

Az oktató:

Palágyi Kálmán
SZTE, Képfeldolgozás és Számítógépes
Grafika Tanszék
6720 Szeged Árpád tér 2.
214-es szoba (tetőter)

(62) 546 197
palagi@inf.u-szeged.hu
www.inf.u-szeged.hu/~palagi

Kurzusanyagok

Digitális képfeldolgozás (BSc)
pub/Digitalis_kepfeldolgozas

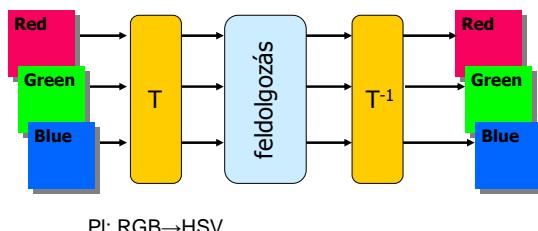
Képfeldolgozás haladóknak (MSc)
pub/Kepfeldolgozas_haladoknak

Színes képek feldolgozása

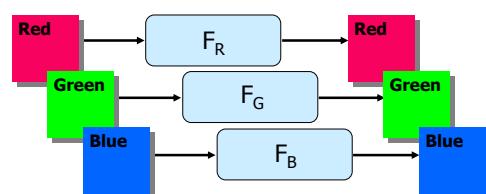
Feldolgozási módok

- alkalmas színmodellre való áttéréssel
- skaláris (komponensenkénti)
- vektoros

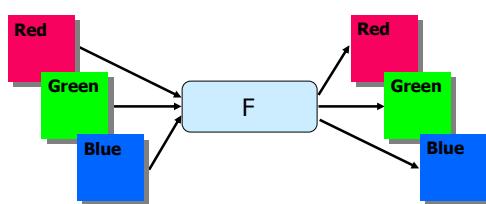
Áttérés alkalmasabb színmodellre



Skaláris feldolgozás



Vektoros feldolgozás



Témakörök

- Kvantálás
- Pont-operációk, hisztogram-transzformációk
- Szűrés, simítás
- Éldetektálás
- Szegmentálás
- Kódolás, tömörítés

Kvantálás

Egy true-color (3×8 bits/pixel, RGB) képen $256^3 = 16777216$ szín fordulhat elő.

A színes képek kvantálásakor a cél az, hogy „kevés” színnel adjuk/jelenítsük meg a képeket „számottevő” minőségromlás nélkül.

A kvantáláshoz meghatározandó:

- a színek egy részhalmaza (*colormap, palette*) és
- egy leképezés az eredeti színhalmazból az adott részhalmazba.

true-color



16-szín

8-szín



4-szín



A kvantálás menete

1. Végezzünk statisztikai elemzést az eredeti képre.
2. A statisztika alapján válasszuk ki a colormap-et.
3. Határozzuk meg valamennyi eredeti szín colormap-beli reprezentánsát (adjuk meg a leképezést, a CLUT-ot (*color look-up table*)).
4. Képezzük a kvantált képet.

A kvantálás alapkategóriái

Az eredeti színtér régiókra bontása (ahol az egy régióból eső színeket a colormap ugyanazon színébe képezzük le) az alábbi 2 módon történhet:

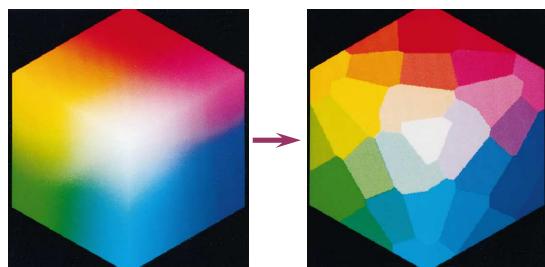
Uniform (képfüggetlen):

az eredeti színtér egyformája méretű régiókra bontjuk.

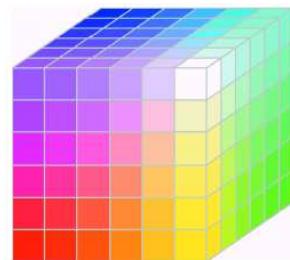
Nem uniform (képfüggő):

az eredeti színtér felosztása függ a színek eloszlásától az adott képen.

A kvantálás mint klaszterezés

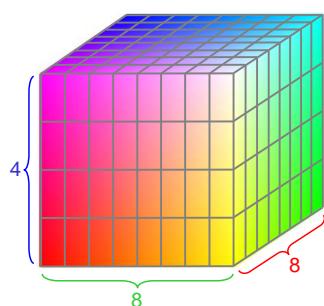


Uniform kvantálás – példa



Az RGB kocka felbontása
 $6 \times 6 \times 6 = 216$ színre/régióra
(*Web-Safe Colors*)

Uniform kvantálás – példa



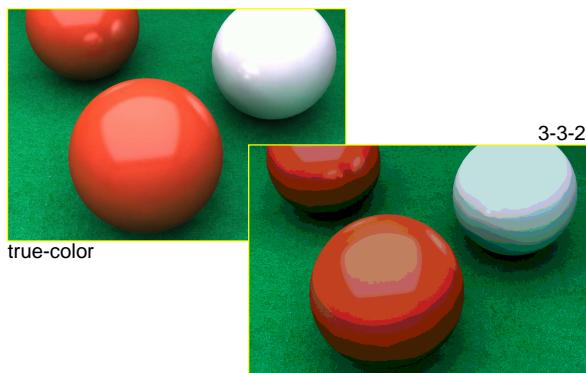
A true-color RGB kocka felbontása $8 \times 8 \times 4 = 256$ színre/régióra
(az R- és a G-komponens 3-3-bites, a B-komponens 2-bitben ábrázolt)

Uniform kvantálás – példa



3-3-2

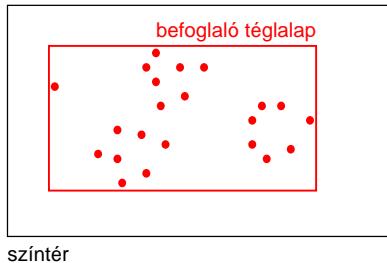
Uniform kvantálás – példa



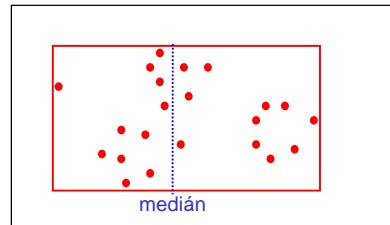
Nem uniform kvantálás – Center/Median Cut algoritmus

1. Határozzuk meg a színtérben a képen előforduló színek befoglaló téglalapot.
2. Rendezzük a színeket a leghosszabb doboz-oldal tengelye mentén.
3. Vágjuk szét az adott dobozt a mediánnál.
4. Ismételjük az eljárást a 2. lépéstől, amíg el nem értük az előre adott doboz-számot (pl. 256-ot).
5. Rendeljük minden dobozhoz az átlagszínét.

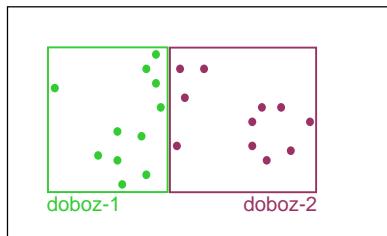
Median Cut algoritmus – 1.



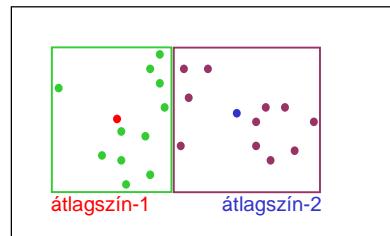
Median Cut algoritmus – 2.



Median Cut algoritmus – 3.



Median Cut algoritmus – 5.



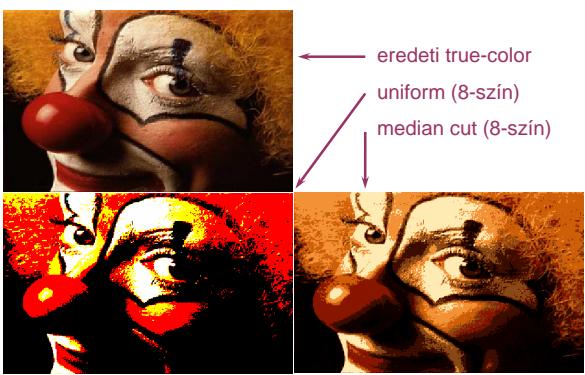
Median Cut algoritmus – példa



Median Cut algoritmus – példa



Median Cut ↔ uniform



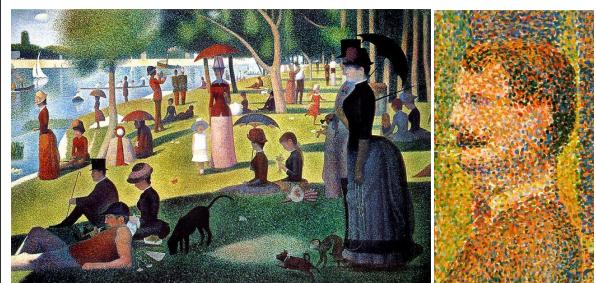
További módszerek kvantálásra és a kvantálási hiba csökkentésére

- Octree method
- Self-Organizing Map
- K-means
- Fast Adaptive Dissection
- Dithering, error diffusion halftoning
- ...

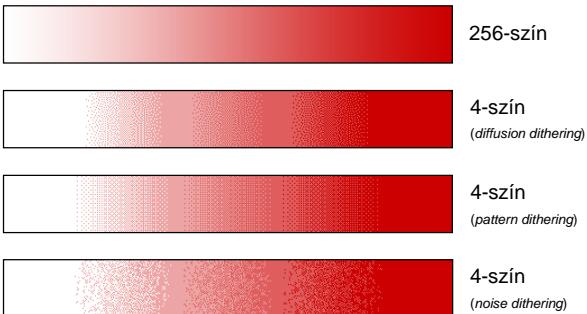
Octree algoritmus – példa



Dithering – Seurat



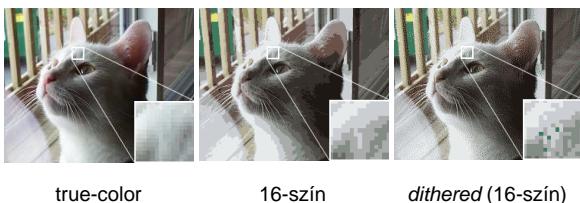
Dithering – példa



Dithering – példa



Dithering – példa

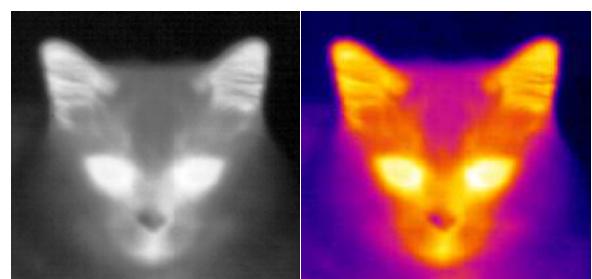


Error diffusion halftoning – példa

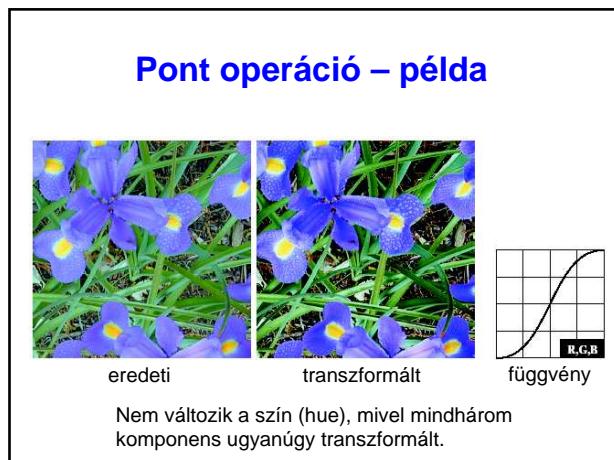
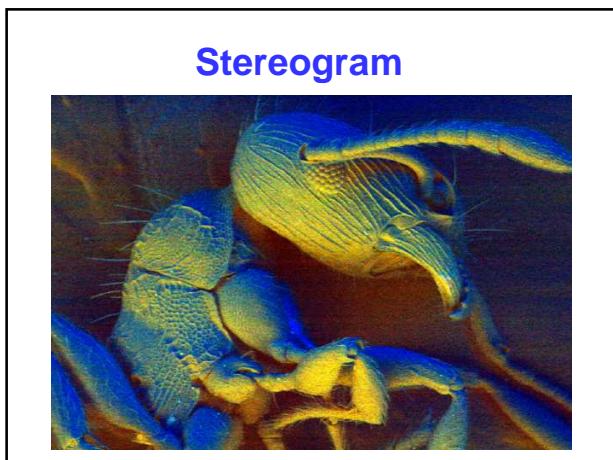
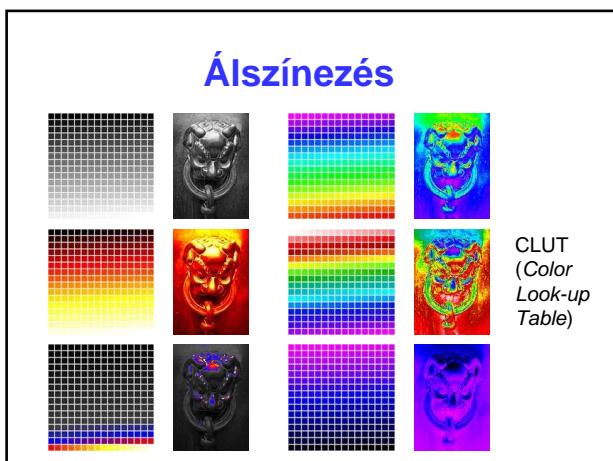
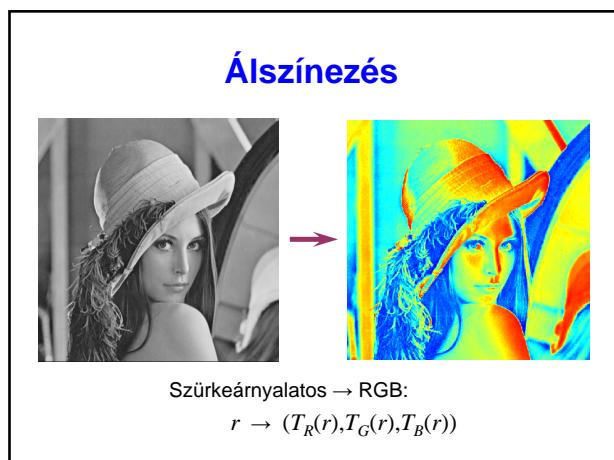
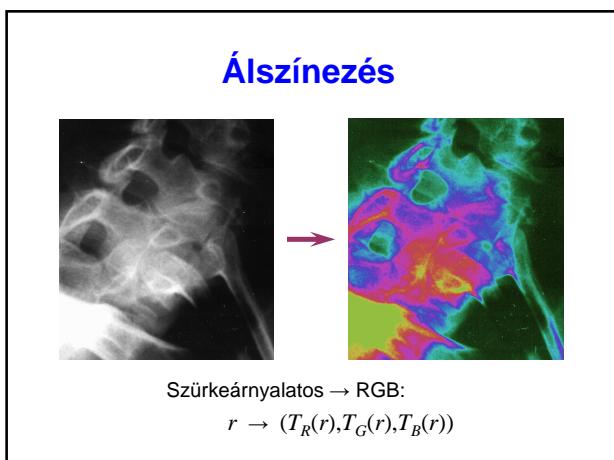


Pont-operációk, hisztogram transzformációk

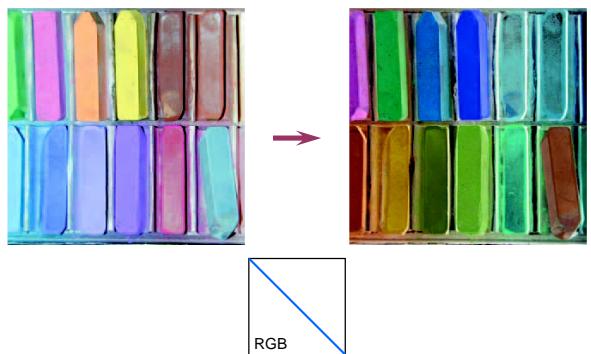
Álszínezés



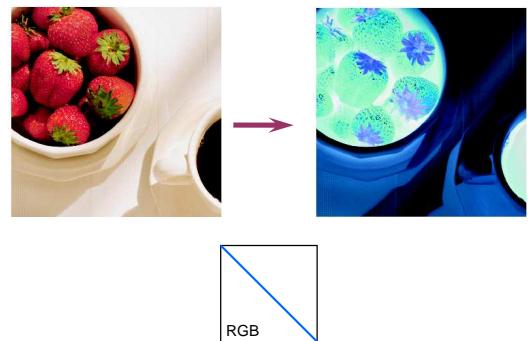
Szürkeárnyalatos → RGB:
 $r \rightarrow (T_R(r), T_G(r), T_B(r))$



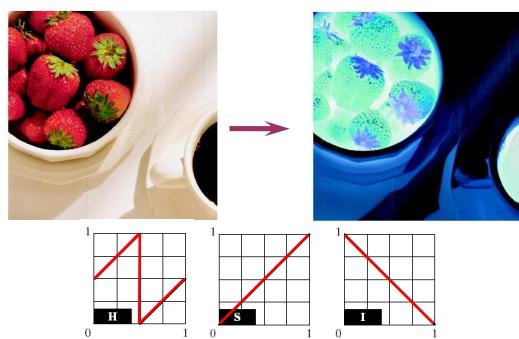
Invertálás / negálás (RGB)



Invertálás / negálás (RGB)



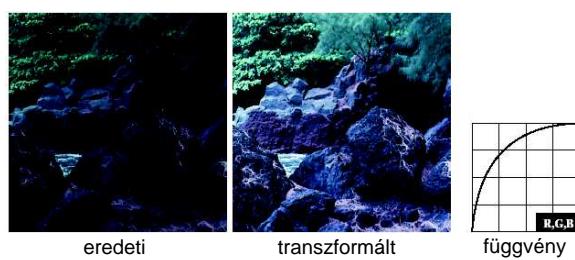
Invertálás / negálás (HSI)



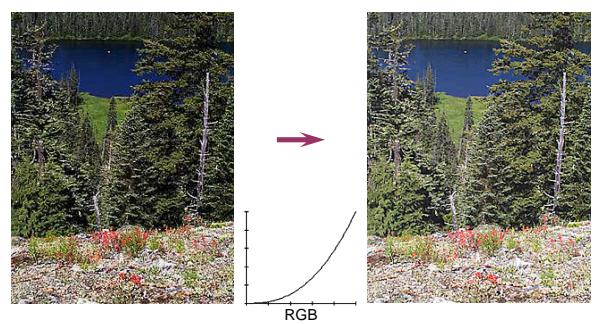
Pont operáció – példa

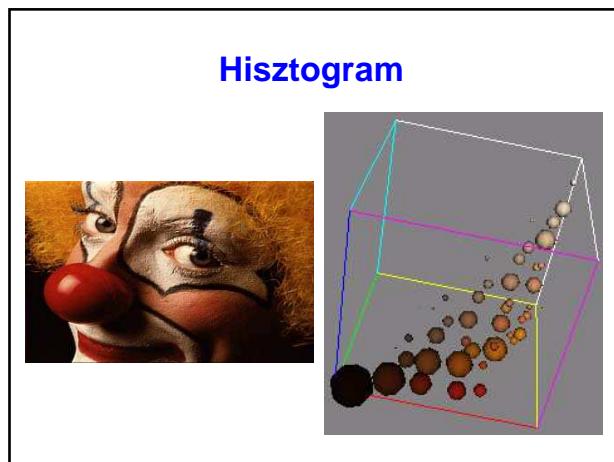
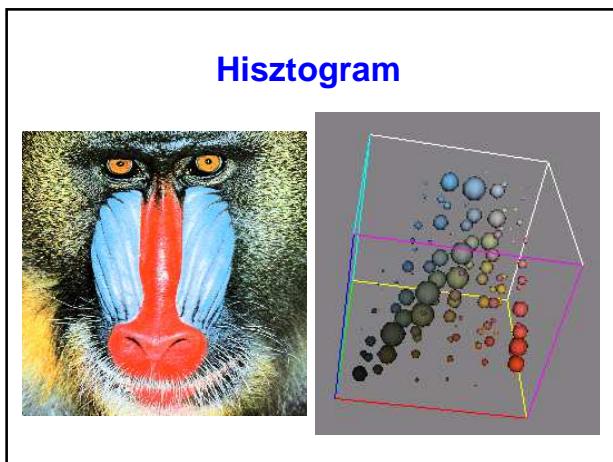
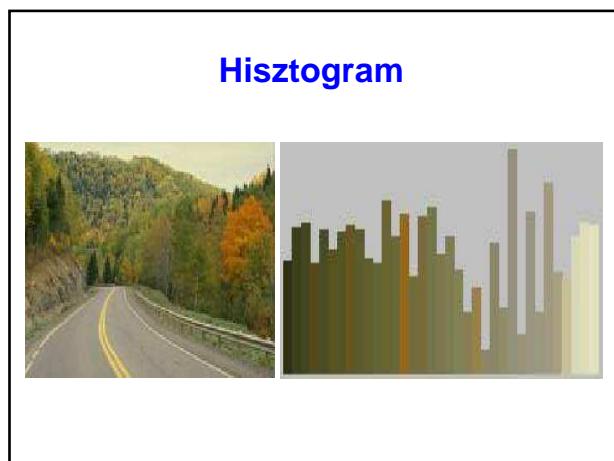
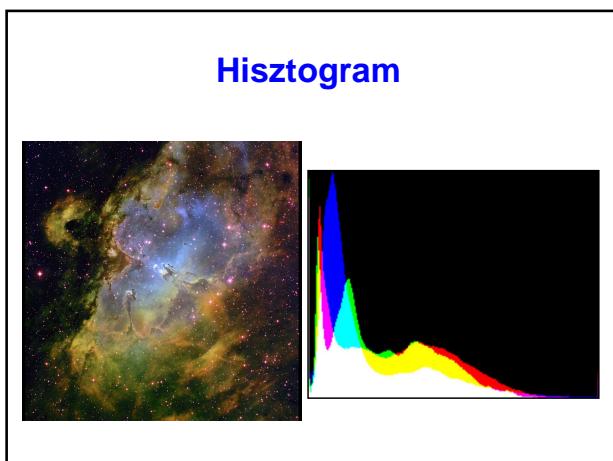
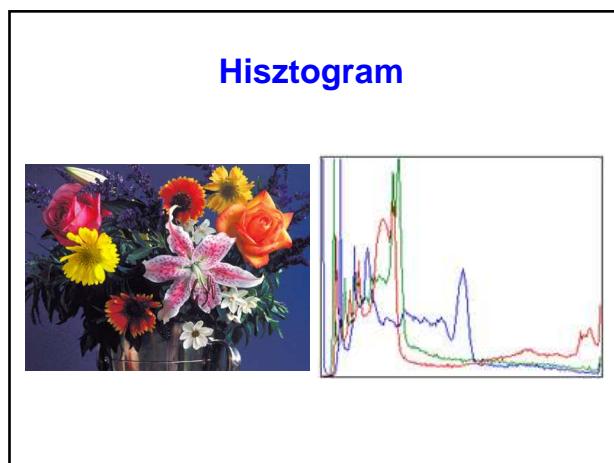
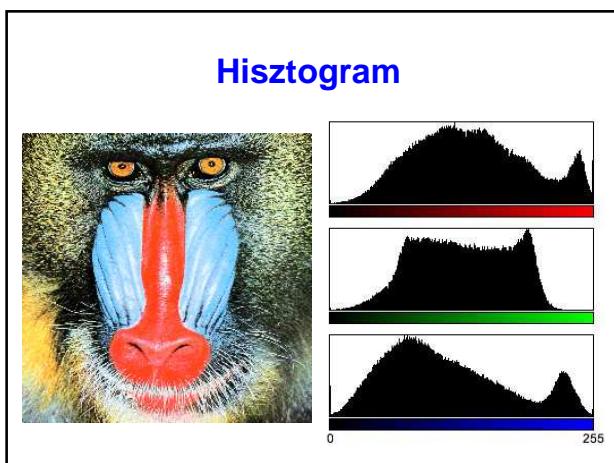


Pont operáció – példa



Gamma korrekció





Hisztogram kiegyenlítés

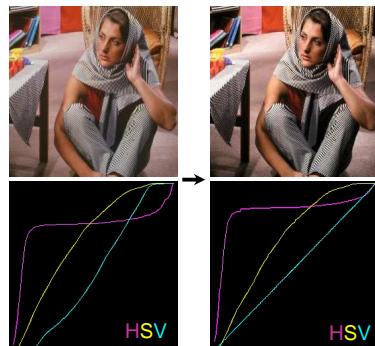


eredeti sötét kép

hisztogramkiegyenlítés
egyenként az R-, a G-
és a B-csatornákon

hisztogramkiegyenlítés
az YIQ-ra konvertált
kép Y (*lumiance*)
komponensén

Hisztogram kiegyenlítés



... a HSV
V-komponense
alapján
(képek és
felhalmozott
hisztogramjaik).

Hisztogram kiegyenlítés



eredeti HSI

kiegyenlítés a *Saturation*
komponens alapján

Szűrés, simítás

Konvolúció

színes (RGB) kép:

$$f = (f_R, f_G, f_B)$$

konvolúciós maszk:

$$g = (g_R, g_G, g_B)$$

konvolvált kép:

$$f * g = (f_R * g_R, f_G * g_G, f_B * g_B)$$

Simítás – példa



eredeti RGB

mindhárom komponens
ugyanazon maszkkal simítva

Simítás – példa



eredeti HSI

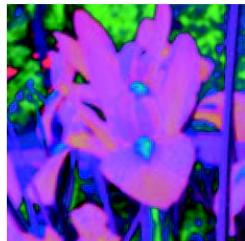


csak az *Intensity* komponens simított

Simítás – példa



eredeti HSI



mindhárom komponens ugyanazon maszkkal simítva

Simítás – példa



az RGB kép minden komponensének simítása 5*5-ös átlagoló maszkkal



- konverzió HSI-be
- csak az *Intensity* simítása 5*5-ös átlagoló maszkkal
- konverzió RGB-be



különbsékgép

Élesítés – példa

Kontrasztfokozás Laplace transzformációval:



az RGB kép minden komponensének transzformálása



- konverzió HSI-be
- csak az *Intensity* transzformálása
- konverzió RGB-be



különbsékgép

Mediánszűrés – példa



eredeti zajos kép



szűrt kép

Éldetektálás

Éltípusok a színes képeken

Az élek a színes képeken is fontosak vizuális információt hordoznak.

A színes képeken kétféle él különböztethető meg:

- **intenzitás-él**
(valamely irányban egy képpont környezetében az intenzitások jelentősen eltérnek),
- **szín-él**
(valamely irányban egy képpont környezetében az színek jelentősen eltérnek, bár az intenzitásban nincs lényeges változás).

Az éldetektálás alapmódszerei színes képeken

- Élek detektálása komponensenként/csatornánként, majd az egyes éltérképek kombinálása/fúziója.
- A szín és az intenzitás szeparálása, majd éldetektálás a szín- és intenzitás komponenseken, végül pedig az egyes éltérképek kombinálása/fúziója.

Éldetektálás csatornánként



élek az egyes színkomponenseken

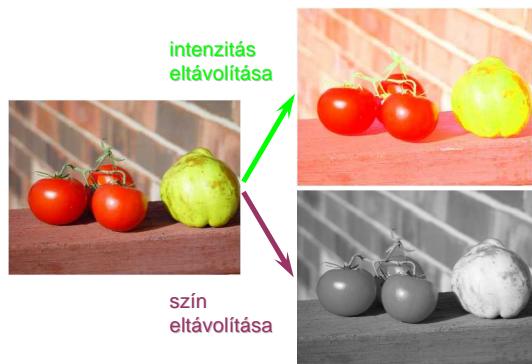
Éldetektálás csatornánként



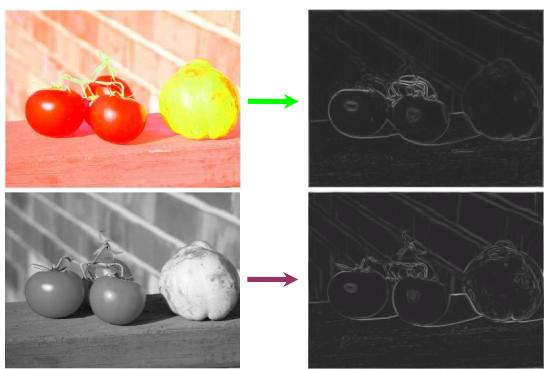
Éldetektálás csatornánként



Szín és intenzitás szeparálása



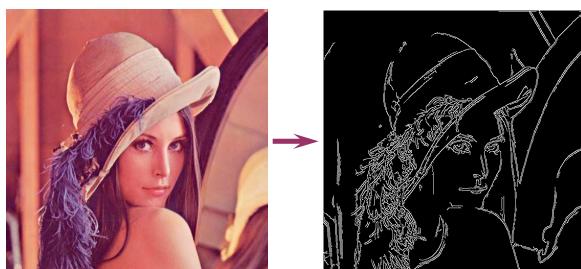
Éldetektálás a komponenseken



Az élképek kombinálása

- átlagolás
- súlyozott átlagolás
(a színmennyiségekkel)
- a domináns szín élképén lévő érték
- összegzés
- négyzetösszegek gyöke

Éldetektálás – példa

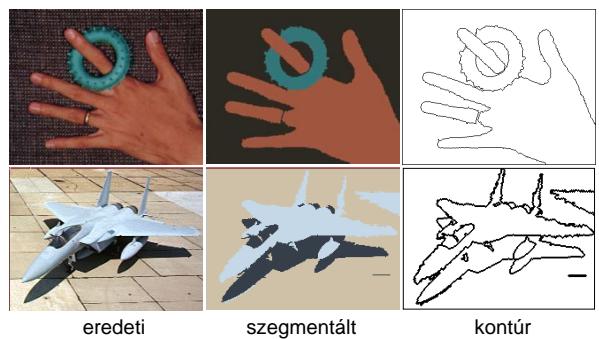


Éldetektálás – példa

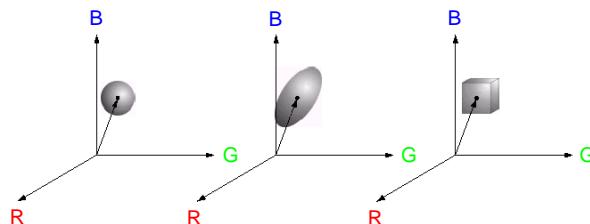


Szegmentálás

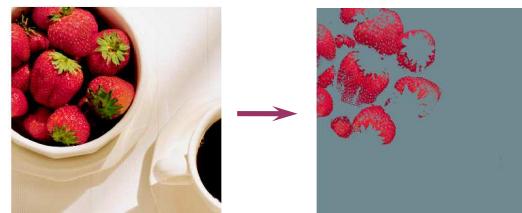
Szegmentálás



Szín-szeletelés



Szín-szeletelés – példa



Az RGB-kocka $(0.7, 0.15, 0.2)$ -körüli 0.2 sugarú gömbtartomány intenzitásait az $(1.0, 0.0, 0.0)$ pontba, a többet pedig a $(0.5, 0.5, 0.5)$ -be viszi.

Szegmentálás klaszterezéssel

A klaszterezés (*clustering*) a felügyelet nélküli tanulás (*unsupervised learning*) legfontosabb problémája, melynek feladata egy címkézetlen adathalmaz struktúrájának megtalálása, vagyis a klaszterek (hasonló elemek halmaza) megadása.

Néhány módszere:

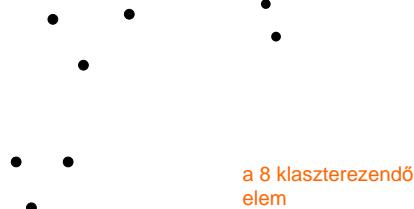
- K-means
- fuzzy C-means
- hierarchikus
- minimális költségű feszítőfa
- maximális/minimális vágás

K-means klaszterezés

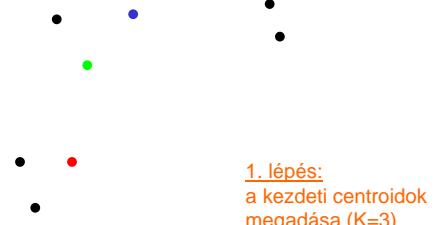
Az eljárás (*J.B. MacQueen, 1967*):

1. Adjunk meg K elemet kezdeti centroidokként (mint a klaszterek reprezentáns elemeit).
2. minden elemet rendeljünk ahoz a klaszterhez, melynek centroidjához a legközelebb esik.
3. Számoljuk újra a klaszterek centroidjait.
4. Ismételjük a 2. és a 3. lépést mindaddig, míg elmozdulnak a centroidok.

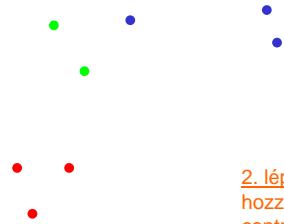
K-means – példa



K-means – példa

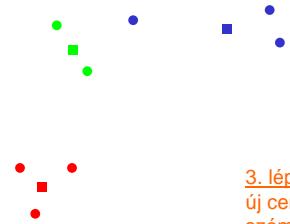


K-means – példa



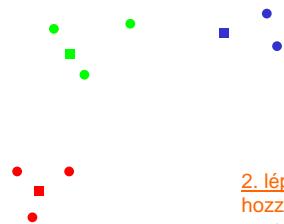
2. lépés:
hozzárendelés a
centroidekhez

K-means – példa



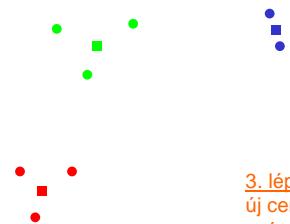
3. lépés:
új centroidek
számítása

K-means – példa



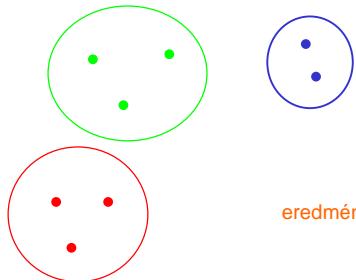
2. lépés:
hozzárendelés a
centroidekhez

K-means – példa



3. lépés:
új centroidek
számítása

K-means – példa



eredmény

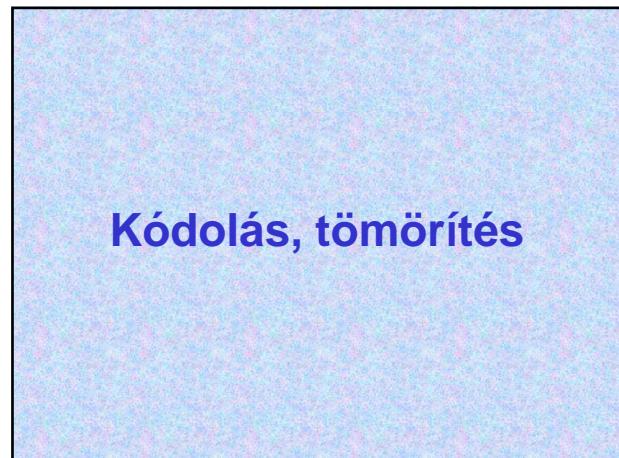
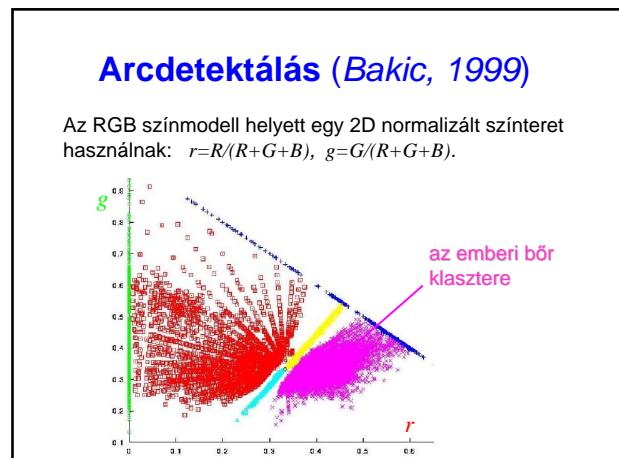
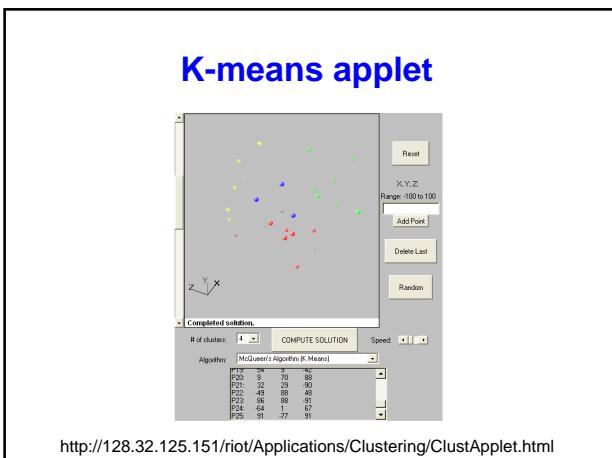
K-means – példa



eredeti RGB



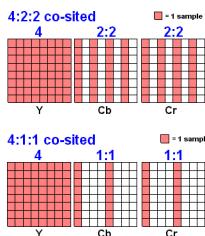
a K-means eredménye
(K=5)



- ### Egyeszerű tömörítési trükkök
- a színmélység csökkentése
(pl: $(8,8,8) \rightarrow (3,3,2)$)
 - a szín és az intenzitás szeperálásával durvább mintavételezés a színtartományban
 - Kvantálás – CLUT (color look up table)
(pl: a 256 jellemző 3*8-bites szín kiválasztása)



Durvább mintavételezés a színtartományban



Áttérés indexelt képre – példa

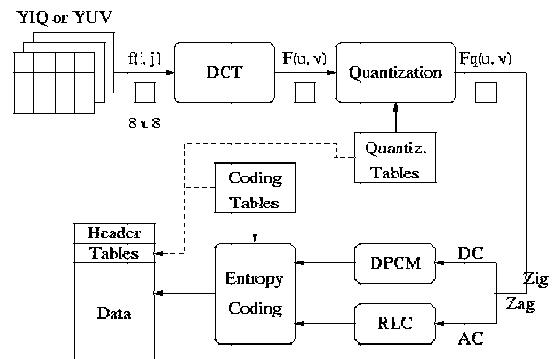


JPEG

A színes képek tömörítésének algoritmusa:

1. konvertálás az YIQ vagy az YUV színtérbe (az intenzitás és a jobban tömöríthető szín szeparálása, mivel a szem a fényességre jobban érzékeny)
2. DCT (a 8x8-as blokkok redundanciájának azonosítása)
3. kvantálás (egész osztás a kvantáló táblák paramétereivel – veszteséges)
4. kódolás (Huffman – veszteségmentes)

JPEG



JPEG – példa



JPEG ↔ GIF (PNG)

– színmélység:

JPEG: 24 bits/pixel
GIF: 8 bits/pixel (olcsóbb monitorokhoz)

– élmegőrzés:

JPEG: nagy tömörítési aránynál hajlamos az élek elkenésére, jó a fotókra
GIF: különösen alkalmas grafikákhoz, vonalrajzokra

– tömörítési arány:

JPEG: ~20:1 aránynál még nincs vizuális veszteség
GIF: 3:1 arányt hoz az indexelés, amit az LZW ~5:1-re javít