Journal of Research in Science and Engineering (JRSE)
ISSN: 1656-1996 Volume-6, Issue-9, September 2024
DOI: 10.53469/jrse.2024.06(09).07

Design of Machine Vision-Based Measuring System for the Pin Height of Camshaft Holder

Changjiang Mo[1], Zan Huang[2], Xiaojing Zhang[3]

[1]Office of Scientific Research, Lingnan Normal University, Zhanjiang 524048, Guangdong, China
[2]School of Mechanical and Electrical Engineering, Lingnan Normal University, Zhanjiang 524048, Guangdong, China
[3]School of Electronics and Electrical Engineering, Lingnan Normal University, Zhanjiang 524048, Guangdong, China
**Correspondence Author, mochj@lingnan.edu.cn*

Abstract: ***In order to meet the precise measurement requirements of pin height of camshaft holder , the system designed a fixture model based on dual cameras and dual backlight with cross structure, and proposed an algorithm that uses dual end pre-selection to fit the line, and calculate the distance between two lines by average value. The Python program can automatically obtain the pin height data, compare it with the preset error subsequently, and determine whether the work-piece passes the test at last. The experiment shows that the relative error value of the measurement is less than 0.3%, which meets the production requirements of the factory.***

Keywords: Machine vision, Line fitting, Fixture, Modbus protocol.

1. Introduction

The engine is the core of the automobile, and the design and manufacture of its key components have an important impact on the performance [1],[2] life-cycle and reliability of the whole machine. Among them, the camshaft, as a key component of the engine, mainly drives and controls the opening and closing of the valves of each cylinder, and its performance directly affects the running performance and service life of the automobile engine at work. It can be seen that the stability of the camshaft's working position determines whether the engine can work normally. It needs to use the camshaft holder to lock its position after putting the camshaft into the engine block. It can be more conveniently aligned with the engine block, and the position of the camshaft can be better limited by hitting the screws at a later stage after pressing the metal pins at both ends of the camshaft holder. At this time, there are certain requirements on the two sides of the pins pressed into the camshaft holder, it should be not too high and too low, it would leave a certain gap when too high, or affect the stability of the camshaft installation when too low.

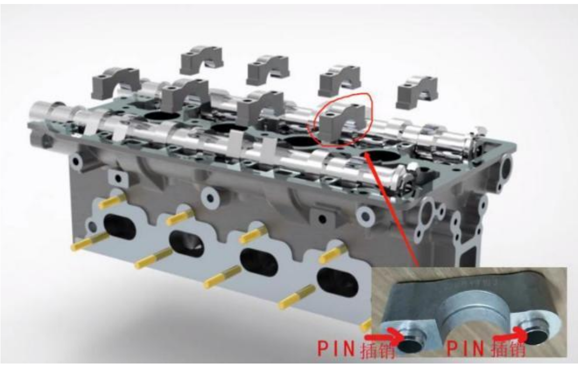


Figure 1: Camshaft holder in Engine

In some factories, the pin height measurement of work-piece (Camshaft holder) is mainly done by manual operation of micrometer and reading with naked eyes. This method has many shortcomings, such as reading accuracy and determining whether there is a human instability problem; the reading time is longer, and it can not provide an interface with the subsequent mechanical devices, and so on.

Ziping Wan [3] et al. of National University of Defense Technology proposed a new machine vision-based precision measurement system for work-piece contour dimensions, using LabVIEW as the platform to develop the system software, and adopting the improved Canny algorithm to recognize the image edges, which makes the measurement speed faster and the efficiency of part contour information extraction higher.

Aiming at the problem of complex environment and interference in industrial production line, Jiang Junhua [4] et al. proposed a method based on the combination of deep learning technology and binocular IR structured light depth camera to realize a part detection and localization system based on SSD-Mobilenet neural network and industrial camera, which can carry out real-time category detection and three-dimensional coordinate acquisition of the target work-pieces. Experiments show that the system has the advantages of strong anti-interference ability, fast detection frame rate, high accuracy, high positioning accuracy, etc., and can quickly and accurately guide the manipulator work.

Gadelmawla [5] et al., Department of Mechanical Engineering, Faculty of Engineering, Qassim University, realized online precision measurement of gear dimensions by machine vision based on CCD camera. The authors built a vision system and used it to capture images of gears to be measured or inspected. A specially developed software (named GearVision) internally uses Microsoft Visual C++ to analyze the captured images and perform the measurement and inspection process. This was verified by measuring two sample gears and comparing the calculated parameters with the actual values of the gear parameters.

Summarizing the current research situation at home and abroad, machine vision and deep learning technologies have begun to be applied in the field of industrial inspection at home and abroad, and have achieved certain results, and are constantly developing in the direction of faster, more accurate and intelligent. This project combines the on-site use of the environment and related technical requirements, the use of dual light source and dual camera configuration, proposed the use of dual-lens, the use of industries computer and machine vision OpenCV algorithm [6],[7],[8] to measure the height of the camshaft holder's pin, determine whether it is a qualified product solution, which can solve the shortcomings of the current manual measurement, improve the accuracy of the measurement of the work-piece.

2. Overall System Scheme

The system considers the issue of mechanical structure and cannot use other sensor solutions attached to the original production equipment. Therefore, a high sales visual measurement solution that does not affect the original equipment is adopted. The measurement system, as shown in Figure 2, uses an industrial computer as the main control board and installs two 500W pixel value high-definition cameras with USB interfaces. They are aimed at the pin parts on both sides of the measured camshaft holder. When the user receives the interface signal of the level change (PLC signal), they start shooting and take two high-resolution pictures. After sending them back to the industrial computer system for image recognition, image processing and calculation are carried out. If the calculation error is within the specified range, a green light is displayed in the middle of the interface program, otherwise a red light is displayed.

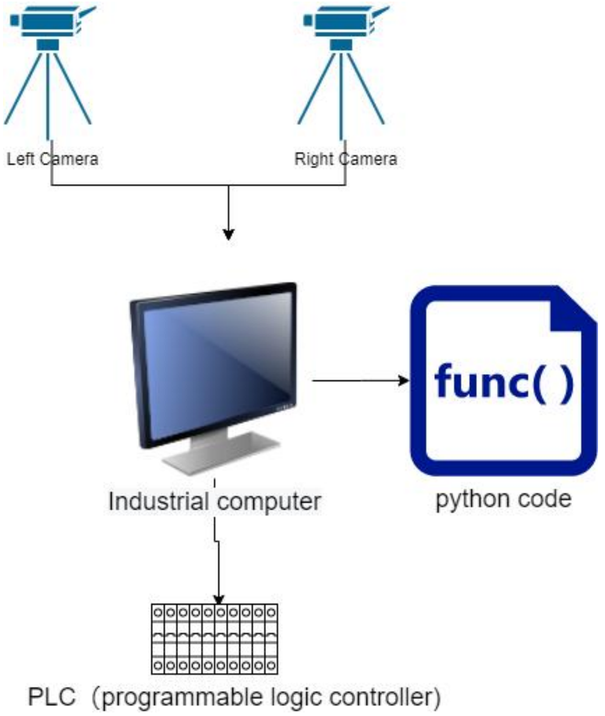


Figure 2: System Design Diagram

3. Design and Implement of Data Forwarding Software System

3.1 Fixture Design

3.1.1 Fixture design

In order to increase the production speed, the engine camshaft holder formed after stamping the pins are not placed elsewhere but measured in-line at the stamping process equipment itself. Therefore, the fixture should consider both the installation position of the camera light source, which does not impede the original production process, as well as the non-stop stamping process resulting in the stability of the fixture position and other issues. A fixture was designed that can be adjusted to fix the position of the object to be measured and the camera, so that the dimensions on the photo form a fixed proportional relationship with the real-world physical dimensions and the distance can be measured accurately. In addition, different bases can be designed according to different work-pieces, adapting different parameters for measurement, as shown in Figure 3, the camera and light source are distributed on both sides of the measured work-piece, and the bottom of the measured work-piece has an interchangeable base design.

Figure 3: Fixture module

3.2 Camera Calibration

When using machine vision to handle physical distance measurement, the camera needs to be calibrated. The dimensions on the picture are pixel distances, and after calibration, the conversion relationship between pixel distances and physical spatial distances can be derived to calculate the actual physical dimensions.

The distortion and intrinsic parameters of a camera are inherent characteristics of the camera itself, and can be used continuously after calibration. However, due to the fact that the camera itself is not an ideal pinhole imaging model and calculation errors, the results obtained when calibrating with different images may vary. If the projection error is generally small, the calibration results can be used.

A monocular camera is proposed to be used to calculate the actual physical height of an object or target in a photograph using similar triangles principle. Suppose there is a target or object with height H. The target or object is then placed at a distance of H from the camera. This target is then placed at a distance of D from the camera and the camera is used to take a picture of the object and measure the pixel height of the object P. This gives the formula for the focal length of the camera:

$$F=PD/H \tag{1}$$

When replacing the object or target, a similar triangle can be used to calculate the height of the object or target:

$$H'=D'P'/F \quad (2)$$

In this system, the measuring fixture and camera are fixed, i.e. D=D', and the formula can be further simplified as:

$$H'=P'(H/P) \quad (3)$$

Using a standard work-piece calibration, after determining D and H, the calibration parameter P can be obtained in the system, and the work-piece can be replaced. the final value of the work-piece height H' can be obtained by inputting real time P ' value into the above equation.

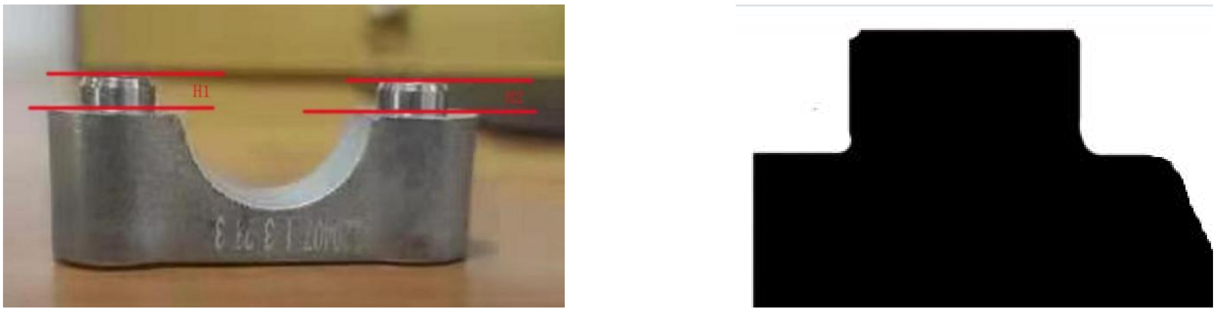
3.3 Camera Light Solution

The camera obtains a projection image of the object being measured, the schematic diagram is shown in Figure 4, 4(a) is the height of the pin of the work-piece to be measured, 4(b) is the projection of one of the pins.

Using a projection light source, the camera and light source are distributed on both sides of the object being measured. After obtaining the projection map of the measured object, the algorithms such as image de-noising, smoothing, fixing lines and calculating are carried out.

3.4 Graphic Processing Algorithm

The flowchart for processing machine vision images using Python+OpenCV in industrial computer is shown in Figure 5:



(a) Camshaft holder (b) Partial projection of the pin

Figure 4: Schematic diagram of the measured object and its projection image

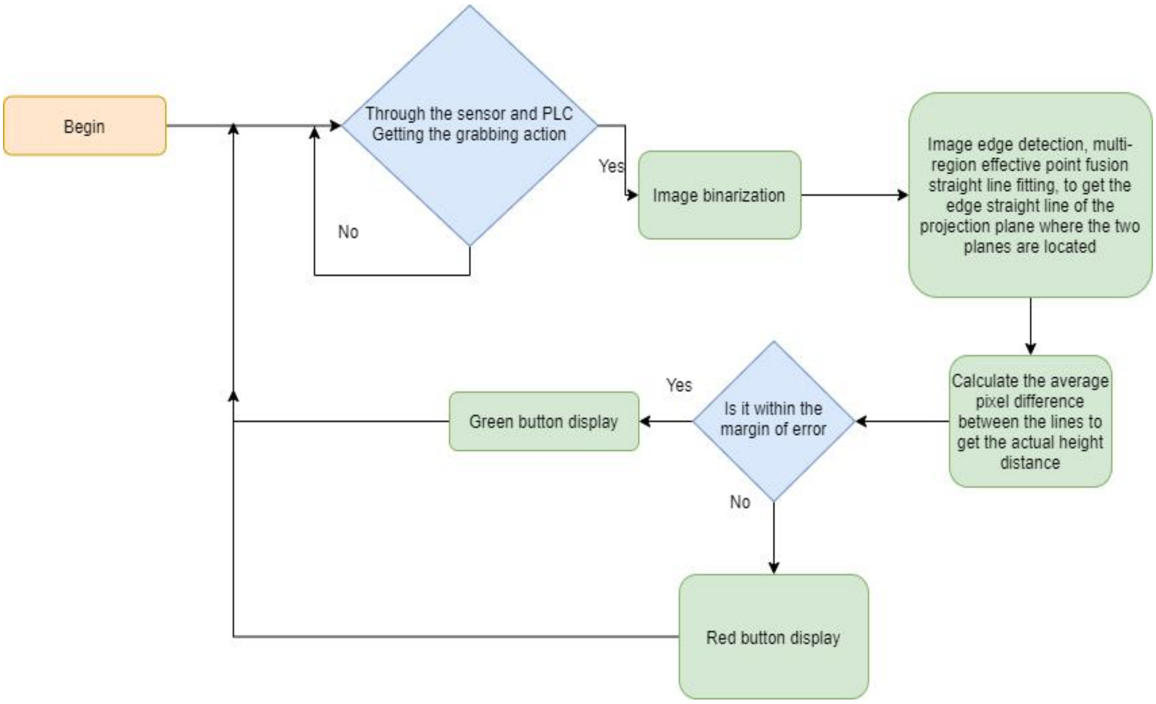


Figure 5: Flowchart of algorithm

Given multiple points(n), a straight line is then fitted to these points, this most common algorithm is the least squares method of fitting a straight line with multiple constraint equations. The core idea of the least squares method of fitting a straight line is that the sum of the squared differences between all the sample values and their corresponding model values is used as the objective loss function, and when the value of the objective loss function achieves a minimum value, the model is considered to be a fit of all the samples.

When the derivative is zero, the function reaches its extremum, so the parameters can be solved by taking the partial derivatives of the objective function with respect to each parameter and making the partial derivatives 0. The parameters obtained when the partial derivative is 0 constitute the solution of the least squares method.

Suppose there are n points (xi, yi) ($0 \le i < n$) and assume that they are fitted to a straight line Y = ax + b. Then for each xi, its fitted value is Yi = axi + b. The objective function is then:

$$f(x) = \sum_{i=0}^{n-1} (Y_i - y_i)^2 = \sum_{i=0}^{n-1} (ax_i + b - y_i)^2 \quad (4)$$

Find the partial derivatives of f(x) with respect to a and b, respectively:

$$\frac{\partial y}{\partial a} = \sum_{i=0}^{n-1} 2(ax_i + b - y_i)\, x_i \; = 2\sum_{i=0}^{n-1} (ax_i^2 + bx_i - x_i y_i) = 2(a\sum_{i=0}^{n-1} x_i^2 + b\sum_{i=0}^{n-1} x_i - \sum_{i=0}^{n-1} x_i y_i) \quad (5)$$

$$\frac{\partial y}{\partial b} = \sum_{i=0}^{n-1} 2(ax_i + b - y_i) = 2(a\sum_{i=0}^{n-1} x_i + \sum_{i=0}^{n-1} b - \sum_{i=0}^{n-1} y_i) = 2(a\sum_{i=0}^{n-1} x_i + bn - \sum_{i=0}^{n-1} y_i) \quad (6)$$

Let the above partial derivatives be zero to obtain a system of quadratic equations:

$$\begin{cases} a\sum_{i=0}^{n-1} x_i^2 + b\sum_{i=0}^{n-1} x_i = \sum_{i=0}^{n-1} x_i y_i \\ a\sum_{i=0}^{n-1} x_i + bn = \sum_{i=0}^{n-1} y_i \end{cases} \quad (7)$$

Letting:

$$A = \sum_{i=0}^{n-1} x_i^2 \,, B = \sum_{i=0}^{n-1} x_i \,, C = \sum_{i=0}^{n-1} x_i y_i \,, D = \sum_{i=0}^{n-1} y_i$$

Then:

$$\begin{cases} aA + bB = C \\ aB + bn = D \end{cases} \quad (8)$$

Solving the above system of equations to obtain a and b gives the linear fit parameters:

$$\begin{cases} a = \frac{nC-bD}{nA-B^2} \\ b = \frac{DA-BC}{nA-B^2} \end{cases} \quad (9)$$

However, the drawback of this algorithm is that once there are one or more outliers among these points, the straight line fitted by the algorithm will be completely deviated.

In order to prevent the least squares algorithm from removing or reducing the influence of outliers during fitting estimation, a double ended positioning point selection mean algorithm is proposed. Firstly, rectangular box point selection is performed on the left and right end regions of the upper edge line segment of the pin height and the lower edge line segment in contact with the base. Secondly, the selected points are substituted into the least squares algorithm to obtain the upper and lower fitting lines of the pin. Finally, calculate the average distance between the corresponding points of the two lines to obtain the sales height. The core code and steps are as follows:

Step1 Using the least squares algorithm fits a line to obtain the slope and intercept of a primary equation(k,b), the function prototype is below:

```
def Cal_kb_linear_fitline1(edgex,wbox,parts):
    loc = []
    if parts==1:
        offset=0
    else:
        offset=4
    for x1 in range(wbox[0+offset][0],wbox[1+offset][0]):
        for y1 in range(wbox[0+offset][1],wbox[1+offset][1]):
            if edgex[y1,x1]>0:
                loc.append([x1,y1])
    for x1 in range(wbox[2+offset][0],wbox[3+offset][0]):
        for y1 in range(wbox[2+offset][1],wbox[3+offset][1]):
            if edgex[y1,x1]>0:
                loc.append([x1,y1])
    loc = np.array(loc)
    output = cv2.fitLine(loc, cv2.DIST_L2, 0, 0.01, 0.01)
    k = output[1] / output[0]
    b = output[3] - k * output[2]
    p0x=output[2]
    p0y=output[3]
return k,b
```

Setp2 Using vector algorithm to get distance of a point to a line, the formula goes below:

$$\text{distance} = \frac{\left|\overrightarrow{\text{line\_point1} - \text{point}} \times \overrightarrow{\text{line\_point2} - \text{point}}\right|}{\left|\overrightarrow{\text{line\_point1} - \text{line\_point2}}\right|} \quad (10)$$

in formula (10), three points are interconnected to form three vectors. The cross product of two vectors is the area of a parallelogram. While "distance" is the height of the same parallelogram, The third vector is the base corresponding to that height, and because their multiplied areas are equal, the following equation is formed to calculate the distance from a point to a line. In the code below, "np.cross" implements the result of the cross product of two vectors, "np.linalg.norm" gets the modulus of a line formed by two points, The function is defined below:

```
def point_distance_line(point,line_point1,line_point2):
    vec1 = line_point1 - point
    vec2 = line_point2 - point
    distance = np.abs(np.cross(vec1,vec2))/
    np.linalg.norm(line_point1-line_point2)
return distance
```

Step3 Calculate average distance between two lines marking for the height of the camshaft holder, the code segment is below:

```
for x1 in range(wbox[0][0],wbox[3][0]):
    y1=k*x1+b
    pt1=(x1,int(y1))
    point =np.array(pt1)
    line_point1 =np.array(pt3)
    line_point2 = np.array(pt4)
    dist=point_distance_line(point,line_point1,line_point2)
    total=total+dist
    cnt=cnt+1
    #print('x1=%d,y=%d,dist=%.3f pixels'%(x1,y1,dist))
if cnt!=0:
    res[0]='pixels= %.3f pixels'%(total/cnt)
    res[1]='dist= %.3f mm'%(total*ps[tt]/cnt)
else:
    res[0]=0
    res[1]=0
```

3.5 PLC Interface

XC2-16T-E model of Xinjie products is used, as shown in Figure 6, after downloading the corresponding program, linking to the measurement system by IO interface, the industrial control computer through the MODBUS protocol [9] to obtain the signal when the sensor measurement signal is detected, computer can start taking pictures, proceed the image and data processing.

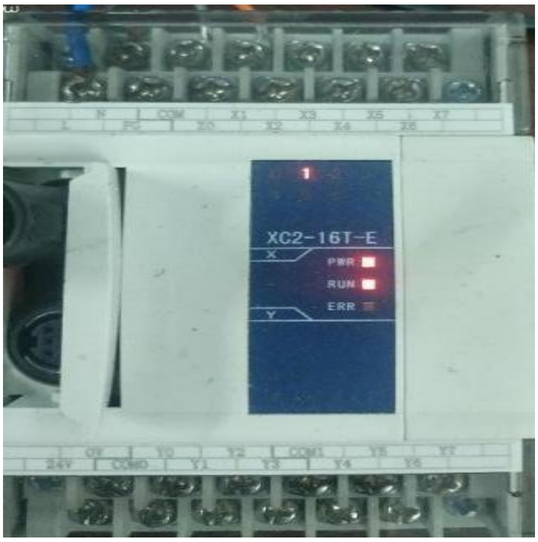


Figure 6: PLC

After installing third-party Modbus library in Python [10], the project completed the debugging of read-write communication with the PLC data unit and executed it using the Multiplayer process, effectively calling the performance of multi-core computers without affecting the execution of the main program. The function segment is as follows:

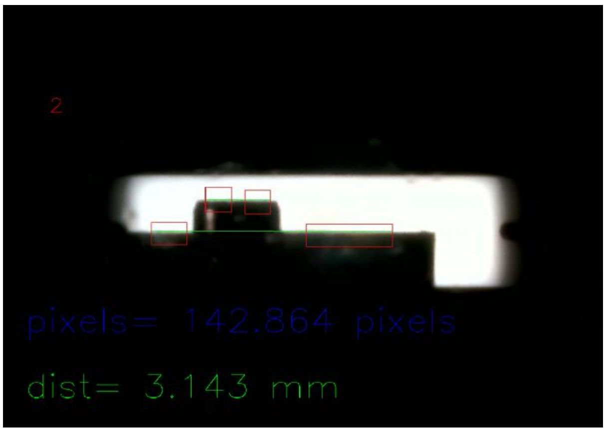
```
def mod(num):
   red=[]
   red1=[]
   red2=[]
   alarm = ""
   PORT="com7"
   try:
      master = modbus_rtu.RtuMaster(serial.Serial(port=PORT, baudrate=19200, bytesize=8, parity='E', stopbits=1))
      master.set_timeout(5.0)
      master.set_verbose(True)
      i=1
      while (True):
         if (num[5]==-1):
            break
         time.sleep(0.1)
         i+=1
         red = master.execute(1, cst.READ_COILS, 0, 1)
         varD = (master.execute(1, cst.READ_HOLDING_REGISTERS, 4500, 1))[0]
         if varD!=30000:
            num[5]==-1
            num[4]=-2
            break;
         tmp=red[0]
         if tmp==1 and num[2]==0:
            num[1]=1#start signal
            print('start signal')
            time.sleep(0.01)
            red1 = master.execute(1, cst.READ_COILS, 0, 1)
            tmp=red1[0]
            if tmp==1:
               num[1]=2#start signal confirmed after 10ms
               num[2]=1
               print('start signal confirmed after 10ms')
         num[0]=red[0]
         alarm = "Normal"
   except Exception as exc:
      alarm = str(exc)
      num[4]=-1
   print('sub Process quit')
```

The multi-core CPU call function segment statement is as follows:

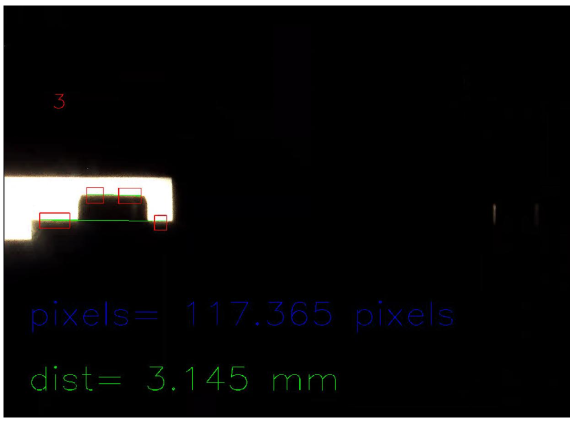
```
num=multiprocessing.Array(ctypes.c_int,[0,0,0,0,0,0])
p = Process(target=mod,args=(num,))
p.Daemon=True
p.start()
```

4. System Experiment Measure Results

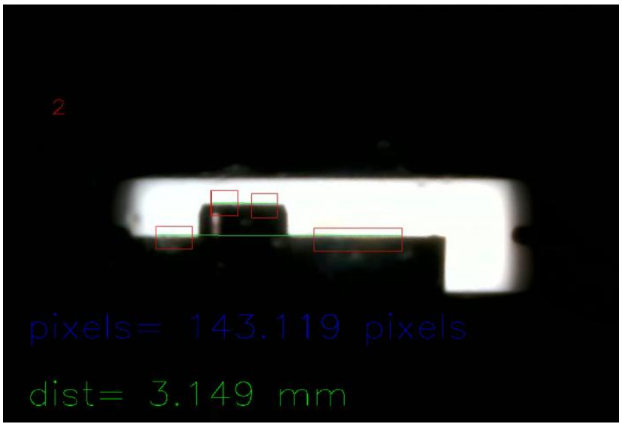
Use the standard pin height of 3.15mm for pin measurement of the camshaft holder, select three measurement results randomly. It is demonstrated left and right side of the pin measurement graph Figure 7 respectively. According to the date in these graphs, the experimental result is recorded and in table 1 where the relative error of about 0.2% of the measured value is calculated as well.



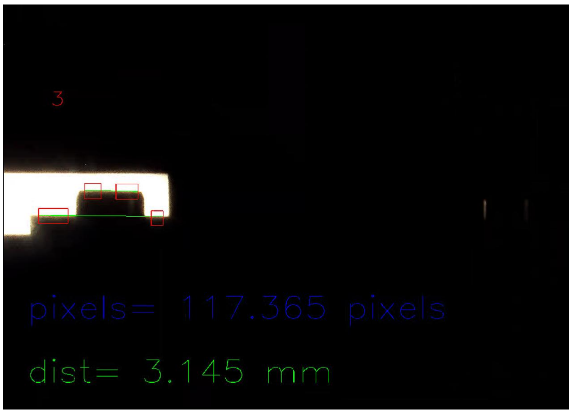
(a)



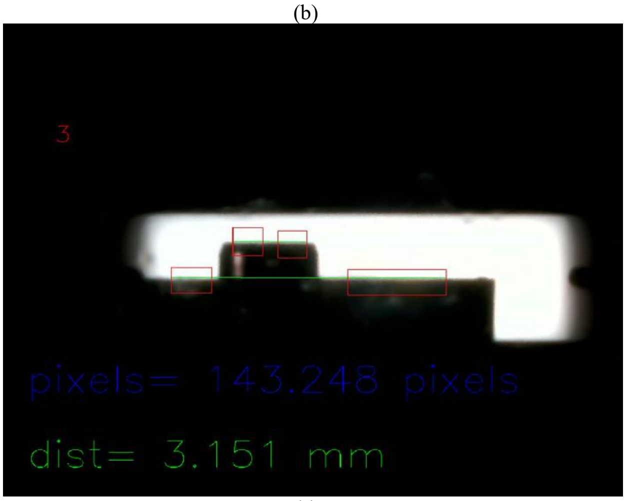
(d)



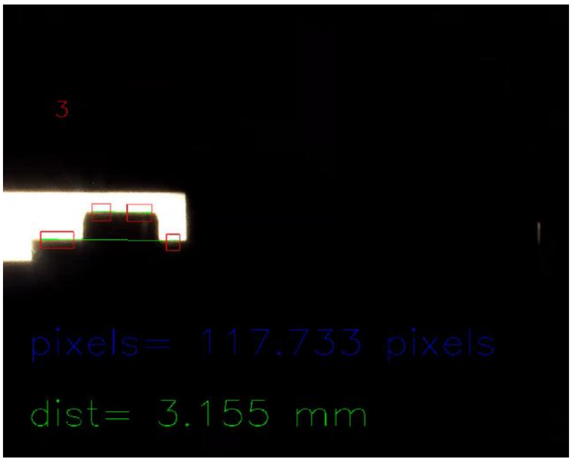
(b)



(e)



(c)



(f)

Figure 7: Pin Height Measurement

Table 1: Measured values and errors

Times	Left pin height (mm)	Relative error on the left side (%)	Right pin height(mm)	Relative error on the right side (%)	Pass
1	3.143	-0.22%	3.145	-0.16%	True
2	3.149	-0.03%	3.145	-0.16%	True
3	3.151	0.03%	3.155	0.16%	True

5. Conclusion

The experimental results show that the system can calculate the pin height of camshaft holder in engine in real time through the image processing algorithm, and after replacing the fixture, by modifying the preset value of the program, it can measure different work-pieces with high compatibility. The system has high measurement accuracy and meets the technical requirements of the project.

Acknowledgment

This work is supported by the 2023 Zhanjiang Science and Technology Plan Project (Project No.2023B01054), Lingnan Normal University Provincial College Student Innovation and Entrepreneurship Project (Project No: S20231057921).

References

[1] Z. Huang, ‘Design and implementation of intelligent camshaft press-fit testing system’, J. Phys. Conf. Ser., vol. 2787, no. 1, p. 012010, Jun. 2024, doi: 10.1088/1742-6596/2787/1/012010.
[2] C. Xu, L. Meng, R. Gong, X. Gong, and A. Li, ‘Camshaft Loosening Diagnosis on the Basis of Generalised Force Recognition at the Centre of Gravity of an Engine’, Math. Probl. Eng., vol. 2021, pp. 1–8, Aug. 2021, doi: 10.1155/2021/1908329.
[3] Z. Wan, ‘Parts Contour Size Measurement System Based on Machine Vision’, Appl. MCU Embed. Syst., vol. 17, no. 12, pp. 32-34+58, Aug. 2017.
[4] J. Jiang, ‘Workpiece Detection and Localization System Based on Neural Network and Depth Camera’, Sens. Microsyst., vol. 39, no. 08, pp. 82–85, Jun. 2019, doi: 10.13873/J.1000-9787(2020)08-0082-04.
[5] E. S. Gadelmawla, ‘Computer vision algorithms for measurement and inspection of spur gears’, Measurement, vol. 44, no. 9, pp. 1669–1678, Nov. 2011, doi: 10.1016/j.measurement.2011.06.023.
[6] B. M. Merzah, M. S. Croock, and A. N. Rashid, ‘Football Player Tracking and Performance Analysis Using the OpenCV Library’, Math. Model. Eng. Probl., vol. 11, no. 1, pp. 123–132, Jan. 2024, doi: 10.18280/mmep.110113.
[7] H. Boxia, H. Yong, X. Rong, and Y. Hongfeng, ‘High-precision automatic measurement of two-dimensional geometric features based on machine vision’.
[8] H. Habeenzu, P. McGetrick, S. Taylor, and D. Hester, ‘UAS-based bridge displacement measurement using

two cameras with non-overlapping fields of view', Autom. Constr., vol. 167, p. 105687, Nov. 2024, doi: 10.1016/j.autcon.2024.105687.

[9] S. Elamanov, H. Son, B. Flynn, S. K. Yoo, N. Dilshad, and J. Song, 'Interworking between Modbus and internet of things platform for industrial services', Digit. Commun. Netw., vol. 10, no. 2, pp. 461–471, Apr. 2024, doi: 10.1016/j.dcan.2022.09.013.

[10] 'Application of Python Automation Tool Combined with OCR Technology in "Financial Checkup" of Universities', Account. Corp. Manag., vol. 5, no. 12, 2023, doi: 10.23977/acccm.2023.051220.

Author Profile

Changjiang Mo received the B.S degree in Education of Electronics Technology Application from Lingnan Normal University in 2003 and the M.S. degree in Computer Technology and Engineering from Chongqing University in 2010. The research fields are intelligent devices, computer network, image recognition, etc.