

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/282407273>

Quick Techniques for Template Matching by Normalized Cross-Correlation Method

Article in British Journal of Mathematics & Computer Science · January 2015

DOI: 10.9734/BJMCS/2015/16461

CITATION

1

READS

1,609

2 authors, including:



Ahmed Ibrahim

Princess Nora bint Abdul Rahman University

8 PUBLICATIONS 56 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



aaawwwwwwwwwww [View project](#)



Quick Techniques for Template Matching by Normalized Cross-Correlation Method

M. I. Khalil^{1*} and Ahmed Ibrahim²

¹Department of Networking and Communication Systems, Princess Nourah bint Abdulrahman University, KSA, Riyadh College of Computer and Information Sciences, Saudi Arabia.

²Department of Computer Sciences, Princess Nourah bint Abdulrahman University, KSA, Riyadh College of Computer and Information Sciences, Saudi Arabia.

Article Information

DOI: 10.9734/BJMCS/2015/16461

Editor(s):

(1) Victor Carvalho, Polytechnic Institute of Cávado and Ave, Portuguese Catholic University and Lusiada University, Portugal.

Reviewers:

(1) Hui Li, Howard University, Washington, USA.

(2) Anonymous, Civil Aviation University of China, China.

Complete Peer review History: <http://sciencedomain.org/review-history/11189>

Original Research Article

Received: 03 February 2015

Accepted: 19 February 2015

Published: 31 August 2015

Abstract

Object recognition is one of the fundamental challenges in signal processing, image processing and computer vision, where the goal is to identify and localize the extent of object instances within an image. A novel approach for performing the matching by normalized cross-correlation method in minimum time is introduced. The template matching by correlation is performed between template w and the image f where the template's position is to be determined in the image. The computing process of correlation coefficient is analyzed and resolved into minute parts or units. These minute units are computed one time only before embedding them in larger blocks and stored in sum tables. The larger blocks are computed in recursive manner, using the sum tables, by adding and/or subtracting minute units from the original block instead of computing them from scratch. Moreover, this technique has been more developed by performing the cross-correlation on the odd or even signal's samples only. The new approach, in its final form, has reduced the cross-correlation calculation time by 90%-94% depending on the image's and template's sizes.

Keywords: Matching by cross-correlation; digital signal processing; image processing; cross-correlation.

1 Introduction

Normalized Cross-Correlation (also called cross-covariance) between two input signals is a kind of template matching. It is generally considered to be the gold standard of many applications [1-3]. However, its high

*Corresponding author: E-mail: magdi_nrc@hotmail.com, azibrahim@pnu.edu.sa;

computational cost is a significant drawback in its real-time application, especially when highly sampled RF signals and an exhaustive search are used [4]. Normalized Cross-correlation can be done in any number of dimensions. One-dimensional normalized cross-correlation between two input signals can be defined as:

$$\gamma(x) = \frac{\sum_s [w(s) - \bar{w}][f(x+s) - \bar{f}_x]}{\sqrt{\sum_s [w(s) - \bar{w}]^2 \sum_s [f(x+s) - \bar{f}_x]^2}} \quad (1)$$

The coefficient, r , is a measurement of the size and direction of the linear relationship between variables x and y .

Given an image $f(x,y)$, the correlation problem is to find all places in the image that match a given subimage $w(x,y)$ (called mask or template). This means that the position of the given pattern is determined by a pixel-wise comparison of the image with a given template, that contains the desired pattern. For this, the template is shifted u discrete steps in the x direction and v steps in the y direction of the image, and then the comparison is calculated over the template area for each position (u,v) . To calculate this comparison, normalized cross correlation is a reasonable choice in many cases [4-7]. The method of choice for matching by correlation is to use the correlation coefficient:

$$\gamma(x, y) = \frac{\sum_{s,t} [w(s,t) - \bar{w}][f(x+s, y+t) - \bar{f}_{xy}]}{\sqrt{\sum_{s,t} [w(s,t) - \bar{w}]^2 \sum_{s,t} [f(x+s, y+t) - \bar{f}_{xy}]^2}} \quad (2)$$

Where w is the template, \bar{w} is the average value of the elements of the template (computed only once). f is the image, and \bar{f}_{xy} is the average value of the image in the region where f and w overlap. The summation is taken over the values of s and t such that the image and the template overlap. The denominator normalizes the result with respect to variation in intensity. The values of $\gamma(x, y)$ are in the range $[-1, 1]$. A high value of $|\gamma(x, y)|$ generally indicates a good match between the template and the image.

There are many approaches for implementing the cross correlation. Most of these approaches are based on the concept of moving a classifier (or object) around over all possible scales and positions, scanning the image and searching for maximal detection responses, which is commonly called Sliding Windows (SW).

Pixel-by-pixel template matching is very time-consuming. For a scene image of size $M \times N$, and the template of size $m \times n$, the computational complexity is $O(m \times n \times M \times N)$.

Because successive reference windows usually overlap, the entire calculation of the numerator in (2,3) is also redundant.

The basic idea behind this work is to pinpoint a collection of building units (vectors), the sum of elements contained in these vectors will be computed once only. These sums will be used to compute the bigger blocks which consist basically of collections of these vectors. Accordingly, the time required to compute the bigger vectors will be reduced resulting in a speed-up of an order of magnitude over the brute force approach of matching method [7].

Assume that a two dimensional array template $w(s,t)$ is to be matched with a two dimensional array image $f(x,y)$ and considering (2), the normalized cross-correlation consists of three terms, i.e., the energy of the template window ($\sum_{s,t} [w(s,t) - \bar{w}]^2$) in the denominator, the energy of the comparison window ($\sum_{s,t} [f(x+s, y+t) - \bar{f}_{xy}]^2$) in the denominator and the standard cross-correlation between these two windows ($\sum_{s,t} [w(s,t) - \bar{w}][f(x+s, y+t) - \bar{f}_{xy}]$) in the numerator. These terms need to be calculated for each pixel in the image $f(x,y)$. This calculation is to be repeated for each template window across the entire signal length. Therefore, the normalized cross-correlation-based template matching method is extremely time consuming.

2 Related Work

The efficient Normalized Cross-Correlation (NCC) calculation method based on sum tables relies on the fact that most calculations are redundant because of the exhaustive search of the comparison windows and high overlap between the reference windows [4]. Peter Nillius [8] attempted to speed up NCC first by transforming each sub-block of the image into the Walsh basis. The Walsh transform expansion can be done very efficiently through a binary tree of filters. Calculating the NCC using the Walsh components requires $2N - 1$ operations instead of $4N + 1$ in a straightforward implementation. A highly parallel implementation of the cross-correlation of time-series data using graphics processing units (GPUs), which is scalable to hundreds of independent inputs and suitable for the processing of signals from "Large-N" arrays of many radio antennas is presented [9]. The computational part of the algorithm, the X-engine, is implemented efficiently on Nvidia's Fermi architecture, sustaining up to 79% of the peak single precision floating-point throughput. M.I. Khalil [10] has introduced Parallel implementation of the cross-correlation execution over the local network, or in some cases over a Wide Area Network (WAN), helps reducing the processing time.

3 Proposed Approach

The proposed method used some of pre-calculated sum tables to avoid repeating redundant computations in the definition of the normalized cross-correlation given by (1,2). Moreover, the proposed method introduced a method for reducing the time required for computing the sum tables.

The term \bar{w} : this term will be computed once only because the contents of the template do not change while sliding over the image:

```
// computing term  $\bar{w}$ 
Sum=0;
for (i=0; i<s; i++)
  for (j=0; j<t; j++)
    sum = sum + w(I,j);
  next j;
next i;

 $\bar{w} = \frac{sum}{s*t};$ 
```

The term $[w(s, t) - \bar{w}]$: this term should be computed once only for each pixel in the template window and this procedure is the same for both the ordinary one and the new approach methods:

```
// computing term  $[w(s, t) - \bar{w}]$ 
for (i=0; i<s; i++)
  for (j=0; j<t; j++)
    d = w (i,j) -  $\bar{w}$ ;
    add d to table-1 at position (i,j);
  next j;
next i;
```

Table 1. Used in the adjacent algorithm

s	t	$w(s, t)$

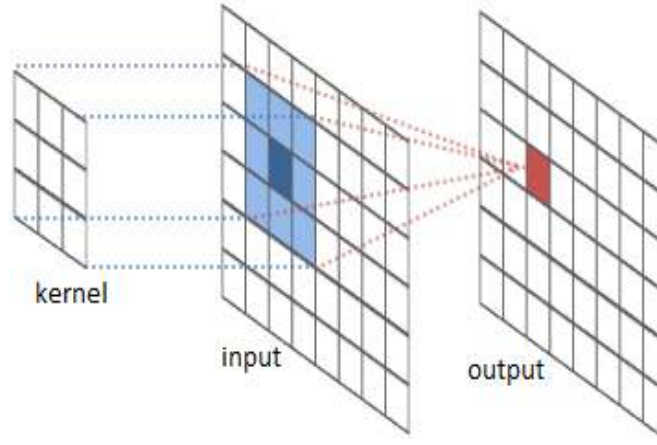


Fig. 1. Object Recognition using Template Matching Method

The term $[f(x + s, y + t) - \bar{f}_{xy}]$: At certain point (x, y) , the ordinary method for computing the term $[f(x + s, y + t) - \bar{f}_{xy}]$ comprises subtracting the value of \bar{f}_{xy} from the value of the pixel at $f(x + s, y + t)$. \bar{f}_{xy} , as previously defined, is the average value of the image in the region where f and w overlap and will be computed every time the template moves over the image. This process will be repeated for all pixels under the template window (s, t) . So, considering the computing process of \bar{f}_{xy} in the ordinary method, every time we should compute \bar{f}_{xy} for all pixels under $w(s, t)$ by adding the pixels values and dividing them by $s \cdot t$. When moving the template horizontally, from pixel $f(x, y)$ to a new pixel $f(x, y+1)$, the value of $\bar{f}_{x, y+1}$ should be recomputed in the same manner.

Inspecting the areas under the template at points (x, y) and $(x, y+1)$ (Fig. 2a and b respectively), it is noticed that the yellow area (Fig. 2c) is common between them. In the new approach, $\bar{f}_{x, y+1}$ can be computed simply by adding the pixels in the blue strip (Fig. 2c) to $\bar{f}_{x, y}$ (the green block in Fig. 2a) and subtracting the pixels in the green strip (Fig. 2c) yielding to the blue block (Fig. 2b). The ordinary method requires s by t add operations, while the proposed one requires only 3 add operations.

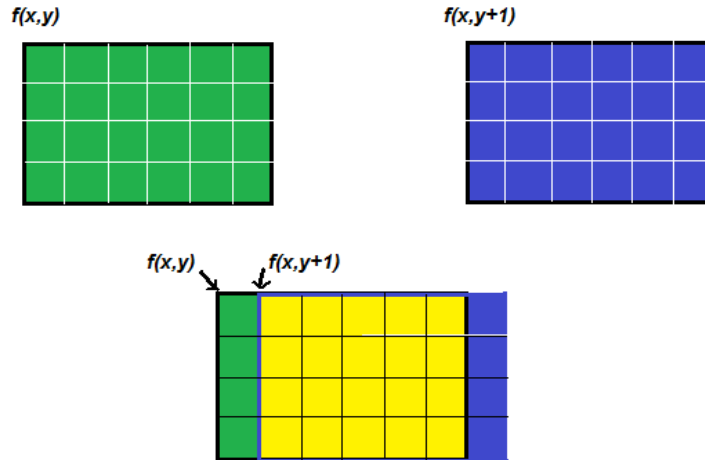


Fig. 2. Area under sliding window: a) at pixel $f(x, y)$ b) at pixel $f(x, y+1)$ c) comparing the area in case a, b

```

// Preparing for computing  $\bar{f}_{xy}$  algorithm:
for (i=0; i<x-s ;i++)
  for (j=0; j<y-t ;j++)
    sum_of_column =0;
    for (m=j; m<j+s ;m++)
      sum_of_column = sum_of_column + f(i,m);
    add sum_of_column to table-2 at position (i,j);
  next j;
next i;

```

Table 2. Used in the adjacent algorithm

x	y	$sum_of_column(x,y)$

```

//computing  $\bar{f}_{xy}$ 
//compute first block at (0,0) covered by the template window
sum =0;
for (i=0; i<s ;i++)
  for (j=0; j<t ;j++)
    sum = sum + sum_of_column from table-2(i,j);
  next j;
next i;
sum_of_block(0,0) = sum;
 $\bar{f}(0,0) = sum\_of\_block(0,0) / (s * t);$ 
add  $\bar{f}(0,0)$  to table-3 at position (0,0);

//compute sum_of_block for rest of image
sum =0;
for (i=1; i<x-s ;i++)
  for (j=0; j<y-t ;j++)
    sum_of_block(i,j) = sum_of_block(i-1,j)+sum_of_culmn from
    table-2(i+s+1,j) – sum_of_column from table-2 (i-1, j);
     $\bar{f}(i,j) = sum\_of\_block(i,j) / (s * t);$ 

    add  $\bar{f}(i,j)$  to table-3 at position (i,j);
  next j;
next i;

```

Table-3. Used in the adjacent Algorithm

x	y	$sum_of_block(x,y)$	$\bar{f}(i,j)$

The number of computations required to calculate the numerator of the NCC coefficient is still comparatively high. Therefore, further simplification of this calculation is required. Accordingly, for quick matching purposes, the normalized cross-correlation algorithm can be carried only on either the odd or even pixels of both the image and the template window.

4 Experimental Results

Three versions of template matching by normalized cross-correlation algorithm have been implemented using Microsoft visual studio C# platform. The first version is “the ordinary” and is based on the sum tables. The second version is “the modified” and is based also on the sum tables beside the utilization of recursive calculation of the some terms in Eq.2. The third version is similar to the second one except that the outer and inner loops in the correlation procedure are modified to deal only on either the odd or even pixels of both the image and the template window. Each of the three versions has been tested on several images with different sizes and a lot of sum-images with different sizes used as templates. Following are two cases of those evaluation experiments. In the first case, a 956x428 image (Fig. 3) in addition to 16 sub-images with sizes 30x35 ~ 183x329 have been used to test the three programs yielding to results list in Table 4 and plotted in Fig.4 respectively. In the second case, a 655x598 image (Fig. 5) in addition to 16 sub-images with sizes 30x50 ~ 200x330 have been used to test the three programs yielding to results list in Table 5 and plotted in Fig.6 respectively.

Table 4. The time of processing for templates with different sizes(first case)

Template dimensions			Time (milliseconds)		
Height	Width	Size (pixels)	Ordinary	Modified	Quick
35	30	1050	65320	31556	1857
55	59	3245	191657	85642	4343
60	85	5100	280053	135245	6763
60	115	6900	382341	174802	8288
90	125	11250	543668	261062	12494
100	150	15000	6935533	324917	16165
100	180	18000	808926	384637	18018
113	176	19888	904080	405752	19621
125	200	25000	1054935	473081	22897
150	200	30000	1093019	504157	25594
170	200	34000	1193019	545289	27312
201	211	42411	1298898	697444	30073
180	250	45000	1402732	654388	31819
175	270	47250	1502915	697031	34066
200	275	55000	1523354	709518	37792
183	329	60207	1695731	798115	38848

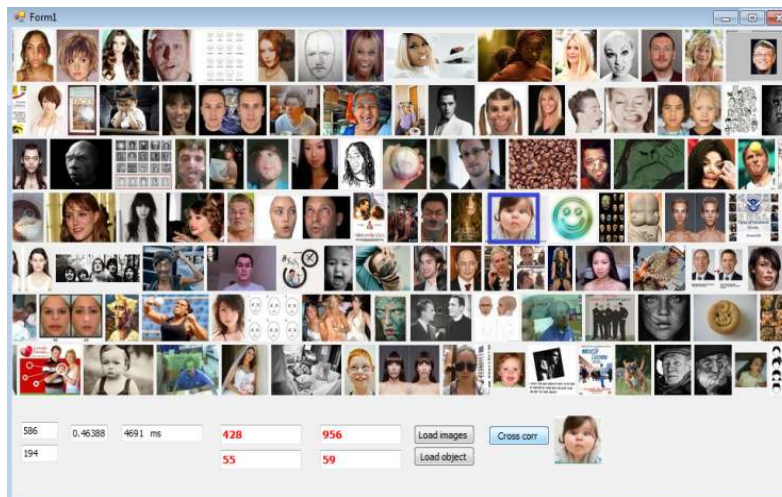
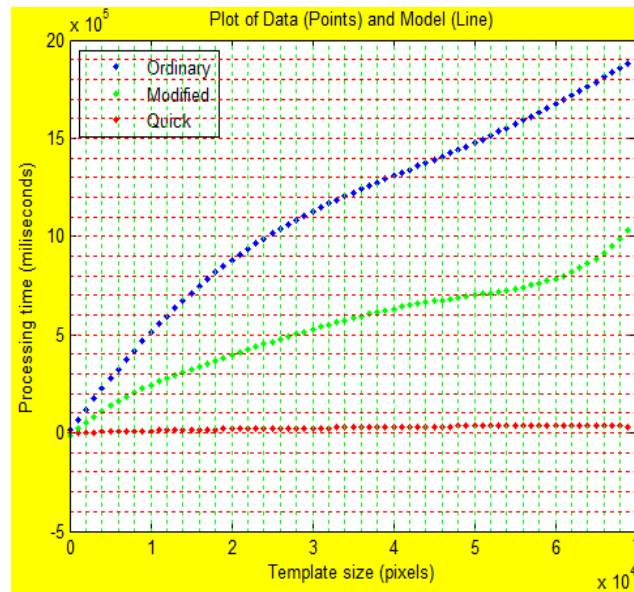


Fig. 3. A snapshot of the first case test

Table 5. The time of processing for templates with different sizes (second case)

Template dimensions			Time (milliseconds)		
Height	Width	Size (pixels)	Ordinary	Modified	Quick
30	50	1500	41253	31850	2389
39	50	1950	54210	40722	2982
75	50	3750	97834	73930	5121
136	89	12104	253097	198470	13130
170	80	13600	273862	212876	13860
190	90	17100	324446	254527	16335
215	85	18275	328314	260560	17244
200	100	20000	363084	296113	18381
187	133	24871	435018	345699	22378
200	140	28000	464747	386119	23856
200	150	30000	492168	388680	25375
210	150	31500	515171	443360	26338
225	175	39375	587751	504293	30834
300	175	52500	574274	544359	33275
300	180	54000	675987	592283	33382
350	170	59500	672706	575036	32936
330	200	66000	751926	620789	35942

**Fig. 4. Comparison between Ordinary, Modified and Quick Correlation-method techniques applied on 956x428 image and different sizes objects**

In the first case, the time of processing using the modified method is reduced to almost 50% of that of the ordinary method. Moreover, the time of processing using the quick method has been extremely reduced to about 1/16 of the time consumed in the ordinary method.

In the second case, the time of processing using the modified method is reduced to almost 75% of that of the ordinary method. The difference in ratio between case one and case two is due to the ratio between the image and template sizes. The quick method has achieved the same reduction as in case one.

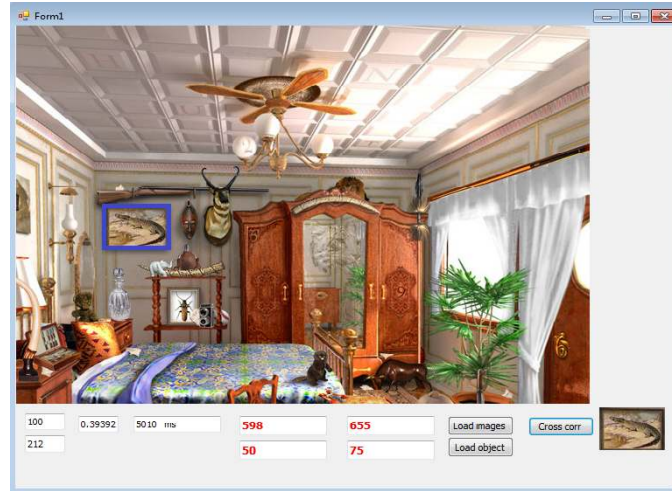


Fig. 5. A snapshot of the second case test

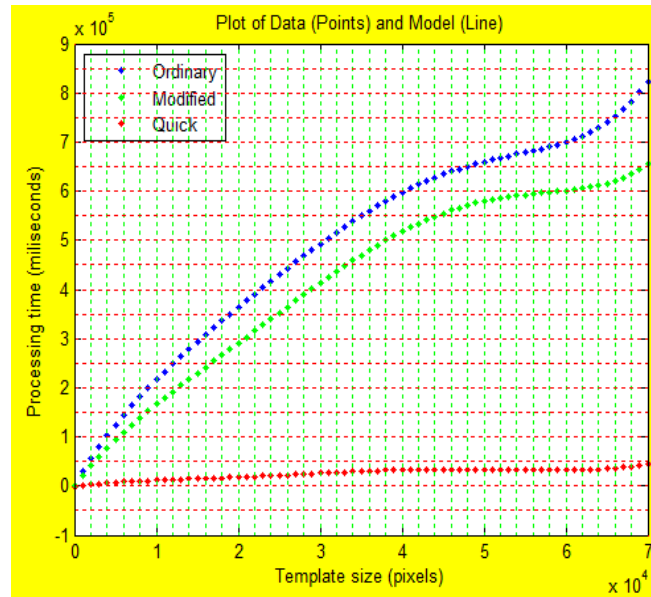


Fig. 6. Comparison between Ordinary, Modified and Quick Correlation-method techniques applied on 655x598 image and different sizes objects

5 Conclusions

Template matching by normalized cross-correlation method has many applications in many fields such as object recognition, signal processing, image processing and computer vision, where the goal is to identify and localize the extent of object instances within an image. However, its high computational cost is a significant drawback in its real-time application, especially when highly sampled RF signals and an exhaustive search are used. In this paper, a new fast algorithm for the computation of the normalized cross-correlation is presented. It is based on using the sum tables and recursive calculations. The consumed time has been reduced comparing with the traditional approaches while maintaining the same high accuracy (correlation coefficient ~ 0.998). Moreover, for quick purposes, the developed approach can be carried only

on the even or odd pixels in both the image and the template window respectively yielding to extreme reduction in processing time.

Competing Interests

Authors have declared that no competing interests exist.

References

- [1] Khalil MI. Car plate recognition using the template matching method. International Journal of Computer Theory and Engineering. 2010;2(5):683-687.
- [2] Linzer E, Tiwari P, Zubair M. High performance algorithms for MPEG motion estimation. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing. 1996;1934-7.
- [3] Abusaeeda O, Evans J, DD, Chan J. View synthesis of KDEX imagery for 3D security X-ray imaging. In Proc. 4th International Conference on Imaging for Crime Detection and Prevention (ICDP-2011); 2011.
- [4] Jianwen, Elias Konofagou E. A fast normalized cross-correlation calculation method for motion estimation. IEEE Trans Ultrason Ferroelectr Freq Control. 2010;57(6):1347-1357.
- [5] Kaie Briechele, Uwe Hanebeck D. Template matching using fast normalized cross correlation. Available:http://i81pc23.itec.uni-karlsruhe.de/Publikationen/SPIE01_BriecheleHanebeck_CrossCorr.pdf
- [6] Lewis JP. Fast normalized cross correlation, industrial light and magic. Available:http://www.academia.edu/653962/Fast_template_matching
- [7] Duda RO, Hart PE. Pattern classification and scene analysis. New York; 1973.
- [8] Fast Block Matching with Normalized Cross-Correlation using Walsh Transforms, TRITA-NA-P02/11, ISRN KTH/NA/P--02/11--SE, CVAP-268.
- [9] Clark MA, La Plante PC, Greenhill LJ. Accelerating Radio Astronomy Cross-Correlation with Graphics Processing Units. arXiv:1107.4264 [astro-ph.IM].
- [10] Khalil MI. Accelerating cross-correlation applications via parallel computing. International Journal of Image, Graphics and Signal Processing (IJIGSP). 2013;5(12):26.

© 2015 Khalil and Ibrahim; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)

<http://sciencedomain.org/review-history/11189>