Computer Vision: Second Assignment (part 1)

Students Giacomo Lugano (5400573), Claudio Tomaiuolo (5630055)

Abstract

Template Matching is an important task in many computer vision applications. It is a high-level machine vision technique that identifies the parts on an image that match a predefined template[1].
Template Matching techniques are flexible and relatively straightforward to use, which makes them one of the most popular methods of object localization. Their applicability is limited mostly by the available computational power, as identification of big and complex templates can be time-consuming [2].
The normalized cross-correlation (NCC) method has high accuracy and strong adaptability, furthermore is routinely encountered in template matching algorithms, such as in facial recognition, motion-tracking, registration in medical imaging, etc.
In this case, we applied the NCC to 6 images representing a red car and a black car turning left. Furthermore, we compared Color-based segmentation and NCC-based segmentation. Then, we considered three windows of different sizes to analyze the efficiency in terms of accuracy and computation time.

Introduction

In signal processing, cross-correlation is a measure of the similarity between two series as a function of the shift of one relative to the other. It is a standard tool for assessing the degree of similarity between two signals. It finds application in pattern recognition, for matching two image patches and for detecting similar features [3].
The technique has several advantages. The first advantage is that cross-correlation is simple to calculate. When used to match a spot at a typical image location, Fourier methods can be used to quickly calculate cross-correlation. The second advantage is that the cross-correlation is independent of translations and scaling in the intensity domain. It is therefore quite independent of illumination variations.

$$C_{fg}(i,j) = \sum_{(u,v)\epsilon R_m} f(u+i, v+j)g(u,v) \tag{1}$$

$$R_m(i,j) = \left\{ u, v \mid x - \frac{m}{2} \leq u \leq x + \frac{m}{2}, y - \frac{m}{2} \leq v \leq y + \frac{m}{2} \right\} \tag{2}$$

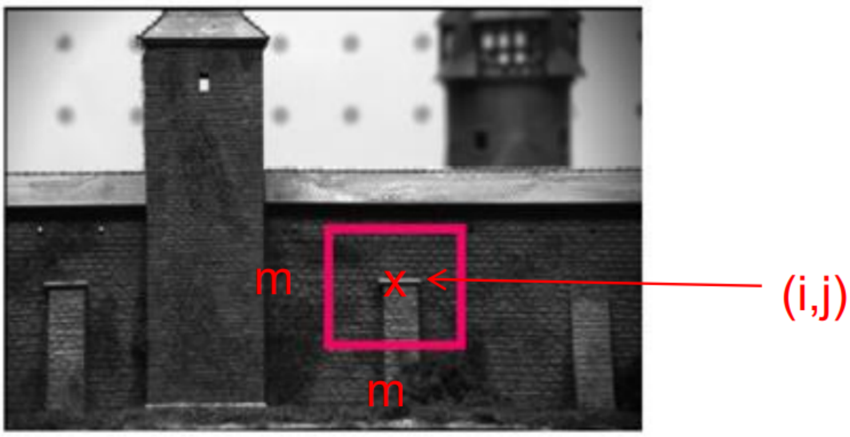

Figure 1: Simple intensity Matching: correlation

Normalized cross-correlation (NCC) has been commonly used as a metric to assess the degree of similarity (or dissimilarity) between two compared images. The main advantage of normalized cross-correlation over ordinary cross-correlation is that it is less sensitive to linear changes in the illumination amplitude in the two compared images. However, it has disadvantages: it cannot be calculated directly with the more efficient FFT (Fast Fourier Transform) in the spectral domain; its computation time increases dramatically as the size of the model window increases, as we will see in our final analyses.

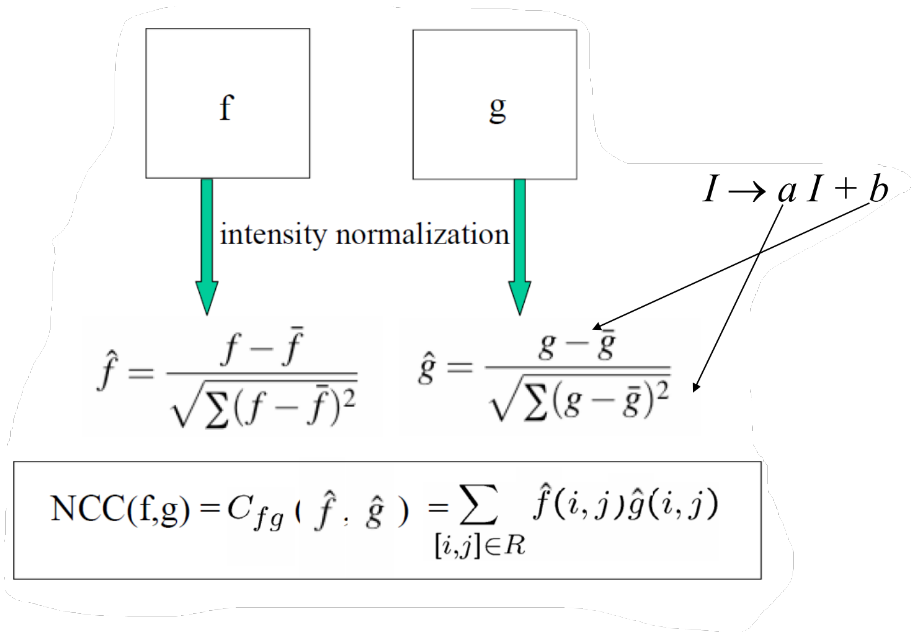

Figure 2: Normalized Cross Correlation

NCC is by definition the inverse Fourier transform of the convolution of the Fourier transform of two images, normalized using the local sums and sigmas. There are several ways of understanding this further, a very simple example is that this normalized cross-correlation is not unlike a dot product where the result is the equivalent to the cosine of the angle between the two normalized pixel intensity vectors.

Normalized Cross Correlation Algorithm Principle

Let S be the matching image with the size of M*N pixels, and T be the template image with the size of m*n pixels. The template image T slides on the matching image S, and the gray correlation value of images S and T is calculated by using the correlation function when the template image T slides to the position (u,v). When the correlation value reaches the maximum value, the position of the search window determines the position of T in S [2].

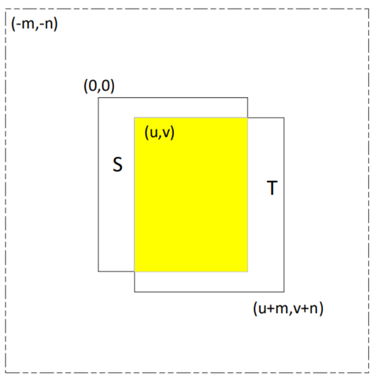

Figure 3: Schematic diagram of image matching principle

The algorithm calculates the degree of image matching between the two images through a normalized correlation measurement formula:

$$R(u,v) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[f(u+i,v+j)-\bar{f}_{u,v}][t(i,j)-\bar{t}]}{\sqrt{(\sum_{i=1}^{m}\sum_{j=1}^{n}[f(u+i,v+j)-\bar{f}_{u,v}]^2)(\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{i,j}[t(i,j)-\bar{t}]^2)}} \quad (3)$$

where f(i,j) is the pixel value of the matching image S in (i,j), t(i,j) is the pixel value of the template image T in (i,j), $\bar{t}$ is the average pixel value of the template image T, and /barf is the average pixel value of the corresponding image S under the current template image T.
Due to the high computational complexity of the above (3), a series of accelerating calculation methods can be adopted in practical application [4].
First, let t'(i,j)=t(i,j)- $\bar{t}$, then the molecule of (3) can be expressed as follows:

$$\sum_{i,j} f(u+i,v+j)t'(i,j) - \bar{f}\sum_{i,j} t'(i,j) \quad (4)$$

Obviously, the last term of (4) is always equal to zero, and the first term of (4) can be expanded to obtain:

$$\sum_{i,j} f(u+i,v+j)t(i,j) - \bar{t}\sum_{i,j} f(u+i,v+j) \quad (5)$$

The first term of (5) can be regarded as the convolution of two signals in the spatial domain, and the convolution for the spatial domain is equivalent to the coefficient multiplication operation based on the frequency domain. Therefore, the first term of (5) is equivalent to the following form:

$$F^{-1}[F(f)F^{*}(t)] \quad (6)$$

where, F is the Fourier transform of the original signal, F* is the conjugate complex operation of the transformed result, and F^{-1} is the inverse Fourier transform of the frequency domain signal.
Next, for the denominator of (3), the latter is the pixel value variance of the template image T. The first term can be simplified as follows:

$$\sum_{i,j}[f(u+i,v+j)]^2 - \frac{[\sum_{i,j} f(u+i,v+j)]^2}{m*n} \quad (7)$$

If (7) is calculated, it is necessary to obtain the cumulative sum of all pixel values and the sum of squared values in the sliding window of the matching image S at (u,v). It can be seen from (7) that direct calculation of correlation coefficient requires a lot of multiplications and additions. The larger the template image and the matching image are, the more additions and multiplications are needed, and the growth is rapid. When a new point is reached, parameters such as correlation value and sum of squares need to be recalculated, which is very wasteful. Therefore, combining the strategies of down sampling and dynamic programming, this paper further optimises the above calculation process.

Results

Considering the following image:

Figure 4: Original image

We read it as a grey scale image. Then, we select a window around the red car and around the black car in order to create the templates.

Figure 5: Black car template

Figure 6: Red car template

After that, we applied the *normxcorr2()* Matlab function to process the normalized cross-correlation between the template and the image.

Figure 7: Example of received result of Normalized cross-correlation function

As final output, we obtained (as shown in Figure N) the position of the red car template, as well as, the position of the black car template.

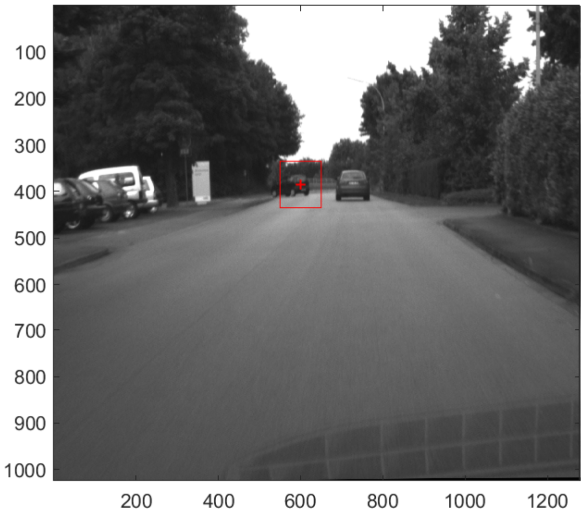

Figure 8: Position of the black car template

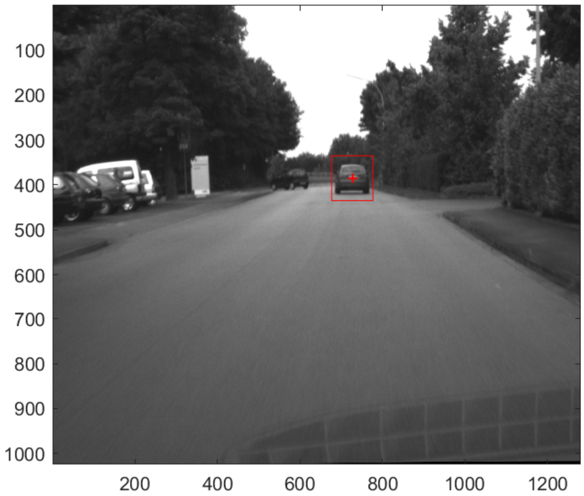

Figure 9: Position of the red car template

These results were obtained for all 6 images, both for the black car and the red car.

Comparing Color-based segmentation and NCC-based segmentation

As we can see from the pictures both the results are great in identifying the desired object, but we must consider that for what regards the color-based segmentation, we need to tune the corresponding mean hue value and related standard deviation with care, indeed the color-based segmentation is preferable in cases where we're looking for an object with a specific color that can't be found somewhere else in the picture, otherwise we could obtain a wrong match as we can see from the segmentation map figure of the red car, where also the parked car is highlighted.

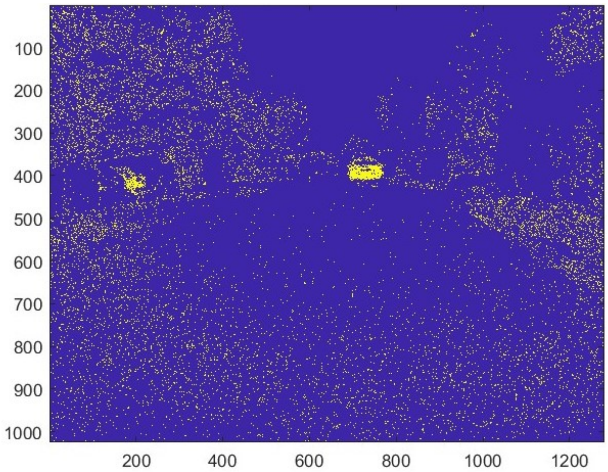

Figure 10: Red car segmentation map

Moreover, the color segmentation, tuned to identify as better as possible the position of an object requires that we consider a image patch on which evaluate the mean and standard deviation value of the hue component smaller respect the object; in this way we obtain a good match but parts of the objects will fall out this hue range causing the bounding box to not cover all the desired object.
For these reasons we think that the NCC-based segmentation is preferable respect the color-based one in all the cases in which the scene is color heterogeneous and the object hasn't a stand out color, or in which the bounding bow is meaningful.
In all the other cases we think that a combination of the two is preferable giving more importance to one or the other based on the color relevance of the object in the scene.

Conclusions

The computation time is a important aspect of Normalized Cross Correlation. So, three different window sizes were applied to find the processing times of each template, around the black car.

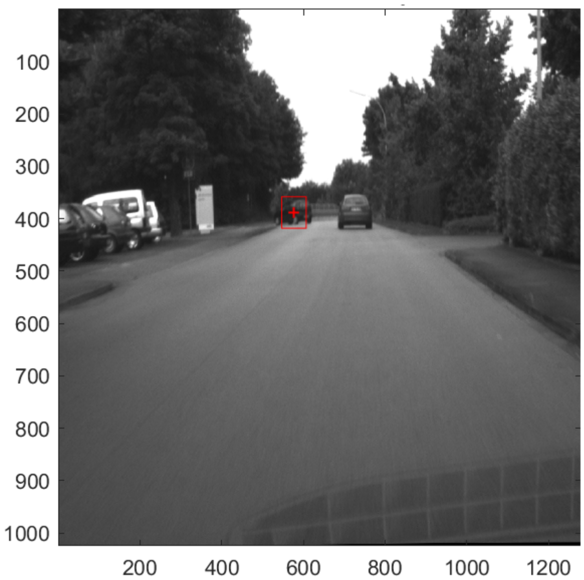

Figure 11: Small template

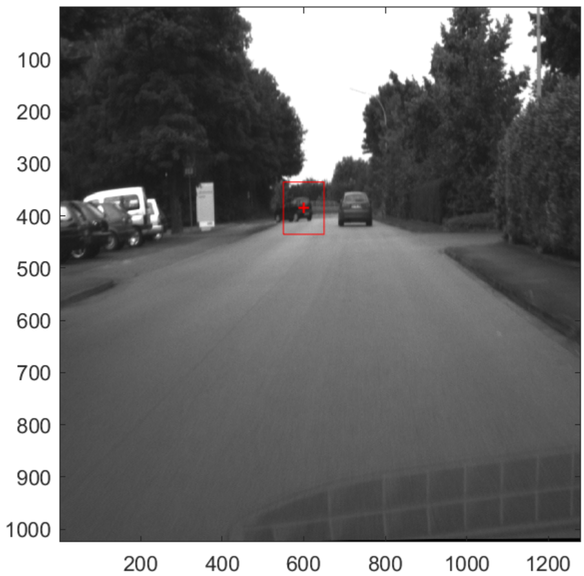

Figure 12: Medium template

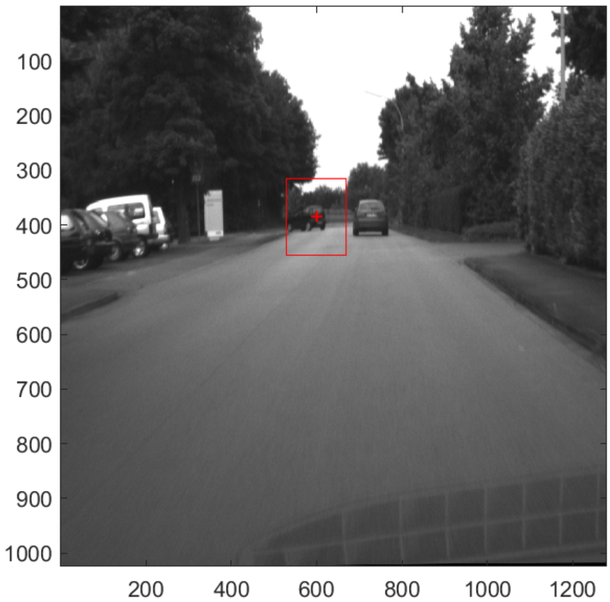

Figure 13: Big template

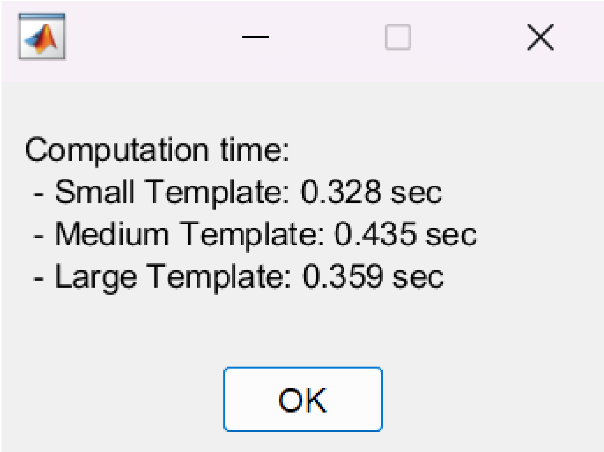

Figure 14: Computation time

From Figure 14, it can be deduced that as the size of the template increases, the computation time of the algorithm increases proportionally due to the greater number of pixels.

References

[1] Jinlian Hu (2008) *Fabric Testing*, Woodhead Publishing Series in Textiles.

[2] Zhongjie Cui (2020) *A Fast Image Template Matching Algorithm Based on Normalized Cross Correlation*, J. Phys.: Conf. Ser. 1693 012163.

[3] Bracewell R. (1965) *"Pentagram Notation for Cross Correlation." The Fourier Transform and Its Applications*, New York: McGraw-Hill, pp. 46 and 243.

[4] Qi Wenfa, Guo Wei (2018) *Robust Authentication for Paper-based Text Documents based on Text Watermarking Technology*, Mathematical Biosciences and Engineering, 16(4): 2233–2249.