**TELKOMNIKA**, Vol.12, No.3, September 2014, pp. 683~688
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v12i3.108

# Research on Keyhole Diameter's Vision Measurement Based on Parallel Technology

**Zhang Chuan-chuan, Zeng Zhi-qiang, Wang Jun-yuan, Du Wen-hua**
North University of China
School of Mechanical and Power Engineering
Xueyuan Rd.3, Taiyuan, 030051 China
e-mail: zhcc22060060@163.com

***Abstract***

*A keyhole diameter of the cartridge's measurement based on machine vision is a part of the cartridge's geometry measurement system, according to the system requirements, to complete the measurement within 5 seconds. A Image Collection System was constructed using computer, CCD camera, LED source, meanwhile a measurement system was compiled by C# on VS2010 platform based on machine vision. Use the Otsu algorithm to extracts the keyhole's edge and near the pixels in order to reduce the computational Canny operator, and use parallel computing in the Canny operator to improve computing speed purposes. Use QueryPerformanceCounter timer for each module timing Canny operator, Canny operator improved computation time is reduced from the original 6s to nearly a hundred ms improved. Meet the time requirements of cartridge geometry measurement system, and other machine vision in which the project can be widely used.*

***Keywords:*** *machine vision; trigger keyhole detection; edge detection; sub-pixel; threshold calculation*

## 1. Introduction

The annual output of some type cartridge reached more than one million, but the diameter of this type cartridge keyhole is only 1mm, Production processes can't completely guarantee the quality of the keyhole machining. when the keyhole tolerance, likely to cause cartridge launch failure, so the military require the production units to accurately measure the keyhole diameter for each cartridge to ensure the overall quality of the cartridge. According to the characteristics of the measurement object, the proposed method using the machine vision measurement. Firstly shooting Partial image of the keyhole and processing this image, then Calculating the diameter of the keyhole, finally comparing with the tolerance of the diameter values and determining whether compliance with the requirements.

In order to improve the accuracy and precision of the measurement, using canny operator edge detection algorithm to detect cartridge keyhole edge point are used when processing this image [1]. Canny operator at the same time improve the accuracy of the computer has increased the amount of computation. Under the conditions of the existing laboratory equipment, running time of Gaussian filter is 67.28 ms, edge contour extraction is 5719.23ms in Canny operator. So the whole time of Canny operator is 5786.61ms, cartridge measurement system can't meet the time requirements. Therefore, in order to improve detection efficiency, a method of using Otsu operator edge coarse positioning to reduce the computation [2] and parallel processing technology is proposed to make sure the system performance to meet the actual requirements.

## 2. The main parameters of the CCD camera and computer

The main flows of measuring system are image acquisition, image analysis, measurement, and outputs the result. To achieve high-speed measurement of the keyhole diameter must be in full knowledge of the size of the CCD camera to capture images of the situation fully hardware performance of your computer.

The hardware consists of an optical illumination system, CCD camera, computer hardware and related auxiliary equipment elements. According to the keyhole imaging requirements and the design of the measuring system, Adjust the collected image, the image pixel size of 512 × 512, while the camera's SNR is 40dB. Computer is using AMD A8-4500M

*Received January 29, 2014; Revised May 26, 2014; Accepted June 10, 2014*

quad-core processor, Windows 7 operating system. Image Measurement System program is written in C # under the VS2010 platform. Image acquisition system schematics and keyhole original image are shown in Figure 1 and Figure 2.

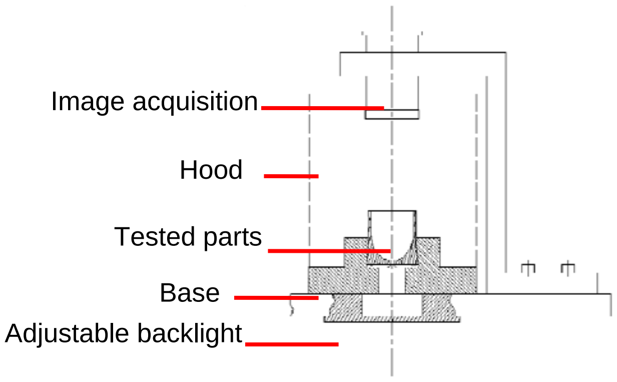


Figure 1. image acquisition system schematics

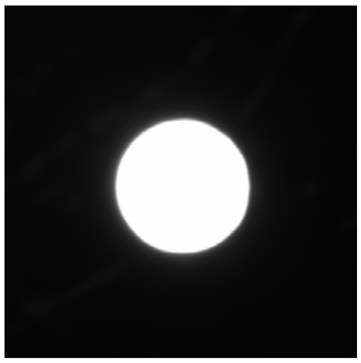


Figure 2. keyhole original image

## 3. **C# Parallel Computing**

Emergence of parallel computing is the inevitable result of the development of computer science, is the same process using a variety of computing resources to solve computing problems. Under the premise of simultaneous, the process can be calculated into small parts, to solve computational problems in Concurrent way. This is an effective means to improve the computer system calculates the speed and processing power. The basic idea is to use multiple processors to solve the same problem collaborative, that the problem is broken down into several parts, each part by a separate processor to parallel computing.

With the rapid development of computer hardware , image processing on multicore processors has already become a fact of existing, but the traditional image processing programming mode must be compatible with the new hardware environment to make the image processing speed to achieve the best application results [3]. C# is integrated TPL (Task Parallel Library) and PLINQ (Parallel LINQ), parallelized applications can be achieved, which will greatly enhance the speed of the application is running.

Static class named *system.Threading.Parallel* provides three important ways *For*, *Foreach* and *Invoke* located two namespaces which are named *System.Threading* and *System.Threading.Tasks*.

### 3.1 Guass filter parallel computing

Gaussian filter is a linear filtering, the value of each pixel, both by itself and the other pixel values through the neighborhood to get a weighted average [4]. The main feature is the Gaussian function is still a Gaussian function after Fourier transform, so the application of fast Fourier transform can put convolution of airspace transformed into the product operations of frequency domain, which greatly reduces the computation time. In practical application, the two-dimensional Gaussian function G(x, y) is decomposed into one-dimensional Gaussian function G(x)and G(y)in the x direction and y direction for image filtering:

$$G(x,y)=(1/2\pi\sigma)\exp(-(x^2+y^2)/2\sigma^2) \quad (1)$$

$$G(x)=(1/2\pi\sigma)\exp(-x^2/2\sigma^2) \quad (2)$$

$$G(y)=(1/2\pi\sigma)\exp(-y^2/2\sigma^2) \quad (3)$$

In actual programming, method of *Foreach* and method of *Partitioner.Create* which are in static class named *System.Threading.Parallel* is combined for parallel computing of algorithm [5].

When the horizontal filtering, two-dimensional array of images is automatically divided into several small two-dimensional array by Parallel.ForEach (Partitioner.Create (0, BmpData.Height), (H) => {}).The loop of each small two-dimensional array in height direction is for (j = H.Item1; j <H.Item2; j + +), in width direction is for (i = 0; i <BmpData.Width; i + +). When the vertical filtering, two-dimensional array of images is automatically divided into several small two-dimensional array by Parallel.ForEach (Partitioner.Create (0, BmpData. Width), (H) => {}). The loop of each small two-dimensional array in width direction is for (i = W.Item1; i< W.Item2; i++), in height direction is for (j= 0; j< BmpData. Height; j++). The filtering statement of a point on the image is temp + = grayValues [i * length1 + rem] * filter [k + radius].

### 3.2 Parallel computing of edge contour extraction

The edge contour extraction procedure is a serial operation before multi-core computers appeared. With the advent of multi-core technology, the method of *Parallel. Invoke* in C # put a picture into four parts to the edge contour extraction [6]. Method of *Parallel.Invoke* is the easiest way to parallelize the serial code. *Parallel.Invoke* put a cartridge keyhole capture original image is divided into four sections for parallel processing as shown in Figure 3.

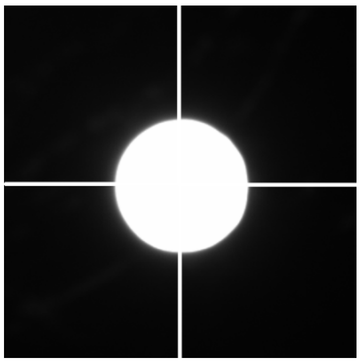


Figure 3. parallel processing image segmentation

## 4. Threshold segmentation and edge detection

For a pixel size of 512 × 512 image fire-hole of the cartridge, which is only a few points around the edge point and the edge contour of edge points need to be processed. Setting two concentric circles with the outline, one slightly larger than the contour edges, slightly more than one contour edge. As shown in Figure 4, only the pixels within the red circle only have the edge contour feature is the object should be extracted.

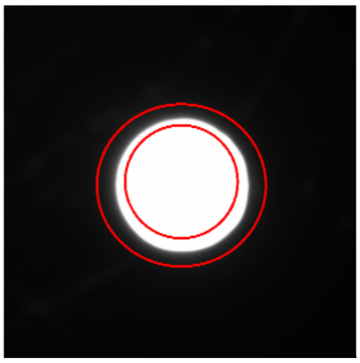


Figure 4. position of the edge to be processed

### 4.1 Improved Otsu threshold calculation

Adding Otsu algorithm after Gaussian filter in Canny operator can achieve automatic threshold calculation, this method is relatively simple and fast[7,8].

The basic calculation method is: the gray range [*Lmin, Lmax*] of the image, the gray value of the pixel is divided into two categories $C_1$ and *C2* in accordance with the threshold value *T*, *C1* is consisted by the pixel which the gray range is [*Lmin, T*], *C2* is consisted by the pixel which the gray range is [*T +1, Lmax*], by the formula (4) classes square error between two types of calculations.

$$\sigma^2 = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)] \qquad (4)$$

In the formula (4), $\omega_1(t)$ is the number of pixels of $C_1$, $\omega_2(t)$ is the number of pixels of $C_1$, $\mu_1(t)$ is the average gray value of the pixels of $C_1$, $\mu_2(t)$ is the average gray value of the pixels of $C_2$. Calculating the optimal threshold at the same time obtaining an image of the same maximum pixel value.

Achieving Otsu algorithm in C #, finding the minimum pixel values $L_{min}$ and the maximum pixel values $L_{max}$ by cycle comparison , and then calculating the optimal threshold *T* of the image. the gray's range of the edge point and near is in the interval [$T$, $L_{max}$], at the same time recording position of these pixels.

Extracting the image edge points and ten points gradient direction near the pixel values of the pixels through a program. As shown in Table 1, the gradient is the biggest when the gray value of the pixel is 204, so this point is determined as an edge point. Optimal threshold of the image is 129. As shown in Figure 5, the intermediate circular part is extracted edge points and near.

Table 1. Gray and gradient near the edge of a ten-point and point-pixel gradient direction

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Gray | 55 | 64 | 73 | 91 | 109 | 154 | 204 | 252 | 254 | 254 |
| Gradient | 4 | 9 | 9 | 18 | 18 | 45 | 50 | 48 | 2 | 0 |

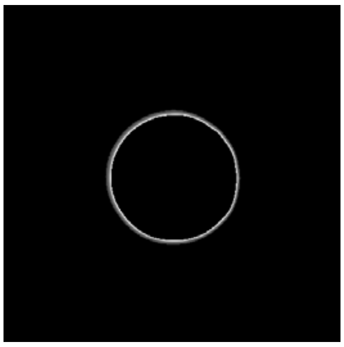
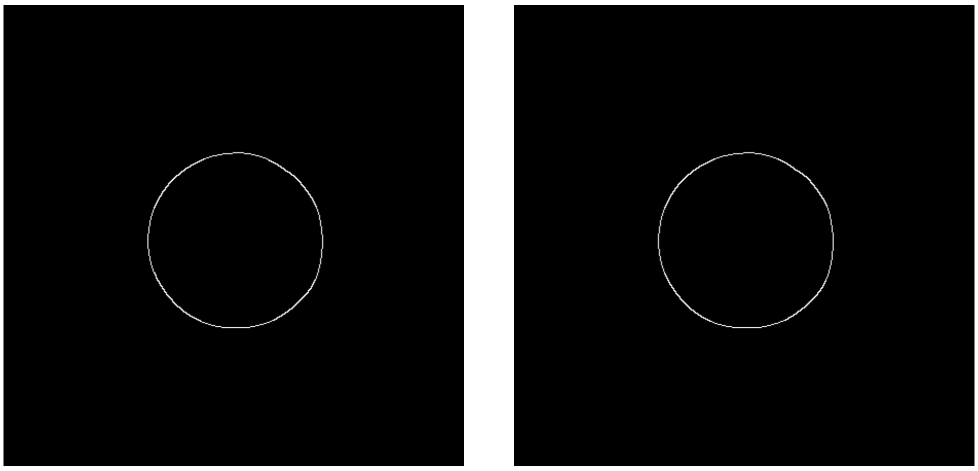


Figure 5. improved Otsu operator Figure

### 4.2 Edge Extraction

Original image contains a total of 262,144 pixels, the improved Otsu operator extracting the number of pixels and near are about 2560. Used by the edge contour extraction point decreased from 262,144 to 2,560. This can greatly increase the edge contour extraction speed. Therefore, when using the C # language, don't need to traverse each pixel of the whole image, only need to calculate Otsu operator extracted pixels. Figure 6 is improved around the edge contour extraction comparison chart, it can be seen from Figure extraction is no different.



(a) The effect of no improved Canny (b) The effect of improved Canny

Figure 6. improved Canny operator before and after the treatment effect comparison chart

## 5. Measurement results and analysis

Using a timer which the lowest identified is 1ms in C# named *QueryPerformanceCounter* record the time of improved Canny operator before and after each module. And for the evaluation of the performance of parallel computing, including speedup and efficiency.

Speedup is defined as: if the execution time of an algorithm execution time of the optimal serial algorithm is $T_s$, parallel algorithms is $T_p$, the ratio of parallel speedup algorithm $S=T_s/T_p$ [9]. Parallel efficiency is defined as: If a speedup of parallel algorithm is $S$, parallel execution threads of nodes is $N$, the parallel efficiency is $S_p=S/N$[10].

(a) Record the time of two kinds of Gaussian filter operations which use technology of parallel computing and don’t use technology of parallel computing, the results are shown in Table 2.

Table 2 the time of Gaussian filter improved and unimproved (ms)

| Gaussian filter | unimproved | improved |
|---|---|---|
| time | 67.28 | 16.45 |

(b) Record the time of three kinds of edge contour extraction algorithm, including unimproved algorithm, threshold segmentation algorithm only and the algorithms that threshold segmentation and parallel computing are using at the same time. The results are shown in Table 3.

Table 3 the time of edge extraction improved and unimproved (ms)

| edge extraction | unimproved | threshold segmentation | threshold segmentation and parallel computing |
|---|---|---|---|
| time | 5719.23 | 645.26 | 145.47 |

By experimental data in Table 2 and Table 3 Comparison following conclusions:

(a) The time of Gaussian filter reduce from 67.28ms to 16.45ms by improved. By analyzing the time of technology which only take threshold segmentation and the time of technology both take threshold segmentation and parallel computing, the time of parallel technology dropped from 645.26 ms to145.47msThe speedup of Gaussian filtering and edge extraction is 4, and then the parallel efficiency of Gaussian filtering and edge extraction is 1.

(b) The time of edge contour extraction, which threshold segmentation firstly, decrease from the original 5719.23ms to 645.26ms. Fall time of this method depends on the magnitude of the number of points near the edge of the edge and threshold segmentation extracted.
(c) Through improving Canny algorithm, computing time from the original 5786.61ms reduced to 161.92ms.

## 6. Conclusion

Using improved Otsu algorithm extract keyhole's edge points and near, can greatly reduce the amount of computation Canny operator. On the basis of multi-core technology, using parallel technology on Gaussian filtering and contour extraction algorithm in Canny operator, significantly reduced the time Canny operator image processing to improve the operating speed. Make full use of Canny operator edge extraction accuracy advantages while overcoming the disadvantages of Canny operator consumes a long time. The improved Canny operator meet time requirements of the measurement system.

## Acknowledgment

This paper supported by the Graduate Innovation Program of Shanxi (20133095), Shanxi Province Science Foundation（2013011025-1）, the Specialized Research Fund for the Doctoral Program of Higher Education of China (20131420120002)

## References

[1] John Canny. A computation approach to edge detection. *IEEE Trans Patten Analysisand Machine Intelligence.* 1986; 8(11): 679- 697.
[2] Xiangyang X, Enmin S, Lianghai J. Characteristic analysis of threshold based on Otsu criterion. *Acta Electronica Sinica.* 2009; 37(12): 2716-2719.
[3] Belean B, Borda M, LeGal B, et al. *FPGA technology and parallel computing towards automatic microarray image processing.* Telecommunications and Signal Processing (TSP), 2011 34th International Conference on IEEE. 2011: 607-610.
[4] Wang Wen-yuan. Selecting the Optimal Gaussian Filtering Scale via the SNR of Image. *Journal of Electronics & Information Technology.* 2009;10: 2483-2487.
[5] Bingjing M, Bo X, Xiaomei C, et al. *Parallel Computing Rendering In Specific Remote Sensing Image Processing*. Proc. of SPIE. 7850: 785028-1.
[6] Prajapati HB, Vij SK. *Analytical study of parallel and distributed image processing*. Image Information Processing (ICIIP), 2011 International Conference on IEEE. 2011: 1-6.
[7] Kumar S, Pant M, Ray A. *Differential evolution embedded Otsu's method for optimized image thresholding*. Information and Communication Technologies (WICT), 2011 World Congress on IEEE. 2011: 325-329.
[8] XU Xiang-yang, Song En-min, Jin Liang-hai. Characteristic Analysis of Threshold Based on Otsu Criterion. *Acta Electronica Sinica*. 2009;12: 2716-2719.
[9] Tang S, Lee B S, He B. *Speedup for Multi-Level Parallel Computing.* Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International IEEE. 2012: 537-546.
[10] Hao Xiao-yun, Fan Yu-mei. How to Demonstrate Parallel Efficiency with Picture. *Computer Engineering and Applications*. 2003;19: 118-119+232.