

K-means: An implementation for clustering pixels in a bitmap image

2023 - 2024



Contenido

1 K-means algorithm

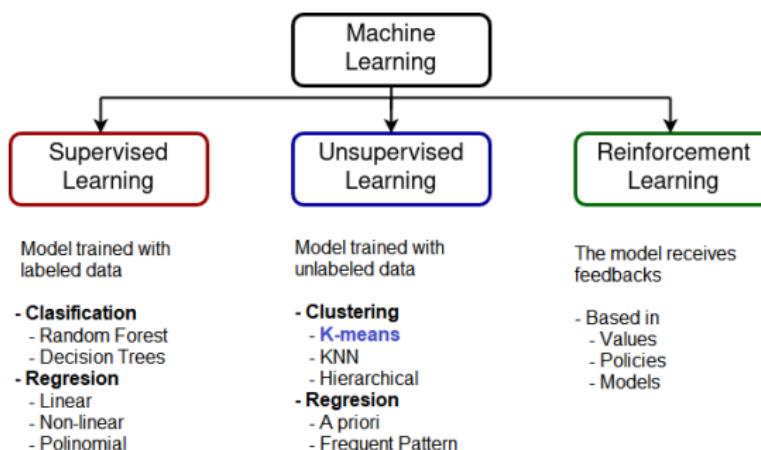
2 Graphical example

3 K-means for BMP images

K-means

Definition

- It is an unsupervised *clustering* algorithm.
- Used to group each element of an unlabeled data collection into one of the K defined groups.



Description

Overall description

- A target number **K** is defined, which indicates the **number of clusters** or groups to use. Each cluster is represented by its **centroid**, which is the central value.
- Each element of the data set is assigned to the closest cluster. To do this, the **distance** of the element towards the centroids is calculated, and the minimum is obtained. The distance function can be complex to define depending on the problem to be treated; for example, if each data has a large number of attributes.
- Once all the data is assigned to the clusters, the **values are recalculated** for each centroid to obtain the new midpoint of all the data belonging to that cluster. The **mean** of the values can be used.

Pseudocode

(Step 1) Selecting the value of K (number of clusters)

(Step 2) Initialize the K centroids randomly.

convergence = False

while (!convergence)

(Step 3) Assign each data point to the nearest cluster.

(Step 4) Update the centroids with the mean

if ((all the centroids remain the same))

 convergence = True

Contenido

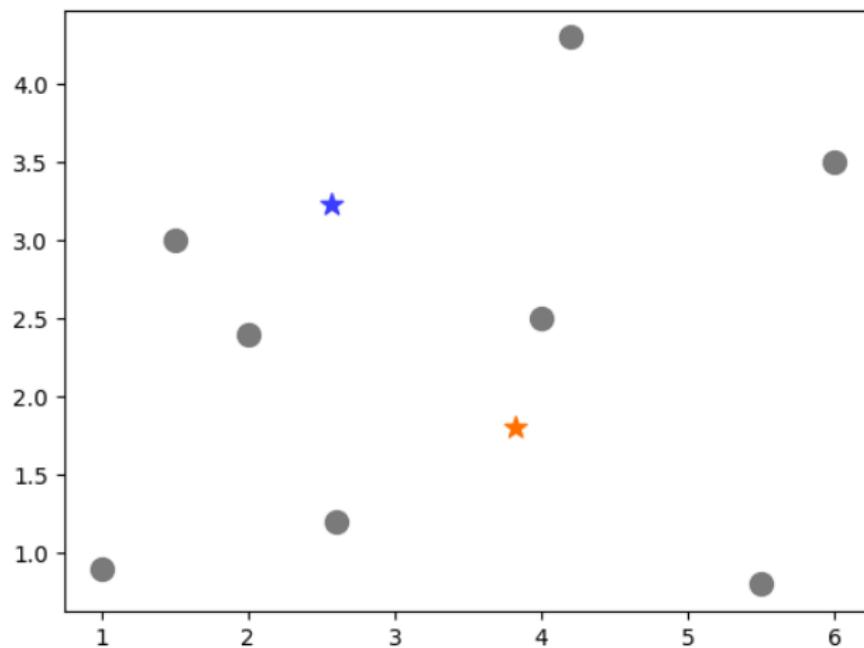
1 K-means algorithm

2 Graphical example

3 K-means for BMP images

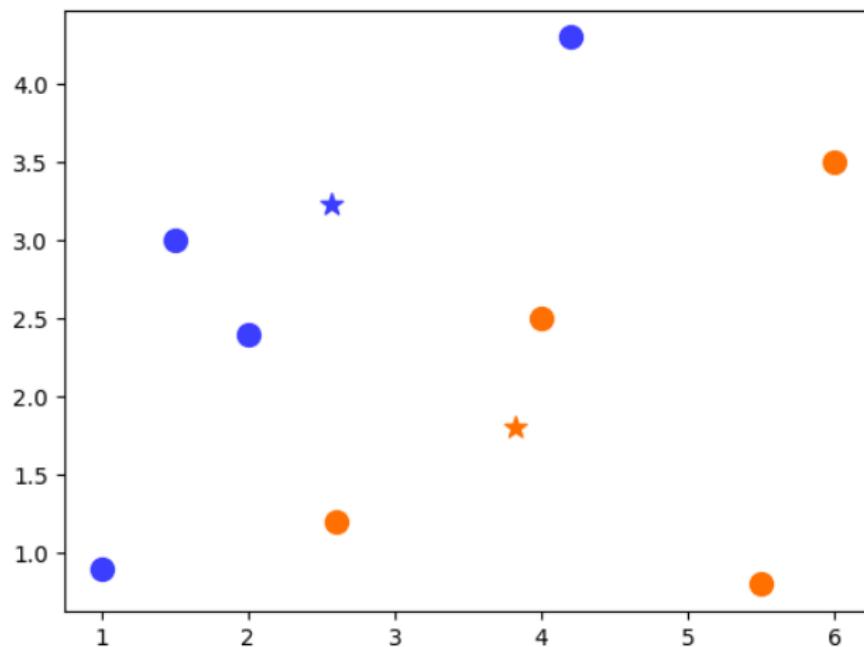
Step 2 (1/7)

Location of two centroids randomly:



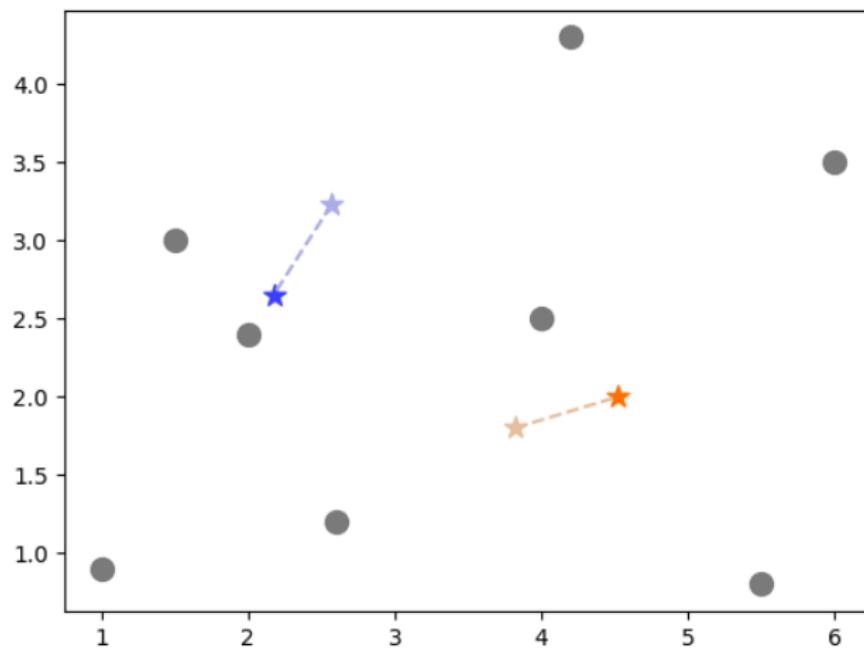
Step 3 (2/7)

Assignment of points to clusters based on the nearest centroid:



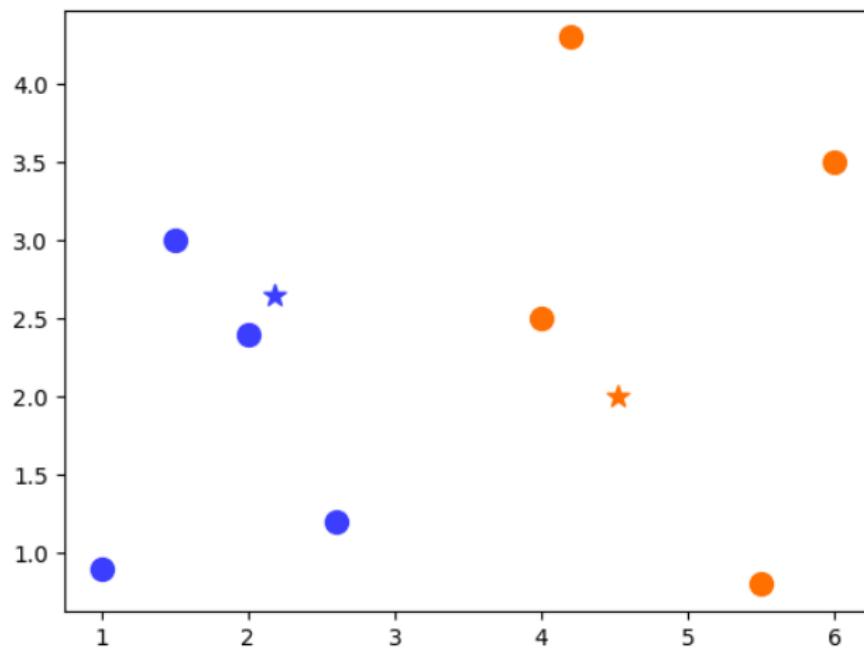
Step 4 (3/7)

Repositioning of centroids based on the mean of the points:



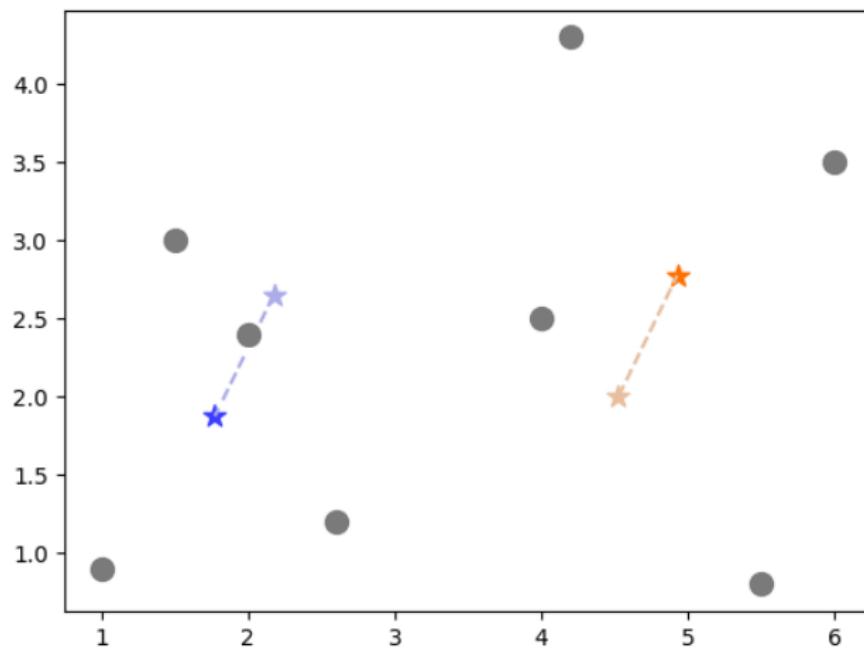
Step 3 (4/7)

Assignment of points to clusters with repositioned centroids:



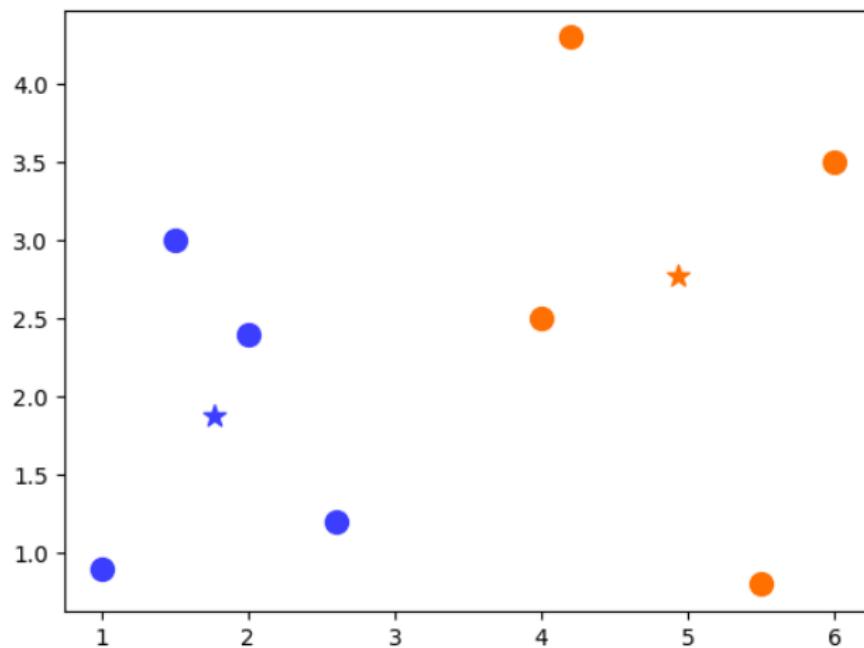
Step 4 (5/7)

Repositioning of centroids based on the mean of the points:



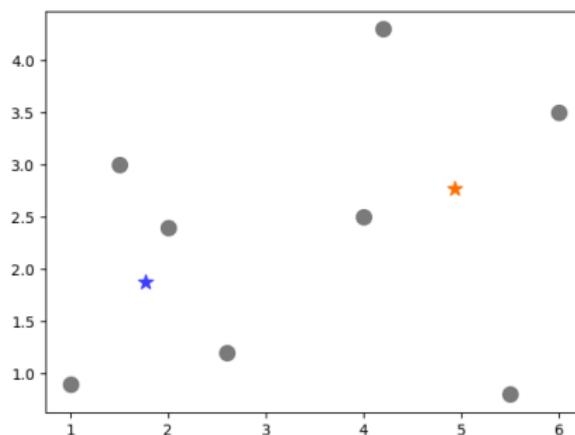
Step 3 (6/7)

Assignment of points to clusters with repositioned centroids:



Step 4 (7/7)

Repositioning of centroids based on the mean of the points:



As the position of the centroids has not changed compared to the last iteration, the algorithm converges and stops.

Contenido

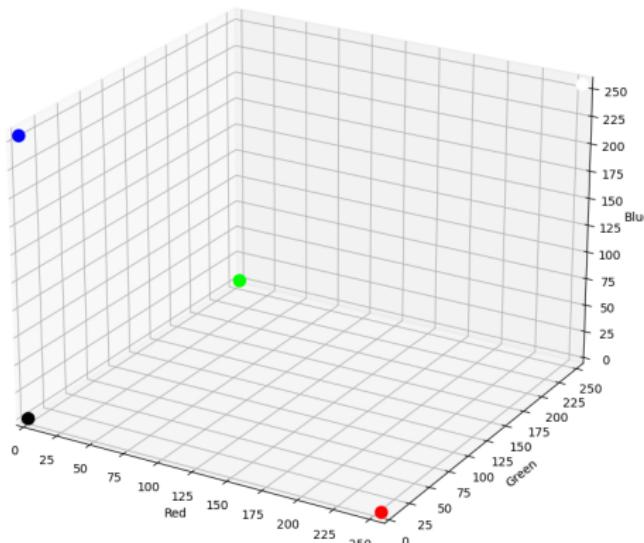
1 K-means algorithm

2 Graphical example

3 K-means for BMP images

K-means for BMP images

- The points (x, y, z) represent each pixel of the image through its red, green, and blue (RGB) values.
- Objective: reduce the number of colors in an image using K-means.
- Each cluster represents one of the K representative colors of the image. Its centroid indicates the RGB value of that color.



- $(0,0,0) = \text{Black}$
- $(255,0,0) = \text{Red}$
- $(0,255,0) = \text{Green}$
- $(0,0,255) = \text{Blue}$

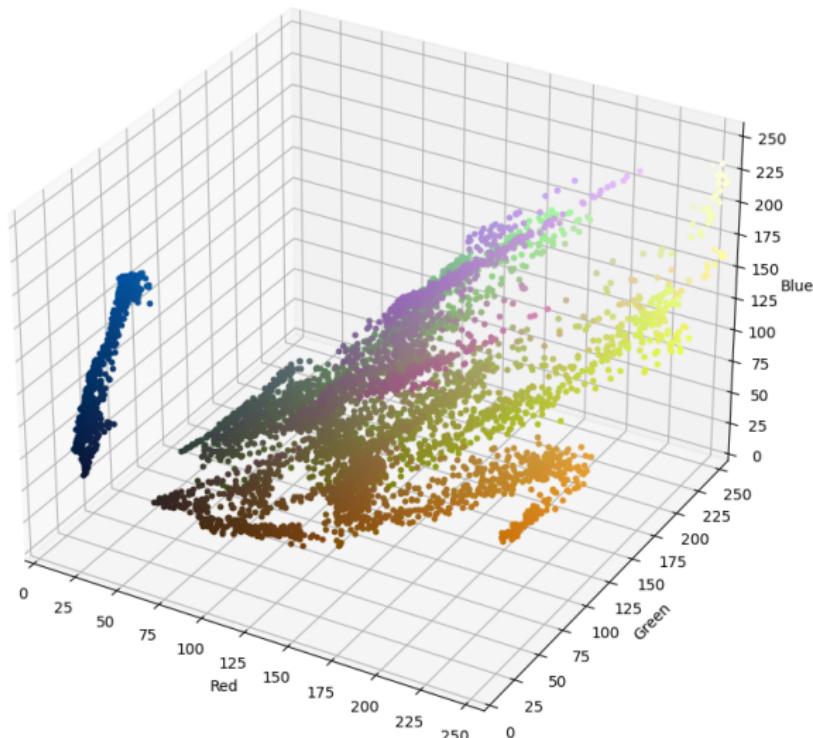
Example

- The image on the left has a color depth of 24 bits, giving it $2^{24} = 16,777,216$ possible colors.
- The second image contains the 10 most representative colors.



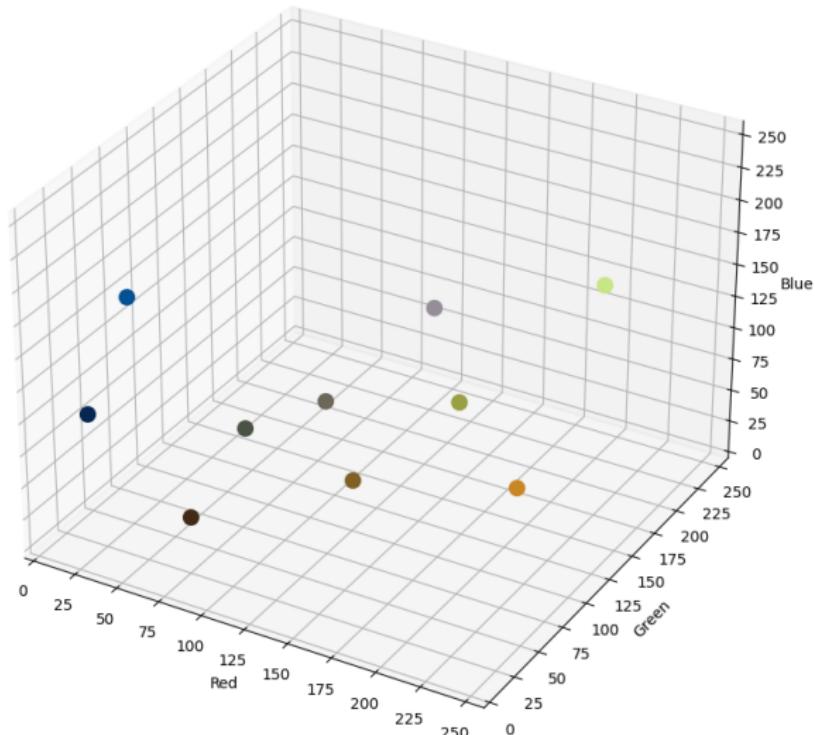
Example in Three-dimensional Space (1/3)

Three-dimensional representation of a subset of RGB pixels:



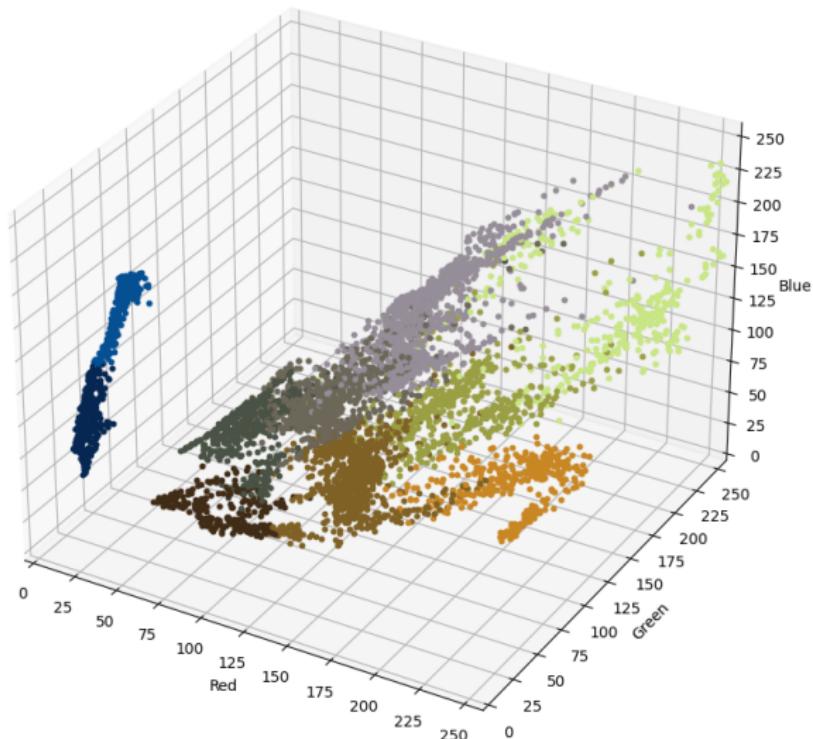
Example in Three-dimensional Space (2/3)

Calculated centroids by the K-means algorithm: (K=10):



Example in Three-dimensional Space (3/3)

Pixels colored according to the cluster they belong to:



Pseudocode

(Step 1) Select the value of K (number of clusters)

(Step 2) Initialize the K centroids randomly

convergence = False

while (!convergence)

(Step 3.a) Calculate the distance of each pixel to each cluster

(Step 3.b) Assign each pixel to the nearest cluster

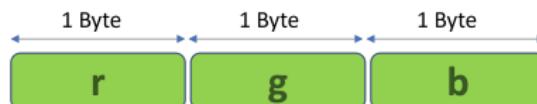
(Step 4) Update the centroids with the mean of the r, g, b values of all the pixels belonging to it

 if (all centroids remain the same)

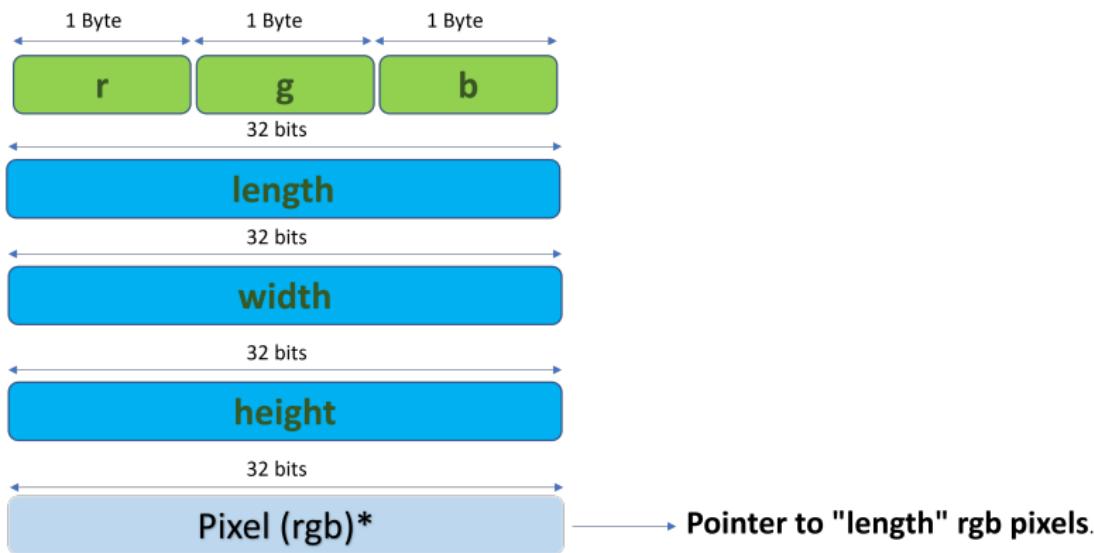
 convergence = True

Data representation (1/2)

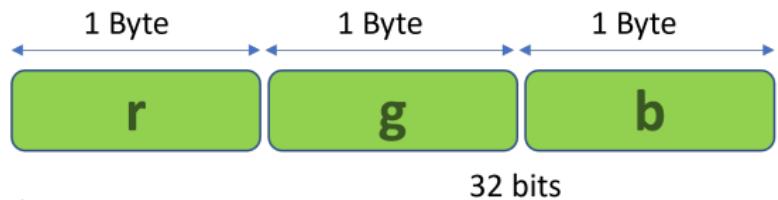
pixel (rgb)



image



Data representation (2/2)

cluster**Centroid pos:****Num_points:**

Points attached to this clu

Mean: