Pattern Analysis and Applications
https://doi.org/10.1007/s10044-021-00997-7

INDUSTRIAL AND COMMERCIAL APPLICATION

A real-time two-stage and dual-check template matching algorithm based on normalized cross-correlation for industrial vision positioning

Fengjun Chen[1,2] · Jinqi Liao[1,2] · Zejin Lu[2] · Jiyang Lv[2]

Received: 3 November 2020 / Accepted: 4 June 2021

Abstract
In this paper, a fast template matching algorithm of two-stage and dual-check bounded partial correlation (TDBPC) based on normalized cross-correlation (NCC) of single-check bounded partial correlation is proposed. According to the principle of continuous rows, the template and the sub-image under matching are divided into three subregions to obtain two upper boundary terms of NCC and get two checking conditions then. In this way, it is possible to quickly eliminate matching points that cannot provide a better cross-correlation score regarding the current best candidate. Generally, to get the highest cross-correlation score, the sub-image has to traverse through the whole image. In addition, the two-stage search strategy of coarse–fine proposed in this paper can further reduce the calculation and improve matching efficiency. The initialization parameters are selected experimentally or automatically. Experimental results show that the TDBPC algorithm proposed in this paper can solve high computational complexity and long matching time of NCC template matching and make it possible to achieve real-time template matching in industrial vision positioning fields. The feasibility of this algorithm in practical application is proved.

Keywords Vision positioning · Real-time template matching · Dual-check bounded partial correlation · Two-stage search strategy

1 Introduction

Template matching is a technique to find the location of a given reference template in a target scene or target image. It is often used in the field of image processing. It can also be applied to automatic visual positioning in industrial applications. For example, Chen et al. [1] proposed an automated vision positioning system using the template matching algorithm of image pyramid hierarchical search strategy, which realized high-efficiency, high-precision positioning and dicing of semiconductor chips in an automatic dicing saw. Zhong et al. [2] used a blob-based template matching algorithm to distinct polycrystalline chips, fragmentary chips, and standard chips. Besides, template matching algorithms can also be used in metal surface defect detection[3, 4]. Connell et al. [5] used a template matching algorithm to implement an online character recognition system, quickly identifying handwritten characters. Therefore, template matching has many applications in industrial vision positioning.

The core of this technique lies in the algorithm used by template matching. The core algorithm of template matching can be divided into feature-based and grey-based template matching algorithms. The feature-based matching algorithm based on scale-invariant feature transform (SIFT) is one of the most popular algorithms[6]. Wu et al. [7] proposed a non-contact palmprint verification method that consists of three steps: image preprocessing, SIFT feature extraction

✉ Fengjun Chen
abccfj@126.com

Jinqi Liao
1466128137@qq.com

Zejin Lu
1206963586@qq.com

Jiyang Lv
937252991@qq.com

[1] State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, HUN College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, Hunan, China

[2] National Engineering Research Center for High Efficiency Grinding, Hunan University, Changsha 410082, Hunan, China

Published online: 13 June 2021

and matching, and matching refinement. A speeded-up robust feature (SURF) is proposed based on SIFT, which is several times faster than SIFT descriptor and has better stability on multiple images[8]. Teh et al. [9] proposed a method of analyzing image moments, and various types of moments can be used in image recognition. Some fundamental image processing issues such as image representation ability, noise sensitivity, and redundancy of information are appropriately addressed. The Zernike moment has been proven to be a very powerful descriptor, but its computation is very complicated. Hwang et al. [10] proposed a method to reduce computation complexity by using the Zernike 2-D basis functions instead of Zernike 1-D radial polynomials to decrease the amount of calculation. Wee et al. [11] proposed a method that minimizing geometrical and numerical errors of Zernike moment by mapping all the pixels of the discrete image inside the unit disk to reduce the amount of calculation. Revaud et al. [12] used the phrase information and improved the Zernike moment to provide a more accurate similarity measure. The feature-based template matching algorithm is complicated and time-consuming. As a result, it is challenging to meet real-time requirements. Conversely, the gray-based template matching algorithm is more suitable for real-time needs in practical applications.

The grey-based template matching algorithm includes the difference and the similarity measure method. Barnel et al. [13] proposed the difference measure of the sum of absolute difference (SAD). Bei et al. [14] presented an accelerated method of partial distortion elimination (PDE). Another accelerated method of successive elimination algorithm (SEA) based on SAD is also proposed by Li et al. [15]. Wang et al. [16] used SEA to achieve fast estimation for motion vectors. And Atallah [17] presented a SAD acceleration algorithm with lower time complexity. In addition, Essannouni et al. [18] proposed an adjustable SAD matching algorithm using the frequency domain, which realizes a transformation from a spatial domain to a frequency domain. Although the SAD algorithm does not require complicated calculations, it is sensitive to noise and light intensity and not very adaptable. Therefore, the similarity measure of normalized cross-correlation (NCC) appeared in template matching [19]. Although NCC in raw form is not efficient, various techniques have been developed to accelerate it with the characteristics of versatility and adaptability, such as fast Fourier transform (FFT), the most common method to speed up this algorithm [20–24]. Stefano et al. [25] proposed a bounded partial correlation (BPC) algorithm of the upper boundary of NCC to accelerate the algorithm, and the acceleration effect of different upper boundaries is different [26, 27]. Wen-Chia et al. [28] combined the BPC algorithm with circular projection to achieve a more adaptable template matching algorithm. To further improve the efficiency of template matching, a top-down search strategy using image pyramids was adopted[29]. A two-stage search strategy was also adopted, which first matched the sub-template and then used the whole template to perform a complete matching in the candidate points[30, 31]. A heuristic iterative method was used to select a best set of templates in the template for matching rather than the whole template [32].

Due to the significant computation of the NCC algorithm, this paper proposes a tighter upper boundary dual-check bounded partial correlation (DBPC) based on single-check BPC. A subsampling technique that minimizes the amount of sampling calculation is used to realize a two-stage search strategy of coarse–fine matching. The principle of selecting three initialization parameters for the DBPC algorithm is carried out experimentally or automatically. Experimental results show that the accuracy and efficiency of the improved NCC algorithm (TDBPC) are high.

2 Related work

The similarity measure between the template and a subwindow of the target image is the core part of the template matching algorithm. A good review is given in [33]. The most common similarity measure methods include the sum of absolute differences (SAD), the sum of squared differences (SSD), and normalized cross-correlation (NCC). In recent years, algorithm research is to realize template matching in complex scenes or unconstrained environments. Itamar et al. [34] proposed a novel similarity measure for template matching named deformable diversity similarity, robust to complex deformation, significant background clutter, and occlusions. Simon et al. [35] presented a robust template matching algorithm: occlusion aware template matching (OATM), which is superior in its ability to match under significant deformation and occlusion. Oron et al. [36] proposed a template matching algorithm named Best-Buddies Similarity (BBS), and it is robust against complex geometric deformation and high levels of outliers. A template matching algorithm called A-MNS[37] is presented, and this algorithm can overcome challenges such as background clutter, occlusion, arbitrary rotation transformation and nonrigid deformation. Therefore, the research goal has been to put forward a more robust template matching algorithm in the complex scene or unconstrained environment in the past few years. Nowadays, the proposed algorithms have been more and more complex with stronger performance. However, the real-time performance of the algorithms in practical industrial vision positioning applications is ignored.

Therefore, research for template matching algorithms has also focused on efficiency. To achieve real-time template matching, many strategies of accelerating algorithms have been applied. Accelerated methods of partial distortion elimination (PDE) [14] and successive elimination algorithm

(SEA) [15] are used in template matching based on SAD. And the strategy of bounded partial correlation (BPC) [25] to speed up template matching based on NCC is also applied. The PDE, SEA, BPC accelerated strategies are essentially a "pruning strategy" used to speed up the original algorithms.

NCC is a versatile similarity measure, while it is not efficient in spatial domain form, especially for real-time template matching algorithms. In this paper, a pruning method using two (weaker and stronger) upper bounds of the normalized cross-correlation (NCC) named TDBPC algorithm is applied, which can resolve the compatibility between the theory of template matching algorithm and practical applications, and make it possible to achieve real-time template matching in industrial vision positioning fields.

3 Bounded partial correlation

3.1 Single-check bounded partial correlation based on NCC

NCC is a classical similarity measure for template matching. It is said that the most correlated matching point is the location of the template in the scene image for template matching with a single target by calculating a correlation between the template and the scene image. Let T be a template of size $M \times N$ and I the scene image of size $W \times H$. The NCC at (x, y) is defined as

$$\mathrm{NCC}(x,y) = \frac{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)\cdot T(i,j)}{\sqrt{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2}\cdot\sqrt{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}} \tag{1}$$

The numerator of Eq. (1) is denoted as $C(x, y)$, representing the correlation term between the template and the matching sub-image. The denominator represents a product of L_2 norms $|I(x, y)|_2$ of the matching sub-image and L_2 norms $|T|_2$ of the template. The former of the denominator in Eq. (1) can be computed very efficiently using a recursive technique, known as box-filtering [38], which makes the calculation independent of the template area and requires only four elementary operations per image position. The latter of the denominator in Eq. (1) can be precomputed at initialization time. The position with the largest NCC score is the best matching position.

Set an upper boundary of $C(x, y)$ to $\alpha(x, y)$, then

$$\alpha(x,y) \geq C(x,y) = \sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I(x+i.y+j)\cdot T(i,j) \tag{2}$$

By normalizing $\alpha(x, y)$, an upper boundary for $NCC(x, y)$ was obtained as follows:

$$\frac{\alpha(x,y)}{|I(x,y)|_2\cdot|T|_2} \geq \frac{C(x,y)}{|I(x,y)|_2\cdot|T|_2} = \mathrm{NCC}(x,y) \tag{3}$$

Given an initialized similarity threshold η, then a check condition at matching point (x, y) is obtained as follows:

$$\frac{\alpha(x,y)}{|I(x,y)|_2\cdot|T|_2} < \eta \tag{4}$$

If condition (4) is satisfied, the matching process can proceed with the next point without calculating $C(x, y)$. It is guaranteed that the current matching point does not correspond to the new maximum correlation. If not, it is necessary to compute $C(x, y)$, normalize it and recheck the condition (5).

$$\frac{C(x,y)}{|I(x,y)|_2\cdot|T|_2} < \eta \tag{5}$$

If condition (5) is false, it indicates that the similarity of a current matching point is close to maximum similarity. Assign a greater similarity to η, record the position of the current point, and then match the next point. If condition (5) is satisfied, it indicates that the current point cannot provide a better similarity than a candidate point, thereby directly dealing with the next point. As the matching process progresses, η will increase to further accelerate the matching speed. It is just like a "pruning" algorithm, a very well-known and used algorithm in the computer science community using an upper bound to reduce the complexity of computing of $C(x, y)$ so that the matching speed can be accelerated as the matching process progresses.

How to choose a suitable $\alpha(x, y)$ is the key to the whole algorithm. Stefano et al. [26] proposed an upper boundary algorithm that divides the template and sub-image under matching into two subregions according to the same principle, forming two sub-terms of $\alpha(x, y)$: partial correlation term $P(x, y)$ and partial term $\beta(x, y)$

$$\begin{aligned}\alpha(x,y) &= P(x,y) + \beta(x,y)\\ &= \sum_{j=0}^{n}\sum_{i=0}^{M-1} I(x+i,y+j)\cdot T(x,y)\\ &\quad + \sqrt{\sum_{j=n+1}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2}\cdot\sqrt{\sum_{j=n+1}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}\end{aligned} \tag{6}$$

Substituting (4) to form a single-check BPC algorithm, and the partition diagram is in Fig. 1a.

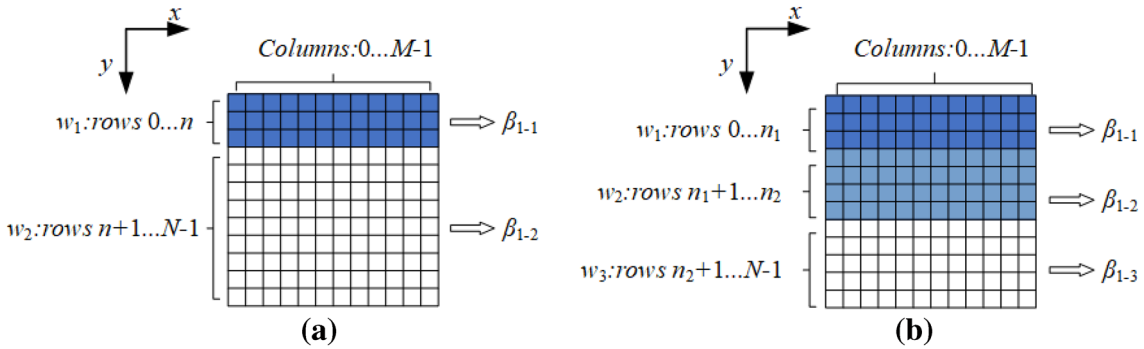

Fig. 1 Row-based partitioning rules of including **a** single-check partitioning; **b** dual-check partitioning

3.2 Dual-check bounded partial correlation (DBPC) algorithm

In this paper, a DBPC algorithm is proposed based on the BPC2 algorithm[25], which divides the template and matching sub-image into three subregions according to a similar principle, as shown in Fig. 1b. w_1 is taken as a subregion for calculating partial correlation term and w_2+w_3 as a subregion for calculating partial term. Then we can get the first upper boundary of Eq. (7)

$$\begin{aligned}\alpha_1(x,y) &= P_1(x,y)+\beta_1(x,y)\\ &= \sum_{j=0}^{n_1}\sum_{i=0}^{M-1} I(x+i,y+j)\cdot T(i,j) + \sqrt{\sum_{j=n_1+1}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2}\cdot\sqrt{\sum_{j=n_1+1}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}\end{aligned} \tag{7}$$

and obtain first check condition

$$\frac{\alpha_1(x,y)}{|I(x,y)|_2\cdot|T|_2} < \eta \tag{8}$$

Accordingly, w_1+w_2 are taken as a subregion for calculating the partial correlation term and w_3 as a subregion for calculating the partial term. Then we can get the second upper boundary Eq. (9) tighter than Eq. (7)

$$\begin{aligned}\alpha_2(x,y) &= P_2(x,y)+\beta_2(x,y)\\ &= \sum_{j=0}^{n_2}\sum_{i=0}^{M-1} I(x+i,y+j)\cdot T(i,j) + \sqrt{\sum_{j=n_2+1}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2}\cdot\sqrt{\sum_{j=n_2+1}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}\end{aligned} \tag{9}$$

and obtain a second check condition

$$\frac{\alpha_2(x,y)}{|I(x,y)|_2\cdot|T|_2} < \eta \tag{10}$$

The partial correlation term of Eqs. (7) and (9) can be directly computed, and the partial term of two equations can be subdivided into the partial term of the template and matching sub-image. And the former can be precomputed at initialization time, while the latter need to be efficiently computed using the box-filtering technique. Additionally, the square sum of the pixel of three subregions is efficiently calculated using three filters by box-filtering technique, and the corresponding formulas are defined as follows:

$$\beta_{1\text{-}1}(x,y,M-1,n_1) = \sum_{j=0}^{n_1}\sum_{i=0}^{M-1} I(x+i,y+j)^2 \tag{11}$$

$$\beta_{1\text{-}2}(x,y,M-1,n_2) = \sum_{j=n_1+1}^{n_2}\sum_{i=0}^{M-1} I(x+i,y+j)^2 \tag{12}$$

$$\beta_{1\text{-}3}(x,y,M-1,N-1) = \sum_{j=n_2+1}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2 \tag{13}$$

Thus the fast compute of two upper boundaries significantly reduced the computation of $C(x, y)$.

According to Fig. 1b, the selection of n_1 and n_2 directly determines the number of rows occupied by each subregion. Therefore, two correlation ratios are defined as follows:

$$C_{r1} = \frac{n_1 + 1}{N}; C_{r2} = \frac{n_2 + 1}{N} \tag{14}$$

The parameter value of C_{r1} and C_{r2} is critical to affecting matching efficiency.

3.3 Two-stage and dual-check bounded partial correlation (TDBPC) algorithm

The procedures of the DBPC algorithm are as follows and shown in Fig. 2a:

Step1: Initialize η and C_{r1}, C_{r2}, and compute $|T|_2$ and Eqs. (11), (12), (13).

Step2: Compute the first upper boundary $\alpha_1(x, y)$ of Eq. (7).

Step3: Judge the first check condition of Eq. (8). If satisfied, proceed with the next point by skipping the current point; if not, compute the second upper boundary $\alpha_2(x, y)$ of Eq. (9) and judge the second check condition of Eq. (10). If satisfied, proceed with the next point by skipping the current point; if not, NCC(x, y) needs to be computed.

Step4: Judge Eq. (5). If satisfied, it is not the best matching point; if not, it is close to the best matching point. Then the maximum similarity value is assigned to η, and the position (x, y) of the current point is recorded.

Step5: Traverse the whole scene image and repeat step2 ~ step4 until a most similar matching point is found.

To further improve the efficiency of template matching, a two-stage search strategy of coarse–fine is adopted in the DBPC algorithm. The original scene image and template are subsampled based on pixel area relationship to get sampling image and template used in coarse matching. The sampling factor k depends on the size of the image and is automatically selected to minimize the total number of operations required by both searches. Following a principle of minimizing computation, the optimal sampling factor f can be computed by Eq. (15). The first term of this function represents the number of operations executed by subsampling images. The second term represents the number of operations carried out in a 4 $k \times 4$ k neighborhood of the original scene image. This subsampling method is different from the "image pyramid" method [39, 40]. Equation (15) is used to dynamically select the size of the sampling factor for different images instead of the fixed sampling factor like "image pyramid" so that matching efficiency can be very high. Use

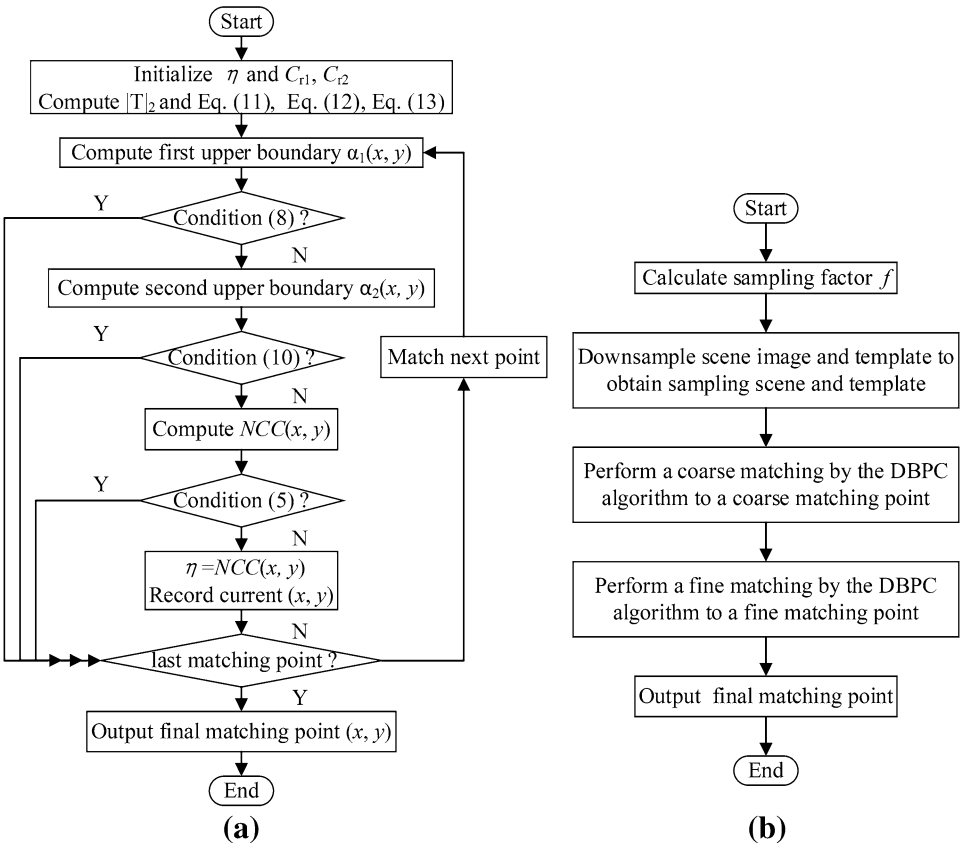

Fig. 2 Flow charts of **a** DBPC algorithm and TDBPC algorithm

the DBPC algorithm to perform coarse matching by matching the sampling scene image and template image to obtain the best coarse matching point (x_c, y_c). A best coarse matching point (x_c, y_c) is mapped to the original scene image to obtain a mapped point ($f\times x_c$, $f\times y_c$).

What's more, the effect of this mapping depends on the result of the coarse matching stage. If the coarse matching stage is correctly matched, then the mapping process is correct. Finally, fine matching of the full template is performed in a $4f\times 4f$ neighborhood centered on a mapped point ($f\times x_c$, $f\times y_c$) to obtain a final matching point.

$$f = \arg\min_k \{\frac{(W-M+1)(H-N+1)MN}{k^4} + (4k)^2 MN\} \tag{15}$$

So, the procedures of the TDBPC algorithm proposed in this paper are as follows and shown in Fig. 2b:

Step 1: Compute an optimal sampling factor f by Eq. (15).
Step 2: Downsample to obtain sampling scene image and the template according to f.
Step 3: Use the DBPC algorithm to perform a coarse matching by matching the sampling scene image and template image to obtain a best coarse matching point.
Step 4: Map the best coarse matching point to the original scene image, and perform a fine matching used by the DBPC algorithm in a $4f\times 4f$ neighborhood centered on a mapping point to obtain a final matching point.

4 Determination of initialization parameters of the DBPC algorithm

4.1 The metric of algorithm efficiency

A total number of matching points P was computed when searching for a scene image in the DBPC algorithm, as below:

$$P = (W-M+1)\cdot(H-N+1) \tag{16}$$

The number of points satisfying that first check condition of single-check BPC in Eq. (8) is S_1. The number of points satisfying that second check condition in Eq. (10) is S_2. Then we can define two elimination ratios, which represents a proportion of elimination points by checking conditions to the total points P, as follows:

$$E_{r1} = S_1/P,\ E_{r2} = S_2/P \tag{17}$$

A metric of efficiency R is introduced to measure the efficiency of this algorithm, which represents the ratio of a total number of operations required between the proposed algorithm and standard NCC(x, y) during calculating per point. The smaller the value of R is, the faster the computation of the proposed algorithm and the higher the efficiency. For the standard NCC(x, y) function, the total number of operations for calculating the similarity per point is ($M{\cdot}N+4$).

For single-check BPC, the total computation operations Q_1 and R_1 are

$$Q_1 = (P-S_1)\cdot M\cdot N + S_1\cdot M\cdot(n_1+1) + 8\cdot P \tag{18}$$

$$R_1 = \frac{Q_1}{P\cdot(M\cdot N+4)} = \frac{\left[(1-E_{r1}) + E_{r1}\cdot C_{r1}\right]\cdot M\cdot N + 8}{M\cdot N+4} \tag{19}$$

For the DBPC algorithm, conditions (8) and (10) need to be considered simultaneously. The total computation operations Q_2 and R_2 are

$$\begin{aligned} Q_2 =& (P-S_2)\cdot M\cdot N + (S_2-S_1)\cdot M\cdot(n_2+1) \\ &+ S_1\cdot M\cdot(n_1+1) + 12\cdot P \end{aligned} \tag{20}$$

$$\begin{aligned} R_2 &= \frac{Q_2}{P\cdot(M\cdot N+4)} \\ &= \frac{\left[(1-E_{r1}) + (E_{r2}-E_{r1})\cdot C_{r2} + E_{r1}\cdot C_{r1}\right]\cdot M\cdot N + 12}{M\cdot N+4} \end{aligned} \tag{21}$$

The test used four 8-bit grayscale images with different resolutions in the board detection field. The template for each image is obtained from another image collected in the same conditions. The image and corresponding template are shown in Fig. 3. The specific parameters of the image are shown in Table 1.

4.2 The effect of η on matching efficiency R_2

The check conditions (8) and (10) show that initialization threshold η affects the matching efficiency. If it is too low, the accelerated effect becomes not obvious. Inversely, it may incorrectly match because the maximum similarity in real matching may be less than the given η. In the test, we keep $C_{r1}=0.4$, $C_{r2}=0.2$, $\eta=0.94$, 0.95, 0.96, 0.97, respectively, and change R_2 to analyze the effect of η on matching efficiency.

As shown in Fig. 4, R_2 is less than 50% for all images, indicating that the DBPC algorithm can reduce computation by more than half to the standard NCC(x, y). And with the increase of η, R_2 gradually decreases, indicating that the matching efficiency increases. $R_2<22\%$ for image2 and $R_2<31\%$ for image4 shows that different images have different sensitivity at η. Therefore, with C_{r1} and C_{r2} unchanged, η only needs to prevent exceeding the maximum actual similarity. The closer the selected value is to the actual maximum similarity, the smaller the value of R_2 and the higher the matching efficiency of the algorithm.

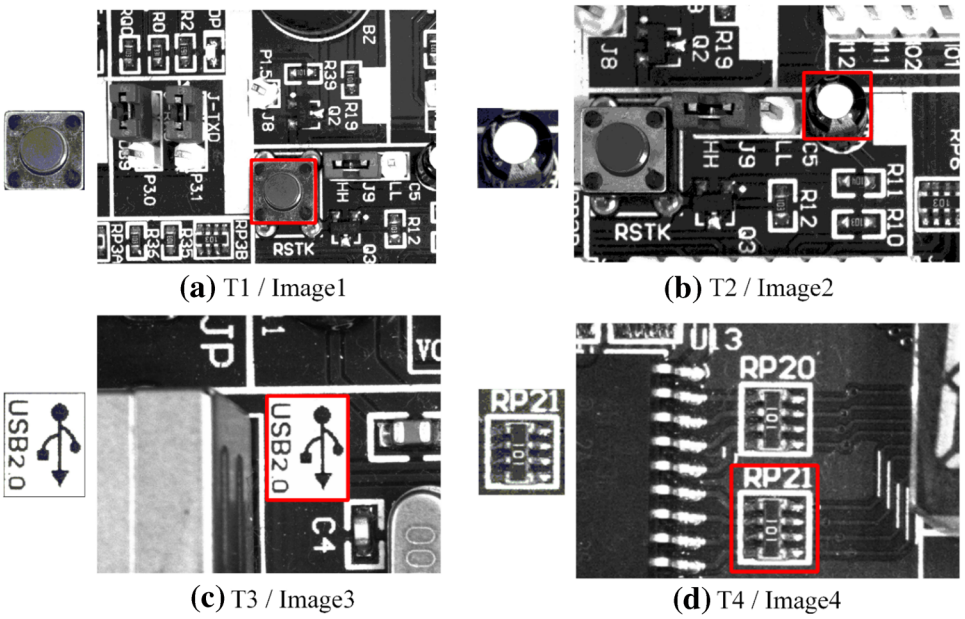

Fig. 3 Test image of different resolution for circuit board

Table 1 The parameters of images

Scene image	$W \times H$ (pixel)	Template	$M \times N$ (pixel)	Matching point;	Maximum similarity
Image1	1280×960	*T*1	242×238	[581,567]	0.978054
Image2	960×640	*T*2	167×165	[167,165]	0.993716
Image3	640×480	*T*3	151×194	[151,194]	0.996353
Image4	480×320	*T*4	114×145	[194,178]	0.99232

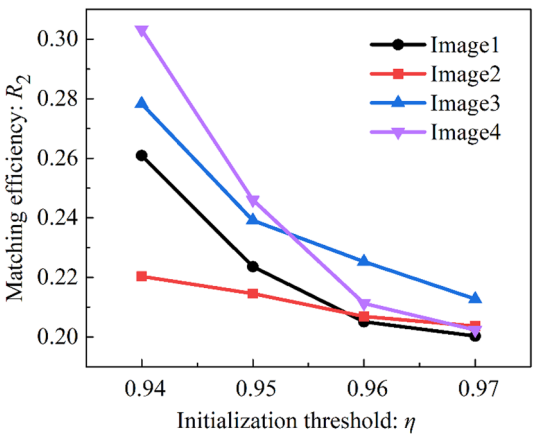

Fig. 4 Matching efficiency R_2 with initialization threshold η

Therefore, in the practical application of industrial visual positioning, generating a random point algorithm is proposed to automatically determine the initialization threshold η to avoid the uncertainty of manually specifying the initialization threshold η. The strategy to automatically determine the initialization threshold η is as follows: If there is no prior knowledge related to the matched target

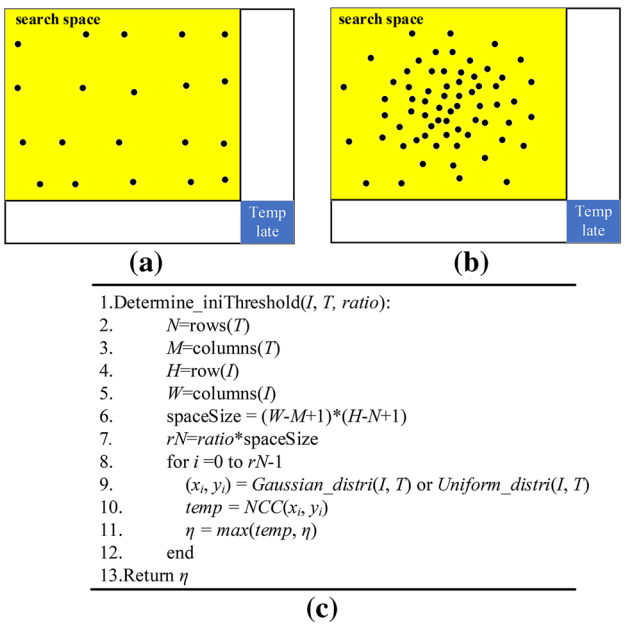

Fig. 5 Two methods for automatically determining initialization threshold η **a** schematic diagram of uniform distribution, **b** schematic diagram of Gaussian distribution, **c** The pseudocode of determining initialization threshold

position, the uniform distribution algorithm can be used to generate random points, as shown in Fig. 5a. Otherwise, the Gaussian distribution algorithm can generate random points, as shown in Fig. 5b. The pseudocode of determining the initialization threshold is shown in Fig. 5c, where *ratio* represents the ratio of the number of random points to the entire search space. The *Gaussian_distri* and *Uniform_distri* functions indicate that the coordinates of random points are obtained through Gaussian distribution or uniform distribution, respectively. The number of random points generated can be determined according to the size of the search space; if the search space is megapixels, the number of random points generated can be determined by taking up 0.001 to 0.005 of the entire search space. If the search space is less than megapixels, the number of random points generated can be determined by taking up 0.01 to 0.05 of the entire search space. Assuming that the number of random points is $\boldsymbol{rN}$, then the determination strategy of initialization threshold η is:

$$\eta = \max_{i=0}^{rN-1} \mathrm{NCC}(x_i, y_i) \tag{22}$$

4.3 The effect of C_{r1} on elimination ratio E_{r1} and matching efficiency R_1

From Eqs. (7) and (9), the selection of n_1 and n_2, namely C_{r1} and C_{r2}, directly affects the value of the two upper boundary $\alpha_1(x, y)$ and $\alpha_2(x, y)$ and, further, affects matching efficiency of condition (8) and (10). Firstly, C_{r2} and η are kept constant, and only the influence of C_{r1} on the elimination ratio E_{r1} and the R_1 is considered. The algorithm is tested using the above images. In the test, we keep $\eta = 0.95$, $C_{r2} = 0.9$ unchanged, and C_{r1} was sequentially taken from 0.1 to 0.8 by an increment of 0.1. By observing the E_{r1} and R_1, the influence of C_{r1} on the matching efficiency can be analyzed.

As is shown in Fig. 6a, with the increase of C_{r1}, E_{r1} increases overall and finally tends to 1. The varying degree of E_{r1} has a turning-point around $C_{r1} = 0.3$ in all images. When $C_{r1} < 0.3$, E_{r1} increases dramatically with the increase of C_{r1}. Once $C_{r1} > 0.3$, E_{r1} begins to rise to 1. It is indicated that there is such a turning point of C_{r1} (minimum C_{r1}) where E_{r1} can be maximized. However, Fig. 6b shows that the trend of R_1 decreases firstly and then increases with the increase of C_{r1}, and there is a minimum R_1 in theory. It demonstrates that there is a maximum efficiency theoretically. Additionally, it can be seen that R_1 is less than 50%, and its trend changes when C_{r1} is between 0.2 and 0.4, which verifies that there is an optimal R_1 that makes this algorithm's efficiency optimal. Considering both Fig. 6a, b, it is possible to have an optimal C_{r1}, making E_{r1} big enough and R_1 small enough. As the corresponding relationship between R_1 and C_{r1} is discrete, it is not easy to find a special C_{r1} for different images in practice. But it is easier to find an optimal range of C_{r1}. Consequently, when C_{r1} is in the range from 0.2 to 0.4, the computation can be reduced by more than 50% compared to the standard $NCC\ (x, y)$.

4.4 The effect of C_{r1} and C_{r2} on matching efficiency R_2

The relationship between R_2 and parameter pair (C_{r1}, C_{r2}) is discrete. Since $0 < C_{r1} < C_{r2} < 1$, R_2 can be tested using 36 parameter pairs shown in Table 2.

In the test, we keep $\eta = 0.95$ and the 36 parameter pairs are used to test the above four images, and then observe the change of R_2 to analyze the algorithm's efficiency. The tested results are shown in Fig. 7.

The horizontal coordinate represents the sequence number of the parameter pair. These 36 parameter pairs (C_{r1}, C_{r2}) have different effects on R_2 of different images. But tested results of four images fluctuate. All of them have seven troughs and R_2 is less than 30%, which shows that

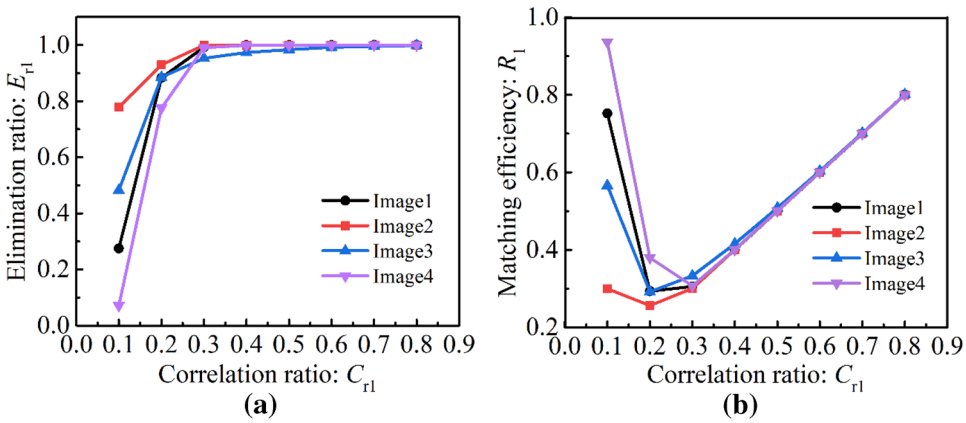

Fig. 6 **a** Elimination ratio E_{r1} and **b** matching efficiency R_1 with C_{r1}

Table 2 Tested parameter pairs of correlation ratio

Sequence	C_{r2}	C_{r1}	Sequence	C_{r2}	C_{r1}	Sequence	C_{r2}	C_{r1}
1	0.2	0.1	13	0.6	0.3	25	0.8	0.4
2	0.3	0.2	14	0.6	0.2	26	0.8	0.3
3	0.3	0.1	15	0.6	0.1	27	0.8	0.2
4	0.4	0.3	16	0.7	0.6	28	0.8	0.1
5	0.4	0.2	17	0.7	0.5	29	0.9	0.8
6	0.4	0.1	18	0.7	0.4	30	0.9	0.7
7	0.5	0.4	19	0.7	0.3	31	0.9	0.6
8	0.5	0.3	20	0.7	0.2	32	0.9	0.5
9	0.5	0.2	21	0.7	0.1	33	0.9	0.4
10	0.5	0.1	22	0.8	0.7	34	0.9	0.3
11	0.6	0.5	23	0.8	0.6	35	0.9	0.2
12	0.6	0.4	24	0.8	0.5	36	0.9	0.1

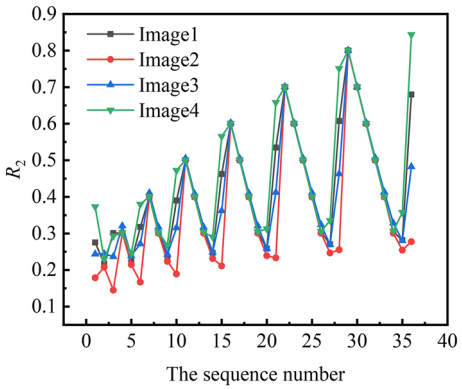

Fig. 7 Matching efficiency R_2 with (C_{r1}, C_{r2})

Table 3 Tested results of 60 images

Sequence	(C_{r1}, C_{r2})	Frequency
3	(0.1, 0.3)	0.114286
6	(0.1, 0.4)	0.097619
5	(0.2, 0.4)	0.092857
1	(0.1, 0.2)	0.083333
2	(0.2, 0.3)	0.080952
10	(0.1, 0.5)	0.07619
9	(0.2, 0.5)	0.071429

the computation can be reduced by more than 70% compared to the standard NCC(x, y) and the effect is remarkable. However, seven parameter pairs for each image troughs are not the same. Therefore, the above 36 parameter pairs are further used to test 60 images of different sizes, 36 R_2 data are obtained for each image, sorted in order from small to large. For each image, selecting the parameter pair corresponding to the first 7 data so that 420 parameter pairs can be obtained. Then seven parameter pairs are selected according to frequency, and they can be referenced in actual industrial template matching. Table 3 shows that the number of occurrences of the first seven parameter pairs in the 420 data exceeds 60% of the total data, indicating that these seven parameter pairs are reliable as references in actual industrial template matching. Further, if one does not want to determine which one of the seven parameter pairs is used as a parameter of algorithm performance through experiments. By default, selecting a parameter pair (0.2, 0.4) as the algorithm's default performance parameter can also achieve a good acceleration effect.

5 Test of matching efficiency

Sixty-six images on natural lighting are divided into six groups. In every group, one image was used for the source of the template and the other 10 for testing. The tested images are collected with the background of the industrial PCB board template matching positioning application. The template image per group is the same, but the position of scene images is different. In other words, scene images of different target positions are matched with the same template to facilitate the subsequent algorithm efficiency test. Images of each group are shown in Fig. 8. The information of each group is shown in Table 4. The TDBPC algorithm is formed by combining the DBPC algorithm and a two-stage search strategy, which is used to test match efficiency in this paper.

From Sect. 4, the selection of the algorithm performance parameters, (C_{r1}, C_{r2}) and η, is very important to the performance of the TDBPC algorithm. It would be preferable

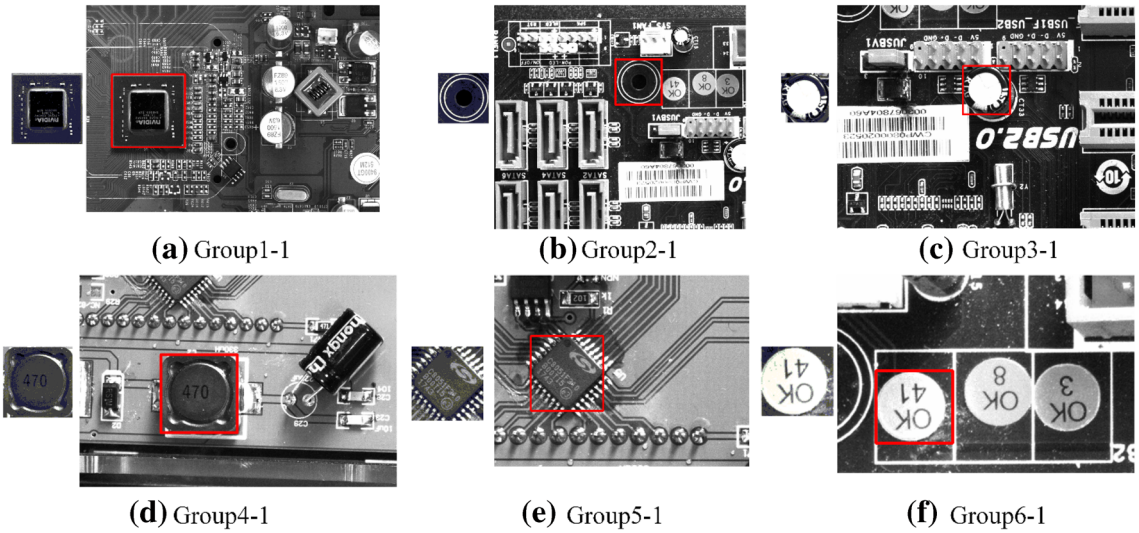

Fig. 8 Test image and corresponding template selected in each group

Table 4 The size of scene images and templates

Sequence	Scene image(pixel)	Template(pixel)
Group1	2560×1440	509×507
Group2	1440×1280	264×260
Group3	1280×960	202×208
Group4	960×640	231×236
Group5	640×480	182×188
Group6	480×320	124×117

for the selection of parameters to be automated rather than obtained through experiments. Therefore, there are two different ways to determine these parameters. Firstly, the parameters are automatically selected. Initialization threshold η is automatically determined by the random number method (see Sect. 4.2), and correlation ratio parameter pairs (C_{r1}, C_{r2}) are by default parameter pairs (see Sect. 4.4). Secondly, the parameters are obtained through experiments. The template matching algorithm depends on the image data itself. On the other hand, in practical applications with a fixed scene, the tested parameters obtained by the corresponding images are returned to the application itself. This approach is more in line with the needs of practical applications to get almost optimal parameters. Therefore, performance parameters obtained through experiments may be more efficient than automatically determined parameters. But relatively speaking, it takes a little more time, depending on the quality of the automatically selected parameters.

Firstly, an image and corresponding template of each group were chosen in Fig. 8. After initializing threshold η, the TDBPC algorithm is used to test 7 referenced parameter pairs. Because of a large amount of calculation, the matching efficiency R_2 in coarse matching is taken as a metric to select an optimal (C_{r1}, C_{r2}). The matching result of the six chosen images is shown in Table 5.

Based on the minimum R_2, the initialization parameter (C_{r1}, C_{r2}) used by six groups of test images was selected in Table 5. And the test platform is the Windows platform equipped with a 2.3 GHz processor, Intel(R) Core(TM) i5-4200U CPU and 4.00 GB of RAM. Use C++ to run a program of the algorithm in the integrated development

Table 5 Matching efficiency R_2 in coarse matching for 7 referenced (C_{r1}, C_{r2}) and parameter selection

Sequence	η	Referenced:(C_{r1}, C_{r2})							Selected: (C_{r1}, C_{r2})
		(0.1,0.2)	(0.1,0.3)	(0.2,0.3)	(0.1,0.4)	(0.2,0.4)	(0.1,0.5)	(0.2,0.5)	
Group1-1	0.86	0.457062	0.364058	0.299895	0.404163	0.273981	0.495183	0.298983	(0.2,0.4)
Group2-1	0.91	0.371212	0.352257	0.291426	0.386281	0.255896	0.472865	0.272925	(0.2,0.4)
Group3-1	0.96	0.312256	0.214651	0.233464	0.262388	0.250185	0.311941	0.268722	(0.1,0.3)
Group4-1	0.97	0.119945	0.131727	0.207211	0.143797	0.207189	0.155944	0.207244	(0.1,0.2)
Group5-1	0.9	0.913734	0.495006	0.483474	0.524662	0.501598	0.587613	0.553018	(0.2,0.3)
Group6-1	0.98	0.16935	0.202745	0.22493	0.24045	0.221122	0.27972	0.221318	(0.1,0.2)

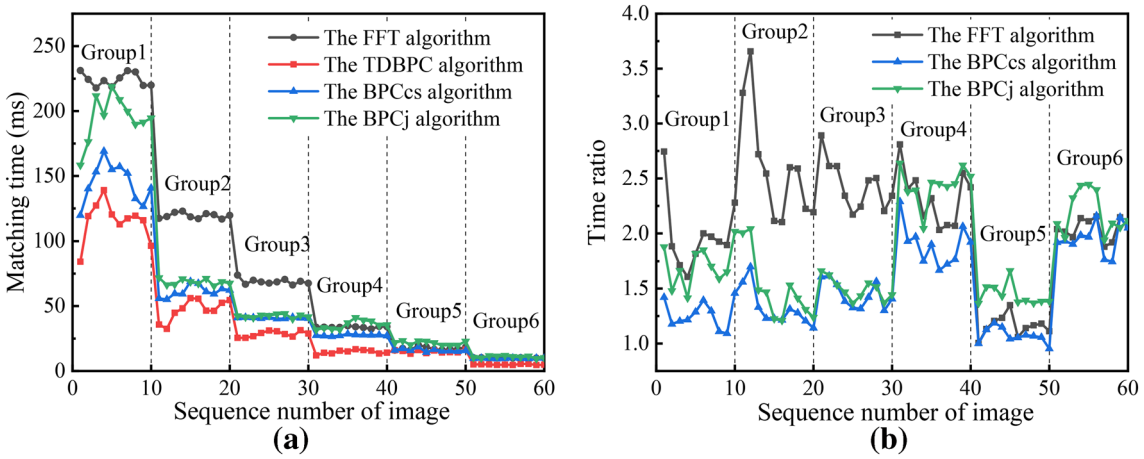

Fig. 9 **a** Matching time of four algorithms, **b** matching time ratio of three algorithms to TDBPC algorithm

Table 6 The average matching time (ms) of six groups of images for four algorithms

Algorithms	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
The TDBPC algorithm	115.162	47.302	28.407	14.630	14.996	4.975
The FFT algorithm	224.210	119.391	68.901	33.793	17.318	10.136
The BPCcs algorithm	144.619	61.244	40.884	27.541	16.044	9.703
The BPCj algorithm	194.622	68.479	42.349	35.581	21.541	10.844

environment of VS2017. The TDBPC algorithm proposed in this paper is compared with template matching based on FFT in Ref. [23], single-check BPCj in Ref. [25], and single-check BPCcs with another upper boundary in Ref. [26]. These algorithms are also strategies to accelerate NCC calculations comparable to the algorithms in this paper. All algorithms are developed based on Opencv, a third-party computer vision open source library. All tested under the same platform and the same data, and the test time is also averaged after 30 tests. The test results of the six group images are shown in Fig. 9 and Table 6.

The results of the four algorithms are similar, and the efficiency of the TDBPC algorithm is prioritized concerning the other three algorithms. However, the accelerated effect of this algorithm is different for different size scene images and templates. The matching time of the same group of images fluctuates due to the different illumination. The matching efficiency of the TDBPC algorithm is about 2 times that of the FFT-based NCC algorithm. Also, this efficiency is more evident in group1 ~ group4. The matching efficiency of the TDBPC algorithm is about 1.8 times that of the BPC_j algorithm, and it is more evident in group one, four and six. The matching efficiency of the TDBPC algorithm is about 1.5 times that of the two-stage BPCcs algorithm, and it is more apparent in group one, four and six. With the increase in image resolution, the matching efficiency of the TDBPC algorithm increases significantly compared with the other three algorithms. Besides, Fig. 9a shows that the average matching time used by the TDBPC algorithm to match an image with a resolution of 2560×1440 about 3.7 million pixels is about 115 ms, which means that it can process 522 images per minute in practical application. When matching an image with a resolution of 480×320 about 150,000 pixels, the average time is about 5 ms. Therefore, its rapidity meets practical requirements as well.

6 Conclusion

In this paper, by selecting reasonable η, C_{r1}, C_{r2} of the DBPC algorithm, combining with a two-stage search strategy, a TDBPC algorithm based on single-check BPC of NCC was proposed. This TDBPC algorithm can resolve the compatibility between the theory of template matching algorithm and practical applications, and make it possible to achieve real-time template matching in industrial vision positioning fields. It is indicated that the closer the initialization threshold η is to the maximum similarity of the actual image, the higher the matching efficiency. A relative optimal parameter pair (C_{r1}, C_{r2}) can be selected from 7 referenced parameter pairs given in this paper to make a template matching for different images. The experiments show that the matching efficiency of this algorithm is about 2 times, 1.8 times and 1.5 times of algorithm based on FFT (fast Fourier transform),

two-stage BPCj and two-stage BPCcs by selecting appropriate initialization parameters, respectively, and the feasibility of this algorithm in practical application is proved.

7 Future work

The proposed algorithm is still an accelerated algorithm of the NCC template matching. The matching speed and accuracy of the algorithm are quite high, but the adaptability of the algorithm is not very strong. Therefore, we will research the adaptability of the algorithm in future research work to correctly and quickly match the target in complex scenes or unconstrained environments such as image rotation, different scales, complex illumination, image deformation and multiple targets under meeting the requirement of industrial vision positioning.

Acknowledgements The funding was provided by the National Natural Science Foundation of China (Grant Nos. 51975204).

References

1. Chen F, Ye X, Yin S, Ye Q, Huang S, Tang Q (2018) Automated vision positioning system for dicing semiconductor chips using improved template matching method. Int J Adv Manuf Technol 100:2669–2678
2. Zhong F, He S, Li B (2017) Blob analyzation-based template matching algorithm for LED chip localization. Int J Adv Manuf Technol 93:55–63
3. An Z, Wang Y (2018) A metal surface defect detection method based on nonlinear diffusion and image difference. Surf Technol (in Chinese) 47:277–283. https://doi.org/10.16490/j.cnki.issn.1001-3660.2018.06.040
4. Wan Y, Cui P, Xu L, Yu H (2019) Anti-icing performance of micro-nano composite texture based on image processing technology. Surf Technol (in Chinese) 48:54–58. https://doi.org/10.16490/j.cnki.issn.1001-3660.2019.08.008
5. Connell SD, Jain AK (2001) Template-based online character recognition. Pattern Recognit 34:1–14
6. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
7. Wu X, Zhao Q, Bu W (2014) A SIFT-based contactless palmprint verification approach using iterative RANSAC and local palmprint descriptors. Pattern Recognit 47:3314–3326
8. Bay H, Ess A, Tuytelaars T (2008) Speeded-up robust features (SURF). Comput Vis Image Underst 110:346–359
9. Teh CH, Chin RT (1988) On image analysis by the methods of moments. IEEE Trans Pattern Anal 10:496–513
10. Hwang SK, Kim WY (2006) A novel approach to the fast computation of Zernike moments. Pattern Recognit 39:2065–2076
11. Wee CY, Paramesran R (2007) On the computational aspects of Zernike moments. Image Vis Comput 25:967–980
12. Revaud J, Lavoue G, Baskurt A (2009) Improving Zernike moments comparison for optimal similarity and rotation angle retrieval. IEEE Trans Pattern Anal Mach Intell 31:627–636
13. Barnea DI, Silverman HF (1972) A class of algorithms for fast digital image registration. IEEE Trans Comput C–21:179–186
14. Bei CD, Gray RM (1985) An improvement of the minimum distortion encoding algorithm for vector quantization. IEEE Trans Commun com-33:1132–1133
15. Li W, Salari E (1995) Successive elimination algorithm for motion estimation. IEEE T Image Process 4:105–107
16. Wang HS, Mersereau RM (1999) Fast algorithms for the estimation of motion vectors. IEEE T Image Process 8:435–439
17. Atallah MJ (2001) Faster image template matching in the sum of absolute value of differences measure. IEEE Trans Image Process 10:659–663
18. Essannouni F, Thami ROH, Aboutajdine D, Salam A (2007) Adjustable SAD matching algorithm using frequency domain. J Real Time Image Process 1:257–265
19. Briechle K, Hanebeck UD (2001) Template matching using fast normalized cross correlation. Opt Pattern Recogn XII 4387:95–102
20. Cooley J, Lewis P, Welch P (1967) Application of the fast fourier transform to computation of Fourier integrals, Fourier series, and convolution integrals. IEEE Trans Audio Electroacoust 15:79–84
21. Rao KR, Kim DN, Hwang JJ (2010) Fast Fourier transform algorithms and applications. In: Signals and communication technology
22. Park CS (2015) 2D discrete Fourier transform on sliding windows. IEEE Trans Image Process 24:901–907
23. Kaso A (2018) Computation of the normalized cross-correlation by fast Fourier transform. PLoS One 13:e0203434
24. Lewis JP (1995) Fast template matching. In: Vision interface 95, Canadian image processing and pattern recognition society Quebec City, Canada, May 15, vol 19, pp 120–123
25. Stefano LD, Mattoccia S (2003) Fast template matching using bounded partial correlation. Mach Vis Appl 13:213–221
26. Stefano LD, Mattoccia S (2003) A sufficient condition based on the Cauchy-Schwarz inequality for efficient template matching. In: International conference on image processing, pp 269–272
27. Mattoccia S, Tombari F, Di Stefano L (2008) Fast full-search equivalent template matching by enhanced bounded correlation. IEEE Trans Image Process 17:528–538
28. Wen-Chia L, Chin-Hsing C (2012) A fast template matching method with rotation invariance by combining the circular projection transform process and bounded partial correlation. IEEE Signal Proc Lett 19:737–740
29. Wong RY, Hall EL (1978) Sequential hierarchical scene matching. IEEE Trans Comput 27:359–366
30. Vanderburg G, Rosenfeld A (1977) Two-stage template matching. IEEE Trans Comput 1:104–107
31. Choi MS, Kim WY (2002) A novel two stage template matching method for rotation and illumination invariance. Pattern Recognit 35:119–129
32. Krattenthaler W, Mayer K, M.Zeiler (1994) point correlation: a reduced-cost template matching technique. In: International conference on image processing, pp 208–212
33. Ouyang W, Tombari F, Mattoccia S, Di Stefano L, Cham WK (2012) Performance evaluation of full search equivalent pattern matching algorithms. IEEE Trans Pattern Anal Mach Intell 34:127–143
34. Itamar I, Mechres R, Zelnik-Manor L (2017) Template matching with deformable diversity similarity. In: The IEEE conference on computer vision and pattern recognition, pp 175–183
35. Korman S, Milam M, Soatto S (2018) OATM: occlusion aware template matching by consensus set maximization. In: The

IEEE conference on computer vision and pattern recognition, pp 2675–2683
36. Oron S, Dekel T, Xue T, Freeman WT (2018) Best-buddies similarity—Robust template matching using mutual nearest neighbors. IEEE Trans Pattern Anal Mach Intell 40:1799–1813
37. Lai J, Lei L, Deng K, Yan R, Ruan Y, Jinyun Z (2020) Fast and robust template matching with majority neighbour similarity and annulus projection transformation. Pattern Recognit 98:107029
38. McDonnell MJ (1981) Box-filtering technique. Comput Graph Image Process 17:65–70
39. MacLean J, Tsotsos J (2000) Fast pattern recognition using gradient-descent search in an image pyramid. In: International conference on pattern recognition, pp 873–877
40. Takei R (2003) A new grey-scale template image matching algorithm using the cross-sectional histogram correlation method. Dynax Corporation Fuchu-City, Fuchu-cho