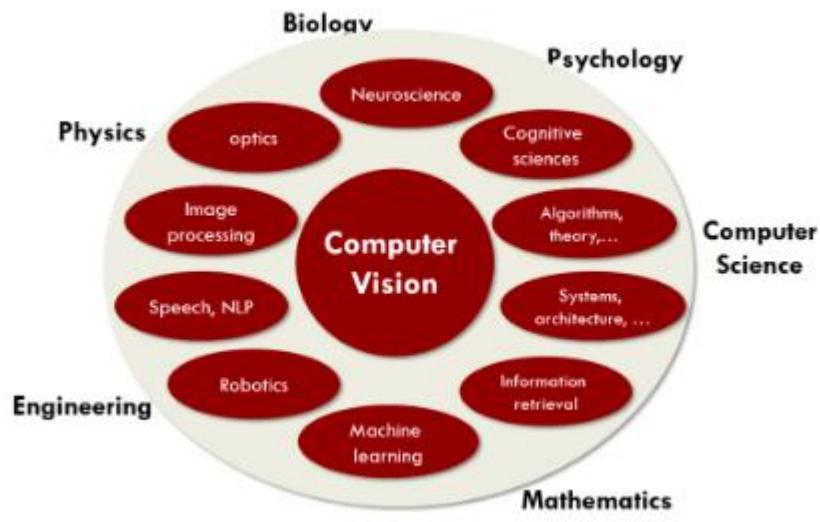


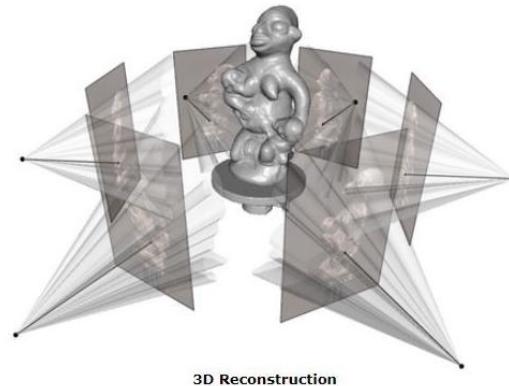
Gépi látás

Gépi látás



Alkalmazások

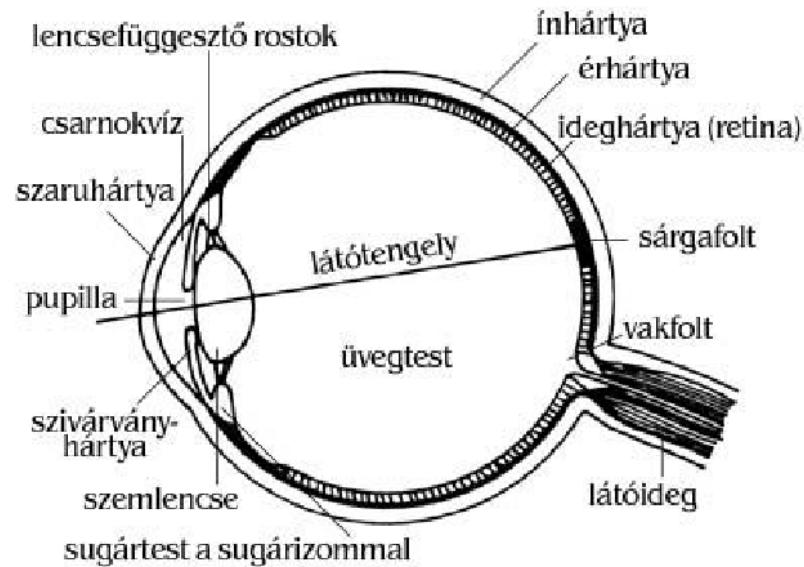
- Speciális effektek
- 3D modellezés
- 3D rekonstrukció
- Biometrikus azonosítás:
arc , ujjlenyomat,
vénaszkenner stb.
- Karakter felismerés
- Vizuális keresés
mobiltelefonnal
- Önvezető autók
- Automatic checkout
- Vizuális alapú
interakciók
- Augmented reality
- Virtual reality



Emberi látás

A fény útja a szemben

- A fény a **szaruhártyán** megtörve jut a szembe - ez lényegében egy **fénygyűjtőlencse**
- A **szivárványhártya** (iris) csökkenti a szembe jutó fény mennyiségét (szűri), a **pupilla** pedig **fényreteszként** funkcionál
- A **szemlencse** a **második gyűjtőlencse**; a belépő fénysugarakat a recehártyára (retinára) fókuszálja



Emberi látás

- A látóhártyában (retina) a kétféle fényérzékelő sejtek közül a „csapok” **képesek felismerni a színeket**. Ezek a csapok a sárgafolt (macula lutea) területén helyezkednek el.
- A különböző **csapok** három alapszín felismerésére képesek: **vörös, zöld és kék**. A különböző színárnyalatok felismerése e három alapszín keveréséből alakul ki. A tökéletes színlátáshoz a különböző színeket felismerő csapok megfelelő aránya és tökéletes működése szükséges.
- Az érzékelő elemekhez idegek kapcsolódnak, az ingereket a szemidegen keresztül az agy felé továbbítják. A látás **érzékelő elemei (fotoreceptorai)**, az esti fényben működő mintegy 130 millió **pálcika (a sötét-világos megkülönböztetésére alkalmas)** és a nappali fényben működő, mintegy 7 millió **csap (a színlátást és éleslátást szolgálják)**.
- Az esti látás elemei, a **pálcikák nem látnak színeket, de rendkívül érzékenyek**.
- A nappali látás elemei a **csapok, kevésbé érzékenyek és színesen látnak**, mert a csapokban három különböző pigment található, melyek egyike a vörös, másik a zöld, és a harmadik a kék fényre érzékeny.

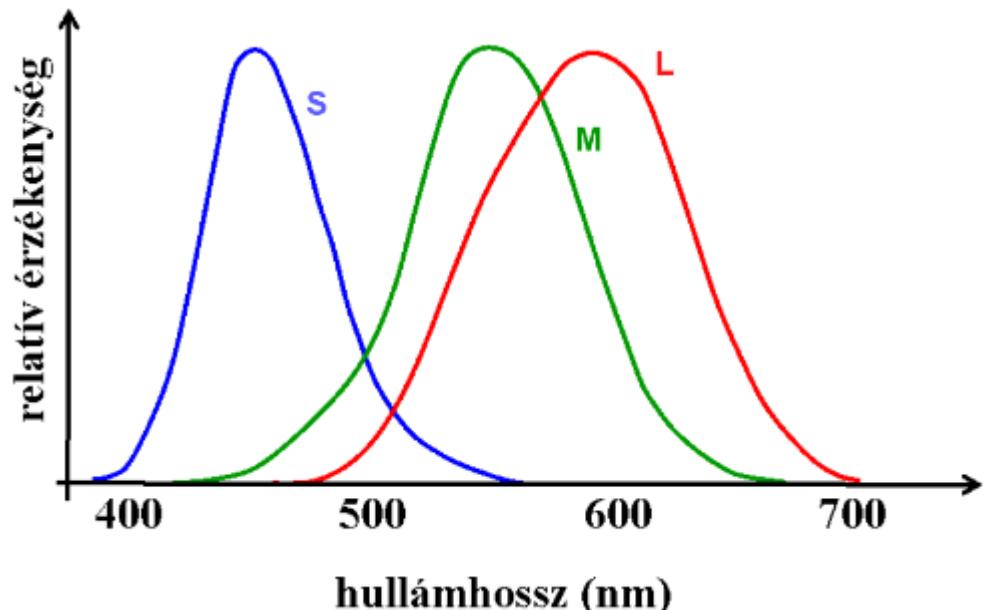
Emberi látás

Csapok

Háromféle csap található a szemben:

- S csap: 420nm körüli fényre a legérzékenyebb (kék)
- M csap: 530nm körüli fényre a legérzékenyebb (zöld)
- L csap: 560nm körüli fényre a legérzékenyebb (vörös)

Nem ugyanannyi van belőlük:
az L:M:S arány kb. 8:4:1



Emberi látás

A különböző hullámhosszú fény másképp törik

- Ahhoz, hogy a szemlencse ezeket is a retinára tudja fókuszálni a sugárizomnak módosítania kell a szemlencse alakját

Ugyanaz történik, mint amikor közelre, vagy távolra nézünk! (közeli tárgy domborúbb szemlencse, távoli tárgy kevésbé domború szemlencse)

- a piros szín azért kelt közelség érzetet, mert ugyanaz játszódik le piros felületre fókuszáláskor, mint amikor közelebbre nézünk
- a kék színnél pedig ugyanaz játszódik le, mint amikor távolabba nézünk, ezért kelt tágasabb érzést

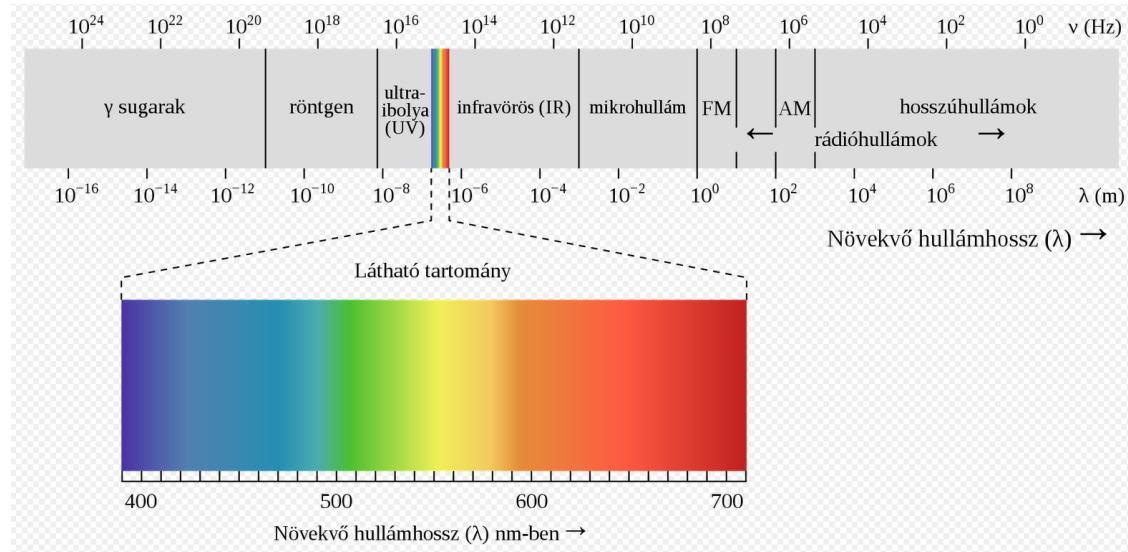
Emberi látás

- Amikor fény ér egy fotoreceptort egy kémiai reakció indul el, aminek eredményeképp egy neurális jelet küldenek az agy felé, úgynévezett fotopigmentet
- Ha egy csap egy adott hullámhosszra 30%-ban érzékeny az azt jelenti, hogy átlagosan 10-ből 3-szor fogja abszorbeálni az olyan hullámhosszú fénykomponenst és küld jelet az agy felé
- A fényreceptorok észleléseit a látóideg továbbítja az agy felé
A látóideg csatlakozási pontja a szemgolyóhoz a vakfolt, itt nincsenek sem csapok, sem pálcikák
- A **látogödör**, ami a vakfolttól oldalra található az **éleslátás helye**, a közepe a foveola, ahol kizárolag csapok találhatóak (150000 csap per mm)
- A foveolátótól kifelé haladva a csapok egyre ritkábbak és a pálcikák váltják fel őket

Elektromágneses spektrum

A fény a teljes elektromágneses spektrumtartomány látható része, amelyet hullámhosszával jellemezhetünk. A látható fény 380-780 nm hullámhossztartományba esik. Az egyes hullámhosszaknak színek felelnek meg.

A különböző hullámhosszúságú fotonok más-más érzetet keltenek a szemlélőben, ezeket az érzeteket nevezük színeknek



Emberi látás

Út az agyba

agy felé küldött információ nem egyszerűen a háromféle csap által leadott jel

Ehelyett az S, M, L által leadott jelek összege vagy különbsége indul az agyba

Az agy felé három idegkötegen halad a színinformáció

- A = M+L: az M és az L csapok által adott válaszok összege, akromatikus csatorna
- R/G = M - L: vörös-zöld különbség
- B/Y = S - A: a kék és az akromatikus csatorna (kb sárga) különbsége

Azaz **1D-s intenzitás és 2D-s szín (króma) információból rekonstruálunk**

Emberi látás

Az agy felé továbbított csatornák azt is befolyásolják, hogy egy adott színt mennyire tartunk **telítettnek** (saturated) → Nagyjából a színt érzékelő kromatikus csatornák és az akromatikus csatorna válaszának aránya határozza meg ezt - Ezért is van, hogy például a sárgát (=erős akromatikus válasz) nem tartjuk annyira telítettnek, mint akármilyen pirosat vagy kéket (=alacsony akromatikus válasz)

Színek a számítógépen

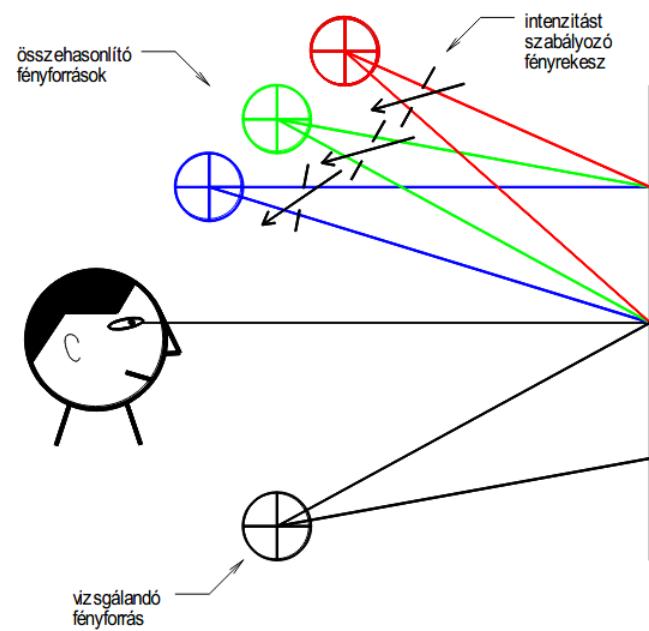
- International Commission on Illumination, 1931.: hogyan lehetne egy "standard" leírást adni arra, hogy egy ember miképp érzékeli a színeket
- A kísérletek egyik eredménye volt, hogy bármely szín előállítható három, megfelelő szín keverékeként => Bármely színérzet kódolható egy **számhármassal, tristimulus értékkel**

Színterek

A színterek a színek ábrázolására használható virtuális térbeli koordináta-rendszer, ahol az egyes színek tulajdonságait azok koordinátái fejezik ki.

RGB színtér

- 1931: Az additív színmegfeleltetés alapkísérlete: minden színinger létrehozható 3 egymástól független színinger additív keverékeként. A függetlenség alatt azt értjük, hogy a három színinger közül egyik sem hozható létre a másik kettő additív keverékeként.



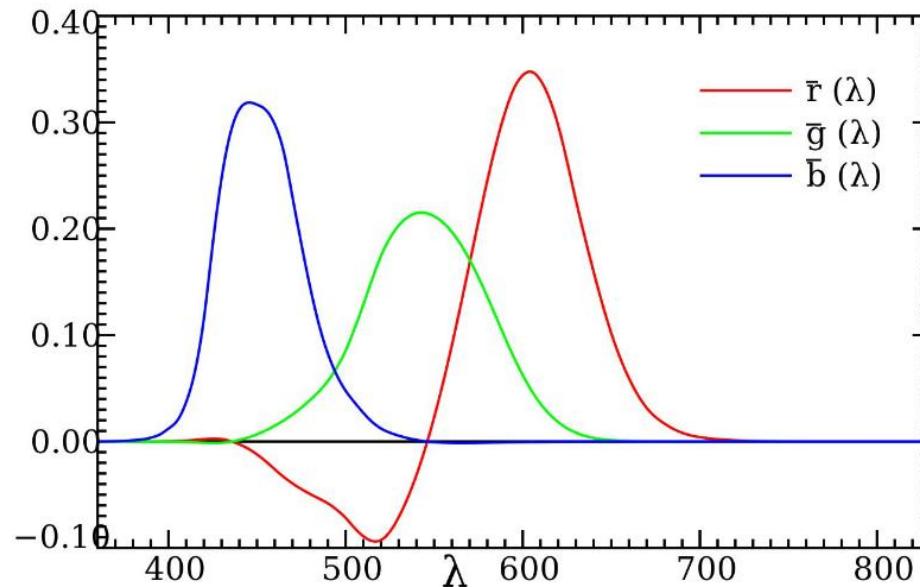
RGB színtér

Az RGB színtér egy olyan **additív színmodell**, ami a **vörös**, **zöld** és **kék** fény különböző mértékű keverésével határozza meg a különböző színeket. Az elnevezése ezen három alapszín angol megfelelőinek első betűiből ered:
Red (piros) 700 nm hullámhosszú monokromatikus fény,
Green (zöld) 546,1 nm hullámhosszú monokromatikus fény
Blue (kék) 435,8 nm hullámhosszú monokromatikus fény
Az RGB skálán egy színt az határoz meg, hogy milyen intenzitású a három komponense.
egységnyi alapszíneket összekeverve fehéret kapunk

RGB színtér

- Ha adott S spektrumú fény R, G, B koordináit akarjuk kiszámolni, akkor csak képezzük S és a színillesztő függvények skaláris szorzatát

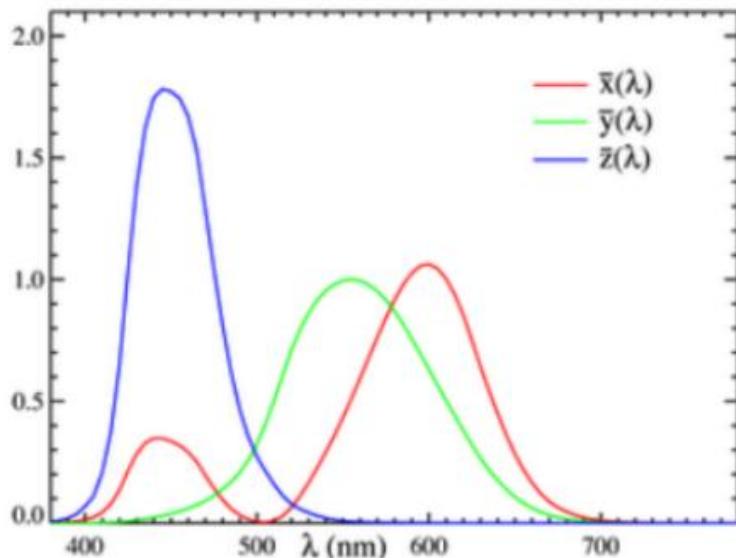
$$R = \int_0^{\infty} S(\lambda) \bar{r}(\lambda) d\lambda$$
$$G = \int_0^{\infty} S(\lambda) \bar{g}(\lambda) d\lambda$$
$$B = \int_0^{\infty} S(\lambda) \bar{b}(\lambda) d\lambda$$



CIE XYZ színtér

Az RGB térrel az első problémát az jelentette, hogy negatív koordinátákra is szükség volt bizonyos hullámhosszú monokromatikus fény érzékeléséhez hasonló fényérzet kialakításához XYZ színtér: a látható spektrumbeli fények csak pozitív súlyokkal kikeverhetőek
Y: a fényességet adja

CIE XYZ színtér

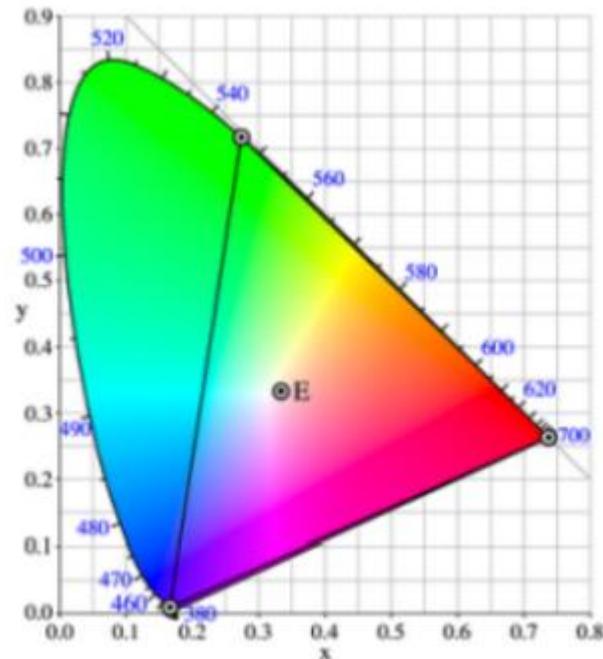


RGB-ből egy lineáris leképezés átvisz XYZ-be:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{b_{21}} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{0.176,97} \begin{bmatrix} 0.490\,00 & 0.310\,00 & 0.200\,00 \\ 0.176\,97 & 0.812\,40 & 0.010\,630 \\ 0.000\,0 & 0.010\,000 & 0.990\,00 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

CIE XYZ színtér

- színtkoordináták segítségével a színeket színtkoordináta-rendszerben lehet ábrázolni
- A görbe vonalain helyezkednek el a legtelítettebb színek, a spektrumszínek
- minden színt egy színpont jellemzi
- Két pont közötti egyenesen lévő színek a két pontból kikeverhetők.
- Három pont által adott háromszög belsejében lévő színek a három pontból kikeverhetők.
- A színpatkó NEM háromszög alakú, ezért nincs három olyan szín, amikből az összes szín kikeverhető lenne.
- egységnyi alapszíneket összekeverve fehéret kapunk
- színességi koordináták:
Színingert három számmal jellemzhetjük: x,y és Y



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

$$x + y + z = 1$$

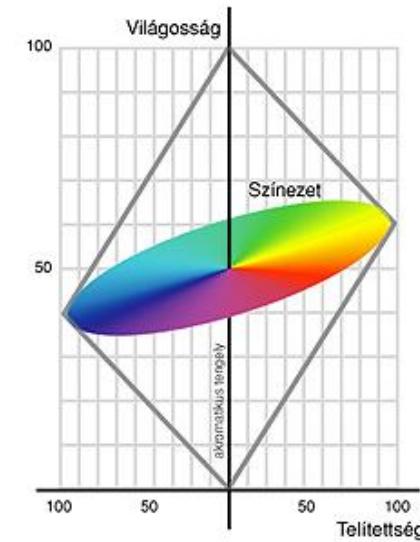
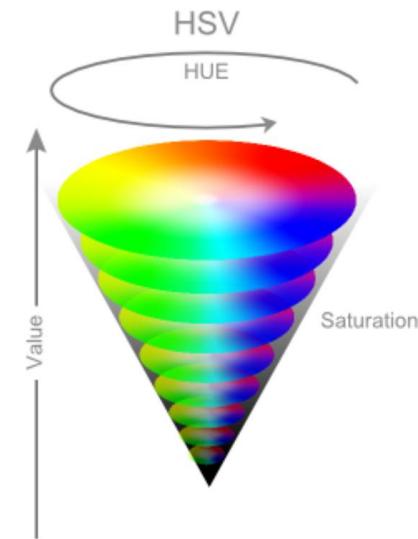
HSV színtér

Cél: a színkoordináták az ember által érzékelt szubjektív fogalmakhoz adaptáltak legyen

- Világosság L(lightness), vagy V (Value): 0 és 100 százalék között, 0 teljesen fekete legalsó végpont, fehér legfelső végpont
- Színezet (Hue): a szám 0 és 360 fok között van kifejezve
- Telítettség (Saturation): 0 és 100 százalék között mennyi szürkét tartalmaz, telítettebb élénkebb hatás

a három színjellemzőt egy henger, (pontosabban egy kettőskúp) terében lehet a legszemléletesebben ábrázolni

- A színezet a henger főkörén fut körbe
- A telítettség sugárirányban (a vízszintes síkban) változhat
- A világosság a henger tengelyének irányában (függőleges síkban) helyezkedik el.



Digitális képfeldolgozás

- A számítógép a képi információkat is digitális adatokként kezeli, így a kép minden jellemzőjéhez valamelyen számot rendel.

Pixel: A kép legkisebb egysége

Minden képpont (pixel) elérhető a koordinátája alapján.

Digitális kép típusai:

- Vektorgrafikus kép:
- Pixelgrafikus kép: képpontokból áll, jellemző a képpontok és a színek száma

Vektorgrafikus kép

Alkalmazási területek:

- Mérnöki tervezés (CAD)

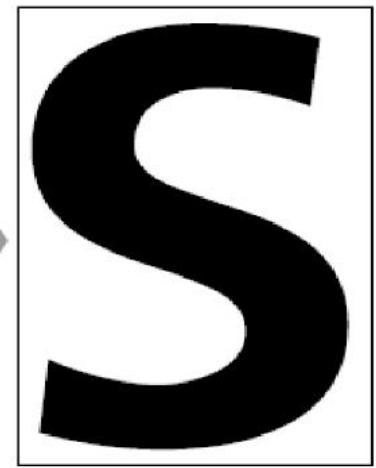
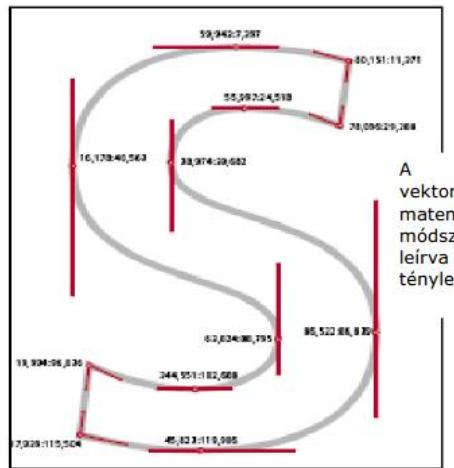
- Térképészeti (GIS)

- Kiadványszerkesztés

- Animáció és filmgyártás

megjelenített kép elemeit a számítógép matematikailag leírható vonalakra görbékre bontja, majd ezek egyenleteit tárolja

- az árnyalatokat nehézkesen kezeli
- nagyításkor a felbontás nem romlik mivel csak a csomópontok koordinátái változnak, maga a képet leíró függvény nem



Pixelgrafikus kép

- Alapegysége és a felbontás egysége a képpont (pixel)
- a képek külön tárolt képpontokból épülnek fel
- minden képpont tulajdonságait numerikus értékek határozzák meg (színmélység)
- korlátlan színhasználat;
- a pixelméret csak bizonyos határok között módosítható;
- képméret változáskor minőségromlás;
- a képeknek nagy a helyigénye;

Pixelgrafikus kép

Felbontás: DPI (dot per inch)

Ha egy kép 300 DPI-s, akkor 1 inch hosszon 300 képpontból áll

A felbontás 2x-es növelésével a kép mérete a négyzetesen nő

Nagyfelbontású képeket nyomdai alkalmazásra, archiválásra, vagy egyéb speciális feladatokhoz használunk.

Képábrázolási módok

Bittérképes kép

- az egyes képpontokhoz tartozó információt egy bit hordozza (fekete vagy fehér)

Szürkeárnyalatos kép

csak a szürke és árnyalatai jelenhetnek meg egyszerre

legfeljebb 256 árnyalat

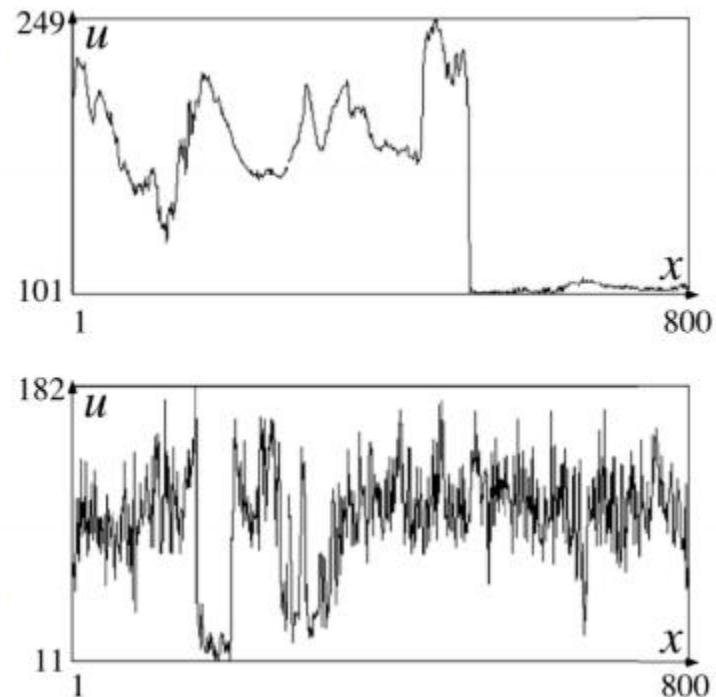
a kép pontonként 8 biten ábrázolható

Színes kép

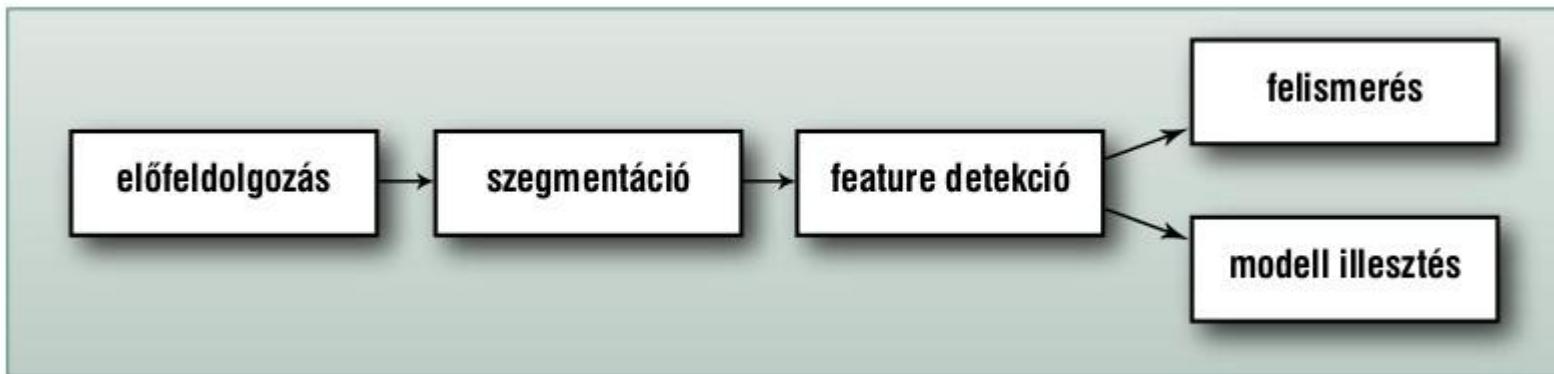
Pixelenként 3 db érték (pl. RGB, HSV stb.)

Hisztogram

Adott pixelértékek milyen gyakran fordulnak elő



Gépi látás folyamata



Alakzatok

- **Él:** egy nagyobb, a kontúrra merőleges intenzitás-változás
- **Sarok:** egy hirtelen forduló a kontúron
- **Vonal:** egy keskeny, hosszú régió
- **Folt:** egy kompakt régió



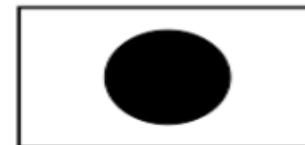
edge



corner



line



blob

Előfeldolgozás

bemenet és a kimenet is egy-egy kép

Feladatok:

- Fölösleges információk eldobása → átalakítás szürkeárnyalatos képpé
- Zajszűrés
- Élesítés
- Kontraszt erősítése

Zajtípusok

- **Additív képfüggetlen, azaz fehér zaj:**

$$g(x, y) = f(x, y) + v_{\text{add}}(x, y)$$

ahol $f(x, y)$ az inputkép, $g(x, y)$ az outputkép, $v(x, y)$ a zaj. Ez a tipikus csatornazaj (jeltovábbítási zaj, transmission noise).

- **Nemkorrelált multiplikatív zaj:**

$$g(x, y) = f(x, y) \cdot v_{\text{mult}}(x, y)$$

Ez a televíziós rasztersorokra jellemző amplitudó-moduláció (változás).

Zajtípusok

- **Kvantálási zaj (hiba):**

$$v_{\text{kvant}}(x, y) = g_{\text{kvant}}(x, y) - f_{\text{eredeti}}(x, y)$$

Az eredeti jelérték folytonos, a kvantált jelérték diszkrét, a különbség véletlen zajként jelenik meg.

Só-és-bors zaj (salt-and-pepper, vagy peak noise): Ez a pontszerű, a képpel nem korreláló, véletlen zaj legtöbbször szélsőértékű (fekete és fehér). Jellemző egyes fajta űrfelvételekre.

Képszűrés

Digitális képfeldolgozás központi fogalma és legfontosabb művelete

- a képtérben működő: közvetlenül a képértékekkel operálnak
- más térben működő: például frekvenciatartományban

Lokális operátorok

Legyen $f(x, y)$ a bemeneti (input) kép, $g(x, y)$ a kimeneti (output) kép

Az (x, y) pontban az eredmény csak a pont környezetétől függ:

$g(x, y) = T[f(x, y)]$, ahol T a környezeten definiált operátor

Képszűrés

- **Rekurzív operátorok:**

- az aktuális eredmény a bemenettől és az előző eredményektől is függhet. A kimenet nincs elválasztva a bemenettől, és az operátor működése során a bemenet módosul, mert a bemeneti képmátrixba írjuk be az eredményt. Ennek a hatása annál jelentősebb, minél nagyobb a környezet, az ablak.

- **Nem rekurzív operátorok:**

- az eredmény csak a bemenet aktuális környezetétől függ
- A kimenet el van választva a bemenettől, és a működés során a bemenet nem módosul, így a művelet hatása korlátozódik a környezetre.

Képszűrés

3x3-as átlagszűrők

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{6} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

5x5-ös átlagszűrők

$$\frac{1}{81} \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{100} \begin{bmatrix} 0 & 3 & 4 & 3 & 0 \\ 3 & 6 & 7 & 6 & 3 \\ 4 & 7 & 8 & 7 & 4 \\ 3 & 6 & 7 & 6 & 3 \\ 0 & 3 & 4 & 3 & 0 \end{bmatrix}$$

Képszűrés

- Példa konvolúciós szűrő alkalmazására

3	2	8	7	8	8
2	2	7	8	7	7
2	3	9	9	8	8
1	2	9	9	7	8
2	2	8	8	8	8
2	3	7	7	9	7

$$* \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} =$$

—	—	—	—	—	—
—	4	—
—	—
—	—
—	—
—	—	—	—	—	—

3	2	8	7	8	8
2	2	7	8	7	7
2	3	9	9	8	8
1	2	9	9	7	8
2	2	8	8	8	8
2	3	7	7	9	7

$$* \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} =$$

—	—	—	—	—	—
—	4	6	—
—	—
—	—
—	—
—	—	—	—	—	—

Képszűrés

Képszél probléma kezelése

Heurisztikus megoldások vannak rá:

- Töltsük ki nullákkal!
 - Ez a legegyszerűbb megoldás, amely azonban nemkívánatos, erős mesterséges éleket eredményezhet
- Töltsük ki az eredménykép átlagértékével!
 - kevésbé erős mesterséges éleket kaphatunk és nem változtatjuk meg az eredménykép értéktartományát.
- Töltsük ki a legközelebbi kiszámított pixelértékkel!
 - Akkor célszerű alkalmazni, ha minden áron el akarjuk kerülni a mesterséges élek megjelenését.

Képszűrés

Lineáris képszűrők

Átlagszűrő:

- Képtérben működő lineáris simítószűrő
- A súlyok nem negatívak, nem nőnek a középponttól való távolsággal, és 1 az összegük
- A gyakorlatban a súlyok gyakran egész számok, és a maszk alkalmazása után a súlyok összegével normálják az eredményt.

Képszűrés

Átlagszűrők típusai:

- **Dobozszűrő (box filter):** a legegyszerűbb és a leggyorsabb, azonos súlyokkal rendelkező átlagszűrő
Egy $(2M + 1) \times (2N + 1)$ -es méretű ablakban az eredmény a képértékek egyszerű, nem súlyozott átlaga:

$$g(x, y) = \frac{1}{(2M + 1) \times (2N + 1)} \sum_{x'=-M}^M \sum_{y'=-N}^N f(x + x', y + y')$$

Képszűrés

Átlagszűrők típusai:

- **Gauss-szűrő:** legelterjedtebb átlagszűrő

Súlyokat a normáleloszlás (Gauss-eloszlás) adja

A Gauss-szűrő maszkja körsimmetrikus, mert csak az r -től függ. Az exponens miatt a maszk harangalakú, a σ paraméter szabályozza a sűrő méretét. Nagyobb σ nagyobb szűrőt és erősebb simítást eredményez.

$$w_G(x, y) = \frac{1}{\sum_{(x,y) \in W} e^{-\frac{r^2(x,y)}{2\sigma^2}}} e^{-\frac{r^2(x,y)}{2\sigma^2}}$$

ahol $r^2(x, y) = x^2 + y^2$

Képszűrés

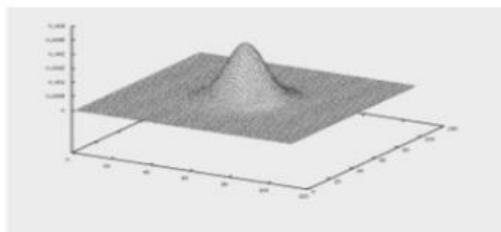
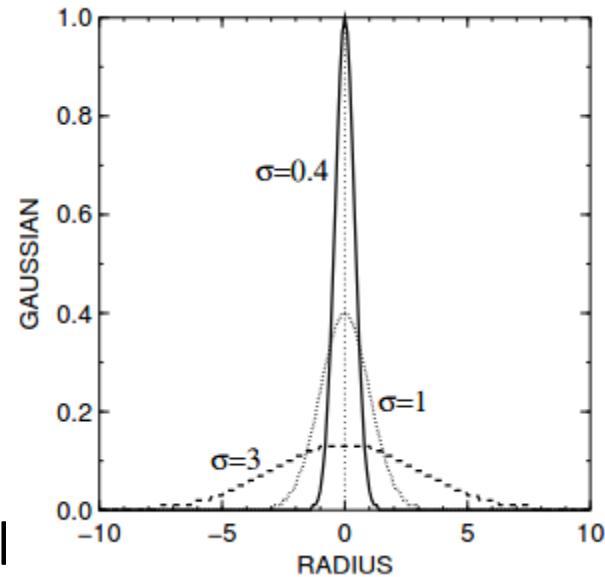
Átlagszűrők típusai:

- Gauss-szűrő

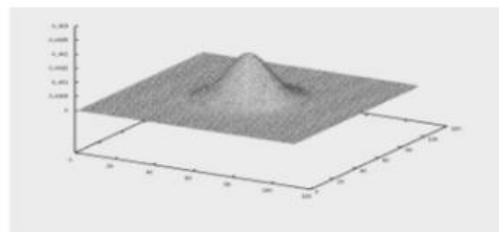
Diszkretizáláskor $w_G(r)$ -t elvágjuk $r_{\max} = k\sigma$ -nál

2,5 → mert ebben benne van a szűrőtérfogat döntő része

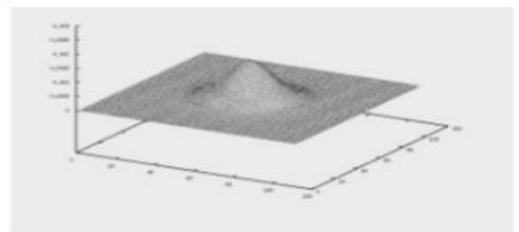
A Gauss-szűrő szeparálható (2D-s szűrő helyett két 1D-s szűrő alkalmazható) $w_G(x, y) = w_G(x) \cdot w_G(y)$



$$\sigma = 9$$



$$\sigma = 10$$



$$\sigma = 11$$

Képszűrés

Simitószűrés alkalmazásai és tulajdonságai:

- Simítószűrővel **zajszűrést** elsősorban a **nulla átlagú fehér zaj esetén** végezhetünk, mert az átlagban az ellenkező előjelű zajok véletlenszerűen semlegesítik egymást. Minél nagyobb a szűrő, annál nagyobb a semlegesítés valószínűsége és a zajcsökkentés mértéke. Vannak azonban negatív mellékhatásai: a **kontrasztcsökkenés** és az **élelmosódás** (edge blurring).
- **Alulmintavételezésre és a felbontás csökkentésére:** egyre nagyobb méretű Gauss-sz szűrőt alkalmazzuk egy képre, a finom részletek egyre jobban tűnnek el, majd szűrés után minden 2. oszlop és sor törlése → **képpiramis** jön létre
- **Mértéktér felépítése:** egy képből Gauss-szűréssel nyert képsorozat növekvő σ mellett. Ez az adatstruktúra hatékony, változó részletességű képelemzést tesz lehetővé
- Hibás bemeneti értékek (outlier-ek) nagy mértékben befolyásolhatják az eredményt. Az outlier-ek a normális zajszinten felüli, teljesen hibás adatok. Pl só-bors zaj is az, így ennek eltávolítására kevésbé alkalmas

Képszűrés

Mediánszűrő

A mediánszűrő eredménye az ablakban levő értékek mediánja

Példa: egy 3×3 -as ablakban az értékek:

(1, 1, 3, 2, 5, 4, 4, 12, 11)

akkor a rendezett sorozat:

(1, 1, 2, 3, 4, 4, 5, 11, 12) és a medián a 4.

Képszűrés

Mediánszűrő

A medián meghatározása nemlineáris művelet P és Q számsorozatra

$$Med(\alpha P) = \alpha Med(P), \text{ de}$$

$$Med(P + Q) \neq Med(P) + Med(Q)$$

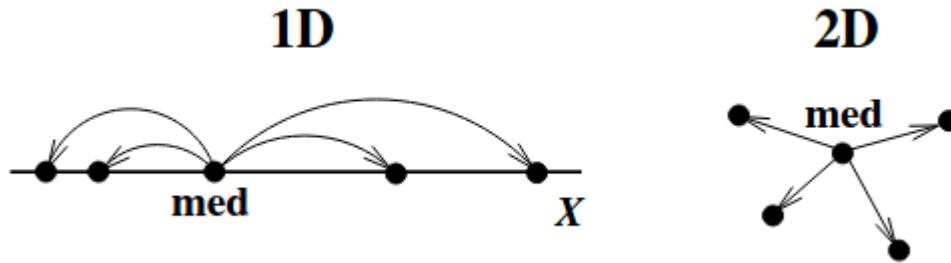
Az átlaggal ellentében a medián robustus statisztikai mennyiségeg (robust statistics). Ha a hibás adatok aránya kevesebb mint 50%, nem befolyásolják az eredményt.

Képszűrés

Mediánszűrő

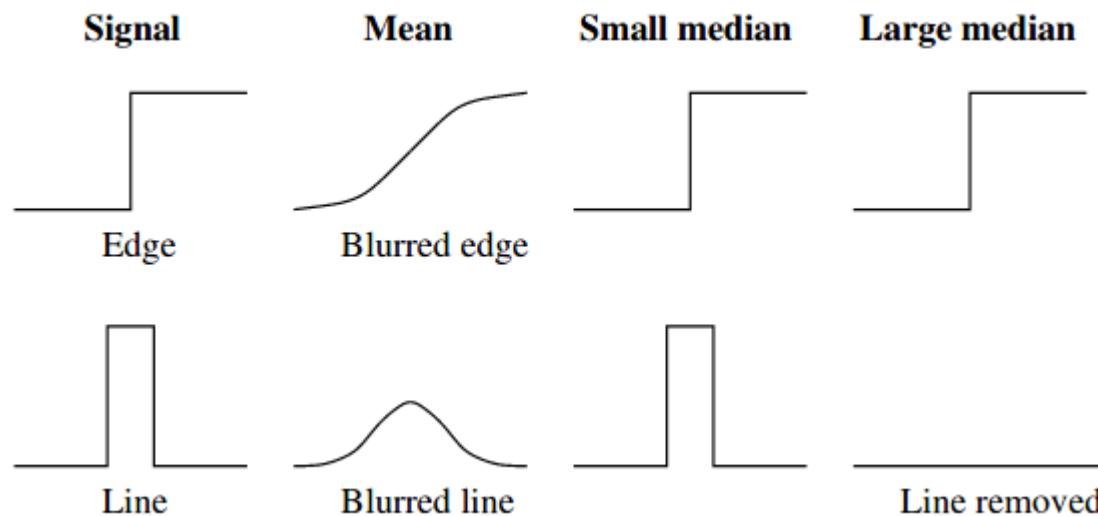
Tulajdonságai:

- mediánszűrő eltávolítja a só-és-bors zajt úgy, hogy nem mossa el az éleket és nem csökkenti a kontrasztot.
- A mediánszűrő törli a vékony vonalakat, ha a vonalvastagság kevesebb mint a szűrőméret fele. Ilyenkor a háttérpixelek többségben vannak az ablakban és ők adják mediánt.
- A mediánszűrő lekerekíti a sarkokat.
- A vektoros mediánszűrőt vektormezők javítására és simítására használják. A szűrő módosítja a hibás vektorokat, amelyek elütnek a környezettől. Iteratív alkalmazásával elsimítjuk a vektormezőt.



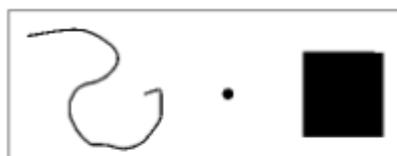
Képszűrés

Átlagszűrő és mediánszűrő összehasonlítása



Képszűrés

Dobozszűrő és mediánszűrő összehasonlítása



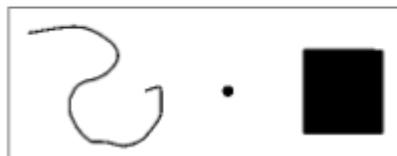
bemeneti kép



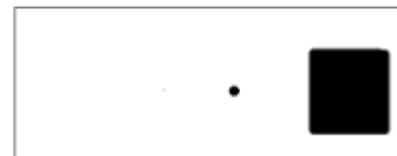
doboz 5×5



doboz 9×9



medián 3×3



medián 7×7

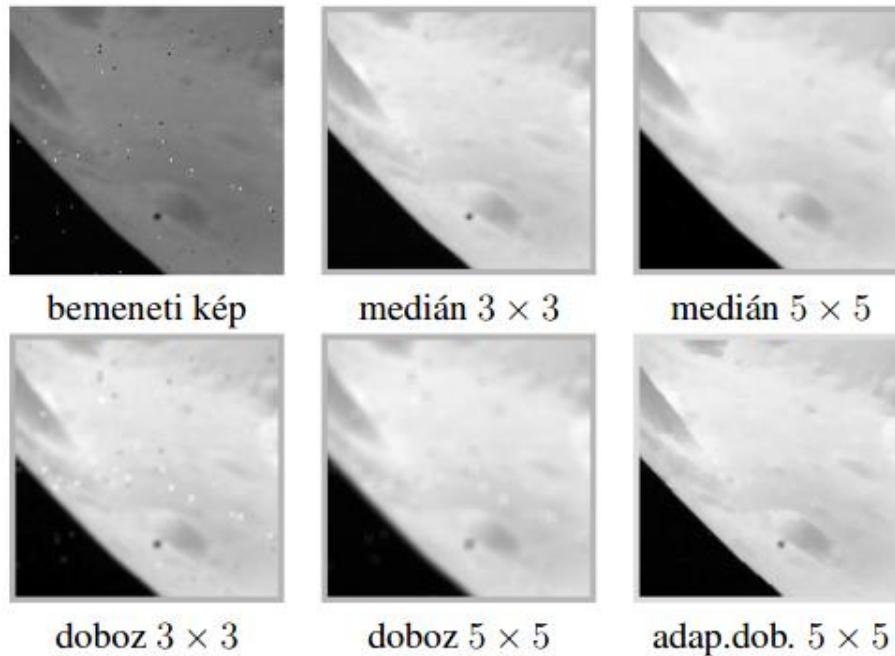


medián 17×17

Képszűrés

Dobozszűrő és mediánszűrő összehasonlítása só-bors zajra

Dobozszűrő nem szünteti meg a só-bors zajt, csak elmosa
Adaptív dobozszerő és a mediánszűrő megszüntette



Képszűrés

Laplace-szűrő:

A folytonos Laplace-operátor

$$\Delta f(x, y) \doteq \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

A deriváltakat különbségekkel közelítjük

$$\frac{\partial f}{\partial x} \rightarrow [-1 \ 1 \ 0]$$

$$\frac{\partial^2 f}{\partial x^2} \rightarrow [-1 \ 1 \ 0] - [0 \ -1 \ 1] = [-1 \ 2 \ -1]$$

$$\Delta f \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Képszűrés

Laplace-szűrő:

$$\Delta f(x, y) \approx f(x, y) - Av(x, y) = f(x, y) * w_L.$$

$$\frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

4 szomszéd

8 szomszéd

$$Av(x, y) \doteq \frac{1}{4} [f(x-1, y) + f(x, y-1) + f(x+1, y) + f(x, y+1)] \quad \text{szomszédos}$$

képelemek átlaga

Képszűrés

Laplace-szűrő:

Tulajdonságai:

- Az eredmény közel áll az eredeti és a simított kép különbségéhez. A lassú képváltozásokat levonjuk, a gyors változások megmaradnak. Ha nincs változás, nulla az eredmény (válasz, response).
 - A kimeneti kép értéktartománya elvileg $[-255, 255]$. Egy pixel és a szomszédjai különbsége azonban gyakran kicsi, ezért a gyakorlatban az értéktartomány lényegesen szűkebb.
 - A Laplace-szűrő kiemeli az intenzitás-változásokat és a finom részleteket: kontúrok, foltokat, vékony vonalakat.
 - A szűrő zaj-érzékeny, mert magasrendű deriváltakat tartalmaz. Egy simítószűrőt alkalmazhatunk előtte, hogy a képfüggvény deriválható legyen.
-
- **Laplacian-of-Gaussian (LoG) szűrő** a Laplace- és a Gauss-szűrő kombinációja: A Gauss-szűrő alkalmazása után a képfüggvény simább, deriválhatóbb lesz, ezért w_{LoG} kevésbé zajérzékeny

$$w_{LoG} = w_G * w_L$$

Képszűrés

Laplace-szűrő:

az eredmény negatív is lehet

Két lehetőségünk van:

- abszolútérték leképezés: elveszítjük az információt egy részét, mert elvész a változás előjele. Viszont jól láthatjuk azokat a képrészleteket, ahol finom változások vannak.
- normalizált érték leképezés: az értékeket leképezzük a [0, 255] tartományra és így ábrázoljuk az eredményképet. Ez nem jár információ-veszteséggel, de gyakran kevésbé szemléletes képet eredményez.

Képszűrés

Laplace-szűrő



bemenet



Laplace abszolút

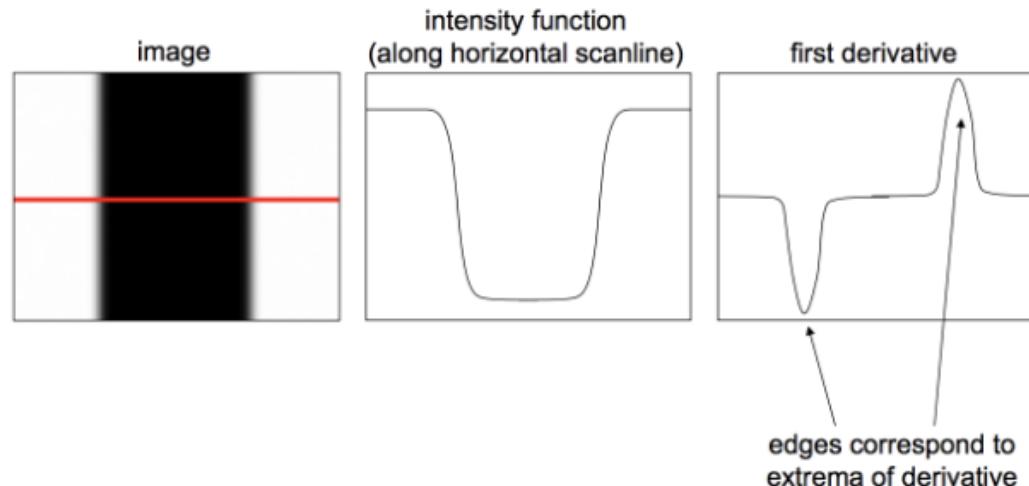


Laplace normalizált

Éldetektálás

Az éldetektálás egy gyakori képfeldolgozási feladat, amely segítségével be tudjuk határolni a képen látható régiókat, tárgyakat

Él: egy nagyobb, a kontúrra merőleges intenzitás-változás

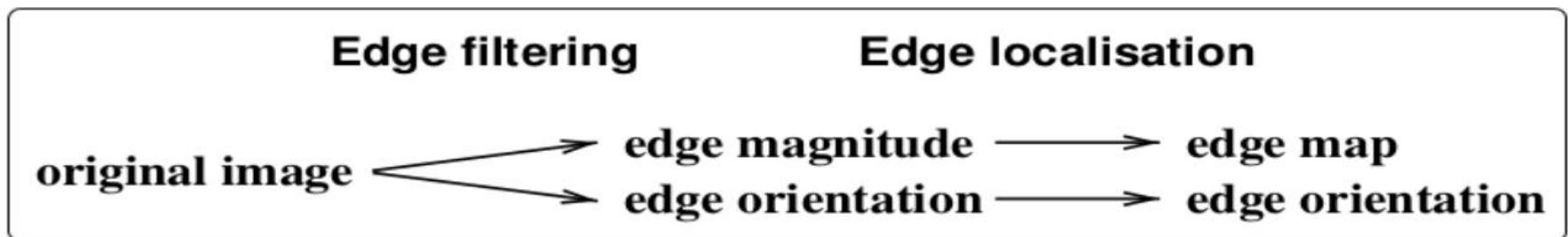


Éldetektálás

Éldetektálás folyamata:

1. **élszűrés**: élszűrő élekre reagál, vagyis felerősíti az éleket és elnyomja a kis változású régiókat.

2. **éllokalizáció**: utófeldolgozás eltünteti a zajos és ún. Fantom éleket



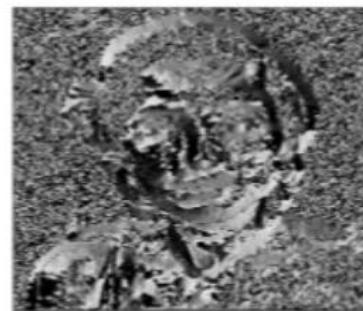
Éldetektálás



eredeti kép



élerősség



élorientáció



élkép



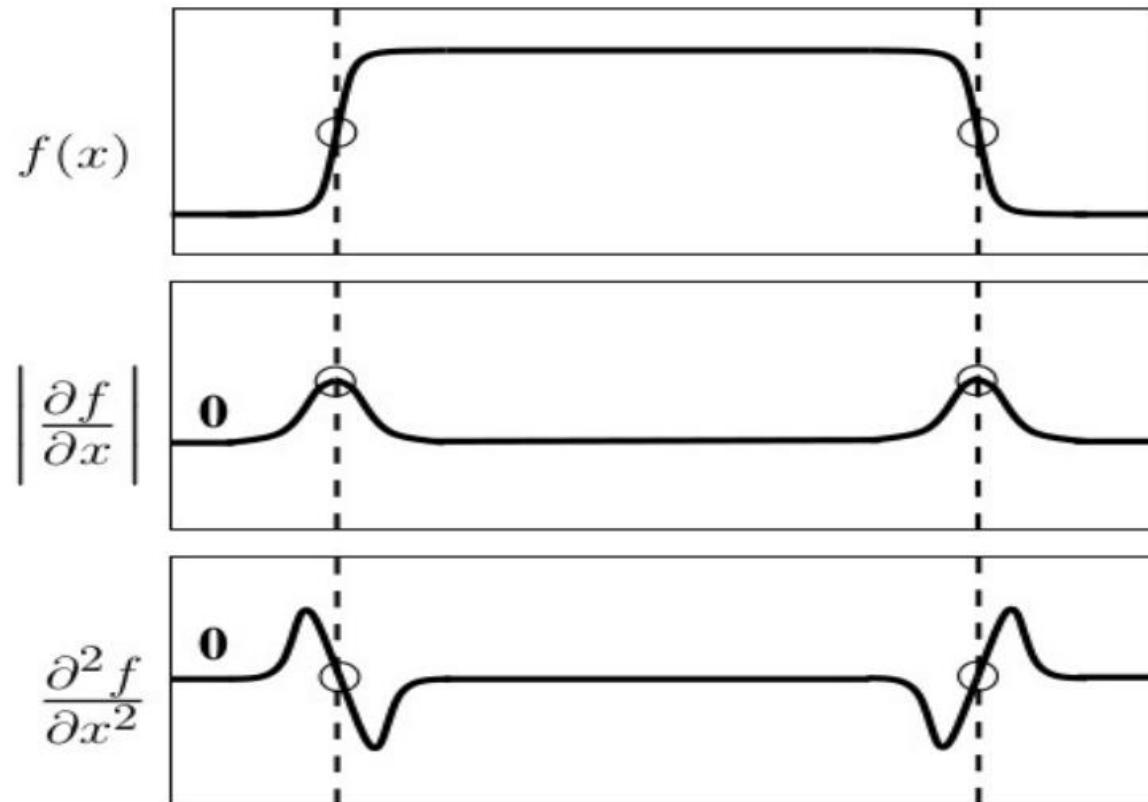
felső vonal



alsó vonal

Éldetektálás

Él és deriváltak kapcsolata:



Éldetektálás

Élszűrők:

a képfüggvény deriváltjait alkalmazzák, hogy felerősítsék az ére merőleges intenzitás-változásokat és elnyomják az ilyen változásokat nem tartalmazó régiókat

Leggyakoribb operátorok:

- Gradiens operátor
- Laplace operátor

$$\nabla f(x, y) \doteq \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

$$\Delta f(x, y) \doteq \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Éldetektálás

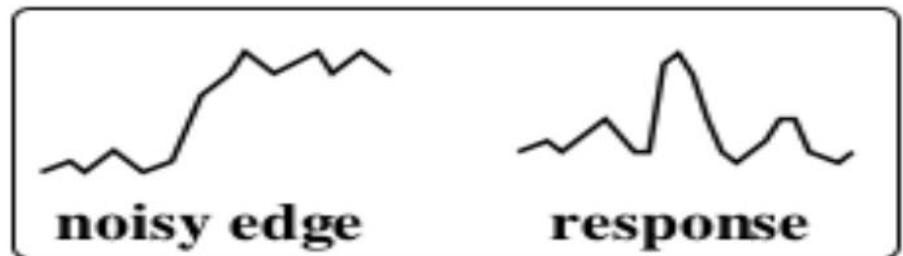
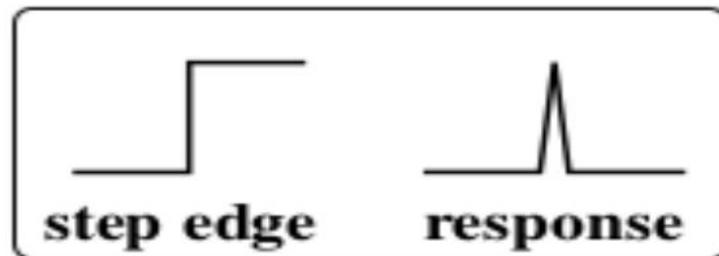
Élszűrők:

Jó lineáris élszűrő kritériumai:

1. Legyen nulla az eredmény ott, ahol nincs képváltozás.
2. Legyen jó a detektálás, azaz legyen minimális az alábbi események előfordulása:
 - hamis, zajos élek detektálása (false positives)
 - valós élek elvesztése (false negatives)
3. Legyen jó a lokalizálás: a detektált él a lehető legközelebb legyen a tényleges élhez.
4. A szűrő legyen izotróp: az eredmény ne függön az él orientációjától.
5. A szűrő egy élet csak egyszer jelezzen (single response): legyen minimális a valós él körüli hamis lokális maximumok száma.

Éldetektálás

egy zajos, elmosott él több szomszédos maximumot produkál



Éldetektálás

Gradiens élszűrők

$$\nabla f(x, y) \doteq \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) = (f_x, f_y)$$

A legkisebb méretű 3×3 -as gradiensszűrők esetén a parciális deriváltakat különbségekkel közelítjük, ezzel az X és Y irányú, G_x és G_y deriváltmaszkokat kapjuk:

$$f * G_x = f_x$$

$$f * G_y = f_y$$

G_y a G_x 90 fokos elfordítottja

Éldetektálás

- Gradiens élszűrők

3x3-as maszszűrők G_x komponensei (vertikális maszk)

$$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Prewitt

$$\frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

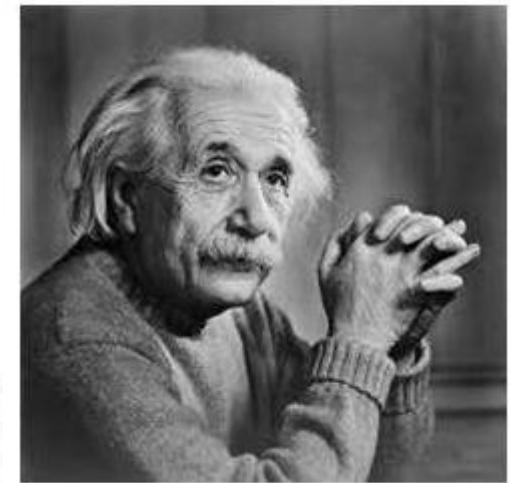
Sobel

$$\frac{1}{2+\sqrt{2}} \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

izotróp

Éldetektálás

Prewitt-élszűrő



Vertikális maszk (G_x) alkalmazása Horizontális maszk (G_y) alkalmazása

Kimeneti kép:

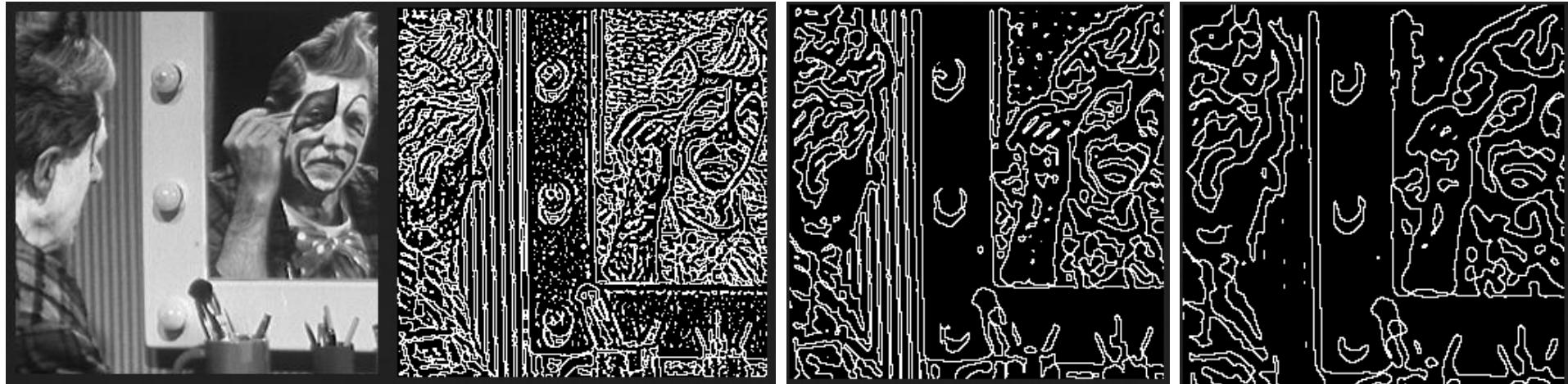
$$M(x, y) = \sqrt{f_x^2 + f_y^2}$$

$$\Theta(x, y) = \arctan \frac{f_x}{f_y}$$

Éldetektálás

LoG (Laplacian-of-Gaussian) szűrő

Élek 2. derivált nulla átmenetei



Gauss-szűrő paramétere:

$$\sigma=1.0$$

$$\sigma=2.0$$

$$\sigma=3.0$$

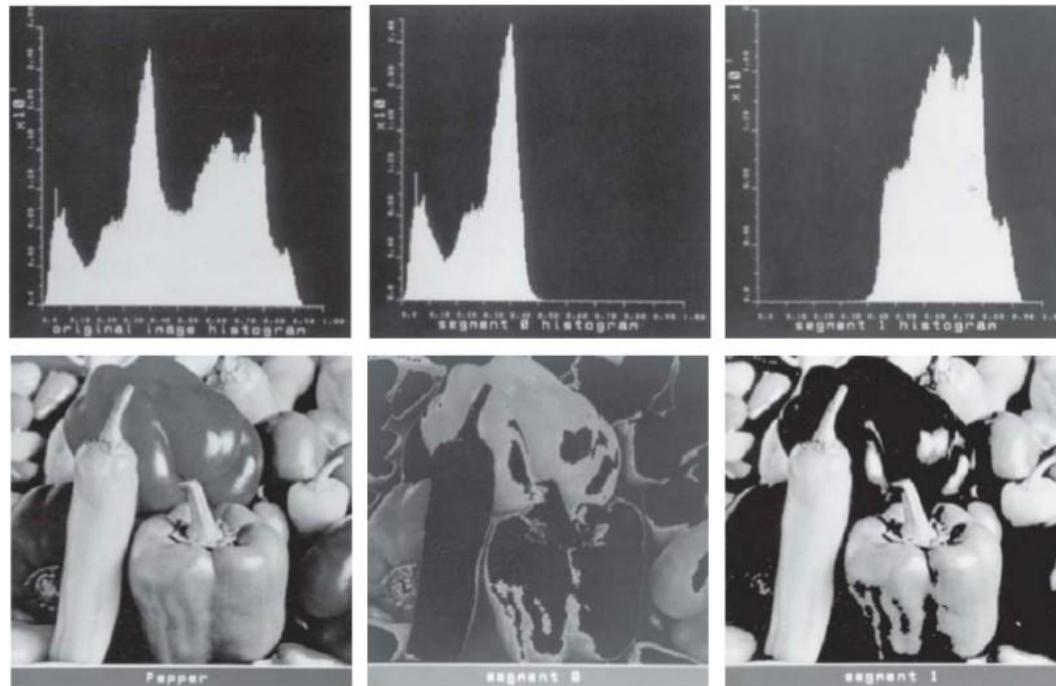
Szegmentálás

- Fontos területek kiválasztása: kép érdekes és érdektelen részeinek meghatározása
- Legtöbbször szín és világosság alapján
- Navigációs alkalmazásoknál például nem érdekes az égbolt, forgalomszámlálásnál az útfelület

Szegmentálás

Intenzitás alapján, küszöbözéssel:

Nagy homogén területek egyenletes intenzitás-értékek



Szegmentálás

Régió alapú szegmentálás: célja felosztani az I képet n darab R_1, \dots, R_n összefüggő és homogén régióra.

Egy régió homogén, ha valamelyik teljesül:

- $|I_{\max} - I_{\min}|$ kicsi;
- bármelyik $I(x, y) \in R$ pixelre $|I(x, y) - I_{\text{mean}}|$ kicsi, ahol I_{mean} a régió átlaga;
- a σ_R intenzitás szórás a régióban kicsi.

Leggyakrabban alkalmazott eljárások:

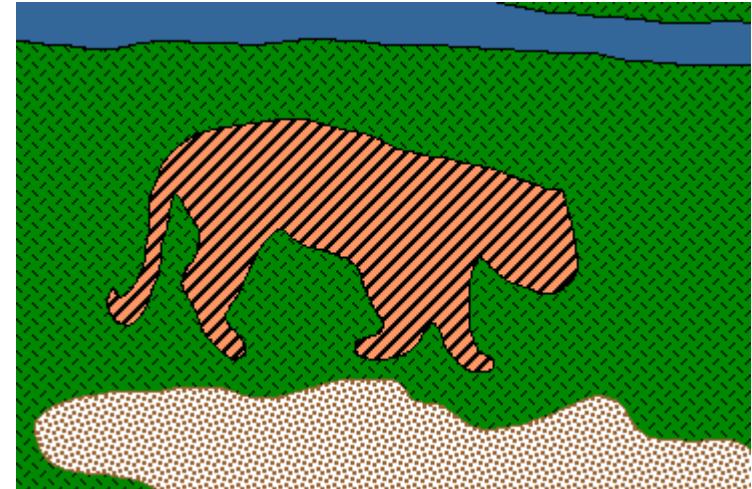
- Régió növesztés
- Régiók darabolása és egyesítése

Szegmentálás

Régió alapú szegmentálás:

A szegmentálás eredménye a következőktől függ:

- milyen képi tulajdonságokat használunk
⇒ intenzitás, szín, textúra
- hogyan hasonlítjuk össze a tulajdonságokat
- mekkora változásokat tolerálunk régiótól régióig



Szegmentálás

Pixel-felhalmozás: régió alapú szegmentálási eljárás

1. Inicializálás

- Kiválasztunk N darab s_i magpontot és egy T küszöböt.
- Magpontokkal inicializálunk N régiót: $R_i^{(0)} = s_i$.
- Inicializáljuk a régiók átlagértékeit: $M_i^{(0)} = I(s_i)$.

2. Iteráció, k -ik lépés

- Megvizsgáljuk az összes $R_i^{(k)}$ minden **határpixelének** 8-szomszédjait.
- Ha van olyan új szomszéd, p , amelyre $|I(p) - M_i^{(k)}| \leq T$, akkor p -t hozzáadjuk $R_i^{(k)}$ -hez.

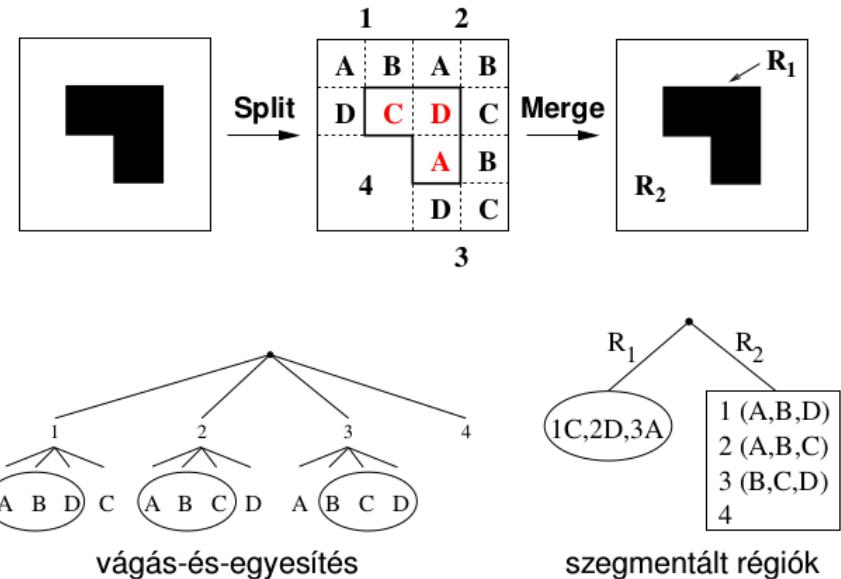
3. Megállunk, ha nem tudunk tovább növeszteni; különben, felfrissítjük az összes $M_i^{(k)}$ -t és iterálunk.

Szegmentálás

Vágás és egyesítés:

Vágás és egyesítés

1. Föntről lefelé (top-down)
 - felosztjuk a képet egyre csökken méretű R_i kockákra
 - megállunk, ha az összes kocka homogén: $P(R_i) = \text{TRUE}$
⇒ az eredmény egy négyesfa
2. Lentről fölfelé (bottom-up)
 - minden szinten egyesítünk két szomszédos R_i és R_j régiót,
 - ha $P(R_i \cup R_j) = \text{TRUE}$
3. Iteráljuk a két fázist, amíg van új felosztás vagy egyesítés



Szegmentálás

Egyéb szegmentálási módszerek:

- Él alapú: objektumok kontúrkeresése élkövetéssel
- Textúra, szín alapú: régiók textúráját, színét használják
- Mozgás alapú: objektumok szegmentálása mozgás alapján

Objektum felismerés

Különböző feladatok:

- Osztályozás: A kép tartalmazza a keresett objektumot? Pl: Van a képen kutya?
- Kép keresés: pl: Google photos
- Detektálás:
 - hol található a keresett objektum a képen?
 - geometriai és szemantikai tulajdonságok kinyerése: például két tárgy távolsága, Piros a közlekedési lámpa?

Objektum felismerés

Kihívások:

- Kategóriák száma:
 - Ember 10-30 ezer objektum kategóriát tud felismerni
 - Legjobb felismerő módszerek 200 kategória detektálásra, 1000 kategória klasszifikációra
- Nézőpont változása
- Megvilágítás
- Skálázás: egy kategóriába tartozó objektumok különböző méretűek lehetnek
- Deformáció: például ember különböző pozíciókat vehet fel
- Takarás (occlusion): egyik objektum a másikat
- Osztályon belül különböző változatok: például szék

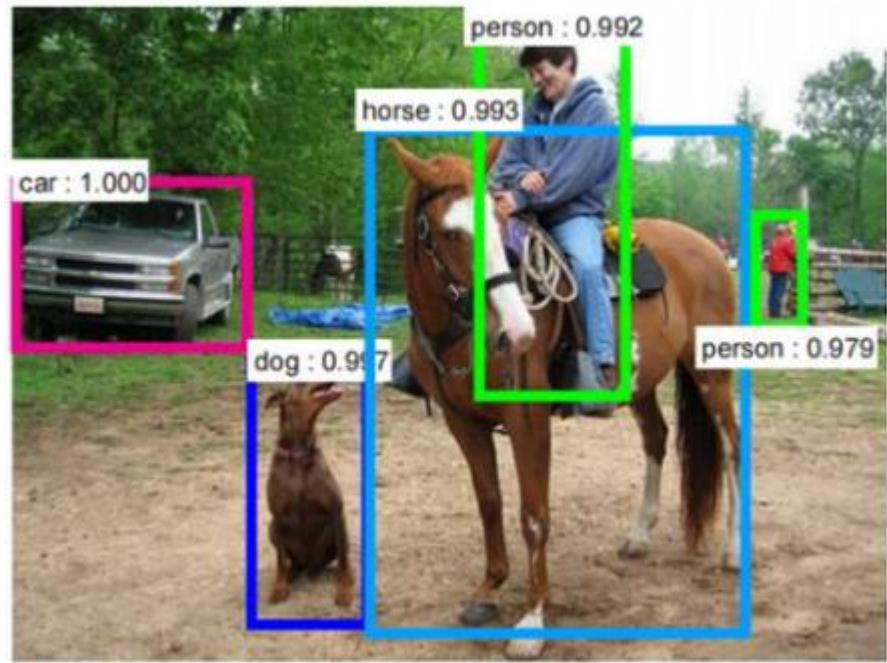


Objektum felismerés

- **Osztályozás:** minden képhez egyetlen címke tartozik
 - Azaz a képen egyetlen objektum látható, és az az egész képet betölti, ezért a feladatot osztályozási feladatként tudjuk megfogalmazni
- **Osztályozás + lokalizálás:** egyetlen objektum nem tölti ki a teljes képet vagy több objektum van a képen

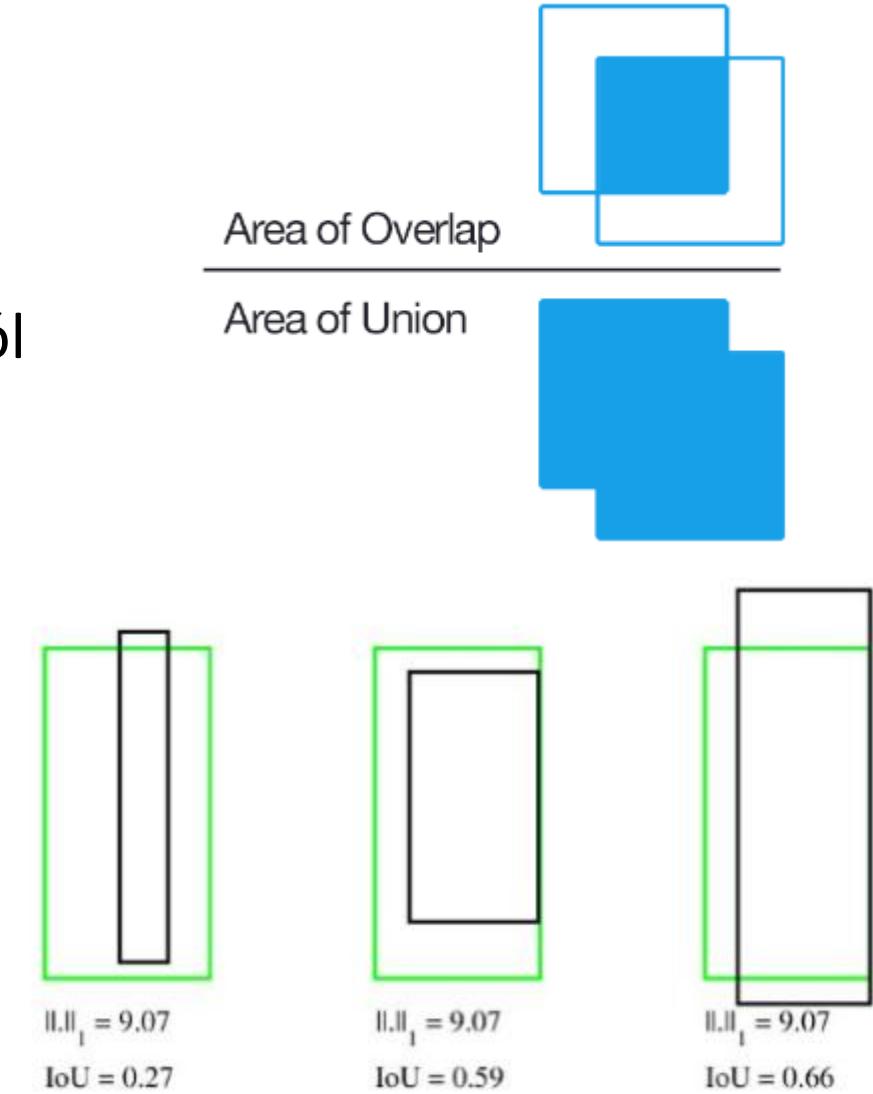
Objektum felismerés

Osztályozás



Objektum felismerés

- Az osztálycímke diszkrét érték, vagy eltaláltuk, vagy nem
- A befoglaló téglalap valódítól való eltérése viszont folytonos érték
- A pontossága mérésére az intersection over union (IoU) értéket szokás használni
- Általában akkor tekintjük helyesnek a detektálást, ha a címke helyes, és $\text{IoU} \geq 0,5$



Objektum felismerés

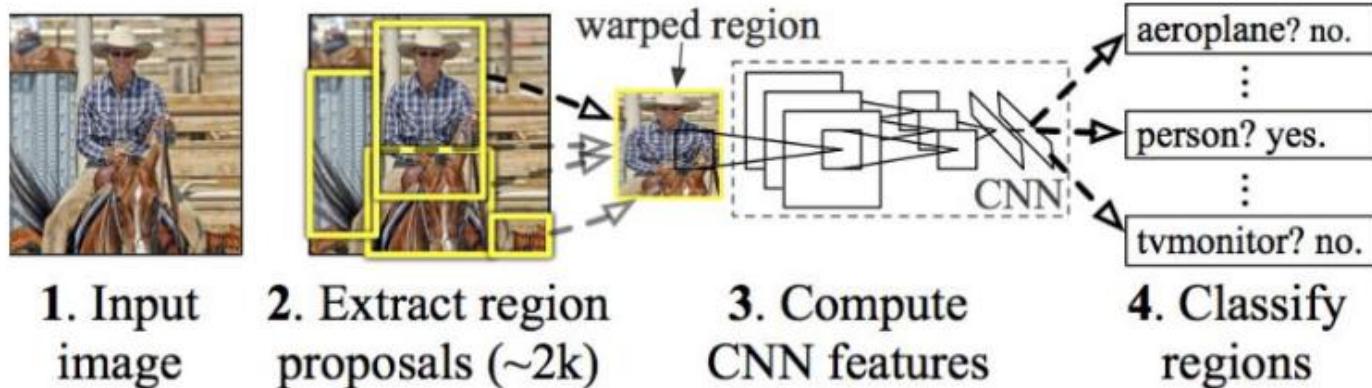
Párhuzamosan kell elvégezni az objektumok osztályozását és detektálását

-Probléma: Az összes lehetséges pozíciót és ablakméretet végig kellene próbálni → túl sok lehetőség

Ráadásul sok alkalmazásban ennek valós időben kellene működnie

Objektum felismerés

- Az **R-CNN** esetén ez egy hierarchikus képszegmentáló-klaszterező algoritmus, amely képeként kb. 2000 régiójavaslatot állít elő
- A javasolt régiókat egységes méretűre/arányúra konvertáljuk 3 különböző tanuló algoritmusból áll össze, ezek külön-külön vannak tanítva:
 - Ezeket egy **CNN** segítségével feldolgozzuk (**jellemzőkinyerés**)
 - Az **osztályozást** egy **lineáris SVM**-mel végezzük
 - A régiók utólagos pontosítása is lehetséges (ANN)



Objektum felismerés

Fast R-CNN

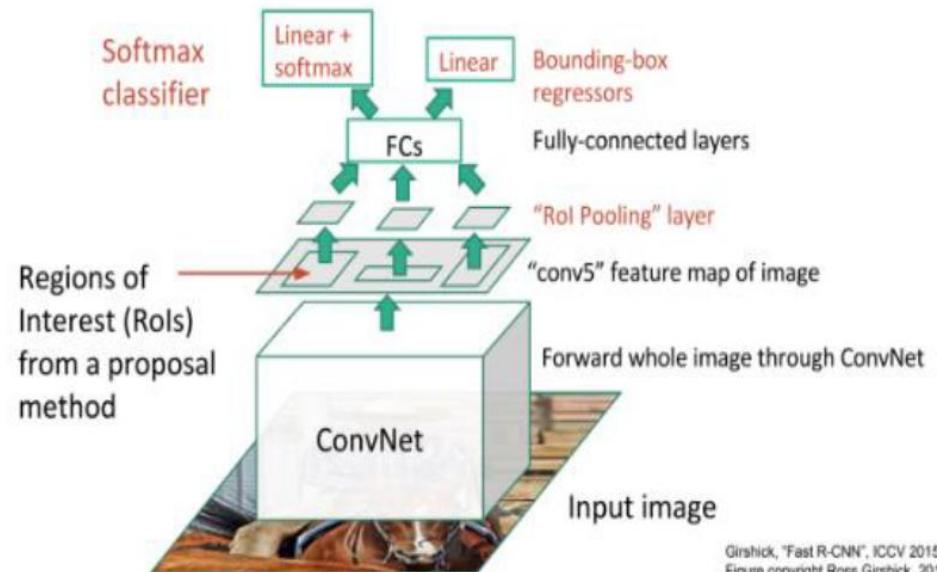
A Fast R-CNN modellben két módosítást javasoltak.

- a három tanuló egyesítése egyetlen tanulómodellé. Ehhez csak az SVM-et kellett lecserélni egy fully connected osztályozó rétegre
- **R-CNN lassú**, mert **egyetlen képen 2000-szer** kell lefuttatni a CNN jellemzőkinyerést (minden javasolt régiót külön-külön) → Fast R-CNN-nél nem kell minden régióra külön kiszámolni a CNN-részt

Objektum felismerés

Fast R-CNN

- A teljes képet ráeresztjük a a CNN-re **egyszer**
- Majd a kapott konvolúciós jellemzőkből vágjuk ki az egyes régiókat leíró részeket
- ROI Polling layer: 7x7-es tulajdonság térkép kinyerése minden ROI régióhoz



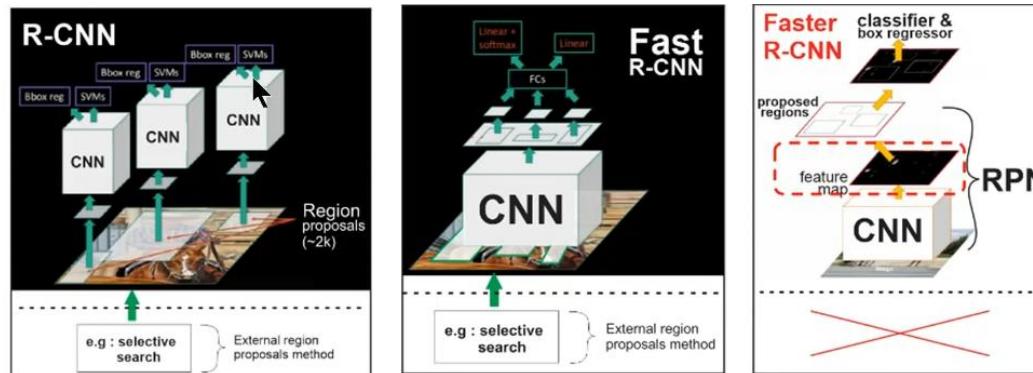
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015.

Objektum felismerés

Faster R-CNN

- A fast R-CNN annyival gyorsabb az R-CNN-nél, hogy a futási időt itt már a régiójavasló algoritmus dominálja
- A faster R-CNN (2016) megpróbálja a régiójavaslatokat is egy neuronháló segítségével előállítani (Region Proposal Network)

Ehhez a konvolúciós rétegek kimenetét használja inputként, nem az eredeti képet



	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image	50 seconds	2 seconds	0.2 seconds
Speed-up	1x	25x	250x
mAP (VOC 2007)	66.0%	66.9%	66.9%

Optikai karakterfelismerés

A mesterséges intelligencia jelfeldolgozó és generalizációs képességeit kiaknázva képes magas hatékonysággal nyomtatott, papír alapú dokumentumokon lévő karaktereket felismerni.

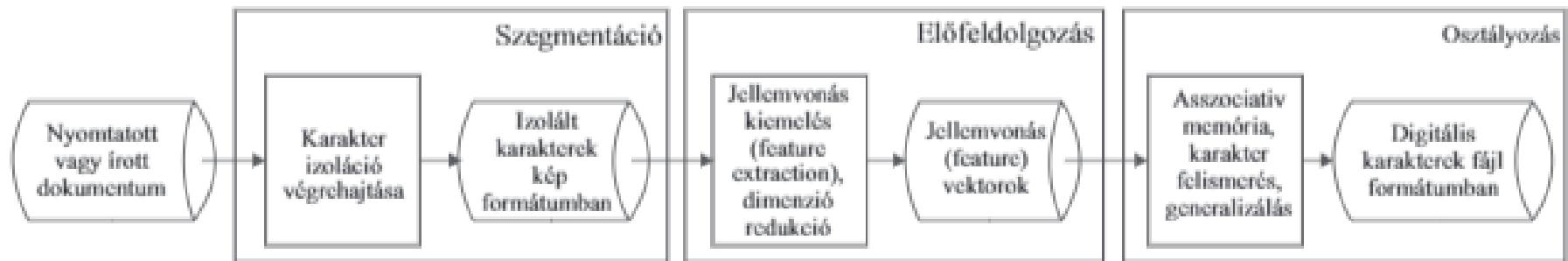
Nehézségek:

- Nagy zajarány a papíralapú dokumentumnál: pl. apró folt a papíron, tinta-elmosódás, tintahiány, homályos háttér, apró gyűrődés a papíron, túl közelí vagy egybeolvadó betűk, betű dőlésszögének ingadozása stb.
- Kézírás még nehezebb a személyiségejegyek miatt

Optikai karakterfelismerés

- A nyomtatott latin vagy latin-rokon karakterek felismerése megoldott problémának tekinthető, az OCR-rendszerek igen hatékonyan képesek felismerni ezeket.
- A kézírás felismerése azonban még napjainkban is aktív kutatási terület, hiszen ez jóval összetettebb feladat.

Optikai karakterfelismerés



Optikai karakterfelismerés

Szegmentáció

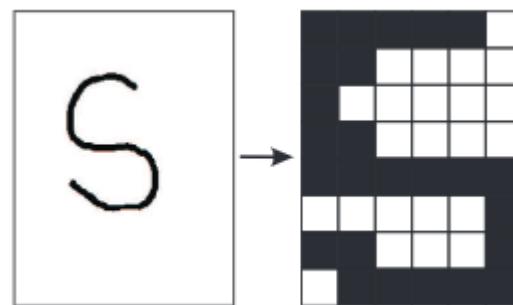
- Cél: a karakterek közötti éles határ megtalálása annak érdekében, hogy téves minták ne kerüljenek osztályozásra (pl két fél karakter).
- A szegmentáció feladata lehet az is, hogy a karakterdőlésszögeket, karakterméreteket normalizálja.
- A szegmentáció további feladata az is, hogy kiszűrje a csak karaktereket tartalmazó szöveges részeket.
(például képek is lehetnek a dokumentumban)



Optikai karakterfelismerés

Optikai előfeldolgozás

a bemeneti minta komplexitásának csökkentésére szolgál, és annak legjellemzőbb vonásait emeli ki → komplexitáscsökkentéssel járó digitalizáció



Optikai karakterfelismerés

Osztályozás

Például mesterséges neurális hálózattal

