Research Article Vol. 3, No. 10 / 15 October 2020 / OSA Continuum 2912

Sub-pixel dimensional measurement algorithm based on intensity integration threshold

MENGLIN CHU,[1,2] WEI HUANG,[1,*] MINGFEI XU,[1] SHUQIANG JIA,[1] XIAOFEI ZHANG,[3] AND YONGNAN LU[2,4]

[1]*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, No. 3888 Dong Nanhu Road, Changchun 130033, China*
[2]*University of Chinese Academy of Sciences, No. 19(A) Yuquan Road, Beijing 100049, China*
[3]*Changchun University of Science and Technology, No. 7089 Weixing Road, Changchun 130022, China*
[4]*Dalian Institute of Chemical Physics, Chinese Academy of Sciences, 457 Zhongshan Road Dalian 116023, China*
[*]*huangw@ciomp.ac.cn*

Abstract: In this paper, we proposed a sub-pixel measurement algorithm based on intensity integration threshold (IIT). The proposed method can localize the sub-pixel edges in an inexpensive way by calculating the integration of the intensity across the edge and finding the point where the integration reaches the threshold. Comparative tests show our method realized better efficiency and robustness in practical applications than other state-of-the-art algorithms.

1. Introduction

Measurement technology based on computer vision has advantages of non-touch and high efficiency. Therefore, it has been widely used in industrial productions [1]. A typical computer vision dimensional measurement system mainly consists of illumination system, telecentric lens and detectors. In practical measurement vision system, backlight illumination can notably enhance the contrast of image, and telecentric lens can avoid measurement errors caused by distortion. Apart from the hardware facilities, whether the dimension of an object can be measured accurately, depends on the accuracy of image processing algorithms. Therefore, to improve accuracy, stability and calculate efficiency of the edge extraction algorithm is still an important task.

Sub-pixel edge detection is of higher accuracy than pixel edge detection in computer vision. As for traditional methods of sub-pixel accuracy dimensional measurement, researchers often use pixel edge detection operators (such as Sobel, Canny and LoG) to extract edge pixels, then calculate sub-pixel edges based on the intensity levels of edge pixels and nearby pixels for dimensional measurement. Therefore, the fundamental pixel-accuracy edge detection plays an important role. When the test image is in high quality, both Sobel and Canny will output smooth and precise edge pixels. Sobel is a simple operator which use only two 3×3 filters without other sophisticated calculations, and it is faster than Canny.

After edge pixel detection, researchers use the intensity levels of pixel points and their neighbor pixels to calculate sub-pixel edges. Most of the existing sub-pixel algorithms can be divided into moment-based methods [2–4], reconstructive methods [5–7] and fitting methods [8–11]. Moment operator has good noise immunity, but it needs large amounts of convolution operation, so moment operators will cost longer time to find sub-pixel edges. The accuracy of reconstructive methods mostly depends on the edge model. Among fitting methods, parabolic fitting is in a relatively high efficiency but it is not precise enough. Erf function fitting is more accurate, but it also consumes more time [12].

Nowadays, Hagara et al. proposed a sub-pixel algorithm based on approximation of the edge with erf function [13], compared it with moment-based edge operator, spatial moments, and

#402101 https://doi.org/10.1364/OSAC.402101
Journal © 2020 Received 9 Jul 2020; revised 5 Oct 2020; accepted 6 Oct 2020; published 15 Oct 2020

OSA CONTINUUM

wavelet transform in 1-D images, showed the superiority their method [12]. Fabijańska et al. proposed a sub-pixel edge detection algorithm reconstructs image gradient function at the edge using the Gaussian function [14]. Flasia et. al. proposed a method for subpixel straight lines detection using a version of the Savitzky-Golay filter and determined subpixel edge location by fitting a Gaussian function to orthogonal sections of the coarse edge image [15]. Von Gioi et al. proposed a sub-pixel edge detecting algorithm which combines Canny operator and Devernay sub-pixel correction and chained the obtained edge points [16]. Yang et al. proposed a sub-pixel measurement system of circle outer diameter based on Zernike moment and quadratic polynomial interpolation [17]. Seo et al. presents a subpixel edge localization method based on the adaptive weighting of gradients (AWG) and a method that uses the squared weighting of gradients (SWG), then combines the SWG and AWG selectively to obtain the best localization [18].

In this paper, we proposed a fast and reliable dimensional measurement algorithm in sub-pixel accuracy which is different from the existing algorithms. Our method firstly uses Sobel operator to obtain edge pixels from the test image. Then, it extracts a sequence of pixels across each edge pixel. Finally, along the sequence of pixels, we calculate the indefinite integral of the pixels' intensity and the subpixel edge is located where the integral of the pixels reaches the threshold. The threshold is often determined by erf-function model or physical photo calibrating experiments. In the experiments, we use a vision system to get pictures of the test objects, compare other algorithms with ours in edge detection and dimensional measurement, and proved that our algorithm can effectively realize high accuracy in sub-pixel edge detection and dimensional measurement.

This paper is organized as follows. Section 2 introduced Sobel operator and our intensity integration threshold method. In Section 3, we illustrated our vision system, and carried out experiments using test images we obtained from the vision system to compare the edge extraction efficiency and accuracy of Sobel operator, Sobel operator + Zernike moment, Sobel operator + erf function fitting and the method we proposed in this paper. We also evaluated the repeatability accuracy of our algorithm comparing with Sobel operator + erf function fitting. According to experimental results, our algorithm shows high precision and great efficiency. Finally, Section 4 concludes this paper.

2. Sub-pixel edge detection

2.1. Sobel operator

Considering both efficiency and accuracy, we choose Sobel operator to extract edges in pixel accuracy. Sobel operator use two edge detection operators:

Horizontal operator:

$$\begin{aligned} G_x = &-f(x-1,y-1) - 2f(x,y-1) - f(x+1,y-1) + f(x-1,y+1) \\ &+2f(x,y-1) + f(x+1,y-1) \end{aligned} \tag{1}$$

Vertical operator:

$$\begin{aligned} G_y = &-f(x-1,y-1) + f(x+1,y-1) - 2f(x-1,y) + 2f(x+1,y) \\ &-f(x-1,y+1) + f(x+1,y+1) \end{aligned} \tag{2}$$

to perform convolution with the whole picture, as shown in Fig. 1 (a),(b). The output intensity level of each pixel is

$$G = \sqrt{G_x + G_y} \tag{3}$$

OSA CONTINUUM

If G is greater than the threshold, this pixel can be classified as edge pixel. Meanwhile, the gradient direction of intensity levels in a point can be calculated as

$$\theta = \arctan(\frac{G_x}{G_y}) \tag{4}$$

2.2. Intensity integration threshold algorithm

Firstly, we need to determine the gradient direction of the edge pixels.

As for one edge pixel $f(x, y)$, we get its 3×3 neighbors, and perform convolution using the horizontal and vertical operators mentioned above. If $G_x(x, y) > G_y(x, y)$, this point can be defined as horizontal edge point, we then extract points along the direction where the intensity levels change along, that is the vertical direction. If $G_y(x, y) > G_x(x, y)$, this point can be defined as vertical edge point, we then extract points along the horizontal direction.

Along the direction where the intensity levels change, we extract the intensity of nearby points. Set I as the intensity and l as the location. Figures 2(a) and 2(b) are the $I - L$ figures of the simulated continuous edge and the dispersed edge. The area of the gray part is the integration we require.

(a)

-1	-2	-1
0	0	0
1	2	1

(b)

-1	0	1
-2	0	2
-1	0	1

Fig. 1. Sobel operator (a) Horizontal (b) Vertical

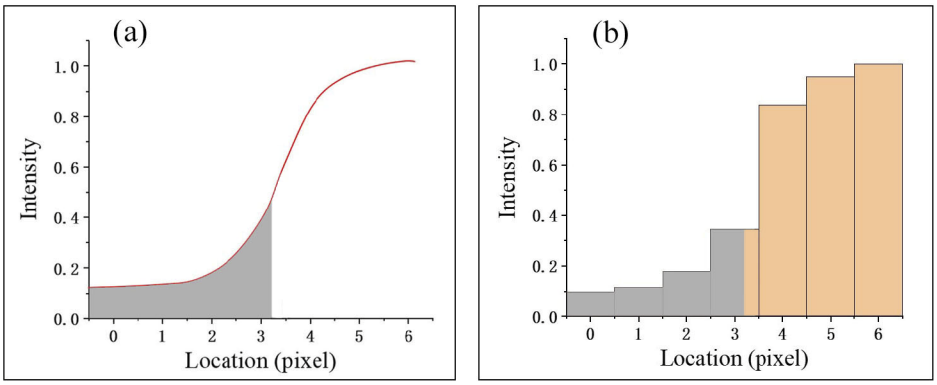

Fig. 2. (a) $I - L$ figure of the simulated continuous edge (b) $I - L$ figure of the dispersed edge

Then, the intensity integration threshold k needs to be determined. k can be determined by calibration experiments. After determining k, the pixel intensity values are integrated from left to right. When the integrated value reaches the threshold, here the position t is the edge position of the sub-pixel to be measured as showed in Eq. (1).

$$\int_0^t I(l)dl = k \tag{5}$$

OSA CONTINUUM

In the actual image, although the pixel intensity can be fitted as a continuous curve [as shown in the Fig. 2(a)], considering the efficiency, we still use discrete intensity values for integration [as shown in the Fig. 2(b)]. In the calculating process, we simplify the integration to summation.

At first, arrange the intensity from small to large. If a total of N points are taken, a series of intensity values $I_1 \sim I_N$ are obtained. Then, calculate the intensity sum E_ns of the first n items, which is equivalent to integrating the intensity of the first n points.

$$E_n = \sum_{i=1}^{n} I_i, n = 1, 2, \ldots, N \tag{6}$$

Find the E_n,E_{n+1} closest to the threshold, so that

$$E_n < k \cap E_{n+1} > k \tag{7}$$

The sub-pixel edge position l coordinates where the integral just reaches the threshold.

$$l = \frac{k - E_n}{I_n} + n \tag{8}$$

The flexibility of IIT algorithm is that the number of points N can be determined according to the blur of the tested edge, and the threshold can be changed according to different lighting conditions to achieve more accurate sub-pixel edge detection results.

In this paper, to realize best result apply for our vision system, we set the original point as the center, and get its nearest 6 points for calculating the intensity integration, which means N=7. The threshold k=0.9 is determined by the calibration process in 3.2.

3. Comparative experiments and accuracy evaluation

3.1. Vision system and the relationship between actual dimensions and pixels

In this part, we make a brief introduction of the vision system we use in this paper.

Our vision system is mainly based on backlight illumination, bi-telecentric lens and CMOS camera, as shown in Figs. 3 and 4. The outgoing light of the illumination is parallel light, and part of the light is blocked by the test object. The bi-telecentric lens can be simplified as shown in Fig. 4. The incident light and the outgoing light of the bi-telecentric lens are both parallel lights. Consequently, a clear shade formed by test object is projected on the CMOS sensor.

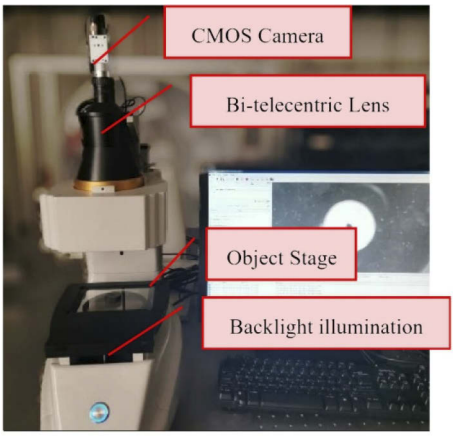

Fig. 3. Experiment device

OSA CONTINUUM

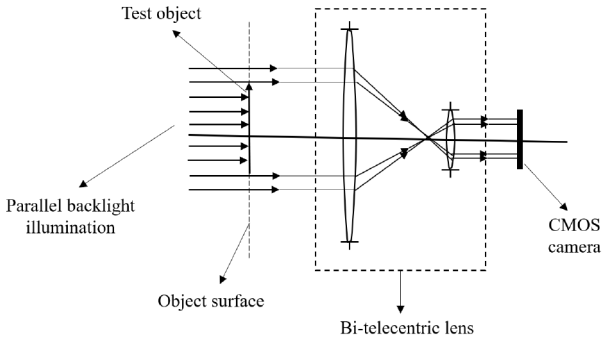

Fig. 4. Schematic diagram of optical path

In our vision system, the magnification of the lens $\beta = -0.11$. Distance between the bottom of the test object and the first face of the lens is 80mm. The resolution of the CMOS is $2448px \times 2048px$, the dimension of each pixel is $3.45\mu m \times 3.45\mu m$.

In the following tests, we use MATLAB on the computer of 8 GB RAM, Intel Core i5-8250U CPU.

The process of edge detection and dimensional measurement in this part can be described as follows (Fig. 5):

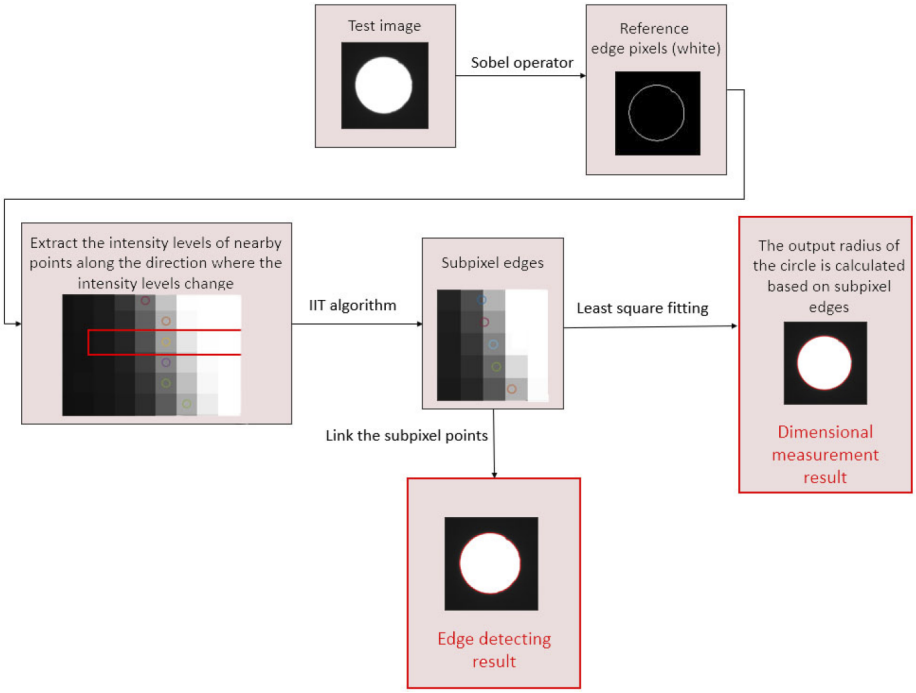

Fig. 5. The flow chart of our sub-pixel measurement algorithm

OSA CONTINUUM

3.2. Calibration

The edge position sought by this algorithm depends largely on the integration threshold. For a machine vision measurement system, if it can locate the edge accurately, then for any object of known size, the exact size can be obtained based on the calibration result.

In this part of the experiment, we selected two objects to be measured: the standard rectangular gauge [Fig. 6(a)] and the inner diameter ring gauge [Fig. 6(b)]. The width of the standard rectangular gauge is 40mm. The inner diameter of the ring gauge is 3.003mm. the accuracy of both objects is 0.001mm. During the process of obtaining test pictures, the lighting conditions remain stable. As shown in Fig. 6, the two objects have such characteristics: the measured shape of the rectangle standard gauge is a straight edge, and the test area is opaque; the shape we measure for the inner diameter ring gauge is the inner diameter [Fig. 7(b)] and the measured area is transparent [Fig. 7(a)].

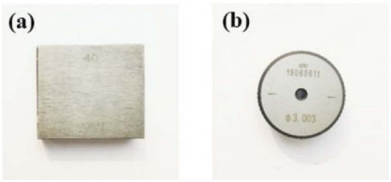

Fig. 6. The measured objects (a) rectangle standard gauge (b) inner diameter gauge

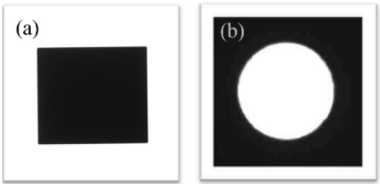

Fig. 7. (a) Test image of rectangle standard gauge (b) Test image of inner diameter ring gauge

Set $q = \frac{pixel\ number}{size}$ as the relationship between pixels and the actual size of a test object. The pixel number/size of the rectangle standard gauge is q_1, the pixel number/size of the inner diameter ring gauge is q_2. Theoretically, if the selected threshold is appropriate, $q_1 = q_2$. If the selected threshold is too low, the tested light-transmitting area is increased, and the tested opaque area is reduced, which will cause $\frac{q_1}{q_2}$ to be less than 1; otherwise, the ratio will be greater than 1. Therefore, here we set up an experiment, taking 8 photos of each object under test (Fig. 8), each time using a different threshold, and using the IIT algorithm to obtain the average dimension. Then we calculate and list the q of the two test objects, and get the ratio $\frac{q_1}{q_2}$ listed in the table below (Table 1).

As can be seen from Table 1, the experiment conforms our theoretical prediction of the change of q with threshold k, and we have obtained the threshold value when $q_1 = q_2$: k=0.9, and obtained the calibration result: 1pix corresponds to 0.0389mm. In addition, it can be seen that the value of the rectangle standard gauge is very stable, because its size is relatively large(40mm), and the slight change of the edge has little effect on the calibration result. However, the value of the inner diameter ring gauge changes more drastically, because its size is relatively small(3.003mm), and the change of the edge position will have a very large impact on the dimensional measurement.

In this part, as a comparison, we introduce erf function fitting algorithm to perform the same edge detection and dimensional measurement on the two tested objects. Erf function

OSA CONTINUUM

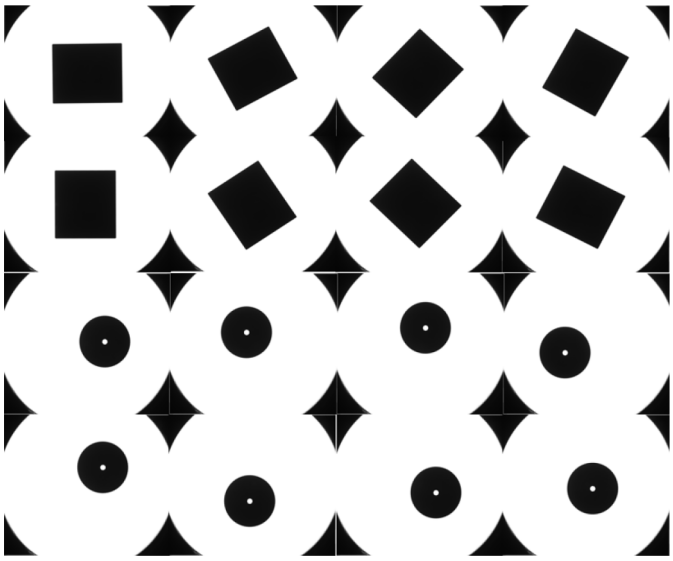

Fig. 8. The measured pictures taken on the machine vision measurement device

Table 1. Calibration based on intensity integration threshold

Threshold k	q of rectangle standard gauge (q1)	q of inner diameter ring gauge(q2)	$\frac{q1}{q2}$
0.1	0.0390	0.0379	1.0290
0.2	0.0390	0.0380	1.0263
0.3	0.0389	0.0381	1.0236
0.4	0.0389	0.0382	1.0209
0. 5	0.0389	0.0382	1.0209
0.6	0.0389	0.0383	1.0183
0.7	0.0389	0.0385	1.0130
0.8	0.0389	0.0387	1.0078
0.9	0.0389	0.0389	1.0000
1.0	0.0389	0.0391	0.9974
1.1	0.0388	0.0394	0.9898
1.2	0.0388	0.0396	0.9848
1.3	0.0388	0.0398	0.9799
1.4	0.0388	0.0400	0.975
1.5	0.0388	0.0402	0.9701
1.6	0.0388	0.0403	0.9677
1.7	0.0388	0.0404	0.9653
1.8	0.0387	0.0405	0.9630

fitting method is a sub-pixel edge detection algorithm widely used in machine vision and image processing with high accuracy and great robustness. An important point is that the principle of erf function fitting is to take the position where the gray level changes the most as the edge point. This feature determines that the erf function fitting algorithm cannot be calibrated by modifying a threshold like the IIT method. Therefore, here we merely measured the average dimension

OSA CONTINUUM

pixels of the two test objects using the 8 images each, and obtain the q values corresponding to the actual size for comparison. The result is listed in Table 2.

Table 2. Calibration test of Erf function fitting method

	Actual dimension	Average pixels	$q = \frac{pixel\ number}{size}$
Rectangle standard gauge	40.000mm	1028.7	0.0389
inner diameter ring gauge	3.003mm	77.7904	0.0386

When using the erf function fitting algorithm to measure the edges of two different objects, the measured value of $q = \frac{pixel\ number}{size}$ is different. This shows that the erf function fitting algorithm needs more corrections to ensure the consistency of dimensional measurement for objects of different sizes and different types of edges.

3.3. Accuracy evaluation

The error source of edge detection and size measurement in machine vision mainly lies in the image acquisition process: the resolution of the detector is limited, and the true edge of the object cannot be reproduced completely and smoothly on the test pictures. In a machine vision system, the edges of objects in the image always present discrete grayscale changes and oscillatory effect. In the context of dimensional measurement, this discrete gray-scale change and oscillatory effect will magnify the error caused by the object displayed at different positions in the view field, thus affects the repeatability accuracy. In addition, lens distortion and adjustment errors, unevenness of the light source, and lack of flatness of the stage can also cause errors in edge detection and dimensional measurement. Since the vision measurement device in this article has been calibrated, the error generated by the hardware facility can be ignored within the central field of view. At this time, most of the errors come from the discrete grayscale changes and the oscillatory effect. A high-precision edge detection and dimensional measurement algorithm can greatly smoothen the oscillatory effect, make the detected edge closer to the actual contour of the object, and eliminate the error caused by the movement of the object in the field of view as much as possible. Therefore, in this article, the standard for evaluating the accuracy of the algorithm is mainly the repeatability accuracy. Based on this, we designed the following accuracy experiments.

3.3.1. Experiments on inner diameter gauge

For each ROI in the 8 test pictures taken from the inner diameter gauge [Fig. 7(b)], we use both Sobel + erf function fitting and our IIT algorithm to obtain sub-pixel edges and least-square fit the edge points into circles. We obtain the radius (in pixels), and calculate the errors compare to the average. The test results are shown in Table 3 and Table 4.

Table 3. Radius values (in pixels) of the inner diameter ring gauge in different positions

	Radius (in pixels)							
Sobel + erf function fitting	77.8060	77.9308	77.7498	77.9667	77.8507	77.6944	77.6667	77.6582
Errors	0.0156	0.1404	-0.0406	0.1763	0.0603	-0.0960	-0.1237	-0.1322
Sobel + Intensity integration threshold	77.0259	77.2118	77.2296	77.1775	77.3329	77.0670	76.9603	77.1614
Errors	-0.1199	0.0660	0.0838	0.0317	0.1871	-0.0788	-0.1855	0.0156

According to Table 4, the MSE of the proposed algorithm is very close to that of the erf function algorithm. However, the dimensional measurement result of the algorithm in this paper is closer to the actual dimension 3.003mm.

OSA CONTINUUM

Table 4. Average radius and MSE (in pixels) of the inner diameter ring gauge

	Average pixels	Corresponding dimension	MSE
Sobel + erf function fitting	77.7904	3.0260mm	0.0123
Sobel + Intensity integration threshold	77.1458	3.0001mm	0.0128

3.3.2. Experiments on more complicated situations

In this part, we use 4 different algorithms (Sobel operator, Sobel operator + Zernike moment, Sobel operator + erf function fitting and Sobel operator + IIT algorithm) to detect edges in more complicated situations, and calculated the MSEs between the fitted shapes and the detected edges.

Figure 9(a) is a picture of six identical electronic components. Figure 10 is a picture of the backplane of the cameras on a cellphone. Both of the test pictures are taken by the measurement device in this paper.

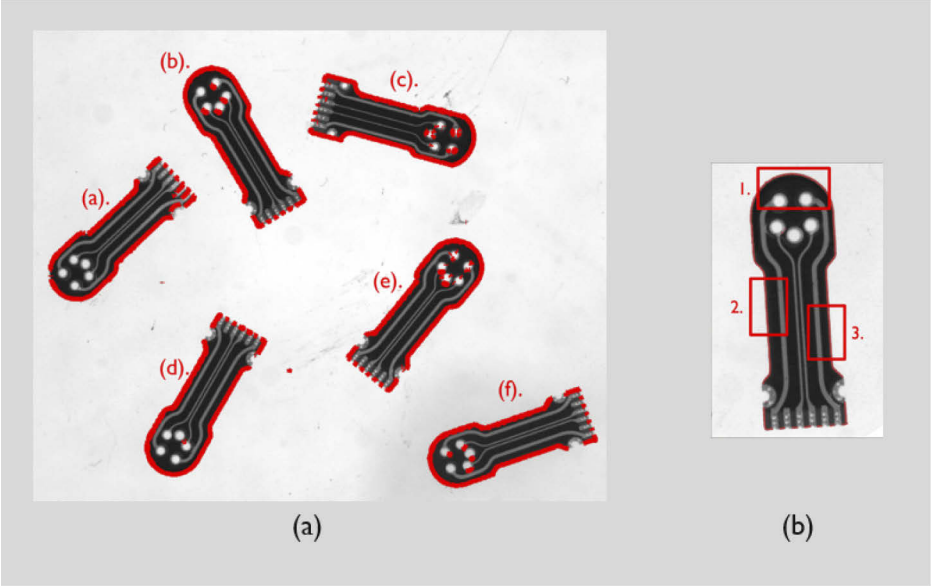

Fig. 9. Test pictures of electronic components (a) six test objects. (b) ROIs of each test object.

For the picture of electronic components, the target objects are marked as (a)-(f) in Fig. 9(a), and the ROIs (regions of interest) are marked as (1)-(3) in Fig. 9(b). The MSEs between the fitted edges and the detected edges are listed in Table 5.

For the picture of the backplane of a cellphone, the ROIs are marked as (h)-(n) in Fig. 10. The MSEs between fitted shapes and the detected edges are listed in Table 6. Figure 11 is the upper right corner of the small circle (n) and the edges detected with 4 different algorithms.

According to the MSEs in Table 5, IIT together with Erf function fitting performs better than Sobel and Sobel + Zernike moment. The proposed IIT algorithm performs better than erf function fitting in detecting circles in Fig. 9, but it is not as good as erf function fitting in detecting straight lines. On the situation in Table 6, the IIT algorithm performs better than Erf function fitting in (h), (i), (j), (l) and (m).

Figure 6 helps illustrating the difference between the test algorithms. The green curves are the fitted outline and the red curves are the edge detected by algorithms. Sobel operator performs well in Fig. 6(a), but it can only extract pixel-accuracy edges. As Zernike moment method

OSA CONTINUUM

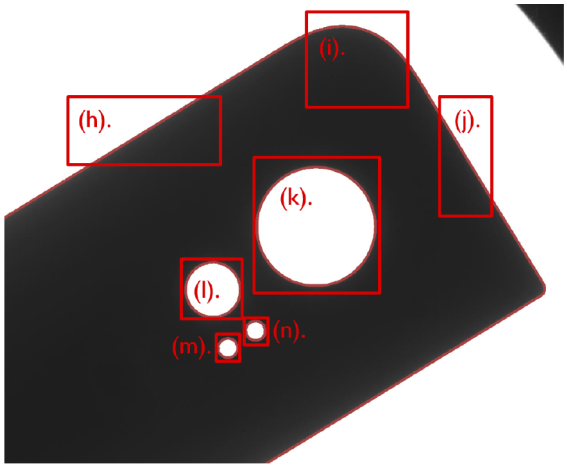

Fig. 10. Test pictures of the backplane of a cellphone

Table 5. MSEs (in pixels) of the electronic components

Algorithm	Sobel			Sobel + Zernike moment			Erf function fitting			IIT		
MSE	1	2	3	1	2	3	1	2	3	1	2	3
(a)	0.2744	0.0744	0.0643	0.1857	0.0571	0.0798	0.0357	0.0252	0.0156	0.0557	0.0384	0.0258
(b)	0.1432	0.0431	0.0578	0.2055	0.0527	0.0920	0.0904	0.0265	0.0358	0.0599	0.0135	0.0466
(c)	0.1452	0.0181	0.1036	0.2592	0.0087	0.2137	0.1197	0.0446	0.0395	0.0763	0.0372	0.0496
(d)	0.1281	0.0170	0.0259	0.1725	0.0320	0.0521	0.0906	0.0362	0.0152	0.0521	0.0168	0.0383
(e)	0.1328	0.0639	0.0576	0.1927	0.0760	0.0663	0.0551	0.0198	0.0335	0.0677	0.0615	0.0298
(f)	0.1256	0.0567	0.0771	0.2461	0.2248	0.0933	0.0717	0.0517	0.0391	0.0647	0.0545	0.0538
Average	0.1582	0.0454	0.0644	0.2103	0.0752	0.0995	0.0772	0.0340	0.0296	0.0627	0.0370	0.0407

Table 6. MSEs (in pixels) of the backplane of a cellphone

MSE	Sobel	Zernike moment	Erf function fitting	IIT
(h)	0.0514	0.0638	0.0283	0.0258
(i)	0.0951	0.1307	0.0799	0.0476
(j)	0.0456	0.0637	0.0322	0.0232
(k)	0.1611	0.1810	0.0805	0.1164
(l)	0.0990	0.1025	0.0991	0.0523
(m)	0.1247	0.2029	0.0812	0.0471
(n)	0.0769	0.1388	0.0128	0.0222

OSA CONTINUUM

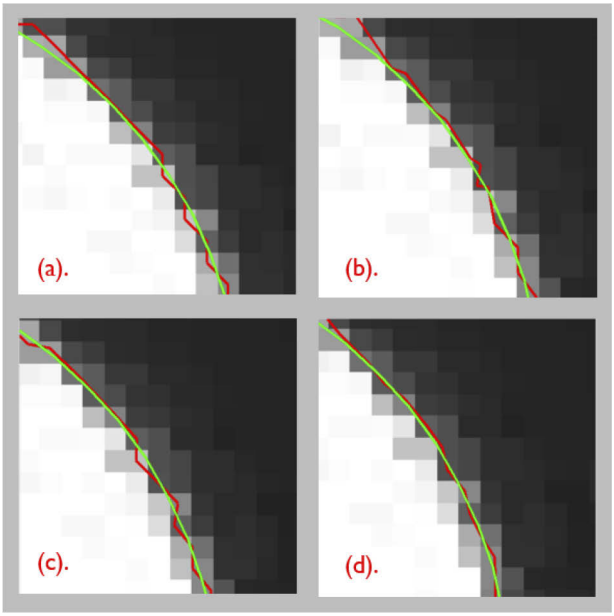

Fig. 11. the upper right corner of the small circle (n). (a). Sobel operator (b). Zernike moment (c).erf function fitting (d). IIT

is taking more account of the neighbor intensity, it tends to extract an edge with more local characteristics. Therefore, the oscillatory effect cause by the discrete detector pixels has been tested through it according to Fig. 6(b). Erf function fitting is better than Sobel method. The IIT algorithm shows a smooth outline and therefore verified the data in Table 6.

In general, the error level of the IIT algorithm when detecting edges is equivalent to that of the erf method. But according to the content in 3.2 and 3.3.1, the advantage of the IIT algorithm lies in its accuracy and the consistency of detection of inner and outer edges. And in 3.4, it also showed better efficiency than the erf method.

3.4. Efficiency Evaluation

The test image is the picture of an inner diameter ring gauge. The size of the test picture is $820px \times 820px$ (Fig. 12). In this part, we use 4 algorithms (Sobel operator, Sobel operator + Zernike moment, Sobel operator + erf function fitting and Sobel operator + IIT algorithm) to extract edges, then we compared the time they spend. Due to the instability of time consuming in the first few times of running a program in MATLAB, the efficiency results we list in Table 1 come from the stable situation after running the program several times. Before every efficiency test, we cleared all data. We collect 4 time consuming results for each algorithm.

Fig. 12. The picture used in efficiency evaluation ($820px \times 820px$)

OSA CONTINUUM

According to Table 7, although only use Sobel operator have high computational efficiency, it can't locate edge in sub-pixel accuracy and it is difficult to meet the requirements of high-precision measurement. Zernike moment cost more time than Sobel operator. The erf function method has high accuracy, but the fitting process is more complicated and takes a long time. The IIT algorithm proposed in this paper has easy implementation, high accuracy, and relatively high computational efficiency as it only taking 46.58% of the time of erf function fitting method.

Table 7. Efficiency evaluation

Algorithm	Sobel	Sobel + Zernike moment	Sobel+ erf function fitting	Sobel + Intensity integration threshold
Time spent/seconds	0.021630	0.185399	1.892403	0.896478
	0.022248	0.188130	1.944752	0.846993
	0.020890	0.188375	1.851850	0.861300
	0.022535	0.181659	1.807678	0.886847
Average time spent	0.021826	0.185641	1.874171	0.872905

4. Conclusions

In this paper, we proposed a dimensional measurement algorithm based on intensity integration threshold (IIT). This algorithm can extract edges in sub-pixel accuracy efficiently, and obtain precise dimensions by fitting the edges into geometric shapes. Compare to the erf function fitting method, our algorithm greatly increases the efficiency (53.42%) without reducing the detection accuracy. It is a practical, relatively cheap and efficient edge detection and dimensional measurement algorithm which can meet the actual needs of high-precision dimensional measurement.

Funding

Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences;State Key Laboratory of Applied Optics.

Acknowledgment

The authors thank the State Key Laboratory of Applied Optics for supporting this work.

Disclosures

The authors declare no conflicts of interest.

References

1. C. Connolly, "Machine vision advances and applications," Assembly Autom. **29**(2), 106–111 (2009).
2. A. J. Tabatabai and O. R. Mitchell, "Edge location to subpixel values in digital imagery," IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-6**(2), 188–201 (1984).
3. E. P. Lyvers, O. R. Mitchell, M. L. Akey, and A. P. Reeves, "Subpixel measurements using a moment-based edge operator," IEEE Trans. Pattern Anal. Mach. Intell. **11**(12), 1293–1309 (1989).
4. S. Ghosal and R. Mehrotra, "Orthogonal moment operators for subpixel edge detection," Pattern Recognit. **26**(2), 295–306 (1993).
5. P. Rockett, "The accuracy of sub-pixel localization in the Canny edge detector," in *Proc. British Machine Vision Conf. Nottingham*, UK, (1999).
6. D. G. Bailey, "Sub-pixel profiling," in *Proc. 5th Int. Conf. Information Communications and Signal Processing*, Bangkok, Thailand, (2005), pp. 1311–1315.
7. T. Hermosilla, E. Bermejo, A. Balaguer, and L. A. Ruiz, "Non-linear fourth-order image interpolation for subpixel edge detection and localization," Image Vis. Comput. **26**(9), 1240–1248 (2008).

OSA CONTINUUM

8. F. Devernay, "A non-maxima suppression method for edge detection with sub-pixel accuracy," in INRIA Research Report, (1995).
9. Y. Zheng, W. D. Qian, J. Luo, and S. F. Zhao, "Study on sub-pixel edge location method based on curve fitting," Optical Technique **33**, 386–389 (2007).
10. R. L. B. Breder, V. V. Estrela, and J. T. D. Assis, "Sub-Pixel Accuracy Edge Fitting by Means of B-Spline," in *IEEE International Workshop on Multimedia Signal Processing*, (Rio de Janeiro, BRAZIL, 2009), pp. 467–471.
11. G. S. Xu, "Sub-pixel edge detection based on curve fitting," in *Second International Conference on Information and Computing Science*, (Manchester, UK, 2009), pp. 373–375.
12. M. Hagara and O. Ondráček, "Comparison of Methods for Edge Detection with Sub-pixel Accuracy in 1-D Images," in *3rd Mediterranean Conference on Embedded Computing*, (2014).
13. M. Hagara and P. Kulla, "Edge Detection with Sub-pixel Accuracy Based on Approximation of Edge with Erf Function," Radioengineering **20**, 516–524 (2011).
14. A. Fabijańska, "Gaussian-Based Approach to Subpixel Detection of Blurred and Unsharp Edges," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, 2014), 641–650.
15. A. G. Flesia, G. Ames, G. Bergues, L. Canali, and C. Schurrer, "Sub-pixel straight lines detection for measuring through machine vision," in *IEEE International Instrumentation and Measurement Technology Conference*, 2014), 402–406.
16. R. G. V. Gioi and G. Randall, "A Sub-Pixel Edge Detector: an Implementation of the Canny/Devernay Algorithm," Image Process. On Line **7**, 347–372 (2017).
17. S. Yang, Y. Wang, and H. Guo, "Sub-pixel Measurement System of Circle Outer Diameter Based on Zernike Moment," in *International Conference on Artificial Intelligence and Engineering Applications*, (2017), pp. 894–903.
18. S. Seo, "Subpixel Edge Localization Based on Adaptive Weighting of Gradients," IEEE Trans. on Image Process. **27**(11), 5501–5513 (2018).