Università degli studi di Genova

DIBRIS

Informatics, Bioengineering
Systems Engineering, Robotics and System Engineering

Computer Vision

Assignment 2

Report Laboratory 5

Student:
Andrea Chiappe,
Fabio Guelfi

November 15, 2023

1 Note

This report and the Matlab code is the same for Andrea Chiappe and Fabio Guelfi

Abstract

In Robotics and many other engineering branches, there is the necessity to analyse images and enable better understanding and manipulation of specific elements within the images.
Two important techniques are Normalized cross correlation-based segmentation and the Harris Corner Detection to identify and locate with great accuracy specific objects or points also in more complicated situation, when color based segmentation or other methods can fail.

Introduction

This lab's goal is to implement two image manipulation methods, the Normalized cross correlation-based segmentation and the Harris Corner Detection, important for the extraction of meaningful information from images. Both techniques are resistant to many variation parameter of the image.
The NCC is useful for locating specific **object** in different images because it can be resistant to different variation of the parameters,as lighting, motion, scale, and deformations of the object, during a video (sequence of images).
The Harris Corner Detection, instead, is useful for matching the key points in an image, that can be used, for example, in the comparison with a picture different form the previous one for scaling, rotation, translation or different point of acquisition.
These Key point are the **corners**, that are point of high local intensity variation on the picture.
The main scope is to have high performance techniques also when the image hasn't high quality or has environment imperfection.

This lab is divided into two points:

- In the first point we have six different images and need to create a window around one car, apply a normalized cross correlation (NCC) to detect the car inside the images. Also we need to analyze the result of matching and compare it with the color segmentation methods done in Lab4 (previous one).

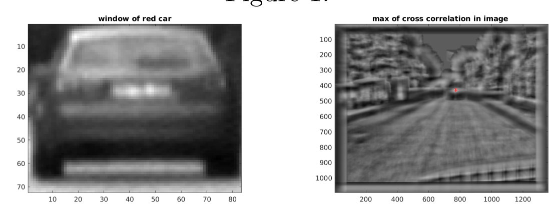
- The second point is about the Harris corner detection and apply it on the i235.png image to find all the interesting point of the image where there are changing of intensity.

Methods

NCC Based Segmentation

The first objective is to recognize the red car inside the six images. For doing this we chose a window around the red car in the first image of set *ur_c_03a_01_0376.png*. The size of the window we chose is: $72x83$ around the red car, showed in the left of Figure 1.After choosing the window we do the norm cross-correlation of the window with the image. For this operation in *matlab* we use the function $normxcorr2$. This functions return the matrix with correlation coefficients. The coefficient with high value correspond to the part of image more similar at the template. In the right part of Figure 1 is shown the result of $normxcorr2$ with the maximum of the matrix. In this case the maximum of cross-correlation is perfect coincident with the center of the template we chose.

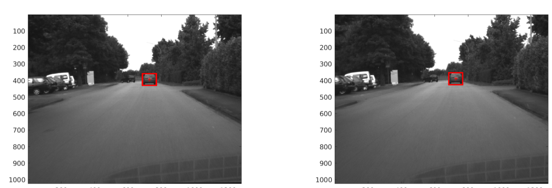
Figure 1:



(a) (b)

After testing the window on the first image, we do the cross-correlation of the template with other six images. The result of the detection of the first and the sixth images are shown in Figure 2

Figure 2:



(Image1) (Image6)

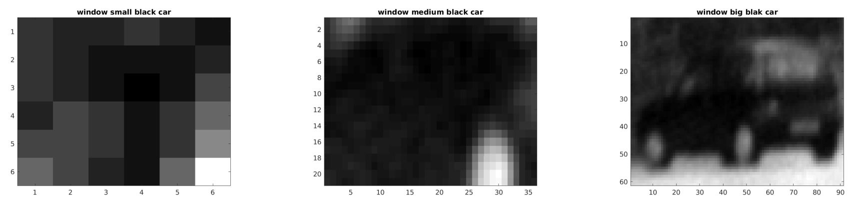
Now we do again all the procedure for detect the black car turn on left. In this case we will use three different windows, and we compute the different computational time. The dimension of the windows we chose are:

1. $5x5$ pixels
2. $20x35$ pixels

3. $60x90$ pixels

The windows are showed in Figure 3

Figure 3:



(5x5 pixel) (20x35 pixel) (60x90 pixel)

We do the normalized cross-correlation of this tree windows with all the 6 images of our data set. We compute the sum and the mean of the computational time for each window. We obtain different result from this test.

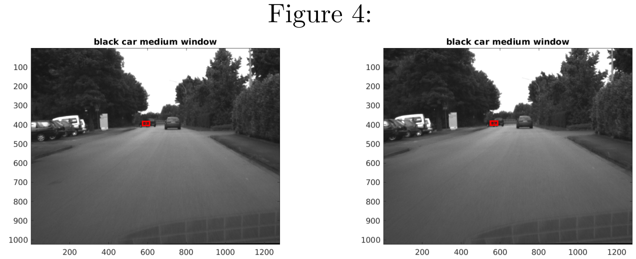
The smallest window is the faster about the computational time (mean is about only 0.131sec), but, for our choice, it doesn't catch always the car. Upgrading of a few pixel this window, the car will be always detected.

The medium one detects the car in all the images and the computational time is slower with respect to the previous case and the mean is around 0.2421 seconds.

The last window we choose, also detects in all the images the black car, but it has the longest elaboration-time, in fact the mean of computational time is around 0.6050 seconds.

In the figure 4 there is the result of the NCC with the medium window because is the window with the better relationship between the accuracy in detection and the computational time than medium and big window.

Figure 4:



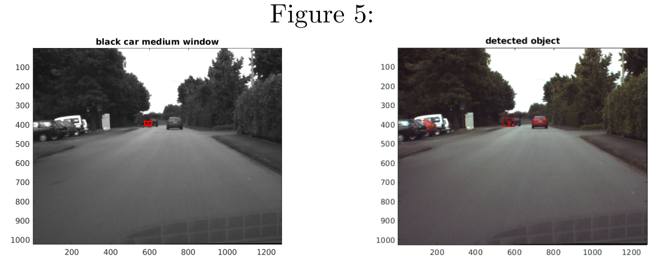
Compare NCC with color_based segmentation

Now the next passage of the lab is to compare the result obtain with the NCC with detection did with color based segmentation. We obtain two different result respect the two different car to detect.

- For the red car the result obtain from the two different technique are similar.
- For the detection of the black car we had a best result for the NCC technique.

The analysis of this part is about: the NCC technique is more efficient in the case I have the template of what need to find in the images. It is sensible to the change of point view point, but in our case, choosing a reasonable size of the template, and also the small movement of the car from first to last images, it work very good. So for all the images, choosing a reasonable template, it can recognize all the images the red and black car, and find all the time in the images, the center (maximum of cross-correlation) of the template we choose, so the detection is very good.
The color based segmentation worked also good about the detection of the red car, thanks that the red color is only the in the car, there isn't any red color in the background or in other part of the scene. So the algorithm cant' find red color in any other part of the figure. For the same reason, detecting the black car turning on the left is more difficult. The background of the black car is dark, near to grey and black. In this case the algorithm, based on color segmentation detect all the black part on the background of the car and it is less precise than NCC. In the Figure 5 are showed the result of detecting the black car, on the left the ncc with the medium window and on the right the color_based_Segmentation

Figure 5:



(normalized cross correlation) (color_based_segmentation)

Harris Corner detection

After that the *image i235* has been transformed in a gray-scale image, the goal is to compute the square of the derivatives on each pixel, also using the $Sobel's$

filters, and smoothing them with a *low pass Gaussian Filter*
Whit the Harris Detector formulas:

$$R = det(M) - k * (trace(M)^2)$$

where:

- R = Harris response for each pixel
- M = gradient matrix = $\begin{bmatrix} I_x^2 & I_y I_x \\ I_x I_y & I_x^2 \end{bmatrix}$
- k = parameter to rule the corner sensitivity

it is possible to compute the Regions map, that is the the matrix that contains the Harris Response of each pixel.

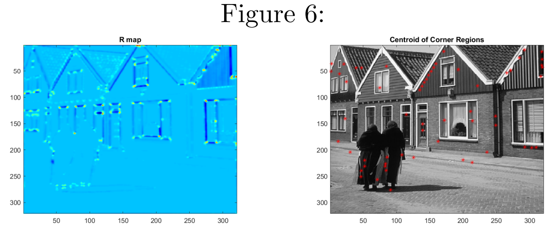
The goal is to subdivide each pixel between flat, edge and corner regions. This is possible comparing the R value to a threshold so:

- $R < -\ threshold$: Pixel to edge regions
- $R > threshold$: Pixel to corners regions
- Other : Pixel to flat regions

As threshold we used: $threshold = 0.3 * max(Rmap)$

So, the last step is to get the centroids of the blobs of the corner detected, through *regionprops()* function and show it as figure.

Figure 6:



(a) (b)

This images represents the map of the Harris corner results (a) and the centroids on the Harris Corner detected (b).

2 Conclusions

Now we can do the considerations of this laboratory.
The first consideration we can do is about the method to choose, the normalized cross correlation or the color based segmentation. For detecting the red car, the two technique have the similarity result.
The *NCC* catch perfect the red car, and the color segmentation has non problem to detect its, thanks the fact the red car have dark background, and there aren't any big area of red color in the images. Maybe if we had more movement of the car, from first images to the last one, the color based segmentation it would be more efficient. This fact because the cross correlation is sensible to the change of the point of view.
For the black car, the *NCC* works better than color segmentation. This happen because the background of the car is dark, with the similar color of black car, so it is less performing. The black car is turning left in our six test images but the cross correlation don't have any problem to detect its, also tanks the fact that the car don't move a lot from first image (where we take the template) to the last image. Another things to considerate is about the region that the to different algorithm detect. In the case o *NCC* it detect all the time the maximum cross correlation between the image an the template. After we are able to trace e rectangle round the car with the template's dimensions. With the color based segmentation every image have a different size of the window. It happen because the algorithm find the region with the most concentration of that color we put in input, so in every images the rectangle around the car has a different size.
An other consideration is about choosing the reasonable template's size for normalized cross correlation template. We test three different size and compute the computational time for each one. The relation between size and computational time is directly proportional: if size is big, it needs more computational time. We decide that the best window for detecting is the middle one. The smallest windows is very fast but not always catch the car, and the big one, although detect the car in all the test images, have a bigger computational time than the middle one.
Considering the Harris Corner Detection we can see some areas with a bigger density of corners detected, here there are important changes of intensity and meaningful gradients in more directions. These points mark salient features of the image, because there, as we notice, there are structures or things separated by the image
After the overlapping of the centroid of the corners and the original image, we think that in the image there are too much corners detected by the algorithm, so we think that taking a bigger threshold may be a good idea to have less meaningful points on the image and so to reduce the resources of computation usage in a next elaboration of the image. To have a bigger threshold there is the need to take a bigger scalar that multiplies *max(Rmap)*.
Adjusting the threshold it is possible to reduce the probability to have false positive or negative.