Gear Recognition and Positioning Based on Image Processing

Kai Wang [a], Hongfei Li, Jianyong Li

School of Shandong University of Science and Technology, Shandong 266590, China.

[a] 1838975646@qq.com

Abstract

Aiming at the problem of shape recognition and position recognition in gear machining, the problem is mathematically modeled. Through the Canny edge processing algorithm and the improved Hough transform algorithm, the gear geometry and coordinates are quickly identified in the image data. The position of the gears is identified. By identifying the circular outline in the image, the center coordinates of the part are calculated and the algorithm is improved. The simulation is programmed in Python and the simulation platform is pycharm. Simulation experiments show that the algorithm can quickly and accurately identify the position of each part.

Keywords

Canny algorithm, Hough transform, Python.

1. Introduction

With the rapid development of computer technology and optoelectronic technology, visual inspection technology has been widely used[1]. Based on modern optics, it integrates science and technology such as optical electronics, computer graphics, information processing, and computer vision, and uses images as a means or carrier for detecting and transmitting information[2]. Due to the high precision and speed of computer vision detection, the detection results are relatively stable, and it has the ability of online detection, real-time analysis and real-time control. Therefore, it has made important contributions to improving production efficiency, ensuring product quality and improving product accuracy.

At present, the traditional part recognition methods mainly include Fourier feature, SIFT[3] (Scale-invariant feature transform), support vector machine[4] and algorithm deformation, but the effect is not good. Applying the above algorithm to the identification of gears can not meet the requirements of less time consumption and high robustness. Therefore, according to the characteristics of gears, a method for quickly identifying gears is proposed, and the position of gears is realized.

2. Image Edge Processing Based on Canny Algorithm

2.1 Canny Algorithm Introduction

Canny edge detection is a technique for extracting useful structural information from different visual objects and greatly reducing the amount of data to be processed. It has been widely used in various computer vision systems. General criteria for edge detection include: (1) Detecting edges with a low error rate means that you need to capture as many edges as possible in the image as accurately as possible. (2) The detected edge should be accurately positioned at the center of the real edge. (3) The given edge in the image should only be marked once, and if possible, the noise of the image should not produce false edges.

To meet these requirements, Canny used the variational method. The optimal function in the Canny detector is described using the sum of four exponential terms, which can be approximated by the first derivative of the Gaussian function.

Among the commonly used edge detection methods, the Canny edge detection algorithm is one of the methods that is strictly defined and can provide good and reliable detection. It has become one of the most popular algorithms for edge detection because it has the advantages of three standards for edge detection and simple implementation.

2.2 Implementation of Canny Edge Detection Algorithm

In order to minimize the effect of noise on the edge detection results, noise must be filtered out to prevent false detection caused by noise. To smooth the image, a Gaussian filter is used to convolve with the image, which smoothes the image to reduce the apparent noise effects on the edge detector. The generating equation of the Gaussian filter kernel is given by:

$$H_{ij} = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left(i-(k+1)\right)^2+\left(j-(k+1)\right)^2}{2\sigma^2}\right);1\le i,\mathrm{j}\le(2k+1) \tag{1}$$

The edges in the image can point in all directions, so the Canny algorithm uses four operators to detect horizontal, vertical, and diagonal edges in the image. The edge detection operator (such as Roberts, Prewitt, Sobel, etc.) returns the first derivative value of the horizontal G_x and the vertical direction G_y, thereby determining the gradient G and the direction θ of the pixel.

$$G = \sqrt{G_x^2 + G_y^2} \tag{2}$$

$$\theta = \arctan\left(G_y / G_x\right) \tag{3}$$

Non-maximum suppression is an edge sparse technique, and the effect of non-maximum suppression is on the "thin" side. After the gradient calculation of the image, the edge extracted based only on the gradient value is still very blurred. For standard 3, there is and should only have an accurate response to the edge. Instead of maximal suppression, it can help suppress all gradient values outside the local maximum to zero. The algorithm for non-maximum suppression of each pixel in the gradient image is:(1) Compare the gradient intensity of the current pixel with two pixels along the positive and negative gradient directions.(2) If the gradient intensity of the current pixel is the largest compared to the other two pixels, the pixel point remains as the edge point, otherwise the pixel point will be suppressed.

After applying non-maximum suppression, the remaining pixels can more accurately represent the actual edges in the image. However, there are still some edge pixels due to noise and color variations. In order to solve these spurious responses, it is necessary to filter the edge pixels with weak gradient values and preserve the edge pixels with high gradient values, which can be achieved by selecting the high and low thresholds. If the gradient value of the edge pixel is higher than the high threshold, it is marked as a strong edge pixel; if the gradient value of the edge pixel is less than the high threshold and greater than the low threshold, it is marked as a weak edge pixel; if the gradient value of the edge pixel is smaller than A low threshold will be suppressed. The choice of threshold depends on the content of a given input image.

Pixels that are divided into strong edges have been identified as edges because they are extracted from the real edges in the image. However, for weak edge pixels, there will be some debate as these pixels can be extracted from real edges or due to noise or color changes. In order to obtain accurate results, weak edges caused by the latter should be suppressed. Typically, weak edge pixels caused by real edges will be connected to strong edge pixels, while noise responses are not connected. In order to track the edge connections, by looking at the weak edge pixels and their 8 neighborhood pixels, as long as one of them is a strong edge pixel, the weak edge point can remain as a true edge.

3. Circular Recognition Based on Hough Transform

3.1 Implementation of Hough Transform

The Hough transform mainly uses the global features of the part image to directly detect the outline of the part image. In other words, it is an effective method to connect the edge pixels together to form a closed boundary area. The Hough transform was originally applied only to the detection of straight lines. Due to its excellent characteristics, it has been gradually applied to other geometric shapes such as circles and rectangles.

As shown in Figure 1. To ensure the correctness of the center positioning, eliminate the outlier noise and overlap problems, use equation (4) to remove the outer points in the detection:

$$\left|P_1P_m^2 - P_1P_{n+1}^2 - P_{n+1}P_m^2\right| \le \varepsilon_1 \tag{4}$$

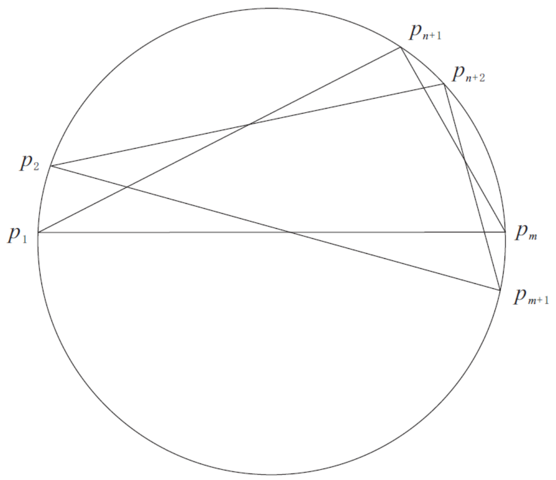

Fig. 1 Principle of circle detection algorithm

The coordinates of the center of the circular target can be calculated by the following formula:

$$\begin{cases} x_0 = \left(p_{1x} + p_{mx}\right)/2 \\ y_0 = \left(p_{1y} + p_{my}\right)/2 \end{cases} \tag{5}$$

3.2 Improvement of Hough Transform

First, sort the boundary points. Starting from the first boundary point, search for the boundary points connected to them in turn until a boundary search is completed. If it is a complete circular boundary, start with any point and search for each connected. Point, store the sorted boundary points in the array; then divide the boundary pixel into three segments, starting from the first point of each segment, selecting one point A(x_a , y_a),B(x_b , y_b),C(x_c, y_c) for each segment to get three points, through Three points solve a circle and radius. Looking for a circle and a radius are shown in Figure 2.

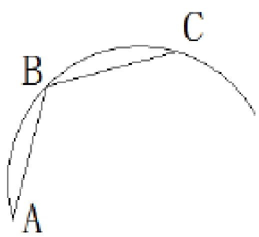

Fig. 2 Find a schematic diagram of the center and radius

Traverse each boundary, get a series of center and radius values, store the center and radius. Determine the difference between all the centers and the radius, according to the set threshold, remove the boundary points that deviate significantly from the true radius and the center value. Finally, the ratio of the number of points on the circumference to the number of the whole boundary points is judged. If the ratio is large, it means that most of the points are on the same circumference. Otherwise, the boundary cannot form a circle. In practical applications, in order to improve the center of the circle And the accuracy of the radius, the pixel coordinates generally take the midpoint of the pixel. At the same time, in order to ensure that different boundaries may be on the same circumference, first find each possible circle, then merge the same boundary according to a certain threshold, so In the boundary detection, if a circle is detected as a few arcs, the same applies. Finally, the center and radius are averaged to obtain the center and radius.

4. Simulation

Simulate experiments with Python 3, The raw data is shown in Figure 3.

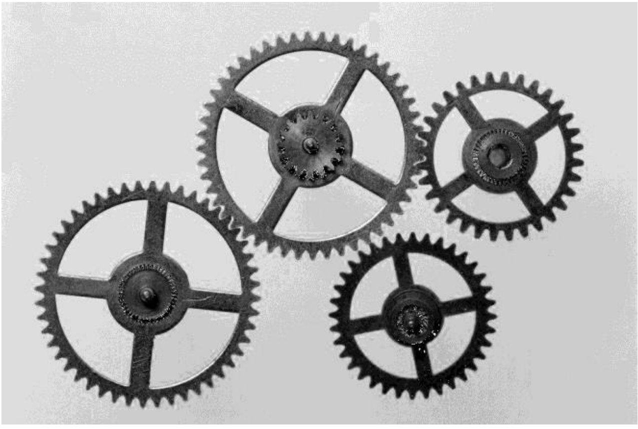

Fig. 3 Raw data image

The image before and after Canny edge processing on the image is shown in Figure 4.

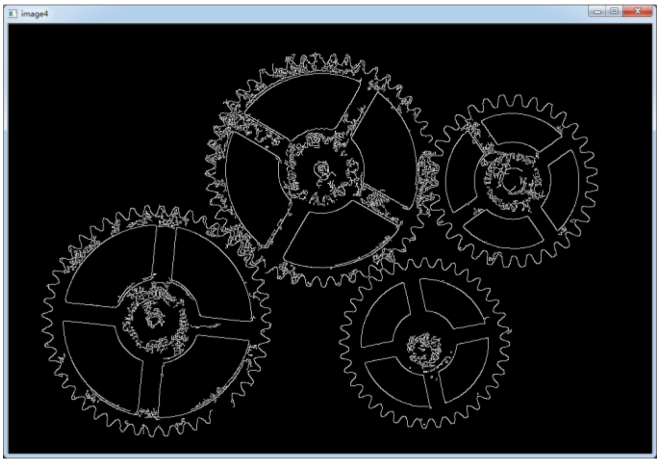

Fig. 4 Canny edge processing

The image center of the part image is determined as shown in Figure 5.

Three sets of experiments were carried out, and the experimental results show that the proposed algorithm can complete the identification of the center position of the part. Under the premise of maintaining the same accuracy as the standard Hough transform, the efficiency of circular detection is relatively improved. Table 1 shows the time consumed by the Hough transform before and after the simulation.

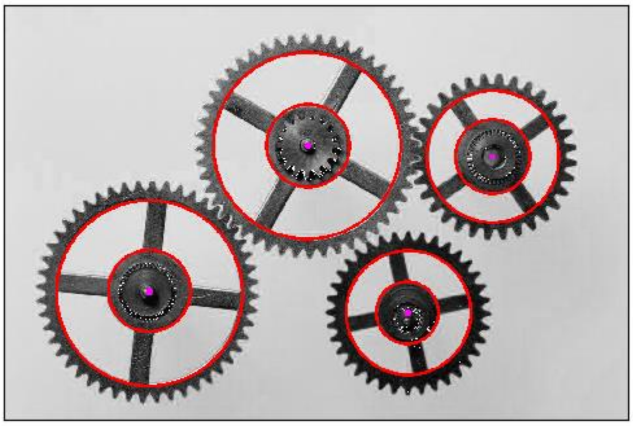

Fig. 5 Center recognition effect map

Table 1. Improved time-consuming comparison before and after operation

	Hough transform	After operation
DATA1	0.45583 s	0.06574 s
DATA2	0.97566 s	0.13562 s
DATA3	1.34226 s	0.94278 s

5. Conclusion

This paper first proposes an algorithm based on the combination of canny algorithm and Hough transform. By identifying the circular contour in the image, the center coordinates of the gear are calculated, and then the algorithm is improved. The simulation proves that the algorithm can accurately identify the algorithm. Gear parts, and the time complexity of the algorithm is greatly reduced.

References

[1] Li Shaohui. Visual inspection of the position and size of mechanical parts [D]. Tianjin University, 2008. (In Chinese).
[2] Chen Xiangwei. Research on Key Technologies of Computer Vision Inspection for Mechanical Parts [D]. Jilin University, 2005. (In Chinese).
[3] Schölkopf B. Choosing multiple parameters for Support Vector Machines[J]. 2008.
[4] Zhai Kaihua, Su Zhenwei, Huang Mingfei. Application of Support Vector Machine in Mechanical Parts Identification[J]. Electronic Technology Application, 2008, 34(11): 108-110.