Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 9, No. 2, 2024

Tool Damage Detection Method Based on Improved Threshold Segmentation Algorithm

Lunwen Peng[1], Shenghu Pan[2]

[1]Department of Mechatronics Engineering, Southwest Petroleum University, Chengdu 610500, China
[2]Oil and gas equipment technology Sharing and Service Platform of Sichuan Province 610500, China

Abstract: In order to solve the problem that the current tool wear defects are difficult to be collected by the visual inspection system, an Otsu threshold segmentation algorithm based on particle swarm optimization is proposed.to detect tool wear The algorithm improved update strategy for inertia coefficients which effectively expanding the search scope of the algorithm and shortens the running time of the algorithm. By adding a perturbation equation to the particle swarm, solved the problem of traditional particle swarm optimization algorithms easily falling into local optima. Finally, an experimental platform is built to verify the effectiveness of the algorithm. This inspection method can achieve the identification of tool damage areas and the measurement of tool damage amount, and has advantages such as high recognition accuracy and fast running speed compared to traditional Otsu algorithm, Canny algorithm ,local threshold segmentation and so on. The research results have certain reference value for the actual tool defect detection system.

Keywords: Tool wear; Visual detection; Otsu: PSO.

1. Introduction

At present, cutting is still one of the important machining processes, and the tool as the core tool in cutting, the quality of the tool will affect the machining accuracy and quality of the machine tool. In actual processing, it is usually determined by worker experience whether the tool has reached scrap conditions. This method will result in a considerable number of tools that have not yet reached the scrapping conditions to be replaced. If the tool has been excessively worn and has not been replaced, it will significantly reduce the dimensional accuracy of the parts and even cause safety accidents. Therefore, in order to accurately measure the wear of the tool, it is necessary to detect the tool wear area.

The method of detecting and measuring tool state can be divided into direct measurement method and indirect measurement method. Indirect measurement method is to detect the characteristic changes of cutting force, sound, vibration, workpiece surface texture and other related physical quantities caused by tool wear through sensors, so as to measure the tool indirectly.

The indirect measurement method has some disadvantages: First, to obtain the characteristics of tool wear from the signal requires the operator to know the relevant domain knowledge; Secondly, sensor-based tool detection devices are often only applicable to a certain type or type of tool, and the generalization is poor and the signal detected by the sensor is doped with a large number of interference signals, which will affect the test results [1].

The direct method is to detect the defect directly on the image of the tool. Ye Zukun et al. [2] proposed a damage detection method based on local threshold segmentation. The whole image is divided into many small pixel blocks, the segmentation threshold of each pixel block is obtained, and the whole pixel is scanned according to the maximum segmentation threshold, and the tool damage location is obtained. However, due to the complex surface texture of the tool, the acquired tool image contains a lot of noise, which is easy to select the noise as the segmentation threshold and lose the damage location information. Deng Xiaopeng et al. [3] combined Otsu algorithm and region growth method to propose an adaptive regional growth defect detection method, which can automatically locate the tool wear area and identify the size of tool damage. However, Otsu algorithm is susceptible to the influence of tool texture and reflection, resulting in difficulty in locating the tool defect location. Zhou Junjie [4] et al. proposed a method of edge detection followed by pixel detection. Firstly, Canny operator is used to rough position the image, then the sub-pixel edge detection algorithm of Zernike moment is used to improve the measurement accuracy, and finally the pixel points are simulated to complete the measurement. However, this method is very sensitive to grayscale fluctuations, and it is easy to identify the inner texture of the tool as the boundary information, which affects the recognition accuracy, and the edge contour discontinuity is easy to occur. Li Shanshan et al. [5] obtained the tool defect region through the region growth method, and obtained the tool characteristic value through the minimum external matrix. However, it is difficult to calculate the damage amount accurately because the gray value of the tool damage surface fluctuates greatly. Liu Jianchun et al. [6] proposed an adaptive threshold segmentation algorithm, which combined collinear contours to fit the edge contour and extract the defect area. However, due to the small defect area of the tool, it is easy to identify the non-wear area as the wear area.

2. Tool image damage detection method

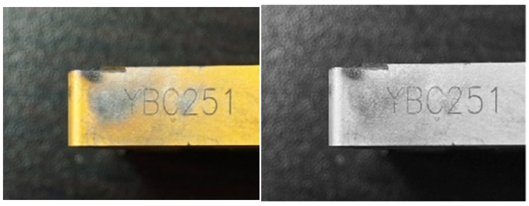
2.1. Image preprocessing

2.1.1. Image graying

RGB model is a color model defined according to human eye perception. RGB model converts color images into 8-bit gray images according to different proportions of three-channel pixel values. According to experiments, this ratio column coefficient has the best effect in processing color images [7]. R, G and B respectively represent the three color components of red, green and blue. In general,, and the formula is as follows:

$$Gray = 0.299R + 0.587G + 0.114B \tag{1}$$

The image is imported into the software platform for gray level conversion, and the original image is taken on the experimental platform. The image before conversion is shown in Figure a), and the effect of the tool after gray level conversion is shown in Figure b).



a) Tool raw image b) Tool gray image

Figure 1. Image graying

2.1.2. Sub heading

In the process of image acquisition, transmission and compression, the camera will be affected by light, which will lead to the acquisition of images containing noise, noise will lead to the loss of key information in the image, and noise may bring certain interference to the identification of defects. The function of image filtering is to eliminate the noise in the image and dilute the irrelevant information, retain the effective information in the image, and improve the signal-to-noise ratio So it is necessary to use the mean filtering algorithm to eliminate the noise and smooth the image

The mean filtering algorithm is a method to smooth the image contour. The algorithm sets a window (2a+1)(2b+1), sets each pixel on the image as the center point of the window for traversal, and assigns a value to the center pixel with the median value of the pixel proximity. The average value of each window pixel is:

$$g_{(r,h)} = \frac{1}{(2a+1)(2b+1)} \sum_{i=-a}^{a} \sum_{j=-b}^{b} g'_{(r-i,h-j)} \tag{2}$$

Where, g (r,h) is the average gray value of each pixel in the window, and g '(R-I, H-j) is the gray value of each pixel in the window.

2.2. Otsu threshold segmentation method.

Threshold segmentation method is a common image segmentation method, the main purpose of threshold segmentation is to classify each pixel in the image through the difference of pixel gray value, and realize the distinction between background and target, so as to extract the interesting target from the image.

Suppose an image has L gray levels, ni (0< i< L-1) is the number of pixels in different gray levels, then the total number of pixels N of the image is:

$$N = \sum_{i=0}^{L-1} n_i \tag{3}$$

Normalize the grayscale histogram:

$$\sum_{i=0}^{L-1} p_i = 1 \tag{4}$$

Then the probability that the gray value of the pixel in the figure is i is:

$$p_i = \frac{n_i}{N} \tag{5}$$

Select a threshold value K, and divide pixels into two intervals X0 and X1 according to different gray levels. X0 is the pixel with gray level [0, k-1], and X1 is the pixel with gray level [K, L-1]. Then the probability of the pixel falling into the two intervals is:

$$x_0 = \sum_{i=0}^{K-1} p_i, x_1 = \sum_{i=K}^{L-1} p_i = 1 - x_0 \tag{6}$$

The mean gray values of X1 and X2 are:

$$\mu_0 = \frac{1}{x_0} \sum_{i=0}^{K-1} ip_i, \mu_1 = \frac{1}{x_1} \sum_{i=K}^{L-1} ip_i \tag{7}$$

The variance between X0 and X1 classes is:

$$\begin{aligned} \sigma^2 &= x_0(\mu_0 - \mu)^2 + x_1(\mu_1 - \mu)^2 \\ &= x_0 x_1(\mu_1 - \mu_0) \end{aligned} \tag{8}$$

The classical Otsu segmentation method is to iterate the value of threshold K in the gray level in the range of [0, L-1] until the optimal segmentation threshold is obtained. In this paper, the Improved Particle Swarm Optmization algorithm is used to optimize the optimal threshold of Otsu algorithm, so as to improve the convergence speed and accuracy of the algorithm.

2.3. PSO algorithm.

Particle Swarm Optimization (PSO) abstracts a flock of birds into particles without volume and mass, takes the position of individual birds or food as the optimal solution, and the particles fly at a certain speed in the solution space, using the information of the group to interact with the information of the optimal individual. In order to adjust its flight speed, the entire particle swarm converges towards the optimal individual under the condition of ensuring population diversity [10]. The algorithm steps are as follows:

First, let particle swarm size be N, search space dimension be D, position vector Xi={x1, x2, x3, xiD}, velocity vector Vi{V1, V2, V3, ViD,}. i is the particle number, the position vector of the particle represents the position of the solution in the solution space, and the velocity vector represents the direction and rate of the particle moving in the solution space. The particle swarm finds the optimal solution through continuous iteration, and the particle iteration formula is as follows:

$$X_{iD}(t+1) = X_{iD}(t) + V_{iD}(t+1) \tag{9}$$

Where, t is the number of iterations, ω is the inertia weight of the particle, representing the influence of the previous iteration speed on the current speed. c1 and c2 are acceleration factors, c1 controls the learning of individual particles, and c2 controls the mutual learning between particles and particle swarm. Pbest is the optimal solution reached by the individual, Gbest is the optimal solution reached by the group.

The determination of inertia weight is a very important part of particle swarm optimization algorithm, which will directly affect the convergence speed and accuracy of the algorithm. The value of ω is high and the search range of the algorithm is large. The value small algorithm of ω converges quickly. The traditional PSO algorithm has some defects, such as low precision and slow convergence speed, because its inertia weight ω is constant. At present, most studies [11][12] modify the parameter part of the algorithm, such as improving the decline strategy of inertia weight, such as linear decline strategy of inertia weight, that is, the inertia weight decreases with the increase of the number of iterations.

The updated formula is:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \frac{\mathrm{m}}{M} \tag{10}$$

ωmax, ωmin are the maximum and minimum inertial weights. m is the current number of iterations, and M is the total number of iterations. The adaptability of this update strategy is wide, and the convergence of the algorithm is improved on the basis of guaranteeing speed and accuracy, but there are some defects in this algorithm. In the early stage of the algorithm, more emphasis should be placed on search to improve the optimization ability of the algorithm, but the linear decline strategy of inertia weight will lead to the weak search ability of the particle swarm in the early stage and can not find the global optimal solution. In the later stage, the algorithm should focus more on convergence, and the more the inertia weight linear decline strategy converges, the slower the algorithm runs. To sum up, in order to improve the convergence speed and accuracy of the algorithm, the inertia coefficient cosine decreasing strategy is adopted to improve the optimization ability of particle swarm in the early stage and the convergence ability of particle swarm in the later stage. The updated formula is as follows:

$$\omega_k = \frac{1}{2}[\cos(\pi \frac{m}{M}) \times (\omega_{\max} - \omega_{\min}) + (\omega_{\max} + \omega_{\min})] \tag{11}$$

A large number of studies have found that when the maximum inertia weight value ωmax is 0.9 and the minimum inertia weight value ωmin is 0.4[13], the algorithm performance will be greatly improved. Set the maximum number of iteration M of the algorithm to 100 and bring it into the formula for calculation.

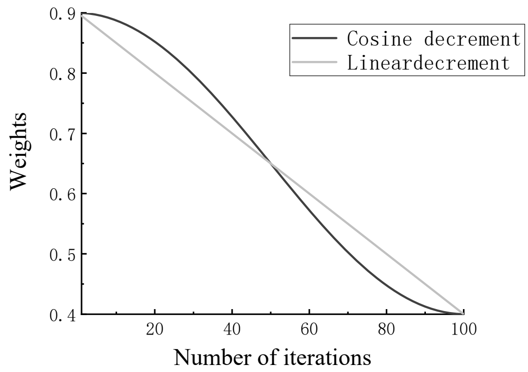


Figure 2. Inertia weight coefficient comparison

he results are shown in Figure 2. It can be seen that in the early stage of iteration, the value of the inertia coefficient ω is large, and the particle swarm has a strong ability to conduct global search, which is conducive to the group's global learning and avoids the algorithm falling into the local optimal solution. In the late iteration, the inertia weight value ω is smaller, the particle local search ability is stronger, and the search speed is faster, which is conducive to improving the search efficiency of PSO algorithm.

However, the search result of the algorithm may still be the local optimal solution. However, it can be confirmed that when the particle swarm converges towards the global optimal solution, the individual optimal value in the next iteration must be higher than the individual optimal value in the last iteration. Therefore, judging the size of the individual optimal position response function beforeand after the iteration can determine whether a single particle is evolving [14], that is:

$$A(i,t) = \begin{cases} 1 & f_{\sigma}(xiD,t) > f_{\sigma}(xiD,t-1) \\ 0 & f_{\sigma}(xiD,t) = f_{\sigma}(xiD,t-1) \end{cases} \tag{12}$$

Where, it represents whether the I-th particle in the t iteration is evolving towards the particle individual optimization.

$$P_{s(t)} = \frac{\sum_{t=1}^{N} A(i,t)}{N} \tag{13}$$

Where, N represents the total number of particles, and Ps(t) is a value between [0,1], the size of which represents the number of individuals in the whole particle swarm evolving toward the individual optimal position. If Ps(t) is large, most of the particles in the population have not reached the global optimal position; if Ps(t) is small, it is considered that most of the particles in the population have reached the global optimal position. If we set the termination condition as $Ps(t) < 0.05$, it indicates that the particle swarm may reach the optimal position of the population. At this time, the algorithm continues to perform three iterations. If the fitness value of the particle swarm after iteration is still higher than that of the previous iteration, it is determined that the particle swarm has reached the optimal position of the population for the first time. However, the particle swarm may still be a local optimal solution at this time. After recording the current optimal position, a perturbation [15][16] is added to the particle swarm to make the particle swarm abandon the current optimal solution and find the global optimal solution again. After adding the disturbance, the updated equation is:

$$s_1 = e^{\frac{f_i - f_\sigma}{|f_i| + \varepsilon}}, s_2 = e^{f_i - f_k} \tag{14}$$

$$x_{iD}(t+1) = x_{iD}(t) + s_1 rand()[Pbest_{iD} - x_{iD}(t)] \\ + s_2 \mathrm{rand}()[Gbest_{iD} - X_{iD}(t)] \tag{15}$$

Where, s1 and s2 are the updated acceleration factors, are the fitness value of the i th particle, are the global optimal fitness value, and are the minimum value of non-zero. The difference between equation (16) and equation (9) is that by introducing the acceleration factor update equation, perturbations of different intensity are added to the particle swarm, so that the particles jump out of the current search area, search again, and get closer to the global optimal position, so as to prevent the particle swarm from falling into the local optimal solution.

2.4. Improved Otsu threshold segmentation process of particle swarm optimization algorithm.

The algorithm flow is shown in Figure 4. The implementation steps of the algorithm are as follows:

1) The tool image is preprocessed to obtain the gray level histogram of the image, and then the gray level histogram is normalized.

2) The inter-class variance formula (8) is derived according to equation (6) and equation (7). In order to find the maximum segmentation threshold for inter-class variance, the improved and optimized PSO algorithm is adopted.

3) Set the particle swarm size to N, randomly initialize the particle position and initial velocity, and set the maximum inertia weight value ωmax, minimum inertia weight value ωmin, acceleration coefficient c1,c2, maximum number of iterations, etc.

4) Use formula (8) as fitness function to calculate the fitness value of particles, and update the individual optimal position Pbest and global optimal position Gbest for each particle according to the fitness value.

5) Use equations (9) and (10) to calculate the position and velocity of the particle

Update: In the iterative process, the inertia coefficient of the particle swarm is reduced by the cosine according to equation (12).

6) Iterate for three consecutive times to determine f (xm+) > Whether f(xm) is true (the state function fitness value of this iteration is greater than the state function fitness value of the last iteration) If yes, jump to Step 5), otherwise return to step 4).

7) Use formula (16) to add perturbations to the particle swarm.

8) Three consecutive iterations to determine f(xm+1)> If yes, go to Step 7. If no, go to Step 3.

9) Calculate the updated particle fitness value and record the result

To the individual optimal position and the global optimal position. Then, we continue to use formula (9) and (10) to update the particle position and velocity.

10) If the number of continuous invariable particle swarm optimal adaptation values reaches the specified value or the number of algorithm iterations reaches the set value. Then the global optimal position Gbest is output as the optimal segmentation threshold K.

11) Set K as the segmentation threshold for image segmentation.

3. Experimental Results and Analysis

In order to verify the effectiveness of the application of the improved particle swarm optimization Otsu algorithm in the process of tool defect detection, this paper built an image acquisition platform, as shown in FIG. 4, which mainly consists of 5 million pixel CCD camera, 18FA0828-6MP industrial lens, dot matrix light source, base and camera support [18]. The base is placed on a horizontal table, and the base is provided with a dot matrix light source to provide a good image acquisition environment for the camera. The camera bracket is vertically fixed on the base, the industrial camera is fixed through the beam bracket and is horizontal with the vertical bracket, and the image detection table is vertically fixed with the vertical bracket.

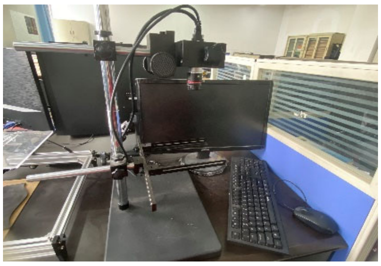


Figure 3. Image acquisition device

Before preparing for testing, it is first necessary to clean the tool. Then fix the tool on the image detection table, adjust the distance between the camera and the tool by adjusting the beam support to get a clear image of the tool damage. Finally, the CCD industrial camera is used to collect the information of the tool back tool face and then transmit it to PC for the next step of image processing.

PC configuration:
Intel(R)Core(TM)i5104004,8GRAM,MATLAB R2022a.

In order to verify the effectiveness of the tool detection algorithm proposed in this paper, 4 tools with different wear degrees were selected as samples for experiments. The real wear of the tool was measured by the microscope. At the same time, the algorithm in this paper was used to carry out repeated experiments on four tools, record the experimental results and calculate the average accuracy. The average accuracy rate is calculated as follows:

$$\xi = (1 - \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - y_i|}{y_i}) \times 100\%$$

Where, is the average accuracy, is the true value, is the measured value, and n is the number of experiments. The measured values were selected to calculate the average damage width of the tool surface.

Figure 3 shows the recognition effect of the improved PSO+Otsu algorithm for four tools with different wear degrees and different lighting conditions. It can be seen that in different lighting environments, the algorithm has a good effect on noise processing, and can accurately and effectively extract the tool damage area. #1 is the first tool, #2 is the second tool, #3 is the third tool, and #4 is the fourth tool. The figure shows the average result of the damage amount of the four tools by the algorithm in this paper. (a) is the tool damage length, and (b) is the average width of the tool damage area. (c) is the tool damage area of the knife.

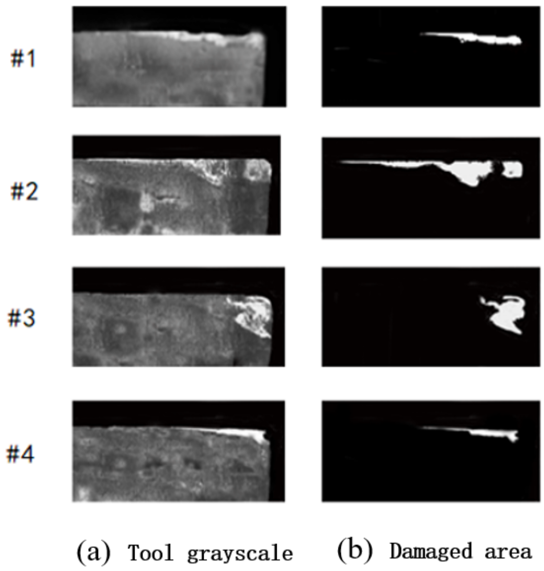


Figure 4. Effectiveness of the proposed algorithm in identifying different tool damage

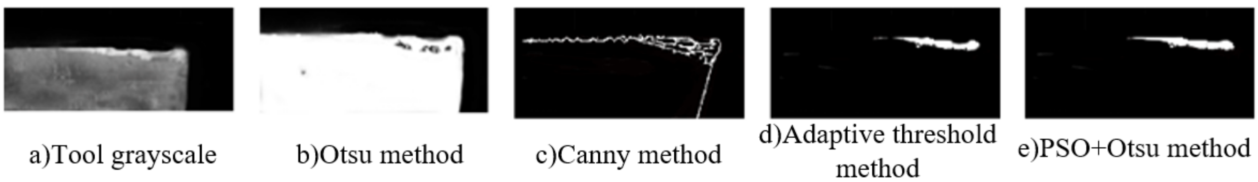


Figure 5. Different algorithms detect tool defect areas

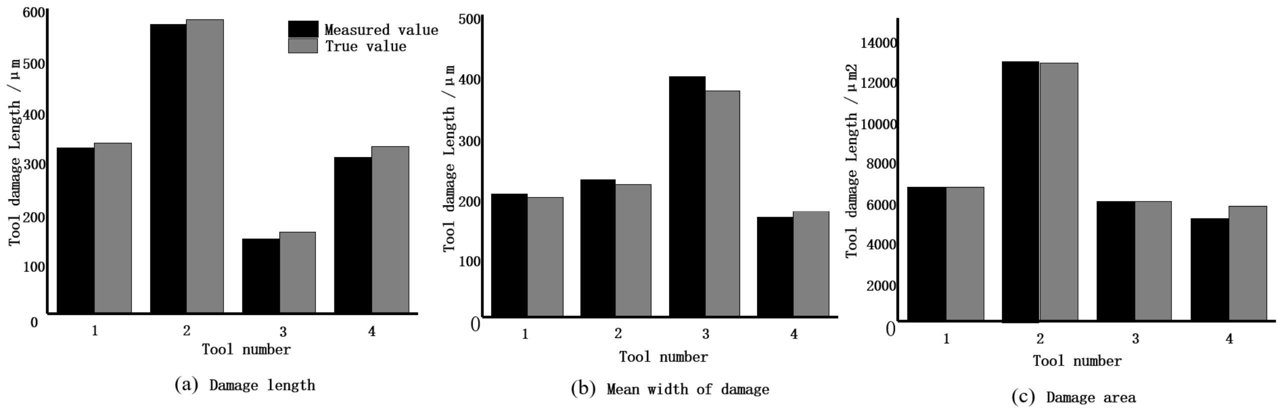


Figure 6. The PSO-Otsu algorithm was improved to detect the result of tool damage

In order to verify the superiority of the proposed algorithm, Otsu method, Canny edge detection algorithm, adaptive threshold segmentation method and improved PSO+Otsu segmentation algorithm are used respectively. Taking the vertical turning tool as an example, the same turning tool is tested five times, and the average running time, iteration times and accuracy of each algorithm are recorded. The quantitative analysis of the detection result data shows that the traditional threshold segmentation method is susceptible to the influence of illumination, which makes it difficult to locate the tool defect location, and even unable to identify the damaged area in severe cases, as shown in Figure 11 (b). However, the traditional edge detection method is prone to problems such as discontinuity of edge

contour, as shown in Figure 11 (c), and cannot form a closed area to calculate the tool damage area. The tool damage detection algorithm used in this paper can automatically and effectively identify the tool damage area, and the recognition effect is outstanding, the recognition efficiency is faster, and the anti-interference is good.

Table 1. Four Scheme comparing

Algorithm	Run time /ms	Average accuracy	Number of iterations
Otsu method	577	-	202
Canny method	496	74.85	-
Adaptive threshold method	456	85.84	189
PSO+Otsu method	189	96.16	22

As can be seen from FIG. 5 and FIG. 6, the tool damage detection method adopted in this paper can automatically and effectively identify the size of the complete tool damage area, with high accuracy, clear identification area and good effect. According to Figure 6 (b) and Table 1, although the traditional Otsu segmentation method has many iterations and long operation time in searching for the optimal threshold of tool segmentation, it still fails to identify the tool damage area. As shown in Figure 6 (c) and Table 2, defects such as edge loss appear in the image of Canny edge detection method, and the algorithm has a long running time, low accuracy and poor effect. According to FIG. 6 (d) and FIG. 6 (e) in combination with Table 1, it can be seen that both of them can clearly obtain the tool damage contour and the image is clear, and the improved PSO+Otsu algorithm can reach 96.16% accuracy in detecting the average width of the tool surface after the tool, which is at least 9.81% higher than the other six algorithms, and also has advantages in terms of running time and number of iterations. It can be seen that the improved PSO+Otsu algorithm proposed in this paper has significantly improved the accuracy and processing speed of tool image damage detection compared with Otsu segmentation method, Canny edge detection method, local threshold segmentation method and other algorithms.

4. Summary

(1) Aiming at the problems of the traditional Otsu algorithm, such as large computation amount, long running time and easy local optimal solution of PSO algorithm, a particle swarm optimization of Otsu threshold segmentation tool damage image visual detection algorithm is proposed. The algorithm adopts the inertia coefficient cosine decreasing strategy to improve the convergence speed of the algorithm, and adds the disturbance equation. When the particle swarm algorithm falls into the local optimal solution, the disturbance is added to the particle swarm to increase the ability of the algorithm to jump out of the local optimal solution.

(2) An experimental detection platform was set up in this paper to detect the tool. The experimental results show that this method can effectively identify the tool damage area and extract the tool damage amount, which has obvious advantages over the traditional detection algorithm. It can provide strong technical support for in-place detection of tool damage.

(3) All the methods in this paper are implemented on the PC side, and the algorithm in this paper is simple and the calculation amount is small. Therefore, the method in this paper can be transplanted to the embedded platform for code adjustment and optimization, so as to apply to more application scenarios.

Acknowledgment

Sichuan Natural Science Foundation 2022NSFSC2002

References

[1] DING Yuhui, Cao Yan, Fu Leijie, BAI Yu. Research status and Prospect of Image Processing Technology in Tool Wear Detection [J]. Manufacturing Technology & Machine Tool, 2020,(4): 56-62.

[2] Ye Zukun, Li Heng, Cha Wenbin, He Yan, Wang Yulin. Tool damage visual detection method using local threshold segmentation [J]. Journal of Xi 'an Jiaotong University,2021, vol. 55 (4): 52-60.

[3] Deng Xiaopeng, Wang Yan, Hong Yu, Hu Xiaofeng. Micro-drilling and Milling Tool wear Detection Method using adaptive region Growth [J]. Journal of Xi 'an Jiaotong University,2021, vol. 55.

[4] Zhou Junjie, Yu Jianbo. Online Measurement of Machining tool wear based on Machine vision [J]. Journal of Shanghai Jiao Tong University,2021, vol. 55 (6): 741-749.

[5] Li Shanshan1, Liu Libing1, Li Li1 et al. Nc Tool Wear State Detection Method Based on Region Growth Method [J]. Manufacturing Technology & Machine Tool, 2017, (2):132-135.

[6] Liu Jianchun, Jiang Junjie, Zou Pilgrimage. Research on Wear Detection Method of End Milling Cutter Based on Machine Vision [J]. Manufacturing Technology & Machine Tool,2020,(1): 136-140.

[7] Zhang Guihua, Feng Yanbo, Lu Weidong. Gray-scale and feature region acquisition of image processing [J]. Journal of Qiqihar University: Natural Science Edition, 2007,23(4):49-52.

[8] Tang Jian. Research on solar multi-band image registration method [D]. Kunming University of Science and Technology.

[9] Lu Yan. Research on image segmentation based on threshold algorithm [D]. Chongqing University,2011

[10] Liu Guihong, Zhao Liang, Sun Jinguang, Wang Wang. An improved particle swarm optimization algorithm for Otsu image threshold segmentation [J]. Computer Science,2016,(3):

[11] Dong Hongbin, Li Dongjin, Zhang Xiaoping. A Particle swarm optimization algorithm for Dynamic Adjustment of inertia weights [J]. Computer Science,2018,45(2):98-102139. (in Chinese).

[12] AO Yongcai, Shi Yibing, Zhang Wei, Li Yanyun. Improved particle swarm optimization based on adaptive inertia weights [J]. Journal of University of Electronic Science and Technology of China,2014,(6): 874-880.

[13] Cheng Wansheng, Zang Xizhe, Zhao Jie, CAI Hegao. Optics and Precision Engineering,2008,(10): 1907-1912. Improved PSO inertia factor for Otsu Threshold Search [J].

[14] LIU Shenxiao, Wang Xuechun, Chang Chaowen. Otsu image segmentation method based on improved particle swarm optimization [J]. Computer Science,2013,(8): 293-295.

[15] Zhang Xinjuan, Lei Xiujuan. Application of improved PSO algorithm in two-dimensional optimal threshold image segmentation [J]. Computer Engineering and Applications, 2011, (26).

[16] Tian Xinghua. Research on improved particle swarm optimization based on Chaotic mapping [D]. Qingdao University, 2020.

[17] Xu Zu. Iso 3685-1977 (E) Life test of single-edge turning tools [J]. Tool Technology,1979,(4): 55-65.

[18] Qin Guohua, Yi Xin, Li Yilan, Xie Wenbin. Optical Precision Engineering,2014, vol. 22 (12): 3332-3341.