UNIVERSITE DE François Rabelais

Ecole d'ingénieurs Polytechnique de l'Université de Tours

Memory Pattern Recognition Project

Presented in contest of

MASTER

Computer Aided Decision Support

by :

JARRAYA Hana

Characterization of the FFT-based correlation distortion for binary template matching

Supported the: 13/01/2015, in front of the jury:

DR. Patrick Martineau

Chairs: DR. Nicolas Ragot

Chairs: DR. Romain Raveaux

Supervisor: DR. Mathieu Delalandre

Abstract

Although it is well known that cross correlation can be efficiently implemented in the transform domain, the form of cross correlation preferred for template matching applications provides an alternative approach, Fourier Transform, using duality between correlation in spatial domain and multiplication infrequency space. As Known Fourier transform implementation can seep up the process of template matching. This short project presents the characterization of Fourier transform based template matching in binary form.

Content

Figures

Introduction

To validate the **Manga copyright protection**, M. Delalandre, M. Iwata and K. Kise proposed an approach to verify the copyright in [8]. This work is based on the similarity between the image (original) and its appropriate template (copy), called template matching process.

Template matching is a technique used in classifying an object by comparing portions of images with another image. Finding a given template in an image is typically performed by scanning the image and evaluating the similarity with the template. When the scanning is concerned with the entire image template matching is optimal. Near-duplicate Manga image Application considers a special case of template matching where the templates are binary. In order to perform the comparison between a template and a candidate sub-image, a function measuring the degree of similarity (or dissimilarity) is computed. When the images are obtained in a binary form, binary similarity functions can be applied [9]. These binary measures require significantly less resource compared to the ones working in the gray domain [10]. The computation is performed in absence of square, multiplications, summing up operations and floating-point coding required while computing similarity between gray-level images. In addition, the binary similarity measures operate as detectors presenting several interesting properties for object recognition, image registration or detection [11, 8].

In this project, we will review first some definitions related to template matching. We will present then a system exploiting Fourier transform processing supporting fast and accurate binary template matching for Manga copyright protection. This system is compared to a logical operator-based template matching.

This memory is organized into three chapters as follows: The first chapter, entitled state of art, will introduce in the first place concepts that we need to understand in order of rightly conducting our work and in the second a formulation of the template matching process in binary case detailing some its advantages. Subsequently we introduce the problematic of our project. In the second chapter we will explain the Fourier transform technique than the algorithm of Fourier transform based template matching. Finally, the third chapter will be devoted to the experimental evaluation of our work. It will allow us, in particular, an interpretation of the performance of the Fourier transform base binary template matching approach comparing to another measure.

Chapter I: State of art

In this chapter, we define correlation and convolution in the first place. We specify a specific type of comparison technique between images is the template matching. This type is based on different similarity measures. We choose in this project to present the similarity measure: Cross correlation derivative from Lpnorm. We define also the similarity measure on binary images which are the basic inputs in our project.

1. *Correlation definition:*

Correlation operation is a signal analysis concept basic to linear systems [1]. Spatial correlation is the process of moving a filter mask over the image and computing the sum of products at each location. In other words, the first value of correlation corresponds to zero displacement of the filter, the second corresponds to one unit displacement, and so on as shown in Fig1. In 2-Dimension, the correlation is between an image f(x,y) and a filter w(x,y) of size m*n, denoted w(x,y) ⊗f(x,y). it's given by equation form:

$$w(x,y) \otimes f(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

With a = (m-1)/2 and b =(n-1)/2 that m and n are two odd numbers. This equation is evaluated for all values of the displacement variables x and y and that all elements of w visit every pixel in f where f has been padded appropriately [2].

Padding: To perform correlation, the two functions f and w have to overlap in their parts. To guarantee that, padding f with enough 0s on either side to allow each pixel in w to visit every pixel in f. if the filter w is of size m*n, we pad the image with minimum of m-1 rows of 0s at the top and bottom and n-1 columns of 0s on the left and right [2]. As Fig1 shows padding a 2D image f with zeros at each side where (a) Image f(x,y) with size 5*5, (b) Filter w(x,y) with size 3*3 and (c) Padded f with 2 rows of 0s at the top and bottom and 2 columns of 0s at the right and left.

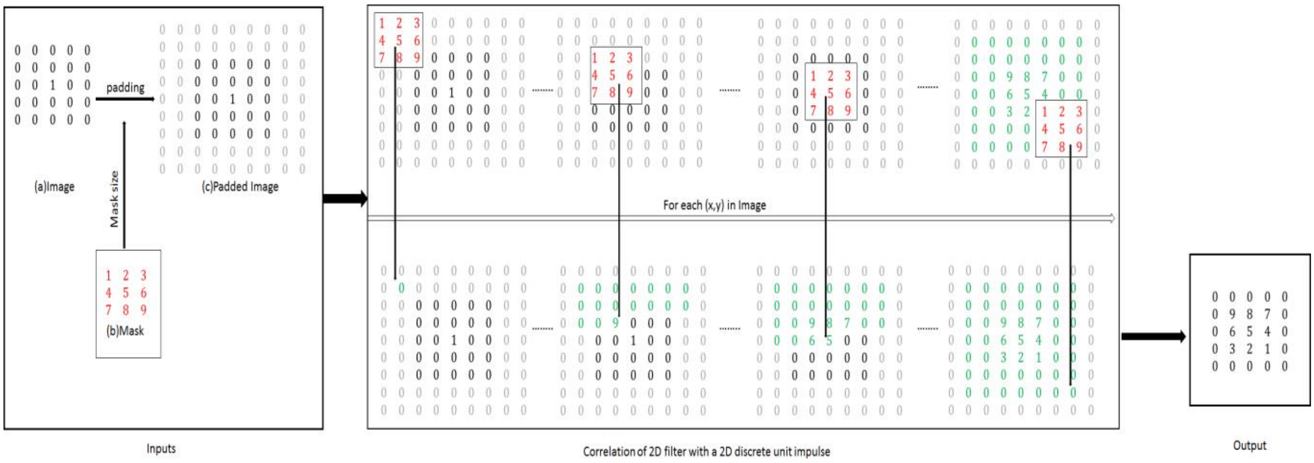

Figure 1 Correlation process

2. *Convolution definition:*

To perform convolution all we do is rotate one function by 180° and perform the same operations as in correlation. So, the filter mask can be rotated by 180° before sliding the sum of products just explained. Or, in a similar manner, the convolution of w(x,y) and f(x,y) denoted by w(x,y)* f(x,y), is given by the expression

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Where the minus signs on the right flip f (i.e., rotate it by 180°). Flipping and shifting f instead of w is done for notational simplicity and also to follow convention. As it turns out, it makes no difference which of the two functions we rotate [2], as Fig2 shows.

A fundamental property of convolution is that convolving a filter w with a unit impulse f that contains all 0s and a single 1 yields a copy of the function at the location of the impulse. Otherwise correlating w with a unit impulse f yields a result that is a copy of w, but rotated by 180°.

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

Figure 2 Convolution of 2D filter with a 2D discrete unit impulse

3. *Template Matching:*

Template matching ***TM*** is a comparison technique between a template ***T*** and a given image ***I***. It determines if T is contained in I and its position based on the pixel intensity value [3]. In other words, R.O. Duda and P.E.Hart, in [6], defined template matching techniques as they "attempt to answer some variation of the following question: Does the image contain a specified view of some feature, and if so, where?"

3.1.Template Definition:

In [4], M. S. Nexon and A. S. Aguado define a template T as" a sub image that contains the shape that we are trying to find"(Fig.4). Moreover, in [3], M. Storring and T. B. Moeslund said *"template is a copy of a part of an image which contains the object or a part of it. The template is also called reference template, correlation kernel, or correlation mask".*

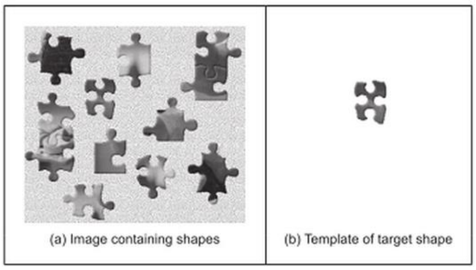

Figure 3 Illustrating Template matching [4]

3.2.Template Matching Process:

Template matching is conceptually a simple process to find image part in the whole image using distance similarity measure. We center the template on an image point and count up how many points in the template **matched** those in the image. The process is repeated for the entire image and the point which led to the best match, the maximum count, is deemed to be the point where the shape (given by the template) lies within the image [4].

For example, we define the template T of Fig.3 (b) and the image I of Fig.3 (a). T is first positioned at the origin and then matched with the image to give count which reflects how well T matched that part of I at that position. The count of matching pixels is increased by pone for each point where the brightness of T matches the brightness of I. This is similar to the process of correlation/convolution. The difference here is that's points in the image are matched with those in the template, and the sum is of the number of matching points as opposed to the sum of the products. The best match is the template T is placed at the position where the rectangle is matched to itself [4].

4. *L_p Norm Similarity measures based Template matching*

A template matching TM uses a measure of similarity between T and I. This is often called correlation. T is correlated with I in which the position of the object should be determined. Ideally the correlation should result in a peak positioned where the target is and no answer elsewhere. Depending on the correlation techniques a large or small value indicates high similarity. An example of the result of similarity measure is shown in Fig.4. Here the peaks indicates the position where the best match is obtained [3].

The template matching problem is to choose the template position that minimizes a given similarly measure between the template T and the image I at a given position (i,j) in the image with (i,j) in I [7]. Several measures of similarity are presented in the literature to calculate the correlation between the image and the template. In this work we define the squared Euclidian distance as the sum of squared differences SSD. The problem is the minimization of SSD given by the equation 1:

$$\min e= \sum_{x,y \in w} I(_{x+i,y+j} - T_{x,y})^2 \quad (1)$$

A common approach to compute similarity is to use the Lpnorm function within gray-level images Eq. (2). With p = 1 we get the Sum of Absolute Differences (SAD) while by taking p = 2 we get the Sum of Squared Distances (SSD).

$$\min e= \sum_{x,y \in w} I|(_{x+i,y+j} - T_{x+i,y+j})|^{p\,1/p} \quad (2)$$

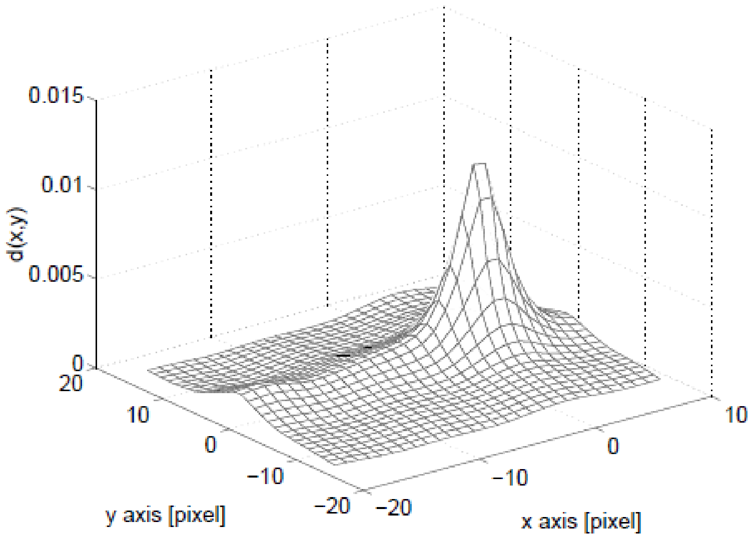

Figure 4 Result of template matching. D(x,y) is distance between T and I.

The axis x and y are position of pixel in space [3].

5. Cross correlation based template matching:

The squared error criterion can derive alternative forms, by considering that eq.1 can be written as

$$\text{Min } e= \sum_{x,y \in w} I^2{}_{x+i,y+j} - 2\, I_{x+i,y+j} T_{x,y} + T^2{}_{x,y} \quad (3)$$

The last term does not depend on the template position i,j. As such, it's constant and con not be minimized. Thus, the optimum in this equation can be obtained by minimizing

$$\text{Min } e= \sum_{x,y \in w} I^2{}_{x+i,y+j} - 2\, I_{x+i,y+j} T_{x,y} \quad (4)$$

If the first term

$$\sum_{x,y\in W} I^2_{x+i,y+j} \quad (5)$$

also is approximately constant, then the remaining term gives a measure of the similarity between the image and the template. That's, we can maximize **the cross correlation** between the template and the image. Thus, the best position can be computed by

$$\text{Max } e = \sum_{x,y\in W} I_{x+i,y+j} T_{x,y} \quad (6)$$

6. Binary Form Case:

A particular implementation of template matching is when the image and the template are binary. In this case, the binary image can represent regions in the image or it contain the edges (black and white pixels). The advantage of binary images is that the amount of computation can be reduced. That is , each term in eq.1 will take only two values: it will be one when $I_{x+I,y+j}=T_{x,y}$ and zero otherwise[4]. Thus eq.1 can be implemented as:

$$\max e = \sum_{(x,y)\epsilon W} \overline{Ix+\imath, y+\jmath \oplus Tx, y} \qquad (7)$$

Where the symbol $\overline{\oplus}$ denotes the exclusive NOR operator. This equation can be easily implemented and require significantly less resource than the original matching function working in the gray domain [4]. The computation is performed in absence of square, multiplications, summing up operations and floating point coding required while computing similarity between gray-level images [8].

We consider X = ($x_1 \ldots$ xm … x_n) and Y =(x_1 … $x_m \ldots$ x_n) as n-dimensional binary vectors where X is one of the template $X_1, \ldots, X_C$ and Y is the image to compare. These binary vectors are used within a δ_m(u, v) function Eq. (8) where u, v = 0, 1 and y_m, x_m are the m_{th} elements of Y and X respectively. We define then in Eq. (9) $n_{u,v}$ as the number of occurrences where x_m = u and y_m = v ∀m. The term n_{11} denotes the positive matches, i.e. the number of 1 bits that match between y_m and x_m. The term n_{00} is the negative matches, i.e. the number of 0 matching bits. The terms n_{10}; n_{01} denote the number of bit mismatches - the first where pattern x_m has a 1 and pattern y_m has a 0, and vice-versa.

$$\delta_m(u, v)\begin{cases} 1 \; if \text{ xm } = \text{ u and ym } = \text{ v} \\ 0 \;\; otherwise \end{cases} \quad (8)$$

$$n_{u,v} \sum_{m=1}^{n} \delta m(u, v) \quad (9)$$

7. Problematic

The computational cost of template matching is large. If the template is square and of the size m*m and it's matched to an image of size N*N, since the m^2pixels are matched at all image points (except for the border), the computational cost is (N^2m^2). This is the cost for position invariant template matching. Any further parameters of interest can increase the computational cost in proportion to the number of values of the extra parameters. This clearly a large penalty and so a direct digital implementation of template matching is slow. Accordingly, this guarantees interest in fast techniques that can deliver same result, such as using Fourier implementation based on Fast Transform Calculus.

8. Conclusion

At this chapter we have conducted a definition on the correlation/convolution as a process similarity analysis between two images. We defined the template matching process based on Lp norm similarity measure. The major convenient is the computation cost of template matching process. And we presented how is the template matching process is similar to correlation process using Fourier Theory. Knowing the fact that Fast Fourier transform speed up the process of template matching. Therefore, our goal is to provide Fourier transform based binary template matching algorithm. In the next chapter, we present the details of this approach called Fast Fourier Transform based binary template matching.

Chapter II: Fast Fourier Transform based binary template matching

This section will define the process of the main idea of the project. First, we define the Fourier transform based template matching implementation. Then, we represent the results and their interpretation.

1. Fourier Transform based template matching formulation:

When dealing with the L_p norm based similarity measures the template matching can be achieved through a correlation operator. In fact, the Fourier transform actually gives an alternative method to implement template convolution and to speed it up, for large templates. In Fourier transforms, the process that is dual to convolution is multiplication [4]. Fast Fourier Transform algorithms can rapidly compute the Fourier Transform; that is FFT appears as a good candidate to support template matching [9].

The matching process in eq.6 is correlation not convolution. Thus, we need to express the correlation in terms of convolution. This can be done as follows. First, we can rewrite the correlation denoted by $\otimes$ in eq.6 as:

$$I \otimes T = \sum_{(x,y)\in W} I x', y' T x' - i, y' - j \quad (10)$$

Where x’=x+i and y’=y+j. Convolution denoted by * is defined as

$$I * T = \sum_{(x,y)\in W} I x', y' T i - x', j - y' \quad (11)$$

Thus, in order to implement template matching in the frequency domain, we need to express eq.10 in terms of eq.11. This can be achieved by considering that

$$\text{I}\otimes T = I * T' = \sum_{(x,y)\in W} I x', y' T x' - i, y' - j \quad (12)$$

Where

$$T' = T - x, -y \quad (13)$$

That is, correlation is equivalent to convolution when the template is changed according to eq.13. this equation reverses the coordinate axes and it corresponds to horizontal and a vertical flip.

In the frequency domain, convolution corresponds to multiplication. As such, eq.12 can be implemented by

$$\text{I}\otimes T = I * T' = FT^{-1}(FT(I) \times FT(T')) \quad (14)$$

Where FT denotes Fourier Transformation. Note that the multiplication operator actually operates point by point, so each point is the product of the pixels at the same position in each image(operator .*in matlab). This computationally faster than its direct implementation, given the speed advantage of the FFT.

To implement this equation, we compute T' by flipping the template and then computing its fourier transform FT(T').

Note that one assumption is that the transforms are of the same size, even though the template's shape is usually much smaller than the image. The solution is to include extra zero values(zero-padding) to make the image of the template the same size as the image.

2. *Fast Fourier Transform implementation:*

The code to implement template matching by Fourier, is given in Code Fig.5. First, the implementation takes the image and the flipped template. Second, the template and the image need to be zero padded. In fact the image of size 256*128 is padded by 128 zeros on right/left sides and 256 zeros on bottom/down sides. Its size is then 768*384. But the size should be squared, so 1024*1024. The same is done for the template. Third, the transforms are evaluated. The required convolution is obtained by multiplying the transforms and then applying the inverse. The resulting image is the magnitude of the inverse transform. This process can be formulated using brightness. As appropriate. The maximum frequency domain value, indicates the position of the template and gives a value for its size. So as a result we have matched points $n_{u,v}$: n_{00} negative matched points, n_{11} positive matched points, n_{10} and n_{01} mismatched points, as explained by eq.9.

```
%Fourier Transform Convolution

function [ n00,n01,n10,n11] = FTconv(inputimage,template)

%image and template pixels 0 or1
T = double(1 - template/255)
I= double(1 - inputimage /255)

% flipping T
Tf=rot90(Template,2);

% not image and not template NotI=(not(I));
NotTf=(not(Tf));

%Frourier transform of I, T, notI, notT
FI=(fft2(I,1024,1024));
FT=fft2(Tf,1024,1024);
FnotI=fft2(NotI,1024,1024);
FnotT=fft2(NotTf,1024,1024);

%convolution map M(u,v)
M00=(FnotT.*FnotI);
M01=(FnotT.*FI);
M10=(FT.*FnotI);
M11=(FT.*FI);

%inverse TF of Map and take magnitude part
R00=real(ifft2(M00));
R01=real(ifft2(M01));
R10=real(ifft2(M10));
R11=(real(ifft2(M11)));

%matched_points_max_values
[n11,i11]=max((R11(:)));
[n10,i10]=max((R10(:)));
[n00,i00]=max((R00(:)));
[n01,i01]=max((R01(:)));
n=n00+n01+n10+n11
end
```

Figure 5 Code Fourier Transform Convolution

3. Conclusion

We presented the Fourier Transform formulation for template matching. Then we defined the Fast Fourier transform based binary template matching algorithm that measures the similarity between an image and its appropriate template.

Chapter III: Results and discussion

In the previous chapter we have much detail the adopted method. Definitely we need to assess how our work has overcome the stated constraints. Therefore, this chapter presents an experimental evaluation with a demonstration that is justified by comparing the experimental results associated with each step made any throughout the project.

1. Data Description

The data used for the implementation of our work are defined as some Manga image registration results, the RMI_{btm} (Registered Manga Images “binary template matching”) dataset obtained in [8] , between legal and illegal version of Manga pages. The next Figure gives some examples of Manga registered images extracted from this dataset. The total dataset is composed of 7688 images including 3844 template models and 3844 matched Regions of Interest (RoI). In our work we use only 435 images as examples from the RMI_{btm} dataset. These images are binary that means the pixels values are 0 or 1.

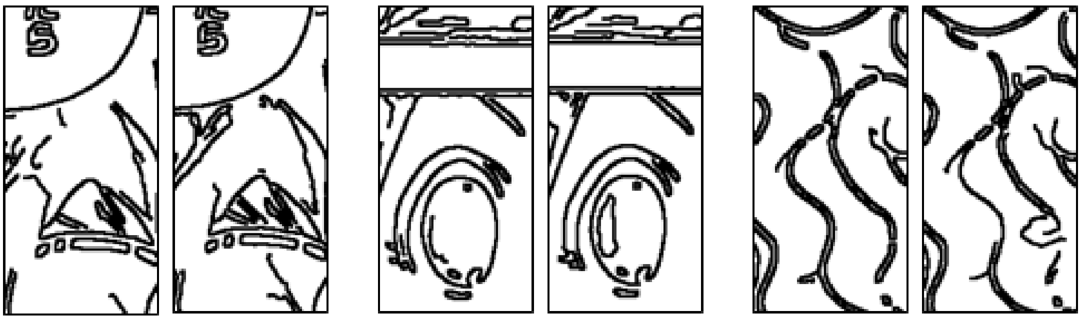

Figure 6 Some examples of registered Manga images left – matched RoI “Region of Interest”, right – the model template

2. Environment:

This project is based on the use of Matlab version R2013a and Asus laptop that processor is x64, Intel core i5 with 6Go RAMemory.

3. Test protocol

In order to test our work , we follow the following steps:

- Step1: Calculate the FFT similarity measure for matching between the image and the template.
- Step2: Calculate the exclusive NOR operator for matching between the image and the template.
- Step3: We compare the FFT measure with logical operator results

4. *experimental* ***Results***

We rank the results according to already defined in Chapter.2 approach. We compare the values of FFT by varying the bit value {0,1} for a binary image and its appropriate template and counting the positive and negative matched points and mismatched points demonstrated in Fig8 with the values obtained by exclusive Nor operator, respectively .

We represent the process of Fourier transform based template matching algorithm applied on one Manga image and its template as an example in Fig.8.

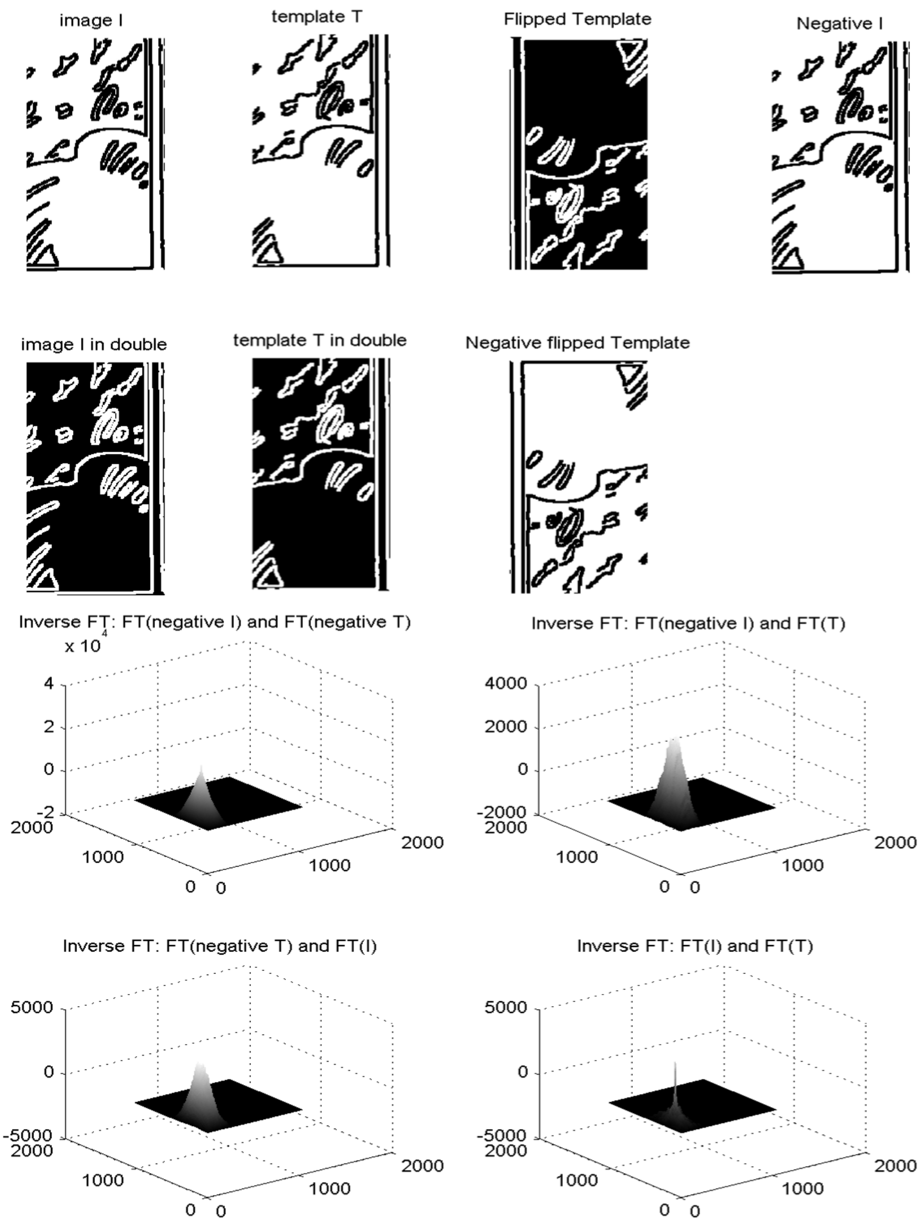

Figure 7 Example for Manga image processed in Fourier Transform based Template matching algorithm results

We use 436 Manga images and templates. These images are binary that means the pixels values are 0 or 1. We calculate the number of negative and positive matched and mismatched points between each template and correspondent image $n_{u,v}$ where $(u,v) \in \{01\}$.

We compare the values that we get from Fourier transform based template matching and exclusive NOR operator, as shown in Fig.8, 9, 10 and 11. We represent the error between them as absolute differences in probability density in a table. Besides, every figure demonstrates the matched or mismatched point value from Fourier transform in function of according point using exclusive NOR operator.

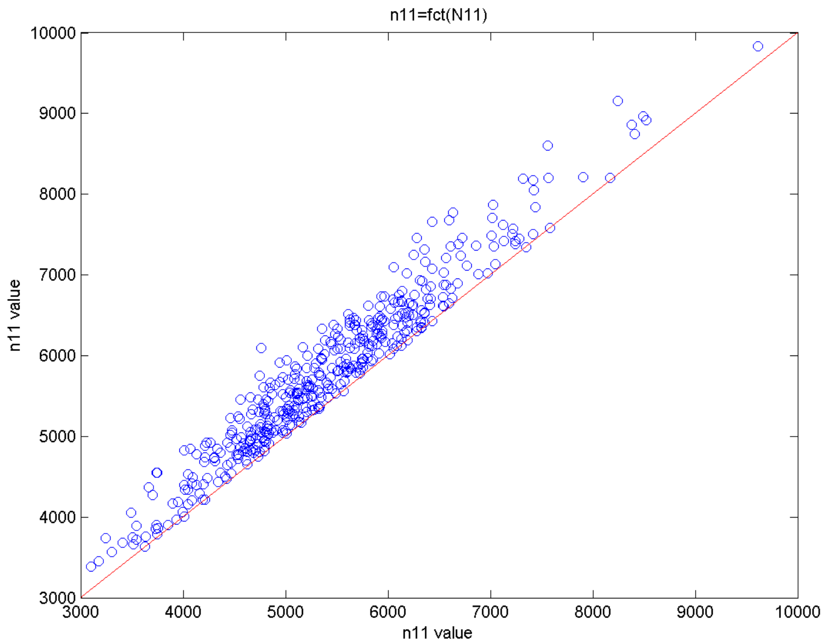

Figure 8 Fourier transform dependent linearly of exclusive NOR operator based template matching for Positive matched points (u,v)=(1,1)

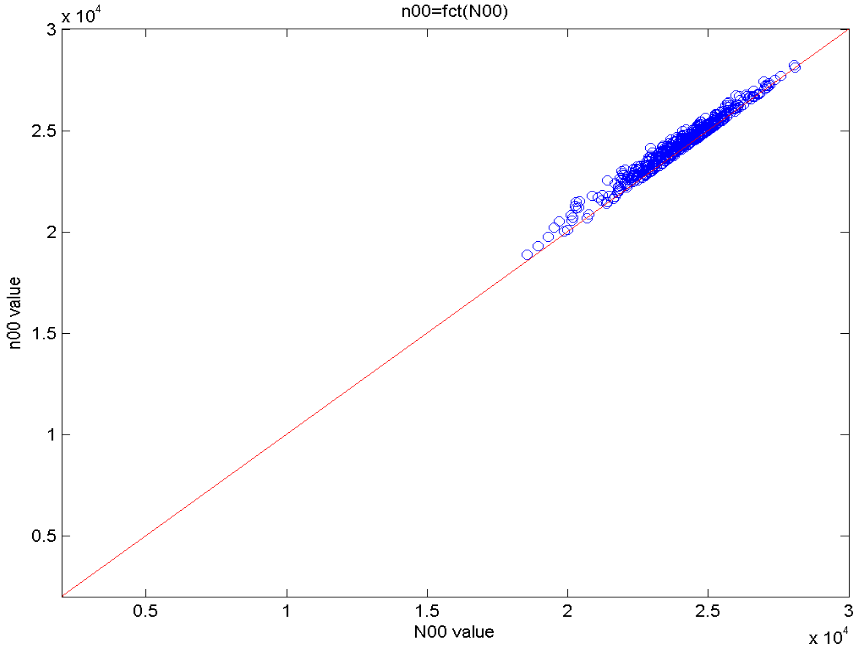

Figure 9 Fourier transform dependent linearly of exclusive NOR operator based template matching for negative matched points (u,v)=(0,0)

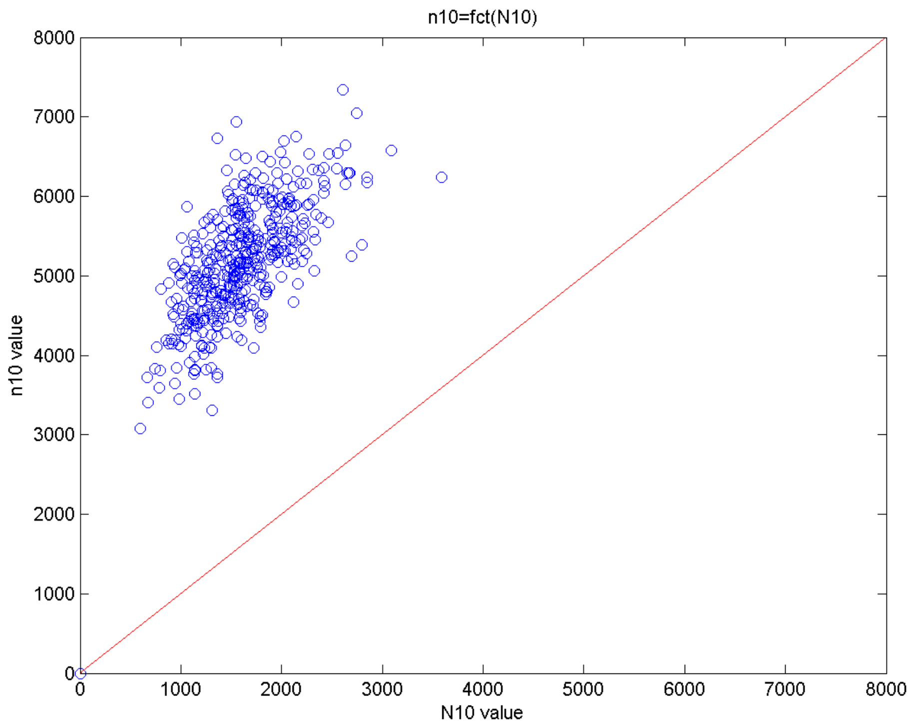

Figure 10 Fourier transform dependent linearly of exclusive NOR operator based template matching for Mismatched points (u,v)=(1,0)

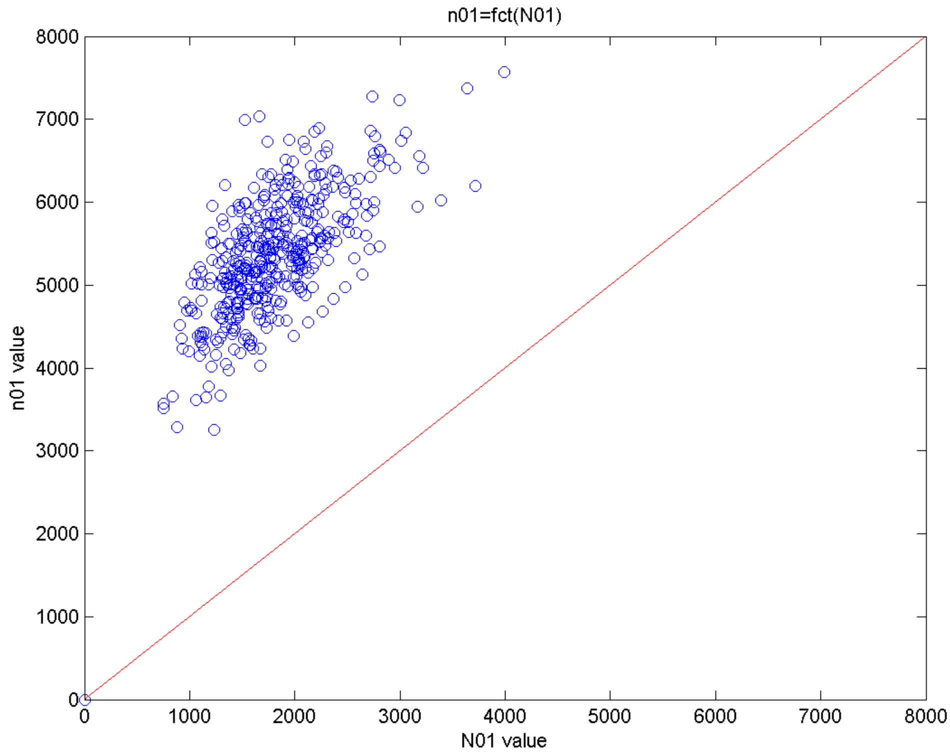

Figure 11 Fourier transform dependent linearly of exclusive NOR operator based template matching for Mismatched points (u,v)=(0,1)

5. Interpretation and discussion

Fig.8 illustrates the results of template matching in the Fourier domain using the image and the template as shown in Fig.7. Fig.7 left bottom shows the flipped and padded template. The Fourier transforms are given in Fig.7 right bottom, respectively. These transforms are multiplied point by point. When the inverse Fourier transformed is applied, the result fig.8 down shows the template best matched the image. Fig.8 down shows a zoom of the region where the peak is located. This peak indicates the maximum that calculates the best match between image and template in the 4 different cases, based on eq.8. These cases are as following:

- The maximum positive matches between the positive image and the positive template is defined by the $n_{1,1}$ as the number of occurrences (the number of 1 matching bits)where $I(x,y)=1$ and $T(x,y) = 1$ $\forall(x,y) \in 1024^2$.
- The maximum negative matches between the negative image and the negative template is defined by $n_{0,0}$ .the term $n_{0,0}$is the negative matches, i.e. the number of 0 matching bits.
- The maximum mismatches between the positive image and the negative template is defined by the n_{01}, and inversely by n10. The terms n_{10}; n_{01} denote the number of bit mismatches - the first where pattern T(x,y) has a 1 and pattern I(x,y) has a 0 $\forall(x,y) \in 1024^2$, and vice-versa.

To evaluate the method Fourier transform based template matching, we compare with the exclusive NOR operator method. Easily, we have the results of this logical operator applied on the binary Manga images database. Fig.8,9, 10 and 11 display how linear the function $n_{u,v}=N_{u,v}$ to visualize the approximation/error value between these two measures.

The error is defined by the absolute difference between the Fourier transform measure results, as $n_{u,v}$, and the logical operator measure results, as $N_{u,v}$, that (u,v) belong to{0,1}, normalized by the global size 12*256, wihin 436 images. It's represented in the following table:

u	***v***	***Error value***
0	0	0.0097
0	1	0.1083
1	0	0.1091
1	1	0.0116

Figure 12 Error probability values in diferent matching cases

Clearly, the case of negative and positive matching that (u,v)={(0,0),(1,1)}, fig.8 and 9 are closely to a linear function dependently to the number of matched points.

In the mismatching cases that (u,v)={(0,1),(1,0)}, the fig.9 and 10 represent an affine function.

In the table Fig.12 also, we can notice the error gap is bigger in the cases of mismatching.

So, the Fourier Transform gives approximatively the same results of Logical operator results the Fourier transform can represent the matching point in cases of positive and negative matching.

So we can interpret this result as the Fourier transform can't give a good a result of template matching in case of mismatched point.

The binary image is sparse and the mismatched points defines the sparse noise in the template. Knowing that Fourier theory assumes that an image replicates spatially to infinity in frequency domain [4]. The sparse noise is embedded in high frequency error not supported in the binary domain when using FFT. As a conclusion, Fourier transform can't perform in case binary image with sparse noise.

6. Conclusion:

In this chapter we have presented the data used in our work and the tools. Then, we describe the protocol we adapt during the test. Finally, we figure the results by well interpreted experimental curves.

Conclusion

This project began to succeed the TU_4 module in International CADS (Computer Aided Decision Support) Master with the subject "**Characterization of the FFT-based correlation distortion for binary template matching** ". We have through this work addresses several aspects related to the template matching process dealing with Manga Copyright protection Application.

This project thesis report summarizing our work was organized as follows:
We introduced for first chapter concepts as we considered important to our work. We have defined the steps of process the template matching and the problematic.
Furthermore, we have demonstrated Fourier theory based template matching approach and its implementation, in second chapter.
A third chapter was used to, Experimental assessment, describing the Experimental results of our work and to their interpretation for validation the Fourier transform based binary template matching.

In this project the goal is to **test the FFT-based correlation distortion for binary template matching** by referring to results obtained by logical operator based binary template matching. the performance of this approach is proven when we process with positive or negative matching ($I(x,y)=T(x,y)=\{(0,0)\ ,(1,1)\}$. The mismatching cases are interpreted by sparse noise. Here, the problem was enhanced: Fourier transform provides bad results of matching measures. In fact, Fourier transform replicates image spatially to infinity that's why it has a difficulty in sparse image. Our case the binary image is sparse. Therefore, Fourier transform based template matching has a major inconvenient that it cannot detect the matching with sparse noise.

As we can see the FFT is optimal in searching according to the computation cost, but it's not attractive in a way of similarity measure. And, some research prove that the standard similarity measures are not valuable in the binary case. So, as research perspective, it's needed to implement an algorithm for Binary template matching case.

Bibliographie

[1] Yarlagadda and R.K.Rao, Analog and Digital Signals and Systems. 2010, XXVIII, chapter 2.

[2] Rafael C. Gonzales and Richard E. Woods. Digital image processing, Third edition. Chapter 3, section 3.4.2, p148-150.

[3] paper named an introduction to template matching

[4] Mark S.Nixon and Alberto S. Aguado, Feature extraction and image processing for computer vision, third edition Chapter 5 section3, p222-233.

[5] J.P. Lewis, Fast Template Matching, Vision Interface 95, Canadian Image Processing and Pattern Recognition Society, Quebec City, Canada, May 15-19, 1995, p. 120-123.

[6] R.O. Duda and P.E.Hart, Pattern Classification and Scene Analysis, New York: Wiley, 1973.

[7] Mathieu Delalandre, Fast Optimal Binary Template Matching Using Upper Bound Measure Approximation. Laboratory of Computer Science University of Tours, Tours city, France

[8] M. Delalandre, M. Iwata, K. Kise. Fast and Optimal Binary Template Matching, Application to Manga Copyright Protection. Workshop on Document Analysis System (DAS), pp. 298-303, 2014.

[9]S. Choi, S. Cha, and C. Tappert, “A survey of binary similarity and distance measures,” Journal of Systemics, Cybernetics and Informatics (SCI), vol. 8, no. 1, pp. 43–48, 2010.

[10]N. Roma, J. Victor, and J. Tome, “A comparative analysis of cross-correlation matching algorithms using a pyramidal resolution approach,” in Workshop on Empirical Evaluation Methods in Computer Vision (ECCV). Machinz Perception and Artificial Intelligence, vol. 50. World Scientific, pp. 117–142, 2002.

[11] S. Cha and al. Enhancing Binary Feature Vector Similarity Measures Journal of Pattern Recognition Research (JPRR), vol 1, pp. 63-77, 2006.