

**Kóródi Mihály**

# **Számítógépes grafika mindenkinek**

**BBS-INFO, 2016.**

Minden jog fenntartva! A könyv vagy annak oldalainak másolása, sokszorosítása csak a kiadó írásbeli hozzájárulásával történhet.

A könyv nagyobb mennyiségben megrendelhető a kiadónál:  
BBS-INFO Kft. 1630 Bp. Pf. 21. Tel.: 407-17-07

A könyv megírásakor a szerző és a kiadó a lehető legnagyobb gondossággal járt el. Ennek ellenére a könyvben előfordulhatnak hibák. Az ezen hibákból eredő esetleges károkért sem a szerző sem a kiadó semmiféle felelősséggel nem tartozik, de a kiadó szívesen fogadja, ha ezen hibákra felhívják figyelmét.

Papírkönyv ISBN 978-615-5477-46-1  
E-book ISBN 978-615-5477-49-47-8

Kiadja a BBS-INFO Kft.  
1630 Budapest, Pf. 21.  
Felelős kiadó: a BBS-INFO Kft. ügyvezetője  
Nyomdai munkák: Biró Family Nyomda  
Felelős vezető: Biró Krisztián

# Tartalomjegyzék

<b>Tartalomjegyzék.....</b>	<b>3</b>
<b>1. Bevezetés .....</b>	<b>7</b>
<b>2. Grafikai alapismeretek .....</b>	<b>9</b>
2.1 A számítógépes grafika alapfogalmai.....	9
2.1.1 A digitális képfeldolgozás.....	9
2.1.2 Grafikus primitívek .....	12
2.1.3 Képek megjelenítése grafikus kijelzőn .....	13
2.1.4 Színterek.....	16
2.1.5 Színkódolás, színmélység .....	18
2.2 Rasztergrafika .....	21
2.2.1 A rasztergrafikus modellezés alapjai.....	21
2.2.2 Élsimító algoritmusok .....	23
2.2.3 Képtömörítési eljárások.....	24
2.2.4 Rasztergrafikus képfájl formátumok.....	25
2.2.5 Képek konvertálása.....	28
2.3 Vektorgrafika.....	29
2.3.1 A vektorgrafikus modellezés alapjai .....	30
2.3.2 Transzformációk.....	31
2.3.3 Alakzatok .....	33
2.3.4 Útvonalak.....	34
2.3.5 Betűképek.....	36
2.3.6 Vektorgrafikus képfájl formátumok.....	38
<b>3. Képek rendszerezése, szerkesztése és megosztása .....</b>	<b>42</b>
3.1 IrfanView .....	43
3.1.1 A program képességei .....	44
3.1.2 Letöltés és telepítés .....	46
3.1.3 Képnézegetés, alpműveletek .....	48
3.1.4 Képek vágása, átméretezése, mentése .....	50
3.1.5 Csoportos szerkesztési műveletek .....	52
3.1.6 A Miniatúrák böngészési mód .....	56
3.1.7 Képernyőkép készítése.....	57
3.2 Windows Fotótár .....	59
3.2.1 Beszerzés és telepítés .....	60
3.2.2 Általános jellemzők.....	60
3.2.3 Több képet egyesítő eszközök .....	62
3.3 Linuxos képnézegetők .....	63

3.3.1 GNOME Image Viewer .....	64
3.3.2 Gwenview .....	65
3.3.3 gThumb .....	66
3.3.4 Shotwell .....	68
3.4 Fotók a weben .....	70
3.4.1 Felhőszolgáltatások .....	71
3.4.2 Fotók a Facebookon .....	72
3.4.3 Google Fotók .....	74
<b>4. GIMP .....</b>	<b>79</b>
4.1 Általános jellemzők .....	81
4.2 Letöltés és telepítés .....	81
4.3 A felhasználói felület .....	82
4.3.1 Dokkolható párbeszédablakok .....	84
4.3.2 Megnyitás, új kép létrehozása .....	85
4.3.3 Mentés és exportálás .....	87
4.3.4 Navigáció a munkaterületen .....	88
4.3.5 Értékek megadása .....	89
4.3.6 Színek és a Színpipetta-eszköz .....	90
4.3.7 Távolságok mérése és a Mérőeszköz .....	93
4.4 Rétegek .....	94
4.4.1 A Rétegek lap .....	94
4.4.2 Rétegtípusok .....	95
4.4.3 Alapvető rétegműveletek .....	97
4.4.4 Vízjel készítése .....	100
4.4.5 Dinamika növelése .....	103
4.5 Kijelölések .....	104
4.5.1 Általános kijelölő eszközök .....	105
4.5.2 Egy gyakori kijelölési művelet .....	111
4.5.3 „Intelligens” kijelölő eszközök .....	113
4.5.4 Műveletek kijelölésekkel .....	116
4.6 Útvonalak .....	119
4.6.1 Útvonal szerkesztése .....	120
4.6.2 Az útvonal, mint kijelölés .....	122
4.6.3 Kijelölés útvonallá alakítása .....	124
4.6.4 Rajzolás útvonalakkal .....	126
4.7 Transzformációk .....	127
4.7.1 Globális transzformációs műveletek .....	128
4.7.2 Transzformációs eszközök .....	132
4.7.3 Ferde horizont vízszintesítése .....	139
4.8 Rajzolás, festés .....	141
4.8.1 Ceruza és ecset .....	142
4.8.2 Festékszóró .....	144

4.8.3 Radír .....	144
4.8.4 Tus.....	146
4.8.5 Kitöltés.....	147
4.8.6 Színátmenet.....	147
4.8.7 Rajzolás segédvonalak mentén.....	148
4.9 A Szövegeszköz .....	150
4.9.1 Betűtípusok .....	152
4.9.2 Szöveg útvonalra illesztése .....	152
4.9.3 Kép körbefuttatása szöveggel.....	154
4.10 Retusáló eszközök .....	155
4.10.1 Másolóeszközök .....	155
4.10.2 Elmosás, maszatolás .....	159
4.11 Csatornák és rétegmaszkok.....	162
4.11.1 Csatornák .....	162
4.11.2 A gyorsmaszk eszköz .....	166
4.11.3 Rétegmaszkok.....	167
4.11.4 Rétegek összemosása .....	168
4.11.5 Háttér eltávolítása maszkolással .....	170
4.12 Klasszikus grafikai gyakorlatok .....	174
4.12.1 Tükröződés .....	174
4.12.2 Kijelölt terület átszínezése.....	175
4.12.3 Szépia.....	176
4.12.4 Vintage szűrő.....	178
4.12.5 Galériaborító .....	182
4.12.6 Üveges hatású gomb.....	186
4.12.7 Fejcsere .....	189
4.13 GIF animációk .....	190
4.13.1 Képkockák kézi szerkesztése .....	190
4.13.2 Animációs szűrők alkalmazása .....	193
4.13.3 GIMP animációs pluginek.....	197
<b>5. Inkscape .....</b>	<b>200</b>
5.1 Letöltés és telepítés.....	201
5.2 A felhasználói felület.....	202
5.3 Általános jellemzők .....	204
5.3.1 Mentés és exportálás.....	207
5.3.2 Navigáció, zoom, mérés .....	210
5.3.3 Rajzelemek, csoportosítás .....	213
5.3.4 A kijelölési és csomópont-szerkesztési mód.....	216
5.3.5 Kitöltés és körvonal .....	224
5.4 Szabályos alakzatok és útvonalak rajzolása.....	229
5.4.1 Szabályos alakzatok.....	229
5.4.2 Kézzel húzott útvonalak .....	235

5.4.3 Szövegobjektumok.....	244
5.4.4 Igazítás és elrendezés .....	247
5.4.5 Dokumentumbeállítás .....	254
5.4.6 Exportálás PNG fájlba .....	257
5.5 Összetett formák szerkesztése .....	259
5.5.1 Halmazműveletek.....	259
5.5.2 Területek kitöltése.....	264
5.5.3 Színátmenetek.....	268
5.5.4 Fények és árnyékok.....	271
5.5.5 Minták hajlítása, útvonaleffektek.....	274
5.6 Vektorizálás .....	279
5.6.1 Bitképek vektorizálása egyetlen útvonallá .....	281
5.6.2 Többmenetes vektorizálás.....	285
5.6.3 Vektorgrafikus ábrázolás .....	287
5.6.4 Térképek és útvonaltervek.....	291
5.7 Precíz szerkesztési lehetőségek.....	292
5.7.1 A pillangó technika.....	293
5.7.2 Szerkesztés rácsvonalak mentén.....	296
5.7.3 Segédvonalak.....	298
5.7.4 Rétegek alkalmazása.....	299
<b>6. Grafikai tervezés .....</b>	<b>301</b>
6.1 Egységes arculat.....	302
6.2 Arány, egyensúly és színek.....	304
6.3 A célközönség .....	307
6.4 Tehetség, kreativitás.....	307
6.5 Referenciaképek, ötletgyűjtés másoktól .....	308
<b>7. Digitális rajztáblák .....</b>	<b>310</b>
<b>8. Nyílt forrású grafikai programok használata .....</b>	<b>313</b>
<b>Irodalomjegyzék .....</b>	<b>315</b>

# 1. Bevezetés

A körülöttünk létező világ érzékelésének legmeghatározóbb módja a látás, a vizuális befogadás. A valódi háromdimenziós tárgyak észlelése és azok egyszerű értelmezése mellett mi emberek bámulatos elvonatkoztató képességgel is rendelkezünk, melynek köszönhetően egy-egy látott forma, vizuális információ mögött lényegesen több mondanivaló is rejtőzhet számunkra. Ahogyan egy különös formájú felhőben is megláthatjuk a vágató lovat, úgy egy papírlapra firkált ákom-bákomban is gyakran fellelhetjük a képzőművészeti alkotást. Ha most egy kicsit félretesszük az absztrakt művészetet, nagyon leegyszerűsítve egy-egy ilyen grafikai alkotás minél jobban hasonlít az általunk ismert valóságra, annál jobban sikerültebbnek, értékesebbnek mondhatnánk, ám még ennél is nagyobb értéket képviselhet, ha a látvány élethűségén kívül tartogat valami rendkívülit, megfoghatatlant. A kép így alkalmassá válik, hogy megmozgassa absztrakciós képességeinket is, de akár egy már soha vissza nem térő, különleges pillanatot is megörökíthet.

A számítógéppel létrehozott grafikákkal sincs ez másként. A 3D-s animációs filmek világa például méltán nevezhető annak az ágazatnak, amellyel a legtöbben már találkozhattunk, legyen szó bármely embercsoportról, bármely korosztályról. Azzal együtt, hogy az utóbbi években egyre több és egyre fejlettebb animációs filmmel találkozhatunk, a hagyományos filmek esetében is egyre gyakrabban alkalmaznak CGI (*Computer-Generated Imagery*, azaz *Számítógépen Létrehozott Kép*) technikákat. Olyan valóság-hű, ugyanakkor a valóságtól messzemenőig elrugaszkodó vizuális élményhez juthatunk az így készült trükkök által, amelyeket a valóságban lefilmezni többnyire lehetetlen lenne. Mindez természetesen megköveteli egy csapat elhivatott grafikus alkalmazását, akik megérdemlik, hogy művésznek nevezzük őket a maguk szakmájában. Nem szabad persze megfeledkeznünk arról sem, hogy az alkotás elkészítéséhez szükség van egy őket kiszolgáló, elképesztően nagy grafikai számítási teljesítményre képes számítógépes környezetre is.

A számítástechnika egyes ágazatain belül vitathatatlanul a grafika az egyik, amely az utóbbi két évtizedben a legrohamosabban fejlődött, és teszi ezt napjainkban is töretlenül. Ebből a fejlődésből az otthoni felhasználók számára elérhető grafikus hardverelemek azok, amik a legkevésbé sem maradnak ki, amit nagyban segít az, hogy egyre nagyobb bonyolultságú, azaz egyre nagyobb integráltsági fokú áramköröket képesek előállítani egyre

kisebb szilíciumlapkákön. Az évekkel ezelőtt még asztali számítógépen is tiszteletre méltó grafikai feldolgozó teljesítmény napjainkra már egy okostelefonban vagy egy táblagépben is ott lakozik. Ez a jelenség már önmagában is jelentősen felkavarja az informatikai és szórakoztató elektronikai üzletágat, a határok a PC, a mobiltelefon és a játékkonzol között jelentősen összesimosódnak, és már a televíziók is többmagos processzorral, fejlett operációs rendszerrel és internetes alkalmazásokkal rendelkeznek. Ennek köszönhetően a számítógépeken készített grafikák felhasználási területének listája és célközönsége is egyre bővül, beépül mindennapjainkba, és többé már nem csak a billentyűzet és monitor előtt ülőkre korlátozódik. Napról napra ezrek és ezrek kapcsolódnak be ebbe a rohamosan fejlődő ágazatba akár a munkájuk miatt, akár csak saját örömeikre, és persze a legideálisabb esetben e kettő indok össze is találkozhat egymással. Mint általában a számítástechnika valamennyi felhasználási területén, a grafikánál is sokan vannak, akik bőven beérik egy alapszintű tudással, de vannak, akik az alapos ismeretek elsajátítására törekcsenek. Mindkét esetben fontos, hogy az alapismerektől kezdve, azokat kellőképpen megismerve építsük fel azt, ahová, amilyen szintre el szeretnénk jutni.

A könyv tartalomjegyzékét áttekintve látható, hogy egyaránt fogunk találkozni elméleti és gyakorlati témakörökkel, ám mindent összevetve mégis inkább a gyakorlati megvalósítások ismertetése lesz a meghatározó. Bár az elméleti alapok mögötti alapos számításokat átugorjuk, és a minden részletre kiterjedő, kimerítő magyarázatokat is igyekszik nélkülözni a könyv, mindamelllett a gyakorlati ismeretek tanulmányozásába csakis azt követően tanácsos belevágnunk, miután a kapcsolódó legalapvetőbb elméleti ismeretekre már szert tettünk. Előfordulhat persze, hogy sokaknak ez az elméleti rész már ismert tudást tartalmaz, mégis be kell látnunk, hogy elkerülhetetlen azok áttekintése, különösképpen egy alapvetően kezdőknek szóló műben. Tanácsos tehát minden olyan alapfogalmon átrágnunk magunkat, amelyek szükségesek lehetnek ahhoz, hogy felkészülten vágassunk bele a gyakorlati részek tárgyalásába.

Mindazonáltal még ebből a viszonylag alaposnak mondható elméleti bevezetőből is hiányozni fognak azok a témakörök, amelyek a webre szánt, generált számítógépes grafika előállításának szempontjából nélkülözhetők. Így például különösebben nem fogunk foglalkozni a DTP (*Desk Top Publishing*) témakörével, azaz a kiadványszerkesztéssel, továbbá a 3D-s grafikát is csak érintőlegesen tárgyaljuk. A 2D raszter- és vektorgrafika kérdéskörét viszont megpróbáljuk alaposan kivesézni. Tesszük mindezt legalább olyan szinten, hogy az olvasó az itt tárgyalt grafikai programokat akár még a könyvben bemutatottakon túlmenően is képes legyen magabiztosan használni, vagy bátorítást kapjon továbbiak megismeréséhez.



## 2. Grafikai alapismeretek

A számítógépes grafika viszonylag könnyen behatárolható területekre korlátozódik az informatikán belül, emiatt nem is túl bonyolult feladat azt bizonyos szempontok szerint kategorizálni, majd egy-egy kategóriát tovább részletezni. A felhasználási területek ugyan rendkívül szerteágazóak, legyen szó webgrafikák készítéséről, digitális képek javításáról, kiadványszerkesztésről vagy akár számítógépes animációkról, ám végső soron mindig ugyanazokkal az alapfogalmakkal és az azokkal kapcsolatos grafikai műveletekkel találjuk szembe magunkat.

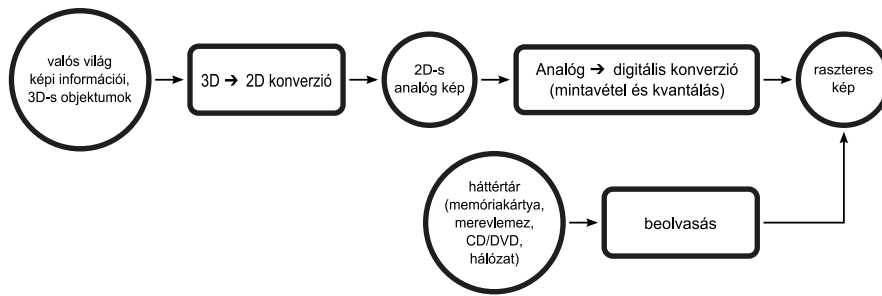
### 2.1 A számítógépes grafika alapfogalmai

A számítógépes grafika fogalomkörébe azokat az eljárásokat soroljuk, amelyek adattovábbítási és adatfeldolgozási szerepet töltenek be a számítógéphez kapcsolódó grafikus perifériák és a számítógép között. Maguk a grafikai adatok többféleképpen is bekerülhetnek a számítógépbe: vagy mi magunk, az általunk megadott parancsok és paraméterek szerint állítjuk elő azokat a számítógép erőforrásainak felhasználásával, vagy pedig a valóság egy-egy pillanatának leképezésével digitalizáljuk és juttatjuk be feldolgozásra.

Az általános meghatározás szerint az általunk, teljesen az alaptól kiindulva előállított grafikai adatok esetében, melyek aztán a csatlakoztatott grafikus kiviteli perifériákkal megjeleníthetőek (képernyő, nyomtató), *generatív* számítógépes grafikáról beszélhetünk. *Digitális képfeldolgozás* történik abban az esetben, ha a valóság képadatait digitális eszközökkel leképezve, akár fényképezőgéppel, akár szkennelvel vagy bármi ezekhez hasonlóval juttatjuk be a számítógépbe további feldolgozásra, átalakításra.

#### 2.1.1 A digitális képfeldolgozás

A számítógéppel csak digitális képadatokat lehet feldolgoztatni, így aztán a valóság képadatainak először valamilyen módszerrel be kell kerülniük a számítógépbe. Ezt bármilyen eszközzel hajtjuk is végre, az eredmény ugyanolyan formátumú: egy egymás alatt és mellett sorban elhelyezkedő pontokból álló rácsháló, azaz *raszteres* digitális kép (1. ábra).



1. ábra

A valós világ objektumai háromdimenziós térben helyezkednek el, amelyek a fényt különbözőképpen verik vissza, árnyékot vetnek vele vagy akár saját maguk bocsájtják ki azt. A képfeldolgozás első lépéseként a képbeviteli eszköz érzékelőjével a valós világ 3D-s tárgyairól egy nagyon rövid időintervallumban, a látható tartományba eső elektromágneses hullámokból egy 2D-s *analóg* képet hozunk létre. Ezután ez a kép a *mintavételezés* során egy rácshálóra feszítődik rá, amely sok-sok apró négyzetből áll. Ezek az apró négyzetek a *képpontok* vagy idegen szóval élve *pixelek* (*picture element*). A rácsháló sorainak és oszlopainak száma adja meg a kép *felbontását*. Ennek a felbontásnak a nagysága nagyrészt az érzékelőtől függ, azaz annak képesnek kell lennie a megfelelő részletességű analóg kép előállítására.

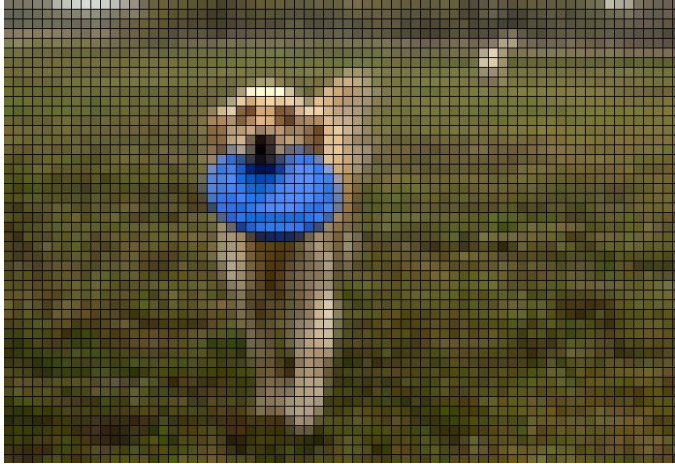
A következő folyamat a *kvantálás*, amelynek köszönhetően a képpontok, azok valódi színe és fényessége alapján egy előre meghatározott színskálából, a lehetséges legjobban közelítő értéket veszik fel. Hogy miért csak egy megközelítő értéket tudnak felvenni, annak oka, hogy a valós világban a színek és árnyalatok száma végtelen, ám a számítógépen csak korlátozott számban állnak ezek rendelkezésünkre. A színskála színeit hagyományosan a vörös, a zöld és a kék színek keverékéből határozzuk meg, ezekből kapunk egy RGB (*Red, Green, Blue*) számkódot. A mintavételezés és kvantálás után kapjuk meg tehát a raszteres képet, amely a kép felbontásának megfelelő oszlopokkal és sorokkal rendelkező rácshálóból áll, a háló minden egyes négyzetében pedig egy-egy szín kódja található. Ezt nevezzük összességében *bittérképnek*.

Az 1. ábrán egy további rövid folyamatot is végigkövethetünk, amikor is a képállomány már egy adathordozón létezik és a további digitális feldolgozáshoz szükséges raszteres képet már csak egy egyszerű beolvasási művelet kell, hogy előállítsa.

A raszteres kép tehát a további feldolgozásra szánt vagy közvetlenül megjelenítendő adathalmaz, egyelőre ez még nem maga a képfájl. Úgy is mondhatnánk, hogy a képfájl csupán a raszteres kép kivetülése, hogy az a

későbbiek során is felhasználható legyen. Egy szkennert esetében ugyanez az adathalmaz már alkalmas arra, hogy közvetlenül átkerüljön a számítógépre további feldolgozásra, amelyet követően a számítógépen már a szkennert szoftvere kell hogy elvégezze a további tömörítési és átalakítási eljárásokat, hogy végül egy képállományt kaphassunk. Fényképezőgépek és kamerák esetében ezek a további feladatok is még mind a készüléken belül hajtódnak végre, hogy aztán a képek annak memóriakártyáján eltárolásra kerüljenek.

A 2. ábrán egy 70x48 képpontból álló raszteres kép látható erősen kinagyítva.



2. ábra

Miután megkaptuk a feldolgozásra alkalmas adatok halmazát, már csak a számítógépen, a számítógép szoftverein és rajtuk a sor, hogy kezdjünk vele valamit. Mindazokat a számítógépes eljárásokat, utómunkálatokat, amelyek során az eredeti kép vagy annak bizonyos része az általunk kiadott parancsok szerinti változáson esik át, *képfeldolgozásnak* nevezzük.

A digitális képfeldolgozás egy következő lehetséges lépése az *alakfelismerés*, amelynek keretében a kép egyes területeit a képfeldolgozó szoftver különféle algoritmusok segítségével beazonosítja és elkülöníti (*szegmentálás*).



3. ábra

Alakfelismeréses feldolgozást végeznek például a műholdas felvételek elemzését végző szoftverek, amelyek az utakat, az épületeket és egyéb objektumokat képesek elkülöníteni a nem fontos részekről (3. ábra). Ugyanide sorolhatjuk még például a Facebook közösségi oldal vagy a digitális fényképezőgépek arcfelismerő funkcióját is. A fényképezőgépek egyszerű arcfelismerő rendszerének elsődleges feladata az arc automatikus fókuszba hozása (*Face Priority AF*), amely során már előre elkészül egy átmeneti raszteres kép, a képet átvizsgálva az azon lévő sötét foltok alapján a gép szoftvere pillanatok alatt megtippeli az arc helyzetét, változtat az optika fókuszpontján, majd visszajelzést küld a felhasználónak, hogy készen áll a már megfelelően élesített kép elkészítésére. Az újabb arcfelismerő rendszerek a pislogás kiszűrésére is alkalmasak. Végül, ha már megemlítettünk néhány alakfelismerő megvalósítást, ne feledkezzünk meg az általunk valószínűleg leggyakrabban alkalmazott rendszerről sem: a vonalkód olvasóról.

### 2.1.2 Grafikus primitívek

Az alakfelismerés során és a generatív grafikában is többnyire szabályos geometriai alakzatokkal dolgozunk. Ezek az alapvető alakzatok a *grafikus primitívek*, melyek sarok- vagy tárgypontjai koordinátákkal meghatározhatók. A kétdimenziós grafikában az alábbiakkal találkozhatunk:

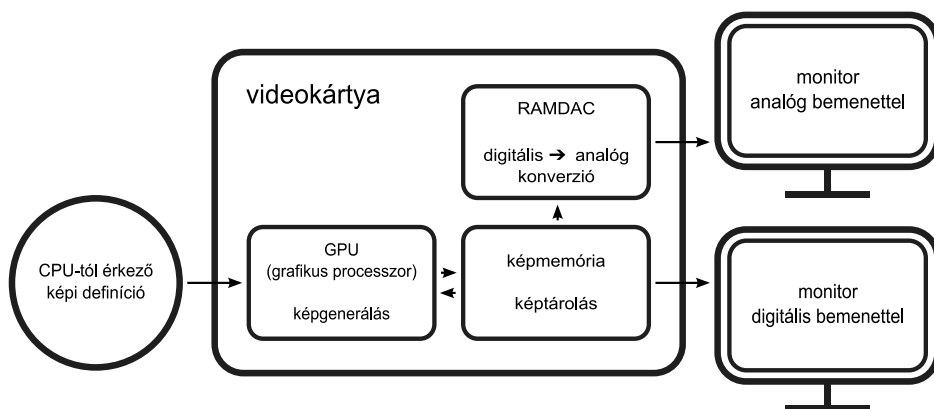
- **pont**: az egyetlen koordinátapárral meghatározott legegyszerűbb alakzat
- **vonall**: kezdő- és végponttal, valamint esetleges töréspontokkal rendelkező grafikai elem
- **görbe**: kezdő- és végponttal rendelkezik, amely egy vagy több csomópont mentén, szabályos ívben görbül, vonalvezetését segédvonalak rögzítik
- **sokszög (poligon)**: egyenes vonalakból álló, területet ábrázoló alakzat, melynek kezdő és végpontja egybeesik
- **kör és ellipszis**: szabályos ívvel rendelkező, az egymással szemben lévő oldalaival szimmetrikus, területet ábrázoló alakzat

A vonal, sokszög és a kör vagy ellipszis rendelkezhet színnel, vonalvastagsággal és vonalstílussal. A sokszöget és a kört vagy ellipszist – mivel területet meghatározó alakzatok – kitöltő színnel vagy kitöltő mintázattal is elláthatjuk. Számos esetben a sokszögek helyett csak a háromszöget sorolják a primitívekhez, mivel háromszögek segítségével bármilyen sokszög megalkotható.

Az iménti felsorolásból kimaradt, de gyakran a betűket és feliratokat is a primitívek közé szokás sorolni, mivel mind a raszter-, mind a vektorgrafikában többnyire önálló rajzobjektumokként kezeljük őket.

### 2.1.3 Képek megjelenítése grafikus kijelzőn

A képmegjelenítési feladatok nagy részét ma már a számítógép grafikus kártyája kell, hogy elvégezze a központi processzorral szorosan együttműködve. A grafikus kártyát sokféleképpen nevezhetjük: a videokártya, videoadapter, videovezérlő, grafikus adapter elnevezések mind a számítógép ugyanazon hardverösszetevőjét jelölik. Kivitelezését tekintve a videokártya, mint alkatrész, létezhet bővítőkártya vagy alaplapra integrált változatban is. Manapság a hagyományos képmegjelenítési feladatokon túlmenően már minden modern videokártya képes valamilyen szinten 3D-s számításokat végezni, amely jelentősen tehermentesíti a CPU-t és lehetővé teszi a modern, nagy részletgazdagsággal rendelkező 3D-s alkalmazások, azaz leginkább játékok futtatását. A grafikus hardver rohamos fejlődése tette lehetővé azt is, hogy megjelenhessenek az első grafikus operációs rendszerek, amelyek napjainkra már nélkülözhetetlenné váltak, és a funkcióikat, az általuk nyújtott felhasználói élményt tekintve szintén folyamatosan fejlődnek.



4. ábra

A fenti ábrán láthatjuk, hogy a központi processzor felől érkező képi információk miként teszik meg útjukat, hogy végül a monitorra vagy más megjelenítő eszközre (televízió, projektor) kerülhessenek. A **grafikus processzor** (*Graphics Processing Unit*) a videokártya központi egysége, amelynek fő feladata, hogy mindennemű, grafikák létrehozásával és megjelenítésével kapcsolatos feladatot átvegyen a CPU-tól. A grafikai

számítások eredményét a GPU a *képmemóriába* küldi tovább, amely a raszteres képi információk ideiglenes tárolását végzi el. A legújabb videokártyákon és megjelenítő eszközökön már található digitális, DVI (*Digital Visual Interface*) csatlófelület is, így a digitális jel innentől már közvetlenül is továbbítható. Korábban minden esetben szükség volt még a kártyák **RAMDAC** (*RAM Digital-Analog Converter*) egységére is, hogy a digitális képi információkat analóg jelekké alakítva, a hagyományos, analóg csatlófelületen keresztül kommunikáló megjelenítő eszközök értelmezni tudják.

A legelterjedtebb analóg csatlófelület a D-SUB (DE-15), amelyet biztosan már jól ismerünk: a csatlakozó vége kék színű és trapéz formájú. Bár e csatlakozótípus használatára egyre ritkábban van szükség, azt továbbra is szinte minden új videokártya és noteszgép tartalmazza. A D-SUB csatlakozót emlegetik még VGA (*Video Graphics Array*) vagy RGB csatlakozó neveken is.

A videokártyát elhagyva a képadatok rasztergrafikus kijelzőre kerülnek megjelenítésre, amelyek közül a leggyakrabban alkalmazott kiviteli eszköz a monitor. A monitorokat elsősorban azok képmegjelenítési technológiája szerint különíthetjük el úgy, mint **CRT** (*Cathode Ray Tube*, azaz katódsugárcsőves), **LCD** (*Liquid Crystal Display*, azaz folyadékkristályos), **PDP** (*Plasma Display Panel*, azaz plazmaképernyős), valamint **OLED** (*Organic Light-Emitting Diode*, azaz szerves alapú fényemittáló diódás).

Az LCD újabb generációit a kijelzőpanelen megjelenő további vékonyfilm tranzistoros aktív mátrix réteg alapján TFT-LCD-nek vagy csak röviden **TFT**-nek (*Thin Film Transistor*) is hívják, továbbá a legújabb TFT kijelzők háttérvilágítását már LED-ek biztosítják, ezért gyakran emlegetik őket LED kijelzőknek is. A LED-es háttérvilágítással nagyobb fényerőt, kevesebb napfény-visszaverődést és alacsonyabb energiafogyasztást tudnak elérni. Az OLED kijelzők esetében szerves anyagok felhasználásával előállított LED-eket használnak a pixelek egyenként történő megvilágításához. Az ilyen kijelzők nagyon jó betekintési szöggel rendelkeznek, egészen nagy fényerőt is képesek kiadni, miközben a fekete igazán mély fekete maradhat. Az OLED leggyakrabban alkalmazott változata az **AMOLED** (*Active Matrix OLED*, azaz aktív mátrixos OLED), amely egy vékonyfilm tranzistoros réteggel bővített megoldás. Mindegyik technológiának megvannak az előnyei és hátrányai, de ezeket most nem részletezzük egyenként.

Csak érdekességképpen, a '70-es években még gyakoriak voltak a vektoros kijelzők is, amelyek képe szabályos vonalak és görbék kiszámításával állt össze. Ezek a megjelenítő eszközök az oszcilloszkópokhoz hasonló CRT monitorok voltak. Bár esetükben nem volt szükség a ferde vonalak élsimítására, azonban csak egyszerűbb formák megjelenítését tették lehetővé, így mára már teljesen eltűntek, és leváltották őket a modern raszteres kijelzők.

Ahogy azt már megtudtuk, a raszteres kép tovább már fel nem osztható részét pixelnek, vagy másképpen képpontnak nevezzük. A számítógépen tárolt raszteres kép által behatárolt rácsháló egy elemének esetében ugyanúgy képpontról beszélünk, mint ahogyan a monitor kijelzőjének felületén található elemi egységről. Mire a számítógépből a kijelzőre kerül a képi információ, számos transzformáció következik be, melyeket a számítógép grafikus hardvere és a szoftverek végeznek el. Ideális esetben a számítógépen tárolt raszteres kép felbontása kisebb vagy egyenlő a kijelző felbontásával és így a transzformációk ellenére a raszteres kép ugyanazon képpontja a képernyőn is ugyanott és ugyanazzal a színkóddal fog megjelenni, azaz a képet natív felbontásban láthatjuk. A nagyítási és kicsinyítési transzformációk történhetnek a grafikai szoftveren belül a kép szerkesztése során vagy a számítógép grafikus perifériáján, az általunk megadott felbontásnak megfelelően.

A *képernyő felbontása* alatt a kijelző teljes felületén, vízszintesen és függőlegesen elhelyezkedő képpontok számát értjük. Ennek alapja eredetileg a VGA szabvány, amely vízszintesen 640, függőlegesen 480 képpontot jelent a klasszikus, 4:3-as *képaránnyal*. Ennek további, gyakoribb változatai az SVGA (800x600), XGA (1024x768), SXGA (1280x1024) és UXGA (1600x1200). Manapság a monitor és notebook kijelzők esetében a szélesebb képarányok, azaz a 16:10 vagy a 16:9 a meghatározóbbak, amelyek eredetileg a mozifilmek képaránya miatt kezdtek egyre inkább terjedni, de a hétköznapi munka során is kényelmesnek bizonyulnak. Olyan képernyő felbontások születtek így, mint a napjainkban gyakori WXGA 16:9 (1366x768), a WXGA+ (1440x900) vagy FullHD (1920x1080), stb. Ezekkel a felbontásokkal a gyártók a natív felbontást, azaz a valós képpontszámokat jelölik. Azt a felbontást, amivel a legélesebb és legrészletesebb képet láthatjuk az adott képernyőn. A monitorok természetesen alacsonyabb felbontásban is képesek működni, ekkor azonban a kép részletessége – különösen a nem CRT monitorok esetében – látványosan gyengébbé, elnyújtottá válik, de ritkán előfordulhat a kicsi, középre zárt, fekete szegélyekkel határolt megoldás is.

A *képpontok távolságát* az egy hüvelyk hosszúságú szakaszon elhelyezkedő képpontok számával határozzuk meg, amely vízszintes és függőleges irányban többnyire megegyezik. Mértékegységét képernyők esetében általában PPI-ben (*Pixels Per Inch*) határozzák meg, a nyomdatechnikában, szkennereknél pedig DPI-ben (*Dots Per Inch*). Ugyancsak DPI-ben határozzák meg az egerek érzékenységét is.

A ma forgalomban lévő képernyők nagyjából 96 és 130 PPI közötti képpont távolsággal rendelkeznek. Ennél magasabb értéknek monitorok és noteszgépek esetében jelenleg nincs értelme, hiszen ugyanazok a grafikai

elemek más kijelzőkön sokkal kisebbnek hatnának. Mobil készülékek esetében azonban már más a helyzet, esetükben ma már a 200-300 PPI vagy még ennél is nagyobb érték az elfogadott. Viszonyításképpen, egy 96 PPI-s kijelző egyetlen négyzethüvelyknyi területén 96x96, azaz összesen 9216 elemi képpont található meg. A CRT monitorok esetében gyakori a pixelméret megadása is, amely a legmodernebb típusoknál jellemzően 0,22 mm.

A nagy felbontásnak igazán a nyomdatechnikában van jelentősége, ugyanis a képernyőn elfogadható élességűnek tűnő kép kinyomtatva már közel sem lesz ugyanannyira részletgazdag. Nyomtatáskor, az egy hüvelykre jutó pontok számára is inkább a felbontás kifejezéssel szokás hivatkozni. Természetesen a nagyobb felbontás több vízszintes és függőleges képpontot, ezáltal nagyobb fájlméretet is igényel. A generált grafikák esetében általában a 300 DPI a minimális, nyomtatásban már elfogadható felbontás, a mai lézernyomtatók általában 600-1200 DPI nyomtatási minőségre képesek.

A képernyőkön tehát a kisebb képpont távolság is elegendő, azonban a számítógépnek a képet másodpercenként többször is meg kell változtatnia. A képernyőknek van egy ide vonatkozó, további jellemzőjük is, mégpedig az, hogy egy másodperc alatt hányszor rajzolják újra a megjelenített képet. Ezt röviden úgy nevezzük, hogy *képfrissítési frekvencia*, amely a mai kijelzők esetében tipikusan 60 Hz körül már teljesen elfogadható. A régebbi katódsugárcsőes monitoroknál és televízióknál fontos szempont volt a minél magasabb, lehetőleg 70 Hz-nél nagyobb képfrissítési frekvencia, mivel azok esetében a kép valóban másodpercenként annyiszor villant fel. Az alacsonyabb képfrissítés a szem gyors elfáradását, látásromlást okoz. A többi kijelzőtípus esetében, azok működéséből adódóan a monitorra küldött jel képfrissítési frekvenciája nem kell, hogy 60-70 Hz-nél jelentősen magasabb legyen, a kép folyamatosnak tűnik, nincs vibrálás.

#### 2.1.4 Színterek

A színterek a színek ábrázolására szolgáló különféle rendszerek. Segítségükkel lehetőségünk van a valós világ végtelen számú színét rendszerbe foglalni, a nekünk szükségeset kiválasztani és azt az adott színtér saját szabványának megfelelő formában, színkóddal helyettesíteni.

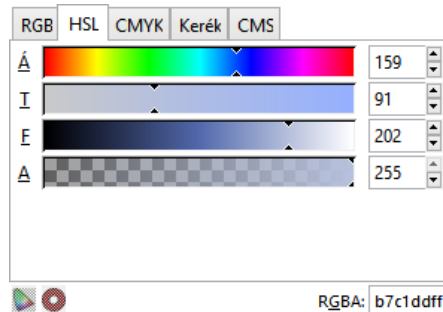
A színek pontos definiálására mindig is nagy igény volt a tudomány és művészetek több ágában is. Szükség volt egy skálára, amely segítségével egyértelműen és szabványos módon beazonosíthatóak a valós világ színei. Az egyik korai, már jól használható színteret *Albert H. Munsell* alkotta meg 1905-ben, aki a színeket a színárnyalat (*hue*), az intenzitás (*value*) és a színtelítettség (*chroma*) írja le. Egy ezen alapuló későbbi szín-szabvány,



amelyet nemzetközileg is elfogadtak, az 1931-ben megjelent *CIE XYZ*. Ez a szabvány az emberi szem színérzékelő receptorainak tartományán alapul, amely így megkönnyíti a megfelelő szín kiválasztását.

Az emberi érzékeléshez jobban alkalmazkodó színtereknek gyakorlati megvalósításai is vannak, melyek közül a leggyakoribb a *HSL*, ami az angol *Hue* (Árnyalat), *Saturation* (Telítettség) és *Lightness* (Fényesség) szavak kezdőbetűiből származik. Egészen hasonló hozzá a *HSV* (H, S, *Value*), amelyre gyakran hivatkoznak még *HSB* (H, S, *Brightness*) néven is.

A HSL és HSV színterek a 421. ábrán, a hátlap belső oldalán találhatóak. Ezek a színterek ugyan nem szabványosítottak, de a mai képszerkesztő programokban is találkozhatunk velük, mivel segítségükkel sokkal könnyebben tudjuk kiválasztani a kívánt színt, amely aztán az *RGB* szabványnak megfelelő szín kódját kapja meg (5. ábra). A gyakorlatban tehát jellemzően a HSL, a HSV vagy egy ezeken alapuló színtér segítségével választjuk ki a kívánt színt, majd azt a grafikai szoftver átalakítja az RGB-nek megfelelő színkóddá.



5. ábra

A televíziózásban és a monitorok esetében is az egyes képpontok három további apró pontból vagy pálcikából, egy vörösből, egy zöldből és egy kékből állnak össze, melyekből a felhasználó szabad szemmel már csak a végeredményt, az összekevert színt látja. Ez valóban azt jelenti, hogy ezzel a három alapszínnel, azok megfelelő arányú kombinációjával gyakorlatilag bármilyen szín előállítható a képernyőkön. Az RGB egy *additív*, azaz összeadó színkeverési eljárást tesz lehetővé, ugyanis az alapszínek különböző mértékű hozzáadásával határozza meg az egyes színeket. Ha a képernyő egy képpontjában mindhárom alapszín maximális intenzitással világít, fehéret kapunk. Ezzel a módszerrel az úgynevezett elsődleges fényforrások színeit tudjuk definiálni. A könyv hátsó borítójának belső oldalán, a 422. ábrán, az LCD panelek egy apró darabkájának elvi felépítését

láthatjuk (nem valódi nagyítás), amelyen jól kivehetőek a képpontokat felépítő vörös, zöld és kék pálcikák.

A képernyőkön való megjelenítésen kívül szükség van még a számítógépen tárolt képek nyomtatására is. Ellentétben az additív színkeveréssel, nyomtatáskor az egyes színek egymás erősítése helyett egymást épp lefedik, ennél fogva a nyomtathatóság miatt nélkülözhetetlen egy ennek a célnak megfelelő színkeverési eljárás bevezetése. A *szubtraktív*, azaz kivonó színkeverés esetében az elsődleges színek kiegészítő párjaival, a ciánkék, a bíborvörös és a sárga színekkel határozzuk meg a nyomtatáshoz szükséges alapszíneket. Ebből következően ennek a színkeverési eljárásnak az elnevezése *CMY* (*Cyan, Magenta, Yellow*). A CMY esetében a három szín maximumából a fekete színt kapjuk meg eredményül.

Az RGB alapszínek közvetlen kapcsolatban vannak a CMY alapszínekkel. Minden színnek létezik egy *komplementer*, más néven kiegészítő színpárja. (A komplementer szó a latin *complementum* (kiegészítés) szóból ered és a színábrázoláson kívül még számos tudományágban alkalmazzák.) Egy adott színnek és annak komplementer színpárjának egyesítésével mindig az adott színkeverési módszerrel előállítható maximális értéket kapjuk meg eredményül. Például a teljesen vörös szín komplementere a ciánkék, így a kettő együtt fehér színt eredményez RGB, és feketét a CMY színkeverési modellben. Az RGB és CMY színkeverést és az egyes színpárokat szemlélteti a 423. ábra a könyv hátsó borítójának belső oldalán.

A gyakorlatban persze mindez nem ennyire egyszerű, ugyanis a nyomtatásban nem jeleníthető meg a képernyővel teljesen megegyező minőségben mindegyik szín, így nem lenne lehetséges elegendően fekete színt kikeverni sem. A probléma kiküszöböléséhez a CMY modellt kiegészítették még egy további értékkel is, amely a hozzáadott fekete mértékét határozza meg. Így került bevezetésre a nyomdatechnikában a *CMYK* színtér, amelynél az utolsó betű a *Key* angol szóból származik. Az eljárásból adódóan ezt a technikát *négyszínnyomásnak* is nevezik.

### 2.1.5 Színkódolás, színmélység

A valós világ színeinek digitális feldolgozásához a szükséges tárhelyet is hatékonyan kell kihasználni. Lehetetlen a világ minden színét definiálni és tárolni, de a képfeldolgozás gyors fejlődésének köszönhetően napjainkban ezt akár rendkívül nagy megközelítéssel is végezhetjük. Amikor a számítástechnika még csak hőskorát élte, még nem álltak rendelkezésre a valódi színeket a jelenlegi mértékben visszaadni képes megjelenítők és így nem volt szükség óriási színskálára sem, azaz a színek tárolása kevesebb

**biten** is legendónek bizonyult. Olyannyira így volt ez, hogy legelőször még egy képponthez egyetlen bit tartozott csupán, azaz vagy világított az a képpont, vagy nem. Egy rastergrafikus kép esetében a nagyobb számú bit felhasználása több szín tárolását teszi lehetővé, amihez nagyobb tárterület is szükséges. A digitális kép bármely képpontjához tartozó bitek számával határozzuk meg egy kép **színmélységét**. A manapság elérhető óriási tárhelyek ellenére napjainkban sincs minden egyes kép esetében szükség az összes szín tárolására, hanem lehetőségünk van az adott igényekhez igazodva kisebb kapacitást igénylő, a weboldalakon vagy a különféle felhasználói felületeken gyorsabban megjelenő képek előállítására és tárolására is.

A számítástechnika fejlődése során többféle színmélység-szabványt is kidolgoztak, ezek közül a következőkben az elterjedtebbeket, a jelenleg is használatosakat vesszük sorba (ezeket szemlélteti a 424. ábra a könyv végén, a hátsó borító belső oldalán).

**Egyetlen bitet** használ a **Monochrome** színskála, azaz az adott képpont vagy világít, vagy nem. Mivel így csak a fekete és a fehér áll rendelkezésre, nem is igazán beszélhetünk színekről, ám a kép egyes részletei a pontozásos színszórás alkalmazásával mégis viszonylag jól kivehetőek.

A **2 bites** színkódolással már négy színt vagy egyes esetekben a fekete és fehér között még két szürkeárnyalat megjelenítését érhatték el. Ilyen volt az IBM 1981-ben bemutatott CGA (*Color Graphics Adapter*) szabványú grafikus kártyája is, amely az első megjelenítési szabvány volt az IBM PC-hez.

A **4 bites** színmélységgel, azaz a 16 színnel már gyakran találkozhattunk a korai grafikus operációs rendszereknél, a régi grafikus alkalmazásoknál és számítógépes játékoknál. Ma is többnyire ezek a színek jelentik a főszíneket és adnak alapot a további árnyalatok kiválasztásához. Az EGA (*Enhanced Graphics Adapter*) megjelenítési szabványban is ez a 16 szín jelentette a használható színskálát.

256 szín megjelenítésére alkalmas a **8 bites** színkódolás, amely többek között az alacsony felbontású VGA (*Video Graphics Array*) szabványos színmélysége is. A 256 szín, különösen a pontozásos színszórással együtt, már gyakran elegendő ahhoz, hogy a részletesség megtartásával, viszonylag kis felbontású képeket, például az internetes oldalak egyes grafikai elemeit tároljuk és jelenítsük meg vele. A szabvány szerint a 8 bit úgy oszlik el a három színkomponens között, hogy a vörös és a zöld kap 3-3 bitet, a kék pedig 2-t. Ez abból adódik, hogy az emberi szem a kék szín árnyalatai között képes a legkevésbé megállapítani a különbséget. A valós színek **szürkeskálába** mentésekor szintén egy 8 bites, a fehér és fekete közötti 256 árnyalatot tartalmazó palettát kapunk.

Napjainkban már kevésbé gyakori a **16 bites** színmélység használata, ami 65 536 szín megjelenítését teszi lehetővé. A módszert a 8 bites rendszerektől való megkülönböztetés miatt **High Color** színskálának is nevezik. Mivel az emberi szem a zöld színre a legérzékenyebb, azaz a zöld árnyalatai között képes a legkönnyebben meglátni a különbségeket, ezért a 16 bitet 5:6:5 arányban osztották el a három szín között. Az 5 bit 32, a 6 bit 64 szám tárolására alkalmas, ezáltal a vörös és a kék esetében 0-tól 31-ig terjed a skála, a zöldnél pedig 0-tól 63-ig. Ezzel a színmélységgel már különösen részletgazdag képek is tárolhatók, ám a módszer megmaradt egyfajta átvezető technikának a valós színek tárolása és azok megjelenítése felé.

A leggyakrabban alkalmazott **24 bites RGB True Color**, azaz „valós” színskála esetében már 3x8 bit áll rendelkezésünkre. 8 biten 0-tól 255-ig lehetséges (decimálisan) elszámolni, ezáltal a vörös, a zöld és a kék is rendelkezhet bármelyik egész értékkel e két szám között. Könnyen kiszámolhatjuk ebből, hogy így összesen nagyjából 16,7 millió szín áll rendelkezésünkre. A 0,0,0 eredményezi a feketét, a 255,255,255 a fehéret, a 255,0,0 a teljesen piros színt, a 255,0,255 a lila és így tovább. Amikor a három szám megegyezik, vagy nagyon közel áll egymáshoz, biztosak lehetünk benne, hogy valamilyen szürke színt kell, hogy kapjunk.

Találkozhatunk még **32 bites** színmélységgel is, amely esetében szintén ugyanaz a színválaszték áll rendelkezésünkre, továbbra is 3x8 biten, de egyes esetekben a memória-hozzáférés szempontjából egyszerűbb a 4x8 bit alkalmazása. A negyedik, fennmaradó bitet szokás még az **alfa-csatorna**, azaz az átlátszóság mértékének meghatározására is felhasználni, ahogyan az az **RGBA** (*Red, Green, Blue, Alpha*) színskála esetében a gyakorlatban meg is történik.

Már a fenti, decimális formában is jól definiálhatóak a színek, de annak a lehetőségét is kihasználjuk, hogy a 255 tízes számrendszerbeli számjegy a tizenhatos számrendszerben három helyett még éppen felírható kettő karakterrel. Ennek köszönhetően a teljes RGB színkód hat darab **hexadecimális** számjegy kombinációjában is megadható. Például a lila szín hexa kódja így ff00ff, azaz 255 (ff) vörös, 0 (00) zöld és 255 (ff) kék. A legtöbb grafikai szoftver képes a hexadecimális színkódok értelmezésére és a színpalettáról hexadecimális kódba való azonnali átalakításra.

A kisebb méretű, önálló grafikai objektumokat tartalmazó képek publikálásra szánt változatait gyakran nem a teljes, 24 bites skála alkalmazásával mentik el, ugyanis lehetőségünk van kizárólag az alkalmazott színek megtartásával, önálló **paletta** felhasználásával, a képfájl színeit kevesebb biten tárolni. Az ilyen képeket **indexelt** képeknek hívjuk. Önálló grafikai objektumok lehetnek például az internetes oldalakon a gombok, a keretek elemei, a grafikus operációs rendszer asztalán látható

ikonok stb. Mivel az ilyen képfájlok mérete rendkívül kicsi, letöltésük pillanatok alatt végbemegy. Az indexelt képek többnyire mind 8 bitesek, de gyakoriak a 256-nál kevesebb színt tartalmazó változatok is. Az indexelt képeket a valós színeket tartalmazó eredeti képekből történő konvertálással készíthetjük el valamilyen képszerkesztő program segítségével. Mivel az egyes grafikai objektumok mögött külön képfájlok, azokon belül önálló színpaletták töltődnek be, a végeredmény egy teljes értékű, valós színekkel rendelkező felhasználói felület lesz.

A profi fotófeldolgozásban ma már gyakori a „16 bites” képek használata is, amely nem összetévesztendő a hagyományos, 16 bites színmélységgel, ugyanis ez a 16 bit csatornánként (színek, plusz az átlátszóság) értendő. Bár a bittérképes képformátumok legnagyobb része továbbra is „csak” a csatornánkénti 8 bites színmélységet támogatja, az igazán lenyűgöző részletességű képek képszerkesztő programban való feldolgozásánál esetenként szükség lehet a 16 bites képek alkalmazására.

## 2.2 Rasztergrafika

Már az előző fejezetekben is találkozhattunk a rasztergrafikus képfeldolgozás alapjaival, a képponttal, mint a rasztergrafikus kép alapegységével, illetve az annak megjelenítésével kapcsolatos alapfogalmakkal. A képpontok működéséből adódóan felmerül még néhány további, csak a raszteres képekre jellemző tulajdonság, amelyek megismerésével együtt a képeken elvégezhető műveletek, az alkalmazható módosítások korlátai is világosabbá válhatnak.

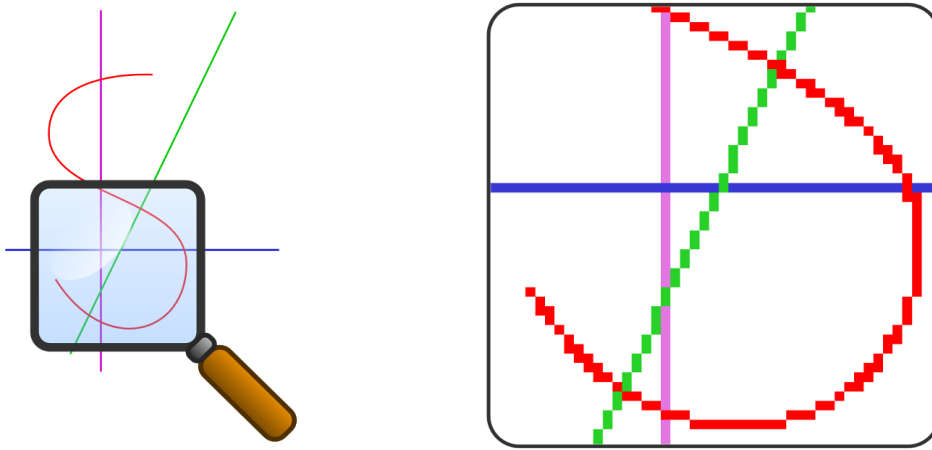
Ha már szót ejtünk az alapvető rasztergrafikus jellemzőkről és az elvégezhető műveletekről, akkor mindenképpen fontos, hogy a szintén szorosan idekapcsolódó képtárolási lehetőségeket se hagyjuk ki. Foglalkozni fogunk tehát továbbá még a raszteres képfájlok tömörítési módszereivel és a hosszú évek során egyre jobbra fejlesztett, majd napjainkban is használt képfájl formátumokkal is.

### 2.2.1 A rasztergrafikus modellezés alapjai

A raszteres kép képpontokból, másnéven pixelekből áll, modelltere egy kétdimenziós koordináta-rendszer, amely az oszlopaival és soraival egy szabályos rácshálót alkot. A képpontok a rácsháló egészértékű koordinátáinak felelnek meg. Amennyiben a képpontok mérete és egymástól való távolsága elegendően kicsi, a képpontokat már nem pontokként, hanem egybefüggő képként láthatjuk.

Nagyon lényeges tudnivaló, hogy a rastergrafikus kép csak az eredeti méretben képes biztosítani az elkészítéskor megkapott részletességet. Nagyítás során mindenképpen egy homályosabb, életlenebb képet kapunk, kicsinyítéskor a már eredeti képen is vékony vonalakkal szegélyezett, apróbb részletek tűnhetnek el. Mindkét esetben a grafikai szoftver a saját algoritmusaival próbálja a lehető legjobb képminőséget elérni, de mindennemű méretváltoztatás és a képen belüli részek transzformációja (például elforgatás) a minőség romlásához, a kép részletességének elvesztéséhez vezet.

A rastergrafikus modellezés során a szabályos geometriai formák rácshálón történő megjelenítésére keressük a lehető legjobban közelítő megoldásokat. A négyzet alakú képpontok miatt a raszteres térben csak a pontok, egyenesek és négyszögek és azok csakis tökéletesen vízszintes és függőleges vonalakkal történő rajzolása eredményez hibátlan, éles rajzolatot. Mindennemű görbe és elferdített vonal leképezése rastergrafikus algoritmusok bevezetését igényli, azaz a számítógépnek ki kell számítania, hogy a függőleges vagy vízszintes koordinátáik szerint nem megegyező képpontokat milyen útvonalon kösse össze. Ezt szemlélteti a 6. ábra is.

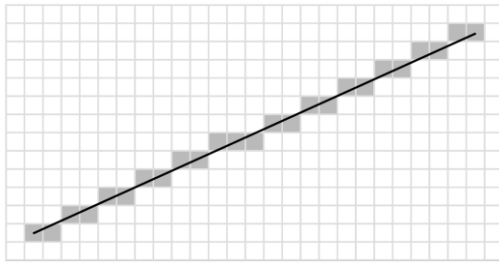


6. ábra

Láthatóan a nagyítás nélküli változatban teljesen elfogadható, éles vonalakat láthatunk és a valójában bevont képpontok csak erős nagyításban érhetőek tetten. Azokat az algoritmusokat, amelyek segítségével meghatározásra kerül, hogy két tetszőleges koordinátájú képpont összeköttetéséhez a rácsháló mely további képpontjainak bevonása szükséges, **digitális differencia-elemző algoritmusoknak** (DDA, azaz *Digital Differential Analyzer*) nevezzük.

A DDA algoritmusok alapjául Bresenham vonalkiszámító algoritmus szolgál, amely az egyes képpontok középpontjaiból képzett szakaszok

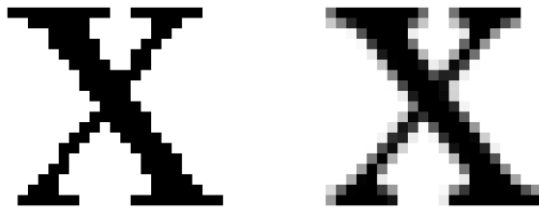
felezésével, a valós vonalhoz mindig jobban közelítő képpontot választja (7. ábra). Ma már több különféle, az eredetit kiegészítő és annak apróbb hiányosságait kiküszöbölő algoritmus is létezik, amelyeket a grafikus szoftver vagy közvetlenül a videokártya is elvégezhet.



7. ábra

## 2.2.2 Élsimító algoritmusok

A vonalkiszámító algoritmussal nagyon gyorsan meghatározhatóak a végpontokat összekötő képpontok, azonban az eljárás egy újabb problémát vet fel, mégpedig a ferde vonalak nem kívánt lépcsőzetességét, amely még nagyítás nélkül is viszonylag könnyen észrevehető. A kirajzolt alakzatok megjelenését javíthatjuk, ha az éleken végigfutó képpontokat valamilyen szinten elhalványítjuk, illetve további szomszédos képpontokat vonunk be, ahogyan azt a 8. ábrán is láthatjuk. Ezeket a módszereket *élsimításnak*, angol kifejezéssel élve *anti-aliasing*-nak nevezzük.



8. ábra

A legrégebbi élsimító algoritmus a *supersampling* nevű technika. Ennek során minden egyes képpontot ideiglenesen felbontanak úgynevezett subpixelekre, de az eredeti pixelhez képest mindig egy kicsit eltolva, majd az eredményt összemossák a kiszámított subpixelek értékeinek átlagolásával.

A supersampling mellett még számos további élsimító algoritmus létezik. Valamelyikük lényegesen gyorsabban még elégséges eredményt hoz és van, amelyekkel látványosan jobb eredményt érhetünk el, de csak nagy számítási teljesítmény árán, hosszabb feldolgozási idő után. Bizonyos helyzetekben, például a 3D-s számítógépes játékok esetében az is fontos szempont lehet,

hogy az algoritmus inkább igényeljen nagyobb memória sáv szélességet, mint számítási teljesítményt, vagy fordítva. A megfelelő algoritmus megválasztása mindig az adott feladattól függően kerül kiválasztásra.

### 2.2.3 Képtömörítési eljárások

A rastergrafikus képfájlokhoz szükséges tárolókapacitás elsősorban a felbontástól és a színmélységtől függ. Még a közepesen nagy felbontású, 24 bites raszteres képek is akár sokszor 10 MB tárhelyet vehetnek igénybe tömörítés nélkül, amelyek amellet, hogy a helyet foglalják, visszaolvasásuk és mentésük is időbe telik. A tárhely optimális kihasználása szempontjából mindenképpen kézenfekvő volt a különféle képtömörítési módszerek kidolgozása, de ezen túlmenően egyes képtömörítési eljárások az adatátviteli hibák kiküszöbölését és adattitkosítást is végezhetnek.

A digitális képek tömörítését a különféle *képi redundanciák* teszik lehetővé. Redundanciának nevezzük, amikor a közlésben az egyértelmű megértéshez elegendő minimumon felül, többlet továbbítódik. Az informatikában, egyes esetekben szükség lehet a többletre (például redundáns, azaz biztonsági okokból megduplázott tápegység egy szerverben), más esetekben viszont a többlet felesleget jelent. A rastergrafikus képfájl redundanciájának csökkentésével a valóban elmentett fájl már sokkal kisebb tárhelyen elfér, majd visszatöltéskor, a tömörítés visszafejtésével újra az eredeti képet kapjuk meg. Alapvetően három redundanciátípust különböztetünk meg képek esetében:

- **Kódolási redundancia:** amikor kevesebb színt tartalmazó színtér is elegendő lenne egy nagy színmélységű, több szín tárolására is alkalmas képen.
- **Képi redundancia:** amikor a kép egy-egy nagyobb területén a képpontok ugyanolyan színnel rendelkeznek, vagy a kép egyes rétegei között csak apró eltérések vannak, mégis minden képpont tulajdonsága egyenként rögzítésre kerül.
- **Pszichovizuális redundancia:** amikor olyan színek, árnyalatok vagy képi elemek is rögzítésre kerülnek, amiket az emberi szem egyébként nem is képes érzékelni és hiányuk általában csak erős nagyításban lenne észrevehető.

A képtömörítési módszerek két nagy csoportba oszthatóak: *veszteséges* (*lossy*) és *veszteségmentes* (*lossless*) tömörítési eljárások. A veszteséges tömörítésnél a pszichovizuális redundancia csökkentését alkalmazzák, míg a veszteségmentes esetében a kódolási és képi redundancia által biztosított



felesleges többletet használják ki. A veszteség vizuálisan értendő, azaz a veszteséges tömörítések alkalmazásakor az eredeti kép valamilyen szinten torzul, a korábban tökéletesen egyszínű területek összemaszatolódnak. A tömörítés mértéke, ezáltal a torzulás veszélyének csökkentése természetesen állítható, ám kisebb tömörítéssel jelentősen nagyobb fájlméretet kapunk. Veszteségmentes tömörítés alkalmazásával kisebb eséllyel érhetünk el igazán kis fájlméretet, azonban a raszteres kép minden képpontja az eredetinek megfelelő marad. A 9. ábrán láthatjuk egy kép veszteségmentes (bal) és veszteséges (jobb oldali) tömörítésének eredményét nagyításban:



9. ábra

A veszteséges tömörítési eljárásokkal rendkívül kis képfájlokat nyerhetünk, a kapott fájl mérete többnyire az eredeti, tömörítetlen képfájl tört részére esik vissza. Digitális fényképek esetében például ideális használatuk, mivel a valóság leképezésével nyert, nagy részletgazdagságú képen a tömörítésből adódó apró torzulások a kép eredeti méretében szinte egyáltalán nem érzékelhetőek. Érdekes tehát az alkalmazásuk, de többnyire csak azzal a megköttéssel, hogy egynél több alkalommal lehetőleg ne mentsünk egy már veszteséges tömörítés használatával elmentett képet. Generált számítógépes grafikák esetében is mindenképpen ajánlott az eredeti, valamelyik veszteségmentes tömörítést használó képfájl formátumban is megtartanunk művünket.

## 2.2.4 Rasztergrafikus képfájl formátumok

A képfájl típusok általában egy fejlécből és az egyes képpontokat definiáló bittérképet tartalmazó adatrészből állnak. A fejlécben a fájl típusra vonatkozó adatok, a képleíró bejegyzések (mint például a vízszintes és függőleges képpontok száma) és esetlegesen a használt színpaletta található.

A legjobb kiindulópont a képformátumok megismeréséhez a BMP (*Bitmap*, azaz bittérkép) formátum, ugyanis alapesetben, önálló paletta nélkül, csak nagyon kicsi fejléccel rendelkezik, továbbá az adatrészában

minden egyes képpont számára a színmélységnek megfelelő bitet tartja számon. Ennek megfelelően egy 100x100 képpont felbontású, 24 bites színmélységgel elmentett képfájl 30 054 bájt méretű lesz. Mivel egy-egy képpont 24 / 8, azaz 3 bájtot foglal, így könnyen kiszámolhatjuk, hogy a fejléc mérete  $30\ 054 - (100 * 100 * 3) = 54$  bájt helyet, az adatrész pedig 30 000 bájtot igényel.

Rengeteg raszteres képfájl formátum létezik, közülük csak azokat vesszük sorba, amelyeket érdemes ismernünk, illetve amelyekkel minden bizonnyal még találkozni fogunk:

- **BMP:** *Bitmap*, egy Microsoft által kidolgozott formátum, kismértékben használhatunk benne tömörítést is, amely veszteségmentes. Az 1 bites monochrome-tól a 32 bitesig mindegyik színmélységgel használható, ismeri az indexelt színkezelést, önálló színpalettája lehet és az átlátszóság tárolására is alkalmas az *alpha channel* használatával.
- **PCX:** A *ZSoft* szoftvercég által fejlesztett *PC Paintbrush* program korai verziójában az alapértelmezett formátum. Ismertté azáltal vált, hogy a programot a Microsoft felvásárolta és a Windows részévé tette. A PCX újabb verziója a 24 bites színmélységet és az átlátszóságot is támogatja (bár ez utóbbit nem minden grafikai szoftver képes kezelni), veszteségmentes tömörítést használ.
- **JPG** vagy **JPEG:** *Joint Photographic Experts Group*, veszteséges tömörítést használó, kiváló tömörítési hatékonysága miatt a legelterjedtebb formátum a digitális fényképek tárolásában. A maximálisan elérhető képfelbontásban 65535x65535 képpontot tartalmazhat, főként a szürkeskála és RGB (24 bit) színmélységeket alkalmazhatjuk esetében és nem támogatja az átlátszóságot.
- **GIF:** *Graphics Interchange Format*, animációk tárolására is alkalmas képfomátum. Veszteségmentes tömörítést használ és legfeljebb 256 szín, azaz 8 bites színmélység vagy szürkeskála használatát teszi lehetővé. Az animációk tárolását úgy valósítja meg, hogy egyetlen képfájl több réteget, ezáltal több képet is tartalmazhat. A képek közötti váltás időtartama, valamint a folyamatos ismétlődés ki/be kapcsolása mentéskor határozható meg. Az interneten nagyon gyakori formátum, a weboldalak (főként animált) grafikai elemeinek (például gombok) tárolásában. A fájlípust 1987 óta használják.
- **PNG:** *Portable Network Graphics*, egy fiatal, 1995 óta fejlesztett, veszteségmentes tömörítést használó formátum. A legtöbb ismert színmélységet támogatja, használhatunk benne szürkeskálát, indexelt képet és átlátszóságot is. Az *Adam-7* algoritmus használatával lehetőség

van a nagyobb felbontású képek fokozatos megjelenítésére is először kis felbontásban, majd a teljes letöltéskor a legjobb minőségben. Számos operációs rendszerben ikonként is használhatunk PNG képeket. Napjainkra az egyik legelterjedtebb képformátummá vált.

- **TGA:** *Truevision TGA (Targa)*, eredetileg szintén egy képszerkesztő program formátuma volt. Egyszerűsége, ezáltal gyors visszakódolhatósága miatt napjainkban is széles körben alkalmazzák például videofájlokban vagy számítógépes játékokban a textúrák tárolásához. A színmélységek a 8 bitestől a 24-ig választhatók, támogatja az átlátszóságot is. A mentés során ki/be kapcsolható veszteségmentes tömörítési módszere kevésbé hatékony fényképek és egyéb nagy részletgazdagsággal bíró képek esetében, de jól teljesít ábráknál, vonalrajzoknál.
- **TIFF:** *Tagged Image File Format* (.tif fájlnev-kiterjesztéssel). A formátum többféle tömörítési eljárás alkalmazását is lehetővé teszi, így néhány veszteségmentes tömörítési algoritmus mellett a JPEG, veszteséges tömörítést is választhatjuk mentéskor, ez teszi ezt a képformátumot rendkívül rugalmassá. Szintén támogat minden ismert színmélységet és az átlátszóságot.
- **RAW:** A formátumot gyakran hívják digitális negatívként is, ugyanis a RAW fájl típus mentésére alkalmas fényképezőgépekben, ennek használatával, a gép érzékelőjéből a lehető legminimálisabb feldolgozással kerülhetnek a képek közvetlenül mentésre. Kimaradnak tehát mindennemű színcorrekcióval és JPEG tömörítéssel együtt járó esetleges hibák, az eredményül kapott fájl viszont nem kezelhető még azonnal képfájlként, további szoftveres feldolgozást igényel. Minden fényképezőgép gyártónak megvannak a saját maga által preferált RAW formátumai, ezért nem beszélhetünk egységes megoldásról. A legtöbb RAW formátum veszteségmentes tömörítést használ, de léteznek veszteséges, de nagyon jó minőségben tömörítő változatok is.
- **XCF:** *eXperimental Computing Facility*, amely a *GIMP* képszerkesztő program és az ahhoz kapcsolódó grafikus függvénytár fejlesztését elkezdő kutató-fejlesztői csoport neve. Az .xcf képfájl formátum a *GIMP* saját, natív formátuma, így veszteségmentes tömörítéssel képes tárolni a kép valamennyi adatát, beleértve a rétegeket, a csatornákat, az útvonalakat, az átlátszóságokat, sőt még az aktuális kijelölést is. Az előző formátumokkal ellentétben az XCF már nem alkalmas csereformátumként való alkalmazásra, mivel ezt csak a *GIMP*, annak más verziói és még néhány további képszerkesztő program (korlátozott módon) képes megfelelően kezelni.

- **PSD:** *Photoshop Document*, az *Adobe* cég *Photoshop* képszerkesztő programjának saját, natív formátuma. Szintén minden olyan adat elmentésére alkalmas (rétegek, átlátszóságok, maszkok stb.), amelyek a visszatöltés után a munka folytatásához szükségesek. Az elkészült képfájlt szintén csak a célprogram képes minden szolgáltatással együtt használni. Igazán nagy, akár maximum 300 ezer x 300 ezer képpont felbontású képek szerkesztéséhez a *Photoshop* egy külön formátummal, a *PSB*-vel (*Photoshop Big*) is rendelkezik.
- **PSP** vagy **PSPIMAGE:** a *Paint Shop Pro* képszerkesztő program saját, natív formátuma 2004-ig a *Jasc Software*, majd attól kezdve a *Corel* gondozásában.

Egyes rasztergrafikus képszerkesztő programok néha tartalmaznak bizonyos vektorgrafikus rajzelemeket is (például útvonalak készítésére alkalmas eszközökkel rendelkeznek), sőt ezen túlmenően például a *Photoshop* és *Paint Shop Pro* képesek a vektoros típusú rétegek kezelésére és saját, natív fájlformátumukon belüli megőrzésére.

Mindegyik általános raszteres képfájlról (például *BMP*, *PNG*, *JPG*) elmondható, hogy a képen belül az egyes grafikai objektumok egyenként nem szerkeszthetőek újra egyszerűen. Amennyiben egy mentést és bezárást követően további módosításokat szeretnénk végezni, akkor az újbóli megnyitás után ismét a teljes képet kell szerkesztenünk és egy további mentéssel a teljes képfájlt felülírunk. Erre a problémára nyújtanak megoldást a képszerkesztő programok a rétegek, útvonalak és a speciális, vektoros eszközök által, illetve a feliratok szövegobjektumként való kezelhetőségével. Emiatt különösen fontos, hogy munkánkat mindig tartsuk meg az adott képszerkesztő program saját, natív képfájl formátumában is. Sőt, még a natív képfájlokról is érdemes a nagyobb változtatások előtt biztonsági másolatot készítenünk, és végül csak a kész képet elmentenünk illetve exportálnunk valamelyik köztes formátumba.

## 2.2.5 Képek konvertálása

Lehetőségünk van ugyanazt a képet különféle képfájl típusok között elmenteni, ezáltal *konvertálni*. Ehhez olyan grafikai szoftverre van szükségünk, amely legalább mindkét fájltypust támogatja. Az eredeti, konvertálandó képfájl formátum esetében már az is elegendő, ha az adott grafikai szoftver legalább megfelelően beolvasni képes azt. Ilyen grafikai szoftverek lehetnek a különféle képszerkesztő vagy képnézegető, képrendszerező alkalmazások.

A képszerkesztő programba az adott szoftver mindennemű, saját formátumától eltérő fájltypus beolvasását *importálásnak* nevezzük. Importálásakor a grafikai program a beolvasott képet átalakítja a saját, belső formátumának megfelelően. Ha ezt követően a képet már a belső formátumban mentjük, akkor egyszerűen mentést, ha ismét egy idegen típusú választunk, *exportálást* végzünk.

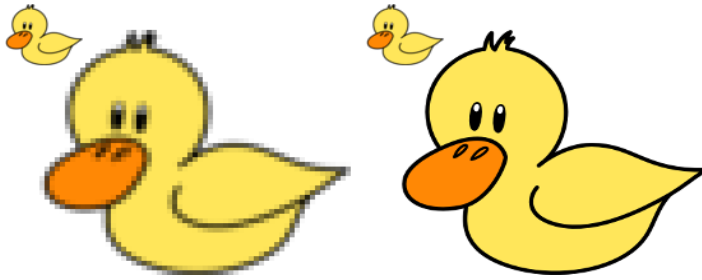
Az egyes képfájl típusokba való konvertálásakor többnyire meghatározhatjuk a tömörítés mértékét is, amely veszteségmentes tömörítéseket alkalmazó fájltypusok esetében fordított arányban áll a visszatöltés sebességével, veszteséges tömörítéskor pedig a kép minőségével. Például a PNG használatával mentéskor 10 féle tömörítési szint választható ki a 0-tól (nincs tömörítés) a 9-ig (maximális tömörítés), amely a visszakódolás sebességét, azaz a megnyitásra fordított időt nagyban befolyásolja. Hasonlóképpen a veszteséges tömörítésű JPG esetében a nagyobb tömörítési mérték rosszabb minőségű képet eredményez. Éppen emiatt a JPG-be való exportálás (azaz konvertálás) során a legtöbb programban nem is a tömörítés mértékét, hanem a képminőséget állíthatjuk be százalékosan.

## 2.3 Vektorgrafika

A számítógépes grafika ezen ágának esetében egy teljesen más jellegű technikáról és megközelítésről beszélhetünk. A vektorgrafikáról nem jelenthető ki, hogy jobb vagy rosszabb lenne, mint a rasztergrafika. Napjainkban bizonyos feladatok egyszerűen már nem lennének megoldhatóak vektorgrafika használata nélkül, számos feladat elvégezhető mindkét módszerrel is, míg megint mások csak raszteres módszerekkel oldhatóak meg. Emellett bármilyen vektorgrafikus képpel is rendelkezünk, annak a képernyőn történő megjelenítése csakis annak a háttérben történő raszteressé alakításával történhet meg a korábbi fejezetekben már említett folyamat során.

A rasztergrafikáról már tudjuk, hogy modellterében csupán rengeteg egymás mellett szabályosan elhelyezkedő képpont alkotja meg a képet. A képpontok között nincsen valódi kapcsolat, a végső képen azok nem alkotnak egy-egy közös objektumot, amellyel a későbbiekben még további műveleteket hajthatnánk végre. Teljesen másképpen történik ez a vektorgrafika esetében, melynek alapja, hogy a kép minden egyes része egy-egy önálló rajzobjektumhoz kell, hogy tartozzon. Ezeket az objektumokat tetszés szerint nagyíthatjuk, nyújthatjuk, forgathatjuk, egymásba vonhatjuk stb. Nevéből is kikövetkeztethetően, a vektorgrafikában a geometriai alakzatokat adott koordinátájú kiindulópontokból egy-egy meghatározott méretű és irányú vektorral építjük fel. A legegyszerűbben a raszter- és vektorgrafika közti különbségre egyértelműen rávilágítani a

nagyíthatósággal szokás (10. ábra): a raszteres kép csak minőségromlással nagyítható, miközben a vektorgrafikus kép méretét bármekkora növelhetjük, az élek vagy határvonalak akár egy futballpálya méretűre nagyított képen is tökéletesen egyenesek maradnak.



10. ábra

Bár a tetszőleges nagyíthatóság gondolata egy rendkívül meggyőző érv lehetne a vektorgrafika használata mellett, a túl sok részlettel rendelkező képeket, így például a fényképek többségét nem lehetséges a részletek teljes megtartása mellett vektorossá alakítani. Ennélfogva a kétdimenziós vektorgrafika általában a generált képek, logók és illusztrációk elkészítése során az ideális választás.

### 2.3.1 A vektorgrafikus modellezés alapjai

A vektorgrafika fogalmát elsősorban a kétdimenziós grafikával kapcsolatban használják, de a különféle háromdimenziós modellező programok szintén vektorosak. A 3D grafikában, a térbeli modellezés során szintén alapkövetelmény az objektumok külön-külön történő elkülöníthetősége, azok független transzformálhatósága, tulajdonságaik változtathatósága. A különbség az alkalmazható objektumok és a modellter kiterjedésében keresendő, továbbá a 3D-s térben az objektumok leképezése nem csak egy, hanem elméletben végtelen irányból történhet.

A kétdimenziós vektorgrafikus modellterben az rajzobjektumok kezdőpontját a vízszintes X és a függőleges Y tengely által határolt koordináta-rendszerben határozzuk meg, ahol a két tengely találkozási pontja az *origo*. A tengelyek alkotta modellterben egy pontot a hozzá vezető *helyvektorral* lehet azonosítani. A helyvektor koordinátái lebegőpontosak, azaz a rasztergrafikával ellentétben itt nem vagyunk csak egész számokkal meghatározható koordináták közé szorítva. A lebegőpontos számokat alkalmazó *Descartes* koordináta-rendszereket *világkoordináta-rendszereknek* nevezzük. A helyvektorok valós számokkal való meghatározása lehetővé teszi, hogy az objektumokat a modellter tetszőleges pontjában elhelyezzük. Mivel az objektumok tulajdonságait a vektorgrafikus képfájl belső *adatbázisában*

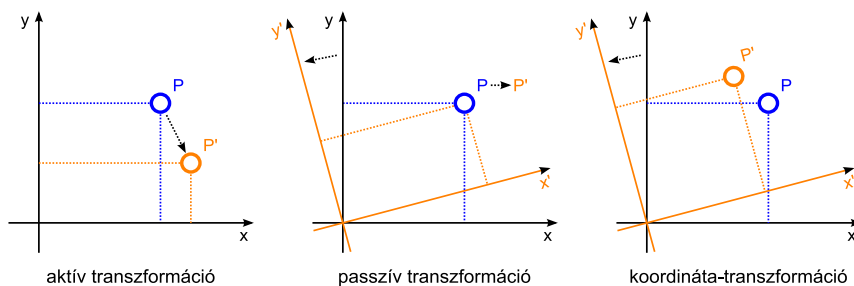
tároljuk, így minden objektum egyedileg visszakereshető, transzformálható, az egyes objektumok között strukturális kapcsolatok alakíthatóak ki vagy akár a meglévő objektumokból újabbak is létrehozhatóak.

Az egyes geometriai objektumok strukturális kapcsolatrendszerére többrétű lehet. Egyrészt a modellter minden objektuma között *alá-fölérendeltségi viszony* áll fenn, amely a kétdimenziós modellterben alapértelmezés szerint hátulról előre halad, de a sorrend bármikor tetszőlegesen megváltoztatható. A hierarchikus kapcsolat mellett lehetséges különféle *mellérendeltségi viszony* kialakítása is, amikor az egyes objektumok szerkezetüket, illeszkedésüket vagy megjelenésüket tekintve kapcsolódnak egymáshoz.

A lebegőpontos helyvektorok miatt a vektorgrafikus modellterben az objektumok egymáshoz történő pontos igazításához egyéb eszközöket kell igénybe vennünk. Ehhez elsődlegesen a vektoros képszerkesztők esetében többnyire rendkívül fejlett, igazító és elrendező kezelőpanel áll a rendelkezésünkre. Másrészt, az objektumok köré segédvonalakat (*guides*) vagy a teljes dokumentumra kiterjedően, nagyon aprólékosan beállítható rácshálót (*grids*) is alkalmazhatunk. Mindezek mellett még így is előfordulhat, hogy az egyes koordinátákkal muszáj egyszerűbb számításokat elvégeznünk, hogy az objektumokat esztétikusan, ugyanakkor nagy pontossággal elhelyezhessük.

### 2.3.2 Transzformációk

A vektorgrafikus modellterben többféle transzformáció alkalmazható, melyeket a matematika, a fizika és a gépészet területén is többféleképpen definiálnak, de eltekintve ettől, alapjaikban véve a transzformációk három jól elkülöníthető csoportba oszthatók:



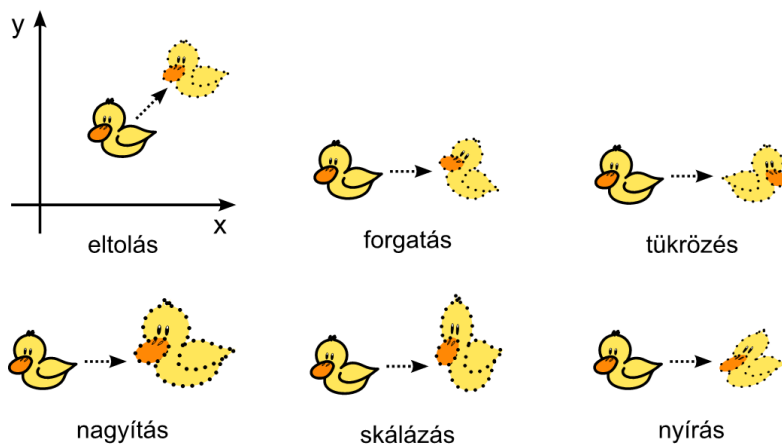
11. ábra

- **Aktív transzformáció** esetében az objektum egyes tárgypontjainak pozíciója változik meg, amely akár a teljes koordináta rendszer nélkül is érzékelhető. A matematikában meghatározott ponttranszformációk (*linear transformation*) aktív transzformációnak minősülnek, a nézőpont

mindvégig megmarad rögzített állapotában (az ezt szemléltető ábrán épp egy eltolásos transzformációt láthatunk).

- **Passzív transzformációról** beszélhetünk, amikor a koordináta-rendszer helyzete változik meg, miközben az objektumok tárgypontjai rögzítve maradnak, ezáltal azok az új, megváltozott rendszerben már új koordinátákat kapnak.
- **Koordináta-transzformációnak** minősül, amikor egy másik koordináta rendszerbe való váltás történik, miközben az új értékek az eredetieknek megfelelőek kell, hogy legyenek. Az objektumok tárgypontjai egymáshoz és a koordináta rendszerhez képest is változatlan pozícióban maradnak, csupán az új rendszerhez idomulnak, nézőpontváltás történik.

Az aktív transzformációknak azon csoportja, amelyek során az objektum tárgypontjai közötti párhuzamos egyenesek a transzformáció végeztével is párhuzamosak maradnak, **affin transzformációknak** nevezzük. Ezen transzformációk csoportján belül továbbiakat is megkülönböztetünk:



12. ábra

- **Egybevágósági transzformációknak** tekinthetjük azokat a módosításokat, amelyek során az objektum összes pontja közötti távolság, az azok között bezárt szögek és az objektum méretaránya sem változik. Ide tartozik az **eltolás** (*translation*), a **forgatás** (*rotation*) és a **tükrözés** (*flipping* vagy *mirroring*).
- A **hasonlósági transzformációk** során a pontok által bezárt szögek, a méretarányok, ezáltal az objektum alakja nem változik, viszont a méretarány megtartásával a teljes objektum mérete igen. Ezek a műveletek a **nagyítás** és a **kicsinyítés**, avagy az arányos átméretezések.