

A Robust Template Matching Algorithm Based on Reducing Dimensions

Y. M. Fouda^{1,2}

¹College of Computer Science and Information Technology, King Faisal University, Al-Ahsa, Kingdom of Saudi Arabia

²Mathematics Department, Faculty of Science, Mansoura University, Mansoura, Egypt
Email: yfoudah@kfu.edu.sa, yfoudah@gmail.com

Received 10 February 2015; accepted 1 April 2015; published 2 April 2015

Copyright © 2015 by author and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Template matching is a fundamental problem in pattern recognition, which has wide applications, especially in industrial inspection. In this paper, we propose a 1-D template matching algorithm which is an alternative for 2-D full search block matching algorithms. Our approach consists of three steps. In the first step the images are converted from 2-D into 1-D by summing up the intensity values of the image in two directions horizontal and vertical. In the second step, the template matching is performed among 1-D vectors using the similarity function sum of square difference. Finally, the decision will be taken based on the value of similarity function. Transformation template image and sub-images in the source image from 2-D grey level information into 1-D information vector reduce the dimensionality of the data and accelerate the computations. Experimental results show that the computational time of the proposed approach is faster and performance is better than three basic template matching methods. Moreover, our approach is robust to detect the target object with changes of illumination in the template also when the Gaussian noise added to the source image.

Keywords

Image Processing, Template Matching, Vector Sum, Sum of Square Difference

1. Introduction

One of the most important issues in image analysis is automatically extracting useful information from an image. This information must be explicit and can be used in subsequent decision making processes. The more commonly used image analysis techniques include template matching and statistical pattern recognition. We can

classify the types of analysis we wish to perform according to function. There are two main types of functions we will wish to know about the scene in an image. First, we may wish to ascertain whether or not the visual appearance of objects is as it should be, *i.e.* we may wish to inspect the objects. The implicit assumption here is, of course, that we know what objects are in the image in the first place and approximately where they are. The second function of image analysis is location. If we don't find out the objects, we may wish to know where they are. The location of an object requires the specification of both horizontal and vertical coordinates. The coordinates may be specified in terms of the image frame of reference where distance is specified in terms of pixels.

Many of applications of computer vision simply need to know whether an image contains some previously defined object or whether a pre-defined sub-image is contained within a source image. The sub-image is called a template and should be an ideal representation of the pattern or object which is being sought in the image. The template matching technique involves the translations of the template to every possible position in the image and the evaluation of the measure of the match between the template and the image at that position. If the similarity measure is large enough, then the object can be assumed to be present. Various difference measures have different mathematical properties, and different computational properties have been used to find the location of template in the source image. The most popular similarity measures are the sum of absolute differences (SAD), the sum of squared difference (SSD), and the normalized cross correlation (NCC). Because SAD and SSD are computationally fast and algorithms are available which make the template search process even faster, many applications of gray-level image matching use SAD or SSD measures to determine the best match. However, these measures are sensitive to outliers and are not robust to variations in the template, such as those that occur at occluding boundaries in the image. However, although the NCC measure is more accurate, it is computationally slow. It is more robust than SAD and SSD under uniform illumination changes, so the NCC measure has been widely used in object recognition and industrial inspection such as in [1] and [2]. An empirical study of five template matching algorithms in the presence of various image distortions has found that NCC provides the best performance in all image categories [3].

Many techniques have been developed to increase the template matching process. Chen and Hung [4] classify these techniques into three classes. First class saves the computations by reducing the number of positions searched. On the other hand, class two try to reduce the computational cost of the matching error for each search position. The techniques in the first and second classes can be combined to further improve the efficiency and this kind of hybrid methods are classified as the third class. Techniques in class one perform the matching error calculations and comparison within a partial search set which is a subset of the complete search set. The efficiency of these techniques depends on the number of the selected search positions, while the resulted minimum matching error depends on how the search positions are selected. Techniques founded in [5]-[7] belong to this class. For example the coarse-to-fine strategy divides the search process into several search steps. Starting from the origin position (0, 0), SADs of several coarsely-spaced search positions are calculated and the one with the minimum SAD is selected as the new starting position of the next step. This procedure is repeated several times with smaller and smaller spacing between the search positions until the search positions with spacing of one pixel are examined. The final search position with minimum SAD is selected as the search result.

Techniques in class two accelerate the calculation of matching error for each search position. Instead of calculating matching error, SAD, these techniques calculates a partial matching error which need less computation than SAD and whose value is less than or equal to SAD. One simple technique in this class is to subsample the pixels in the matching blocks. For example, the partial sum of absolute difference can be calculated by using a quarter of pixels regularly subsampled in each matching block [8]. Another technique, called the partial distance method [9] and [10], can be used for speeding up the computation. For each search position, the partial sum of absolute difference is calculated. During the calculation process, if one of these partial sum absolute differences is larger than the minimum matching error computed so far, the calculation for this search position can be terminated and the calculation of the partial sum of absolute difference behind can be saved. The techniques in class three combine the techniques in the previous two classes, as described in [8] and [11] to further improve the efficiency. Another technique in this class is the hierarchical method [12] which first estimates the coarse result of the motion vector in the lower resolution image, and then refines the result in the higher resolution image within a small search region centering at the coarse result.

Reducing data in the image by converting the image from 2-D into 1-D is a new strategy in template matching introduced by [13]. He used ring projection transform to convert the 2-D template in a circular region into a 1-D gray level signal as a function of radius. Fouda [14] convert the 2-D images into 1-D vector information by summing-up the intensity values in all rows in the vertical direction. His method was sensitive to noise. In this

paper, we modify the 1-D template matching algorithm to making it more robust to noise. Instead of the image scanned in one direction we scan the image in two directions to increase the features in the 1-D information vector. Then the matching process can be done under some degree of noise. The rest of the paper is organized as follows. Section 2 introduces the basic techniques in template matching. The description of proposed algorithm and its complexity analysis will be described in Section 3. Simulation and comparison results for NCC, SAD, and coarse-to-fine (CTF) standard are reported in Section 4. Then we state conclusions in Section 5.

2. Template Matching Basic Techniques

In this section we introduce the most important three algorithms in template matching. Starting by normalized cross correlation technique is computationally expensive. But it is very robust against noise under different illumination conditions, so it has been widely used in object recognition and industrial inspection. The second is the sum of absolute difference algorithm which is best computationally than normalized cross correlation technique. But it is not robust to intensity and contrast variations, so it can be used in some applications such as feature tracking and block motion estimation in video compression. The last is the coarse-to-fine (CTF) technique which can be reduce the computational cost by using block averaging to decrease the spatial resolution of the template and the source image. It apply the low-resolution (“coarse”) template to the low-resolution source, and using the full-resolution (“fine”) template only when the coarse template’s degree of mismatch with the source is blew a given threshold.

2.1. Normalized Cross Correlation (NCC)

NCC has been commonly used as a metric in pattern matching to evaluate the degree of similarity between template and blocks in source image. The main advantage of the NCC over other techniques is that it is less sensitive to linear changes in the amplitude of illumination in the two compared template and block. Furthermore, the NCC is confined in the range between -1 and 1 . The setting of detecting threshold is much easier than other techniques. The NCC does not have a simple frequency domain expression. It cannot be directly computed using the more efficient fast Fourier transform in the spectral domain. Its computation time increase dramatically as the window size of the template gets larger [15].

In pattern matching applications, NCC working as follows: one finds an instance of a small template in a large source image by sliding the template window in a pixel by pixel basis, and computing the normalized correlation between them. The maximum value or peaks of the computed correlation values indicate the matches between a template and subimage in the source image. **Figure 1** shows the surface plot correlation coefficient values between the template T(a) in **Figure 5(a)** and blocks in the source image in **Figure 4(a)**. The experimental results

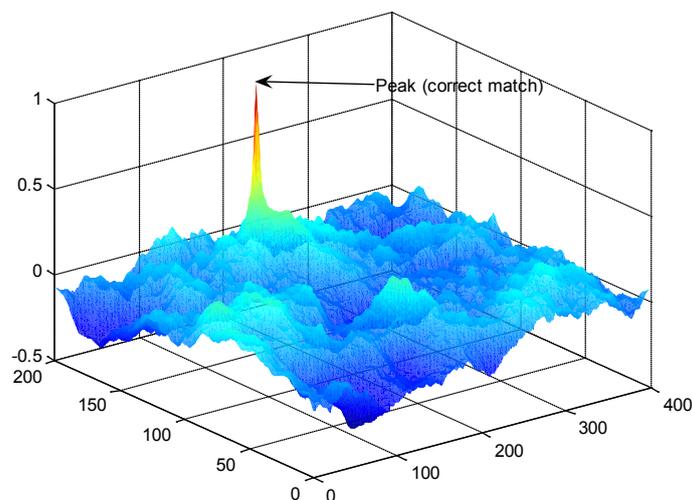


Figure 1. Surface plot of correlation coefficient between template T(a) in **Figure 5(a)** and green image in **Figure 4(a)**.

given that the peak value 0.999. This value in the surface at position (200, 175) in the source image is the correct match for template T(a).

The values of NCC used for finding matches of the template $T(i, j)$ of size $n \times n$ in a source image $S(x, y)$ of Size $p \times q$ is defined as:

$$\lambda(i, j) = \frac{\sum_{x=1}^n \sum_{y=1}^n (S(i+x, j+y) - \bar{S}(i, j))(T(x, y) - \bar{T})}{\sqrt{\sum_{x=1}^n \sum_{y=1}^n (S(i+x, j+y) - \bar{S}(i, j))^2 \sum_{x=1}^n \sum_{y=1}^n (T(x, y) - \bar{T})^2}}, \quad 1 \leq i < (p-n), 1 \leq j < (q-n) \quad (1)$$

where

$$\bar{S}(i, j) = \frac{1}{n \times n} \sum_{x=1}^n \sum_{y=1}^n S(i+x, j+y) \quad \text{and} \quad \bar{T} = \frac{1}{n \times n} \sum_{x=1}^n \sum_{y=1}^n T(x, y) \quad (2)$$

Direct computation of $\lambda(i, j)$ require $n \times n = n^2$ (addition/ multiplication) operations at each point (i, j) in the source image where $1 \leq i < (p-n)$, $1 \leq j < (q-n)$. Then the operations in Equation (1) is proportional to $n^2(p-n)(q-n)$. So the computational cost of NCC is $O(n^2(p-n)(q-n))$ which is very time consuming.

2.2. Sum of Absolute Difference (SAD)

The SAD is another commonly used similarity measure in pattern matching. According to [16] from a maximum likelihood perspective, SAD measure is justified when the additive noise distribution is exponential. Template matching applications often use the similarity measure SAD. However, this measure is not invariant to brightness and contrast variations which occur in many practical problems [17]. SAD is faster than NCC algorithm because it requires neither multiplication nor division operations.

Assume we have a template image $T(x, y)$ of size $n \times n$ is to be matched within a source image $S(x, y)$ of Size $p \times q$ where $(p > n)$ and $(q > n)$. For each pixel location (x, y) in the source image we compute an $(p-n) \times (q-n)$ matrix C where $C(x, y)$ is of the form

$$C(x, y) = \sum_{i=1}^n \sum_{j=1}^n f(S(x+i, y+j), T(i, j)), \quad 1 \leq x \leq (p-n), 1 \leq y \leq (q-n) \quad (3)$$

where $f(x, y) = |x - y|$, the minimum value of the matrix C given us the location of the best match for the template in the source image. If one of the values of C is zero, the local block is identical to the template. **Figure 2** shows the surface plot of matrix C when the SAD algorithm search about template T(a) in **Figure 7(a)** in the noise image in **Figure 6(b)**. We notice that the minimum value of matrix C is 32,384 which gives us the true position (100, 80) for T(a) in the source image.

The computation of $C(x, y)$ requires a number of operations proportional to the template area $(n \times n)$. These operations are computed for each (i, j) in the source image where $1 \leq i < (p-n)$, $1 \leq j < (q-n)$. Then the computational cost for SAD method is $O(n^2(p-n)(q-n))$ the same complexity in NCC algorithm but the SAD method is faster than NCC method because the number of operations (comparison) in SAD is less than number of operations (addition, multiplication, and divisions) in NCC for each position (i, j) in the source image.

2.3. Coarse-to-Fine (CTF)

The coarse-to-fine strategy, proposed by Rosenfeld and Vander Brug [18] [19], is well-known approach to reduce the computational cost of template matching. This strategy uses a low resolution template and its corresponding low resolution source image for initial coarse matching. Matching between high resolution template and original source image is applied for fine matching only when there is high similarity in the coarse matching. The coarse-to-fine technique works as follows: Upon creating the resolution levels for both template and source image, a search is conducted with the coarse template and its source image. The resulting pixel location provides a coarse location of the template pattern in the next lower level of the source image. Therefore, instead of per-

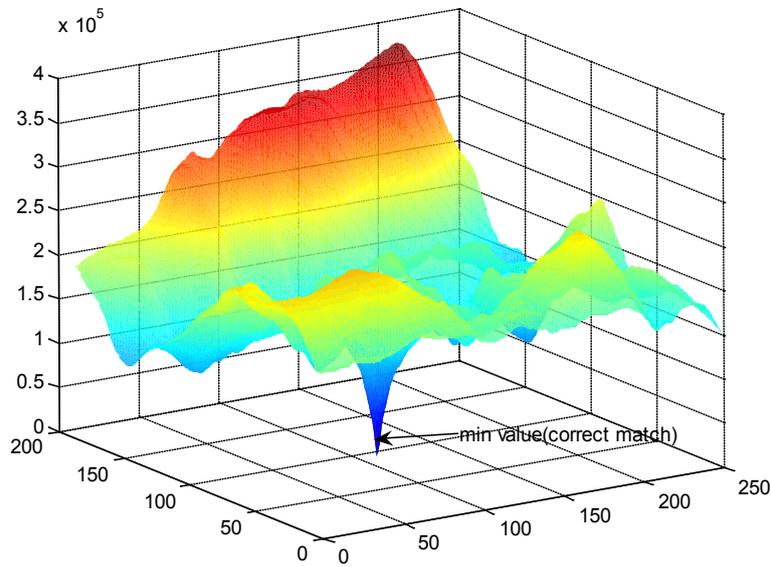


Figure 2. Surface plot of sum of absolute difference between template T(a) in Figure 7(a) and the noise two-man image in Figure 6(b).

forming a complete search in the next level resolution, one requires to only search a close vicinity of the area computed from the previous search. This sequence is iterated until the search in the original source image is searched.

The image resolution levels is based on reducing the dimensions of the image by a factor f , a predefined positive integer, at each resolution level. Assume we start with an image $g^0(x, y)$ and let $g^r(x, y)$ be the r th level of the resolution levels. The value of a pixel $g^r(x, y)$ on level r can be obtained from the average values of corresponding 2×2 neighboring pixels $g^{r-1}(2x, 2y)$, $g^{r-1}(2x+1, 2y)$, $g^{r-1}(2x, 2y+1)$, and $g^{r-1}(2x+1, 2y+1)$ on level $(r-1)$. In other words, $g^r(x, y)$ can be obtained by the following equation when $f = 2$:

$$g^r(x, y) = \frac{1}{4} \left(g^{r-1}(2x, 2y) + g^{r-1}(2x+1, 2y) + g^{r-1}(2x, 2y+1) + g^{r-1}(2x+1, 2y+1) \right) \quad (4)$$

Figure 3 shows the process by which the resolution is enhanced from the low resolution shown in **Figure 3(c)** where Equation (4) was applied for template matching T(b) in **Figure 7(b)**.

To compute the computational cost for coarse-to-fine method assume that the size of the source image is $p \times q$. For each level of the resolution process, calculation of the sum of 2×2 neighboring pixels requires $3(p-1)(q-1)$ additions (see Equation (4)). However, using the idea for fast calculation of the sum developed in [20], the complexity can be reduced to be $(2p-1)(q-1)$ additions for each level. If the block size is $n \times n$ the overhead for construction the resolution process is $(n/4)(2p-1)(q-1)$. Since there are $(p/n)(q/n)$ template blocks in the source image, the computation overhead for each template block is $(n/4)(2p-1)(q-1) / [(p/n)(q/n)] = n^3 (1/2 - p/2 - q/4 + 1/4)$ which is $O(n^3)$. This shows that the complexity of coarse-to-fine method is better than NCC and SAD techniques.

3. The Proposed One-Dimensional Algorithm (M1D)

Fouda [14] proposed a template matching technique depending on converting the image data from 2-D into 1-D. His method is efficient for the time but it is sensitive to the noise. The proposed M1D method is a modification for 1-D method to avoid its sensitive to noise. In our method, we try to make the template matching is more robust to noise. This can be done by adding a new features to information vector to be of size $1 \times (m+n)$ instead of $1 \times n$. First, the amount of data analyzed reduce by transforming the template image and all blocks in the

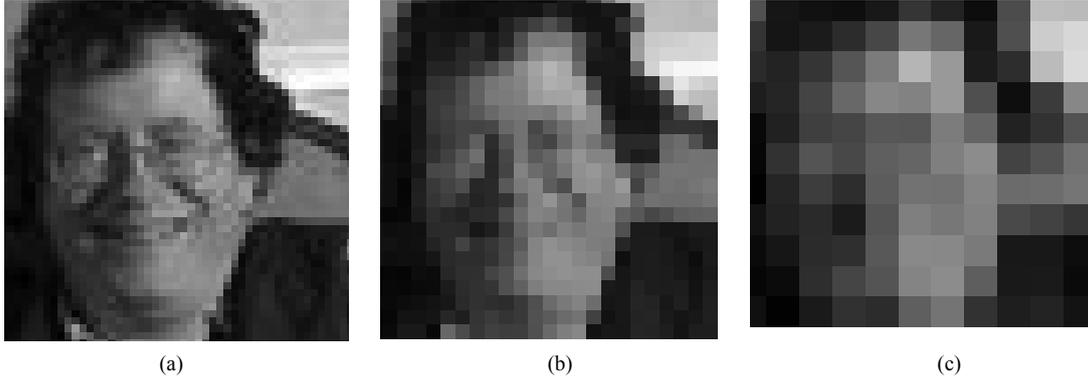


Figure 3. Three resolution levels for template T(b) in Figure 7 by coarse-to-fine method using Equation (4). (a) level 0; (b) level 1; (c) level 2.

source image which has the same size of template from two dimensional into one dimensional information vector. The new information vector data consists of two parts. The first part is the summation of all intensity values for each column in the 2-D image (see Equation (5)). The second part is the summation of all intensity values for each row in the 2-D image (see Equation (6)). In this case we scanned the 2-D image in two directions (vertical and horizontal) to make our technique more robust against noise. The new information vector (see Equation (7)) will be used in the matching process instead of 2-D image. Subsequently, this allows the search to be performed with fewer data, while still taking all pixels intensity values into account. Secondly, the sum of squared differences is used to measure the likeness between 1-D template and all possible 1-D blocks in the source image. Another measure can be used such as sum of absolute difference or Euclidean distance between template and all possible blocks in the source. Finally, the decision will be taken based on the likeness values. The block in the source with minimum similarity value will be the best match for template in the source.

To formalize the problem, suppose that we have a source image S of size $p \times q$ and template image T of size $m \times n$ where $m < p$ and $n < q$. The problem is to find the correct match for the template T in the source S .

First, we scan the template image in the vertical direction by adding up the intensity values of rows for each column. Then the first part NT_1 in 1-D information vector for the template is constructed and can be given by:

$$NT_1(i) = \left(\sum_{i=1}^m T(i,1), \sum_{i=1}^m T(i,2), \dots, \sum_{i=1}^m T(i,n) \right) \quad (5)$$

Next, we scan the template image in the horizontal direction by adding up the intensity values of columns for each row. Then the second part NT_2 for the template in the 1-D template information vector is constructed and given by the following formula:

$$NT_2(i) = \begin{pmatrix} \sum_{i=1}^n T(1,i) \\ \sum_{i=1}^n T(2,i) \\ \vdots \\ \sum_{i=1}^n T(m,i) \end{pmatrix} \quad (6)$$

Now, the template image $T(i, j)$ converted to 1-D template information vector NT of length $m+n$ which can be formulated by the following equation:

$$NT(i) = (NT_1(i), NT_2'(i)) \\ = \left(\sum_{i=1}^m T(i,1), \sum_{i=1}^m T(i,2), \dots, \sum_{i=1}^m T(i,n), \sum_{i=1}^n T(1,i), \sum_{i=1}^n T(2,i), \dots, \sum_{i=1}^n T(m,i) \right), \quad (7)$$

where $T(i, j)$ is the pixel value at location (i, j) of the template image.

Secondly, for each pixel (i, j) in the source image S , $1 \leq i < p - m$ and $1 \leq j < q - n$, we determine a block $B(i, j)$ of size $m \times n$. All these blocks are scanned in a similar way as in the template image to obtain NB_1 and NB_2 by the following formulas:

$$NB_1(i, j) = \left(\sum_{k=i}^{m+i-1} B(k, j), \sum_{k=i}^{m+i-1} B(k, j+1), \dots, \sum_{k=i}^{m+i-1} B(k, n+j-1) \right) \quad (8)$$

and

$$NB_2(i, j) = \begin{pmatrix} \sum_{k=j}^{n+j-1} B(i, k) \\ \sum_{k=j}^{n+j-1} B(i+1, k) \\ \vdots \\ \sum_{k=j}^{n+j-1} B(m+i-1, k) \end{pmatrix} \quad (9)$$

Then a 1-D information vector is constructed also as in the template image by the following formula:

$$\begin{aligned} NB(i, j) &= (NB_1(i, j), NB_2'(i, j)) \\ &= \left(\sum_{k=i}^{m+i-1} B(k, j), \sum_{k=i}^{m+i-1} B(k, j+1), \dots, \sum_{k=i}^{m+i-1} B(k, n+j-1), \sum_{k=j}^{n+j-1} B(i, k), \sum_{k=j}^{n+j-1} B(i+1, k), \dots, \sum_{k=j}^{n+j-1} B(m+i-1, k) \right), \\ &\quad \forall 1 \leq i < (p-m) \quad \text{and} \quad 1 \leq j < (q-n). \end{aligned} \quad (10)$$

where $B(k, j)$ is the pixel value at location (k, j) of the source image.

Thirdly, the likeness between template image and each corresponding block in the source are measured by sum of square difference distance between NT and NB . All these distances compute and store in new storage $C(i, j)$ where

$$\begin{aligned} C(i, j) &= \left(\sum_{k=i}^{m+i-1} B(k, j) - \sum_{i=1}^m T(i, 1) \right)^2 + \left(\sum_{k=i}^{m+i-1} B(k, j+1) - \sum_{i=1}^m T(i, 2) \right)^2 + \dots \\ &\quad + \left(\sum_{k=i}^{m+i-1} B(k, n+j-1) - \sum_{i=1}^m T(i, n) \right)^2 + \left(\sum_{k=j}^{n+j-1} B(i, k) - \sum_{i=1}^n T(1, i) \right)^2 \\ &\quad + \left(\sum_{k=j}^{n+j-1} B(i+1, k) - \sum_{i=1}^n T(2, i) \right)^2 + \dots + \left(\sum_{k=j}^{n+j-1} B(m+i-1, k) - \sum_{i=1}^n T(m, i) \right)^2, \end{aligned} \quad (11)$$

where $1 \leq i < p - m$ and $1 \leq j < q - n$.

The position (\bar{i}, \bar{j}) at which the smallest value of $C(i, j)$ is obtained corresponds the left upper corner of the best match for the template in our proposed method.

Now we discuss the computational cost considered in our proposed method. First the cost of converting the template and all blocks in the source into 1-D can be ignored in our proposed method, since for the template this need only be done once, and when we are calculating the computational cost for every position of the blocks in the source, the contribution due to summing the blocks become negligible compared with computing matrix C in Equation (11). The main complexity of our proposed method depends on the computations found in the matrix C in Equation (11). This equation can be written as:

$$C(i, j) = \sum_{k=i}^{m+i-1} \sum_{k'=j}^{n+j-1} f(B(k, j), T(i, k')), \quad 1 \leq i < (p-m) \quad \text{and} \quad 1 \leq j < (q-n) \quad (12)$$

for some function f where $f(x, y) = (x - y)^2$ by expanding $f(x, y)$ into $x^2 + y^2 - 2xy$ of the definition of matrix C : The matrices corresponding to the x^2 term and (respectively) y^2 term are easy to compute in $O(mn)$ time as in [21]. The matrix corresponding to the xy term can be computed in $O(\log(mn))$ as in [22]. Then for every value in the matrix C the complexity for xy term is $O((p - m)(q - n)\log(mn))$. Finally, if we take the maximum complexity for the three terms x^2 , y^2 , and xy , the complexity of our proposed method not exceed $O((p - m)(q - n)\log(mn))$ which is more efficient than coarse-to-fine method.

4. Experimental Results

In this section, we show the efficiency improvement of the proposed algorithm for one dimensional based template matching. In order to make the template matching more robust the proposed method added a new feature for the vectors made in one dimensional algorithm. To compare the efficiency of the proposed algorithm, we also implement the normalized cross correlation (NCC), sum of absolute difference (SAD), coarse to fine (CTF), and one dimensional (1-D) methods. These algorithms were implemented in a Matlab 7.0 on a Laptop with an Intel® Core™2 Duo CPU T7500 @ 2.20 GHz and 1.99 GB RAM. Two types of images are used for the testing purpose RGB images and gray scale images. Greens image of size 300×500 and its noisy version as the source image is a representative for RGB case (see [Figure 4\(a\)](#) and [Figure 4\(b\)](#)). Six template images of size 100×100 selected from the original greens image as shown in [Figure 5](#). The templates in [Figures 5\(d\)-\(f\)](#) are the brighter version (increase brightness from 22% to 30%) of the original templates in [Figures 5\(a\)-\(c\)](#), respectively. Two-man image of size 240×300 and its noisy version as the source is a representative for gray scale case (see [Figure 6\(a\)](#) and [Figure 6\(b\)](#)). Six template images of size 50×50 selected from the original two-man image as shown in [Figure 7](#). The templates in [Figures 7\(d\)-\(f\)](#) are the brighter version (increase brightness from 21% to 24%) of the original templates in [Figures 7\(a\)-\(c\)](#), respectively.



(a)



(b)

Figure 4. RGB image case (a) source greens image and (b) the noisy greens image added with Gaussian noise with variance 0.01.

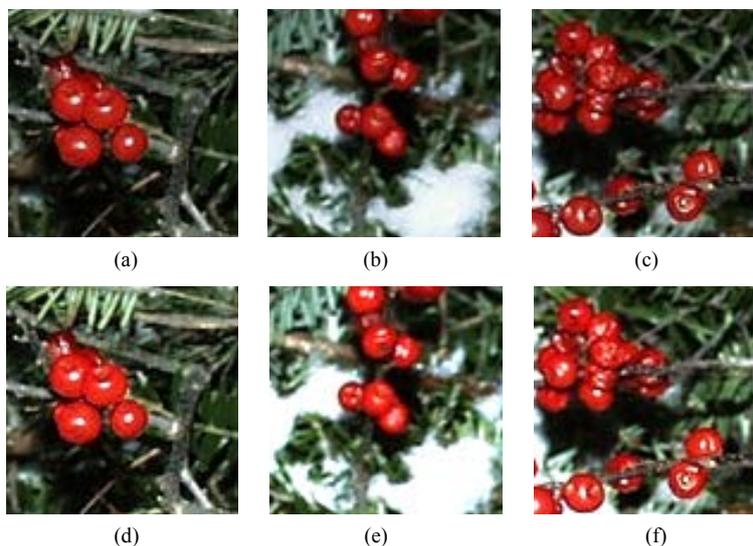


Figure 5. Six template from green image (a)-(c) three template sample of size 100×100 , (d) bright version of (a) by 22%, (e) bright version of (b) by 30%, and (f) bright version of (d) by 24%.

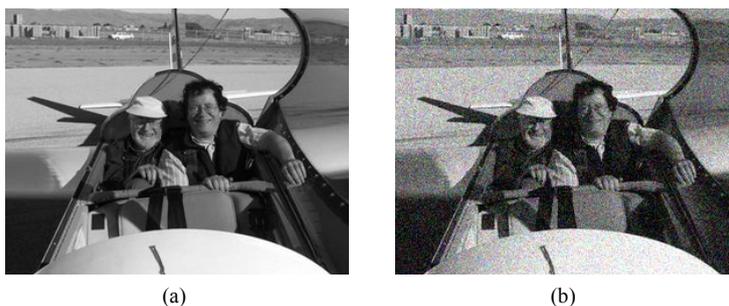


Figure 6. Gray scale image case (a) Source two-man image and (b) the noisy two-man image added with Gaussian noise with variance 0.01.

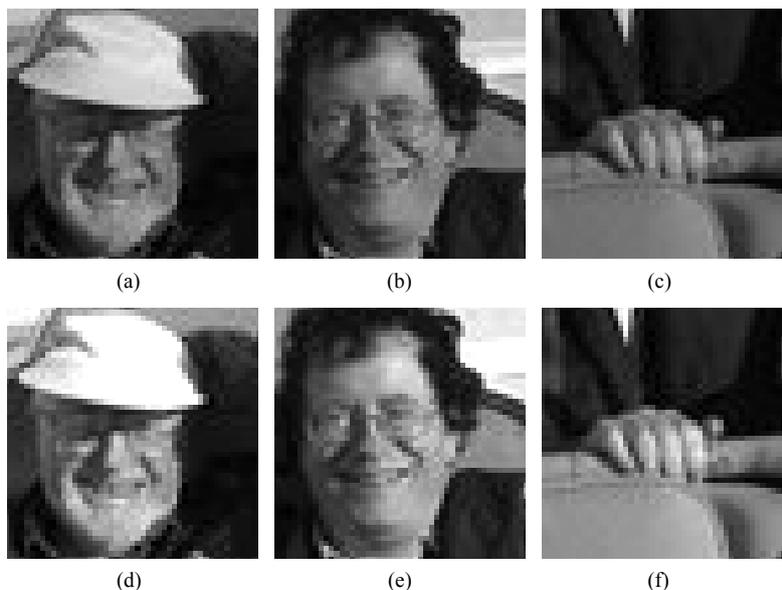


Figure 7. Six templates from two-man image (a)-(c) three template sample of size 50×50 , (d) bright version of (a) by 24%, (e) bright version of (b) by 21%, and (f) bright version of (d) by 23%.

Let us start by the RGB case through the greens image **Figure 4(a)** and its templates in **Figure 5**. To compare the robustness and the efficiency of the proposed algorithm, we add random Gaussian noises with mean 0 and variance 0.01 onto the source image as shown in **Figure 4(b)** and compare the performance of the pattern search on the noisy image. NCC, SAD, CTF, and the proposed MID algorithm are guaranteed to find the correct match from the source image, so we only focus on the comparison of search time required for these algorithms. The execution time required for these algorithms applied on the greens image in the clear case and noisy version are shown in **Table 1** and **Table 2**, respectively.

The results in **Table 1** and **Table 2** are several experiments were carried out with three different templates T(a), T(b), and T(c) and its brightness version in **Figure 5**. The true position for these templates in the source image are (200, 175), (50, 15), and (15, 150) respectively. When the brightness are increased for the three templates (T(d), T(e), and T(f) in **Table 1**) or the noises are added to the source (**Table 2**), All the algorithms are given the same position except 1-D algorithm given (200, 164), (50, 19), and (15, 160) respectively, which are a false positions (indicated by * in **Table 1** and **Table 2**). This ensures the fact that 1-D algorithm is sensitive to noise because it depends on a little numbers of features. But 1-D algorithm gives the best running time as in **Table 1** and **Table 2** when the source and template are free from noise. On the other hand the proposed MID algorithm gives a correct match when the noise are added to the template and/or source and in the same time, the running time is better than NCC, SAD, and CTF algorithms. **Figure 8** and **Figure 9** shows the performance of the proposed algorithm compared with other algorithms except 1-D algorithm which fail to find the correct match under noise conditions. It is clear that the proposed algorithm outperform the others in the two cases where the noise are added for template and/or added to the source.

Secondly the proposed algorithm was tested under the gray scale case through the two-man image **Figure 6(a)** and its templates in **Figure 7**. As in the RGB case the random Gaussian noise was added with mean 0 and variance 0.01 onto the two-man source image as shown in **Figure 6(b)**. **Table 3** and **Table 4** show the experimental results of applying different templates of size 50×50 as shown in **Figure 7**. The templates in **Figures 7(d)-(f)** are the brighter versions of the original templates in **Figures 7(a)-(c)**, respectively. All of these experimental results show the significantly improved efficiency of the proposed algorithm for the one dimensional based pattern matching compared to NCC, SAD, and CTF. For example in the clear case from **Table 1**, if we take the template T(d) we see that the improvement of SAD, CTF, 1-D, and MID are 64.47%, 72.58%, 95.65%, and 92.21% respectively. Then the 1-D is the best running time but he give a false match (indicated by *) for the templates

Table 1. Execution time by seconds of applying NCC, SAD, CTF, 1-D and the proposed MID algorithm with the six templates shown in **Figures 5(a)-(f)** and the source image shown in **Figure 4(a)**.

Seconds	T(a)	T(b)	T(c)	T(d)	T(e)	T(f)
NCC	70.25	71.03	69.98	69.73	69.76	74.03
SAD	45.61	46.06	45.92	45.65	45.98	45.7
CTF	23.59	49.53	34.33	23.59	49.89	33.91
1-D	8.17	6.41	8.66	6.65*	8.28*	6.61*
MID	12.36	11.86	11.72	12.33	10.22	11.56

Table 2. Execution time by seconds of applying NCC, SAD, CTF, 1-D and the proposed MID algorithm with the six templates shown in **Figures 5(a)-(f)** and the noise source image shown in **Figure 4(b)**.

Seconds	T(a)	T(b)	T(c)	T(d)	T(e)	T(f)
NCC	75.49	74.84	79.67	74.86	79.73	78.08
SAD	47.42	48.37	47.56	45.73	45.93	45.78
CTF	23.75	51.15	35.36	23.64	51.31	33.94
1-D	7.98	8.05	8.56	8.22*	6.59*	6.61*
MID	10.25	12.78	12.87	12.55	11.87	12.76

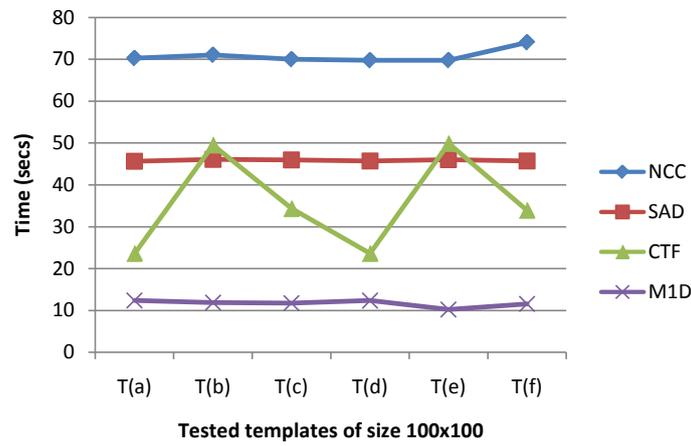


Figure 8. Performance of the proposed algorithm under clear greens image.

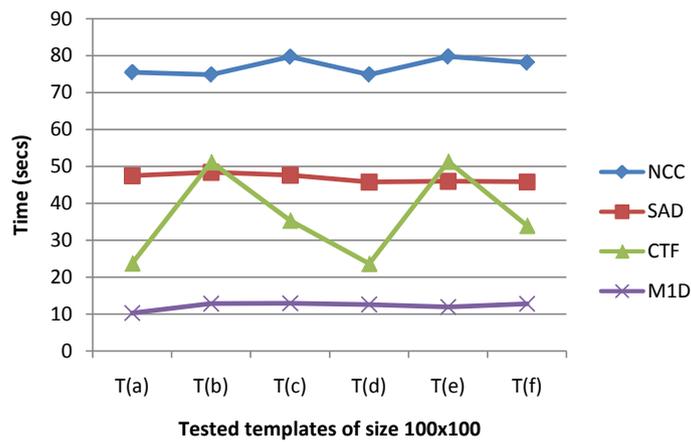


Figure 9. Performance of the proposed algorithm using greens image with noise.

Table 3. Execution time by seconds of applying NCC, SAD, CTF, 1-D and the proposed M1D algorithm with the six templates shown in Figures 7(a)-(f) and the source image shown in Figure 6(a).

Seconds	T(a)	T(b)	T(c)	T(d)	T(e)	T(f)
NCC	21.47	21.59	23.33	20.42	21.88	24.47
SAD	7.18	7.28	7.31	7.2	7.2	7.19
CTF	6.14	5.17	3.99	5.6	5.06	3.95
1-D	0.87	0.87	0.89	0.89*	0.91*	0.89*
M1D	1.64	1.66	1.61	1.59	1.62	1.61

Table 4. Execution time by seconds of applying NCC, SAD, CTF, 1-D and the proposed M1D algorithm with the six templates shown in Figures 7(a)-(f) and the noise source image shown in Figure 6(b).

Seconds	T(a)	T(b)	T(c)	T(d)	T(e)	T(f)
NCC	23.11	23.64	22.76	23.47	22.53	24.76
SAD	7.5	7.2	7.15	7.44	7.42	7.23
CTF	5.76	5.11	3.92	5.54	4.98	4.06
1-D	0.89	0.91	0.92	0.92*	0.92*	0.88*
M1D	1.58	1.56	1.56	1.59	1.59	1.58

T(d), T(e), and T(f). The proposed algorithm MID give improvement 92.21% and give a correct match under the noise conditions. In addition, the proposed algorithm gives a correct match under illumination changes it outperforms the other algorithms in the running time.

The execution time in **Table 3** and **Table 4** is less than execution time of **Table 1** and **Table 2** because the sizes of two-man image and its templates less than the size of green image and its templates. From **Table 3** we notice that 1-D algorithm is the best and it is give a correct result when the source and templates (T(a), T(b), and T(C)) are clean. But when the brightness are added to templates T(d), T(e), and T(f) the 1-D algorithm is fail to find a correct match. In this context, the proposed algorithm MID give a correct match for brightness templates and also it gives the best results compared with other algorithms. From **Table 4** when the noises are added to the source all algorithms given a correct match except the 1-D algorithm is fail when the brightness are added to the templates, but it gives the best result when templates are clean. Finally, we can say that, if the source and templates are clean 1-D gives a correct match and best running time. But if there is a noise in the source or in the templates the proposed algorithm gives a correct match and best results compared with other algorithms. **Figure 10** and **Figure 11** show the performance of the proposed algorithm compared with other algorithms except 1-D algorithm which fail to find the correct match under noise conditions. It can be found that our method is about 14 times faster than NCC method, about five times faster than SAD method, and about three and half times faster than CTF method in the total computation time. That the model one-dimensional information vector, due to the transformation process, allow estimation by simple sum calculation rather than a complex calculation. Therefore, the matching time of proposed algorithm is reduced.

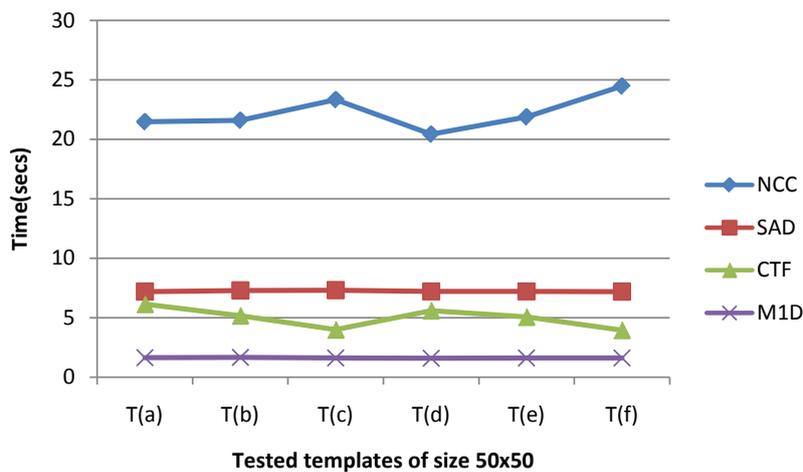


Figure 10. Performance of the proposed algorithm (MID) under clear two-man image.

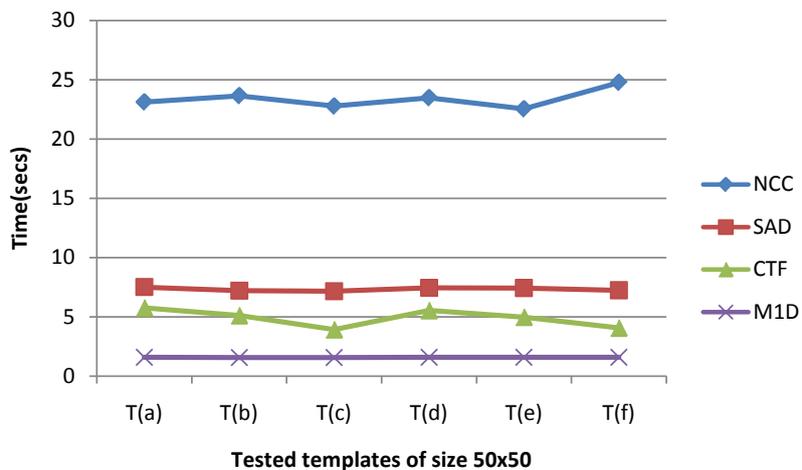


Figure 11. Performance of the proposed algorithm (MID) using two-man image with noise.

5. Conclusions

In this paper, we proposed a one-dimensional full searching algorithm for template matching. It depends on reducing image data form $m \times n$ to $m + n$. So it performs better than three basic two-dimensional algorithms in template matching (NCC, SAD, and CTF). The proposed algorithm scans the template and blocks in the source image in vertical and horizontal directions. Therefore, it is robust against noise when the changes of illuminations are occurred in the template image. Also experiments demonstrate that the proposed algorithm is robust to detect the object when the Gaussian noises are added to the source image.

It has been shown theoretically and experimentally that the computational cost of the proposed algorithm is $O((p-m)(q-n)\log(mn))$, where the size of the source image is $p \times q$ and the size of the template image is $m \times n$. The proposed algorithm resolves of the shortcomings of the image-based approach where the computational cost is huge, and expands the usefulness of template matching. To make the template matching more robust, future developments will take into consideration the correlation coefficient among sum vectors instead of the sum of square differences.

References

- [1] Du-Ming, T. and Chien-Ta, L. (2003) Fast Normalized cross Correlation for Detect Detection. *Pattern Recognition Letters*, **24**, 2625-2613. [http://dx.doi.org/10.1016/S0167-8655\(03\)00106-5](http://dx.doi.org/10.1016/S0167-8655(03)00106-5)
- [2] Costa, C.E. and Petrou, M. (2000) Automatic Registration of Ceramic Tiles for the Prop Use of Fault Detection. *Machine Vision and Applications*, **11**, 225-230. <http://dx.doi.org/10.1007/s001380050105>
- [3] Li, R., Zeng, B. and Liou, M.L. (1994) A New Three-Step Search Algorithm for Block Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **4**, 438-442. <http://dx.doi.org/10.1109/76.313138>
- [4] Chen, Y., Hung, Y. and Fuh, C. (2001) Fast Block Matching Algorithm Based on the Winner-Update Strategy. *IEEE Transactions on Image Processing*, **10**, 1212-1222. <http://dx.doi.org/10.1109/83.935037>
- [5] Koga, T., Inuma, K., Hirano, A., Iijima, Y. and Ishiguro, T. (1981) Motion Compensated Interframe Coding for Video Conferencing. *Proc. Nat. Telecommunications Conf.*, **4**, G5.3.1-G5.3.5.
- [6] Jain, J.R. and Jain, A.K. (1981) Displacement Measurement and Its Application in Interframe Image Coding. *IEEE Transactions on Communications*, **29**, 1799-1808. <http://dx.doi.org/10.1109/TCOM.1981.1094950>
- [7] Ghanbari, M. (1990) The Cross-Search Algorithm for Motion Estimation [Image Coding]. *IEEE Transactions on Communications*, **38**, 950-953. <http://dx.doi.org/10.1109/26.57512>
- [8] Liu, B. and Zaccarin, A. (1993) New Fast Algorithms for the Estimation of Block Motion Vectors. *IEEE Transactions on Circuits and Systems for Video Technology*, **3**, 148-157. <http://dx.doi.org/10.1109/76.212720>
- [9] Bei, C.D. and Gray, R.M. (1985) An Improvement for the Minimum Distortion Encoding Algorithm for Vector Quantization. *IEEE Transactions on Communications*, **33**, 1132-1133.
- [10] Gersho, A. and Gray, R.M. (1992) Vector Quantization and Signal Compression. Kluwer, Norwell.
- [11] Shi, Y.Q. and Xia, X. (1997) A Thresholding Multiresolution Block Matching Algorithm. *IEEE Transactions on Circuits and Systems for Video Technology*, **7**, 437-440. <http://dx.doi.org/10.1109/76.564124>
- [12] Nam, K.M., Kim, J.S., Park, R.H. and Shim, Y.S. (1995) A Fast Hierarchical Motion Vector Estimation Algorithm Using Mean Pyramid. *IEEE Transactions on Circuits and Systems for Video Technology*, **5**, 344-351. <http://dx.doi.org/10.1109/76.465087>
- [13] Lin, Y.-H. and Chen, C.-H. (2008) Template Matching Using the Parametric Template Vector with Translation, Rotation, and Scale Invariance. *Pattern Recognition*, **41**, 2413-2421. <http://dx.doi.org/10.1016/j.patcog.2008.01.017>
- [14] Fouda, Y.M. (2014) One-Dimensional Vector Based Template Matching. *International Journal of Computer Science and Information Technology*, **6**, 47-58. <http://dx.doi.org/10.5121/ijcsit.2014.6404>
- [15] Tsai, D. and Lin, C. (2003) Fast Normalized cross Correlation for Defect Detection. *Pattern Recognition Letters*, **24**, 2625-2631. [http://dx.doi.org/10.1016/S0167-8655\(03\)00106-5](http://dx.doi.org/10.1016/S0167-8655(03)00106-5)
- [16] Sebe, N., Lew, M.S. and Huijsmans, D.P. (2000) Toward Improved Ranking Metrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 1132-1143. <http://dx.doi.org/10.1109/34.879793>
- [17] Mahmood, A. and Khan, S. (2012) Correlation Coefficient Based Fast Template Matching through Partial Elimination. *IEEE Transactions on Image Processing*, **21**, 2099-2108. <http://dx.doi.org/10.1109/TIP.2011.2171696>
- [18] Rosenfeld, A. and Vanderbrug, G.J. (1977) Coarse-Fine Template Matching. *IEEE Transactions on Systems, Man and Cybernetics*, **7**, 104-107.

- [19] Vanderbrug, G.J. and Rosenfeld, A. (1977) Two-stage Template Matching. *IEEE Transactions on Computers*, **C-26**, 384-393. <http://dx.doi.org/10.1109/TC.1977.1674847>
- [20] Lee, C. and Chen, L. (1997) A Fast Motion Estimation Algorithm Based on the Block Sum Pyramid. *IEEE Transactions on Image Processing*, **6**, 1587-1591. <http://dx.doi.org/10.1109/83.641419>
- [21] Gonzalez, R.C. and Woods, R.E. (1992) Digital Image Processing. Addison-Wesley, Reading.
- [22] Atallah, M.J. (2001) Faster Image Template Matching in the Sum of the Absolute Value of Differences Measure. *IEEE Transactions on Image Processing*, **10**, 659-663. <http://dx.doi.org/10.1109/83.913600>