Computer Vision: Lab Session n.4
Color-based segmentation

Francesca Canale 4113133
Filippo Gandolfi 4112879
Marco Giordano 4034043

16/04/2019

1 Introduction

The aim of this fourth lab is to study the color based segmentation on different frames and then to detect the red car in the images studying the Hue values of the original frames.
The process of partitioning a digital image into multiple segments is defined as image segmentation. Segmentation aims to divide an image into regions that can be more representative and easier to analyze. Such regions may correspond to individual surfaces, objects, or natural parts of objects. Typically image segmentation is the process used to locate objects and boundaries (e.g., lines or curves) in images. Color image segmentation that is based on the color feature of image pixels assumes that homogeneous colors in the image correspond to separate clusters and hence meaningful objects in the image. In other words, each cluster defines a class of pixels that share similar color properties. To be able to describe colors, we need to know how people respond to them. The simplest question is to understand which spectral radiances produce the same response from people under simple viewing conditions.

We have been given six different frames to work with, below we report the grayscale images obtained with MATLAB function *rgb2gray()*:

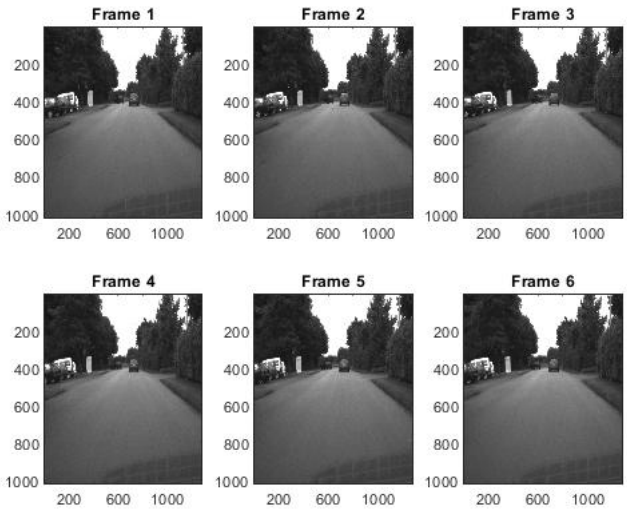

Figure 1: Grayscale images

2 Images channels

Color spaces are oriented either towards hardware or towards applications:

- RGB model is related to color monitors;
- CMY model to color printers;
- HSV model that corresponds closely with the way humans describe and interpret color.

In order to split the images in the three RGB and HSV channel we implemented a function *dividing_channel()*: the three RGB channels are already contained in the original images and regarding the HSV channels it uses MATLAB function *rgb2hsv()* that converts an RGB colormap to an HSV colormap. The resultant images obtained for the first frame are shown in subsections 2.1 and 2.2, it is possible to see also the images obtained for the others frames in the folder images.

2.1 RGB channels

The RGB color space is a linear color space that formally uses single wavelength primaries:

- 645.16nm for Red;
- 526.32nm for Green;
- 444.44nm for Blue.

The main purpose of the RGB color model is for the representation and display of images in electronic systems, as computers, tv and photography.

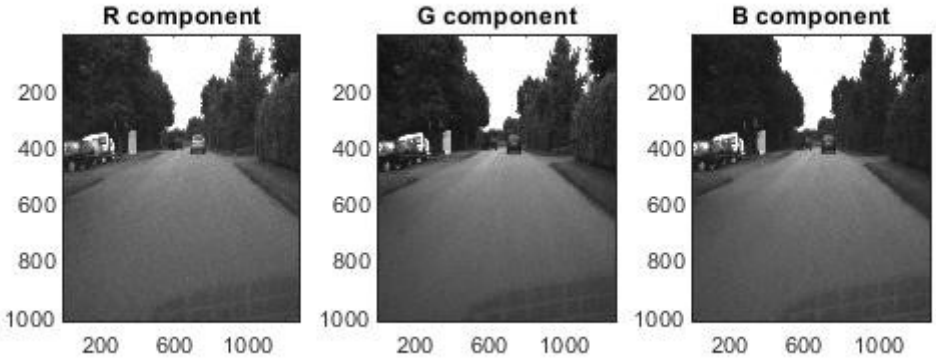

Figure 2: RGB channels of frame 1

2.2 HSV channels

The coordinates of a color in a linear space may not necessarily encode properties that are common in language or are important in applications. Perceptually meaningful dimensions:

- Hue, the property of a color that varies in passing from red to green;
- Saturation, the property of a color that varies in passing from red to pink;
- Value, the property that varies in passing from black to white.

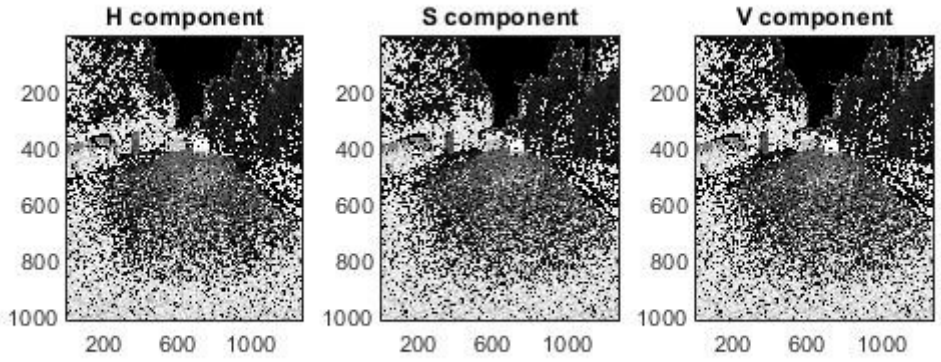

Figure 3: HSV channels of frame 1

3 Evaluating Hue component values

In order to evaluate the Hue component of the red car area we implemented a function *hue_evaluation()*. Firstly, we selected in the first frame image "ur_c_s_03a_01_L_0376.png" the area corresponding to the red car. Then we computed the mean using function *mean2()* and the standard deviation with function *std2()*.
This is useful for the next steps because it will allow us to recognize in the images blobs with similar properties.

4 Segmenting the red car

The shape of an object can be described either in terms of its boundary or in terms of the region it occupies.

- Region-based representation requires image segmentation in several homogeneous region;
- Image regions are expected to have homogeneous characteristics (e.g. intensity, color) that are different in each region;
- Thus, edge detection and region segmentation are dual (complementary) approaches in image analysis.

In order to segment the red car in the images we implemented a function *segmentation()* that creates a mask. Using the mean and standard deviation already computed it puts a '1' in the mask if the original Hue value of the images is between: $mean - 0.2 * standard\_deviation$ and $mean + 2 * standard\_deviation$. At this point this is the image that we obtained for the first frame:

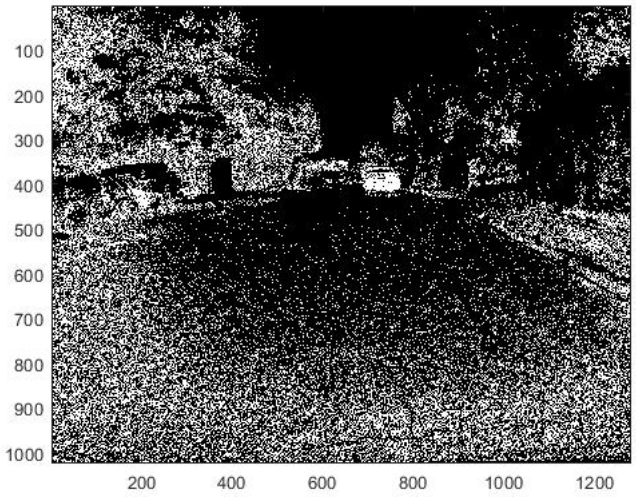

Figure 4: Mask after first threshold of frame 1

The problem is that the algorithm that we used to compute the threshold is not optimal and so the resultant images present a lot of noise.
To avoid this behaviour we used 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.

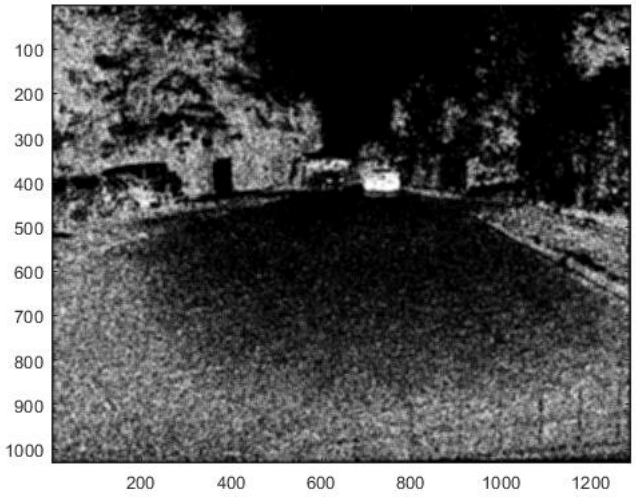

Figure 5: Mask after moving average filter of frame 1

We can see that the car is clearer respect to other points. Finally, we make again the mask with the same values of threshold computed before.

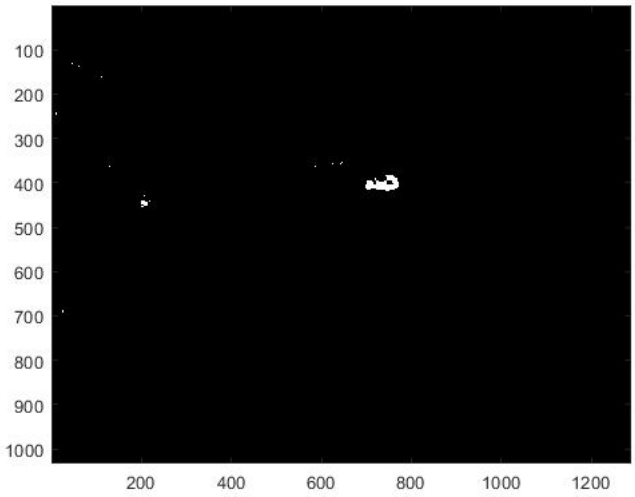

Figure 6: Final mask of frame 1

The resultant images for the other frames are available in the folder images.

5 Car detection

To finally detect the red car we implemented a function *detect_car()*. It uses MATLAB function *regionprops()* that finds Area, Centroid and BoundingBox of blobs of white pixel. In order to discard small blobs we take in consideration only the ones that have area greater than 300. Finally we plot a $*$ on the position of the centroid and a red boundingbox.

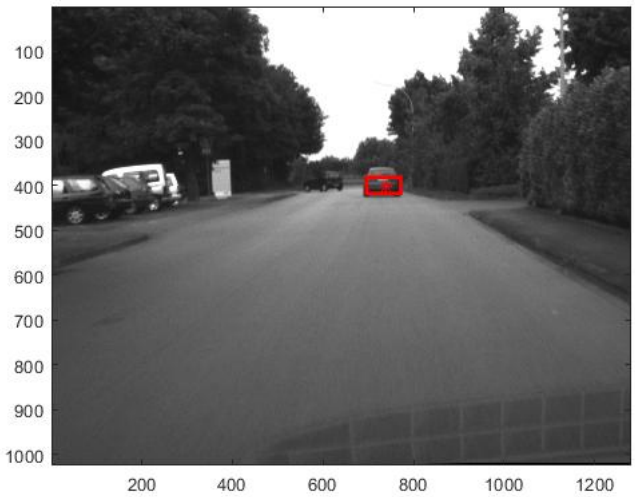

Figure 7: Detected red car first frame

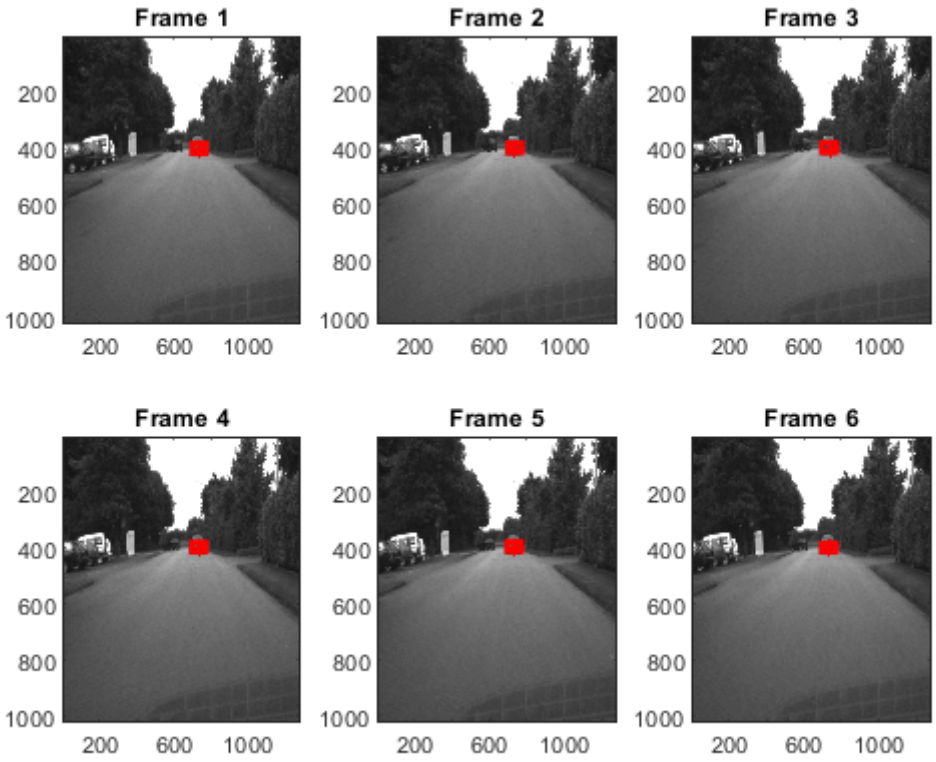

Figure 8: Detected red car all frames