

# Segmentacija slike

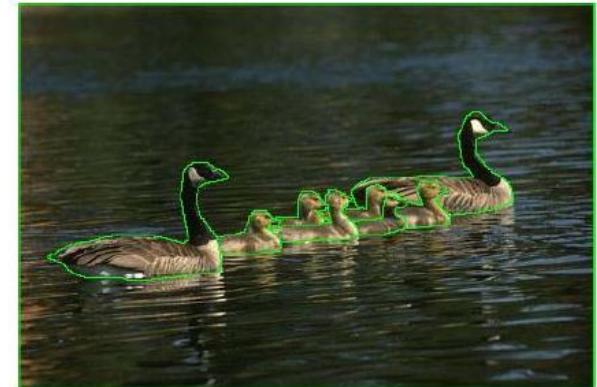
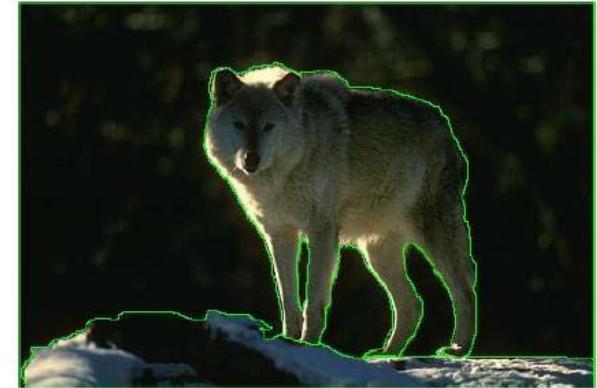
primenom klasterovanja

# Segmentacija slike – cilj

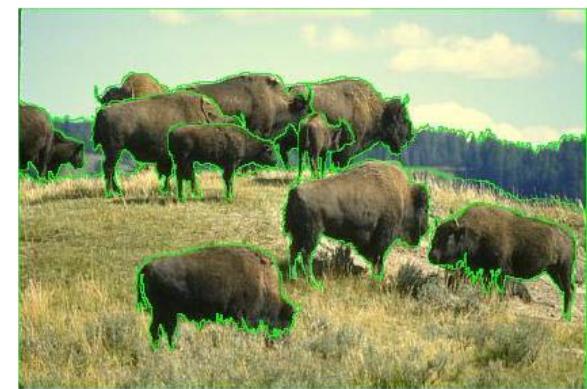
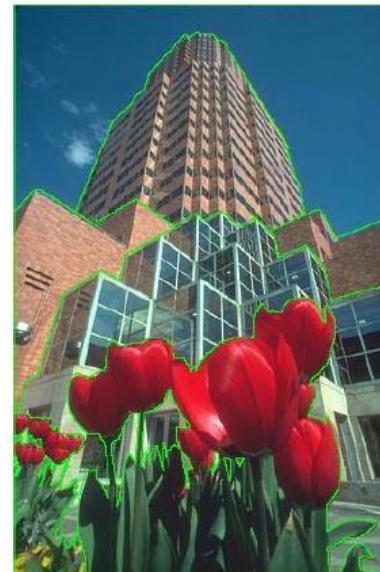
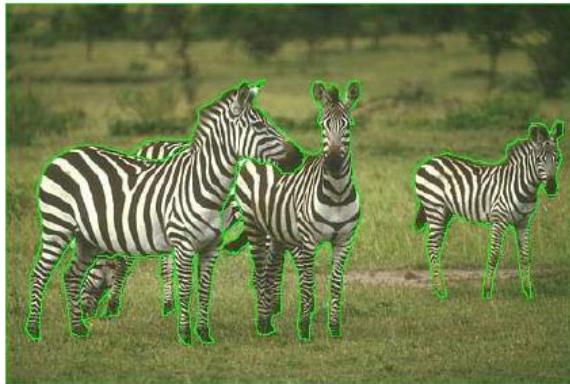
---

- Cilj: izdeliti sliku (ili video) na smislene regije
  - svetle regije, regije određene boje, regije sličnih karakteristika na medicinskim slikama, regije koje se pomeraju na isti način u videu,...
- Cilj segmentacije je da se pojednostavi i/ili promeni predstava slike u nešto što bi bilo smislenije i lakše za analizu
- Predstavlja jedan od ključnih koraka u analizi slike jer rezultati segmentacije mogu da utiču na sve naredne procese analize slike

# Segmentacija slike – cilj

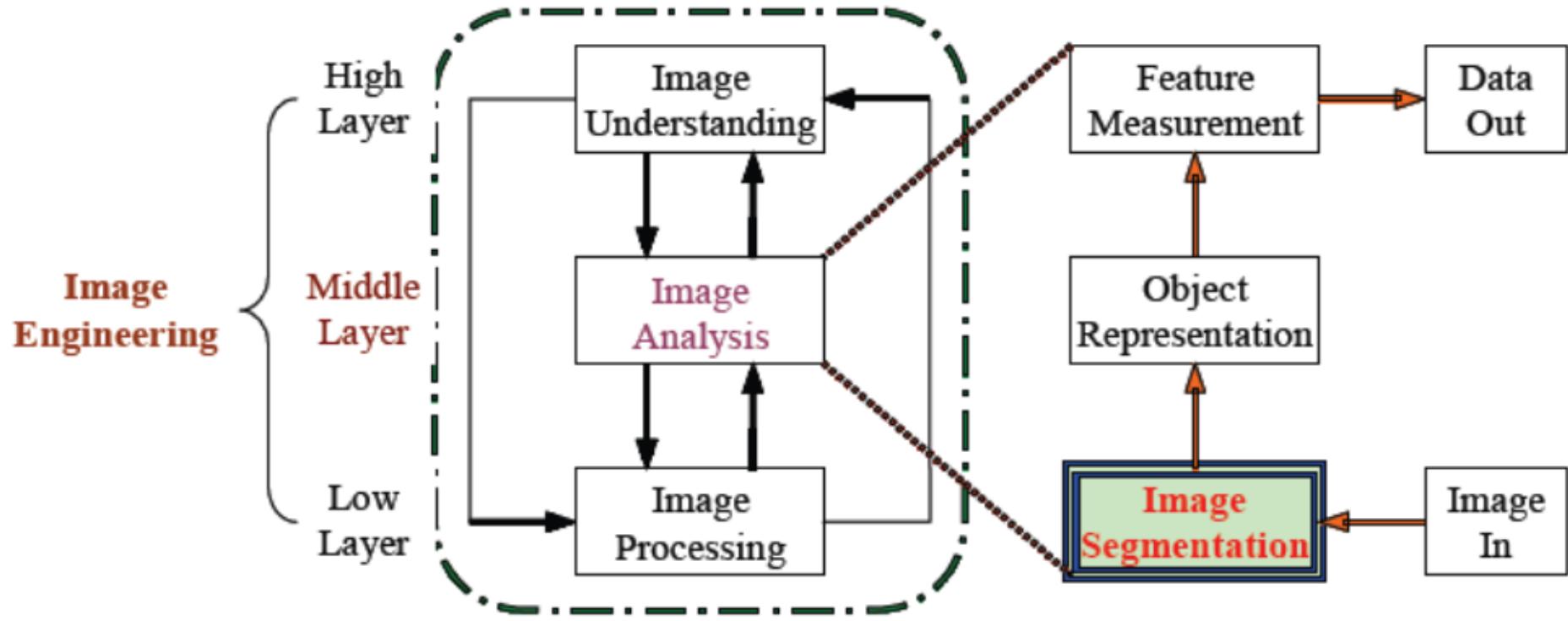


# Segmentacija slike – cilj

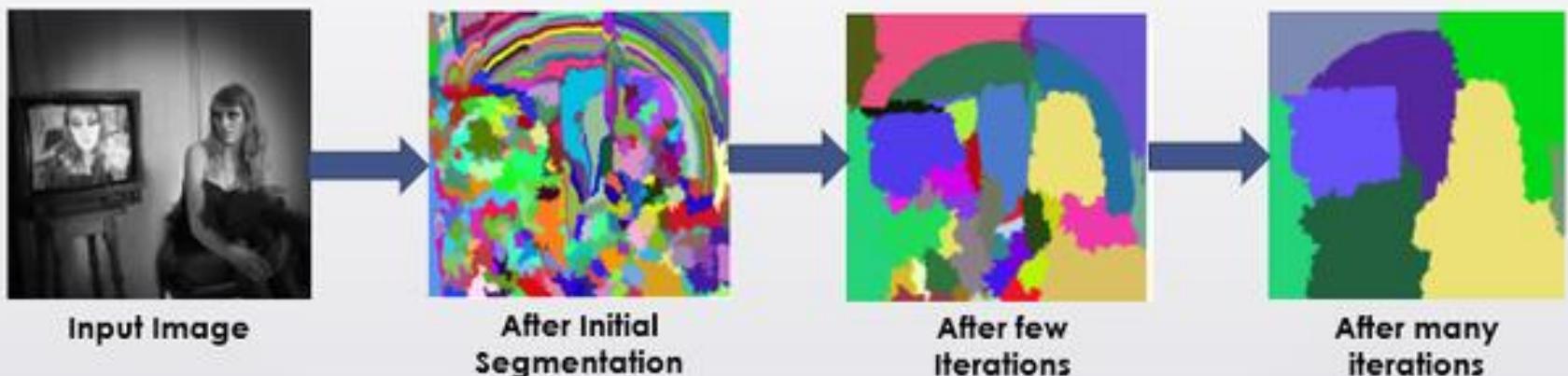
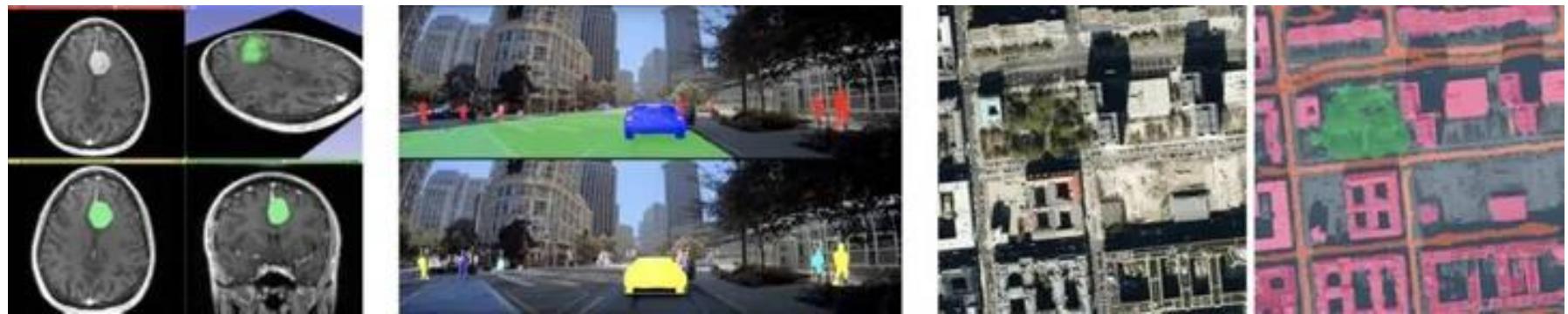


# Segmentacija slike – cilj

- Cilj: ekstrakcija informacija i reprezentacija objekta



# Primena segmentacije



# Segmentacija slike – izazovi

---

- Utvrđivanje *ground truth*
  - Manuelna segmentacija je dugotrajan i mukotrpan proces
  - Sem toga, rezultujuća segmentacija je subjektivna (dvosmislenost podataka, šum)
  - Subjektivne segmentacije različitih eksperata će se najverovatnije razlikovati
- Potreba za integracijom informacija visokog nivoa
- Zbog svega navedenog, trenutni stav je da se segmentacija evaluira u smislu krajnjeg cilja (celokupnog sistema čiji je deo)

# Definicija segmentacije

---

- Segmentacija slike je operacija particonisanja slike na kolekciju povezanih skupova piksela
- Neka  $R$  predstavlja celokupnu regiju slike. Segmentacija je particonisanje  $R$  na podregije  $R_1, R_2, \dots, R_n$  tako da važi:
  1.  $\bigcup_{i=1}^n R_i = R$  – svaki pixel mora da pripada nekoj regiji
  2.  $R_i$  je povezana regija,  $i = 1, 2, \dots, n$  (po definiciji susedstva)
  3.  $R_i \cap R_j = \emptyset$  za sve  $i$  i  $j$  takve da je  $i \neq j$
  4.  $P(R_i) = \text{TRUE}$  za sve  $i = 1, 2, \dots, n$  – uslov se odnosi na karakteristike koje regija treba da zadovolji (npr. svi pixels u regiji su iste boje)
  5.  $P(R_i \cup R_j) = \text{FALSE}$  za  $i = 1, 2, \dots, n$  – različite povezane regije su međusobno različite u smislu karakteristika koje uniformna regija treba da zadovolji

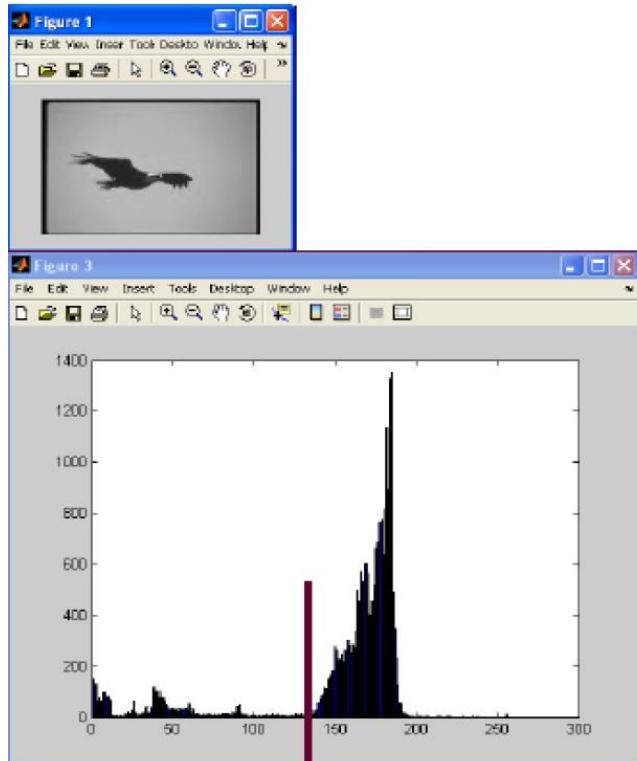
# Segmentacija slike

---

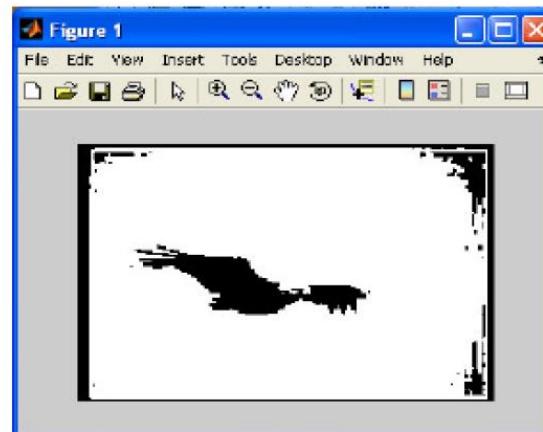
- Dakle, sve što treba da uradimo jeste da definišemo sličnost (homogenost)
  - Ali, šta je to što želimo da bude slično u svakoj regiji?
  - Da li možemo izdvojiti kriterijume koji će rezultovati time da regije budu objekti od značaja?
- Primeri pristupa:
  - Pristupi bazirani na histogramu
  - Pristupi bazirani na klasterovanju
  - *Region growing*
  - *Split-and-merge*
  - Morfološki pristupi
  - Pristupi bazirani na grafovima

# Segmentacija bazirana na histogramu

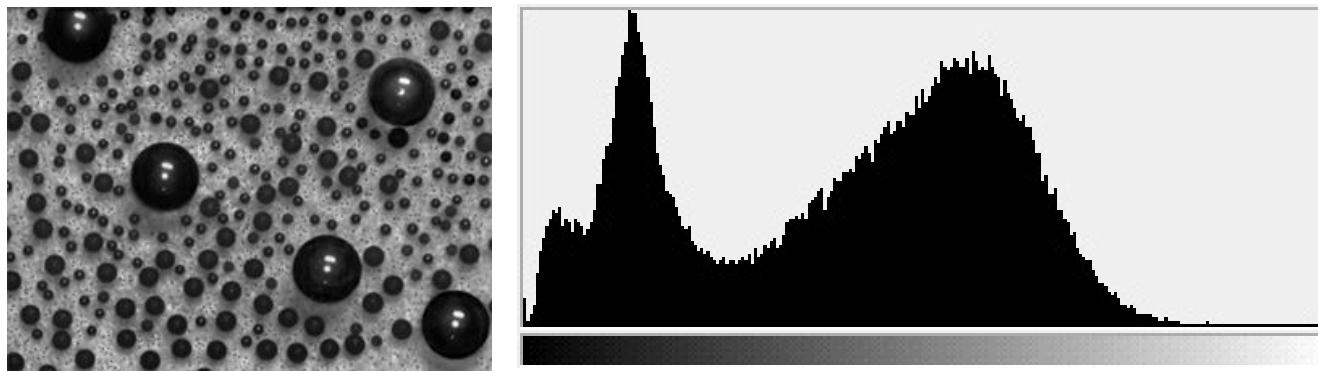
- Koliko „naradžastih“ piksela ima na slici?
- Na ovaj tip pitanja se može odgovoriti posmatranjem histograma



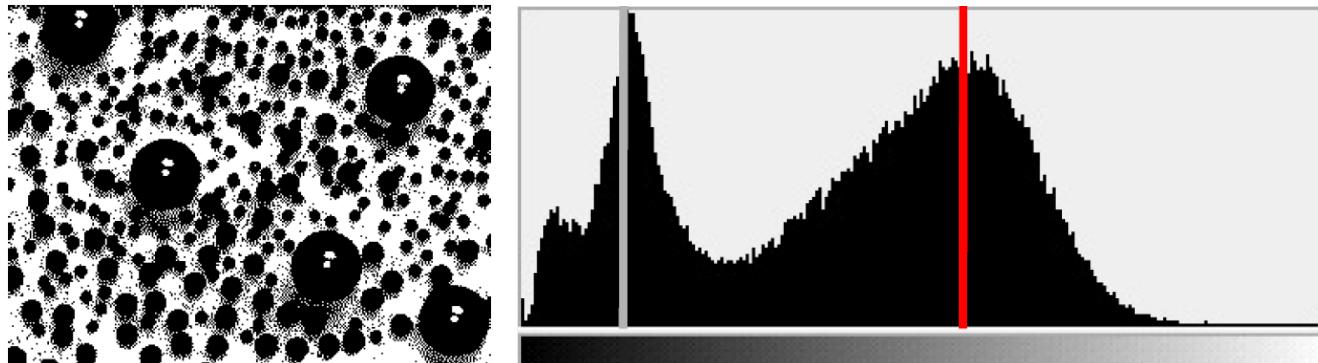
`imshow(B > 140)`



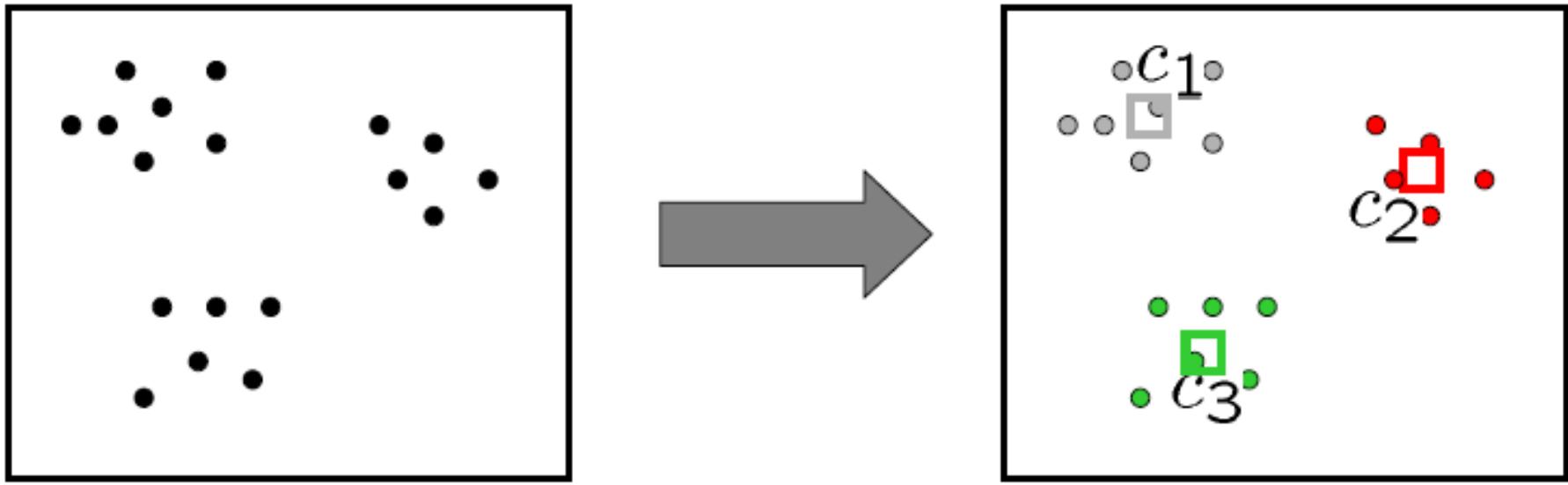
# Segmentacija bazirana na histogramu



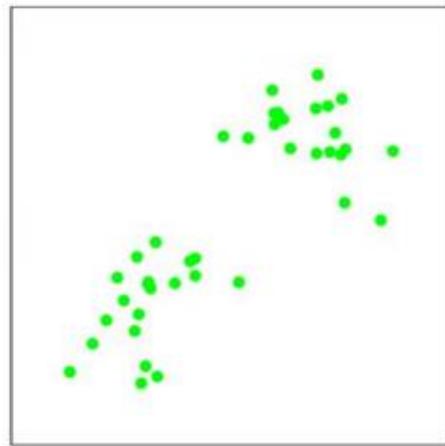
- Koliko modaliteta ima na slici?
- Ovo možemo rešiti tako što smanjimo broj boja na  $K$  i mapiramo svaki piksel na najbližu boju
- Na primer, ako koristimo dve boje:



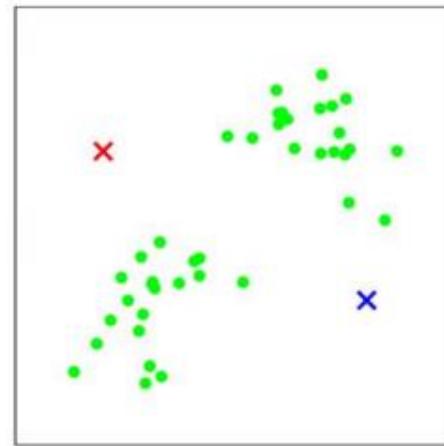
# Segmentacija bazirana na klasterovanju



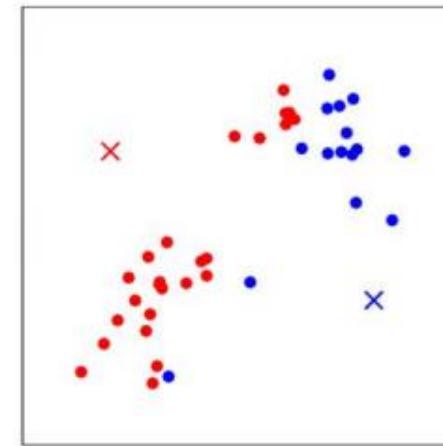
# K-means



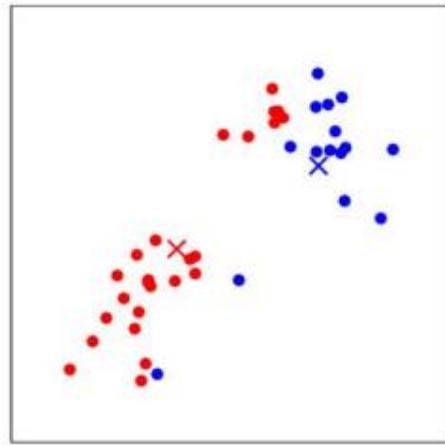
(a)



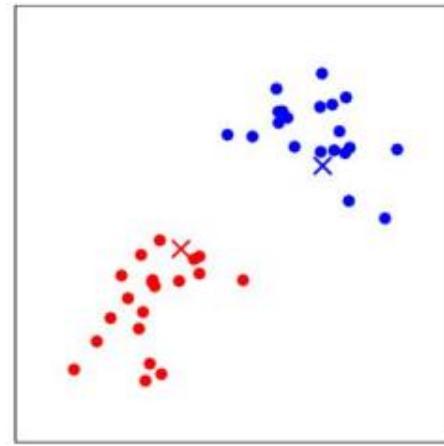
(b)



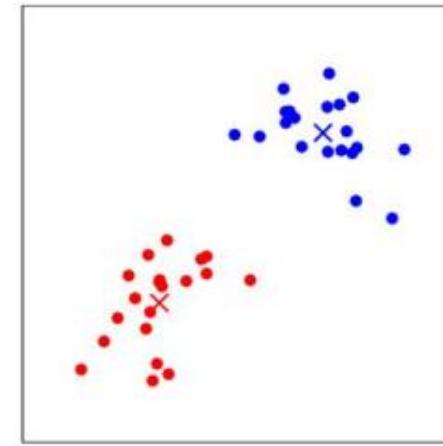
(c)



(d)



(e)



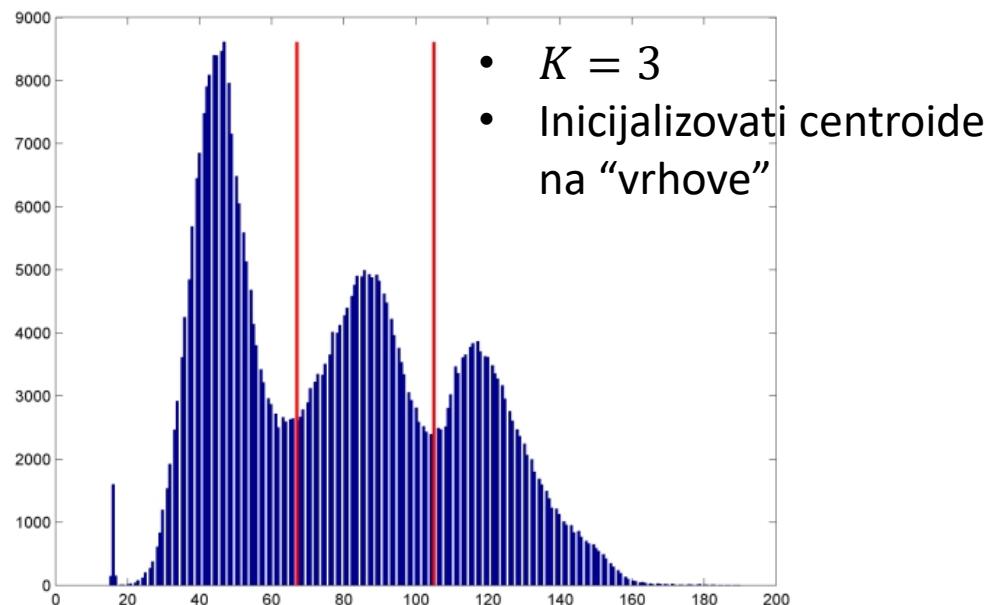
(f)

# K-means

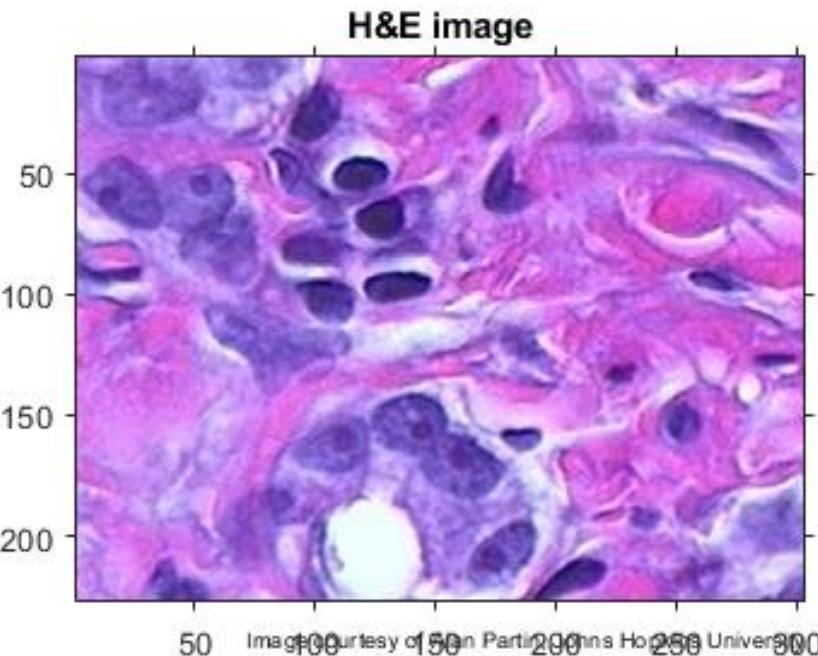
<b>Ulaz</b>	<ul style="list-style-type: none"><li>• <math>T = \{x^{(i)}, i = 1, \dots, N\}</math> – trening skup</li><li>• <math>k</math> – broj klastera</li></ul>
<b>Postupak</b>	<ul style="list-style-type: none"><li>• Na slučajan način inicijalizovati centroide <math>\mu_1, \dots, \mu_k</math></li><li>• Dok ne dođe do konvergencije:<ul style="list-style-type: none"><li>• Za svaku opservaciju <math>x^{(i)}</math> pronaći klaster sa najbližim centroidom (i dodeliti opservaciju tom klasteru):<math display="block">z^{(i)} \leftarrow \arg \min_j \ \mu_j - x^{(i)}\ _2^2</math></li><li>• Ažurirati centroide klastera prema opservacijama koje su im dodeljene (<i>mean</i>)<math display="block">\mu_j = \frac{1}{n_j} \sum_{i:z^{(i)}=j} x^{(i)}</math><math display="block">n_j \text{ - broj opservacija u klasteru } j</math></li></ul></li></ul>
<b>Izlaz</b>	<ul style="list-style-type: none"><li>• <math>\{(x^{(i)}, z^{(i)}), i = 1, \dots, N\}</math></li></ul>

# Segmentacija bazirana na klasterovanju

- Varijante *k-means* algoritma:
  - Različiti načini inicijalizacije centroida
  - Različiti kriterijumi stopiranja
  - Određivanje korektnog broja klastera ( $k$ ) za datu sliku



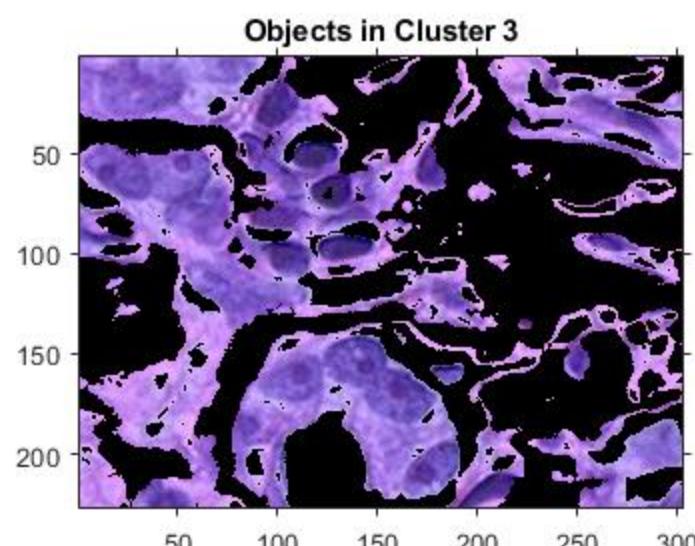
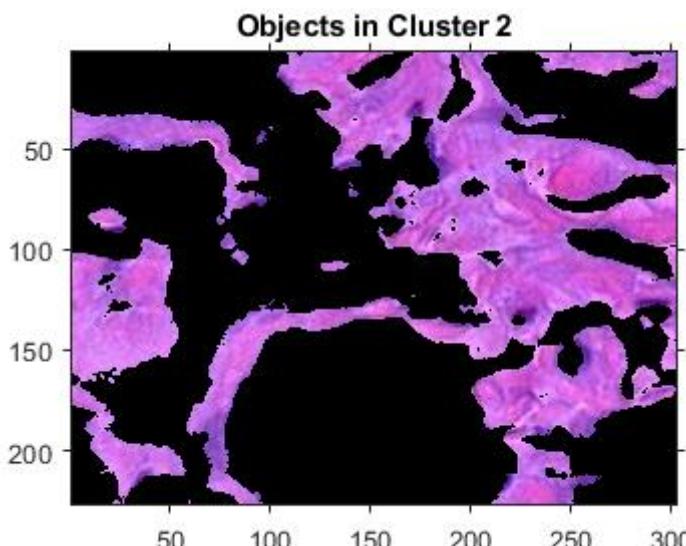
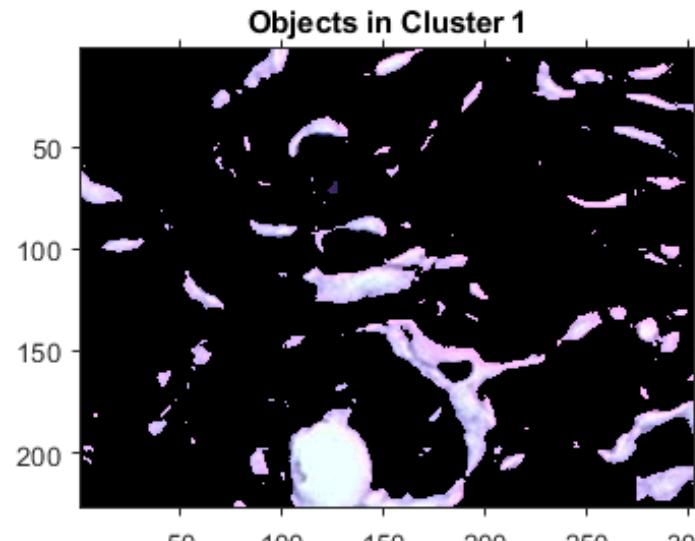
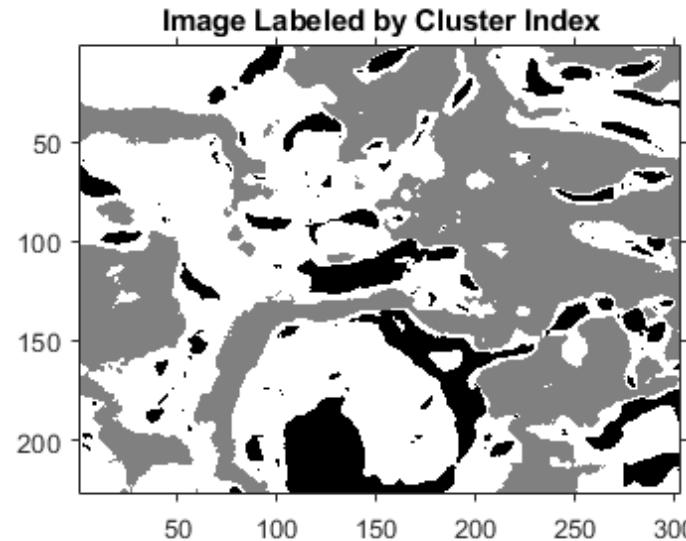
# Klasterovanje prema boji – primer



- Slika: mikroskopski snimak tkiva obojenog hematoksilinom i eosinom
- Ovo je standardna metoda koja pomaže patologima da razlikuju tipove tkiva
- Želimo da klasterujemo tkiva prema boji

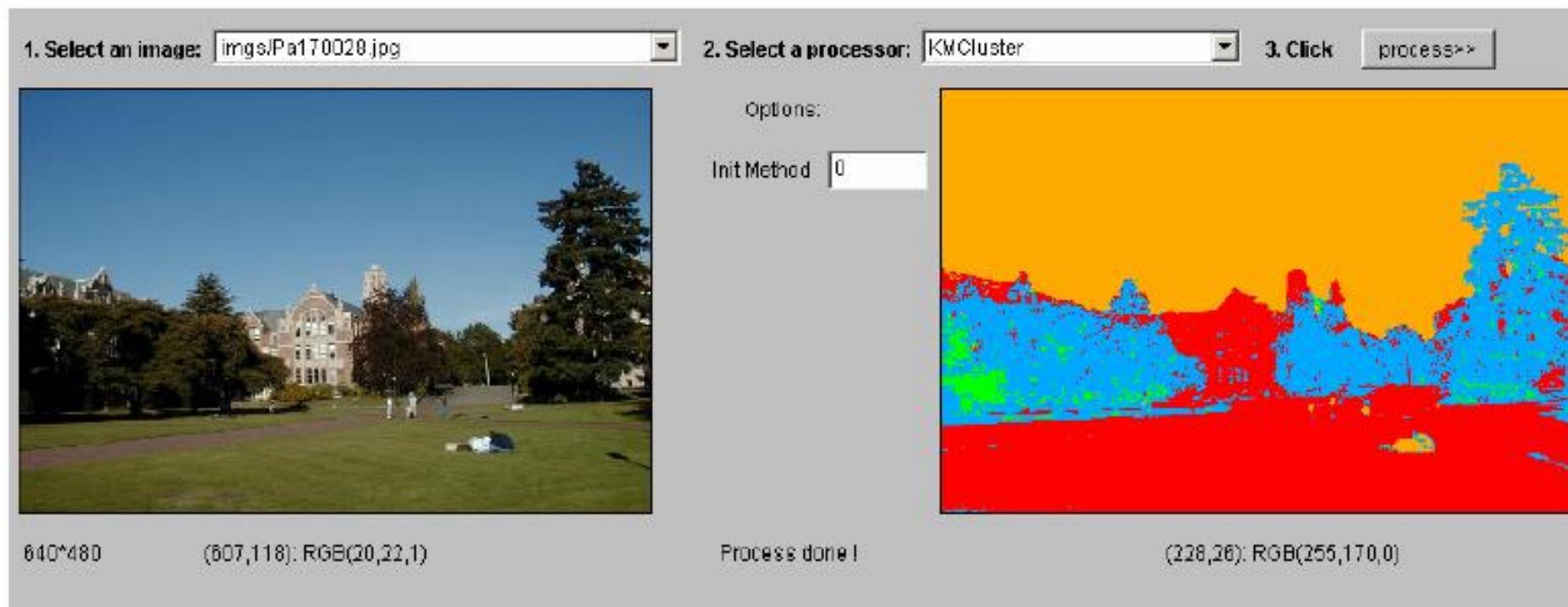
- Želimo klasterujemo po boji (nezavisno od osvetljenja)
  - Koristimo LAB model boja (L - osvetljenost, A – gde se boja nalazi na crveno-zelenoj osi i B – gde se boja nalazi na plavo-žutoj osi)
  - Koristimo samo informacije iz A i B
  - Na slici su prisutne plava, bela i ružičasta boja → tri klastera

# Klasterovanje prema boji – primer



Ako dalje želimo da izdvojimo jezgra  
ćelija (tamnije boje)  
možemo da koristimo  $L$  kanal i globalni prag

# Primer: *k-means* klasterovanje po boji



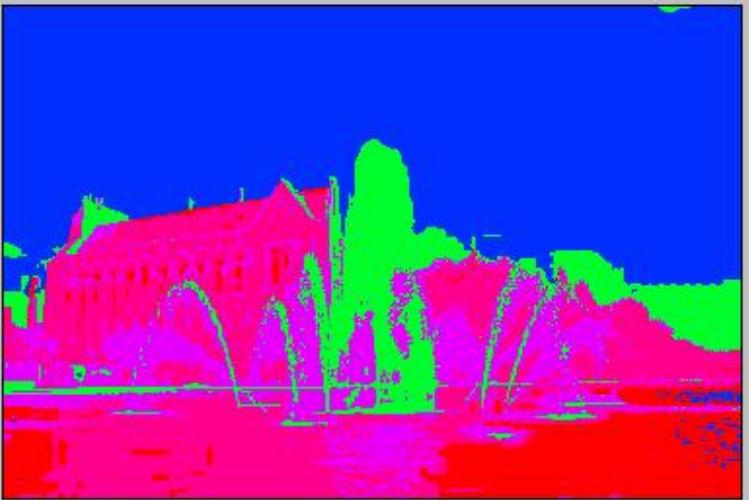
# Primer: *k-means* klasterovanje po boji

1. Select an image:  2. Select a processor:  3. Click

Options:  
Init Method

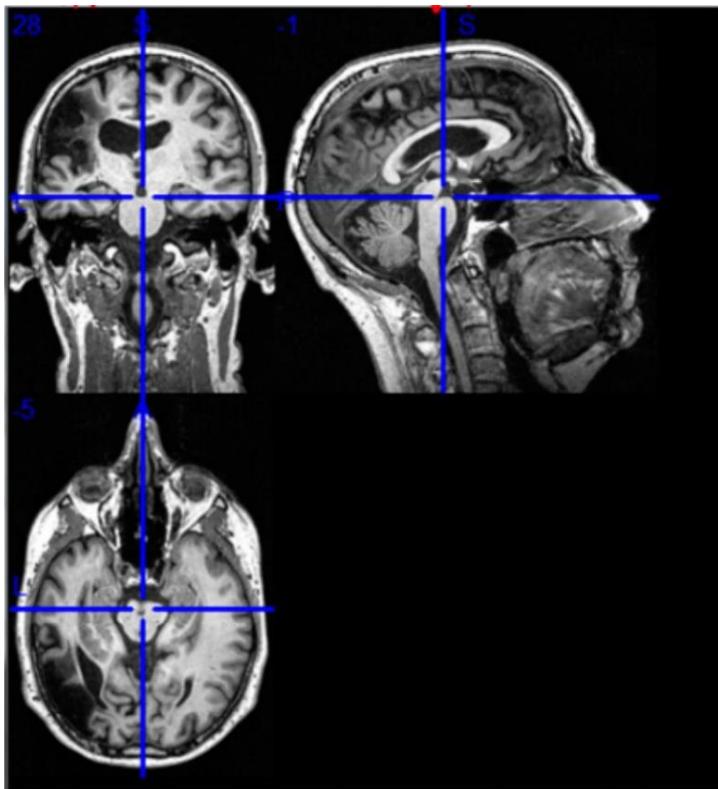


640\*480 (636,95): RGB(102,130,151)



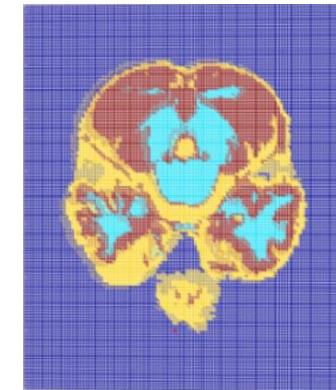
Process done ! (590,209): RGB(0,46,255)

# Primena: procesiranje medicinskih slika



- MRI snimak. Na ovim snimcima postoje tipično tri grupe:
  - *White matter*
  - *Gray matter*
  - *Cerebral Spinal Fluid (CSF)*
- Želimo da klasifikujemo voksele (3D piksele) u ove tri kategorije
- Domensko znanje:
  - Vokseli istog intenziteta treba da pripadaju istoj grupi
  - Susedni vokseli treba da pripadaju istoj grupi

$$v(x, y, z) = \begin{bmatrix} \alpha \cdot I(x, y, z) \\ \beta \cdot x \\ \beta \cdot y \\ \beta \cdot z \end{bmatrix} \quad D(v_i, v_j) = \|v_i - v_j\|$$



# Primena: procesiranje medicinskih slika

Segmentation

Grey matter

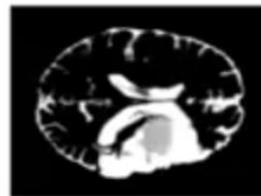
White matter

CSF

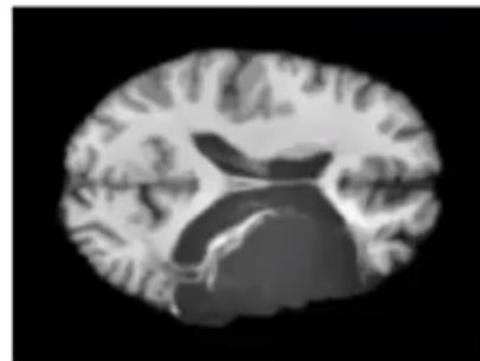
K-means  
Segmentation



SPM8  
Segmentation

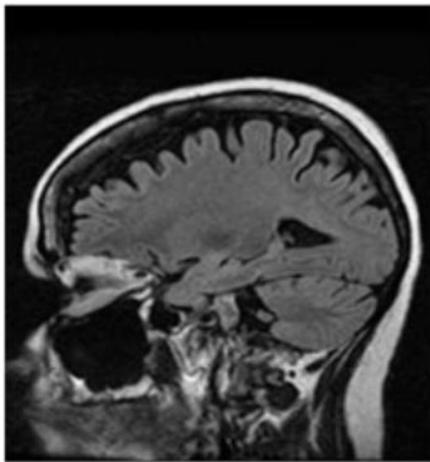


Original Image

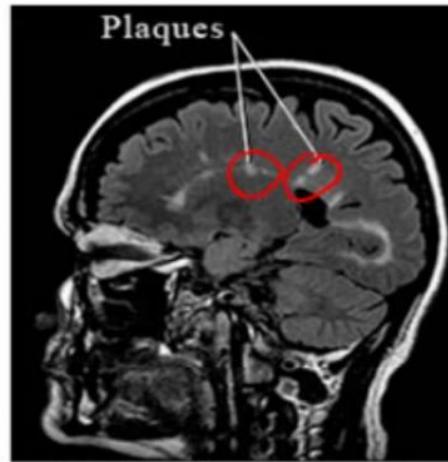


\*SPM8 is a public toolbox supported by Matlab

# Primena: procesiranje medicinskih slika



Healthy Brain

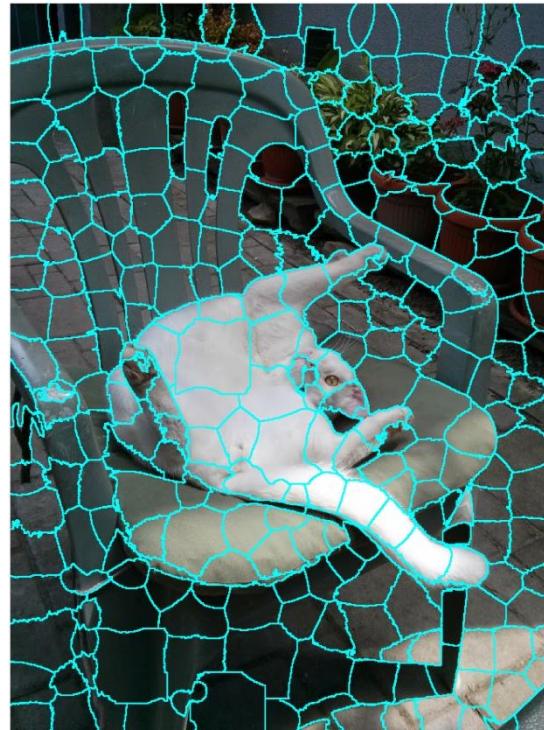


Brain with damage  
(Lesions or plaques)

- Primena: pokušaj da se identifikuju
  - *Stroke lesion*
  - *Perilesional region*
- Ovo bi pomoglo da se predvidi oporavak ili reakcija na terapiju
- Tipično, ovo se crta ručno, što je vremenski zahtevan i subjektivan proces
- Problemi: šum, slabe ivice, nehomogenost podataka

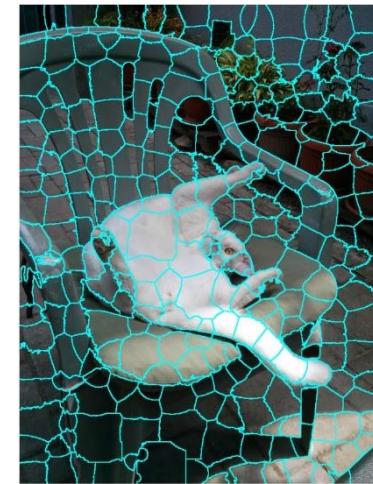
# Superpixels

- Rezultati koje smo dobili *k-means* klasterovanjem u odnosu na boju (ne i lokaciju) nisu loši u odnosu na izdvojene boje
- Međutim, izdvojene oblasti su dosta diskonektovane. Želeli bismo malo više prostorne koherentnosti
- Modifikacija *k-means* koja se često koristi u obradi slike: *superpixels*
  - Pronalaze se kontinualne regije koje su sličnog intenziteta/boje



# Superpixels

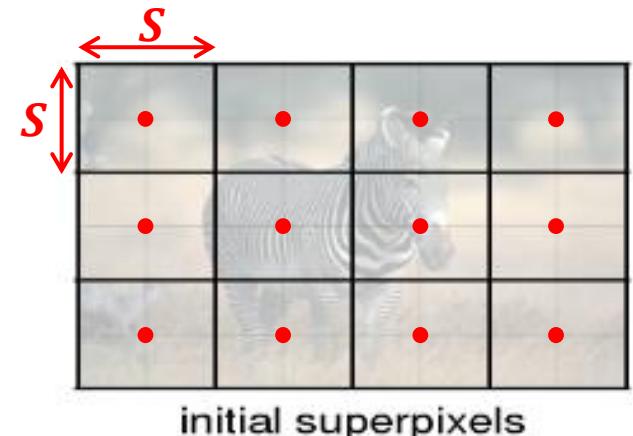
- Kompaktna reprezentacija slike
  - Sa hiljade superpiksela možemo predstaviti milione piksela
- Bolja reprezentacija za kasniju segmentaciju
  - „drži stvari zajedno“
  - Algoritmi za segmentaciju će biti efikasniji ako koriste superpiksele umesto piksela
- Algoritam
  - Modifikacija *k-means*
  - Pored boje se koristi i prostorna povezanost
  - Jedna varijanta algoritma: SLIC (*Simple Linear Iterative Clustering*)



# SLIC (*Simple Linear Iterative Clustering*)

- Klasteruje se 5D vektor  $[l, a, b, x, y]$ 
    - $(l, a, b)$  - LAB model boja;  $(x, y)$  - prostorna informacija
1. Inicijalizacija centroida  $C_k = [l_k, a_k, b_k, x_k, y_k], k \in \{1, \dots, K\}$  odabirom piksela na regularnoj mreži koraka  $S$

- Pomeriti centroide na poziciju sa najmanjim gradijentom u  $3 \times 3$  susedstvu
- Time izbegavamo da postavimo centroide na ivicu ili na piksel koji predstavlja šum

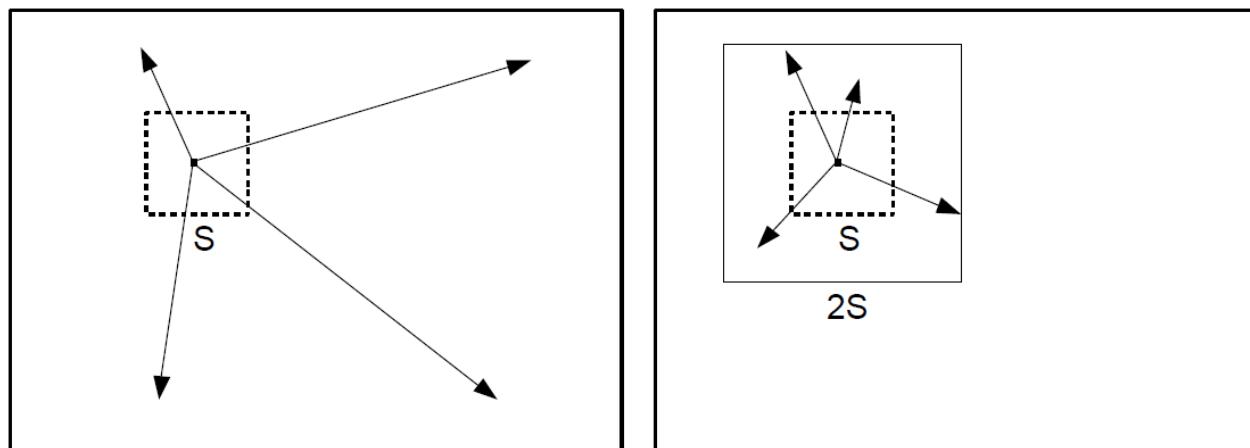


Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11), pp.2274-2282.

# SLIC (*Simple Linear Iterative Clustering*)

2. Za svaki centroid  $C_k$ ,

- Za svaki piksel  $i$  iz  $2S \times 2S$  susedstva  $C_k$  izračunati udaljenost  $D$  između  $C_k$  i  $i$
- Ako je  $D < d(i)$  (udaljenosti piksela  $i$  od centroida kome je piksel prethodno dodeljen), dodeli piksel  $i$  centru  $C_k$  i postavi  $d(i) = D$



(a) standard  $k$ -means searches  
the entire image

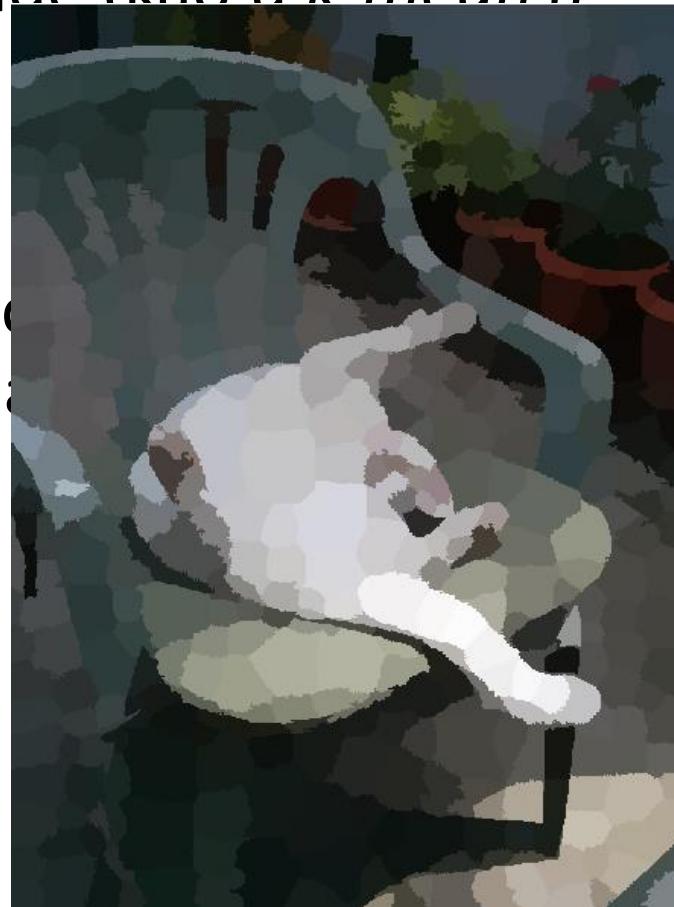
(b) SLIC searches  
a limited region

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11), pp.2274-2282.

# SLIC (*Simple Linear Iterative Clustering*)

## 3. Ažuriraj centroide (kao u *k-means*)

- Ponavljati korake
- Na kraju, (opciono) alocirati svaku boju svakog piksela prosekom boje klusta u kojem se ona nađe



Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11), pp.2274-2282.

# SLIC (*Simple Linear Iterative Clustering*)

- Problem: kako definisati distancu  $D$ ?
  - Boja piksela u LAB prostoru  $[l, a, b]$  se kreće u poznatom opsegu
  - Pozicija piksela  $[x, y]$  se kreće u opsegu koji zavisi od veličine slike
  - Ova dva opsega mogu biti veoma različita. Moramo normalizovati bliskost u prostoru boja i prostornu bliskost kako jedna ne bi imala mnogo jači uticaj od druge

$$d_c = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}, \quad d_s = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$D = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2}$$

- $N_c$  - maksimalna distanca u prostoru boja
- $N_s$  - maksimalna prostorna distanca – već smo je ograničili  $N_s = S = \sqrt{N/K}$ , gde je  $N$  broj piksela, a  $K$  broj klastera

# SLIC (*Simple Linear Iterative Clustering*)

- Za  $N_c$  je malo teže odrediti šta može biti
  - Rastojanja boja mogu veoma varirati od klastera do klastera i od slike do slike
  - U praksi, fiksiramo  $N_c$  na neku konstantu  $m$  koja reguliše nagodbu prostorne distance i distance u prostoru boja
    - Veliko  $m$ : superpikseli će biti kompaktniji
    - Malo  $m$ : superpikseli će se više vezati za ivice na slici
- Ovim se mera distance pojednostavljuje na:

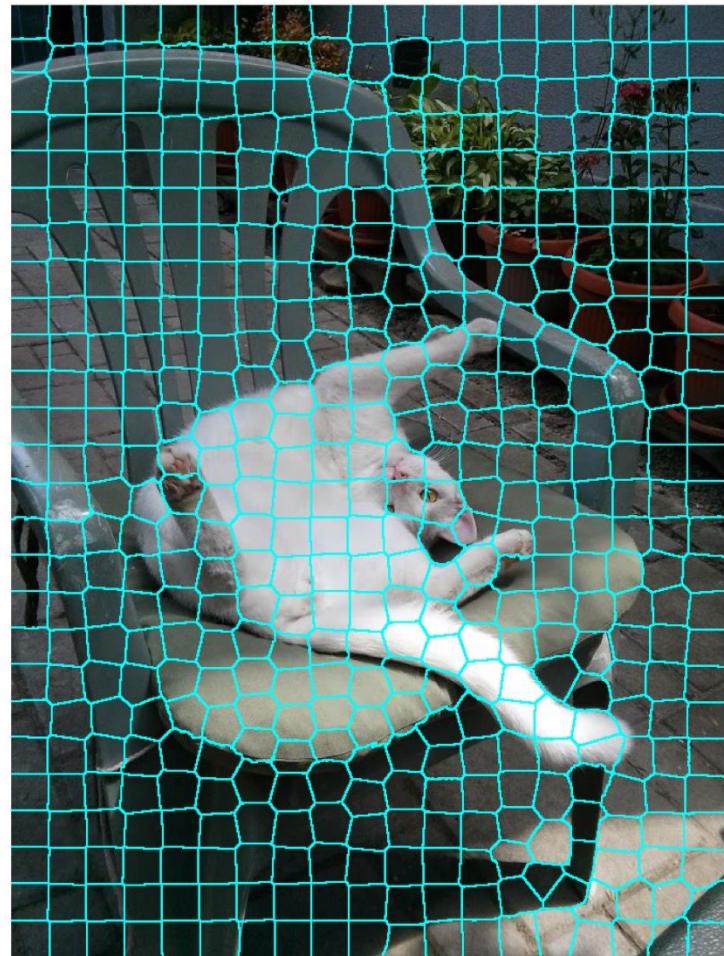
$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2}$$

# Superpikseli: kompaktnost



$m = 5$

superpikseli prate ivice



$m = 100$

superpikseli su kompaktniji

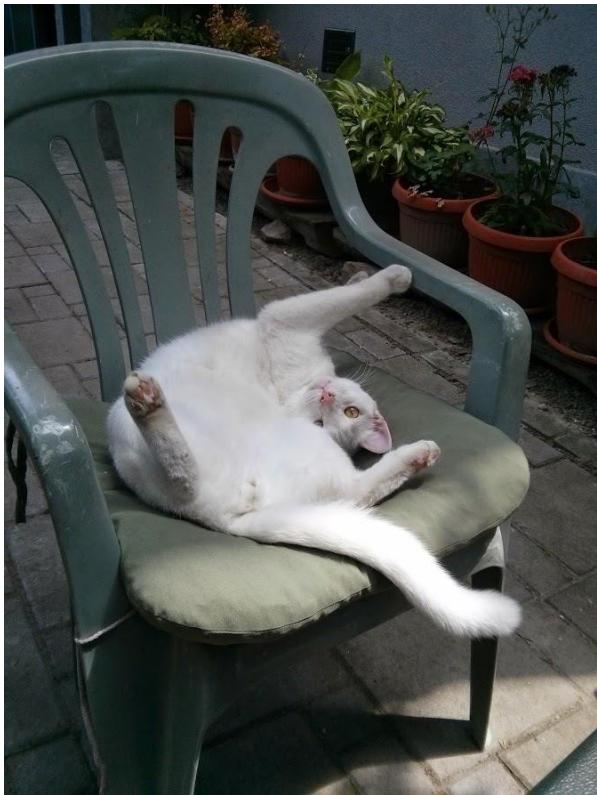
# Broj superpiksela



Fig. 1: Images segmented using SLIC into superpixels of size 64, 256, and 1024 pixels (approximately).

- Podešavanje prema tome koliko želite da superpikseli prate ivice

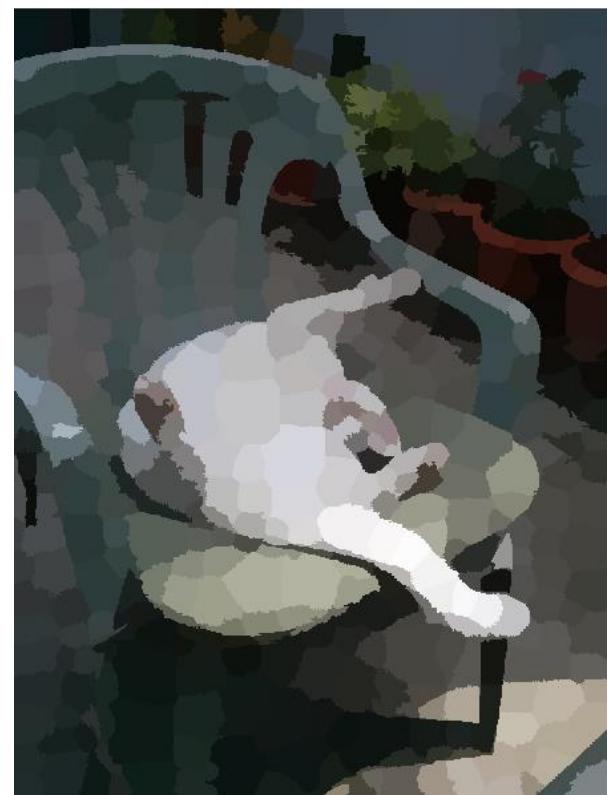
# Broj superpiksela



$881 \times 668 = 595\,188$



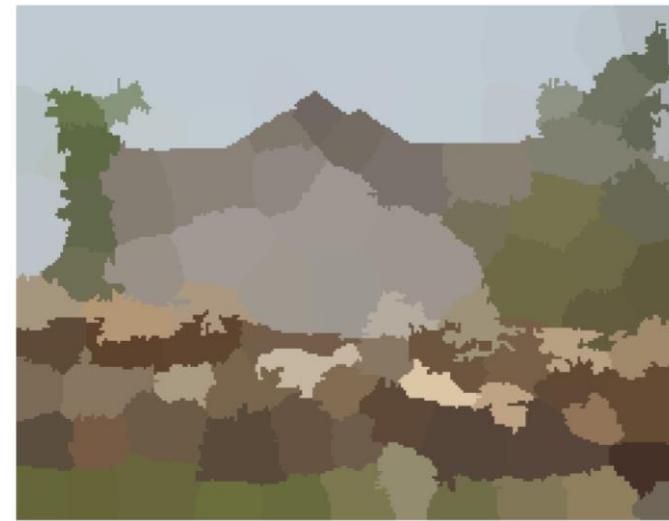
500 000 superpiksela,  
Svaki piksel je zamenjen  
prosekom boje klastera



500 superpiksela,  
Svaki piksel je zamenjen  
prosekom boje klastera

# Segmentacija

- Nad superpixelsima možemo pustiti algoritam koji će ih grupisati u regije
  - Klasterovanje višeg nivoa:
  - Superpixelsima smo uklonili šum na originalnoj slici koji bi mogao smetati algoritmu za segmentaciju
  - Važne stvari su ostale, a uklonili smo nevažne stvari



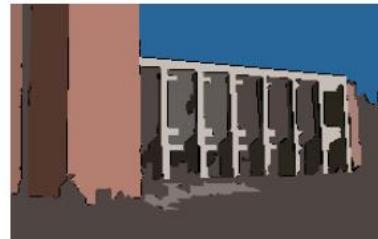
# Segmentacija bazirana na klasterovanju

- Pored boje, klasterovanje može da koristi i druga obeležja. Na primer, teksture

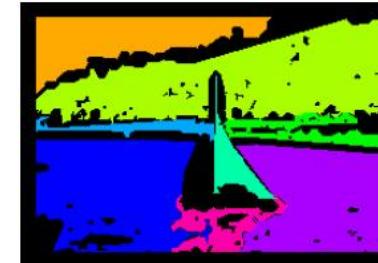
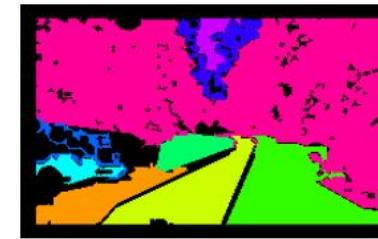
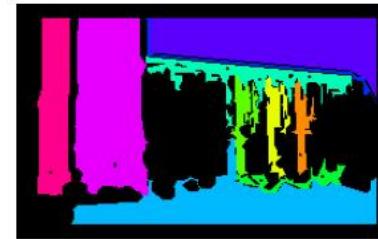
Original Images



Color Regions

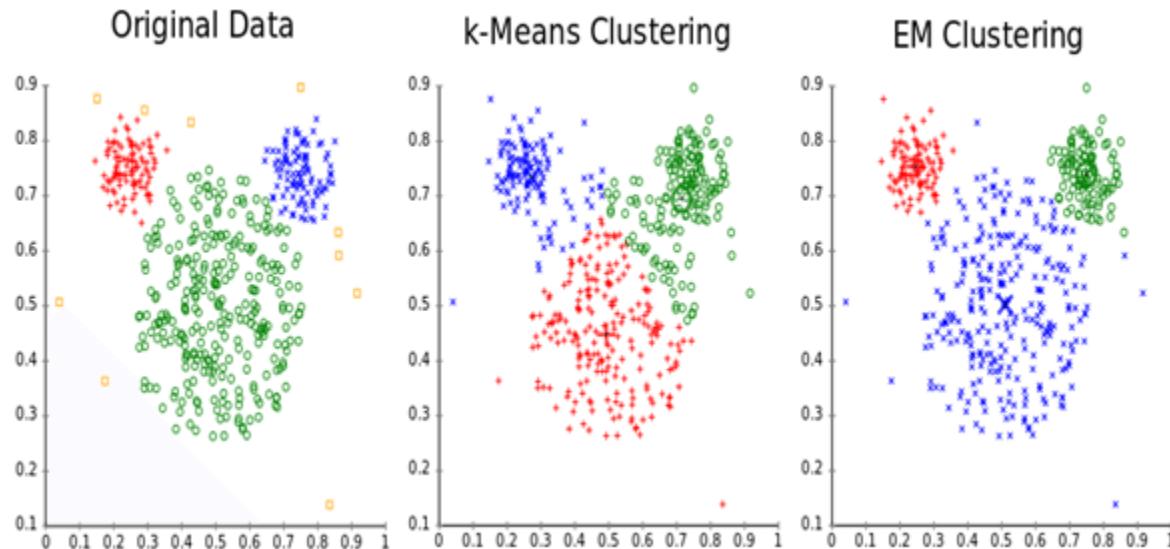


Texture Regions

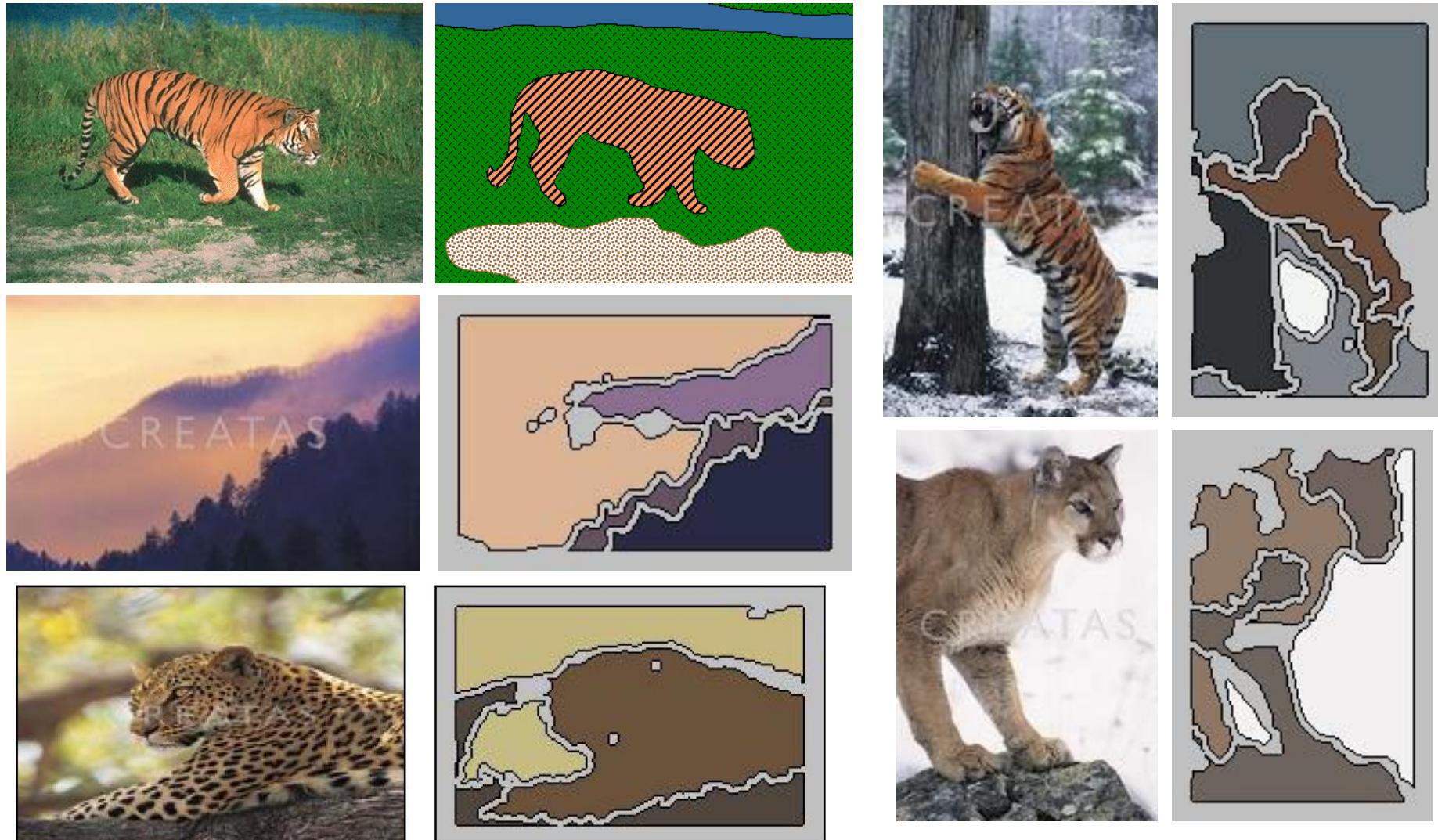


# Segmentacija bazirana na klasterovanju

- Problem: segmentacija bazirane na histogramu i klasterovanju mogu proizvesti „neuredne“ regije
  - Kako se ovo može ispraviti?
- Model Gausovih Mešavina (*Gaussian Mixture Model*, GMM) se može koristiti kao probabilistički model za klasterovanje
  - Svaki klaster se modeluje Gausovom raspodelom
  - Za svaki klaster moramo odrediti parametre Gausijana



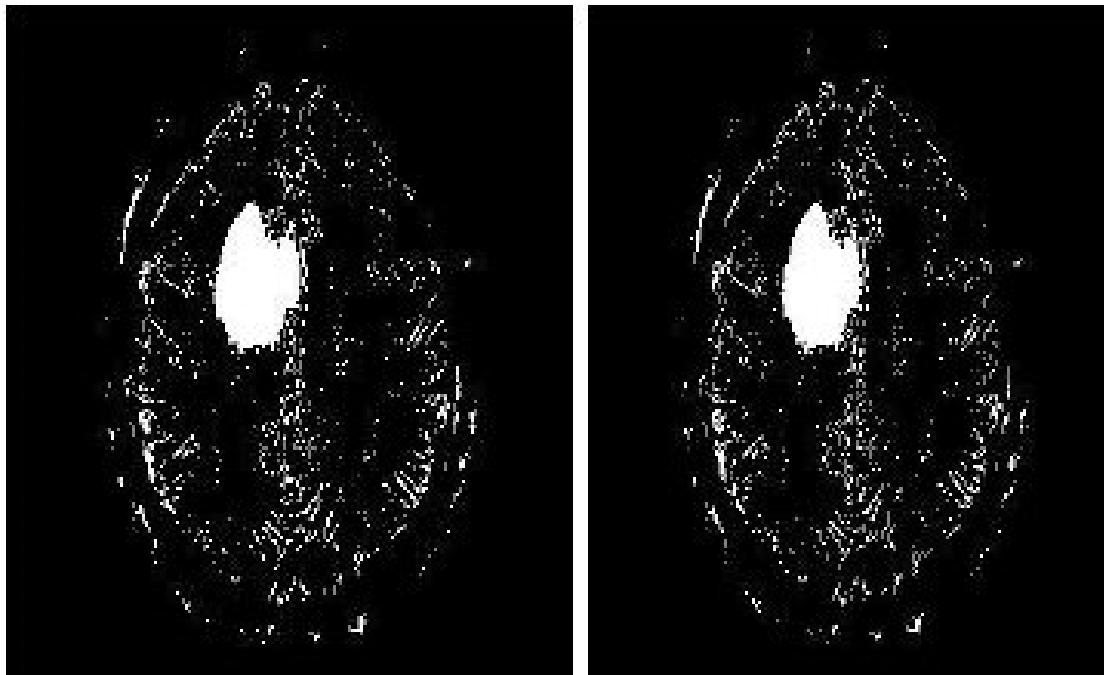
# Primeri – UC Berkeley *Blobworld*



C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: a System for Region-Based Image Indexing and Retrieval (long version)," EECS Department, University of California, Berkeley, Tech. Rep. UCB/CSD-99-1041, 1999.

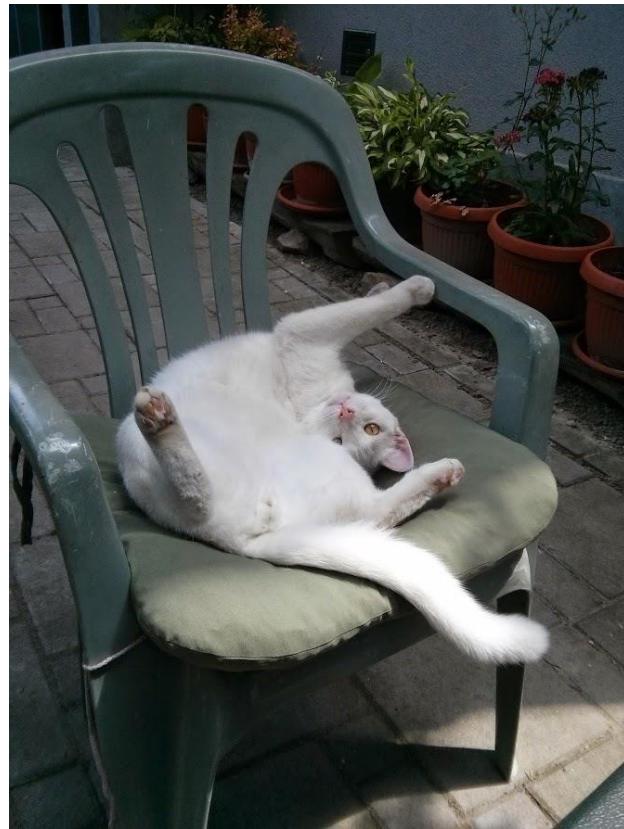
# Segmentacija pomoću rasta regija

- *Region growing*
- Ako radimo segmentaciju pragom, možemo dobiti gomilu nepovezanih regija (iznad praga)
  - Binarne slike
  - Možemo primeniti morfološke operacije da malo pročistimo rezultate



- Umesto toga bismo nekada želeli jednu povezanu regiju sastavljenu od piksela koji pripadaju određenom susedstvu
- Recimo da želimo da kliknemo na jedan pixel regije i da automatski pronađemo *blob* piksela koji su na neki način slični toj tački

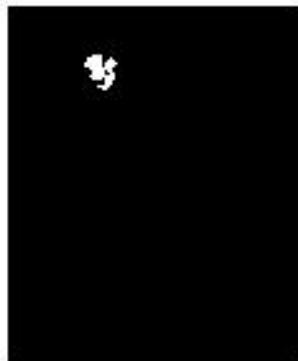
# Poređenje segmentacije: rast regionala i prag



# Segmentacija pomoću rasta regija

- Počinje od jednog piksela potencijalne regije

Iteration 5



Iteration 10



Iteration 20



- Regija se proširuje dodavanjem susednih piksela

Iteration 40



Iteration 70

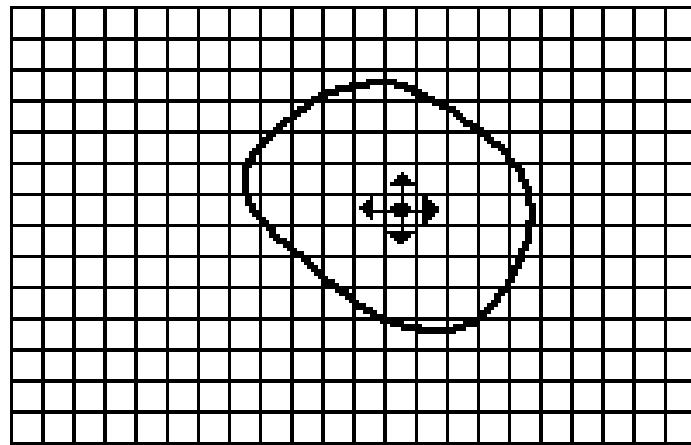


Iteration 90



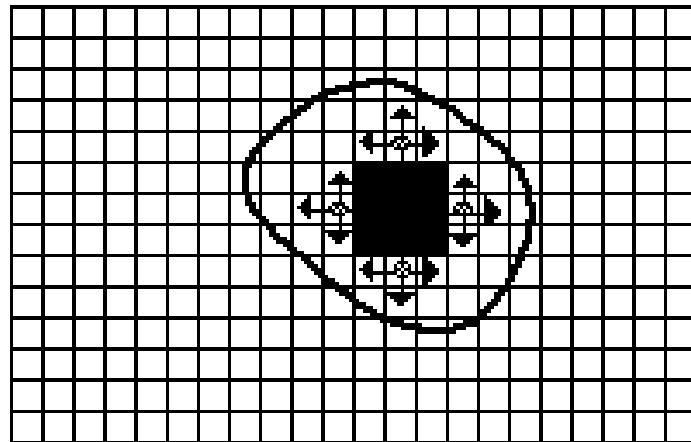
- Sve dok se potencijalni pikseli za dodavanje ne pokažu previše različitim

# Segmentacija pomoću rasta regiona



- Seed Pixel
- ↑ Direction of Growth

(a) Start of Growing a Region

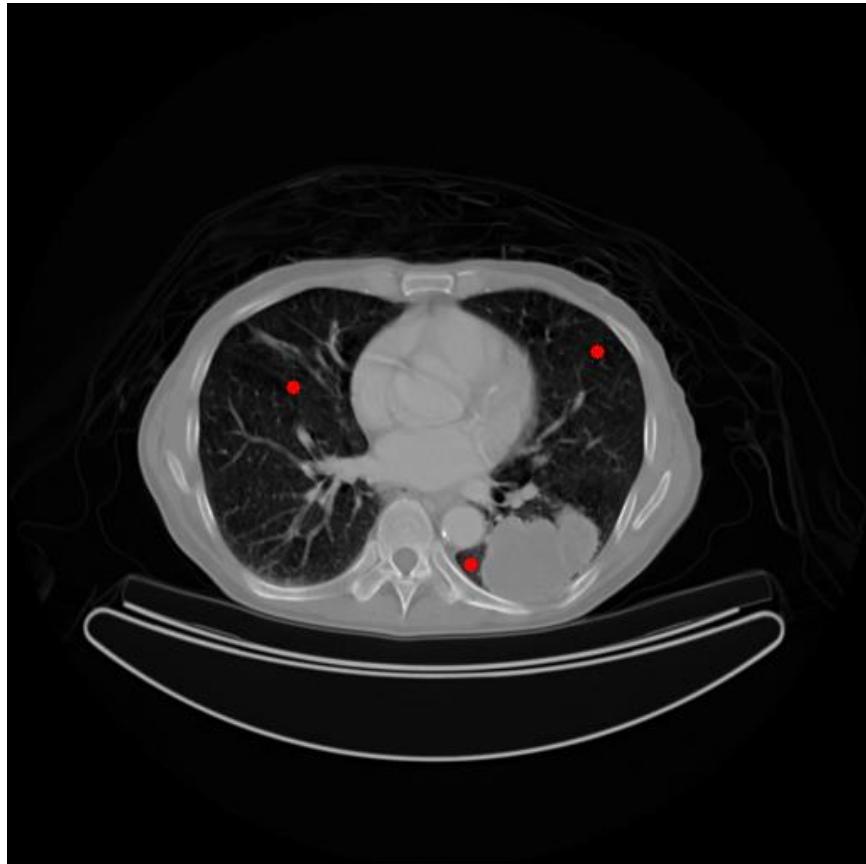


- Grown Pixels
- Pixels Being Considered

(b) Growing Process After a Few Iterations

# Biranje polaznog (seed) piksela

- Od ulazne slike pravimo binarnu „*seed image*“ sliku koja sadrži lokacije od interesa



- Ručni odabir piksela
- Automatski:
  1. Segmentiramo sliku
  2. Svaku povezanu komponentu redukujemo na jednu tačku (centroid, erozija,...)

# Kada dodati susedni piksel u regiju?

---

- Može biti kriterijum zasnovan na sličnosti intenziteta
  - Dodajemo sve tačke koje su povezane sa  $(x, y)$  (8-susedstvo) čiji je intenzitet sličan intenzitetu te tačke:  $|I(x, y) - I(x_i, y_i)| \leq T$
- Obično se koriste statistički testovi da se odredi koji pikseli se mogu dodati u regiju
  - Regija je populacija sa sličnom statistikom
  - Koristi se statistički test da se proveri da li se sused na granici regije uklapa u regiju populacije

# Segmentacija pomoću rasta regiona

- Neka je  $R$  regija koja trenutno sadrži  $N$  piksela
- Neka je  $p$  piksel susedan regiji
- Srednja vrednost  $\mu$  i varijansa  $\sigma^2$ :

$$\mu = \frac{1}{N} \sum_{(r,c) \in R} I(r, c)$$

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

# Segmentacija pomoću rasta regiona

- $T$ -statistika:

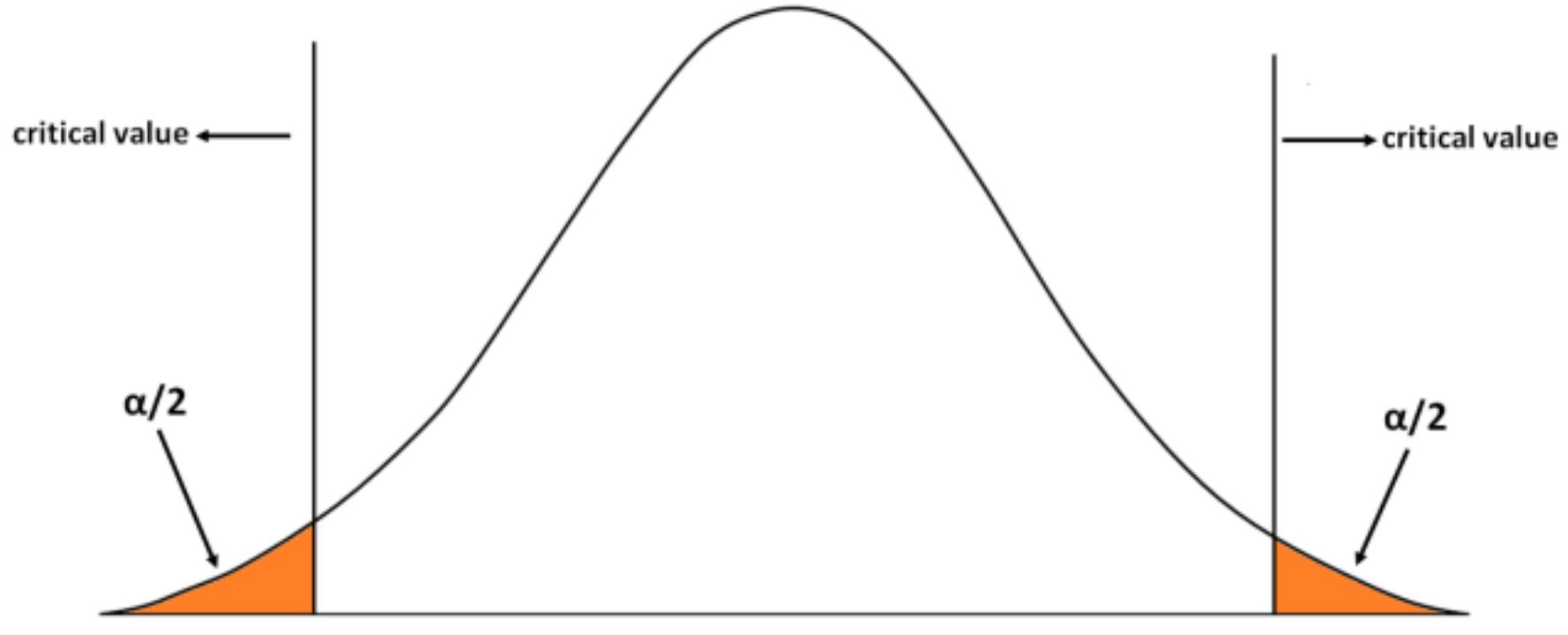
$$T = \left( \frac{(N - 1)N}{N + 1} (p - \mu)^2 / \sigma^2 \right)^{\frac{1}{2}}$$

- Ima  $T_{N-1}$  distribuciju ako su svi pikseli u  $R$  i test piksel  $p$  nezavisni i identično distribuirani (*independent and identically distributed, i.i.d.*) Gausijani

# Segmentacija pomoću rasta regiona

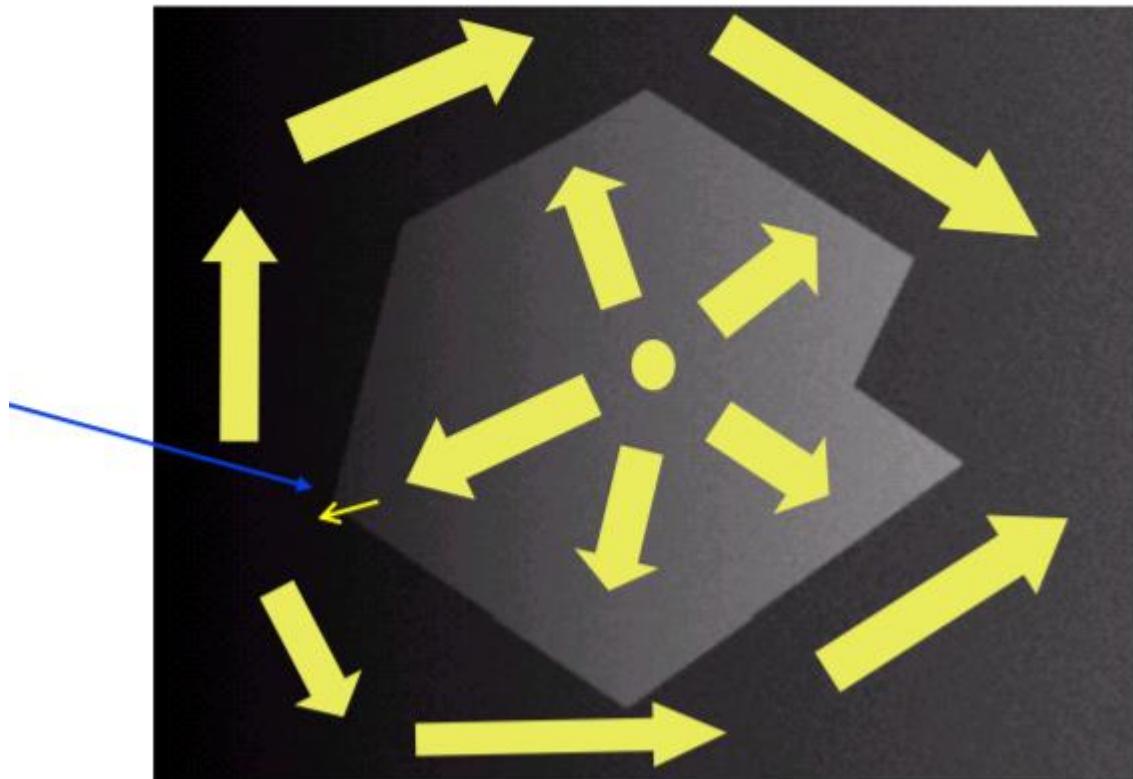
- Za  $T$  distribuciju, iz statističkih tabela možemo očitati verovatnoću  $P(T \leq t)$  za dati stepen slobode i željeni nivo pouzdanosti (*confidence level*)
  - Potrebno je odabratи orgovarajući prag  $t$
- Ako je izračunato  $T \leq t$  za željeni nivo pouzdanosti,  $p$  se dodaje u regiju  $R$  i ažuriraju se  $\mu$  i  $\sigma^2$  regije
- Ako je izračunato  $T$  preveliko, smatra se da nije verovatno da  $p$  pripada populaciji piksela u  $R$ . Započeti novu regiju

# Segmentacija pomoću rasta regiona

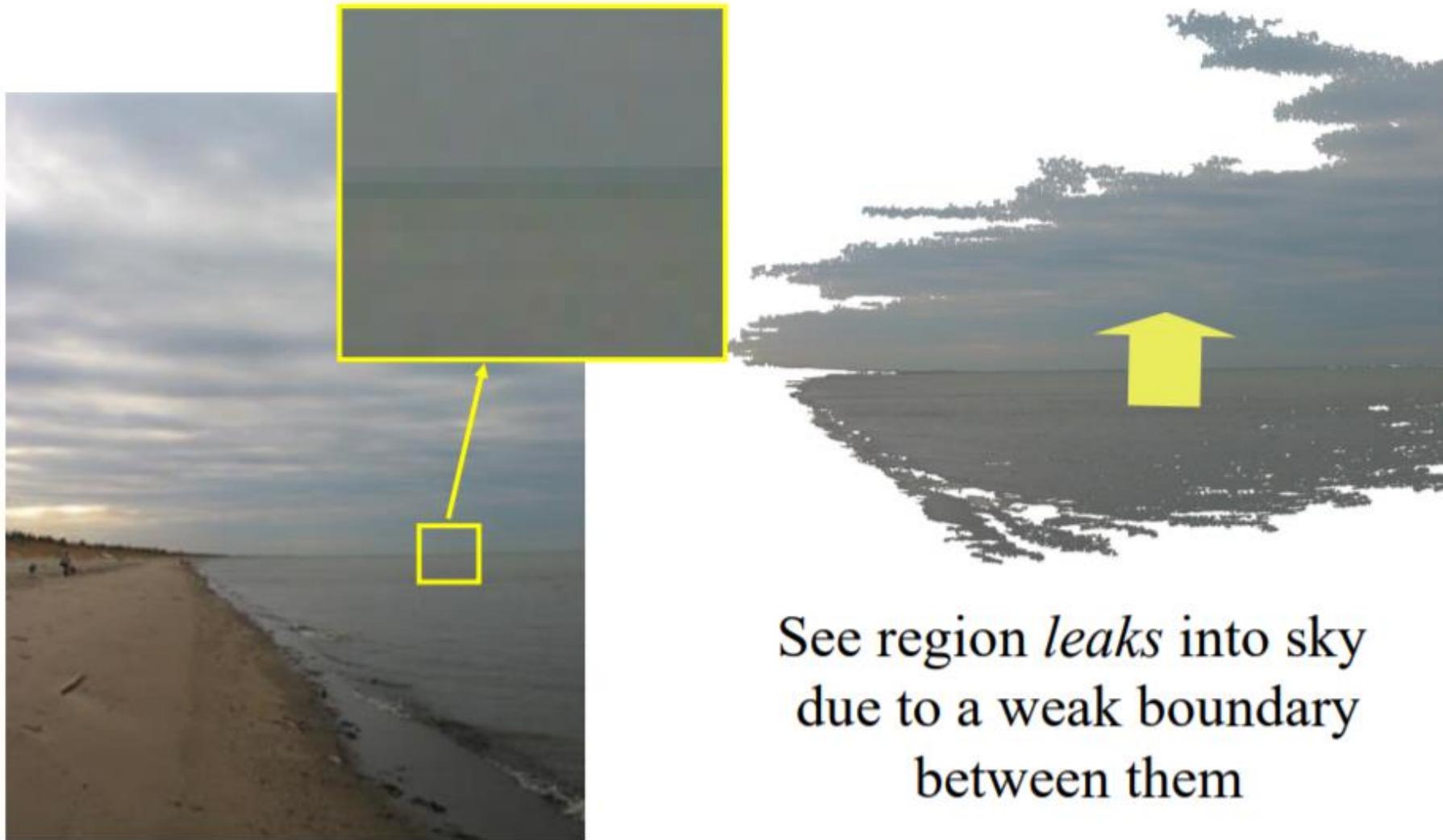


# Segmentacija pomoću rasta regiona - problem

Regija može da  
„procuri“ kroz jednu  
„slabu tačku“ granice



# Segmentacija pomoću rasta regiona - problem



# Cela slika

---

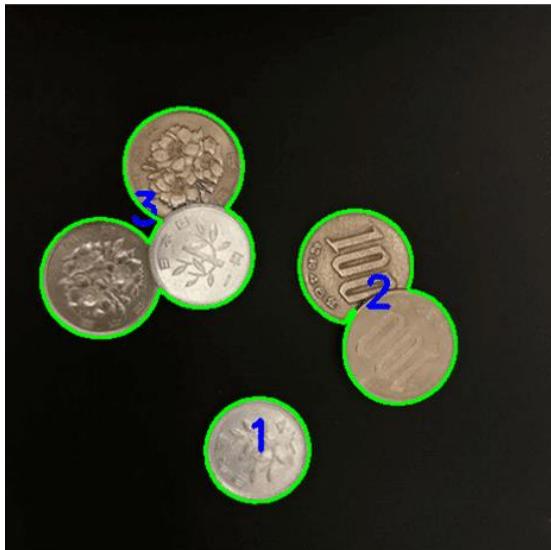
- Kada se rast jedne regije zaustavi, možemo odabrati novi *seed* piksel koji ne pripada dosadašnjim regijama i započeti ponovo
- Možemo ovaj proces ponavljati do svi pikseli ne pripadaju nekoj regiji
- Tipično rezultuje veoma dobrom segmentacijom koja dobro odgovara ivicama
- Međutim, započinjanje od određenih *seed* piksela i puštanje da regija naraste pre razmatranja drugih regija unosi pristrasnost u koristi regija koje su prve segmentirane

# Cela slika

---

- Pristrastnost ima više neželjenih efekata:
  1. Trenutna regija dominira – nejasnoće oko ivica ne moraju biti dobro razrešene
  2. Različit izbor *seed* piksela daje različite rezultate
  3. Problem ako se nasumično odabran *seed* piksel nađe na ivici
- Postoje tehnike za simultani rast regija koje rešavaju ove probleme

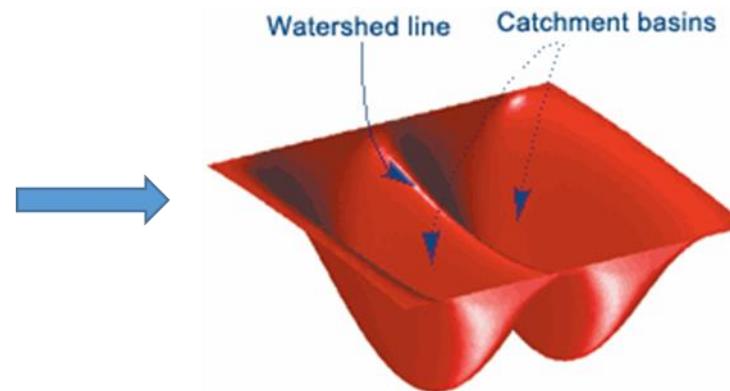
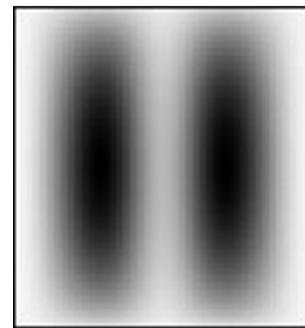
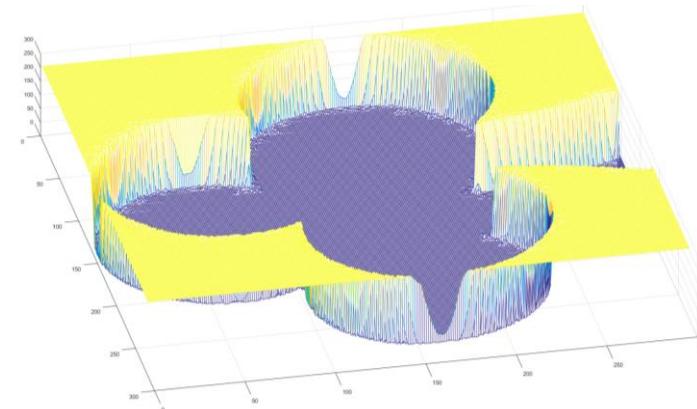
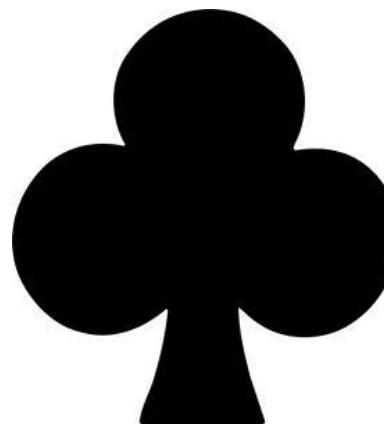
# Watershed segmentacija (razvođe)



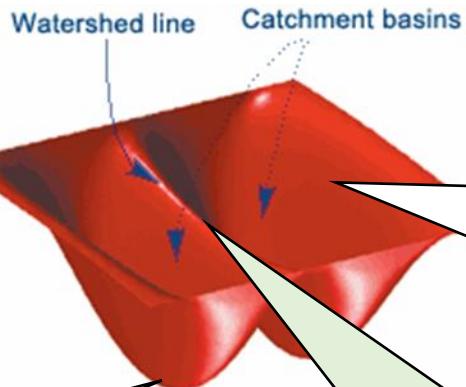
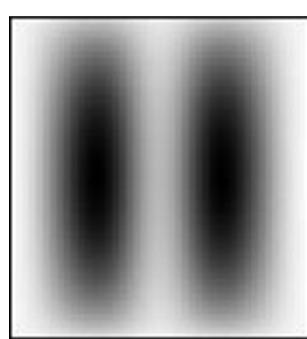
Tipično se koristi za rešavanje jedne od najtežih operacija u procesiranju slike – odvajanje sličnih objekata koji se dodiruju

# Watershed segmentacija (razvođe)

- Slika može biti interpretirana kao 3D površina sa „dolinama“ i „planinama“



# Watershed segmentacija (razvođe)

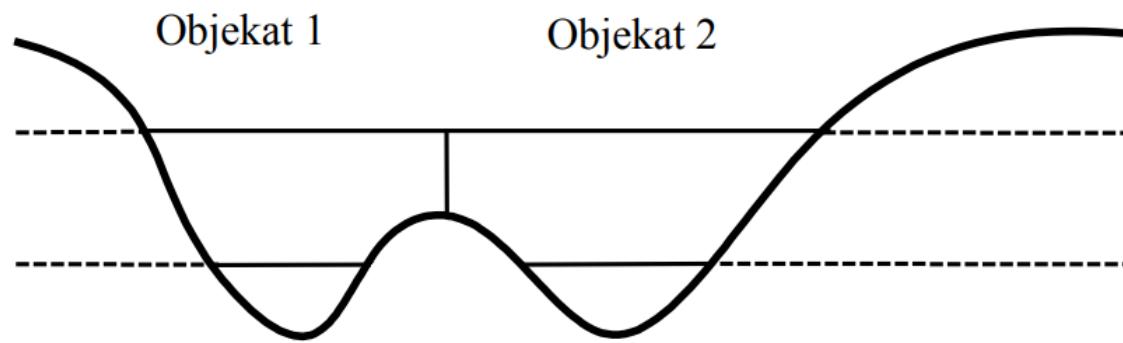


Tačke koje pripadaju minimumu regije

*Watershed lines:* tačke sa kojih kap vode ima jednaku verovatnoću da padne u više od jednog minimuma

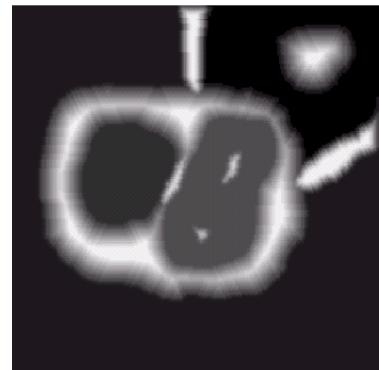
Ovo su tačke koje želimo da detektujemo – razdvajaju dva objekta reprezentovana „dolinama“

# Watershed segmentacija (razvođe)

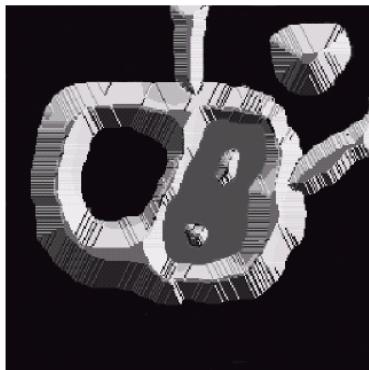


- Prepostavimo da u svakoj tački minimuma ima „rupu“ i da se površ zaroni u vodu
- Voda bi ušla kroz rupe u minimumima i poplavila površinu
- Kako bi se izbeglo da se voda koja dolazi od dva različita minimuma izmeša, pravi se „brana“ kad god bi se desilo spajanje vodenih površina
- Konačno, jedina stvar koja je vidljiva na površini bi bile ove „brane“. Ove „brane“ se zovu *watershed lines*.

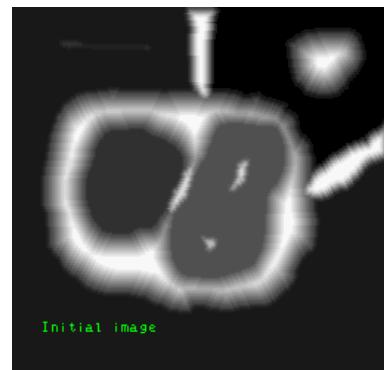
# Watershed segmentacija



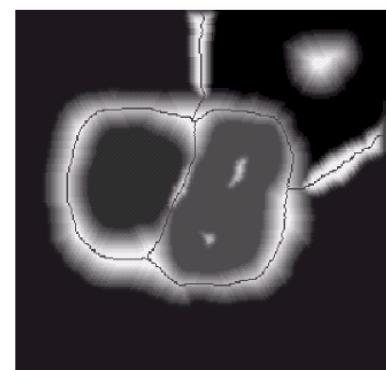
Originalna slika



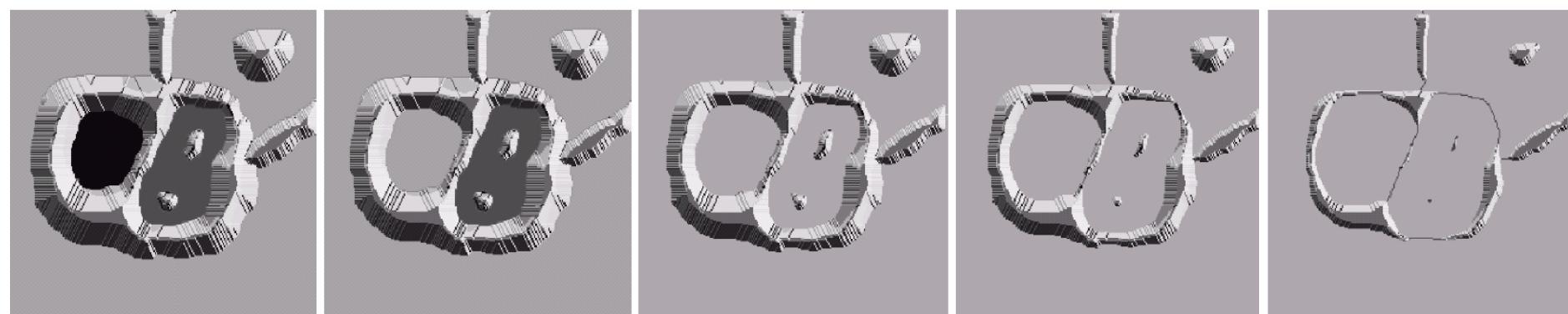
Topološki pogled



Initial image

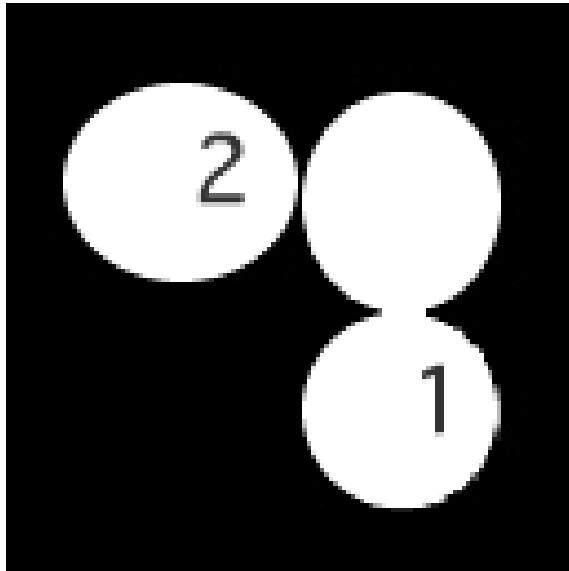


Rezultat

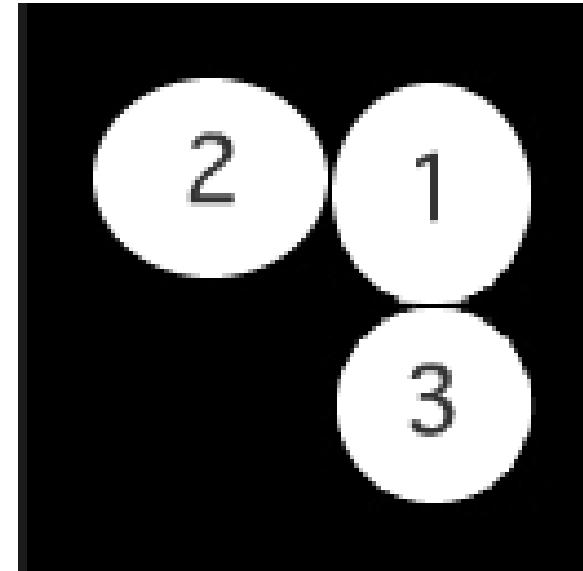


# Watershed segmentacija

Blob analysis



Watershed processing



The connected portions  
cannot be separated well



The connected portions  
can be separated

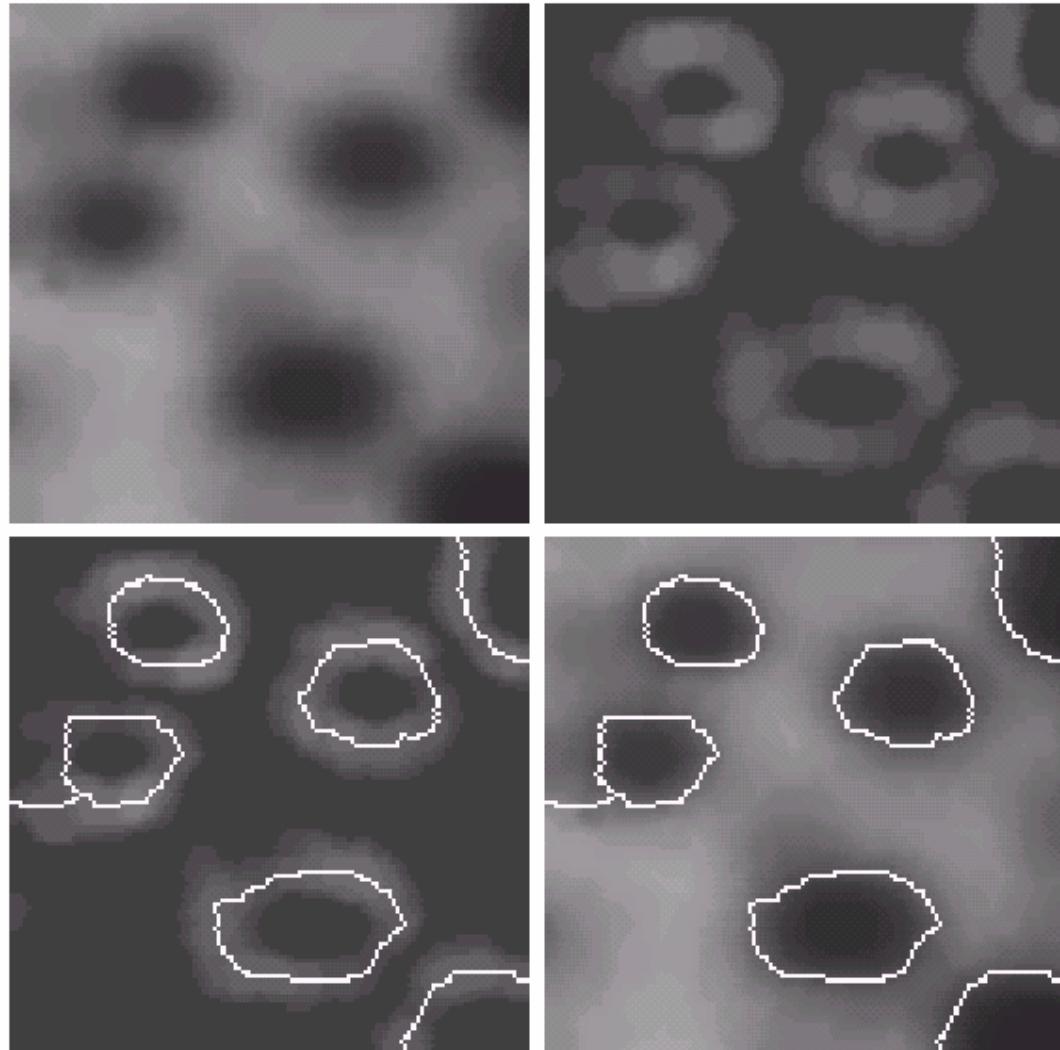
# Watershed segmentacija

a b  
c d

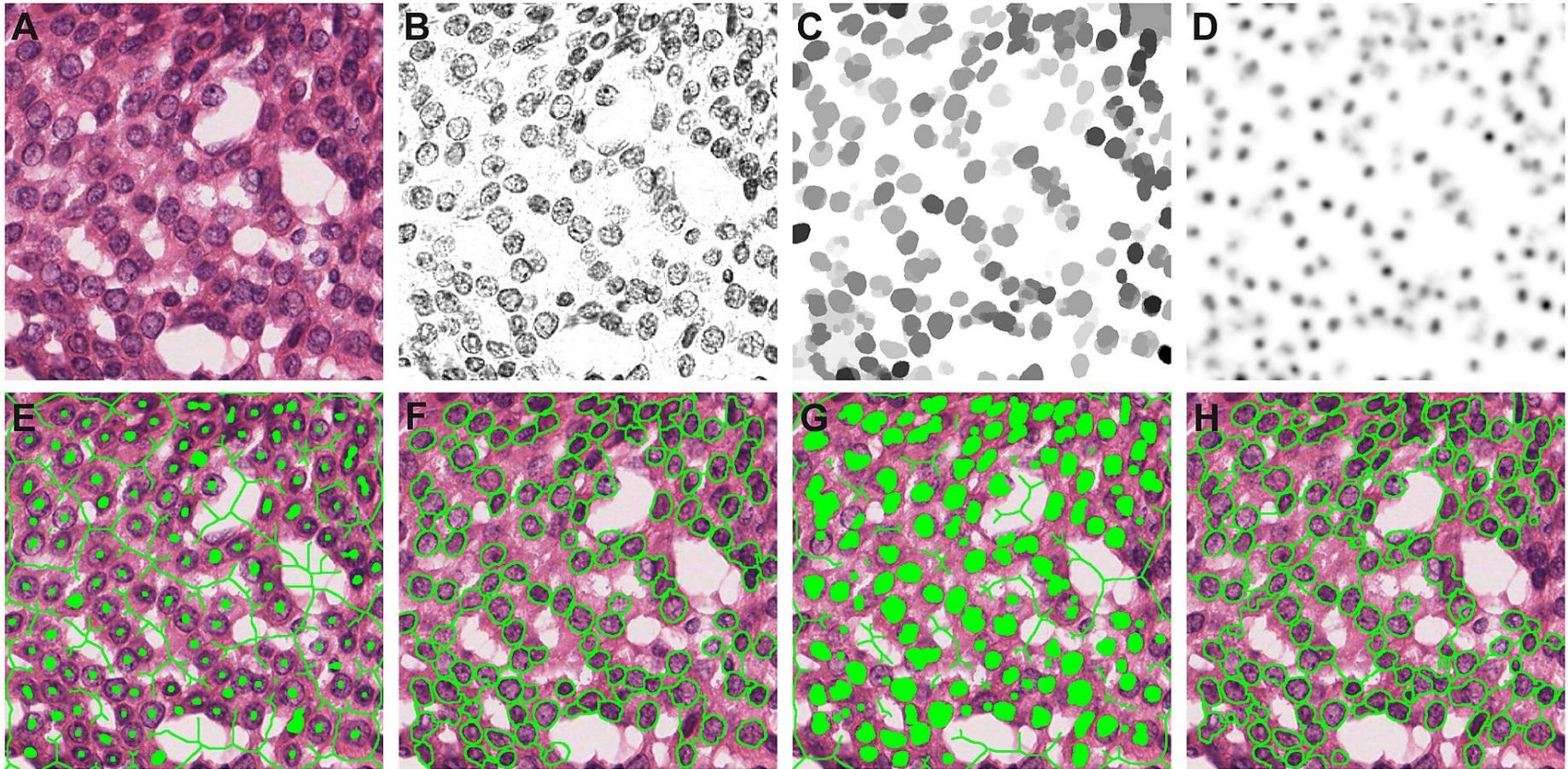
**FIGURE 10.46**

(a) Image of blobs. (b) Image gradient.  
(c) Watershed lines.  
(d) Watershed lines superimposed on original image.  
(Courtesy of Dr. S. Beucher,  
CMM/Ecole des Mines de Paris.)

- Jedna od glavnih primena je u ekstrakciji gotovo uniformnih (*bloblike*) objekata od pozadine
- Ivice dobijene segmentacijom su zatvorene konture

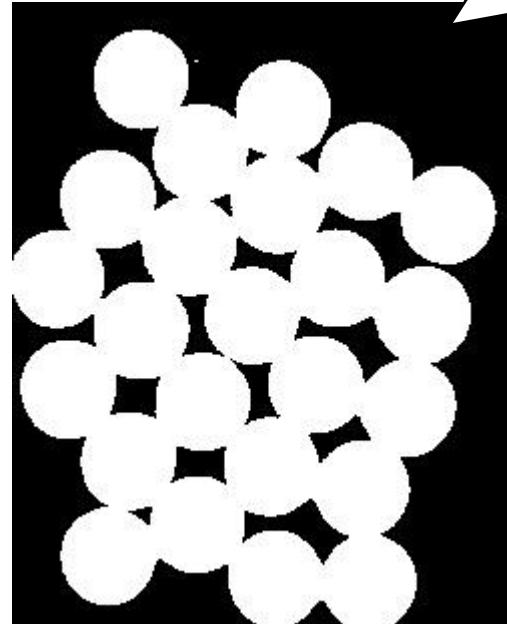


# Watershed segmentacija



A) Original image. B) Hematoxylin channel. C) Pre-processed image (hematoxylin channel processed with series of morphological operations). D) Fast radial symmetry transform (FRST). E) FRST foreground and background markers. F) Watershed segmentation with FRST markers. G) Regional minima foreground and background markers. H) Watershed segmentation with regional minima markers

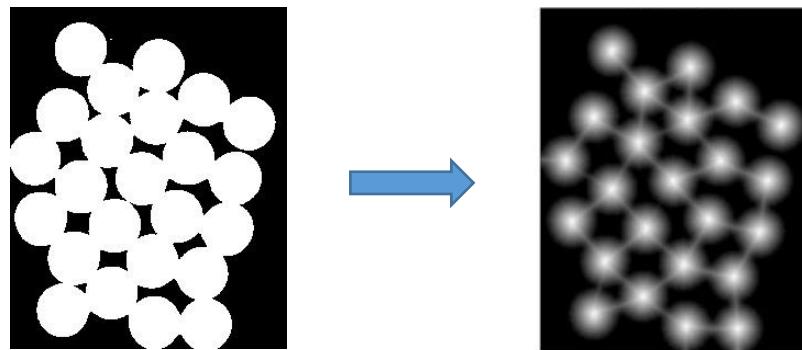
# Watershed segmentacija



Otsu, Opening, Closing

# Watershed segmentacija

- Moramo izmeniti sliku u drugu sliku čiji *catchment basins* predstavljaju objekte koji trebaju biti identifikovani



- Gradijent može biti korišćen za *grayscale* slike. *Catchment basins* bi u teoriji odgovarali homogenim regijama određenog nivoa sive na slici
- *Distance transform* može biti korišćena na binarnim slikama
  - Ideja je da se kreiraju granice koje su najdalje moguće od centara objekata koji se preklapaju
  - *catchment basins* bi odgovarali objektima od interesa (*foreground*)

# Distance transform

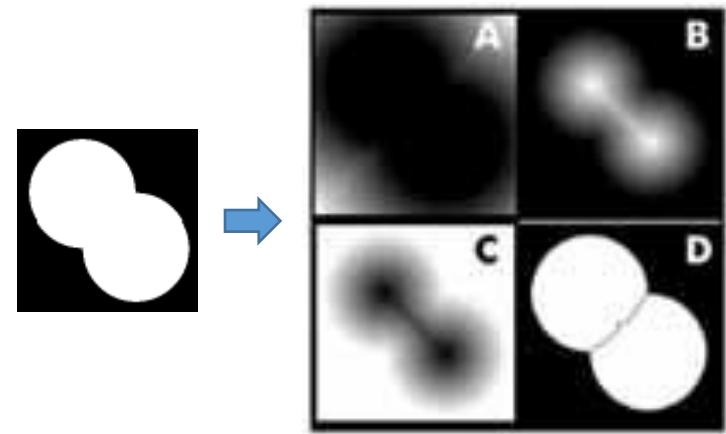
- Rastojanje svakog piksela do najbližeg ne-nula piksela

0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

Binary image matrix

0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	0	0	1	2	1	0	0	0
0	1	1	1.41	1	1	1	1	0
0	0	0	1	2	2	2	1	0
0	0	0	1	2	3	2	1	0
0	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

Distance transform matrix

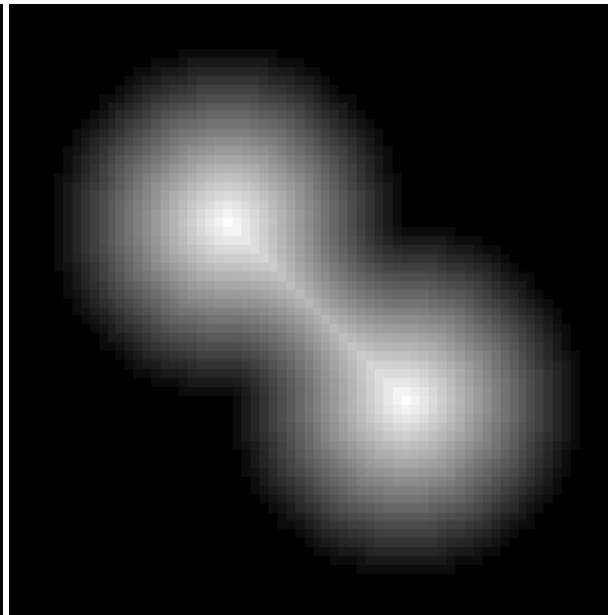


- A: distance transform
- B: A nije korisno jer imamo jedan catchment basin preko cele slike – napravimo komplement
- C: invertujemo da napravimo catchment basins
- D: Primjenjen *watershed*

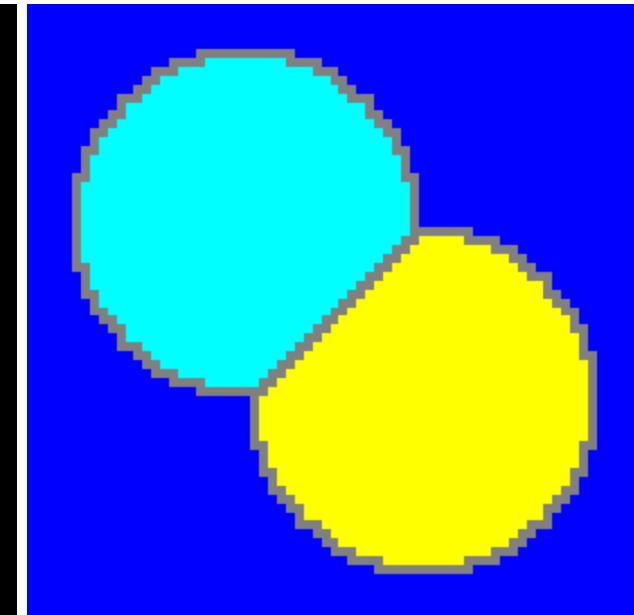
# Watershed segmentacija



Binarna slika



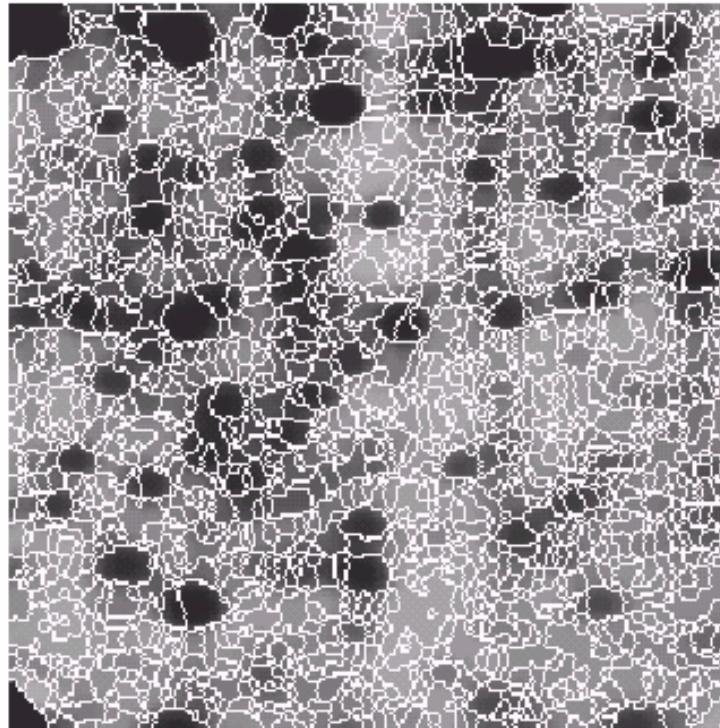
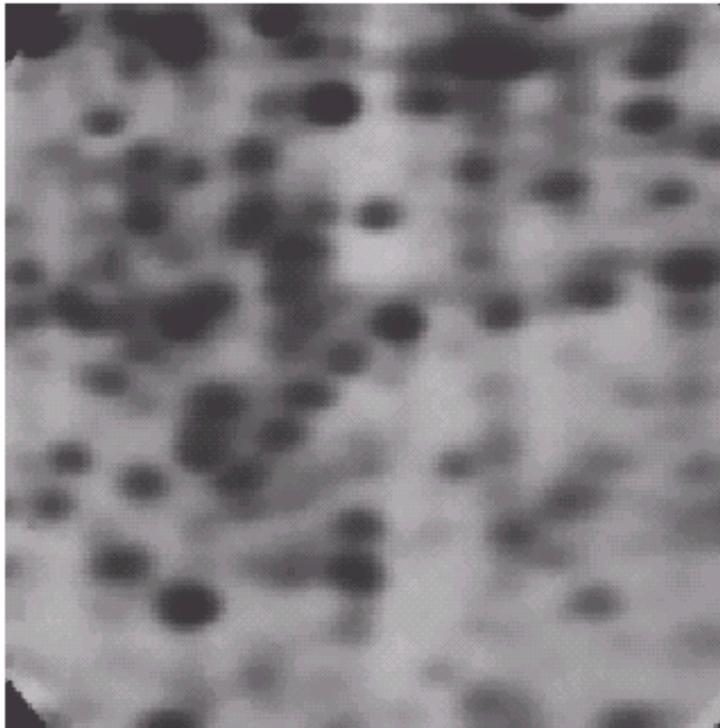
*Distance transform*  
komplementa binarne  
slike



*Watershed transform*  
primenjena na rezultat  
*distance transform*

# Problem

- Slika može biti prekomerno segmentirana usled šuma (veliki broj minimuma od kojih nisu svi relevantni)



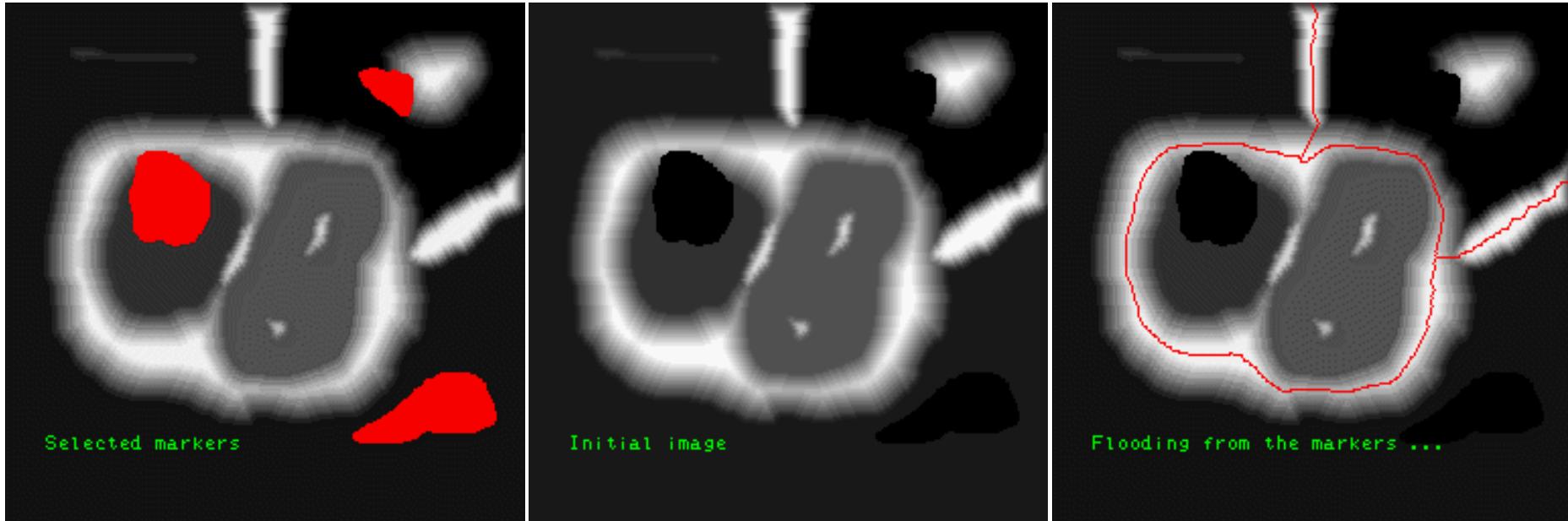
a b

**FIGURE 10.47**

(a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident.  
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

- Praktično rešenje: ograničiti broj dozvoljenih regija

# Watershed segmentacija



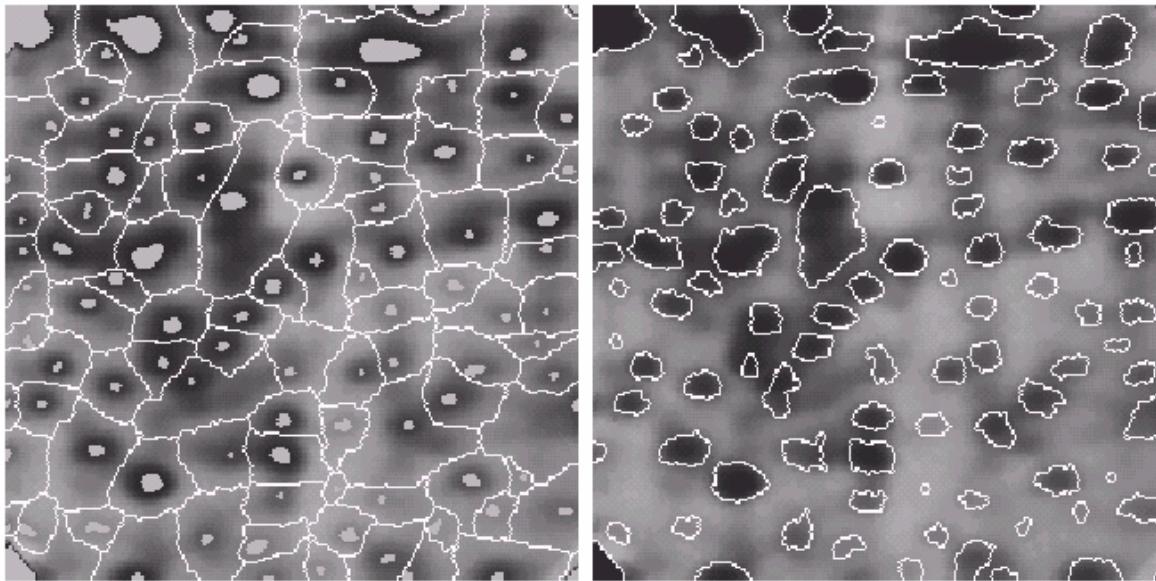
- Način da kontrolišemo prekomernu segmentaciju je da koristimo predefinisan skup markera kako bismo specificirali samo dozvoljene regionalne minimume (ubacivanje domenskog znanja)
- Trebaju nam:
  - Interni markeri (*Internal markers*) – unutar objekata od interesa
  - Eksterni markeri (*External markers*) – nalaze se na pozadini

# Watershed segmentacija

---

- Interni markeri
  - Svaki odgovara jednom objektu
  - Okruženi su tačkama veće „nadmorske visine“
- Eksterni markeri
  - Dele slike na regije gde je dozvoljeno da se nalaze lokalni minimumi
- Ovi markeri uvode ograničenja u proces segmentacije
  - U literaturi ima mnogo preporuka kako odabratи markere
  - Od jednostavnih algoritama baziranih na intenzitetu do kompleksnijih: boja, oblik, lokacija, tekstura,...
  - Uvodimo domensko znanje u proces segmentacije

# Watershed segmentacija

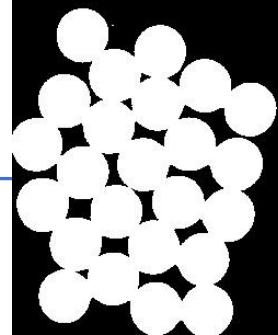


a b

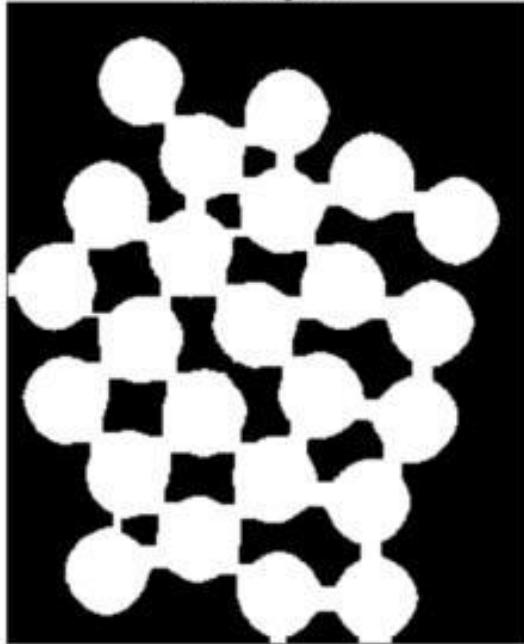
**FIGURE 10.48**  
(a) Image showing internal markers (light gray regions) and external markers (watershed lines).  
(b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

- Pronalazak internih markera: *Gaussian smoothing* + primena praga i (opciono) morfološke operacije
- Na datu sliku se primeni *watershed segmentacija* – dobijene tačke se proglose eksternim markerima
- Ove linije efikasno dele sliku na regije gde svaka ima jedan interni marker i deo pozadine
- Problem se redukuje na to da se svaka regija podeli na dva dela – može se primeniti *watershed* ili neka druga tehnika

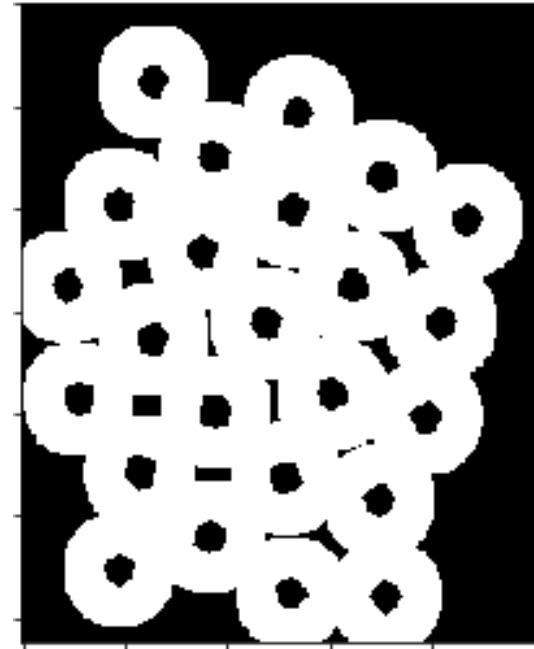
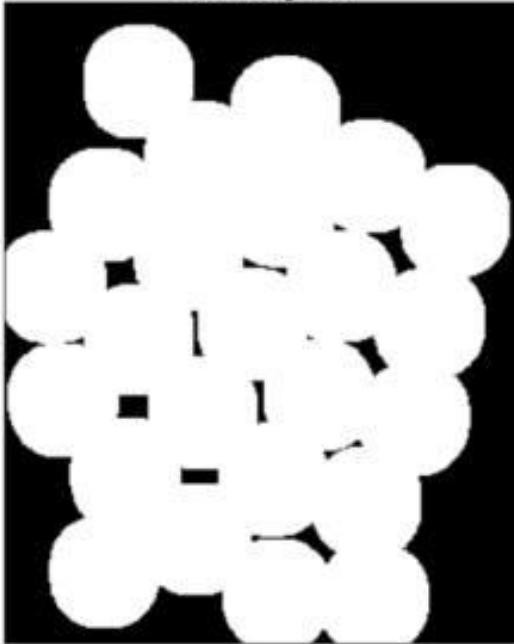
# Watershed segmentacija



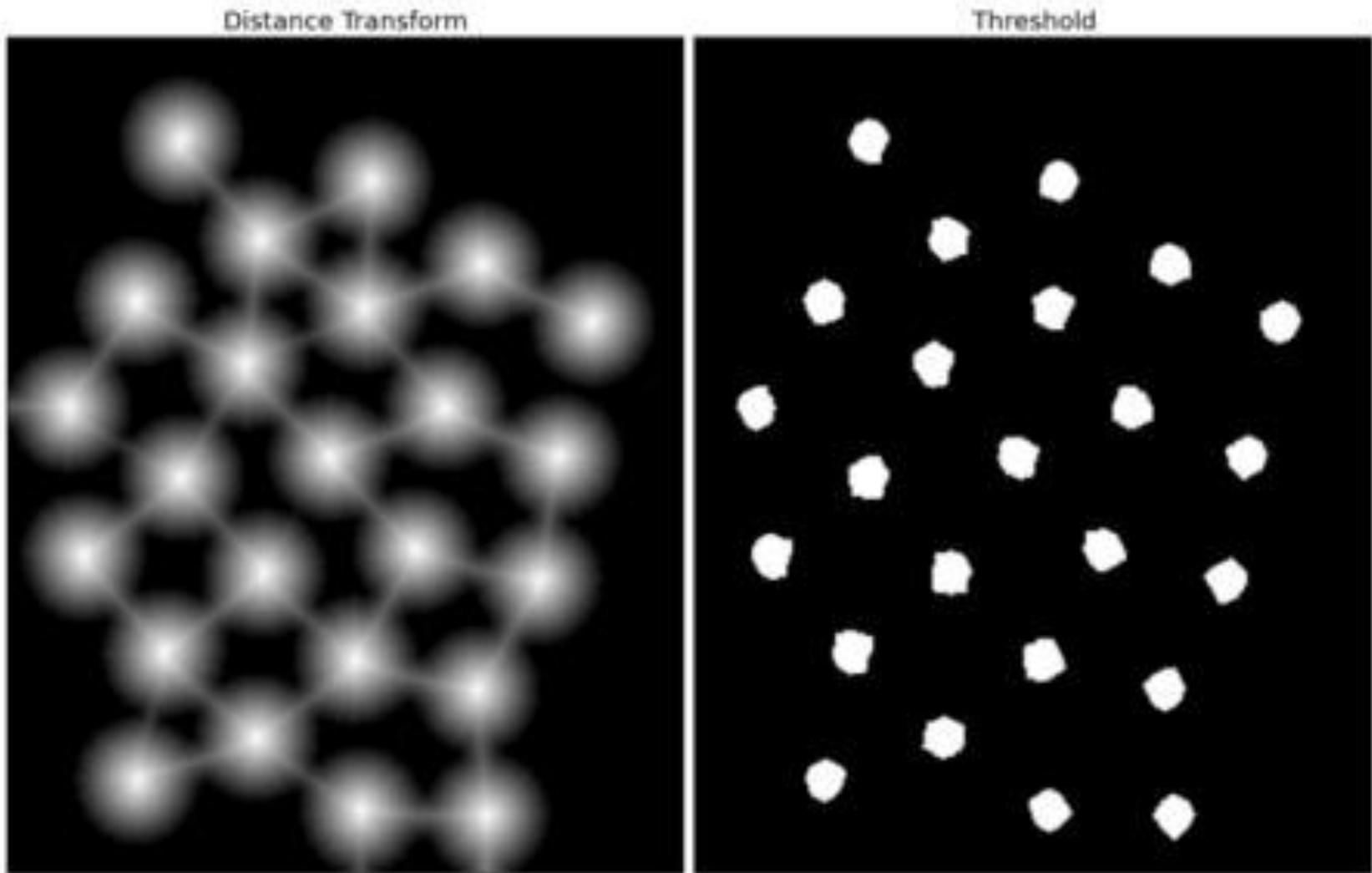
White Region shows  
Sure Foreground



Black Region shows  
Sure Background

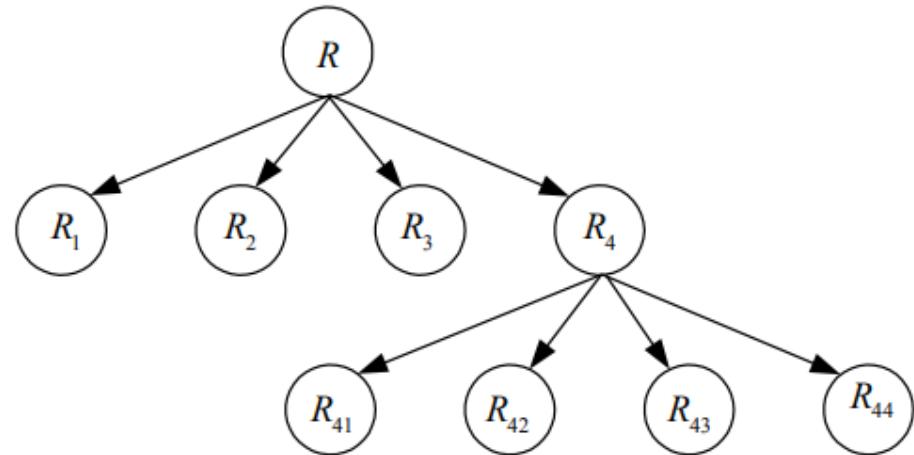
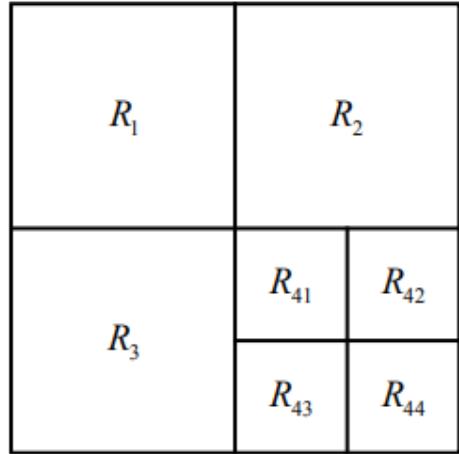


# Watershed segmentacija



# Segmentacija pomoću razdvajanja i spajanja regionala (*split-and-merge*)

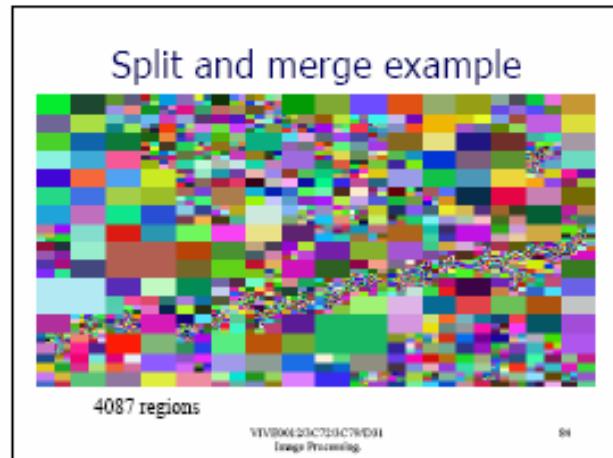
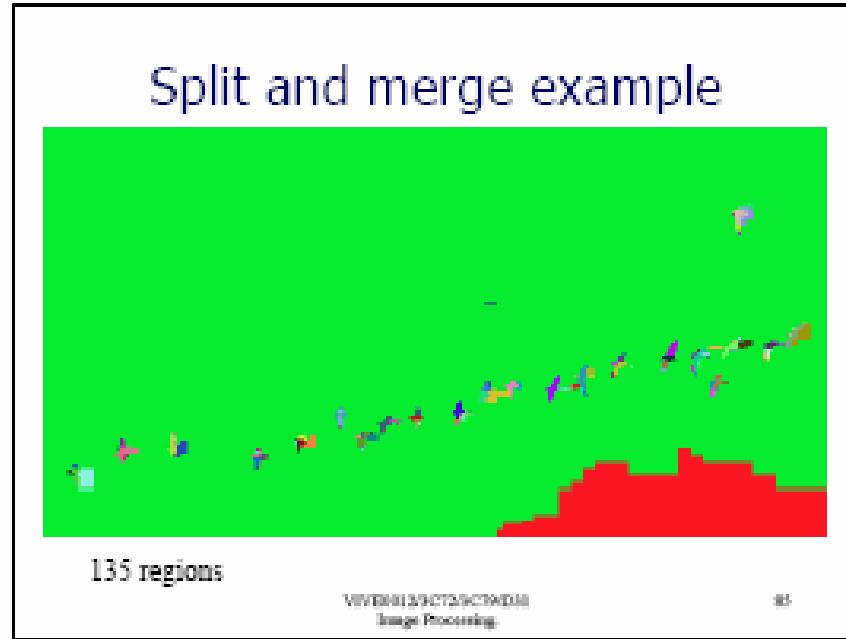
1. Za početnu regiju uzeti celu sliku
2. Ako regija nije uniformna, podeliti je na 4 kvadranta
3. Spojiti sve susedne dovoljno slične regije (isto uniformno obeležje)
4. Ponavljati korake 2-3 iterativno, sve dok nema više podele ili spajanja



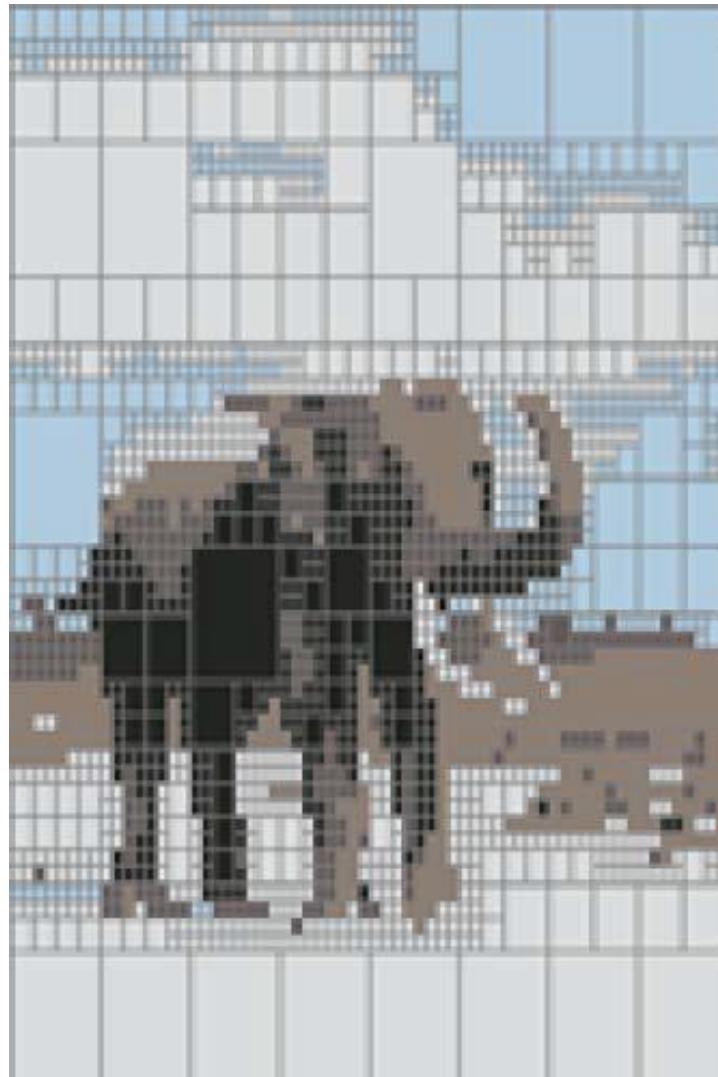
# Segmentacija pomoću razdvajanja i spajanja regionala (*split-and-merge*)

- Moramo definisati:
  1. uniformnost/sličnost
    - Možemo uraditi na različite načine
    - Npr. razlika najveće i najmanje osvetljenosti piksela u regionu
    - Npr. varijansa osvetljenosti (ili druga statistička mera)
  2. Prag
    - Kada je regija dovoljno uniforma da je ne delimo na regije
    - Kada su regije dovoljno slične da možemo da ih spojimo
  3. Kriterijum zaustavljanja
    - Kada nema više deljenja/spajanja
    - Definisati najmanju veličinu bloka
- Nedostatak ove metode je blokovska struktura regionala u segmentiranoj slici

# Segmentacija pomoću razdvajanja i spajanja regionala (split-and-merge)



# Segmentacija pomoću razdvajanja i spajanja regionala (*split-and-merge*)



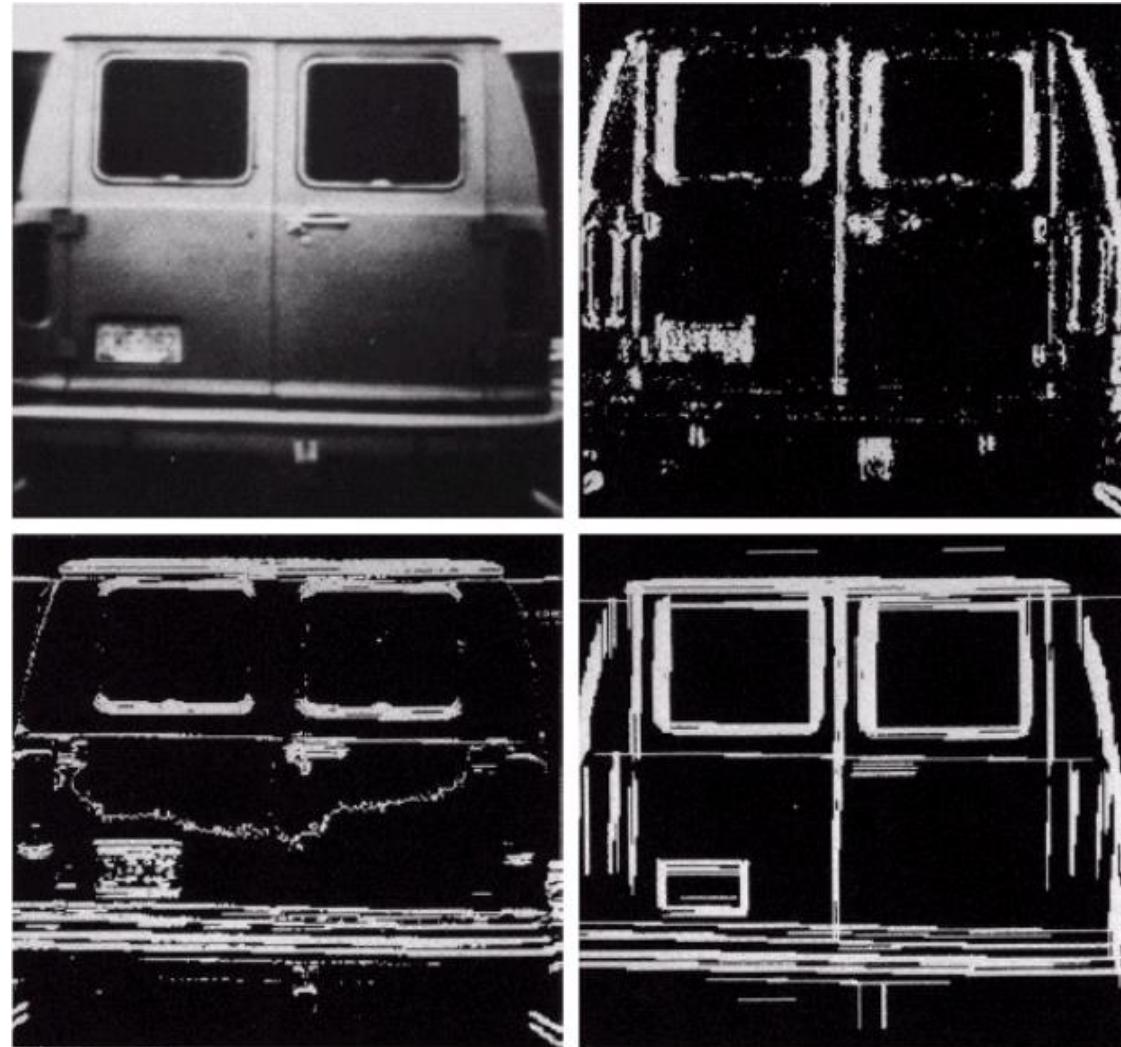
# Segmentacija pomoću granica regiona

- Metode izdvajanja ivica generišu isprekidane granice objekata, a ne zatvorene krive
  - Posledica prisustva šuma, neuniformnog osvetljenja i drugih uticaja
- Za segmentaciju, moramo spojiti ivice u zatvorene konture
- Možemo koristiti
  - Lokalno povezivanje: analiza okoline pikela koji je označen nekom od metoda za detekciju ivica kao ivična tačka
  - Metode fitovanja polinomima ili pravolinijskim segmentima
  - Globana analiza: npr. Hough transformacija – tačke se povezuju ako pripadaju istoj krivoj određenog oblika



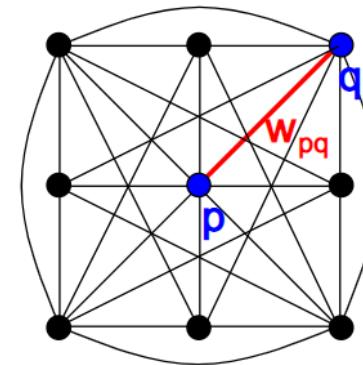
# Lokalno povezivanje ivice

- Cilj: izdvojiti pravougaone objekte koji su kandidati za tablicu
- Gore desno i dole levo:  $G_x, G_y$  (Sobel)
- Povezivanje ivičnih regiona čija je magnituda gradijenta veća od 25, a ugao gradijenta se ne razlikuje za više od  $15^\circ$
- Prepoznavanjem odnosa stranica, lako možemo izdvojiti tablicu od ostalih kandidata

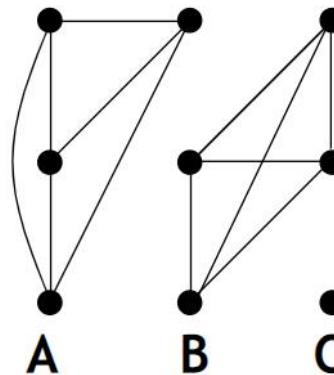


# Segmentacija slike bazirana na grafovima

- Slika je predstavljena grafom
  - Čvorovi odgovaraju pikselima (ili superpikselima)
  - Potpuno povezan graf
  - Grana između čvorova  $p$  i  $q$  ima težinu  $w_{pq}$  koja odgovara sličnosti čvorova



- Cilj je particonisanje čvorova u odvojene klastere
  - Sličnost čvorova unutar svakog klastera treba da bude velika, a sličnost čvorova različitih klastera treba da bude mala
  - Podelu ćemo uraditi brisanjem grana između sličnih segmenata



# Segmentacija slike bazirana na grafovima

- Primer sličnosti čvorova:

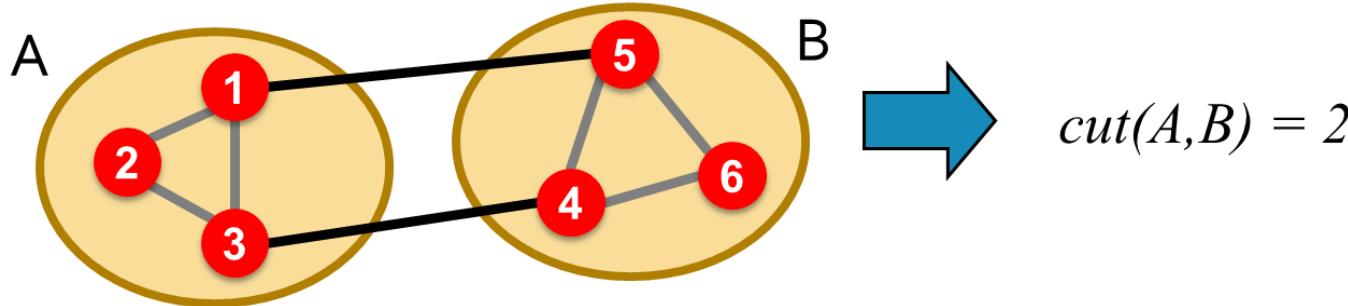
$$w_{pq} = \frac{1}{dist(p, q)} e^{-\frac{1}{2\sigma^2} \|I_p - I_q\|^2}$$

- Ako je  $I_p \approx I_q \rightarrow e^{-\frac{1}{2\sigma^2} \|I_p - I_q\|^2} \approx e^0 = 1$
- Ako je  $I_p \neq I_q \rightarrow e^{-\frac{1}{2\sigma^2} \|I_p - I_q\|^2} \approx e^{-\infty} = 0$

# Segmentacija slike bazirana na grafovima

- Graf se može razbiti na dva odvojena skupa čvorova  $A$  i  $B$  uklanjanjem grana koje povezuju ove skupove čvorova

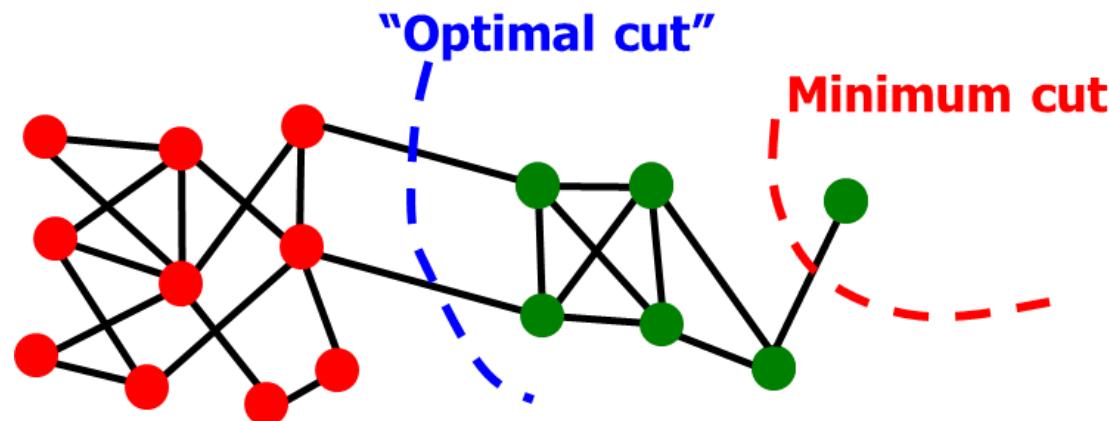
$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$



- Jedan način da se razbije graf jeste da se minimizuje  $cut$

# Segmentacija slike bazirana na grafovima

- Intuitivno rešenje za pronalaženje zajednica: preseci najmanji mogući broj grana
  - Kvalitet klastera je težina konekcija koje vode izvan klastera
- Ne radi uvek... Problem:
  - Favorizuje odsecanje malih grupa čvorova zato što uzima u obzir samo grane koje izlaze iz klastera
  - Ne uzima u obzir grane unutar klastera



# Segmentacija slike bazirana na grafovima

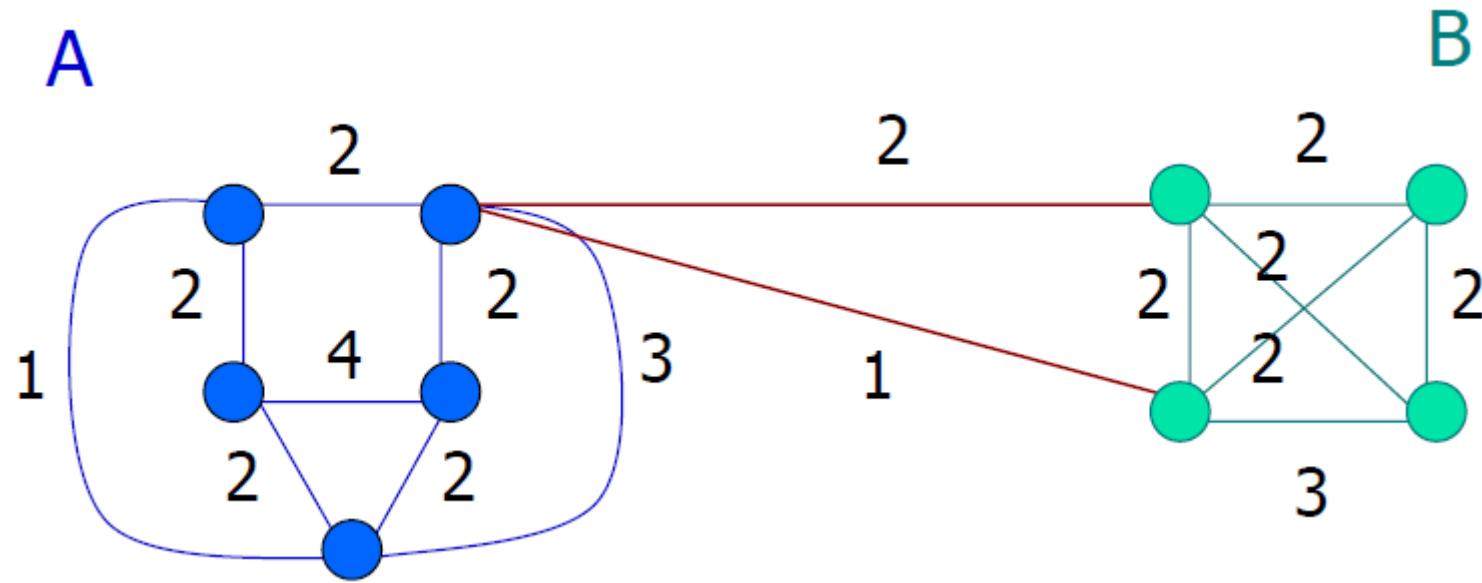
- Shi i Malik su predložili *normalized cut*:

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

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

$assoc(A, V)$  – koliko je skup čvorova  $A$  povezan sa grafom kao celinom (ukupna težina grana sa barem jednim čvorom u  $A$ )

# Segmentacija slike bazirana na grafovima



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