9. Szegmentálás

Kató Zoltán

Képfeldolgozás és Számítógépes Grafika tanszék
SZTE
(http://www.inf.u-szeged.hu/~kato/teaching/)

Szegmentálás célja

- **Partícionáljuk a képet koherens “objektumokra”**
- **Nincs egzakt definíció (alkalmazásfüggő, szubjektív)**

image | human segmentation

http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

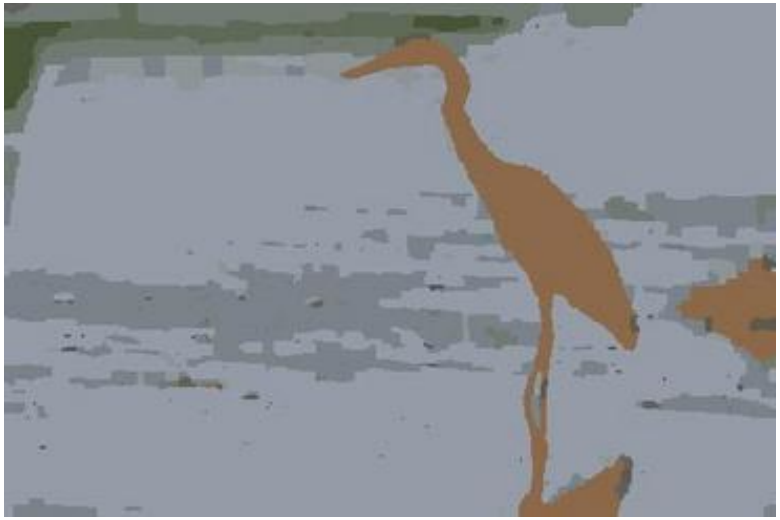
Szegmentálási példák

Terminológia

- Szegmentálás, csoportosítás, perceptuális rendezés (*Segmentation*, *grouping*, *perceptual organization*):
  - az összetartozó jellemzők összegyűjtése, csoportosítása
- Illesztés (*Fitting*):
  - modell társítása a megfigyelt jellemzőkhöz
- Szegmentálás felülről lefelé (*Top-down segmentation*):
  - a pixelek összetartozásának alapja a közös objektumba tarztozás (ehhez ismerni kell a keresett objektumot)
- Szegmentálás lentről fölfelé (*Bottom-up segmentation*):
  - a pixelek összetartozásának alapja az azonos vizuális jellemző (pl. hasonló szín)

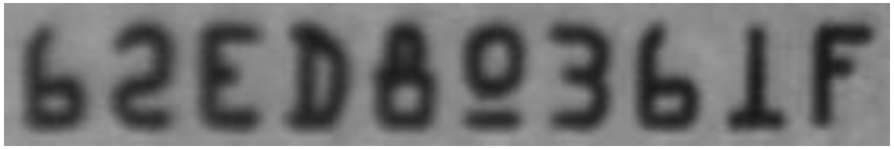
Szegmentálási módszerek

- Szegmentálás mint klaszterezés
  - Küszöbölés, K-means, Mean shift
- Régió alapú szegmentálás
  - Régiónövelés, split & merge
- Kontúr alapú szegmentálás
  - Canny, kontúrkövetés, alakzatillesztés

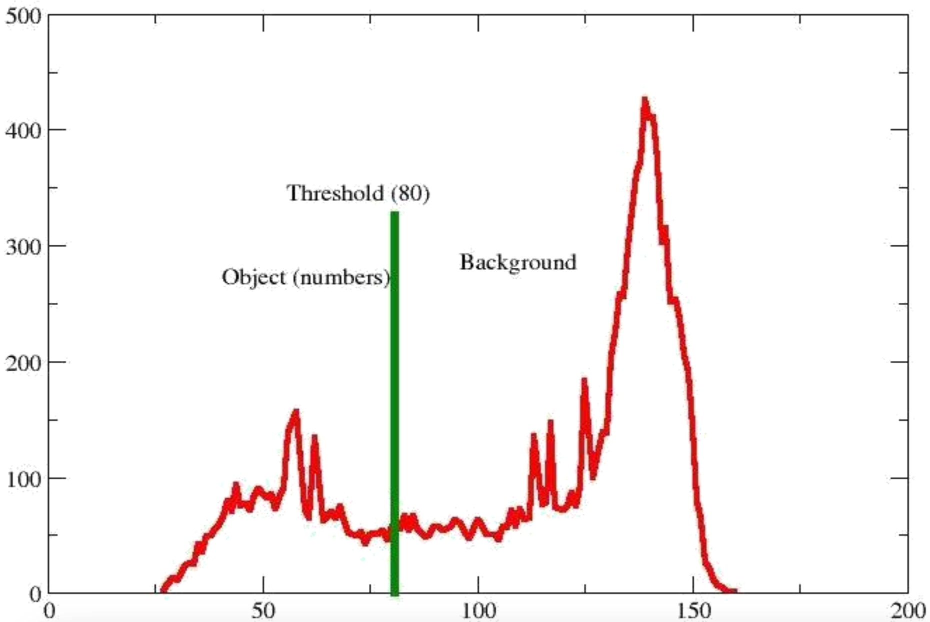


GLOBÁLIS ÉS LOKÁLIS KÜSZÖBÖLÉS

Küszöbölés



- Alapfeltevések:
  - Az objektum és a háttér eltérő intenzitású (nagy kontraszt)
  - Az objektum és a háttér önmagában homogén intenzitású

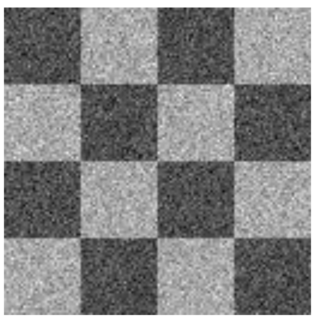


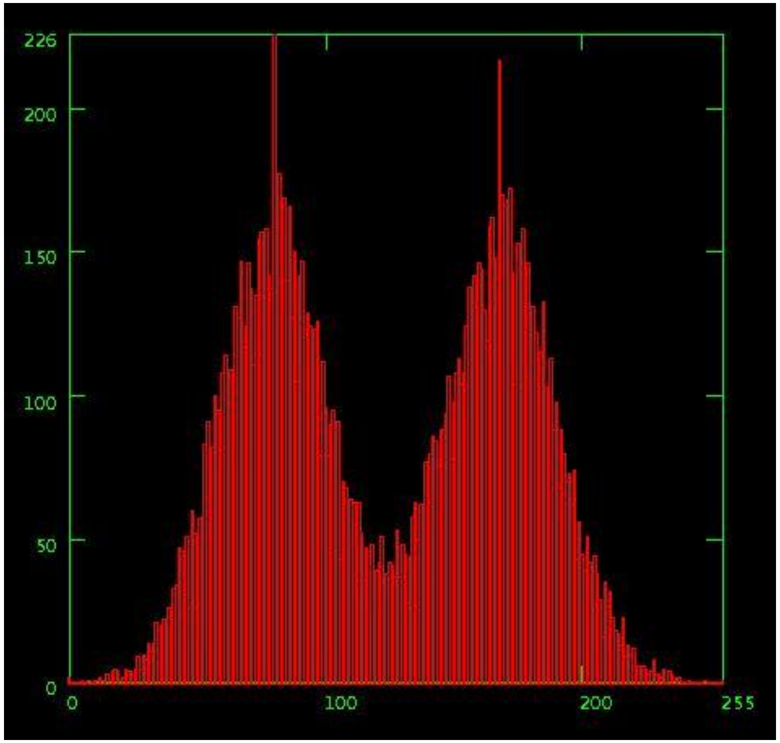
- Például: sötét objektum világos háttéren:
  - ***f(x,y)*** - input kép;
    t(x,y) - szegmentált kép;
    T=küszöbérték

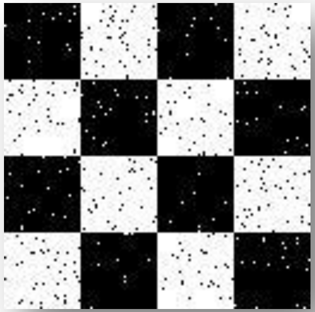
f(x,y)<=T ➔ ***t(x,y)=1*** (objektum)
f(x,y)>T ➔ ***t(x,y)=0*** (háttér)

Küszöbérték meghatározása

- Zajos kép esetében nehéz (~lehetetlen) kielégítő küszöbértéket meghatározni → a szegmentálás inhomogén lesz.
  - Zajszűrés jelenthet megoldást, ha nem túl nagy a zaj…







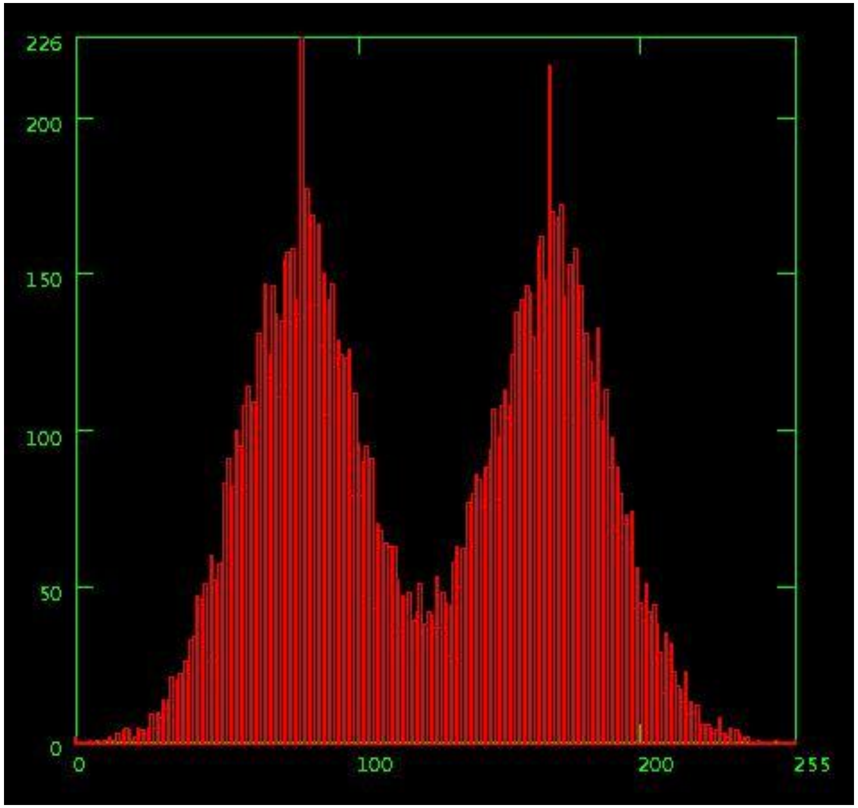
Küszöbérték meghatározása

- Lehet az input képtől függetlenül, manuálisan rögzített érték
  - Egyszerű
  - Kontrollált környezetben jól használható (ipari alkalmazások)
- Adaptív eljárások, melyek az input képhez automatikusan választják ki az optimális küszöbértéket:
  - Globális hisztogram-ból klaszterezéssel számított érték
    - Isodata algoritmus (Yanni)
    - Otsu algoritmus
  - Lokálisan változó érték (egyenetlen megvilágítás)
    - Niblack algoritmus

Isodata algoritmus (Yanni)

- **Jól használható, ha az előtér és a háttér kb. ugyanannyi pixelből áll.**

  1. **Inicializálás: a hisztogramot két részre osztjuk (célszerűen a felezőponton): T_0**
  2. **kiszámítjuk az objektum valamint a háttér intenzitásának középértékét: M_i, m_i**
  3. **Az új küszöbérték a két középérték átlaga: $T_i=(M_i+m_i)/2$**
  4. **Vége ha a küszöbérték már nem változik: $T_{k+1}=T_k$**



Otsu algoritmus

- A bemeneti kép *L* szürkeárnyalatot tartalmaz
- a normalizált hisztogram minden *x* szürkeértékhez megadja az előfordulási gyakoriságát (valószínűségét): p_x
- Az algoritmus lényege: keressük meg azt a *T* küszöbszámot, amely maximalizálja az objektum-háttér közötti varianciát.
- Az előtér/háttér pixelek gyakorisága és középértéke:

$$\text{háttér:}\quad B(T)=\sum_{x=1}^{T}p_x \qquad \text{objektum:}\quad 1-B(T)=\sum_{x=T+1}^{L}p_x$$

$$\text{legyen } m(T)\equiv\sum_{x=1}^{T}xp_x \;\Rightarrow\; \text{a teljes kép középértéke:}\quad \mu\equiv m(L)=\sum_{x=1}^{L}xp_x$$

$$\text{háttér középértéke:}\quad \mu_B=\frac{m(T)}{B(T)};\quad \text{objektum középértéke:}\quad \mu_O=\frac{\mu-m(T)}{1-B(T)}$$

Otsu algoritmus

- **Az előtér/háttér pixelek szórása:**

$$\sigma_B^2 = \frac{1}{B(T)} \sum_{x=1}^{T} (x - \mu_B)^2 p_x, \quad \sigma_O^2 = \frac{1}{1-B(T)} \sum_{x=T+1}^{L} (x - \mu_O)^2 p_x$$

- **A teljes kép szórása:**

$$\sigma^2 = \sum_{x=1}^{T} (x - \mu)^2 p_x + \sum_{x=T+1}^{L} (x - \mu)^2 p_x = \ldots =$$

$$= \underbrace{B(T)\sigma_B^2 + (1-B(T))\sigma_O^2}_{=\sigma_W^2(T),\ \text{osztályon belüli varianciától függ}} + \underbrace{(\mu_B - \mu)^2 B(T) + (\mu_O - \mu)^2 (1-B(T))}_{=\sigma_C^2(T),\ \text{osztályok közötti varianciától függ}} \equiv \sigma_W^2(T) + \sigma_C^2(T)$$

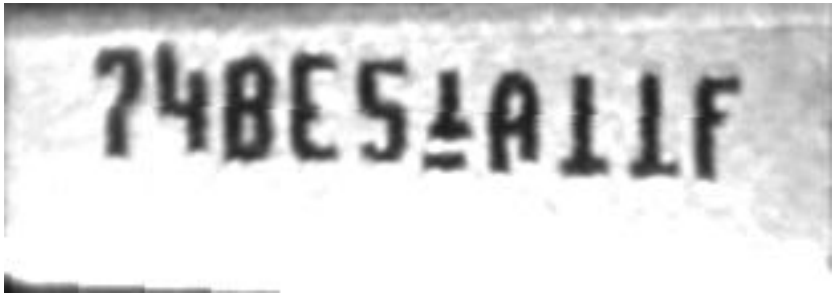
- **nyilván σ^2 konstans, és T-t úgy kell beállítani, hogy $\sigma_C^2(T)$ a lehető legnagyobb legyen:**

$$\sigma_C^2(T) = \ldots = \frac{(\mu(T) - \mu B(T))^2}{B(T)(1-B(T))}$$

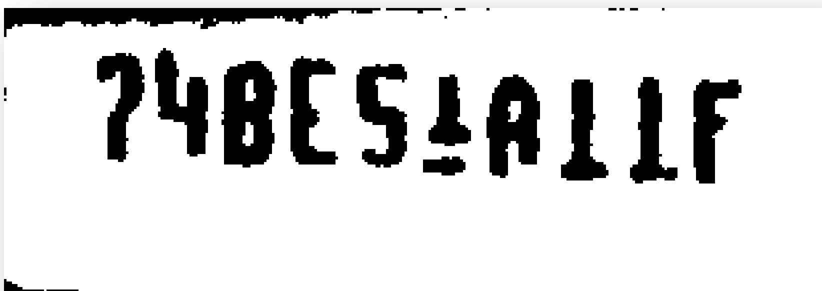
Otsu algoritmus

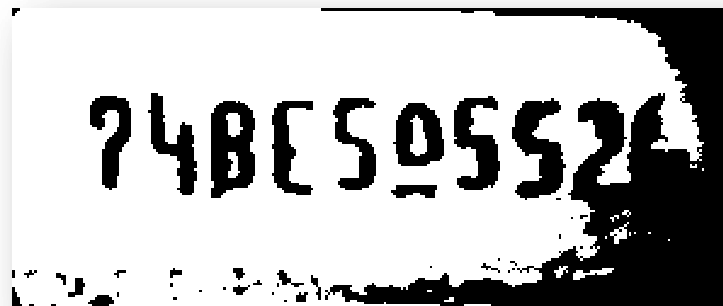
- **A hisztogram elejéről kezdve nézzük meg minden szürkeértéket, mint lehetséges küszöböt:**
  - **Számoljuk ki $\sigma_C^2(T)$ értékét $\mu(T)$ és $B(T)$ segítségével**
  - **mindaddig növeljük T értékét, amíg $\sigma_C^2(T)$ növekszik.**
- **Ez az algoritmus feltételezi, hogy $\sigma_C^2(T)$ -nek egyetlen maximuma van!**

Példák

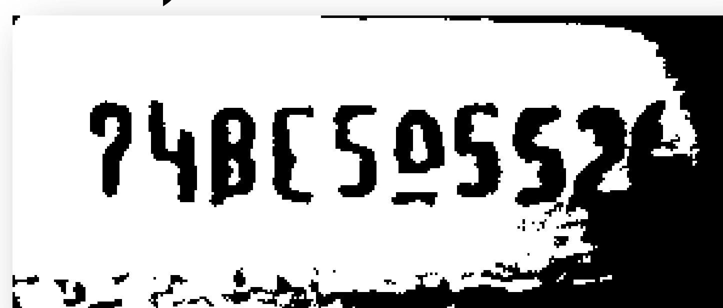


Original





Isodata (Yanni)

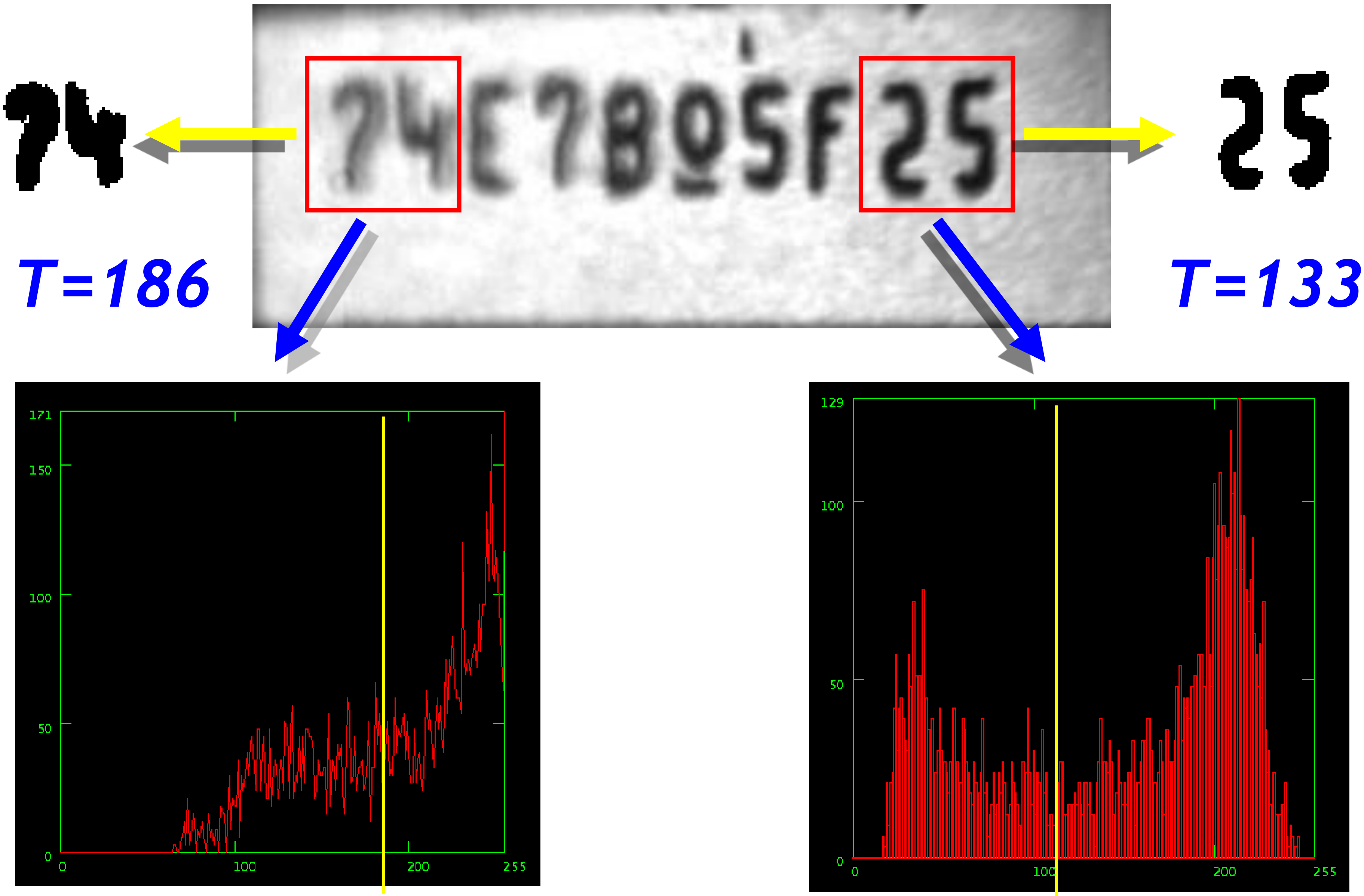


Otsu

Lokálisan változó küszöbérték

- **Mit tehetünk abban az esetben, ha az objektum vagy a háttér nem homogén?**
- **Amennyiben az objektum és a háttér kontrasztja lokálisan továbbra is nagy, akkor alkalmazhatunk lokális küszöbölést.**

Lokális hisztogram + Otsu

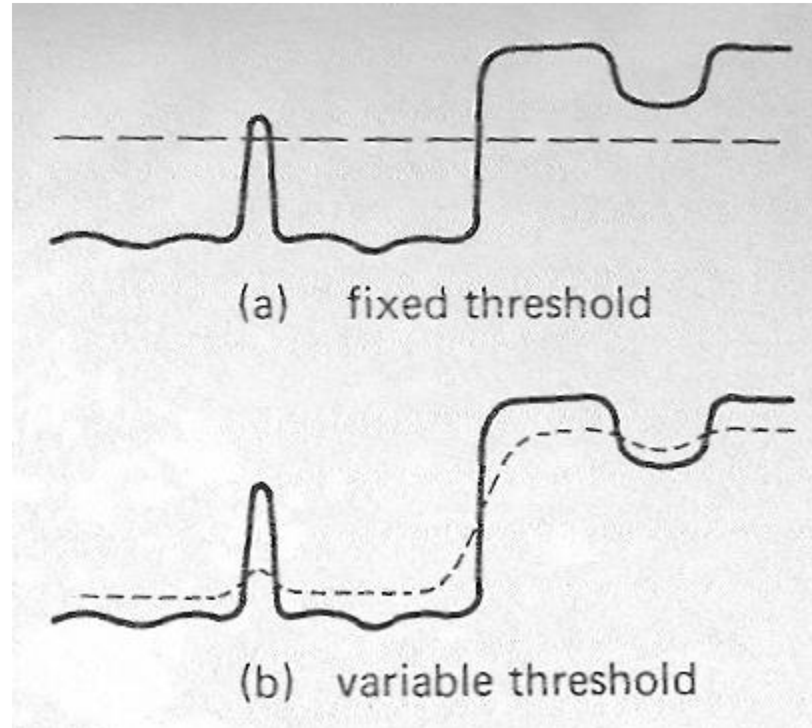


Niblack algoritmus

- **Egyetlen küszöb nem elegendő az objektum és háttér szétválasztásához**
- **➔változó küszöbérték (*T(i,j)*) kell, amely követi az intenzitásváltozásokat:**

$$T(i,j) = \mu(i,j) + k\sigma(i,j)$$

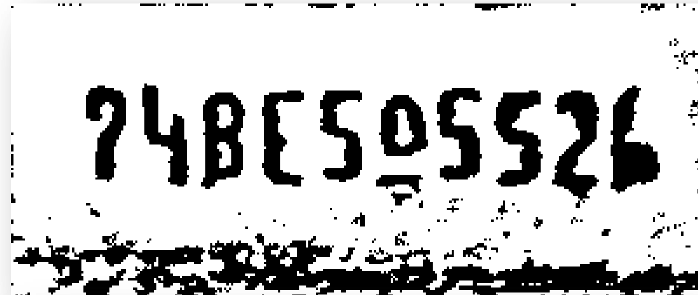
- ➢ ***(i,j)* adott környezetében:**
  - – ***μ(i,j)* - középérték**
  - – ***σ(i,j)* - szórás**
- ➢ ***k* – mennyire vegyük figyelembe a szórást**
  - – **Sötét objektum ➔ k< 0,**
  - – **világos objektum ➔ k>0**
  - – **Általában |*k*| ~ *0.2***



A környezet mérete: elég kicsi ahhoz, hogy a lokális részleteket megőrizze, de elég nagy ahhoz, hogy a zajt elnyomja (általában ***15X15***).

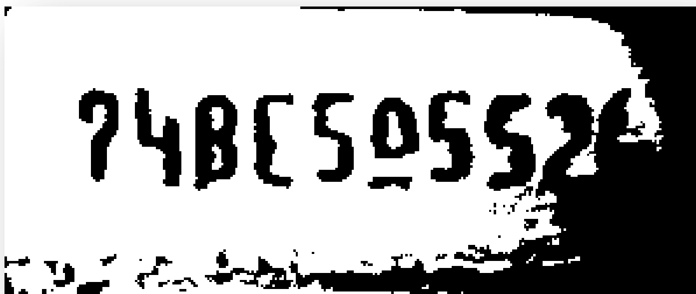
Niblack algoritmus: példák

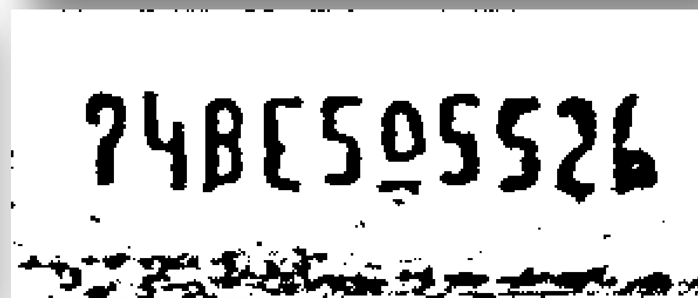
original



Niblack
k=-0.2
30X30

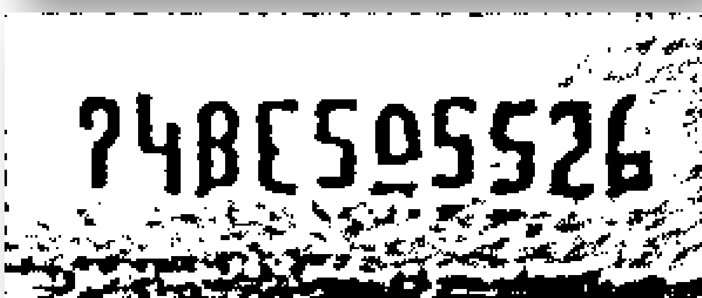
Otsu

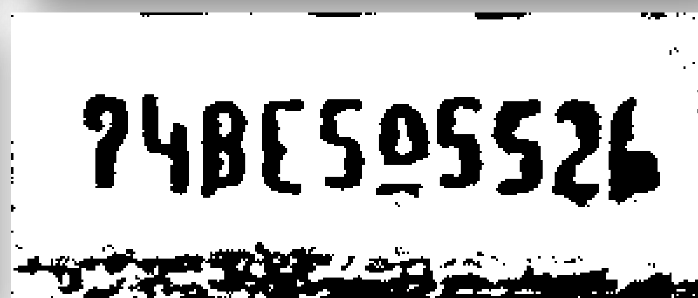




Niblack
k=-0.5
30X30

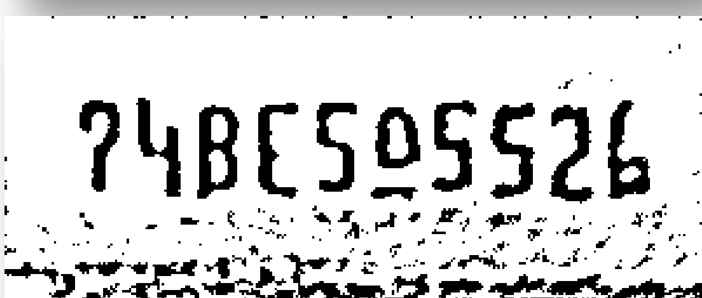
Niblack
k=-0.2
15X15

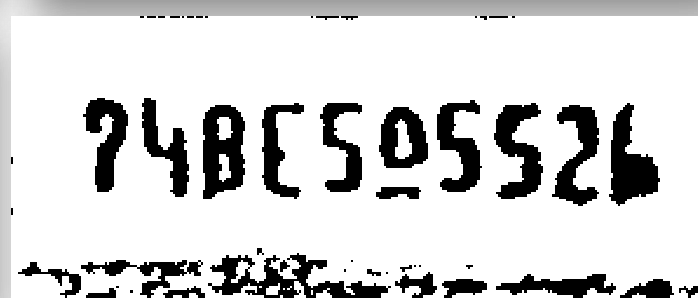




Niblack
k=-0.2
60X60

Niblack
k=-0.5
15X15





Niblack
k=-0.5
60X60

Postprocessing

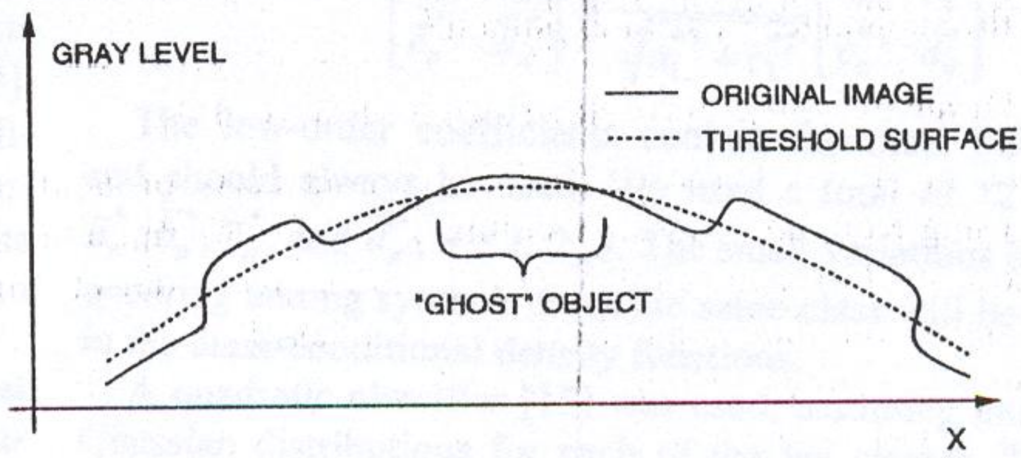
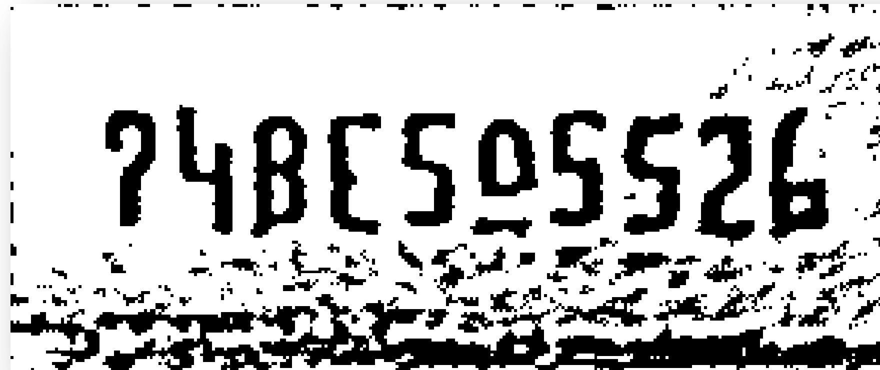


Fig. 4. Cross-section through an image, illustrating how "ghost" objects might occur. Based on a figure in Yanowitz and Bruckstein [24].

- **"ghost" objektumok eltüntetése**
  - **Számitsuk ki az átlagos gradiens értékét az objektumok élei mentén**
  - **Töröljük le azokat az objektumokat, amelyeknek az átlagos gradiense egy adott küszöbérték alatt van**

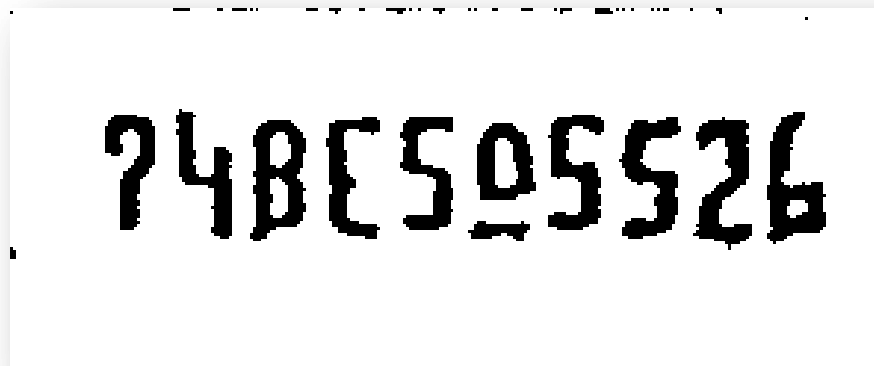
Postprocessing példák

original



Niblack k=-0.2 15X15

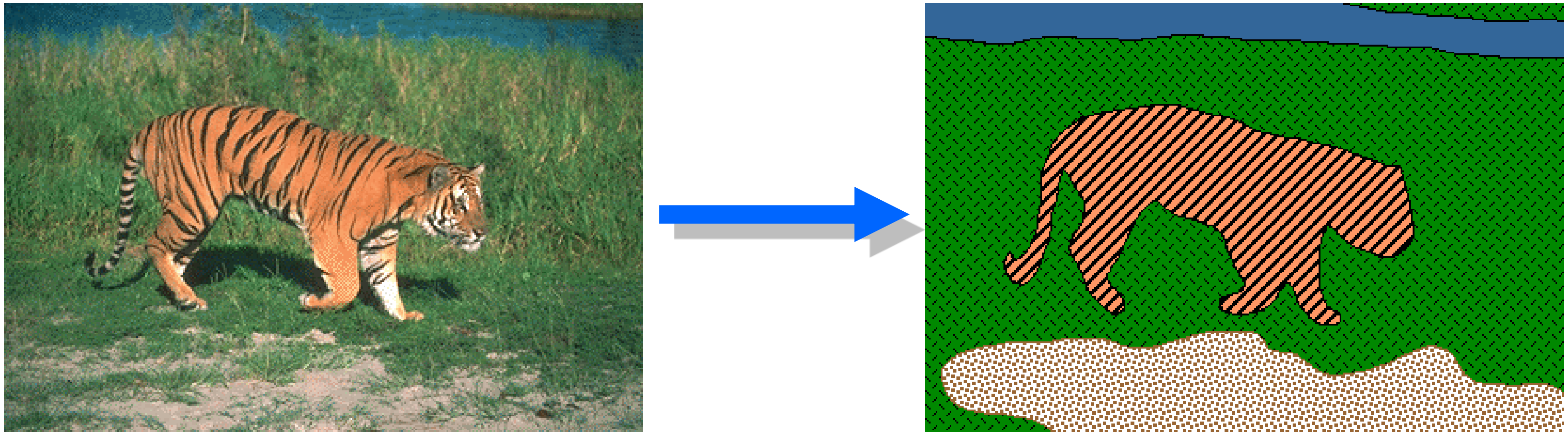
gradiens



postprocess után

RÉGIÓ ALAPÚ SZEGMENTÁLÁS

Szegmentálási kritériumok

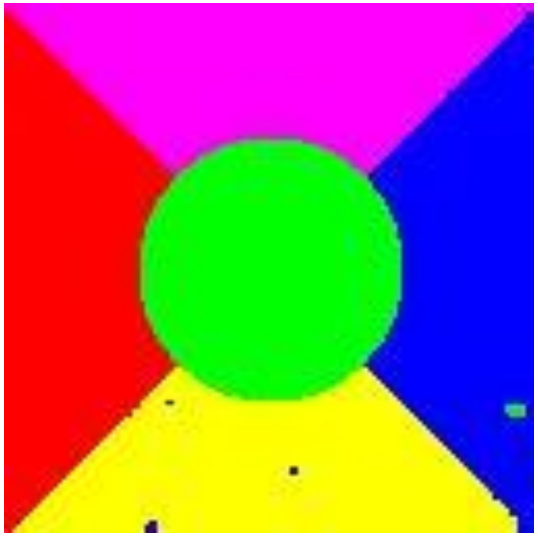


- **Particionáljuk a képet az alábbi kritériumokat kielégítő régiókba (Pavlidis):**

1. $\cup S_i = S$ — A régiók lefedik a teljes képet
2. $S_i \cap S_j = \emptyset,\ i \neq j$ — A régiók nem átfedőek
3. $\forall\ S_i\ ,\ P(S_i) = \text{true}$ — Minden régió kielégíti a homogenitási kritériumot
4. $P(S_i \cup S_j\ ) = \text{false}$, $i \neq j$, S_i adjacent S_j — Szomszédos régiók uniója nem elégíti ki a homogenitási kritériumot

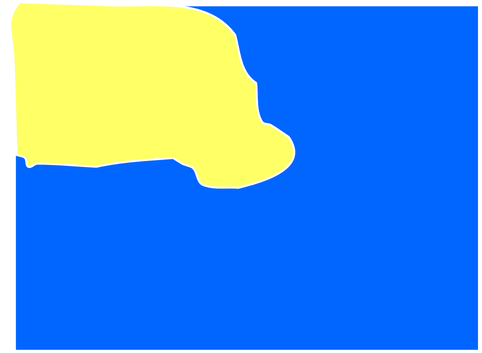
Címkézés

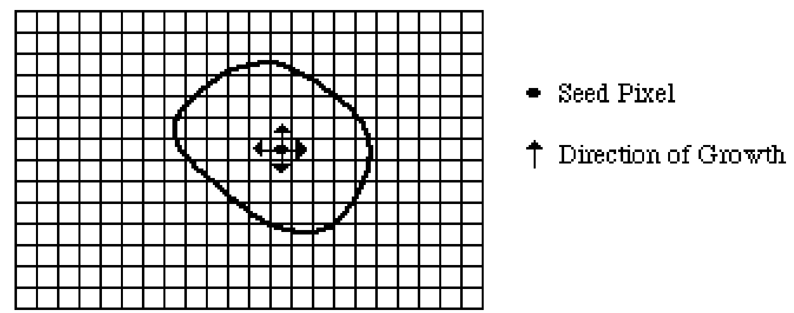
- **A szegmentált képet címkézéssel állítjuk elő:**
  - **Egy adott régióhoz tartozó pixeleket azonos címkével (~szürkeárnyalat) jelöljük meg.**



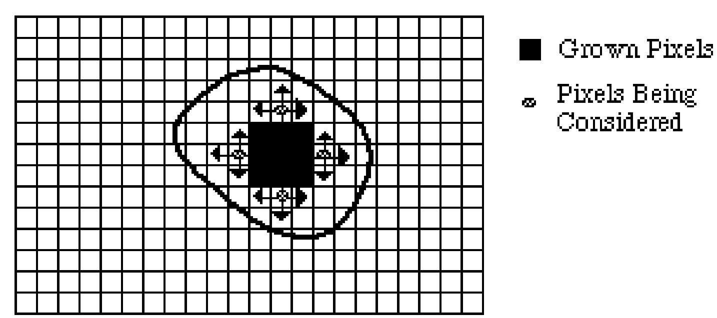
Régiónövelés (Region Growing)

- Inicializáljuk a régió-magokat (seed point)
  - Ezek kiválasztása alkalmazás-függő
  - Lehet egyszerűen az első még nem címkézett pixel
- Növeljük a régiókat a szomszédos pixelek hozzáadásával a **homogenitási kritériumnak** megfelelően
  - Szín távolság a szomszédoktól
  - A régión belüli inhomogenitás mérésével
  - Régió mérete vagy alakja, …





(a) Start of Growing a Region



(b) Growing Process After a Few Iterations

Régiónövelés

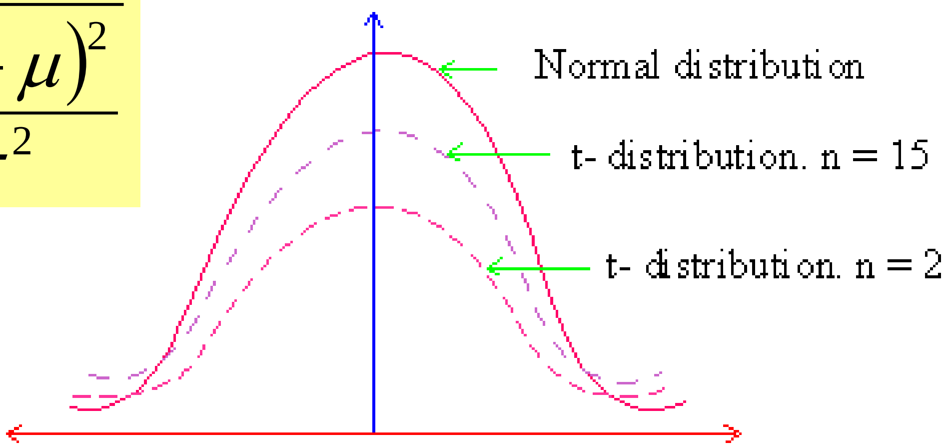
- **Homogenitási kritériumként alkalmazhatjuk pl. az alábbi statisztikai tesztet:**
  - **Legyen *R* az aktuális régió, amely *N* pixelből áll.**
  - ***P* jelölje a vizsgált szomszédos pixel intenzitását.**

$$\mu = \frac{1}{N} \sum_{(x,y)\in R} I(x,y) \qquad \sigma^2 = \frac{1}{N-1} \sum_{(x,y)\in R} (I(x,y) - \mu)^2$$

  - **A τ statisztika:**

$$\tau = \sqrt{\frac{N}{N+1} \frac{(P-\mu)^2}{\sigma^2}}$$

  - **(n-1) szabadsági fokú Student’s eloszlást követ**

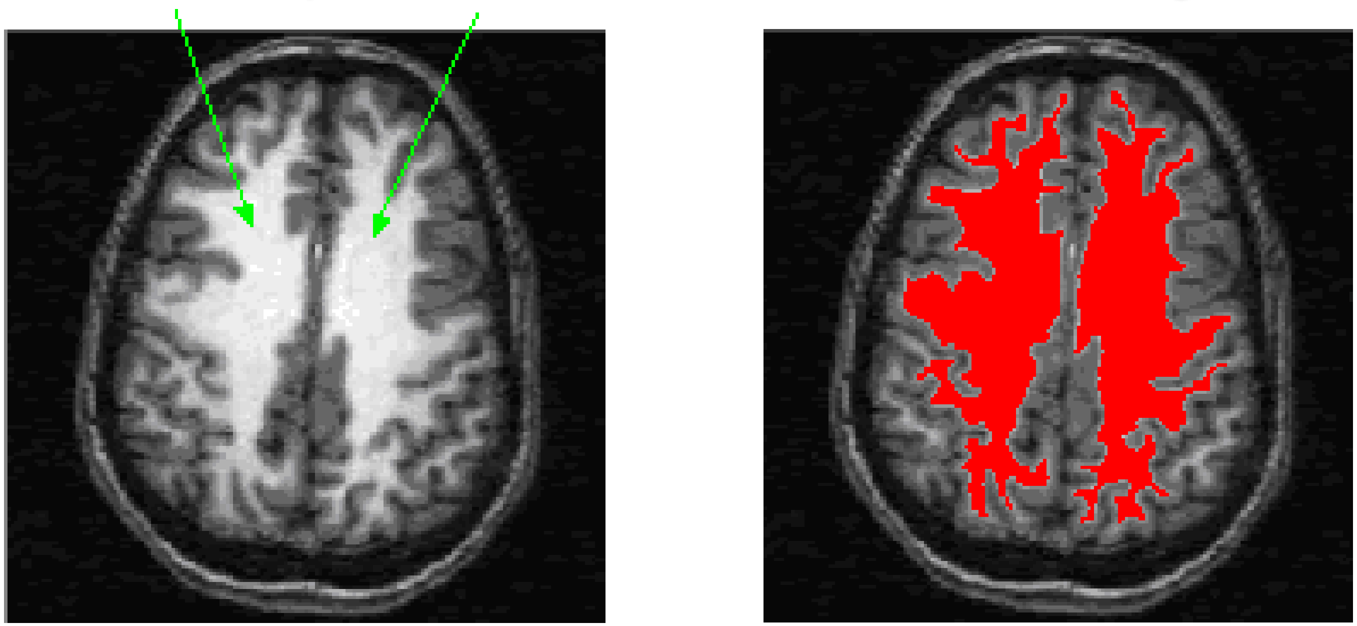


Régiónövelés

- **Egy megfelelő *T* küszöbszámot statisztikai táblázatból választhatunk a megfelelő szabadsági fokhoz és konfidencia szinthez**
  - **Ha $\tau \leq T$, akkor a vizsgált pixelt az *R* régióhoz adjuk**
  - **Ha $\tau > T$, akkor a vizsgált pixel egy másik (új) régióhoz tartozik.**

forráspontok

növesztett régiók



fehérállomány szegmentálása MR agyvizsgálaton

Split & Merge

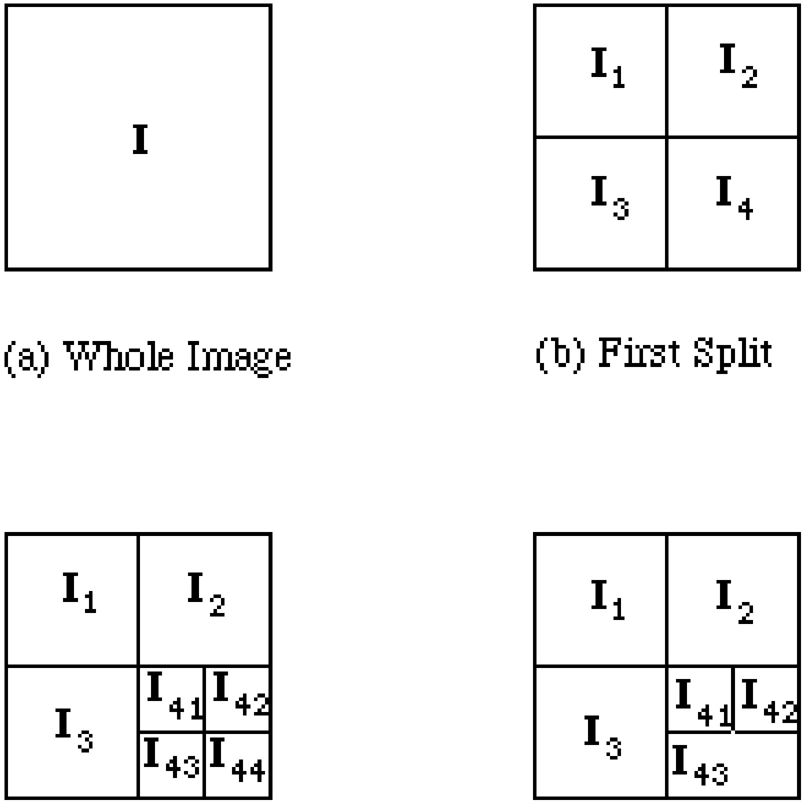
1. P - homogenitási kritérium
2. R_i régiót osszuk fel 4 részre, ha

$$P(R_i) = False$$

3. Vonjuk össze az R_i és R_j szomszédos régiókat, ha

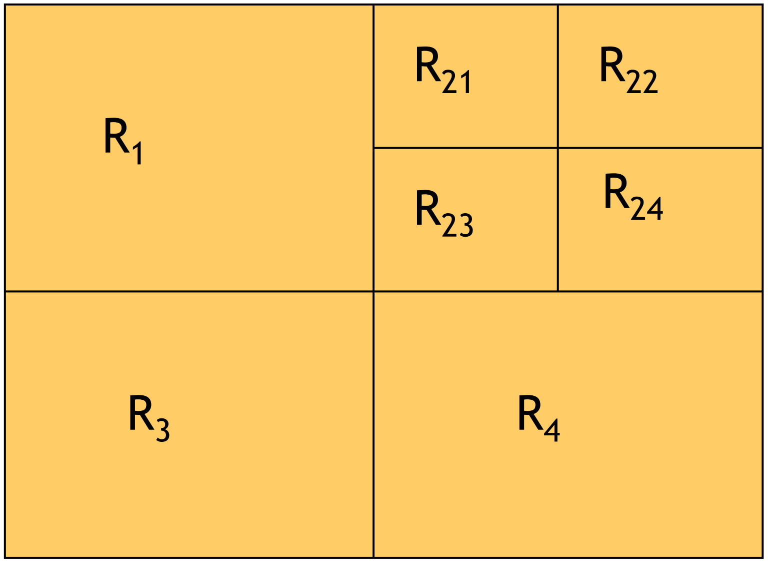
$$P(R_i \cup R_j) = True$$

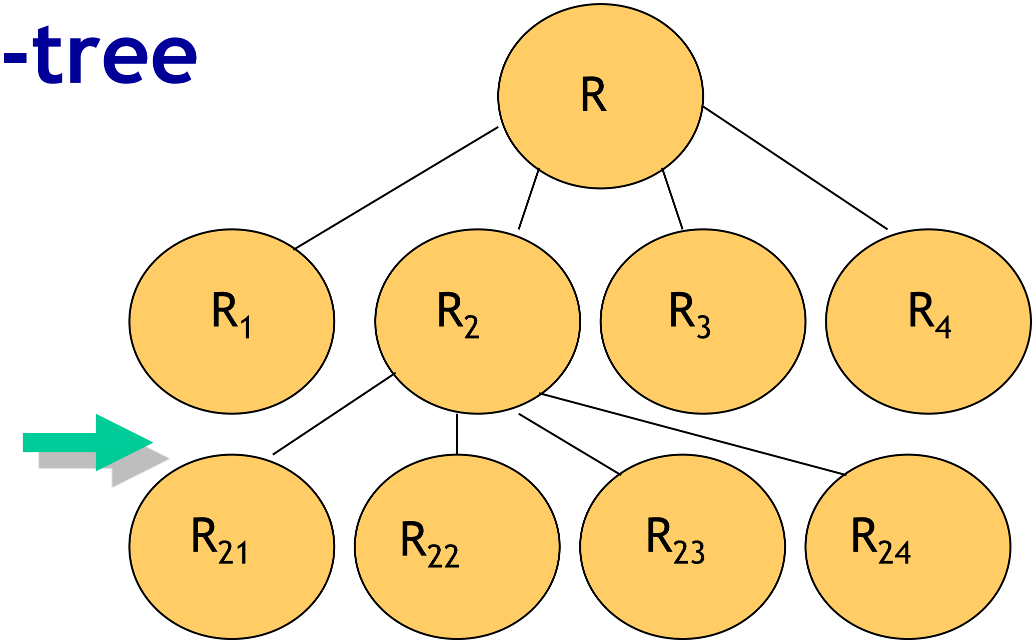
4. Vége, ha nem lehet több régiót feldarabolni vagy összevonni.

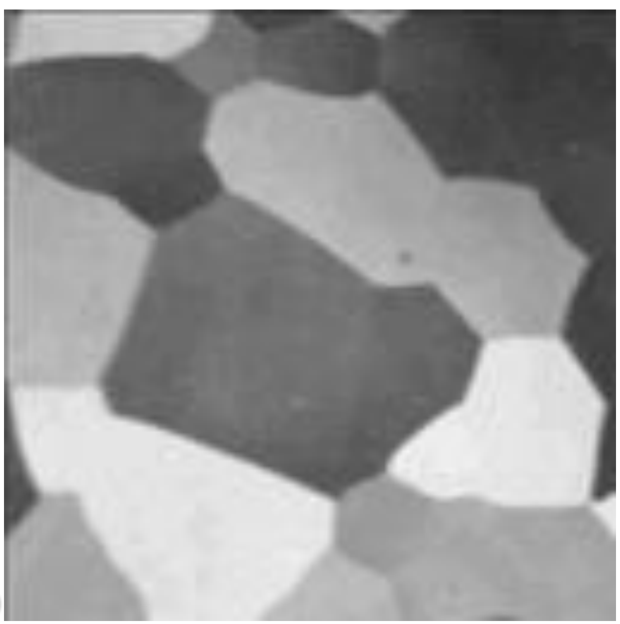


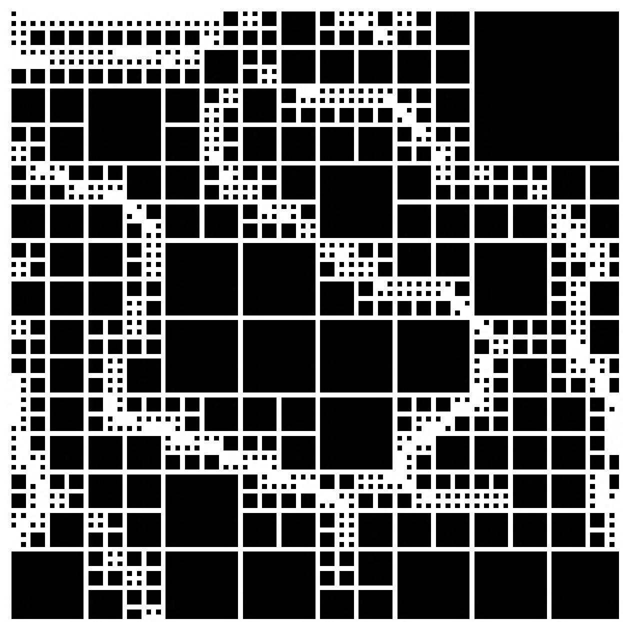
(a) Whole Image (b) First Split (c) Second Split (d) Merge

Split & Merge quad-tree






eredeti kép


dekompozíció

Klaszterezés

- Adott *K* klaszter: C_1,..., C_K .
- A kalszterek középértékét jelölje μ_1,..., μ_K.
- LSE (Least Square Error):

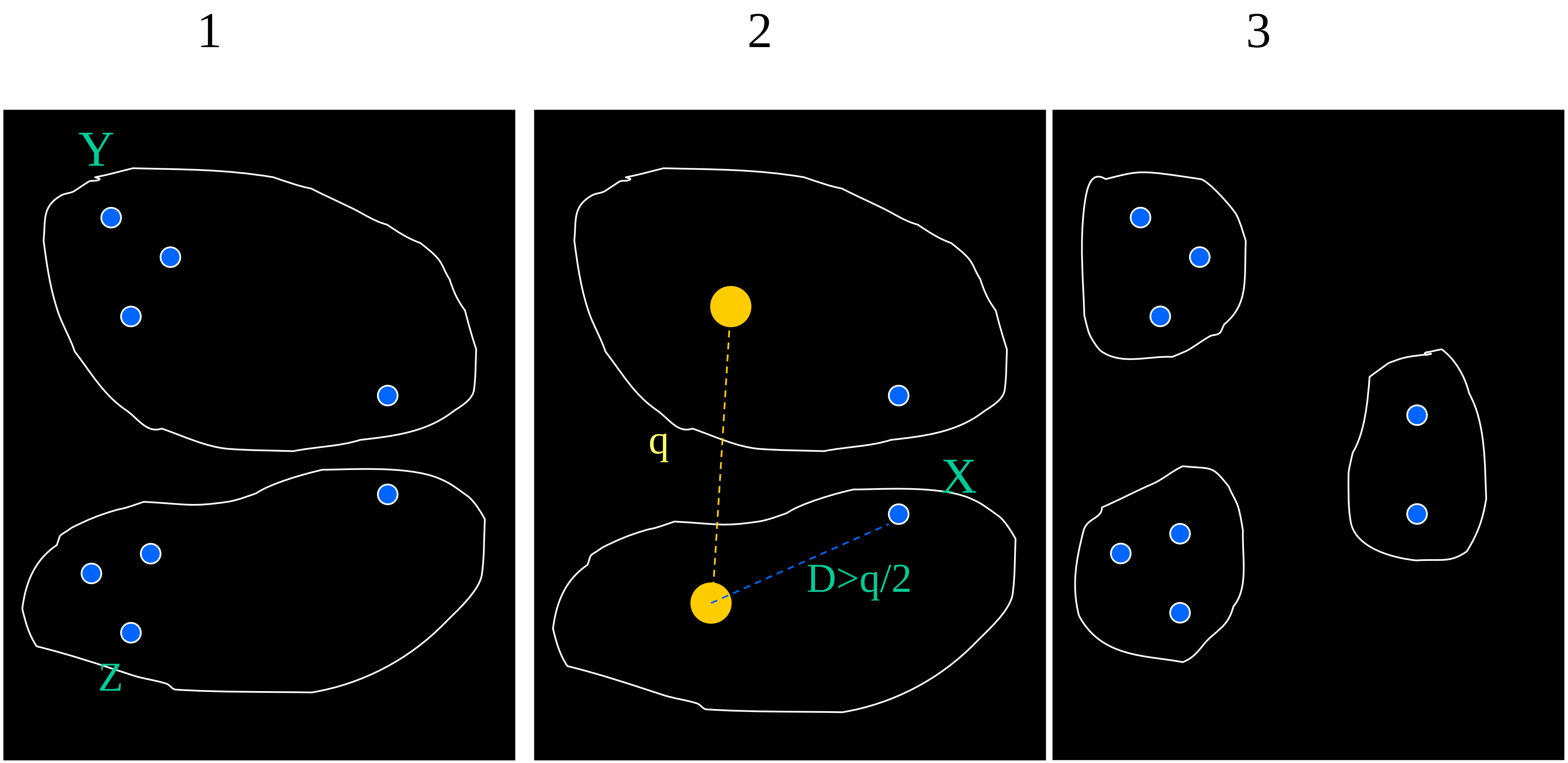
$$D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left\| x_i - \mu_k \right\|^2$$

- Az összes lehetséges particionálás közül válasszuk azt, amelyik minimalizálja *D*-t.
  - Sok lehetséges particionálás ➔ kereséssel nem határozható meg a minimum.

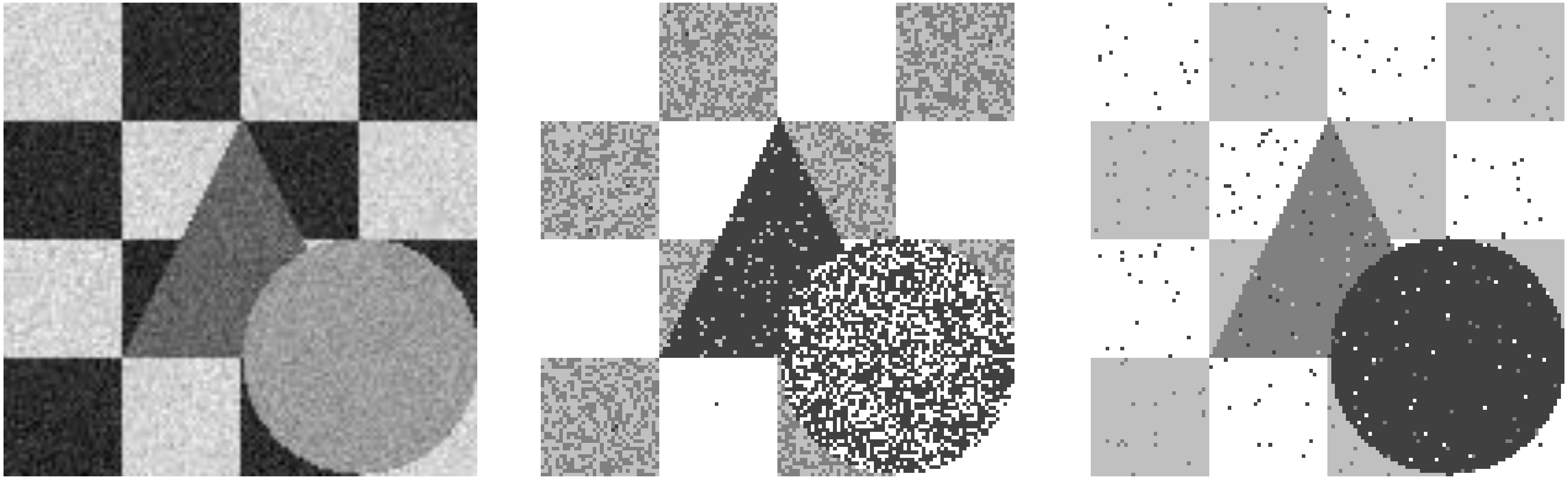
K adaptív meghatározása (Meng-Hee Heng)

1. Válasszunk 2 pontot (*Y* és *Z*), amelyek a legmesszebb helyezkednek el egymástól a feature-térben. Ezek lesznek a kezdeti klaszter-középértékek.
2. A kép minden pontját rendeljük ahhoz a klaszterhez, amelynek a középértéke a legközelebb van.
3. Legyen *D* a maximális távolság, amely egy pont és az ő klaszter-középpontja között van. Jelölje *X* ezt a pontot.
4. Legyen *q* az átlagos távolság a klaszter-középpontok között.
5. Ha $D>q/2$, akkor legyen *X* egy új klaszter-középpont.
6. Ha új klaszter-középpont keletkezett, akkor ➔ 2.

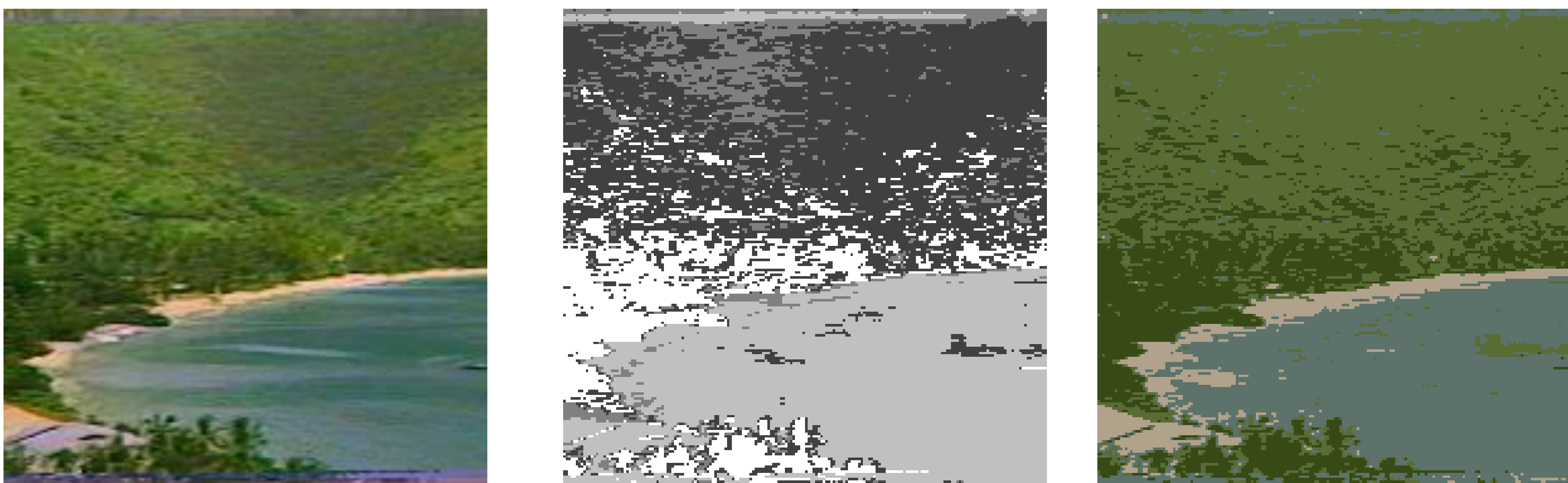
K adaptív meghatározása (Meng-Hee Heng)



K-means példa

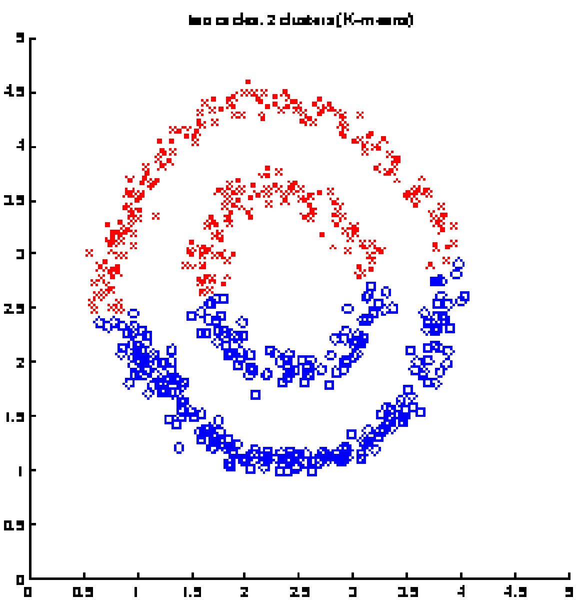


K=4

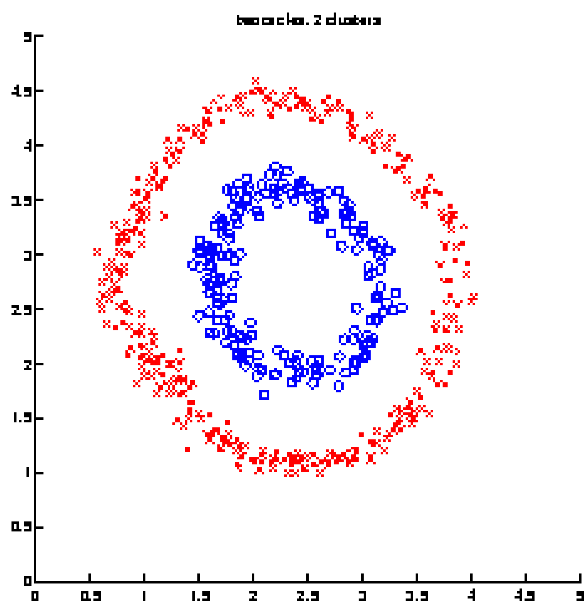


Problémák

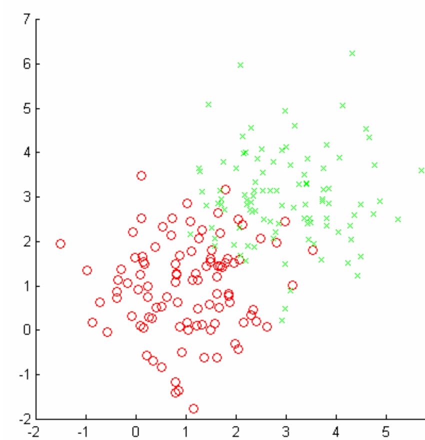
- **Nem konvex klaszterek detektálására közvetlenül nem használható.**
  - ➔Spektrális klaszterezés



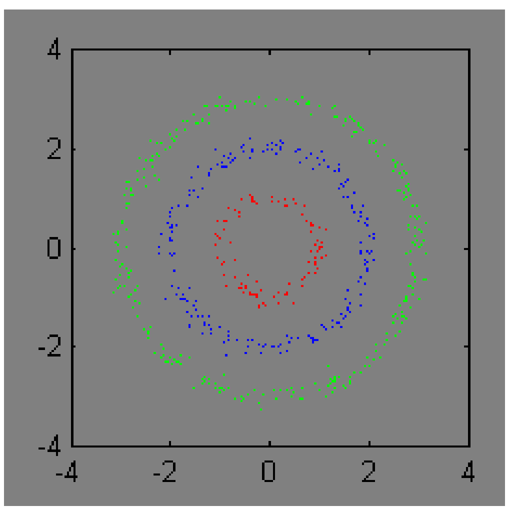
K-means



Spectral



OK

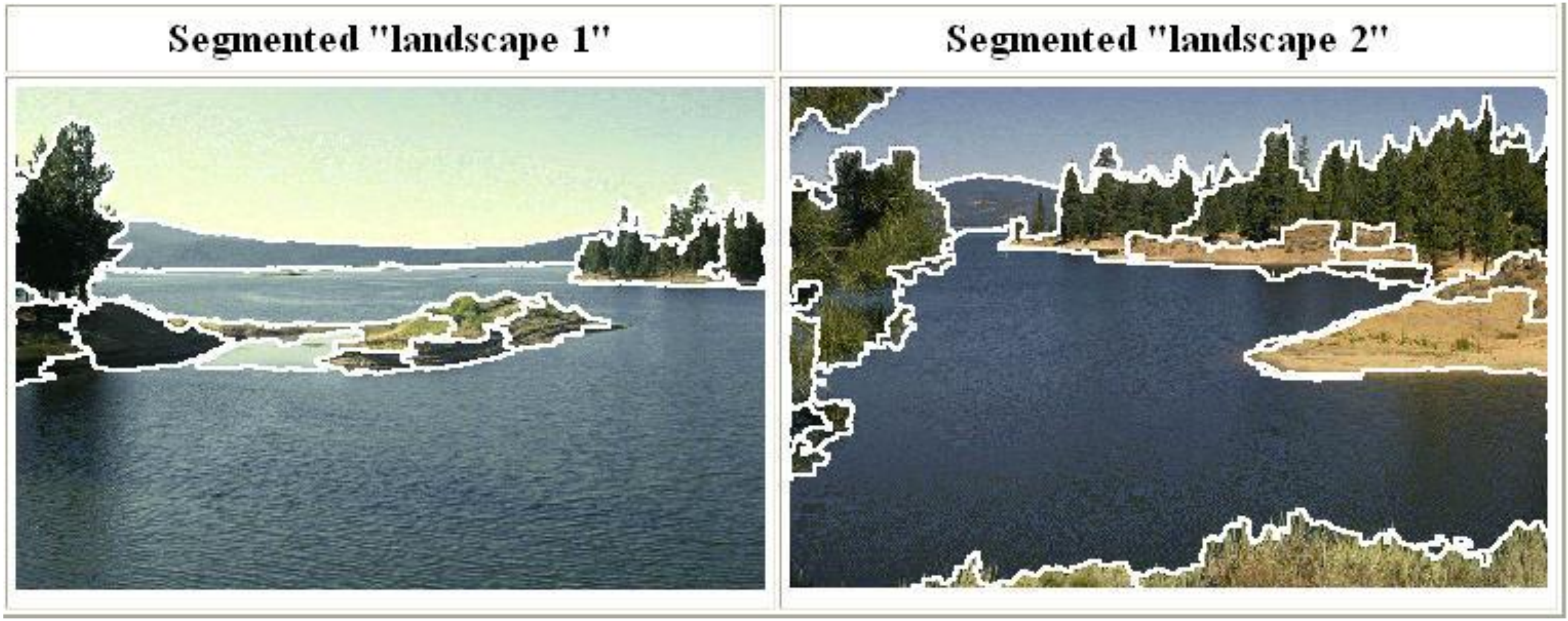


???

MEAN SHIFT

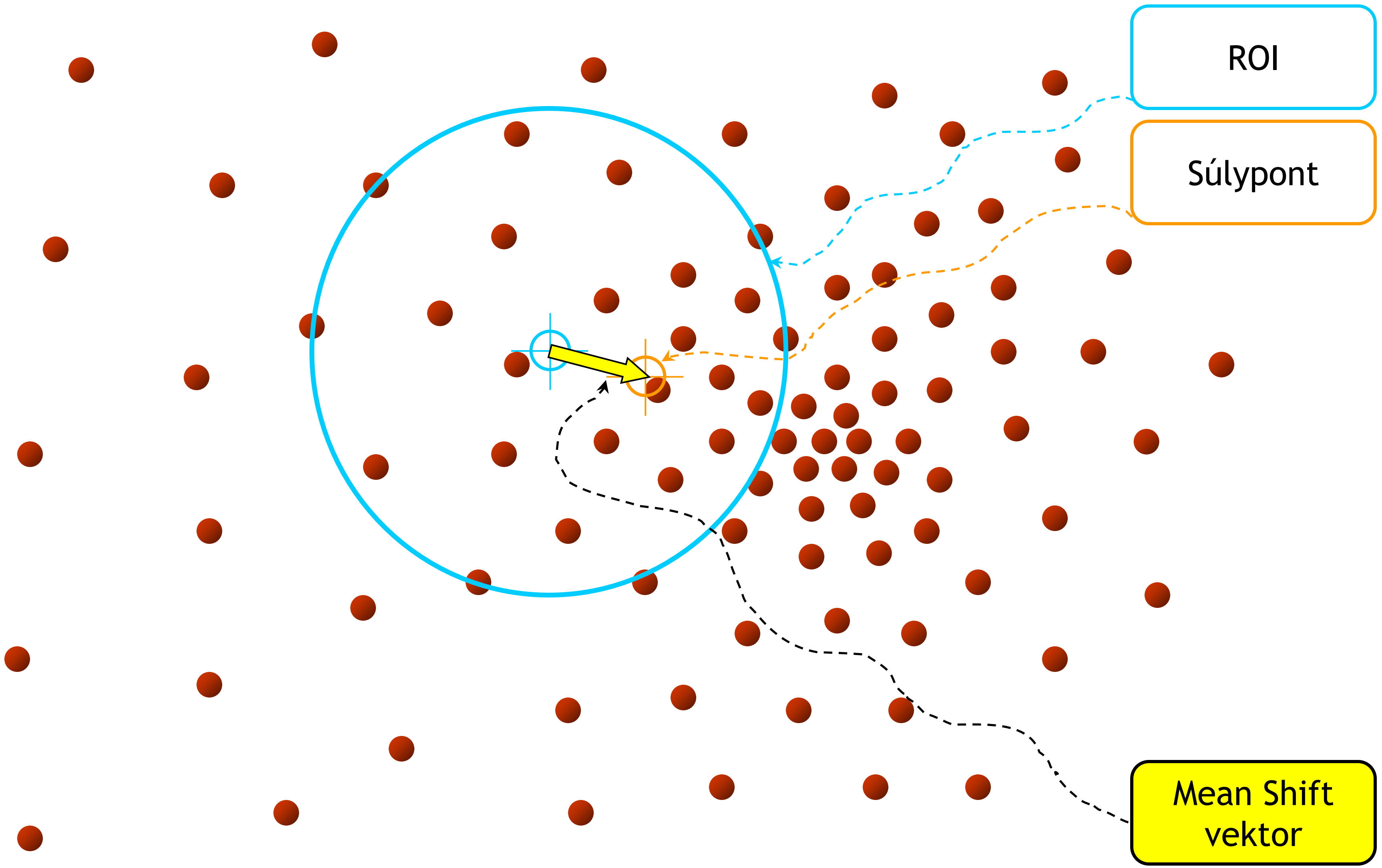
Mean shift szegmentálás

- **Klaszterező eljárás számos alkalmazással**
  - **éltartó simítás, szegmentálás, mozgáskövetés, …**



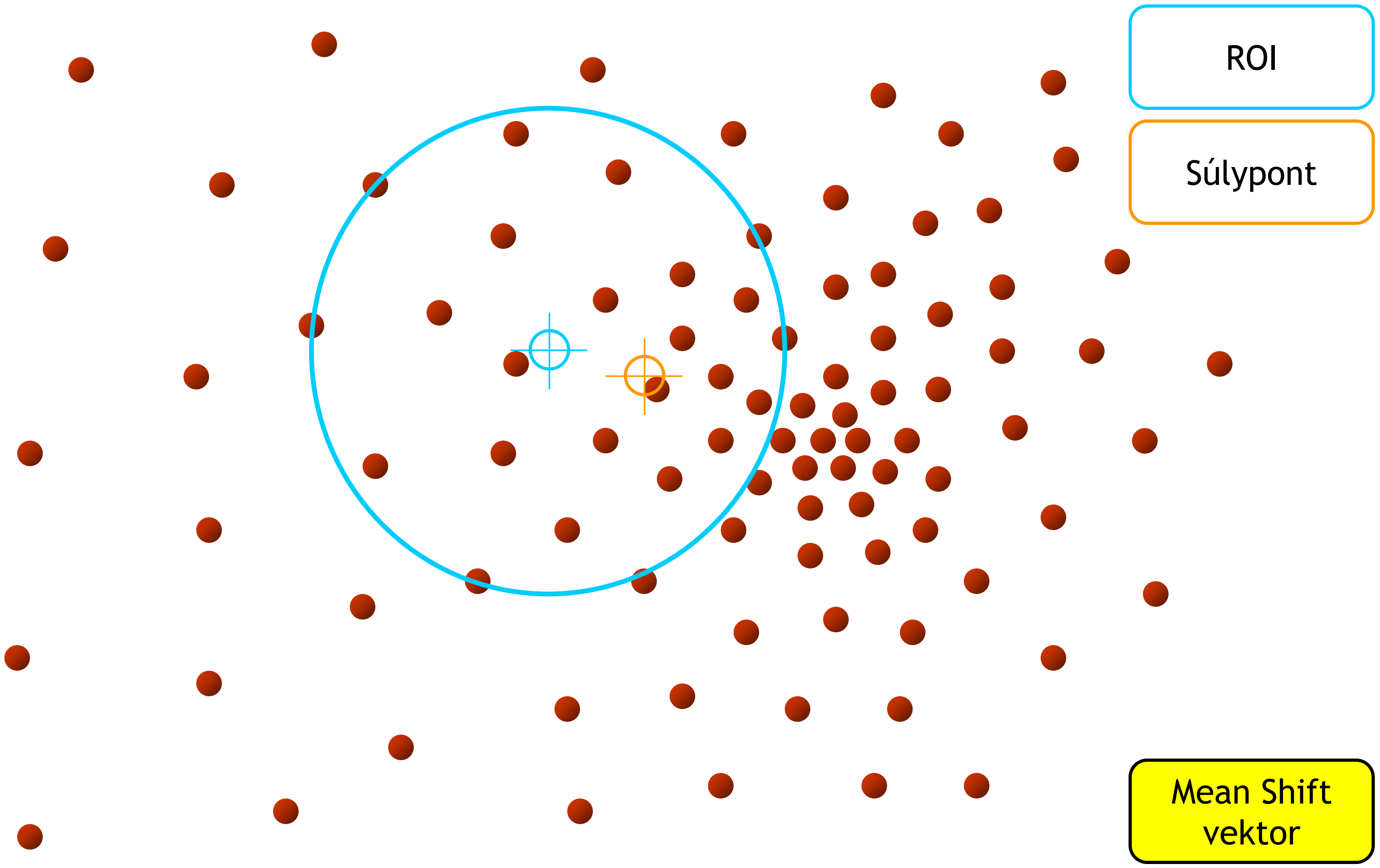
D. Comaniciu and P. Meer, **Mean Shift: A Robust Approach toward Feature Space Analysis**, *IEEE Transactions on Pattern Analyzis & Machine Intelligence*, 2002.

Mean shift működése



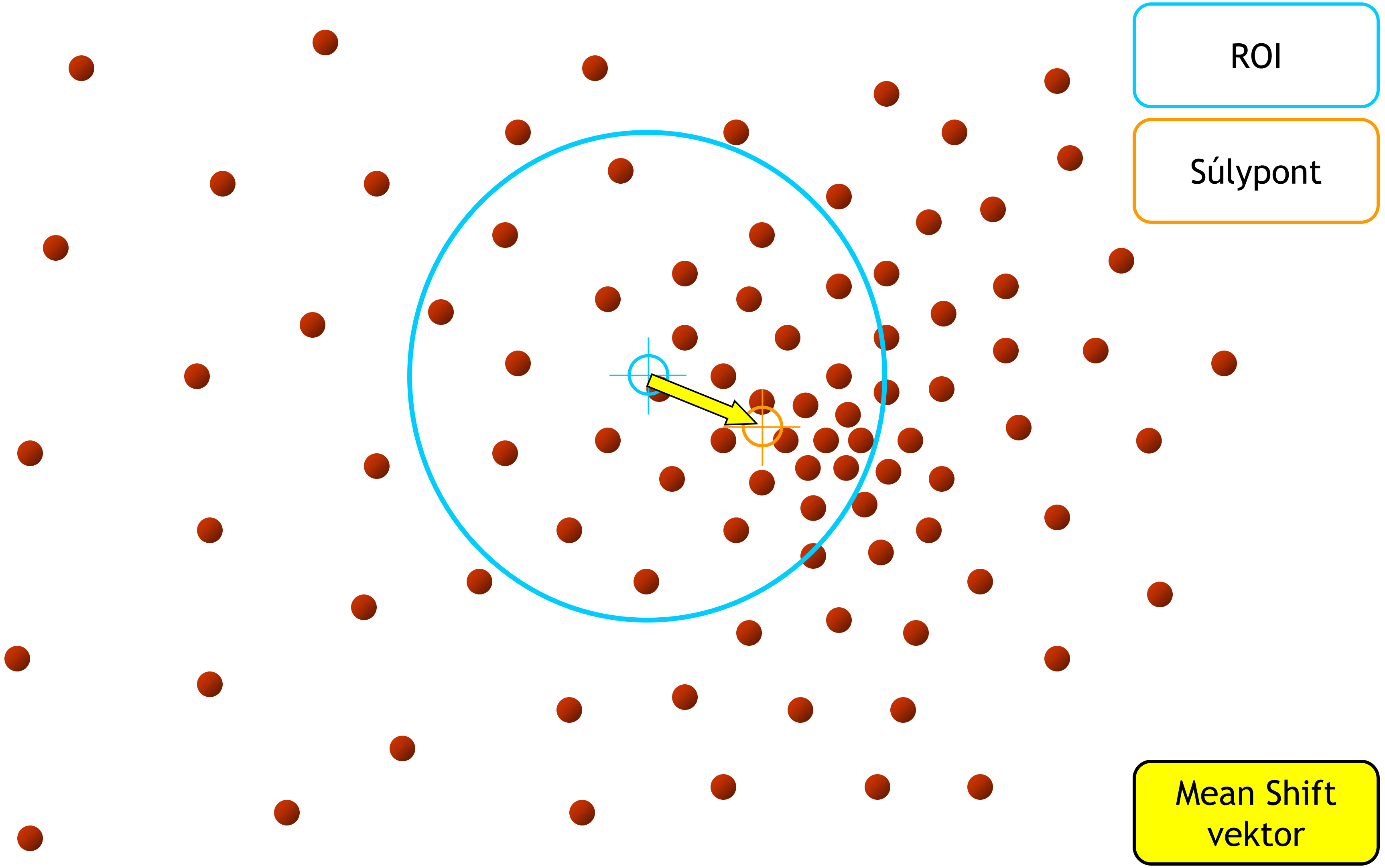
Slide by Y. Ukrainitz & B. Sarel

Mean shift működése



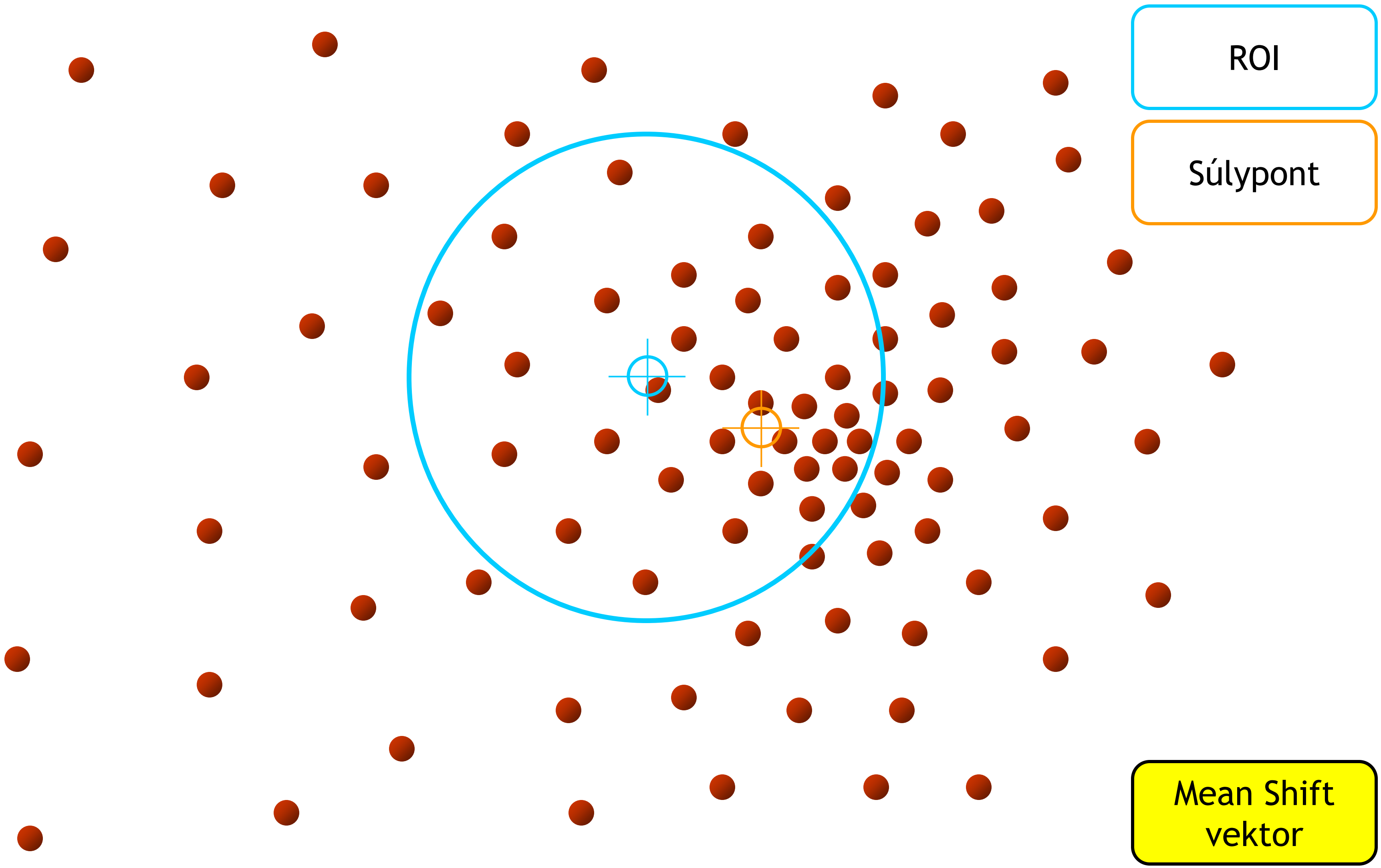
Slide by Y. Ukrainitz & B. Sarel

Mean shift működése



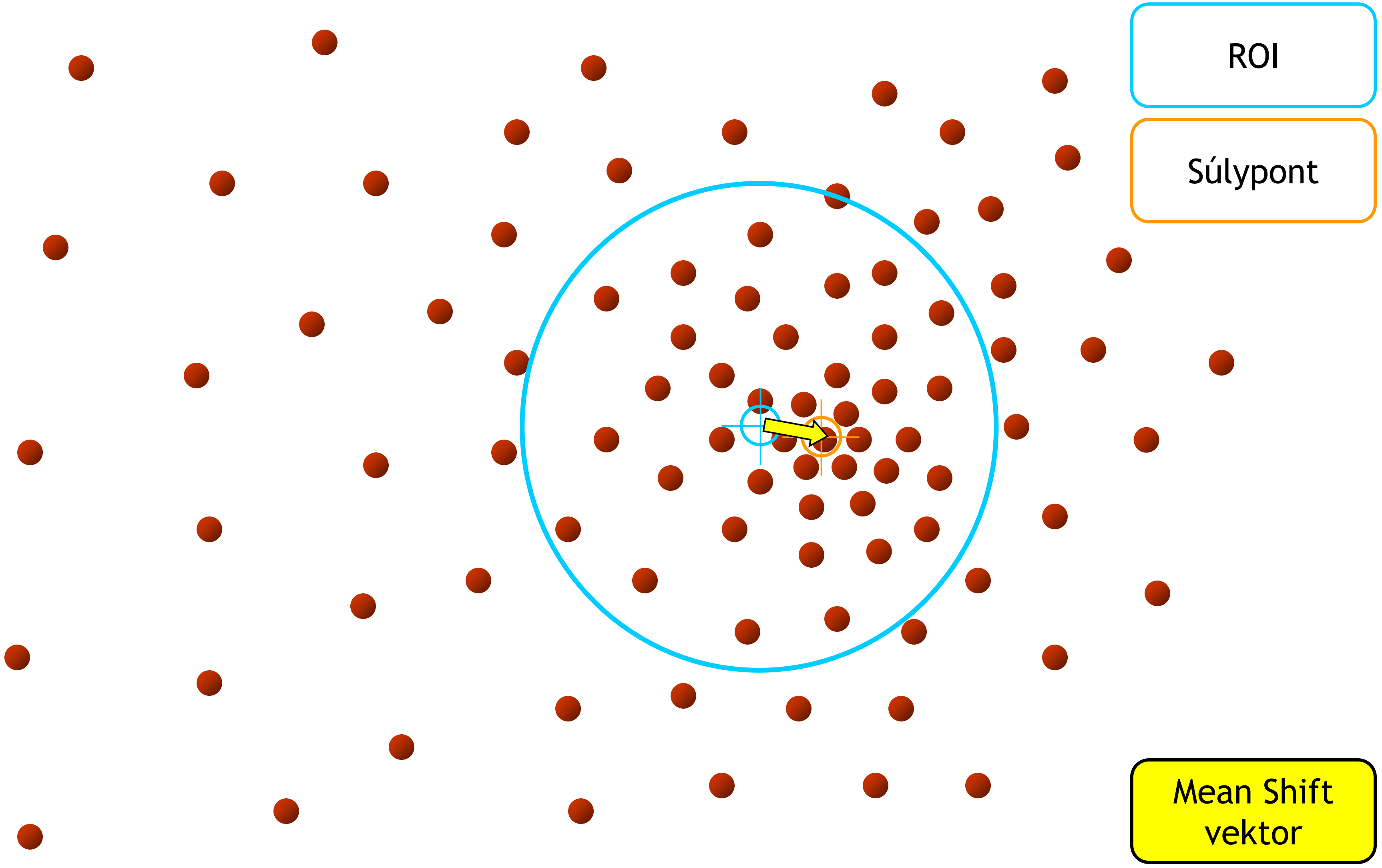
Slide by Y. Ukrainitz & B. Sarel

Mean shift működése



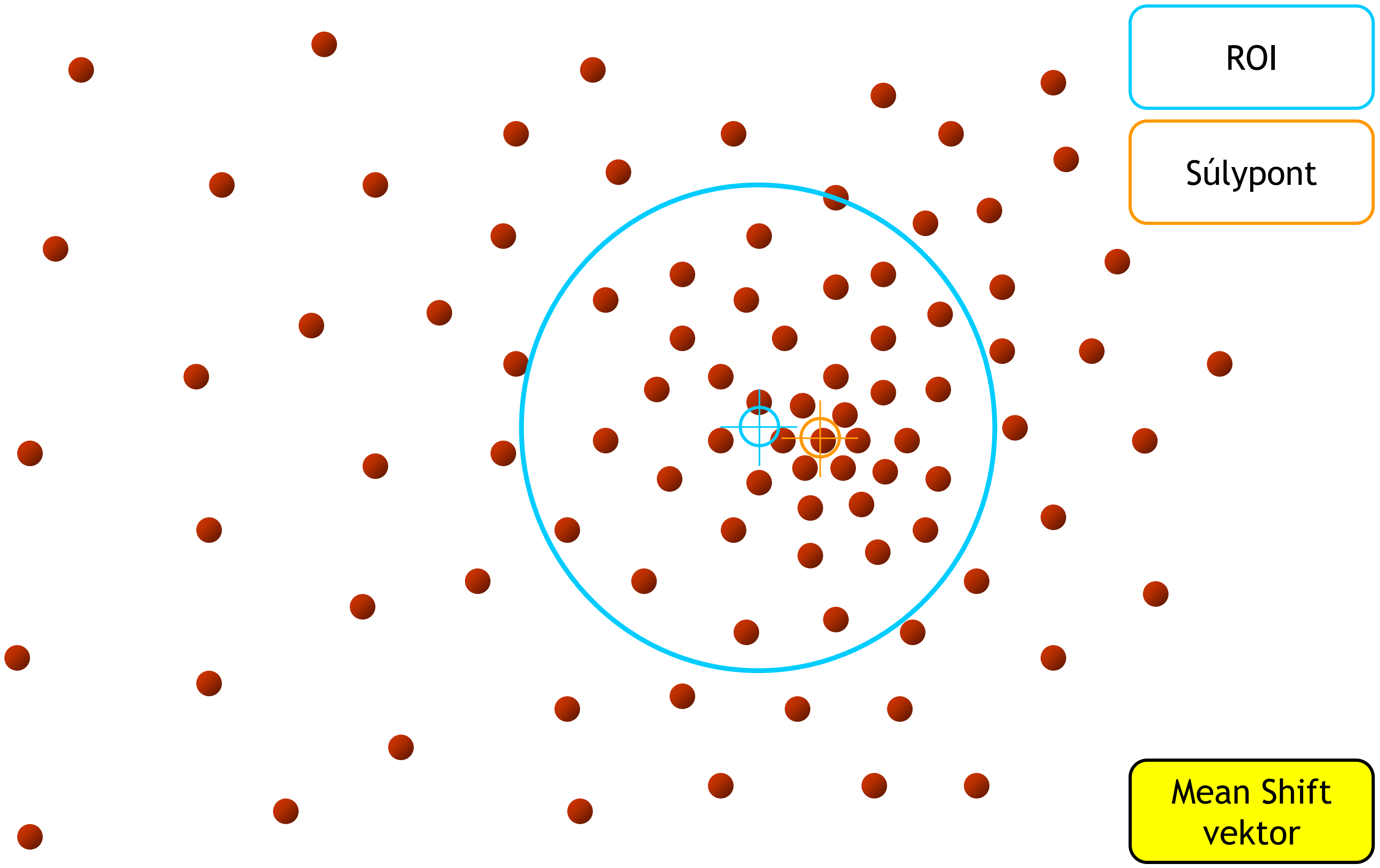
Slide by Y. Ukrainitz & B. Sarel

Mean shift működése



Slide by Y. Ukrainitz & B. Sarel

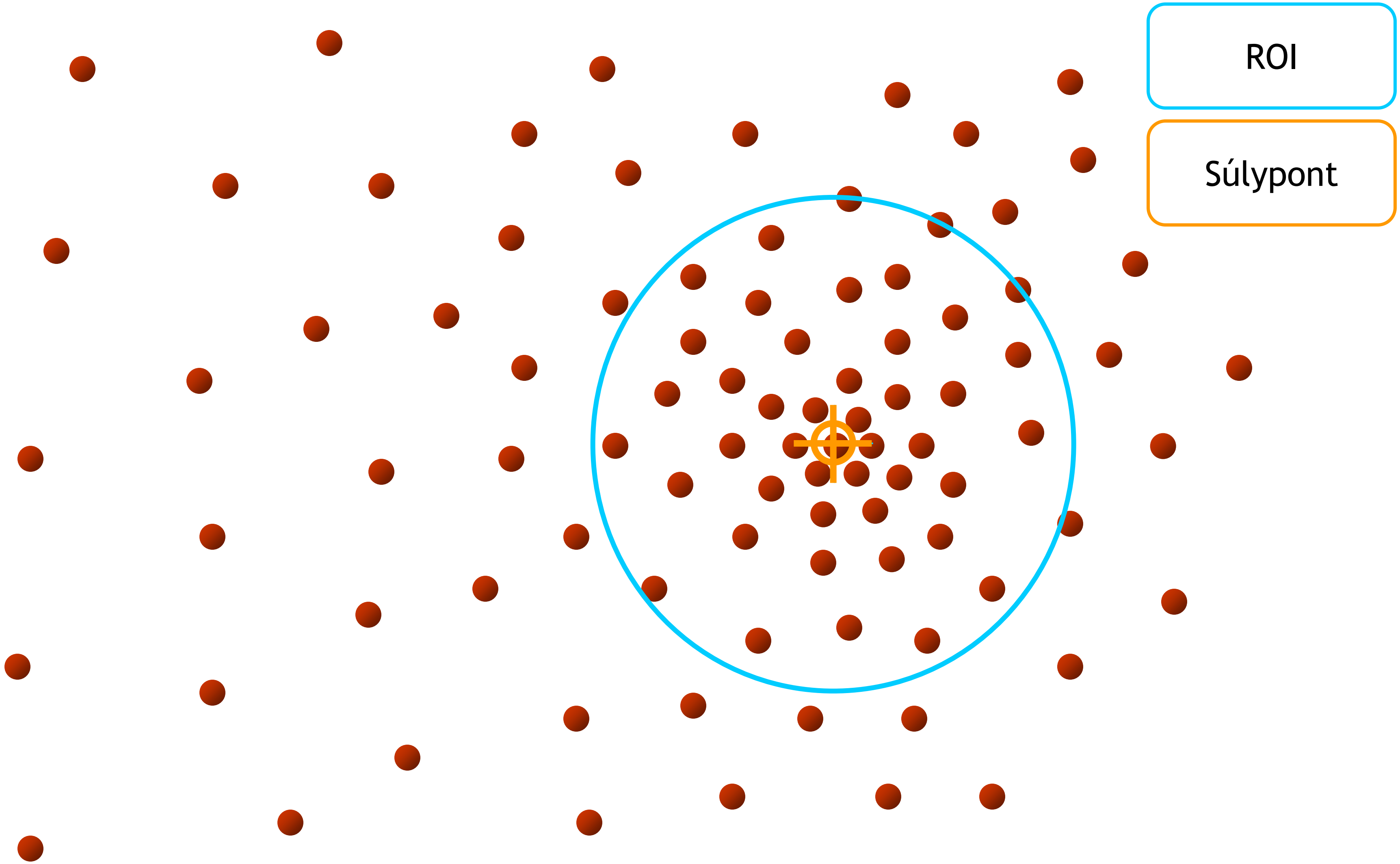
Mean shift működése



Slide by Y. Ukrainitz & B. Sarel

Kató Zoltán: Digitális Képfeldolgozás (Tehetséggondozó program)

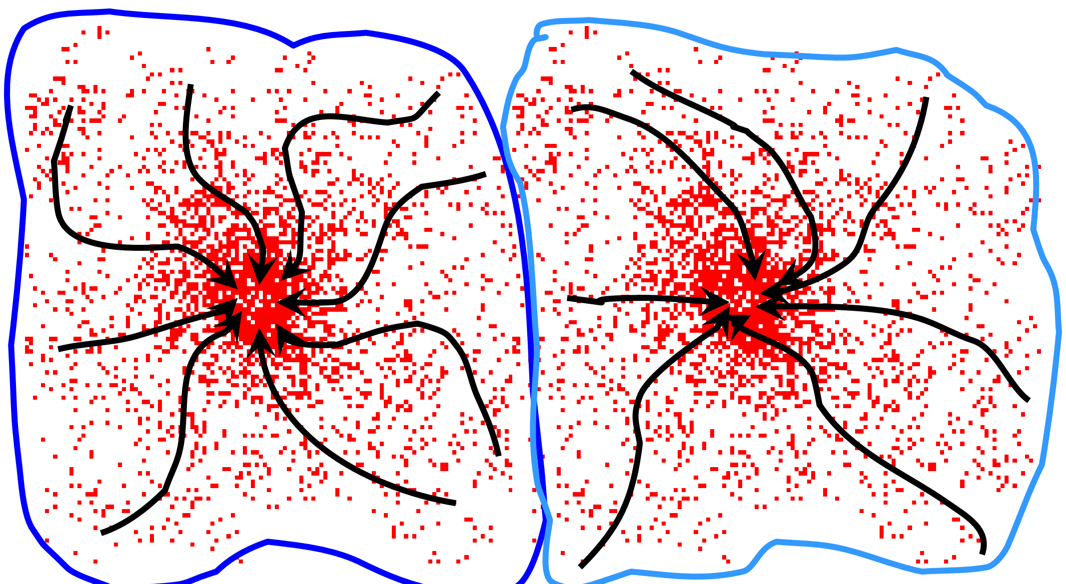
Mean shift működése

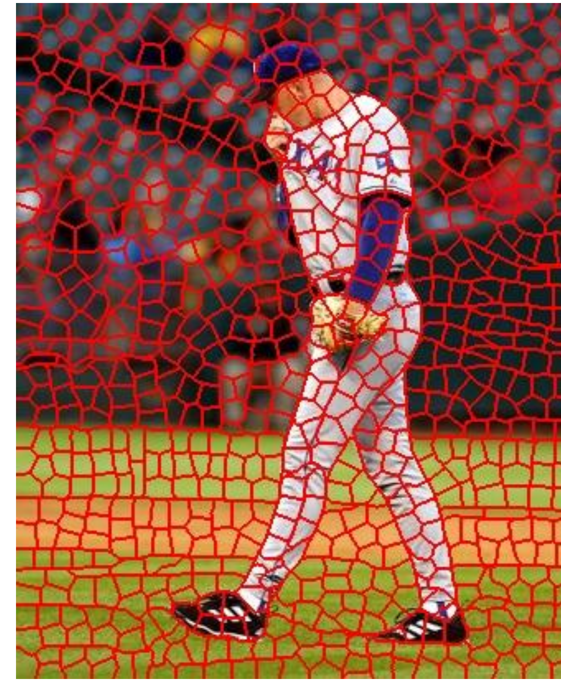


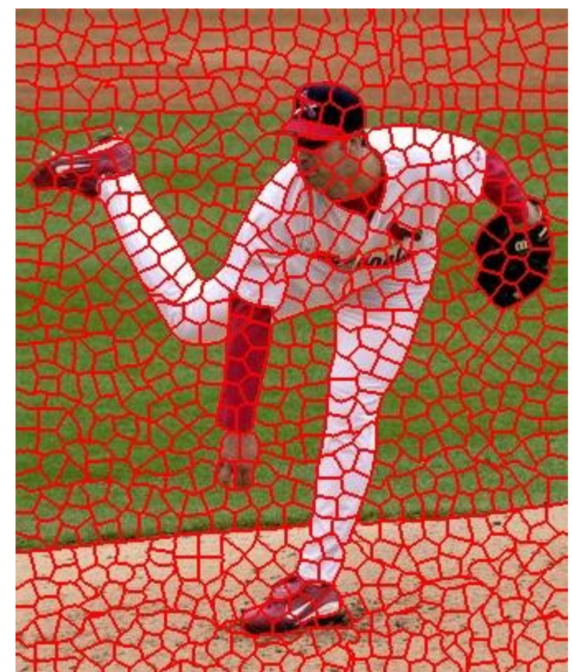
Slide by Y. Ukrainitz & B. Sarel

Mean shift klaszterezés

- **Egy csúcs vonzáskörzetébe tartozó adatpontokat soroljuk egy klaszterbe.**
- **Vonzáskörzet: az a régió, amelyből minden trajektória ugyanabba a csúcsba vezet.**
  - **Pixelérték *és* képi pozíció alapján klaszterezve szuperpixeleket kapunk**







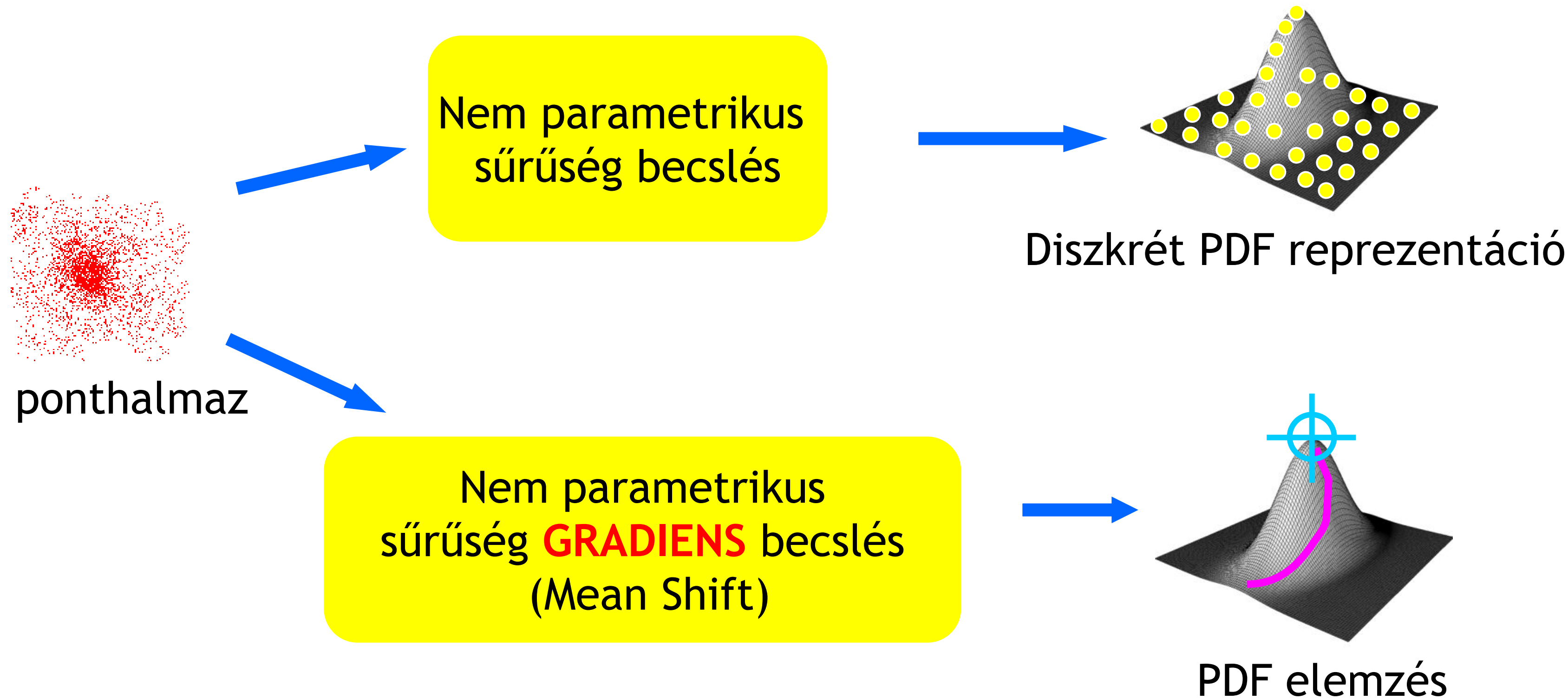
Mean shift algoritmus

1. Input: *n* darab *d*-dimenziós pontból álló halmaz: $\{x_i\}_{i=1..n}$

2. Statisztikai modell: a ponthalmazt egy ismeretlen sűrűségfüggvényű (PDF) valószínűségi változó mintáinak tekintjük. A sűrűségfüggvény paramétereire vonatkozóan semmit nem tételezünk fel (vagyis egy paraméter nélküli modellt alkalmazunk).

3. Cél: minden ponthoz keressük meg az ismeretlen sűrűségfüggvény hozzá legközelebb eső lokális maximumát.

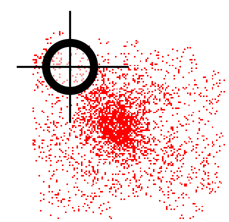
Sűrűség becslés paraméter nélkül



- **Nem parametrikus = nincs feltevés a PDF alakjára nézve (pl. parametrikus, ha feltesszük, hogy Gauss sűrűségfüggvénye van)**
- **A sűrűségfüggvény helyett csak a gradiensét fogjuk meghatározni**

Kernel Density Estimation, Parzen ablak

- Kernel: Az $\{x_i\}_{i=1..n}$ adatpontok függvénye:
  - Statisztikában Parzen-ablakként is ismert $P(\mathbf{x}) = \dfrac{1}{n}\sum_{i=1}^{n} K(\mathbf{x}-\mathbf{x}_i)$
- A kernel egy nem-negatív, valós, integrálható *K*(x) függvény, amely kielégíti a következő feltételeket:
  - Normalizált $$\int_{R^d} K(\mathbf{x})d\mathbf{x} = 1$$
  - Szimmetrikus $$\int_{R^d} \mathbf{x}K(\mathbf{x})d\mathbf{x} = 0$$
  - Exponenciálisan csökken a súlya $$\lim_{\|\mathbf{x}\|\to\infty} \|\mathbf{x}\|^d K(\mathbf{x}) = 0$$
  - $$\int_{R^d} \mathbf{x}\mathbf{x}^T K(\mathbf{x})d\mathbf{x} = c\mathbf{I}$$



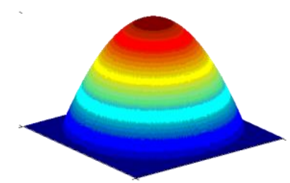
Adatpontok

Kernel példák

- **A gyakorlatban az alábbi egyszerűbb alakok használatosak:**
  - **Ugyanaz a függvény minden dimenzióban:** $K(\mathbf{x}) = c\prod_{i=1}^{d} k(x_i)$
  - **Vektor hosszának függvénye:** $K(\mathbf{x}) = ck\left(\|\mathbf{x}\|\right)$

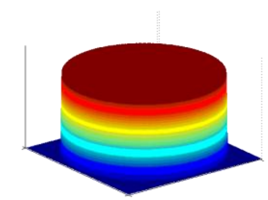
- **Epanechnikov Kernel**

$$K_E(\mathbf{x}) = \begin{cases} c\left(1-\|\mathbf{x}\|^2\right) & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



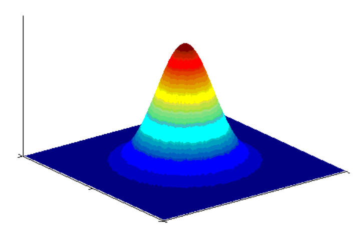
- **Uniform Kernel**

$$K_U(\mathbf{x}) = \begin{cases} c & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



- **Normal Kernel**

$$K_N(\mathbf{x}) = c \cdot \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$$



Kató Zoltán: Digitális Képfeldolgozás (Tehetséggondozó program)

Sűrűség gradiens becslése

- **A sűrűség helyett becsüljük annak gardiensét:**

$$\nabla P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla K(\mathbf{x}-\mathbf{x}_i)$$

- **Ebből a kernelből:** $K(\mathbf{x}-\mathbf{x}_i) = ck\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$

ablakméret

- **Kapjuk:**

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n} \nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \bullet \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

$g(\mathbf{x}) = -k'(\mathbf{x})$

Az elmozdulás (Mean Shift) kiszámítása

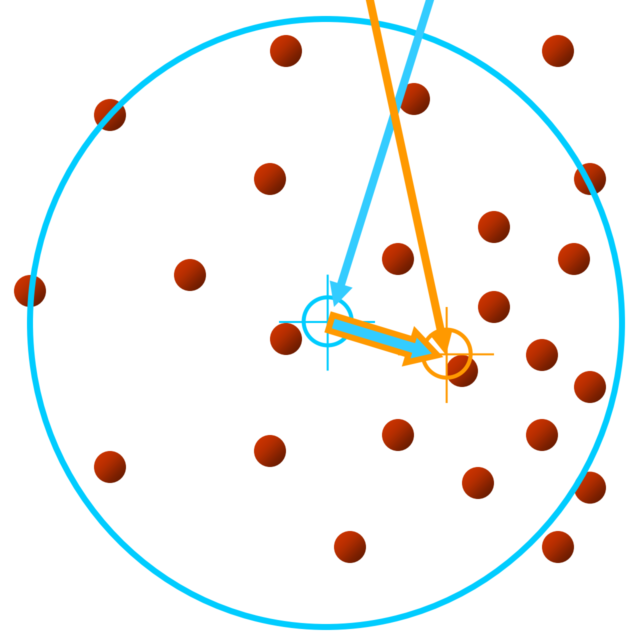
$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n}\nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right]\cdot\left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

Újabb Kernel density estimation !

Számítsuk ki a mean shift vektort

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)} - \mathbf{x}\right]$$

- **Toljuk el az ablakot m(x)-el**



$$g(\mathbf{x}) = -k'(\mathbf{x})$$

Mik lesznek az adatpontok?

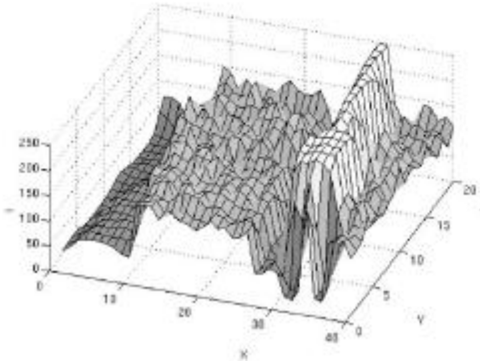
- Egy digitális képet 2 dimenziós rácson vett r-dimenziós vektorok (pixelértékek) reprezentálnak
  - r=1 szürkeárnyalatos kép esetén, 3 színes kép esetén, …
- A rácspontok alkotják a *képteret*
- A szürkeárnyalat, szín alkotja a *radiometriai tartományt*.
- Miután normalizáltuk h_s-el a képteret és h_r-el a radiometriai tartományt, a pozíció és radiometriai vektorokat konkatenálva megkapjuk a közös kép-radiometriai teret, amely $d=r+2$ dimenziós.

Jellemző tér

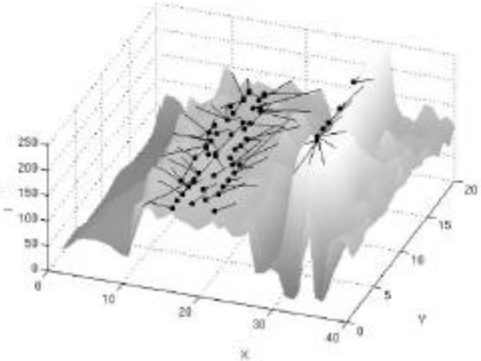
- **Jellemző tér = képtér koordináták + radiometriai érték**
  - **Kernel:**

$$K(\mathbf{x}) = C \cdot k_s\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|\right) \cdot k_r\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|\right)$$

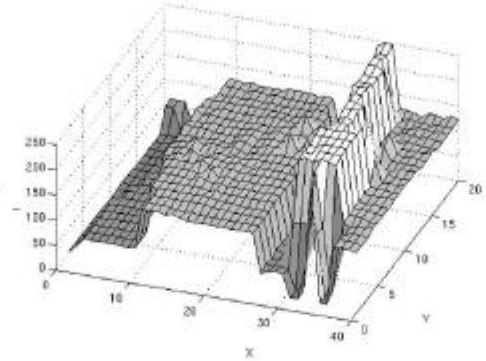
- **Vagyis a képet úgy tekintjük, mint adatpontok a közös kép-radiometriai térben**



Képi adat

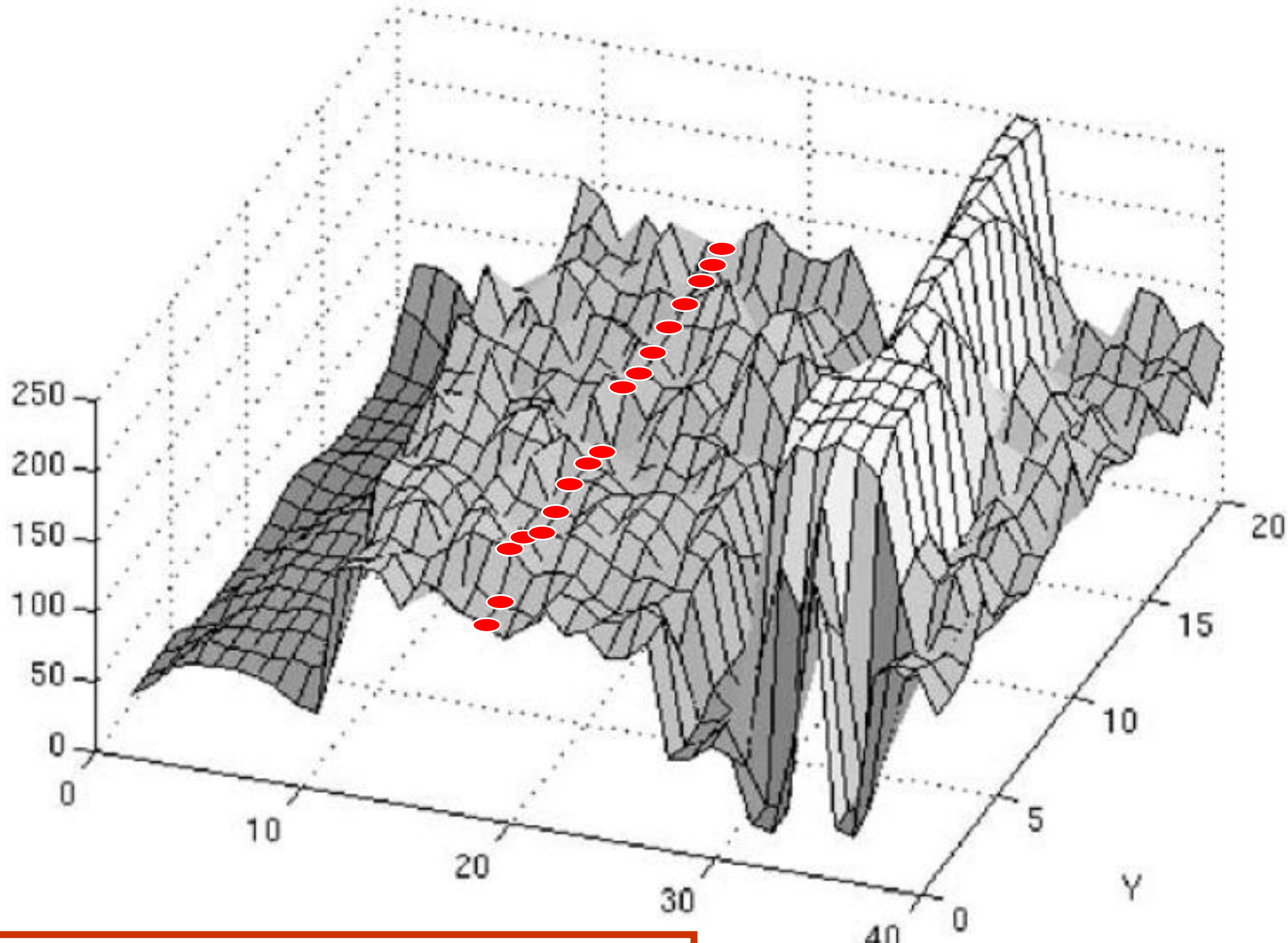


Mean Shift vektorok



Simítás eredménye

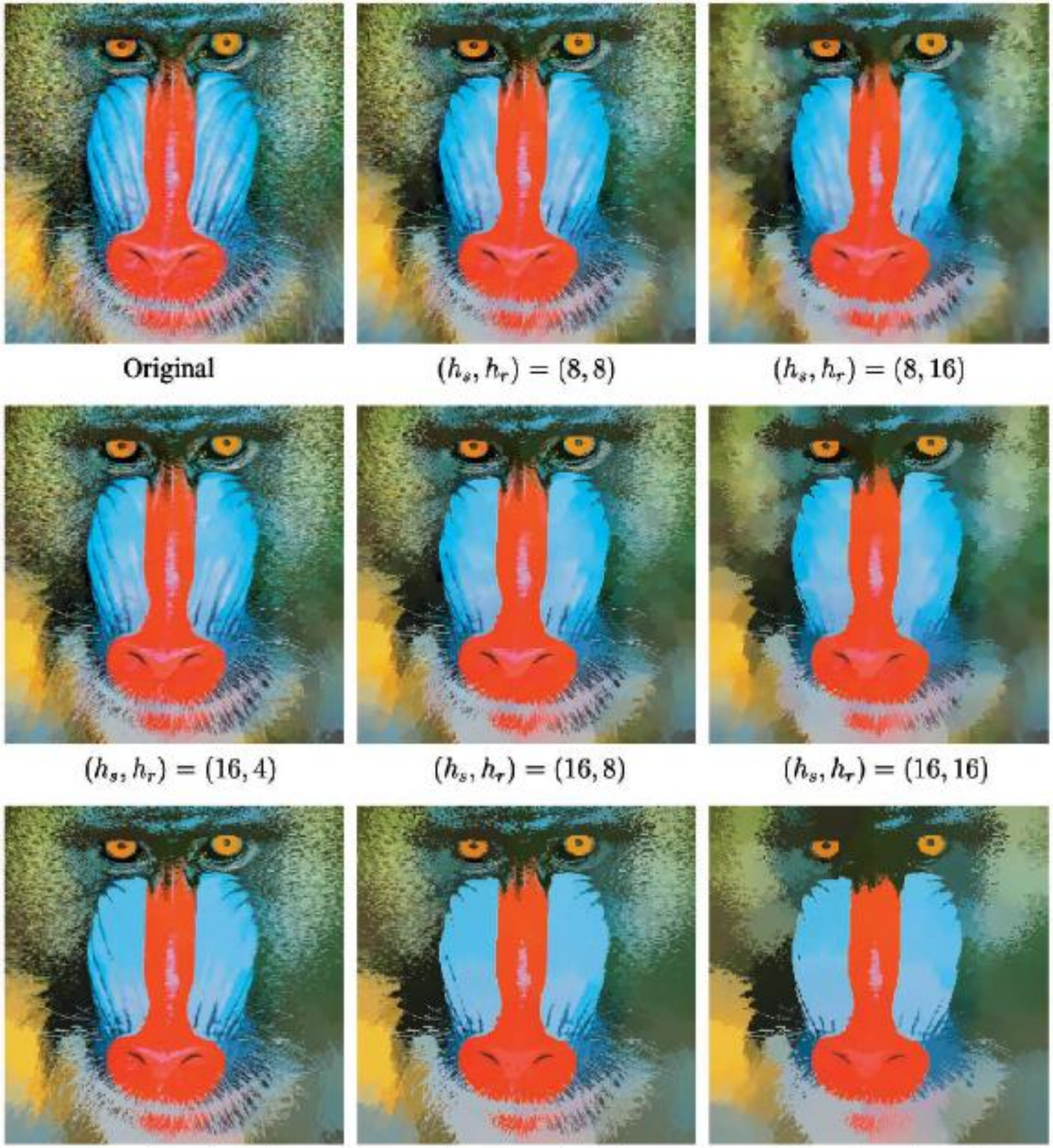
Éltartó simítás



A módusok a sima régiókban keletkeznek

Éltartó simítás

- Az ablakméret hatása a kép- és a radiometriai térben



Original — $(h_s, h_r) = (8, 8)$ — $(h_s, h_r) = (8, 16)$

$(h_s, h_r) = (16, 4)$ — $(h_s, h_r) = (16, 8)$ — $(h_s, h_r) = (16, 16)$

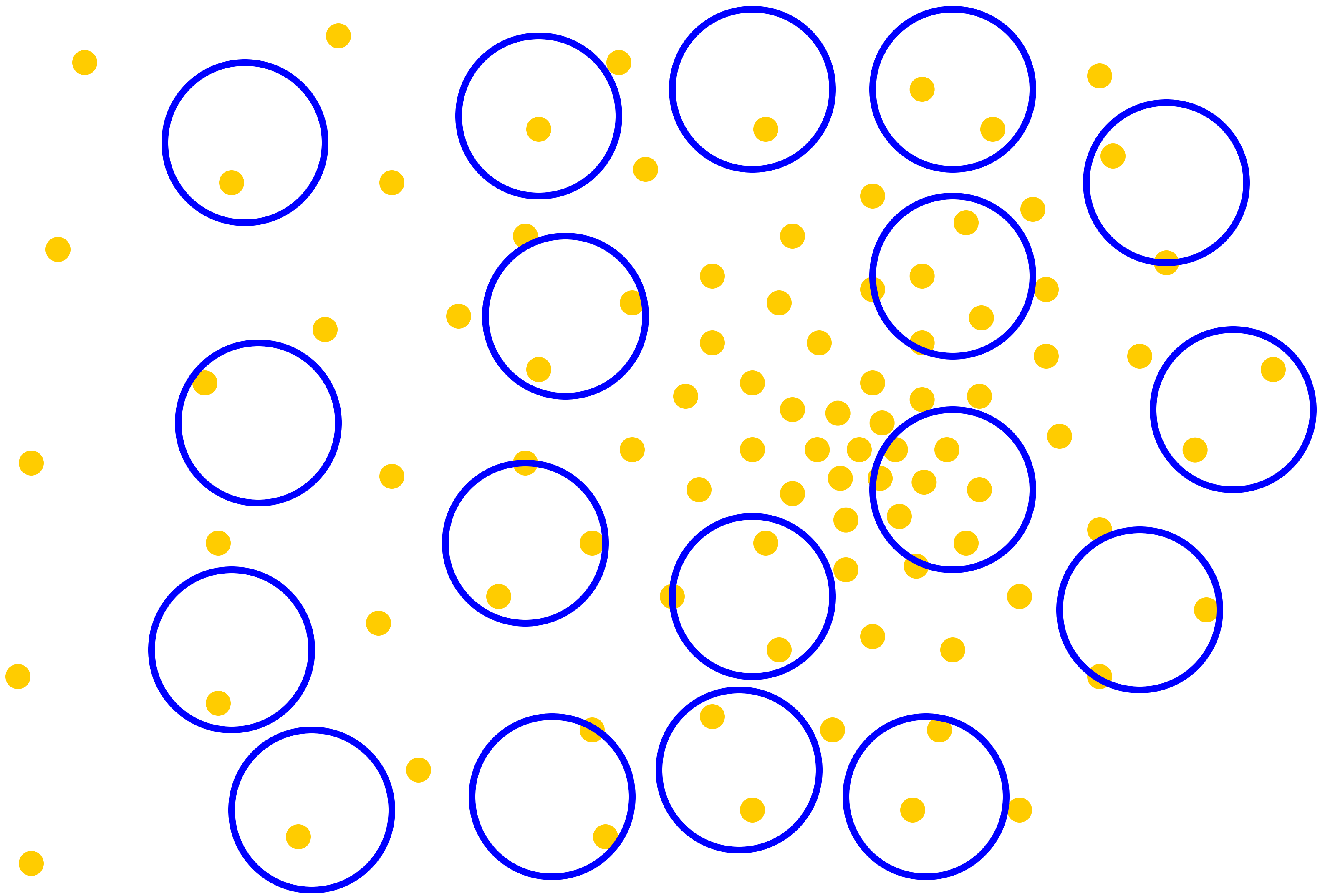
$(h_s, h_r) = (32, 4)$ — $(h_s, h_r) = (32, 8)$ — $(h_s, h_r) = (32, 16)$

Mean Shift szegmentálás

1. Minden pixelre határozzuk meg a legközelebbi módust, és tároljuk (Mean Shift által adott érték)
2. Határozzuk meg azokat a pixeleket, amelyekhez tartozó módusok közel azonosak (<0.5 távolság)
3. Cjmkézzük meg az így kapott pixeleket ➔ régió
4. Esetleg töröljük a túl kicsi régiókat (rendeljük őket a legközelebbi szomszédhoz)

- Mindezt párhuzamosan is végrehajthatjuk

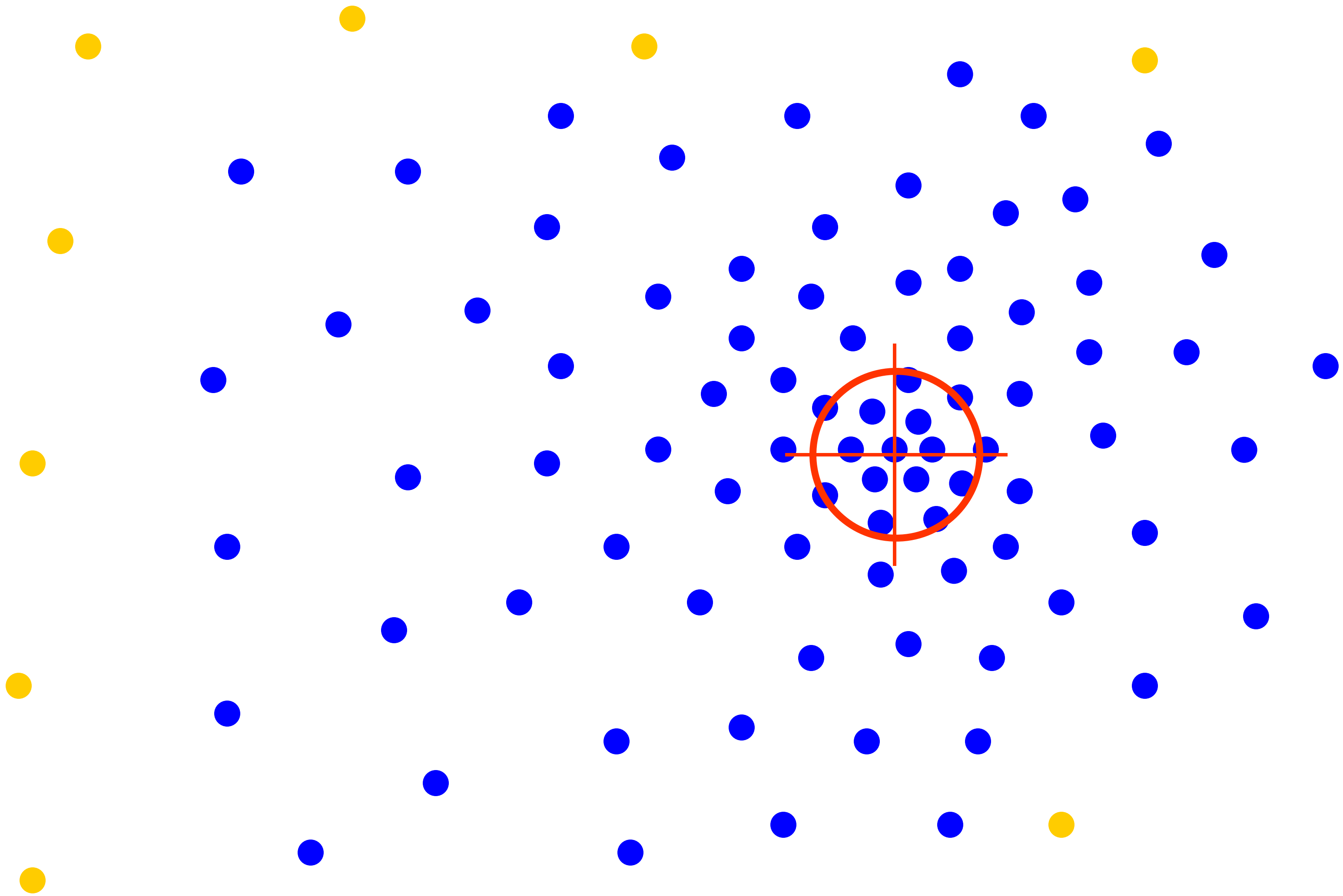
Módusok keresése



Fedjük le a teret nem átfedő ablakokkal

Futtassuk párhuzamosan az algoritmust

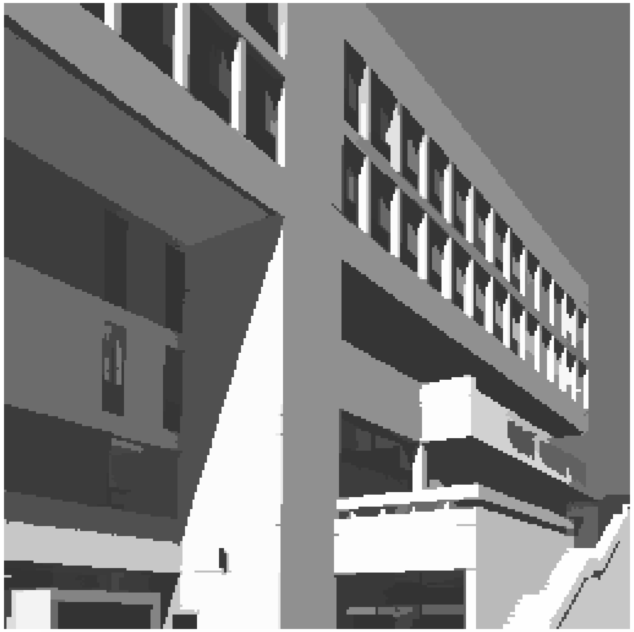
Módusok keresése



A kék pontokon keresztülhaladt az ablak a módus felé

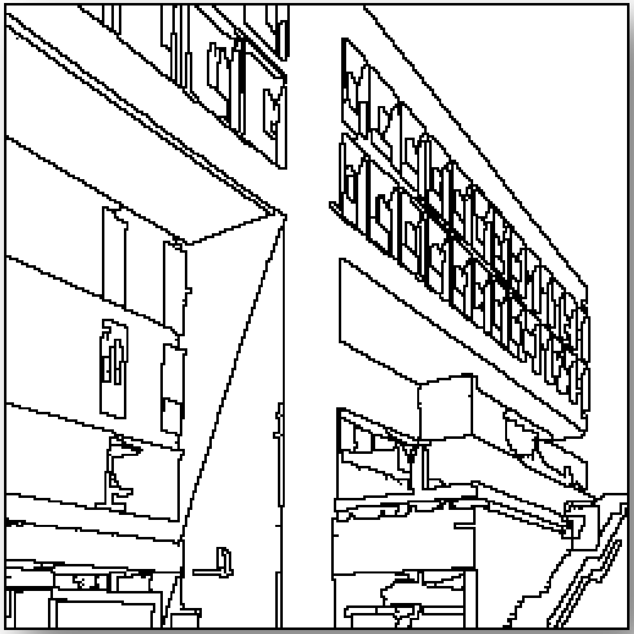
Szegmentálási eredmény

Szegmentált kép

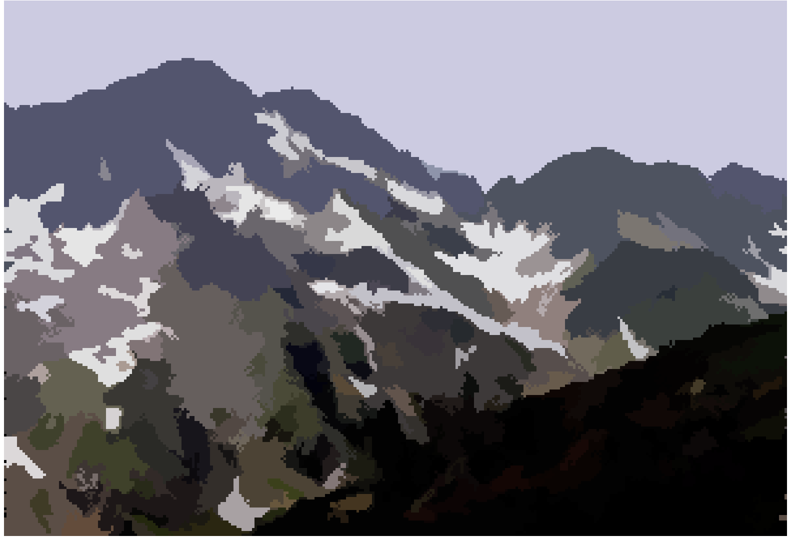


Input kép

Kontúrok



Mean shift szegmentálási eredmények



Kató Zoltán: Digitális Képfeldolgozás (Tehetséggondozó program)

Mean shift előnyök és hátrányok

- Előnyök
  - Tetszőleges alakú klasztereket detektál (K-means csak gömbszerűeket!)
  - Egyetlen paraméter (ablakméret)
  - Tetszőleges számú csúcsot képes megtalálni
- Hátrányok
  - Az eredmény függ az ablakmérettől
  - Nagy számításigényű
  - Rosszul skálázható (a feature-tér dimenziójára érzékeny)

Felhasznált anyagok

- Palágyi Kálmán: Digitális Képfeldolgozás
  /pub/Digitalis_kepfeldolgozas
- Svetlana Lazebnik: COMP 776: Computer Vision
  - http://www.cs.unc.edu/~lazebnik/research/spring08/
- Yaron Ukrainitz & Bernard Sarel: Advanced Topics in Computer Vision, Weizmann Institute
  - http://www.wisdom.weizmann.ac.il/~vision/courses/2004_2/files/mean_shift/mean_shift.ppt
- További források az egyes diákon megjelölve