



An accelerating CPU based correlation-based image alignment for real-time automatic optical inspection [☆]



Chin-Sheng Chen ^{a,*}, Chien-Liang Huang ^a, Chun-Wei Yeh ^a, Wen-Chung Chang ^b

^a Graduate Institute of Automation Technology, National Taipei University of Technology, Taiwan, ROC

^b Department of Electrical Engineering, National Taipei University of Technology, Taiwan, ROC

ARTICLE INFO

Article history:

Received 9 December 2014

Received in revised form 3 July 2015

Accepted 11 September 2015

Available online 26 October 2015

Keywords:

Image alignment

Image pyramid search strategy

Streaming SIMD Extensions 2

Similarity measure

ABSTRACT

This paper proposes an accelerating correlation-based image alignment using CPUs for time-critical applications in automatic optical inspection (AOI). In order to improve computation efficiency, the image pyramid search scheme is combined with the parallel computation. The image pyramid search scheme is employed first to quickly find certain objects in both monochrome and color images with rotation, translation and scaling. Sub-pixel accuracy is then used to attain the more accurate results at the sub-pixel level. In our experimental results, rotation accuracy is smaller than 0.218°, and the speed is increased between 277 and 20,841 times. According to translation, rotation and scaling tests, the errors of rotation, translation and scaling are 0.2°, 2.07 pixel and 0.55%, respectively. These results show that the proposed method is suitable for dealing with correlation-based image alignment for time-critical applications in automatic optical inspection.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Recently, the demands of automatic optical inspection (AOI) have increased significantly as the quality control of manufacturing process had become increasingly important. Template matching is particularly useful for machine vision tasks, where an object within an image must be aligned with a model of the object. In general, two-dimensional (2D) matching under rigid transformations, i.e., translation and rotation, is usually applied in applications of AOI in which the model of the object is generated from an image of the object. In order to account for rotation and multi-object detection, the similarity measure should be calculated with rotated version of the template. However, this induces a significant increment in computational cost because several similarity calculations considering the multi-object detection with rotated template version must be done at each position of the detected region.

Based on correlation similarity measurement, the object model is systematically compared to the image using all allowable degrees of freedom of transformations in the matching phase. The most popular and robust correlation-based similarity measurement is the Normalized Cross Correlation (NCC), which is used to evaluate the similarity between the template image and the compared image. For template matching in AOI, the optimal matching results are indicated by the maximum

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. W-H Hsieh.

* Corresponding author at: Graduate Institute of Automation Technology, National Taipei University of Technology, 1, Sec. 3, Zhongxiao E. Rd., Taipei 10608, Taiwan, ROC. Tel.: +886 2 27712171x4325; fax: +886 2 87733217.

E-mail addresses: saint@ntut.edu.tw (C.-S. Chen), t7669026@ntut.edu.tw (C.-L. Huang), yehcw1009@gmail.com (C.-W. Yeh), wchang@ntut.edu.tw (W.-C. Chang).

NCC correlation, and the NCC coefficient is confined in the range between -1 and 1 . The NCC coefficient is equal to 1 when there is perfect matching, and it also means the two compared images are same.

To conduct the NCC, a template in a scene image by sliding the template window on a pixel-by-pixel basis, and the NCC coefficient is further computed. Therefore, it is a computationally intense problem that aims at finding an object in a scene image. As is well known, template matching based on a full search correlation metric is exhaustively time-consuming, so locating the best object in an image efficiently and quickly is always an important problem in matching process [1].

To solve this problem, numerous approaches have been recently proposed to facilitate this task. Though the several efficient methods are used, the computation burden is still too expensive to calculate the similarity between a template and a block in search image. To simplify the computation procedure, an elimination strategy based on the upper bounds has been applied in the full exhaustive search template matching procedure to rapidly skip the dissimilar candidates in monochrome level images [2] and color images [3]. For the elimination strategy performance, tight bounds are proposed to speed up the correlation computation in [4]. And the authors further propose a coarse-to-fine scheme and two stage techniques for large and small templates based on a partial elimination algorithm [5]. Other efficient searches with the multilevel winner update scheme [6], the branch-and-bound scheme [7] and weak classifiers [8] are also proposed to efficiently accelerate inference by avoiding unnecessary calculation of the NCC metric.

The other optimal manner considered to achieve parallel computing is single instruction multiple data (SIMD) by parallel computing CPUs. In the correlation-based similarity algorithm, SIMD executes repetitive arithmetic operations based on the multiplication and accumulation computations. In previous studies [9,10], the SSE and SIMD instructions are used to quickly calculate the multiplication and accumulation computations of NCC metric. Although efficient correlation computation based on parallel computing methods is proposed, the computation burden dramatically increases when the objects contain rotation and scaling of an image.

Although optimal methods based on the elimination strategy or other efficient search algorithms are proposed, these approaches are unable to satisfy AOI applications if the objects have rotation and scaling. The NCC methods with rotated angle are introduced to solve this problem [11,12]. In [11], the pre-computed score set from rotated templates to the original template are used to recognize the target object by skipping unnecessary computation in the first step and then estimating rotation angle of the target object in the refinement step. In the refinement step, piecewise linear models are used to evaluate the rotation angle by straight lines. However, its alignment accuracy depends on the rotation step angle in the pre-computed score set. Furthermore, bi-cubic interpolation in the matching scores is used to obtain the sub-pixel level location and rotation angle in both steps [12], and accuracy of rotation depends on the interval of the interpolation, so, it cannot guarantee the optimal results after the two step interpolations. In [13], a two stage method is proposed to locate the certain object and estimate the rotation angle of the certain object. In the first stage, the rotation invariant method, ring-projection transform, is used to recognize the possible locations of a template in the scene image. In the second stage, the least second moment is employed to estimate the rotation angle. However, the computation burden is still high in matching process by using ring-projection transform.

In contrast to there correlation-based methods, the gradient based NCC metric for detection of texture-less objects has been proposed [14–16]. Here, the efficient template matching method only considers the image gradients to detect objects, and further use a binary representation of the gradients. Then SIMD units are also applied to save the computations of gradient based NCC metric. The iterative closest point (ICP) is further employed to optimize the pose estimation in [17], but the ICP is impractical for the real-time application.

As mentioned above, many studies have proposed optimizing the template matching method, but these methods cannot provide a precise result when the object involves rotation, translation and scaling in time-critical applications.

In AOI applications, the geometric relationship between template image T and inspected image I is constricted, as shown in Fig. 1. There usually exists an indicated inspection pattern (such as the diamond patterns shown in Fig. 1) in the images to alleviate the computational burden for performing real-time template matching. Although the scaling variation between T and I is small because the working distance between the lens and the inspected target is fixed in AOI applications, the scaling still affects the accuracy of alignment results. Hence, the template matching problem for AOI can be formulated as finding the optimal 2-D rigid transformation between T and I , constrained by translation vector \vec{T} , the rotation angle, and the scaling factor.

This study is motivated by developing a real time correlation-based image alignment for monochrome and color images by which the target objects can be detected under rigid transformations and scaling. The method presented here not only can obtain the location of the template image within the inspection image but also can provide precise information on rotation and scaling. In addition, the proposed methods become not only very accurate, but also efficiently real-time oriented by a image pyramid search technique, Streaming SIMD Extensions 2 and sub-pixel accuracy estimation.

The remaining sections of this paper are organized as follows: Section 2 reviews the normalized cross correlation method. Section 3 presents the architecture of this proposed method. Section 4 presents the key processes in detail. Section 5 presents the sub-pixel alignment. Various experimental results are then discussed in Section 6. Finally, the conclusions of this paper are given in Section 7.

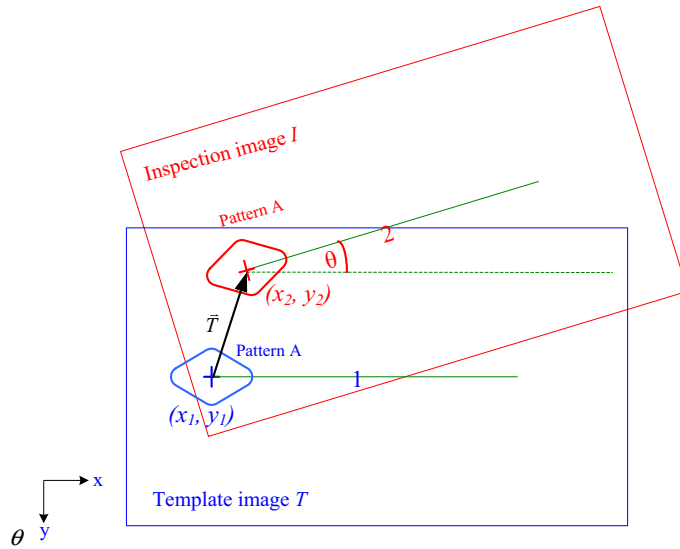


Fig. 1. Geometric relationship between template image T and inspection image I .

2. Review of Normalized Cross Correlation (NCC)

The correlation-based similarity measurement is applied to evaluate the degree of similarity between two compared images. The NCC metric is generally applied for both monochrome and color images, and the definitions of NCC for both monochrome and color images are first carried out [18]. In monochrome images, the NCC similarity coefficient can be used for image alignment between the template image $T(i,j)$ and the inspection image $I(x,y)$, is defined as:

$$\delta_g(x,y) = \frac{\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} [I(x+i,y+j) \cdot T(i,j)] - m \cdot n \cdot \mu_I \cdot \mu_T}{\sqrt{\sigma_I \cdot \sigma_T}} \tag{1}$$

where the size of the template image is $m \times n$; μ_T and μ_I are the gray level averages of the template image and the windowed search image, respectively, i.e.,

$$\mu_T = \frac{1}{m \cdot n} \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} T(i,j) \cdot T(i,j)$$

$$\mu_I = \frac{1}{m \cdot n} \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} I(x+i,y+j)$$

$$\sigma_I = \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} I^2(x+i,y+j) - m \cdot n \cdot \mu_I^2$$

$$\sigma_T = \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} T^2(i,j) - m \cdot n \cdot \mu_T^2$$

$\delta_g(x,y)$ is the degree of similarity between template image and the compared window image at coordinates (x,y) for gray level images.

On the other hand, color images are typically represented with RGB vectors that correspond to the red (R), green (G), and blue (B) channels. The extension of NCC similarity coefficient from gray level images to color images is defined by:

$$\delta_c(x,y) = \frac{\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} [I_c(x+i,y+j) \cdot T_c(i,j)] - 3 \cdot m \cdot n \cdot \mu_{Ic} \cdot \mu_{Tc}}{\sqrt{\sigma_{Ic} \cdot \sigma_{Tc}}} \tag{2}$$

where

$$\begin{aligned}
 T_c(i, j) &= (T_R(i, j), T_G(i, j), T_B(i, j)) \\
 I_c(x + i, y + j) &= (I_R(x + i, y + j), I_G(x + i, y + j), I_B(x + i, y + j)) \\
 \mu_{T_c} &= \frac{1}{3 \cdot m \cdot n} \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} [T_R(i, j) + T_G(i, j) + T_B(i, j)] \\
 \mu_{I_c} &= \frac{1}{3 \cdot m \cdot n} \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} [I_R(x + i, y + j) + I_G(x + i, y + j) + I_B(x + i, y + j)] \\
 \sigma_{I_c} &= \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} \|I_c(x + i, y + j)\|^2 - 3 \cdot m \cdot n \cdot \mu_{I_c}^2 \\
 \sigma_{T_c} &= \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} \|T_c(i, j)\|^2 - 3 \cdot m \cdot n \cdot \mu_{T_c}^2
 \end{aligned}$$

The number of computations required to calculate the NCC similarity coefficient for a color image is three times that for a gray-level image.

In NCC similarity measurement, the dot product term yielding the numerator of Eqs. (1) and (2) represents the cross correlation between the template and the compared window image, and its computation turns out to be the bottleneck in similarity measurement. In order to support both computation of speed and accuracy, the image pyramid search strategy, SIMD units and refinement adjustment are developed.

3. Architecture of the proposed method

Fig. 2 shows an overview of the architecture, which consists of offline model generation and online matching process. There are three input parameters defined by the user: the threshold of similarity Th_{\min} ; the detected range of rotation angle, R_{\min} and R_{\max} ; the detected range of scaling, S_{\min} and S_{\max} .

In offline model generation, the model is generated from the template image of the object to be matched in the online matching process. Here T_l is the multi-resolution images; l is an index of image pyramid levels, L is the maximum index of image pyramid levels; and $\mu_{I,T}$, $\sigma_{I,T}$ and μ_{I,T_c} , σ_{I,T_c} are the mean and the standard deviation of monochrome and color template images, respectively, on image pyramid level l .

In the online matching process, the multi-resolution image of an inspection image, I_l , is built by a fixed image pyramid level L from the offline model generation. Then the NCC similarity measurement with the image pyramid search strategy user defined parameters, the pre-calculated mean and standard deviation, of template to quickly find the objects. Meanwhile, the locations of objects with a similarity score exceeding the threshold of similarity Th_{\min} are registered and stored in object list. The translation estimation is performed by using vector \vec{T} between the centers of the target detected in T and I . Essentially, the image pyramid search strategy begins with an exhaustive search on the highest level and to extract a fixed number of objects from the best matches. Each object is then tracked down to the lowest level of the image pyramids, and the objects with the scores that exceed the threshold of similarity Th_{\min} are reported as the final match. The multi-resolution NCC metrics based on this image pyramid search strategy are further described in Section 4.2. The sub-pixel alignment is done by fitting a second order polynomial to the similarity scores in a $3 \times 3 \times 3$ neighborhood around the position which has maximum score.

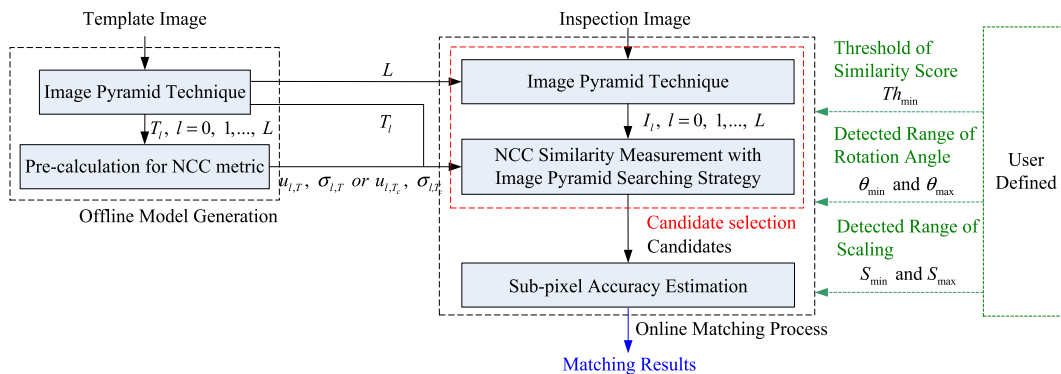


Fig. 2. Architecture of the proposed method.

4. Object selection

The proposed object selection is coarse-to-fine image alignment method. In the coarse level, rotation, translation and scaling of objects are detected by the image pyramid search strategy on a search image.

4.1. NCC similarity of rigid transformation

To consider the objects with the rigid transformation involved in NCC similarity coefficient calculation as shown in Fig. 2, the transformed models are generated by rotating and scaling the compared window image. The similarity coefficient for monochrome and color NCC metric are re-defined from Eq. (1) to be further used to estimate the pose parameters of each candidate as:

$$\delta_g(x, y, \theta) = \frac{\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} [I(x + \hat{i}, y + \hat{j}) \cdot T(i, j)] - m \cdot n \cdot \mu_{I,I}}{\sqrt{\left(\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} I^2(x + \hat{i}, y + \hat{j}) - m \cdot n \cdot \mu_{I,I}^2\right) \cdot \left(\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} T^2(i, j) - m \cdot n \cdot \mu_{I,T}^2\right)}} \quad (3)$$

where $\begin{bmatrix} \hat{i} \\ \hat{j} \end{bmatrix} = \begin{bmatrix} S \times \cos(\theta) & S \times -\sin(\theta) \\ S \times \sin(\theta) & S \times \cos(\theta) \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix}$, θ is rotation angle, S is scaling ratio, \hat{i} and \hat{j} are the shift position in coordinate (x, y) after rotation and scaling.

The gray level averages of the compared window image $\mu_{I,I}$ is modified by

$$\mu_{I,I} = \frac{1}{m \cdot n} \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} I(x + \hat{i}, y + \hat{j}) \quad (4)$$

Subsequently, extension of the NCC coefficient from monochrome images to color images is re-defined by

$$\delta_c(x, y, \theta) = \frac{\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} [I_c(x + \hat{i}, y + \hat{j}) \cdot T_c(i, j)] - 3 \cdot m \cdot n \cdot \mu_{I,I_c} \cdot \mu_{I,T_c}}{\sqrt{\left(\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} \|I_c(x + \hat{i}, y + \hat{j})\|^2 - 3 \cdot m \cdot n \cdot \mu_{I,I_c}^2\right) \cdot \left(\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} \|T_c(i, j)\|^2 - 3 \cdot m \cdot n \cdot \mu_{I,T_c}^2\right)}} \quad (5)$$

where \hat{i} and \hat{j} are the shift position in coordinate (x, y) after rotation φ and scaling S ; and the color average of the compared window image μ_{I,I_c} is modified by

$$\mu_{I,I_c} = \frac{1}{3 \cdot m \cdot n} \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} [I_R(x + \hat{i}, x + \hat{j}) + I_G(x + \hat{i}, x + \hat{j}) + I_B(x + \hat{i}, x + \hat{j})] \quad (6)$$

Therefore, the objects with rotation, translation and scaling are first selected and located by the NCC metric, either Eq. (3) or Eq. (5). Unfortunately, the computation burden dramatically increases when the objects contain rotation, translation and scaling in an image. Hence, the image pyramid technique [19], the SIMD unit and the image pyramid search strategy are applied to reduce the computational burden in the objects selection process, and the other basic idea of these techniques quickly find the object and rapidly eliminate the dissimilarity object on each image pyramid level.

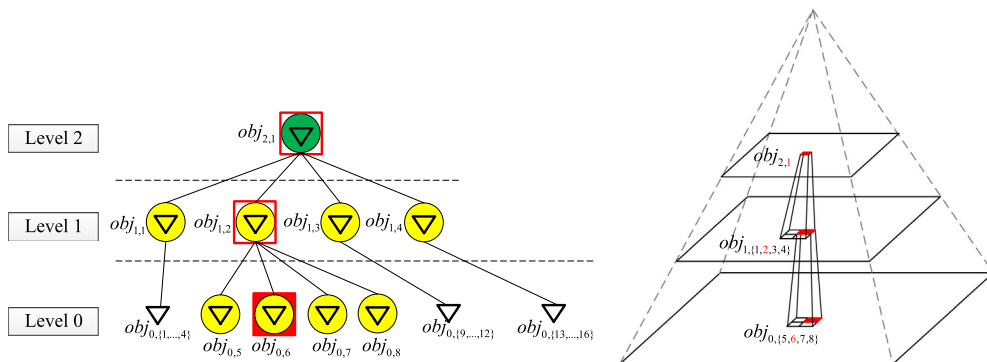


Fig. 3. Image pyramid search strategy.

4.2. Image pyramid search strategy

The image pyramid search framework allows such a target search. The image pyramid hierarchically splits the objects involving information of different image pyramid levels into disjoint subsets. By this strategy, large numbers of the objects can be discarded early during the search process by noticing that their similarity coefficient are lower than the pre-defined threshold of similarity Th_{\min} . Here, the similarity coefficient between the template and the compared window image is computed by Eq. (3) in monochrome image or Eq. (5) in color image.

Fig. 3 shows a case with three image pyramid levels study of search process. The search process starts at the objects on the highest pyramid level. The object with the rotation θ and scaling S at location (x,y) , on the highest pyramid level is searched by computing the similarity coefficient between the template and the compared window image. The object, $obj_{2,1} = \{\theta_{21}, S_{21}, (x_{21}, y_{21})\}$, on level 2, which is indicated by a green circle, is searched in Fig. 3. Any object with similarity score that exceeds the threshold of similarity Th_{\min} is stored in the object list, and the Th_{\min} should be set to the minimum expected object visibility. On the next lower pyramid levels, the search process is based on the objects that have been found on the previous pyramid level, such as the objects $obj_{1,\{1,2,3,4\}}$ that are denoted by a yellow circle are inherited from their parent object $obj_{2,1}$ with a 2×2 search area on image pyramid level 1 and the search rotation angle and scaling ratio are also inherited from $obj_{2,1} = \{\theta_{21}, S_{21}, (x_{21}, y_{21})\}$ with small range of rotation and scaling. Only one candidate, $obj_{1,2} = \{\theta_{12}, S_{12}, (x_{12}, y_{12})\}$, denoted by a red square, with similarity score exceeds the threshold of similarity Th_{\min} . This process is repeated until all match candidates have been tracked down to the lowest pyramid level. The final match result, $obj_{0,6}$, is denoted by a red solid square on the lowest level, as shown in Fig. 3. According to this search strategy, the Th_{\min} is set to skip unnecessary computations. This search strategy, it not only provides efficient matching for a single object, but also is easily extended to multi-object matching.

4.2.1. SIMD units

The correlation-based image alignment method, NCC, executes repetitive operations based on summation, multiplication and multiplication and accumulation (MAC) involved in the calculation of NCC metric. In particular, the calculation bottleneck of the NCC metric is the computation of the summations, multiplications and dot product terms. But the speed can be significantly increased with the SSE2 (Streaming SIMD Extensions 2) instructions that parallel the computation of the NCC metric by performing several multiplications and summations through a single SSE2 instruction. With the advantages of SSE2, the SSE2 instructions are utilized to optimize evaluation of the NCC coefficient when an object has rotation, translation and scaling. The SSE2 is an SIMD extension on the Intel SIMD processor supplementary instruction sets, where SIMD means that the one instruction is applied to multiple data. SSE2 contains 8 registers, called XMM0 to XMM7, each with a width of 16 bytes. That implies that those registers contain 2, 4, 8 or 16 variables with a corresponding width of 8, 4, 2, or 1 byte. Therefore, it can perform the operation on several variables in parallel. Next, Fig. 4 illustrates an example for computation of the arithmetic operation of summation, multiplication and MAC. Here, `__m128i` is a data type for integer arithmetical operations. In Fig. 4(a), `_T` and `_J` are variables of `__m128i` that operate in XMM registers, and they represent the data involved in unpacked in word data type loaded from the two different data sets. The summation of `_T` and `_J` in SSE2 instruction is expressed as:

$$\text{.sum} = \text{.mm_add_epi16}(_T, _J) \quad (7)$$

After executing the eight summations between corresponding items, the summation result is stored into the destination register `.sum`. In Fig. 4(b), `_T` and `_J` also represent the data involved in unpacked in word data type as loaded the two different data sets. The multiplication of `_T` and `_J` in SSE2 instruction is expressed as:

$$\text{.mul} = \text{.mm_mullo_epi16}(_T, _J) \quad (8)$$

After executing the eight multiplication between corresponding items, the result is stored into the destination register `.mul`. In Fig. 4(c), `_T` and `_J` also represent the data involved in unpacked word data type as loaded from the two different data sets. The dot product of `_T` and `_J` in SSE2 instruction is expressed as:

$$\text{.madd} = \text{.mm_madd_epi16}(_T, _J) \quad (9)$$

After executing the eight multiplications and four summations between corresponding items, the MAC result is stored in the destination register `.madd`.

This example shows that the summation and the MAC values of eight values are only executed as a single SSE2 instruction, and it can be computed very quickly by using SSE2 instructions.

4.2.2. Accelerating NCC calculation

As mentioned above, the calculation bottleneck of the NCC metric is the computation of the summations and dot product terms. In NCC metrics for monochrome and color images, Eqs. (3) and (5), the numerator and the denominator effectively use the summation, multiplication and MAC arithmetical operations, such as mean values, and square values and the dot-product terms. For instance, the size of template and compared windows images are 8×8 pixel, the computation burden of the summation, multiplication and MAC requires only 26, 58 and 58 instruction cycles by using Eqs. (7)–(9), respectively.

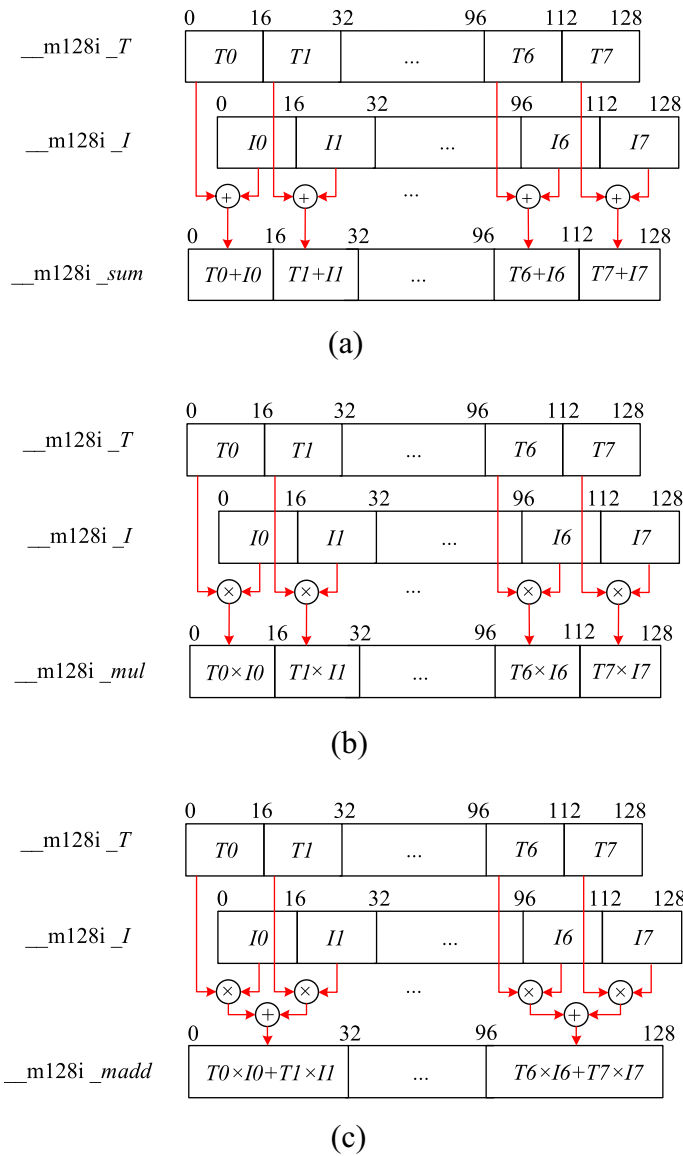


Fig. 4. Arithmetic operations of summation, multiplication and accumulation (MAC) in parallel instruction. (a) Arithmetic operation of summation. (b) Arithmetic operation of multiplication. (c) Arithmetic operation of multiplication and accumulation (MAC).

On the other hand, it requires 64, 640 and 703 instruction cycles without SSE2 instruction. Then for the similarity calculation in this example, the computation instruction cycles are 469 and 3736, with or without SSE2. This shows the computation advantage is about 8 times and also means the NCC metric can be computed very quickly using SSE2 instructions.

5. Sub-pixel alignment

In order to obtain higher accuracy with rotation θ and translation \vec{T} estimation, higher accuracy can be achieved by the idea of parabolic surface fitting. For this the initial transformed parameters are estimated in the final match on image pyramid level 0. To refine the position and rotation, the facet model principle is applied to fit the second order polynomial to the NCC similarity coefficient $\delta(x, y, \theta)$ in a $3 \times 3 \times 3$ neighborhood around the initial parameters. The neighborhood parameter of the translation and rotation are ± 1 and $\pm \Delta\alpha$. Here, the $\delta(x, y, \theta)$ represents the NCC similarity coefficient $\delta_g(x, y, \theta)$ if the template image is monochrome; similarly, the $\delta(x, y, \theta)$ represents the NCC similarity coefficient $\delta_c(x, y, \theta)$ if the template image is in color. The model of the surface that fits the set of the data points is considered as the following general equation:

$$\delta(x, y, \theta) = k_0x^2 + k_1y^2 + k_2\theta^2 + k_3xy + k_4x\theta + k_5y\theta + k_6x + k_7y + k_8\theta + k_9 \quad (10)$$

The solution for the coefficients k_0, \dots, k_9 can be determined by solving the equation

$$\mathbf{Az} = \mathbf{s},$$

where

$$\mathbf{A} = \begin{bmatrix} (x_{\delta_{\max}} - 1)^2 & (y_{\delta_{\max}} - 1)^2 & (\theta_{\delta_{\max}} - \Delta\alpha)^2 & (x_{\delta_{\max}} - 1) \cdot (y_{\delta_{\max}} - 1) & (x_{\delta_{\max}} - 1) \cdot (\theta_{\delta_{\max}} - \Delta\alpha) & (y_{\delta_{\max}} - 1) \cdot (\theta_{\delta_{\max}} - \Delta\alpha) & (x_{\delta_{\max}} - 1) & (y_{\delta_{\max}} - 1) & (\theta_{\delta_{\max}} - \Delta\alpha) & 1 \\ (x_{\delta_{\max}})^2 & (y_{\delta_{\max}} - 1)^2 & (\theta_{\delta_{\max}} - \Delta\alpha)^2 & (x_{\delta_{\max}}) \cdot (y_{\delta_{\max}} - 1) & (x_{\delta_{\max}}) \cdot (\theta_{\delta_{\max}} - \Delta\alpha) & (y_{\delta_{\max}} - 1) \cdot (\theta_{\delta_{\max}} - \Delta\alpha) & (x_{\delta_{\max}}) & (y_{\delta_{\max}} - 1) & (\theta_{\delta_{\max}} - \Delta\alpha) & 1 \\ (x_{\delta_{\max}} + 1)^2 & (y_{\delta_{\max}} - 1)^2 & (\theta_{\delta_{\max}} - \Delta\alpha)^2 & (x_{\delta_{\max}} + 1) \cdot (y_{\delta_{\max}} - 1) & (x_{\delta_{\max}} + 1) \cdot (\theta_{\delta_{\max}} - \Delta\alpha) & (y_{\delta_{\max}} - 1) \cdot (\theta_{\delta_{\max}} - \Delta\alpha) & (x_{\delta_{\max}} + 1) & (y_{\delta_{\max}} - 1) & (\theta_{\delta_{\max}} - \Delta\alpha) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (x_{\delta_{\max}} + 1)^2 & (y_{\delta_{\max}} + 1)^2 & (\theta_{\delta_{\max}} + \Delta\alpha)^2 & (x_{\delta_{\max}} + 1) \cdot (y_{\delta_{\max}} + 1) & (x_{\delta_{\max}} + 1) \cdot (\theta_{\delta_{\max}} + \Delta\alpha) & (y_{\delta_{\max}} + 1) \cdot (\theta_{\delta_{\max}} + \Delta\alpha) & (x_{\delta_{\max}} + 1) & (y_{\delta_{\max}} + 1) & (\theta_{\delta_{\max}} + \Delta\alpha) & 1 \end{bmatrix}$$

$$\mathbf{z} = [k_0 \ k_1 \ k_2 \ k_3 \ k_4 \ k_5 \ k_6 \ k_7 \ k_8 \ k_9 \ 1]^T$$

and

$$\mathbf{s} = \begin{bmatrix} S(x_{\delta_{\max}} - 1, y_{\delta_{\max}} - 1, \theta_{\delta_{\max}} - \Delta\alpha) \\ S(x_{\delta_{\max}}, y_{\delta_{\max}} - 1, \theta_{\delta_{\max}} - \Delta\alpha) \\ S(x_{\delta_{\max}} + 1, y_{\delta_{\max}} - 1, \theta_{\delta_{\max}} - \Delta\alpha) \\ \vdots \\ S(x_{\delta_{\max}} + 1, y_{\delta_{\max}} + 1, \theta_{\delta_{\max}} + \Delta\alpha) \end{bmatrix}$$

where $x_{\delta_{\max}}$, $y_{\delta_{\max}}$ and $\theta_{\delta_{\max}}$ are obtained from the final match on image pyramid level 0, respectively, as described in Section 4.2. There are 10 unknown values in the coefficient matrix \mathbf{z} , and 27 correlation data points are supplied by the 27 neighbors surrounding the initial parameters $(x_{\delta_{\max}}, y_{\delta_{\max}}, \theta_{\delta_{\max}})$, as given in \mathbf{s} . Since \mathbf{A} is a rectangular matrix, a least-squares regression must be used to solve \mathbf{z} , and it can be re-written as:

$$\mathbf{A}^T \mathbf{Az} = \mathbf{A}^T \mathbf{s} \quad (11)$$

Since $\mathbf{A}^T \mathbf{A}$ is a square matrix, the following equation yields the ten unknown parameters as:

$$\mathbf{z} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{s} \quad (12)$$

The refinement can finally be obtained by analytically computing the maximum of Eq. (10). For calculating, Eq. (10) can be re-written as:

$$\delta(x, y, \theta) = [x \ y \ \theta] \begin{bmatrix} k_0 & \frac{1}{2}k_3 & \frac{1}{2}k_4 \\ \frac{1}{2}k_3 & k_1 & \frac{1}{2}k_5 \\ \frac{1}{2}k_4 & \frac{1}{2}k_5 & k_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + [x \ y \ \theta] \begin{bmatrix} k_6 \\ k_7 \\ k_8 \end{bmatrix} + k_9 \quad (13)$$

Finally, the sub-pixel precise position and rotation is obtained by solving Eq. (14). The values for (x, y, θ) can be solved

$$\nabla \delta(x, y, \theta) = \begin{bmatrix} 2k_0 & k_3 & k_4 \\ k_3 & 2k_1 & k_5 \\ k_4 & k_5 & 2k_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} k_6 \\ k_7 \\ k_8 \end{bmatrix} = 0 \quad (14)$$

Finally, the sub-pixel precise position and rotation of the maximum is obtained by solving Eq. (14) with respect to (x, y, θ) . The values for (x, y, θ) can be solved by a 3×3 linear equation system:

$$\begin{bmatrix} x^* \\ y^* \\ \theta^* \end{bmatrix} = \begin{bmatrix} 2k_0 & k_3 & k_4 \\ k_3 & 2k_1 & k_5 \\ k_4 & k_5 & 2k_2 \end{bmatrix}^{-1} \begin{bmatrix} -k_6 \\ -k_7 \\ -k_8 \end{bmatrix} \quad (15)$$

The solution of Eq. (15), noted as (x^*, y^*, θ^*) , are the results of (x, y, θ) with sub-pixel accuracy in the estimation. It should be noted that since this is a closed form solution for the maximum value, the computation is deterministic. The accuracy of this refinement will be further discussed in the following section.

6. Performance

In this section, a series of experimental results show the performance of the proposed method. Various images are used to verify our algorithm in different experiments. As mentioned in Section 3, several parameters are designed for our proposed method. The similarity threshold Th_{\min} is a critical value, since it impacts the performance directly. There are two real world AOI cases are shown in Fig. 11. Here, the certain objects should be recognized and the corresponding poses of the objects are

further estimated. In those cases, the 1000 real captured images are obtained for setting the best Th_{\min} . As a thumb of rule, we set the Th_{\min} as 0.75 to guarantee a recognition rate higher than 99.9%. In this case, the recognition rates are 99.9% and 100% for Fig. 11(a) and (b), respectively. Only one of the 36 objects in a specified capture image is failed to be recognized in Fig. 11(a). The setting of other matching parameters, the rotation range, R_{\max} and R_{\min} , the scaling range, S_{\min} and S_{\max} are dependent on the requirement of applications. In our experiment, the setting of all parameters is defined before the test. The results are shown and discussed in the following subsections. To verify the proposed method, the results are compared with Ref. [13] and the basic NCC template matching with rotation, called NCC_R. All of the experiments were performed in Visual Studio 2008 on a personal computer (PC) with an Intel Core i7 3.4 GHz CPU and 8 GB of memory.

6.1. Rotation estimation

Fig. 5 shows the rotation test images, which are used for evaluating the rotation accuracy of the proposed rotation estimation technique. For rotation accuracy estimation, the synthetic 72 rotated images are generated by rotating the original image with rotation angles from 0° to 359° by 5° step increments, and the parameters of user defined, $Th_{\min} = 0.75$, $R_{\min} = -180$, $R_{\max} = 180$, $S_{\min} = 1$, $S_{\max} = 1$. The proposed method outperforms the others two methods in terms of rotation accuracy under various rotation images. Detailed comparison results are shown in Table 1. To provide an overall accuracy evaluation, two performance indices, the mean and the standard deviation, are used to quantitatively show the performance that computed by 72 synthetic rotated images.

As seen in Table 1, the errors of mean, standard deviation in all test cases that are conducted by our proposed technique are less than 0.218° and 0.307° . Even though the error of standard deviation of NCC_R in Fig. 5(b) and the errors of mean and standard deviation of Ref. [13] in Fig. 5(c) are close to our proposed method. In other cases, our method is superior to other competing methods. This indicates that proposed method is robust and can provide more reliable and accurate results, without being restricted to any special cases.

6.2. Translation, rotation and scaling (TRS)

In this experiment, the test images involved translation, rotation and scaling used to verify the proposed method. Here, only the proposed method was implemented. The test images are shown in Fig. 6 and the test parameters of this experiment are listed in Table 2. In the TRS test, the rotation angle of test parameters are 30° and 45° , the scaling range of test parameters is from 80% to 120%, increasing by steps of 10%. Those 20 test images are used to evaluate the performance. To evaluate the performance for translation, the Euclidean distance between the original translation and the estimated translation is employed to evaluate the result in translation, and the overall errors of the proposed method are shown in Table 5. And the user-defined parameters are $Th_{\min} = 0.75$, $R_{\min} = -180$, $R_{\max} = 180$, $S_{\min} = 0.7$, $S_{\max} = 1.3$. Detailed results are shown in Tables 3 and 4.

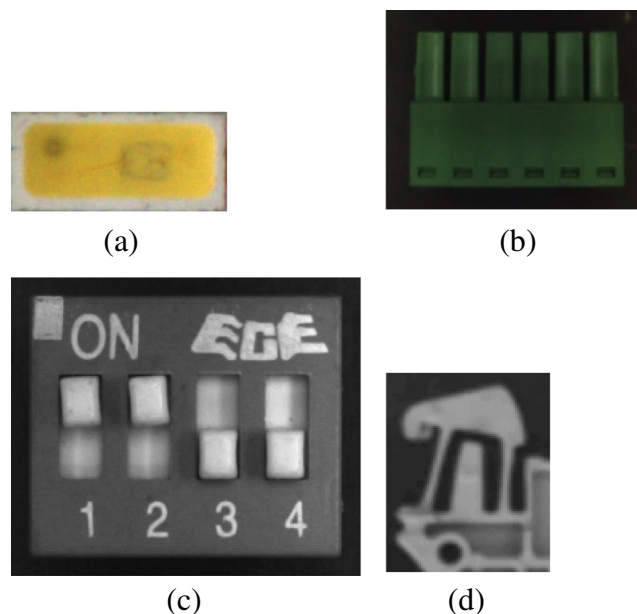
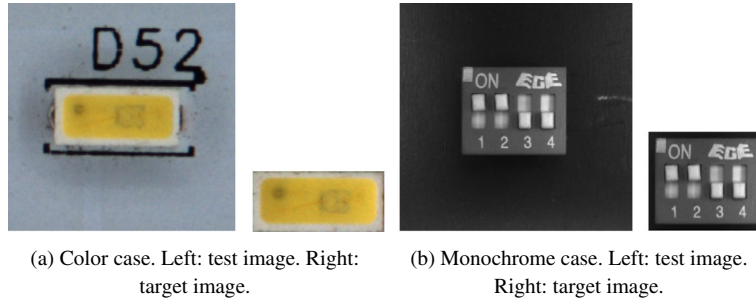


Fig. 5. Images for rotation test. Sizes of (a–d) are 148×68 , 258×196 , 262×224 , 80×90 .

Table 1

Errors of rotation accuracy produced by the proposed method and the others comparative methods.

		Proposed method	Ref. [13]	NCC_R
Fig. 5(a)	Mean (°)	0.169	0.267	0.387
	Std. dev. (°)	0.156	0.251	0.408
Fig. 5(b)	Mean (°)	0.218	1.228	0.221
	Std. dev. (°)	0.307	1.323	0.077
Fig. 5(c)	Mean (°)	0.168	0.168	0.392
	Std. dev. (°)	0.267	0.267	0.127
Fig. 5(d)	Mean (°)	0.167	0.717	0.226
	Std. dev. (°)	0.273	1.105	0.152

**Fig. 6.** Test images for TRS test. The sizes of the TRS test color and monochrome images are 270×270 and 519×507 . The sizes of the TRS test target image are 148×68 and 262×224 corresponding to color and monochrome image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)**Table 2**

Parameters of TRS test.

	Color case	Monochrome case
Translation	(132,137)	(249,244)
Rotation	30° and 45°	30° and 45°
Scaling	80–120%	80–120%

Table 3

Matching results of Fig. 6(a) in TRS test.

Original scaling (%)	Estimated		
	Rotation angle (°)	Scaling (%)	Translation (pixel, pixel)
80	30.004	82.1	(133.16,136.67)
	44.904	79.8	(133.57,137.07)
90	30.045	89.9	(133.29,136.7)
	44.95	89.9	(134.56,134.08)
100	30.263	100	(133.63,138.13)
	44.624	100	(134.62,136.08)
110	30.054	109.5	(132.64,137.96)
	44.444	108.3	(133.59,134.38)
120	30.036	119.6	(132.04,137.71)
	44.444	119.6	(133.46,134.28)

It can be seen in Table 5 that the mean rotation errors of color and monochrome cases are 0.2° and 0.11° , and the mean scaling errors of color and monochrome cases are 0.55% and 0.37%, and then the errors of mean translation in each cases are 2.07 pixel and 1.93 pixel, respectively. For these results, the proposed method can provide precisely matching results when the object simultaneously involved rotation, translation and scaling.

Table 4
Matching results of Fig. 6(b) in TRS test.

Original scaling (%)	Estimated		
	Rotation angle (°)	Scaling (%)	Translation (pixel, pixel)
80	29.688	82	(251.26, 246.87)
	45.073	80.1	(250.49, 246.03)
90	30.236	88.3	(251.93, 244.57)
	45.11	89.5	(250.1, 244.15)
100	30.036	100	(248.89, 243.47)
	45.09	99.6	(249.19, 243.21)
110	30.036	109.8	(248.78, 242.93)
	45.13	107.8	(249.44, 241.82)
120	30.036	119.5	(248.44, 242.73)
	45.11	119.5	(249.12, 241)

Table 5
Errors of the proposed method in TRS test.

Error type		Fig. 6(a)	Fig. 6(b)
Rotation (°)	Mean (°)	0.2	0.11
	Std. dev. (°)	0.2	0.08
Translation (pixel)	Mean (pixel)	2.07	1.93
	Std. dev. (pixel)	1	1.06
Scaling (%)	Mean (%)	0.55	0.37
	Std. dev. (%)	0.7	0.64

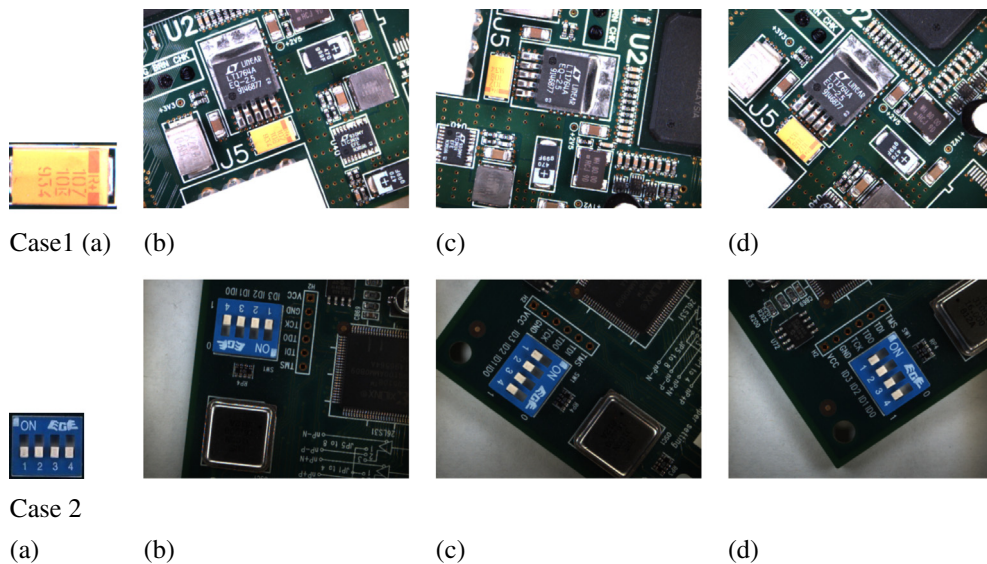


Fig. 7. Case 1 and Case 2 of real captured color test images. The sizes of real captured color images (b–d) are 800×600 , and the sizes of corresponding pattern image (a) are 138×85 and 209×185 . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.3. Efficiency in real captured images

The images are captured when an object with arbitrary moving and rotating for further experiments. To compare the computational burden of the proposed method and the other comparative methods, Ref. [13] and NCC_R, the execution time is used as a performance index. There are six kinds of captured images, color images and monochrome images that are employed to evaluate the computation burden, and these are shown in Figs. 7–11 and corresponding case 1 to case 6. In this experiment, the user-defined parameters are $Th_{\min} = 0.75$, $R_{\min} = -180$, $R_{\max} = 180$, $S_{\min} = 1$, $S_{\max} = 1$. The computation results are shown in Table 6.

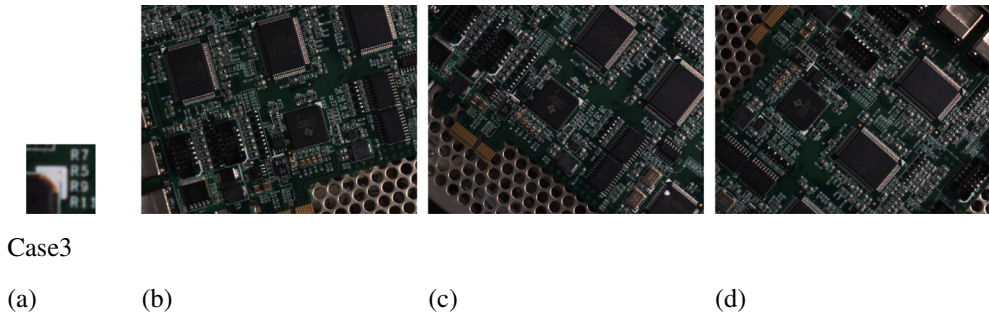


Fig. 8. Case 3 of real captured color test images. The sizes of real captured color images (b–d) are 1360×1024 , and the sizes of corresponding pattern image (a) is 70×70 . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

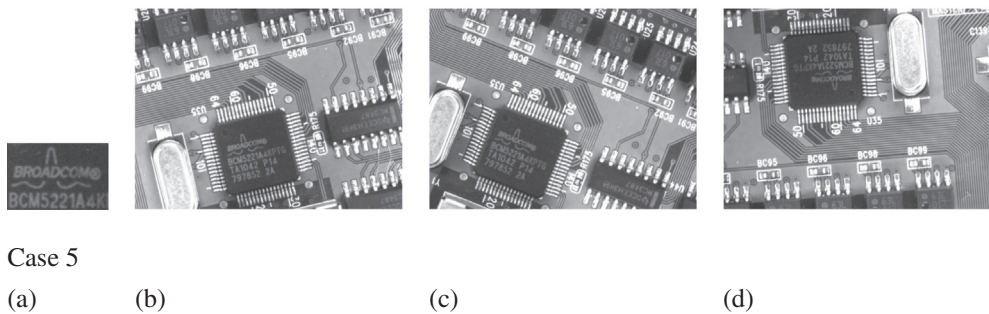
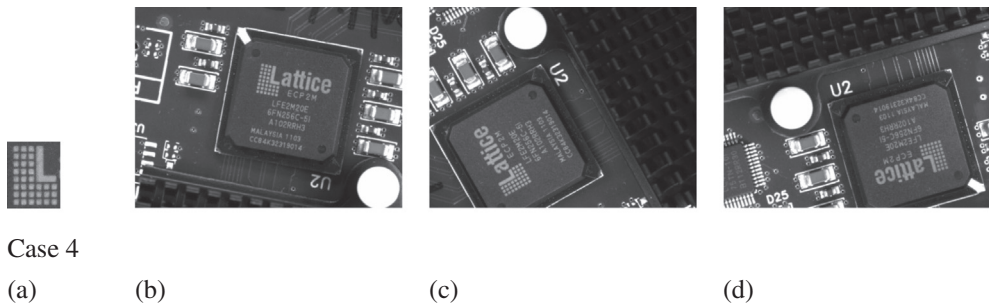


Fig. 9. Case 4 and Case 5 of real captured monochrome test images. The sizes of real captured monochrome images are 800×600 , and the sizes of corresponding pattern image are 80×101 and 141×93 .

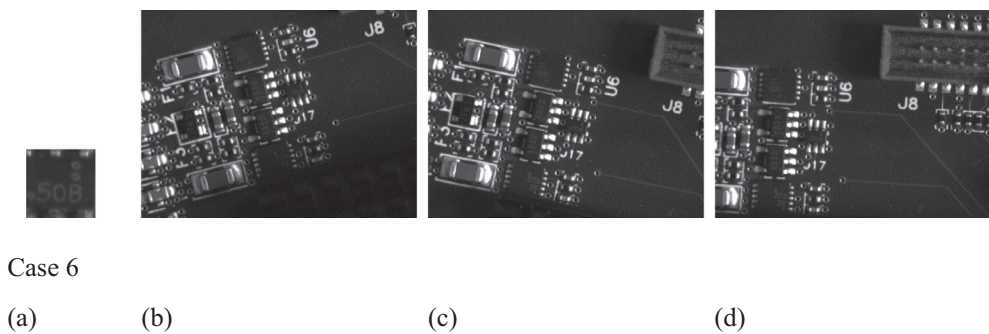
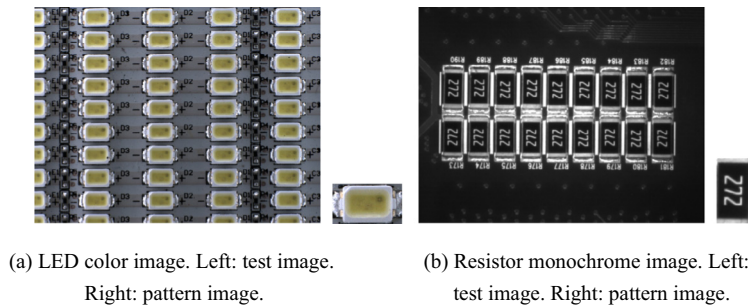


Fig. 10. Case 6 of real captured monochrome test images. The sizes of real captured monochrome images (b–d) are 1360×1024 , and the sizes of corresponding pattern image (a) is 70×70 .



(a) LED color image. Left: test image. Right: pattern image.
 (b) Resistor monochrome image. Left: test image. Right: pattern image.

Fig. 11. Multi-object test images. The sizes of real captured images are 800×600 , and the sizes of corresponding pattern image are 95×55 and 55×100 .

Table 6

The efficiency compare with proposed method, Ref. [13] and NCC_R.

	Execute time (ms)					
	Color			Monochrome		
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Proposed method	57.15	21.43	182.84	37.93	50.65	94.81
NCC_R	15848.92	446624.2	195262.2	34553.69	79072.63	74354.19
Ref. [13]	57063.82	129747.2	189988.5	15176.34	27818.6	32241.23

Table 7

The efficiency of multi-object detection.

Similarity ranking	Execute time (ms)			
	1	5	10	15
LED image	74.29	74.35	74.2	74.25
Resistor Image	66.86	66.6	66.9	66.75

From Table 6 it can be noted that proposed method is notably more efficiency than other compared methods. The computation advantage range comparing to Ref. [13] and NCC_R is between 277 and 20,841 times. In median size of template cases, Case 1, Case 2, Case 4 and Case 5, the computation advantages comparing to NCC_R are 227, 20,841, 910 and 1561 times. On the other hand, the computational advantages are 998, 6054, 400 and 549 times in monochrome cases, respectively. For further comparison with small template cases, Case 3 and Case 6, with color and monochrome images, are used to evaluate the performance. In small cases, the speed-up range is between 340 and 1067 times. Moreover, the proposed method is always faster than the Ref. [13] and the NCC_R least over 340 times despite on small template size.

6.4. Multi-object detection

This experiment is used to verify the proposed method is still efficient when more objects are located in test image. Fig. 11 shows the multi-object test images. There are totally 36 and 18 objects located in Fig. 11(a) and (b), respectively. Here, the parameters of user defined are $Th_{min} = 0.75$, $R_{min} = -180$, $R_{max} = 180$, $S_{min} = 0.9$ and $S_{max} = 1.1$. In this experiment, the top 1, 5, 10 and 15 similarity ranking of targets in Fig. 11(a) and (b) are selected for evaluation by the efficient process in multi-object detection. The results of multi-object detection are shown in Table 7.

As shown in Table 7, it can be seen that the computational burden is not increased when detection number of object is increased. In order to achieve multi-object detection, the all of possible candidates are tracked down to lowest image pyramid and the possible candidates are sorted in each image pyramid levels. By using the proposed search strategy, the results show that our method is efficient for multi-object detection.

After the analysis of various experiments, the proposed method is verified for use in the real word AOI applications because of the convincing results.

7. Conclusion

This paper demonstrates the concept of an accelerating correlation-based image recognition and pose estimations on parallel computing CPUs. The image pyramid search scheme deploys a sufficient condition based on a predefined threshold to

rapidly skip unsatisfactory candidates. The SIMD is also demonstrated how to take advantage of the modern CPUs to calculate similarity measure. They can dramatically reduce the computation burden of the similarity measure in color and monochrome images. Experimental results that the rotation accuracy is smaller than 0.2 degree and the computation advantage is over 227 times show the proposed method is very accurate and efficient when dealing with the correlation-based template matching in real-time automatic optical inspection.

Acknowledgment

The authors would like to acknowledge the financial support of the Ministry of Science and Technology of Taiwan, ROC through its grant MOST 104-2221-E-027-029-MY2.

References

- [1] Ouyang W, Tombari F, Mattoccia S, Di Stefano L, Cham W-K. Performance evaluation of full search equivalent pattern matching algorithms. *IEEE Trans Pattern Anal Mach Intell* 2012;34:127–43.
- [2] Mattoccia S, Tombari F, Di Stefano L. Fast full-search equivalent template matching by enhanced bounded correlation. *IEEE Trans Image Process* 2008;17:528–38.
- [3] Mattoccia S, Tombari F, Di Stefano L. Efficient template matching for multi-channel images. *Pattern Recogn Lett* 2011;32:694–700.
- [4] Mahmood A, Khan S. Exploiting transitivity of correlation for fast template matching. *IEEE Trans Image Process* 2010;19:2190–200.
- [5] Mahmood A, Khan S. Correlation-coefficient-based fast template matching through partial elimination. *IEEE Trans Image Process* 2012;21:2099–108.
- [6] Shou-Der W, Shang-Hong L. Fast template matching based on normalized cross correlation with adaptive multilevel winner update. *IEEE Trans Image Process* 2008;17:2227–35.
- [7] Lampert CH, Blaschko MB, Hofmann T. Beyond sliding windows: object localization by efficient subwindow search. In: 2008 IEEE conference on computer vision and pattern recognition, 2008 CVPR; 2008. p. 1–8.
- [8] Wu T, Toet A. Speed-up template matching through integral image based weak classifiers. *J Pattern Recogn Res* 2014;1:1–12.
- [9] Di Stefano L, Mattoccia S, Tombari F. Speeding-up NCC-based template matching using parallel multimedia instructions. In: 2005 Proceedings seventh international workshop on computer architecture for machine perception, 2005 CAMP; 2005. p. 193–7.
- [10] Kadidlo J, Strey A. Exploiting data- and thread-level parallelism for image correlation. In: 2008 16th Euromicro conference on parallel, distributed and network-based processing, 2008 PDP; 2008. p. 407–13.
- [11] Sassanapitak S, Kaewtrakulpong P. An efficient translation-rotation template matching using pre-computed scores of rotated templates. In: 2009 6th international conference on electrical engineering/electronics, computer, telecommunications and information technology, 2009 ECTI-CON; 2009. p. 1040–3.
- [12] Cui X, Kim H, Park E, Choi H. Robust and accurate pattern matching in fuzzy space for fiducial mark alignment. *Mach Vis Appl* 2013;24:447–59.
- [13] Tsai D-M, Tsai Y-H. Rotation-invariant pattern matching with color ring-projection. *Pattern Recogn* 2002;35:131–41.
- [14] Hinterstoisser S, Lepetit V, Ilic S, Fua P, Navab N. Dominant orientation templates for real-time detection of texture-less objects. In: 2010 IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2010. p. 2257–64.
- [15] Hinterstoisser S, Holzer S, Cagniard C, Ilic S, Konolige K, Navab N, et al. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: 2011 IEEE international conference on computer vision (ICCV). IEEE; 2011. p. 858–65.
- [16] Hinterstoisser S, Cagniard C, Ilic S, Sturm P, Navab N, Fua P, et al. Gradient response maps for real-time detection of textureless objects. *IEEE Trans Pattern Anal Mach Intell* 2012;34:876–88.
- [17] Konishi Y, Kotake Y, Ijiri Y, Kawade M. Fast and precise template matching based on oriented gradients. In: *Computer Vision–ECCV 2012 Workshops and Demonstrations*. Springer; 2012. p. 607–10.
- [18] Tsai D-M, Lin C-T, Chen J-F. The evaluation of normalized cross correlations for defect detection. *Pattern Recogn Lett* 2003;24:2525–35.
- [19] Tanimoto SL. Template matching in pyramids. *Comput Graph Image Process* 1981;16:356–69.

Chin-Sheng Chen received the Ph.D. degree in Mechanical Engineering from National Chiao Tung University in 1999. Presently he is a professor of Graduate Institute of Automation Technology, Taipei TECH. His research interests include machine vision systems, motion control, and mechatronics. He received the Outstanding Research Award from College of Mechanical and Electrical Engineering, Taipei TECH in 2013 and 2014.

Chien-Liang Huang received his B.S. degree from the Department of Mechanical Engineering, National Pingtung University of Science and Technology, Taiwan, in 2007. He is now pursuing a Ph.D. degree in the Graduate Institute of Automation Technology of National Taipei University of Technology, Taiwan. His research interests mainly include machine vision, pattern recognition.

Chun-Wei Yeh received the Ph.D. degree in Electronic, Electrical and Computer Engineering, University of Birmingham, UK, in 2014. His research interests mainly include color morphology, machine vision and pattern recognition.

Wen-Chung Chang received his M.S. degree in Engineering from Stanford University, in 1992, and his M.S., M.Phil., and Ph.D. degrees in Electrical Engineering from Yale University, in 1993, 1996, and 1997, respectively. He is currently an Associate Professor of Electrical Engineering at National Taipei University of Technology. His research interests include robotics, visual servoing, industrial automation, and real-time control applications.