A Fast Tool Edge Detection Method Based on Zernike Moments Algorithm

Liu Zixin, Liu Jinyue，Li Yang, Weng Liming

School of Mechanical Engineering，Hebei University of Technology，Tianjin 300130，China

E-mail Address:1427905290@qq.com

Abstract: due to the fast and high-precision edge detection requirements of tools in projection measurement, a fast sub-pixel edge detection algorithm for tool images is proposed. The algorithm first uses the Ostu threshold to perform the background segmentation of the tool image, then uses boundary tracking to obtain the boundary points of the image, and designs the corresponding correction operator to obtain the edge information of the tool; then according to the edge gray model, the Zernike moments are used to calculate the gradient direction of the edge points, and the two-dimensional model of the edge gray is converted into the one-dimensional model, and the sub-pixel edge of the tool image can be accurately obtained by calculating the gray distribution by spline interpolation. Experimental results show that this method has better noise immunity and faster detection speed..

1. Introduction

Nowadays, there is an increasing demand for product accuracy in the machining process. And manufacturing industries, and the geometric parameters of the tool have an important influence on the machining accuracy. Therefore, pre-adjustment and detection of the tool prior to machining are indispensable prerequisites for high-precision machining. At present, the measurement methods of the tool are mainly divided into manual measurement and automatic measurement. The manual measurement method is easy to be interfered, so the detection accuracy is low, and the microscopic condition of the tool cannot be detected in detail; However, the automatic measurement is the direct measurement of the geometric parameters of the tool through a special inspection device. The automatic tool detection equipment on the market today is mainly divided into mechanical and machine vision-based electronic video tool presetters. Among them, the electronic camera tool presetter is an optical image acquired by a CCD camera through a projection method, and then the computer records and processes the captured image. The geometric parameters of the tool are extracted through a digital image processing algorithm, which has a non-contact, high precision, and speed. Fast, high degree of automation, and it is the current mainstream tool parameter measurement method. In the tool vision inspection system based on the projection method, the edge information of the tool image is the basis for subsequent acquisition of tool geometry parameters, so the edge detection of the tool image is very important. The sub-pixel edge detection algorithm uses the software algorithm to process the image on the basis of the invariant system hardware conditions, and further subdivides the pixel unit. The processing methods include fitting method, interpolation method and moment method. The moment-based sub-pixel edge detection method uses the integral operator of the moment to locate the sub-pixel edge and is insensitive to noise, but the computational complexity of

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

the method of moments is high [1]; The fitting-based sub-pixel edge detection algorithm obtains high-accuracy edge positions by assuming a gray-scale model and fitting it, but it needs to be iteratively solved, and its efficiency cannot be guaranteed[2]. Although the sub-pixel edge detection algorithm has been perfected and developed, its computational complexity is high, and even iterative calculations are required in some specific algorithms, while the tool measurement system requires real-time detection of the tool, therefore, a fast sub-pixel edge detection algorithm is needed to locate the edge of the tool.

1.1. Otsu Threshold for Image Segmentation

The image is divided into a background and a tool projection by a gray-level threshold. The variance between classes represents the difference between the background and the tool projection. The threshold when the variance is maximized is the optimal threshold[3]. Assuming that the gray level of the image is L-level, wherein the number of pixels of the i-th level is N_i, the total number of pixels is:

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

The probability of the k-th pixel appearing in the image is:

$$P_i=\frac{N_i}{N} \tag{2}$$

Assuming that the target and background pixels are w_0 and w_1 respectively, and the segmentation threshold is 't', then the probability of occurrence of each class is:

$$\begin{cases} w_0=\sum_{i=0}^{t} P_i \\ w_1=\sum_{i=i+1}^{N-1} P_i = 1-w_0 \end{cases} \tag{3}$$

The average gray value of each class is:

$$\begin{cases} u_0 = \dfrac{u(t)}{w_0} \\ u_1 = \dfrac{u_T - u(t)}{1-w_0} \end{cases} \tag{4}$$

The formula for the variance between classes is:

$$\delta^2(k) = w_0(u_T-u_0)^2 + w_1(u_T-u_1)^2 \tag{5}$$

Therefore, the optimal threshold k* is:

$$k^* = \arg\max\left(\delta^2(k)\right) \tag{6}$$

1.2. Boundary Tracking

By using insects with the law, the termination condition was changed to boundary between the two en ds of the image, that is to complete the tracking [4]. After the above threshold value is used to binarize t he image, the pixel value of the tool projection image is 0, and the background pixel value is 1. The al gorithm steps are as follows:

(1) It takes the pixel in the upper left corner of the image as the starting point, and traverses the boundary point of the image counterclockwise. When the pixel value changes, the point before and after the pixel value change is 0 as the starting point of the contour, as shown in Figure 1(a). Get the current point 'a' of the profile.

(2) It searches the 8-pointer of the current point along the outline of the outer tangent of the contour starting from the pixel 'a'. When the pixel with the pixel value of 0 appears for the first time, record the point as the current point, as shown in Figure. 1(b). If the points meet this requirement, the point b closest to the pixel value 0 is selected as the current point, and the above process is repeated to obtain the next current point 'c', as shown in Figure. 1(c).

(3) Step 2 is repeated from the current point of the previous time. If the current point reaches the boun-dary of the image, stop the algorithm, as shown in Figure 1(d).

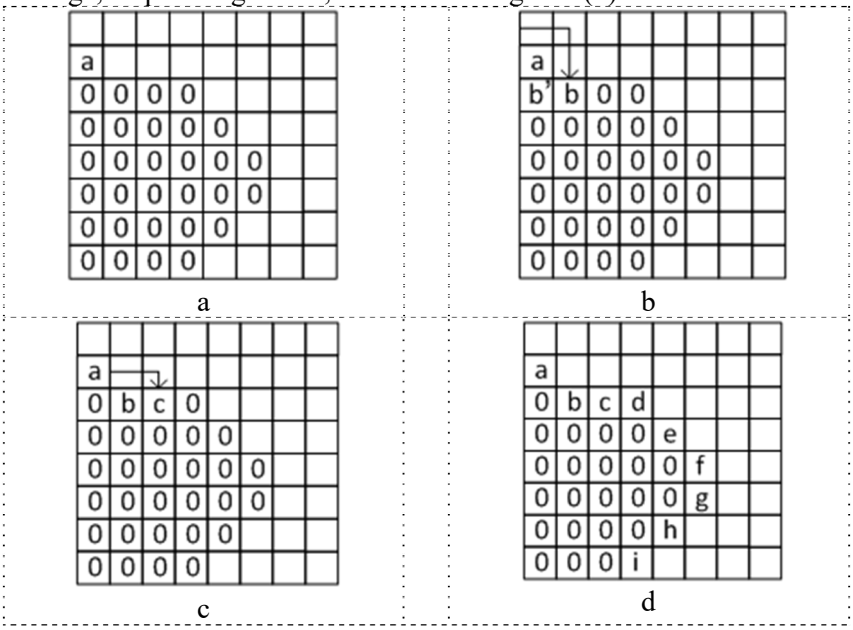


Figure 1 Boundary tracing

After the above tracking is completed, the boundary is the boundary of the edge projection, and the edge of the image is not conceptually equivalent to the boundary of the image. The edge refers to the place where the pixel value in the image mutates, and the boundary point is the interior of the object adjacent to the background point; the basic unit of digital image is pixel, so in the binary i-mage, the boundary point of the object is the pixel inside the object adjacent to the background poi-nt, and the position of the edge is between the boundary point and the adjacent background point.

Therefore, the coordinates of any boundary point c in Figure 1(d) are (x,y), and the coordinates of the corresponding edge point should be (x,y-0.5). In this paper, the boundary points are corrected according to the values of the pixels in the boundary point and its neighborhood, so as to further improve the positioning accuracy of the edge points. Assume that the position of the edge point is affected by the pixel value in the 3×3 neighborhood of the boundary point, and construct the boundary point correction operator:

$$G_x = \begin{bmatrix} f(x-1,y)0.15 & 0 & -f(x+1,y)0.15 \\ -0.8 & 0 & 0.8 \\ f(x-1,y)0.15 & 0 & -f(x+1,y)0.15 \end{bmatrix} \tag{7}$$

$$G_y = \begin{bmatrix} f(x,y-1)0.15 & -0.8 & f(x,y-1)0.15 \\ 0 & 0 & 0 \\ -f(x,y+1)0.15 & 0.8 & -f(x,y+1)0.15 \end{bmatrix} \tag{8}$$

For any point (x, y), the corrected position is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} G_x A \\ G_y A \end{bmatrix} \tag{9}$$

In formula (9), $G_x A$ and $G_y A$ are respectively the convolutions of the templates $G_x A$ and $G_y A$ at the boundary point (x,y).

The edge detection is performed on a standard circle image with a radius of 500. The difference between the position of the boundary point before and after the correction and the actual position is compared, and the difference between the obtained arc radius and the actual radius is fitted to measure the effectiveness of the algorithm.

Table 1 Edge point correction

Outline point	Max(x)	Min(x)	Max(y)	Min(y)	R
Before correction	2499	500	2499	500	999.03
After correction	2499.5	499.5	2499.5	499.5	999.91

According to the data in Table 1, the corrected edge points can correctly reflect the actual edge position.

1.3. Positioning Sub-pixel Edges Based on Zernike Moments

Through edge tracking and boundary correction, we have obtained edge points arranged in sequence. Although the boundary points are corrected but their accuracy cannot reach the sub-pixel level, we need to further determine the position of edge points. The threshold T is calculated by the Otsu algorithm, and a one-dimensional model is extracted from the edge gray two-dimensional model according to the gradient direction of the boundary point.

To convert a two-dimensional model into a one-dimensional model, edge parameters need to be determined, and three Zernike moments. Z_{00}, Z_{11} and Z_{20}, are required to determine edge parameters. The corresponding Zernike orthogonal complex functions are:

$$\begin{cases} V_{00} = 1 \\ V_{11} = x + jy \\ V_{20} = 2x^2 + 2y^2 - 1 \end{cases} \tag{10}$$

According to the rotation invariance theory of Zernike moment [7], the formula can be obtained:

$$\begin{cases} Z_{00} = Z'_{00} \\ Z_{11} = Z'_{11} \\ Z_{20} = Z'_{20} \end{cases} \tag{11}$$

Assuming that the rotated image is parallel to the horizontal direction, we can get:

$$\mathrm{Im}[Z'_{11}] = \sin\varphi \,\mathrm{Re}[Z_{11}] - \cos\varphi \,\mathrm{Im}[Z_{11}] = 0 \tag{12}$$

According to formula (10) ,we can get:

$$\varphi = \arctan\left[\frac{\mathrm{Im}[Z_{11}]}{\mathrm{Re}[Z_{11}]}\right] \tag{13}$$

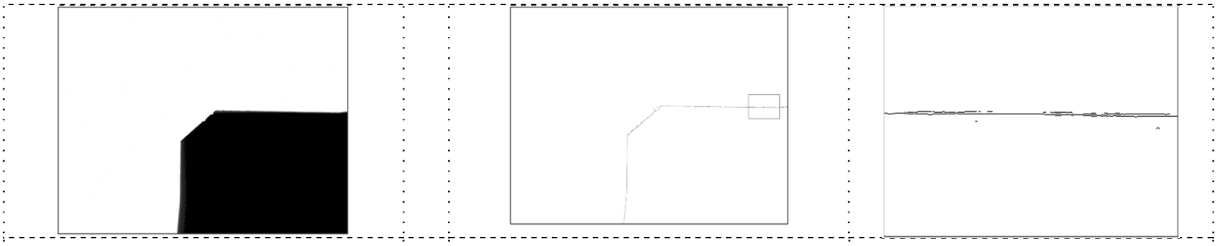
By converting the two-dimensional model of the edge into a one-dimensional model and then using cubic spline interpolation, the sub-pixel edge of the tool is obtained.

2. Experimental Results and Analysis

This paper presents fast edge extraction and sub-pixel edge location methods, and respectively compar ed with the classic edge detection algorithm - Canny operator edge detection algorithm, the classic sub -pixel edge location algorithm - Zernike moment based sub-pixel edge algorithm on operating efficien cy and resistance Noise comparison. The computer used in the experiment was I7-6800, 16G memory, and Window7 operating system. The main program was written in Labview language. Experiments ve rify that the tools used are the same position of the same tool to avoid experimental errors caused by in terference factors.

2.1 Edge Detection Algorithm Experiment

After adjusting each parameter optimally, the Canny operator algorithm uses a 7×7 Gaussian filter te mplate. The upper threshold is set to 110, the lower threshold is set to 40, and edge detection of the too l image shown in Figure 2(a) is performed using the Canny operator. And I get the best experimental results,and the experimental results are as follows:



(a) Tool actual image (b) Canny Edge (c) An enlarged view of the edge

Figure 2 Canny operator edge detection

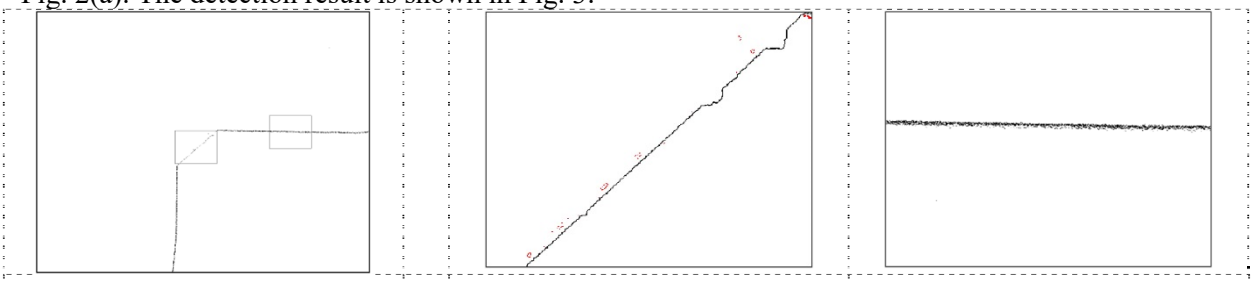
The detection time of Canny operator is about 300ms, and after adjustment of various parameters, it still can't realize all edge single responses. In addition, as can be seen from Figure 2(c), the algorithm still generates multiple responses at the fuzzy edge position. And it has a great impact on the real-time performance of the system.

2.2 Sub-pixel Edge Algorithm Experiment

In traditional sub-pixel edge algorithm based on Zernike moments, the Zernike template size is N*N. The Zernike moment method can be used to sub-pixel accuracy correction at any edge position (x_i, y_i) of the pixel level to obtain (x_i, y_i) :

$$\begin{bmatrix} \overline{x_i} \\ \overline{y_i} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \frac{N}{2} l \begin{bmatrix} \cos(\varphi) \\ \sin(\varphi) \end{bmatrix} \tag{14}$$

When N=7, gray-level difference threshold $K_{\min} = 40$, vertical distance threshold $L_{\max} = 0.14$, the experiment results are best. The sub-pixel edge detection is performed on the edge projection image in Fig. 2(a). The detection result is shown in Fig. 3:



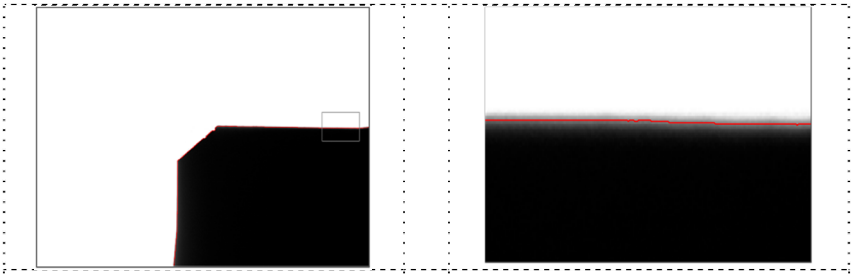
(a) Subpixel edge (b) Edge local magnification 1 (c) Local edge magnification 2

Figure 3 tool image detection results

Figure 3 (b) is the sub-pixel edge detection result, which takes about 11s; if the pixel edge detection operator, such as the Canny operator after the coarse positioning of the edge of the image after the sub-pixel edge detection, the detection time can be reduced To 500ms, still can not meet the real-time requirements of the system. As shown in Fig. 3(c), the edge detected by the sub-pixel edge is not a single edge. The reason is that the tool is a complex space object. When the image is captured, not all the edges are on the positive focal plane, and some of the edges are defocused. Ambiguity, but the above algorithm does not distinguish, it will produce multiple responses to the fuzzy edge.

2.3 Experimental Research on Fast Edge Extraction Algorithm Based on Zernike Moments

The fast edge extraction algorithm proposed in this paper uses Ostu threshold segmentation images an d boundary tracking and correction to obtain image edges:



(a) Tool actual image (b) Enlarged view of the edge

Figure 4 Tool Image and Edge

Fig. 4(a) shows the edge detection result of the actual image, and Fig. 4(b) shows the partial fuzzy edge. It can be seen that the method still has a single edge response to the blurred contour; and the complete boundary is obtained by tracking the boundary points without the need for the entire width. The image is convoluted, which effectively improves the efficiency of the algorithm. The time required for the Canny operator is approximately 300 ms, while the method requires only about 20 ms.

The sub-pixel edge location algorithm proposed in this paper can obtain a single image edge after edge extraction, and solves the problem of multiple responses generated by the fuzzy edge; and taking the edge point as the center, the edge gray can be obtained along the normal direction of the edge point. One-dimensional distribution.

Table 2 Edge gray distribution

x_i	-4	-3	-2	-1	0	1	2	3	4
y_i	194	178	150	135	120	97	85	76	64

The cubic spline interpolation function interpolates the data in Table 2 to obtain a continuous curve. The interpolation results are shown in Figure 5.

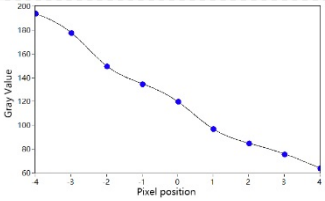


Figure 5 Edge gray distribution and interpolation curve

A one-dimensional model is extracted from the edge gray two-dimensional model according to the gradient direction of the boundary point.

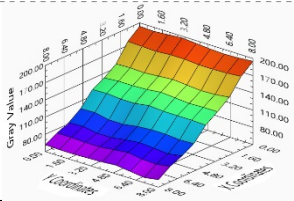


Figure 6 Edge gray two-dimensional model

Fig. 6 is a two-dimensional distribution model of the actual edge grayscale. This model can be regarded as a surface formed by the projection of the edge grayscale one-dimensional model in the vertical direction of the gradient. The edge grayscale can be obtained by the expression (13). The two-dimensional model was transformed into a one-dimensional model for processing, and the sub-pixel edge position was obtained by calculating the gray distribution by the spline difference.

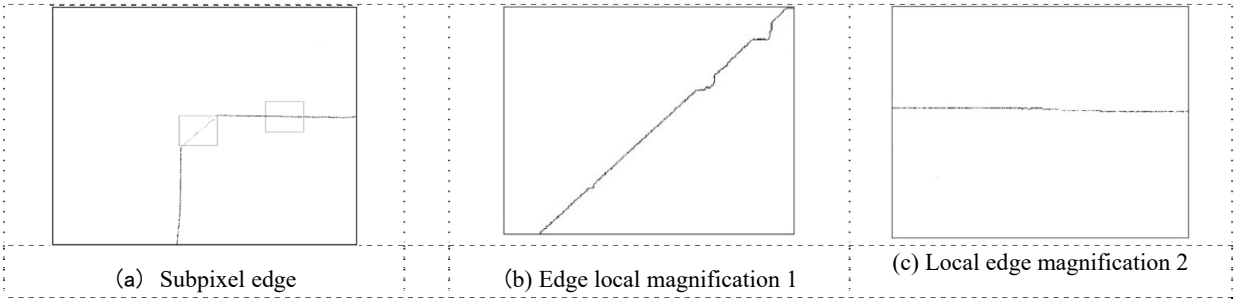


Figure 7 tool image detection results

On the basis of obtaining the edge points, the sub-pixel edge position is shown in Fig. 3(a). As shown in Fig. 7(b), the problem of multiple responses generated by the fuzzy edge is better resolved and takes about 100ms. .m

Table 3 The length of time for each algorithm

Algorithm	Canny Algorithm	Sub-pixel Edge Algorithm	Fast Edge Positioning Algorithm Based on Zernike Moments
Time Consuming	300ms	500ms	130ms

3. Experimental Results and Analysis

The tool edge detection based on Zernike moment algorithm proposed in this paper can be reduced to 130ms in total time. The accuracy can achieve a single response of the fuzzy edge position, and the speed can meet the real-time detection of the system. The algorithm proposed in this paper can be used in the real-time system of tool detection. It has been well applied and laid a good foundation for the subsequent measurement of tool parameters. Therefore, the method proposed in this paper for rapid edge detection is reasonable.

References

[1] Kaur A, Singh C. Sub-pixel edge detection using seudo Zernike moment[J]. *International Journa of Signal Processing Image Processing & Pattern Recognition*, 2011, 4.
[2] Hermosilla T, Be rmejo E, Balaguer A, et al. Non-linear fourth-order image interpolation for subpixel edge detection and localization[J]. *Image & Vision Computing*, 2008, 26(9):1240-1248.
[3] Zhou Yang. Research on Visual Inspection Technology of Image Type Knife Adjuster[D]. Jiangnan University,2011
[4] Liu Lishuang. Research on Electronic Camera Presetting Measuring Instrument[D]. Tianjin University，2006
[5] Zhou Yazhou. Sub-pixel edge detection method based on cubic spline interpolation[D]. Changchun Industrial College,2014
[6] Tai Guohua,Yi Xing,Li Yiran ,et al. Tool wear automatic detection and detection system[J]. Optical Precision Engineering, 2014, 22(12):3332-3341.
[7] Guo Xiao. Ultrasonic Image Segmentation Based on Zernike Moments and Level Sets[D]. Southeast University,2015
[8] Kou Xingyuan. Research on Intelligent Machine Tool Detection Technology Based on Image Processing[D]. Beijing University of Posts and Telecommunications，2012
[9] Chen Chen. Tool image contour feature detection[D]. XiAn Industrial University，2013