Fast Feature Extraction Method for Faults Detection System

Hongmin Wang, Xiaohui Zhu(✉), Xiangyong Niu, and Ping Xue

School of Automation, Harbin University of Science and Technology, Harbin 150080, China
l2l0158334@qq.com

Abstract. The feature extraction based on machine learning is significant in the detection system. The boundary information, the circumference and the area are the essential features in the identification and the classification of flaws. In order to get those information, this paper proposed a novel algorithm to get the boundary information using the boundary tracking, and to make each flaw independent by establishing a balanced binary search tree for data storage. By scanning the image and the image boundaries based on binarization transformation, there is no need to fill the region, nor need to use the chain code to count the number of regions and the boundary information. According to the established balanced binary search tree, we can calculate the number of the pixel of the area of each fault, the edge information of the boundary, and the circumference. The algorithm has the advantages of fast speed, less computation, better noise suppression and accurate results.

Keywords: Image processing · Boundary tracking · Freeman chain code Balanced binary search tree

1 Introduction

During the manufacturing process of lens, lots of flaws accompanied such as point flaw, plume, scratches, air bubbles etc. At present, our country mainly adopts artificial detection methods. The use of artificial detection wastes a lot of labor, which is based on the experience of quality judgment and fault classification. By the influence of the difference of worker's personal status and experience, the result is subjective and can not be standardized. With the rapid development of national economy, people pay more attention to highly requirement for the quality of optical lenses, and the demand for the lens is becoming bigger and bigger. It's irreversible to realize the automatic detection and classification of flaw for optical lens. In the detection system of lens based on machine vision, it's an essential part of the whole system to obtain feature from the image after binarization processing [1]. SIFI(Scale-invariant feature transform) is a commonly used descriptor as the local characteristics. The SIFT algorithm [2] proposed by LOWE is to find the feature point on different spatial scales, and to calculate the direction of the key point. Those point will not be changed by the light intensity or the affine transformation. To an extent, the SIFI algorithm can solve the problems such as the affine transformation, the projection transformation [3–6], and the target occlusion.

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 297–306, 2018.
https://doi.org/10.1007/978-3-319-73317-3_35

SIFI algorithm is stable. However, due to the massive detection of feature point, it's too complicated and dissipative. SURF (Speeded Up Robust Feature) is more efficient and simple to operate than SIFI. Due to the variable shape of the fault for the lens, both feature descriptor mentioned above are not suitable for the feature extraction of faults for detection system. In a similar way, commonly used Harris corner detection operator and CSS (Curvature Scale Space) corner detection operator do not apply to optical lens fault detection system. Realizing the automation of optical lens fault detection is a process of constantly learning from the artificial detection. According to the quality of the optical lens classification standard, the boundary of the fault information, the area and the perimeter are the main basis of fault detection and classification. Those characteristics can be the input of the neural network and SVM (Support Vector Machine) to learn to identification and classification [7].

2 Edge Following Algorithm Based on Balanced Binary Search Tree

Due to the changeable shape of the optical lens, different kinds and different level of the flaws have different influence on the optical lenses. Taking an example of the scratch, we need the area, perimeter features and all the boundary point to fit the line. In order to obtain these characteristics, there are two commonly used solution as follows.

Plan a: First, detect each edge of image and fill the internal. Second, traverse images, and label each connected domain. In order to label the domain correctly, regard the first point of the traversed domain as the starting point. The region is labeled in a boundary tracking manner until the region is fully labeled. Starting scanning from the beginning of the mark to find the next starting point, unless the whole image is processed.

From this scheme, the digital image after processing, the edges and the internal of each individual fault have the same gray value, and each individual fault's gray value is different. It means that one byte used to describe a digital image pixel can only describe 256 faults. If the number of the faults are more than 256, the more byte are needed to describe a digital image pixel, which causes a waste of space. When to obtain the perimeter or the information of a flaw's boundary, the traverse is needed, which causes the inefficiency.

Plan b: Boundary tracing and described by the chain code.

Chain code is a coded representation of boundary point, using a specific direction and length of links. Scan in the order of the bottom-up, from left to right. Find the edge point named a as a starting point for the edge tracking, and mark it as an already tracking pixels. If another unmarked boundary point b is found, then update the current point to b. Continue to track the edge from point b, until all points are marked [8–12].

From this scheme, it can successfully help track the closed area of the border generally. However, when boundary adhesion appears, lots of the edge points will be missed. If the chain code is to describe the boundary, only the starting point of the

boundary points is in absolute coordinates, and the offset of other points is represented by the related direction. If all the boundary points of a fault are needed, it needs to restore the original image by the chain code. Each boundary point needs to be determined according to the location of last point, which causes the edge information processing more complex and inefficiency.

In order to solve the problems above and to improve efficiency, the third plan is proposed here.

Plan c: Boundary tracking method is to scan the image in the order of bottom-up, from left to right. Each flaw is a individual collection with a balanced binary search tree. In order to facilitate the follow-up calculation of the area and the perimeter, the multilevel nested balanced binary search tree is adopted. The map in the STL (Standard Template Library) of *c* plus plus is referenced to explain. Element of the map appears as a pair with real value and the key value at the same time [13, 14]. In the first layer of the balanced binary search tree, the number of lines *i* starts from zero as the key value. The independent bank of all the boundary points where *i* is set in array is the real value. The second floor are unique numbers as key values. To peer number *i*, there is a corresponding balanced binary search tree as a real value. The tree structure is shown in Fig. 1.

According to the algorithm shown in Fig. 2, the steps of the algorithm are presented as follows:

Step1: Find the edge point from lower left boundary points, current point, as a starting point named s in the border. Mark the starting points. At the same time, put the point into the established balanced binary tree in the first layer.
Step2: Scan from the current point along the scanning direction, as shown in Fig. 3. The default direction is 0. The tracking principle is: If the tracking point is the boundary point, update the point to the new current point, and put the point into the first layer of the established tree. At the same time, change the direction 90° counterclockwise. If the tracking point is not a boundary point, the scan direction will be modified the 45° clockwise.
Step3: Continue to track the boundary from the current boundary point, and repeat the step 2 until the current boundary point backs to the start point *s*. Set direction 0, and continue to scan the new boundary.

This scheme can make up for both of these scenarios. After a border tracking, all the boundary points are classified by each individual defect. Later, when the boundary information points are extracted, the area and the perimeter are calculated. Only extracted points are needed to scan, without scanning the entire digital image. This improves the efficiency of time and space. Figure 4 is the original image of the scratch section. Figure 5 shows the image after the Fig. 4 traced edge by the plan *c*. And then we set the stage for the next calculation. In Fig. 4, under the condition of the boundary of a fault information, only three faults in the balanced binary search tree are going to be scanned, and you can get all the boundary of the fault information and feature, instead of scanning the whole image.

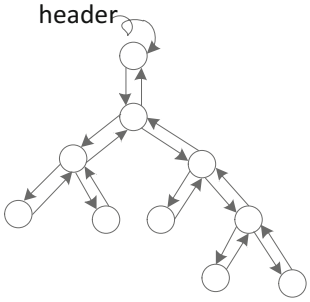


Fig. 1. Structure of tree

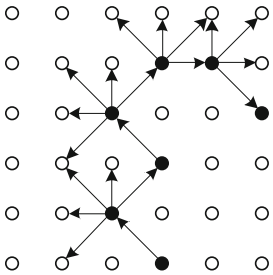


Fig. 2. Algorithm diagram

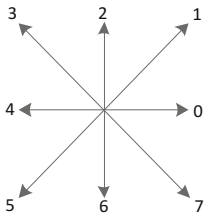


Fig. 3. 8-direction

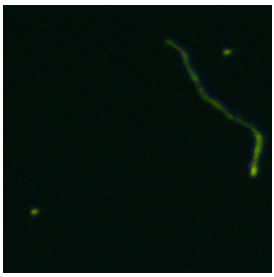


Fig. 4. The original picture of the scratch

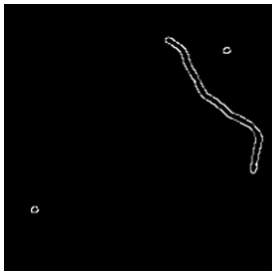


Fig. 5. The post-processing portion

3 Perimeter Calculation Based on Balanced Binary Search Tree

The length of the closed area connected by a common boundary point center or the sum of regional boundary points can be regarded as the perimeter.

3.1 Kinds of Common Algorithm to Calculate the Perimeter

(1) Using the 8 - direction chain code which is shown in Fig. 6 direction chain code to calculate the perimeter. The serial number in the chain code is even when the number is even code, odd number when odd code. The distance to the odd code is $\sqrt{2}$. The distance to the even code is 1. According to even code number, the odd number of code, its formula is as follows:

$$N = n_o + \sqrt{2}n_e. \tag{1}$$

N is the perimeter. n_e is the number of the even number in code, n_o is the odd number.

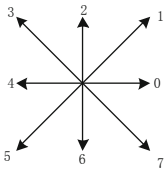


Fig. 6. Direction chain code

(2) Use Euclidean distance formula to calculate the perimeter, two points (x_1, y_1) and (x_2, y_2), its formula is as follows:

$$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2}$$

L is the perimeter between the two points.

3.2 Improved Algorithm to Calculate the Circumference

When boundary adhesion appears, tracking by traditional boundary tracking method based on chain code, a lot of the edge points will be missed. So Freeman Chain Code is abandoned here. Using Euclidean distance formula to calculate the circumference is complex, time-consuming and space-consuming. So it's not a good choice here. The improved algorithm is based on the balanced binary search tree, which has storied all the boundary point. The number of the corresponding pixels where the line number is i is ni. The total number of boundary points is as follows:

$$L = \sum_{i=0}^{i=N} n_i \tag{3}$$

L is the perimeter of the boundary points.

4 Area Calculation Based on Balanced Binary Search Tree Area

The sum points of the boundary and its total pixels within the element or the area of the integral surrounded by a closed curve can be the area wanted.

4.1 Kinds of Common Algorithm to Calculate the Area [15, 16]

(1) Pixel accumulative method. Calculating area often counts the boundary points and its total pixels within the element. The formula is as follows:

$$S = \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} f(x, y) \tag{4}$$

S is the area wanted. n is line number and m is the row number. $f(x,y)$ gives the gray value of the point (x,y).

(2) On the x-y plane. Using the Green formula to calculate area surrounded by a closed curve. The format is as follows:

$$S = \frac{1}{2}\oint (xdy - ydx) \tag{5}$$

S is the area wanted.
After binarization:

$$S = \frac{1}{2}\sum_{i=0}^{n-1}(x_i(y_{i+1} - y_i) - y_i(x_{i+1} - x_i)) = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - y_i x_{i+1}) \tag{6}$$

S is the area wanted.

(3) For each individual area, establish the minimum circumscribed rectangle model. The length and width of the rectangle is a and b, then, the format is as follows:

$$S = a \times b \tag{7}$$

S is the area wanted.

By the same token, the maximum and minimum circle model, the average round model, and elliptical model and other model which are equivalent to those model have the similar effect.

4.2 Improved Algorithm to Calculate the Area

Pixel accumulative method is easy, but it is space-consuming. If the area of the flaw is very large, the pixel adds one by one, thus the time of the operation is too long, and it reduce the efficiency of operation in the system. When using the Green theorem, it needs to be carried out by crossover operation point by point, and it's both time-consuming and space-consuming. The improved algorithm is based on the balanced binary search tree, which has storied all the boundary points. The corresponding pixels where the line number is i is on the tree. The format is as follows:

$$S = \sum_{i=0}^{n-1}(y_{i\max} - y_{i\min}); \tag{8}$$

S is the area wanted. y_{imax} and y_{imin} are the maximum and the minimum, n is the total number of the line.

5 Experimental Results and Analysis

5.1 The Results and Analysis of the Boundary Tracking

The traditional boundary tracking method based on the storage of the 8-direction chain code, when encountered intersection, is easily miss detection. Which will result in the

inaccuracy of the data, even lead to the detection failure. Thus graph (a) is the original picture. Graph (b) is the result of the boundary following by the common method based on the Freeman chain code. As is shown in the picture (b), When boundary adhesion appears, tracking by traditional boundary tracking method based on chain code will cause a lot of edge points missed. Graph (c) is the result of the boundary following by the improved algorithm proposed in the paper. The data are stored in established balanced binary search tree. So the boundary tracking algorithm based on the balanced binary search tree is available in this paper.

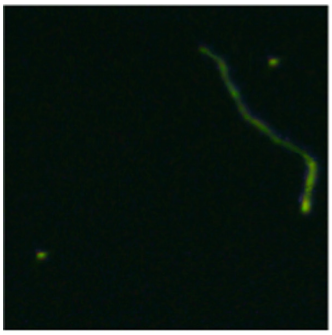
a） b) c)

5.2 The Analysis of the Detection Effects for the Optical Lens Detection

Graph (d) is the original picture adopted by the detection system. Graph (e) is the picture of the lens after binarization processing. Graph (f) is the graph (e) after edge following by the algorithm proposed in this paper. The algorithm of boundary tracing can get complete and accurate information of boundary. And by scanning it once, creating a balanced binary search tree for storage, the full boundary information and the data needed for the subsequent feature can be extracted. By reducing the amount of data from the entire digital image to the boundary point of all the defects, the time of calculation is greatly reduced, and the speed of the computation is faster.

d) e) f)

Graph (g) is the original image of the scratch. Graph (h) is the image after being processed by the Sobel operator. Graph (i) is the image processed by the Prewitt operator. The Sobel operator and the Prewitt operator are based on first derivative edge detection operator, through the calculation of the gradient of the image to detect the image edge. So the ability of anti-noise is poor, which brings out a lot of unnecessary points. Graph (j) is the image processed by the LOG operator which produces a large amount of noise and non-demand boundary areas. Therefore, those kinds of edge processing algorithm will not be applied to the rapid feature extraction of optical lens defect.



g)

h)

i)

j)

5.3 The Study for the Influence Factors of Digital Image Processing Speed

From the table below, it can be seen that the time of imaging processing has the intimate relationship with the size of the image, the sum of the flaws and the size of the flaw. In the same resolution, the larger size of the image is, the more data is needed to be processed, and the more time is needed to handle it. Similarly, the more high-resolution the digital image is, the more time it takes to process the image. Due to a certain amount of time is needed for boundary tracking and data storage, when the size and resolution are certain, the larger number of defects in the digital image, the

more average area of the defects, or the greater average length of the period, the more time it will take to handle the problem. From the Table 1, it is obvious to see that the algorithm proposed this page is much faster than the common method.

Table 1. Detection time of unimproved and improved algorithms

Item value size (pixel)	2592 × 1944	2592 × 1944	2592 × 1944	432 × 324
Detection time of the unimproved algorithm(s)	2.125	3.351	4.142	0.131
Detection time of the improved algorithm(s)	0.712	0.924	1.235	0.026
Sum of the flaws	52	87	116	32
Average perimeter of the flaws (pixel)	1792	1881	1824	52
Average area of the flaws (pixel)	8842	8163	9457	231

6 Conclusion

This paper proposes a edge tracking algorithm based on the balanced binary search tree. Calculate the perimeter and the area of the optical lens by the data storied. The novel algorithm is simplified, and it responds fast. Relying on this method, only once is needed to scan the image based on binarization transformation. It's no need filling area. All the needed message will be extracted, without using Freeman Chain Code.

References

1. Liu, H., Shen, J., Guo, S.: Digital image processing using Visual C++. China Mach. Press **6**, 154–156 (2010)
2. Chen, Y., Shang, L.: Improved SIFT image registration algorithm on characteristic statistical distributions and consistency constraint. Optik-Int. J. Light Electron Opt. **127**(2), 900–911 (2016)
3. Lin, H., Du, P., Zhao, W., et al.: Image registration based on corner detection and affine transformation. In: 2010 3rd International Congress on Image and Signal Processing (CISP), vol. 5, pp. 2184–2188. IEEE (2010)
4. Goodman, J., Weare, J.: Ensemble samplers with affine invariance. Commun. Appl. Math. Comput. Sci. **5**(1), 65–80 (2010)
5. Wang, W., Zhang, D., Zhang, Y., et al.: Robust spatial matching for object retrieval and its parallel implementation on GPU. IEEE Trans. Multimedia **13**(6), 1308–1318 (2011)
6. Wang, P., Chen, Q., Chen, H., et al.: A new affine-invariant image matching method based onSIFT. In: International Symposium on Photoelectronic Detection and Imaging 2013: Infrared Imaging and Applications. International Society for Optics and Photonics, vol. 8907, p. 89072D (2013)
7. Yeh, J.P.: Detecting edge using support vector machine. Adv. Mater. Res. **588**, 974–977 (2012). Trans Tech Publications

8. Sun, X.H.: Digital image processing: the principle and algorithm (2010). (in Chinese). 孙燮华. 数字图像处理: 原理与算法 (2010)
9. Liu, L.L.: New algorithm and its application for edge detection in morphology. J. Infrared Millimeter Waves **17**(5), 386–390 (1998)
10. Zhao, J., Xu, Y., Jiao, Y.: The fast arithmetic study of image edge detection based on the order morphology. Acta Electronica Sinica **36**(11), 2195–2199 (2008)
11. Chuang, C.H., Lie, W.N.: A downstream algorithm based on extended gradient vector flow field for object segmentation. IEEE Trans. Image Process. **13**(10), 1379–1392 (2004)
12. Zhao, Y., Chen, H., Wang, S., et al.: An improved method of detecting edge direction for spatial error concealment. J. Multimedia **7**(3), 262–268 (2012)
13. Jie, H.: The Annotated STL Source. Huazhong University of Science and Technology Press, WuHan (2002). (in Chinese). 侯捷. STL 源码剖析. 华中科技大学出版社 (2002)
14. Tang, J., Millington, S., Acton, S.T., et al.: Surface extraction and thickness measurement of the articular cartilage from MR images using directional gradient vector flow snakes. IEEE Trans. Biomed. Eng. **53**(5), 896–907 (2006)
15. Xie, J., Li, S., Lin, G.: The method of calculating the area and circumference based on the boundary trace. Electron. Technol. Softw. Eng. (9), 119–120 (2014). (in Chinese). 谢家龙, 李林升, 林国湘. 基于边界跟踪的多连通区域面积和周长的计算方法. 电子技术 与软件工程 (9), 119–120 (2014)
16. Zhou, X., Chen, Y., Hu, W.: Tree traversal binary image boundary tracing algorithm based on cross-point. Comput. Appl. Softw. **31**(2), 230–232 (2014). (in Chinese). 周秀芝, 陈洋, 胡文婷. 基于交叉点的树遍历二值图像边界跟踪算法. 计算机应用与软件 **31**(2), 230–232 (2014)