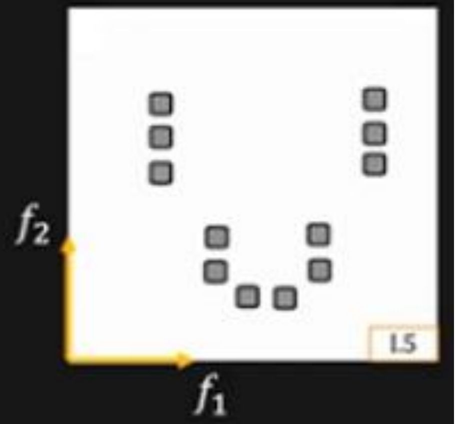
K-Means and Mean Shift

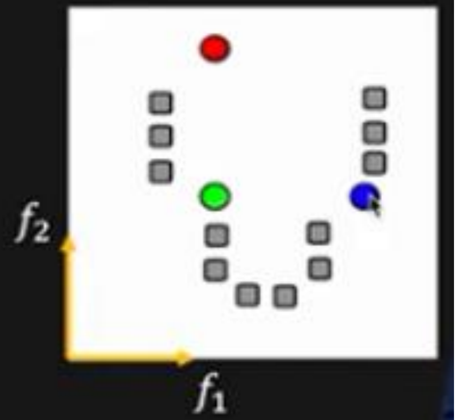
3-Means Clustering Example

Problem: Segment the given pixel feature distribution into 3 clusters.



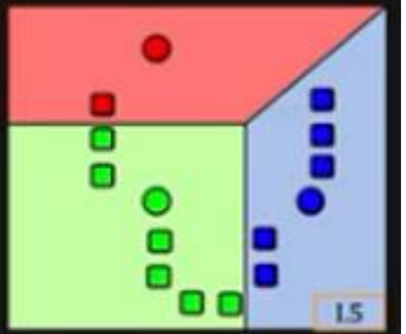
Solution:

Step 1: Randomly generate the initial centroids (means) of the 3 clusters.

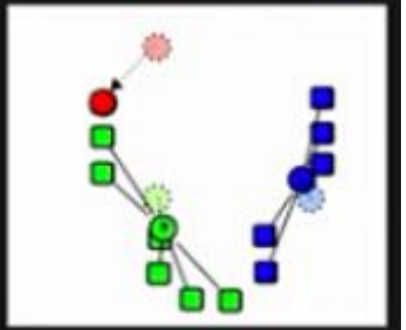


3-Means Clustering Example

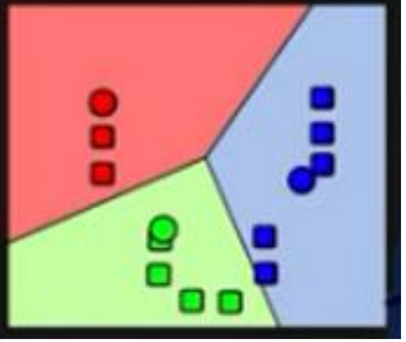
Step 2: Create 3 clusters by assigning each feature point to the nearest mean.



Step 3: Recompute the mean of each cluster.



Step 4: Repeat steps 2 and 3 until convergence.



k-Means Clustering

Given: Image with N pixels and number of clusters k.

Task: Find the k clusters.

Clustering:

1: Pick k points randomly as the initial centroids (means) $\{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k\}$ of the k clusters in feature space.

2: For each pixel $\mathbf{x}_j$ find nearest cluster mean $\mathbf{m}_i$ to pixel's feature $\mathbf{f}_j$ and assign pixel to cluster i.

3: Recompute mean for each cluster using its assigned pixels.

4: If changes in all k means is less than a threshold ε, stop. Else go to step 2.

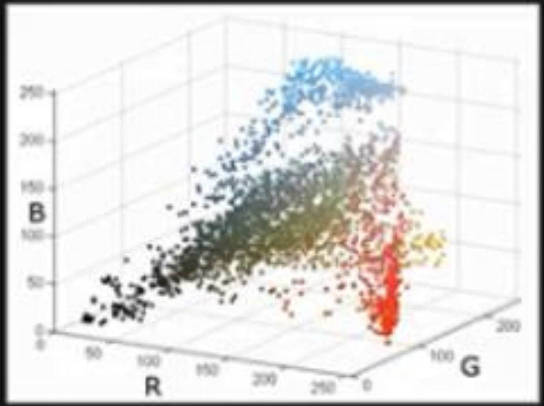
[Lloyd 1957, MacQueen 1967]

k-Means Initialization Methods

- **Method 1**: Select k random feature points as initial centroids. If two points are very close, resample.
- **Method 2**: Select k uniformly distributed means within the range of the distribution.
- **Method 3**: Perform k-means clustering on a subset of pixels and use the result as the initial means.

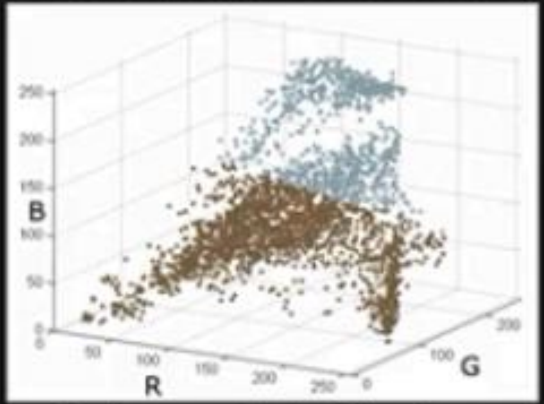
k-Means Clustering Results (k = 2)

Input Image



Pixel RGB Color Distribution
(Color of feature point ≡ Color of image pixel)

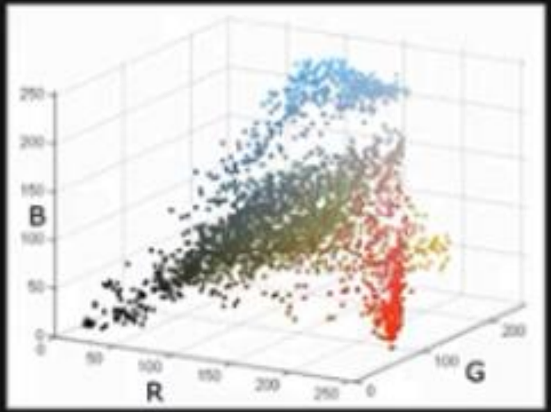
Segmented Image
($k = 2$; $\{R, G, B\}$-space)



Color-coded Clusters
(Color of feature point ≡ Color of image segment)

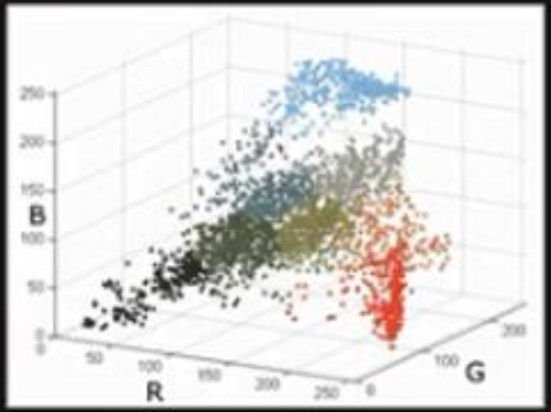
k-Means Clustering Results (k = 8)

Input Image



Pixel RGB Color Distribution
(Color of feature point ≡ Color of image pixel)

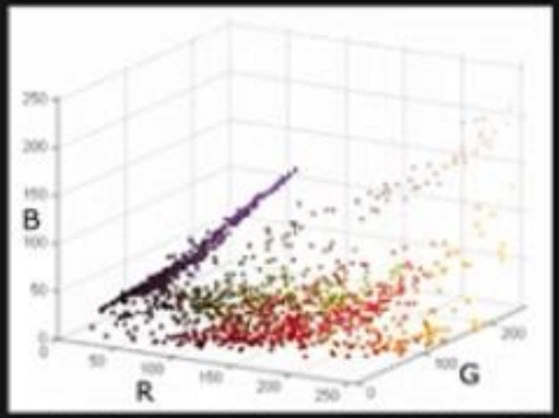
Segmented Image
($k = 8$; $\{R, G, B\}$-space)



Color-coded Clusters
(Color of feature point ≡ Color of image segmen

k-Means Clustering Results (k = 16)

Input Image



Pixel RGB Color Distribution

Segmented Image
($k = 16$; $\{R, G, B\}$-space)

Segmented Image
($k = 16$; $\{R, G, B, x, y\}$-space)

Note: Disjoint regions could belong to a single cluster.

- **Measuring similarity between observations**
- **Euclidean distance**: Most common method to measure distance between observations, when observations include continuous variables is the Euclidean distance.
- Let observations $u = (u_1, u_2, \ldots, u_q)$ and $v = (v_1, v_2, \ldots, v_q)$ each comprise measurements of q variables.
- The Euclidean distance between observations u and v is
- $d_{u,v} = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + (u_q - v_q)^2}$

- **Measuring similarity between observations**
- **Euclidean distance**: Most common method to measure distance between observations, when observations include continuous variables is the Rectilinear distance.
- Let observations $u = (u_1, u_2, \ldots, u_q)$ and $v = (v_1, v_2, \ldots, v_q)$ each comprise measurements of q variables.
- The Rectilinear distance between observations u and v is
- $d_{u,v} = |u_1 - v_1| + |u_2 - v_2| + \cdots + |u_q - v_q|$

K-means Clustering

- **Example**

Cluster the following eight points (with (x, y) representing locations) into three clusters A1(2, 10) A2(2, 5) A3(8, 4) A4(5, 8) A5(7, 5) A6(6, 4) A7(1, 2) A8(4, 9).

Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2).

The distance function between two points $a=(x1, y1)$ and $b=(x2, y2)$ is defined as: $\rho(a, b) = |x2 - x1| + |y2 - y1|$.

- Use k-means algorithm to find the three cluster centers after the second iteration.

Solution:

Iteration 1

		(2, 10)	(5, 8)	(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(2, 10)	0	5	9	1
A2	(2, 5)	5	6	4	3
A3	(8, 4)	12	7	9	2
A4	(5, 8)	5	0	10	2
A5	(7, 5)	10	5	9	2
A6	(6, 4)	10	5	7	2
A7	(1, 2)	9	10	0	3
A8	(4, 9)	3	2	10	2

Iteration 1

		(2, 10)	(5, 8)	(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(2, 10)	0	5	9	1
A2	(2, 5)	5	6	4	3
A3	(8, 4)	12	7	9	2
A4	(5, 8)	5	0	10	2
A5	(7, 5)	10	5	9	2
A6	(6, 4)	10	5	7	2
A7	(1, 2)	9	10	0	3
A8	(4, 9)	3	2	10	2

Cluster 1	Cluster 2	Cluster 3
(2, 10)	(8, 4)	(2, 5)
	(5, 8)	(1, 2)
	(7, 5)	
	(6, 4)	
	(4, 9)	

Example Continued

- Next, we need to re-compute the new cluster centers (means). We do so, by taking the mean of all points in each cluster.

- For Cluster 1, we only have one point A1(2, 10), which was the old mean, so the cluster center remains the same.

- For Cluster 2, we have ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6)

- For Cluster 3, we have ((2+1)/2, (5+2)/2) = (1.5, 3.5)

Example Continued

		(2, 10)	(6, 6)	(1.5, 3.5)	
	Point	Dist Mean 1	Dist Mean 2	Dist Mean 3	**Cluster**
A1	(2, 10)	0	8	7	1
A2	(2, 5)	5	5	2	3
A3	(8, 4)	12	4	7	2
A4	(5, 8)	5	3	8	2
A5	(7, 5)	10	2	7	2
A6	(6, 4)	10	2	5	2
A7	(1, 2)	9	9	2	3
A8	(4, 9)	3	5	8	1

Example Continued

- Next, we need to re-compute the new cluster centers (means). We do so, by taking the mean of all points in each cluster.

- In Cluster 1, we have points 1 and 8. Therefore the centroid is: $((2+4)/2,(10+9)/2)=(3,9.5)$
- In Cluster 2, we have points 3,4,5 and 6. Therefore, the centroid is: $((8+5+7+6)/4,(4+8+5+4)/4)=(6.5,5.25)$
- For Cluster 3, we have points 2 and 7. Therefore, the centroid is: $((2+1)/2,(5+2)/2)=(1.5,3.5)$

Example Continued

		(3, 9.5)	(6.5 ,5.25)	(1.5, 3.5)	
	Point	Dist Mean 1	Dist Mean 2	Dist Mean 3	**Cluster**
A1	(2, 10)	1.5	9.25	7	**1**
A2	(2, 5)	5.5	4.75	2	**3**
A3	(8, 4)	10.5	2.75	7	**2**
A4	(5, 8)	3.5	4.25	8	**1**
A5	(7, 5)	8.5	0.75	7	**2**
A6	(6, 4)	8.5	1.75	5	**2**
A7	(1, 2)	9.5	8.75	2	**3**
A8	(4, 9)	1.5	6.25	8	**1**

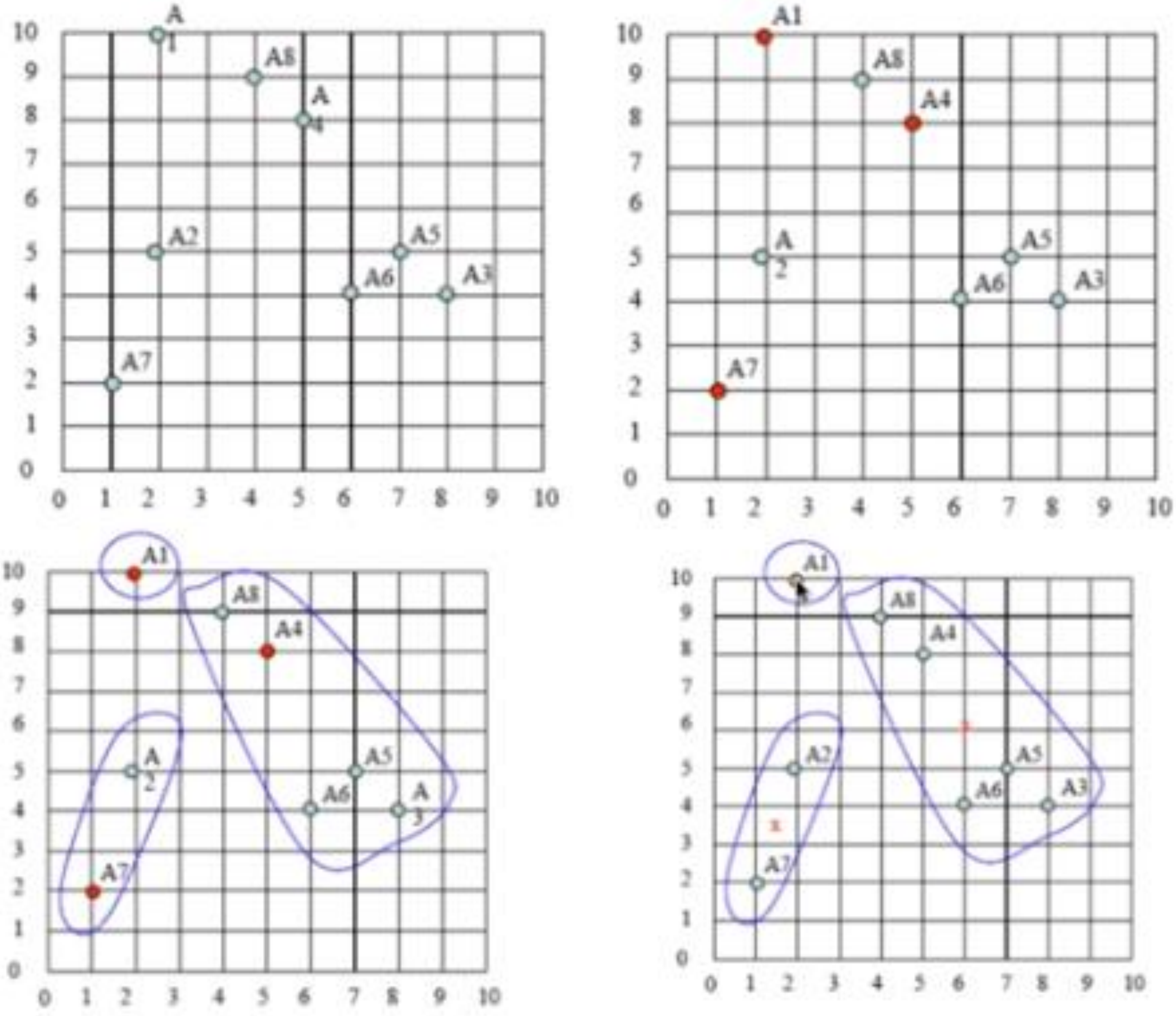
Example Continued

- Next, we need to re-compute the new cluster centers (means). We do so, by taking the mean of all points in each cluster.

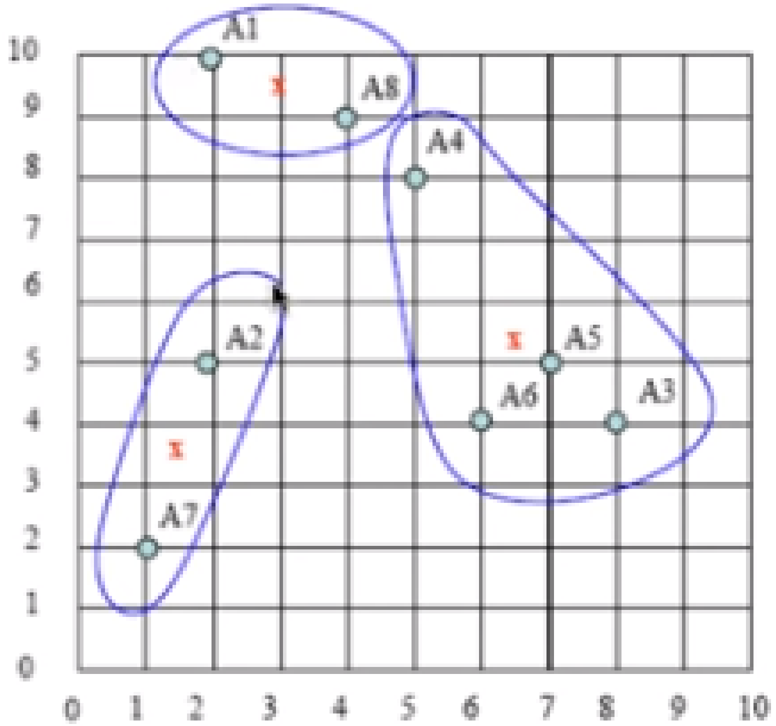
- In Cluster 1, we have points 1, 4, and 8. Therefore the centroid is: $((2+5+4)/2,(10+8+9)/2)=(3.67,9)$
- In Cluster 2, we have points 3,5 and 6. Therefore, the centroid is: $((8+7+6)/4,(4+5+4)/4)=(7,4.3)$
- For Cluster 3, we have points 2 and 7. Therefore, the centroid is: $((2+1)/2,(5+2)/2)=(1.5,3.5)$

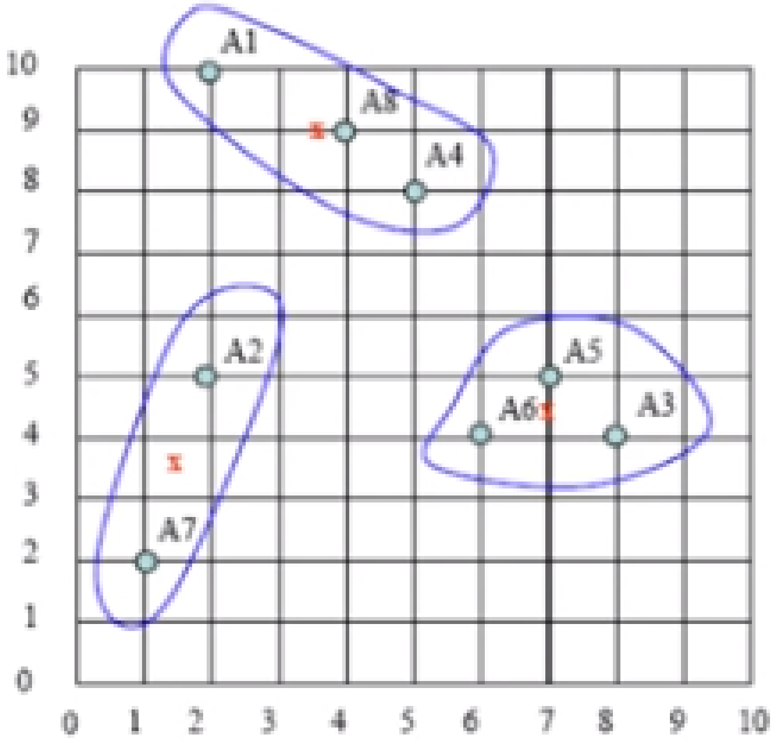
Example Continued

		(3.67, 9)	(7 ,4.3)	(1.5, 3.5)	
	Point	Dist Mean 1	Dist Mean 2	Dist Mean 3	**Cluster**
A1	(2, 10)	2.67	10.7	7	**1**
A2	(2, 5)	5.67	5.7	2	**3**
A3	(8, 4)	9.33	1.3	7	**2**
A4	(5, 8)	2.33	5.7	8	**1**
A5	(7, 5)	7.33	0.7	7	**2**
A6	(6, 4)	7.33	1.3	5	**2**
A7	(1, 2)	9.67	8.3	2	**3**
A8	(4, 9)	0.33	7.7	8	**1**


A1
A2
A3
A4
A5
A6
A7
A8
0 1 2 3 4 5 6 7 8 9 10

Next two Iterations

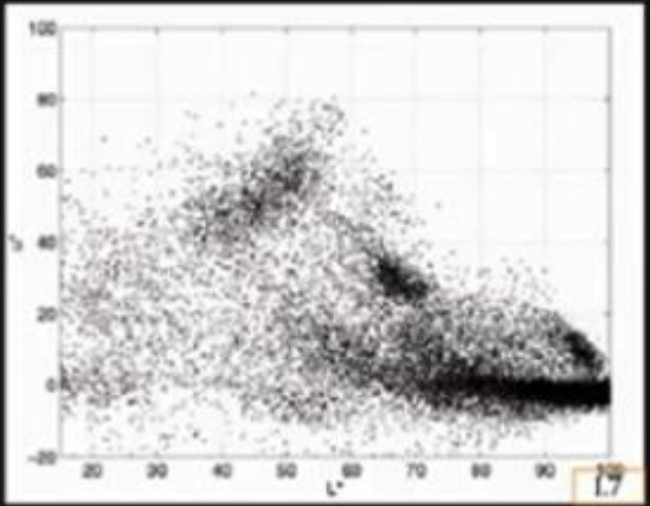




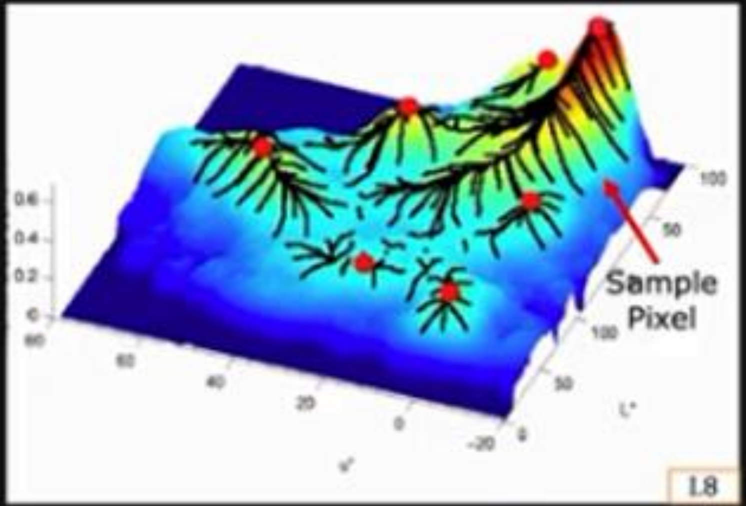
k-Means Clustering: Comments

- Simple and reasonably fast
- Need to pick the number of clusters k
- Sensitive to initialization
- Sensitive to outliers

The Concept of Mean Shift



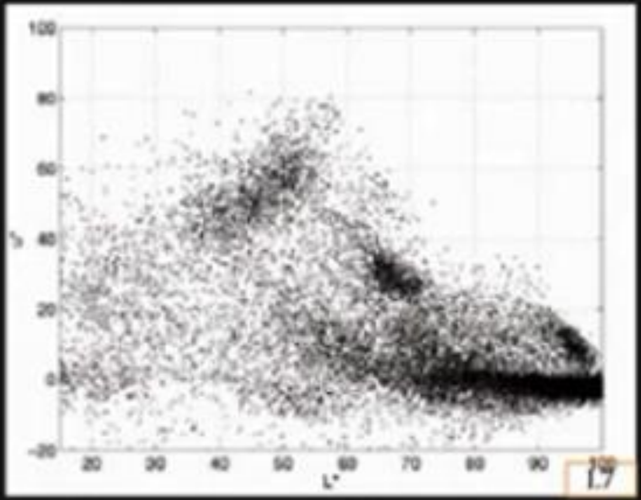
Pixel Feature Distribution



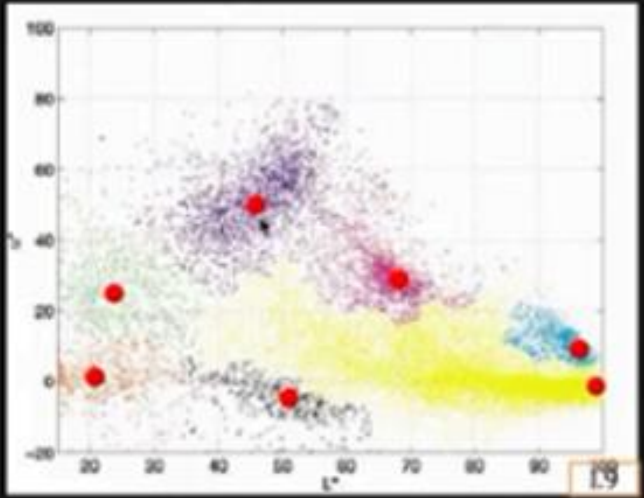
Normalized Density

- Each hill represents a cluster.
- Peak (mode) of hill represents "center" of cluster.
- Each pixel climbs the steepest hill within its neighborhood
- Pixel assigned to the hill (cluster) it climbs.

The Concept of Mean Shift



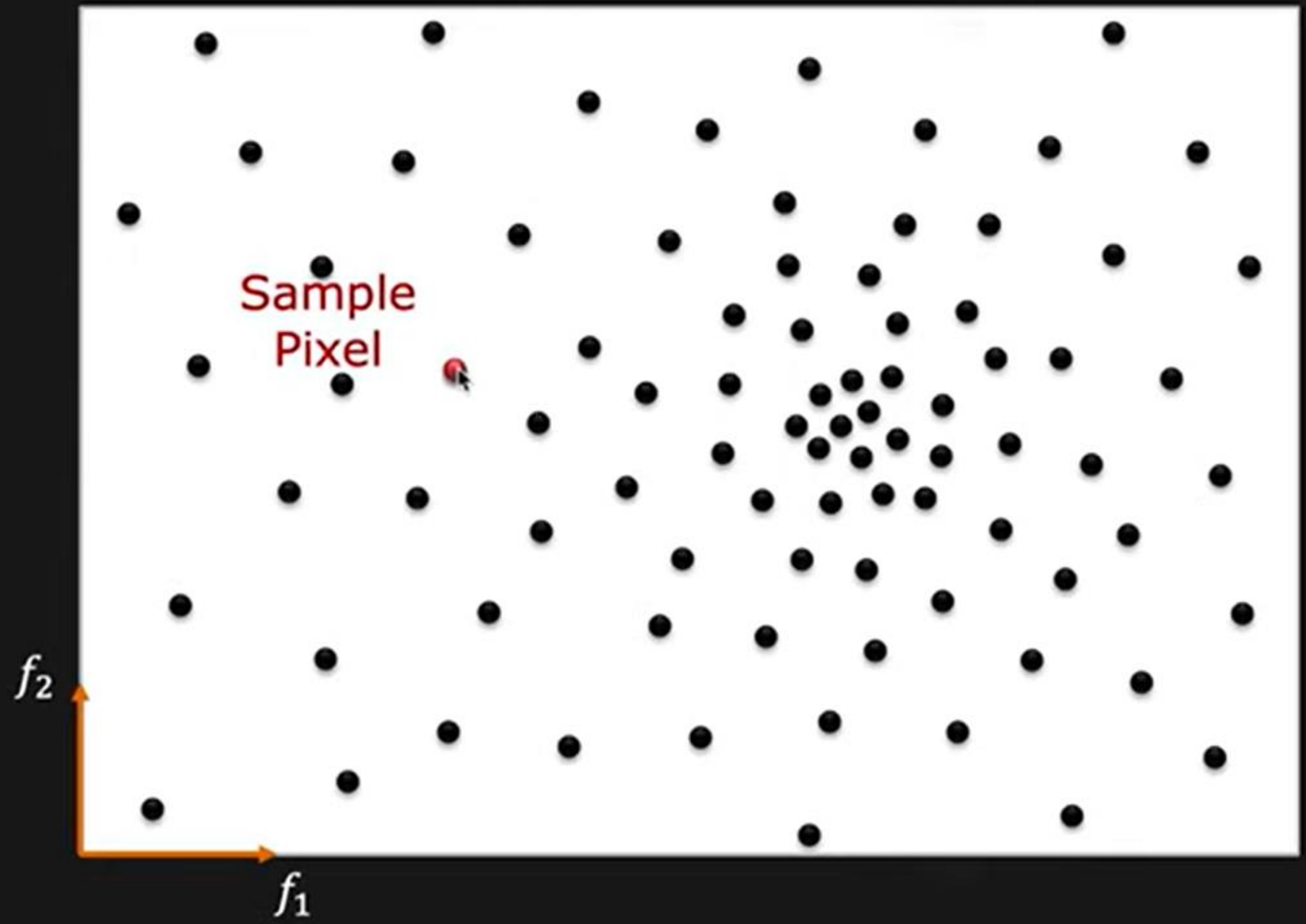
Pixel Feature Distribution



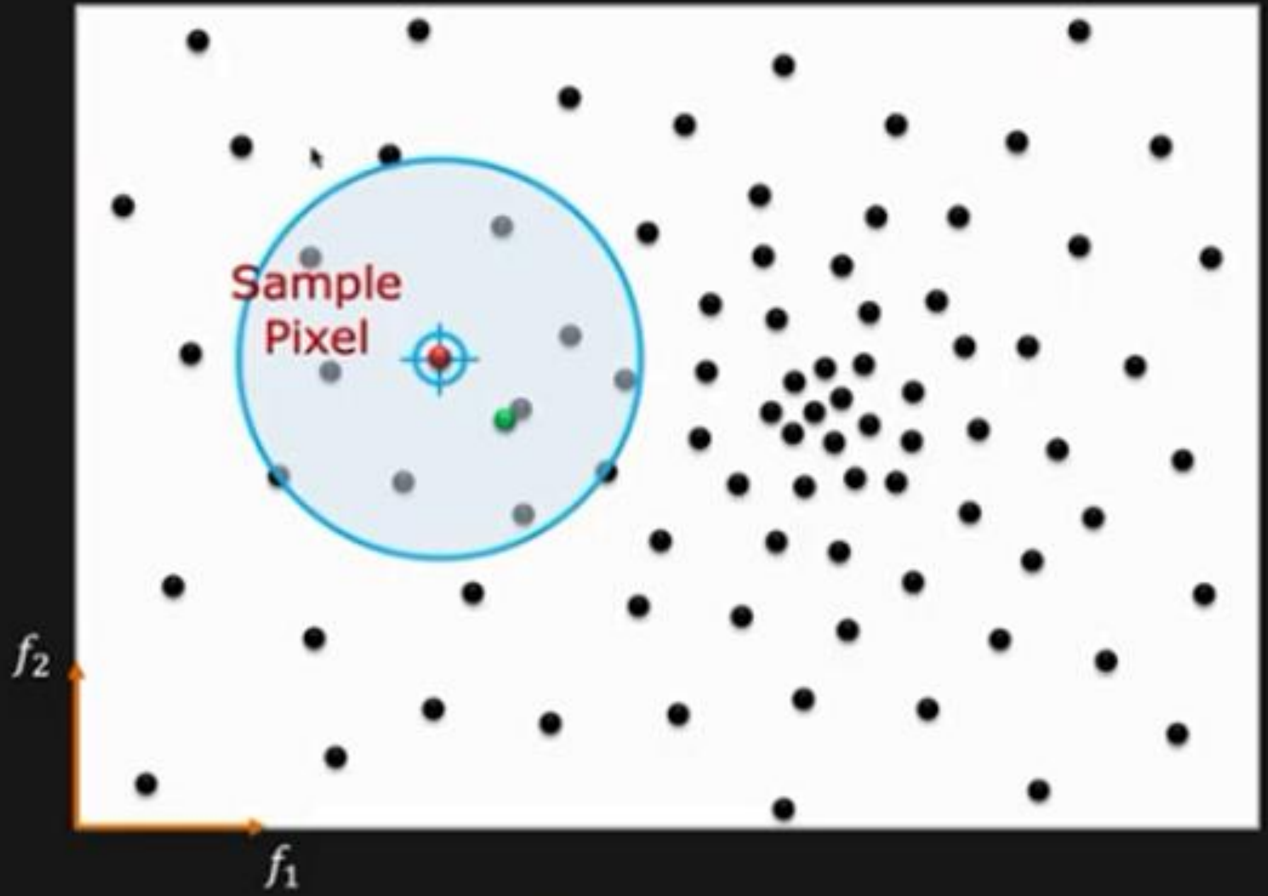
Labeled Clusters and their Centers

- Each hill represents a cluster.
- Peak (mode) of hill represents "center" of cluster.
- Each pixel climbs the steepest hill within its neighborhood
- Pixel assigned to the hill (cluster) it climbs.

Hill Climbing using Mean Shift



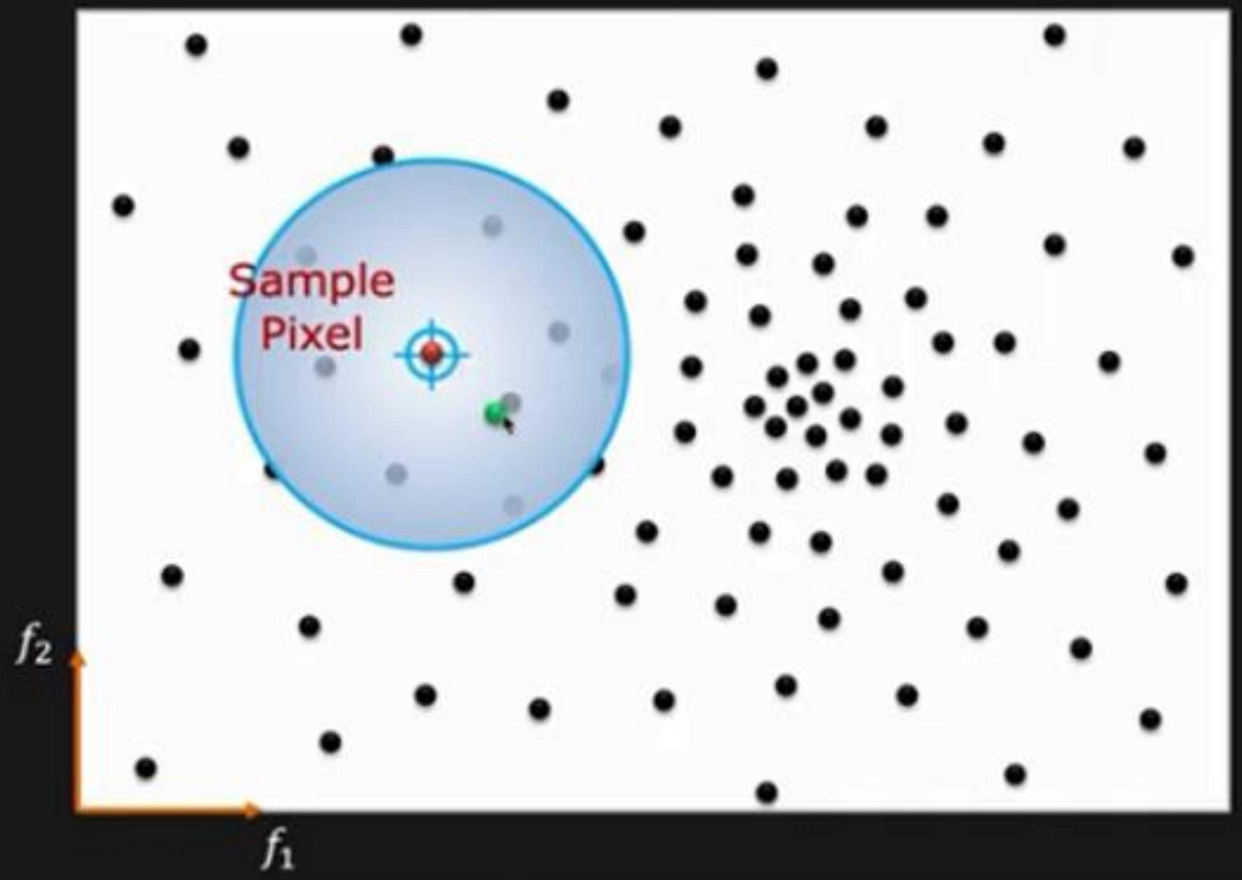
Hill Climbing using Mean Shift



Compute Centroid (•) within a Window of size W.

Use Simple Means or...

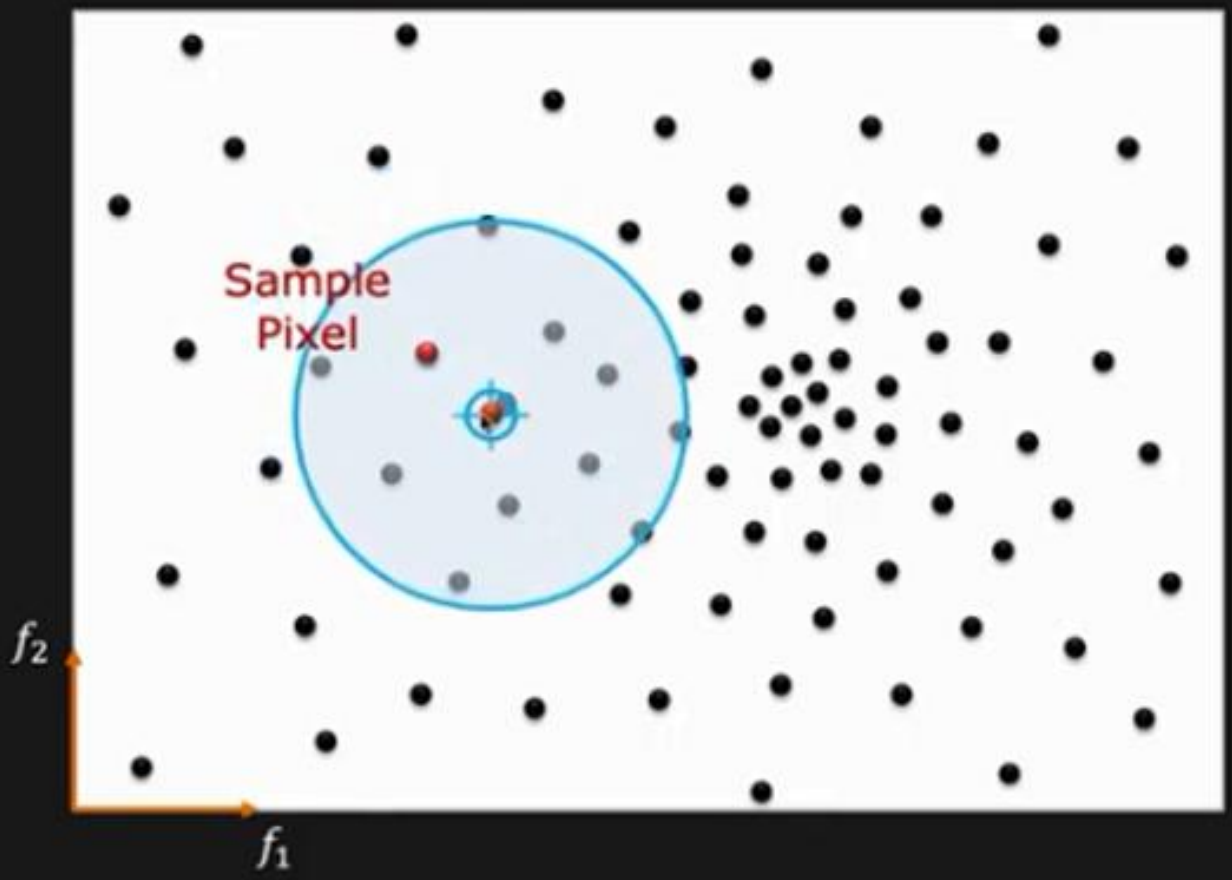
Hill Climbing using Mean Shift



Compute Centroid ($\bullet$) within a Window of size W.

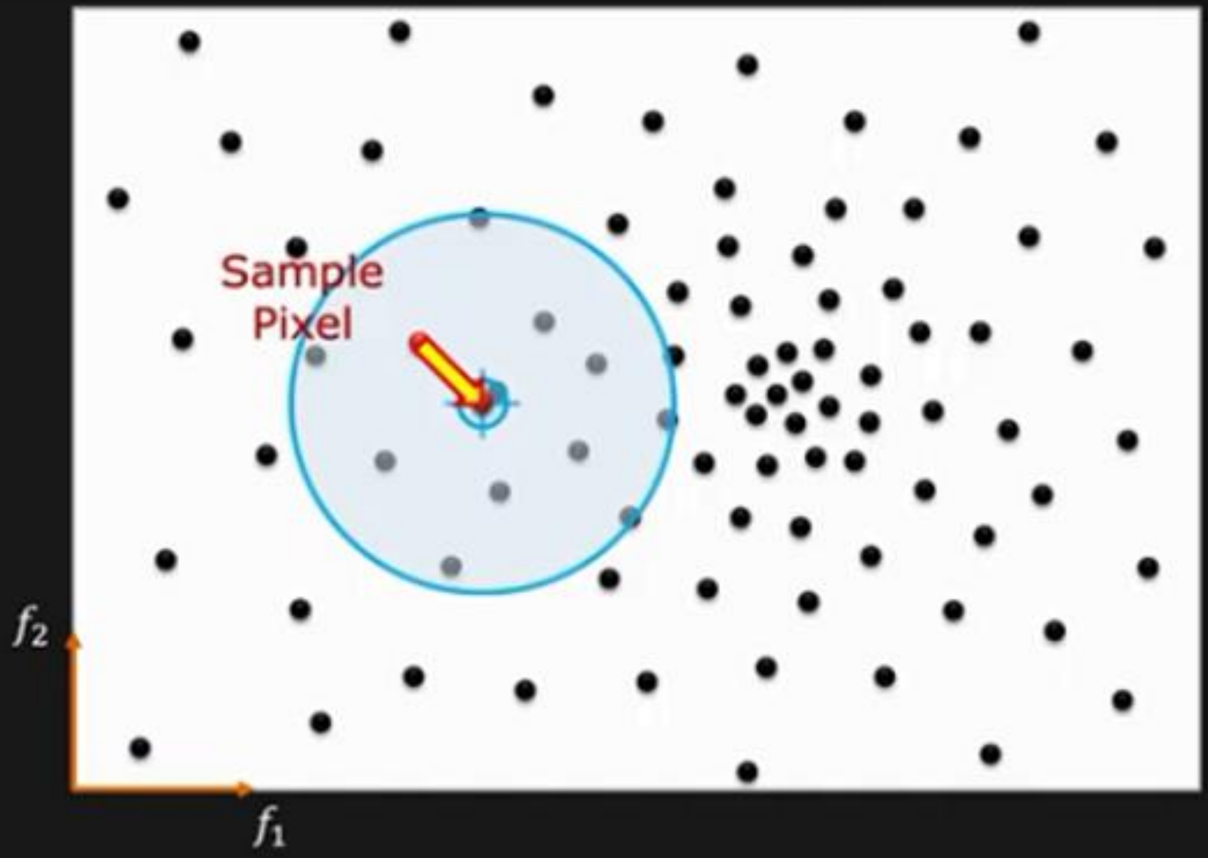
Use Weighted Means.

Hill Climbing using Mean Shift



"Shift" the Window to the Centroid

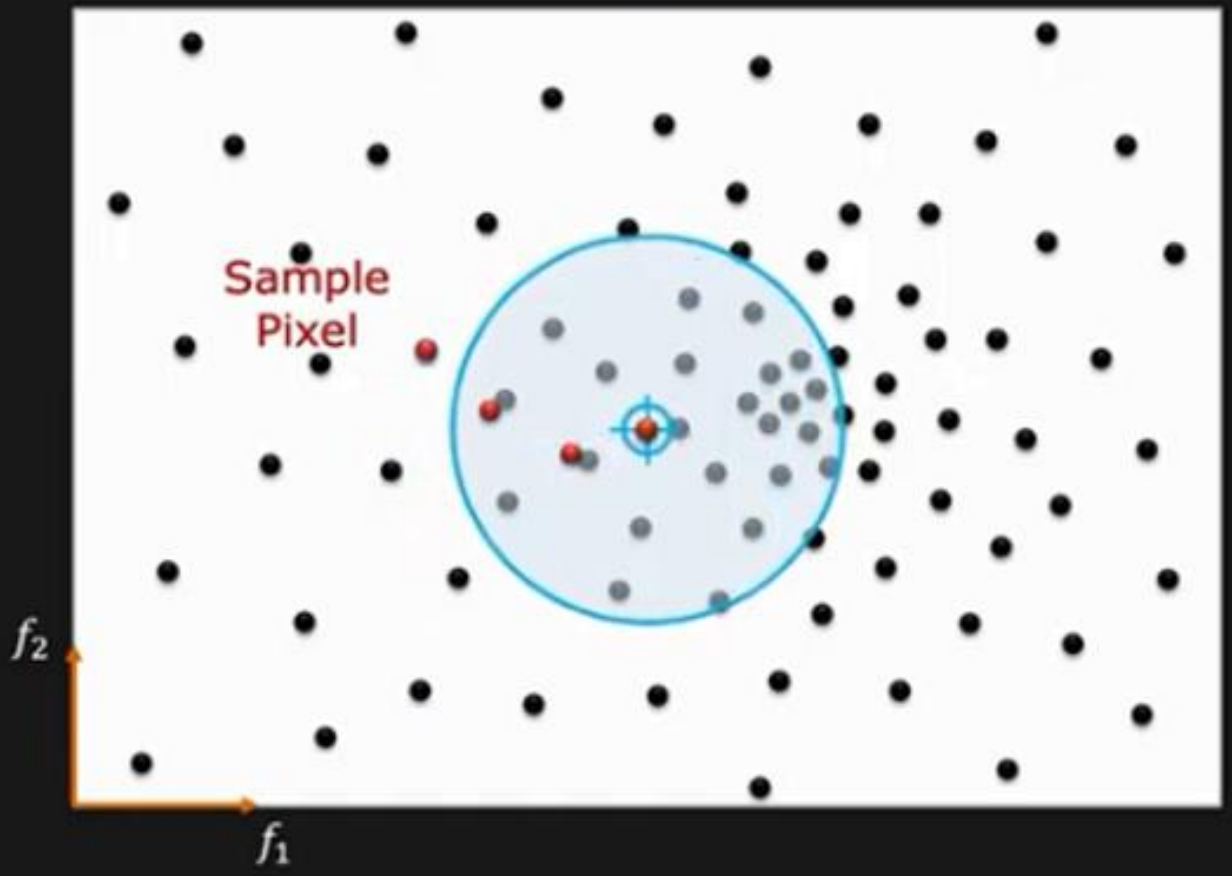
Hill Climbing using Mean Shift



"Shift" the Window to the Centroid

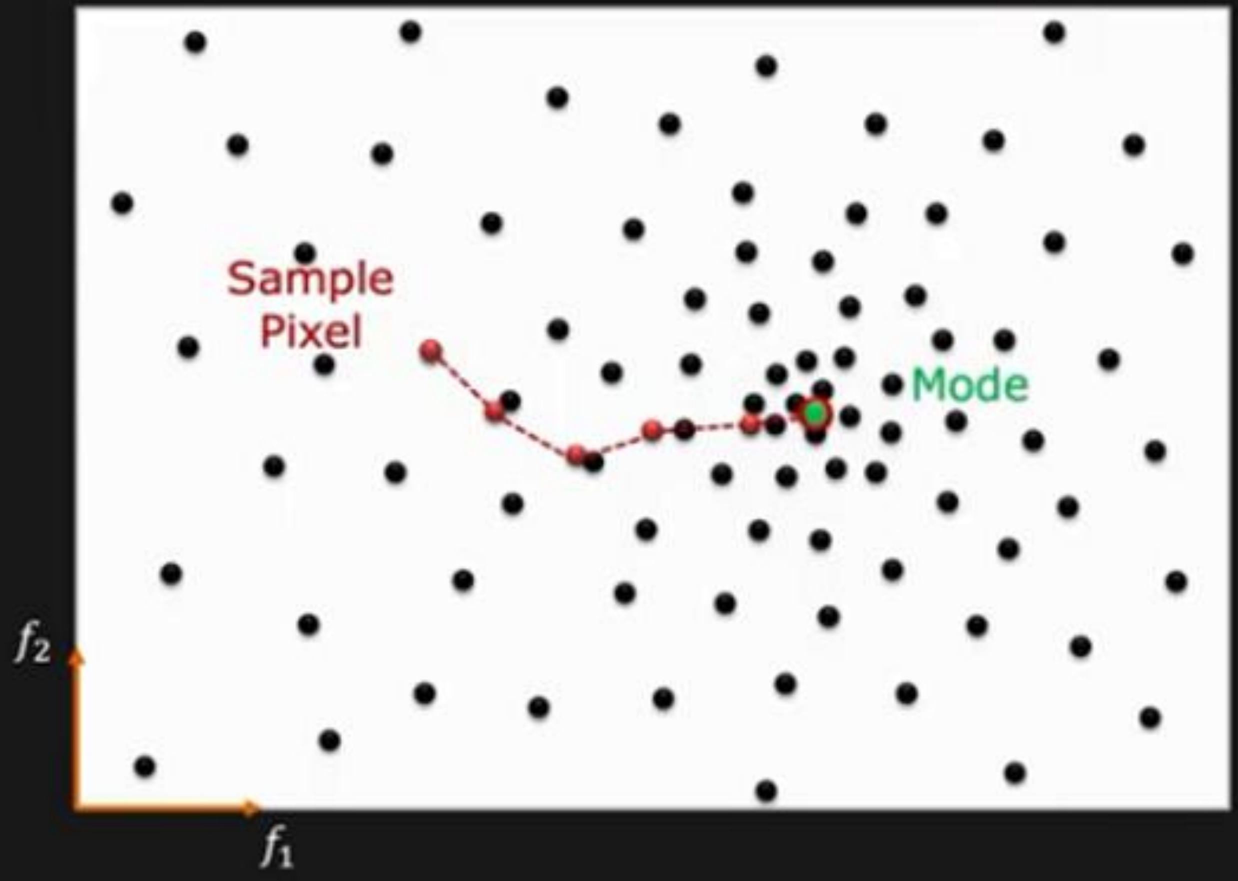
: Mean Shift Vector

Hill Climbing using Mean Shift



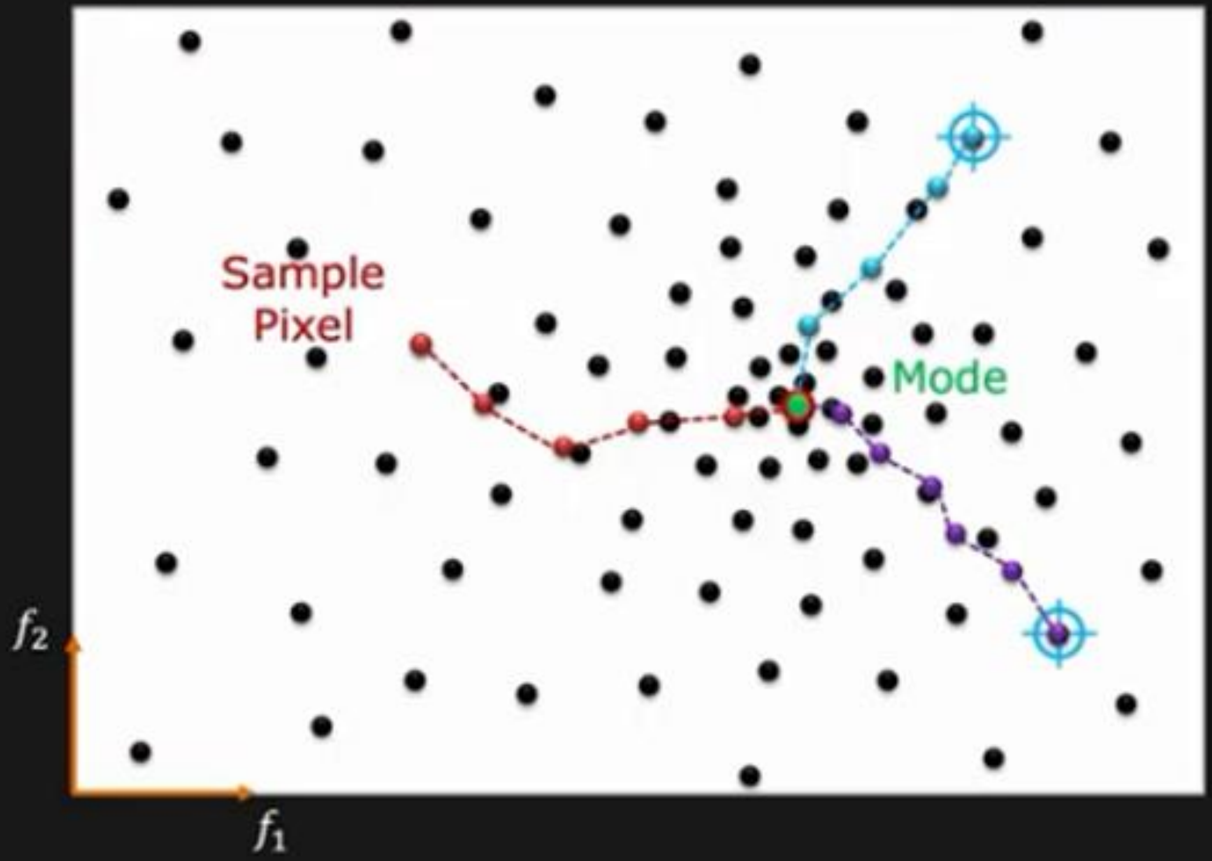
Repeat...Until Convergence.

Hill Climbing using Mean Shift



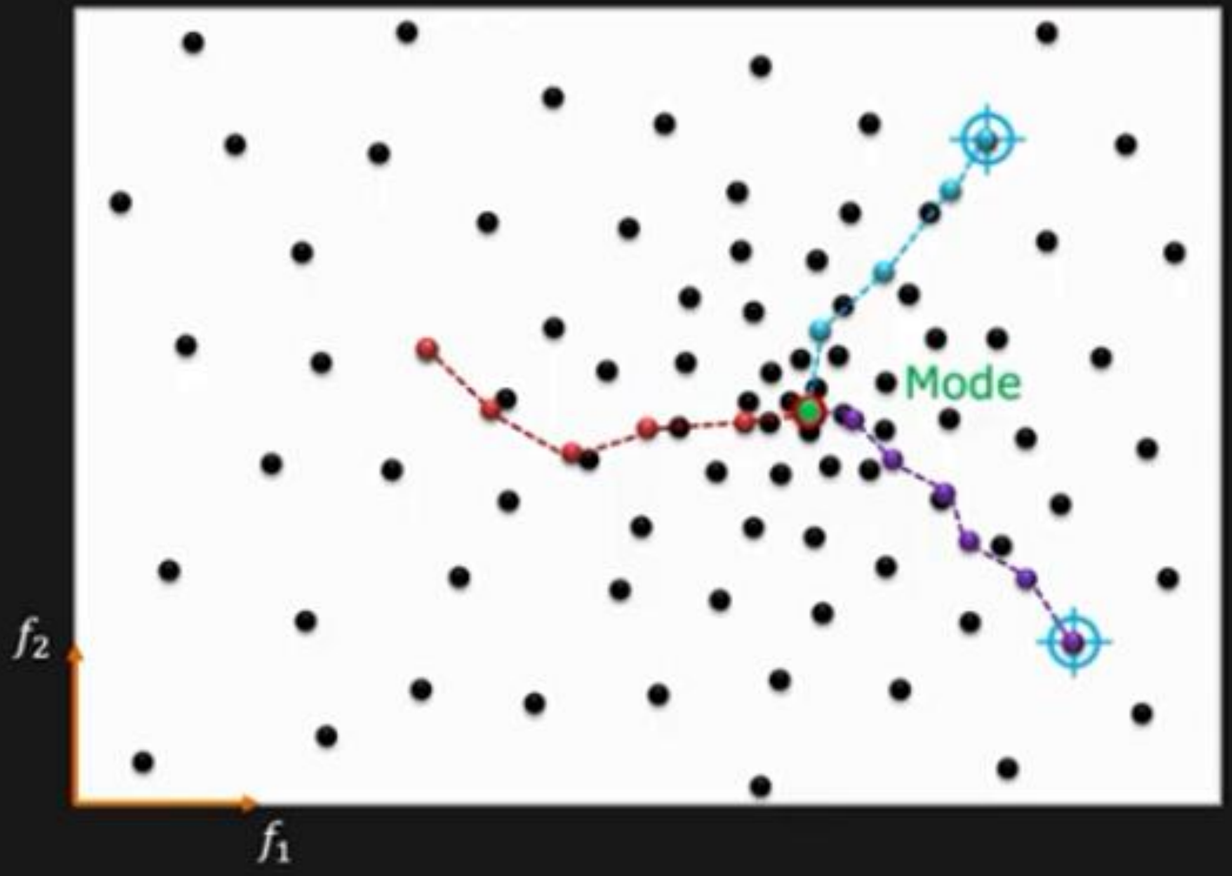
Declare mode and assign it a cluster label.

Hill Climbing using Mean Shift



Repeat…for all pixels!

Hill Climbing using Mean Shift



Features that converge to same mode belong to same cluster.

Means Shift Algorithm

Given: Distribution of N pixels in feature space.

Task: Find modes (clusters) of distribution.

Clustering:

1: Set $\mathbf{m}_i = \mathbf{f}_i$ as initial mean for each pixel i.

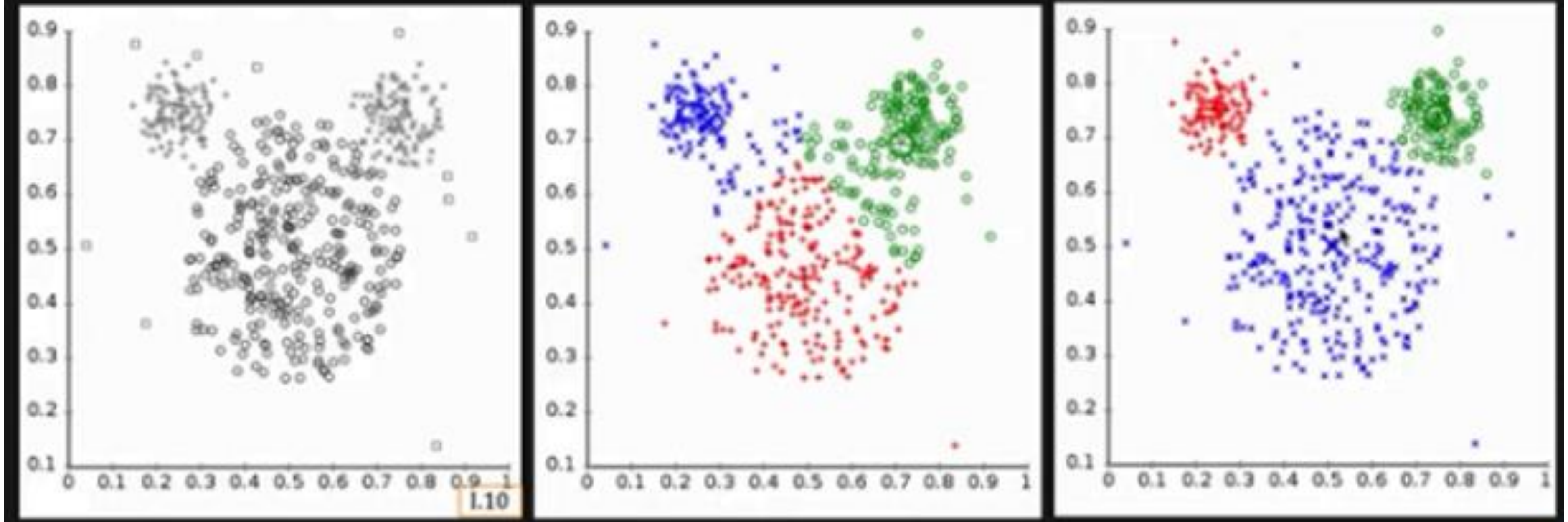
2: Repeat the following for each mean $\mathbf{m}_i$:

- a: Place window of size W around $\mathbf{m}_i$.
- b: Compute centroid $\mathbf{m}$ within the window. Set $\mathbf{m}_i = \mathbf{m}$.
- c: Stop if shift in mean $\mathbf{m}_i$ is less than a threshold ε. $\mathbf{m}_i$ is the mode.

3: Label all pixels that have same mode as belonging to same cluster.

[Fukunaga 1975, Comaniciu 2002]

k-Means vs. Mean Shift



Original Data | k-Means (k=3) | Mean Shift

k-Means vs. Mean Shift

Input Image

k-Means (k=16)

Mean Shift (W=21)

Mean Shift Segmentation Results

Examples of Mean Shift Segmentation

Mean Shift: Comments

- Simple but computationally expensive
- Finds arbitrary number of clusters
- No initialization required
- Robust to outliers
- Clustering depends on window size W