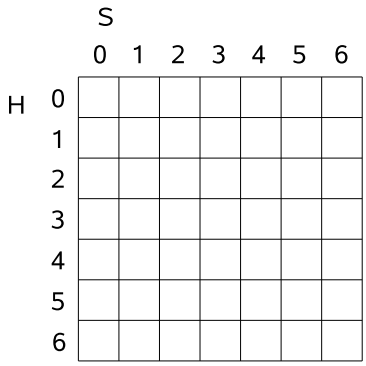
1 Pre-Recorded Tasks

1.1 K-Means Clustering

Some pixel values of an image in HSV space are given:

k		1	2	3	4	5	6	
$H[k]$	$=[$	1	1	2	4	4	5	$]$
$S[k]$	$=[$	6	4	4	1	2	2	$]$

1. Draw a 2-D diagram with hue and saturation as coordinates and mark the positions where the input image contains value pairs with '×'!



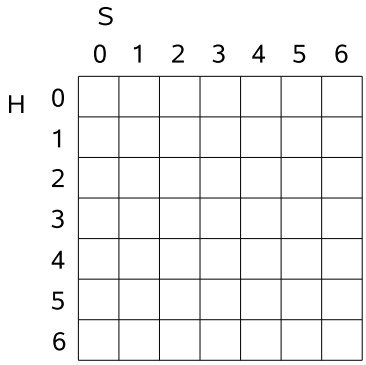
2. K-Means clustering with $K = 2$ clusters shall be performed. The initial centroids of the clusters A and B are given as $\mathbf{c}^{(A)}_{start}[H, S] = (5, 1)$ and $\mathbf{c}^{(B)}_{start}[H, S] = (4, 3)$. Mark the cluster centroid locations with '∘' and draw the nearest neighbor cluster border to the diagram!

3. Perform the nearest neighbor classification based on the cluster centroids $\mathbf{c}_{start}$. Decide for each data point k to which class (A or B) it belongs and write down the results in the table below!

k		1	2	3	4	5	6	
$H[k]$	$=[$	1	1	2	4	4	5	$]$
$S[k]$	$=[$	6	4	4	1	2	2	$]$
Assigned class:								

4. Perform the next step of the k-means algorithm and calculate the new cluster centroids $\mathbf{c}^{(A)}_{step1}[H, S]$ and $\mathbf{c}^{(B)}_{step1}[H, S]$!

5. Draw a new diagram with the original data points and the new cluster centroids $\mathbf{c}_{step1}^{(A)}[H, S]$ and $\mathbf{c}_{step1}^{(B)}[H, S]$. Draw the new nearest neighbor cluster border as well!



6. Perform the nearest neighbor classification based on the new cluster centroids $\mathbf{c}_{step1}$. Decide for each data point k to which class (A or B) it belongs and write down the results in the table below!

k		1	2	3	4	5	6	
$H[k]$	$=[$	1	1	2	4	4	5	$]$
$S[k]$	$=[$	6	4	4	1	2	2	$]$

Assigned class:

2 Self-Study Matlab Tasks

2.1 Chroma Keying

The term *keying* refers to the replacement of a certain color with a new background. We want to focus on creating a digital mask that tells apart the foreground from the background. In the following, we use color information only. Some images are provided in `~/SHARED_FILES/vcc/Ex8/`.

1. Load a greenscreen image from the given files. At first, all processing shall be done in RGB color space. Use the red color channel and extract a histogram with 256 bins. Use the MATLAB function `hist(img_red(:), 256)`!

   We now want to extract a typical color value of the background. We can assume that the upper left corner shows only background. Calculate the mean value of each color channel for a 20×20 region!

2. Use a single color channel and its corresponding mean value for segmentation of the image. Pixels with values close to the background color should be assigned to the background, whereas pixels with a larger color distance should belong to the foreground. Can you find a suitable distance that can be used for single color segmentation? Display the resulting mask!

3. Now, we want to segment the image based on two color channels. A pixel only belongs to the background if it is within a specific color range in both channels. Find a reasonable distance and display the resulting mask!

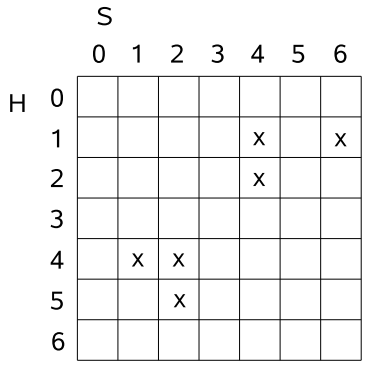
1 Pre-Recorded Tasks

1.1 K-Means Clustering

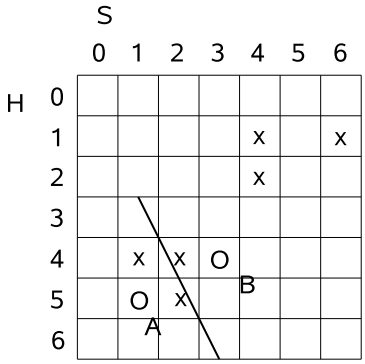
Some pixel values of an image in HSV space are given:

k		1	2	3	4	5	6	
$H[k]$	$=[$	1	1	2	4	4	5	$]$
$S[k]$	$=[$	6	4	4	1	2	2	$]$

1. Draw a 2-D diagram with hue and saturation as coordinates and mark the positions where the input image contains value pairs with '×'!



2. K-Means clustering with $K = 2$ clusters shall be performed. The initial centroids of the clusters A and B are given as $\mathbf{c}^{(A)}_{start}[H, S] = (5, 1)$ and $\mathbf{c}^{(B)}_{start}[H, S] = (4, 3)$. Mark the cluster centroid locations with '○' and draw the nearest neighbor cluster border to the diagram!



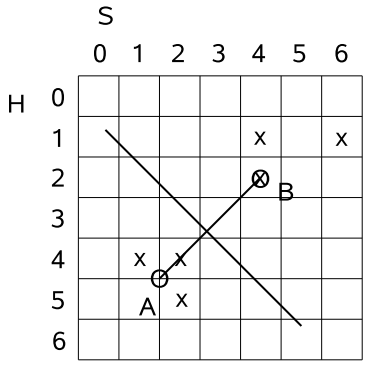
3. Perform the nearest neighbor classification based on the cluster centroids $\mathbf{c}_{start}$. Decide for each data point k to which class (A or B) it belongs and write down the results in the table below!

k		1	2	3	4	5	6	
$H[k]$	$=[$	1	1	2	4	4	5	$]$
$S[k]$	$=[$	6	4	4	1	2	2	$]$
Assigned class:		B	B	B	A	B	A	

4. Perform the next step of the k-means algorithm and calculate the new cluster centroids $\mathbf{c}^{(A)}_{step1}[H, S]$ and $\mathbf{c}^{(B)}_{step1}[H, S]$!

$$\mathbf{c}^{(A)}_{step1}[H] = \tfrac{1}{2}(4+5) = 4.5 \qquad \mathbf{c}^{(B)}_{step1}[H] = \tfrac{1}{4}(1+1+2+4) = 2$$
$$\mathbf{c}^{(A)}_{step1}[S] = \tfrac{1}{2}(1+2) = 1.5 \qquad \mathbf{c}^{(B)}_{step1}[S] = \tfrac{1}{4}(2+4+4+6) = 4$$
$$\mathbf{c}^{(A)}_{step1}[H,S] = (4.5, 1.5) \qquad \mathbf{c}^{(B)}_{step1}[H,S] = (2, 4)$$

5. Draw a new diagram with the original data points and the new cluster centroids $\mathbf{c}^{(A)}_{step1}[H,S]$ and $\mathbf{c}^{(B)}_{step1}[H,S]$. Draw the new nearest neighbor cluster border as well!



6. Perform the nearest neighbor classification based on the new cluster centroids $\mathbf{c}_{step1}$. Decide for each data point k to which class (A or B) it belongs and write down the results in the table below!

k		1	2	3	4	5	6	
$H[k]$	= [	1	1	2	4	4	5	]
$S[k]$	= [	6	4	4	1	2	2	]
Assigned class:		B	B	B	A	A	A	

2 Self-Study Matlab Tasks

2.1 Chroma Keying

The term *keying* refers to the replacement of a certain color with a new background. We want to focus on creating a digital mask that tells apart the foreground from the background. In the following, we use color information only. Some images are provided in `~/SHARED_FILES/vcc/Ex8/`.

1. Load a greenscreen image from the given files. At first, all processing shall be done in RGB color space. Use the red color channel and extract a histogram with 256 bins. Use the MATLAB function `hist(img_red(:), 256)`!

   We now want to extract a typical color value of the background. We can assume that the upper left corner shows only background. Calculate the mean value of each color channel for a 20×20 region!

2. Use a single color channel and its corresponding mean value for segmentation of the image. Pixels with values close to the background color should be assigned to the background, whereas pixels with a larger color distance should belong to the foreground. Can you find a suitable distance that can be used for single color segmentation? Display the resulting mask!

3. Now, we want to segment the image based on two color channels. A pixel only belongs to the background if it is within a specific color range in both channels. Find a reasonable distance and display the resulting mask!