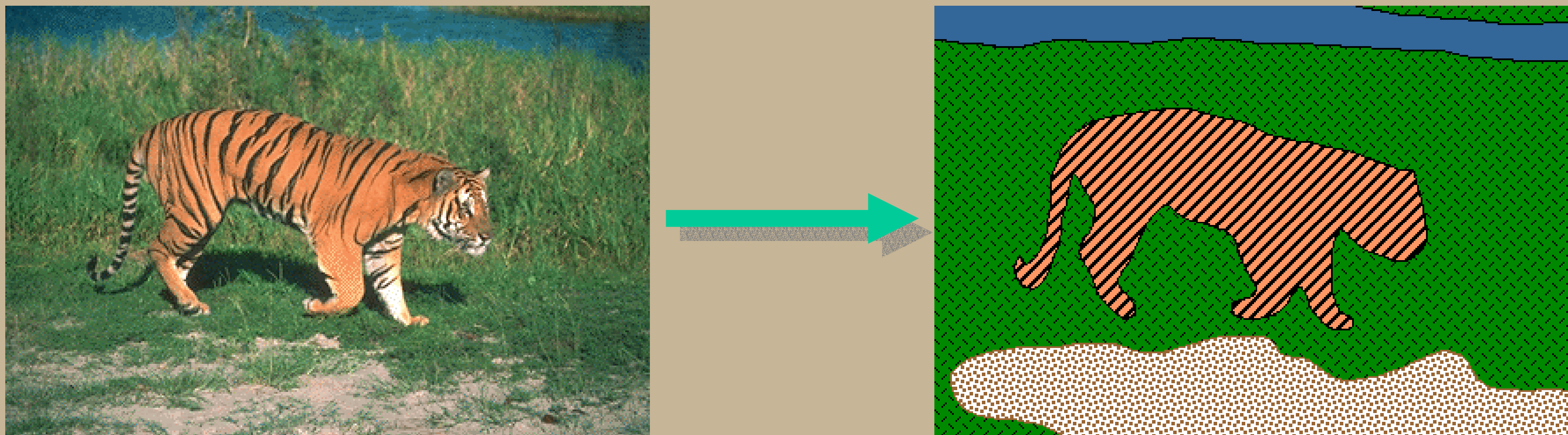
7. Régió alapú szegmentálás

Kató Zoltán

http://www.cab.u-szeged.hu/~kato/segmentation/

Szegmentálási kritériumok



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

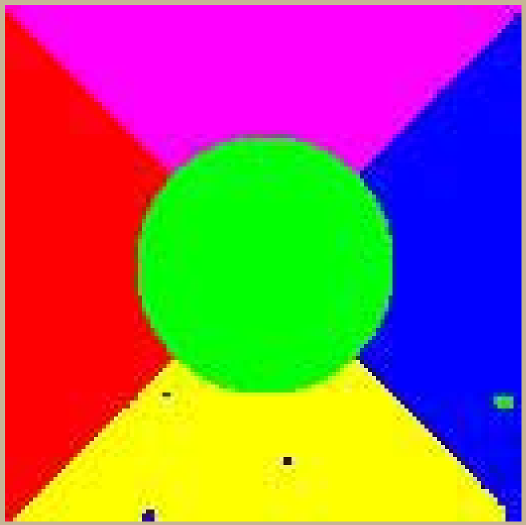
1. $\cup S_i = S$ — A régiók lefedik a teljes képet
2. $S_i \cap S_j = \emptyset,\ i \neq j$ — A régiók nem átfedőek
3. $\forall\ S_i\ ,\ P(S_i) = \text{true}$ — Minden régió kielégíti a homogenitási kritériumot
4. $P(S_i \cup S_j\ ) = \text{false}$, $i \neq j$, S_i adjacent S_j — Szomszédos régiók uniója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

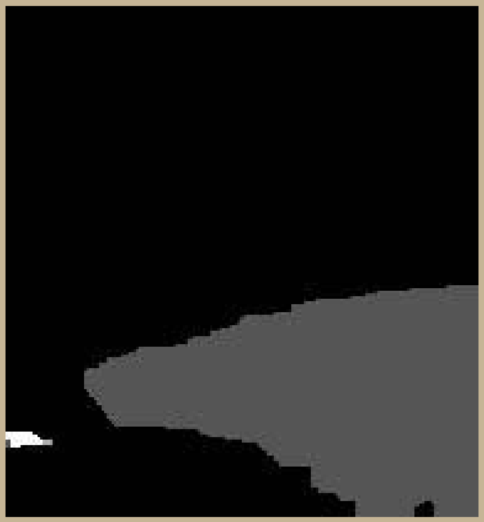
Homogenitási kritérium

- A szegmentáláshoz tehát definiálnunk kell egy homogenitási kritériumot
  - Milyen tulajdonságok figyelembevételével és
    - Intenzitás
    - Szín
    - Textúra
    - Mozgás
  - Hogyan definiáljuk a hasonlósági kritériumot úgy, hogy
  - A kialakult régiók megfeleljenek a képen látható objektumoknak vagy azok értelmes részeinek.

Címkézés

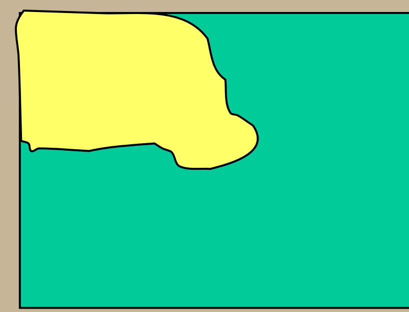
- A szegmentált képet címkézéssel állítjuk elő:
  - Egy adott régióhoz tartozó pixeleket azonos címkével (~szürkeárnyalat) jelöljük meg.

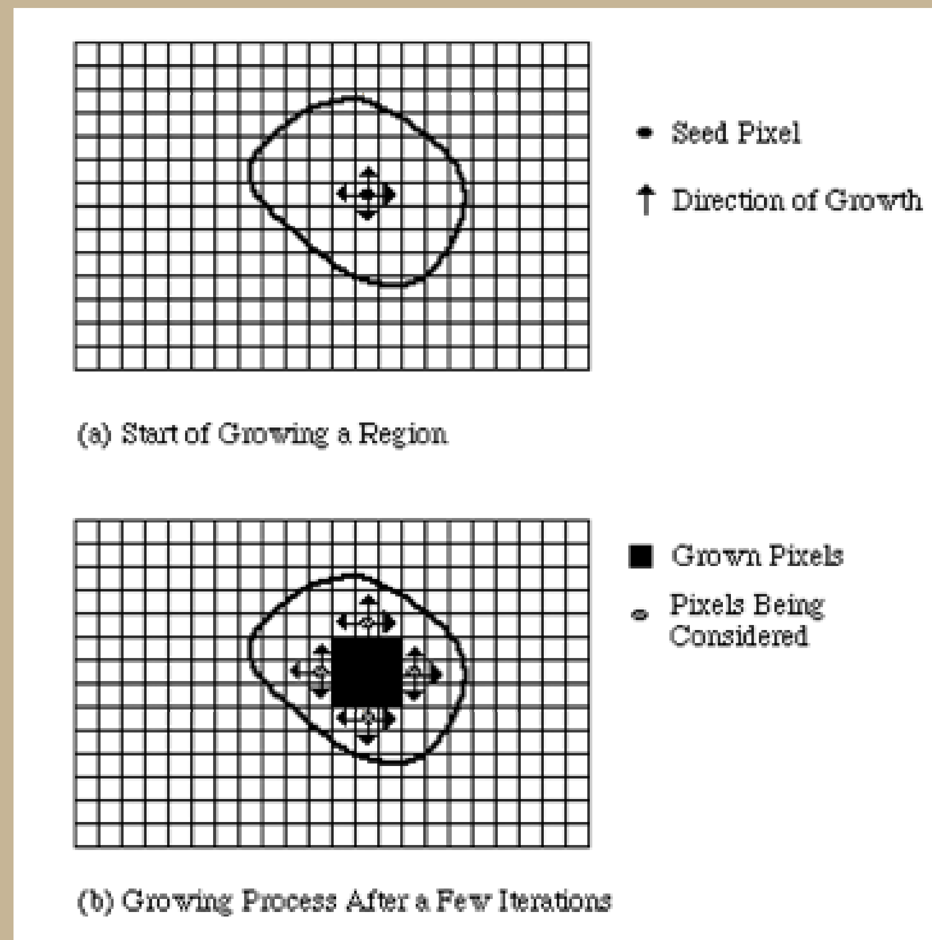





Régiónövelés (Region Growing)

- 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

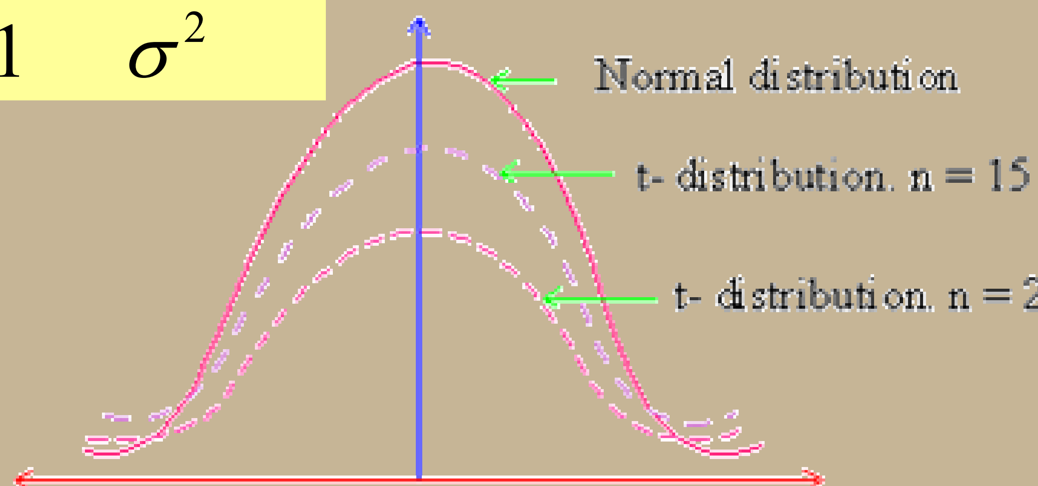
(b) Growing Process After a Few Iterations

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.

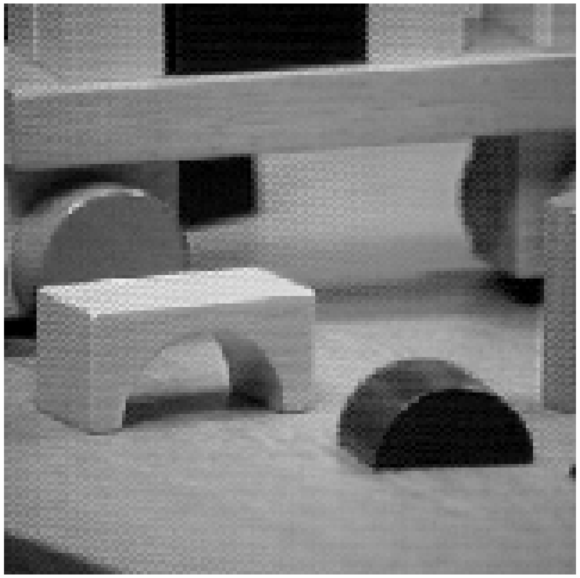
$$\mu = \frac{1}{N} \sum_{(x,y)\in R} I(x,y) \qquad \sigma^2 = \frac{1}{N-1} \sum_{(x,y)\in R} \left(I(x,y) - \mu\right)^2$$

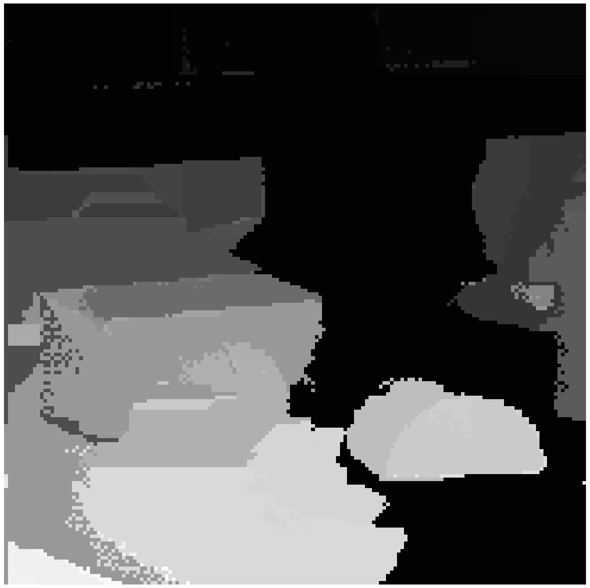
  - A τ statisztika: $$\tau = \sqrt{\frac{N}{N+1} \frac{(P-\mu)^2}{\sigma^2}}$$
  - (n-1) szabadsági fokú Student's eloszlást követ



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.





Split & Merge

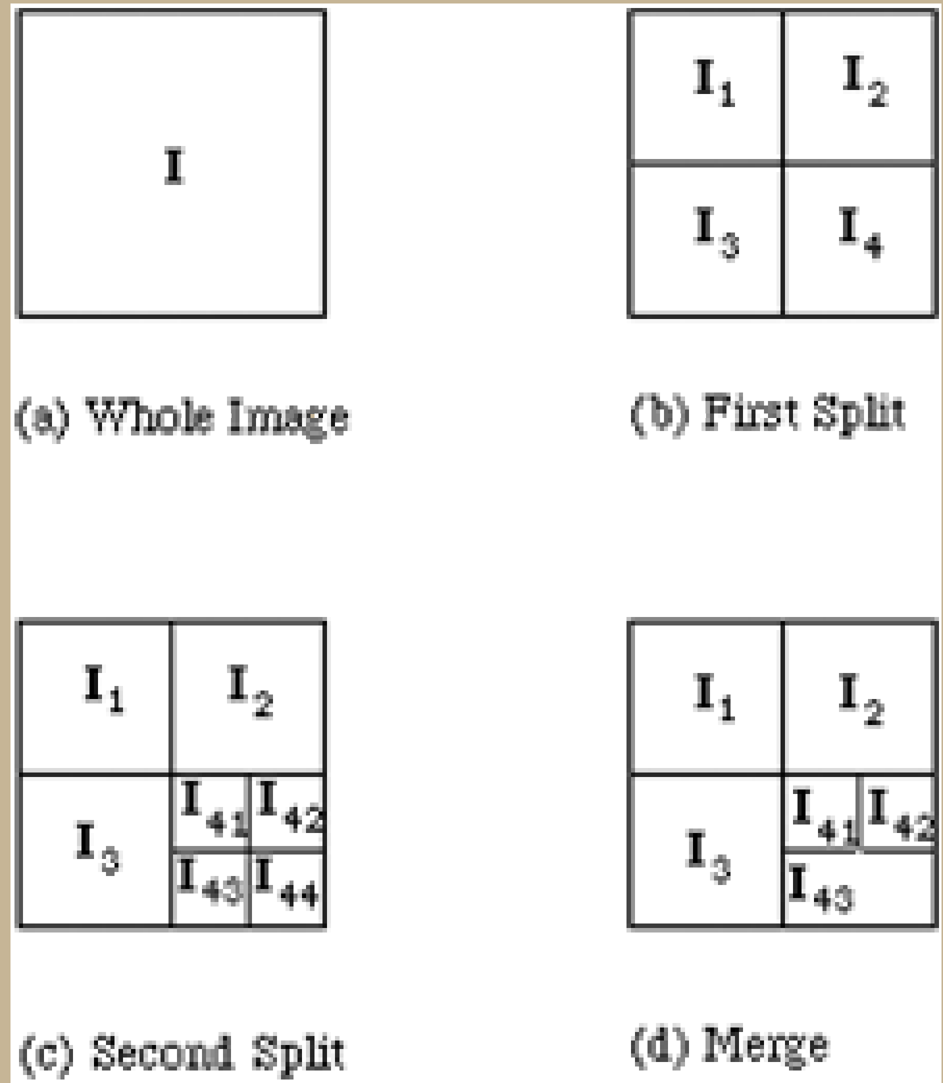
1. ***P*** – homogenitási kritérium
2. ***R_i*** régiót osszuk fel 4 részre, ha

$$P(R_i) = False$$

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

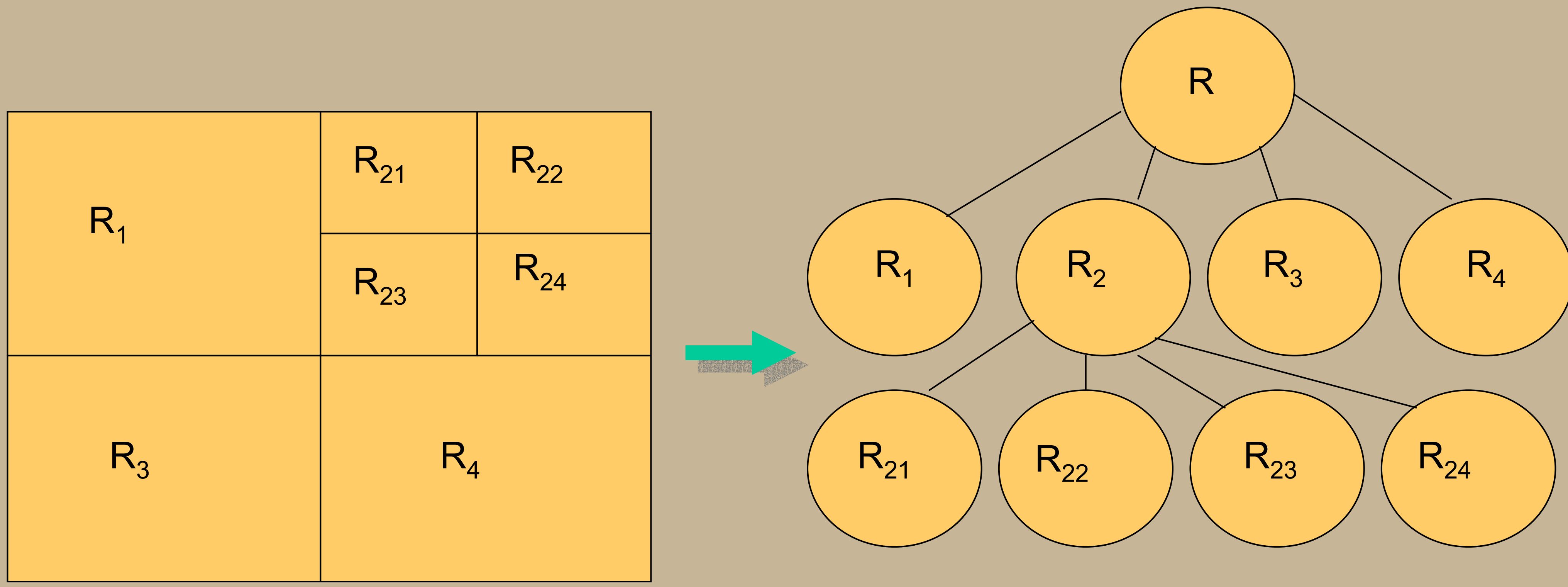
$$P(R_i \cup R_j) = True$$

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

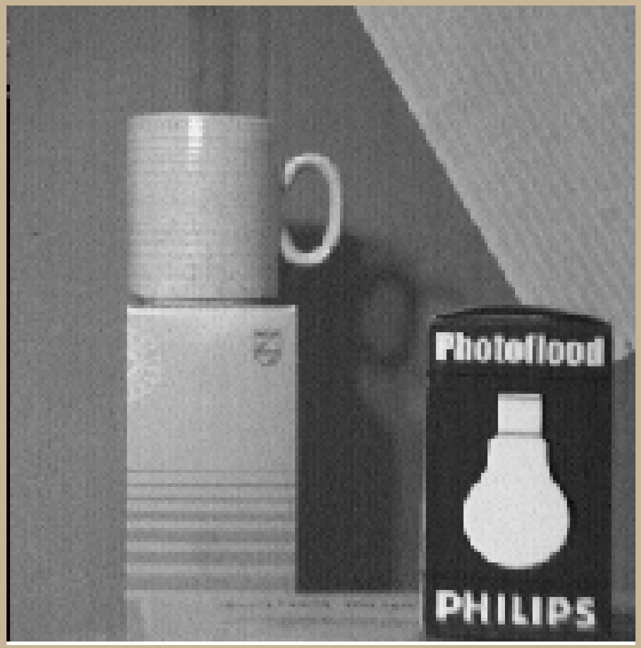


(a) Whole Image (b) First Split (c) Second Split (d) Merge

Quad-tree



Split & Merge



original

split

merge

Klaszterezés

- Adott ***K*** klaszter: $C_1, \ldots, C_K$.
- A kalszterek középértékét jelölje $\mu_1, \ldots, \mu_K$.
- LSE (Least Square Error):

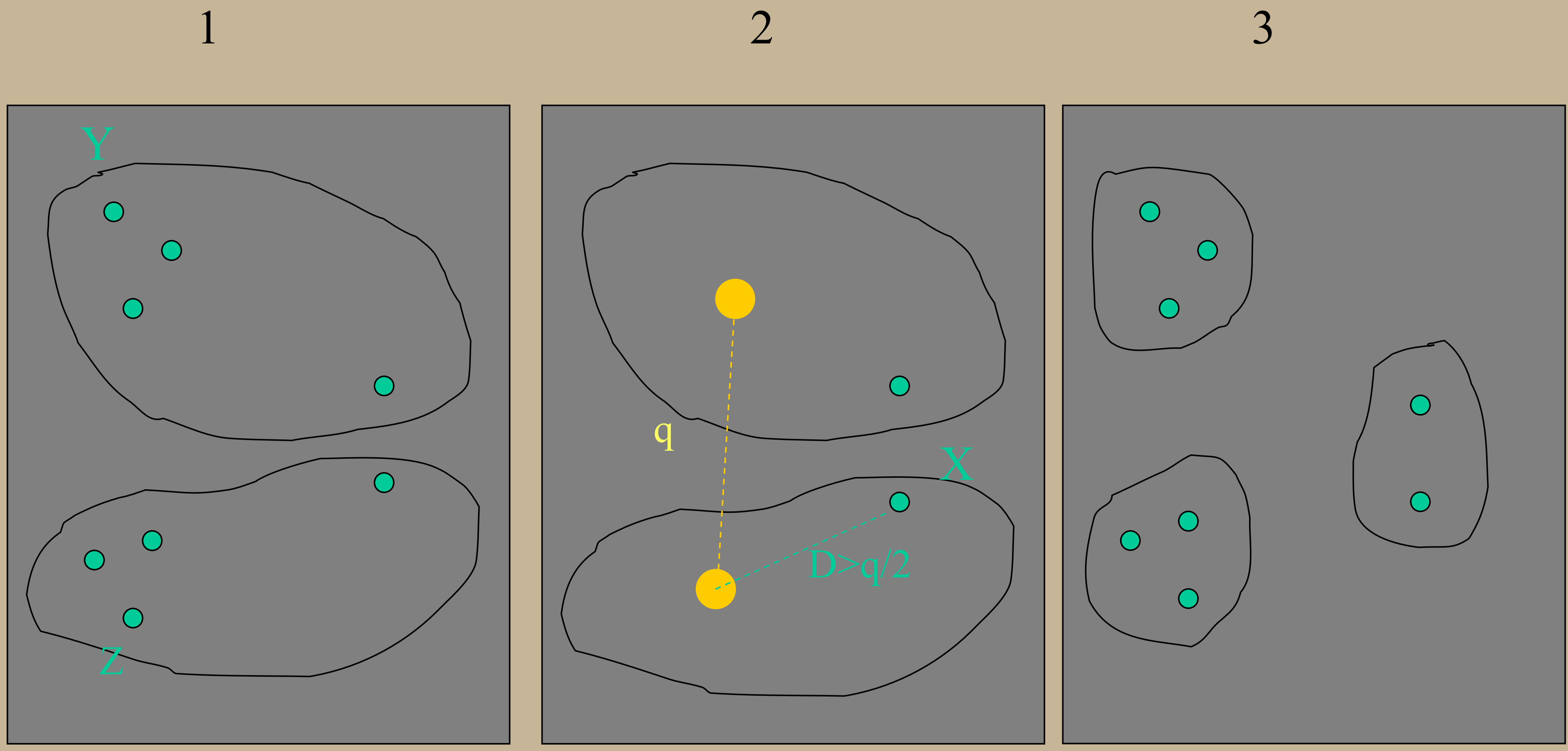
$$D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left\| x_i - \mu_k \right\|^2$$

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

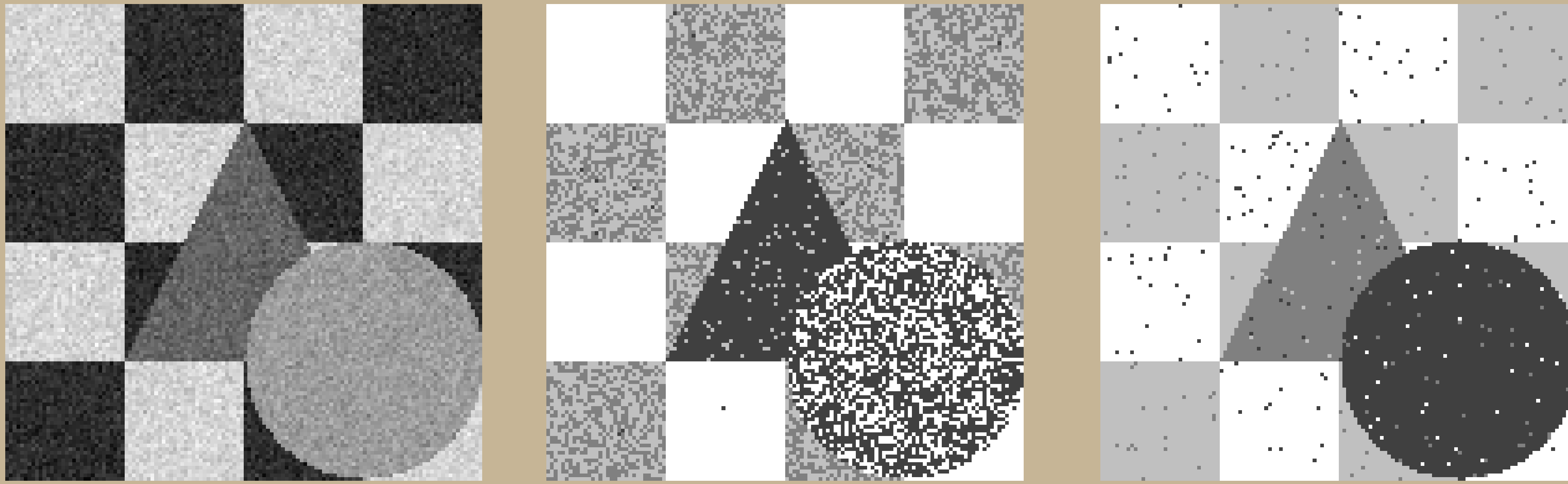
K-means (Meng-Hee Heng)

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.

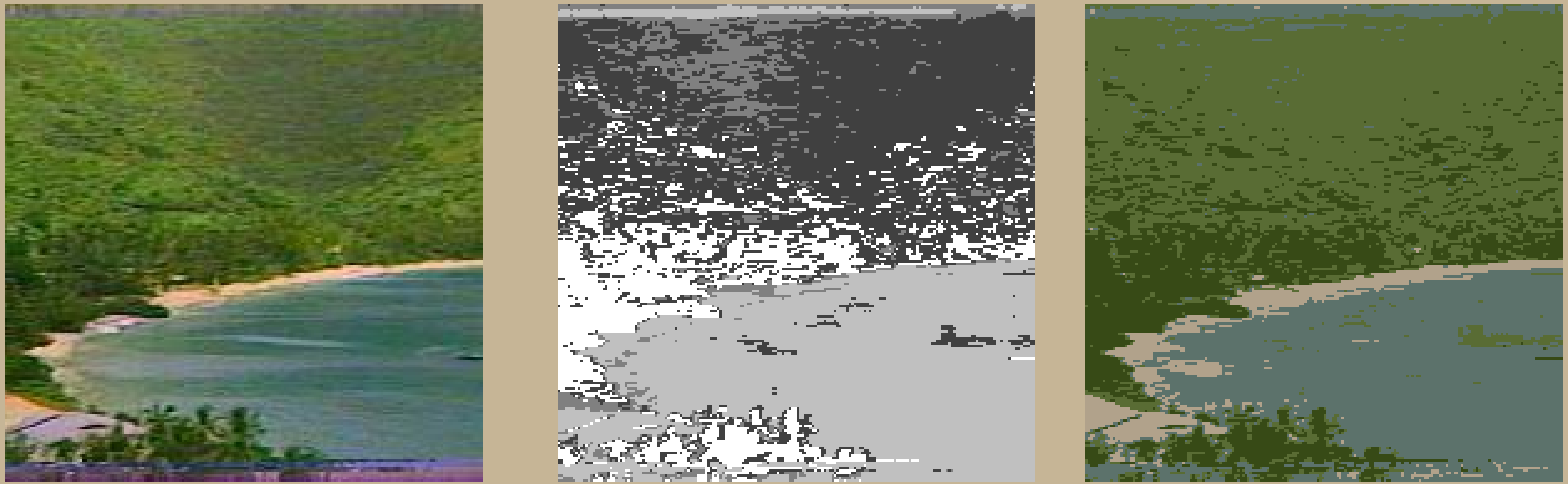
K-means (Meng-Hee Heng)



K-means példa

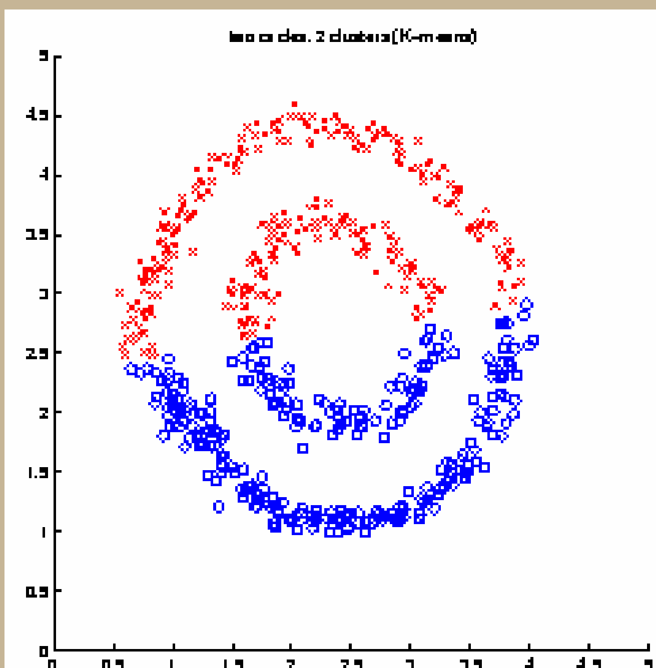


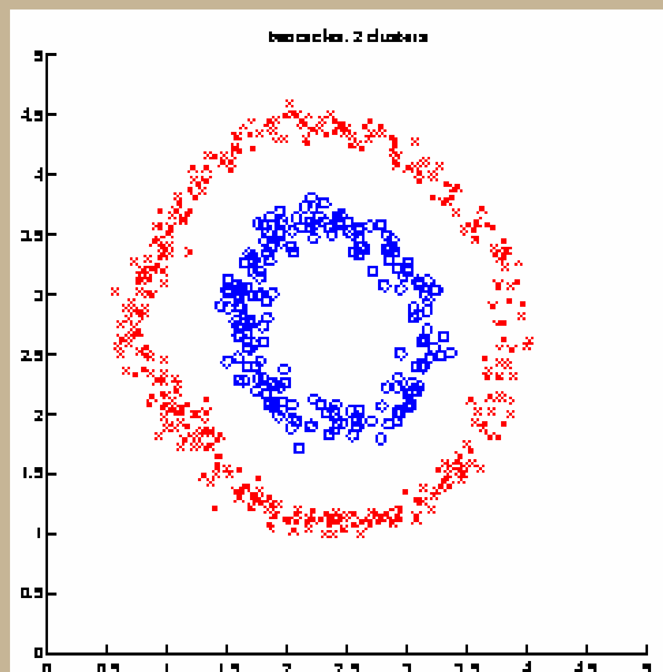
K=4



Problémák

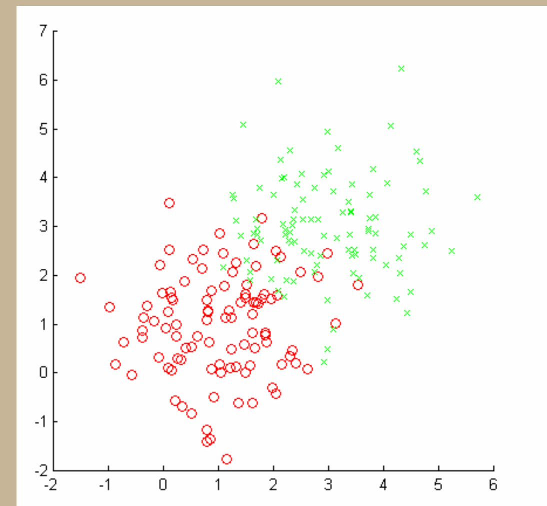
- Nem konvex klaszterek detektálására közvetlenül nem használható.
  - ➔Spektrális klaszterezés



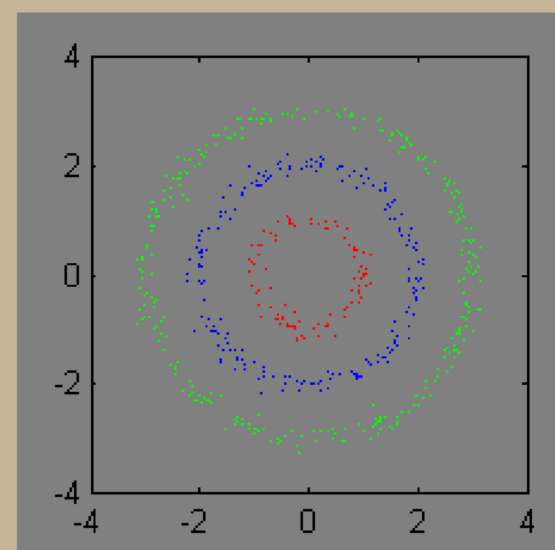


K-means

Spectral



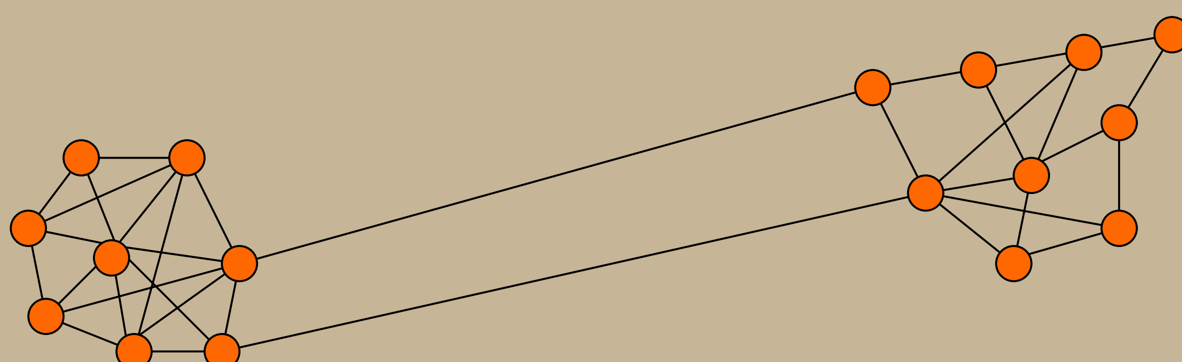
OK



???

Normalized Cut (Shi-Malik)

- Reprezentáljuk a képet gráfként:
  - a gráf pontjai pixelek vagy pixelek kis csoportja
  - a gráf éleihez rendelt súly a pixelek hasonlóságát jellemzi.
- Cél: Particionáljuk a gráf pontjait diszjunkt részgráfokra úgy, hogy
  - a hasonlóság egy részgráfon belül nagy,
  - de a részgráfok között kicsi.

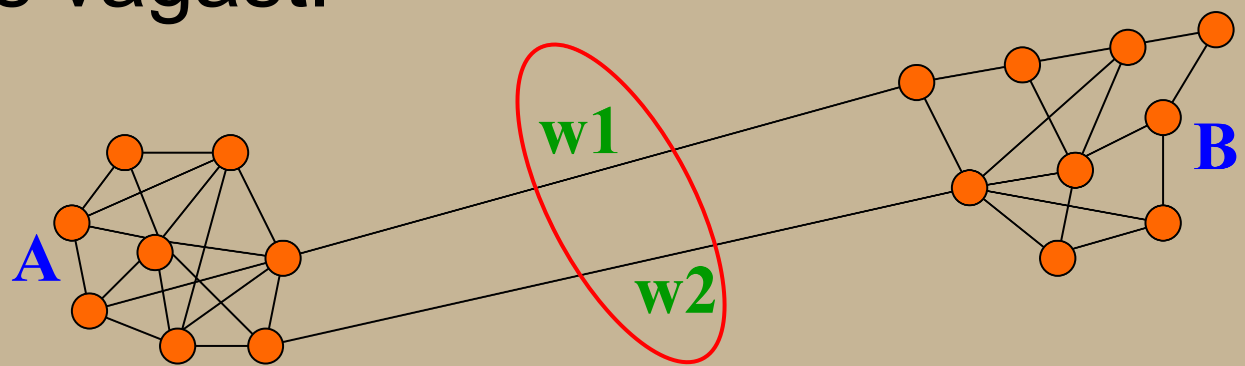


Minimális vágás

- Legyen ***G=(V,E)*** egy gráf.
  - Minden ***(u,v)*** élhez rendeljünk egy ***w(u,v)***-vel jelölt súlyt, amely az ***u*** és ***v*** közötti hasonlóságot reprezentálja.
- A ***G*** gráfot két diszjunkt ***A*** és ***B*** részgráfra bonthatjuk úgy, hogy töröljük az ***A*** és ***B*** között átmenő éleket.
- Legyen a vágás költsége:

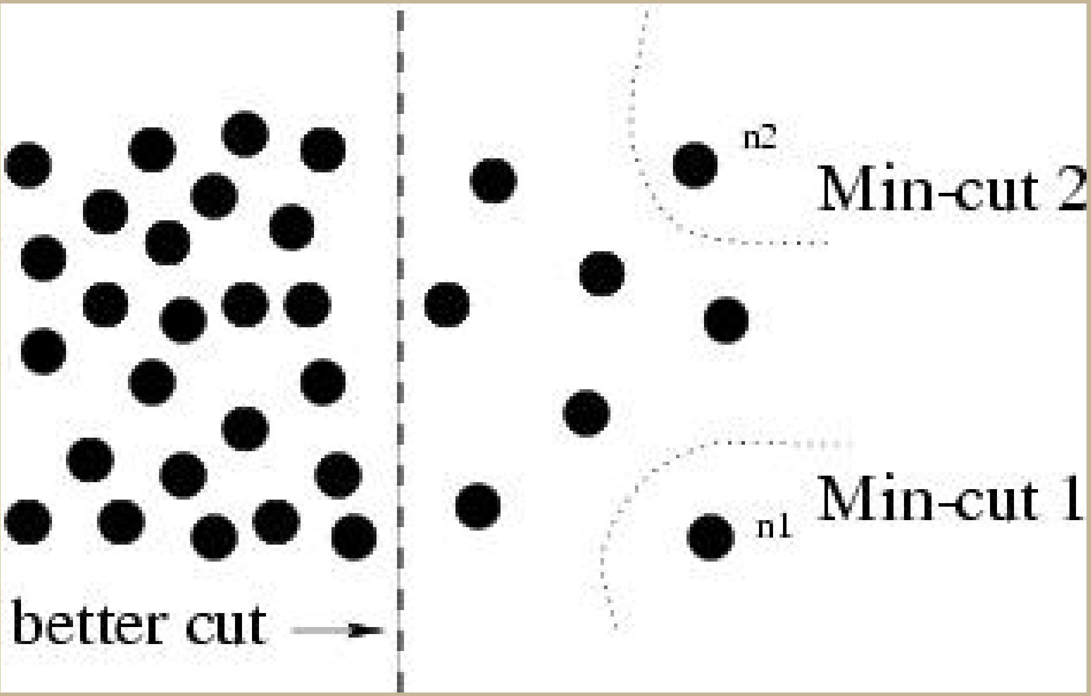
$$\mathrm{cut}(A,B) = \sum_{u \in A, v \in B} \mathrm{w}(u,v)$$

- ***G***-t particionálhatjuk úgy, hogy megkeressük a minimális vágást.



Normalizált vágás

- A minimális vágás hajlamos kis méretű részgráfokat (~régiókat) előállítani (kevesebb él átvágása kisebb költséget jelent…)



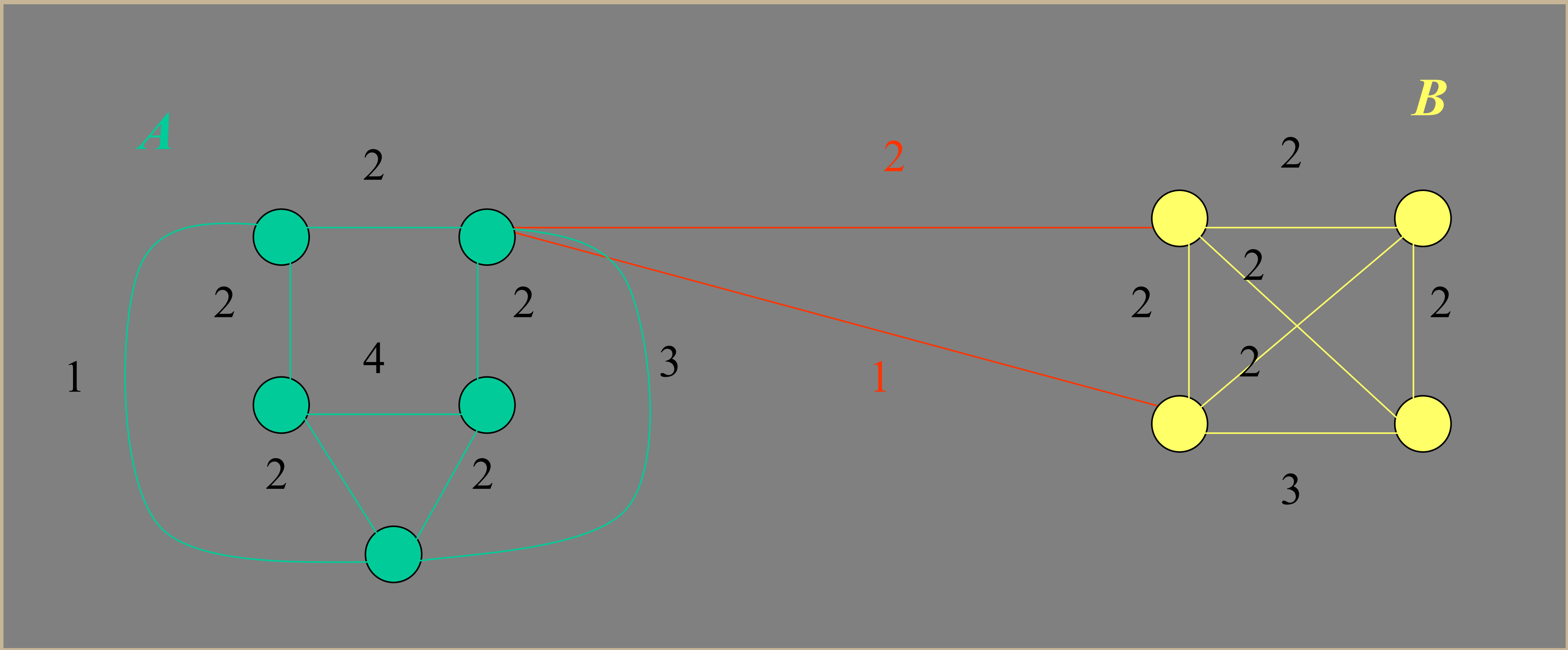
- Normalizált vágás a megoldás:
  - Egy ***A*** részgráf mennyire erősen kapcsolódik az egész ***G=(V,E)***-hez:

$$\mathrm{asso}(A,V) = \sum_{u \in A, t \in V} \mathrm{w}(u,t)$$

  - Normalizált vágás:

$$\mathrm{Ncut}(A,B) = \frac{\mathrm{cut}(A,B)}{\mathrm{asso}(A,V)} + \frac{\mathrm{cut}(A,B)}{\mathrm{asso}(B,V)}$$

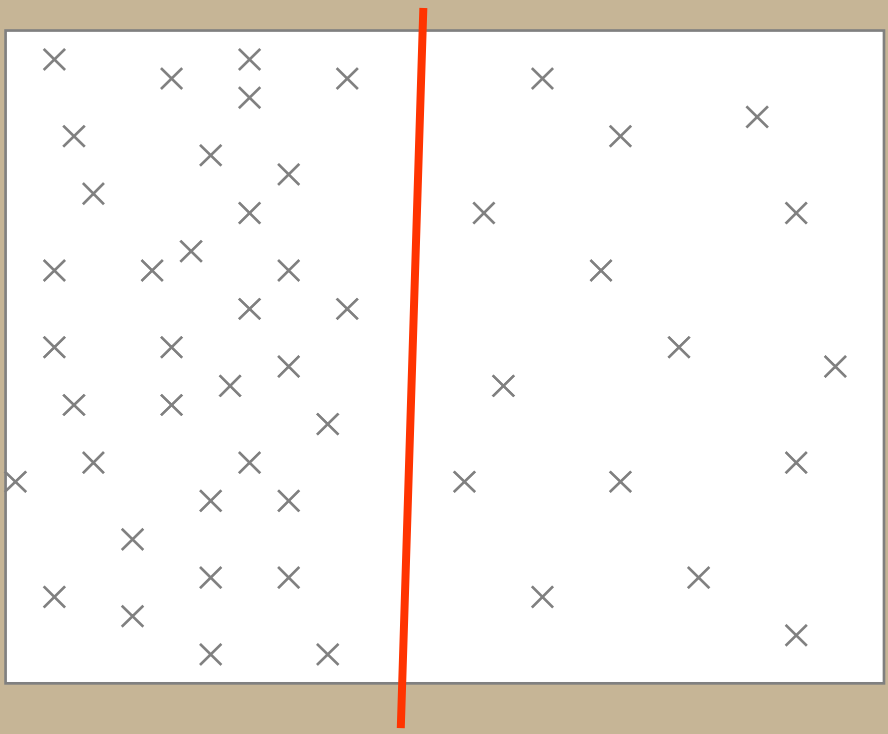
Normalizált vágás: Példa

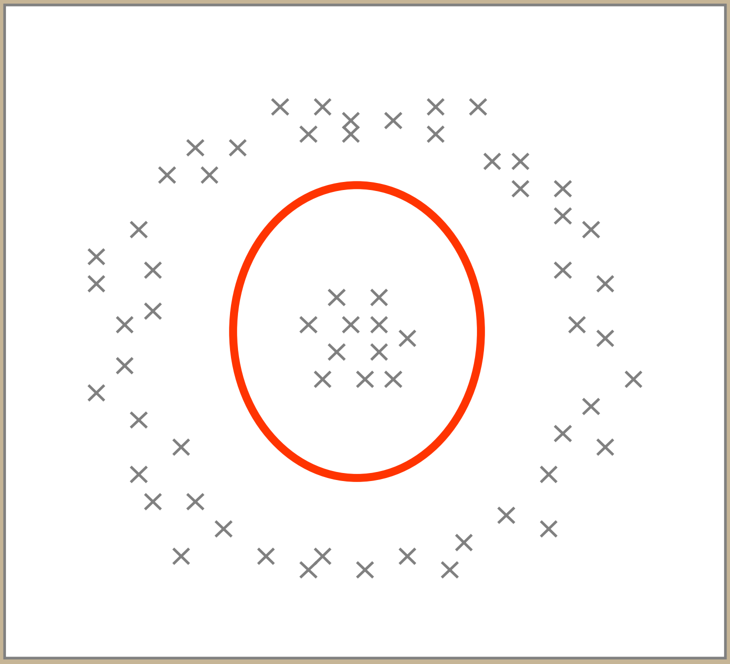


$$\mathrm{Ncut}\,(A,B) = \frac{\mathrm{cut}(A,B)}{\mathrm{asso}(A,V)} + \frac{\mathrm{cut}(A,B)}{\mathrm{asso}(B,V)} = \frac{3}{21} + \frac{3}{16}$$

Normalized Cut (NCut) kritérium

$$\hat{A} = \min_{A \subset V}(\mathrm{Ncut}(A, \overline{A}))$$





NP-Hard!

NCut Mátrix formában

- ***W(i,j)*** – súly mátrix (szimmetrikus)
- ***D(i,i)*** – az ***i*** csomóponthoz tartozó súlyok összege (diagonális)
- Levezetés után ezt kapjuk:
- Megoldás: általánosított sajátérték probléma.
  - Optimális megoldás a **második legkisebb** sajátérték
  - Az ***y'*** megoldás valós lesz, amelyet diszkrét értékekké kell alakítani
    - Például 0 értékkel küszöböljünk

$$D(i,i)=\sum_j w(i,j)$$

$$minNCut=\min_y \frac{y^t(D-W)y}{y^t Dy}$$

$$1)\ y_i \in \{1,-b\} \quad 2)\ y^t D1=0$$

NP-Complete

$$(D-W)y=\lambda Dy$$

Hogyan definiáljuk a súlyokat?

$$w(i,j) = \exp\left(\frac{\left\|F_i - F_j\right\|^2}{\sigma_F}\right) \times \begin{cases} \exp\left(\frac{\left\|X_i - X_j\right\|^2}{\sigma_X}\right) & \text{if } \left\|X_i - X_j\right\|^2 < r \\ 0 & \text{otherwise} \end{cases}$$

- ***X(i)*** - az ***i*** csomópont helye a képen
- ***F(i)*** - az ***i*** csomópont tulajdonság-vektora
  - intenzitás, szín, textúra, mozgás, …
- Vegyük észre, hogy ***w(i,j)=0*** azokra a csomópontokra, amelyekhez tartozó pixelek a képen távol esnek egymástól.

NCut szegmentáló algoritmus

1. Az input képből készítsük el a ***G=(V,E)*** gráfot, illetve a ***W*** hasonlósági (~súly) mátrixot.
2. Oldjuk meg a $(D-W)x=\lambda D x$ általénosított sajátérték problémát
3. A második legkisebb sajátvektor alapján particionáljuk ***G***–t két részgráfra.

- Hogyan particionáljunk kettőnél több részre?
  - Rekurzívan futtassuk a fenti algoritmust a részgráfokra
  - Vagy használjuk a magasabbrendű sajátvektorokat

NCut eredmények