

Képmegjelenítés és animáció 3D rendszereken



Tóth Balázs György – Jakab László



**NEMZETI
KÖZSZOLGÁLATI EGYETEM
BUDAPEST**

SZÉCHENYI 



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

OKOSVÁROS-TECHNOLÓGIÁK
A technológia fejlődésének irányai és hatása
VIII. kötet

Sorozatszerkesztő:

Sallai Gyula

Tóth Balázs György – Jakab László

KÉPMEGJELENÍTÉS ÉS ANIMÁCIÓ 3D RENDSZEREKEN



Nemzeti Közszerológati Egyetem
Közigerzerológati Továbbkérzerológati Intézet
Budapest, 2020

A kötet a Nemzeti Közszolgálati Egyetem **KÖFOP-2.1.2-VEKOP-15-2016-00001** „**A jó kormányzást megalapozó közszolgálat-fejlesztés**” projektje keretében, a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Karán létesült „**Okos város – okos közigazgatás**” kutatóműhelyben (2017/162/BME-VIK) készült.

Szakmai lektor:

Kiss Bálint tszv. egyetemi docens, BME VIK

A kézirat lezárásának dátuma:

2017. december 31.

© Nemzeti Közszolgálati Egyetem
Közigazgatási Továbbképzési Intézet, 2020
© Jakab László, 2020
© Tóth-Balázs György, 2020

A mű szerzői jogilag védett. Minden jog, így különösen a sokszorosítás, terjesztés és fordítás joga fenntartva. A mű a kiadó írásbeli hozzájárulása nélkül részeiben sem reprodukálható, elektronikus rendszerek felhasználásával nem dolgozható fel, azokban nem tárolható, azokkal nem sokszorosítható és nem terjeszthető.

TARTALOMJEGYZÉK

1. Bevezetés	7
2. Képrögzítő és képmegjelenítő technológiák	8
2.1 A megjelenítés minősége	10
2.2 A digitális képrögzítés.....	12
2.2.1 A digitális képfolyamok tömörítése	13
2.3 A digitális megjelenítők	14
2.4 A kiterjesztett valóság eszközei.....	21
3. Virtuális világ	27
3.1 Geometriai modellek	28
3.1.1 Sokszöggeometriák	28
3.1.2 Görbe alapú reprezentációk.....	32
3.1.3 Magasságmezők.....	36
3.1.4 Térfogati modellek	37
3.2 A virtuális objektumok felületi jellemzői	37
3.3 Transzformációk.....	42
3.3.1 Koordinátarendszerek.....	42
3.3.2 Transzformációs hierarchiák	43
4. Animáció	45
4.1 Képlet animáció.....	46
4.2 Keyframe animáció	51
4.3 Pálya animáció	55
4.4 Csúcspont animáció.....	56
4.5 Csontváz animáció	58
4.5.1 Virtuális csontváz	58
4.5.2 Bőrözés.....	61
4.5.3 Bekötési póz	62
4.5.4 Inverz kinematika	63
4.5.5 Animációs állapotok és kombinációik.....	64
4.6 Fizikai szimuláció	65

5. 3D képszintézis	68
5.1 Inkrementális képalkotás.....	68
5.2 Sugárkövetés	71
5.3 Globális illumináció	71
5.4 Tértfogati modellek megjelenítése	72
6. A grafikus hardver	75
6.1 A grafikus hardver története	75
6.2 Képelőállítás a grafikus hardver segítségével.....	76
6.2.1 A grafikus csővezeték	77
6.2.2 Textúrázás.....	79
6.2.3 Blending művelet	82
Irodalomjegyzék	84

BEVEZETÉS

A *Képmegjelenítés és animáció 3D rendszereken* kismonográfia célközönsége az okos város és az e-közigazgatás-fejlesztés témakörében érintett vagy a témakör iránt érdeklődő önkormányzati vezetők, képviselők és szakemberek, az állami tisztviselőkről szóló 2016. évi LII. törvény 1. §-a szerinti állami tisztviselők, egyetemi hallgatók, továbbá a képzésükben, illetve továbbképzésükben részt vevő oktatók és a digitális technológia adta lehetőségek iránt nyitott kutatók-fejlesztők. Célunk a tárgykörök tematikus, jól követhető módon való bemutatása.

A rövid bevezetést követően a 2. fejezetben összefoglaljuk a korszerű képrögzítő és képmegjelenítő technológiákat, ismertetve mindazokat az ismereteket, amelyek a rendszerek működésének, előnyeiknek és korlátaiknak megértéséhez szükséges. Olyan technikai és műszaki megoldásokra térünk ki, ahol a cél a színes, jó minőségű és mozgás bemutatására is képes képek jelenlegi technológiájának összefoglalása, ezért nem foglalkozunk sem a hagyományos film- és papíralapú fényképezéssel, sem a ma már szinte egyáltalán nem használt analóg technikákkal, TV-vel és monitorokkal sem. Külön kitérünk a kép minőségét meghatározó jellemzőkre, a digitális technikák alapelveire, az adatfolyamok csökkentését szolgáló tömörítésekre, majd áttérünk a megjelenítők ismertetésére. A 2. fejezet a kiterjesztett valóság eszközeinek bemutatásával zárul.

A 3. fejezetben a virtuális világ leírásához szükséges geometriai modellek áttekintése után térünk rá azok felületi jellemzőire. A gyorsabb, valósághűbb megjelenítés érdekében bevezetünk koordináta-transzformációkat és a 4. fejezetben az animációk témakörével kapcsolatos ismereteket foglaljuk össze. Az ebben a fejezetben található speciális ismeretek vezetnek el a fizikai szimuláció elveire.

Az 5. fejezetben jutunk el a háromdimenziós képek létrehozásához szükséges módszerekig. Kitérünk a megvilágításhoz és a térfogati modellek megjelenítéséhez szükséges eszközökre, és az ezután következő fejezetben pedig a grafikus hardverek előnyeire.

Reményeink szerint ez a kismonográfia az eredeti célkitűzéseknek megfelelően hozzájárulhat a tématerület iránt érdeklődők ismereteinek bővítéséhez.

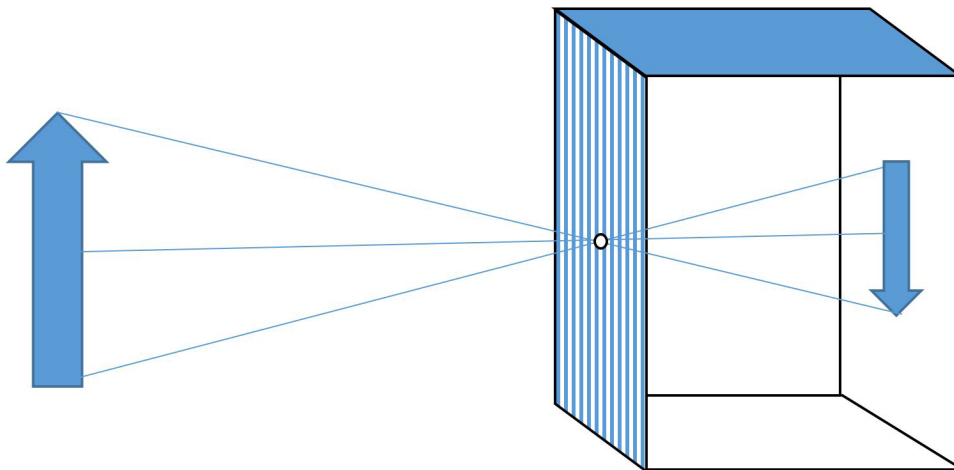
A szerzők¹

¹ Dr. Tóth Balázs György, a BME Villamosmérnöki és Informatikai Kara Irányítástechnika és Informatika Tanszékének egyetemi docense. PhD fokozatát a BME-en 2014-ben szerezte. Kutatási területe a valós idejű számítógépes grafika és a grafikus hardver általános célú számításokra történő alkalmazása.

Prof. Dr. Jakab László, a BME Villamosmérnöki és Informatikai Karának egyetemi tanára, dékánja (2016-2019), az Okos város - okos közigazgatás kutatóműhely szakmai vezetője. Villamosmérnök diplomáját a BME-n szerezte meg 1981-ben. A műszaki tudomány kandidátusa 1992-ben, habilitált doktor 2013-ban, az MTA doktora címet pedig 2014-ben szerezte meg. Jelenleg az Elektronikai Technológia tanszék munkatársa.

2. KÉPRÖGZÍTŐ ÉS KÉPMEGJELENÍTŐ TECHNOLÓGIÁK

A képkötés, képmegjelenítés mérnöki megközelítésben a fényképezéssel kezdődött. Jelen monográfia szempontjából lényeges kiemelnünk, hogy ez a technika, illetve technológia síkbeli ábrázolást, úgynevezett leképezést jelent. Ez egyszerűen visszavezethető a középkori „camera obscura” módszerhez (2.1. ábra), amikor egy zárt doboz egyik oldalán egy kicsi lyukat fúrunk, s a doboz másik oldalán fordított kép jelenik meg, minden különösebb leképezési elem, például lencsék vagy tükrök alkalmazása nélkül.



2.1. ábra. Camera obscura

Forrás: a szerző saját szerkesztése

Ami különleges, hogy a doboz hátsó oldalán a kép minden távolságban, minden dobozméret esetén éles, nincs fókusztávolság és élesre állítás. Az, hogy a kép úgynevezett leképezés, vagy más szóval projekcióval is előállítható, már a középkorban is ismert volt, használtak félig átlátszó papírt, amelyre a képet jellemző vonalak egyszerűen átmásolhatók voltak. Általános szabályokat is le lehetett fektetni, például a párhuzamosak perspektivikus ábrázolásban egy pontban találkoznak, a horizont meghatározása a találkozási pontok által meghatározott vonalban van, és így tovább [Budó 1999].

Azt mindenesetre megállapíthatjuk, hogy térbeli, vagyis háromdimenziós modellek nélkül a térérzékelés megjelenítése síkon sokféle kompromisszum árán valósítható meg, de ez a mai napig az egyik legáltalánosabb és legelterjedtebb módszer, némi kiegészítő tulajdonsággal felruházva. Érdekes, hogy tapasztalataink alapján egy közönséges fényképen is jól tudunk térbeli pozíciókat azonosítani, a képen látottakat térben elrendezni, egyfajta „virtuális” teret a képhez hozzárendelni. Ebben természetesen nagy szerepe van a hely és a tárgyak méretének, szokásos elhelyezésük ismeretének.

Térbeli látásunkat két, egymástól egy bizonyos távolságban elhelyezkedő szemünknek köszönhetjük. Vagyis annak, hogy a két szem különböző pozíciójából két különböző képet látunk, amit agyunk egy nem túl egyszerű algoritmus felhasználásával egyesít, távolságinformációk felruházásával. Ez biológiai vagy evolúciós értelemben a túlélésünket és a fennmaradásunkat is biztosította. Ezt a térbeli látást egyszerűen utánozhatjuk például szemüvegek használatával. Az egyes szem csak a hozzá rendelt képet látja, a két, kismértékben eltérő, különböző szögből, illetve pozícióból felvett kép egy-egy, különböző szemhez rendelése erős térhatás illúzióját teremti (2.2. ábra). Hasonló hatás holografikus módszerek segítségével szemüveg nélkül is elérhető, de ezt napjaink 3D mozijaiban nemigen alkalmazzák. Ennek a technikának az ára magas, a nézőpontok, vagyis a nézők száma korlátozott, a kép mérete, felbontása illetve színmélysége elmarad a szemüveges rendszerekétől. A helyzetet csak súlyosbítja, hogy sokan vannak, akik az efféle virtuális képeket nem szeretik, fejfájásra, rosszullétre panaszkodnak, ez a módszerek általános elterjedését akadályozza. Jelen monográfiának nem célja ennek a problémának, illetve az ehhez kapcsolódó megoldási lehetőségeknek a részletesebb kifejtése, a továbbiakban a második módszerre, a két dimenzióban, síkban megjelenített képi ábrázolások nézőpont szerinti elforgatására, a háromdimenziós virtuális bejárására, egy látszólagos ablakon keresztül szemlélésére koncentrálnunk.



2.2. ábra. Sztereoszkóp szemüvegek filmszínházi felhasználásra. A moziban egyidejűleg két filmet vetítenek a vászonra, a speciális, úgynevezett polarizációs szűrős szemüvegek biztosítják, hogy mindkét szem a számára vetített, kissé különböző nézőpontból készült képet lássa

Forrás: a szerző saját felvétele

Aki látott már, vagy játszott akár mobiltelefonon, számítógépen vagy valamilyen speciális konzolon háromdimenziós játékot, azt tudja, hogy a hatás igen meggyőző, sok esetben valószínű lehet. Ez persze erősen függ az ábrázolás minőségétől, a formák, színek, árnyékok, felületek valóságosságától, a megjelenítés sebességétől.

2.1 A megjelenítés minősége

Első közelítésében megvizsgáljuk a képminőség kérdését, amit az általános vélekedés a kép élességével azonosít. Tekintsünk el az analóg rendszerek képen belül folyamatosan változó intenzitásának kvantitatív jellemzésétől! Éljük most be annyival, hogy az ilyen eszközök leírása bonyolult, nem is minden esetben ad egyértelműen képviselhető, szubjektív szempontoktól független eredményeket. A történelmi fejlődéstől eltekintve kezdjük tehát közvetlenül a digitális képábrázolással.

A digitális technológiában a kép pontokból áll. Ezt bárki könnyen ellenőrizheti: közel kell menni a digitális LCD-TV-hez vagy monitorhoz. Egy bizonyos távolságon belül a vonalak, határoló részletek szétesnek, négyzet alakú elemekre bomlanak. Szemünk felbontó, részletmegkülönböztető képessége határozza meg ezt a távolságot. Átlagos esetben egy méter távolságban két, egymástól egy milliméterre eső pontot tudunk megkülönböztetni. Ehhez persze jó fényviszonyok és megfelelő, kontrasztos színösszeállítások kelljenek. A számok és az arányok szerencsére egyszerűek. Könnyen megjegyezhetjük, hogy egy 1,2 m x 0,6 m képméretű (48 inches) full HD televízió kb. 2000 x 1000 képpontját közelítőleg 1 m távolságból kell nézni ahhoz, hogy a képpontok ne legyenek megkülönböztethetők. Tekinthejtük ezt persze az optimális távolságnak is, közelebből a kép szubjektív élessége csökken, távolabbról pedig a méretek lesznek kisebbek.

Ha egy ilyen úgynevezett full HD kép esetében kétmillió (2000 x 1000) képpontról beszélünk, akkor nem mondtunk semmi a színekről. A kép szürke árnyalatú, az egyes pontok fényességét a hozzájuk rendelendő intenzitásértékkel jellemezzük. Nyolcbites rendszerben kettő a nyolcadikon, vagyis 256 megkülönböztethető intenzitás- vagy fényességértékünk van, amit a korszerű eljárásokban alapszínenként egyenként, egymástól függetlenül határozunk és jelenítünk meg. Megjelenítők, monitorok, TV-képernyők esetén az alapszínek szinte kizárólagosan a vörös, a zöld és a kék. Ezt RGB (red, green, blue) -rendszernek hívják, és a szabvány pontosan meghatározza, hogy a látható fény különböző hullámhosszú komponenseinek mely tartományai számíthatók az egyes színösszetevőkhöz. A kép felvételekor a három színt egyszerű passzív színszűrőkkel választják szét. A képrögzítést nagyon leegyszerűsítve úgy is elképzelhetjük, hogy három különböző CCD kamerával a három képet egyidejűleg, de a három különböző alapszínek megfelelő színszűrőn keresztül vesszük fel. A kamerákban az érzékelőpontok száma 1920 oszlopban és 1080 sorban elrendezve található (ez felel meg a korábban említett kb. 2000 x 1000 képpontnak), ami egyébként éppen az ugyancsak szabvány 6:9-es képaránynak felel meg. Mozgóképek esetén természetesen legalább 25 képet rögzítünk másodpercenként, ennyi feltétlenül szükséges ahhoz, hogy a mozgásokat folyamatosnak érzékeljük. A megjelenítés is lényegében ennek megfelelően történik.

A kijelzőn, ami esetünkben egy full HD monitor vagy TV, a 0,5 mm-es méretű képpontok három különböző színű részből állnak, erről bárki könnyen meggyőződhet, ha egy legalább 1 m x 0,5 m-es méretű vagy annál nagyobb monitor vagy TV képét közelről megnézi. Ennek persze igen jelentős következményei is vannak. A kép 3 x 2 millió képpontból áll, és mindegyik képpontot 8 bit (ami egyébként pontosan 1 bájt adatmennyiségnek felel meg) fényességérték jellemez, akkor egyetlen ilyen kép 6 MB, vagy 48 millió bit adatmennyiséget jelent. Ha ebből másodpercenként 25 képre van szükségünk, ahhoz 150 MB/másodperc, vagy 1,2 Gbit/másodperc adatsebességgel kell az adatokat továbbítanunk, illetve venniük és megjeleníteniük (2.3. ábra). A háromdimenziós megjelenítéshez alapesetben, például a mozikban és bizonyos 3D televíziókban elterjedten használt polarizációs szemüveges megoldás esetén ennek kétszeresére lenne szükség. Egy-egy, egymástól különböző kép kell, egymástól függetlenül, de ugyanakkor egyide-

júleg a két szemünk számára. A jelenleg elterjedt internetsebességek, 10–50 Mbit/másodperc sávszélesség mellett nem tennék lehetővé ezt az adatfolyamot. Tájékoztatásul megemlíthetjük, hogy a Műegyetem belső hálózatán 100 Mbit/másodperc a felhasználók számára biztosított sebesség, mialatt a Villamosmérnöki és Informatikai Kar teljes „I” épülete napközben átlagosan 2-3 Gbit/másodperc sávszélességet használ. Hogyan lehetséges mégis full HD TV-adások átvitele a jelenlegi infrastruktúra és lehetőségek felhasználásával?

CHECKING SATELLITE: HOTBIRD 13°E >> 11681 >> Arte HD																
Complete table of SATELLITE: HOTBIRD 13°E																
<input type="radio"/> rement <input type="radio"/> Synthesis <input checked="" type="radio"/> Display all <input type="radio"/> DLS/Slideshow <input type="radio"/> Graph. VIDEO <input type="radio"/> Graph. AUDIO <input type="radio"/> Graph. DATA <input type="radio"/> Graph. MULTIPLEX <input type="radio"/> Oth																
Last update 17/01/2018 08:23:13																
SID	VIDEO							AUDIO				DATA				
	V-PID	Mode	Bitrate (Mb/s)			Gop	Resolution	Format	A-PID	Lang	Mode	Bitrate (kb/s)	Fe	D-PID	Type	Bitrate (kb/s)
			Min	Ave	Max											
307	571	MPEG4	3.4	5.1	7.3	73	1920x1080	1.77 (16:9)	572	fra	AC3 2/0	128	48.0	575	Sub-title	2
									573	qaa	AC3 2/0	128	48.0	576	Sub-title	2
									574	qad	AC3 2/0	128	48.0	578	Data (AIT)	22
									577	deu	AC3 2/0	128	48.0	579	Data (DSM-CC)	19

2.3. ábra. Az Arte HD műholdcsatorna mért jellemzői. Az 1920 x 1080 full HD felbontású kép átvitele átlagosan 5,1 Mbit/másodperc adatátviteli sebesség mellett történik

Forrás: www.digitalbitrate.com internetes tesztoldal

A választ két irányból is megkaphatjuk. Az egyik rész megoldás magában a képátviteli szabályokban található, alpból nem kötelező minden képváltáshoz a teljes képet átvinni, megjeleníteni. Az időben egymás utáni egyik képen csak a páros, míg a másikban a páratlan sorok jelennek meg, ezeket félképeknek nevezzük, a vizuális élmény alig csökken, különösen lassan változó képeknél, az adatfolyam viszont a felére csökken. A módszert váltott soros megjelenítésnek nevezzük, a betűjele „i” az angol „interlaced” kifejezésből, szemben a minden képet teljes egészében átvivő progresszív, „p” jelű technikával. Jellemző HD (high definition) nagy felbontású méretek és megjelenítési módok a 720p, 1080i illetve az 1080p, ahol a szám a képen az oszlopok számára utal.

A kép- (fénykép) illetve a videófelbontások is nőnek a hardver- és szoftverkomponensek fejlődésével. Új lehetőség a már készülékekben is hozzáférhető UHD (ultra high definition) ultranagy felbontás, amit „4K”-nak is szoktak nevezni, a képméret 3840 x 2160 pont, illetve még ennél is nagyobb az UHD (8K), ahol 7680 x 4320 képponttal, vagyis az előző méret kétszeresével, képterület négyszeresével számolhatunk.

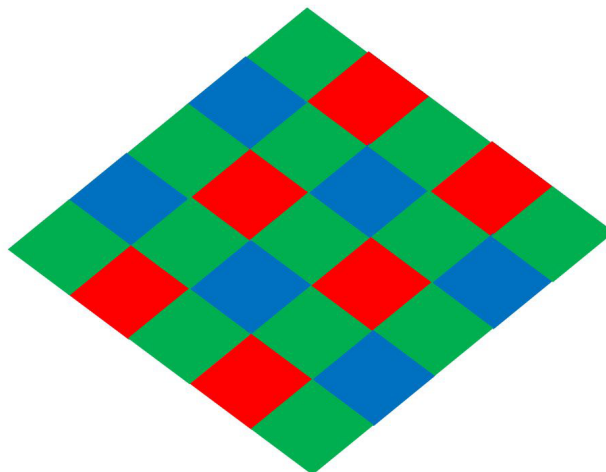
A felbontás és képméret viszonyában egy fontos tényezőre viszont feltétlenül fel kell hívunk a figyelmet. A megfigyelő szempontjából az az előnyös, ha olyan távolságból nézi a képet, ahonnan a képpontokat már nem tudja megkülönböztetni, de a képméret a legnagyobb, vagyis a lehető legközelebb van a legelőnyösebb pozíció. A szemünk felbontása 1 szögperc, ez nem függ a szemlélődés távolságától, vagyis csak a kijelző, monitor, illetve TV egyes kicsi képpontjainak mérete fogja a távolságot meghatározni. A nagyobb felbontásszám nagyobb, a látóteret jobban kitöltő képméretet fog eredményezni. A szemünk szempontjából a 0,3 mm-es képpontmérethez 1m távolság tartozik, a képméret, valamint a sorok, illetve oszlopok számának hányadosa megadja a képpont abszolút méretét, ennek viszonyából a 0,3 mm-hez pedig kiadódik, hogy mennyivel közelebről vagy távolabbról előnyös a kép tanulmányozása.

A második „részmegoldás” az adatok mennyiségének a csökkentésére a három különböző színű alapkép képpontszámának a csökkentése. Erre több módszer is van, a legelterjedtebb az úgynevezett Bayer-szűrés. Ennél a módszernél a zöld kép minden második pixelét (képpontját) sakktáblaszerűen eldobjuk, a kék, illetve a piros komponensnél pedig csak minden negyediket tartjuk meg. Így a három kép összes pixelének száma megegyezik a szürkeárnyaltos képével, az adattartalom, illetve a mennyiség is azonos. Meg kell jegyezni, hogy a módszert elsősorban a képfelvételhez dolgozták ki, kompakt fényképezőgépekben, videókamerákban, mobiltelefonokban alkalmazzák széleskörűen. Mivel szorosan kapcsolódik a témakörhöz, érdekes lehet egy rövid áttekintés a digitális képfelvétel és -rögzítés eszközeiről, illetve módszereiről is. A továbbiakban erre haladunk.

2.2 A digitális képrögzítés

A képérzékelésre ma szinte kizárólagosan félvezetőalapú fényérzékeny detektormátrixokat használnak. A fényérzékelő mátrixok kis méretű detektorokat tartalmaznak – amelyeket a kijelzők alapeleméhez hasonlóan pixeleknek nevezünk – sorba, illetve oszlopba rendezve. A szabványos képméretnek megfelelően például egy full HD érzékelő mátrix 1980 x 1080 darab pixelt tartalmaz. Az érzékelők teljes méretüket tekintve a fényképezőgép, kamera vagy mobiltelefon méretéhez illeszkedő módon 5-10 mm oldalhosszúságúak, amiből következik, az érzékelőmátrix oldalhosszúságát osztva az oszlopok vagy sorok számával, hogy a pixelek 3-7 mikron (ezredmilliméter) méretűek, vagyis az átlagos hajszál tizedével mérhetők össze. Valamennyi érzékelőelemen megvilágítás hatására szabad töltések, elektronok keletkeznek, ezek a töltések megfelelő áramköri környezetben elektromos jellé alakíthatók, vagyis a detektált fényintenzitások időfüggő elektromos feszültség formájában kinyerhetők. A jelenleg elterjedt két alapvető érzékelőtípust a CCD (charge coupled device, vagyis töltéscsatolt) és a CMOS (complementary metal oxid semiconductor, vagyis komplementer fémoxid félvezető) érzékelőket elsősorban a töltések gyűjtése és a kiolvasás módja különbözteti meg egymástól, a felhasználó, vagyis a mi szempontunkból tulajdonságaik nem térnek el jelentősen. Nem meglepő, hogy vannak gyártók, amelyek minkét rendszerben forgalmaznak eszközöket és kész kamerákat is. Az 1980 x 1080 elemű képérzékelő mátrix minden eleme 8 bit-es 255 szintben megkülönböztethető jelet ad, videóalkalmazásokban progresszív információ kiolvasás mellett 25 teljes képet másodpercenként.

Az adatebességre az előző, a képmegjelenítésre számolt sávszélességet kapunk. Vegyük figyelembe, hogy ilyen érzékelő estén még csak egy szürkeárnyaltos (köznyelvi fordulattal fekete-fehér) képünk van! Három ilyen érzékelővel tudunk szabványos RGB-képeket készíteni, azonban a bejövő fény színek szerinti szétválasztása geometriai fényutat, színszűrővel ellátott tükröket, vagyis nagy méreteket eredményez. A HD felbontású televíziós sportközvetítéseknél látható nagyméretű, állványra szerelt, kameraman által mozgatott színes, full HD felbontású kamerák a hatvanas évek analóg fekete-fehér kameráira emlékeztetnek, de ennek nem a kevéssé integrált elektronikai elemek, hanem a megfelelő mennyiségű fényt gyűjtő és szín szerint szétválasztó, a három érzékelőre szétosztó optika az oka. Nyilvánvaló, hogy ez az egyszerű, kompakt kézi fényképezőgépekben, kamerákban, illetve mobiltelefonokban nem alkalmazható technika. A megoldás a Bayer-algoritmus (2.4. ábra) szerinti szűrők integrálása az érzékelőmátrixra.



2.4. ábra. A Bayer-algoritmus szerint kialakított szűrő
 Forrás: a szerző saját szerkesztése

Sakktáblaszerűen például minden világos mező egy zöld szűrőt kap, a sötét mezők pedig egyenletesen elosztva kék és piros színűek lesznek. A kiolvasott képet, amely mindhárom színről információt tartalmaz, szín szerint szétválogatjuk, majd a képből hiányzó elemeket úgynevezett interpolációval, közelítéssel kitöltjük. Ahol üres pixeleket találunk, azokra a közelükben található pixelek értékeiből például lineáris közelítéssel kiszámolható értékeket tekintjük érvényesnek. Zöld pixelből minden második tartalmaz információt, az üres mezőket a két szomszéd számtani középértékével töltjük fel. Az eredmény három, a teljes 3 x 2 megapixel adatot tartalmazó adatfolyam lesz. Persze tudnunk kell, hogy a teljes optikai felbontást csak a teljes képre értelmezhetjük, a zöld információnk ennek csak a fele, a kék és piros képi adatok csak a teljes kép negyedét teszik ki. Vizuálisan a különbség alapból nem jelentős, bár két képet, egy teljes értékűt és egy Bayer-algoritmussal egyszerűsített képet összehasonlítva a különbség azért látható.

Összefoglalva az eddigieket, az eredeti adatmennyiséget a váltott soros képmegjelenítéssel a felére, a Bayer-algoritmussal pedig a maradék harmadára csökkentettük, vagyis némi kompromisszumok árán, de a hatodrésztére vettük vissza. Még a három dimenzióról nincsen szó, de már „csak” 200 Mbit/másodpercnél tartunk.

A tapasztalatok szerint a jó minőségű full HD videófolyamhoz körülbelül 10 Mbit/másodperc sebességre van szükség. Csak érdekességként jegyezzük meg, hogy a 2008-as pekingi olimpia magyarországi kísérleti HD-közvetítéséhez a hazai fogadó állomások optikai üvegszálak kapcsolatán 50 Mbit/másodperces adatfolyamot (streamet) kaptak. Hogyan csökkentjük tovább a szükséges 20-as osztóval, tényezővel az adatmennyiséget?

2.1.1 A digitális képfolyamok tömörítése

A megoldást adattömörítésnek hívjuk. Erre alapvetően két módon van lehetőségünk. Az egyik elv, hogy nem minden képet, vagyis kétmillió bájtnyi információt továbbítunk, hanem csak a képek közötti különbséget. A szükséges képváltási időn belül, a másodperc 25-öd része alatt, néhány sporteseménytől eltekintve nincsenek nagy változások, a kép nagyszámú pixeleinek információja nem változik, és felesleges ezeket minden alkalommal megismételni. Természetesen a változásokra korlátozva az adatfolyamot nagy a

veszélye annak, hogy elveszítjük az alap képi információkat, ezért a teljes képet bizonyos időközönként meg kell ismételni. További lehetőség, hogy egyetlen képen belül sem gyorsak a térben a változások, ritkán fordulnak elő egyetlen pixel szélességű vonalak vagy néhány pixel kiterjedésű, erősen a környezetüktől eltérő területek. Ha csak a térbeli változásokra koncentrálunk, bárki könnyen kipróbálhatja egy egyszerű képszerkesztő programmal, hogy a képtömörítés mértékének növelésével milyen jelentős mértékben csökken a kép adatmérete, és milyen kevéssé romlik a minősége.

Ma a korábbi tömörítési módszerek (mpeg, mpeg2, xvid és egyébek) mellett leginkább elterjedt az MP4-es formátum (H.264 jelöléssel is előfordul). Ennek részletei számunkra most nem annyira fontosak, egy dologra azonban feltétlenül ki kell térni. Mind a kép tömörítése, mind a tömörített adatfolyam kibontása (kicsomagolása) nagyon számításigényes. Ezt a fényképezőgépek, kamerák, telefonok speciális, erre a célra tervezett és gyártott hardver áramköri elemekkel, célszámítógépekkel oldják meg az esetek többségében. Szoftver megoldások is rendelkezésre állnak, de igen erőforrás-igényesek (processzorsebesség, memória, grafikus memória). Ha valaki HD-videók szerkesztésére adja a fejét, annak számolnia kell azzal, hogy szerkeszteni, műveleteket végezni csak a teljesen „kicsomagolt” képeken lehet, és az eredményt utána újra tömöríteni kell, hogy az értelmes méretben tárolható legyen. Nem véletlen, hogy a mai napig lehet szalagos videokamerákat vásárolni, bár a piacon ma már csak kompakt, kis méretű kazettákra rögzítő készülékek elérhetők. A szalagon van hely, a szükségesnél kisebb, enyhe tömörítés pedig egyszerűsíti a feldolgozást, szerkesztést.

2.3 A digitális megjelenítők

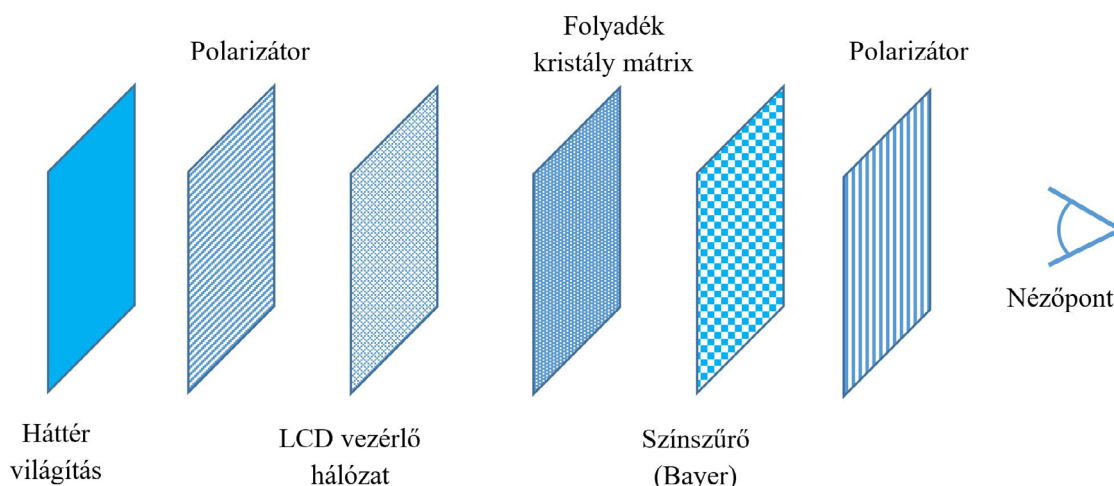
Mivel a monográfia egyik célja a képmegjelenítés technikájának összefoglalása, beszélnünk kell a monitorok, televíziók, mobiltelefon-kijelzők és virtuális szemüvegek működéséről, tulajdonságairól is. A továbbiakban összefoglaljuk a jelenleg elérhető, elterjedtebb módszereket, és kitekintünk a jövőbeli fejlődési irányokra. Nem foglalkozunk a történeti fejlődéssel, ahogy ezt a képrögzítés esetében is tettük az analóg módszerek és technológiák teljes figyelmen kívül hagyásával.

Manapság legelterjedtebben az úgynevezett LCD-eszközökkel találkozunk. Az LCD egy mozaikszó, folyadékkristályos eszközt jelent (liquid crystal device). A folyadékkristályok olyan anyagok, amelyek mechanikai szempontból folyadékhoz hasonlóan viselkednek, de belső rendezettségük van, amely bizonyos tulajdonságok erős irányfüggéséhez (anizotrópiájához) vezethet. Lényeges, hogy külső elektromos térrel szerkezetük, így tulajdonságaik is megváltoztathatók. A hetvenes évek elején jelentek meg a folyadékkristályos kijelzős eszközök, ezek közül például a mai napig kapható órák. Fontos tudnunk, hogy bár sokféle folyadékkristályos anyag létezik, ezek belső szerkezete jelentősen különbözhet is, a mi szempontunkból az elektromosan megváltoztatható optikai tulajdonságaik a meghatározóak. Mit jelent ez?

A folyadékkristályos anyagok a fény polarizációját forgatják el. Azért, hogy ezt pontosan értsük, néhány alapvető fizikai fogalmat tisztáznunk kell. A fény elektromágneses hullám. Nem különbözik a rádióhullámoktól, amelyeket antennákkal sugárzunk ki és fogunk fel, s amelyekben elektromágneses energia terjed, amit elektromos és mágneses térerősséggel írunk le. Az elektromos térerősség az elektromos töltésekre, például az elektronokra ható erővel jellemezhető, ennek azonban iránya is meghatározó. Az erő és így az elektromos térerősség is ezért matematikai szempontból egy vektorral jellemezhető mennyiség, melynek nagysága (abszolút értéke) és iránya is van. Az elektromágneses hullámban mind az elektromos, mind a mágneses térerősség iránya a hullámterjedésre

merőleges, ezért az elektromágneses hullámot transzverzális hullámnak nevezzük. Könnyen belátható, hogy ez egy elég speciális tulajdonság, mert például hanghullámok esetében a levegőben terjedő nyomáskülönbség még csak nem is vektormennyiség, mivel a nyomásnak mint fizikai mennyiségnek nincs kitüntetett iránya, minden irányban azonos érték. Szilárd testekben, például a vasúti sínben, mind a terjedési irányra merőleges változások (deformációk), mind a terjedési iránnyal párhuzamos sűrűség anomáliák, longitudinális hullámok is terjedhetnek, attól függően, hogy milyen irányú a „gerjesztés”. Minden attól függ, milyen irányból ütjük meg például kalapáccsal a sínt. Mindenesetre az elektromágneses hullámoknak nincs longitudinális változata, de szerintem az olvasó sem bánja, ha ennek mélyebb fizikai okaira most nem térnénk ki. Normál esetben a fényforrások által kibocsátott fényben a térerősségvektorok tetszőleges irányba mutathatnak, egy izzólámpa, fénycső vagy természetes fényforrások, például a Nap esetében is a fényben mindenféle térerősségirányok előfordulhatnak. Ezt úgy nevezzük, hogy a fény polarizálatlan, nincsenek benne kitüntetett, rögzített térerősségirányok. Vannak olyan anyagok, jellemzően nagy és hosszú molekulákkal rendelkező polimer szerves vegyületek, amelyek szűrik a polarizálatlan fényt, csak a szerkezetükhöz képest kitüntetett irányú térerősségvektorokat engedik át, a többi elnyelik. Ezekből polarizátorok készíthetők, amelyeket fényképezéshez, üvegfelületekről visszaverődő fény kiszűréséhez jól lehet használni. A polaroid napszemüvegek is hasonló elven működnek, kirakatokról, autószelelvédőkről, ablakokról visszaverődő zavaró fény csökkentésére alkalmasak.

A polarizátoron áthaladó fény viszont már jellemző térerősségirányokkal rendelkezik, ezért polarizáltak nevezzük. Fontos megjegyeznünk, hogy a polarizátorok a rájuk jellemző áteresztési térerősségirányra merőleges polarizációjú fényt nem engedik át, pontosabban fogalmazva erősen csillapítják. Ennek persze az a következménye, hogy két polarizátort egymás után elhelyezve és őket úgy forgatva, hogy az áteresztési polarizációs irányuk egymásra merőleges legyen, az eredmény átlátszatlan lesz. A belépő polarizálatlan megvilágításból az első polarizátor olyan fényt enged át, amit viszont a második blokkol. Helyezzünk a két polarizátor közé egy olyan elektromosan vezérelhető folyadékkristályt, amely a fény polarizációját elforgatja, s máris megkaptunk egy fényintenzitás-modulátort, amivel az egyes képpontok fényességét beállíthatjuk. Így működnek az LCD-kijelzők (2.5. ábra). Korábban a kijelzőkön belül az elektromos elektrodaelrendezéssel szegmenseket alakítottak ki, így lehetett például hét rombusz alakú szegmensen számokat és betűket megjeleníteni (lásd órák). A technológia fejlődésével lehetőség nyílt kis pixel méretben és nagy elemszámmal LCD-megjelenítőket készíteni. Jegyezzük meg, hogy amire feltétlenül szükségünk van, azok a polarizátorok és a megfelelő pixel kialakításhoz szükséges elektrodaelrendezés és azok vezérlése, például kétmillió független pixel esetére másodpercenként 25 alkalommal. Ami viszont még nincs, az a fény, mivel az LCD-struktúráknak nincs önálló, saját fénye. Ráadásul a kijelzőnk még nem színes.



2.5. ábra. Az LCD-megjelenítő felépítése

Forrás: a szerző saját szerkesztése

Az LCD-kijelzőkhöz, megjelenítőkhöz fényforrásra van szükség. Órák esetében a külső nappali fény és az LCD mögé elhelyezett tükör már elegendő a kontrasztos felirat vagy szám kijelzéséhez, monitorok, TV-k, mobiltelefonok esetében háttérmegvilágításra van szükség. Ennek a háttérmegvilágításnak a kijelző teljes felületén egyenletesnek és azonos színűnek kell lennie, a színes megjelenítés érdekében pedig lehetőleg fehérnek. Ennek a háttérvilágításnak a megvalósítása az LCD-monitorok egyik legnehezebb feladata. Korábban a feladatot úgynevezett hidegkatódos fénycsővel, a kijelző háta mögé hajtogatott fehér fénycső „kígyóval”, manapság pedig fehér LED-ekkel, fénykibocsátó diódákkal oldják meg.

A fénycsövek alacsony nyomású gázkisülést alkalmazó fluoreszcens fényforrások. A fénycsőben 7-10 mbar, vagyis a légköri nyomás körülbelül egy századának megfelelő nemesgáztöltés van. Ez korábban neongáz volt, ezért is hívta a magyar köznyelv neonsöveknek ezeket a lámpákat. A neonnak a fénykibocsátásban nincs szerepe, csupán semleges hordozója annak a higanygáznak, amely az elektromos gerjesztés következtében az ultraibolya hullámhossz tartományban 254 nm hullámhosszon sugároz. Szobahőmérsékleten a higany akár folyékony halmazállapotba is visszarendeződhet, ezért kikapcsolt állapotban kompakt fénycsövekben kicsi rázogatóra kopogó hangot is adhat. A higany által kibocsátott ultraibolya fény gerjeszti a fénycső falára felvitt úgynevezett fluoreszcens, megvilágítás hatására más hullámhosszon világító fényport, ami válaszképpen a látható tartományban sugározni kezd. A fénycső színe a felvitt fluoreszcens anyagok összetételétől függ, ezek bonyolult, a foszforhoz hasonló tulajdonságú vegyületek, a fehér fény a különböző komponensek kék, zöld és piros fényéből tevődik majd össze. Az eddigiekből következik, hogy a kijelzők, monitorok vastagságát éppen a háttérvilágításhoz szükséges elemek határozták meg. Normál, világításban is használt fénycsövekben izzó, meleg katódot használnak, alacsonyabb gyújtási és üzemelési feszültség és jó hatásfok kapcsolódik hozzá, kijelzőkben melegedési és élettartam-megfontolások miatt a hidegkatódos elrendezések terjedtek el.

A fénycsöves háttérvilágításokat fokozatosan, de teljesen kiszorítják a LED-es megoldások. A LED-TV, illetve kijelző elnevezés kicsit félrevezető. Látni fogjuk később, hogy önálló fényű pixelekkal rendelkező úgynevezett OLED-megoldások is elterjedőben vannak, amelyek mind felépítésükben, mind működésükben jelentősen különböznek

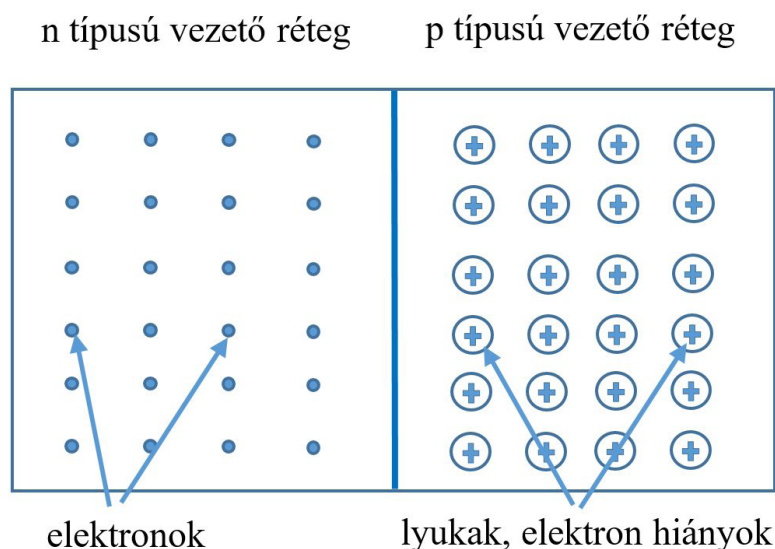
az LCD-monitoroktól, és hosszú távon például a virtuális szemüvegek alapelemeivé válhatnak. De ne ugorjunk még nagyon előre!

A LED, ahogy erre korábban is utaltunk „light emitting diode”, vagyis fénykibocsátó dióda [SALEH 1991]. Az elnevezés pontosan leírja az eszközt, ami egy félvezető dióda, két elektródával. Elektromos kapcsolattal rendelkező egyenirányító elem, a szó klaszikus értelmében. Félvezető anyagból készül. Az anyagokat vezetési tulajdonságaik alapján három csoportba osztjuk. A félvezetők elektromos vezetőképességük vagy ellenállásuk alapján a jól vezető fémek és a jól szigetelő dielektrikumok között találhatók. Bennük a szabad töltéshordozók mennyisége alacsonyabb, mint a fémekben, és lényegesen magasabb, mint a szigetelőkben, ahol gyakorlatilag nincsenek szabad töltéshordozók. A három csoport közötti különbséget úgynevezett sáv szerkezet diagramon szoktuk szemléltetni, ahol az anyagban található elektronenergiákat ábrázoljuk a függőleges tengelyen, és az energia függvényében megkülönböztetünk tartományokat. A legalacsonyabb tartomány az úgynevezett vegyértéksáv, itt található az anyagban az alkotórészek atommagjaihoz kötött elektronok, beleértve a külső elektronpályák elektronjait is. A legfelső tartomány a vezetési sáv, ahol a magasabb energiájú, gerjesztett vagy szabad elektronok energiaszintjeit ábrázoljuk. A két tartományt egy úgynevezett tiltott sáv választja el egymástól, ebben az energiasávban nem lehetnek elektronok, ennek alapvetően kvantummechanikai okai vannak. A jelenség arra vezethető vissza, hogy egyetlen atomban az elektronenergia szintjei sem vehetnek fel tetszőleges értékeket, az elektron „pályák” kvantáltak. Ez minden elemre igaz. Dielektrikumok vagy szigetelők esetén az anyagban elektronok gyakorlatilag csak a vegyértéksávban találhatóak, szabad, elektromos vezetésre alkalmas töltéshordozók nincsenek. Ezen a helyzeten megelégedéssel, vagyis energiaközléssel kis mértékben lehet javítani, de az anyag gyakorlatilag szigetelő marad széles hőmérséklet-tartományban. Ilyen anyagok az üvegek, kerámiák. Fémekben a vezetési sáv és a vegyértéksáv átfed, a tiltott sáv eltűnik, nagy mennyiségű szabad töltéshordozó, elektron áll rendelkezésre, kis elektromos ellenállású vezetésre alkalmasak, mint például a réz és az alumínium. Félvezetőkben a tiltott sáv közepesen széles, így szobahőmérsékleten is található elektronok a vezetési sávban, ami alpból egy, a fémeknél gyengébb, de a szigetelőknél lényegesen jobb elektromos vezetést tesz lehetővé. A hőmérséklet növelésével a vezetési sáv elektronjainak száma gyorsan nő, a vezetési tulajdonságok javulnak. Érdekes, hogy fémekben a hőmérséklet növekedése éppen ellentétes hatású, mivel a növekvő hőmérséklet az atomok hőmozgását növeli, így azok akadályozzák az elektronokat az elektromos vezetésben. Az elemi félvezetők a periódusos rendszer IV. oszlopában találhatóak. Fontos megjegyeznünk, hogy a félvezetőket egykristályos szerkezetbe rendezve használjuk fel. Az egykristályok olyan belső szerkezettel rendelkeznek, ahol az atomok egymáshoz képest egy elemi cellának nevezett térfogatban egymáshoz képest mindig azonos pozícióban helyezkednek el, és a teljes anyag ennek az elemi cellának a geometriai eltolásával, ismétlésével „előállítható”. Fizikailag természetesen az egykristálykészítés speciális eljárást, olvadékból, illetve oldatból való kikristályosítást jelent.

Különleges tulajdonsága a félvezetőknek, hogy vezetési tulajdonságaik adalékolással, bizonyos anyagoknak a félvezetőkbe történő bevitelével megváltoztathatók. Az adalékanyagok egyik csoportját donoroknak nevezzük. Ezek, a periódusos rendszer V. oszlopában található elemek a külső elektronhéjukon eggyel több elektront tartalmaznak, mint az elemi félvezetők. Ezek a többletelektronok felszabadulnak, amikor a donoranyagok beépülnek a félvezetők kristályrácsába. Legjellemzőbb donor-adalékanyag a foszfor. A donorral adalékolt félvezetőket „n” (negatív) típusú félvezetőknek nevezzük. Az adalékok másik csoportja az akceptorok. Az akceptorok a periódusos rendszer III. oszlopának

elemei, külső elektronhéjukon eggyel kevesebb elektron található, mint az elemi félvezetőkben. Beépülve a félvezetők kristályrácsába ezek mellett az adalékatomok mellett a vegyértéksávban egy elektronhely keletkezik. Ilyen anyag jellemzően a bór. Akceptorok esetén a helyzet elég különleges, mivel szabad töltéshordozó nem keletkezik, az anyag vezetőképessége mégis javul. Ennek oka, hogy ezeket az elektronhelyeket a szomszéd atom elektronjai könnyen be tudják tölteni. Ekkor ott keletkezik egy elektronhiány, amit lyuknak is nevezünk. Ez a lyuk szabad pozitív töltésként tud „mozogni” az anyagban, pontosan ellentétes irányba, mint ahogy a lyukakat betöltő elektronok teszik. A jelenséget lyukvezetésnek nevezzük. Az ilyen félvezetőt „p” (pozitív) típusú félvezetőnek hívják.

Nézzük meg, mi történik akkor, ha egy adalékolt „n” és „p” típusú félvezető párra feszültséget kapcsolunk. Abban az esetben, amikor a pozitív feszültség a „p” típusú, a negatív pedig az „n” típusú oldalon van, a pozitív feszültség taszítja a lyukakat és vonzza az elektronokat, míg a másik, a negatív oldal pont ellenkezően viselkedik. Az anyagra kapcsolt elektromos tér egymás felé tereli a lyukakat és az elektronokat, amelyek a két anyag határfelületén rekombinálódnak, vagyis a vezetési sáv elektronjai betöltik a vegyértéksáv elektronhiányait. Az elektronok vezetési sávnyi energiájukat elveszítik, ez általában az anyag melegedését okozza. Az ily módon kapcsolt feszültséget nyitóiránynak, az áramot pedig nyitóirányú áramnak nevezzük. Ellentétes polarizáció, vagyis feszültségek esetén a „p” oldali negatív feszültség magához vonzza a „lyukakat”, az „n” oldali pozitív feszültség pedig az elektronokat. A két töltéshordozó-típus egymástól eltávolodik, az elektromos tér őket szétválasztja, az elektromos vezetés megszűnik. Ez a záróirány-polarizáció, vagy feszültség esete, a záróirányú áram nagyságrendekkel kisebb a nyitóirányúnál, így működik, persze egy kicsit leegyszerűsítve, az egyenirányító félvezető dióda.



2.6. ábra. A p-n átmenet

Forrás: a szerző saját szerkesztése

Említettük, hogy a vezetési sávból az elektron a vegyértéksávba kerül, energiáját el kell veszítenie, azt a környezetének, a körülötte található atomoknak adja át. Ezek energiája, rácsrezgése, vagyis hőmérséklete nőni fog. Lehet-e ebből az energiából fénykibocsátás? Elveszítheti-e úgy ez az elektron az energiáját, hogy közbe egy foton keletkezik?

A válasz igen, bár elemi félvezetőben a jelenség valószínűsége igen kicsi, tízezer rekombinációból jó, ha egy jár fénykibocsátással. A jelenség nemhogy fényforrásnak nem alkalmas, de még kimutatni sem könnyű, mivel a foton mélyen az anyagban, a két különböző típusú félvezető határfelületén (amit egyébként p-n átmenetnek nevezünk [2.6. ábra]) keletkezik és utána a környező atomokon el is nyelődik, abszorbeálódik. Ez az elemi félvezetők anyag-, illetve sáv szerkezetének a következménye. Szerencsére vegyület-félvezetőkben a helyzet ettől különböző. A vegyület-félvezetők a periódusos rendszer III. és V. oszlopából, illetve a II. és VI. oszlop elemeiből, elempárjaiból állíthatók össze. Legtipikusabb a GaAs, a gallium-arszenid, amely az elemi félvezetőkhez hasonló sáv szerkezettel rendelkezik, azokhoz hasonlóan adalékolható, p-n átmenetet képezve belőle dióda-ként használható, de a mi szempontunk igen előnyös tulajdonsága, hogy a nyitóirányú áramot biztosító töltéshordozó-rekombináció során nagy valószínűséggel fotonok keletkeznek, vagyis a jelenség fény kibocsátásával jár. A keletkezett, kibocsátott foton hullámhossza az anyag tiltott sáv-szélességének a függvénye, gallium-arszenid esetében 870 nm, ami a látható 400–700 nm-es hullámhossz-tartományon kicsit kívül esik, és a közeli infravörösben található. Ha visszagondolunk eredeti célkitűzésünkre, hogy LCD-kijelzőkhöz fehér háttérvilágítást keresünk, hát ez a fény semmiképpen nem fehér.

A LED-ek esetében természetesen még meg kellett oldani, hogy az anyag belsejében keletkező fény elnyelődés nélkül kiszabaduljon a diódából, de az ilyen elektronikai technológiai részletek már tényleg túlmutatnak e monográfia tárgyán és célkitűzésein.

Ahogy említettük, a kibocsátott fény hullámhossza a tiltott sáv szélességének függvénye. A látható, piros, zöld és kék színekhez olyan vegyület-félvezetőkre van szükség, amelyekben a tiltott sáv szélessége ezeknek a hullámhosszaknak megfelel. Ezek három, illetve négy összetevőjű vegyületek, ahol az összetevők arányával kis mértékben, de a LED színe is állítható. Piros fény keltésére alkalmas a GaAlAs, gallium-alumínium-arszenid, zöld fényt bocsát ki a GaAsP, gallium-arszenid-foszfid vagy a GaP, gallium-foszfid, míg a kék fényhez egészen a kilencvenes évekig kellett várni, a SiC, szilícium-karbid, illetve az InGaN, indium gallium nitrid anyagok technológiájának kidolgozásáig.

Itt kell megjegyeznünk, hogy a fehér fény keltéséhez félvezető fényforrások esetén is van lehetőség a fluoreszcencia jelenségét a fénycsövekhez hasonlóan kihasználni. Egyetlen kék LED és megfelelő fényporok kombinációjával a zöld és piros fénykomponens is előállítható, a nagy teljesítményű fehér LED-fényforrások esetében, amelyeket világításra is használhatunk, ez egy elterjedt technológia [Feynman 1985].

Eljutottunk odáig, hogy félvezető fényforrás háttérvilágítással rendelkező, most már technológiai szempontból nagy méretben és nagy felbontással, illetve pixelsűrűséggel is rendelkező LCD-kijelzőket, -monitorokat, -TV-eket tudunk készíteni. Ez jelenleg a leggyakoribb technika, amely mobiltelefon-kijelzőkben, okos-órákban egyaránt megtalálható.

Lehet-e olyan kijelzőket készíteni, amelyeknek minden eleme, pixele saját fényrel rendelkezik, vagyis nincs szükség háttérvilágításra és súlyos fényvesztéseket okozó polarizátorrétegekre. Nagyobb méretek esetén a válasz egyszerű, hiszen néhány milliméteres diszkrét LED-elemekből is lehet nagyméretű paneleket készíteni, ezek alkalmasak lehetnek stadionokban, buszmegállóknak még akár képek, videók továbbítására is. Kisebb, milliméter alatti tartományban azonban valami másra van szükség, olyan LED-megoldásokra, ahol integráltan valósíthatók meg a kijelző-, képmegjelenítő funkciók, és ezzel el is jutottunk a mai legkorszerűbb megoldásokhoz, az OLED-, vagyis „organic”, szerves LED-rendszerekhez.

A dolog kémiája elég bonyolult, ezért erre most nem térünk ki. Fizikai szempontból az OLED-panelek egyetlen, általában üveg hordozóra integrált polimer félvezető diódamátrix-rendszerek, ahol az egyes, különböző színű, piros, kék, zöld pixelek intenzi-

tásának egymástól független vezérlését is a hordozón kell megvalósítani oly módon, hogy a rendszer széles környezeti hőmérséklettartományban és hosszú élettartamot biztosítva legyen hibátlanul működőképes. Ezek a rendszerek kis méretekből megtalálhatók mobiltelefonokon, okosórákon, nagyobb méretben kijelzők, TV-k is kaphatók, bár még nagyon drágák.

OLED-példának álljon itt a Sony 21 M Ft-ért árult televíziója (2017. októberi kiskereskedői ár), a teljes műszaki specifikációjával.

Képernyő	
TV-képernyő típusa	LED
Képátló (cm)	252 cm
Képátló (inch)	100 inch
Ívelt kijelző	Nincs
Képarány	16:9
HD-felbontás	Ultra HD
Felbontás	3840 x 2160 pixel
Betekintési szög	178 °
3D megjelenítés	
3D Tv	Igen
3D technológia	Aktív szemüveges
2D - 3D konverzió	Van
Tuner	
DVB-T	Van
DVB-T2	Van
DVB-C	Van
DVB-C2	Nincs
DVB-S	Van
DVB-S2	Van
Elektronikus műsorújság (EPG)	Van
Teletext	Van
PAP	Nincs
Hang	
Hangteljesítmény	2x10W
Dolby Digital	Van
DTS	Nincs
SRS	Nincs
Funkciók	
HDMI-CEC-támogatás	Van
DLNA-támogatás	Nincs
PVR-támogatás	Van
Böngésző	Van
Vezeték nélküli szabványok	Wi-Fi Direct

Csatlakozók	
	HDMI
	Scart
	Kompozit
	
	USB-csatlakozó
	Ethernet
	
	Wi-Fi
	Optikai hangkimenet
	Koaxiális hangkimenet
Fejhallgató-csatlakozó	
CI+ interfész	
Méreték és fogyasztás	
Szélesség (talppal)	225.8 cm
Magasság (talppal)	139.3 cm
Mélység (talppal)	50.4 cm
Súly (talppal)	144.8 kg
Energiafogyasztás bekapcsolva	413 W
Éves energiafogyasztás	573 kWh/év
Energiaosztály	B
További tulajdonságok	
HDMI-bemenetek száma	4 db
Scart-bemenetek száma	1 db
USB-portok száma	3 db

Ez ma minden szempontból csúcskategóriájú készülék, nem csak az ára miatt. Vegyük észre viszont, hogy 145 kg tömegű és bekapcsolt állapotban 0,5 kW fogyasztású!

Ami nagyon lényeges, hogy ezen az OLED-elven már viszonylag egyszerű a virtuálisvalóság-szemüvegek készítése is. Van lehetőség olyan átlátszó rétegek létrehozására, hogy a virtuális képek a valós környezettel együtt jelenjenek meg.

2.4 A kiterjesztett valóság eszközei

A kijelző, megjelenítő lehetőségek mellett szükségünk van más funkciók megvalósítására is az olyan eszközök okosításához, intelligensé tételéhez, mint a mobiltelefonok, virtuális szemüvegek, okosórák stb. Ezek a funkciók, illetve technológiák a pozíció és helyzet meghatározása, a mozgás, a haladó és forgó mozgás érzékelése, a gyorsulás meghatározása és mérése.

Kezdjük a pozíció, helyzet meghatározással, amelyre szabad térben a GPS, a Global Positioning System nyújt lehetőséget. A rendszer alapja a 20 km magasságban keringő 24 műhold, amelyek keringési ideje rövid, csupán 12 óra, de három-négy műhold a nap 24 órájában a föld bármely pontjáról látható. A műholdak saját atomórával felszerelve pontos ütemezésben jeleket sugároznak, amelyek tartalmazzák a műhold pontos helyzetét a kisugárzás pillanatában. A földi GPS-vevő a jeleket a műholdtól való távolság függvényében különböző időpontban veszi. A vétel időpontjából és az üzenet adattartalmából, vagyis a műhold üzenetküldésének időpontjából, illetve pontos helyzetéből a megfigyelő és a műhold távolsága meghatározható. Több műholdkapcsolat esetén az

eszköz a helyzetet háromszögeléssel kiszámolja, nemcsak a földrajzi, földfelszíni koordináták, de a magasság értékek is meghatározhatók. A rendszer jellemzően néhány méter pontos, de kiegészítő elemekkel pontossága jelentősen növelhető. Az amerikai GPS-rendszer mellett épül az európai, az orosz és a kínai megvalósítás is, ennek elsősorban stratégiai jelentősége van, de a rendszerek jól ki is egészíthetik egymást.

A készülék mozgásának, elforgatásának, a kijelzőre merőleges irányának meghatározására szinte kizárólagosan úgynevezett MEMs-érzékelőket, szenzorokat alkalmaznak. A MEMs elnevezés a mikro-elektro-mechanikai rendszer rövidítése, és ilyen elemek készítését kis méretekben a szilícium félvezető technológia fejlődése és a szilícium irányfüggő marásának, maratásának a lehetősége teremtette meg. Az érzékelő elemek geometriáját minden esetben magán a szilíciumhordozón alakítják ki, az esetek többségében membrán, konzol (félhíd), illetve elektródák között lengő kis tömeg formájában. A teljes méretek a 10 mikrontól az 1 mm-ig terjedhetnek, a technológia szükség esetén 1 mikron alatti részletek kidolgozását is lehetővé teszi. A mérés elve, a mozgás elektromos jellé alakítása többféle módon történhet, ennek részletei jelen monográfiában nem lényegesek. Elég tudnunk, hogy a Föld mágneses terének és a gravitációs gyorsulás irányának egyidejű érzékelésével az olyan egyszerű mobil alkalmazásoktól, mint az iránytű vagy a vízmérték egészen a mozgás háromdimenziós időbeli leírása is lehetséges. Egy korszerű és okos eszközben akár több mint tíz különböző MEMs-érzékelő is lehet, ezek több pontban hőmérsékletet és nyomást is mérnek. Például a légköri nyomás változásából az okosóra meg tudja mondani hány emeletet másztunk meg, miközben a gyaloglásra, lépésre utaló mozgások, gyorsulások hiányában a liftezést figyelmen kívül hagyja.

Kiterjesztett valóság (Augmented Reality, vagy AR) alkalmazások alatt olyan rendszereket értünk, amelyekből az egyszerűbbek például mobiltelefonra már most is elérhetők, s ahol a kamera valódi képére olyan többletinformációkat helyezünk, mint például a képen látható elemek, tárgyak egyéb adatai. De lehet ilyen a közüzemi szolgáltatók számára kifejlesztett rendszer, amely például az elektromos vagy a csatornahálózat specifikus adatait jeleníti meg a valós helyszínen. Minden ilyen alkalmazás esetén fontos kiemelnünk, ahogy az eddigiekben elég részletesen is tárgyaltuk, hogy mind a pontos kamerahelyzet és -irány gyors meghatározása, mind pedig a képpel együtt mozgatott óriási adatmennyiség kezelése nagy kihívás.

Virtuális valóság (Virtual Reality, VR) alatt a számítógép által létrehozott, mesterséges környezetet értjük, amely lehet nagyon részletesen kidolgozott, s amelyben akár három dimenzióban is tudunk közlekedni, és betekinteni az egyes részletekbe. A virtuális valóság megtapasztalásához szimulátorra van szükség, amely egyszerű esetben egy háromdimenziós sztereószemüveg is lehet, fejlettebb változatban repülőgép-szimulátorokhoz hasonló készülékekbe beülve mozgás-, gyorsulás- és repülésélmény is átélhető. Nagyobb méretekben egy digitálisan elkészített háromdimenziós mozifilm egyszerű, passzív szemüvegen keresztül nézve is erős virtuálisvalóság-élményt tud adni.

Mind a kiterjesztettvalóság-, mind a virtuálisvalóság-rendszerekben és -alkalmazásokban lehetőség van az interaktivitás megteremtésére, vagyis a felhasználó a virtuális világban önállóan dönt, tájékozódik, és esetleg az eseményeket is irányítja, irányíthatja. A képminőség, színhúság elérése, a megvilágítás, árnyékhatások, mozgások szimulációja magas szintű számítógépes grafikai ismereteket és óriási számítási kapacitást feltételez, ezekre a célokra gyakran a szuperszámítógépekkel összemérhető sebességű és kapacitású grafikai processzorokat (GPU-kat) használunk. Mindezek ismertetésére részletesen kitérünk a későbbi fejezetekben.

Ha egyszerűbb háromdimenziós alkalmazásokban gondolkodunk, a hagyományos számítógépek és monitorok, kijelzők mellett is elérhetők olyan szoftvermegoldások, ahol lehetőség van térképek és rajzok alapján digitalizált adatok 3D megjelenítésére. Például csőhálózat, elektromosvezeték-rendszer, szellőzőcsatornák valós idejű ábrázolása, vagy számítógéptomográf-, illetve MRI-felvételek tanulmányozása, ahol a rendszeren vagy a felvételen belüli térbeli mozgásra lehetőségünk van, a hálózatot vagy a vizsgálati eredményt különböző irányokból és méretben megtekinthetjük.

A háromdimenziós tervezőprogramok akár az építészetben, a gépészeti tervezésben akár az autópári fejlesztésekben is igen elterjedtek. Ma az autó, minden alkatrészével, minden elemével együtt először a virtuális valóságban, a számítógépben készül el. Természetesen az egyes rendszerelemek integrációjával nagyobb rendszereket is építhetünk, amelyben szimulációkra van lehetőség, például útfelbontások hatásának elemzése a közlekedésre, tűz terjedése akár különböző időjárási szituációkban épületekben és épületek között. Természeti katasztrófa esetére részletes tervek készíthetők az ilyen adatok felhasználásával.

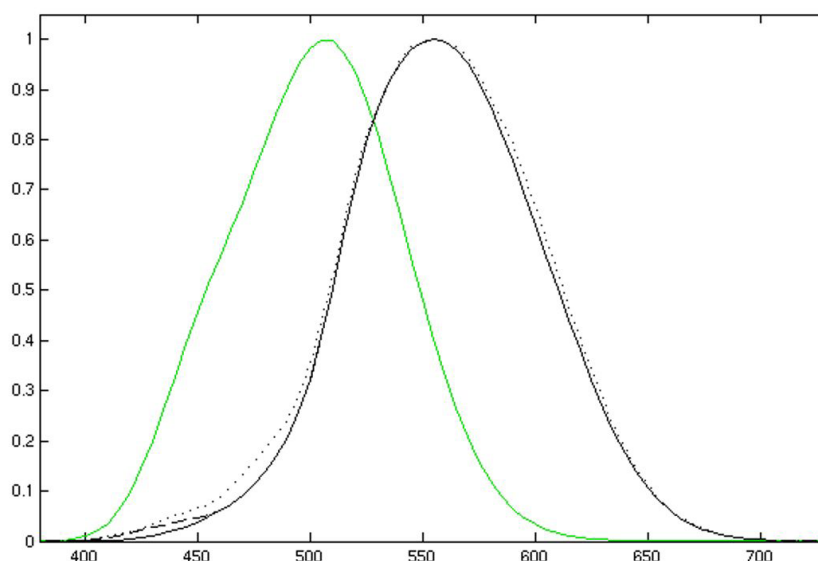
Sok olyan alkalmazást is találunk, ahol nincs feltétlenül szükség a magas szintű interaktivitásra, viszont szempont a képminőség és hűség, például látványtervezéshez vagy oktató animációkhoz.

Elmondhatjuk, hogy a 3D technológiák a mindennapi élet részévé váltak, ezért fontos ismerni a lehetőségeket és korlátokat.

A kijelzők fizikai alapjai témakörben a továbbiakban röviden összefoglaljuk mindazokat az információkat, egy kicsit technikai szempontból is, hogy milyen felhasználói szintű paraméterek várhatók el a jelenleg hozzáférhető rendszerektől [Ábrahám szerk. 1997].

Kijelző által felvett teljesítmény az elfogyasztott elektromos energia adott idő (egy másodperc) alatt. Mértékegysége a watt (W). A fényerősség az a látható (fény)energia (intenzitás), amelyet a fényforrás egy 1 m sugarú gömbfelületen átsugároz. Mértékegysége a kandela (cd). Ennek SI-meghatározása: A kandela az olyan fényforrás fényerőssége, amely 540 000 GHz (555 nm) frekvenciájú monokromatikus fényt bocsát ki, és sugárerőssége ebben az irányban 1/683-ad watt per szteradián. A fényáram: 1 kandela minden irányban egyenletes, pontszerű fényforrás által 1 szteradián térszögbe sugárzott teljesítmény, ennek mértékegysége a lumen (lm). A megvilágítás az egy négyzetméter felületre eső látható fényáram, ennek mértékegysége a lux (lx).

A $V(\lambda)$ görbe az emberi szem fiziológiai válaszfüggvénye a hullámhossz függvényében (1.7. ábra). Megkülönböztetünk fotopikus (fekete) választ a „nagy” intenzitások esetében és szkotopikus (zöld) választ a „kis” intenzitások esetében. Érdekes megjegyeznünk, hogy a $V(\lambda)$ meglehetősen pontosan követi a Nap sugárzásának hullámhosszfüggését, vagyis a szemünk pontosan ott érzékeny a fényre, ahol a környezetben a legtöbb fény található.



2.7. ábra. Az emberi szem érzékenysége a hullámhossz függvényében
 Forrás: https://en.wikipedia.org/wiki/Luminosity_function

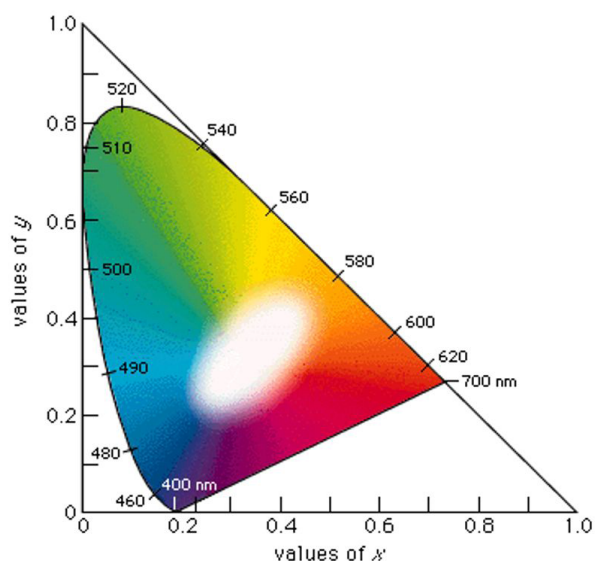
A kijelző eszközeink leírására világítástechnikai paramétereket használunk. Megvilágításra (és kandelára) példák: az 1 kandela kb. egy gyertya fénye, 100 wattos izzó kb. 120 cd, 20 lux alatt gyöngye megvilágításnak számít, 200 lux lenne a nappali megvilágítás, 1000 luxot használunk futballpályákon, 10 000-25 000 lux a napfény. Ide tartozik még a kontraszt, ami $C = (B1 - B2) / B1$, ahol B1 és B2 a világosabb és a sötétebb pixelek/tartományok fényűrsége. Ennek nincs mértékegysége, annál jobb a láthatóság, minél inkább az egyhez van közel. Az éppen elfogadható láthatóság a $C = 0,5$. A kontrasztarány pedig a $CR = B1 / B2$, amit például az LCD-paneleknél adnak meg.

A színek jellemzésére több különböző szabvány létezik. A magyar coloroid színrendszerben megkülönböztetjük a színezetet (Hue) – ez a tulajdonképpeni szín, a telítettséget (Saturation), a világosságot (Brightness). Ezen felül elterjedt színrendszer a már korábban említett RGB (red, green, blue), amit additívnak nevezünk, mivel a színek komponensek összeadódva adják ki az egyes színeket. A három színt 256 (8 bit) különböző szinten megjelenítve éppen azt a 16 millió színértéket kapjuk, ami gyakran a monitorokon feliratként szerepel. A CYMK (cyan, yellow, magenta, black – vagy key) rendszer szubtraktív, vagyis a papírra nyomtatott színeket a visszavert fényből hiányzó színek komponensek fogják meghatározni.

A színdiagramban a három alapszín az R (700 nm), G (564,1 nm), B (435 nm). A három színkoordinátát ábrázoljuk egy transzformáció segítségével:

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z}$$

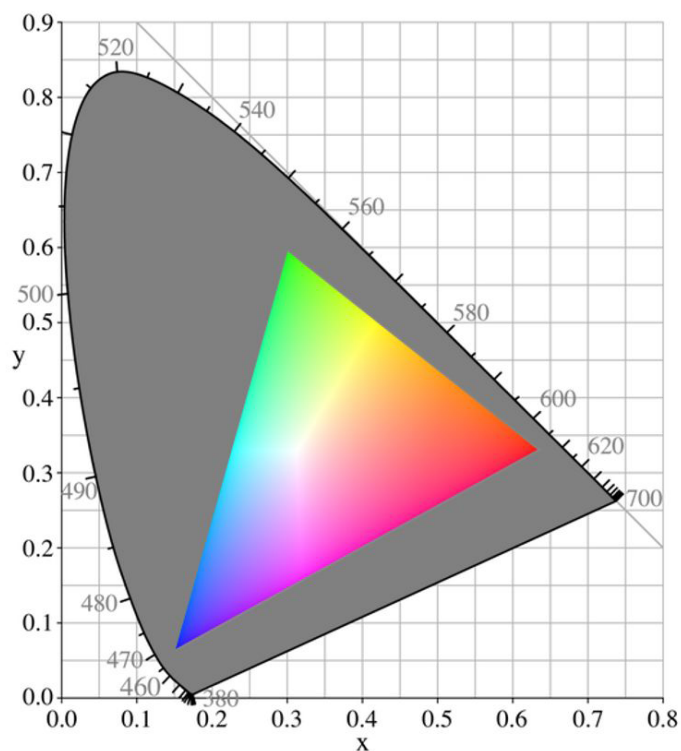
Az eredmény az alábbi ábrán látható:



2.8. ábra. Színkoordináták a transzformáció nyomán

Forrás: Nemzetközi Világítástechnikai Bizottság
(International Commission on Illumination) CIE 1931 szabvány

A gamut, vagy színkorlát fogalmához lényeges, hogy a monitor típusától függ, hogy a színkorlát mekkora hányadát teszi elérhetetlenné a teljes színtartománynak, a CRT (katódsugárcsöves eszközök) esetében a foszforok típusa adja meg a gamut három csúcsát, az LCD esetében ez a háttérvilágítás spektrumától függ [Ábrahám szerk. 1997].



2.9. ábra. Készülékfüggő színkorlát a színtéren belül

Forrás: https://en.wikipedia.org/wiki/CIE_1931_color_space

A képmegjelenítők jellemzői a képméret, ami lehet szabványos oldalarányú 4:3 vagy 16:9, a felbontás az egyenlő képpontok száma a kijelző teljes oldalhosszára, a pixelméret az egy képpont átmérőjének vagy két oldalának a hossza, a képváltási idő a két egymást követő kép között eltelt idő, a képváltási frekvencia pedig a másodpercenként megjelenő képek száma.

A színvisszaadás az egy pixellel megjeleníthető színárnyalatok száma, ez RGB-rendszerben N_3 , ahol N a „szürkeszintek” száma. Láttuk korábban, hogy 8 bites rendszerben ez a 16 millió szín.

Vegyük észre, hogy éppen az előzőekben bemutatott nagyértékű OLED-televízió esetében semmi információ nincs a specifikációban a színhűség és a színkorlát paramétereiről. Ezeket felhasználói szinten, az alkalmazás függvényében magunknak kell meghatároznunk. A fejezet zárásaként álljon itt a Graphisoft Archicad tervezőszoftverjének képe.

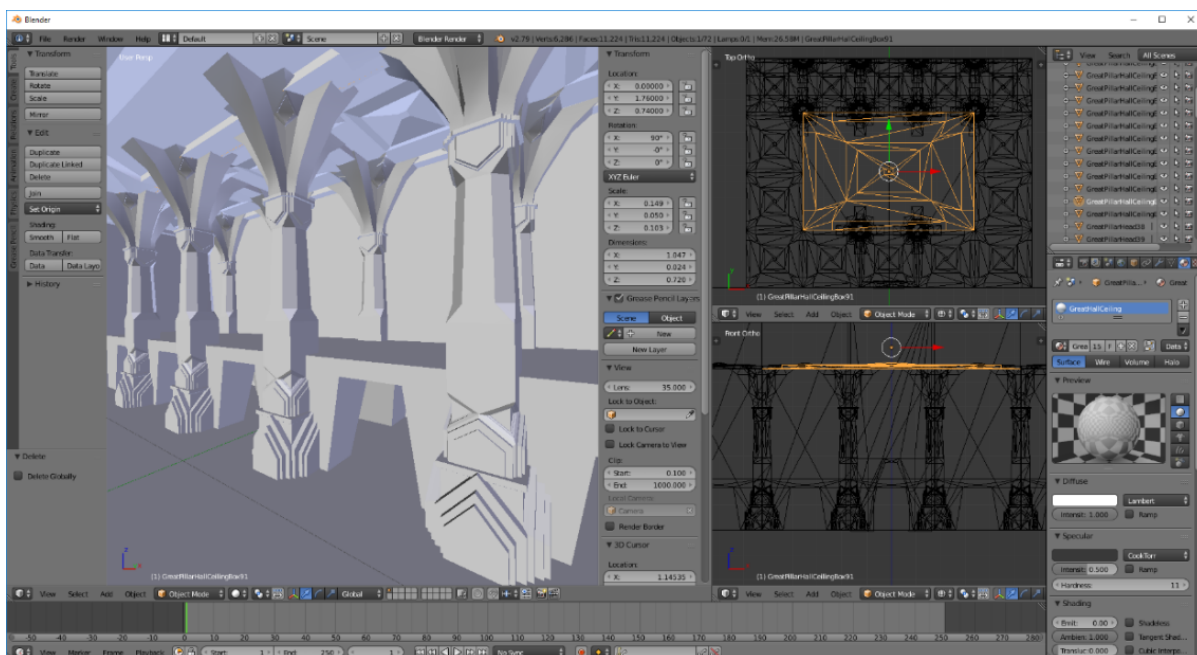


2.10. ábra. A Graphisoft Archicad rendszerével készült kép
Forrás: <https://www.graphisoft.hu/archicad/>

3. VIRTUÁLIS VILÁG

A háromdimenziós képképzés során egy virtuális világot jelenítünk meg, amely a valós világunkat utánozza. Ez azt jelenti, hogy akár csak a körülöttünk lévő világ, a virtuális világ is tárgyakkal épül fel, amelyeknek formája, színe, egyedi jellemzői vannak. A virtuális objektumok formájának megadását hívjuk modellezésnek. A modellezés végeredménye egy olyan reprezentáció, amelyet hatékonyan és egyértelműen tudunk tárolni a számítógépen. Míg a mi fejünkben egy tárgy formájára gondolva olyan dolgok jutnak eszünkbe, hogy gömbölyded vagy szögletes, hosszúka vagy tömzsi, egyik oldalán csúcsosodó stb., és hasonló jellemzőket halva mi magunk is el tudunk képzelni egy tárgyat, addig ezek a kifejezések a számítógépes grafikában nem tudnak egyértelműen leírni egy alakzatot. Az alakzatainkat számsorozatként kell tárolnunk, és segítségül kell hívnunk geometriai formulákat.

A tárgyak felületének leírására több módszer is rendelkezésünkre áll, amelyek közül nem lehet egyértelmű győztest hirdetni, ugyanis mindegyik módszer az alakzatok egy-egy csoportját tudja igazán hatékonyan kezelni [Szirmay-Kalos 2004]. Ezeket a modellezési módszereket és a hozzájuk tartozó kiterjedt eszközkészleteket összetett szoftvereken keresztül tudjuk használni, amelyeket háromdimenziós modellezőszoftvereknek nevezünk. Ezek felülete meglehetősen bonyolult, és használatuk nagy gyakorlatot igényel. A 3.1. ábrán a Blender nevű nyílt forráskódú, háromdimenziós modellező- és animációs szoftver felületét láthatjuk. Ezek a szoftverek nemcsak a forma megadására kínálnak eszközöket, hanem a felületi jellemzők (mint szín, érdesség) és mozgások is megalkothatók velük.



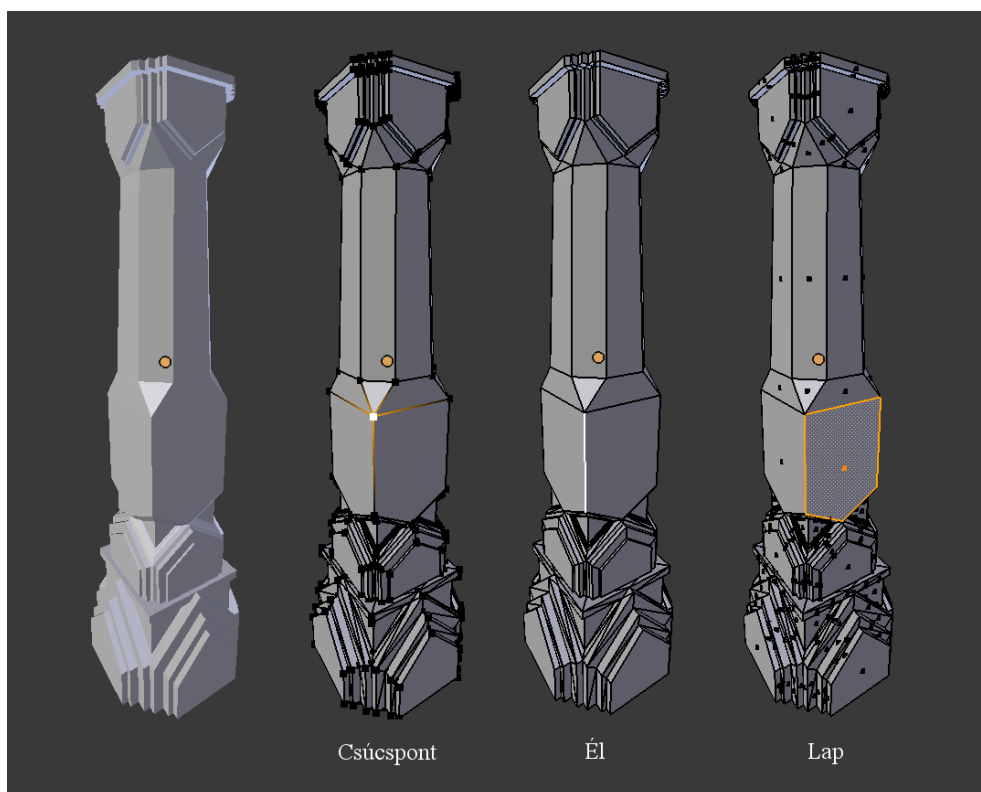
3.1. ábra. A Blender felülete
Forrás: a szerző saját felvétele

A következő alfejezetekben áttekintjük a fő geometriai reprezentációkat és modellezési módszereket.

3.1 Geometriai modellek

3.1.1 Sokszög-geometriák

A valós idejű háromdimenziós megjelenítés során leggyakrabban használt felütleírási forma a poliéder, amelynél a geometriát sokszöglapok alkotják. A sokszög-geometriának három fő alkotóelemük van: pontok, élek és lapok – ezeket a 3.2. ábrán mutatjuk be. A legelemibb egység a csúcspont, amely egy háromdimenziós koordinátával rendelkezik. Két csúcspontot él köt össze. Szomszédos élek kört alkotó sorozata határoz meg egy lapot. Általános esetben a lapok akárhány élet tartalmazhatnak, de a valós idejű megjelenítés során háromszögekkel dolgozunk. Ennek az az oka, hogy csak a háromszög határoz meg egyértelműen egy síkot. Szerencsére bármely sokszög átalakítható háromszögek sorozatává, erre a modellező szoftverek kész eszköztárat biztosítanak. A sokszögekkel történő modellezést poligonmodellezésnek nevezzük.

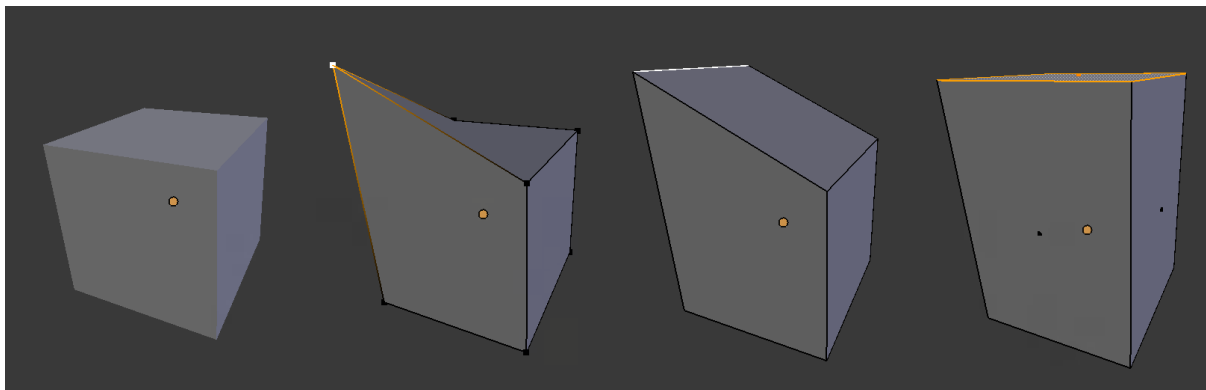


3.2. ábra. Sokszög-geometriák al kotóelemei

Forrás: a szerző saját készítésű ábrája

A poligonmodellezés során egy egyszerű sokszög-geometriából indulunk ki, és elemi műveletek sorozatával alakítjuk, bővítjük a modellt. A kiindulási alap lehet egy téglalap, egyetlen háromszög vagy téglalap. Az elemi műveletek közé tartozik például az alkotóelemek mozgatása, forgatása, átméretezése. A 3.3. ábra azt mutatja be, hogy milyen hatással van a geometriára egy csúcspont, egy él vagy egy lap felfele irányú eltolása. Ennél a műveletnél az alkotóelemek száma nem változik, és szomszédossági viszonyaik

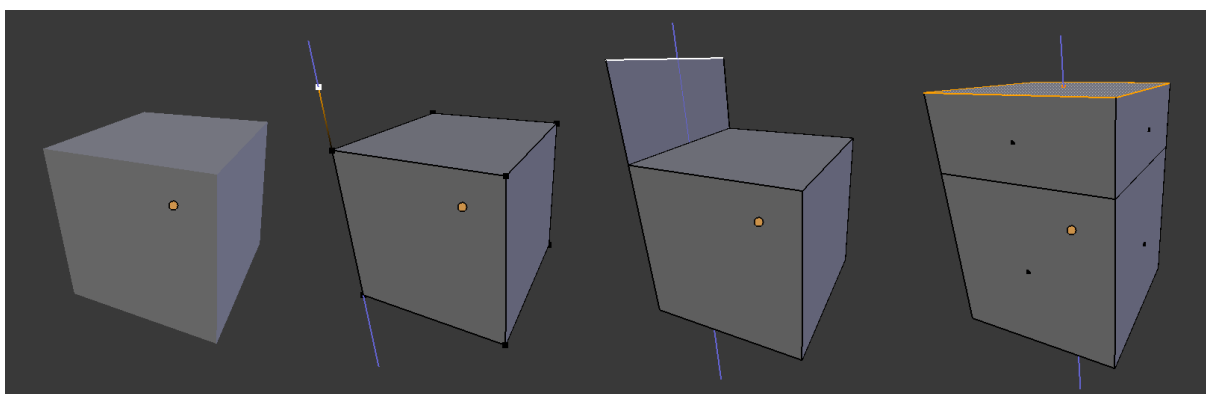
is megmaradnak: ugyanazok a csúcspontok alkotnak éleket, ugyanazok az élek alkotnak sokszöglapokat. Ilyenkor azt mondjuk, hogy a geometria topológiája változatlan marad.



3.3. ábra. Poligon alkotóelemek mozgatásának hatása a geometriára

Forrás: a szerző saját készítésű ábrája

Vannak olyan műveletek, amelyek során a topológia megváltozik. Egy gyakran használt ilyen művelet a kihúzás, amelyet a 3.4. ábra szemléltet egy kocka kiindulási geometrián. A kihúzás bármely alkotóelemre elvégezhető. Egy csúcspont kihúzása azt jelenti, hogy a csúcspontból készül egy másolat, annak pozícióját megváltoztatjuk, és az eredeti, illetve a kihúzott csúcspontot éllel kötjük össze (az eredményt az ábra második alakzatán láthatjuk). Az élkihúzás hasonlóan történik. Az élet lemásoljuk és eltoljuk, illetve a megfelelő csúcspontokat éllel kötünk össze. Ezzel három új él is keletkezik, amelyek az eredeti éllel egy körbe érő zárt élsorozatot alkotnak. Erre az élsorozatra új lapot illesztünk (harmadik alakzat). Egy lap kihúzása pedig úgy történik, hogy a lapot lemásoljuk és eltoljuk, a megfelelő csúcspontokat éllel összekötjük. Ezzel a kihúzott lap és az eredeti lap között egy palást jön létre, melynek oldalait lapokkal zárjuk le (negyedik alakzat). Az eredeti lap, amely most már a test belsejében van, törlődik.



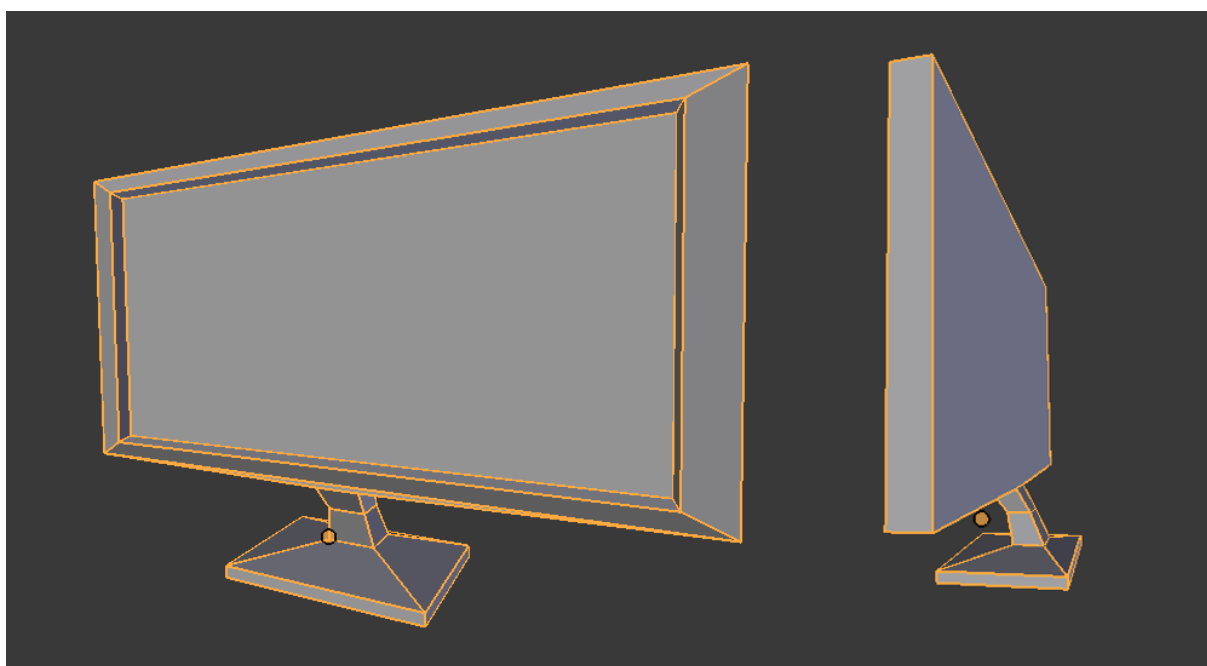
3.4. ábra. Poligon alkotóelemek kihúzásának hatása a geometriára

Forrás: a szerző saját készítésű ábrája

A kihúzás műveletnél meg kell említenünk, hogy a csúcspont- és az élkihúzás megszünteti az eredeti kocka egy fontos pozitív tulajdonságát: a zártságát. A kocka síklapokkal teljesen körülhatárolt test: poliéder. Ezt úgy is megfogalmazhatjuk, hogy bármely pontot is vesszük a háromdimenziós térben, egyértelműen meg tudjuk mondani, hogy az a testen belül vagy kívül van-e. Az él kihúzásának hatására keletkezett egy olyan lap, mely ezt

az egyértelmű döntést most már megakadályozza. Ez a lap nem csatlakozik minden éle mentén más lapokhoz, így az objektum zártsága megszűnt, a kint-bent viszonyok értelmezhetetlenné váltak. A csúcspont kihúzása pedig olyan élt hozott létre, mely csatlakozik laphoz, gyakorlatilag nulla kiterjedésű geometriai elem. Az ilyen geometriákat elfajuló geometriáknak nevezzük, és lehetőség szerint kerülni kell, mert később, a geometria feldolgozása során sok problémát okozhatnak.

Már ezzel a két modellezési eszközzel – az alkotóelemek mozgásával és a lapok kihúzásával – igen változatos geometriákat lehet létrehozni. Példaképpen egy monitor-modellt láthatunk a 3.5. ábrán, amely az alapműveletek egymás utáni végrehajtásával készült. A modellező szoftverek természetesen számos további és bonyolultabb poli-gonmodellezési lehetőséget biztosítanak komplex geometriájú objektumok hatékony létrehozásához.

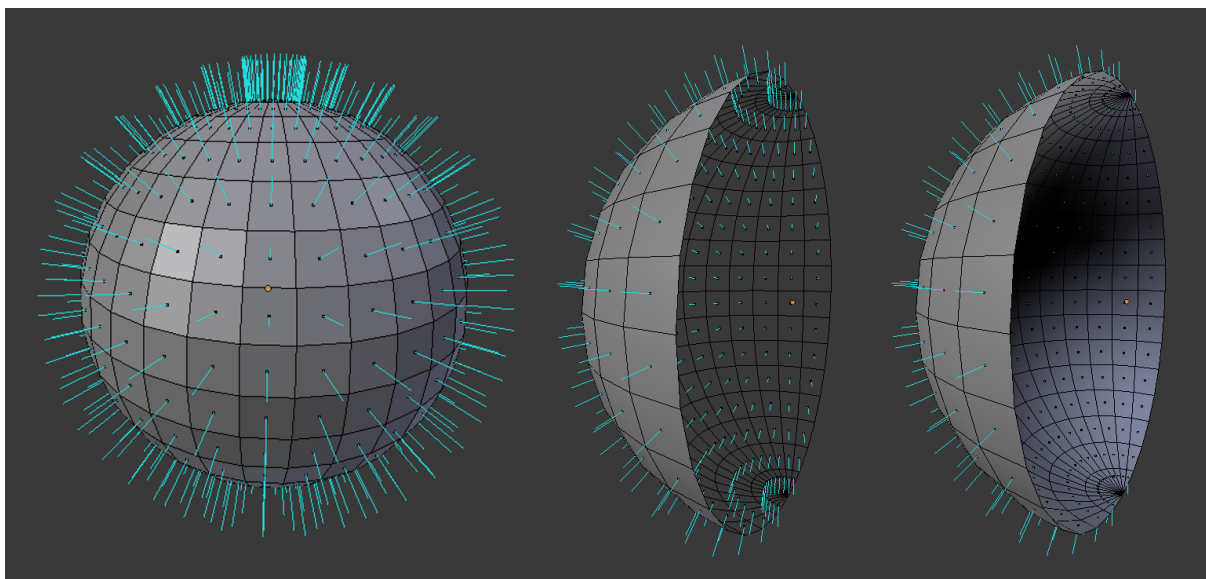


3.5. ábra Lapkihúzással modellezett monitorgeometria

Forrás: a szerző saját készítésű ábrája

A sokszög-geometriákkal kapcsolatban érdemes még megemlítenünk egy-két fontos geometriai jellemzőt. Ezek a testek sokszögekből állnak. Az a jó, ha egy sokszög minden csúcsa ugyanazon a síkon helyezkedik el, így tekinthetjük csak lapnak. Ha mégsem így lenne, fel kell osztani háromszögek sorozatára, amelyek már biztosan egy síkot határoznak meg. Minden síkhoz meghatározhatunk egy irányt, mely a síkra merőleges, ezt hívjuk a sík normálvektorának. Úgy is tekinthetünk rá, mint az az irány, amerre a lap „néz”. Egy síklap esetén valójában két ilyen merőleges irány is van, amelyek egymással ellentétes irányúak, így ezek közül kell választanunk egyet. Ez fogja megmondani, merre van a síklap előtti rész, az ellentétes irányba pedig a síklap mögötti térrészt találjuk. Ezek a lap-normálvektorok fontos tulajdonságok a háromszög geometriáknál. Megfigyelhetjük, hogy egy zárt sokszög-geometria esetén bármilyen nézőpontból is nézünk a felületre, csak felénk néző lapokat láthatunk, a háttal néző lapokat mindig eltakarják a felénk néző lapok. Ezt láthatjuk a 3.6. ábrán, ahol a lapok normálvektorait a lapok közepéből kinövő kis kék pálcikákkal jelöltük.

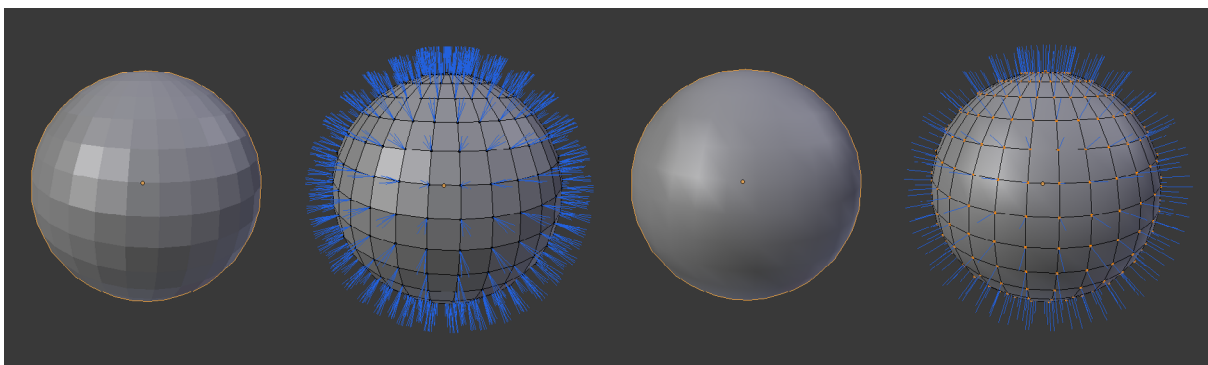
Ez azt jelenti, hogy zárt alakzatoknál elegendő csak a felénk néző lapokat kirajzolni, a többi nem érdemes, mert úgyis el lennének takarva. Ezt hívjuk hátsólapeldobásnak. Előfordulhat, hogy egy geometriát nem tudunk egyszerűen leírni zárt alakzattal, ilyenkor a hátsólapeldobás hatására a geometria bizonyos részei bizonyos kameranézőpontokból láthatatlanná válnak. Ezt mutatja a 3.6. ábra középső, félgömbhöz formájú alakzata is. Ilyenkor a hátsó lapokat is ki kell rajzolni a helyes megjelenítéshez, ahogy ezt a jobb oldali félgömb mutatja.



3.6. ábra. A lapok normálvektorai, félgömb megjelenítése hátsólapeldobással és anélkül
 Forrás: a szerző saját készítésű ábrája

Normálvektorokat nemcsak a lapokhoz, hanem a csúcspontokhoz is rendelhetünk. Ennek az objektum felületi elemeinek fényességértékének számításánál van nagy jelentősége, ahol fontos szerepe van annak, hogy egy felületelem milyen mértékben fordul egy fényforrás irányába. Valójában egy csúcsponthoz nemcsak egy normálvektort tárolhatunk el, hanem annyit, ahány lappal az adott csúcspont határos. Ha egy lap kirajzolásakor mindig a lap normálvektorát használjuk a csúcspontjain az árnyaláshoz, akkor „lapos” árnyalásról beszélünk (flat shading). Ilyenkor egy csúcspont minden hozzá tartozó lap kirajzolásakor más-más normálvektorértékeket használ. Ezt az árnyalást láthatjuk a 3.7. ábra bal oldali két gömbjén. A csúcspontok normálvektorait kék pálcikákkal jelöltük.

Ha a csúcspont minden lap kirajzolásakor ugyanazt a normálvektort használja, mégpedig a környező lapok normálvektorainak valamilyen átlagát, akkor lágy árnyalásról beszélünk (smooth shading). Ezt látjuk az ábra jobb oldali két gömbjén. Ezeknek a csúcsponti normálvektoroknak a kiszámításához a geometriát és a szomszédossági viszonyokat fel kell dolgozni, hogy az átlag normál irányt meg tudjuk határozni. A modellezés részét képezi a normálvektorok helyes beállítása, segítségükkel igény szerint mind éles, mind lekerekített éleket meg tudunk jeleníteni a modelljeinken.

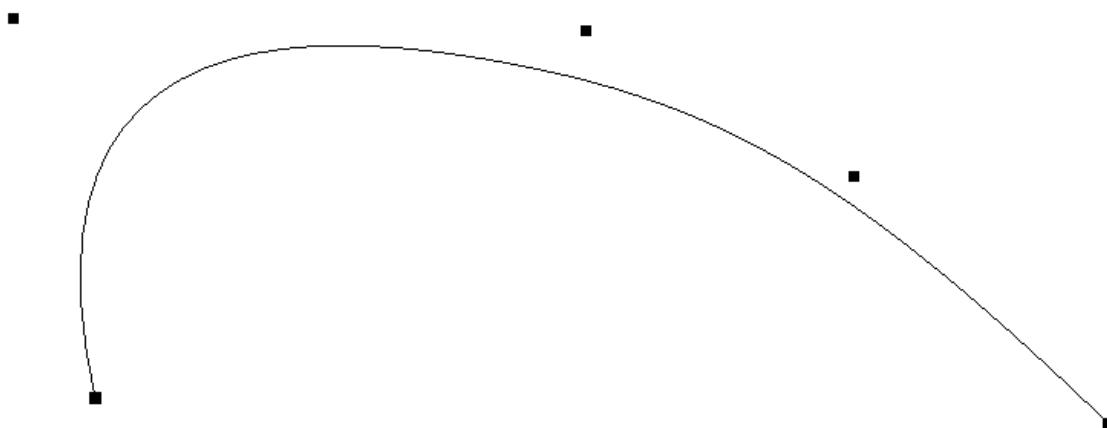


3.7. ábra. Csúcspontok normálvektori flat és smooth árnyalás esetén
 Forrás: a szerző saját készítésű ábrája

3.1.2 Görbealapú reprezentációk

Mint ahogy azt láttuk, a poligon modellek síklapokkal határolt testek, éppen ezért alapvetően éles éllel rendelkező testek modellezésére a leghatékonyabbak. Ha ívelt felületet szeretnénk modellezni, növelnünk kell a csúcspontjaink és sokszögeink számát, és sok apró síklapocskával kell közelítenünk a görbült részeket. Ez meglehetősen időigényes is lehet. Ívelt felületek leírására azonban már jóval a számítógépes grafikai elterjedése előtt voltak módszerek. Ezek olyan mérnöki alkalmazásokban voltak használatosak, mint a gépjárművek karosszériáinak tervezése és gyártása. Ezek a módszerek a felületeket görbék segítségével adják meg.

A görbe egy a háromdimenziós térben futó ívelt vonal. Több görbeprezentáció létezik, de közös bennük, hogy a görbe alakját kontrollpontok határozzák meg. A kontrollpontok meghatározott sorrendben felvett háromdimenziós koordináták. A görbe gyakorlatilag ezeknek a kontrollpontoknak a listáját követve, egyik kontrollpontról a következő felé halad a térben. Megkülönböztetünk interpolációs és approximációs görbéket. Az interpolációs görbe esetén a görbe ténylegesen áthalad minden kontrollpontján, míg az approximációs görbe csak közelít feléjük.

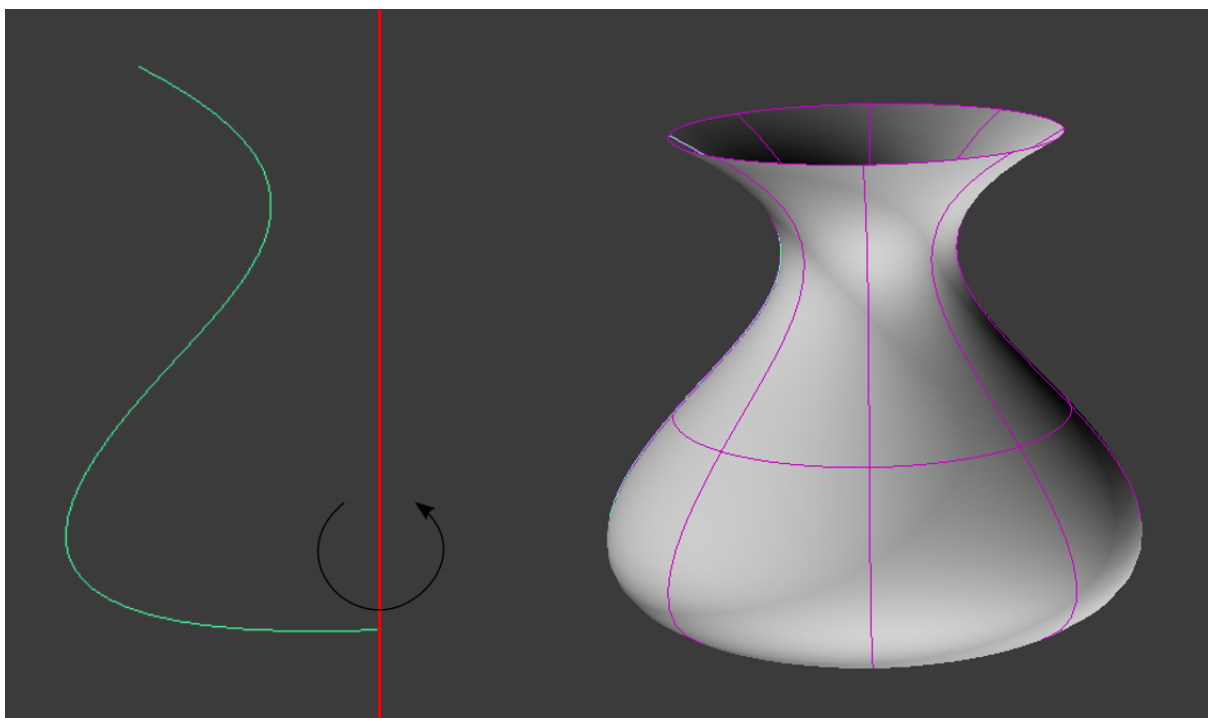


3.8. ábra. NURBS-görbe
 Forrás: a szerző saját készítésű ábrája

A számítógépes grafikai modellezésben gyakran használt görbetípus a NURBS-görbe. Ez alapvetően egy approximációs görbe, de sok modellezőszoftverben interpolációs verzióját is használhatjuk. A 3.8. ábra egy NURBS-görbét mutat be. A csomópontokat fekete téglalapokkal jelöltük. Jól látszik, hogy a görbe a kezdő és végpontokon átmegy, de a köztes csomópontokat csak közelíti, tehát approximációs görbe. A görbe alakját a csomópontok pozíciójának megváltoztatásával módosíthatjuk.

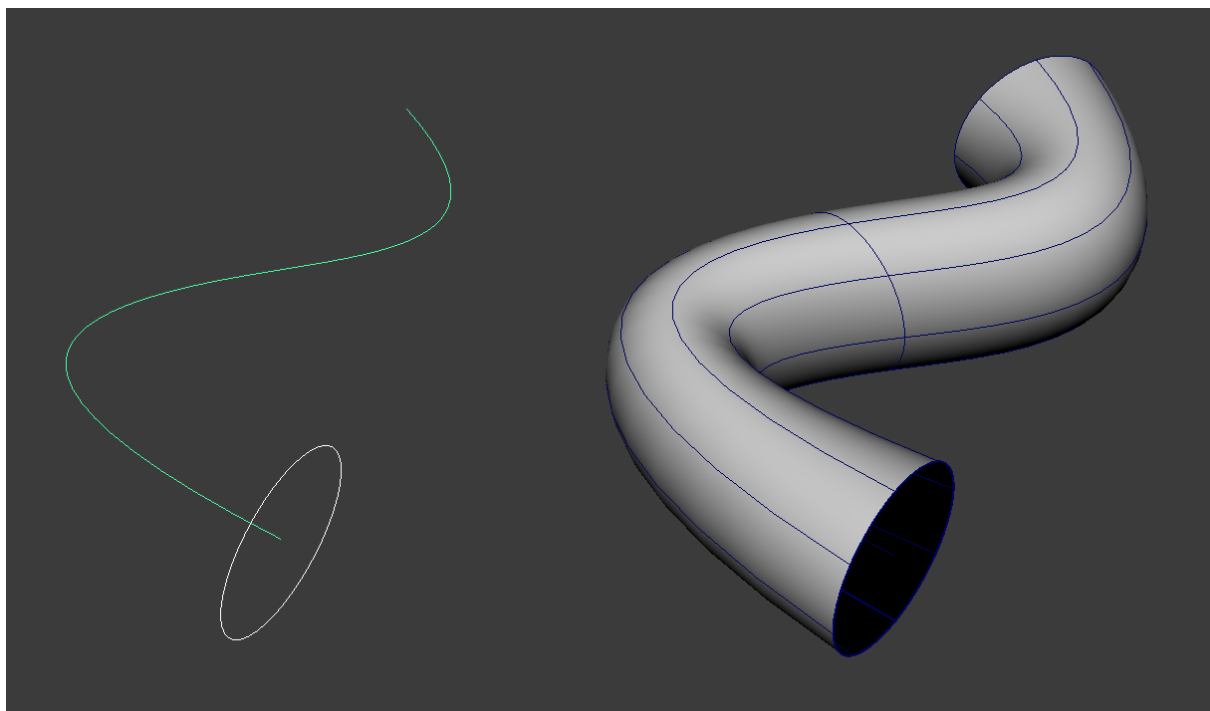
Felületeket görbék segítségével tudunk létrehozni. Erre több lehetőségünk is van, melyből bemutatunk most pár gyakran használt megoldást. Forgástesteket tudunk létrehozni egy görbe egy tengely körüli körbe forgatásával. Ez látható a 3.9. ábrán: a zöld görbét a piros tengely körül teljesen körbe forgatjuk, így eredményül egy váza alakú testet kapunk. A kapott geometria tökéletesen sima, mivel a görbékkel alkotott felületek matematikai definícióját adják a felületnek. A 3.10. ábrán egy kihúzott felületet látunk. Itt két görbére van szükségünk: egy profil görbére és egy útvonal görbére. A kapott felületet a profilgörbe által érintett pontok adják, miközben végigszalad az útvonalgörbén. Ezzel változatos profilú és haladású csőszerű geometriákat tudunk létrehozni. A 3.11. ábrán látható összekötött felületnél több profilgörbét adunk meg, melyek megfelelő pontjait kötjük össze felületté. Így olyan csőszerű alakzatokat tudunk készíteni, melyek hossz tengelyük mentén változtatják profiljukat.

A bemutatott módszerekkel készített felületeket NURBS-felületeknek nevezzük. A NURBS-felületek görbék által határolt felületek, amelyeket szintén kontrollpontok vezérelnek. Egy egyszerű felületre látunk példát a 3.12. ábrán, ahol a felület kontrollpontjait és a kontrollpontok által meghatározott hálót (angol nyelvkörnyezetben: Hull) is jelöltük. Az ábra jobb oldalán kiválasztottunk négy kontrollpontot, és ezeket lejjebb vittük. Érdekes megfigyelni a felület változását. Itt is igaz, hogy a felület a kontrollpontok felé próbál meghaladni, de azokat nem éri el: tehát approximációs felület.

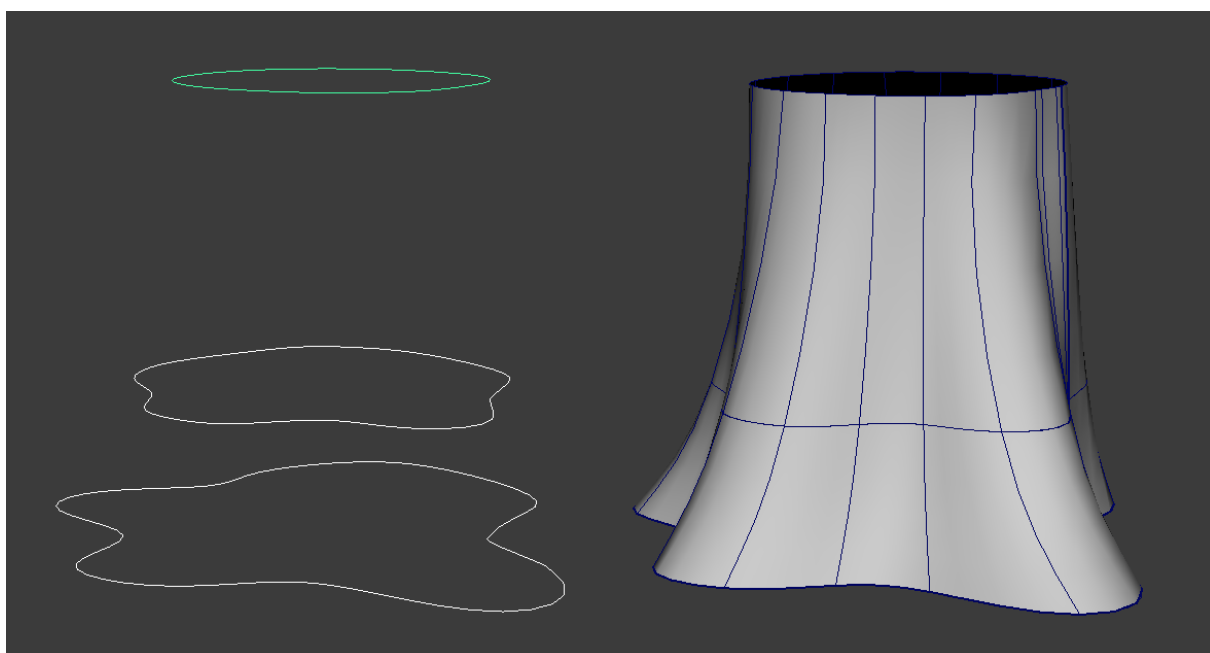


3.9. ábra. Forgásfelület

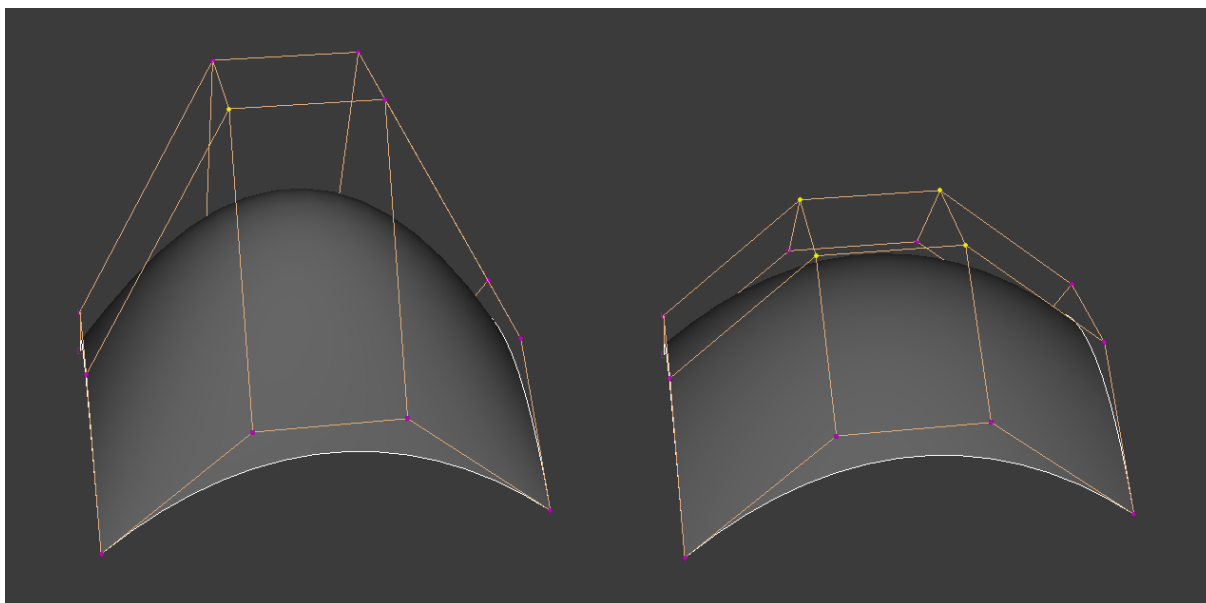
Forrás: a szerző saját készítésű ábrája



3.10. ábra. Kihúzott felület
Forrás: a szerző saját készítésű ábrája



3.11. ábra. Összekötött felület
Forrás: a szerző saját készítésű ábrája



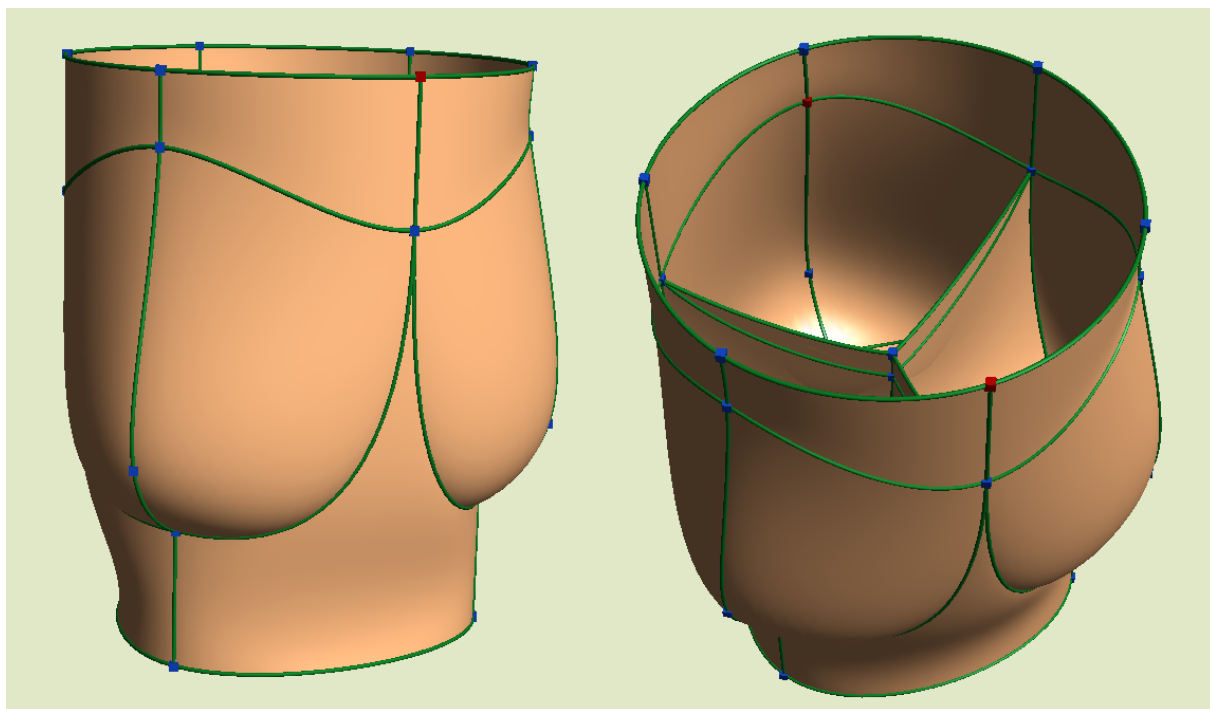
3.12. ábra. NURBS-felület kontrollpontjai és a kontrollpontok mozgásának hatása a felületre
 Forrás: a szerző saját készítésű ábrája

A NURBS-felületek hálója mindig rácsszerkezetű, azaz van négy határoló görbéje, és a belsejét még tetszőleges számban feloszthatjuk, de mindig szabályos rácsot kapunk. Ez hátrányt jelenthet abból a szempontból, hogy nem minden alakzatot könnyű visszavezetni ilyen rácsformára. A forgásfelületeket például nem nehéz: a profilgörbe adja a rács két szemközti oldalélét, annak alsó és felső kontrollpontjai által bejárt körök pedig a másik kettőt. Forgásfelületeknél tehát ez a rács „össze van hajtva”: a két profilgörbeoldal egybeesik, illetve a másik két él zárt, azaz magukba visszatérő görbék. Hasonlóan kell elképzelnünk a kihúzott és az összekötött felületeket is.

NURBS-felületekkel való modellezés során vagy egy rácsot osztunk fel újból és újból az élei mentén ott, ahol újabb részletekre van szükség, vagy egyszerű rácsokat (amelyet a 3.12. ábra is mutat) illesztünk egymás mellé. Ez utóbbi esetén fel kell tudnunk osztani a modellezendő felületet kis rácsok összegére, melyeket szorosan egymás mellé kell illeszteni, és figyelni, hogy szép ívben simuljanak egymáshoz. Ezeket az elemi négyszögletes rácsokat patcheknek hívjuk, és ezt a modellezési módszert patchmodellezésnek.

Nem szabad elfelejtenünk, hogy bár ezeknek a modelleknek a tárolása nagyon hatékony, mert csak csomópontadatokat kell elmentenünk, a megjelenítésüket közvetlenül nem támogatják a mai 3D rendszerek. Ezért minden esetben poligon geometriává kell őket alakítani a megjelenítés előtt, ahol mi állítjuk be a létrejövő poligon geometria részletességét. A patch-ek jó illesztése különösen fontos a poligonhálónak a kialakításakor, ugyanis, ha rosszul érintkeznek, a poligon geometriában lyukak keletkezhetnek.

A négyszögpatchekre bontás elég nagy megkötéseket jelenthet. Ma már léteznek olyan módszerek, melyekkel tetszőleges sokszög alakú felületelemekkel dolgozhatunk. Ebben az esetben szabadon megalkotjuk a modellezett tárgy fontos körvonalait görbékkel, és a görbék által körbezárt térrészeket sima felületekkel szabadon tudjuk kitölteni [Várady 2013]. Erre mutat példát a 3.13. ábra, ahol egy emberi szív aortagyökét és aorta billentyűit modelleztük le szabad felületekkel.

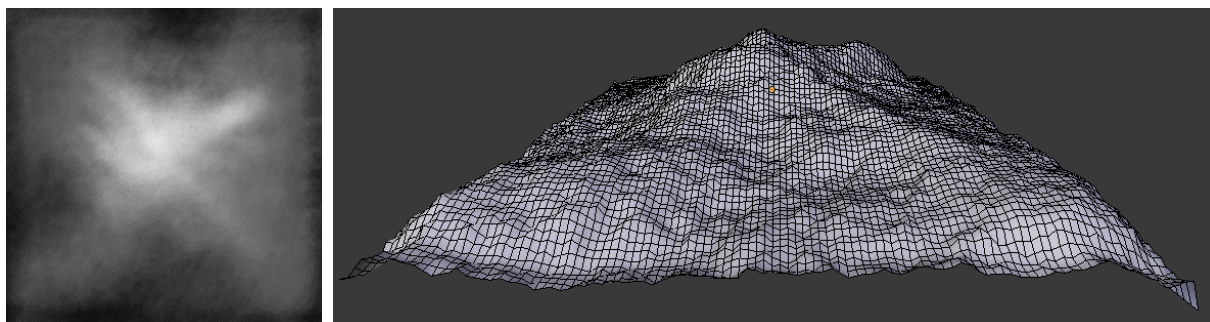


3.13. ábra. Aortagyök- és aortabillentyű-modell sokszögfelületekkel
 Forrás: a szerző saját készítésű ábrája

3.1.3 Magasságmezők

A geometriai modellezés egy speciális formáját alkalmazzák a kültéri jelenetek terepviszonyainak modellezéséhez. A hegyek, völgyek, dombok megalkotásának rendkívül kényelmes módja, ha rajzolunk egy magassági térképet, magasságmezőt. Ez egy olyan szürkeárnyaltos kép, amelynek minden képpontja egy-egy magasságértéket tárol. A fekete képpont nulla magasságot, a fehér teljes magasságot, a szürke árnyalatok pedig köztes magasságokat jelölnek. Hasonlít tehát egy kicsit a térképeink szintvonalaihoz, de itt nem az azonos magasságú területeket jelöljük görbékkel, hanem minden képpontra megadunk egy magasságértéket. Ezeket a magasságtérképeket pedig arra használjuk, hogy egy egyenletes vízszintes rács csúcspontjait kihúzzuk a csúcspontnak megfelelő magasságértékkel. A 3.14. ábrán egy szürkeárnyaltos magasságtérképet és a hozzá tartozó magasságmező-geometriát láthatjuk.

Ezzel a módszerrel könnyen lehet változatos felszíni geometriákat létrehozni. Mivel a kiindulási rácsunk már poligon geometria, a magasságmezőt közvetlenül meg tudjuk jeleníteni.

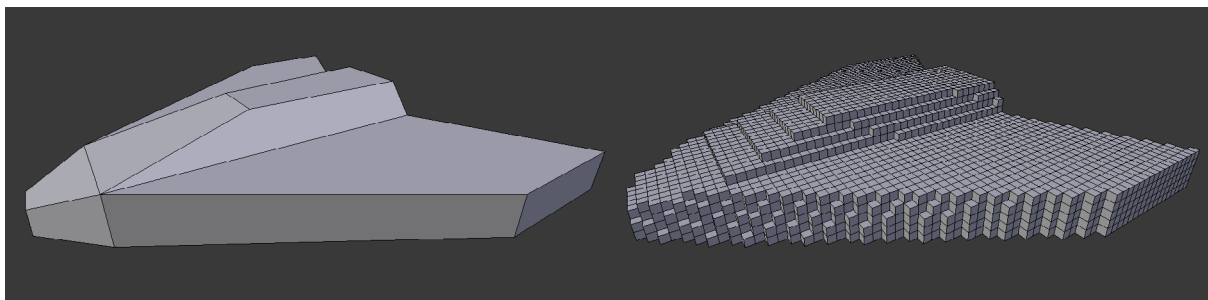


3.14. ábra. Magasságtérkép és a hozzá tartozó geometria
 Forrás: a szerző saját készítésű ábrája

3.1.4 Térfogati modellek

Az eddig tárgyalt modellezési módszerek mindegyike csak az objektumok külső héját írta le. A valós objektumok azonban tömör testek. Ennek a ténynek sokszor nincs jelentősége, mert a külső héj mögé nem szeretnénk nézni, így elegendő annak megjelenítése. Bizonyos esetekben azonban a test belső szerkezete is fontos. Különösen nagy jelentősége van ennek az orvosi alkalmazásoknál, például egy emberi test modelljénél, ahol a belső szervek vizsgálata a fontos.

Ilyen esetekben használhatjuk a térfogati modelleket. A térfogati modellek valójában szabályos háromdimenziós rácson tárolt adatok, amelyeket mezőknek (field) is hívunk. A rács pontjait voxeleknek hívjuk, ami a térfogati elem kifejezésből származik: volume element. Sokféle adatot tárolhatunk a rács pontjaiban: lehet sűrűség (ekkor például sűrűségmezőről beszélünk), lehet szín vagy bármi más jellemző például sebességvektor (sebességmező). Egy orvosi képalkotó eszköz, mint a CT, például sűrűség adatot szolgáltat: a rács minden egyes pontjában azt tároljuk, hogy milyen sűrűségű szövetet mért a berendezés abban a pontban. A sűrűségek pontos értékéből, illetve a hasonló sűrűségű rácspontok által alkotott formákból következtethetünk a szövetek alakjára.



3.15. ábra. Térfogati modellé alakított poligon test

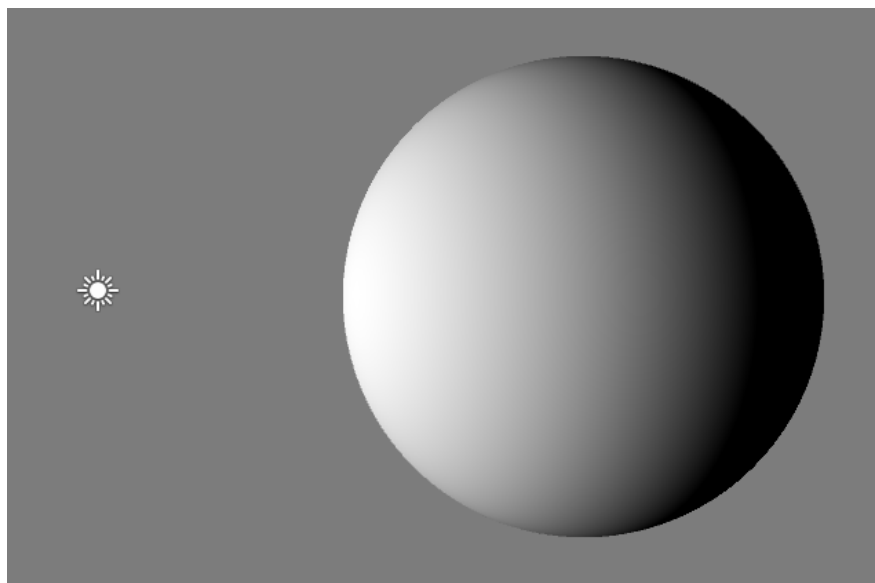
Forrás: a szerző saját készítésű ábrája

A térfogati modelleket főleg orvosi, illetve mérnöki alkalmazásoknál használjuk, segítségünkkel sok fizikai szimulációs probléma hatékonyan megoldható. Ugyanakkor a voxelalapú modellezés egyre népszerűbb. Ennek az lehet az oka, hogy a munkamenet hasonlít a gyerekkori építőkockáinkkal való alkotásra, így nem szükséges bonyolult modellezési technikák elsajátítása. A 3.15. ábrán egy egyszerű úrhajó poligon- és voxelalapú modelljét láthatjuk. Itt a voxelekben csak annyit tárolunk, hogy a testen belülré esik-e az adott voxel, vagy sem, de tárolhatunk emellett színi információt is. A térfogati elemek megjelenítéséről bővebben írunk az 5.4. fejezetben.

3.2 A virtuális objektumok felületi jellemzői

A virtuális tárgyak alakján kívül meg kell adnunk a felületi jellemzőiket is. Ilyen jellemzők például a színe vagy az érdessége. A valós világunk felületei komplex tulajdonságot mutatnak, igazán pontos leírásukra minden pontban minden irányból meg kellene vizsgálnunk, hogy hogyan reagál a reá eső fényre. Ez természetesen lehetetlen, ezért egyszerűbb matematikai modelleket használunk a felületek fényvel való kölcsönhatásainak jellemzésére. Ennek megfelelően különböző anyagmodelleket fejlesztettek ki, amelyek

egy adott felülettípust tudnak jól leírni. Ezek közül a két leggyakrabban használt modell a Lambert és a Phong.



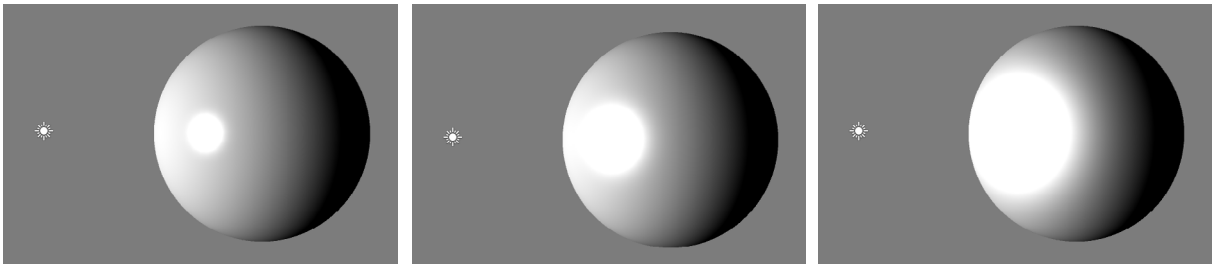
3.16. ábra. Diffúz felület

Forrás: a szerző saját készítésű ábrája

A Lambert-modell diffúz anyagok leírására alkalmas. Diffúz anyagnak tekinthetők például a meszelt falak. Ezekre az anyagokra az a jellemző, hogy a beérkező fényt minden irányban egyenletesen visszaverik, ezért mindegy, milyen szögben tekintünk rájuk, ugyanolyan fényesnek látszanak. A visszavert fényerő csak a világító test fényerejétől, illetve a fénysugarak beesési szögétől függ: minél laposabb szögben érkezik be a fénysugarak, annál sötétebbnek hat a felület. A 3.16. ábra ilyen diffúz felületet mutat be, amelyet bal oldalról világítunk meg (a fényforrást egy kis ikon jelzi a képen).

A valós anyagok jelentős része nem a diffúz, hanem a csillogó anyagok körébe tartozik. Ilyen a legtöbb műanyag, de az emberi bőr is. Ezekre az anyagokra jellemző, hogy bár minden irányban visszaverik a beérkező fénysugarakat, de bizonyos irányokban nagyobb mértékben. Ezek az irányok a tökéletes visszaverődés irányának közelében található irányok. Az, hogy milyen mértékben térnek el ettől a tökéletes visszaverődési iránytól, a csillogó anyagok fontos jellemzője, amelyre a legtöbb szoftverben shininess néven hivatkoznak. Csillogó anyagokat a Phong- vagy a Blinn-moddellel tudjuk megjeleníteni, amelyek kicsit térnek csak el egymástól. Vannak más, komplikáltabb modellek is, de ez a két leggyakrabban használt, főleg a valós idejű grafikában.

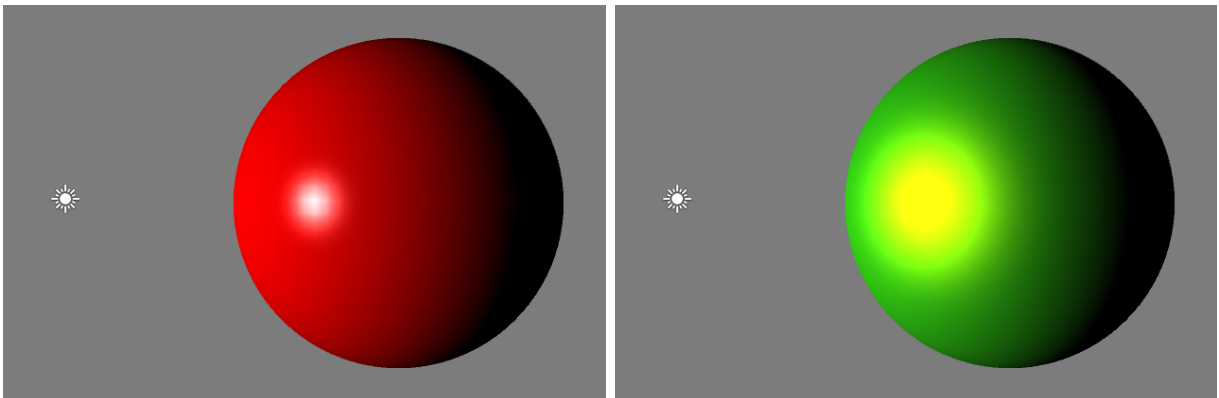
A 3.17. ábra csillogó anyagú gömböket mutat be különböző shininess-paraméterekkel (a paraméter értéke balról jobbra csökken). Fontos megjegyezni, hogy ezeken az ábrákon két anyagmodell is számítunk egyszerre: a Lambert- és a Phong-modell hatását összegezve jelenítjük meg. Tehát a csillogó – más szóval spekuláris – modellek csak a csillanás fénykörét számítják, ehhez még általában hozzáadjuk a diffúz visszaverődéseket is.



3.17. ábra. Csillogó anyagok különböző csillogási paraméterrel
 Forrás: a szerző saját készítésű ábrája

A diffúz és a spekuláris modellekkel már az anyagok széles skáláját elő tudjuk állítani. Az egyik leggyakrabban változtatott paraméter az az anyag színe. Ezen kívül a spekuláris csillanás színét is meg tudjuk változtatni. Nagyon gyakori beállítás, hogy az anyag színét megváltoztatjuk, de a csillanást fehér színűn hagyjuk. Ez azért indokolt, mert sok anyag esetében a tükörszerű visszaverődésekre nem hat az anyag színe, ugyanis ezek a fénysugarak nem hatolnak be az anyagba, hanem csak valamilyen irányban visszaverődnek. A diffúz szín viszont az anyagba hatolt és a felület alól visszaszóródott fénysugarak színe, ahol már igenis számít a felület fényelnyelő tulajdonsága. Ugyanakkor a spekuláris szín megváltoztatásával érdekes anyagokat lehet létrehozni. Mindkettő esetre mutat példát a 3.18. ábra.

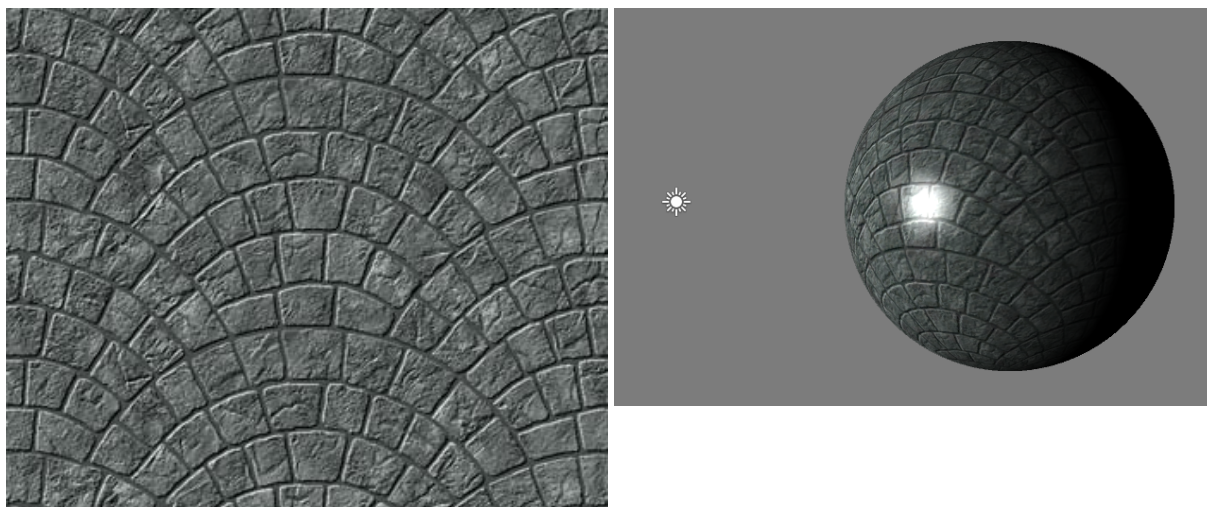
A természetben található tárgyakra ritkán jellemző az, hogy minden felületi pontjuk ugyanolyan színű, sőt rendkívül változatos színmintázatokkal találkozhatunk. Kell hát valamilyen lehetőség, amivel a felület egyes pontjaihoz színértékeket rendelünk. Egy megoldás lehet az, hogy a csúcspontokhoz rendeljük a színértékeket. Ez sajnos részletes mintázatnál túl sok csúcspontot igényelne. E helyett kétdimenziós képekben, ún. textúrákban tároljuk a felület színmintázatát, és ezeket a képeket mint egy csomagolópapírt húzzuk rá a felületre.



3.18. ábra. Csillogó anyagok különböző diffúz és spekuláris színekkel
 Forrás: a szerző saját készítésű ábrája

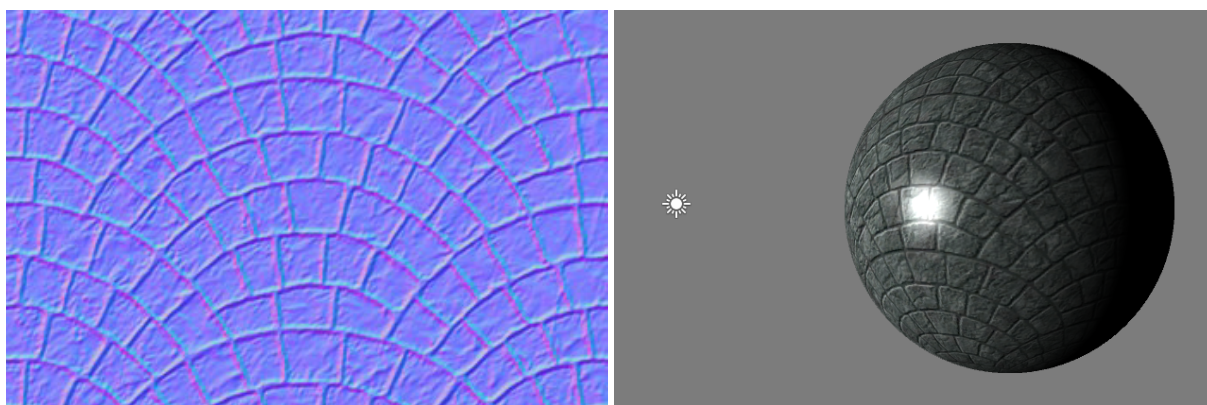
Azt, hogy pontosan hogyan kell egy textúrát a felületre ráhúzni a felület textúra-paraméterezése adja, amelynek előállítása a modellezési lépés után szokott következni. Ez gyakran időigényes művelet, eszközkészletére most nem térünk ki. Ha azonban ez a paraméterezés már adott, egy egyszerű képfájlban tárolhatjuk a színértékeket, és ezt a

felülethez rendelve sokkal részletesebb anyagokat hozhatunk létre. A 3.19. ábrán egy követmintájú textúrát láthatunk, illetve ennek a textúrának a használatával megjelenített csillogó felületű modellt. Láthatjuk, hogy a részletgazdagságot jelentősen növelni tudtuk.



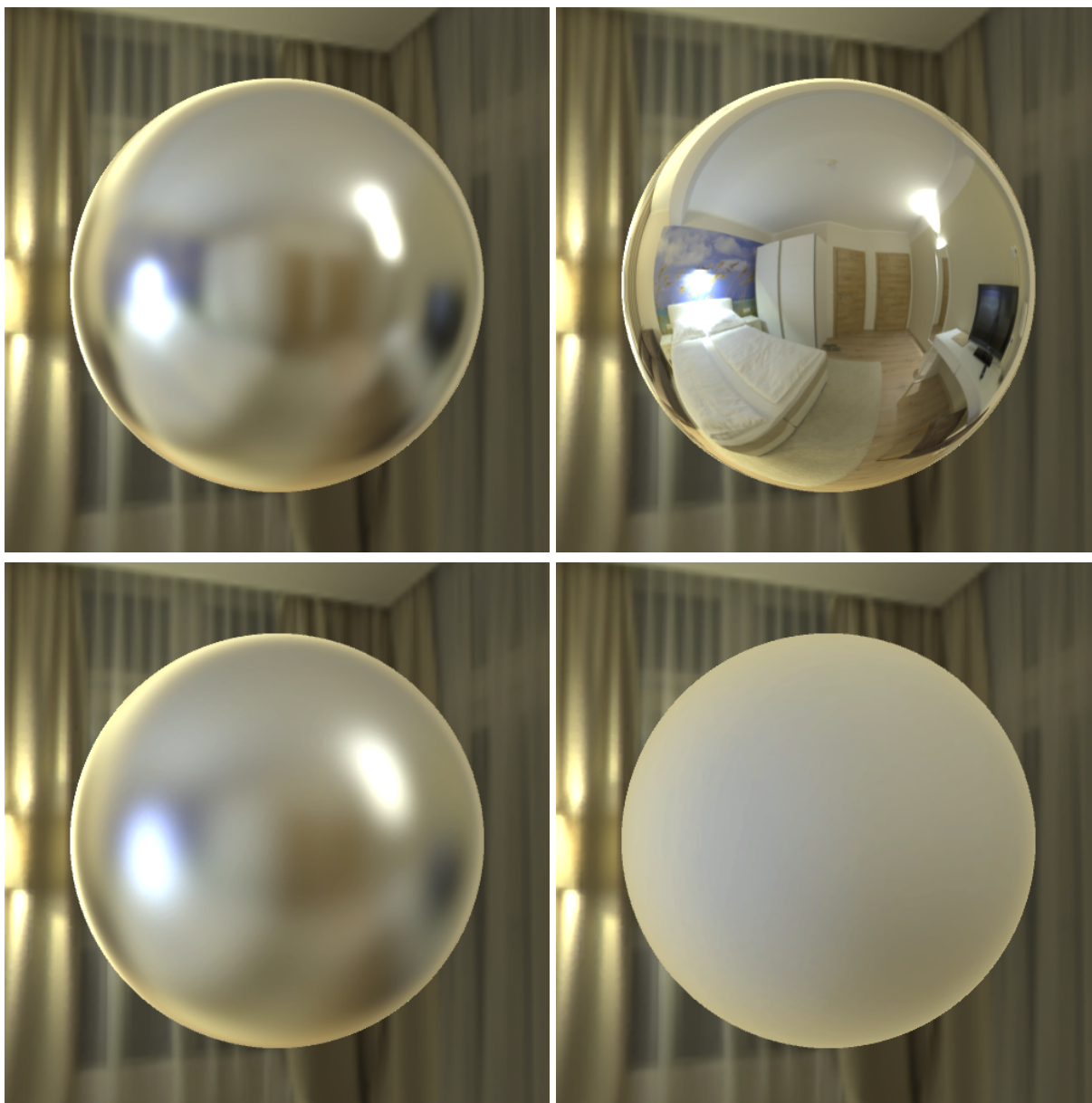
3.19. ábra. Textúrafájl és annak használata a felületen
Forrás: a szerző saját készítésű ábrája

A textúrákban nem csak színi információt tárolhatunk. Eltárolhatjuk például a felület csillogásának mértékét, így pontról pontra változhat a csillogási fénykör nagysága. Ez például az emberi arcbőrnel hasznos, ugyanis azokon a területeken, ahol fesesebb a bőr, ott a csillogási paraméter magasabb értékű. Egy másik gyakran használt textúra-típus a normál textúra. Ezek a textúrák valójában 3D irányértékeket tárolnak a textúra pixeleinek három színcsatornájában. Ezek az irányok pedig a felület irányát adják meg, pontosabban a poligonháló által meghatározott síklapok irányától (normálvektorától) való eltérést. Ezzel a felület részletességét növelhetjük újabb csúcspontok hozzáadása nélkül. Ez a módszer az ún. displacement mapping módszerek közé tartozik [Szirmay-Kalos 2008], csak apróbb felületi változások megjelenítésére alkalmas, így a felület érdekességét, illetve bemélyedéseit, karcolásait tudjuk vele érzékeltetni. A 3.20. ábra egy ilyen normál textúrát és annak alkalmazását mutatja be.



3.20. ábra. Normál textúra és annak használata a felületen
Forrás: a szerző saját készítésű ábrája

Textúrákat használhatunk a környezet visszaverődésének megjelenítésére is. Ilyenkor nincs konkrét fényforrásunk a térben, hanem a környezetből mindenhol érkező fény sugarak. Ezeket a felület visszatükrözheti a tükörrányba, de a felület polírozottságától függően az a tükörkép elmosódhat. Ennek szélsőséges esete a diffúz felületet adja. A környezet színeit tartalmazó képeket környezeti térképeknek (environment map) nevezzük, melyek speciális tárolású képek: 2D koordináták helyett 3D irányvektorokkal címezhetők. A környezeti térképek használatára mutat példát a 3.21. ábra, ahol a különböző polírozottságú felületeket láthatunk.



3.21. ábra. Környezeti térképek használata
Forrás: a szerző saját készítésű ábrája

3.3 Transzformációk

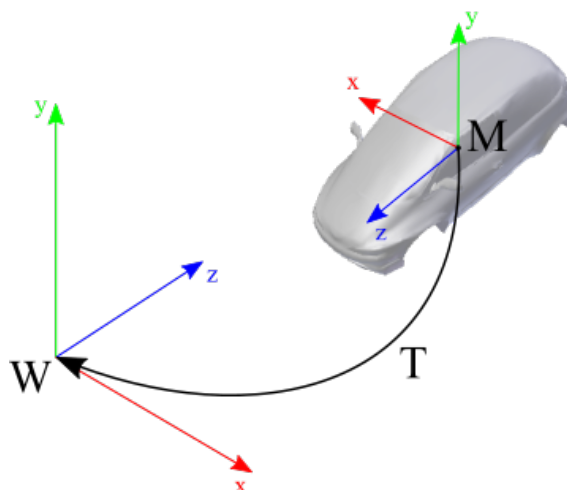
A számítógépes grafikában a geometriai alakzatok megváltoztatását geometriai transzformációnak nevezzük. Ezeknek a transzformációknak egy fontos részét képezik az alaktartó, vagy merev transzformációk, ahol az alakzat formája nem változik, csak a pozíciója, orientációja, esetleg a mérete. A valós világban is leggyakrabban ilyen a transzformációkkal találkozunk: egy autó elhalad előttünk, miközben a kerekei nemcsak haladnak, hanem forognak is; vagy egy nyíló ajtó is valójában a csuklópántok által meghatározott tengely körül végez forgó mozgást.

3.3.1 Koordináta-rendszerek

Pozíciókról, illetve elfordulásokról beszélni csak akkor van értelme, ha van mihez viszonyítanunk őket. Az, hogy valami öt méterre előre van, csak akkor kap értelmet, ha tudjuk, honnan számítjuk az öt métert, és hogy mit is jelent pontosan, hogy előre. A háromdimenziós térben a viszonyítási alapot a koordináta-rendszerek jelentik, amelyeknek több formája lehet, de leggyakrabban derékszögű (Descartes) koordináta-rendszert használunk. A derékszögű koordináta-rendszereknek mindig van egy középpontja (origó), ahol a koordinátatengelyek metszik egymást. A koordinátatengelyek páronként merőlegesek és x , y és z betűkkel szokás jelölni őket. Egy pont pozícióját az origótól mérve, három tengely szerinti elmozdulással, azaz koordinátákkal szoktunk leírni. A forgás leírásának több módja is van, amelyek közül számunkra a fő tengelyek szerinti elfordulások a legkönnyebben értelmezhetők, ezért a legtöbb szoftver ezek beállítását kínálja fel, ugyanakkor a grafikai rendszerek a számításaikhoz nem ezt a leírást szokták alkalmazni bizonyos hátrányos tulajdonságai miatt, amelyek taglalására nem tudunk kitérni.

A virtuális világok leírásánál mindig rögzítünk egy fő koordináta-rendszert, ezt nevezzük világkoordináta-rendszernek, vagy globális koordináta-rendszernek. Ennek középpontja a virtuális világ középpontja, és ennek fő irányai adják meg, hogy a virtuális világban merre is van a „jobbra-balra” (x tengely), „fel-le” (y tengely) és az „előre-hátra” (z tengely). Ugyanakkor a testek geometriai leírását nem ebben a koordinátarendszerben adjuk meg, hiszen egy test elmozdulásakor az összes csúcspontjának koordinátáit el kellene mozgatnunk. A könnyebb kezelhetőség kedvéért a geometriai leírásokat mindig egy saját koordináta-rendszerben adjuk meg, ami a világ koordináta-rendszerétől független. Tehát ha modellezünk egy autót, akkor az autó modellje esetében is tudunk rögzíteni egy középpontot, és x , y , z tengelyeket, amik az autó szemszögéből rögzítenek előre-hátra, jobbra-balra és fel-le irányokat. Az autómódellem csúcspontjainak koordinátáit ebben a koordináta-rendszerben adjuk meg. Ezt a koordinátarendszert nevezzük modellezési, objektumtérbeli vagy lokális koordinátarendszernek.

Egy modell virtuális világba helyezésekor megadjuk az összefüggést a globális koordináta-rendszer és a modell lokális koordináta-rendszere között. Ez az összefüggés a modell transzformációja. A transzformáció tehát megadja azt, hogy az autómódellem középpontját a világ középpontjához képest a világ fő tengelyei mentén hova kell elhelyezni, illetve azt is, hogy az autómódellem lokális fő tengelyei merre álnak a virtuális világban (hogyan forgattuk meg a modellt). Ezt szemlélteti a 3.22. ábra, ahol W -vel jelöltük a világot, és M -mel a modellezési koordináta-rendszereket. A két koordináta-rendszer közötti transzformációt jelöli a T .

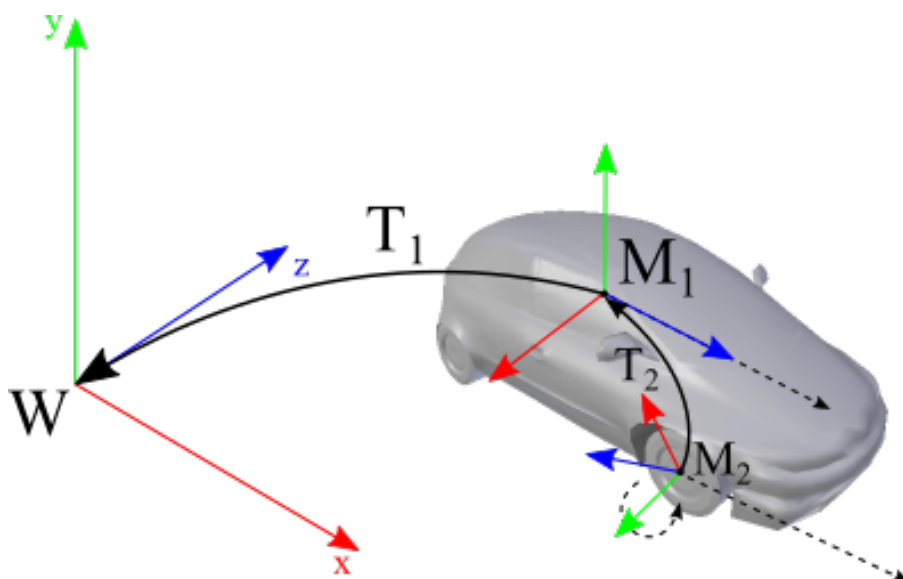


3.22. ábra. Világ- és modell-koordinátarendszerek és viszonyuk
 Forrás: a szerző saját készítésű ábrája

A virtuális világ felépítésekor minden háromdimenziós objektumnak saját lokális koordináta-rendszere és transzformációja van. Ezt a transzformációt világtranszformációnak (world transform) is hívjuk, mert modelltérbeli koordinátákból állít elő világtérbeli koordinátákat.

3.3.2 Transzformációs hierarchiák

A lokális koordináta rendszereket nemcsak a globális koordináta-rendszerekhez viszonyíthatjuk, hanem más lokális koordináta-rendszerekhez is. Tehát egy transzformáció azt is megadhatja, hogy a modell lokális koordináta-rendszeréből hogyan kell áttérni egy másik modell lokális koordináta-rendszerébe. Ennek akkor van jelentősége, ha az egyik modell transzformációja egy másik modellre is hatással van, ekkor transzformációs hierarchiáról beszélünk. Egyszerű példaként tekinthetjük megint az autómodellt, melyet a 3.23. ábrán szemléltetünk. Az autó lokális koordináta rendszerét $M1$ -el jelöltük, a világkoordinátákba vivő transzformációját pedig $T1$ -el. Az autó, előrehaladás során, a lokális koordináta-rendszerének origóját változtatja a világkoordináta-rendszerben (ezt szaggatott nyíllal jelöltük). Ha az autó kerekét külön modellben tároljuk, akkor ennek a modellnek is van saját lokális koordináta-rendszere, melyet $M2$ -vel jelöltük. Ezt a koordináta-rendszert viszont érdemes nem a világhoz, hanem az autó koordináta-rendszeréhez viszonyítani, hiszen a kerék pontosan úgy halad előre, ahogy az autó. Ennek valós alapja is van, hiszen a kerék az autó része, és a kerék felfüggesztési pontja az autóhoz képest mindig fix. Ezt tükrözve a kerék transzformációját az autó rendszerében adjuk meg, tehát $T2$ a kerék lokális koordináta-rendszeréből az autó lokális koordináta-rendszerébe tudja transzformálni a csúcspontjai koordinátáit. A kerék transzformációja ezen kívül forgást is tartalmaz, ami az ő lokális koordináta-rendszerének egy tengelye körüli forgást jelent (ezt szaggatott félköríves nyíllal jelöltük). A kerék lokális koordináta-rendszere tehát világtérben egyenesen halad előre, és közben folyamatosan forog. Az $M2$ -ből W -be vivő transzformáció valójában $T2$ és $T1$ transzformációkból tevődik össze.



3.23. ábra. Transzformációhierarchia
Forrás: a szerző saját készítésű ábrája

A transzformációkat valójában mátrixalakban tároljuk, és az összetett transzformációk az elemi transzformációk mátrixainak szorzatát jelenti, ami szintén egy mátrix. Így az összetett hierarchikus mozgások pontosan ugyanúgy, mátrixformában kezelhetők, mint az egyszerű transzformációk. A transzformációs hierarchiákat érdemes kihasználnunk, mert a testek pozíciójának és elfordulásainak komplex matematikai képletekkel való megadását elkerülhetjük, ha a mozgást fel tudjuk bontani elemi mozgások hierarchiájaként.

4. ANIMÁCIÓ

A körülöttünk lévő világ folyamatos mozgásban, változásban van. Ha azt szeretnénk, hogy a virtuális világunk a valós világot utánozza le, nem elegendő a modellezés és megjelenítés, a mozgást is meg kell valósítanunk. Ezt a mozgást nevezzük animációnak.

Az animáció szó a latin eredetű animo szóból ered, amelynek jelentése: elevenít, élővé tesz. Pontosan ez az, amit tenni szeretnénk: a mozdulatlan, élettelen világba lelket vinni. Mozgás nélkül állóképhez juthatunk csupán, amely csak ideig-óráig tudja lekötni figyelmünket, és távol áll attól, hogy virtuális környezetnek nevezzük.

A számítógépes animáció a hagyományos animációs technikákon alapszik. Hagyományos animációt láthatunk például a rajzfilmekben, gyurmafilmekben. A mozgás ábrázolása már korán megjelenik a művészetekben. Az őskori barlangrajzokon is láthatunk egy mozgás különböző fázisait ábrázoló képeket. A 19. századra már gyakran találkozhatunk a laterna magica (bűvös lámpás) nevű eszközzel, amelyben egy olajlámpás előtt üveglapokra festett képecskéket húztak el. A képecskék előtt elhelyezett lencse segítségével a kép a falra vetült, és a gyakran váltakozó képkockák a mozgás illúzióját keltették. Valójában a mozifilmek is ugyanezen alapulnak.

Egy mozgást akkor látunk folyamatosnak, ha másodpercenként legalább huszonöt mozgásfázist mutatunk. Tehát a bűvös lámpás esetében egy másodpercnyi mozgás megvalósításához huszonöt, de legalább húsz képecskét meg kellett festeni, és ezeket egy másodperc alatt elhúzni a lencse előtt. Kevesebb kép esetén a folyamatos mozgás illúziója elveszik, az egyes állóképek jól megkülönböztethetővé válnak. A filmek esetén szintén tipikusan huszonöt (a gyakorlatban sokszor inkább huszonnégy) vagy harminc képkockát rögzítünk másodpercenként. A másodpercenkénti képkockaszámot FPS-nek (Frames per Second) is szoktuk rövidíteni.

A virtuális valóság alkalmazások esetén ez az FPS általában változó, erősen függ egy képkocka előállításához szükséges időtől. Ebbe beletartozik a megjelenítés ideje, de minden más, a program állapotának frissítésével kapcsolatos számítás ideje is. Ilyen feladatok például a felhasználó által adott parancsok feldolgozása (tipikusan billentyűzet és egérkezelés és az arra adandó válaszok), a virtuális világ objektumainak mozgásának fizikailag pontos számítása (erről bővebben a 4.6 fejezetben írunk), és a gép által irányított szereplők viselkedésének meghatározása. Ha az alkalmazásunk nem túlzottan erőforrás-igényes, akkor az elérhető FPS bőven a minimális 25 fölé esik. Ilyenkor általában érdemes is lekoraltozni egy fix értékre. Ha azonban erőforrás-igényes az alkalmazásunk, ez az FPS erősen ingadozhat, és akár le is csökkenhet a folyamatos mozgáshoz szükséges 25 alá. Ilyenkor szoktuk azt mondani, hogy az alkalmazás „szaggat”: a folyamatos mozgás látszata megszűnik, állóképek szakaszos megjelenítését érzékeljük.

Az animáció megvalósításához tehát a mozgó világ állapotát bizonyos időközönként elő kell tudnunk állítani, és ezt az állapotot megjeleníteni. Az állapot tetszőleges jellemzők rendezett halmazát jelentheti: pozíció-, elforgatás-, méret-, de akár szín- vagy formaparaméterek is az állapot részei lehetnek. Leggyakrabban pozíció- és elforgatásértékeket kezelünk, de egy emberi karakter mozgása során maga a forma is jelentősen változik.

Egy objektum állapotainak változását tehát valamilyen formában tárolnunk kell, amire több lehetőségünk is van. Ezeket a lehetőségeket tárgyalják a következő alfejezetek.

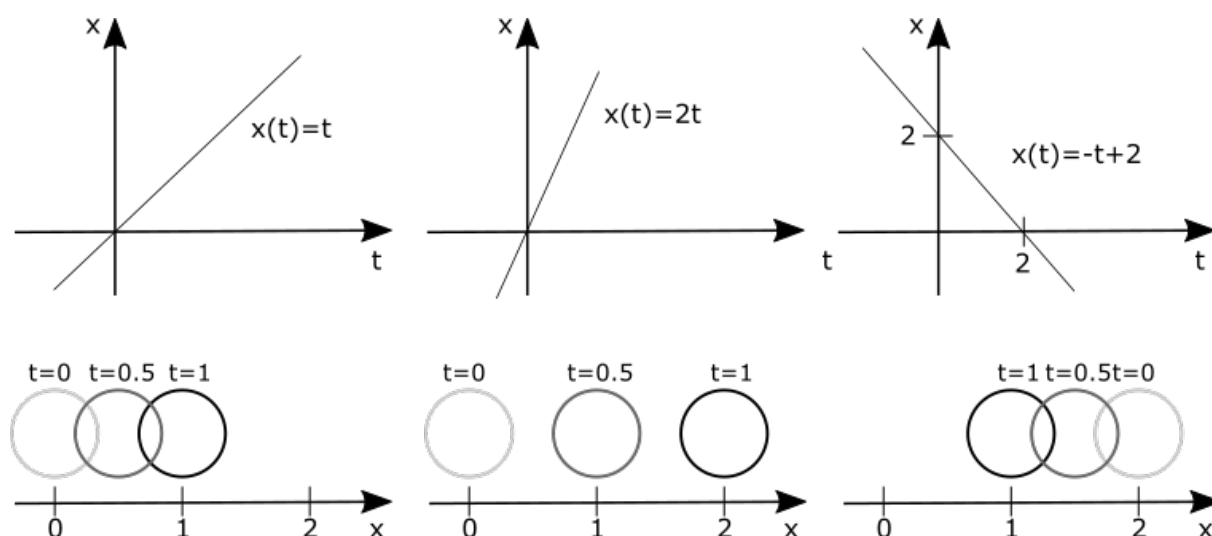
4.1 Képletanimáció

Animáció megvalósításához szükségünk van arra, hogy az objektumok jellemzőinek értékét egy adott időpontban meg tudjuk határozni. Ezen aktuális értékek alapján történik meg ugyanis a megjelenítés. Ha az adott jellemző változása egy matematikai kifejezéssel jól leírható változást mutat, akkor érdemes képletanimációt használni.

Példaképpen tekintsünk egy, a képernyőn vízszintesen, egyenletesen haladó tárgyat. Ha a képernyő vízszintes tengelye az x tengely, ennek a tárgynak a képernyőn látható pozíciójának x koordinátája egyenes arányosságban van az alkalmazás indítása (vagy az animáció indítása) óta eltelt idővel. Az x koordináta tehát megadható egy, az eltelt időn értelmezett lineáris függvénnyel. Képletszerűen ezt így írhatjuk:

$$x(t) = a \cdot t + b,$$

ahol az alkalmazásban eltelt idő, mondja meg az objektum haladási sebességét (és előjelétől függően azt is, hogy balról jobbra, vagy jobbról balra halad a képernyőn), pedig a kezdő időpontbeli kiindulási koordinátáját. Ezt az összefüggést ábrázolhatjuk is egy olyan koordinátarendszeren, amelynek vízszintes tengelyén az időt, függőleges tengelyén az x koordináta értékeit jelöljük. A 4.1. ábrán három lineáris függvényt láthatunk más-más és paraméterértékekkel. A függvények alatt láthatjuk a hozzájuk tartozó animációt is.



4.1. ábra. Lineáris függvények és hozzájuk tartozó animációk

Forrás: a szerző saját készítésű ábrája

Az első esetben és , ami annak felel meg, hogy az objektumunk a időpillanatban az helyen tartózkodik, és egy egységnyi idő elmúltával az helyre mozdul jobbra. Az idő és a távolság konkrét mértékegysége az alkalmazásunktól függ, az animáció jellegét nem befolyásolja, így nem is jeleztük. A második esetben, ahol és szintén az pontból indulunk, de egy egységnyi időlépés után már kétszer akkora távolságra, pontba jutunk. A harmadik esetben, ahol és , az pontból indul az animáció, és a másik irányba, balra

halad a tárgy. Egy egységnyi idő elteltével egy egységnyi távolságot teszünk meg, így a pontba jutunk.

Ezek a példák egyetlen tengely menti elmozdulást, tehát egydimenziós mozgást mutattak meg. Az egyenes vonalú, egyenletes mozgás magasabb dimenziókra való kiterjesztése sem bonyolult, csupán minden dimenzióhoz lesz egy külön lineáris összefüggésünk:

$$x(t) = a_x \cdot t + b_x,$$

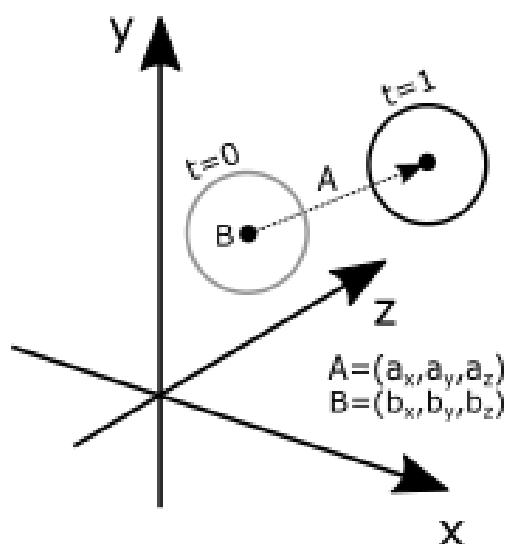
$$y(t) = a_y \cdot t + b_y,$$

$$z(t) = a_z \cdot t + b_z.$$

Ezt felírhatjuk tömörebb formában is:

$$(x(t), y(t), z(t)) = (a_x, a_y, a_z)t + (b_x, b_y, b_z).$$

Ez egy sugár paraméteres egyenlete, melyben a a sugár kezdőpontja, pedig a az iránya. Az animáció során a test a kezdőpontból indul és egységnyi idő alatt az egyenes irányvektora mentén a vektor hosszának megfelelő távolságra jut. Ezt a mozgást ábrázolja a 4.2. ábra.



4.2. ábra. Egyenletes, egyenes vonalú mozgás a háromdimenziós térben

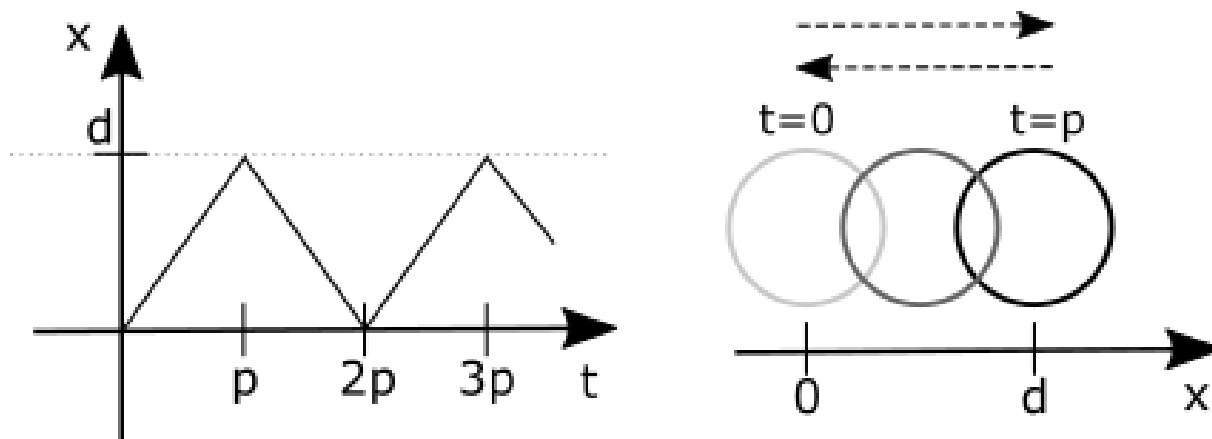
Forrás: a szerző saját készítésű ábrája

A lineáris függvény a legegyszerűbb függvény, amellyel mozgást írhatunk le, ugyanakkor önmagában csak egyenes vonalú, egyenletes mozgás leírására képes. Bonyolultabb mozgásokhoz bonyolultabb függvények vagy függvények kombinációi kellenek.

Nézzünk még egy egyszerű, gyakran felmerülő problémát: egy objektumot szeretnénk egy irány mentén előre-hátra mozgatni. A mozgás egyenes vonalú, és lehet egyenletes is, de menet közben periodikusan irányt változtatunk. Az ilyen mozgást háromszöghullám-függvénnyel írhatunk le, ha maradunk az x tengelyen történő mozgásnál, a következő képletet használhatjuk:

$$x(t) = d \left| 1 - 2 \cdot \text{frac} \left(\frac{t}{p} + \frac{1}{2} \right) \right|,$$

ahol az a távolság, amelyet az objektum megtesz, mielőtt irányt változtat, pedig a mozgás sebességét befolyásolja: időegység alatt fog eljutni az objektum távolságra. Ebben a képletben megjelenik az abszolút érték ($| \cdot |$) és a törtrész (frac) függvény is. A háromszöghullám-függvényt és a hozzá tartozó mozgást mutatja a 4.3. ábra.



4.3. ábra. Háromszöghullám-függvény és a hozzá tartozó mozgás

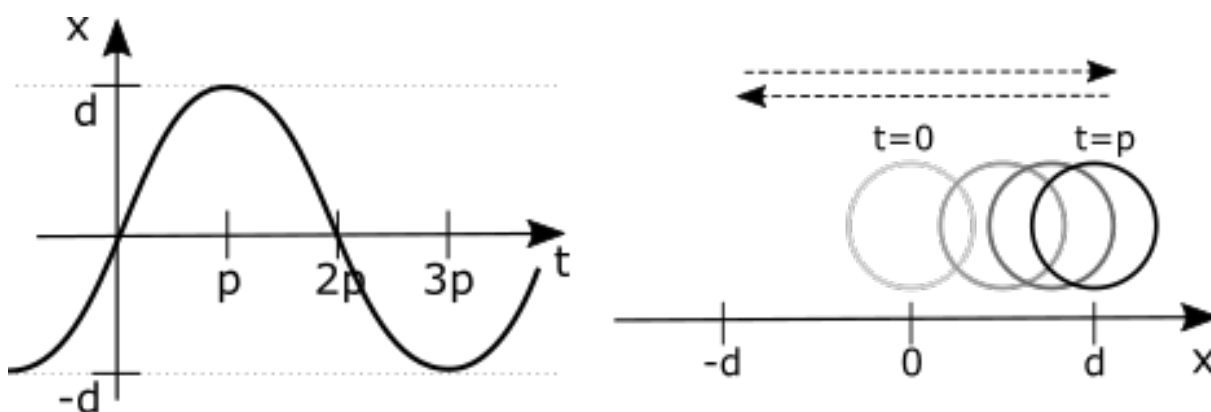
Forrás: a szerző saját készítésű ábrája

Ilyen hirtelen váltású oda-vissza mozgás ritkán figyelhető meg a természetben. A valós mozgásokra általában jellemző, hogy nincs hirtelen változásuk. Ez a matematika nyelvén azt jelenti, hogy folytonosak, és a deriváltjaik is folytonosak, tehát C^1 folytonosak. A folytonosság minden mozgásnál elengedhetetlen, különben hirtelen helyváltatásról, egyfajta teleportálásról beszélnénk. A deriváltak folytonossága csupán simább és természetesebb hatást ad, de nem kötelező, elképzelhető ilyen hirtelen változású mozgás.

Ha szeretnénk simább oda-vissza mozgást elérni, akkor C^1 folytonos függvényt kell keresnünk a probléma megoldásához. Erre tökéletesen megfelel például a szinuszfüggvény:

$$x(t) = d \cdot \sin\left(\frac{\pi}{2p} t\right).$$

Ez a függvény látható az 4.4. ábrán, amelyről leolvashatjuk, hogy egy és között ingadozó értéket ad meg. Ezt egyszerűen átalakíthatjuk, hogy és között ingadozzon, csupán az egyszerűbb képlet miatt maradtunk ennél a tartománynál. A szélsőértéket időegység alatt éri el. A szélsőértéket azonban másképpen közelíti meg, mint a háromszöghullám: a szélsőérték közelébe érve lelassul a mozgás, majd attól távolodva újra felgyorsul. Ez természetesebb hatást ad, mint a hirtelen irányváltás. A szinuszfüggvény radiánban megadott szögeken dolgozik, ezért volt szükség a π -vel való súlyozásra. A mozgás gyorsaságát a paraméter, a távolságot pedig a paraméter változtatásával szabályozhatjuk.



4.4. ábra. Szinuszfüggvény és a hozzá tartozó mozgás.

Forrás: a szerző saját készítésű ábrája

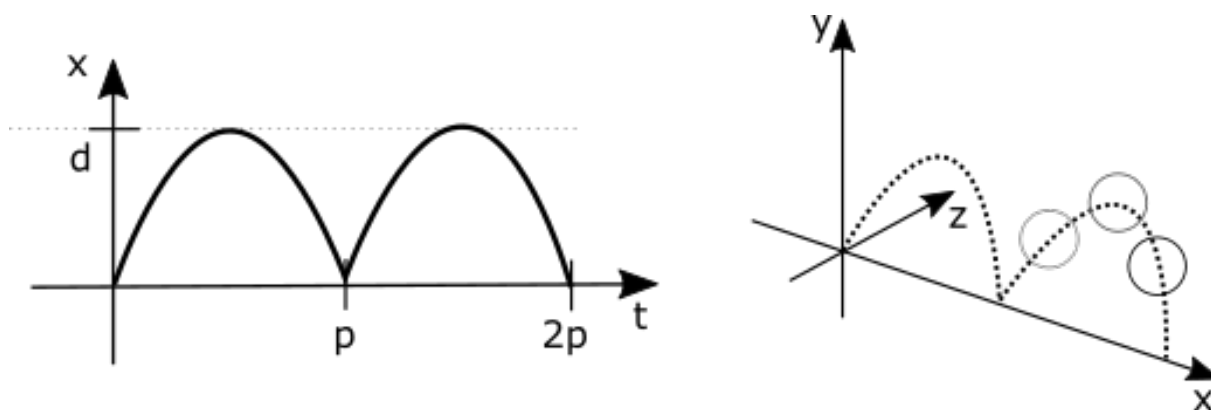
A természetes mozgásra példa egy objektum feldobása és leesése vagy egy labda földön való pattogása. Ez utóbbi szintén egy oda-vissza mozgás, akárcsak az eddig tárgyaltak, de más karakterisztikája van. A felfele haladó ívben úgy viselkedik, mint a szinuszfüggvényünk: fokozatosan lassul az irányváltás előtt, és irányváltás után újból gyorsul. A talajhoz közeledve azonban folytatódik a gyorsulás, és a talajjal való érintkezés után hirtelen irányváltás következik be. A pattogó mozgás megvalósításához a háromszöghullám-függvényből indulhatunk ki, és kombináljuk egy hatványfüggvénnyel:

$$x(t) = d \left(1 - \left| 1 - 2 \cdot \text{frac} \left(\frac{t}{p} \right) \right|^2 \right)$$

Ennek a függvénynek az alakját láthatjuk a 4.4. ábra bal oldalán. Egy előre elhajított test, illetve előre pattogó labda esetében a mozgás már nem egydimenziós, újból egyenletrendszert kell felírunk. Egy irányba történő pattogás a következőképp írható le:

$$\begin{aligned} x(t) &= a \cdot t, \\ y(t) &= d \left(1 - \left| 1 - 2 \cdot \text{frac} \left(\frac{t}{p} \right) \right|^2 \right), \\ z(t) &= 0. \end{aligned}$$

Ezt a mozgást mutatja az 4.5. ábra jobb oldalán látható kép. Azt, hogy a labda milyen magasra pattan, határozza meg, a pattogás gyakoriságát pedig , az előre haladás sebességét pedig . Itt tehát más-más jellegű függvényeket használtunk az egyes tengelyekhez.



4.5. ábra. Pattogó labda mozgása periodikus hatványfüggvénnyel
 Forrás: a szerző saját készítésű ábrája

Utolsó példaképp tekintsük a körmozgást. Egy síkon történő körmozgást leíró egyenletek:

$$x(t) = r \cdot \cos\left(\frac{2\pi t}{p}\right) + c_x,$$

$$y(t) = r \cdot \sin\left(\frac{2\pi t}{p}\right) + c_y,$$

$$z(t) = 0.$$

Itt egy , középpontú, sugarú körön történik a mozgás. A teljes kört időegység alatt tesszük meg.

Képletanimációt tehát akkor érdemes használni, ha matematikailag jól megfogható az adott mozgás. Nagy előnye a tömör formája és a könnyű kiértékelhetősége. A kiértékelésnél az is fontos, hogy bármikor, bármelyik időpontban lekérdezhettük az aktuális értéket. Hátránya, hogy – bár a függvények kombinálásával igen komplex mozgások is megvalósíthatók – nem tudunk minden mozgásra egyszerű képletet felírni, illetve a képletek paraméterei sokszor nem elég intuitívak. A képletanimáció hatékony használatához meg kell tanulnunk függvényekben és azok kombinációiban gondolkodni, ami némi tapasztalatot igényel. Ugyanakkor érdemes egy kis tapasztalatot szerezni a képletanimáció témakörében, ugyanis a legtöbb 3D rendszer felkínál valamilyen szkriptelési lehetőséget, így az objektumainkat közvetlenül képlettel is vezérelhetjük. Az ilyen szkriptek nem igényelnek mély programozói tudást, inkább a függvények ismeretét.

Meg kell jegyeznünk, hogy az itt felsorolt példák mind a pozíció változásáról szólnak. Ez nem jelenti azt, hogy a képletanimáció csak a pozíció animációjára jó, gyakorlatilag bármilyen paraméter változása leírható a módszerrel. Lineáris függvényt használhatunk például egy objektum egyenletes forgására vagy a képernyő fényerejének folyamatos növelésére stb. Háromszöghullám- vagy szinuszfüggvénnyel megvalósíthatjuk ritmikus fel-lebillenéseket, fényforrások fényerejének pulzálását, egy tárgy méretének ritmikus változását stb. A lehetőségek száma végtelen.

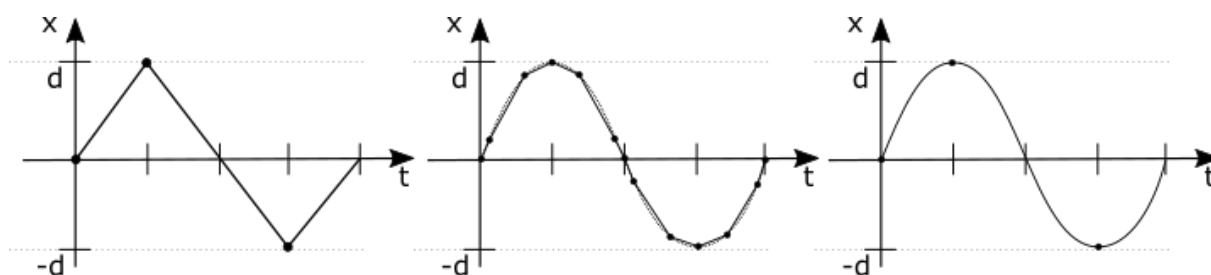
4.2 Keyframe-animáció

A klasszikus animációk során – mint amilyen egy gyurmaanimáció is – először meghatározunk egy kívánt másodpercenkénti képkockaszámot, majd minden képkockát külön rögzítünk. Ez azt jelenti, hogy minden képkockában meg kell határozni a világ új állapotát, annak megfelelően megváltoztatni azt (például kicsit megemelni a gyurmafigura karját), és új képet készíteni. Ez rendkívül időigényes feladat, és a mozgás ilyen formájú ábrázolása rendkívül nagy tapasztalatot igényel. A hagyományos rajzfilmek esetében annyiban javul a helyzet, hogy meg tudunk rajzolni úgynevezett kulcskockákat (keyframe), amelyek egy mozgás fontos állomásait tartalmazzák. Egy kézfelemelésnél például a leeresztett és a felemelt állapot lehet egy-egy kulcsfázis. A köztes képkockák (fáziskockák) következnek a kulcskockákból, akár egy kevésbé tapasztalt rajzoló is meg tudja rajzolni. Ezzel a technikával a mozgás egyenletessége könnyebben biztosítható, mint amikor szigorúan képkockáról képkockára kell haladnunk.

A hagyományos kulcskocka-animáció ötlete átvihető a számítógépes animációra is. Ha ismerjük az objektumok paramétereinek értékét bizonyos kulcsfontosságú időpillanatokban, akkor a köztes értékeket automatikusan tudjuk számítani. Ezt a számítógépes grafikában is kulcskocka-animációknak vagy kulcsolásnak nevezzük.

A képletanimációknál ismertetett példákat újra megvizsgálhatjuk a kulcskocka-animáció szempontjából. Egyenes vonalú egyenletes mozgás megvalósításához egy kezdőpont és egy végpont koordinátára van szükségünk, a két pont közötti értékek egyszerű súlyozással megkaphatók. Hasonló a helyzet az oda-vissza mozgással, csupán ott periodikusan ismétlődik a két szélső koordinátához tartozó kulcskocka. Ha ez a mozgás nem hirtelen vált irányt, akkor több lehetőségünk van. Vagy felvesszünk újabb köztes kulcskockákat, hogy finomítsuk a mozgást, vagy azon változtatunk, ahogy a két szélsőérték közötti köztes értékeket számítjuk, és az egyszerű lineáris átmenet helyett bonyolultabb súlyozást használunk.

A 4.6. ábrán mutatjuk be az előbb vázolt megoldásokat. A bal oldali képen a legegyszerűbb oda-vissza mozgást láthatjuk: felvettünk kulcspontokat a kritikus időpillanatokban, ezeket pontokkal jelöltük az ábrán, és a köztes értékeket sima szakaszátmenettel számítjuk. A középső képen a hirtelen irányváltás finomításához új kulcspillantokat veszünk fel, és ezekben meghatározzuk a kívánt értéket. A köztes időpillanatokban szintén egyszerű lineáris súlyozást használunk, ezzel a kívánt íves szinuszgörbénket törtvonallal közelítjük. A kívánt és a tényleges mozgás közötti különbség jól látszik: szaggatott vonallal berajzoltuk a szinuszgörbét, és ahol a törtvonal jelentősen eltér ettől a görbétől, ott torzul a mozgás. Ennél a megoldásnál több kulcspontot kell felvenni, ami plusz erőfeszítést jelent. Minél több kulcspontot veszünk fel, annál jobban közelítjük a kívánt mozgást, de annál hosszadalmasabb ez a folyamat. A jobb oldali képen a kulcspontok számát minimalizáltuk, megegyezik a bal oldali képen láthatóval, és a súlyozást módosítottuk annak megfelelően, hogy milyen ívben szeretnénk az egyik kulcsállapotból a következőbe eljutni.



4.6. ábra. Periodikus mozgás megvalósítása kulcskeret-animációval

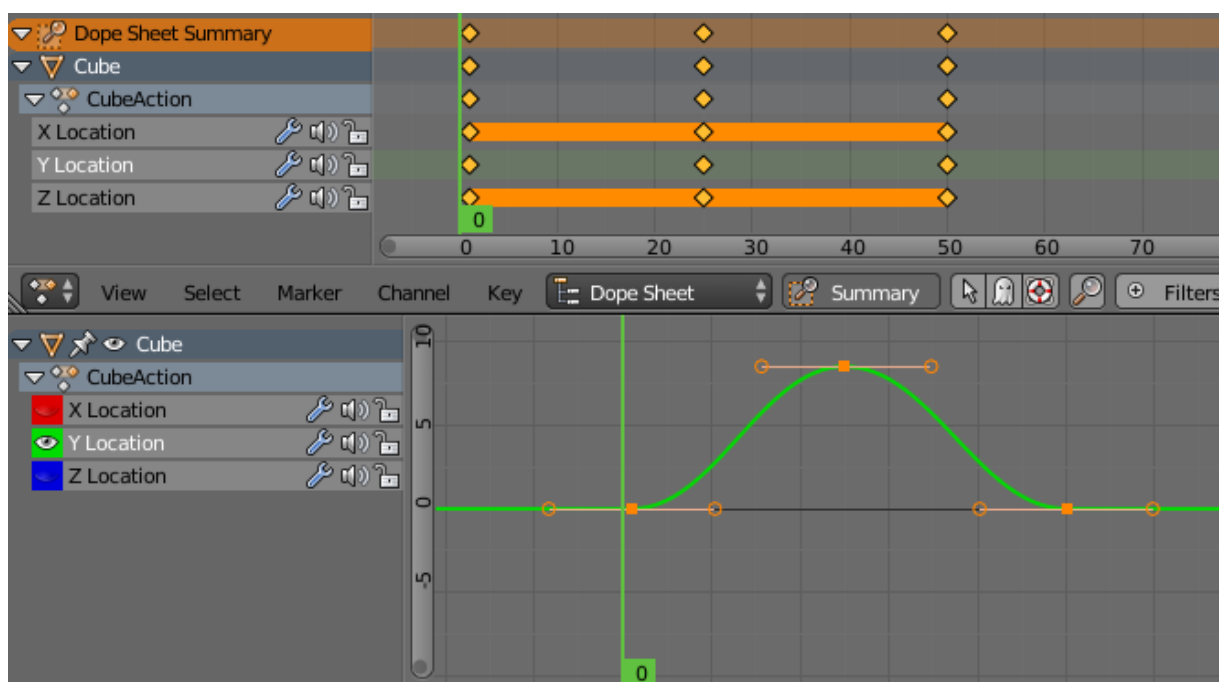
Forrás: a szerző saját készítésű ábrája

A kulcsoláson alapuló számítógépes animációs rendszerek fontos elemei ennek a súlyozásnak a megadási lehetőségei, melyekkel a mozgásgörbéinket kedvünk szerint alakíthatjuk. A legtöbb animációs rendszerben a kulcspontjaink közötti interpolációt egy ún. Bézier-görbével valósítjuk meg. Ennek a görbének a pontos matematikai alakja a gyakorlati felhasználás szempontjából nem annyira fontos, sokkal inkább a működésének megértése. Elemi Bézier-görbék összekapcsolásával olyan interpolációs görbét tudunk előállítani, ami áthalad előre megadott pontokon. A mi esetünkben ezek a pontok a kulcspontokban felvett értékek.

A Bézier-görbe előnye, hogy nagymértékben testre szabható az, hogy két kulcsérték között milyen ívet járjon be a görbe. Ezt úgy tudjuk megtenni, hogy minden csúcspontoz két érintőérték tartozik, egy bal oldali és egy jobb oldali. A jobb oldali azt határozza meg, hogy az előző kulcsértékből milyen ívben fusson be, míg a bal oldali azt mondja meg, hogy következő csúcspont felé milyen ívben távozzon a görbe. Matematikailag ezek az érintők a görbe bal és a jobb oldali deriváltjainak felelnek meg a kulcspontban.

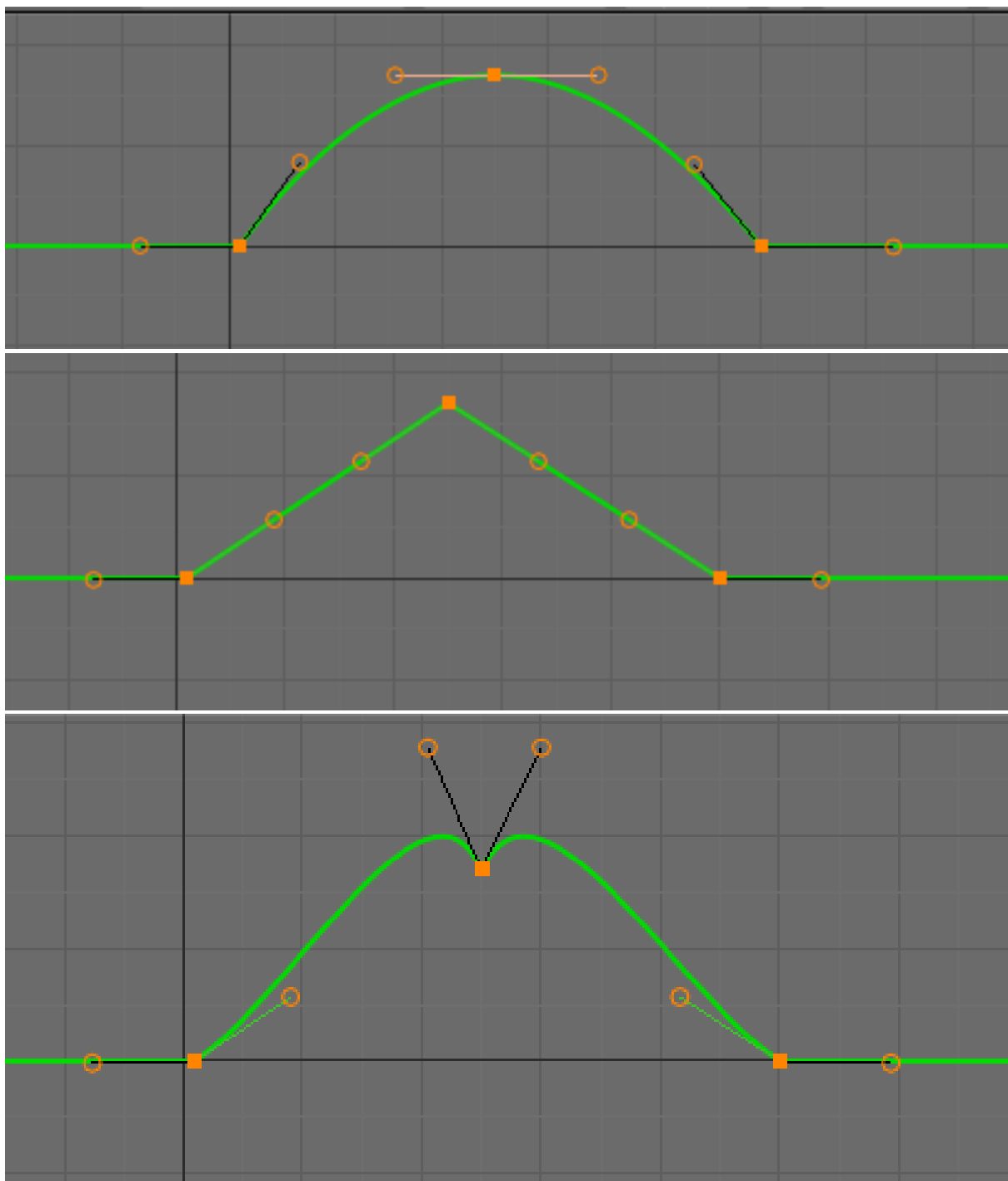
A 4.7. ábra a Blender nevű háromdimenziós modellező és animációs szoftver két animációs szerkesztőablakát mutatja. Ezek az animációs nézetek minden komoly animációs szoftverben megtalálhatók. Az egyik ilyen nézet az ún. DopeSheet, amelyet angolul még ExposureSheet vagy XSheetnek, a magyar hagyományos animációs szaknyelvben pedig expozíciós listának hívunk. A hagyományos animációban az expozíciós lista az egyes képkockák tartalmát határozza meg, tartalmazhat a képkockákhoz tartozó dialógusszövegeket, kameramozgás-utasításokat. Az animációs szoftverekben a DopeSheet egy táblázatszerű megjelenítés, ahol az oszlopok az animáció képkockáinak felelnek meg, a sorok pedig az objektumok időben változó tulajdonságainak (animált tulajdonságok). Ha egy tulajdonsághoz a táblázat egy képkockához tartozó elemében bejegyzés van, akkor az a képkocka kulcskockát jelent.

A 4.7. ábra alsó felén látható nézet az animációsgörbe-szerkesztő ablakot mutatja. Ez gyakorlatilag egy függvényszerkesztő nézet, ahol a vízszintes tengelyen az idő (képkockában vagy másodpercben megadva), a függőleges tengelyén pedig az animált paraméter felvett értékeit ábrázoljuk. Egy objektum mozgásakor valójában három ilyen függvénygörbét jeleníthetünk meg, illetve szerkeszthetünk: a pozíció három fő tengely szerinti koordinátáinak időbeli változását.



4.7. ábra. Expozíciós lista (Dope sheet) és animációsgörbe-szerkesztő ablak a Blenderben
 Forrás: a szerző saját készítésű ábrája

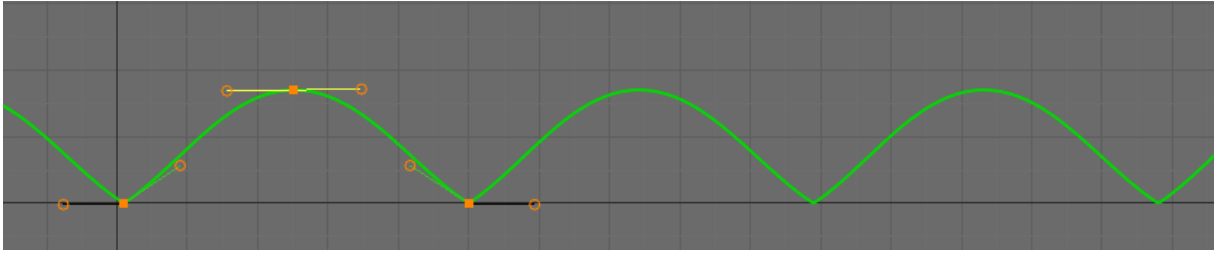
Az animációsgörbe-szerkesztő nézetben a kulcskockákat csomópontokkal jelöljük, a csomópontok között a függvénygörbe alakját csomóponti érintők beállításával szabályozhatjuk. A 4.8. ábra ugyanazon csomópontértékekkel, de más érintőbeállításokkal mutat be animációs görbéket. Az érintőkkel az íves átmenet (felső ábra), a hirtelen átmenet (középső ábra), de teljesen szabad mozgások túllendüléssel (alsó ábra) is megvalósíthatók.



4.8. ábra. Az animációs görbe és így az animáció pályájának szabályozása animációs görbék segítségével

Forrás: a szerző saját készítésű ábrája

A legtöbb animációs rendszer lehetőséget ad ismétlődő animációk lejátszására is. Ilyenkor elég egy periódus animációját elkészíteni, ezt már tudja ismételni a rendszer. Erre mutat példát a 4.9. ábra. Ilyenkor ügyelnünk kell arra, hogy az utolsó kulcspontban és az első kulcspontban beállított értékek megegyezzenek.

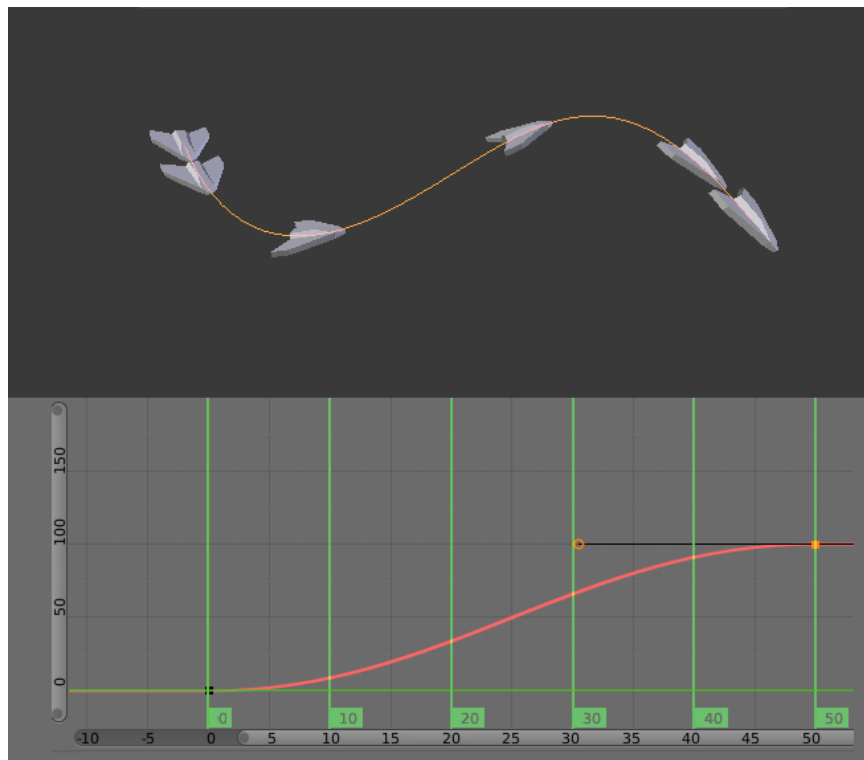


4.9. ábra. Ismétlődő animáció animációs görbéje
 Forrás: a szerző saját készítésű ábrája

4.3 Pályaanimáció

Egy test mozgását legtermészetesebb módon az általa leírt mozgáspálya megrajzolásával adhatjuk meg. Az előző alfejezetben leírt kulcskeret-animáció során a mozgáspálya néhány kulcsfontosságú pontját adtuk meg, a köztes értékek interpolációval adódtak. A koordináták animációjának független megadásával a mozgáspálya csak közvetlenül adódik, kényelmesebb lenne magának a háromdimenziós görbének a megadása. Ezt nevezzük pályaanimációnak, amely során a test egy adott pontja végigfut a felhasználó által megadott görbén.

Maga a görbe bármilyen matematikai formulával megadható görbe lehet, tipikusan Bézier- vagy NURBS-görbéket szoktak használni. A felhasználó a görbe szerkesztése során folyamatos visszajelzést kap a végső mozgáspályáról. A mozgatni kívánt testen ki kell jelölnünk egy pontot, ami a görbén végig fog futni, és meg kell határoznunk a test előre irányát, hogy a testet mindig a görbe haladási irányába tudjuk forgatni. A 4.10. ábra felső képén egy pályaanimációval mozgatott repülő láthatunk az animáció különböző időpillanataiban. A repülő mindig a görbe haladási irányába fordul.



4.10. ábra. Pályaanimáció és a hozzá tartozó animációs időgörbe
 Forrás: a szerző saját készítésű ábrája

A pályaanimáció során fontos az animáció időbeli dinamikája, azaz hogy a pálya egyes szakaszain milyen gyorsan mozog az objektum. A mozgás sebessége nem adódik magából a görbéből, sőt, ha a görbe vezérlőpontjait vennénk alapul, nagyon rosszul járnánk, mert általában oda, ahol a görbén finomítani szeretnénk, több vezérlőpontot veszünk fel, és így ott gyorsabban haladna a test. Az animációs rendszerekben ezért egy külön animációs görbét tudunk megadni arra, hogy a pályát milyen ütemben kell bejárni.

A 4.10. ábra alsó képén láthatunk egy példát egy ilyen animációs görbére. Ez a görbe azt mutatja, hogy a repülő lassan indul el a pályán, középen felgyorsul és végül lassuló mozgással ér a görbe végére. Ez a felső képen jól látszik, ugyanis a kirajzolt időpillanatok egyenlő időközönként lettek felvéve (10 képkockánként), a repülő egyes képei pedig nem egyenletesen helyezkednek el a görbén. Az animációs görbe tehát az animáció ideje (itt képkockában megadva) és a görbén megtett relatív úthossz közötti összefüggést adja meg (jelen példában a 0 egységnyi úthossz jelenti azt, hogy a görbe elején vagyunk, az 50 azt, hogy a görbe közepén és a 100 azt, hogy a teljes hosszán végigmentünk). Ha ez az animációs görbe egy egyenes vonal lett volna, az állandó sebességet jelentett volna az egész útvonalon. Az útvonal-animációk az animációs görbék szabad szerkesztésével kényelmes és intuitív megoldást adnak a mozgások tervezésére.

4.4 Csúcspont-animáció

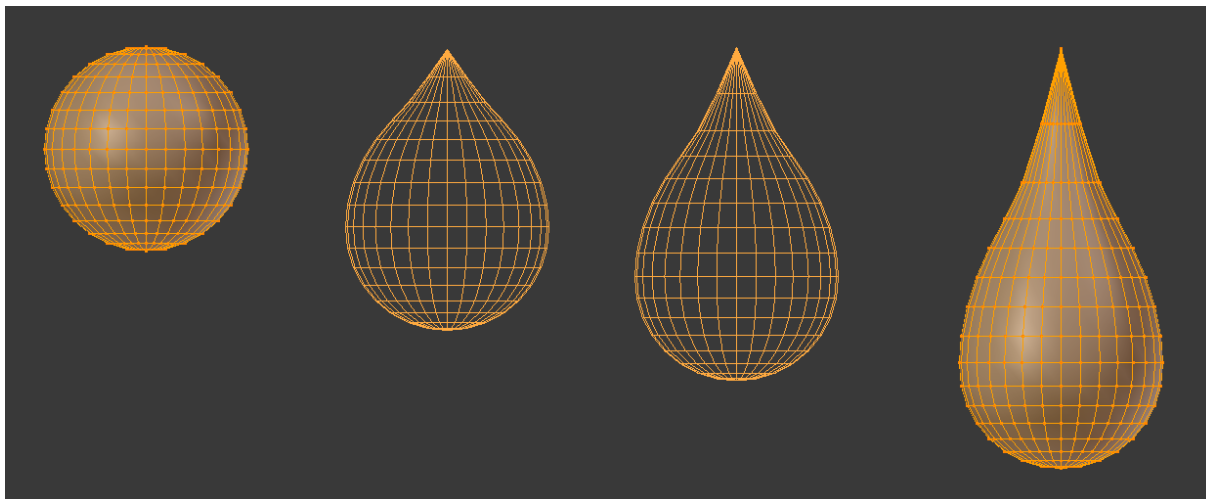
Korábbi példáink során mindig merev testekkel számoltunk, azaz csak a testek pozíciója, elforgatása változott, a test nem változtatta meg az alakját. Ez alól egyedül a skálázás animációja a kivétel: a különböző tengelyek mentén különböző mértékben történő skálázás az objektum alakjának változásához vezet. Ez azonban korlátozott változási módokat tud csak leírni. A valós életben az anyagok jelentős része igen változatos alakváltozáson tud átesni, ezeket hívjuk deformálódó anyagoknak. Ilyen például egy szivacs, egy gumicső vagy egy drapériadarab. A virtuálisvalóság-rendszerekben általában csak speciális esetekben szükséges hasonló anyagok valósághű megjelenítése. Az alakjukat változtató objektumoknak azonban van egy fontos csoportja, amely igen gyakran előkerül a virtuális világokban: ez pedig az emberi vagy állati alakok, szereplők.

Az emberi test mint háromdimenziós modell folyamatosan változtatja az alakját. Az emberi szövetek puha, deformálódó anyagok, amelyek egy merev csontváz köré épülnek. Az izmok munkájának hatására változik meg a csontok aktuális helyzete és a kívülről látható szövetek aktuális alakja. A szövetek alakváltozása között vannak látványosabbak: maga a kar felemelkedése, de vannak kisebb léptékűek: izmok kirajzolódása, ráncok, arcmimika. Ha emberi karaktereket szeretnénk ábrázolni a virtuális világainkban, ezeket az alakváltozásokat meg kell tudni jelenítenünk.

Ha visszagondolunk megint a hagyományos animációs technikákra, ott valójában minden képkockára újból megadjuk a tárgy alakját: a gyurmafigurának behajlítjuk a kezét, vagy újrarajzoljuk egy karakter animációjának a következő képkockáját. A háromdimenziós grafikában is lehetőség van ennek az animációs módszernek a használatára. Kulcsfontosságú időpillanatokban megadhatjuk a test új alakját, a köztes időpillanatokban pedig megpróbáljuk a kulcskockákból származtatni. Ezt hívjuk csúcspont-animációnak, mivel a kulcsfontosságú pontokban a geometriát leíró háromszögháló csúcspontjainak pozícióját adjuk meg.

Fontos, hogy a geometria topológiája nem változhat, azaz pontosan ugyanannyi háromszögből kell állnia, és a háromszögek szomszédossági viszonyai is megmaradnak, csupán a csúcspontok koordinátái változnak. Ez a megkötés teszi lehetővé, hogy a kulcskockák közötti képkockák értékeit automatikusan számítsuk egy egyszerű lineáris

súlyozással: a csúcspontok egyenes vonalon elindulnak a következő kulcskeretben rögzített értékeik felé. A 4.11. ábra egy csepp formálódását mutatja be csúcspont-animációval. A két kulcskockában meghatározott geometriát kitöltve, a köztes képkockákat vonalas megjelenítéssel ábrázoltuk. Az animáció során a topológia nem, csak a geometria változik.



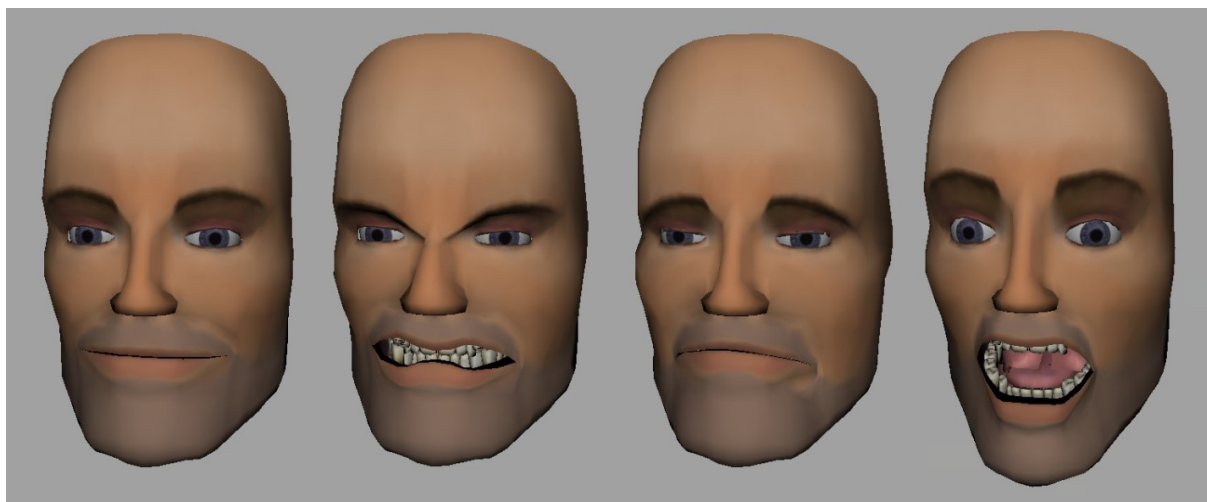
4.11. ábra. Csúcspont-animáció képkockái
 Forrás: a szerző saját készítésű ábrája

Csúcspont-animációt használhatunk az emberi mozgás megjelenítése során is. Bár régebben használtak a teljes emberi karakter megjelenítésére is csúcspont-animációt, erre a feladatra már hatékonyabb eljárások is vannak (lásd 4.5. fejezet). Végtagok animációjára különösen szerencsétlen választás ez a módszer, mert a kulcskockák közötti súlyozás valószerűtlen köztes geometriai állapotokat eredményezhet. Csúcspont-animációt használhatunk azonban az emberi arc mimikáinak és a beszéd megjelenítésére.

Ha az emberi arc mimikáit szeretnénk megjeleníteni, az egyes kifejezésekhez külön-külön elkészítjük a megfelelő geometriát, és mindig a megfelelőt jelenítjük meg. Ha nem szeretnénk hirtelen váltásokat, a semleges arckifejezésű geometria és az aktuális megjelenített kifejezéshez tartozó geometria között súlyozhatunk. Az arckifejezésekhez tartozó geometriákat Blend Shape-eknek hívjuk (4.12. ábra), nevében is utalva a súlyozásra. Valójában ennél is továbbmegyünk, és nemcsak két állapot között súlyozunk, hanem sok állapot lehet aktív, mindegyik egyedi súllyal. Így képesek vagyunk kikeverni egyedi arckifejezést például a mérges és a csalódott arckifejezés egyszerre történő használatával, megfelelő súlyozásokkal.

Bonyolultabb rendszereknél fontos gondosan előkészíteni a blend shape-halmazt, melyből dolgozunk. Bár néhány alap arckifejezés és azok összemosása is sok esetben kielégítő lehet, könnyen kikeverhetünk érvénytelen arckifejezéseket. Például ha mind az ijedt és a csodálkozó arc is kinyitja a modell száját, ha ezeket együtt alkalmazzuk, könnyen túlnyithatjuk a száját, torz geometriát okozva. Ennek elkerülése érdekében felbonthatjuk az összetett kifejezéseket alap mozdulatokra, például izomcsoportok hatásaira, így készíthetünk geometriát a bal szemöldök felemeléséről, a bal szálszél oldalra húzásáról stb. Ezen alap geometriai módosulások szabadon keverhetők, mert külön-külön csúcspontcsoportra hatnak, vagy ha azonosra hatnak is, együttes alkalmazásuknak amúgy sem lenne értelme (például a szemöldök egyszerre történő felemelésének és lehúzásának). Az alapmozdulatok meghatározásához gyakran nyúlnak segítségkép-

pen a Facial Action Coding System (FACS) rendszerhez. Ilyen bonyolult rendszereket általában csak teljes arc mozgáskövetéssel párosítva szoktunk használni, tipikusan a filmiparban, de léteznek még valós idejű arckövető megoldások is, melyeket virtuális valóság rendszerekben is alkalmazhatunk.



4.12. ábra. Arcanimáció a Blend Shape-technikával
Forrás: a szerző saját készítésű ábrája

Az emberi beszéd animációjához elkészíthetjük az összes kiejtett hanghoz tartozó geometriát, és ezeket aktiválhatjuk a mondott szövegnek megfelelően. Ezzel a módszerrel lipsync néven találkozhatunk a számítógépes grafikában (ugyanaz, mint a filmek szinkronhangjaira használt kifejezés).

4.5 Csontváz-animáció

4.5.1 Virtuális csontváz

Egy emberi karakter mozgatásához a csúcspont-animáció helyett olyan megoldást kell keresnünk, amely hatékonyan oldja meg a geometriadeformáció problémáját. Az ihletet pedig vehetjük a természetből is. Az emberi mozgást alapvetően az teszi lehetővé, hogy van egy merev csontvázunk, amelyhez az izmok tapadhatnak. Az izmok összehúzódásának hatására valójában ennek a merev váznak a csontjait mozgatjuk, amelyek magukkal viszik a rájuk tapadó szöveteket. A csontvázunk nagyon fontos tulajdonsága, hogy hierarchikus szerkezetű, azaz a csontok egymáshoz vannak kapcsolva az ízületek által. Az alkar felemelése csak úgy lehetséges, hogy a bicepsz izom a felkarcsonton ered, de az alkarcsontjain végződik. Összehúzódásakor a két végpontját közelítené egymáshoz, mely az ízületek merev összekapcsolása miatt csak a két csont egymáshoz képesti elforgásával oldható meg. A csontok hierarchikus összekapcsolása következtében az alkarcsontok magukkal viszik a kéztőcsontokat, azok pedig az ujjcsontokat.

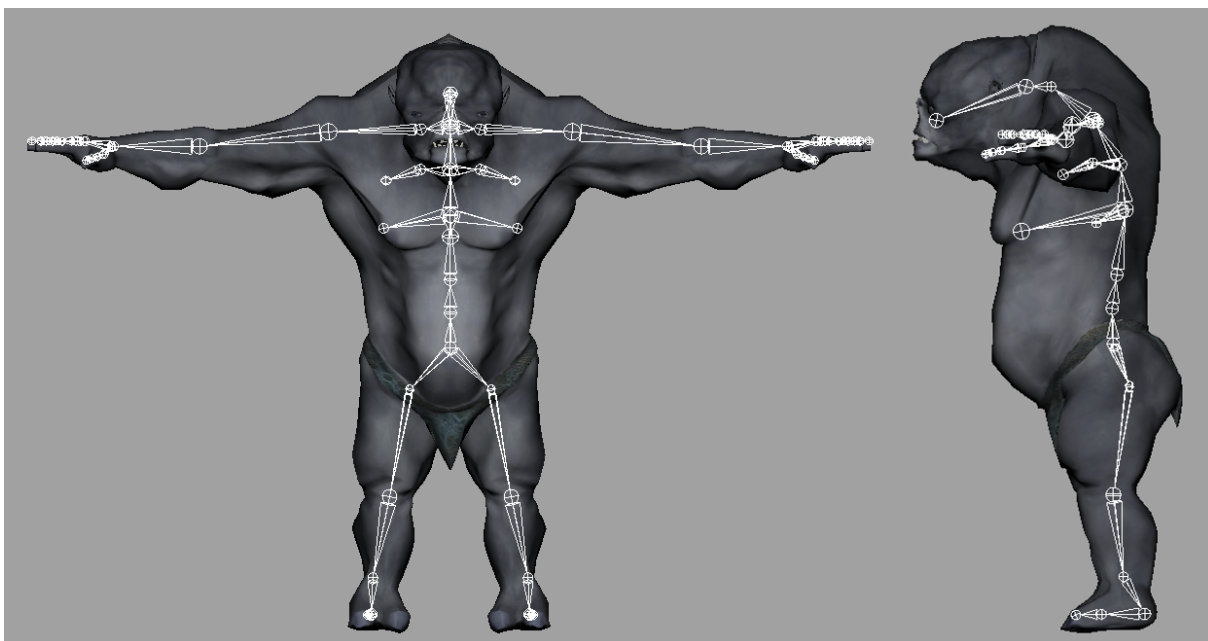
Ugyanezt az elvet követi a csontváz-animáció technikája [Magnenat-Thalman 1988]. A geometriához hozzárendelünk egy hierarchikus vázszerkezetet, a csontvázat. Ehhez a virtuális csontvázhoz nem tartozik geometria, a csontváz egyes csontjainak nincs a miénkhez hasonló hengeres alakja a végén kiszélesedő ízületi formációkkal. A virtuális csontok csak a hierarchia felépítésében és a hierarchikus mozgás leírásában játszanak szerepet. Valójában csak azt tároljuk róla, hogy mely másik csont végpontjából ered, milyen az orientációja, és mekkora a virtuális csonthossza, hogy tudjuk, hogy

a hozzá csatlakozó következő csont honnan eredjen. Azt a csontot, melyből egy adott csont ered, a csont szülőcsontjának nevezzük. Ehhez hasonlóan a csontból eredő újabb csontokat gyerekcsontoknak hívjuk. A virtuális csontvázaknak mindig van egy kitüntetett csontja, amelynek nincs szülőcsontja, hanem belőle indul ki az egész csonthierarchia, ezt a csontot gyökércsontnak hívjuk. A hierarchikus felépítésnek köszönhetően, ha egy csont elfordul a szülőcsontjához képest, az összes gyerekcsontját és annak gyerekeit is forgatja magával.

A csontváz felépítése során általában követhetjük a modellezett élőlény anatómiai leírását, de nem kell azt hűen másolnunk. Egy emberi csontváz például 206 db csontot tartalmaz. Az összes csont modellezése teljesen felesleges lenne, sokszor egy csontcsoport helyettesíthető egyetlen csonttal, ilyen például a tenyér csontjai vagy a lábfej csontjai. Az alkar két csontját szintén egy csontként szoktuk reprezentálni. Az is gyakran előfordul, hogy a gerinc csontjainak számát csökkentjük, mert nincs szükségünk a gerinc pontos működésének másolására, csak épp annyira, hogy a hátat be bírjuk hajlítani. Egy bokszoló megalkotásakor például az egész kézfej egyetlen csonttal helyettesíthető, hiszen úgyszintén a mozgásuknak hatása a kesztyűre. Hasonlóan egy teniszütőt markoló sportoló mozgásakor sem kell törődnünk az ütőre tapadó ujjakkal, hiszen azok biztosan nem fognak elmozdulni egymáshoz képest (vagy legalábbis nyugodtan élhetünk ezzel az egyszerűsítéssel).

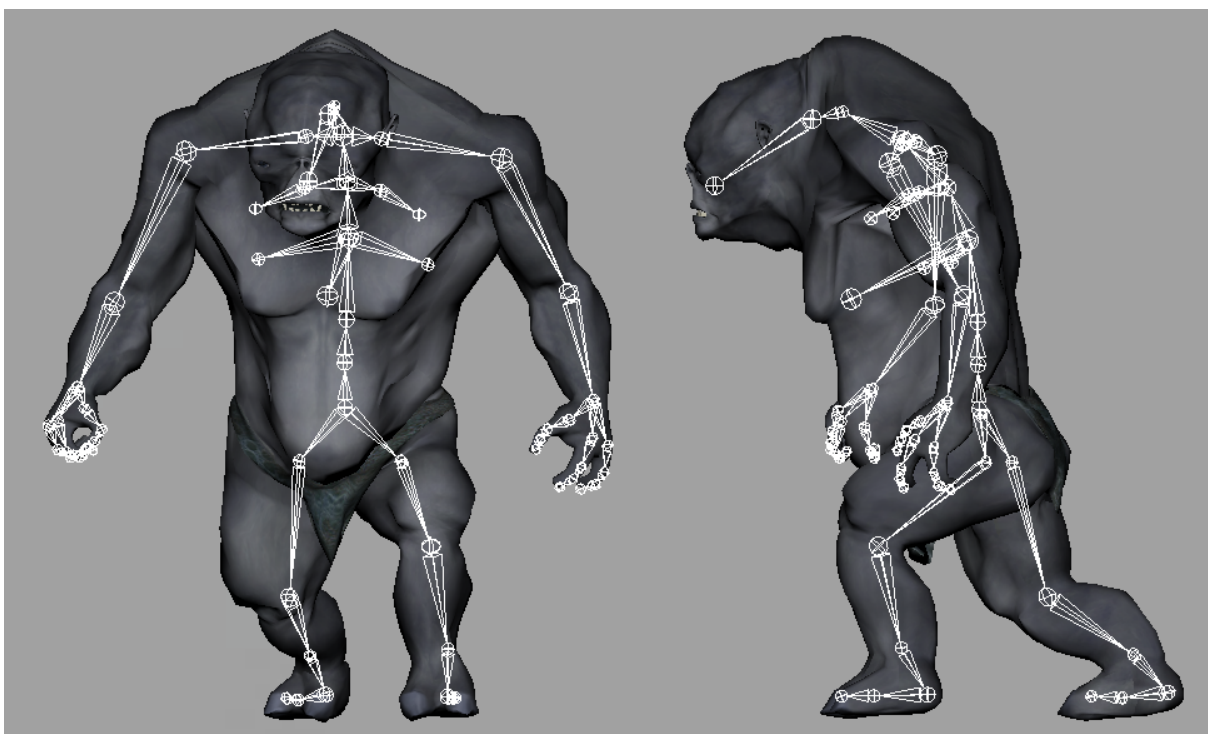
Ugyanakkor az is előfordulhat, hogy olyan csontok jelennek meg a virtuális csontvázban, amelyeket a valós csontvázak csontjainak nem tudnánk megfeleltetni. Tipikus példa erre a szemcsontok megjelenése a virtuális csontvázban. A mi szemünkben nincsenek csontok, de mivel a csontváz-animáció technikájában tetszőleges mozgás csak csontokkal érhető el, ezért a szemgolyó elfordulását is csak a szem geometriájához rendelt virtuális csonttal tudjuk megtenni.

A 4.13. ábra egy humanoid karakter virtuális csontvázát mutatja be. A csontváz csontjait piramis alakkal ábráztuk, a csont a piramis alapjánál ered, és a csúcspontjánál végződik. A csontok végpontjaiban köröket láthatunk, ezek az ízületeknek feleltethetők meg. A piramis alakokat megfigyelve megtalálhatjuk a csontváz gyökércsontját, mely a szemközti nézetben pont a köldöknél található, belőle ered a két láb, illetve a gerincoszlop. Megfigyelhető a lábfej egyszerűsödése, a bordák egyszerűsödése (ezek sok virtuális csontváznál teljesen hiányoznak is), és a gerincoszlop sem áll annyi csontból, amennyiből egy gerinces élőlény gerincoszlopa állni szokott. A 4.14. ábrán ennek a csontváznak egy animációs kulcskockában felvett állapotát látjuk. Az animáció során a karakter épp előre lép, testsúlyát a bal lábára helyezve.



4.13. ábra. Egy humanoid karakter csontváza

Forrás: a szerző saját készítésű ábrája



4.14. ábra. A csontváz egy animációs fázisa

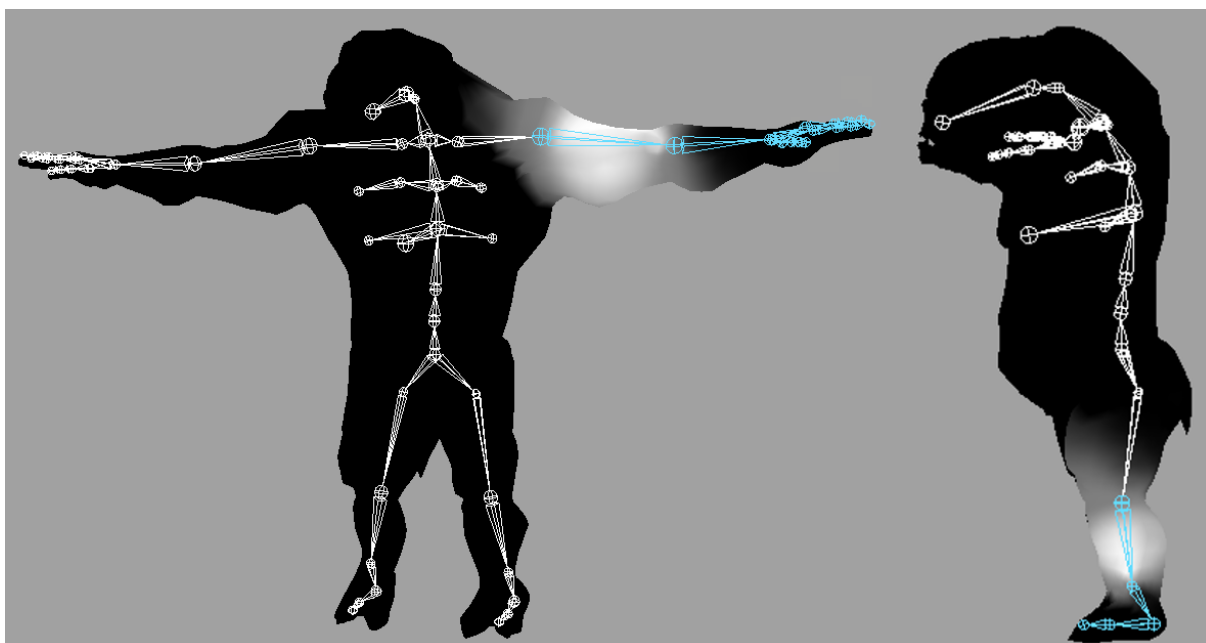
Forrás: a szerző saját készítésű ábrája

A csontváz-animáció során tehát a geometria változása helyett a csontok mozgását tároljuk el. Ez természetes animációs módszer, hasonlít a hagyományos bábanimációhoz, viszont annyi könnyebbségünk van, hogy a kulcskeret-animációt kihasználva elegendő csak a kulcsfontosságú képkockákban eltárolni a csontok elfordulásait, a köztes képkockák értékei automatikusan számíthatók a 4.2 fejezetben leírtak szerint.

4.5.2 Bőrözés

Már csak az az igencsak fontos kérdés marad hátra, hogy a csontváz mozgásának hatására hogyan fog megfelelően változni a geometria alakja. A valódi szövetekhez hasonlóan itt is azonosítjuk, hogy a geometria mely csúcspontjai mely virtuális csonthoz „tapadnak”. A csontok a mozgásuk során viszik magukkal a hozzájuk rendelt csúcspontokat is. A helyzetet annyival bonyolítjuk még, hogy egy csúcspontra nemcsak egy csont lehet hatással, hanem több is, és meg kell adnunk, hogy milyen mértékben hassanak a csúcspontra ezek a csontok. Ezeknek a súlyoknak a megadását egyszer tesszük meg, az animáció során már nem fognak változni. A csontokhoz való csúcspontok és csontsúlyok hozzárendelése a csontváz-animáció egyik kritikus pontja, amely alapvetően befolyásolja az animáció minőségét, így érdemes nagy hangsúlyt fektetni rá.

A 4.15. ábra bal oldalán a karakter bal felkarcsontjának hatását (súlyát) láthatjuk a hozzá rendelt csúcspontokra. Szürkén jelennek meg azok a csúcspontok, amelyekre ez a súly nem zérus, tehát a csonthoz vannak rendelve. Minél világosabb ez a szín, annál nagyobb hatása van a csontnak az adott csúcspontra. A teljesen fekete területek tehát azt jelzik, hogy az adott csúcspontra nincs hatással a kiválasztott csont; a teljesen fehér területek pedig azt, hogy csak az adott csont van rá hatással: egy az egyben felveszi annak transzformációját. Láthatjuk, hogy a felkarcsont tengelyéhez közel eső csúcspontok teljes súlyt kaptak, míg a távolabb esők egyre kisebb súlyokat. Az ábra jobb oldalán a bal lábszárcsonthoz tartozó súlyokat is megmutatjuk példaképpen.



4.15. ábra. A csontváz egy-egy csontjához tartozó csontsúlyok
 Forrás: a szerző saját készítésű ábrája

Az animációs szoftverek megfelelő eszközöket biztosítanak ezeknek a súlyoknak a megadására. A legkényelmesebb mód az, ha a felhasználó egy háromdimenziós virtuális ecset segítségével ráfesti ezeket a súlyértékeket a geometria csúcspontjaira. Konkrét érték megadására általában a teljesen fekete és fehér területeknél van szükség, a köztes területeken általában egy puha szélű ecsettel fokozatosan világosítunk vagy sötétítünk. Általában lehetőség van a súlyok szomszédos csúcspontokból történő átlagolására,

amellyel az éles változásokon finomíthatunk. A geometria csontvázhoz rendelésének és a csontsúlyok megadásának folyamatát bőrözésnek (skinning) nevezzük.

4.5.3 Bekötési póz

Azt már láttuk, hogy a csontváz-animációhoz szükségünk van egy csontvázra és egy kiindulási geometriára, amit a csontvázhoz rendelhetünk. Felmerül azonban kérdés, hogy a kiindulási geometria – és így a csontváz kiindulási helyzete – milyen pózban legyen. A geometriát egy adott pózban tudjuk csak megmodellezni, és ehhez a pózhoz kell a csontváz kiindulási állapotát igazítani, mielőtt a csontvázat és a geometriát egymáshoz rendelnénk. Ezt az állapotot bekötési póznak (binding pose) nevezzük, és fontos a jó megválasztása.

A csontok mozgása szempontjából nem annyira fontos a kiindulási állapot, a geometria modellezése és torzulása szempontjából viszont sokkal inkább. Az alappóz megválasztásánál vannak ésszerű meggondolások, például nem érdemes aszimmetrikus pózt modellezni, ahol a test jobb oldala más pózt vesz fel, mint a bal. Már csak azért sem, mert a modellezést rendkívül felgyorsíthatjuk, ha csak egy oldalt modellezünk meg, és ezt tükrözzük a másik oldalra. Nem érdemes olyan pózt sem modellezni, ahol a testrészek közel kerülnek egymáshoz, például egy keresztbe font kar nagyon rossz választás, mert nehezen férünk hozzá a kar és a mellkas geometriájához, nehezen járjuk körbe modellezés során. Nem érdemes olyan pózt választani, amely extrém ízületi állapotokhoz tartozik, például összegömbölyödött gerinc vagy felhúzott lábak. A karakter legtermészetesebb állapotához közel álló pózt érdemes választani, ami a modellezést is könnyíti.

Egy gyakran használt ilyen póz a 4.13. ábrán bemutatott úgynevezett T póz. Nevét a széttárt karok miatt formált T alakról kapta. A lábak nyújtottak, enyhe terpeszben, hogy minden oldalról hozzájuk tudjunk férni a modellezés során. A gerinc feszítetten egyenes, mint amikor kihúzzuk magunkat. A karok pontosan vízszintesen oldalra vannak emelve, így szintén könnyen hozzáférhetők a modellezés során. Ez a kartartás azért szerencsés, mert a csontok elforgatása pontosan megadható, hiszen pontosan vízszintben vannak. Ezt a tartást egy valódi ember is fel tudja venni, ha referenciaként kell használni (főleg mozgáskövetésnél és valós mozgások rögzítésénél kap ez nagy jelentőséget). Ugyanakkor az oldalra felemelt kar a kar teljes mozgástartományának felénél van, ami a bőrözésen alapuló animációnál fontos.

Sajnos a bőrözéses animációnak megvannak a maga korlátai. Túl nagy csontelfordulásoknál a geometria túlságosan torzulhat, megjelenhetnek csavarodások, nem kívánt kidudorodások, behorpadások. Ezen a csontsúlyok aprólékos megfestésével tudunk segíteni, illetve be tudunk vezetni segédcsonthoz, amelyek megakadályozzák a geometria bizonyos pontjainál a nem kívánt behorpadásokat. Akármilyen körültekintőek is vagyunk a csontozás és súlyfestés során, a hibákat nem tudjuk teljesen kiküszöbölni, de ha a bekötési póztól nem térünk el jelentősen, akkor nem lesznek feltűnőek. Éppen ezért a bekötési póz megválasztásánál azt is érdemes figyelembe venni, hogy a karakterünk milyen mozgásokat fog majd végezni.

Ha egy karakter főleg gyalogol, fut, illetve kényelmes kartevékenységet végez, akkor érdemesebb kevésbé felemelt karral megmodellezni. A körülbelül 45 fokban megemelt karú pózt A póznak hívjuk az általa formált alak miatt. Ez egy igen gyakran használt modellezési póz. A T pózt akkor részesíthetjük előnyben az A pózzal szemben, ha tudjuk, hogy a karakterünk gyakran emeli fel a karjait, de a testéhez közel is sokat fogja tartani. Ilyenkor a mozgástartomány felénél, tehát vízszintesen érdemes a karokat tartani.

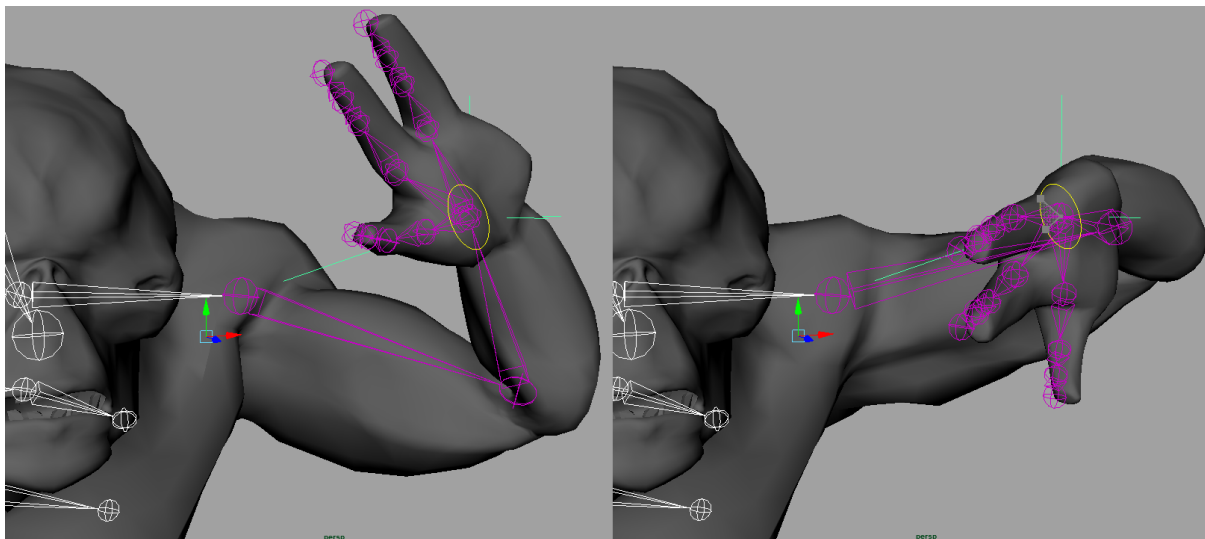
4.5.4 Inverz kinematika

A csontváz-animáció során tehát a csontokat mozgatjuk, a geometria automatikusan változik a csontok mozgására. Bár a csontok képkockáról képkockára történő mozgása rendkívül időigényes feladat, mégis vannak eszközeink, amelyek ebben támogatnak minket. Egy ilyen eszköz az inverz kinematika. A kézfej magunk elé emelésekor a felkarcsontunk és az alkarcsontunk forgatásai is megváltoznak. Ezzel két probléma is van: először is az animációhoz megalkotása során két csonttal is törődnünk kell, másodsor pedig ha pontosan szeretnénk pozicionálni a magunk elé tartott kézfejet, akkor a két csont elfordulását egymással összhangban kell tudnunk beállítani, hogy a kézfej a helyére kerüljön.

Azt az animációs módszert, ahol egy csontokból álló hierarchikus lánc végpontjának pozícióját a lánc elejéről indulva, egymás után a csontok forgatását megadva kapjuk meg, előreemenő kinematikának nevezzük. Ennek ellentéte az inverz kinematika (IK), ahol a hierarchia lánc végpontjának pozícióját közvetlenül adjuk meg, és megpróbáljuk automatikusan számítani a lánc egyes csontjainak az elfordulásait úgy, hogy a végpont a helyére kerüljön.

Erre mutat példát a 4.16. ábra, ahol egy zöld vonal mutatja a hierarchikus lánc kezdő- és végpontja közötti szakaszt. Itt tehát felvettünk egy inverz kinematikai segédeszközt a vállízület és a csuklóízület között. Az IK végpontját megváltoztatva a rendszer automatikusan számítja a felkarcsont és alkarcsont elfordulásait, hogy a csukló a lehető legközelebb kerüljön a kívánt végponthoz.

Fontos itt megjegyezni, hogy a csukló adott pontra emelését sokféleképpen megtehetjük. Valójában az IK kezdő- és végpontja által meghatározott vonal mint tengely körül az egész hierarchikus lánc tetszőleges szöggel elfordulhat, továbbra is a helyére kerül a végcsont. Így általában lehetőségünk van az animációs rendszerekben ennek az elfordulásnak a szabályozására.



4.16. ábra. A karbehajlítása inverz kinematika segítségével

Forrás: a szerző saját készítésű ábrája

A 4.16. ábrán egy sárga kör jelöli ezt a segédeszközt. Az ábra bal oldalán láthatjuk, hogy ennek a körnek az elforgatása (az elforgatás szöge is látszik a körlapon) a kart oldalra, a testtől elemelte, de a csukló továbbra is a kívánt helyen maradt. IK-lánccokat szoktak

használni a karokra és a lábakra a mozgások animációinak létrehozásakor, de a gerinc hajlítására is lehet inverz kinematikát alkalmazni.

4.5.5 Animációs állapotok és kombinációk

A csontváz-animációt valós idejű rendszereknél általában úgy használjuk, hogy egy karakter jellemző mozgásait előre eltároljuk, és a program futása során mindig a karakter állapotának megfelelőt játsszuk le. Ilyen mozgás lehet a futás, járás, ugrás. Mivel a mozgás és a karakter mint az alkalmazásunk egy szereplőjének állapota között erős összefüggés van, az animációk lejátszása az alkalmazás logikájának szerves részét képezi. Ezért sok játékmotorban fejlett animációs állapotrendszert találhatunk (ilyen például a UnityMecanim rendszere).

A megjelenítés szempontjából érdekes még az, hogy nem feltétlenül kell ahhoz ragaszkodnunk, hogy a karakterünk egyszerre csak egy animációs állapotban lehet. Ennek egy igen egyszerű példája a járás és futás. Ha a karakterünk az álló pozícióból hirtelen vált járásba, majd onnan futásba, az nem ad természetes hatást, hiszen a valós életben folyamatos átmenetet látunk ezen mozgások között. Ezt a folyamatos átmenetet elérhetjük úgy, hogy az előre eltárolt animációkat keverjük, azaz egyszerre játsszuk le a futás- és a járásanimációt, de ezek hatását súlyozottan jelenítjük meg. Ha a két animáció össze van hangolva, szép eredményt kaphatunk. Így például a járásanimáció is előre nyújtja a lábat, illetve hajlítja a combot, csak kisebb mértékben, mint a futásanimáció. A két animáció keverésével a láb hajlítása valahol a két animáció között lesz, gyakorlatilag kocogni fog a karakter. Ahogy a futásanimáció súlyát növeljük, és vele párhuzamosan a járásét csökkentjük, a mozgás átmegey teljes futásba.

Az animációkat nemcsak keverni, hanem összeadni is lehet. Ez gyakorlatilag azt jelenti, hogy mindkét animáció teljes súllyal szerepel, mindkettő hatása teljes mértékben érvényesül. Ennek a futás- és a járásanimációnál nem igazán van értelme, mert a futás lábhajlítására ráadva a járás lábhajlítását túlzott mozgást kapnánk, de más esetekben jó hasznát tudjuk venni. Tegyük fel például, hogy a virtuális karakterünk a kezébe tud venni valamilyen szerszámot, és azzal valamilyen műveletet tud végezni. Ez egy játék esetében lehet például egy kard és a vele való vágás mozdulat. Ha a karakterünk tud járni, futni, ugrani, sőt guggolni is, akkor elvárhatnánk, hogy tudjon állva, járva, futva, ugorva és guggolva is suhintani a kardjával. Ekkor el kellene készítenünk a járás, futás, guggolás és ugrás animációkat, és ezeknek a karddal való suhintással kombinált változatait is. Ha a használható eszközök száma vagy a használati módjuk száma nő, drasztikusan megugrik az elkészítendő animációk száma is.

Ezt elkerülhetjük úgy, hogy bizonyos animációkat csak bizonyos csontcsoportokra definiálunk. Az előző példánál maradván a suhintás csak a karokat, esetleg az egész felsőtestet érinti, míg a járás, futás, ugrás, guggolás pedig csak a lábakat. Ezen animációk teljes súlyú összeadásával már nincs gond, mert külön csontcsoportokra hatnak, és így jelentősen csökkent az elkészítendő animációk száma. Az animációk ilyen keverésére a legtöbb animációs rendszerben van lehetőség.

4.6 Fizikai szimuláció

Az eddig tárgyalt animációs módszereknél a felhasználó feladata volt a tárgyak mozgáspályájának megtervezése. Bár sok hatékony módszer van a kezünkben a mozgások tervezésére, bizonyos mozgások élethű megvalósítása meglehetősen nehéz feladat, és nem is biztos, hogy előre elvégezhető egy animációs fázisban, animátor által. Ilyen például egy virtuális világban lévő objektum földre esése, vagy más tárgyakkal való ütközése. Az objektumok ütközésekor fellépő kölcsönhatásokat a fizika törvényei írják le. Az összefüggésekben számos paraméter szerepel, amelyek így befolyásolják az objektumok mozgását. Gondoljunk csak arra, hogy két objektum ütközésekor, ha az egyik objektum a másik szélét találja el, akkor az nemcsak irányt változtat, hanem megpördül, forgásba kezd. A forgás mértéke függ attól is, hogy az ütközés hol következik be a két objektum között, tehát a mozgási lehetőségek száma olyan nagy, hogy azokra előzetes animációs sémákkal nem lehet felkészülni.

A virtuális világokban fellépő mozgások fontos részét képezik a fizikai szimulációk által irányított mozgások. Ezeket a mozgásokat olyan szimulációs keretrendszerek számítják, amelyek képesek kezelni a tárgyak ütközéseit és a newtoni fizika törvényszerűségeit. A fizikai rendszereknek két fő feladatuk van. Az egyik az ütközések detektálása. Mielőtt bármilyen fizikai képletet kiértékelhetnénk, azonosítani kell azokat az objektumokat, amelyek interakcióba lépnek egymással. Ez az ütközésetektálás feladata.

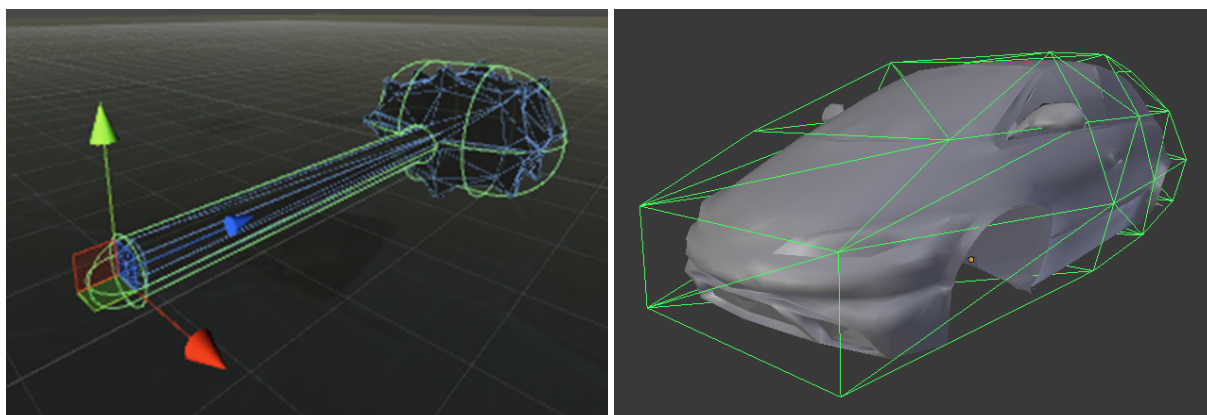
Az ütközések során azt vizsgáljuk, hogy két geometriai alakzat között van-e átlapolódás. Ebből következik, hogy a fizikai rendszernek is szüksége van egy leíróra a geometriáról. Mivel a testek átlapolódásának vizsgálata komplex számítást is igényelhet, a fizikai rendszerek általában egyszerűbb geometriával dolgoznak, mint az objektum végső megjelenése. Az ütközésetektáló rendszer szempontjából az az ideális, ha az objektumok alakját meg tudjuk adni matematikai formában, ezért gyakran használunk téglatest, gömb, henger és más egyszerűen leírható alakzatokat (ezeket primitív alakzatoknak hívjuk). A virtuális világ fizikai geometriai reprezentációit fizikai befoglalóknak vagy ütköző-geometriának hívjuk (találkozhatunk még a kizáró kifejezéssel is).

Sok objektum gond nélkül leírható matematikai formában. Egy labdának például tökéletesen megfelel egy gömb, akárcsak egy biliárdgolyónak. Tulajdonképpen a tökéletes gömbök esetében ez a reprezentáció pontosabb lesz, mint a grafikai leírás, hiszen a grafikai geometriát mindig háromszöghálóvá alakítjuk, ami csak közelítheti a tökéletes gömböt. A legtöbb esetben azonban a fizikai befoglaló csak egy közelítést ad a tárgy tényleges alakjára. Például egy dinnye is leírható gömb alakban, bár a valóságban nem tökéletes gömb. Ugyanez igaz egy almára is, a fizikai szimuláció szempontjából sokszor eltekinthetünk az alma száránál és a csészeleveléknél található bemélyedések és maga a szár fizikai modellezésétől, ugyanakkor a grafikai megjelenésben biztos, hogy meg fognak jelenni ezek a részletek.

Ha egy testet nem tudunk jól közelíteni primitív alakzattal, használhatunk összetett alakzatot is, amely több primitív alakzat „összeragasztásából” adódik. Erre mutat példát a 4.17. ábra (Komplex geometriák közelítése primitívekkel és egyszerűsített fizikai befoglalóval) bal oldalán látható buzogány alakzat. Bár a teljes geometriát nem tudjuk egyetlen primitívvel közelíteni, a nyél és a fej külön-külön jól helyettesíthető egy-egy kapszulával. Ezen befoglaló alakzatok merev összekapcsolása már olyan összetett fizikai leíró eredményez, amely jól közelíti a grafikai reprezentációt.

Vannak azonban olyan alakzatok, amelyek nem bonthatók fel ilyen könnyen primitívekre. Ilyen testek esetén használhatunk egy egyszerűsített háromszög-geometriát, amely a testet jól közelíti, de alacsony geometriai komplexitású. Ezen alakzatok felé

azonban van egy fontos követelményünk: konvex alakzatoknak kell lenniük. A 4.17. ábra jobb oldalán látható autógeometriát egy egyszerűsített konvex alakzattal közelítettük.



4.17. ábra. Komplex geometriák közelítése primitívekkel és egyszerűsített fizikai befoglalóval

Forrás: a szerző saját készítésű ábrája

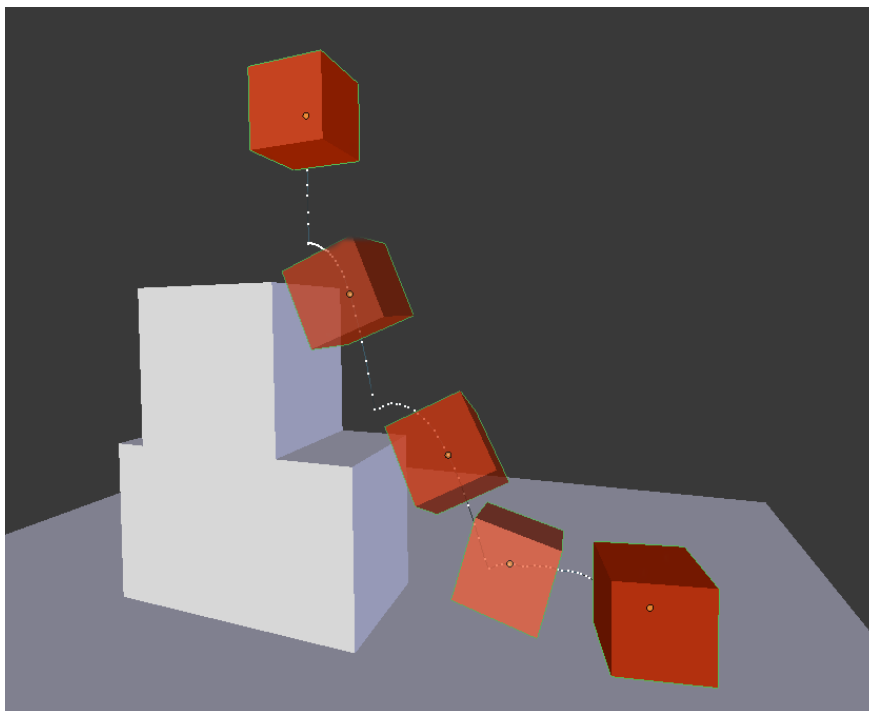
Elmondhatjuk tehát, hogy a fizikai rendszer által használt geometria csak ritka esetekben esik egybe a ténylegesen megjelenített geometriával, annál jóval egyszerűbb vagy analitikus formában megadható alakzatokat használunk.

Ha a fizikai rendszernek sikerült azonosítania az egymásba lógó alakzatokat, a következő nagy feladata az, hogy az objektumok közötti kölcsönhatásokat a dinamika törvényeinek megfelelően számítsa. Ezt a számítást időről időre el kell végezni, és a végeredményül kapott új pozícióknak és elfordulásoknak megfelelően a virtuális világot újrarajzolni. Logikusnak tűnik tehát a fizikai szimuláció minden képkockában történő újraszámítása. Figyelnünk kell azonban arra, hogy a fizikai rendszerek csak kis időlépéseket képesek stabilan számítani, így a frissítési gyakoriságuk tipikusan 1/60 másodperc körül mozog. Ha a virtuális világ kirajzolása ennél több időt vesz igénybe, akkor egy kép elkészítése közben több fizikai szimulációs lépést is futtatunk. Ha a megjelenítés nem időigényes és jóval a 60 FPS fölé esik, akkor viszont bizonyos képkockáknál kimaradhatnak fizikai szimulációs lépések.

A fizikai rendszerekben általában háromféle fizikai szereplőt szoktunk megkülönböztetni: statikus, dinamikus és kinematikus. A statikus objektumok soha nem változtatják meg a helyzetüket, semmilyen erő és ütközés nem hat rájuk. A dinamikus objektumokra hatnak a környezet erői, például a gravitációs gyorsulás, és részt vesznek ütközésekben. A mozgásuk megfelelő számításához el kell látnunk őket bizonyos fizikai jellemzőkkel. Az egyik legfontosabb ilyen jellemző a tömeg. Minél kisebb tömegű egy test, annál könnyebben fellökik őt más testek, és az erőhatásokra is nagyobb sebességváltozással reagálnak. Ha két dinamikus objektum egymásnak ütközik, lepattannak egymásról. Ugyanakkor ha egy dinamikus objektum statikus objektummal ütközik, csak a dinamikus objektum pattan le a statikusról. A statikus objektumok tehát részt vesznek a fizikai számításokban, csak olyan, mintha végtelen tömegűek, így elmozdíthatatlanok lennének. A harmadik csoport, a kinematikus a statikusra hasonlít abban a tekintetben, hogy végtelen tömegűnek tekinthető, azonban a helyzete nem változatlan. A statikus és kinematikus testek megkülönböztetése hatékonysági okokból szükséges. A kinematikus objektumokat nem a fizikai rendszer irányítja, hanem a felhasználó, tipikusan valamilyen animáció által. A fizikai világban azonban hatásuk van, végtelen tömegű mozgó testként minden útjukba kerülő dinamikus objektumot fellökhetnek. Kinematikus objektum lehet például egy biliárdjátékban a dákó, amelynek mozgását a golyóval való ütközés nem

befolyásolja, végtelen tömegű testként teljes kinematikus energiáját átadja a meglökött golyónak. A biliárdgolyók dinamikus testek, a biliárdasztal pedig statikus fizikai objektum.

A 4.18. ábra egy fizikai szimulációt mutat be, ahol egy kockát ejtünk le egy lépcsőzetes geometriára. A kocka útját szaggatott vonallal berajzoltuk, és néhány képkockában kirajzoltuk a kockát is, hogy lássuk az elfordulásának változását is. Ennél a szimulációnál a kocka dinamikus objektum, míg a lépcsők és a talaj statikus objektumok voltak. Minden geometriát téglatestekkel közelítettünk.



4.18. ábra. Fizikai szimuláció

Forrás: a szerző saját készítésű ábrája

A dinamikus objektumok fizikai tulajdonságai közül megemlíthetjük még a pattogás mértékét (bounciness), ami azt adja meg, hogy az ütközés után az objektum energiájának hány százalékát veszíti el. Ha nem veszít, akkor a végtelenségig pattog, mindig ugyanolyan magasra visszapattan, mint ahonnan indítottuk. Egy gumilabda például az energiájának jó részét megtartja. Ha minden energiát elveszítünk, akkor gyakorlatilag nincs visszapattanás ütközés után. A medicinlabda vagy egy szivacs az energiájának nagy részét elveszíti. A fizikai rendszerek képesek még a súrlódás figyelembevételére is, illetve megadhatunk egy csillapítási faktort is az objektumainkhoz, aminek hatására a test folyamatosan veszít a sebességéből (ez lehet haladási vagy forgási sebesség is). A csillapítással meg tudjuk akadályozni, hogy az objektumaink a végtelenségig guruljanak, vagy szimulálhatunk közegellenállást is.

5. 3D KÉPSZINTÉZIS

A fotorealisztikus képalkotás során az elsődleges célunk, hogy az előállított kép a szemlélőben a valódi világ benyomását keltse. Ennek érdekében a virtuális világunk modelljeit (3.1 fejezet) megfelelő részletességgel kell megalkotnunk, illetve az anyagjellemzőiket (3.2 fejezet) is az elérni kívánt minőségnek megfelelően kell megválasztanunk. Amennyiben nem csak egy állóképet szeretnénk látni, akkor a virtuális világ megalkotásakor a jelenetekben (színtérben) mozgásokat is meg kell komponálnunk (4. fejezet), mind a jelenetben részt vevő szereplők, mind a jelenetet felvevő kamera számára.

Amennyiben a virtuális világunk felépítése megfelelő, akkor a következő lépés a jelenethez tartozó képkockák kiszámítása, amely során a valódi fizikai világban lejátszódó fényterjedési folyamatokat kell lemásolnunk. A fény-felület kölcsönhatás leírására szolgáló úgynevezett árnyalási egyenlet [Kayija 1986] nagyon összetett, így valamilyen közelítő módszert kell alkalmaznunk. A gyakorlatban alkalmazható módszerek a fizikai hatások bizonyos részét elhanyagolják, ami természetesen rontja a valósághűséget, így a megcélzott felhasználási területnek megfelelő kompromisszumot kell kötnünk.

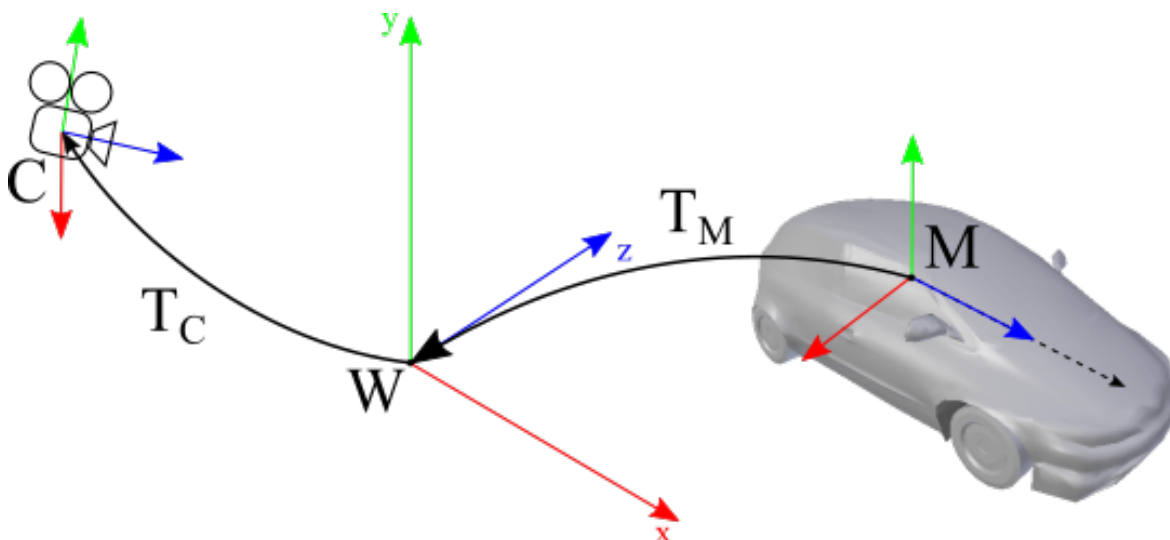
5.1 Inkrementális képalkotás

Az interaktív rendszerek esetében a képelőállítás ideje legalább olyan fontos, mint a képek valósághűsége. Mivel az emberi szem már másodpercenként 25 képkocka esetén is folyamatosnak látja a mozgást, ezért minimálisan ezt a képelőállítási sebességet el kell érniünk, azaz egy-egy képkocka kiszámítására legfeljebb 40 ms idő jut. Ezen időkorlát mellett a gyakorlatban valamilyen inkrementális képszámító algoritmust szokás felhasználni, amely a megfelelő sebesség elérése érdekében nagyon sok egyszerűsítéssel él.

Az inkrementális képalkotás alapvető célja az elvégzett számítások lehető legkisebbre szorítása, így az egyes lépések során elvégzett műveleteket az elemi képpontok helyett mindig a lehető legnagyobb egységeken végezzük. Ez az egység a háromszög, amelyet csúcspontjaival és azok tulajdonságaival írunk le. A virtuális világunk modelljét ennek megfelelően háromszögekből kell felépítenünk, ami a gyakorlatban nem jelent túlzott megkötést. Fontos, hogy a virtuális világunknak csak azt a részét jelenítsük meg, amely látható a kamera nézőpontjából, ezzel is csökkentve a felesleges számítási műveleteket. Ennek megfelelően a biztosan nem látható háromszögeket a lehető leghamarabb el kell dobni, amit az inkrementális képalkotás során vágásnak nevezünk. Végül fontos, hogy a képelőállítás során szükséges műveleteket mindig abban a koordináta-rendszerben végezzük el, amelyben az a lehető legegyszerűbben számolható.

Ahogy azt a 3.3.1 fejezetben láttuk, a virtuális világunkat felépítő modellek általában a saját modellezési koordináta-rendszerükben vannak leírva, így az első lépés a kép előállítása során az objektumok elhelyezése egy közös koordináta-rendszerben, amit a modelltranszformáció alkalmazásával tehetünk meg. Annak érdekében, hogy könnyen el lehessen dönteni egy-egy háromszögről, hogy az látható-e a kamera nézőpontjából, a világkoordináta-rendszert átranzformáljuk egy olyan koordináta-rendszerbe, amelynek origója a kamera pozíciója, és amelyben a kamera a tengely irányába néz. Ezt a koordináta-rendszert kamerakoordináta-rendszernek nevezzük. Ezt az 5.1. ábra

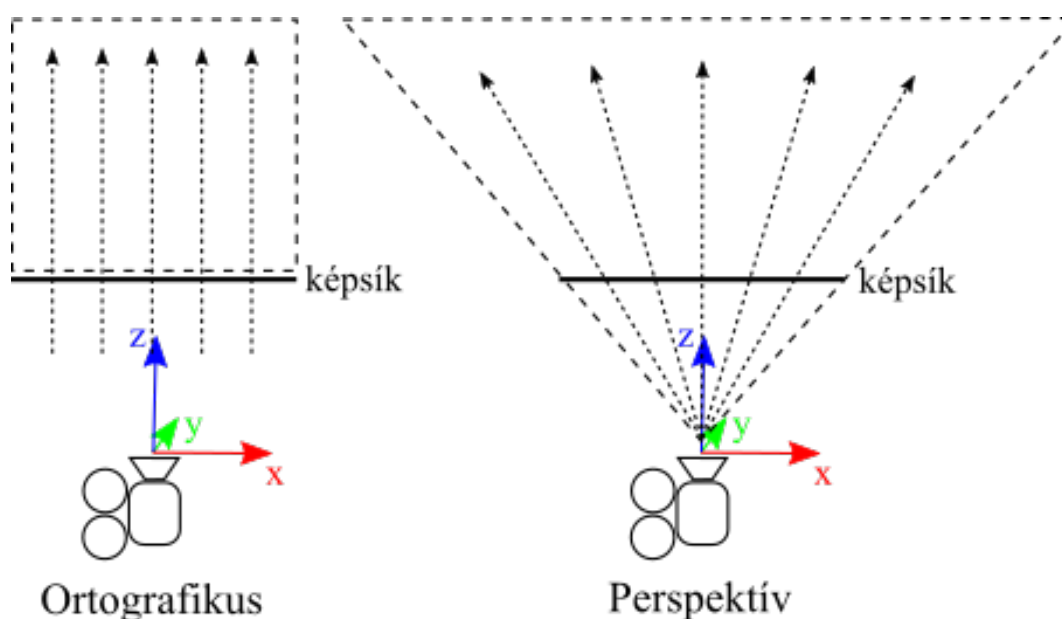
mutatja be, ahol a modell koordináta-rendszerét jelöli, a világkoordináta-rendszert, a kamera koordináta-rendszerét pedig C . A modell csúcspontjait tehát először segítségével világkoordinátákba, majd onnan T_M transzformáció segítségével kamerakoordináta-rendszerbe alakítjuk.



5.1. ábra. Kameratranszformáció

Forrás: a szerző saját készítésű ábrája

Az utolsó transzformációs lépés a vetítés, amely során azt határozzuk meg, hogy a kijelző téglalapjára hogyan kell transzformálni a koordinátákat. A 3D alkalmazásunk ablakának a koordináta-rendszerét képernyőkoordináta-rendszernek nevezzük. Mivel ez kétdimenziós, ezért valahogy el kell hagynunk egy koordinátát a 3D koordinátákból. Ezt nevezzük vetítésnek, és gyakorlatilag a koordináta elhagyását jelenti, a vetítési transzformáció pedig előkészíti a koordinátáinkat a koordináta elhagyására. Alapvetően két vetítési módot szoktunk használni, melyeket az 5.2. ábra mutat be. Ortografikus (vagy párhuzamos) vetítés során a kamerakoordináta-rendszerben meghatározunk egy fő tengelyekkel párhuzamos oldalú befoglaló dobozt, ennek sarkai fogják meghatározni a képernyő sarkait, amelyre rajzolunk. A befoglaló dobozon kívül eső felületelemek nem kerülnek kirajzolásra. Az ábrán szaggatott nyilakkal jelöltük az egy képponthoz tartozó felületelemeket. A perspektív vetítés során meghatározzuk a képsíkot, amelyre rajzolni szeretnénk (ami mindig a kamera síkjával párhuzamos), és a képernyősík sarokpontjait összekötve a kamerakoordináta-rendszer origójával egy gúlát kapunk. Ezt hívjuk nézeti gúlának, és csak azokat a felületi elemeket rajzoljuk ki, amelyek a gúlán belül esnek. Itt is szaggatott nyilakkal jelöltük az egy képponthoz tartozó felületelemeket, melyeken látszik, hogy a kamera origójában találkoznak. A perspektív vetítés feladata, hogy a teret úgy torzítsa, hogy ezek a sugarak párhuzamossá váljanak, ami után a koordináta már eldobható.



5.2. ábra. Vetítési transzformációk
 Forrás: a szerző saját készítésű ábrája

Az így módosított térben a kamera látóterére vágás már egyszerűen elvégezhető. A vágás során azon háromszögeket, amelyek minden csúcspontja kívül esik a kamera látóterén, eldobjuk. Azon háromszögeket, amelyeknek bármilyen kicsi része is a látóterén belülre esik, felbontjuk a látható és nem látható darabjaira úgy, hogy a látható részek továbbra is háromszögeket alkossanak. Az így megszürt háromszöghalmazzt ezután rávetítjük a képsíkra, és meghatározzuk, hogy potenciálisan mely képpontokban jelenhetnek meg a végső képen. Ezt a lépést rasterizálásnak nevezzük, amely során a háromszögeket felbontjuk fragmensekre úgy, hogy minden olyan képpontba kerüljön fragmens, amelyben a háromszög potenciálisan látható. A fragmensek előállításánál a háromszög csúcspontjaihoz rendelt tulajdonságokat az egyes fragmensek csúcspontoktól való távolsága alapján súlyozva interpoláljuk. Az egyes fragmensekhez ezek után már minden olyan információ a rendelkezésünkre áll, amely ahhoz szükséges, hogy a színüket meghatározzuk (3.2 fejezet). A fragmensek színének meghatározása után már csak annak az eldöntése következik, hogy az azonos képpontra eső fragmensek közül végül melyik is látszik. Ehhez el kell döntenünk, hogy mely fragmens volt a legközelebb a kamerához a virtuális világban. A döntéshez általánosan a z-buffer algoritmust alkalmazzuk. Az összes fragmens feldolgozása után a kép elkészült, és megjelenítésre kész.

Az inkrementális képszintézis során a színteret felépítő háromszögeket önállóan dolgozzuk fel, így a fényterjedés számítása során a virtuális világot felépítő objektumok között fellépő hatásokat csak igen korlátozottan tudjuk figyelembe venni. Olyan viszonylag egyszerű jelenségek megjelenítéséhez, mint az objektumok árnyéka, illetve a tükör vagy fénytörő jellegű anyagok alkalmazása, külön erőfeszítéseket kell tennünk. Amennyiben a képelőállítás nem kell valós időben elvégeznünk, akkor alkalmazhatunk olyan algoritmusokat is, amelyek ezeket a fényterjedési jelenségeket és még sokkal többet is képesek természetes módon kezelni.

5.2 Sugárkövetés

A sugárkövetés algoritmus [Glassner 1989] megpróbálja a fény terjedését szimulálni úgy, hogy a fénysugarak és a felületek ütközését figyelembe vegye. A kép előállításánál minden egyes képpontra különállóan oldjuk meg a takarási és árnyalási feladatot, hasonlóan az inkrementális képkalkotáshoz, azonban nemcsak a kamerához legközelebbi felületi pontokat vesszük figyelembe a számítások során, hanem a fénysugár útját is közelítjük a színtérben. Ez a közelítés a fény-felület találkozásánál a lehetséges visszaverődések és törések közül a geometriai optikának megfelelő ideális esetekben képes a többszörös visszaverődés és törés követésére. Bár a többi úgynevezett inkohérens esetet elhanyagoljuk, az alkalmazott közelítés kevésbé durván egyszerűsíti az árnyalási egyenletet.

A képelőállítás folyamata az inkrementális algoritmushoz hasonlóan a kiszámítandó kép minden egyes képpontjához hozzárendel egy sugarat, amely mentén meghatározza a legközelebbi felületi pontot. Ebben a pontban kiértékeljük a megvilágítást úgy, hogy a felületi pont és a nézeti irány alapján meghatározzuk a felület esetleg saját sugárzását, illetve az ambiens megvilágítás hatását. Ezek után meghatározzuk a tükör- és törési irányból érkező sugársűrűséget, amelyet a felületi tulajdonságoknak megfelelően vesszük figyelembe. A két speciális irányban végzett számítások nagyon hasonlóak a nézeti sugár mentén végzett számításokhoz, azzal a különbséggel, hogy a sugár irányát nem a kamera és a képpont közepe, hanem a felületi pont és a tükör-, illetve törőirány határozza meg. Az többi irányból a felületi pontba érkező fénysűrűség meghatározását az algoritmus úgy egyszerűsíti, hogy a színtérben található minden egyes fényforrásra meghatároz egy láthatósági értéket, amely felhasználásával a képen árnyékok is megjelenhetnek. A láthatósági érték meghatározásához a felületi pontból minden egyes fényforrás felé elindítunk egy úgynevezett árnyéksugarat, és a sugár mentén megvizsgáljuk, hogy van-e metszés valamely színtérbeli objektummal a fényforrásnál közelebbi pontban. Amennyiben nincs metszéspont, akkor a fényforrás látható a felületi pontból, így annak kibocsátott sugársűrűségét teljes egészében figyelembe vesszük. Amennyiben a felületi pont és a fényforrás között valamely objektum helyezkedik el, akkor annak a fénytörési együtthatójával súlyozzuk a fényforrás felől érkező sugársűrűséget. Természetesen az árnyéksugár mentén is figyelembe vehetnénk a fénytörési jelenségeket, azonban ennek számítása lényegesen bonyolultabb lenne, így a gyakorlatban elhanyagoljuk ezt.

5.3 Globális illumináció

A korábban bemutatott két algoritmus az árnyalási egyenlet erősen egyszerűsített közelítését alkalmazta a képelőállítás során. Az egyszerűsítések miatt nem vettük figyelembe, hogy a fény akár többszörös visszaverődés után is kamerába juthat, így egy sor összetett fényterjedési jelenséget nem tudunk reprodukálni. A globális illuminációs algoritmusok a képelőállítás sebessége helyett a lehető legvalóságosabb végeredményre helyezik a hangsúlyt, ennek megfelelően a céljuk az árnyalási egyenlet minél pontosabb kiértékelése. A számítások bonyolultsága miatt ezek az algoritmusok valós idejű képkalkotásra nem alkalmasak, így ismertetésüktől most eltekintünk.

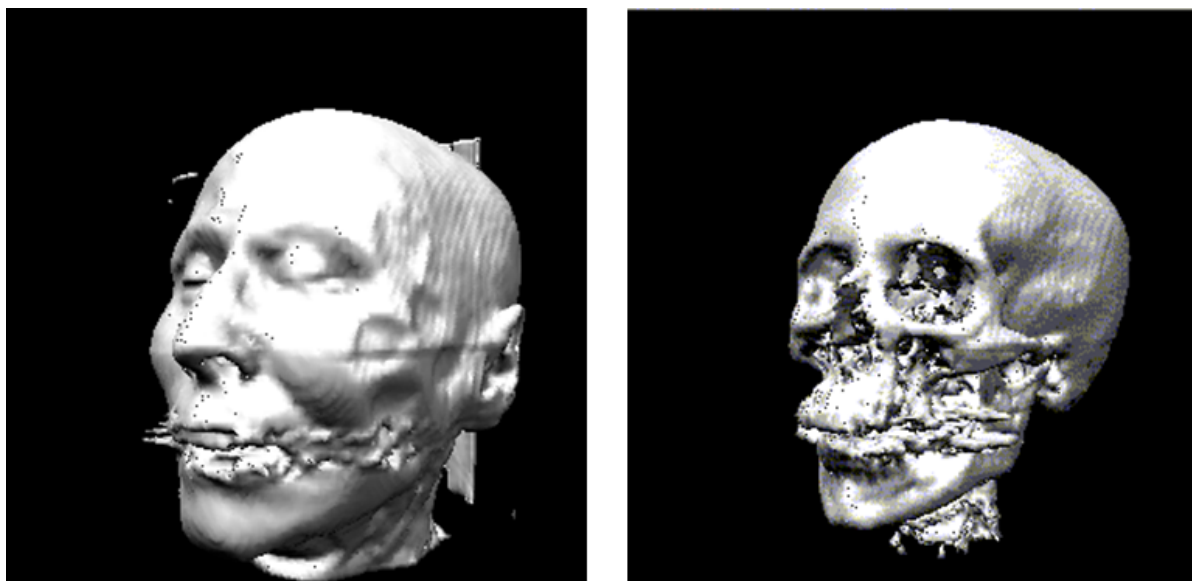
5.4 Tértfogati modellek megjelenítése

A térfogati modellekről már szót ejtettünk a 3.1.4 fejezetben. A térfogati modellek közös jellemzője, hogy a reprezentálandó térrész valamely tulajdonságának folytonos modelljét valamilyen finomságú mintavételezett értékekkel közelítik. Ennek megfelelően a modell a sima geometriai közelítésekhez képest jelentős többletinformációt tartalmaz, megjelenítése során lehetőség nyílik az modellen belüli viszonyok feltérképezésére. Ebben a fejezetben röviden áttekintjük a térfogati adatok megjelenítésére szolgáló alapvető módszereket. Az egyszerűség kedvéért a továbbiakban feltételezzük, hogy a térfogati modell sűrűség jellegű értékeket tartalmaz, de a bemutatott algoritmusok tetszőleges tulajdonságok esetén alkalmazhatók.

A térfogati modellek kiértékelésénél alapvetően azonos értékű régiókat keresünk, illetve ezen régiók térbeli eloszlására és egymáshoz képesti helyzetére vagyunk kíváncsiak. Az azonos értékű régiókat izofelületnek nevezzük, és ugyanúgy meghatározhatjuk az alapvető tulajdonságaikat, mint a geometriai modellek esetén. A térfogati modell minden pontjában meghatározhatjuk az azon a ponton átmenő izofelület normál vektorát, illetve a felület görbületét is.

A térfogati modellek megjelenítésére szolgáló algoritmusok két nagy csoportba sorolhatók aszerint, hogy a diszkrét adatot közvetlenül, úgynevezett direkt módszerekkel vagy valamilyen átmeneti reprezentációt alkalmazva indirekt módszerekkel dolgozzuk fel.

Az indirekt módszerek közül a leggyakrabban alkalmazott az úgynevezett masírozó kockák algoritmus [Lorensen 1987], amely segítségével a térfogati modell egy szintfelületéhez képezhetünk közelítő háromszöghálós reprezentációt (5.3. ábra). Ezek után az előállított geometriai modellt tetszőleges háromszögalapú geometriát kezelni képes algoritmussal jeleníthetjük meg. A geometriai modell további előnyös tulajdonsága, hogy a legtöbb modellező eszközben tovább tudjuk finomítani, illetve felhasználhatjuk véges elemes mechanikai szimulációkban is. Az algoritmus hátránya, hogy általában nagyszámú háromszöget hoz létre, így gyakran szükségessé válik a geometriai modell további feldolgozása a minőség javítása érdekében. Ilyen utófeldolgozási lépés a viszonylag kis görbületű területeken a háromszögek összevonása, illetve a nagy görbületű részeken valamilyen simítás és újrafelosztás alkalmazása. A háromszöghálós megközelítés szakaszonként lineárisan közelíti a felületet, így az utófeldolgozással együtt is előfordulhat, hogy a végső modellen zavaró élek jelennek meg. Az előzőekben említett hátrányos tulajdonságok miatt a masírozó kockák módszerét csak olyan esetben érdemes alkalmazni, ha mindenképpen geometriai modell előállítására van szükség.



5.3. ábra. A masírozó kockák algoritmussal létrehozott geometriai modellek
 Forrás: a szerző saját készítésű ábrája

A direkt térfogatvizualizációs módszerek esetén közvetlenül a diszkrét térfogati adatot értékeljük ki és jelenítjük meg [Csébfalvi 2001]. Természetesen az indirekt módszerekhez hasonlóan lehetőségünk van egy választott értékű szintfelület megjelenítésére, az úgynevezett isosurface raycasting módszer segítségével. Ez a módszer a megjelenítendő kép minden egyes pixeléhez meghatároz egy nézeti sugarat, és a sugár mentén megkeresi azt a pontot a térfogati modellben, amely pontban a térfogati modell értéke már meghaladja a választott küszöbértéket (5.4. ábra).

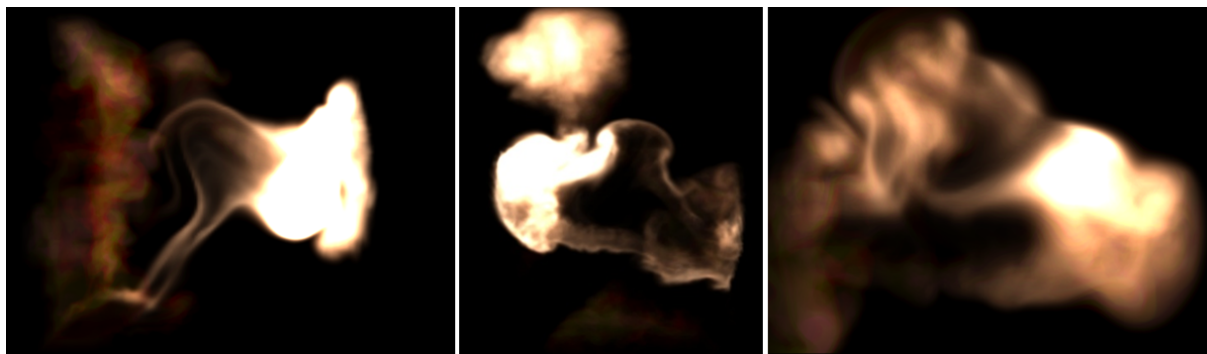


5.4. ábra. Direkt szintfelület-megjelenítés
 Forrás: a szerző saját készítésű ábrája

A szintfelületek mellett vizsgálhatjuk a térfogati modellben rögzített tulajdonság térbeli eloszlását is. Ehhez egy olyan megjelenítési modellt szükséges alkalmaznunk, amely átlátszó közegként képes kezelni a térfogati adatokat úgy, hogy minden sűrűségértékhez hozzárendel egy átlátszósági értéket is. Ez a hozzárendelés általában egy egyszerű leképező függvénnyel történik, amely a modellben található sűrűségértékek értelmezési tartományát a zárt tartományra képezi le. Az így definiált átlátszósági érték megadja azt, hogy az adott térrészen áthaladva mekkora része nyelődik el a belépő fénynek.

A kamerából egy pixelen áthaladó sugár mentén az átlátszósági információt felhasználva meghatározhatjuk, hogy milyen mélyen látunk bele a térfogatba, és onnan milyen sugársűrűség jut a kamerába. A sugár mentén a kamera irányából elindulva minden

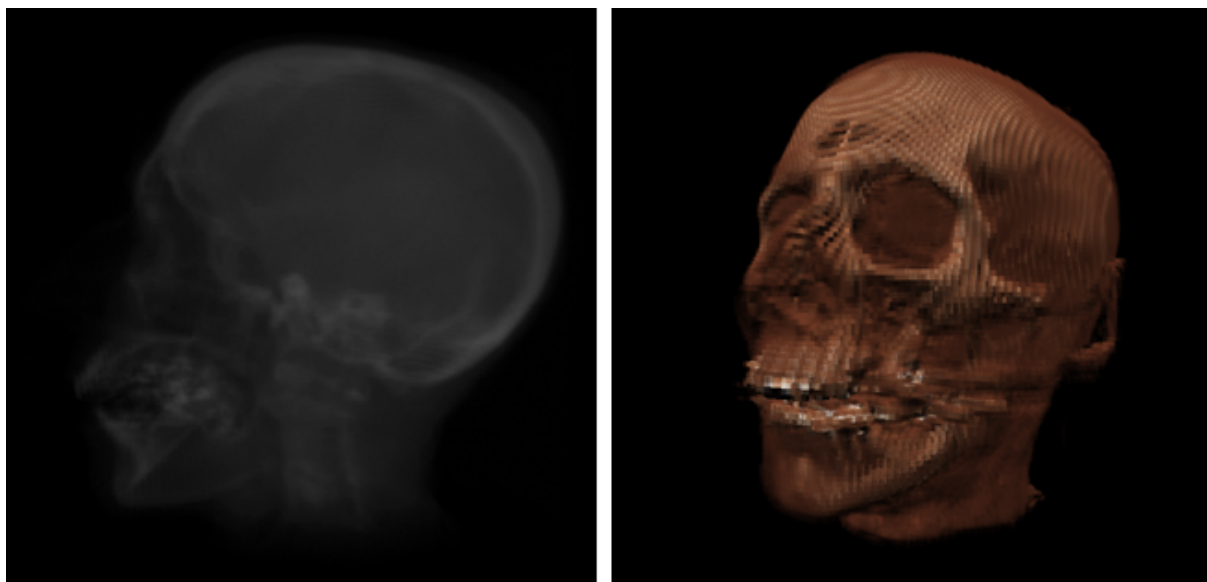
egy lépés során kiértékeljük az adott pont sűrűségét és a hozzá tartozó átlátszóságot. Ezek után az átlátszósági faktorról és a megtett lépés hosszával arányosan csökkentjük az adott térrész hozzájárulását a sugár mentén mért sugársűrűséghez. Ezt a módszert láthatósági sugárkövetésnek nevezzük, és általában fényelnyelő anyagok, például köd és füst megjelenítésére alkalmazzuk (5.5. ábra).



5.5. ábra. Láthatóság sugárkövetéssel megjelenített modell

Forrás: a szerző saját készítésű ábrája

A láthatósági sugárkövetéssel ellentétes megközelítés a fénysugárkövetés módszere, amely során egy adott pixelen áthaladó sugár mentén a térfogati modell legtávolabbi pontjából indulunk, és az egyes lépések során a kamera felé haladunk. Amennyiben csak a sugár mentén megtett út alatti elnyelődést összegezzük, akkor a térfogati modell egy röntgenképszerű vetületét kapjuk (5.6. ábra). Természetesen a mintapontokban közelíthetjük a környezetből visszavert sugársűrűséget is, amelyet hozzáadhatunk az adott szakasz mentén a sugársűrűséghez.



5.6. ábra. Fénysugárkövetéssel megjelenített modell

Forrás: a szerző saját készítésű ábrája

6. A GRAFIKUS HARDVER

Az interaktív grafikus alkalmazások esetén a komfortos használathoz másodpercenként 60 képkocka kiszámítása és megjelenítése szükséges. Az egyes képkockák felbontása (azaz a megjelenítendő képpontok száma) a kijelző paramétereitől függ, de az esetek többségében ez legalább 1920 x 1080 képpont. Ez optimális esetben 124 416 000 képpont kiszámítását jelenti minden egyes másodpercben, azaz egy-egy képpont színének előállítására kevesebb, mint $5\mu\text{s}$ áll rendelkezésre. Jól érzékelhető, hogy a feladat megoldására speciális eszközre van szükségünk, így a valós idejű képelőállítás napjainkra szinte kizárólag az úgynevezett grafikus processzorok (GPU-k) felhasználásával történik.

6.1 A grafikus hardver története

A grafikus megjelenítésre alkalmas eszközök már a 60-as években is léteztek, azonban ezek a korai rendszerek még vektorgrafikus megjelenítésre voltak képesek, azaz egy elektronsugár mozgatásával görbékből építették fel a képet. A mai értelemben vett rastergrafikus rendszerek a 70-es évek közepén jelentek meg, és közös jellemzőjük, hogy a megjelenítendő kép szabályos négyzetrácsba szervezett képpontokból áll össze. Ez a módszer nagyobb szabadságot biztosított a részletgazdag képek megjelenítésében, így csakhamar teljesen kiszorította a vektorgrafikus rendszereket.

Az első rastergrafikus eszközök tipikusan a 2D megjelenítés hatékonyságának javítására törekedtek, így a korai ablakozó rendszerek igényeinek megfelelően a vonalak, négyszögek és a grafikus memóriában tárolt képrészletek (sprite-ok) gyors megjelenítésére voltak felkészítve. A 3D megjelenítésre ezeken az eszközökön még korlátozottan volt lehetőség, mivel a teljes képelőállítási folyamatot a központi processzoron kellett elvégezni, és az eredményül előálló képet kellett a grafikus hardver memóriájába másolni.

A 90-es évekre a 2D grafikus gyorsítást nyújtó hardverek már széles körben elterjedtek az otthoni rendszerekben is, és egyre nagyobb igény mutatkozott a 3D grafikus megjelenítés iránt is. Ezt az igényt elsősorban a számítógépes játékok gerjesztették, mivel a játéktérmi gépek és a televízióra köthető játékkonzolok által nyújtott élmény az otthoni számítógépeken ebben az időben megközelíthetetlen volt. A 90-es évek közepére jelentek meg az első otthoni számítógépbe illeszthető grafikus kártyák, amelyek a gyors 2D megjelenítés mellett már képesek voltak korlátozottan a 3D grafikus számítások támogatására is. Ezen hibrid megoldások mellett megjelentek a teljes inkrementális 3D képelőállítást támogató eszközök is, azonban gyártástechnológiai korlátok miatt ezek az eszközök nem támogatták a 2D megjelenítést. Az évtized végére azonban a technológia fejlődése már lehetővé tette a 2D és 3D feldolgozást támogató egységek integrálást, és így létrejöttek az első modern grafikus egységek, amelyek már a rasterizáció folyamata mellett a csúcspontok transzformációját és a csúcspontonkénti megvilágítás számítását is támogatták.

A 2000-es évek elején jelentek meg azon otthoni felhasználásra szánt grafikus egységek, amelyek már lehetővé tették a csúcspont- és a fragmensfeldolgozás során alkalmazott algoritmusok módosítását. Ezekben az eszközökben a csúcspont- és a fragmensszámítást eltérő felépítésű egységek végezték. A csúcspontszámításokra szolgáló

egység nagyobb rugalmasságot nyújtott, azonban lassabban végezte a számításokat, míg a fragmensfeldolgozást végző egységek a csúcspontok számához képest akár nagyságrendekkel több fragmens miatt sokkal szűkebb korlátok között működtek. Ez a kettősség az évtized közepére megszűnt, és a specializált egységek helyét átvették az általános célú számításokat is támogató komponensek. Ezzel megnyílt a lehetőség a korábbiaknál rugalmasabb feldolgozásra, így sorra jelentek meg az újabb programozható feldolgozási lépések a képelőállítás folyamatában.

Napjainkra a grafikus hardver a képelőállítás mellett alkalmassá vált általános célú számításokra is, így egyre gyakrabban találkozhatunk velük a gépi látás, a neurális hálózatok vagy akár az összetett szimulációs feladatok és az orvosi képalkotás [Tóth 2013] területén is.

6.2 Képelőállítás a grafikus hardver segítségével

A grafikus processzor a számítógép fő processzorától (CPU) függetlenül végzi a képek előállítását és megjelenítését a képernyőn. A központi processzor feladata a grafikus hardver működésének vezérlése és annak ellátása a megjelenítendő virtuális világ leírásával, jellemzően egy általános 3D programozói felületen (API-n) keresztül. A legelterjedtebb 3D API-knak az OpenGL, a Direct3D és újabban a Vulkan tekinthetők, amelyek a felhasználási célokban és az alapvető koncepciójukban nagyban hasonlítanak egymásra.

A grafikus hardver a képelőállítás folyamatának szempontjából az inkrementális képszintézis algoritmusát modellezi úgy, hogy a képelőállítás folyamán lehetőséget biztosít az egyes feldolgozási lépések befolyásolására. A szigorúan egymást követő feldolgozási lépésekből összeállított munkafolyamatra csővezeték-ként (pipeline) hivatkozunk. A csővezeték elvárt működésének leírására az úgynevezett leírók (render state) szolgálnak, amelyek egyértelműen leírják a csővezeték viselkedését. A feldolgozandó adatokat a rajzoló hívások (render call) adják meg, amelyek a csővezeték bemenő adathalmazának leírására szolgálnak. A csővezeték kimenete elsősorban a végső kép, amely jellemzően egy színes képpontokat tartalmazó adathalmaz, de emellett kiegészítő információk is megjelenhetnek (például mélységinformáció, objektumazonosítók, illetve tetszőlegesen, a felhasználó által előállított adat stb.).

A megjelenítendő virtuális világot objektumok halmazaként írjuk le, ezek alkotják a csővezeték bemeneti adathalmazát. Az egyes objektumok geometriai felépítését jellemzően a saját modellezési koordináta-rendszerükben írjuk le, amelyet a virtuális világba elhelyezésük során transzformálunk a világkoordináta-rendszerbe. A geometriai felépítés leírására nagyon sok matematikai módszer áll rendelkezésünkre, amelyek közül leggyakrabban a háromszöghálót alkalmazzuk, kiegészítve azt vonal és pont primitívekkel. A primitíveket jellemzően a csúcspontpozíciójukkal, normálvektorukkal, csúcspontszínükkel és textúrakoordinátaikkal írhatjuk le.

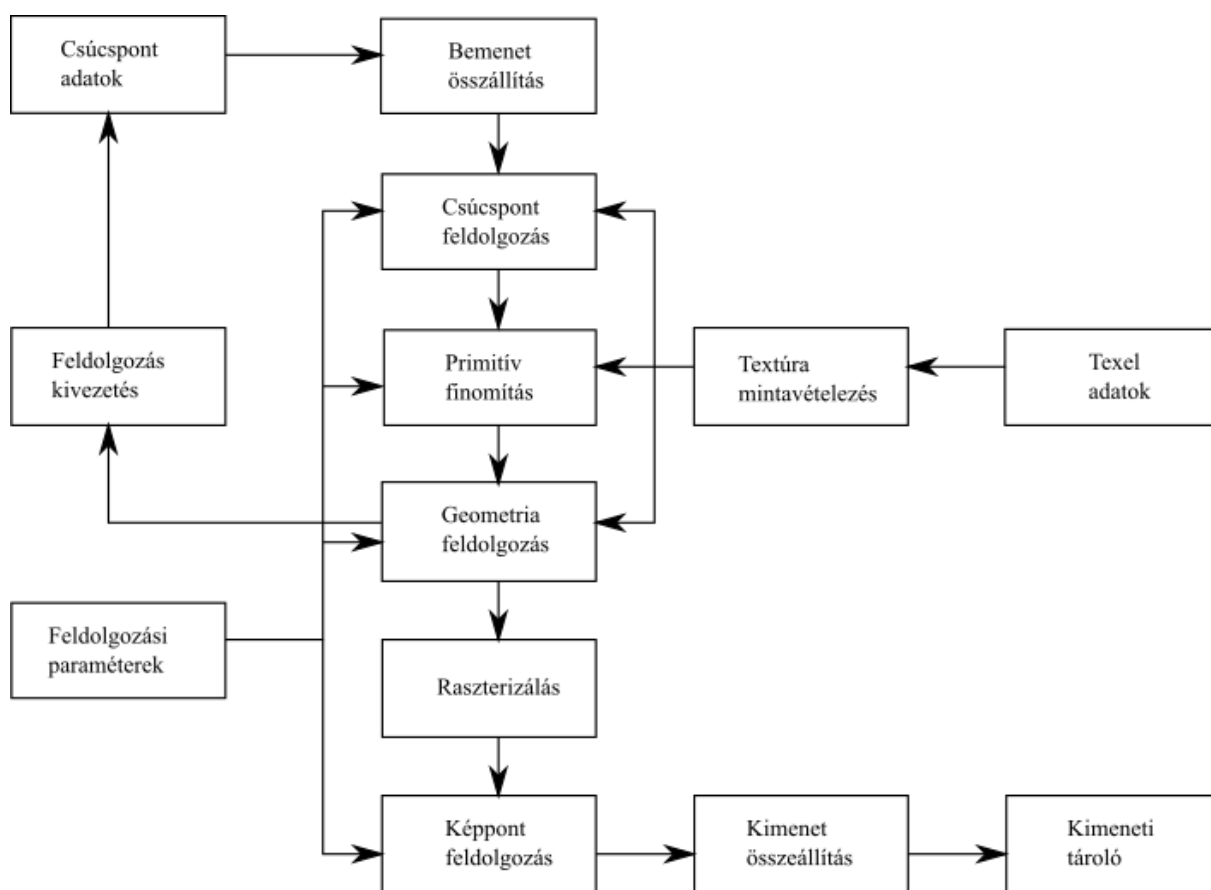
6.2.1 A grafikus csővezeték

A modern grafikus csővezeték adatfolyam-feldolgozó hálózatként működik, amely több mint fél tucat adatfeldolgozási lépcsőt tartalmaz. A feldolgozási lépcsők között található előre rögzített algoritmusokat végrehajtók és szabadon programozhatók is.

A lényeges feldolgozási lépések az alábbiak (6.1. ábra):

- **Bemenet-összeállítás:** Az első lépcső feladata, hogy a felhasználó által megadott adatokat tartalmazó tárolókat olvasva összeállítsa a csővezeték számára értelmezhető primitív leírásokat. Az összeállított primitív leírásokhoz az adott hardverre jellemző kiegészítő információk is hozzárendelődnek (például primitívazonosító, példányazonosító, csúcspont-azonosító), amelyek segítik a későbbi feldolgozási lépések hatékony működését. Az ebben a lépésben feldolgozott primitívek a következő lépésként a csúcspontfeldolgozóban folytatják az útjukat.
- **Csúcspontfeldolgozás:** Ez a feldolgozási lépés az előző lépcsőtől érkező primitívek csúcspontjait dolgozza fel. Jellemzően transzformációs, bőrzési, morfolási vagy akár csúcspontonkénti megvilágítás számolást végez. A csúcspontárnyaló egyszerre egy csúcspontot dolgoz fel, és a kimenete is egyetlen csúcspont lesz. A feldolgozási lépcső alapvető feladata a virtuális világ leírásának feldolgozása úgy, hogy a későbbi megvilágításszámítási lépések számára értelmezhető legyen azáltal, hogy a csúcsponthoz rendelt tulajdonságokat (például pozíciót, normálvektort) a modellezési koordináta-rendszerből a világkoordináta-rendszerbe transzformálja, amelyben a kamera és a fényforrások vannak megadva. A világkoordináta-rendszerben adott primitíveket ezek után a kameratranszformáció segítségével módosítjuk úgy, hogy a csúcspontok paraméterei a kamera pozíciójának és irányultságának megfelelőek legyenek. Majd a perspektív vetítés pedig úgy torzítja a teret, hogy a kamerába érkező vetítésugarak párhuzamosak legyenek, lyukkameraként imitálva a valódi fizikai kamerák működését. Végül a nézeti transzformációt alkalmazva a primitívekre áll elő a végső pozíció, ahova a primitíveket ki kell rajzolni a végső képen.
- **Primitívfinomítás:** A primitívfinomítási lépés lehetővé teszi a bemeneti primitívek hatékony felbontását kisebb egységekre. A feldolgozásfolyamat három lépésből áll, amelyek a finomítás paraméterezésére, a rögzített algoritmusú felbontásra és a finomítás kiértékelésére szolgálnak. Az első lépés a beérkező primitívhalmozat egészíti a finomítást végző hardver számára szolgáló további paraméterekkel. Ezek közül a legfontosabb a felbontási szintek meghatározása, amely azt adja meg, hogy a primitívhalmozat hányszor kell felbontani kisebb primitívekre a feldolgozás során. A finomítás kiértékelése során a létrejött új primitívek utólagos feldolgozására nyílik lehetőség. Ennek során történik az új primitívek összeállítása és azok pozíciójának, illetve egyéb tulajdonságainak számítása.
- **Geometriafeldolgozás:** Ebben a lépésben az eddig feldolgozott csúcspontokból újra összeállítja a csővezeték a kirajzolható primitívet (pontot, törtvonalat, háromszöget), és lehetőséget nyújt a beérkezett primitív alapján egy vagy több kimenő primitívet létrehozni. Az ideérkező primitíveket a feldolgozásuk után a csővezeték eldobja, és az újonnan generált primitívek veszik át a helyüket a további feldolgozás során.

- **Vágás:** A végleges geometria kialakítása és annak a kamerának megfelelő transzformációja után lehetőség nyílik azon primitívek eltávolítására, amelyek a kamera látótérén kívül esnek. Ezt a lépését nevezzük vágásnak, amely során a csővezeték eldobja azokat a primitíveket, amelyek teljes egészében kívül esnek a látótéren, és levágja azon primitíveket a látótérre, amelyeknek egy vagy több csúcsa kívül esik azon.
- **Raszterizálás:** A raszterizálás folyamán az eddig csúcspontjaikkal adott primitíveket felbontja a rendszer úgynevezett fragmensekre, amely az eddigi 3D adatokat képezik le a képen megjeleníthető 2D adatokra. A raszterizáció során megtörténik a 3D primitívek nézeti gúlára vágása és leképzése a képernyőkoordináta-rendszer 2D terébe, amely megadja, hogy az adott fragmensnek hol kell megjelennie a kimeneti tárolóban. A primitívek csúcspontjaiban adott tulajdonságinformációk a raszterizálás során interpolálva rendelkeznek hozzá a fragmensekhez.
- **Képpontfeldolgozás:** A raszterizáló által létrehozott fragmensek minden információval rendelkeznek a fragmens végső színének számításához, így a fragmensek egymástól függetlenül kiértékelhetők. Ezt a kiértékelést végzi a képpontfeldolgozás lépése, amely az árnyalási technikák széles körét teszi megvalósíthatóvá. Ezek közé tartoznak a széles körben alkalmazott fizikai alapú visszaverődés-számítások, de akár bizonyos utófeldolgozási algoritmusok is.
- **Kimenet-összeállítás:** A grafikus csővezeték utolsó lépése gondoskodik a kiszámított fragmensek elhelyezéséről a kimeneti tárban. Ennek során nemcsak a fragmensek színinformációja használható, hanem a fragmensekhez rendelt mélység és különféle külső maszkok is. A rendelkezésre álló információk alapján a kimenet-összeállítás során rekonstruálható a fragmensek egymást takarása (z-buffer), amely eredményeként a 3D térből 2D-be leképzett primitívek megfelelően fogják egymást takarni. Áttetsző primitívek esetén a lehetőség nyílik az áttetszőségi információ alapján a fragmensek összemosására is úgy, hogy az aktuális fragmenshez rendelt szint kombináljuk a kimeneti tárban található színnel a fragmenshez rendelt áttetszőségi érték és egy előre megadott összemosó függvény alapján.



6.1. ábra. A modern grafikus csővezeték felépítése

Forrás: a szerző saját készítésű ábrája

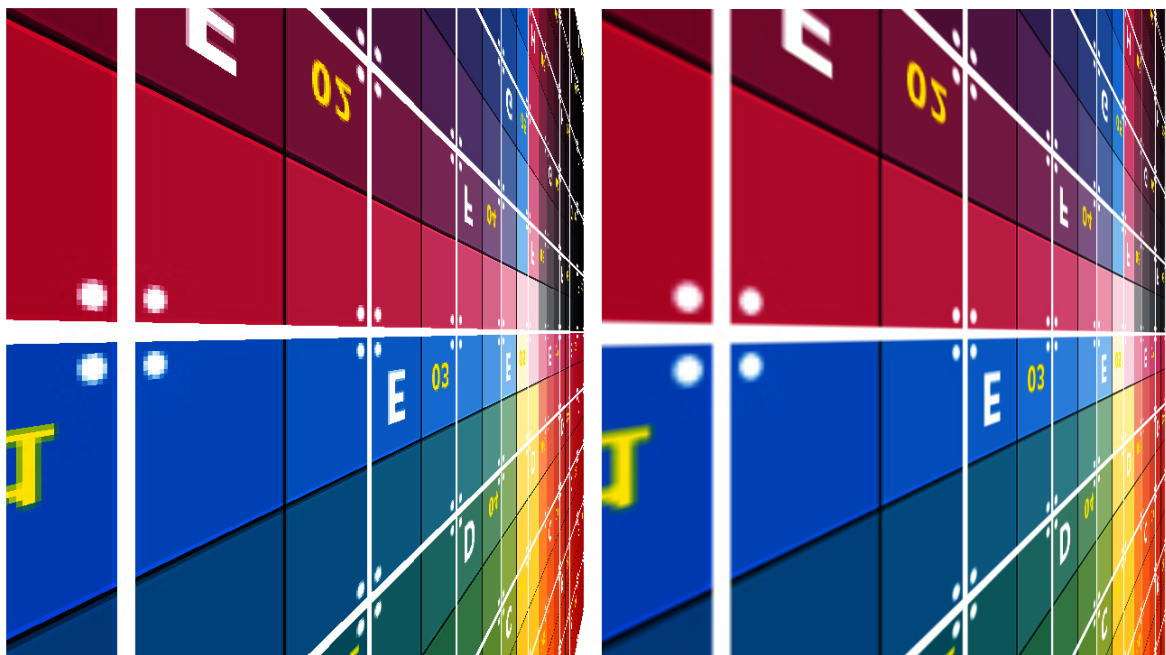
A grafikus csővezeték a grafikus hardver által nyújtott lehetőségeket az egyes feldolgozási lépések különböző mértékű programozhatóvá tételével teszi elérhetővé a felhasználó számára. Az alkalmazandó algoritmusok leírását úgynevezett árnyaló programok segítségével adhatjuk meg. A teljes mértékben programozható feldolgozási lépések a csúcspontfeldolgozás, a primitívfinomítás első és utolsó lépése, a geometriafeldolgozás és a képpontfeldolgozás. A többi feldolgozási lépésben általában a paraméterek finomhangolására nyílik lehetőségünk. Az árnyaló programokat jellemzően magas szintű programozási nyelvek segítségével készíthetjük, amelyek lényegesen kibővítik a 3D API által nyújtott lehetőségeket. A legelterjedtebb árnyaló nyelvek a GLSL, az OpenGL, illetve a HLSL a Direct3D esetén. Az árnyaló nyelvek támogatják a 3D számítások esetén gyakran alkalmazott matematikai függvényeket, illetve a textúrák mintavételezésének különböző eseteit.

6.2.2 Textúrázás

A grafikus hardver támogatja a textúrák alkalmazását is a virtuális objektumok megjelenítése során. Tekintve, hogy a textúrák helyettesítik az objektumok anyagjellemzőit és finom részleteit, ezért az előállított kép minőségében kulcsszerepet játszanak.

A virtuális objektumokat felépítő primitívek csúcspontjaihoz textúratérbeli koordinátákat rendelhetünk, amelyek azt határozzák meg, hogy a textúrában tárolt információ milyen módon képződik le a primitív csúcspontjai közötti síkra. A raszterizálás során a textúrákoordinátákat is interpolálja a rendszer, így folyamatos átmenetet kapunk a textúratérben is. Fontos megemlíteni, hogy a textúrákoordináták interpolációja során

a grafikus hardver automatikusan figyelembe veszi a perspektív transzformációból fakadó torzulásokat, és az interpolált értékeket ennek megfelelően számítja ki. Az interpolált textúratérbeli koordináták nem feltétlenül esnek a textúra pixeleinek (texel) középpontjaira, ezért a grafikus hardver lehetőséget biztosít a texelek szűrt olvasására úgy, hogy a mintapontok közé új értékeket vesz fel a legközelebbi texelek értéke alapján, illetve kiolvasható a textúra úgy is, hogy mindig a legközelebbi eső texelben tárolt értéket kapjuk vissza (6.2. ábra).

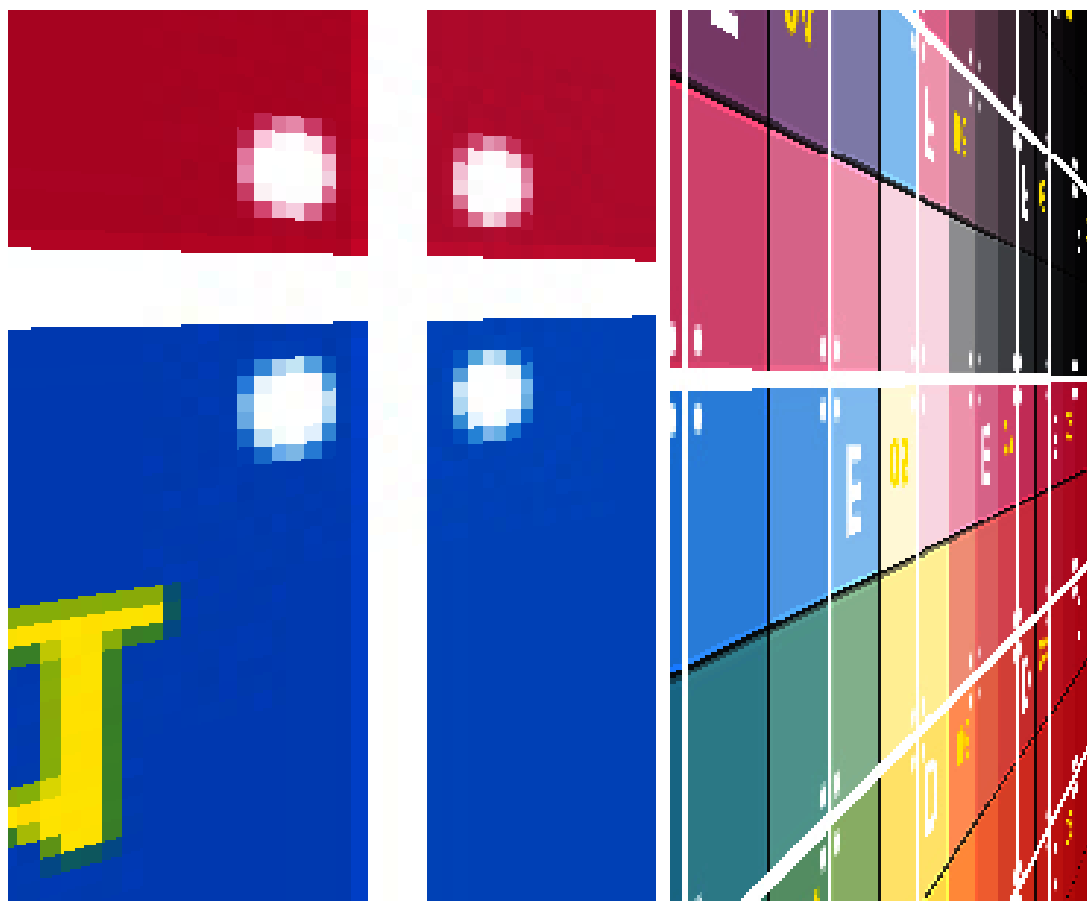


6.2. ábra. Legközelebbi texelközépponttal vagy lineáris szűréssel mintavételezett textúra

Forrás: a szerző saját készítésű ábrája

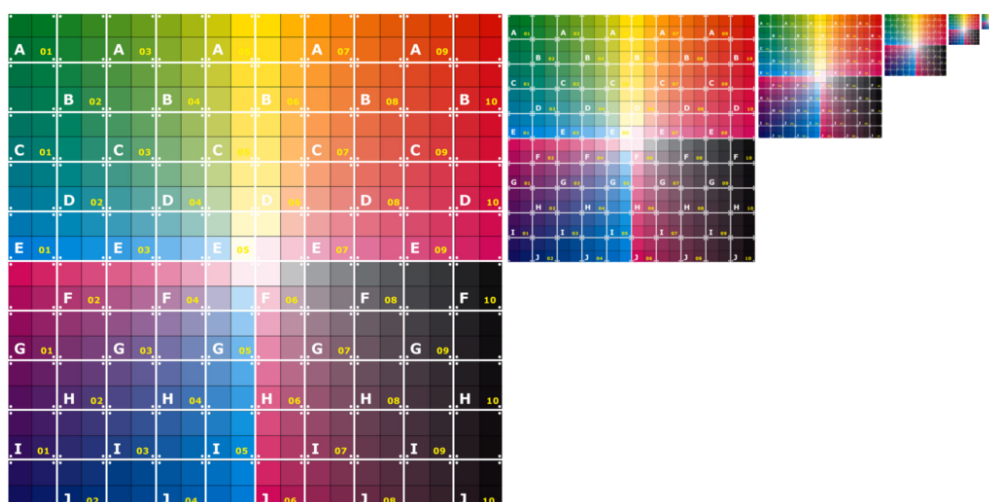
Fontos kitérni arra is, hogy a térbeli primitívek képsíkra történő leképzése során a legjobb mintavételezési minőséget akkor kapjuk, ha a képsík két pixele közötti távolság közel azonos a két pixelben interpolált textúrákoordináták különbségével. Ez azonban a legtöbb esetben nem áll fenn, így az alul- illetve túl-mintavételezésből fakadó hibák keletkezhetnek (6.3. ábra). Az alul-mintavételezés esetén a textúrában tárolt információt hiányosan kapjuk meg, ami a részletek eltűnését eredményezi, ezzel szemben túl-mintavételezés esetén több képponthoz is azonos texel információja rendelődik, így zavaró blokkosodás jön létre. Ezt a blokkosodást figyelhetjük meg az 5.3. ábra bal oldalán látható képen a körök szélein. A jobb oldali képen pedig az alul-mintavételezés miatt a felület jobb oldala felé haladva egyre nehezebben ismerhetők fel a textúra mintái.

A mintavételezési hibák csökkentése érdekében a grafikus hardver többféle technikát is a rendelkezésünkre bocsájt, amelyek együttes alkalmazásával lényegesen javítható a megjelenített kép minősége. Az egyik ilyen lehetőség az úgynevezett mip-mapek használata, amely az egyes textúrákhoz több részletességi szintet tud rendelni. A textúra mintavételezése során a rendszer azt a részletességi szintet választja ki, amelyben a texelek közötti távolság legközelebb van a képsíkra leképzett textúrákoordináták közötti távolsághoz.



6.3. ábra. A nagyításból és kicsinyítésből fakadó hibák
 Forrás: a szerző saját készítésű ábrája

Ezzel a mintavételezésből fakadó hibák minimálisra csökkenthetők. A mip-map egyes szintjeinek felbontása piramisszerűen csökken (6.4. ábra), és az egyes szintekhez tartozó texelértékeket a rendszer képes automatikusan generálni egy átlagoló szűrés segítségével. Természetesen lehetőségünk van a textúrapiramis egyes szintjeihez saját texelértékeket is rendelni, amely segítségével még pontosabban tudjuk megadni azokat a részleteket, amelyeket mindenképpen meg szeretnénk őrizni.

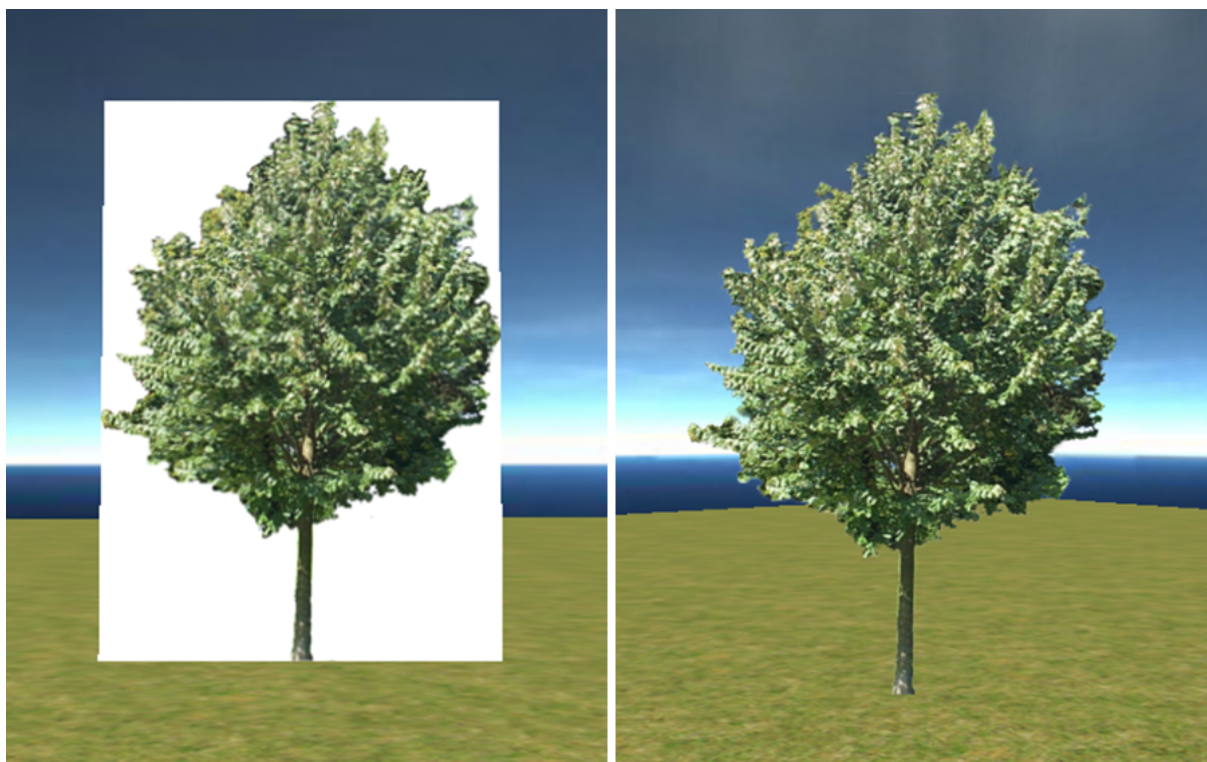


6.4. ábra. A textúrapiramis
 Forrás: a szerző saját készítésű ábrája

6.2.3 Blendingművelet

Az inkrementális képelőállítás során lehetőségünk van áttetsző anyagokat is modellezni és megjeleníteni. Szokásos módon az áttetszőséget egy új színek komponens bevezetésével adhatjuk meg, amely a vörös, zöld és kék színcsatorna mellett az úgynevezett alfa (átlátszósági) csatornában jelenik meg. Az áttetsző objektumok megjelenítéséhez a grafikus hardver blending egysége nyújt támogatást. A blending során az átlátszó objektumokhoz tartozó fragmensek értékei nem írják felül a kimeneti tárolót, hanem a beírandó értéknek és a kimeneti tár aktuális tartalmának valamilyen súlyozott összege kerül kiírásra. A súlyozott érték kiszámításához alkalmazandó képlet általában három komponensből áll össze. Az első a súlyozást végző alapegyenlet, amelyet általában előre definiált függvényekből választhatunk ki. Az elterjedt függvények az összeadás, kivonás, minimum és maximum számítása, de egyes grafikus hardverek további függvényeket is felkínálhatnak. A másik két komponens, amelyet a számításhoz felhasználhatunk, az új érték és a kimeneti tárban szereplő értékek súlya az egyenletben.

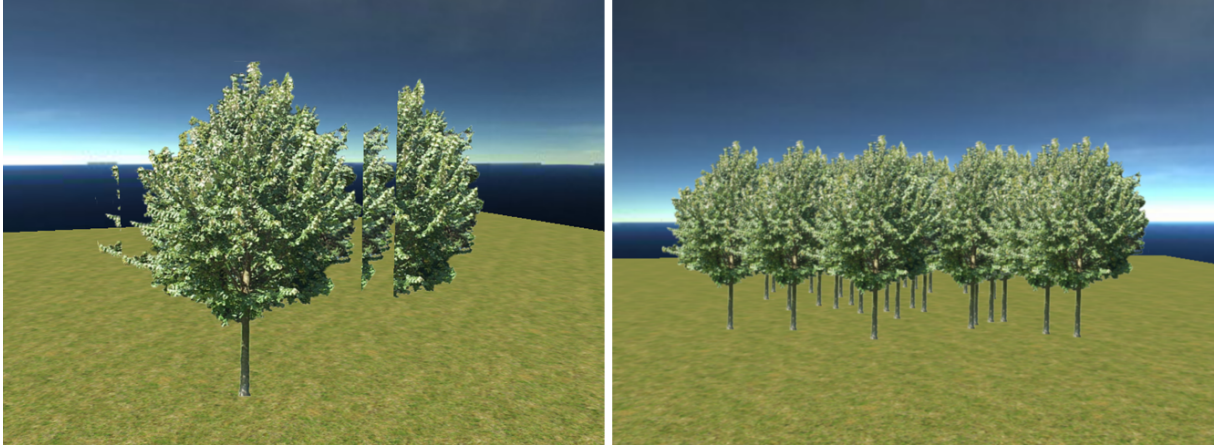
A blendingműveletet az áttetsző objektumok megjelenítése mellett alkalmazhatjuk a virtuális világunk geometriai felépítésének egyszerűsítésére is. Szokásos megoldás olyan objektumok esetén, ahol a geometriai felépítés nagyon összetett, de valójában a teljes formát ismerjük fel, és a részletek önmagukban érdektelenek, hogy csak az objektum képét helyezzük el egy nagyon egyszerűsített modellen. A 6.5. ábra egy fa egyszerűsített modellje látható úgy, hogy a fa képét egy olyan négyszögre illesztettük rá, amelyet úgy helyezünk el a színtérben, hogy minden esetben a kamera nézeti irányára merőleges legyen. A fa körüli fehér képpontok átlátszóságát megnövelve a fa képe beolvad a környezetébe.



6.5. ábra. Blending nélkül és blendinggel illesztett képek

Forrás: a szerző saját készítésű ábrája

Fontos megemlíteni, hogy a súlyozásnál alkalmazott függvényekben a tagok nem felcserélhetők, így a helyes eredmény eléréséhez szükséges a fragmensek sorrendjének helyes megválasztása is. Először mindig az átlátszatlan primitiveket kell feldolgoznunk, majd az áttetszőket a kamerától való távolságuk alapján csökkenő sorrendbe rendezve. A hibás sorrendezésből fakadó hibák jól megfigyelhetők az 6.6. ábra bal oldali képén, amelyen a fák geometriai reprezentációját véletlenszerű sorrendben rajzoltuk ki, míg a jobb oldali képen a képsíktól való távolság alapján rendeztük őket. Ez a rendezés azonban általános esetben nehezen megoldható probléma, ezért erre az alkalmazott modellek kiválasztásánál különös figyelmet kell fordítani.



6.6. ábra. A hibás és helyes sorrendezés eredménye *blending* esetén
Forrás: a szerző saját készítésű ábrája

Az átlátszatlan objektumokhoz tartozó fragmensek esetén ez a probléma nem jelentkezik, mivel a z-buffer algoritmusnak köszönhetően mindig csak a képsíkhöz legközelebb eső fragmens jelenik meg a kimeneti tárbán.

IRODALOMJEGYZÉK

Ábrahám Gy. (1997): *Optika*. Panem Könyvkiadó, Budapest.

Bahaa E. A. Saleh – Teich, Marvin Carl (1991): *Fundamentals of Photonics*. John Wiley & Sons, Inc.

Budó Á. – Mátrai T. (1999): *Kísérleti Fizika III. Optika és Atomfizika*. Budapest, Nemzeti Tankönyvkiadó.

Csébfalvi B. (2001): *PhD thesis. Interactive Volume-Rendering Techniques for Medical Data Visualization*. Technische Universität Wien, Institut für Computergraphik und Algorithmen.

Glassner, A. S. (1989): *An Introduction to Ray Tracing*. London, Academic Press.

Kayija, J. T. (1986): The rendering equation. *Computer Graphics (SIGGRAPH'86 Proceedings)*, 143–150.

Lorensen, E. W. – Cline, E. H. (July 1987): Marching Cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, Vol. 21, Nr. 4.

Magnenat-Thalmann, N. – Laperrière, R. –Thalmann, D (1988): Joint-Dependent Local Deformations for Hand Animation and Object Grasping. *Proc. Graphics Interface'8*. 26–33.

Richard P. Feynman – R. B. Leighton – M. Sands (1985): *Optika/Anyaghullámok* (Mai fizika 3.). Budapest, Műszaki

Szirmay-Kalos L. – Umenhoffer T. (2008): Displacement Mapping on the GPU - State of the Art. *Computer Graphics Forum*, Vol. 27, No. 6. 1567–1592.

Szirmay-Kalos L. – Antal Gy. – Csonka F. (2004): *Háromdimenziós grafika, animáció és játékfejlesztés*. Budapest, ComputerBooks.

Tóth B. (2013): *Efficient sampling methods for photorealistic rendering on the GPU*. PhD thesis. Budapest.

Várady T. – Salvi P. – Rockwood, A. (2013): Transfinite Surface Patches Using Curved Ribbons. *EuroGraphics 2013 – Short Papers*. 5–8.

A Nemzeti Közszerológáti Egyetem kiadványa



Kiadó:

Nemzeti Közszerológáti Egyetem
Közigerázgatósi Továbbképzési Intézet

www.uni-nke.hu

Felelős kiadó:

Prof. Dr. Kis Norbert rektorhelyettes
Címe: 1083 Budapest, Üllői út 82.

Olvasószerekztő:

Dorogi Katalin

Tördelőszerkezttő:

Friebert Máté

ISBN 978-963-498-391-0 (elektronikus)