Lecture 18-2 Region-based Segmentation（chapter 10.4-10.6 ）

Yuyao Zhang, Xiran Cai PhD

zhangyy8@shanghaitech.edu.cn caixr@shanghaitech.edu.cn

SIST Building 2 302-F/302-C

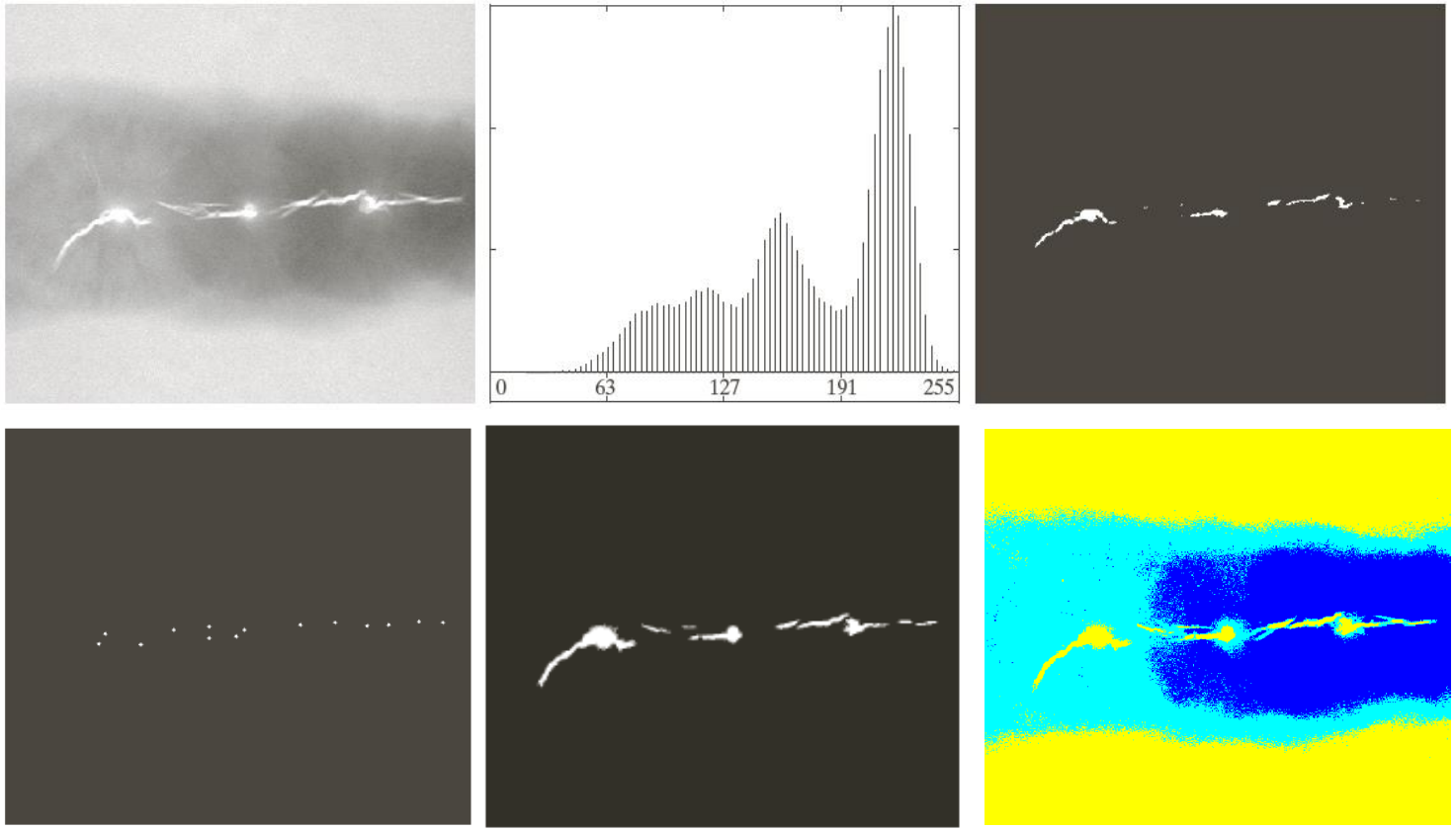
Course piazza link:
piazza.com/shanghaitech.edu.cn/spring2021/cs270spring2021

Outline

- Region-based Segmentation
  - Region growing
  - Region spilt and merge
  - Clustering and Super-pixel

Basic region growing

- If we only want “common” pixels near one point.
  1) from input image $I(x, y)$ get a binary “seed image” $S(x, y)$ for locations of interest. (e.g. by thresholding).
  2) reduce seed connected components down to single point each.
  3) let $T(x, y) = 1$ if $I(x, y)$ satisfies some predicate/condition and 0 else. (e.g. (x, y) is 8-connected to seed point (x_i, y_i) and $|I(x, y) -$



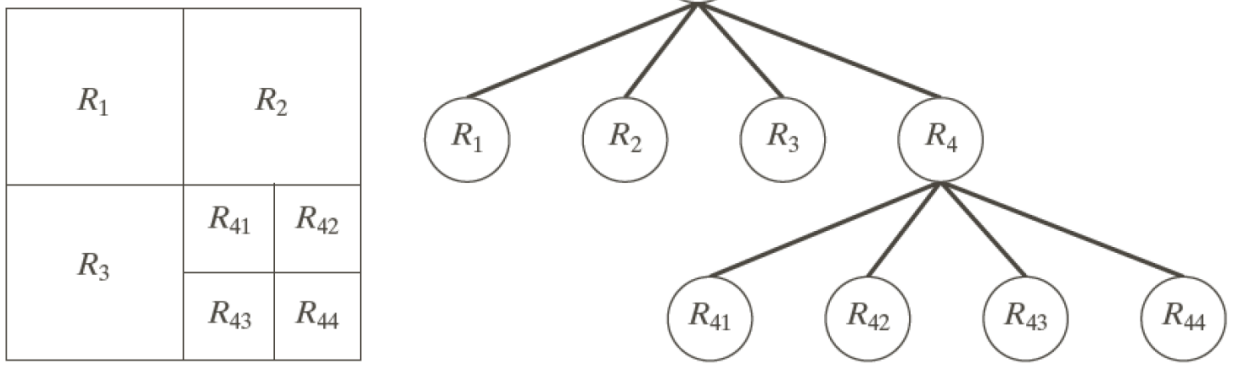
Try this

- Check the “regiongrow.m” function in discussion folder.

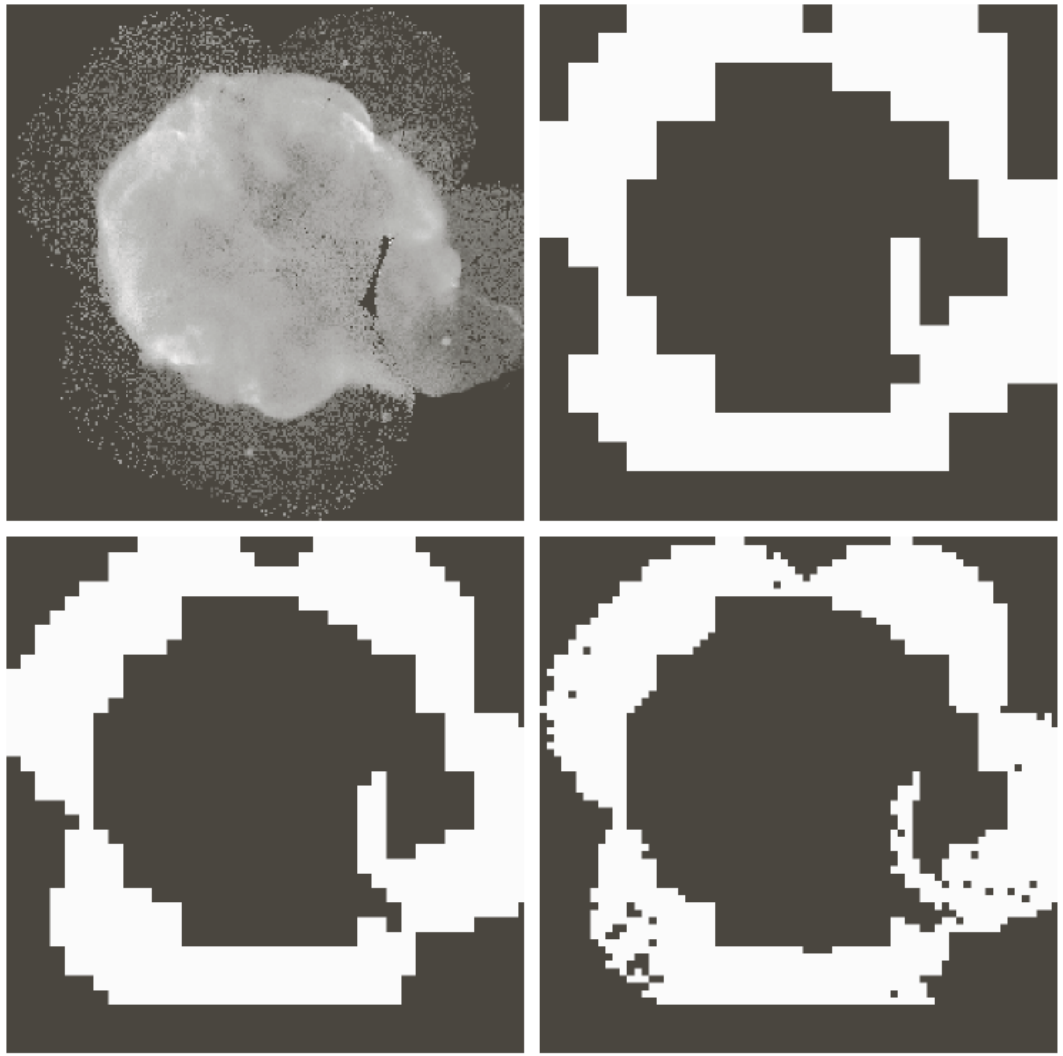
Region split and merge

➢ Steps

1. Split into four disjoint quadrants any region R_i for which $Q(R_i) = False$ (need to specify a minimum quadregion size beyond which no further splitting is carried out;
2. Merge any adjacent regions R_j and R_k for which $Q(R_j \cup R_k) = True$;
3. Stop when no further merging is possible.



Region Splitting and Merging



K-means Clustering

➢ Algorithm:

1) Specify an initial set of clusters centers $m_1, m_2, \cdots, m_k \in \mathcal{R}^n$.

2) For each $x_i \in \mathcal{R}^n$ in dataset, assign it to closest cluster

$$x_i \in cluster_j \text{ if } \|x_i - m_j\| < \|x_i - m_k\| \quad (k \neq j)$$

3) Update the mean m_j = average value of all x in cluster j.

$$m_j = \frac{1}{c_j} \cdot \sum_{x \in c_j} x \quad j = 1,2, \cdots, k$$

4) Keeps altering 2) and 3) until stop changing.

E.g. Image histogram

E.g. Color space k-means cluster.

K-means Clustering

➢ Algorithm:

1) Specify an initial set of clusters centers $m_1, m_2, \cdots, m_k \in \mathcal{R}^n$.

2) For each $x_i \in \mathcal{R}^n$ in dataset, assign it to closest cluster

$$x_i \in cluster_j \text{ if } \|x_i - m_j\| < \|x_i - m_k\| \quad (k \neq j)$$

3) Update the mean m_j = average value of all x in cluster j.

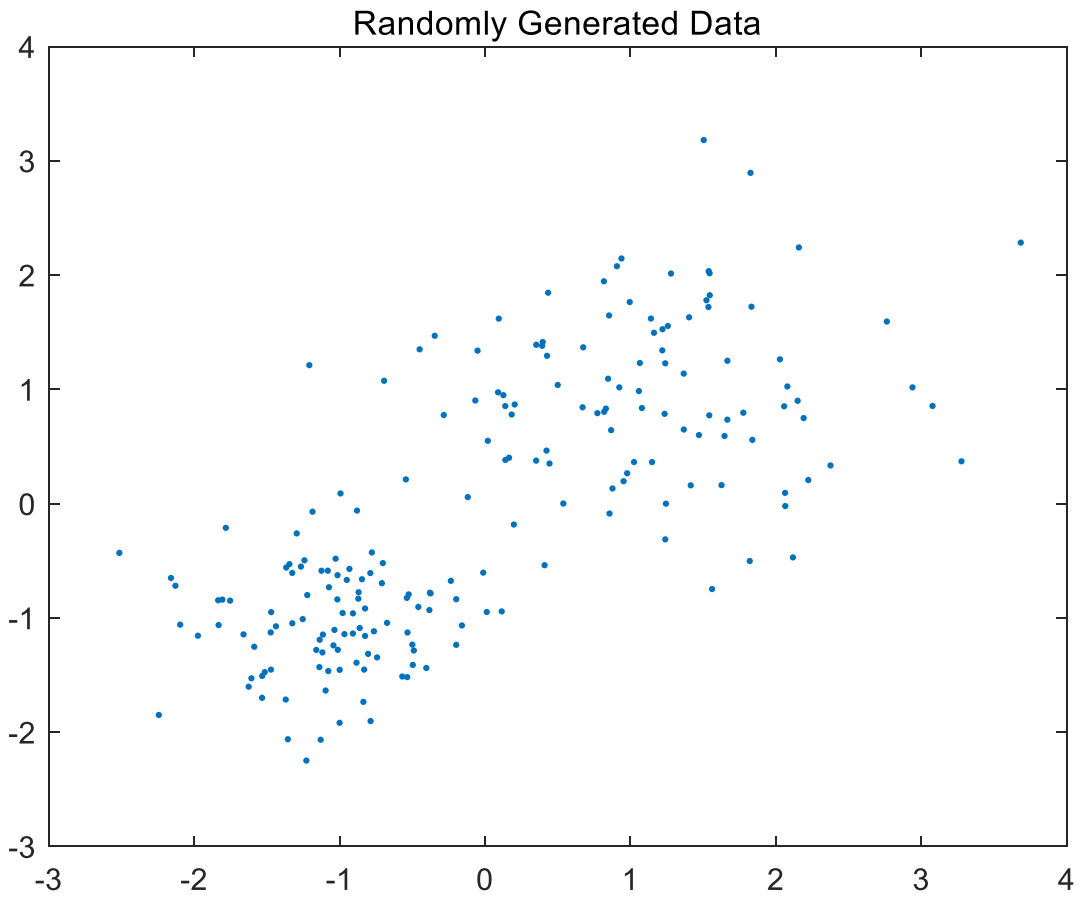
$$m_j = \frac{1}{c_j} \cdot \sum_{x \in c_j} x \quad j = 1,2, \cdots, k$$

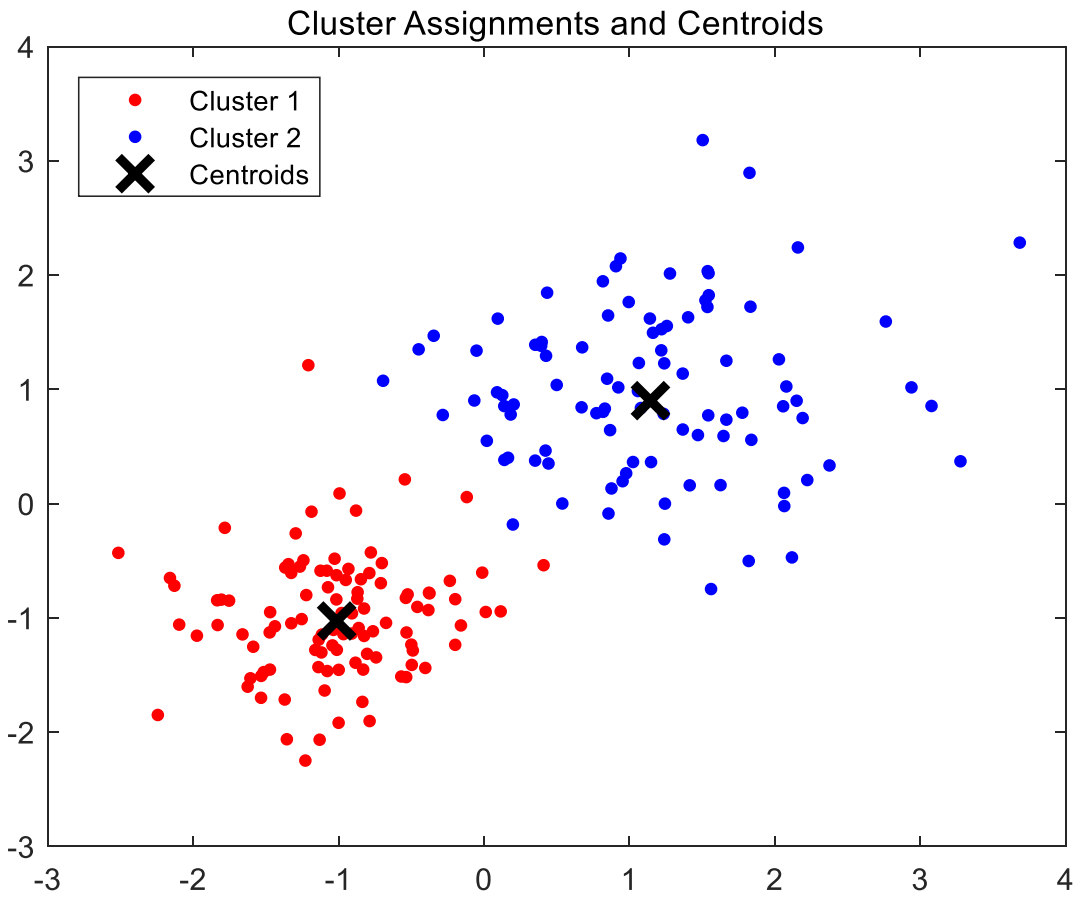
4) Keeps altering 2) and 3) until stop changing.

E.g. Image histogram

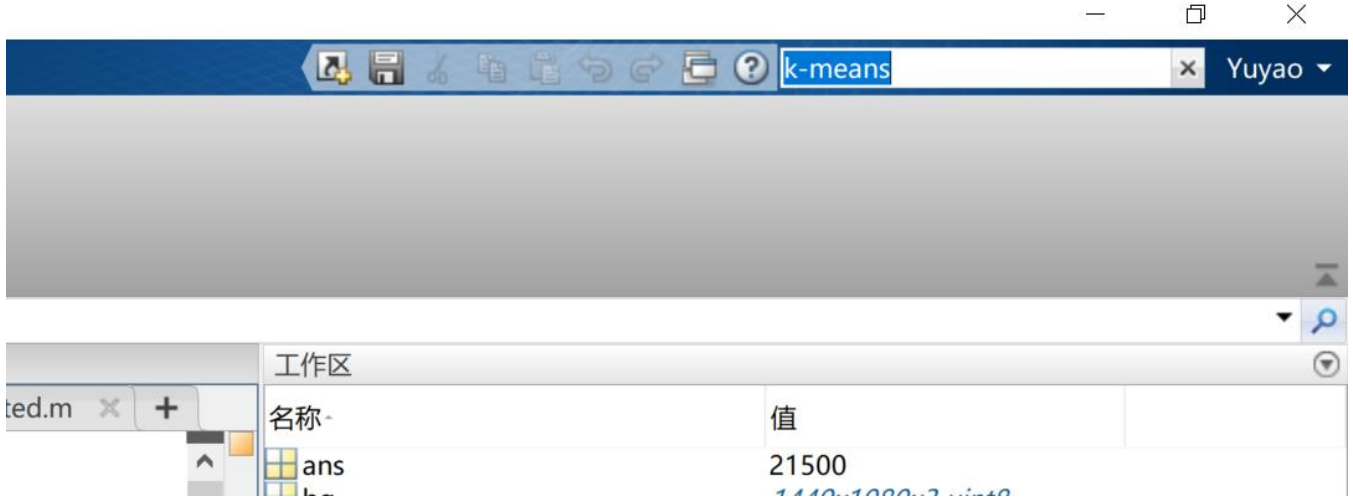
E.g. Color space k-means cluster.

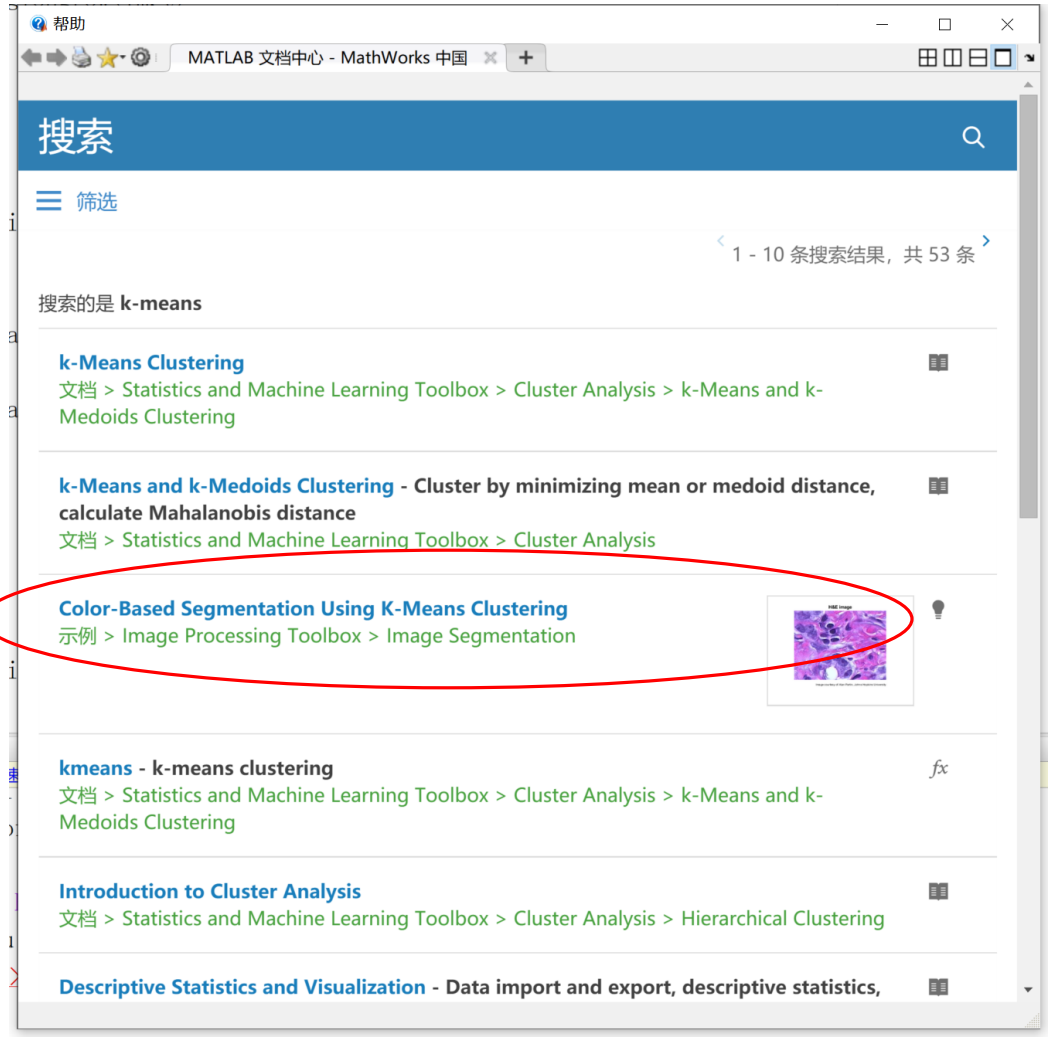
K-means demo





Try this





H&E image

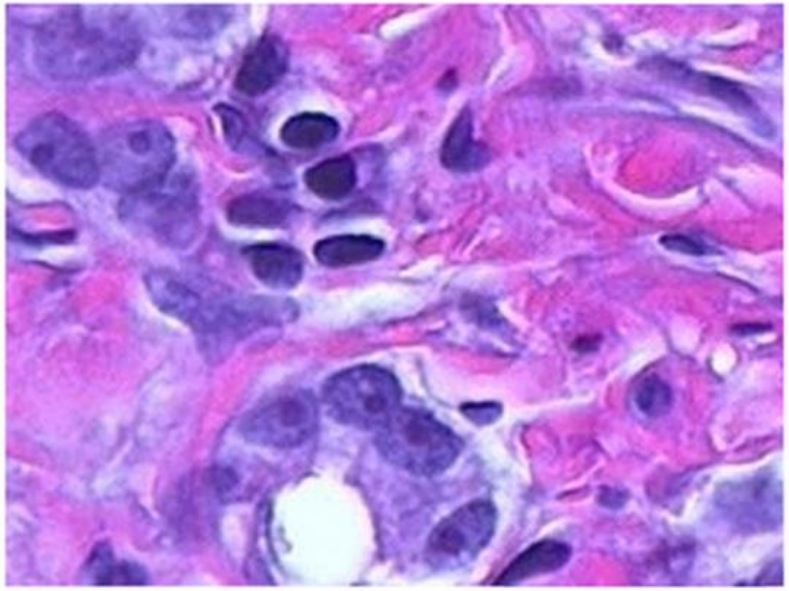
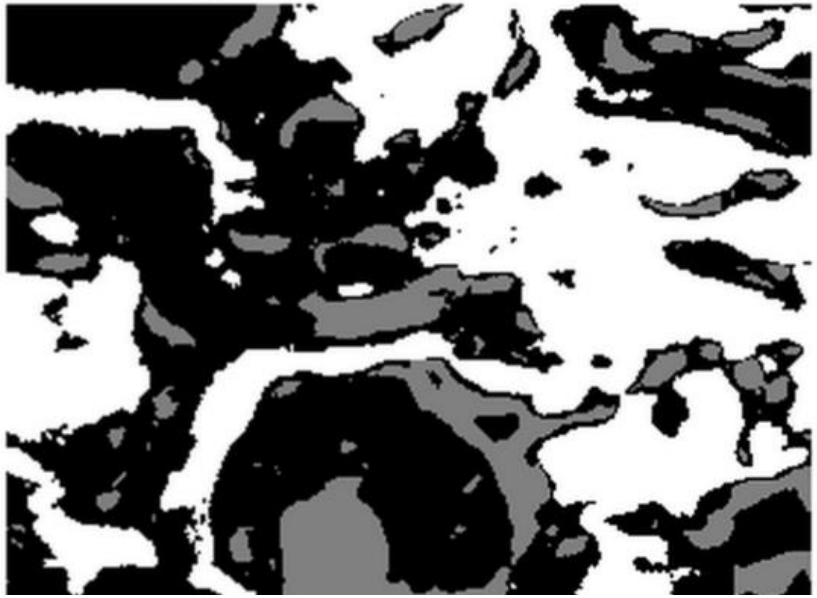
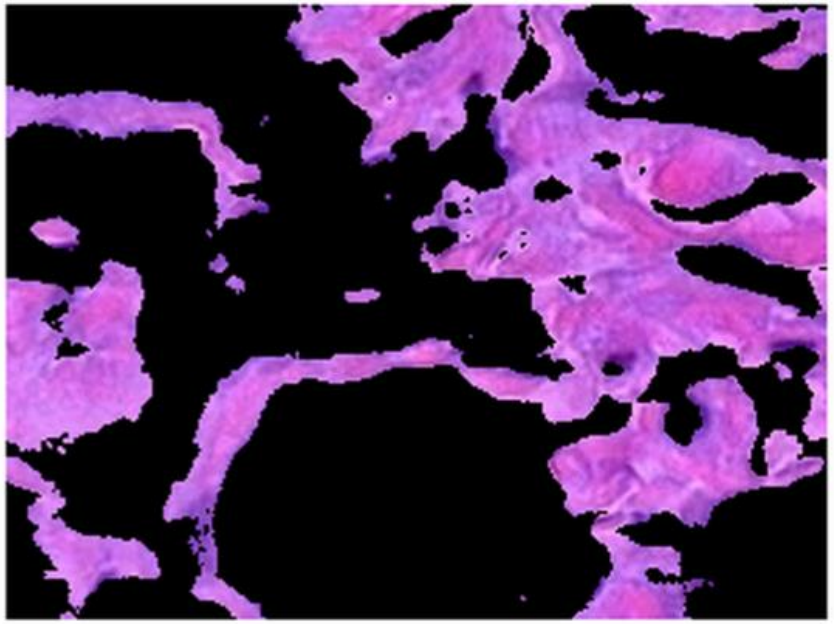


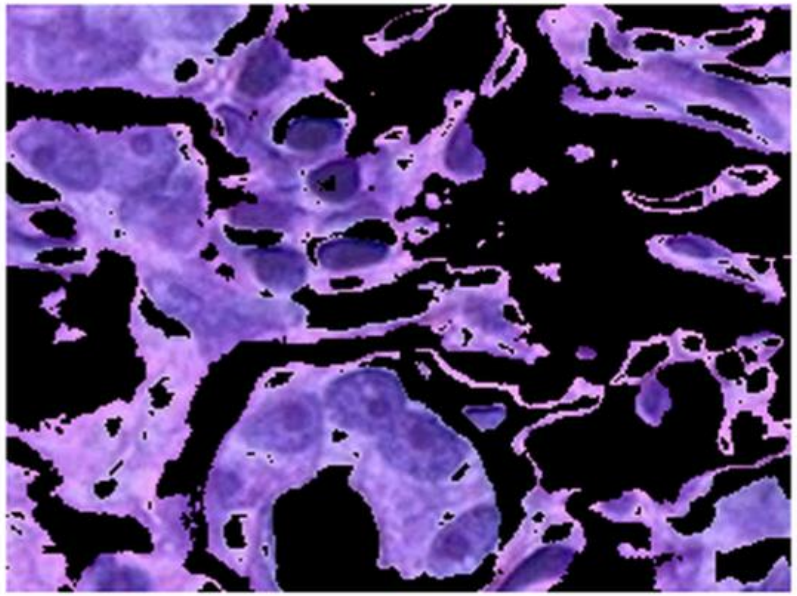
image labeled by cluster index



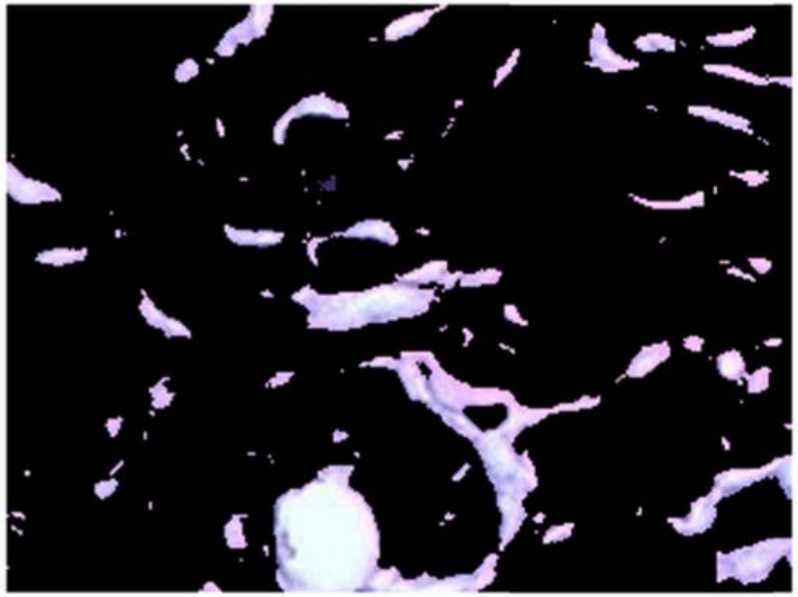
objects in cluster 3



objects in cluster 1



objects in cluster 2



Super-pixel

- Modification of K-means used in image processing;
- Regions of image that are contiguous and have similar intensity/color.
- Why doing this?
  - More compact (e.g. thousands of super-pixels could represent millions of pixels).
  - “Keeps things together” better for subsequent segmentation; computationally efficient.

Super-pixel

Idea: Clustering 5-D vectors $[r, g, b, x, y]$

1) initialize super-pixels center by sampling N locations on a region grid in image plane.

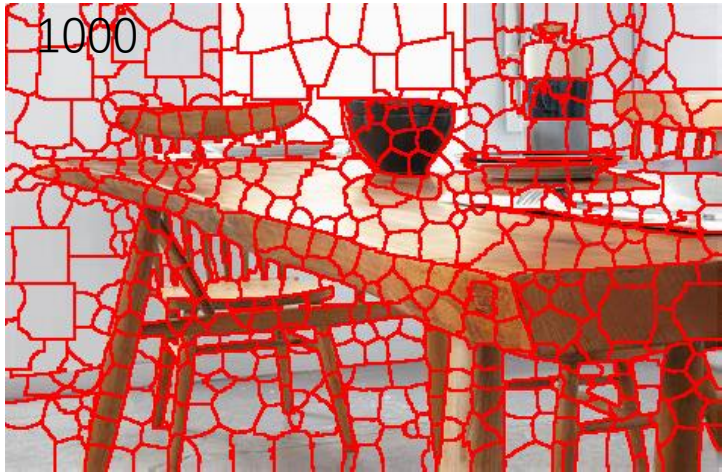
- ❖ Move slightly within 3x3 neighborhood to lie on lowest gradient position (Don't want to start on an edge).

2) For each cluster center $m_{\mathbf{i}}$, compute distance bwt $m_{\mathbf{i}}$ and each pixel in a neighborhood of $m_{\mathbf{i}}$.

- ❖ Only search in neighborhood, the region size depends on # of *N*.
- ❖ Assign pixel to cluster $\mathbf{i}$ if its distance is better than its current value.

Try this

- Check the “mysuperpix.m” function in discussion folder.



Take home message

- **Pixel-based segmentation:**

  each pixel is segmented based on gray-level values, no contextual information, only histogram.

  -Example: hough transform

- **Region-based segmentation:**

  considers gray-levels from neighboring pixels by

  – including similar neighboring pixels (region growing),

  – split-and-merge,

  – or super-pixel segmentation.

- **Edge-based segmentation:**

  Detects and links edge pixels to form contours.

Take home message

- Region based methods are robust because:
  - Regions cover more pixels than edges and thus you have more information available in order to characterize your region
  - When detecting a region you could for instance use texture which is not easy when dealing with edges
  - Region growing techniques are generally better in noisy images where edges are difficult to detect
- The edge based method can be preferable because:
  - Algorithms are usually less complex
  - Edges are important features in an image to separate regions
- The edge of a region can often be hard to find because of noise or occlusions
- Combination of results may often be a good idea