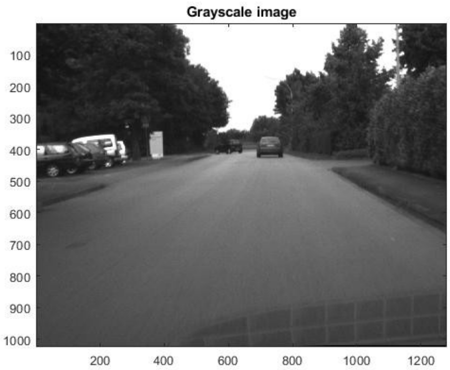
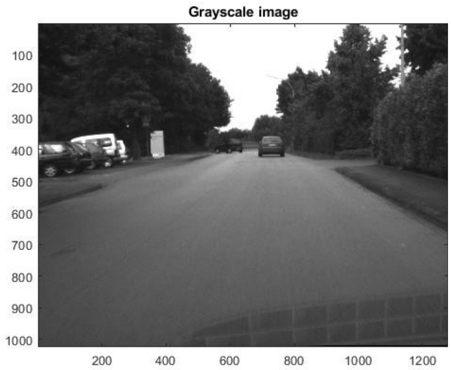
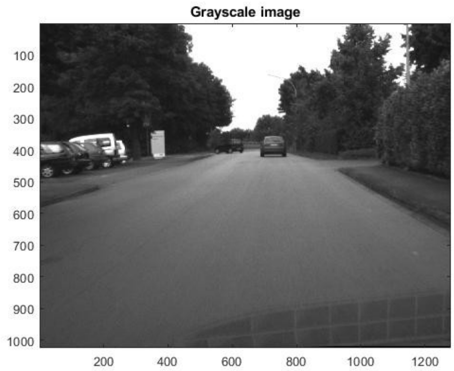
Computer Vision Lab session n. 4
09/04/2018

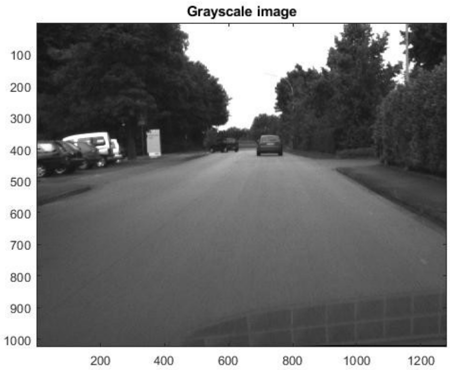
Color-Based Segmentation

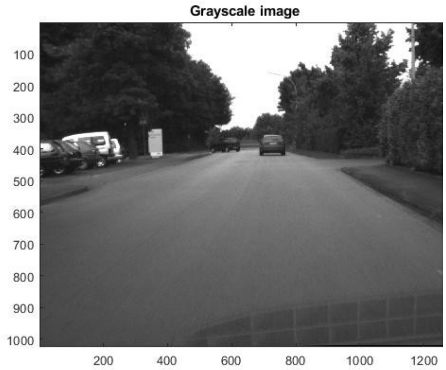
The aim of this lab session is to study and evaluate an image in rbg and hsv colormap. Then we must detect the red cars using the hue value of the original frames

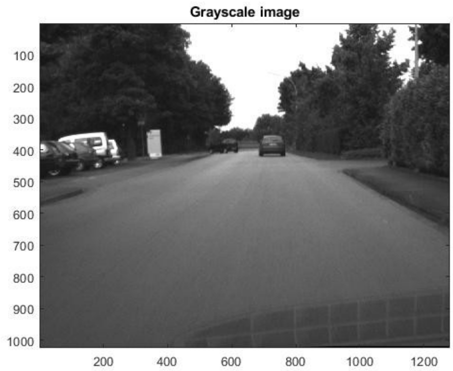






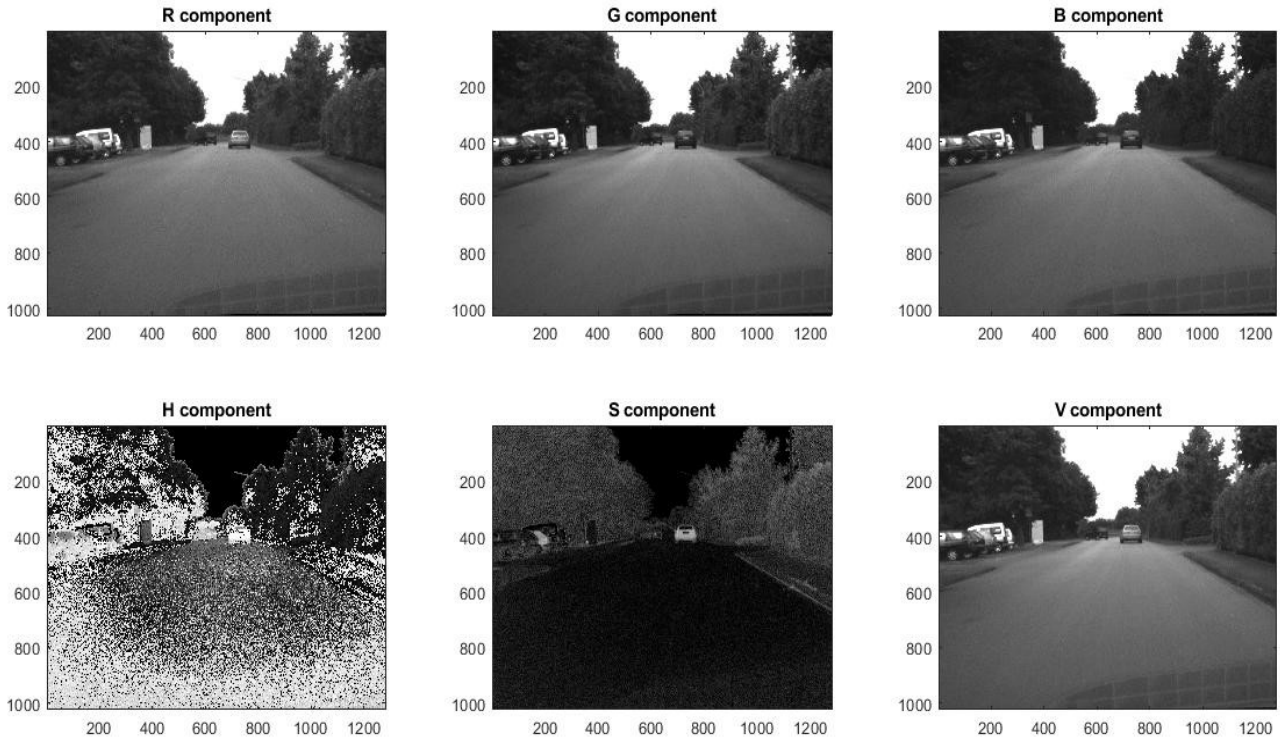






Davide Torielli
Luca Morando
Fabio Fusaro

Colormaps (colorMaps.m)
In this function we simply use rgb2gray and rgb2hsv to convert the image in different colormaps and show the single channels (red, green, blue and hue, saturations, intensity).

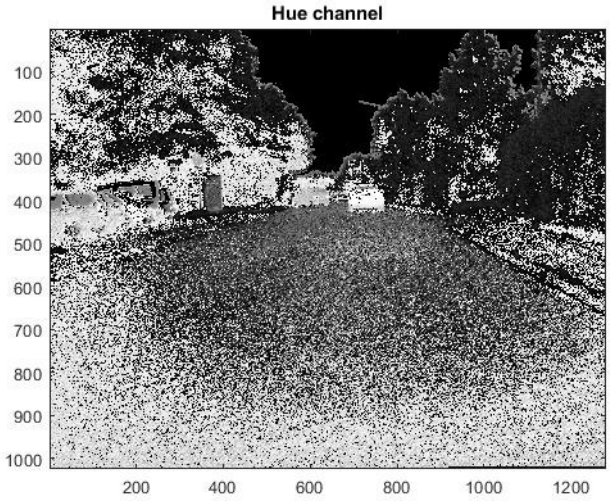


In the above picture is displayed the first frame (0376).
To detect car for the next step the best is to choose the hue channel because it is the one which differentiate better the colors. The saturation channel is better only in this case because the colored things in the original image are only the two cars (we would have seen objects of different colors all mapped in white)

Davide Torielli
Luca Morando
Fabio Fusaro

Studying the hue values for red pixels (maskHue.m)

Looking only the hue channel, we detect the cluster which correspond to red pixel (the car's one)

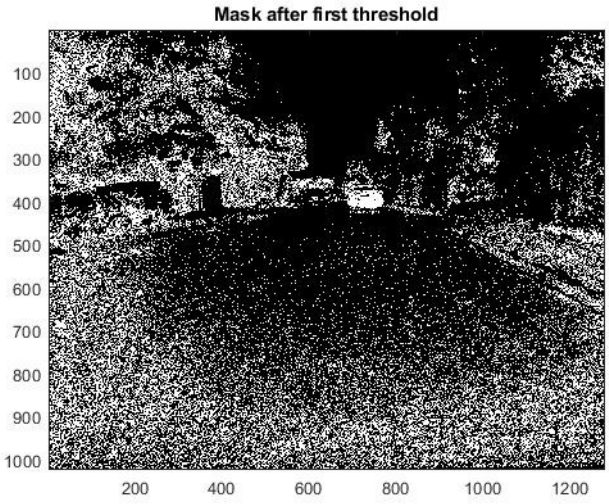


We use two methods:

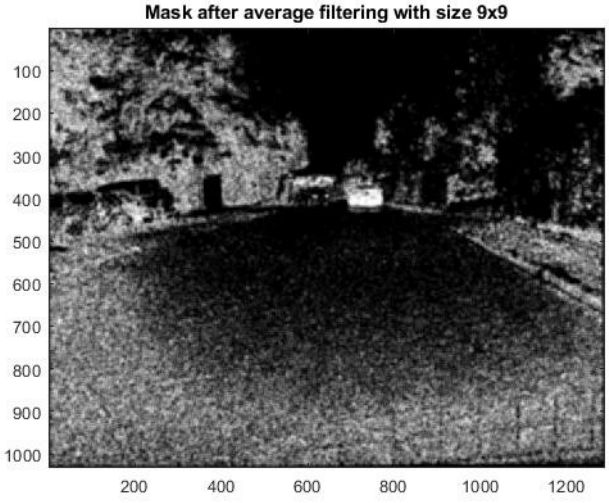
1. The first is bit "hardcoded": we choose a threshold of 0.98 based on the fact that the color that we have to detect is red, that is mapped as value near 1 in the hue channel.
   We simply use a boolean mask that has 1 if pixel's hue is greater than 0.98 (pixel red detected) and 0 if not.

2. With this method theorically should be possible detect the car also if it was of another color because we choose a range for the threshold based on the color of the car.
   First we choose the area of the image where the car is (simply taking right rows and colums of the image). In particular we try to take the pixel's car where the red is stronger (a sort of color picker).
   For these pixels, we take the mean and the standard deviation and put 1 in the mask if the original hue value is between mean-0.2*standard_deviation and mean+2*standard_deviation.

Davide Torielli
Luca Morando
Fabio Fusaro



The problem with this method is that the threeshold is not so precise as the first method, so we have more noise. To solve this, we use a moving average filter to make single white sparse point more “gray” and to keep the dense white pixels of the car “more” white than the others.
A good size of this filter for those images is 9x9.

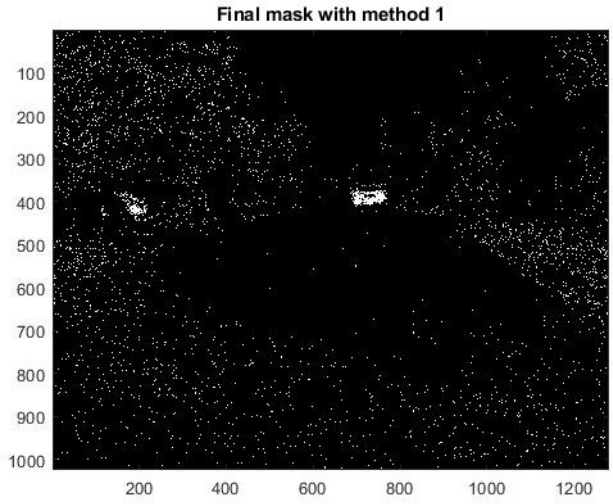


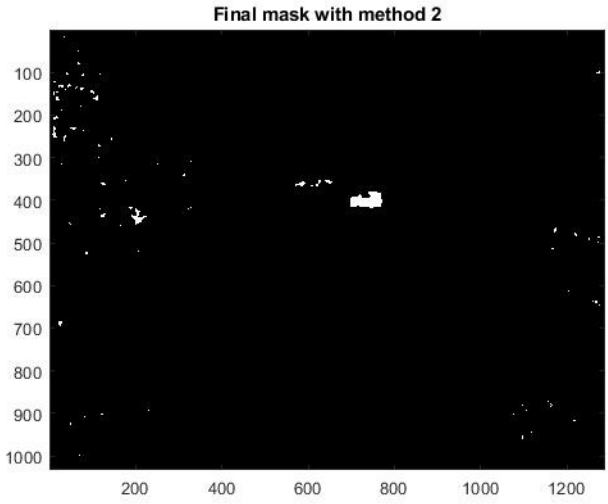
We see that the car is clearer respect to other points.
Finally, we make again the mask with a lower threshold (mean-0.5*standard_deviation as lowthreshold) because after the average filter all the pixels are a little further from white (1 value)

Another problem with this method is that the car is moving in the different frames, so we should take different rows and different columns for the

calculation of mean and standard deviation each time the frame changes. In the our code, we pick an area which is pretty good for all frames.

Here the comparison between the two methods:





Davide Torielli
Luca Morando
Fabio Fusaro

DetectCar (detectCar.m)

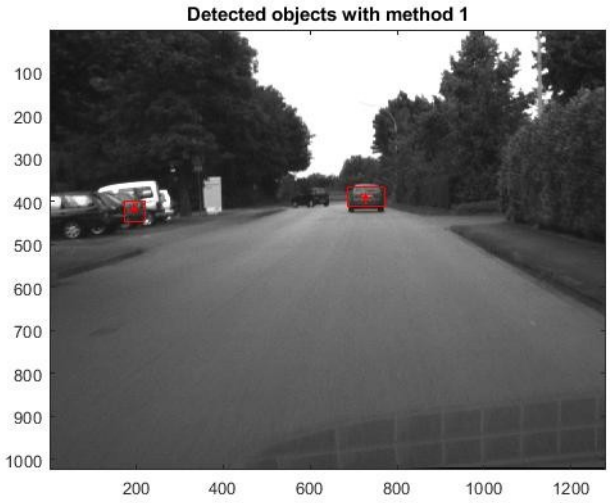
We now have a good mask with clusters of point. To finally detect the real car, we use the matlab function regionProps(), to find Area, Centroid and BoundingBox of clusters of white pixel.

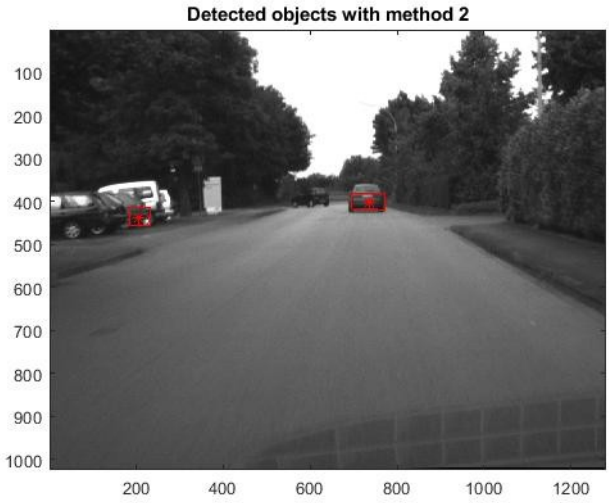
The detectCar function takes the minClusterArea as parameter, so cluster which have too little Area are discarded.

For the frame 0376 with Area greater than 200 we detect both red car, with Area greater than 550 we detect only the car on the road, because it is a cluster of more "red" pixels.

Note that with different frames it is possible that area of clusters changes, so this minClusterArea are not so precise. However with value 200 we detect the car in the road (the important one) in all frames.

In the images below we study the 0376 frame with minClusterArea 200.





Davide Torielli
Luca Morando
Fabio Fusaro