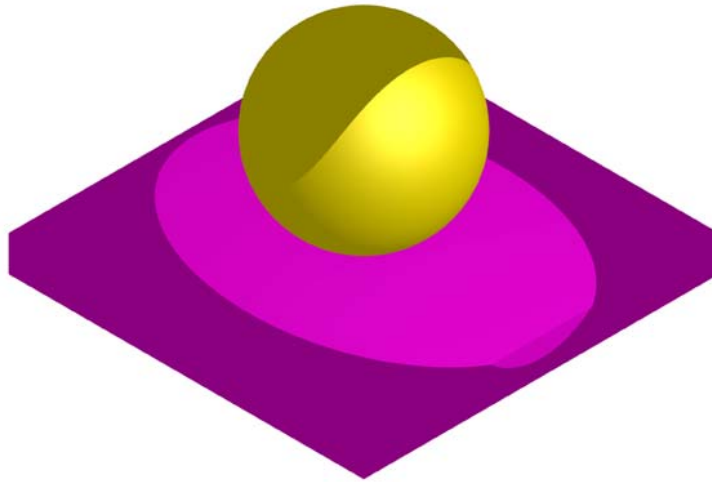
Pontszerű fényforrást reflektorra is tehetünk. Ez egy fénykúp megadását jelenti (mintha egy ernyőt helyeznénk a fényforrásra). A reflektorhatáshoz meg kell adnunk a fénykúp tengelyének irányát és fél nyílásszögét, továbbá azt, hogy a fény intenzitása hogyan csökken a kúp tengelyétől a perlel felé haladva. Reflektor fényforrásból csak akkor jut el fény az ábrázolt felület valamely pontjára, ha a pontot és a fényforrást összekötő egyenes és a fénykúp tengelyének δ szöge kisebb, mint a kúp β fél nyílásszöge (7.3. ábra), vagyis ha $\delta < \beta$. Az I_l fényt kibocsátó reflektornak a δ függvényében bekövetkező intenzitáscsökkenését a

$$I_l \cos^r \delta$$

kifejezéssel írhatjuk le, ahol r nemnegatív egész szám. $r = 0$ esetén a fény intenzitása nem változik a kúpon belül.



7.3. ábra. Reflektor fényforrás



7.4. ábra. Reflektor fényforrás hatása

7.3. Anyagtulajdonságok

A felületi pontok színének meghatározásában a fényforrások mellett fontos szerepe van a megvilágított felület anyagtulajdonságainak. Az ábrázolás szempontjából a legfontosabb optikai anyagjellemzők:

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

A fényvisszaverési tulajdonságokat az alábbi konstansokkal adhatjuk meg:

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

Ezek a konstansok tapasztalati mennyiségek, nem azonosíthatók a felület anyagának valamely fizikai jellemzőjével.

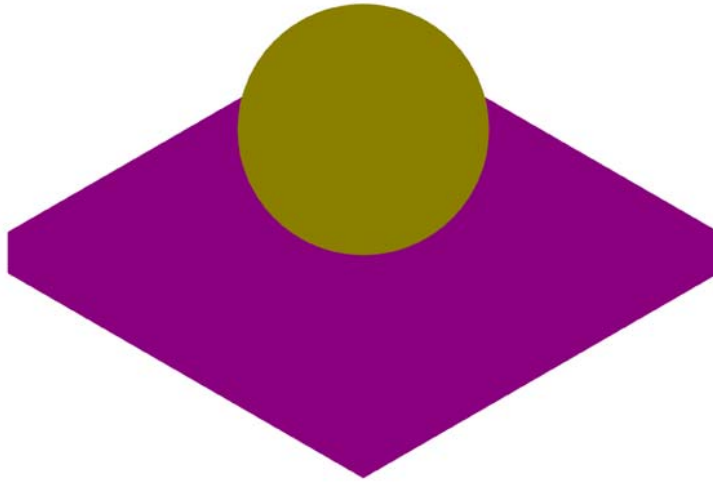
Egy felületi pontból a szembe eljutó környezeti fény

$$E_a = k_a * I_a$$

alakban írható fel, ahol I_a a megjelenített térrészben lévő összes környezeti fény (a fényforrásoktól független és a fényforrásoktól függők), k_a pedig a felület környezeti fény visszaverődési

együtthatója. \mathbf{k}_a komponensei (RGB) a $[0., 1.]$ intervallumbeli valós számok lehetnek. A továbbiakban a $*$ operátor azt a speciális vektorműveletet jelöli, amely két vektorhoz olyan harmadik vektort rendel, amelynek komponensei az eredeti vektorok megfelelő komponenseinek a szorzata, tehát a $\mathbf{k}_a = [k_{a_R} \ k_{a_G} \ k_{a_B}]^T$ és $\mathbf{I}_a = [I_{a_R} \ I_{a_G} \ I_{a_B}]^T$ vektorok $*$ szorzata

$$\mathbf{k}_a * \mathbf{I}_a = \begin{bmatrix} k_{a_R} I_{a_R} \\ k_{a_G} I_{a_G} \\ k_{a_B} I_{a_B} \end{bmatrix}.$$

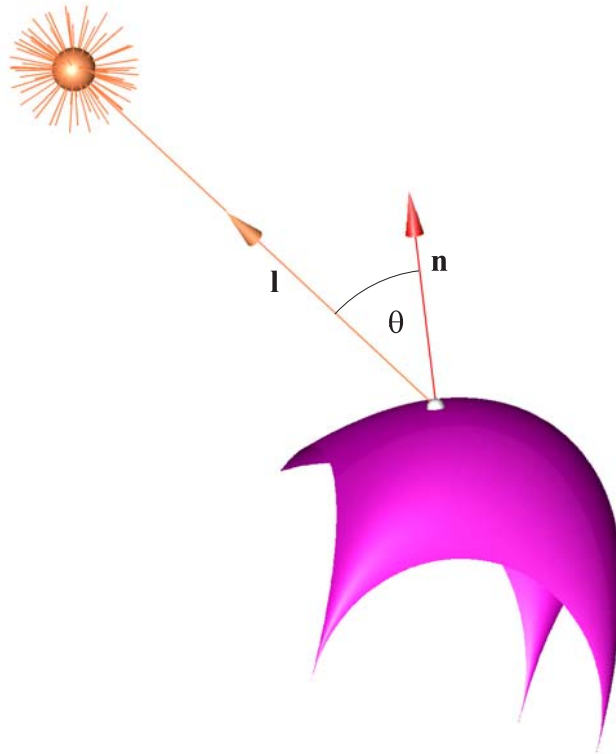


7.5. ábra. Csak környezeti fénnel megvilágított objektumok

Egy felületi pontból a szembe eljutó szórt visszaverődési fény mennyisége függ a fényforrás I_l szórt fénykomponensétől, a fényforrásból a felületi pontba beérkező fénysugárnak a felületi normálissal bezárt θ szögétől (7.6. ábra), valamint a k_d felületi konstanstól. Az összefüggés

$$E_d = k_d * I_l \cos \theta, \quad \cos \theta = \frac{\mathbf{l} \cdot \mathbf{n}}{|\mathbf{l}| |\mathbf{n}|}$$

alakban írható fel. A szórt visszaverődés tehát független a nézési iránytól, és csak akkor vesszük figyelembe, ha $\theta \in [0, \pi/2]$. A k_d konstans vektort a felület szórt visszaverődési együtthatójának nevezzük, komponensei a $[0, 1.]$ intervallumbeli értékek lehetnek.



7.6. ábra. Szórt visszaverődés

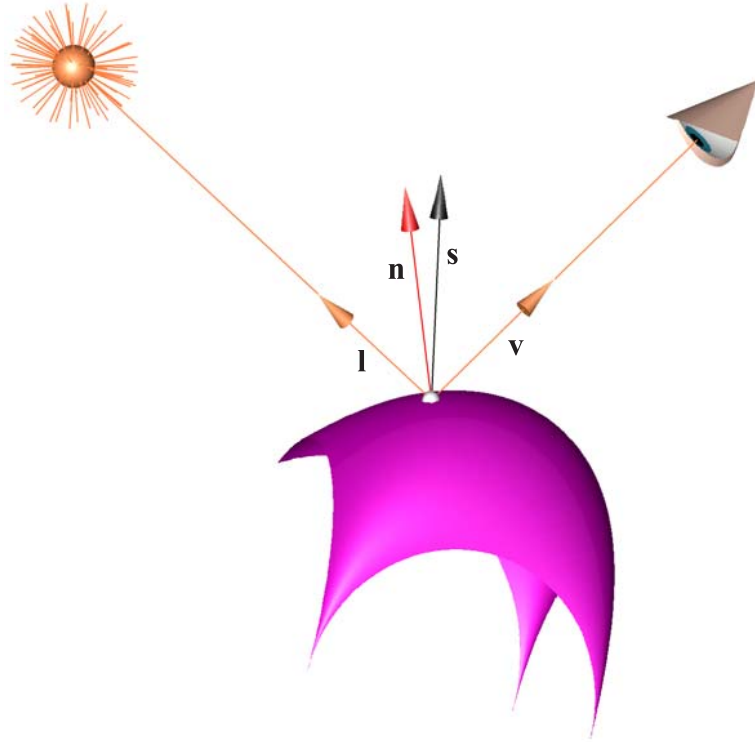
A tükrözött fényvisszaverődés a nézési iránytól is függ, hiszen egy ragyogó tárgyat nézgetve azt tapasztaljuk, hogy a fejünket mozgatva a fényes terület is mozog. Ezen kívül szerepet játszik még a csúcspontbeli normális, a fényforrás pozíciója, az anyag tükröző visszaverődési együtthatója és az anyag ragyogási tényezője. A 7.7. ábra jelöléseit használva, a szembe eljutó tükrözött fény

$$S = \left\{ \begin{array}{ll} 0, & \text{ha } \mathbf{l} \cdot \mathbf{n} \leq 0 \\ \max \left\{ \left(\frac{\mathbf{s} \cdot \mathbf{n}}{|\mathbf{s}| |\mathbf{n}|} \right)^e, 0 \right\} \mathbf{I}_s * \mathbf{k}_s, & \text{ha } \mathbf{l} \cdot \mathbf{n} > 0 \end{array} \right\}$$

ahol

$$\mathbf{s} = \mathbf{l}/|\mathbf{l}| + \mathbf{v}/|\mathbf{v}|,$$

I_s a csúcspontba eljutó tükrözött fény, k_s az anyag tükröző visszaverődési együtthatója, e pedig az anyag ragyogási kitevője. Matt felülethez kis, csillogó felülethez pedig nagy e érték tartozik. Egy tökéletes fényvisszaverő felület esetén e értéke végtelen lenne.



7.7. ábra. Tükröző visszaverődés

Több fényforrás esetén összegezni kell a fényösszetevőket. Tehát n fényforrás esetén

$$\mathbf{V} = \mathbf{I}_l + \mathbf{I}_g * \mathbf{k}_a + \sum_{i=0}^{n-1} T_i R_i (\mathbf{A}_i + \mathbf{D}_i + \mathbf{S}_i)$$

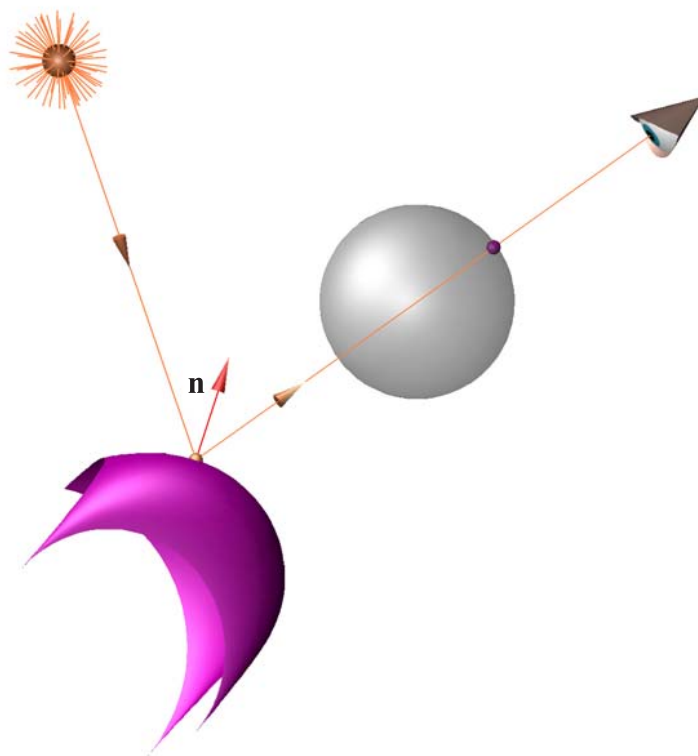
alakban írható fel, ahol

- \mathbf{I}_l az objektum által kibocsátott fény;
- \mathbf{I}_g a globális környezeti fény;
- T_i az i -edik fényforrás tompulása;
- R_i az i -edik fényforrás reflektor együtthatója;
- $\mathbf{A}_i = \mathbf{I}_{ai} * \mathbf{k}_a$ az i -edik fényforrás környezeti fénykomponensének és az anyag környezeti fény visszaverődési együtthatójának a szorzata;
- \mathbf{D}_i az i -edik fényforrásból a szembe eljutó szórt fény;
- \mathbf{S}_i az i -edik fényforrásból a szembe eljutó tükrözött fény.

7.4. Fényképszerű képek létrehozása

A térbeli objektumok ábrázolása során a valószerűség fokozása érdekében további anyagtulajdonságokat vehetünk figyelembe, és modellezhetjük az objektumok kölcsönhatását is. Így igen jó minőségű, fényképszerű (photo realistic) képeket hozhatunk létre.

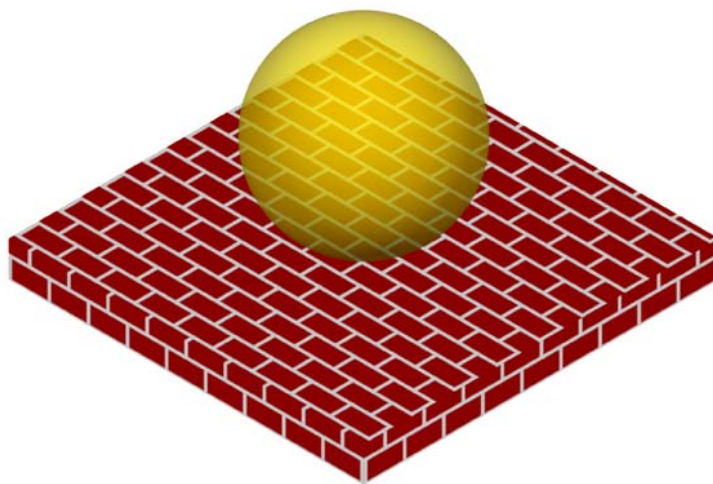
7.4.1. Átlátszóság



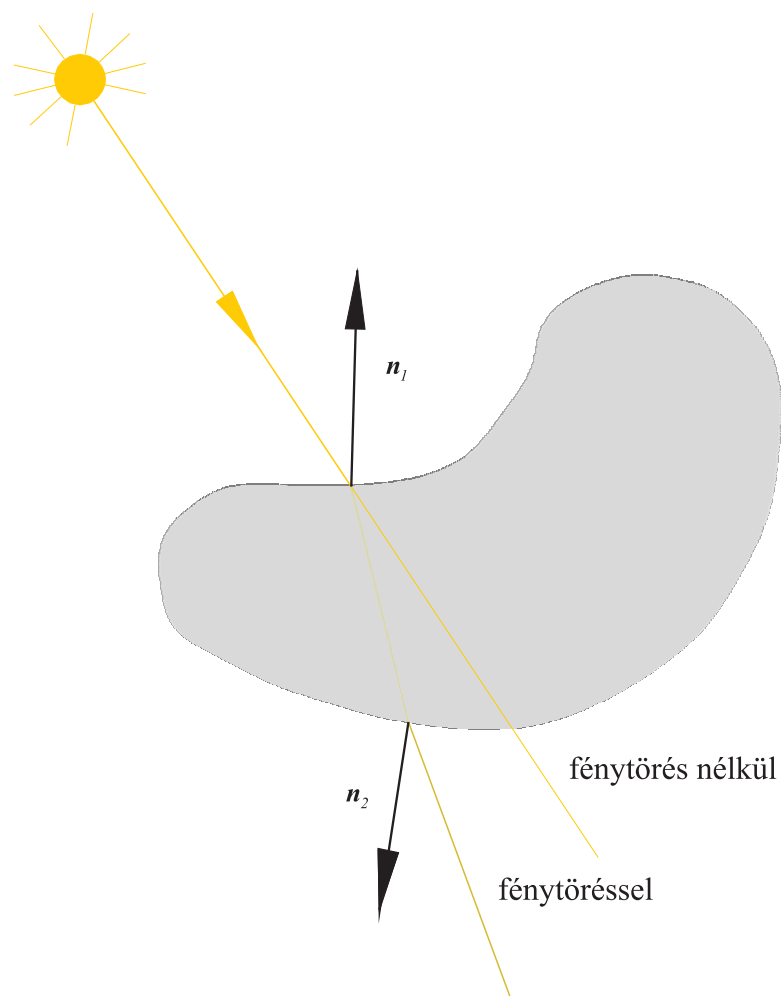
7.8. ábra. Átlátszóság vizsgálata

Az eddigi anyagtulajdonságok mellett figyelembe vehetjük a testek átlátszóságát is (7.8. ábra). Kétféle átlátszóságot különböztethetünk meg, a fénytörés mentes és a szórásmentes fénytöréssel járó. Bár a fénytörés nélküli átláthatóság - a fénysugár irányváltoztatás nélkül halad át a testen - nem valószerű, sok CAD alkalmazás számára kedvező, például amikor egy autó karosszériájának egy része átlátszó, és így látjuk az ülések elhelyezkedését is (ezt szokták röntgenképnek is nevezni).

A fénytöréssel járó átlátszóság esetén a fénysugár a testbe való be- és kilépéskor megváltoztatja irányát (7.10. ábra). Ennek modellezése sokkal összetettebb feladat. Mindkét esetben meg kell adni az átlátszóság mértékét, az úgynevezett átlátszósági együtthatót, ami egy valós szám a $[0., 1.]$ intervallumból. Ez egyetlen skalár, és minden színkomponensre (RGB) ezt alkalmazzuk.



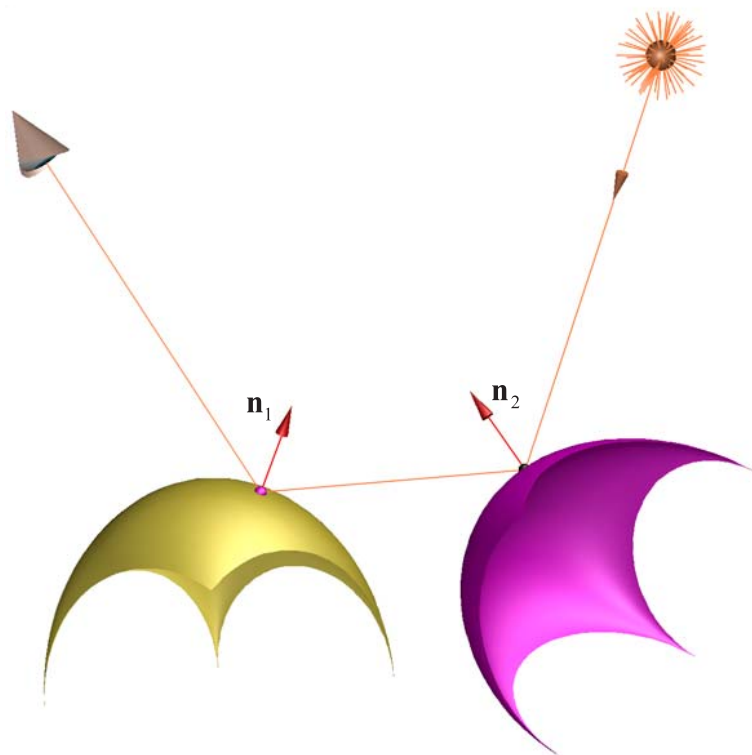
7.9. ábra. A gömbön átlátszik a talapzat mintázata



7.10. ábra. Átlátszóság fénytöréssel és fénytörés nélkül

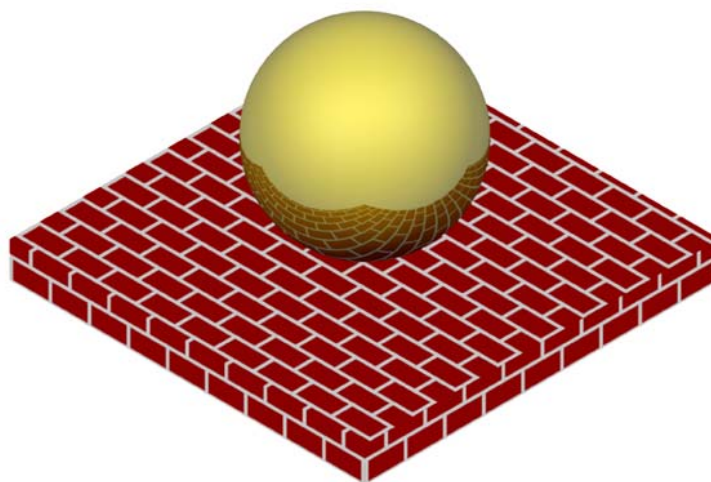
7.4.2. Árnyék és tükröződés

A valószerűségnek egy igen fontos eleme az ábrázolandó térrészben elhelyezkedő objektumok optikai kölcsönhatásai. Két ilyen effektusra térünk ki, ezek az árnyék és az objektumok tükröződése egymáson. Tükröződés esetén azt vizsgáljuk, hogy az adott felületi pontban valamely más test látszik-e (tükröződik-e), lásd a 7.11. ábrát. Ennek érdekében meg kell határozni a tükröződő sugárnak (félegyenesnek) az objektumokkal alkotott metszéspontjait, és ezek közül ki kell választani a kiindulási ponthoz legközelebbit.

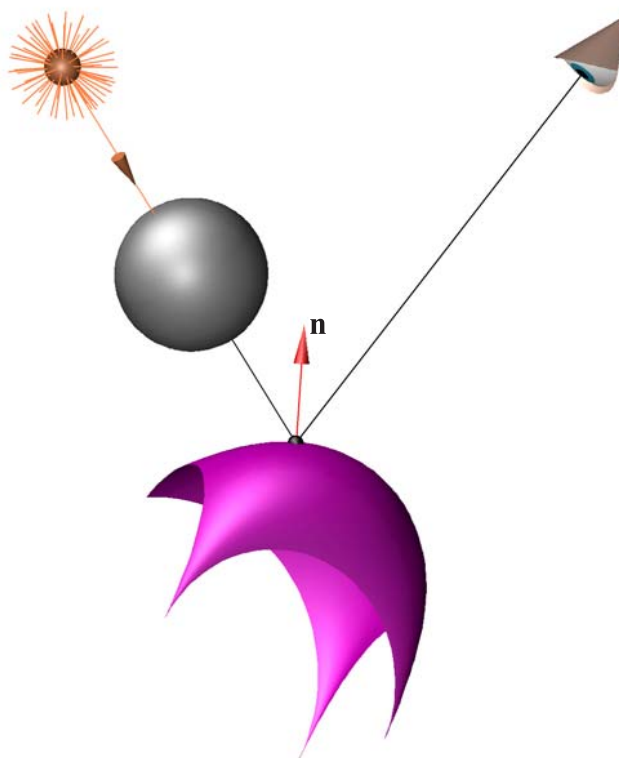


7.11. ábra. Tükröződés modellezése

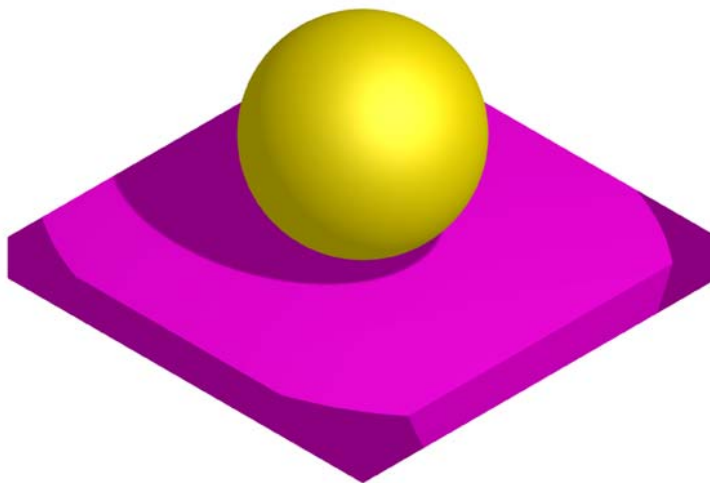
Árnyék esetén azt kell vizsgálnunk, hogy a fényforrás és a vizsgált alakzat között van-e valamilyen tárgy (7.13. ábra).



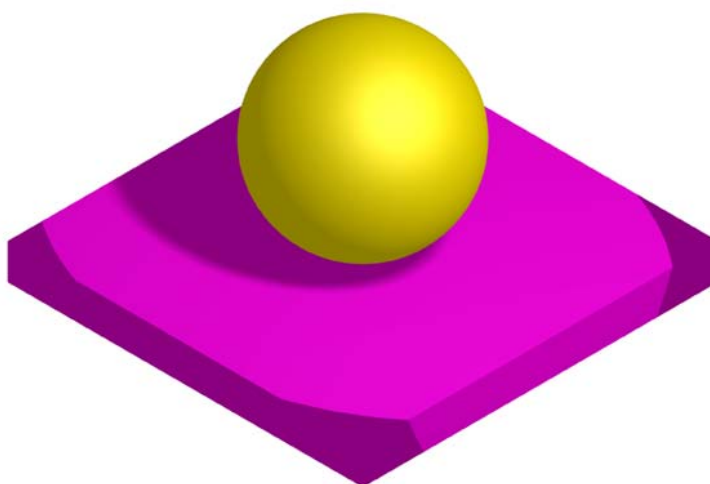
7.12. ábra. A gömbben tükröződik a talapzat



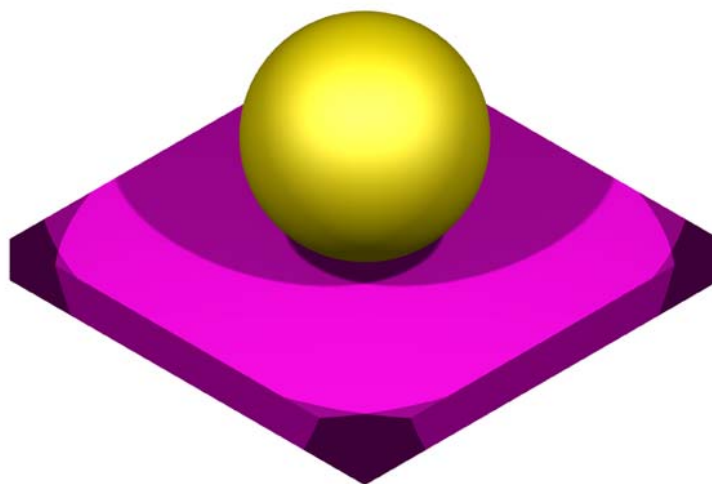
7.13. ábra. Árnyékvizsgálat



7.14. ábra. Az árnyék határa kontrasztos



7.15. ábra. Az árnyék és a környezet közötti átmenet fokozatos (soft shadow)

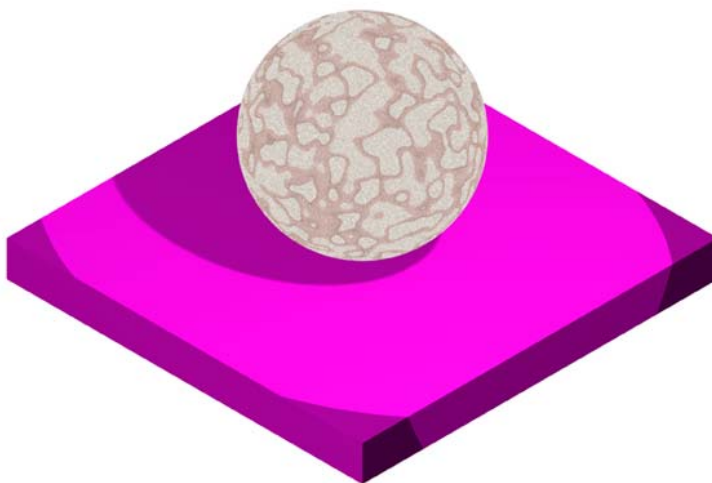


7.16. ábra. Két reflektor hatására keletkező árnyék

7.4.3. Textúra

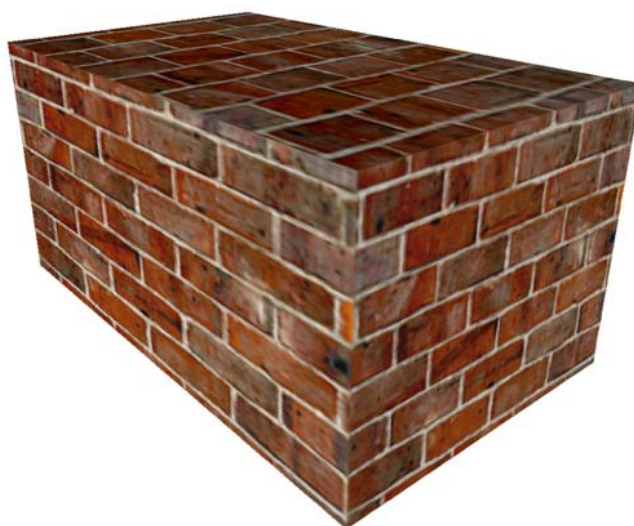
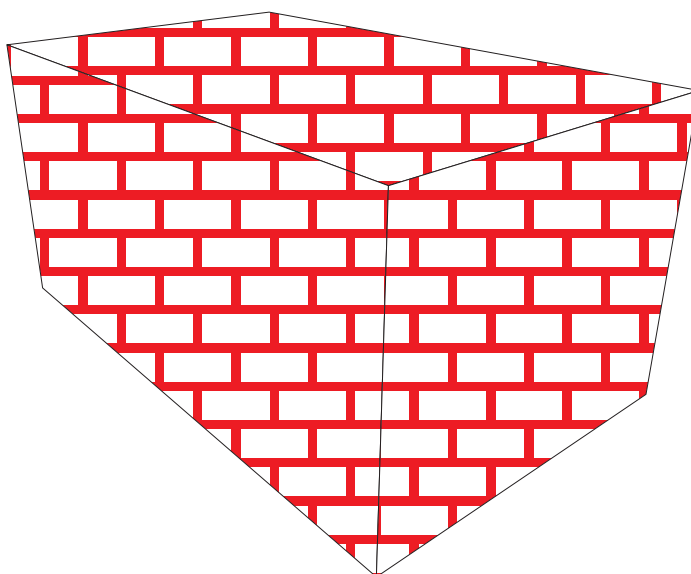
Az eddigiek során azt feltételeztük, hogy az ábrázolandó objektum homogén színű, és a fényviszonyoknak megfelelően az objektum képére illeszkedő pixelek színét ezen színből kiindulva kell meghatározni. A valóságban azonban a testeken lehetnek minták, illetve a testnek az anyagára jellemző mintázata van. A megjelenítés során ezt is célszerű figyelembe venni. Ezt az eljárást textúraleképezésnek nevezzük. A textúra a felület mintázatát tartalmazó kép, amit ismételve fel kell feszíteni a felületre.

Egyes rendszerek többretegű, az egyes rétegek átlátszóságának mértékét is megadó úgynevezett 3D-s textúrákat is megengednek. A textúrát előállíthatjuk valamilyen algoritmussal, de egy fénykép raszterizálásával kapott pixeltömbbel is. A 7.17. ábrán a gömb textúrázott.



7.17. ábra. Textúrázott gömb

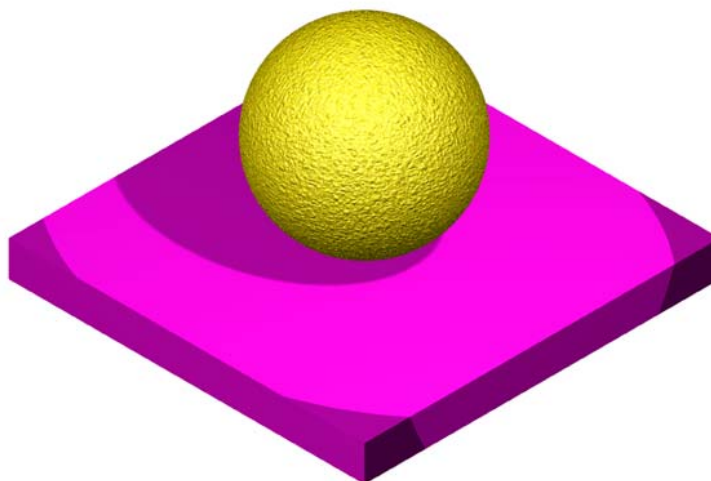
A textúrázás és a poligonok képének mintával való kitöltése között óriási a különbség, lásd a 7.18. ábrát. Kitöltéskor a poligonok képét töltjük ki mintával, a textúrázásnál viszont a térben visszük rá a felületre a mintázatot (mintha matricát ragasztanánk rá, vagy tapétáznánk), majd a textúrával együtt ábrázoljuk az objektumot. Ennek az a következménye, hogy a textúrázás adott esetben felerősíti a perspektív torzítást, tehát hangsúlyozza az alakzat térbeliségét, a mintával való kitöltés viszont rontja ezt a hatást.



7.18. ábra. Kitöltés mintával és textúrázás

7.4.4. Felületi érdesség

A számítógéppel létrehozott képek általában sima, fényes, fémesen csillogó testeket, felületeket ábrázolnak. Bár a valóságban is vannak ilyen testek, de nem jellemzőek, ezért a képek valószerűségének fokozása érdekében célszerű a felületi érdességet is szemléltetni. A felületi érdesség geometriai értelemben azt jelenti, hogy például a határoló lap nem sík, hanem a síkból kisebb-nagyobb besüllyedéssel, kitüremkedéssel létrehozott görbült felület. A felületi érdesség geometriai modellezése nem célszerű, hiszen a síktól csak kis mértékben eltérő, a számítások során síknak tekinthető lapok görbült felületként való leírása rendkívül elbonyolítaná a legegyszerűbb feladatok megoldását is. Éppen ezért a modellezés során a felületi érdességet csak egy attribútumként adjuk meg és tároljuk az adatbázisban, és csak a szemléltetés során vesszük figyelembe.



7.19. ábra. Érdesített gömb

A felületi érdesség szemléltetését (bump mapping) általában úgy érjük el, hogy a normálisokat megváltoztatva (mintegy felborzolva), a felületről visszavert fény irányát változtatjuk, aminek vizuális hatása a felület érdessége lesz (7.19. ábra). A felületi normálisok változtatása történhet akár véletlenszerűen is, de a textúrához hasonlóan valamilyen algoritmussal, bit-térképpel is.

8. fejezet

Görbék leírása és megjelenítése

A 4. fejezetben ismertettük a grafikus alapelemeket, amiből a számítógépen a képek felépíthetők. Ebben a fejezetben görbék leírásával és megjelenítésével foglalkozunk.

8.1. Görbék leírása

Görbék leírására alapvetően három lehetőség van:

- az $y = f(x)$ úgynevezett explicit,
- az $F(x, y) = 0$ implicit,
- és az $\mathbf{r}(t)$, $t \in [a, b]$ paraméteres leírási mód.

Ezek közül az első kettő csak síkgörbék, a harmadik mind sík-, mind térgörbék leírására alkalmas. A legegyszerűbb görbe, az egyenes esetén ezek a jól ismert

$$y = mx + b;$$
$$ax + by + c = 0$$

és

$$\mathbf{r}(t) = \mathbf{a} + t\mathbf{d}$$

alakok.

A számítógéppel segített tervezés számára a paraméteres leírás a legfontosabb. Az implicit leírást sokkal ritkábban használjuk, az explicitet pedig alig, mert ez nagyon nagy mértékben függ a koordináta-rendszer-től. A paraméteres leírás azért is hasznos, mert mind síkgörbe, mind térgörbe leírható vele. A paraméteres és implicit leírások közötti áttérés általában nem lehetséges. Nagyon jó lenne, ha mindig át lehetne térni, mert mindkét leírási módnak vannak előnyei és hátrányai. Annak eldöntésére például, hogy egy pont illeszkedik-e a görbére az implicit alak a legalkalmasabb, a kirajzoláshoz viszont a paraméteres.

Az $\mathbf{r}(t) = [x(t) \ y(t) \ z(t)]^T$, $t \in [a, b] \subset \mathbb{R}$ skalár-vektor függvénnyel megadott pontok halmazát (differenciálgeometriai értelemben) görbének nevezzük, ha az $\mathbf{r}(t)$ függvény

- kölcsönösen egyértelmű és mindkét irányban folytonos;
- t szerint folytonosan differenciálható;

- a differenciálhányados sehol sem tűnik el.

Az utóbbi tulajdonság tehát azt jelenti, hogy $\dot{\mathbf{r}}(t) = [\dot{x}(t) \ \dot{y}(t) \ \dot{z}(t)]^T \neq \mathbf{0}, \forall t \in [a, b]$. A második és harmadik feltétel biztosítja az érintő létezését a görbe tetszőleges pontjában. Ezt az előállítást reguláris előállításnak is szokták nevezni. Ha a harmadik komponens nulla, akkor síkgörbe leírását kapjuk. A továbbiakban, ha külön nem szólunk róla, a görbe lehet mind sík- mind térgörbe.

Egy görbének lehet többféle paraméteres alakja is. Tekintsük ugyanis az $\mathbf{r}(t), t \in [a, b]$ görbét, valamint a $t = t(u), u \in [c, d]$ függvényt. Feltételezzük továbbá, hogy a $t = t(u)$ függvény kölcsönösen egyértelmű, mindkét irányban folytonos megfeleltetést hoz létre az $[a, b]$ és $[c, d]$ intervallumok között, $t(u)$ folytonosan differenciálható és $\frac{d}{du}t(u) \neq 0, \forall u \in [c, d]$. Ezen feltételek teljesülése esetén az $\mathbf{r}(u) = \mathbf{r}(t(u))$ skalár-vektor függvény eleget tesz a görbe definíciójában megkövetelt feltételeknek, továbbá az $\mathbf{r}(u)$ ugyanazt a görbét írja le, mint $\mathbf{r}(t)$. A paraméternek a fenti tulajdonsággal rendelkező transzformációját megengedett paramétertranszformációnak nevezzük.

A paramétertranszformációt leíró függvény a fenti feltételek miatt szigorúan monoton, differenciálható függvény, melynek differenciálhányadosa nem tűnik el. A görbét növekvő paraméterértékek szerint bejárva, a görbe egy irányítását határozzuk meg. Ha a paramétertranszformációt leíró függvény szigorúan monoton növekvő, akkor a görbe megtartja irányítását, ha szigorúan monoton csökkenő, akkor az irányítás ellentétes lesz. A számítógépi grafikában és a számítógéppel segített geometriai tervezésben gyakran használjuk azt a paramétertranszformációt, amikor az $[a, b]$ intervallumról a $[c, d]$ intervallumra a

$$t(u) = \frac{b-a}{d-c}(u-c) + a, u \in [c, d]$$

függvénnyel térünk át. A továbbiakban felsorolt görbetulajdonságok, görbéket jellemző mennyiségek függetlenek a parametrizálástól, azaz a megengedett paramétertranszformációval szemben invariánsak, tehát a görbe belső tulajdonságait fejezik ki.

8.1.1. Ívhossz

Tekintsük az $\mathbf{r}(t), t \in [a, b]$ görbét, osszuk fel az $[a, b]$ intervallumot n egyenlő részre! Legyen $t_i = a + i\Delta t, \Delta t = (b-a)/n$ és $\Delta \mathbf{r}_i = \mathbf{r}(t_{i+1}) - \mathbf{r}(t_i)$! A t_i paraméterértékekhez tartozó \mathbf{r}_i görbepontokat kössük össze egyenes szakaszokkal! A görbébe ilyen módon beleírt töröttvonal hossza

$$s_n = \sum_{i=0}^n |\Delta \mathbf{r}_i| = \sum_{i=0}^n \left| \frac{\Delta \mathbf{r}_i}{\Delta t} \right| \Delta t.$$

Az n értékét növelve s_n egyre jobban megközelíti a görbe ívének a hosszát, vagyis $\lim_{n \rightarrow \infty} s_n = s$, ahol s a görbe ívhossza. Belátható, hogy az $\mathbf{r}(t), t \in [a, b]$ görbe a és t paraméterei közé eső ívének a hossza

$$s(t) = \int_a^t |\dot{\mathbf{r}}(t)| dt$$

alakban írható fel.

Az $\mathbf{r}(t)$ függvényre szabott feltételek miatt az $s(t)$ függvény eleget tesz a megengedett paramétertranszformációval szemben támasztott követelményeknek, ezért a $t(s) = s^{-1}(t)$ paramétertranszformációt végrehajtva a görbe paramétere az ívhossz lesz. Az ívhossz paramétert természetes paraméternek is szokás nevezni, mert ez a görbére jellemző, annak belső geometriai tulajdonságát kifejező paraméter. A továbbiakban az ívhossz-paraméter szerinti deriváltat vesszővel, a tetszőleges paraméter szerinti deriváltat pedig ponttal jelöljük. Az ívhossznak, mint paraméternek elsősorban az elméleti vizsgálódásoknál van fontos szerepe. A gyakorlat szempontjából nem játszik központi szerepet, mivel az $s^{-1}(t)$ függvény általában nem állítható elő zárt alakban. Az ívhosszat is többnyire a beírt töröttvonal hosszával közelítjük az egzakt formula kiértékelése helyett.

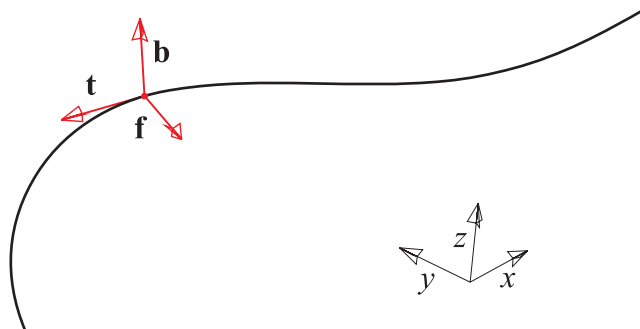
8.1.2. Kísérő triéder

A görbéknek megkülönböztetünk úgynevezett lokális - a görbe egy pontjának valamely környezetében érvényes - és globális - a görbe egészére érvényes - tulajdonságait. A görbék lokális tulajdonságainak vizsgálatához gyakran használunk egy speciális koordináta-rendszert, az úgynevezett kísérő triédert, mely a görbén pontról pontra változik.

Feltételezzük, hogy az $\mathbf{r}(t)$ függvény kétszer folytonosan differenciálható és $\ddot{\mathbf{r}}(t) \neq \mathbf{0}, \forall t \in [a, b]$. Ekkor a görbe tetszőleges $\mathbf{r}(t)$ pontjában meghatározhatjuk a

$$\begin{aligned} \mathbf{t} &= \frac{\dot{\mathbf{r}}(t)}{|\dot{\mathbf{r}}(t)|}, \\ \mathbf{b} &= \frac{\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)}{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|}, \\ \mathbf{f} &= \mathbf{b} \times \mathbf{t} \end{aligned}$$

vektorokat.



8.1. ábra. Térgörbe kísérő triédere

A $\mathbf{t}, \mathbf{f}, \mathbf{b}$ páronként merőleges egységvektorok egy derékszögű Descartes-féle koordináta-rendszert határoznak meg, ahol \mathbf{t} az egységnyi hosszúságú érintővektor, \mathbf{f} -et főnormálisnak, \mathbf{b} -t pedig binormálisnak nevezzük. A $\mathbf{t}, \mathbf{f}, \mathbf{b}$ vektorhármast kísérő triédernek nevezzük (8.1. ábra), a \mathbf{t} és \mathbf{f} által kifeszített sík a simulósík, a \mathbf{f} és \mathbf{b} vektorok által meghatározott sík a normálsík, a \mathbf{b} és \mathbf{t} által meghatározott pedig a rektifikáló sík.

Ha a paraméter az ívhossz, akkor

$$\begin{aligned} \mathbf{t} &= \mathbf{r}'(s), \\ \mathbf{f} &= \frac{\mathbf{r}''(s)}{|\mathbf{r}''(s)|}, \\ \mathbf{b} &= \mathbf{t} \times \mathbf{f}. \end{aligned}$$

8.1.3. Görbület, simulóköör

Egy egyenes érintővektorai egymással párhuzamos, azonos irányítású vektorok. Az összes többi görbénél azonban ez nem így van, vagyis a görbén haladva az érintővektorok különböző irányúak. A görbe egyenestől való eltérése sebességének egy mérőszáma a görbület.

Tekintsük az $\mathbf{r}(s)$ görbét (s az ívhossz szerinti paraméter) és legyen t_0 az s_0 , \mathbf{t} pedig az s paraméterértékhez tartozó pontban az érintővektor! Jelölje $\Delta\alpha$ a t_0 és \mathbf{t} vektorok szögét és legyen $\Delta s = |s - s_0|$! Ekkor a

$$\lim_{s \rightarrow s_0} \frac{\Delta\alpha}{\Delta s} = \kappa(s_0)$$

mennyiséget az $\mathbf{r}(s)$ görbe s_0 -beli görbületének nevezzük. Belátható, hogy ha a paraméter az ívhossz, akkor

$$\kappa(s) = |\mathbf{r}''(s)|,$$

tetszőleges t paraméter esetén pedig

$$\kappa(t) = \frac{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|}{|\dot{\mathbf{r}}(t)|^3}.$$

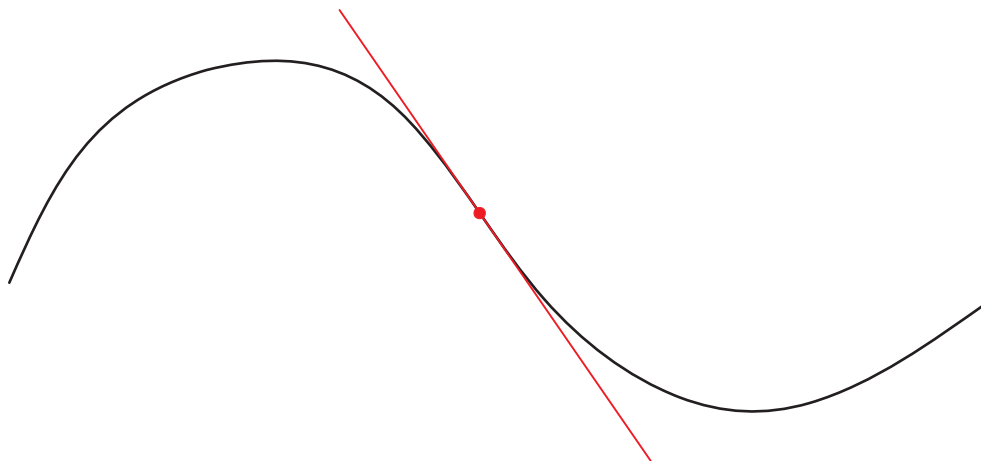
A görbület természetesen a paraméterezéstől független geometriai jellemzője a görbének (pontosabban a görbének az adott pontjában), és egybevágósági transzformációkkal szemben invariáns. Más megközelítésben azt mondhatjuk, hogy az s paraméter szerint végighaladva az $\mathbf{r}(s)$ görbén, a görbe $\mathbf{r}'(s)$ érintővektora a mindenkor binormális vektor körül $\kappa(s)$ szögsebességgel forog. A görbület definíciójából következik, hogy az egyenes görbülete azonosan nulla. A síkgörbék közül csak az egyenesnek és a körnek konstans a görbülete.

A síkgörbék el nem tűnő görbületéhez előjelet is rendelhetünk. Ennek egyik kézenfekvő módja, hogy az $\mathbf{r}(t) = \begin{bmatrix} r_x(t) & r_y(t) & 0 \end{bmatrix}^T$ síkgörbe

$$\kappa(t) = \frac{|\dot{r}_x(t)\ddot{r}_y(t) - \ddot{r}_x(t)\dot{r}_y(t)|}{|\dot{\mathbf{r}}(t)|^3}$$

görbületének számlálójában elhagyjuk az abszolút értéket, vagyis a görbület előjele a $\det(\dot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t))$ előjele legyen.

Az $\mathbf{r}(t)$ síkgörbe valamely t_0 paraméterértékhez tartozó $\mathbf{r}(t_0)$ pontját a görbe inflexiós pontjának nevezzük, ha a görbe t_0 belüli érintőjét a görbe $\mathbf{r}(t_0)$ -ban metszi. Könnyen belátható, hogy az $\mathbf{r}(t_0)$ pont pontosan akkor inflexiós pontja a görbének, ha a pontbeli görbület nulla és a pont környezetében a görbület előjelet vált. Szemléletesen azt mondhatjuk, hogy az inflexiós pontban az érintő mindkét oldalán van görbe, bármilyen kicsiny környezetét is vesszük a pontnak (8.2. ábra).



8.2. ábra. Síkgörbe inflexiós pontja

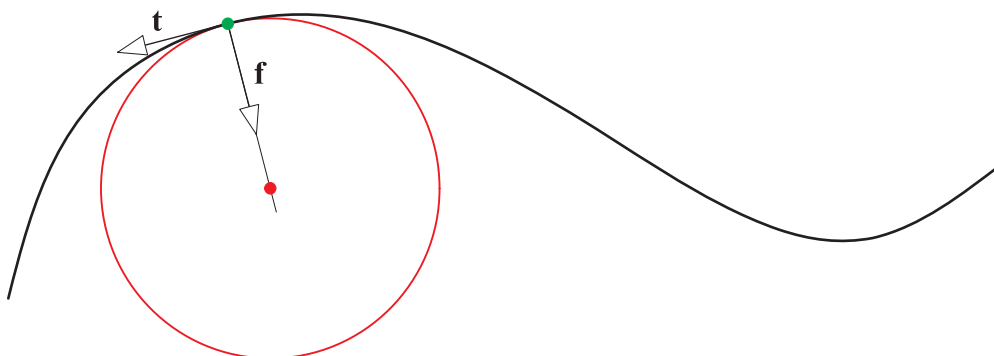
Az $\mathbf{r}(t)$ görbe t paraméterhez tartozó pontjában a görbe simulóköre az a kört értjük, mely síkja a görbe t -beli simulósíkja, $\rho(t)$ sugara pedig a t beli görbület reciproka, azaz

$$\rho(t) = \frac{1}{\kappa(t)}$$

és $\mathbf{c}(t)$ középpontjára

$$\mathbf{c}(t) = \mathbf{r}(t) + \rho(t) \mathbf{f}(t)$$

teljesül (8.3. ábra). A simulókör tehát csak akkor értelmezhető, ha a görbület nem nulla; illetve nulla görbület esetén azt mondhatjuk, hogy a simulókör sugara végtelen, vagyis a körből egyenes lesz.



8.3. ábra. A simulókör középpontja a görbe főnormálisán van

8.1.4. Torzió

A síkgörbék binormálisai párhuzamosak, azaz a simulósíkjai egybeesnek, a térgörbe simulósíkjai azonban folytonosan változnak, tehát binormálisai nem párhuzamosak. Egy görbe adott pontjában a síkgörbétől való eltérés sebességének mérőszáma a torzió.

Tekintsük a háromszor folytonosan differenciálható $\mathbf{r}(s)$ görbét (s az ívhossz szerinti paraméter), jelölje $\Delta\beta$ a $\mathbf{b}(s_0)$ és $\mathbf{b}(s)$ binormálisok szögét, és legyen $\Delta s = |s_0 - s|$. Ekkor a

$$\lim_{s \rightarrow s_0} \frac{\Delta\beta}{\Delta s} = |\tau(s_0)|$$

menyiséget az $\mathbf{r}(s)$ görbe s_0 -beli torziójának nevezzük.

Belátható, hogy ívhossz paraméter esetén

$$\tau(s) = \frac{(\mathbf{r}'(s), \mathbf{r}''(s), \mathbf{r}'''(s))}{\kappa^2(s)} = \frac{(\mathbf{r}'(s), \mathbf{r}''(s), \mathbf{r}'''(s))}{|\mathbf{r}''(s)|^2}$$

($(\mathbf{r}', \mathbf{r}'', \mathbf{r}''')$ a vektorok vegyes szorzatát jelöli), tetszőleges t paraméter esetén pedig

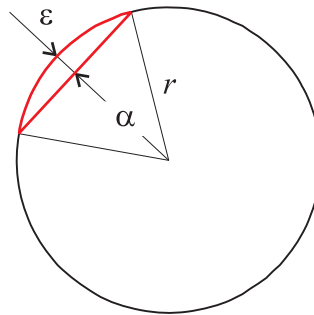
$$\tau(t) = \frac{(\dot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t), \dddot{\mathbf{r}}(t))}{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|^2}.$$

A síkgörbék torziója azonosan nulla. Ennek megfordítása is bebizonyítható, vagyis, ha egy görbe torziója minden pontban nulla, akkor a görbe síkgörbe. Mindössze három olyan görbe van, melynek mind a görbülete, mind a torziója konstans. Ezek az egyenes, a kör és a hengeres csavarvonal.

8.2. Görbék megjelenítése

A görbéket a beléjük írt töröttvonal segítségével ábrázoljuk, vagyis húrokkal közelítjük a görbét. A cél az, hogy a képernyőn szép sima vonalat lássunk, ne lehessen észrevenni azt, hogy szakaszokkal közelítettünk. Ennek érdekében kontrollálni kell az ív és a közelítő húr távolságát, másrészt az egymást követő szakaszok szögét. Az ív és a húr távolságának csökkentése javítja a kép minőségét. El kell azonban kerülnünk a húrok hirtelen irányváltását is, vagyis azt, hogy az egymást követő szakaszok nagy szöget zárjanak be.

A fenti célok elérése tetszőleges görbe esetén nem egyszerű feladat. A megoldás ugyanis a görbe görbületétől, pontosabban annak változásától függ. A konstans görbületű görbékkel nincs gond.

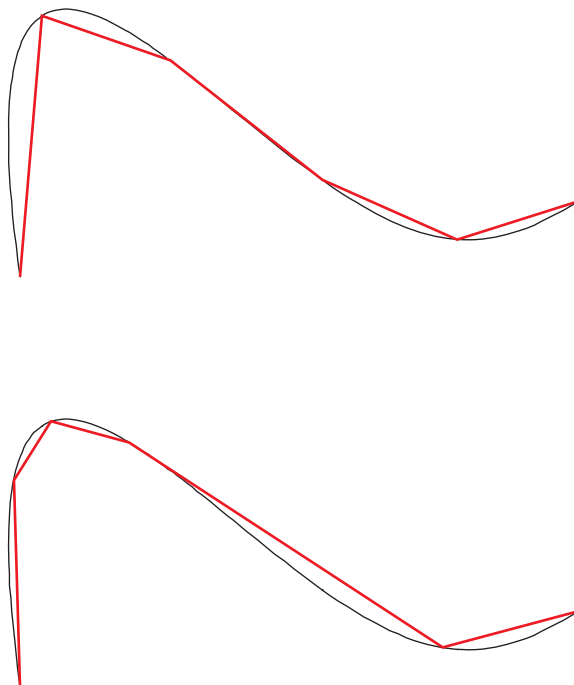


8.4. ábra. Kör közelítése előírt pontossággal

A kör előírt pontosságú rajzolását könnyű megvalósítani. Ha előírjuk ugyanis, hogy a közelítő húr és az ív távolsága ε legyen (8.4. ábra), akkor könnyen kiszámíthatjuk az ehhez tartozó α középponti szöget. A 8.4. ábrán látható, hogy

$$\varepsilon = r(1 - \cos(\alpha/2))$$

Ha tehát a kört (vagy körívet) olyan egyenlő ívekre osztjuk, melyek középponti szöge $\leq \alpha$, akkor a kívánt hatást érjük el.



8.5. ábra. Görbe közelítése beírt töröttvonallal; a felső képen az értelmezési tartományt egyenlő részekre osztottuk, az alsó képen figyelembe vettük a görbületet

Csak az egyenes és a kör konstans görbületű síkgörbe, vagyis általában nincs a kör esetéhez hasonló egyszerű eljárás. A paraméteres görbék esetén követhetjük azt az egyszerű utat, hogy az értelmezési tartományt n egyenlő részre osztjuk, és az osztáspontokhoz tartozó görbeponthoz kötjük össze szakaszokkal. n növelésével egyre jobb közelítést kapunk. Ez az eljárás működőképes, kielégítő eredményre vezet, de nem optimális. A gond az, hogy a görbület változásától függetlenül, a görbe minden részét egyre több és több részre osztjuk, amíg a kívánt eredményt el nem érjük. Az optimális eljárás figyelembe veszi a görbület változását, és a nagy görbületű részeket több, a kis görbületűeket kevesebb részre osztja. A 8.5. ábrán ugyanannak a görbének két közelítését láthatjuk, mindkét esetben öt szakaszt használtunk. A felső ábrát a paramétertartomány egyenlő részre osztásával kaptuk, az alsó ábránál figyelembe vettük a görbület változását.

Vannak olyan görbemegadási módok, amelyek megkönnyítik a görbék előírt pontosságú, optimális kirajzolását. Ez az a görbeosztály, amelynél a görbe egy poligonnal (a kontrollpoligonnal) adható meg, és a görbét a kontrollpoligon sarkainak ismételt levágásával kapott poligonok sorozatával egyre pontosabban közelítjük meg. Maga a görbe ezen eljárás határértékeként kapott görbe. (Ezeket a görbéket angolul összefoglalóan subdivision curves-nek nevezzük.)

Ilyen felosztási algoritmus a számítógépi grafikában és a geometriai modellezésben is rendkívül népszerű Bézier-görbékre is létezik, ez a de Casteljau-algoritmus.

Adottak a tér $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ pontjai és $t \in [0, 1]$ skalár. A pontokat összekötő töröttvonal $\mathbf{b}_i, \mathbf{b}_{i+1}$ oldalait osszuk fel $t : (1 - t)$ arányban, és az osztáspontokat jelöljük $\mathbf{b}_i^1(t)$ -vel! Így

a kiindulásnál eggyel kevesebb számú pontot kapunk. Az osztáspontok által meghatározott töröttvonal oldalait ismét osszuk fel $t : (1 - t)$ arányban, és ezt az eljárást addig folytassuk, míg egyetlen osztáspontot kapunk. Ezen rekurzív felosztás formális leírása

$$\mathbf{b}_i^r(t) = (1 - t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t);$$

$$(r = 1, \dots, n), (i = 0, 1, \dots, n - r),$$

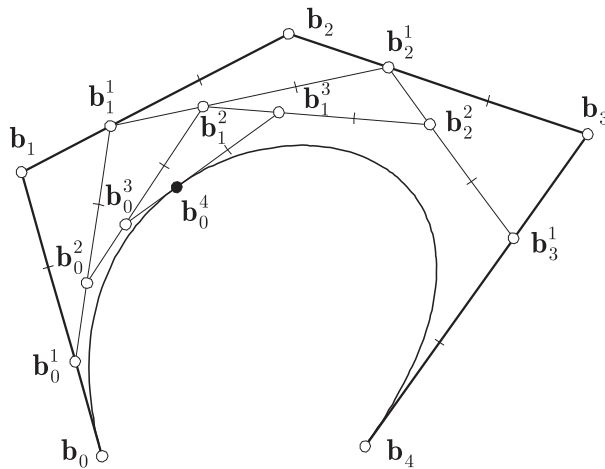
ahol

$$\mathbf{b}_i^0 = \mathbf{b}_i (i = 0, 1, \dots, n).$$

A $\mathbf{b}_0^n(t), t \in [0., 1.]$ pontok által meghatározott görbe a Bézier-görbe, magát a felosztási algoritmust de Casteljaú-algoritmusként nevezzük. A felosztás során kapott pontok a következő háromszög alakba rendezhetők

$$\begin{array}{ccccccc} & & & & & & \mathbf{b}_0 \\ \mathbf{b}_0 & & & & & & \\ \mathbf{b}_1 & \mathbf{b}_0^1(t) & & & & & \\ \mathbf{b}_2 & \mathbf{b}_1^1(t) & \mathbf{b}_0^2(t) & & & & \\ \vdots & & & \ddots & & & \\ & & & & \mathbf{b}_0^{n-1}(t) & & \\ \mathbf{b}_n & \mathbf{b}_{n-1}^1(t) & \mathbf{b}_{n-2}^2(t) & \dots & \mathbf{b}_1^{n-1}(t) & \mathbf{b}_0^n(t) \end{array}$$

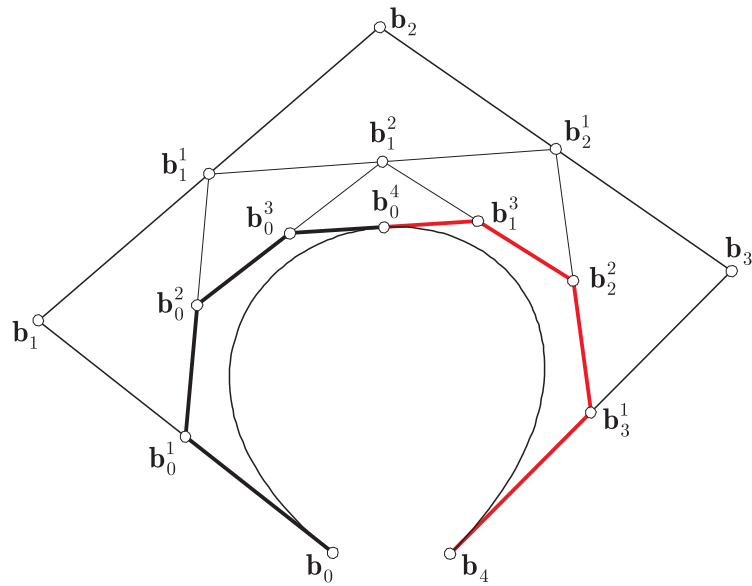
Az $n = 4$ esethez tartozó szerkesztést mutatja a 8.6. ábra.



8.6. ábra. A $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_4$ kontrollpontokkal adott Bézier-görbe $t = 1/3$ pontjának szerkesztése a de Casteljaú-algoritmussal

A de Casteljaú-algoritmussal nemcsak a Bézier-görbe pontja kapható meg, hanem a görbe két, az eredetivel megegyező fokszámú Bézier-görbére bontható. Az egyik a $[0., t]$ intervallum fölötti, amelynek kontrollpontjai $\mathbf{b}_0^j(t)$, $(j = 0, 1, \dots, n)$, a másik pedig, szimmetria okok miatt, a $[t, 1.]$ intervallum fölötti görbe, amelynek kontrollpontjai $\mathbf{b}_j^{n-j}(t)$, $(j = 0, 1, \dots, n)$. A 8.7. ábrán látható negyedfokú Bézier-görbét a 0.5 paraméterértéknél vágtuk ketté.

Ezt a tulajdonságot felhasználva az alábbi kirajzoló algoritmus hozható létre. Tekintsük a $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ kontrollpontokkal adott $\mathbf{b}(t)$ Bézier-görbét, és legyen adott az ε pontosság, ami



8.7. ábra. Bézier-görbe kettévágása a de Casteljau algoritmussal

a közelítő egyenes szakaszoknak (a húroknak) az általuk helyettesített ívtől mért távolságának a felső korlátja!

1. Ha $b_0 = b_n$, azaz a görbe zárt, akkor vágjuk ketté a görbét a de Casteljau-algoritmussal a $t = 1/2$ paraméterértéknél, és az így kapott két darab n -edfokú Bézier-görbére hajtjuk végre az algoritmus további lépéseit.
2. Vegyük a b_0 és b_n kontrollpontokat összekötő egyenest, és keressük meg a tőle legtávolabb fekvő b_j , $(0 < j < n)$ kontrollpontot, és ennek a távolságát jelöljük m_j -vel.
3. Ha $m_j \leq \varepsilon$, akkor a görbét a b_0, b_n szakasszal helyettesítjük. Egyébként osszuk fel a görbét a $t = j/n$ paraméterértéknél a de Casteljau-algoritmussal két n -edfokú Bézier-görbére, és mindkét görbére a 2. lépéstől folytassuk a feldolgozást.

Ez a módszer konvergens és gyors. A 3. lépésnél tulajdonképpen bármely $t \in (0, 1)$ értéknél kettévághatnánk a görbét, azonban, ha b_j van legtávolabb az egyenestől, akkor ő húzza el tőle leginkább a görbét. Ha pl. rasteres megjelenítőt használunk, akkor az $\varepsilon = 1$ pixel garantáltan sima, folytonos, esztétikus eredményt ad, és mindezt viszonylag kevés egyenes szakasszal éri el. Az algoritmus ugyanis a görbület változását figyelembe véve állítja elő a közelítő töröttvonalat.

9. fejezet

Felületek leírása és szemléltetése

Ebben a fejezetben röviden összefoglaljuk a felületek különböző leírásait, majd ismertetjük a felületek különböző szemléltetési lehetőségeit.

9.1. Felületek leírása

A felületek leírására is három alapelethezőség van. Ezek

- a $z = f(x, y)$ kétváltozós függvénnyel leírt, úgynevezett explicit felületleírás;
- az $F(x, y, z) = 0$ implicit;
- és az $s(u, v), (u, v) \in \mathbb{R}^2$ vektor-vektor függvénnyel leírt felület.

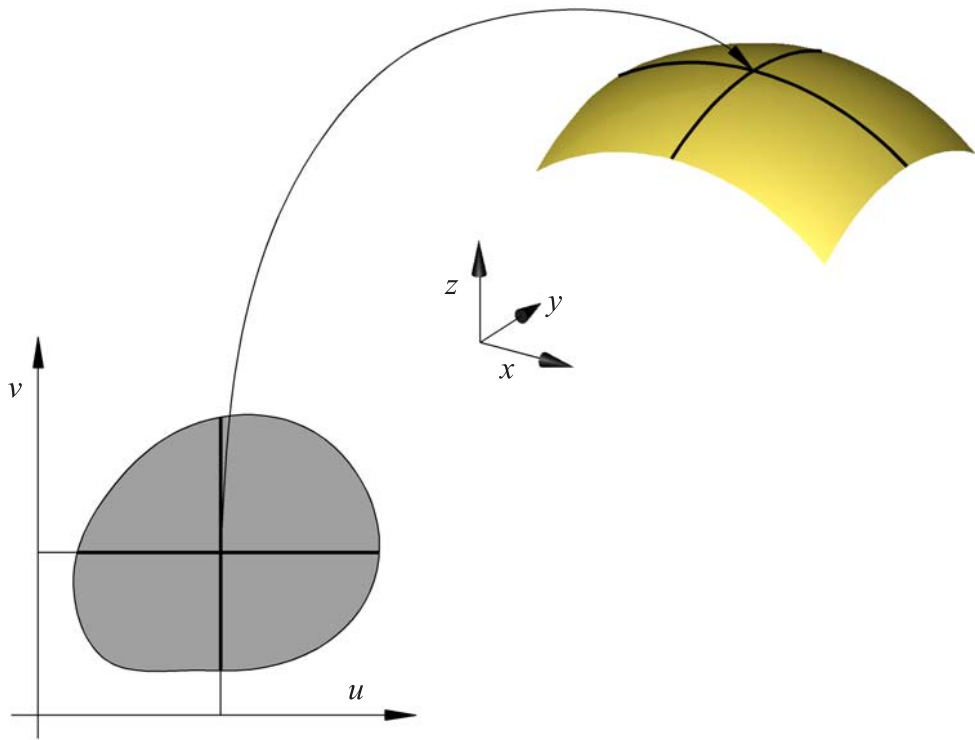
Egy felület többféleképpen is leírható. A számítógéppel segített geometriai tervezés számára a paraméteres előállítási forma a legfontosabb. Az implicit leírás speciális feladatok esetén - például metszés, testmodellezés - kerülhet előtérbe, az explicit előállítás azonban a CAGD szempontjából elhanyagolható, bár természetesen nem teljesen kizárt.

Az explicit leírás koordinátarendszer-függő, ezért ritkán alkalmazható, mivel megtörténhet, hogy egy koordinátatranszformáció után a felület az új rendszerben már nem is írható le explicit alakban. Ilyen szempontból a paraméteres alak a legjobb, mivel itt az értelmezési tartomány és az értékkészlet teljesen szétválasztott (9.1. ábra). Az áttérés egyik leírási formáról a másikra általában nem lehetséges.

Differenciálgeometriai értelemben vett elemi felületen olyan

$s(u, v) = \begin{bmatrix} r_x(u, v) & r_y(u, v) & r_z(u, v) \end{bmatrix}^T, (u, v) \in G \subset \mathbb{R}^2$ kétparaméteres vektorfüggvény által meghatározott pontthalmazt értünk, ahol G egyszeresen összefüggő részhalmaza a síknak (paramétersíknak), és az $s(u, v)$ függvényre a következők teljesülnek:

- $s(u, v)$ kölcsönösen egyértelmű, mindkét irányban folytonos;
- $s(u, v)$ a G tartomány minden pontjában folytonosan differenciálható;
- és $\frac{\partial}{\partial u}s(u, v) \times \frac{\partial}{\partial v}s(u, v) \neq \mathbf{0}, \forall (u, v) \in G$ esetén.



9.1. ábra. Felület paraméteres megadása

A G tartományt paramétertartománynak nevezzük. Az alkalmazások során a paramétertartomány általában a koordinátatengelyekkel párhuzamos oldalú téglalap, azaz $u \in [u_0, u_1]$, $v \in [v_0, v_1]$. Differenciálgeometriai értelemben felületen véges sok elemi felület egyesítésével előállított összefüggő alakzatot értünk, ahol az egyesített felület bármely pontjának van olyan környezete, mely az alakzathoz elemi felületet metsz ki.

Egy felületnek nemcsak többféle előállítási formái lehetnek, hanem azok egy előállítási formán belül is többféleképpen leírhatók. Tekintsük az $s(u, v)$, $(u, v) \in G \subset \mathbb{R}^2$ felületet, és a $\hat{G} \subset \mathbb{R}^2$ egyszeresen összefüggő tartományt. Feltételezzük továbbá, hogy a G és \hat{G} tartományok között van egy kölcsönösen egyértelmű leképezés, mely folytonosan differenciálható és a differenciálhányados sehol sem tűnik el.

Ekkor az $s(u(\hat{u}, \hat{v}), v(\hat{u}, \hat{v}))$ differenciálgeometriai értelemben vett felület, és megegyezik az $s(u, v)$ függvénnyel előállított felülettel.

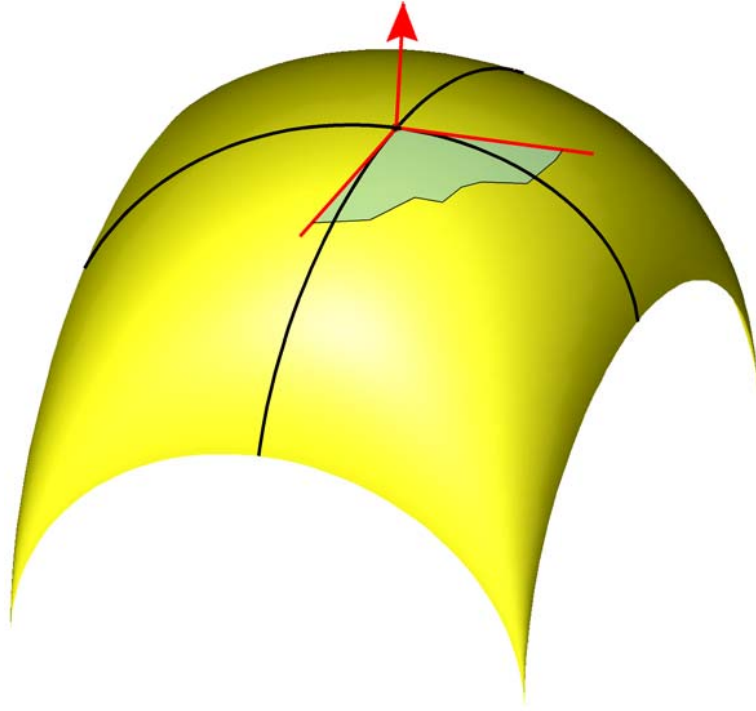
A paramétertartomány fenti tulajdonságú transzformációját megengedett paramétertranszformációnak nevezzük. A továbbiakban ismertetendő felületi jellemzők a megengedett paramétertranszformációval szemben invariánsak, a felület belső, geometriai jellemzői. A CAGD-ban gyakran használt az úgynevezett affin (vagy lineáris) paramétertranszformáció, amikor az $[u_0, u_1] \times [v_0, v_1]$ tartományról az $[\hat{u}_0, \hat{u}_1] \times [\hat{v}_0, \hat{v}_1]$ tartományra térünk át az

$$\begin{aligned} u &= \frac{u_1 - u_0}{\hat{u}_1 - \hat{u}_0} (\hat{u} - \hat{u}_0) + u_0; \\ v &= \frac{v_1 - v_0}{\hat{v}_1 - \hat{v}_0} (\hat{v} - \hat{v}_0) + v_0 \end{aligned}$$

függvényekkel.

9.1.1. Felületi görbe, érintősík, normális

Tekintsük az $\mathbf{s}(u, v), (u, v) \in G$ felületet és a G paramétertartományon az $\mathbf{u}(t) = [u(t) \ v(t)]^T$ görbét. Az $\mathbf{u}(t)$ görbének a felületen az $\mathbf{r}(t) = \mathbf{s}(\mathbf{u}(t)) = \mathbf{s}(u(t), v(t))$ pontthalmaz felel meg, ami a feltételek miatt egy (differenciálgeometriai értelemben vett) görbe. Az ilyen görbét felületi görbéknek nevezzük. Kiemelt szerepük van azoknak a felületi görbéknek, melyekre $u = t, v = konstans$, vagy $v = t, u = konstans$ teljesül. Ezeket a felület paramétervonalainak nevezzük. A felület bármely pontján pontosan egy u - és egy v -paramétervonal halad át.



9.2. ábra. Felület adott pontján áthaladó paramétervonalak, a pontbeli érintősík és normális

Az $\mathbf{s}(u, v)$ felület tetszőleges pontján áthaladó felületi görbéknek a közös pontban vett érintői egy síkban vannak, mégpedig az

$$\mathbf{s}_u = \frac{\partial}{\partial u} \mathbf{s}(u, v) \text{ és } \mathbf{s}_v = \frac{\partial}{\partial v} \mathbf{s}(u, v)$$

vektorok által kifeszített síkban, mivel tetszőleges $\mathbf{r}(t) = \mathbf{s}(u(t), v(t))$ felületi görbe esetén

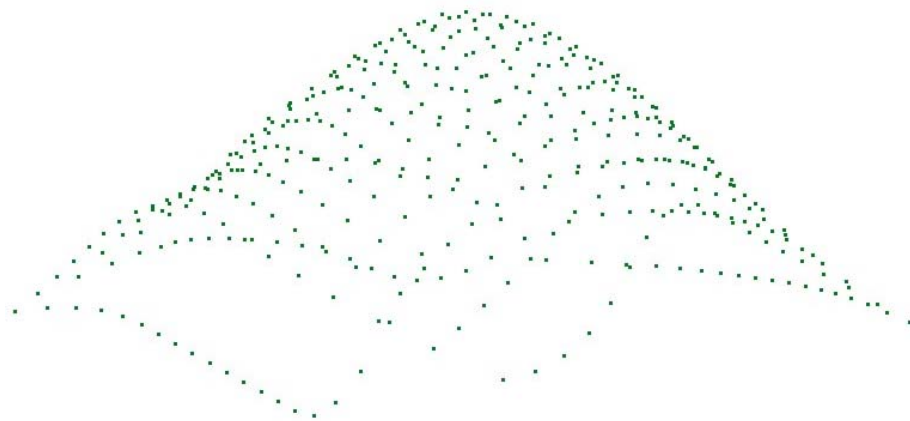
$$\begin{aligned} \frac{d}{dt} \mathbf{r}(t) &= \frac{d}{dt} \mathbf{s}(u(t), v(t)) = \\ &= \frac{\partial}{\partial u} \mathbf{s}(u, v) \frac{d}{dt} u(t) + \frac{\partial}{\partial v} \mathbf{s}(u, v) \frac{d}{dt} v(t), \end{aligned}$$

vagyis az érintővektor az \mathbf{s}_u és \mathbf{s}_v vektorok lineáris kombinációja. Ezt a síkot a felület adott pontbeli érintősíkjának nevezzük. A síkot felfeszítő \mathbf{s}_u és \mathbf{s}_v vektorok a felület adott ponton áthaladó u - illetve v -paramétervonalainak az érintői. Az érintősíkra merőleges $\mathbf{s}_u \times \mathbf{s}_v$ vektort a felület normálvektorának, az egységnyi hosszúságú

$$\mathbf{n} = \frac{\mathbf{s}_u \times \mathbf{s}_v}{|\mathbf{s}_u \times \mathbf{s}_v|}$$

vektort pedig a felület normálegységvektorának nevezzük (9.2. ábra). Megjegyezzük, hogy a normálvektor irányítása paramétertranszformáció során megváltozhat, de csak globálisan, tehát a felület minden pontjában.

9.2. Felületek szemléltetése

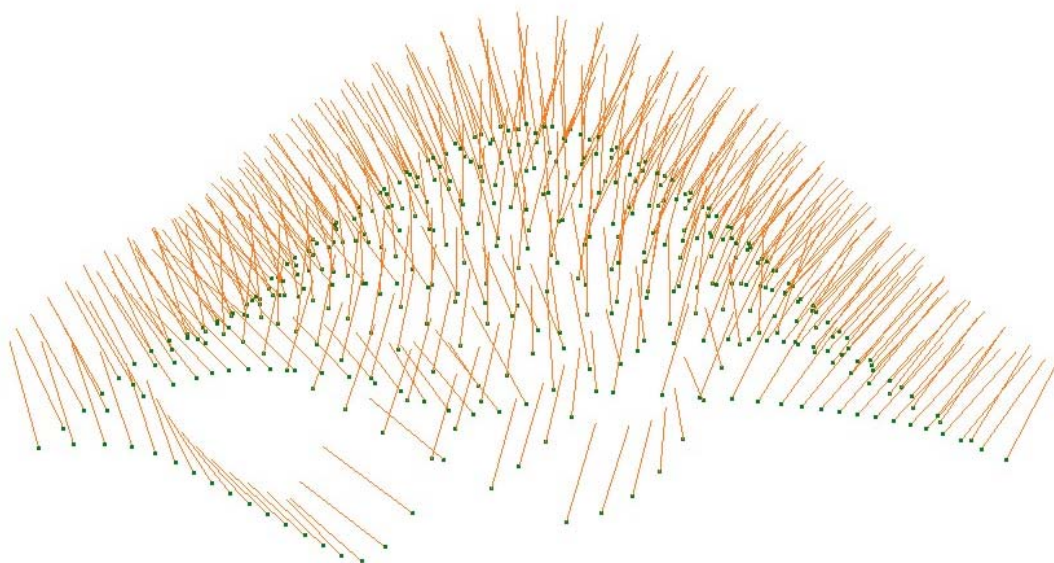


9.3. ábra. Felület szemléltetése ponthálóval

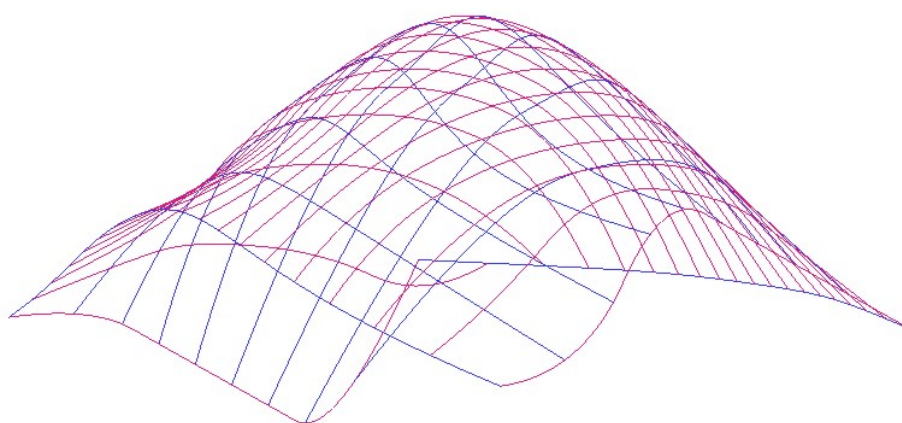
A paraméteres alakban leírt felületek megjelenítésére több lehetőség van. Megtehetjük, hogy a paramétertartományban pontokat veszünk fel, kiszámítjuk a hozzájuk tartozó felületi pontokat és azokat megjelenítjük (lásd a 9.3. ábrát).

A felület változásának szemléltetésére a felületi normálisokat is megjeleníthetjük (9.4. ábra).

A felület paramétervonalainak seregeivel (u , illetve v irányú görbeseregekkel) is jól szemléltethetjük a felületet (9.5. ábra).



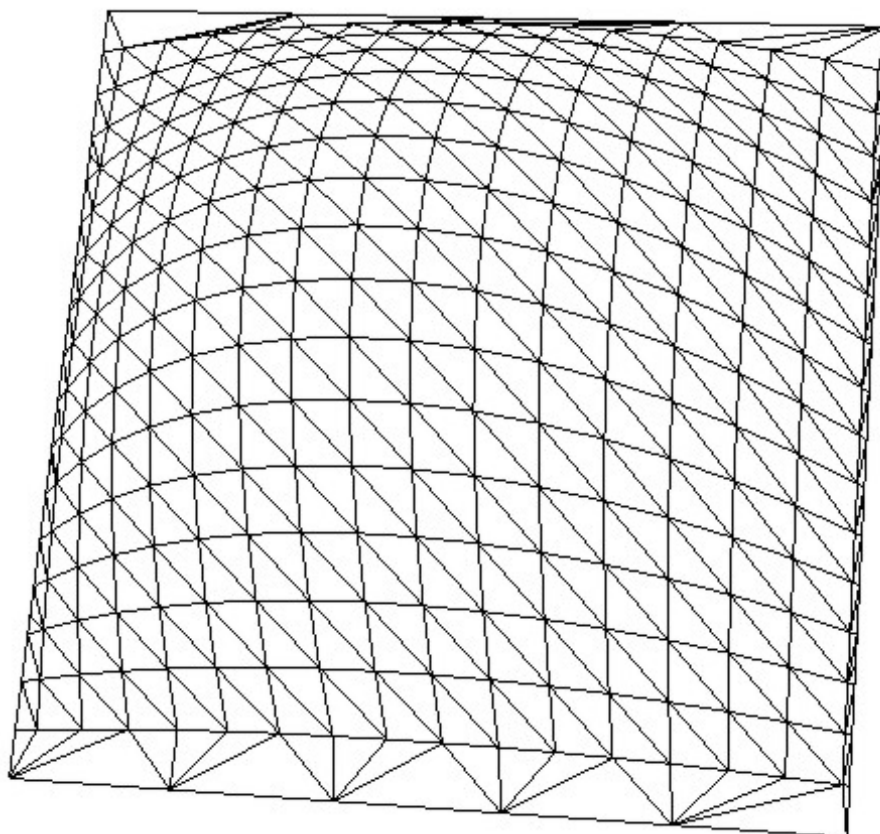
9.4. ábra. Felület szemléltetése pontokkal és normálisokkal



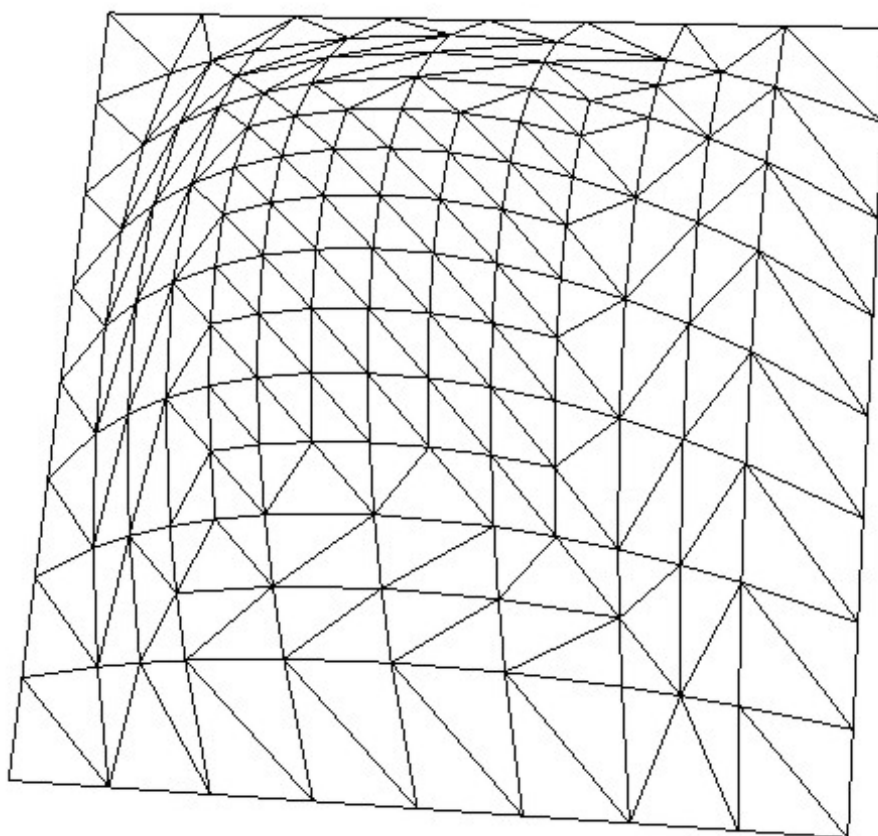
9.5. ábra. Felület szemléltetése paramétervonalakkal

Ha színes, árnyalt képet akarunk létrehozni, netalán textúrázni akarjuk a felületet, vagy a valószerűséget fokozó optikai hatásokat akarunk érvényesíteni, akkor legtöbbször poligonhálóból közelítjük a felületet, majd a közelítő poligonhálót jelenítjük meg. Ezt az eljárást csak a fénysugárkövető módszer alkalmazásakor kerülhetjük el, de ezt nagy számítási igénye miatt csak ritkán használjuk, kevés rendszer teszi lehetővé.

Háromszögekből álló hálót alkalmazunk leggyakrabban, mivel így biztosított, hogy a poligonháló elemei síkidomok legyenek. A felületek optimális közelítése poligonhálóból nagyon összetett feladat. A görbék töröttvonalal való közelítéséhez hasonlóan, a jó közelítő poligonháló a nagy görbületű felületrészeknél sűrű (több kisebb poligonból áll), a kis görbületű részeket pedig nagyobb poligonokkal közelíti. Arra is ügyelni kell, hogy elkerüljük a nagyon kis szöget tartalmazó (tű- vagy fogpiszkáló-szerű) háromszögeket. A 9.6. ábrán egy rossz megoldást láthatunk, a háromszögek mérete, sűrűsége nem veszi figyelembe a felület görbületét, és a széleken egyenetlenek a háromszögek. A 9.7. ábrán ugyanannak a felületnek egy sokkal jobb közelítését láthatjuk.



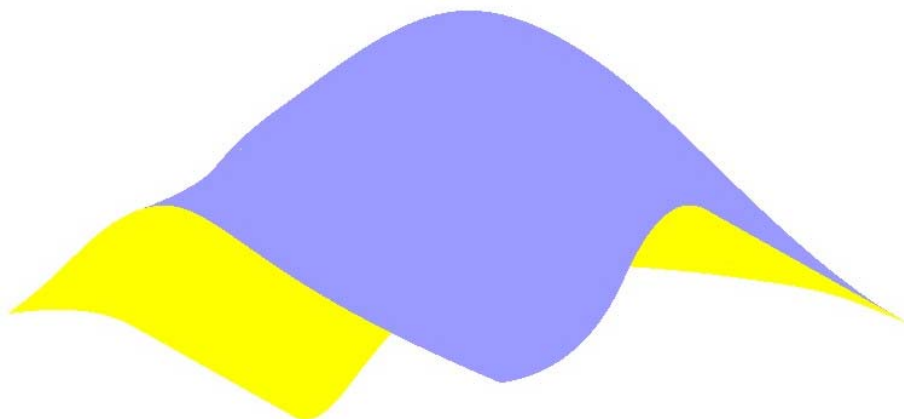
9.6. ábra. A közelítő poligonháló nem veszi figyelembe a felület görbületének változását



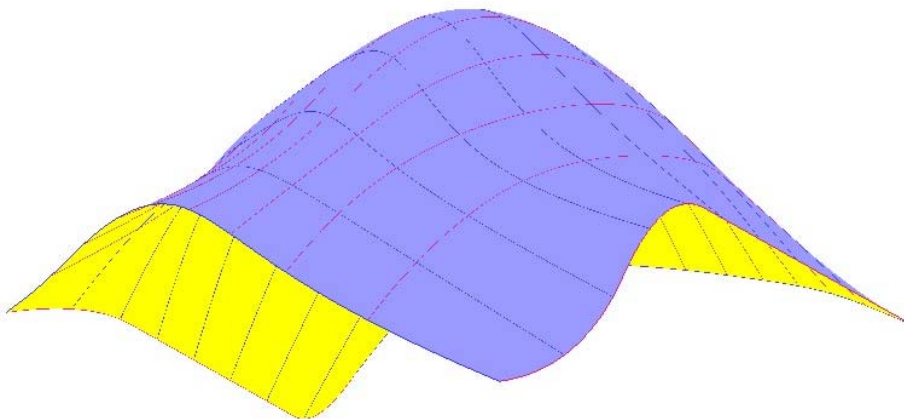
9.7. ábra. A közelítő poligonháló figyelembe veszi a felület görbületének változását és elkerüli a tűszerű háromszögeket

Központi szerepük miatt a poligonhálók megjelenítését részletesen tárgyaljuk.

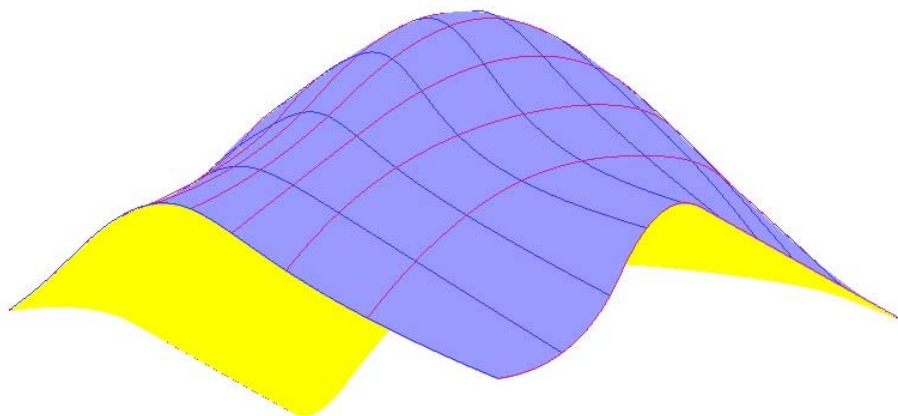
A 9.8. ábrán poligonhálóval ábrázoltunk egy felületet. A felületre a paramétervonalak (vagy egyéb görbék) is rárajzolhatók. Ilyenkor azonban újabb problémába ütközünk, ugyanis a megjelenítő algoritmus hol a felület, hol a felületre illeszkedő görbe szerint fogja a megfelelő pixelt kiszínezni (lásd a 9.9. ábrát). Ez tulajdonképpen nem hiba, hiszen a felületi pont és a rá illeszkedő görbepont ugyanolyan távol van, tehát a láthatóságukat nem lehet eldönteni. Ilyen esetben kis csaláshoz folyamodunk, a felületre illeszkedő görbe helyett, attól azonos távolságra lévő görbét rajzolunk ki (a távolság persze igen kicsi). Az eredmény a 9.10. ábrán látható.



9.8. ábra. Felület szemléltetése árnyalt poligonhálóval



9.9. ábra. Árnyalt felület paramétervonalakkal



9.10. ábra. A felületre illeszkedő görbe helyett, attól azonos távolságra lévő görbét ábrázolunk

9.2.1. Poligonhálók árnyalt megjelenítése

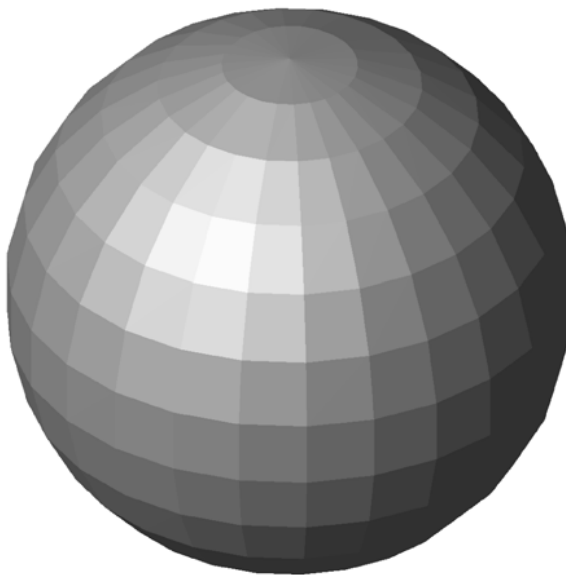
Külön figyelmet érdemelnek a poligonhálók, mivel egyrészt a geometriai modellezőrendszerek gyakran poliéderrel, poligonhálóval közelítik a testeket, a görbült felületeket is, másrészt a mai grafikus kártyák poligonokat, poligonhálókat tudnak gyorsan, hatékonyan megjeleníteni.

Konstans árnyalás

Poliéderek, poligonhálók árnyalt megjelenítésére a legegyszerűbb megoldás az úgynevezett konstans árnyalás. Ebben az árnyalási modellben minden lapnak csak egy pontjában számítjuk ki a színt, és az egész lapra ezt alkalmazzuk. Ez csak akkor ad teljesen kielégítő megoldást, ha a következők teljesülnek:

- a fénysugarak párhuzamosak,
- a nézőpont végtelen távoli pont (párhuzamos vetítés),
- az ábrázolt objektum valóban poliéder, nem pedig egy görbült felületet közelítő poligonháló.

Görbült felületet közelítő poligonháló esetén az eredmény lehangoló, a közelítő poligonok élei látszanak (9.11. ábra).



9.11. ábra. Konstans árnyalás

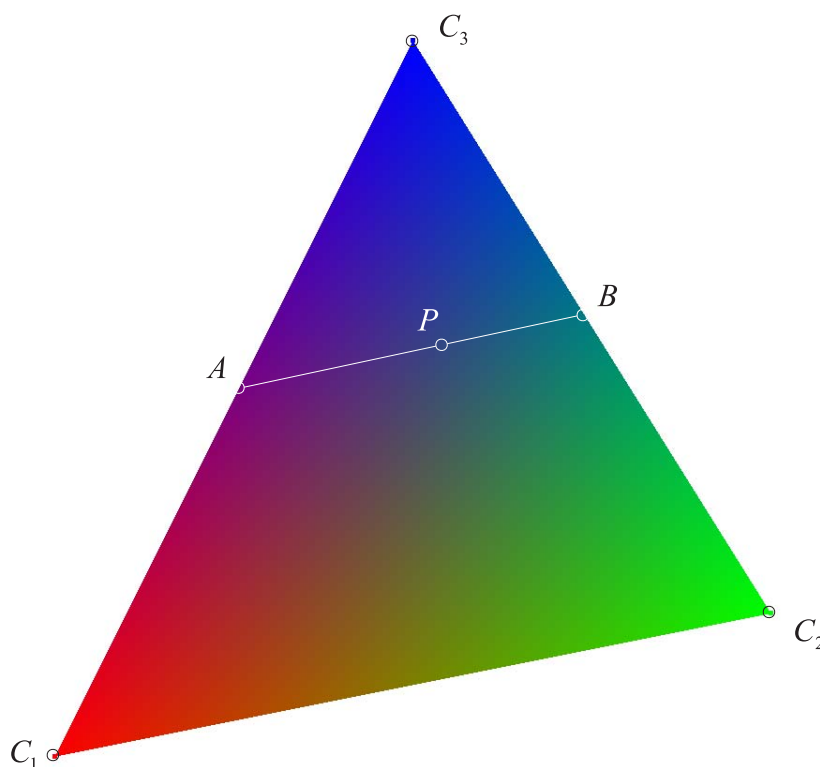
Különös figyelmet kell fordítanunk a poligonhálóval közelített görbült felületek esetén a felületi normálisokra, melyek az árnyalásnál központi szerepet játszanak. Ha ugyanis a közelítő poligonháló lapjainak normálisait vesszük figyelembe az árnyalási számításoknál, akkor a normálisok ugrásszerű változása miatt a színárnyalatban is ugrásszerű változás fog bekövetkezni az élek mentén, ezért a közelítő poligonháló élei kisebb-nagyobb mértékben mindig látszanak. A probléma kiküszöbölése érdekében a közelítő poligonháló minden csúcsában külön megadjuk még a normálist is. Ennek egzakt módja az lenne, hogy meghatározzuk az eredeti felület

normálisát az adott pontokban. Ez azonban nem mindig lehetséges, illetve célszerű, ezért sok esetben csak közelítjük ezt az egzakt normálist a poligonháló normálisaiból kiindulva. Egy elterjedt közelítési mód a csúcspontban találkozó lapok normálisainak átlagolása, azaz annak a vektornak a meghatározása, mely komponensei a csúcsponthoz tartozó normálisok megfelelő komponenseinek számtani közepe. Az így kapott normális tehát általában nem lesz merőleges egyetlen lapra sem, de elég jól közelíti az eredeti (nem síklapokkal határolt) felület normálisát.

Görbült felületeket közelítő poligonháló esetén konstans árnyalás helyett valamilyen folytonos árnyalást (smooth-shading) alkalmazunk. Két ilyen módszert ismertetünk, a Gouraud- és a Phong-féle árnyalási modellt.

Gouraud-féle árnyalás

A Gouraud-féle árnyalás esetén kiszámítjuk a színt a csúcspontokban a csúcspontbeli normálisok segítségével. A lap élei mentén a csúcspontok színének lineáris interpolációjával határozzuk meg a színt, a lap valamely belső pontjában pedig a 9.12. ábra szerinti kettős lineáris interpolációval, azaz először az A és B pontokban, majd ezek lineáris interpolációjával a P pontban.



9.12. ábra. A poligon belső pontjaiban kettős interpolációval határozzuk meg a színt

A Gouraud-féle árnyalási modell hátránya, hogy a test körvonala nem lesz sima, azaz elárulja, hogy poliédermodellt használtunk.

További hátrány mutatkozik a nagy tükröző visszaverődésű részeknél, valamint jelentkezhet az úgynevezett Mach-sáv hatás, ami az emberi szem számára ugrásszerű fényintenzitás

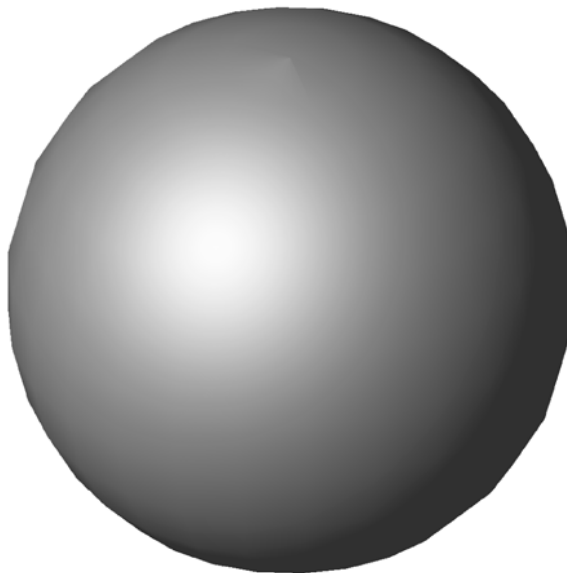
változást jelent. Ez jelentősen csökkenthető a modell finomításával, azaz több approximáló lap használatával, ami azonban a modell tárolási igényét és a feldolgozás idejét növeli.



9.13. ábra. Gouraud-féle árnyalás

Phong-féle árnyalás

A Phong-féle árnyalás csak abban különbözik a Gouraud-félétől, hogy nem a színt interpolálja, hanem a normálisokat. Tehát a lap élei mentén a csúcspontbeli normálisok lineáris interpolációjával határozzuk meg a normálist, a lap valamely belső pontjában pedig kettős interpolációval. Így minden egyes pontban ki kell számítani a színt az interpolációval kapott normálist felhasználva. Ez természetesen több számítást igényel, de az eredmény is jobb minőségű lesz. Egyetlen negatív jelenség marad, a képhatár szögletessége.



9.14. ábra. Phong-féle árnyalás

10. fejezet

Animáció

Az egymás után gyorsan levetített képek sorozatát az emberi szem (és az agy) mozgásnak érzékeli. Ehhez mozifilmek esetén másodpercenként 24 képkockát vetítenek le. A számítógépel készített mozgókép technikai feltétele, hogy a videomemória két egyenértékű képsík (színpuffer) tárolására legyen alkalmas. Az egyiket megjelenítjük, miközben a másikat írjuk (rajzolunk rá), majd a két képsík szerepét felcseréljük. Ha csak egy képsíkot használunk és a kép letörlése után hozzuk létre az újat még viszonylag egyszerű kép esetén is szaggatott, villogó lesz a kép, nem fogunk folyamatos mozgást érzékelni.

Megkülönböztetünk valós idejű és nem valós idejű animációt. A valós idejű animáció esetén a számítógép menet közben - a két képsík cseréje közötti időben - állítja elő a képet (végzi el az összes szükséges számítást). Előfordulhat azonban, hogy a számítógépünk nem elég gyors ehhez, ilyen esetben két lépésben készítjük az animációt. Előbb létrehozuk az egyes képkockákat és a háttértárolón tároljuk őket, majd innen beolvassuk a megfelelő sebességgel "levetítjük" a tárolt képeket. Az utóbbi eljárást nevezzük nem valós idejű animációnak.

A mozgás érzékeltetésére az ábrázolt alakzatot és a nézőpontot (a vetítés középpontját, a kamerát) változtathatjuk. Ha környezetéből kiragadott objektumot (objektumokat) akarunk körüljárni, azt megtehetjük a modell forgatásával (modelltranszformáció). Ha az objektum környezetét is ábrázoljuk, akkor a nézőpontnak (kamerának) kell mozogni az objektum terében. A kamera mozgásához nem elegendő a pályagörbét megadni, hiszen az is lényeges információ, hogy a kamera merre néz, azaz a pályagörbe minden pontjában még egy irányvektort is meg kell adni. Ezen kívül természetesen a valós helyzet szimulálása érdekében a mozgás sebességét, és annak változását is figyelembe kell venni.

Élvezhető animáció létrehozása összetett szintér esetén nem egyszerű feladat, szerencsére manapság már speciális szoftverek állnak rendelkezésre, melyek nagymértékben megkönnyítik az animáció készítését.

A. Függelék

Vektorműveletek geometriai jelentése

Az alakzatokat a számítógéppel való kezelés érdekében valamilyen koordináta-rendszerben kell leírni. A grafikus rendszerek általában derékszögű Descartes-féle koordináta-rendszert használnak. Egy koordináta-rendszer rögzítése után a pontok koordinátákkal (rendezett számhármasokkal) írhatók le. Ezért gyakran célszerű vektorokat használni a könnyebb kezelhetőség és az egységes leírás érdekében. Mind a helyvektoroknak, mind a szabad vektoroknak megvan a maga szerepe. Például az egyenes egy pontjával és irányával egyértelműen megadható. Egy koordináta-rendszer rögzítése után a pontot helyvektor, az irányt szabad vektor írja le.

Az alábbiakban összefoglaljuk a vektorokon értelmezett skaláris, vektoriális és vegyes szorzás geometriai jelentését, és példát adunk ezek alkalmazására a számítógépi grafikában.

A.1. Skaláris szorzat

Az $\mathbf{a} = (a_x, a_y, a_z)$ és $\mathbf{b} = (b_x, b_y, b_z)$ vektorok skaláris (vagy belső) szorzata az

$$\mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z$$

szám. Ennek geometriai jelentése: az \mathbf{a} és \mathbf{b} vektorok hosszának a szorzata szorozva a közbezárt szög koszinuszával, azaz

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \alpha$$

Ha \mathbf{a} és \mathbf{b} nem nullvektor, akkor skaláris szorzatuk pontosan akkor nulla, ha merőlegesek egymásra. Ha pl. $|\mathbf{b}| = 1$, akkor az $|\mathbf{a} \cdot \mathbf{b}|$ szám az \mathbf{a} vektor \mathbf{b} irányára eső merőleges vetületének a hosza.

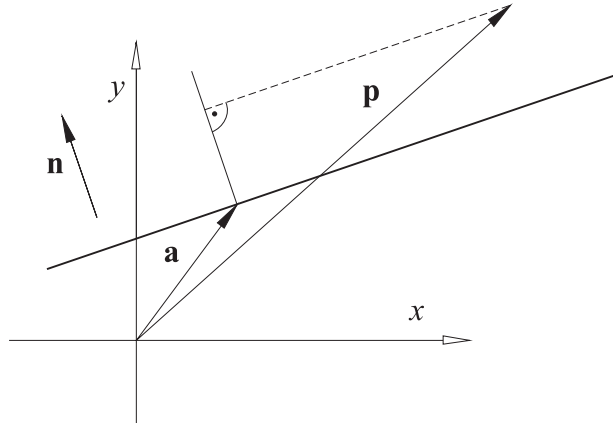
Ha a sík egy egyenesének \mathbf{a} egy pontja, \mathbf{n} pedig egységnyi hosszúságú normálvektora (az egyenesre merőleges egységnyi hosszúságú vektor), \mathbf{p} pedig a sík tetszőleges pontja, akkor a

$$(\mathbf{p} - \mathbf{a}) \cdot \mathbf{n}$$

szorzat a \mathbf{p} pontnak az egyenestől mért előjeles távolsága (A.1. ábra). A szorzat előjele akkor pozitív, ha \mathbf{p} az egyenesnek azon az oldalán van amerre az \mathbf{n} mutat.

A.2. Vektoriális szorzat

Az $\mathbf{a} = (a_x, a_y, a_z)$ és $\mathbf{b} = (b_x, b_y, b_z)$ vektorok vektoriális (vagy külső) szorzata az



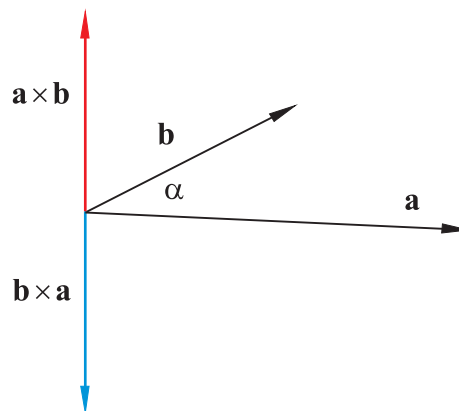
A.1. ábra. Pont és egyenes előjeles távolsága

$$\mathbf{a} \times \mathbf{b} = (a_y b_z - a_z b_y, -a_x b_z + a_z b_x, a_x b_y - a_y b_x)$$

vektor. Az $\mathbf{a} \times \mathbf{b}$ vektor az \mathbf{a} és \mathbf{b} vektorokra, tehát a velük párhuzamos síkokra merőleges (A.2. ábra), hossza pedig

$$|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}| |\mathbf{b}| \sin \alpha,$$

vagyis a közös kezdőpontba tolt \mathbf{a} és \mathbf{b} vektorok által meghatározott háromszög területének a kétszerese.



A.2. ábra. Vektoriális szorzat

Az $\mathbf{a} \times \mathbf{b}$ vektor pontosan akkor lesz nullvektor, ha a vektorok lineárisan összefüggők, azaz ha legalább az egyikük nullvektor, vagy egymással párhuzamosak.

Ha \mathbf{a} és \mathbf{b} az (x, y) koordinátasíkban fekvő vektorok, akkor az $\mathbf{a} \times \mathbf{b}$ vektornak csak a z komponense lehet nullától különböző. Az \mathbf{a} és \mathbf{b} vektorokhoz ezt a skalárt hozzárendelő műveletet a \wedge jellel jelöljük, tehát

$$\mathbf{a} \wedge \mathbf{b} = a_x b_y - a_y b_x.$$

Az $(\mathbf{a} \wedge \mathbf{b})/2$ szám a közös kezdőpontba tolt \mathbf{a} és \mathbf{b} vektorok által felfeszített háromszög előjeles területe. Ennek az előjeles területnek igen sok alkalmazása van a számítógépi grafikában és a számítógéppel segített geometriai tervezésben.

A $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ pontok körüljárási irányát az

$$\mathbf{a} = \mathbf{p}_2 - \mathbf{p}_1,$$

$$\mathbf{b} = \mathbf{p}_3 - \mathbf{p}_1$$

vektorokból nyert $\mathbf{a} \wedge \mathbf{b}$ skalár előjele alapján kapjuk meg. Ha $\mathbf{a} \wedge \mathbf{b} > 0$, akkor $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ pozitív (az óramutató járásával ellentétes) körüljárási irányt határoz meg; ha $\mathbf{a} \wedge \mathbf{b} < 0$, akkor negatív (az óramutató járásával megegyező); $\mathbf{a} \wedge \mathbf{b} = 0$ esetén pedig nem beszélhetünk körüljárási irányról, mivel az adott pontok kollineárisak (közös egyenesre illeszkednek).

A poligonok (zárt töröttvonalak által határolt síkidomok) háromszögekre bonthatók, így ezek előjeles területe is meghatározható. A $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$ csúcspontokkal adott poligon a \mathbf{p}_1 pontból kiindulva a $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$; $\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4$; \dots ; $\mathbf{p}_1, \mathbf{p}_{n-1}, \mathbf{p}_n$ háromszögekre bontható. Az első előjeles területe

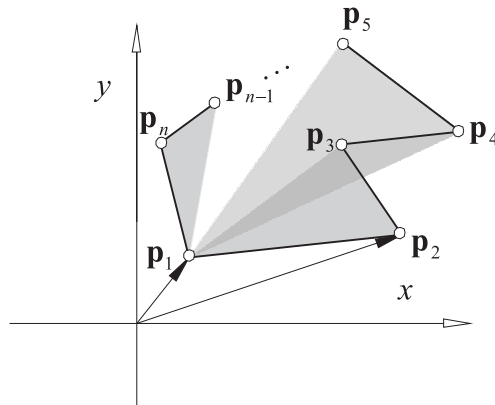
$$T_1 = (\mathbf{p}_2 - \mathbf{p}_1) \wedge (\mathbf{p}_3 - \mathbf{p}_1) / 2$$

a másodiké

$$T_2 = (\mathbf{p}_3 - \mathbf{p}_1) \wedge (\mathbf{p}_4 - \mathbf{p}_1) / 2$$

az i -ediké ($1 \leq i \leq n-2$)

$$T_i = (\mathbf{p}_{i+1} - \mathbf{p}_1) \wedge (\mathbf{p}_{i+2} - \mathbf{p}_1) / 2$$



A.3. ábra. Poligon háromszögekre bontása

Ezen területek előjele a háromszögek körüljárási iránya szerint változik. Az A.3. ábrán pátható például, hogy $T_1 > 0$, $T_2 < 0$ és $T_3 > 0$. A poligon előjeles területe

$$\begin{aligned} T_p &= \sum_{i=1}^{n-2} T_i = \\ &= \left(\sum_{i=1}^{n-2} (\mathbf{p}_{i+1} - \mathbf{p}_1) \wedge (\mathbf{p}_{i+2} - \mathbf{p}_1) \right) / 2 \end{aligned} \tag{A.1}$$

Az (A.1) összeg kis átalakítással sokkal egyszerűbb (sokkal kevesebb műveletet igénylő) alakra is hozható. Figyelembe véve ugyanis, hogy

$$(\mathbf{p}_{i+1} - \mathbf{p}_1) \wedge (\mathbf{p}_{i+2} - \mathbf{p}_1) = \mathbf{p}_{i+1} \wedge \mathbf{p}_{i+2} - \mathbf{p}_1 \wedge \mathbf{p}_{i+2} - \mathbf{p}_{i+1} \wedge \mathbf{p}_1,$$

valamint

$$\mathbf{p}_1 \wedge \mathbf{p}_i = -\mathbf{p}_i \wedge \mathbf{p}_1$$

és bevezetve a $\mathbf{p}_{n+1} = \mathbf{p}_1$ jelölést, az (A.1) összeg

$$T_p = \left(\sum_{i=1}^n \mathbf{p}_i \wedge \mathbf{p}_{i+1} \right) / 2$$

alakra hozható.

A fenti eljárás alkalmas konvex és konkáv poligonok területének kiszámítására is. Egy pontthalmazt konvexnek nevezünk, ha bármely két pontját összekötő szakasz minden pontja a pontthalmazhoz tartozik. A nemkonvex pontthalmazokat konkávnak nevezzük. A síkbeli konvex poligonok belső szögei 180 foknál nem nagyobbak. Ez szemléletesen úgy is megfogalmazható, hogy nem lehet bennük elbújni, azaz a poligon bármely pontjából a poligon összes többi pontját láthatjuk (tehát konvex alaprajzú kiállítóteremben egyetlen teremőr elegendő).

Poligonok előjeles területét gyakran használják a számítógépi grafikában. Például a poligonok látható oldalának meghatározásához ezt is felhasználják a grafikus rendszerek.

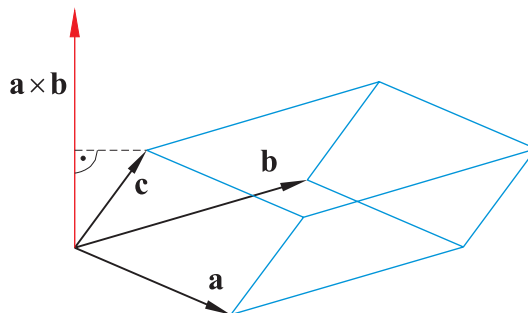
A.3. Vegyes szorzat

Az $\mathbf{a} = (a_x, a_y, a_z)$, $\mathbf{b} = (b_x, b_y, b_z)$ és $\mathbf{c} = (c_x, c_y, c_z)$ vektorok vegyes szorzatát a vektorokból képzett mátrix determinánsával számítjuk ki, azaz

$$(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \det \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{bmatrix}$$

Könnyen látható, hogy

$$(\mathbf{a}, \mathbf{b}, \mathbf{c}) = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$$



A.4. ábra. A vegyes szorzat geometriai jelentése

Mint az A.4. ábra is mutatja, ez a szám a közös kezdőpontba tolt \mathbf{a} , \mathbf{b} , \mathbf{c} vektorok által felfesztett paralelepipedon (parallelogrammák által határolt négyoldalú hasáb) előjeles térfogatával

egyezik meg. Ennek segítségével könnyen kiszámítható a négy csúcspontjával adott tetraéder előjeles térfogata.

A $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$ helyvektorokból képzett

$$\mathbf{a} = \mathbf{p}_2 - \mathbf{p}_1,$$

$$\mathbf{b} = \mathbf{p}_3 - \mathbf{p}_1,$$

$$\mathbf{c} = \mathbf{p}_4 - \mathbf{p}_1$$

vektorok vegyes szorzatából kapjuk a keresett

$$V = \frac{(\mathbf{a}, \mathbf{b}, \mathbf{c})}{6}$$

térfogatot. Ugyanis

$$m = \frac{\mathbf{a} \times \mathbf{b}}{|\mathbf{a} \times \mathbf{b}|} \cdot \mathbf{c}$$

a $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ alapháromszögű gúla magassága, a gúla alapterülete pedig

$$T = \frac{|\mathbf{a} \times \mathbf{b}|}{2}$$

így

$$V = \frac{Tm}{3} = \frac{(\mathbf{a}, \mathbf{b}, \mathbf{c})}{6}$$

A poligonok előjeles területének meghatározásakor követett gondolatmenettel kiszámíthatjuk tetszőleges poliéder (síklapok által határolt véges test) előjeles térfogatát is.

Az $\mathbf{a}, \mathbf{b}, \mathbf{c}$ vektorok vegyes szorzata pontosan akkor nulla, ha a három vektor lineárisan összefüggő. Ez geometriailag azt jelenti, hogy a közös kezdőpontba tolt vektoroknak van közös síkja, vagyis komplanárisok.

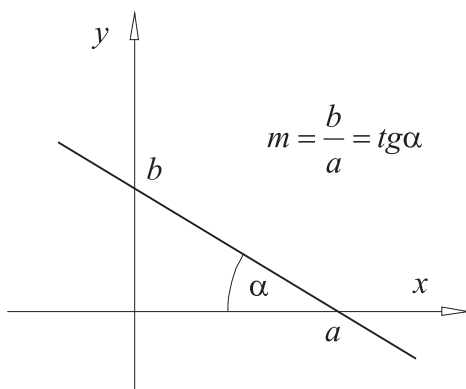
B. Függelék

Egyenes és sík leírása

Az egyenesnek és a síknak - a legegyszerűbb görbének, illetve felületnek - fontos szerepe van a számítógépi grafikában. Nagyon sok, gyakran előforduló feladat oldható meg egyszerűen, ha megfelelően választjuk meg az egyenes, illetve a sík leírását.

B.1. Egyenes leírása

Előbb összefoglaljuk az egyenes jól ismert leírásait, majd néhány feladat megoldását mutatjuk be ezek segítségével.



B.1. ábra. Egyenes explicit leírása

B.1.1. Egyenes explicit leírása

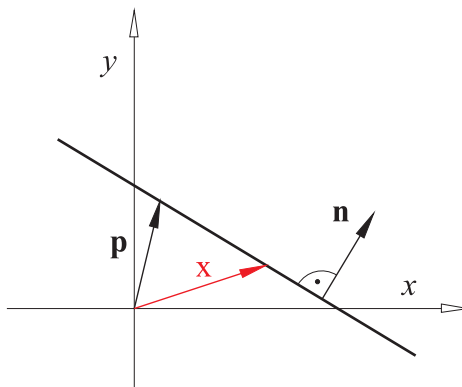
Az (x, y) síkbeli egyenes explicit leírása a középiskolából jól ismert

$$y = mx + b, \quad (\text{B.1})$$

ahol m az egyenes iránytangense (meredekségét kifejező szám), b pedig az egyenesnek az y tengellyel alkotott metszéspontjának y koordinátája (lásd a B.1. ábrát). Egy derékszögű Descartes-féle koordináta-rendszer rögzítése után a (B.1) alakú kifejezések egyértelműen meghatároznak

egy egyenest a síkon. A (B.1) alakú kifejezések és a sík egyenesei között azonban nincs kölcsönösen egyértelmű megfeleltetés, mivel az y tengellyel párhuzamos egyenesek nem írhatók le ilyen alakban.

Az explicit leírásnak ez a legnagyobb hátránya, vagyis az, hogy nagymértékben függ a koordináta-rendszertől. Általában jobb elkerülni a használatát, mivel mindig figyelniünk kell az egyenesnek a koordinátatengelyekhez viszonyított helyzetét, azaz esetszétválasztást kell végrehajtanunk. Olykor azonban a feladat megoldása éppen ezt kívánja, erre láttunk példát a 3.1. szakaszban a DDA algoritmusnál.



B.2. ábra. Egyenes implicit leírása

B.1.2. Egyenes implicit leírása

Legyen \mathbf{p} az egyenes egy ismert pontjának helyvektora, \mathbf{n} egy az egyenesre merőleges vektor (normálvektor), és jelölje \mathbf{x} az egyenes tetszőleges pontjának helyvektorát (lásd a B.2. ábrát)! Ilyen feltételek mellett az

$$\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}) = 0 \quad (\text{B.2})$$

egyenlőségnek kell teljesülni az \mathbf{n} és $\mathbf{x} - \mathbf{p}$ vektorok skaláris szorzatára. Az $\mathbf{n} = (n_x, n_y)$, $\mathbf{p} = (p_x, p_y)$ és $\mathbf{x} = (x, y)$ jelölést használva a (B.2) egyenlőség

$$n_x x + n_y y - \mathbf{n} \cdot \mathbf{p} = 0$$

alakban írható fel. Ezt nevezzük az egyenes implicit egyenletének. A sík bármely egyenesre felírható (B.2) alakban.

Az x és y változók együtthatói az egyenes egy normálvektorának komponensei. A normálvektornak csak az iránya fontos, az irányítása és hossza nem, hiszen az \mathbf{n} vektort tetszőleges nullától különböző λ számmal megszorozva ugyanannak az egyenesnek a leírását kapjuk. Ugyanis az

$$\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}) = 0$$

és

$$\lambda \mathbf{n} \cdot (\mathbf{x} - \mathbf{p}) = 0$$

egyenleteket ugyanazok az \mathbf{x} vektorok elégítik ki. Ezért mindig elérhető, hogy a normálvektor egységnyi hosszúságú legyen, azaz $|\mathbf{n}| = 1$. Ha $|\mathbf{n}| = 1$, akkor a (B.2) egyenletet az egyenes Hesse-féle normálalakjának nevezzük.

Az egyenes implicit alakját ismerve könnyen eldönthetjük, hogy egy adott pont illeszkedik-e az egyenesre. A \mathbf{q} helyvektorú pont pontosan akkor illeszkedik a (B.2) egyenesre, ha

$$\mathbf{n} \cdot (\mathbf{q} - \mathbf{p}) = 0.$$

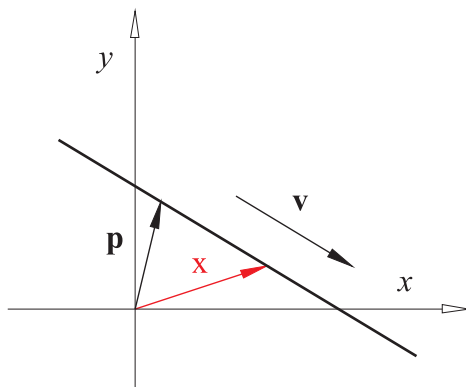
Az implicit alak ismeretében az is könnyen meghatározható, hogy egy pont az egyenes melyik oldalán van. Tekintsük ugyanis a \mathbf{q} pontot és helyettesítsük be a (B.2) egyenletbe! Ekkor a

$$d = \mathbf{n} \cdot (\mathbf{q} - \mathbf{p})$$

skalárt kapjuk eredményül.

- Ha $d = 0$, akkor \mathbf{q} illeszkedik az egyenesre.
- Ha $d > 0$, akkor \mathbf{q} az egyenesnek azon az oldalán van, amerre az \mathbf{n} vektor mutat.
- Ha $d < 0$, akkor \mathbf{q} az egyenesnek az \mathbf{n} vektorral ellentétes oldalán van.

Ha $|\mathbf{n}| = 1$, akkor d a \mathbf{q} pontnak az egyenestől mért előjeles távolsága.



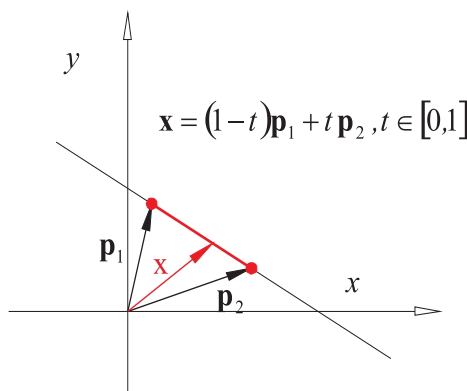
B.3. ábra. Egyenes paraméteres leírása

B.1.3. Egyenes paraméteres leírása

Legyen \mathbf{p} az egyenes egy ismert pontjának helyvektora, \mathbf{v} az egyenessel párhuzamos vektor, és jelölje \mathbf{x} az egyenes tetszőleges pontjának helyvektorát (lásd a B.3. ábrát)! Ekkor az

$$\mathbf{x} = \mathbf{p} + t\mathbf{v}, \quad t \in \mathbb{R} \quad (\text{B.3})$$

egyenlőség teljesül, vagyis az egyenes tetszőleges pontjának helyvektora ilyen alakban állítható elő. A \mathbf{v} vektornak csak az iránya fontos, az irányítása és hossza közömbös. Ilyen alakban nemcsak a sík, hanem a tér bármely egyenesre is felírható. Ezért ez az alak rendkívül fontos szerepet játszik.



B.4. ábra. Végpontjaival adott szakasz pontjai

Tudjuk, hogy a két nem egybeeső p_1 és p_2 pont egyértelműen meghatároz egy egyenest (lásd a B.4. ábrát). A $v = p_2 - p_1$ vektor ennek az egyenesnek az irányvektora, az egyenes paraméteres alakja pedig

$$x = p_1 + t(p_2 - p_1).$$

Ezt átalakítva az

$$x = (1 - t)p_1 + tp_2, \quad t \in \mathbb{R} \quad (\text{B.4})$$

alakot kapjuk a p_1 és p_2 pontokra illeszkedő egyenesre.

Az alkalmazások során gyakran előfordul, hogy nem a végtelen egyenesre, hanem csak a p_1 és p_2 végpontok által meghatározott szakaszra van szükségünk. Ha a (B.4) kifejezésben csak a $t \in [0, 1]$ értékeket engedjük meg, akkor pontosan a szakasz pontjait kapjuk.

Egy lineáris kombinációt konvexnek nevezünk, ha a kombináló tényezők nemnegatívak és összegük 1. Tehát a p_1, p_2 végpontú szakasz pontjait p_1 és p_2 konvex kombinációjaként kapjuk meg.

B.1.4. Egyenessel kapcsolatos feladatok

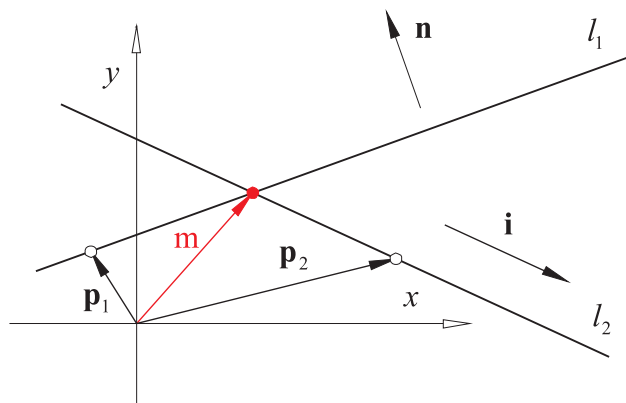
Néhány példát mutatunk arra, hogy az egyenes leírási módjának alkalmas választásával milyen egyszerűen oldhatók meg feladatok. Az itt szereplő feladatok elemiek, nyilván bárki meg tudja oldani, de nem biztos, hogy elég egyszerűen. Fontos szempont a feladatok megoldásánál, hogy a megoldás alapján könnyen lehessen hatékony programot írni.

Két egyenes metszéspontja

Adott az l_1 és l_2 egyenes a síkon, keressük az m metszéspontjukat. Azt feltételezzük, hogy l_1 p_1 pontjával (annak helyvektorával) és n normálvektorával adott; l_2 pedig p_2 pontjával és i irányvektorával (lásd a B.5. ábrát). Ez a feltételezés nem jelent megszorítást, mivel akárhogy is adottak az egyenesek, mindig elő tudjuk állítani ezeket az adatait.

Ilyen feltételek mellett l_1

$$n \cdot (x - p_1) = 0$$



B.5. ábra. Egyenesek metszéspontja

l_2 pedig

$$\mathbf{x} = \mathbf{p}_2 + t\mathbf{i}$$

alakban írható fel. A keresett metszéspontnak mindkét egyenletet ki kell elégíteni, ezért az első egyenletben \mathbf{x} -et a második egyenlet jobb oldalával helyettesíthetjük, amiből

$$\mathbf{n} \cdot (\mathbf{p}_2 - \mathbf{p}_1) - t\mathbf{n} \cdot \mathbf{i} = 0.$$

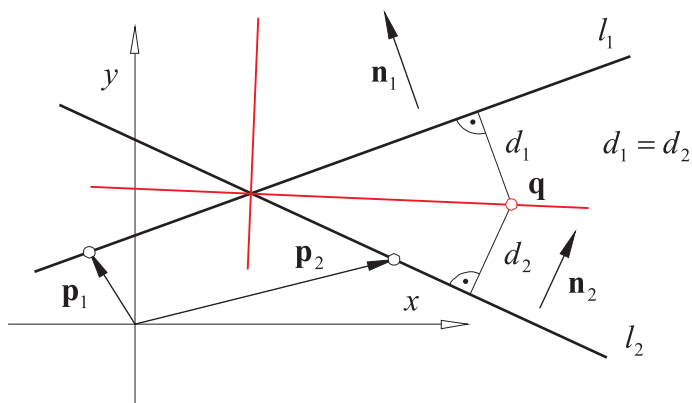
Ha $\mathbf{n} \cdot \mathbf{i} \neq 0$, azaz l_1 nem párhuzamos l_2 -vel, akkor az \mathbf{m} metszéspont paramétere

$$t = \frac{\mathbf{n} \cdot (\mathbf{p}_2 - \mathbf{p}_1)}{\mathbf{n} \cdot \mathbf{i}}$$

(ha $\mathbf{n} \cdot \mathbf{i} = 0$, azaz $l_1 \parallel l_2$, akkor nincs metszéspont), maga a keresett metszéspont pedig

$$\mathbf{m} = \mathbf{p}_2 + \frac{\mathbf{n} \cdot (\mathbf{p}_2 - \mathbf{p}_1)}{\mathbf{n} \cdot \mathbf{i}} \mathbf{i}.$$

Két egyenes szögfelező egyenesei



B.6. ábra. Szögfelező egyenesek

Adott a síkon az l_1 és l_2 egyenes, keressük a szögfelező egyeneseiket. Azt feltételezzük, hogy l_1 \mathbf{p}_1 pontjával és \mathbf{n}_1 normálvektorával, l_2 pedig \mathbf{p}_2 pontjával és \mathbf{n}_2 normálvektorával adott (lásd a B.6. ábrát). Azt is feltételezzük, hogy a normálvektorok egységnyi hosszúak, azaz $|\mathbf{n}_1| = |\mathbf{n}_2| = 1$. Az egyenesek Hesse-féle normálalakja tehát

$$\mathbf{n}_1 \cdot (\mathbf{x} - \mathbf{p}_1) = 0 \quad (\text{B.5})$$

$$\mathbf{n}_2 \cdot (\mathbf{x} - \mathbf{p}_2) = 0 \quad (\text{B.6})$$

Ezekbe a sík tetszőleges \mathbf{q} pontjának helyvektorát behelyettesítve, a pontnak az egyenesektől mért előjeles d_1 és d_2 távolságát kapjuk. A \mathbf{q} pont pontosan akkor van a szögfelezőn, ha $d_1 = d_2$. Ezért a szögfelezők implicit alakját a (B.5) és (B.6) egyenletek összeadásával, illetve kivonásával kapjuk meg, azaz

$$\mathbf{n}_1 \cdot (\mathbf{x} - \mathbf{p}_1) + \mathbf{n}_2 \cdot (\mathbf{x} - \mathbf{p}_2) = 0$$

és

$$\mathbf{n}_1 \cdot (\mathbf{x} - \mathbf{p}_1) - \mathbf{n}_2 \cdot (\mathbf{x} - \mathbf{p}_2) = 0.$$

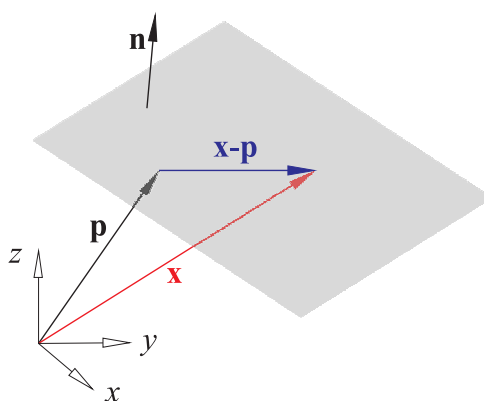
B.2. Sík leírása

B.2.1. Sík explicit leírása

Egy koordináta-rendszer rögzítése után a

$$z = ax + by + c$$

egyenletet kielégítő (x, y, z) koordinátájú pontok egy síkot határoznak meg. A z tengellyel párhuzamos síkok azonban nem írhatók le ilyen alakban, vagyis ez a leírás nagyon függ a koordináta-rendszer megválasztásától. Ezért ezt az alakot ritkán használjuk.



B.7. ábra. Sík implicit leírása

B.2.2. Sík implicit leírása

Legyen \mathbf{p} a sík egy ismert pontjának helyvektora, \mathbf{n} egy a síkra merőleges vektor (normálvektor), és jelölje \mathbf{x} a sík tetszőleges pontjának helyvektorát (lásd a B.7. ábrát)! Ekkor az \mathbf{x} helyvektoroknak az

$$\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}) = 0 \quad (\text{B.7})$$

egyenletet kell kielégíteni. Az $\mathbf{n} = (n_x, n_y, n_z)$, $\mathbf{p} = (p_x, p_y, p_z)$ és $\mathbf{x} = (x, y, z)$ jelölés bevezetésével (B.7) az

$$n_x x + n_y y + n_z z - \mathbf{n} \cdot \mathbf{p} = 0$$

alakban írható fel, amit a sík implicit egyenletének nevezünk. Nyilvánvaló, hogy az \mathbf{n} vektor hossza és irányítása közömbös, tehát az \mathbf{n} és $\lambda \mathbf{n}$, $\lambda \neq 0$ normálvektorok ugyanazt a síkállást határozzák meg. Ezért mindig elérhető, hogy $|\mathbf{n}| = 1$ legyen, amely esetben a (B.7) egyenletet a sík Hesse-féle normálalakjának nevezzük.

Ha a tér tetszőleges \mathbf{q} helyvektorú pontját behelyettesítjük a (B.7) egyenletbe, akkor a

$$d = \mathbf{n} \cdot (\mathbf{q} - \mathbf{p}), |\mathbf{n}| = 1$$

számot kapjuk.

- Ha $d = 0$, akkor q illeszkedik a síkra.
- Ha $d > 0$, akkor q a síknak azon az oldalán van, amerre az \mathbf{n} vektor mutat.
- Ha $d < 0$, akkor q a síknak az \mathbf{n} vektorral ellentétes oldalán van.

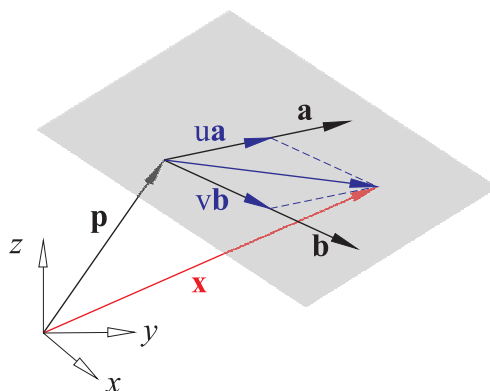
Ha $|\mathbf{n}| = 1$, akkor d a q pontnak a síktól mért előjeles távolsága.

B.2.3. Sík paraméteres leírása

Legyen \mathbf{p} a sík egy ismert pontjának helyvektora, \mathbf{a} és \mathbf{b} a síkkal párhuzamos, lineárisan független vektorok (egyik sem nullvektor és nem párhuzamosak), és jelölje \mathbf{x} a sík tetszőleges pontjának helyvektorát (lásd a B.8. ábrát)! Ekkor az \mathbf{x} vektorok

$$\mathbf{x} = \mathbf{p} + u\mathbf{a} + v\mathbf{b}, \quad u, v \in \mathbb{R} \quad (\text{B.8})$$

alakban írhatók fel. Ezt nevezzük a sík paraméteres alakjának. Látható, hogy a sík végtelen sokféleképpen írható fel (B.8) alakban.



B.8. ábra. Sík paraméteres leírása

B.2.4. Síkkal kapcsolatos feladatok

Két sík szögfelező síkjai

Adott két sík, meghatározandók a szögfelező síkjai. A feladat megoldása analóg a két egyenes szögfelező egyeneseinek meghatározásával. Azt feltételezzük, hogy mindkét síknak a Hesse-féle normálalakját ismerjük. A keresett szögfelező síkok implicit alakja az adott síkok Hesse-féle normálalakjának összege, illetve különbsége lesz.

Sík és egyenes dőféspontja

Adott egy sík és egy egyenes, keressük a metszetüket, azaz a dőféspontot. A feladat megoldása analóg a korábban tárgyalt két egyenes metszéspontjának meghatározásával. Azt feltételezzük, hogy a sík

$$\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}_1) = 0$$

implicit formában, az egyenes pedig

$$\mathbf{x} = \mathbf{p}_2 + t\mathbf{v}$$

paraméteres formában adott. A keresett m metszéspontnak mindkét egyenletet ki kell elégítenie, ezért az első egyenletben az x ismeretlent, a második egyenlet jobb oldalával helyettesíthetjük. Így az

$$\mathbf{n} \cdot (\mathbf{p}_2 - \mathbf{p}_1) - t\mathbf{n} \cdot \mathbf{v} = 0$$

egyenletet kapjuk az ismeretlen t paraméterre. Ha $\mathbf{n} \cdot \mathbf{v} = 0$, akkor az egyenes párhuzamos a síkkal, tehát nincs dőféspont, egyébként pedig

$$t = \frac{\mathbf{n} \cdot (\mathbf{p}_2 - \mathbf{p}_1)}{\mathbf{n} \cdot \mathbf{v}}$$

maga a dőféspont pedig

$$\mathbf{x} = \mathbf{p}_2 + \frac{\mathbf{n} \cdot (\mathbf{p}_2 - \mathbf{p}_1)}{\mathbf{n} \cdot \mathbf{v}}\mathbf{v}.$$

Irodalomjegyzék

- [1] Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F., *Computer Graphics: principles and practice*, second edition, Addison-Wesley, Reading, MA, 1990.
- [2] Juhász, I., *Számítógépi geometria és grafika*, Miskolci Egyetemi Kiadó, Miskolc, 1995.
- [3] Juhász, I., *OpenGL*, elektronikus jegyzet,
<http://iam035.inf.unideb.hu/mobidiak/listdocument.mobi?id=101>, 2003.
- [4] Szirmay-Kalos, L., *Számítógépes grafika*, ComputerBooks, Budapest, 1999.
- [5] Szirmay-Kalos, L., Antal, Gy., Csonka, F., *Háromdimenziós grafika, animáció és játékfejlesztés*, ComputerBooks, Budapest, 2003.