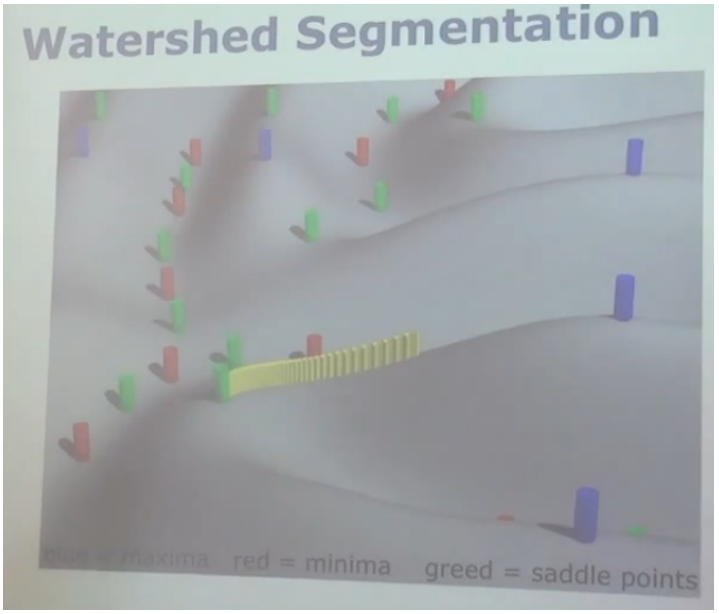
Watershed Segmentation

1. Choose local minima as region seeds (each is given a different label)
2. Add neighbors to priority queue, sorted by intensity value
3. Take top priority pixel from queue
   - If all labeled neighbors have same label, assign to pixel
   - Add all non-marked neighbors
4. Repeat step 3 until finished

Matlab: seg = watershed(im)

Watershed Segmentation



Watershed Segmentation

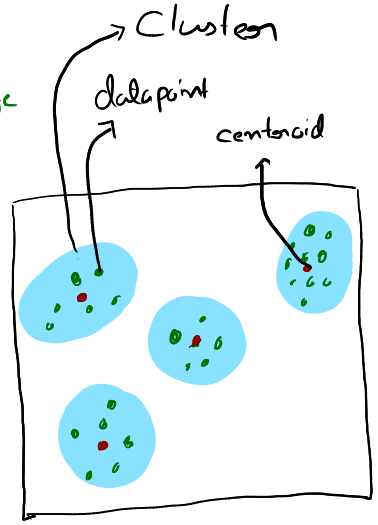
- Popular segmentation approach
- An effective form of hierarchical clustering for image segmentation
- Tends to oversegment images

k-mean Clustering

K-Means Clustering

- Clusters are represented by centroids
- Centroids do not need to be members of the data
- Partitions the data into k clusters
- Objective: Find the k cluster centers and assign the data points to the nearest cluster, such that the squared distances from the cluster centroids are minimized

Can just be average of data points.



K-Means Clustering Algorithm (Informally)

- Iterative procedure
- Initialization: Choose k arbitrary centroids (cluster means)
- Repeat until convergence
  - Assign each data point to the closest centroid
  - Adjust the centroids of the clusters to the mean of the data points assigned to them

Same algorithm, written formally

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

For all $\boldsymbol{m}_i, i = 1, \ldots, k$

$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

Until $\boldsymbol{m}_i$ converge

Re-compute the cluster means $\boldsymbol{m}_i$ using the current cluster memberships

Assign each $\boldsymbol{x}^t$ to the closest cluster

Strength of K-Means

- Easy to understand and to implement
- Efficient $O(nkt)$
  n = #iterations, k = #clusters, t = #data points
- Converges quickly to a **local** optimum
- Probably the most popular clustering algorithm

Weaknesses of K-Means

- User needs to specify #clusters (k)
  (although there are methods to estimate k)
- Sensitive to the initialization strategy:
  Use different seeds
- Sensitive to outliers since all data points contribute equally to the mean
  Strategy: Try to identify/eliminate outliers
- Prefers clusters of approximately similar size (objects are assigned to the nearest centroid)

Mean Shift Clustering

Mean Shift Algorithm

The mean shift algorithm seeks for the **modes** of the given set of points

1. Choose kernel and bandwidth
2. For each point:
   a) Center a window on the point
   b) Compute the mean of the data in the search window
   c) Center the search window at the new mean location
   d) Repeat (b,c) until convergence
3. Assign points that end up at nearby modes to the same cluster

Mean Shift Pros and Cons

Pros

- Good general-practice segmentation
- Flexible in number and shape of regions
- Robust to outliers

Cons

- Have to choose kernel size in advance
- Not suitable for high-dimensional features