



Automated Alignment of U-Notch in Iron Caps Using Machine Vision: A System for Suspended Insulators

Kui Kang¹ , Huiyu Zhang^{2*} , Yu Wang¹

¹ School of Machine and Engineering, Xihua University, 610039 Chengdu, China

² School of Intelligent Manufacturing, Yibin Vocational and Technical College, 644003 Yibin, China

* Correspondence: Huiyu Zhang (huaiyuzhang812@foxmail.com)

Received: 11-29-2023

Revised: 02-27-2024

Accepted: 03-05-2024

Citation: K. Kang, H. Y. Zhang, and Y. Wang, "Automated alignment of U-notch in iron caps using machine vision: A system for suspended insulators," *Precis. Mech. Digit. Fabr.*, vol. 1, no. 1, pp. 1–10, 2024. <https://doi.org/10.56578/pmdf010101>.



© 2024 by the author(s). Published by Acadlore Publishing Services Limited, Hong Kong. This article is available for free download and can be reused and cited, provided that the original published version is credited, under the CC BY 4.0 license.

Abstract: In the automated production line for suspended insulators, precise alignment of the U-shaped notch in iron caps is crucial for effective gluing. This study introduces a system based on machine vision that automates the alignment process. The system initially preprocesses the images of iron caps to segment the U-shaped contour. It utilizes the method of quadratic maximum contour connectivity domain to accurately identify the target U-shaped region. The alignment process involves calculating the coordinates of the largest external rectangle's longest edge and the external circle's center point. These coordinates are instrumental in determining the necessary rotation angle for proper notch alignment. The fixture then adjusts the iron cap based on this calculated angle, ensuring precise alignment. Experimental validations of this system have demonstrated a notch alignment error within 0.5 degrees with 96.51% accuracy and an error within 1 degree with 100% accuracy. The algorithm's execution time is a swift 0.034 seconds. Both the error margins and operational speed satisfy the stringent requirements of the automatic production line.

Keywords: Machine vision; Suspended insulator; Iron cap; Image processing

1 Introduction

Suspension insulators are an important part of high-voltage transmission lines. It ensures connections and insulation between high-potential conductors and low-potential towers. Suspended insulators usually consist of glass discs, steel hinges, and iron caps. The working scenario and composition of a suspended insulator are shown in Figure 1.

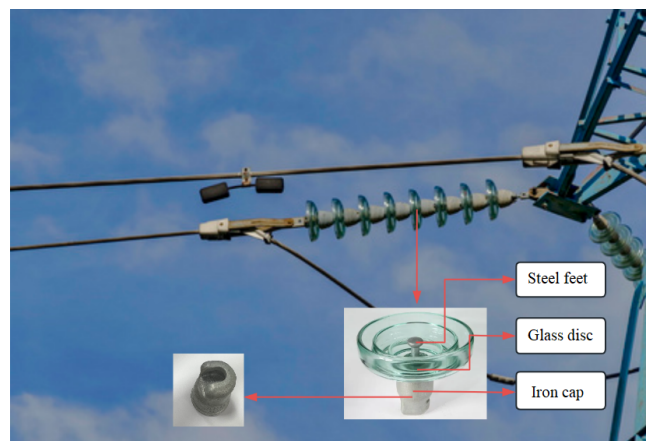


Figure 1. Working scenario and composition of suspended insulators

At present, the traditional production of suspended insulators is carried out manually, including manual alignment assembly, gluing, and stringing. Among them, in the stringing process, the steel hinge needs to be connected to the iron notch, but the direction of the notch is not fixed, and the notch needs to be aligned in order to carry out the stringing. To achieve automation in the production of suspended insulators, the alignment of the iron notch must be absent. In traditional automated production measurement, the typical method is to use calipers, goniometers, or angle gauges to measure a certain parameter of the workpiece several times, then the average value [1]. With the rapid development of science and technology and the popularity of machine vision technology, there are more and more algorithms for image detection and recognition, such as the non-contact rotation angle measurement method based on a monocular camera [2], image direction determination based on feature extraction [3], dynamic angle measurement method based on machine vision [4], 3D vertebra rotation angle automatic measurement direction recognition based on deep learning [5]. The traditional machine vision algorithm based on image processing adopted in this paper has the advantages of non-contact [6], high resolution, and robustness to free motion and rotational fluctuations, which makes it easy to get the angle between the U-shaped notch of the iron cold and the actual alignment [7], and then achieve the notch of the iron cold to realize automatic alignment through the rotation of the iron cold driven by the motor.

2 Image Pre-processing

The visual image of the calibrated area is captured by an industrial camera [8], but in the process of industrial image acquisition, due to the field environment, lighting changes, background interference, camera shake, and other factors will affect the image quality. To reduce the interference information in the image and enhance the target feature information, the original image needs to be preprocessed. The purpose of image preprocessing is to eliminate the noise that exists in the acquisition process of the iron cap image and to improve the image characteristics of the target area, thus improving the robustness of the algorithm. In this paper, the flow chart of the iron cap preprocessing is shown in Figure 2.

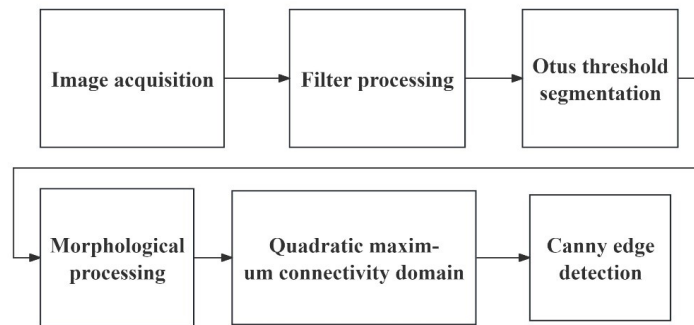


Figure 2. Flowchart of image pre-processing

2.1 Filter Processing

As the filtering process's main goal is to denoise or smooth an image or signal to improve image quality, extract features, reduce noise interference, or provide better input for subsequent image processing algorithms. Images or signals are often subject to various types of noise interference during acquisition or transmission, such as Gaussian noise, pretzel noise, and so on. Filtering processes can reduce the noise level in an image by smoothing or suppressing the noise, improving the quality and clarity of the image, and thus increasing the robustness of the algorithm.

2.1.1 Mean filtering

Mean filtering is also known as linear filtering, where the gray value of the central pixel point is equal to the average value of some surrounding critical domain with the formula [9]:

$$g(x, y) = \frac{1}{N} \sum_{(m,n) \in S} f(m, n) \quad (1)$$

where, S is the critical domain of pixel point (x, y) and N is the number of pixel points in S . Pixel points on the boundary can be left unfiltered and their pixel values retained.

Mean value filtering is fast, but it has its own defects: It can't eliminate the noise, and it will cause the blurring of the details of the image at the same time of denoising. But the details of the U-shaped profile of the iron cap and the outer horseshoe profile are especially important, so this system does not choose to use mean value filtering.

2.1.2 Median filtering

Median filtering is a common image processing technique used to remove noise from an image. It is based on the principle that the neighboring pixels around each pixel are sorted according to the size of the gray value, and then the median value is taken as the new value of the pixel, thus eliminating the effect of noise. Median filtering uses a non-linear method for processing, which ensures that the details of the image are not lost while suppressing the noise, and it is one of the more widely used image processing methods [10]. The number of pixels in the template of median filtering is generally odd, and the target pixel value is equal to the middle value of all pixels in its proximity domain [11] with the formula:

$$g(x, y) = med\{f(m, n), (m, n) \in S\} \quad (2)$$

To select the filtering method with the best effect, this paper adds 3% pretzel noise to the image after gray scaling, and then use the three filtering methods mentioned above to process them separately. The filtering is selected on a 5×5 cross-shaped template. The filtering results are shown in Figure 3, in which picture a is the original image; picture b is the image after adding 3% salt and pepper noise; picture c is the effect of mean filter, which can be seen that the mean filter fails to eliminate the noise but makes the image blurred and even loses the details; picture d is the effect of median filter, which can be seen that the median filter not only completes the noise reduction but also retains the details of the image; Comparing the two different filtering processes, this paper chooses the median filter for filtering.

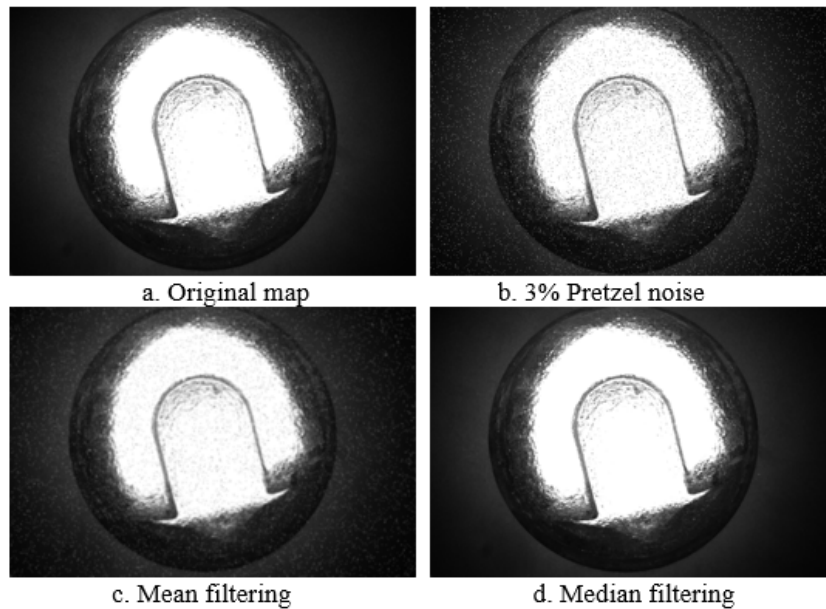


Figure 3. Filter processing

2.2 Otus Threshold Segmentation

After pre-processing, the image quality is improved and the noise is removed. Since this paper only focuses on the topmost U-shaped contour and the contour groove of the iron bubble, the extracted image is binarized to distinguish the pixels on the edges from the other pixels. The principle of binarization is to obtain a threshold value by some algorithm, and then the pixel points within the image are classified into foreground and background categories based on the threshold value.

When fixed thresholding is used under complex illumination conditions, it fails to extract feature points or extracts only a small number of feature points. In this paper, the Otsu thresholding method is used [12], which is an adaptive state-value segmentation algorithm based on the histogram of an image that is able to classify the image into two categories, one for the target and the other for the background. The Otsu thresholding method is suitable for the fields of binarization and image segmentation, where the image is enhanced by applying Gaussian filtering for noise reduction, truncation of adaptive luminance adjustments, and unsharpening of the masking operation. Then, the enhanced image is segmented into subregions of specified size, and the truncated Otsu method is used to calculate the adaptive threshold for each subregion [13]. It has higher classification accuracy and better adaptability than other read-value methods. Advantages of the Otsu Thresholding Method.

(1) The Otsu queuing method is an adaptive threshold selection method without human intervention that can automatically adapt to the complexity of the image and the change in grayscale distribution.

(2) The Otsu queuing method has better segmentation results, higher classification accuracy, and better adaptability than the general gray level-based threshold selection method. The segmentation results of this algorithm are obvious, and the important structures are clear.

(3) The Otsu queuing method is simple to compute, the algorithm complexity is low, and it can be implemented quickly.

The threshold segmentation effect is shown in Figure 4.

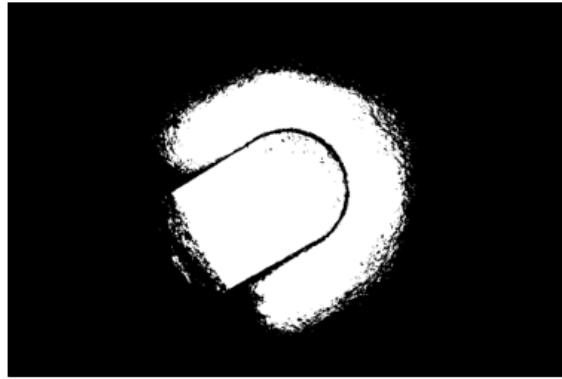


Figure 4. Otsu threshold segmentation

2.3 Morphological Processing

Morphological processing is a commonly used technique in image processing for performing morphological operations on images such as erosion, expansion, open operation, closed operation, etc., which can be used for tasks such as denoising, segmentation, edge detection, etc. of images [14]. In this paper, since the found image is not good enough to distinguish the U-shaped notch completely, erosion operations in morphological processing are used to reduce or eliminate the boundary of the object in the image.

The principle of the erosion operation is to slide a structural element (also known as a kernel or convolution kernel) in the image, compare the center of the kernel with the position of each pixel in the image, and if all the elements of the kernel match with pixels at the corresponding positions in the image (i.e., they are all foreground pixels), the center pixel is set to be a foreground pixel, otherwise it is set to be a background pixel [15]. The erosion operation is actually achieved by gradually eroding away the foreground pixels in the image, and the size and shape of the structure elements affect the erosion effect. The larger the structuring element is, the more obvious the effect of erosion is, and the object will become smaller; while the smaller the structuring element is, the slighter the effect of erosion is, so it can be used for removing small noise points, separating the connections between objects, etc. In this paper, 5×5 structural elements are used to separate the U-shaped contour from the peripheral contour to get the binarized image that is not connected to the periphery at all, which improves the accuracy of contour finding, and the processing results are shown in Figure 5.



Figure 5. Morphological processing

2.4 Secondary Maximum Connectivity Domain

In image processing, a connected field is the set of pixels in an image that have the same pixel value and are connected to each other. The connected field marking operation for a binary image is to mark the connected field from a dot matrix image consisting of white pixels (usually represented by “1” for binary images and “255” for grayscale images) and black pixels (usually represented by “0” for grayscale images) that are adjacent to each other with pixel values. In a dot-matrix image composed of two pixels (usually represented by “1” for binary images and “255” for grayscale images) and black pixels (usually represented by “0”), a set of pixels adjacent to each other with pixel values of “1” or “255” is extracted, and a numerical marking is filled in for the different connectivity domains in the image in a varying degree, and the number of connectivity domains is counted [16]. The number of connected domains is counted. The image is usually traversed at the pixel level, starting from a single pixel and recursively or iteratively finding all pixels with the same pixel value that are connected to that pixel and grouping them into the same connected domain.

After morphological operations, the required part of the image has been segmented to retain the target feature area, specifically the U-shaped contour in the center. Through extensive experimentation, it was discovered that the contour area of the target feature is smaller than that of the peripheral contour yet is the largest when excluding the peripheral contour. Leveraging this key insight, this study introduces a method to find the maximal contour connectivity domain for the second time, calculating the area of the connected domain. Initially, the largest peripheral closed contour is identified and used to generate a mask image, which covers the peripheral contours in black, isolating only the target feature area. However, due to the potential presence of incomplete noise filtering, when seeking the maximal closed contour connected domain a second time, a mask is not generated. Instead, only the largest contour image is retained. This approach successfully extracts and segments the U-shaped feature area. The process of U-shaped contour extraction and segmentation is illustrated in Figure 6.

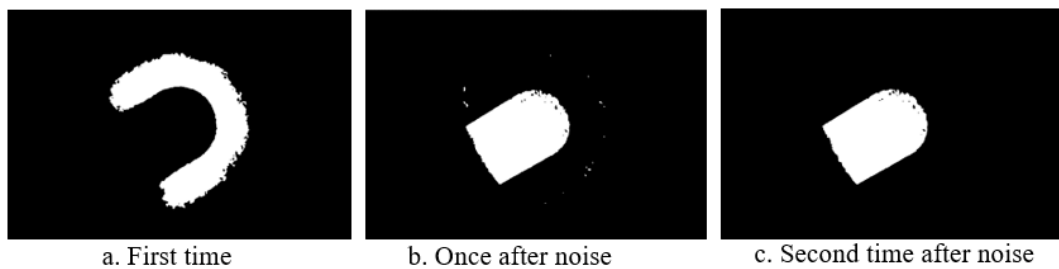


Figure 6. U-profile



Figure 7. Canny edge detection

2.5 Canny Edge Detection

Among many computer vision algorithms, the Canny edge detection algorithm performs significantly better than existing edge detection techniques [17]. When using edge detection for image edge recognition, the choice of edge detection algorithm affects the results in order to obtain a clear image [18]. In order to extract the outer contour edge features of the image, this paper first processed the target image with hole filling, which means that under binarization, there are “white dots” or “black dots” in the image, affecting our calculation of the area inside the contour [19]. In OpenCV, we can use contour filling and the flooding algorithm to achieve hole filling and the flooding algorithm to fill the hole area, so as to achieve the filling of the holes in the image.

Flood fill is an algorithm for filling connected regions in an image [20]. In OpenCV, the flood fill is implemented using the `cv2.floodFill` function [21]. The algorithm starts from a specified seed point and continues to expand the fill in all four directions until it encounters a boundary or fails to satisfy the fill conditions, thus filling regions with

the same pixel values. The flood algorithm is usually used to fill closed regions, remove noise, select regions, and perform other image processing tasks.

After filling, the Canny edge detection algorithm is further used to extract the edge features of the image. Canny edge detection is less affected by image noise and the detected edges are more continuous, with clear edge lines and high detection accuracy [19]. And calculate the contour area; the calculated area within a certain range is to meet the desired U-shaped target feature contour and thus improve the accuracy of image segmentation. The effect is shown in Figure 7.

3 Rotation Angle Measurement

In industrial environments, workpieces are mostly cluttered, and some contour detection algorithms, which often get the coordinate information of the object, find it difficult to get the accurate angle between the workpiece and the actual alignment. Since the U-shaped contour in Figure 7 is composed of a rectangle and a semicircle in proportion, using this information, this paper proposes to establish the minimum outer rectangle and outer circle of the target iron counterfeit, and by extracting the specific straight line as well as the coordinate information of the key point, through the measurement based on the machine vision with the specific straight line as the reference target, the angle on the image can be calculated [22], and the angle between the target workpiece and the actual alignment is acquired information.

3.1 Minimum Outer Rectangle

A minimum outer rectangle is a rectangle of minimum size around a given shape and is commonly used to describe and enclose objects or contours in a found image. For polygonal objects, approximating their shape with their outer rectangles is a common method in the fields of GIS and graphics [23]. After obtaining the information of the boundary pixels of each closed image region, it is easy to filter the pixels to get two special points $P(x_{min}, y_{min})$ and $P(x_{max}, y_{max})$, and then the outer rectangle of the diagonal line connecting these two points is the minimum outer rectangle of the image region.

This article uses the OpenCV library inside the function for `cv2.minAreaRect(cnt)` [21], `cnt` is an array or vector of point sets (which holds the coordinates of the points), and there are an indefinite number of elements in this point set. The rectangle returned by the function is usually a rotated rectangle, which can be described by a center point, width, height, and rotation angle, and the coordinates of the four vertices of the rectangle are returned in the form of coordinates. The coordinates of the four vertices returned are at the bottom point for the first point, which is `box[0]`, and clockwise around the second, third, and fourth points. The relationship between the four vertices is shown in Figure 8.

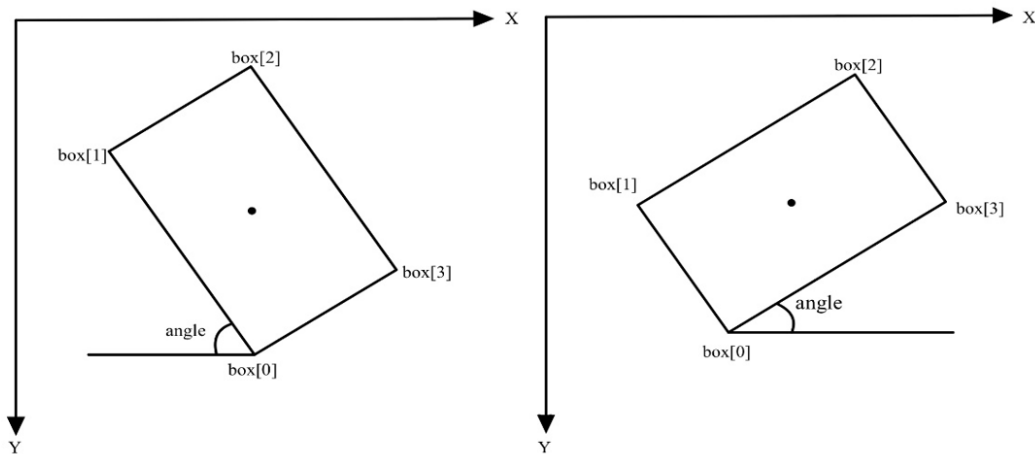


Figure 8. External rectangle vertex relationship

As the position of the four vertices is not fixed, combined with the characteristics of the coordinates of the four vertices of the minimum outer rectangle, through the `box[0]`, `box[1]` and `box[2]` three points to find the length of the `box[0]` and `box[1]`, `box[1]` and `box[2]`, resulting in the length of the rectangle adjacent to the length of the two sides of the distance 1 and distance 2, through the length of the length of the comparison can be determined the long side and short side, can be fixed to get the long side (AB) coordinates, the formula is:

$$\begin{cases} distance\ 1 > distance\ 2 & \rightarrow A = box[0], B = box[1], C = box[2], D = box[3] \\ distance\ 1 < distance\ 2 & \rightarrow A = box[1], B = box[2], C = box[3], D = box[0] \end{cases} \quad (3)$$

$A(x_1, y_1)$ and $B(x_2, y_2)$ in the formula are the coordinates of the two points on the long side, and then use the coordinates of the two points to find the slope k of the long side and the angle θ between the long side and the horizontal according to the formula (4) in turn, and the angle of a in Figure 8 is a positive number, and the angle of b is a negative number.

$$k = \frac{(y_1 - y_2)}{(x_1 - x_2)} \tag{4}$$

$$\theta = \arctan(k)$$

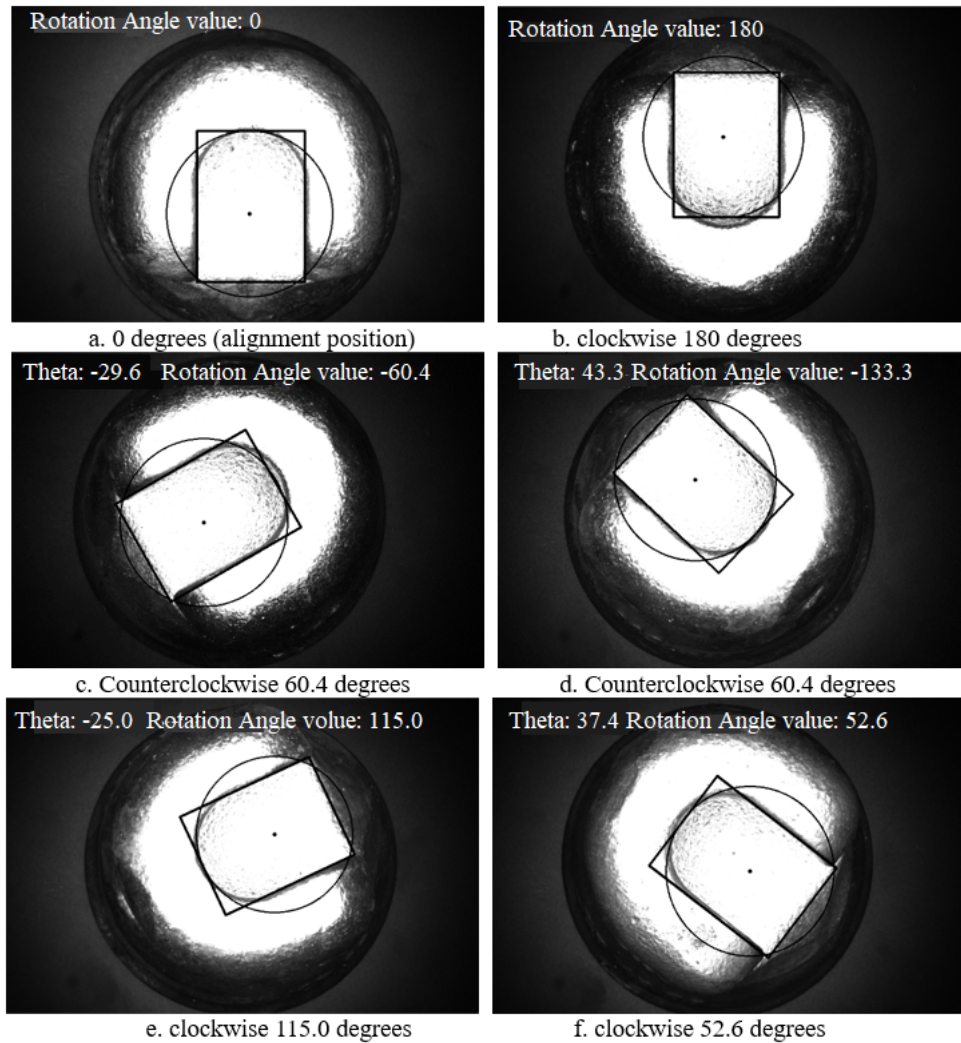


Figure 9. Rotation angle

3.2 Minimum External Circle

A least outer circle is the smallest circle that can enclose a given shape, described by a center coordinate and a radius. In image processing, the minimum enclosing circle is often used to describe and analyze the shape and position of an object [24]. In OpenCV, the function `cv2.minEnclosingCircle()` can be used to compute the minimum enclosing circle [21], which is usually returned as a tuple (center, radius), where center is the coordinates of the center of the circle and radius is the radius.

Since the notch direction of the iron cap has a rotation angle of 360 degrees in the plane, to get the direction of the notch, we also need to judge the external rectangle to determine which side of the notch is in the two short edges. Based on the geometric relationship of the semicircular arc in the U-shaped profile of the iron cap, we can use the OpenCV library `cv2.minEnclosingCircle()` minimum external circle function to get the center of the circle

coordinates (x, y) . Then find the distance from the center of the circle coordinates to the two short sides (AD and BC) with the following formula:

$$\frac{|(y_2 - y_1)x_3 - (x_2 - x_1)y_3 + y_1y_2 - x_1y_2|}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \quad (5)$$

You can find out the distances 3 and 4 from the center point to the two short edges, compare the size of the distance, and the short side of the rectangle with the short distance is the position where the gap is located. When combined with the characteristics of the image, the distance from the center point to the two short edges, and the size of the angle obtained in the formula (4) to judge the size of the angle, the following three situations occur:

(1) It's found that when the angle is negative, the two forms of in subgraphs (b) and (d) of Figure 9 at this time in the distance 3 and distance 4 according to the size of the judgement of the direction of the opening, when distance 3 > distance 4 can be derived from the angle of c is: $90+\theta=-60.4$, when distance 3 < distance 4 it can be concluded that the angle of e is: $90-\theta=115.0$;

(2) When the angle is positive, it is the two forms of in subgraphs (c) and (f) of Figure 9. At this time, in the judgment of the opening direction according to the size of distance 3 and distance 4, when distance 3 < distance 4, it can be concluded that the angle of d is: $-90-\theta=133.3$, when distance 3 < distance 4. When distance 3 < distance 4, the angle of (f) is $90-\theta=52.6$;

(3) As arctan (k) to find the value can only be in $(0, 90)$ of the open area, cannot get 0 and 90 of these two special cases, this time, according to the short side of the gap in the longitudinal coordinates of the gap is equal and the distance 3 and distance 4 size, when distance 3 > distance 4, the angle is 0, as in subgraph (a) of Figure 9; when distance 3 < distance 4, the angle is 180, as in subgraph (b) of Figure 9.

The angle sought is the angle of rotation of the iron cap, with positive values being clockwise and negative values being anti-clockwise. The result of the rotation angle is shown in Figure 9, with subgraph (a) in Figure 9 as the counter-positive position, counter-clockwise rotation for negative values (e.g., in subgraphs (c) and (d) of Figure 9), and clockwise rotation for positive values (e.g., in subgraphs (e) and (f) of Figure 9).

4 Analysis of Experimental Data and Results

This paper combines the laboratory's existing equipment, such as the HuaRay Technology A3600MG100 camera and the M2016-12MP-2 lens, and, based on the PyCharm platform, the OpenCV3.10 vision function library, for the development and experimental verification and analysis. The experimental environment is shown in Figure 10.

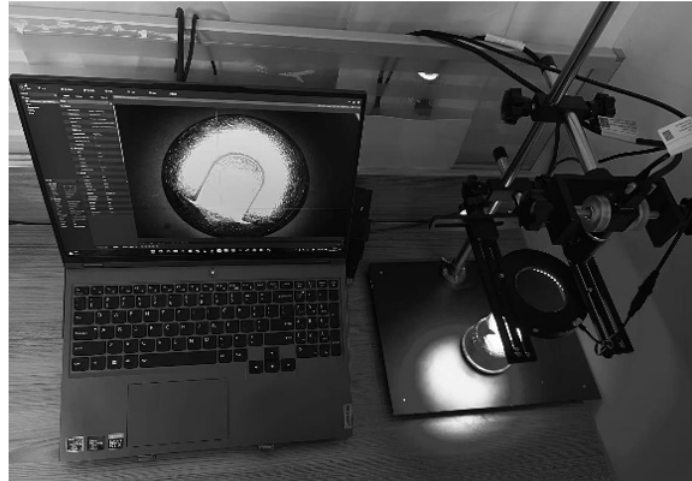


Figure 10. Experimental environment

A total of 20 iron cap pictures with different angles were collected, totaling 732, manually measured to mark the angle value, marked the picture name as the measured value, and verified the value calculated by the algorithm with the measured value.

Using the machine vision-based iron cap U-shaped notch alignment angle measurement system algorithm designed in this paper, repeated test experiments were conducted on different iron cap angles, respectively. The algorithm's single running time, the standardized average deviation of the measurement angle, the maximum error, and the detection accuracy in the range of error allowed to test the three indicators are shown in Table 1.

Table 1. Results of related experiments

Time to Run Once	0.034 s	
Standardized Mean Value	0.3°	
Maximum Error Value	0.8°	
Tolerance Range	±0.5° range	±1° range
Detection Accuracy	96.51 %	100 %

A total of 732 experiments were carried out. The results of the experiments in the iron cap U-shaped notch angle error 0.5° range of accuracy is 96.51%, error 1° range of accuracy is 100%, the algorithm runs a single time 0.034 seconds, the two error ranges and running speed meet the standards required by the automatic production line, and because the manual measurement of the angle may not be as accurate as the results of the detection, this algorithm is higher, but they all meet the system requirements for the accuracy of the U-shaped angle of the iron cap. And because the manually measured angle may not be as accurate as the detected results, the detection accuracy of this algorithm is higher, but all meet the system requirements for the accuracy of the U-shaped angle of the iron cap. It can be seen that within the error range, the machine vision-based iron cap U-shaped notch angle measurement system proposed in this paper can achieve accurate measurement of the iron cap U-shaped angle within the error tolerance.

5 Conclusions

In this system, first of all, the image is filtered to remove the noise to get a better recognition effect after filtering, and then binarized after corrosion to separate the target area from the interference by finding the second largest connected domain area to get the U-type target area after the processing of the above process, and then through the traditional image of the outline of the smallest external rectangular box, the target area positioning After the above process, the target area is positioned by the minimum external rectangular box of the contour in the traditional image, and the positive and negative values of the size of the angle and the minimum external circle center are compared with the distance of the two short edges to determine the direction of the gap. The rotation angle of the gap is then achieved, and the angle derived from the experiment meets the standard required by the automatic production line. However, there may still be deficiencies in this system, such as the fact that the image acquisition interference is large and the algorithm used cannot be very good to get the angle information. In the future, we can use machine learning or deep learning algorithms to use the manually labeled pictures as a data set, divided into a 70% training set, a 15% validation set, and a 15% test set to input the neural network for training, to get the trained model, and then test to get the accuracy rate and combine the accuracy rate to continuously improve the network model to get a better accuracy rate, thus improving the accuracy of neural network detection.

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

- [1] J. Y. Li, J. Li, X. P. Wang, G. Q. Tian, and J. F. Fan, "Machine vision-based method for measuring and controlling the angle of conductive slip ring brushes," *Micromachines*, vol. 13, no. 3, 2022. <https://doi.org/10.3390/mi13030447>
- [2] X. C. Gan, A. B. Sun, X. Ye, and L. Q. Ma, "Non-contact measurement of rotation angle with solo camera," in *Ninth International Symposium on Precision Engineering Measurement and Instrumentation*, Changsha, China, 2014. <https://doi.org/10.1117/12.2180752>
- [3] M. Blumenstein, X. Y. Liu, and B. Verma, "An investigation of the modified direction feature for cursive character recognition," *Pattern Recognit.*, vol. 40, no. 2, pp. 376–388, 2006. <https://doi.org/10.1016/j.patcog.2006.05.017>
- [4] H. Wu, W. Li, and S. Han, "Research on dynamic-angle measurement method for machine vision," in *Proceedings Volume 11341, AOPC 2019: Space Optics, Telescopes, and Instrumentation*, Beijing, China, 2019. <https://doi.org/10.1117/12.2540332>
- [5] X. Huo, H. Li, and K. Shao, "Automatic vertebral rotation angle measurement of 3D vertebrae based on an improved transformer network," *Entropy*, vol. 26, no. 2, 2024. <https://doi.org/10.3390/e26020097>
- [6] M. Malarvel, S. R. Nayak, P. K. Pattnaik, and S. N. Panda, "Machine learning-based approaches," in *Machine Vision Inspection Systems*. John Wiley Sons, Inc., 2021. <https://doi.org/10.1002/9781119786122>

- [7] H. Kim, Y. Yamakawa, T. Senoo, and M. Ishikawa, "Visual encoder: Robust and precise measurement method of rotation angle via high-speed RGB vision," *Opt. Express*, vol. 24, no. 12, pp. 13 375–13 386, 2016. <https://doi.org/10.1364/OE.24.013375>
- [8] W. M. Li, J. Jin, X. F. Li, and B. Li, "Method of rotation angle measurement in machine vision based on calibration pattern with spot array," *Appl. Opt.*, vol. 49, pp. 1001–1006, 2010. <https://doi.org/10.1364/AO.49.001001>
- [9] P. Yugander, C. H. Tejaswini, J. Meenakshi, K. S. Kumar, B. V. N. S. Varma, and M. Jagannath, "MR image enhancement using adaptive weighted mean filtering and homomorphic filtering," *Procedia Comput. Sci.*, vol. 167, pp. 677–685, 2020. <https://doi.org/10.1016/j.procs.2020.03.334>
- [10] H. H. Draz, N. E. Elashker, and M. M. A. Mahmoud, "Optimized algorithms and hardware implementation of median filter for image processing," *Circuits Syst. Signal Process.*, vol. 42, pp. 5545–5558, 2023. <https://doi.org/10.1007/s00034-023-02370-x>
- [11] H. Kurita and T. Maruyama, "An image filter system based on dynamic partial reconfiguration on FPGA," in *Advances in Parallel Computing*, 2014, pp. 540–547. <https://doi.org/10.3233/978-1-61499-381-0-540>
- [12] X. M. Zhang, B. S. Xu, and S. Y. Dong, "Adaptive median filtering for image processing," *J. Comput. Aided Des. Graph.*, no. 2, 2005.
- [13] L. Y. Xiao, C. D. Fan, H. L. Ouyang, A. F. Abate, and S. Wan, "Adaptive trapezoid region intercept histogram based Otsu method for brain MR image segmentation," *J. Ambient Intell. Human. Comput.*, vol. 13, pp. 2161–2176, 2022. <https://doi.org/10.1007/S12652-021-02976-6>
- [14] J. Q. Chen, A. J. Li, F. Zheng, S. S. Chen, W. K. He, and G. P. Zhang, "Research on tire surface damage detection method based on image processing," *Sensors*, vol. 24, no. 9, p. 2778, 2024. <https://doi.org/10.3390/s24092778>
- [15] J. Y. Chiu, "Automated medication verification system (AMVS): System based on edge detection and CNN classification drug on embedded systems," *Heliyon*, vol. 10, no. 9, p. e30486, 2024. <https://doi.org/10.1016/j.heliyon.2024.e30486>
- [16] L. Zhu and J. Yang, "Ancient books Chinese characters segmentation based on connected domain and Chinese characters feature," *Adv. Mater. Res.*, vol. 143-144, pp. 227–231, 2010. <https://doi.org/10.4028/www.scientific.net/AMR.143-144.227>
- [17] D. Sangeetha and P. Deepa, "FPGA implementation of cost-effective robust Canny edge detection algorithm," *J. Real-Time Image Proc.*, vol. 16, pp. 957–970, 2019. <https://doi.org/10.1007/s11554-016-0582-2>
- [18] D. Sundani, S. Widiyanto, Y. Karyanti, and D. T. Wardani, "Identification of image edge using quantum Canny edge detection algorithm," *J. ICT Res. Appl.*, vol. 13, no. 2, pp. 133–144, 2019. <https://doi.org/10.5614/itbj.ict.res.appl.2019.13.2.4>
- [19] L. Yu, H. B. Miao, G. P. Shen, and H. P. Su, "Workpiece recognition and positioning algorithm based on machine vision," *Autom. Instrum.*, vol. 38, no. 6, pp. 29–33, 2023. <https://doi.org/10.19557/j.cnki.1001-9944.2023.06.007>
- [20] Q. Li and J. K. Hu, "Online grading of apples based on machine vision," *Food Mach.*, vol. 36, no. 8, pp. 123–128, 2020. <https://doi.org/10.13652/j.issn.1003-5788.2020.08.022>
- [21] J. Sigut, M. Castro, R. Arnay, and M. Sigut, "OpenCV basics: A mobile application to support the teaching of computer vision concepts," *IEEE Trans. Educ.*, vol. 63, no. 4, pp. 328–335, 2020. <https://doi.org/10.1109/te.2020.2993013>
- [22] H. Cheng, C. G. Cai, Y. Wang, Z. H. Liu, and M. Yang, "A high precision rotating line detection method for the rotation angle measurement based on machine vision," in *4th International Conference on Computer Graphics and Digital Image Processing (CGDIP 2020)*, Kunming, China, 2020. <https://doi.org/10.1088/1742-6596/1627/1/012016>
- [23] M. X. Li, Q. X. Liu, and C. Y. He, "Research and achievement on cigarette label printing defect detection algorithm," *Appl. Mech. Mater.*, vol. 200, pp. 689–693, 2012. <https://doi.org/10.4028/www.scientific.net/AMM.200.689>
- [24] X. M. Li, Y. J. Li, R. Q. Chen, J. L. Chi, X. J. Shi, and H. Lin, "Evaluation of the minimum circumscribed circle based on the chord and its two corresponding minimum angles," *Meas.*, vol. 201, p. 111754, 2022. <https://doi.org/10.1016/j.measurement.2022.111754>