Novel template matching method with sub-pixel accuracy based on correlation and Fourier-Mellin transform

Guangjun Zhang
Beijing University of Aeronautics
and Astronautics
School of Instrumentation Science
and Opto-electronics Engineering
Ministry of Education
Key Laboratory of Precision
Opto-mechatronics Techonology
Beijing 100083
China

Ming Lei
Xulin Liu
Information Center of Beijing
Meteorological Bureau
Beijing 100089
China
E-mail: lmeagle01@163.com

Abstract. Template matching is the process of determining the presence and the location of a reference image or an object inside a scene image under analysis by a spatial cross-correlation. Conventional cross-correlation type algorithms are computationally expensive. Furthermore, when the object in the image is rotated, the conventional algorithms cannot be used for practical purposes. An algorithm for a rotation-invariant template matching with subpixel accuracy is proposed based on the combination of the correlation and Fourier-Mellin transformation when the fluctuating scope of the rotation angle is $[-20\ \text{deg}, 20\ \text{deg}]$. The algorithm consists of two stages. In the first stage, the matching candidates are selected using a computationally low-cost improved correlation algorithm. The operation of AND is adopted to reduce the computational cost for this stage. In the second stage, rotation invariant template matching is performed only on the matching candidates using the cross-correlation algorithm after adjusting image with a Fourier-Mellin invariant (FMI) descriptor, and the matching precision is subpixel by the novel method using the Fermat point. Experimental results show that the proposed method is very robust to Gaussian noise and rotation, and it also achieves high matching accuracy and matching precision. *© 2009 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.3125425]

Subject terms: template matching; rotation invariance; contour matching; 1-D phase correlation; Fourier-Mellin transform; log-polar transform.

Paper 080931RR received Dec. 2, 2008; revised manuscript received Mar. 5, 2009; accepted for publication Mar. 6, 2009; published online May 5, 2009.

1 Introduction

Template matching is a classical problem in scene analysis: given a reference image of an object, decide whether that object exists in a scene image under analysis, and find its location if it does. The template matching process involves cross-correlating the template with the scene image and computing a measure of similarity between them to determine the displacement.[1–3] Since the evaluation of the correlation is computationally expensive, there has been a need for low-cost correlation algorithms for real-time processing. A large number of correlation-type algorithms have been proposed.[4–6] One of the approaches is to use an image pyramid for both the template and the scene image and to perform the registration by a top-down search.[7] Other fast matching techniques use two-pass algorithms, using a subtemplate at a coarsely spaced grid in the first pass, and searching for a better match in the neighborhood of the previously found positions in the second pass.[8]

However, when objects in the image are rotated with respect to each other, the methods described earlier cannot be used, and a set of templates at different orientations is to be used. This procedure is hardly practical for real-time processing when the rotation angle is arbitrary or unconstrained. In such applications, rotation-invariant moment-based methods can be used.[9,10] The moment-based methods, however, require too much computation for practical purposes and are sensitive to noise.

In this paper, we propose a novel fast rotation-invariant template matching method when the fluctuating scope of the rotation angle is $[-20\ \text{deg}, 20\ \text{deg}]$. The method consists of three stages. In the first stage, the matching candidates are selected using an improved correlation algorithm with a computationally low cost. In the second stage, rotation-invariant template matching is performed only on the matching candidates using correlation algorithms after adjusting the image with a Fourier-Mellin invariant (FMI) descriptor.[11,12] In the third stage, the exact location of the template is obtained with subpixel interpolation. The methods in the first and third stages are novel, and an improved method in the second stage is used: The 2-D phase correlation is replaced by 1-D phase correlation to accelerate the registration parameter estimation process.

The remainder of the paper is organized as follows. Section 2 gives a brief overview of convolution and the matching method based on correlation. Section 3 presents our proposed algorithm for candidate selection, the improved FMI, and the subpixel interpolation method. In Sec. 4, various simulation results from the use of the proposed

0091-3286/2009/$25.00 © 2009 SPIE

schemes are reported and compared with results from the conventional methods in terms of the speed of the algorithm, the presence of noise, and the brightness and contrast variations. Last, we give concluding remarks in Sec. 5.

2 Convolution and Correlation Algorithms

The convolution algorithm is often used in digital image processing,[13,14] which is simple and efficient. The convolution algorithm and the correlation algorithm are all based on the template matching. Meanwhile, they bear partial similarities in terms of algorithm [see Eq. (2) and the numerator in Eq. (3)]. Inspired by this a novel and simplified algorithm similar to the convolution algorithm is proposed, which has not only reduced the complexity of the correlation algorithm, but also withstands rotation within certain angles. The theories of the convolution algorithm and the correlation algorithm are as follows.

2.1 *Convolution Algorithm*

For 2-D images, we will explain only the theory of 2-D convolution. The convolution h of the bivariate continuous functions f and g, given in Eq. (1), is similar to that of the 1-D convolution; the only difference is that the two variables x and y are independent. Their 2-D convolution expression is as follows:

$$h(x,y) = f * g = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)g(x-u,y-v)\mathrm{d}u\,\mathrm{d}v. \tag{1}$$

The convolution of the digital image is similar to the continuous function; the only difference is that the value of the independent variable is an integer. The double integration is transformed into a double sum. Therefore, we get Eq. (2) for a digital image:

$$H = F * G \quad H(i,j) = \sum_{m}\sum_{n} F(m,n)G(i-m,j-n), \tag{2}$$

where G is the convolution kernel, F denotes digital images, and H is the result of the convolution. Because the values of F and G are nonzero only within a limited range, we need to calculate only the sum of the field that is overlapped with the nonzero part.

2.2 *Cross-Correlation Algorithm*

The normalized cross-correlation theory is as follows. Let us define a template to be $\boldsymbol{G}_s$ with the size of $M_s \times N_s$ pixels and an image G_r with the size of $M_r < N_r$ pixels, respectively, and $M_s < M_r$, $N_s < N_r$. Define the top-left corner of the image as the origin and the subimage as $G_r(u,v)$ in the image with the size of $M_s \times N_s$ pixels. The normalized cross-correlation coefficient $\rho(u,v)$ between G_s and $G_r(u,v)$ is then defined as[15]

$$\rho(u,v) = \frac{\sum_{i=1}^{M_s}\sum_{j=1}^{N_s}[\boldsymbol{G}_r(i+u,j+v) - \overline{\boldsymbol{G}_r(u,v)}] \times [\boldsymbol{G}_s(i,j) - \overline{\boldsymbol{G}_s}]}{\{\sum_{i=1}^{M_s}\sum_{j=1}^{N_s}[\boldsymbol{G}_r(i+u,j+v) - \overline{\boldsymbol{G}_r(u,v)}]^2\}^{1/2}\{\sum_{i=1}^{M_s}\sum_{j=1}^{N_s}[\boldsymbol{G}_s(i,j) - \overline{\boldsymbol{G}_s}]^2\}^{1/2}}, \tag{3}$$

where the grayscale average of $G_r(u,v)$ and G_s is respectively represented with $\overline{G_r(u,v)}$ and $\overline{G_s}$. $\rho(u,v)$ in the scope of $[-1,1]$. The best matching location is determined by the maximum cross-correlation coefficient corresponding to the correlation peak, and the matching error is determined by the location of the correlation peak and the size and position of the template.

3 Template Matching Method Based on Correlation and Fourier-Mellin Transform

3.1 *Candidate Selection Process*

Enlightened by the convolution algorithm, we considered to simplify the crosscorrelation algorithm so that it will have the algorithmic mode of the convolution algorithm; in this way, the computation will be greatly reduced. However, we must point out that this change just takes the convolution algorithm as reference. Although they seem similar to convolution in appearance, they are different in meaning—the novel algorithm does not have the operational properties of the convolution. The realization of the simplification is as follows.

In Eq. (3), we presumed that

$$\begin{cases} \boldsymbol{G}_{S_{i,j}} = \left\{ \sum_{i=1}^{M_s}\sum_{j=1}^{N_s}[\boldsymbol{G}_s(i,j) - \overline{\boldsymbol{G}_s}]^2 \right\}^{1/2} \\ \boldsymbol{G}_{R_{i,j}} = \left\{ \sum_{i=1}^{M_s}\sum_{j=1}^{N_s}[\boldsymbol{G}_r(i+u,j+v) - \overline{\boldsymbol{G}_r(u,v)}]^2 \right\}^{1/2} \end{cases}.$$

Then, Eq. (3) can be transformed into Eq. (4) as follows:

$$\rho(u,v) = \frac{\Sigma_{i=1}^{M_s}\Sigma_{j=1}^{N_s}[\boldsymbol{G}_r(i+u,j+v) - \overline{\boldsymbol{G}_r(u,v)}] \times [\boldsymbol{G}_s(i,j) - \overline{\boldsymbol{G}_s}]}{\boldsymbol{G}_{R_{i,j}}\boldsymbol{G}_{S_{i,j}}}. \tag{4}$$

Suppose that we study only the binary image. Obviously, to remove the denominator in Eq. (4) and make it similar to the mode of the convolution, we must focus on the denominator in Eq. (4). After studying Eq. (4), we find that $\boldsymbol{G}_{S_{i,j}}$ is a fixed value, while others are all variable, which increases the complexity of the computation. This is because $\rho(u,v)$ is mainly about the degree of the cross-correlation. So we consider just the correlation coefficient based on the template and abandon the computation of $\boldsymbol{G}_{R_{i,j}}$ in Eq. (4), reducing the computational cost. The method is as follows.

The pixel values of the binary image are usually 0 and 255 or 0 and 1. Suppose that the pixel values of the binary image are only 0 and r. Since it is based on the template, the sum of square of each pixel in the template can be used as the denominator, instead of the denominator in Eq. (4). So the correlation coefficient $\rho(u,v)$ can be simplified as

$$\rho(u,v) = \frac{\Sigma_{i=1}^{M_s}\Sigma_{j=1}^{N_s}\boldsymbol{G}_r(i+u,j+v) \times \boldsymbol{G}_s(i,j)}{\Sigma_{i=1}^{M_s}\Sigma_{j=1}^{N_s}\boldsymbol{G}_s(i,j)^2}. \tag{5}$$

Obviously, the denominator of Eq. (5) is a constant. Suppose that it is N. Equation (5) can be further simplified into Eq. (6):

$$\rho(u,v) = \frac{1}{N}\left(\sum_{i=1}^{M_s}\sum_{j=1}^{N_s}\boldsymbol{G}_r(i+u,j+v) \times \boldsymbol{G}_s(i,j)\right). \tag{6}$$

Apparently, Eq. (6) is like the mode of the convolution algorithm, and $0 \leqslant \rho \leqslant 1$. In addition, there are only two pixel values (0 and r). If $r=1$, then the operation in brackets in Eq. (6) is actually equal to the sum of the AND operation of the correspondent pixels in the two images. Otherwise, the operation will be equal to the sum of squares of the AND operation. Because it is a binary image, no matter the former or the latter, their computational performances are the same as that of the AND operation. But the latter offers r^2, which is also a fixed constant. Therefore, Eq. (6) can be changed into Eq. (7):

$$\begin{cases} \rho(u,v) = \dfrac{1}{N}\left[\displaystyle\sum_{i=1}^{M_s}\sum_{j=1}^{N_s}\boldsymbol{G}_r(i+u,j+v)\cdot\boldsymbol{G}_s(i,j)\right] \\ \boldsymbol{G}_r(i+u,j+v)\cdot\boldsymbol{G}_s(i,j) = \begin{cases} 0 \\ 1 \end{cases} \quad \text{if } r=1 \\ \boldsymbol{G}_r(i+u,j+v)\cdot\boldsymbol{G}_s(i,j) = \begin{cases} 0 \\ r^2 \end{cases} \end{cases} . \tag{7}$$

In Eq. (7), ∅ denotes the AND operation. Undoubtedly, it is of great value to make the computation more efficient and the hardware implementation of the algorithm more realizable. On the other hand, there might be mismatching that makes $\rho=1$ where all the pixel values are r in some region of the reference image. For example, when there are only 0 and 1 for pixel values in an image, if the subimage covered by the template is 1, it would cause an error of final correlation degree $\rho=1$ by using Eq. (7) through the AND operation (similar to the AND operation of two images). Therefore, these regions must be removed by forcedly making $\rho=0$ there. Here, the subimage of universal 1 is eliminated through the comparison of the template image pixel sum and that of the subimage, covered by the template. If a subimage has universal 1, the sum of pixel values will be greater than that of the template because there is a region of 0 within the template. The universal 1 or 0 would make a template meaningless. Therefore, we would consider only the subimages in which the pixel value sum is less than that of the template. This would avoid error matching. Meanwhile, for a region of universal 0, it is not necessary to have a matching calculation. For a real matching region, the sum of pixel values should be approximate to that of the template. Thus, we consider only the regions in which the sum of pixel values is greater than half that of the template and less than that of the entire template. Suppose that the sum of the pixel values in the subimage covered by the template is Ψ, making $M < \Psi \leqslant \Phi$, where Φ is the sum of the pixel values in the template. M can be adapted according to the need of precision. Here, $M = \Phi/2$. When the template is being moved, it needs to compute just the sum of the pixel values in the subimage covered by the template within this range. As a result, not only can it accelerate the operational speed, but it can also remove a large amount of false matching positions.

Equations (3) and (7) are used for the same image and template, respectively. Figure 1 shows the distributions of correlation peaks. Figure 1(a) shows the plan view of the correlation peak generated from Eq. (3), and Fig. 1(b) shows the correlation peak generated from Eq. (3). Figures 1(c) and 1(f) show the correlation peak with image rotating by 30 deg. In particular, Figs. 1(c) and 1(d) show the plan view of the correlation peak and the correlation peak, respectively, both generated from Eq. (3). The same in Fig. 1(e) and 1(f), but generated from Eq. (7). In the illustration, the pane stands for the place where the value of ρ is highest, and the location of the pane in Fig. 1(a) is the factual location of the template in the reference image. In Fig. 1, in order to show better comparison of the two algorithms, the denoted matching positions are marked according to the highest correlation peak. But in the following experiments, the candidate matching positions are selected through a definite number of candidates.

Obviously, after the rotation, the classic algorithms have lost the correct matching position [as shown in Fig. 1(c)]. With the refined algorithm, the effective matching positions will still stay near the correct positions [as shown in Fig. 1(e)]. The refined algorithm thus has better flexibility to rotations. Meanwhile, it can isolate the possible matching

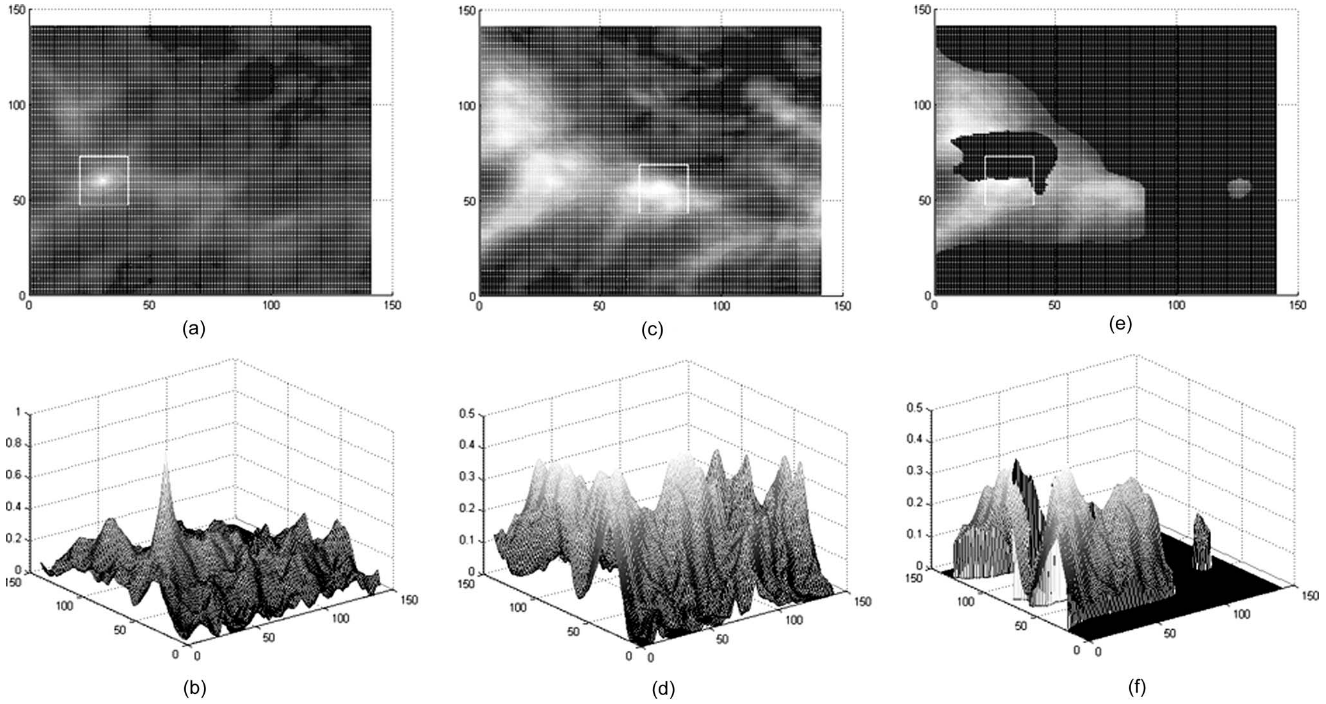


Fig. 1 Correlation peaks. (a) Plan view of the correlation peak generated from Eq. (3); (b) correlation peak generated from Eq. (3); (c) plan view of the correlation peak generated from Eq. (3) (image rotates by 30 deg); (d) correlation peak generated from Eq. (3) (image rotates by 30 deg); (e) plan view of the correlation peak generated from Eq. (7) (image rotates by 30 deg); and (f) correlation peak generated from Eq. (7) (image rotates by 30 deg).

regions within a smaller region [as shown in the light region of Fig 1(e)], which benefits the following fine matching.

Further, it has to be pointed out that matching candidates are not derived through Eq. (7) after binary images are set through binarization. They are derived, based on the adaptive threshold, through the pixel values, either 0 or 1. This has speeded up the computation. Here, the adaptive thresholds are calculated by the grayscale mean values of the subimages. The subimages are divided equally from reference image into n copies according to the template. If the reference image cannot be divided equally into subimages according to the template, the rest of the image that cannot be put in the subimage will calculate its threshold with its neighboring subimage. This binarization is simple and fast. But just like many other binarizations, it cannot keep consistency of binarization for complex images. Therefore, after the coarse matching (CMA) stage, Fourier-Mellin transformation is introduced. Thus, the performance is guaranteed as long as the matching candidates are near the good matching points. Further, with the cost of additional computation, more candidates can be selected in order that some candidates can fall around the fine matching points.

The Fourier-Mellin transformation has the advantage of low cost of matching duration and overlapping range. Even though the matching candidates are selected at locations far from the fine matching points, a correct matching still can be guaranteed. The performance of these two indices has the best effect compared to other algorithms.

3.2 *Image Registration Based on Fourier-Mellin Transform*

The image registration algorithm based on Fourier-Mellin transform is an effective matching method insensitive to noise and with high veracity.[7] It has been applied to image registration, curve matching, and other fields,[11,16,17] where it is used only to estimate registration parameters. To accelerate the registration parameter estimation process, the 2-D phase correlation in Chen et al.[11] is replaced by 1-D phase correlation. The improved image registration theory is as follows.

3.2.1 *Principle of 1-D phase correlation*

Phase correlation is one of the most widely used techniques for image registration in the frequency domain to determine the relative translation between two images.[18,19] This method relies on the translation property of the Fourier transform, referred to as the Fourier translation theorem, which is easy to implement, has less computation complexity, and thus requires less process time due to the high efficiency of fast Fourier transform (FFT). Furthermore, it is robust to interference. However, phase correlation can deal with translations only between two images in the x axis and y axis.[20]

In this paper, with the 1-D signal $f(x)$ as an example, the Fourier translation theory is as follows:

$$F(\omega)\exp(-j2\pi x_0\omega) \leftrightarrow f(x-x_0). \tag{8}$$

Let $f_1(x)$ and $f_2(x)$ be two input signals that differ only by a displacement x_0, i.e.,

$$f_2(x) = f_1(x - x_0). \tag{9}$$

From Eq. (8), the corresponding Fourier transforms $F_1(\omega)$ and $F_2(\omega)$ will be related by

$$F_2(\omega) = F_1(\omega)\exp(-j2\pi x_0\omega). \tag{10}$$

The cross-power spectrum of two signals f_1 and f_2 with $F_1(\omega)$ and $F_2(\omega)$ is defined as

$$C_{12}(\omega) = \frac{F_1(\omega)F_2^*(\omega)}{|F_1(\omega)F_2^*(\omega)|} = \exp\{j[\Phi_1(\omega) - \Phi_2(\omega)]\} = \exp(j2\pi x_0\omega), \tag{11}$$

where $\Phi_1(\omega)$ and $\Phi_2(\omega)$ are the spectral phases of $f_1(x)$ and $f_2(x)$, $F_2^*(\omega)$ is the complex conjugate of $F_2(\omega)$, and the translation theorem guarantees that the phase of the cross-power spectrum is equivalent to the phase difference between the two signals. By taking the inverse Fourier transform of the representation in the frequency domain, we will have a function that is an impulse[19]—that is, it is approximately zero everywhere except at the displacement x_0 that is needed to optimally register the two signals:

$$D_{12}(\omega) = F^{-1}\left(\frac{F_1(\omega)F_2^*(\omega)}{|F_1(\omega)F_2^*(\omega)|}\right) = F^{-1}[\exp(j2\pi x_0\omega)]. \tag{12}$$

Based on the property of the Fourier transform, the inverse Fourier transform of $\exp(j2\pi x_0\omega)$ is $\delta(x-x_0)$. Equation (12) gives the 1-D δ function. So $D_{12}(\omega)$ is a function that is also an impulse.

Since the phase difference for every frequency component contributes equally, the location of the peak will not change if there are noises or part distortion between the two signals.

3.2.2 *FMI descriptor*

The FMI descriptor is translation invariant and represents rotation and scaling as translations in parameter space. Its theory is as follows. Consider image $s(x,y)$ and image $r(x,y)$, among which, $s(x,y)$ is obtained from $r(x,y)$ by the translation, rotation, and scaling transform, i.e.,

$$s(x,y) = r[\sigma(x\cos\alpha + y\sin\alpha) - x_0, \sigma(-x\sin\alpha + y\cos\alpha) - y_0]. \tag{13}$$

Then the corresponding Fourier transforms $S(u,v)$ and $R(x,y)$ will be related by

$$|S(u,v)| = \sigma^{-2}|R[\sigma^{-1}(u\cos\alpha + v\sin\alpha), \sigma^{-1}(-u\sin\alpha + v\cos\alpha)]|, \tag{14}$$

where, we use $|.|$ to denote spectrum amplitude. From Eq. (14), spectrum amplitude is related only to rotation angle α and scaling factor σ, not to displacement x_0 and y_0. Hence, the transformation parameters can be computed by two steps: In step 1, α and σ can be obtained by the spectrum amplitude; in step 2, x_0 and y_0 can be achieved.

Step 1. Achieving the rotation angle α and the scaling factor σ using the phase correlation algorithm. The theory is as follows

$$r_{p1}(\theta, \log\rho) = r_p(\theta,\rho), \tag{15}$$

$$s_{p1}(\theta, \log\rho) = s_p(\theta,\rho), \tag{16}$$

where the spectrum amplitudes of the image r and the image s in polar coordinates are respectively represented with r_p and s_p. Then we can get the following equation:

$$s_{p1}(\theta, \log\rho) = r_{p1}(\theta - \alpha, \log\rho - \log\sigma), \tag{17}$$

or

$$s_{p1}(\theta, \lambda) = r_{p1}(\theta - \alpha, \lambda - \kappa), \tag{18}$$

where $\lambda = \log\rho$, $\kappa = \log\sigma$.

Here, we use the 1-D projection, so the image in polar coordinates can be devided into two 1-D datasets in the x axis and y axis, respectively. Then, Eq. (18) can be divided into the following two equations:

$$s'_{xp1}(\theta) = r'_{xp1}(\theta - \alpha), \tag{19}$$

and

$$s'_{yp1}(\lambda) = r'_{yp1}(\lambda - \kappa). \tag{20}$$

We can see that Eqs. (19) and (20) have the form of Eq. (9) in nature. So s'_{yp1} and r'_{yp1} can be calculated using Eqs. (10) and (11) ordered in log-polar coordinate space, and the same for s'_{xp1} and r'_{xp1}. Then by taking the inverse Fourier transform, the distribution of the correlation coefficient can be achieved. α and κ can be obtained, according to the location of the corresponding peak.

If the logarithm's fundus is e, then

$$\sigma = e^{\kappa}. \tag{21}$$

Thus, α and σ can all be obtained.

Step 2. Achieving the displacement x_0 and y_0 using the phase correlation algorithm. By taking the inverse transform, image $s_1(x,y)$ can be obtained from the image $s(x,y)$ using α and σ from step 1. Equations (10) and (11) are used to image $s_1(x,y)$ and image $r(x,y)$, respectively. Then, by taking the inverse Fourier transform, the distribution of the correlation coefficient can be achieved. The displacement x_0 and y_0 can be obtained according to the location of the corresponding peak.

3.3 *Subpixel Interpolation Method*

Rectify the image by the method mentioned in Sec. 3.2, and then take the right matching point as the center and search by expanding several pixels. (The range of the searching can be expanded according to different demands—obviously, the bigger the range is, the more reliable the result will be, but the more time it will consume.) Then,

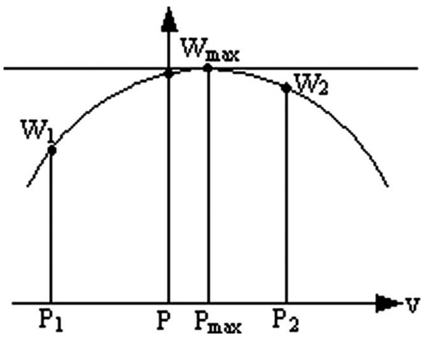


Fig. 2 Parabola interpolation method.

according to the correlation peak, the novel subpixel interpolation method is used to gain the final matching position by the principle of parabola interpolation method (Fig. 2) The concrete steps are as follows.

Making eight longitudinal cuts from the summit of the correlation peak (vertically, horizontally, and diagonally), we will get four sections that can be precisely described by the fitting parabola. In this way, 2-D information can be applied to describe 3-D information. Therefore, four pairs of points at the 0.1-pixel level are obtained: (x_1,y_1), (x_2,y_2), (x_3,y_3), (x_4,y_4). Suppose that the point (x,y) is the minimum value point—that is, the sum of all distances between this point and other points is the smallest—then this point is the Fermat point. Forming function $f(x,y)$ as follows:

$$f(x,y)=[(x-x_1)^2(y-y_1)^2]^{1/2}+[(x-x_2)^2(y-y_2)^2]^{1/2}+[(x-x_3)^2(y-y_3)^2]^{1/2}+[(x-x_4)^2(y-y_4)^2]^{1/2},$$

then the (x,y) with $f_{\min}$ in the preceding expression is the wanted subpixel position. Therefore, we can heighten the interpolation accuracy, making it as accurate as 0.01 subpixel.

Next, we will further explain this method. Vertically, the biggest correlation coefficient can be obtained by using the parabola interpolation method and then putting the result into the equation $y=ax^2+bx+c$ so that we can get the correspondent values in the x axis and y axis, respectively. The coefficient of the equation a, b, and c can be obtained by the three nearby coordinate values of the correlation peaks. Suppose that the correlation coefficient of the matching position is ρ_0 and that its horizontal coordinate value is x_0. We will then get the equation $\rho_0=ax_0^2+bx_0+c$, taking two points near (x_0,ρ_0)—namely, (x_1,ρ_1) and (x_2,ρ_2), respectively—and then solving the following equations:

$$\begin{cases}\rho_0=\alpha x_0^2+bx_0+c\\ \rho_1=\alpha x_1^2+bx_1+c\\ \rho_2=\alpha x_2^2+bx_2+c\end{cases}.$$

Then, the coefficients a, b, and c can be obtained. The real matching position should be the place where the correlation coefficient is the biggest—that is, the highest point $w_{\max}$ in the fitting parabola (Fig. 2). In this condition, the correspondent horizontal coordinate can be obtained along the fitting parabola.

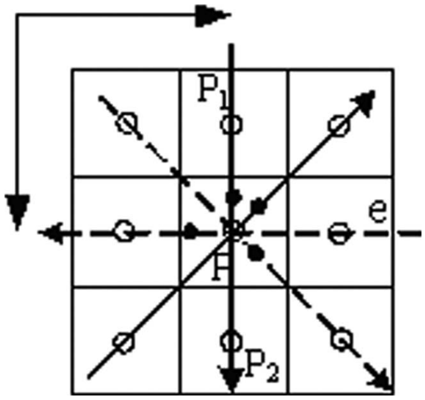


Fig. 3 Eight directions along fitting parabolas.

Similarly, the correspondent vertical coordinate can be obtained along the fitting parabola by using its vertical coordinate—that is, $\rho_0=dy_0^2+ey_0+f$. We will then get the coordinate value (x_1,y_1).

When fitting horizontal and diagonal sections, as mentioned earlier, we can get the correspondent coordinate values of the other three correspondent points—say, (x_2,y_2), (x_3,y_3), and (x_4,y_4). The eight directions can be illustrated by Fig. 3 as follows.

The four points can generally form three shapes—namely, convex quadrilateral, concave quadrilateral, and triangle (see Fig. 4). Making use of basic mathematical knowledge—that is, that the other two sides of a triangle were greater than the third—it is easy to prove that when the four points form a convex quadrilateral, the Fermat point (the point where the sum of the distance between it and other peaks is smallest) is the intersection of the diagonals; when they form a concave quadrilateral, the Fermat point is the point at the bottom of the concave part; and when they form a triangle, the Fermat point is the point in the middle of the line where the three points are in. Because the positions of the points are the same, the conditions when they form a convex quadrilateral or a triangle can be classified as one category. Then by the features of the Fermat point, the coordinates of the subpixel position can be easily obtained.

3.4 *The Matching Algorithm Flowchart*

In the preceding sections, we have carefully analyzed the steps of the proposed algorithm. The main flowchart is shown in Fig. 5, which consists of the following steps:

1. Image preprocessing.
2. Implementing coarse matching and finding matching candidates.

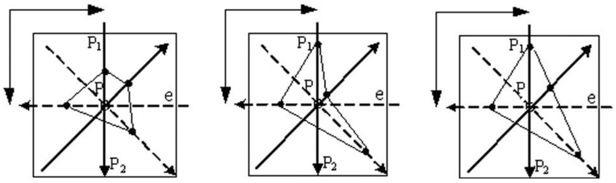


Fig. 4 Position about four points.

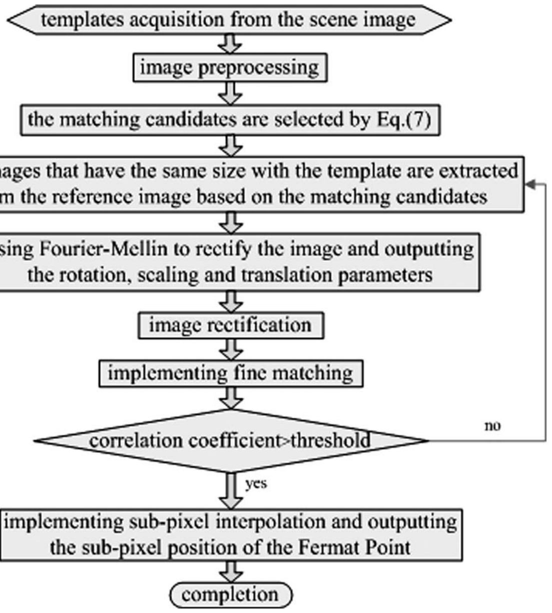

templates acquisition from the scene image
image preprocessing
the matching candidates are selected by Eq.(7)
sub-images that have the same size with the template are extracted from the reference image based on the matching candidates
using Fourier-Mellin to rectify the image and outputting the rotation, scaling and translation parameters
image rectification
implementing fine matching
correlation coefficient>threshold
no
yes
implementing sub-pixel interpolation and outputting the sub-pixel position of the Fermat Point
completion

Fig. 5 The matching algorithm flowchart.

3. Implementing image registration and outputting the translation, rotation, and scaling parameters.
4. Image rectification.
5. Implementing fine matching and outputting the position at pixel level.
6 Implementing subpixel interpolation and outputting the precious position of the subpixel

4 Experimental Results

Experiments were carried out to measure the performance of our algorithm in terms of speed, matching precision, and matching accuracy. As proposed in this paper, methods of computation for selecting candidates, the hierarchical Zernike method,[21] and the circular projection method of computation for selecting candidates in the frequency domain[22] are compared. The hierarchical Zernike method utilizes the pyramid structure of the image and yielded results comparable to ours in terms of accuracy.

4.1 *Test Images*

The proposed algorithm has been tested with four types of variations: rotation, brightness change, contrast change, and added Gaussian noise.

Figure 6 shows a set of templates masked off from the original image, and Figures 7(a)–7(d) show rotated images at arbitrary angles in the scope of [−20 deg, 20 deg]. In particular, Fig. 7(a) shows a set for rotation only and Fig. 7(b) shows a set with additive Gaussian noise in addition to the rotation. Figs. 7(c) and 7(d) have brightness and contrast changes, respectively, both with rotations at arbitrary angles in the scope of [−20 deg, 20 deg]. In all of the images with different settings, the proposed algorithm has located the position of each template precisely.

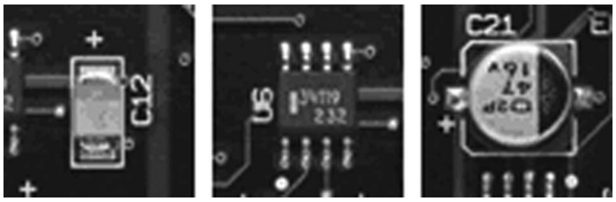


Fig. 6 A set of templates.

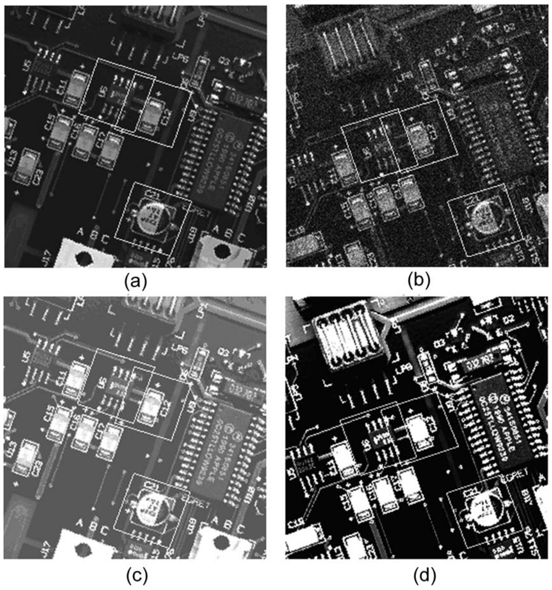


Fig. 7 Test images (a circuit board). (a) Rotated image; (b) rotated image with Gaussian noise; (c) rotated image with brightness variation; and (d) rotated image with contrast variation.

4.2 *Speed Test*

To verify the matching performance of our proposed algorithm in terms of speed, several experiments were carried out with various sizes of templates for a set of 256×256 images. Template sizes varied from 31×31 to 71×71. The proposed method was also compared in terms of the speed to the existing algorithm, the hierarchical Zernike method, and the circular projection method of computation for selecting candidates in the frequency domain. The speed test was performed on a 2.4-GHz Pentium 4 PC (Table 1).

Figure 8 shows the speed comparison of several algorithms. The proposed algorithm turned out to be the fastest.

Table 1 Speed of matching (in s).

Size of template	Proposed method	Circular projection	Hierarchical Zernike
31×31	0.0250	0.0625	0.3855
41×41	0.0420	0.1255	0.6612
51×51	0.0575	0.1811	1.0015
61×61	0.0607	0.2576	1.3232
71×71	0.0799	0.3242	1.6202

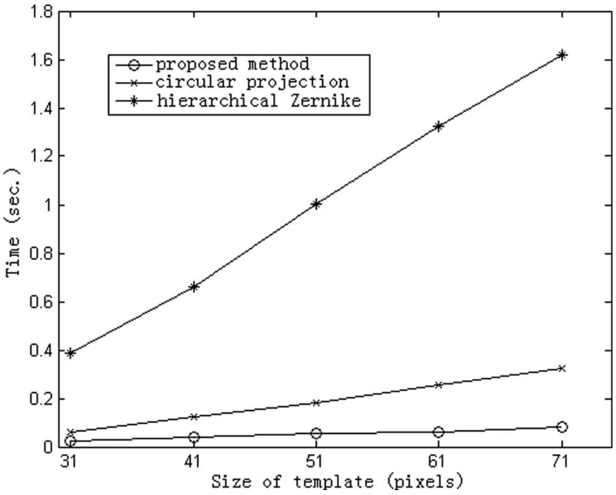


Fig. 8 Speed comparison.

For the time required to select candidates, Figure 9 shows that it increases very slowly in relation to the template size (Table 2).

4.3 *Matching Precision*

To verify the interpolation performance of our proposed algorithm in terms of precision, the matching error was compared with the paraboloid surface fitting method (Fig. 10). Figure 11 shows the relationship between sections and precision. In the experiment, the interpolation algorithm was performed when the matching location is precise. Table 3 shows the experiment results of the matching precision comparison between the paraboloid surface fitting method and the proposed algorithm. Table 4 shows the results of the proposed algorithm under different sections.

In Table 3, for the method of the paraboloid surface fitting, it is obvious that the minimum of the absolute error is 0.019 and 0.068 in the x axis and y axis, respectively, while the maximum of the absolute error is 0.00314 for x and 0.00168 for y, according to the proposed algorithm (four sections). Obviously, the proposed algorithm is more accurate than the method of the paraboloid surface fitting, which improves the matching accuracy by at least an order of magnitude. Figure 10 shows the error comparison for the two methods.

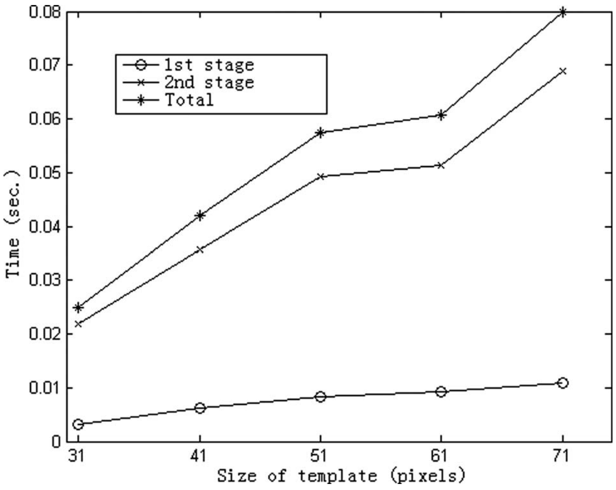


Fig. 9 Speed of each stage (first stage: candidate selection; second stage: fine matching).

Table 2 Speed of each stage (in s).

Size of template	31×31	41×41	51×51	61×61	71×71
First stage Candidate selection	0.0032	0.0063	0.0082	0.0093	0.0109
Second stage Fine matching	0.0218	0.0357	0.0493	0.0514	0.0690
Total	0.0250	0.0420	0.0575	0.0607	0.0799

In Table 4, when one point is selected, it is the point corresponding to the vertex of the fitting parabola along the direction of 135 deg. When two points are selected, they are the vertices of the horizontal and vertical fitting parabolas, respectively; when three points are selected, they are the vertices of the fitting parabolas corresponding to the horizontal, vertical, and 135 deg directions.

Obviously, the precision increases continuously with the increase of the sections (Fig. 11). However, the quantity and complexity of the computation also increase accordingly with more points selected: When only one point is selected, this point is the best matching location; when two points are selected, it is obvious that the midpoint is the best matching location; when three points are selected, the best matching location is the Fermat point. Fig. 4 shows the analytic results for four points, and the complexity of the computation increases tremendously for more than five points.

4.4 *Matching Accuracy*

For the accuracy of our proposed algorithm, a matching ratio was also compared. The matching ratio is defined as the number of correct matches to the number of trials. For this accuracy testing, 30 similar images of electronic circuit boards had been captured, and 15 templates of the area of interest were selected manually from each image, resulting in 450 templates in total. Various types of noise such as Gaussian noise and contrast/brightness changes were added to each image. Then each template was searched for from the corresponding image. The matching ratio indicates how well the queried template has been retrieved at the precise location from the corresponding image. Obviously, the matching ratio depends on the number of candidates for each trial. The ratio becomes higher as the number of candidates for each template increases, but the computation takes longer.

4.4.1 *Rotation angles*

Figure 12 shows the relationship between the accuracy and the rotation angles. The rotation matching test was performed with each rotation of 5 deg. The transverse error Δx and the longitudinal error Δy were required to satisfy the constraint $(N-\Delta x)(N-\Delta y)\geq 80\% N^2$ in order to have 80% of the overlap between the template and the subimage the

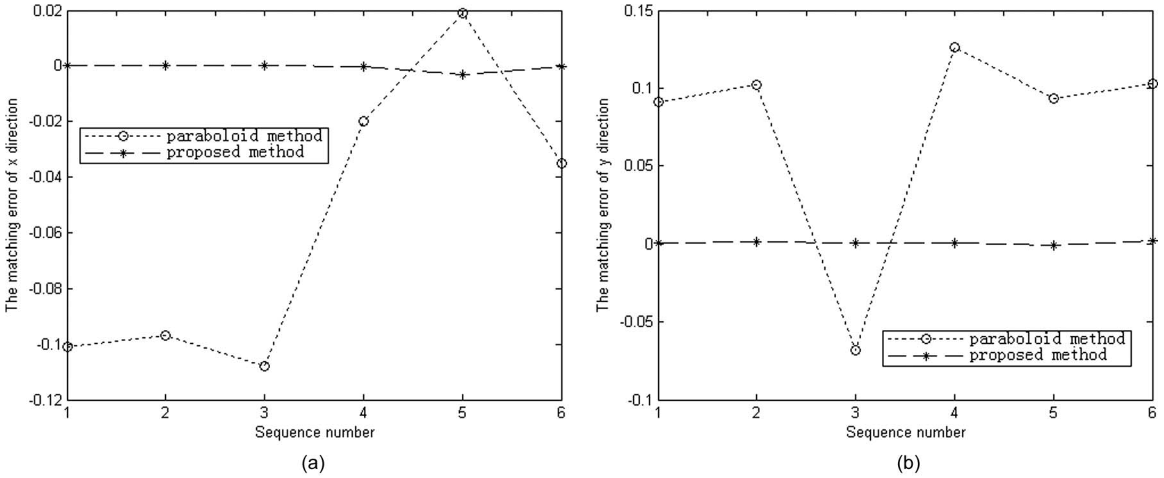


Fig. 10 Comparison of the matching error between the paraboloid surface fitting method and the proposed method. (a) Error comparison on the *x* axis; and (b) error comparison on the *y* axis.

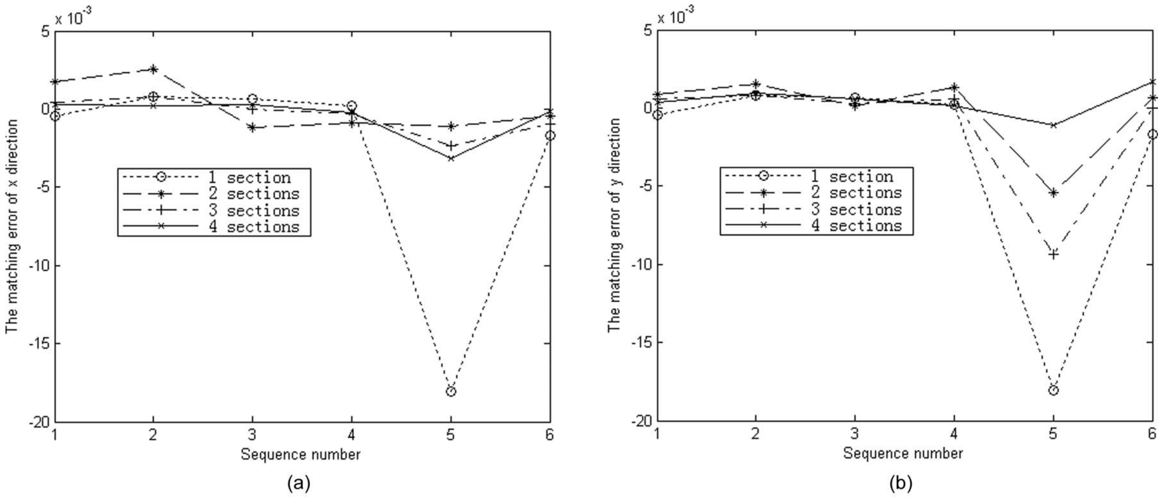


Fig. 11 Comparison of the matching error with different sections. (a) Error comparison on the *x* axis; and (b) error comparison on the *y* axis.

Table 3 Comparison of the experiment results.

Sequence Number	Coordinates	Paraboloid surface fitting method		Proposed method (4 sections)	
		Matching position	Error	Matching position	Error
1	(40, 90)	(40.101, 89.909)	(−0.101, 0.091)	(39.99975, 90.00036)	(0.00025, 0.00036)
2	(175, 262)	(175.097, 261.898)	(−0.097, 0.102)	(174.99980, 261.99904)	(0.00020, 0.00096)
3	(67, 196)	(67.108, 196.068)	(−0.108, −0.068)	(66.99975, 195.99944)	(0.00025, 0.00056)
4	(30, 140)	(30.02, 139.874)	(−0.02, 0.126)	(30.00023, 139.99984)	(−0.00023, 0.00016)
5	(24, 129)	(23.981, 128.907)	(0.019, 0.093)	(24.00314, 129.00107)	(−0.00314, −0.00107)
6	(190, 116)	(190.035, 115.897)	(−0.035, 0.103)	(190.00019, 115.99832)	(−0.00019, 0.00168)

Table 4 Comparison of the experiment results.

		Proposed method (different sections)							
		1 point(1 section)		2 points(2 sections)		3 points(3 sections)		4 points(4 sections)	
SN	Coordinates	Matching position	Error	Matching position	Error	Matching position	Error	Matching position	Error
1	(40, 90)	(40.00043, 90.00043)	(−0.00043, −0.00043)	(39.99827, 89.99911)	(0.00173, 0.00089)	(39.99958, 89.99943)	(0.00042, 0.00057)	(39.99975, 90.00036)	(0.00025, 0.00036)
2	(175, 262)	(174.99917, 261.99917)	(0.00083, 0.00083)	(174.99747, 261.99850)	(0.00253, 0.0015)	(174.99917, 261.99917)	(0.00083, 0.00083)	(174.99980, 261.99904)	(0.00020, 0.00096)
3	(67, 196)	(66.99935, 195.99935)	(0.00065, 0.00065)	(67.00115, 195.99986)	(−0.00115, 0.00014)	(67, 195.99972)	(0, 0.00028)	(66.99975, 195.99944)	(0.00025, 0.00056)
4	(30, 140)	(29.99980, 139.99980)	(0.0002, 0.0002)	(30.00089, 139.99869)	(−0.00089, 0.00131)	(30.00030, 139.99941)	(−0.00030, 0.00059)	(30.00023, 139.99984)	(−0.00023, 0.00016)
5	(24, 129)	(24.01803, 129.01803)	(−0.01803, −0.01803)	(24.00110, 129.00539)	(−0.00110, −0.00539)	(24.00236, 129.00936)	(−0.00236, −0.00936)	(24.00314, 129.00107)	(−0.00314, −0.00107)
6	(190, 116)	(190.00166, 116.00166)	(−0.00166, −0.00166)	(190.00047, 115.99938)	(−0.00047, 0.00062)	(190.00093, 116)	(−0.00093, 0)	(190.00019, 115.99832)	(−0.00019, 0.00168)

same size as the template, which can guarantee obtaining the accurate scaled and rotated factors by using the Fourier-Mellin transform (Fig. 13),

where N is the size of the template, and N^2 is the area of the template.

4.4.2 *Gaussian noise*

Additive white Gaussian noise was added to each image. The plot in Fig. 14 shows the relationship between the accuracy and the number of candidates. As evidenced by the experiment, when 40 candidates were selected for each template, the accuracy ratio ranked highest. The accuracy degrades by only a small portion as the SNR decreases because the binarization reduces the effect of Gaussian noise.

4.4.3 *Brightness/contrast variation*

Figure 15 shows the matching accuracy result when the brightness of the image changes due to the lighting conditions. The proposed algorithm survives by more than 90% even with severe changes (−40 to 40 in gray values) in brightness. This is because the binarization eliminates the effect of the overall brightness change. On the other hand, the correlation algorithms are not sensitive to brightness variation. Figure 16 shows the matching accuracy when the contrast of the image changes. The proposed algorithm turned out to be robust to contrast change too.

Equations (22) and (23) were used to change the brightness and the contrast of the images. The abscissa in Figs. 15 and 16 indicates the value of d in the following equations:

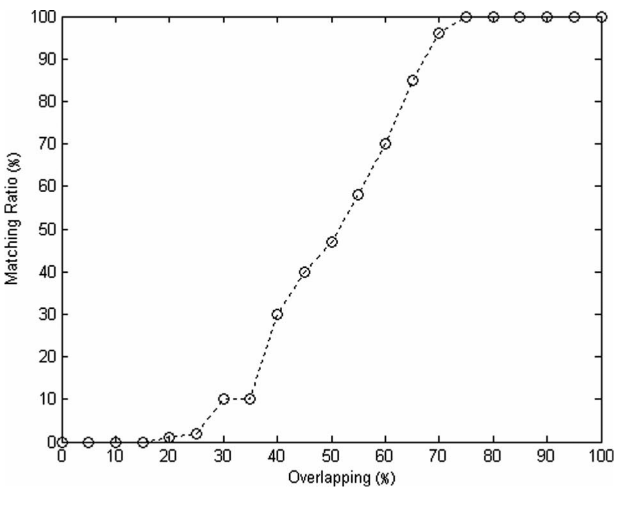


Fig. 12 Matching ratio with overlapping variation.

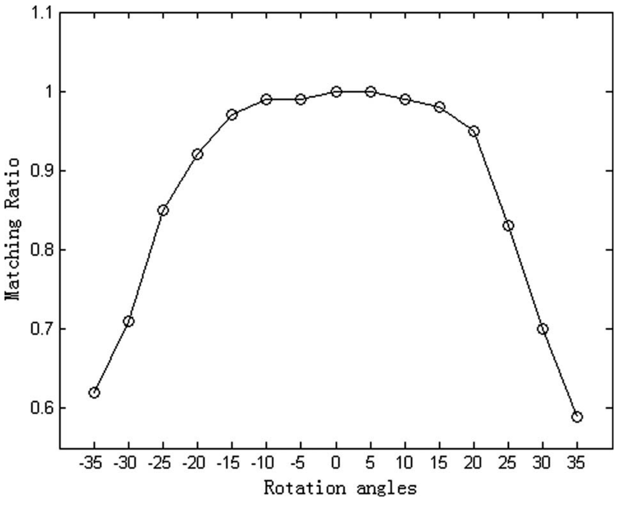


Fig. 13 Matching ratio with rotation angle variation.

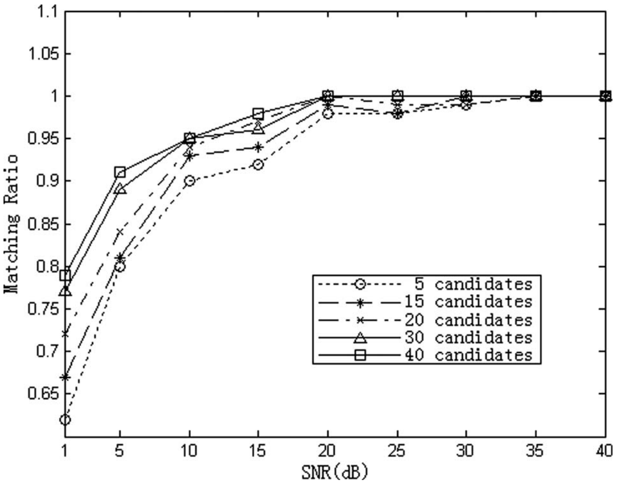


Fig. 14 Matching ratio under Gaussian noise.

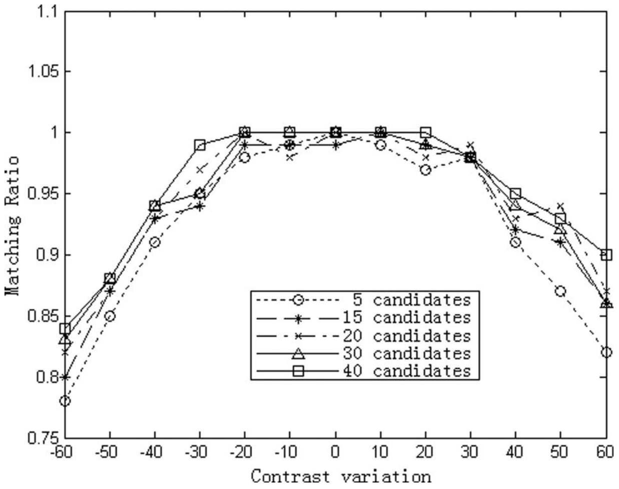


Fig. 16 Matching ratio with contrast variation.

$$f_b = f + 255 \times \frac{d}{100}, \tag{22}$$

$$f_c = f + (f - f_{avg}) \times \frac{d}{100}. \tag{23}$$

4.5 *Terrain Matching Application*

We applied the proposed method for terrain matching using a sand table simulating real terrain. Generally, the real-time 3-D terrain data is recovered from the optical images in the observed scene, which is called the reference evaluation map (REM). Suppose that a reference digital elevation map (DEM) called the airborne-referenced DEM is achieved with the feature matching method and surface natural neighbor interpolation in position A (Ref. 23 and 24. In position B, with the same method, we will achieve REM, which is also called real-time REM. With the proposed method, we match the achieved REM and DEM.

REM and DEM are both converted to contour maps, which are called the REM contour map (RCM) and the DEM contour map (DCM), using the same method as in Yu et al..[25] Thus, the problem of terrain matching is converted to the matching of RCM and DCM.

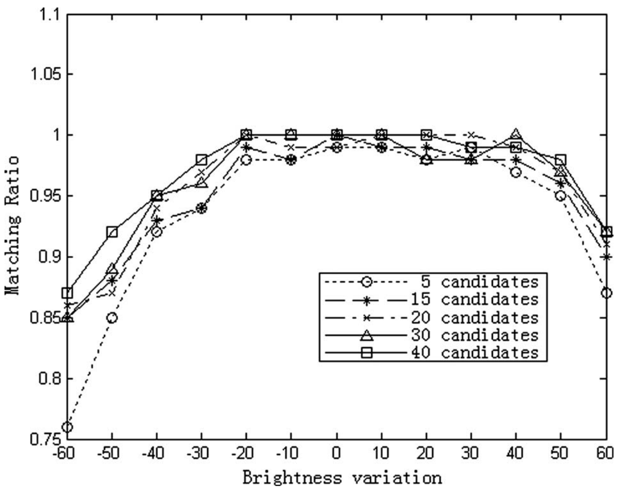


Fig. 15 Matching ratio with brightness variation.

For this particular experiment, the scaling factor between RCM and DCM is 1 because the REM has the same resolution as the DEM. Therefore, we consider only the situation in which the two maps contain rotation and translations.

Additive white Gaussian noise was added to the 3-D terrain data. Figure 17 is the matching results of the REM with −15-dB Gaussian noise, and the joined −15-dB noise can indicate that the fluctuating scope of 3-D data is $[-17.4, 17.4]$ mm, while the real sand table height is 287 mm therefore, the fluctuating scope of 3-D is equivalent to the precision changes $\delta \approx \pm 17.4/287$—that is, about $\pm 6.1\%$. Figure 17(a) shows the sparse feature matching result using the method in Li and Zhang.[23] Figure 17(b) is the sparse feature matching result with −15-dB noise and 20-deg rotation. Figures 17(c) and 17(d) are DEM and REM, respectively. The contours of DEM are shown in Figure 17(e). Figure 17(f) is the contour maps of REM. A set of templates intercepted from Fig. 17(f) is shown in Fig. 17(g), and the size of RCM is 60×60 pixels. Using the method proposed in Sec. 3.2, the rotation angle is 19.8757 deg between RCM and DCM, so the error is 0.1243 deg. RCM overlaid onto DCM is shown in Fig. 17(h).

Experiment 1. Figure 18 shows the matching results of REM and DEM obtained in the different position and height. There are certain differences between them due to surface interpolation, so the constructed contour maps are different. The matching results are that the rotation is 11.2541 deg and the translations are $(-22.7124, 12.3686)$. But the real parameters are not known.

Experiment 2. Figure 19 shows the matching results of REM and DEM obtained in the different position and height. There are a certain differences between them due to surface interpolation, so the constructed contour maps are different. The matching results are that the rotation is

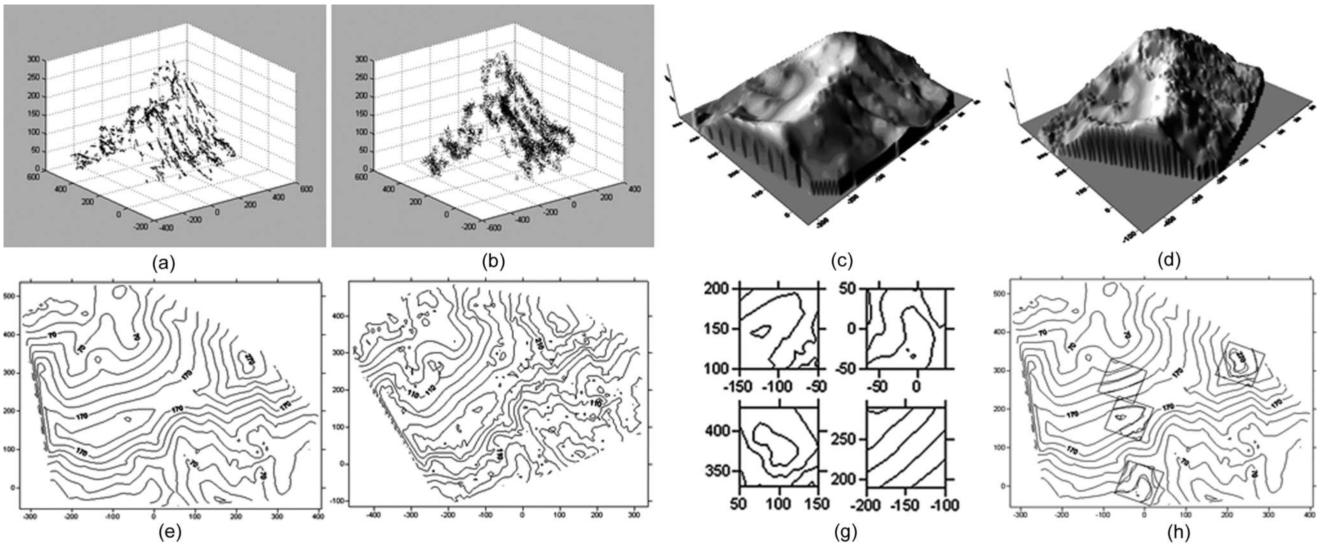


Fig. 17 Matching result of an REM with −15-dB noise to the rotated reference DEM. (a) The sparse feature matching result; (b) the sparse feature matching result with −15-dB noise and 20-deg rotation; (c) reference DEM; (d) REM; (e) contour map of (c); (f) contour map of (d); (g) a set of templates; and (h) RCM overlaid onto DCM.

20.5101 deg and translations are $(-4.9662, -19.0128)$. And the real parameters are also not known.

5 Conclusions

In this paper, we proposed a fast and robust algorithm for template matching of an arbitrarily rotated image when the fluctuating scope of the rotation angle is $[-20\text{ deg}, 20\text{ deg}]$. The proposed method employed to select candidates has efficiently reduced the order of feature space using the AND operation. In the experiment, we showed that it was computationally simple and fairly reliable for selecting candidates. In addition, the time complexity for the candidate selection process increased very slowly compared to the template size. But it is necessary to point out that the ex-

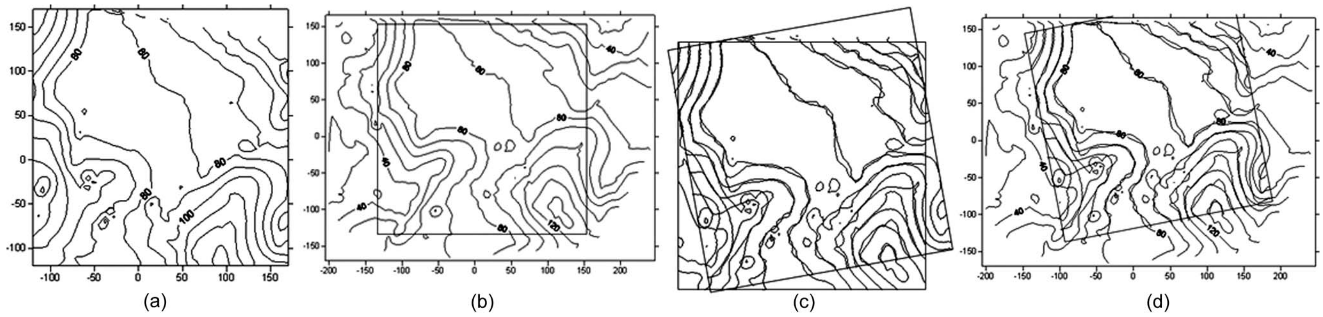


Fig. 18 Matching results of sand table simulating the real terrain. (a) RCM; (b) DCM; (c) matching results; and (d) RCM overlaid onto DCM.

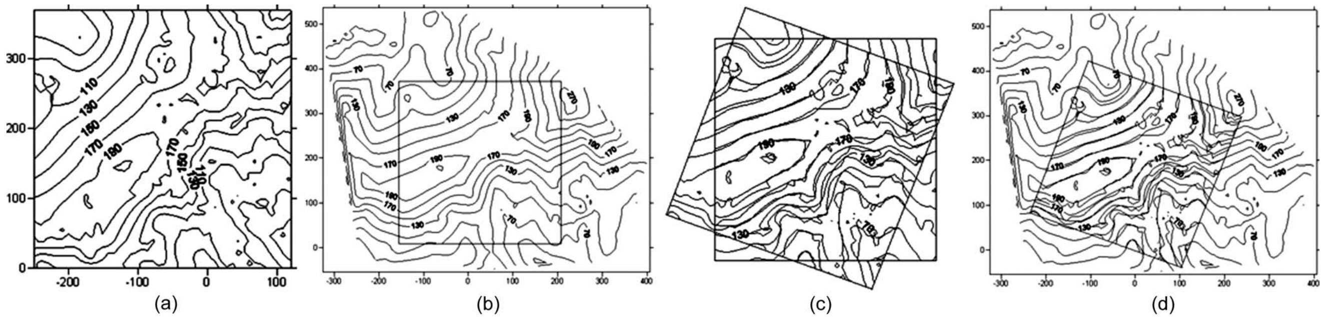


Fig. 19 Matching results of sand table simulating the real terrain. (a) RCM; (b) DCM; (c) matching results; and (d) RCM overlaid onto DCM.

cellent performance of the proposed method employed to select candidates was achieved at the expense of matching accuracy. Therefore, the Fourier-Mellin transform was used to remedy this problem. Experimental results also show that the matching precision is subpixel using a novel method, which is better than the paraboloid surface fitting method.

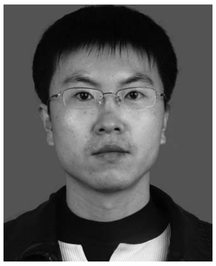
Acknowledgment

We appreciate the effort of the reviewers, whose helpful comments improved this paper.

References

1. J. K. Aggarwal, L. S. Davis, and W. N. Martin, "Correspondence process in dynamic scene analysis," *Proc. IEEE* **69**(5), 562–572 (1981).
2. E. S. Baker and R. D. Degroat, "A correlation-based subspace tracking algorithm," *IEEE Trans. Signal Process.* **46**(11), 3112–3116 (1998).
3. L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.* **24**(4), 325–376 (1992).
4. U. M. Cahnvon and R. Bajcsy, "Adaptive correlation tracking of targets with changing scale," Univ. of Pennsylvania: GRASP Laboratory Technical Report No. MS-CIS-96-22 (1996).
5. R. G. Caves, P. J. Harley, and S. Quegan, "Matching map features to synthetic aperture radar (SAR) images using template matching," *IEEE Trans. Geosci. Remote Sens.* **30**(4), 680–685 (1992).
6. J. P. Secilla and N. Garcia, "Template location in noisy pictures," *Signal Process.* **14**, 347–361 (1987).
7. R. Y. Wong and E. L. Hall, "Sequential hierarchical scene matching," *IEEE Trans. Comput.* **27**(4), 359–366 (1978).
8. A. Rosenfeld and A. Kak, *Digital Image Processing*, 2nd ed., vol. **2**, Academic Press, Orlando (1982).
9. S. W. Lee and W. Y. Kim, "Rotation-invariant template matching using projection method," *Proc. Korean Inst. Telecommun. Electron.* **19**(1), 475–476 (1996).
10. R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," *Graph. Models Image Process.* **54**(5), 438–460 (1992).
11. Q. S. Chen, D. Michel, and F. Deconinck, "A symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(12), 1156–1168 (1994).
12. C. Kan and M. D. Srinath, "Invariant character recognition with Zernike and orthogonal Fourier-Mellin moments," *Pattern Recogn.* **35**(1), 143–154 (2002).
13. V. Martin, "Wavelets and filter banks: theory and design," *IEEE Trans. Signal Process.* **40**(9), 2207–2232 (1992).
14. B. Murat, E. K. Misha, and L. M. Eric, "Wavelet domain image restoration with adaptive edge-preserving regularization," *IEEE Trans. Image Process.* **9**(4), 597–608 (2000).
15. L. D. Stephano and S. Mattoccia, "Fast template matching using bounded partial correlation," *Mach. Vision Appl.* **13**(4), 213–221 (2003).
16. L. A. D. Hutchinson and W. A. Barrett, "Fast registration of tabular document images using the Fourier-Mellin transform," in *IEEE Proc. First Int. Workshop Document Image Analysis for Libraries (DIAL'04)* **16**(12), 1256–1270 (2004).
17. X. Zhao, "A study on robust image curve matching techniques," PhD. thesis, Beijing Institute of Automation, Chinese Academy of Sciences (2004).
18. A. Averbuch and Y. Keller, "FFT based image registration," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 3601–3608 (2002).
19. Reddy, B. Srinivasa and B. N. Chatterji, "An FFT-based techniques for translation, rotation, and scale-invariant image registration," *IEEE Trans. Image Process.* **5**(8), 1266–1271 (1996).
20. C. Siwaphon, C. Orachat, and T. Kiatnarong, "Image registration using Hough transform and phase correlation," in *Proc. Int. Congress on Anti-Cancer Treatment (ICACT)*, pp. 20–22 (2006).
21. S. G. Kim and W. Y. Kim, "Hierarchical grayscale template matching for an arbitrarily oriented object," in *Proc. Korea Signal Process. Conf. 96*, pp. 1335–1338 (1996).
22. M. S. Choi and W. Y. Kim, "A novel two stage template matching method for rotation and illumination invariance," *Pattern Recogn.* **35**(1), 119–129 (2002).
23. H. Li and G. Zhang, "A new fast edge-matching algorithm based on corner constraint and edge constraint," *Proc. SPIE* **6358**, 635815 (2006).
24. M. Yanalak, "Effect of gridding method on digital terrain model profile data based on scattered data," *J. Comput. Civ. Eng.* **17**(1), 58–67 (2003).
25. Q. Yu, J. Tian, and J. Liu, "A novel contour-based 3D terrain matching algorithm using wavelet transform," *Pattern Recogn. Lett.* **25**(1), 87–99 (2004).

Guangjun Zhang received his BS, MS, and PhD degrees from the Department of Precision Instruments Engineering, Tianjin University, China, in 1986, 1989, and 1991, respectively. He was a visiting professor at North Dakota State University from 1997 to 1998. He is currently a professor and dean of the School of Automation Science and Electrical Engineering, BeiHang University, China. His research interests are machine vision, laser precision measurement, optoelectronic precision instrumentation, and artificial intelligence.



Ming Lei received both his BS and his MS from the Automatic Control Department, Daqing Petroleum Institute, China, in 2000 and 2003, respectively. Since 2004, he has been a PhD candidate at Beijing University of Aeronautics and Astronautics, China. His research interests include pattern recognition for machine vision systems and content-based image retrieval.

Xulin Liu biography and photograph not available.