A High Accurate Vision Algorithm on Measuring Arbitrary Contour

Hongwei Xie[1] Kun Sun[2] Yu Liu[2]

1.Guangzhou University, Guangzhou 510006, China
xhw_cn@foxmail.com
2. South China University of Technology, Guangzhou 510640, China
sunbaiwan9527@gmail.com, auylau@scut.edu.cn

Abstract. In order to measure the error of arbitrary contours, a high accurate visual algorithm is proposed. Firstly, a subpixel edge extraction algorithm based on local grayscale fitting is used to extract subpixel contours. And then we use the coarse-to-fine matching strategy, which uses the shape context feature of contours to perform coarse matching, and then based on iterative closest points to complete fine matching step. Finally, the neighborhood algorithm is proposed to calculate the contour error. Experiments show that the accuracy of this algorithm can reach 0.5 pixel, and this algorithm can apply to error measurement of any shape of contours.

Keywords: shape context descriptor; iterative closest points; sub-pixel contour extraction; contour measurement;

1 Introduction

The visual measurement technology of product defects is widely used in industrial applications. Compared with other measurement techniques, machine vision measurement technology has the advantages of high degree of intelligence, good real-time and high precision [1]. Contour measurement is just one of the important applications.

At present, the research on the contour detection and measurement algorithm is basically to divide the extracted contour into straight lines, ellipses, circles and other geometric primitives [2-4], and then fit them respectively [5,6]. Based on the known number of breakpoints, [2] proposed an algorithm to determine the approximate function in the breakpoint interval to divide the contour curve into lines and circles. Based on the detection of arc and the dominant point, [3] divided the contours into straight lines and circles. And then the robustness of this algorithm is improved in [4], which using the adaptive tangential covering for the dominant point detection and the polygonal approximation of the contours on the tangent space. With the least squares method, [5] fitted circles, ellipses, hyperbolas, etc. A new method of nonparametric fitting and elliptic arc is proposed in [6], which improves the measurement accuracy of line segment and ellipse. Although these research works make the robustness and accuracy of contour segmentation and fit increased, these methods can only be used to detect and measure the contours of the regular shapes. Considering the fact that the actual workpiece shapes are varied and the contours may be made up of irregular curves, a new artifact detection algorithm for workpiece contour defects is proposed, which can accurately detect and measure the contour defects of any shape of the workpiece.

The algorithm proposed in this paper can be divided into three steps: extracting the contour, matching the contour and calculating the error. Firstly, the area of the workpiece is extracted on the image, and followed by denoising and closing contour. Then the sub-pixel edge is extracted based on the local area edge extraction algorithm [7]. Secondly, we will match contour by coarse-to-fine matching strategy. Select the sample points first, which are used to calculate the shape context descriptors for initial position registration of contours [8], and then fine matching using the iterative nearest point algorithm [9,10]. Finally, the error of each position in the contour is calculated according to the matching result.

2 Extracting contour

2.1 Image Preprocessing

In order to get a better contour of the workpiece, we use industrial camera with parallel backlight and telecentric lens to collect images. Firstly, the image is done binarization, erosion and dilation separately, followed by smoothing filter to extract region of interest (ROI). Then the first order gradient of ROI is obtained by the convolution of image with Prewitt operator and the pixels whose gradient value is greater than the threshold is noted as the contour edge. However, the contour obtained by the above method is likely not to be closed continuously, which will affect the accuracy of error calculation. So, the contours should be filled with appropriate erosion and dilation.

2.2 Contour Extracting

The existing sub-pixel edge extraction algorithm can be divided into three categories: based on the moment[11], based on the least squares [12] and based on interpolation [13]. The methods based on moments have high accuracy, but it is very sensitive to noise. The methods based on the minimum squared difference determines the edge parameters by minimizing the local energy function, and are robust to the noise, but the computation is time consuming. There are also some computationally efficient methods based on interpolation, but is susceptible to errors. Therefore, a new sub-pixel edge extraction algorithm which is different from the above three categories was proposed in 7. This method can obtain the sub-pixel information accurately, including the position, gray scale difference, normal vector and curvature of edges, based on the function on the neighborhood area. It is also computationally efficient, robust to noise near the edge, and accurate sub-pixel information. And particularly, accurate curvature will play an important role in removal of false matches and fine matching, which is not available in other edge extraction algorithms.

Assume that the edge is a quadratic curve

$$y = a + bx + cx^2 \tag{1}$$

This curve cut up the image into two parts, and the gray values are A and B. Take a 5 × 3 neighborhood centered on the given pixel, as shown in Fig 1.

As shown in Fig.2, the coordinate system is established with the center of the given pixel as the origin, and $F_{(i,j)}$ is the gray value of each pixel $p(i, j)$. The sum of gray values of each column is respectively marked as S_L, S_M, S_R. So

$$S_L = \sum_{n=j-2}^{j+2} F_{i-1,n} = 5B + \frac{A-B}{h^2}L \tag{2}$$

$$S_M = \sum_{n=j-2}^{j+2} F_{i,n} = 5B + \frac{A-B}{h^2}M \tag{3}$$

$$S_R = \sum_{n=j-2}^{j+2} F_{i+1,n} = 5B + \frac{A-B}{h^2}R \tag{4}$$

where L, M, and R denote the area of each column respectively, and the simultaneous equations are solved

$$a = \frac{2S_M - 5(A+B)}{2(A-B)} - \frac{1}{12}c \quad , b = \frac{S_R - S_L}{2(A-B)} \quad , c = \frac{S_L + S_R - 2S_M}{2(A-B)} \tag{5}$$

whereby the curvature of the pixel can be obtained

$$K = \frac{2c}{(1+b^2)^{\frac{3}{2}}} \tag{6}$$

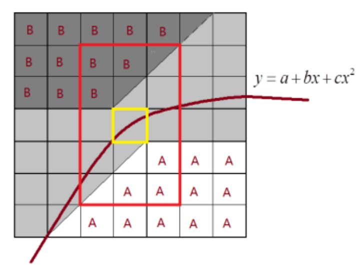

Fig. 1. The pixel in edge curve and its 5 ×3 neighborhood marked with a red box

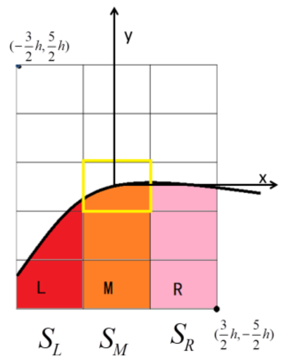

Fig. 2. The neighborhood of the quadratic curve

3 Contour Matching

3.1 Coarse Matching

Get shape context descriptor

Since there are so many pixels of the target contour, it is very time consuming to compute the feature orderly, and the appropriate sampling will improve the matching speed. So firstly we will randomly sample the contours uniformly and get a sparse contour Q.

In 2002, Serge Belongie proposed the shape context descriptor (hereinafter referred to as *SC*), which had achieved a good effect in shape matching. Suppose the contour is $S = \{s_1, s_2, \ldots, s_n\}$, where any $s_i \in S$ is the origin of the log-polar coordinates, and SC of s_i is the distribution of the histogram of other n-1 point.

In the log-polar coordinate system, the coordinate plane is divided into $k=r\times l$ bins, where r is the equal segment number of the angle axis and l is the logarithm axis of the number of segments, to record the distribution of the remaining n-1 points. And this constructs the histogram $h_i(j)$, where i is the i-th point, j represents the j-th bins, where $1 \leq i \leq n$, $1 \leq j \leq k$. This feature is invariant in translation. As the shape context descriptor using the logarithmic distance, which can accurate describe nearby points and approximate describe points faraway, we can take into account the local features and global features, as shown in Fig 3 and Fig 4.

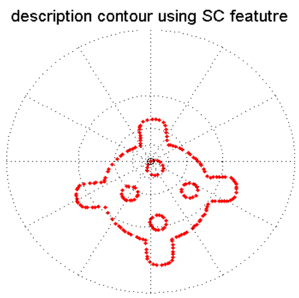

Fig. 3. the description of workpiece with SC descriptor

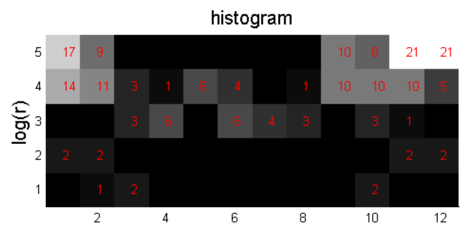

Fig. 4. count the points in each bins of Fig 4 and draw the histogram

Feature matching

Assume $p_i \in P$, $q_i \in Q$, are the random points in contour P and Q respectively, the cost function between the two points can be described as

$$C_{i,j} = C(p_i, q_j) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k)-h_j(k)]^2}{h_i(k)+h_j(k)} \tag{7}$$

where $h_i(k)$ and $h_j(k)$ is the histogram in pi and qi. Thus we can calculate the cost function of all corresponding points between P and Q

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)}) \tag{8}$$

and expect to minimum the total cost function, which is a weighted bipartite graph matching problem. Here we use a more efficient algorithm proposed in [14]to calculate the best match.

Removing mismatch

The mismatch between contour points often occurs, which will reduce the speed and accuracy of the contour matching. Therefore, a rejection function is designed with the curvature of sub-pixel,

$$r = \begin{cases} 1, & c_1 - c_2 \le c_0 \\ 0, & c_1 - c_2 > c_0 \end{cases} \tag{9}$$

where c_1 and c_2 are the curvatures of the corresponding points. When the curvature error of the corresponding point is greater than the threshold value c_0, it would be considered to be false match and the point pair will be discarded. Then, the rotation and translation matrices can be calculated based on matching result.

Calculating rotation and translation matrics

According to the matching results of the corresponding points, the commonly used methods for calculating the rotation and translation matrices are: SVD decomposition [15], quaternion method [16], orthonormal matrix [17], dual quaternion method [18]. Here we use SVD decomposition to solve the problem.

The result is shown in Fig 5.

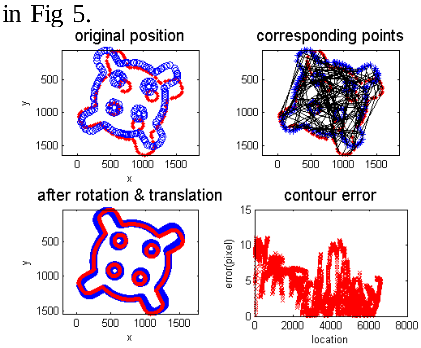

Fig. 5. shape matching with SC descriptor

The first line of pictures are the sampled contours and correspondences with *SC* descriptor (connected with segments). The second line of pictures are the result after coarse matching and the error of each contour point. But it can be seen that the error range reaches 10 pixels, obviously cannot meet the measurement needs. So the more accurate measurement, at pixel or even sub-pixel level, is needed.

3.2 Fine Matching

The coarse matching result is used as the initial position for the *ICP* algorithm. This algorithm essentially is iteratively matching the corresponding points to calculate the optimal rigid transformation matrix, until the condition of stopping the convergence is satisfied. But the algorithm requires that the two contours have been approached to some extent or the initial position has been estimated, otherwise the process of searching for the nearest point will fall into the local optimum. So here we use the framework proposed in [15] to achieve fine matching of contours. The steps are as follows:

1. Select the data. Considering the results of the initial match has been more accurate, so all the points would be use directly. And it can be seen that it has reached the optimal in the first several iterations from Fig 6.
2. Match contour points. Searching the nearest contour point is usually the most time-consuming step. There are three general approaches: The simplest but also most time-consuming method is traversing all $q_j \in Q$ to find the nearest point for all $p_i \in P$ In order to speed up the search process, generally use kD-tree or Delaunay algorithm. Here using kD-tree.
3. Determine the weight function. Considering error of the noise effects and error in the contour extraction process, a new weight function is designed based on the sub-pixel information.

$$w = e^{-(\alpha m_1 + \beta m_2 + \lambda m_3)} \tag{10}$$

where

$$m_1 = \frac{|d_1 - d_2|}{mean(|d_P - d_Q|)},\quad m_2 = \frac{1}{\vec{n}_1 \cdot \vec{n}_2} \frac{1}{mean(|\vec{n}_P \cdot \vec{n}_Q|)},\quad m_3 = \frac{|c_1 - c_2|}{mean(|c_P - c_Q|)}.$$

where m1 is the normalized residual of the nearest points in P and Q, m2 is the normalized product of the normal vector of the nearest point, and m3 represents the normalized difference in the curvature of the nearest point. α, β, γ The corresponding scale factor, α+ β + γ = 1. Here α = 0.5, β = 0. 3, γ = 0.2.
4. Discard partial matching result. Discard the poor quality of the matching results, not incorporated into the error calculation. Here, the minimum weight of 10% of the corresponding points is discarded.
5. Determine the error metric function. After getting the matching result, the rest of the work is to measure the error and minimize the error function. There are two main error measurement methods:
One is calculating point-to-point distance proposed in [9]:

$$\sum_{i=1}^{n} \| w_i (Rp_i + t) - q_i \|^2 \tag{11}$$

Another is calculating point-to-tangent distance proposed in [10]:

$$\sum_{i=1}^{n} \| w_i (Rp_i + T - q_i).n_i \|^2 \tag{12}$$

As the point-to-tangent distance is more accurate to describe the positional relationship of contours, it is better when the high precision is required. Here we adopt this method.

6. Minimum the contour error. Linearize the rotation matrix $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ to $\begin{pmatrix} 1 & -\theta \\ \theta & 1 \end{pmatrix}$. Thus, the rotation angle θ and the translation distance t_x and t_y can be obtained by the least squares method. Although this approximation introduces errors, with the iteration the angle become smaller and smaller, and the linearized rotation matrix will be very close to true value.

In Fig 6 below, first line on the left side of is the coarse matching result, and the right side is fine matching result. The second line on the left side is the contour error, the right is detailed iterations. It can be seen that the error has been completely less than one pixel.

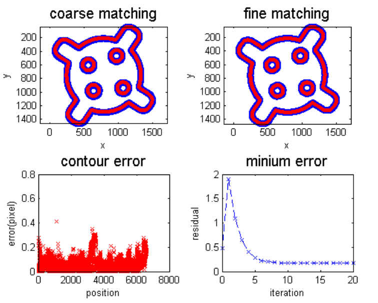

Fig. 6. The contour matching of work-piece

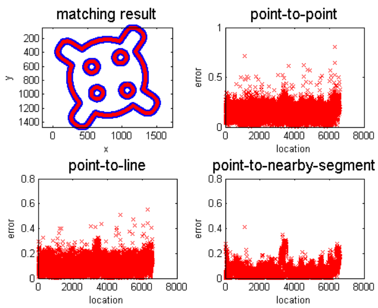

Fig. 7. compare of three kind metric methods

4 Calculating the contour error

The contour error will be calculated after the contour has been fine matched. As the contour is discrete and similar, so it is needed to try to make the results closer to the real value. There are three methods of error calculation:

1. Point to point. For any $p_i \in P$, whose nearest point in Q is denoted q_j, the error is

$$e_i = \| p_i - q_j \| \tag{13}$$

that is, the European distance of these two points. This approach depends on the quantity of points, when the contours are sparse, the results will be very inaccurate. Generally, this method is not used directly.

2. Point to tangent. For any $p_i \in P$, whose nearest point in Q is denoted q_i, the error is

$$e_i = \| (p_i - q_j).n_i \| \tag{14}$$

That is the distance of p_i to q_j where the tangent. It can be seen this method requires high precision of the normal vector.

3. Neighborhood method. Based on the advantages of the above two methods, we design the neighborhood method, that is, calculate the minimum distance from the point to the adjacent segment. For $p_i \in P$, find the n points nearest to p_i in Q, and clockwise connect them to n-1 line segments. Let $\{q'_1,\ q'_2, q'_3, \ldots\ q'_{n-1}\}$ be the minimum distance from pi to the n-1 segment, and it is just the error in contour point qi.

$$e_i = \min\{q'_1, q'_2, \ldots, q'_{n-1}\} \tag{15}$$

Three error measurement methods are compared in Fig 7. It can be seen that the neighborhood method proposed in this paper has the best effect. Because the distance of point-to- tangent is limited by the precision of the normal vector, and distance of point-to-point is limited by the quantity of the contour points, while the neighborhood method minimizes these negative factors, with better robustness.

5 Experiment and Analysis

5.1 Calibration plate experiment

The calibration plate with accuracy of 0.001mm is used to verify the accuracy of the algorithm. As the measurement object, it would be randomly rotated and translated in the field of view, and then the circle, rectangle and triangle on the calibration plate would be extracted and compared to standard contours. The error distributions are as follow:

It can be seen from Fig8-Fig10 that the accuracy of our contour detection and measurement algorithm can up to 0.5 pixels, especially good result on the circle and rectangle. However, the error of triangle is large, this is because the edge of the intersection is very close, affecting the accuracy of contour extraction.

5.2 Workpiece Experiment

The workpiece with precision of 0.01mm is placed in the field of view, and then compared extracted contour with the standard contour. The experimental results are as follows in Fig11.

It can be seen that for irregularly shaped workpiece, the accuracy of the contour measurement can also be within 0.5 pixels, but compared with the calibration plate experiment the error is significantly increased, which is caused by two reasons: First, the accuracy of workpiece is not high, with the fluctuation range of 0.2 pixels. Second, as the complexity of the contour increases, the extraction of corners and other edges of the contour would be less accurate.

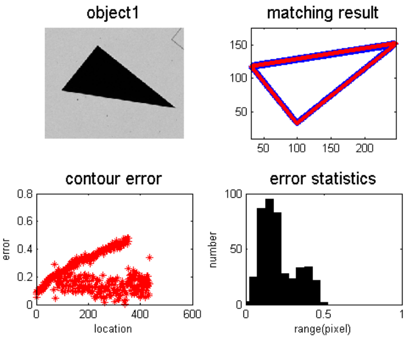

Fig. 8. calibration plate experiment 1

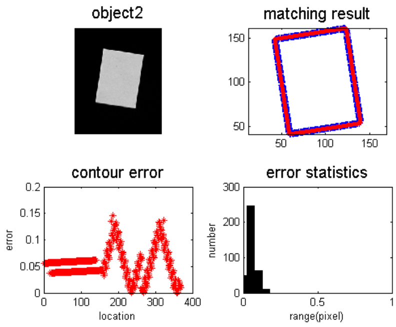

Fig. 9. calibration plate experiment 2

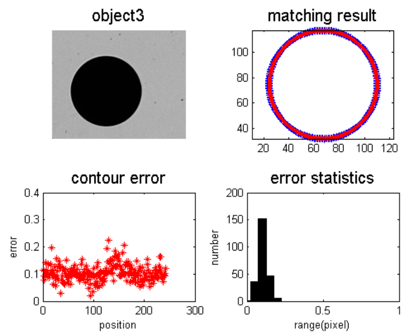

Fig. 10. calibration plate experiment 3

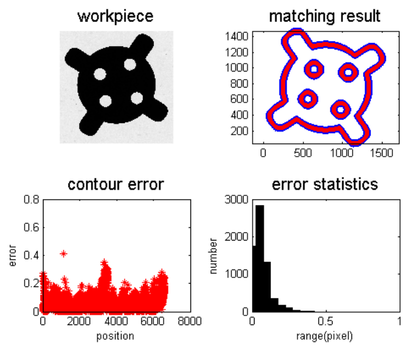

Fig. 11. workpiece Experiment

6 Summary and Outlook

In this paper, a contour error measurement algorithm is proposed to detect and measure the contour error of any shape, and the measurement accuracy can up to 0.5 pixels. The algorithm will greatly improve the application of error detection in industrial and improve the efficiency of production. In this paper, according to the requirements of contour error detection and the characteristics of sub-pixel contours, we design a rejection function to reduce the false matching in *SC*, propose a new weight function to increase the accuracy and robustness of *ICP*, and also design the neighborhood method to calculate the contour error. However, the robustness of the algorithm is not good enough to the affects from the noise point close to the edge. And as the increase of contour points, the extraction of sub-pixel edge and run time of *ICP* will increase accompanied, which is also an important factor restricting practical application of this algorithm. These are all needed to be improved in next steps.

References

1. WangYao-nan, ChenTie-jian, He Zhen-dong, Wu Cheng-zhong. Review on the machine vision measurement and control technology for intelligent manufacturing equipment, Control Theory & Applications, 2015, 32(3): 273-286
2. Chen, J.-M., Ventura, J.A., Wu, C.H.: Segmentation of planar curves into circular and line segments. Image Vision and Computing 14, 71–83 (1996)
3. Nguyen, T.P., Debled-Rennesson, I., Decomposition of a curve into arcs and line segments based on dominant point detection, SCIA '11. (2011) 794-805
4. Phuc Ngo, Hayat Nasser, Isabelle Debled-Rennesson, A discrete approach for decomposing noisy digital contours into arcs and segments. ACCV Workshop 2016, Nov 2016
5. Sung Joon Ahn, Wolfgang Rauh, Hans-JuKrgen Warnecke, Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola, Pattern Recognition 34 (2001) 2283-2303

6. Viorica Patraucean, Pierre Gurdjos, Rafael Grompone von Gioi, A Parameterless Line Segment and Elliptical Arc Detector with Enhanced Ellipse Fitting, ECCV 2012, Part II. LNCS, vol. 7573, pp. 572–585. Springer, Heidelberg (2012)
7. Agustín Trujillo-Pino, Karl Krissian, Miguel Alemán-Flores, Daniel Santana-Cedrés, Accurate subpixel edge location based on partial area effect Accurate subpixel edge location based on partial area effect, Image and Vision Computing 31 (2013) 72–90.
8. Serge Belongie, Jitendra Malik,Jan Puzicha, Shape Matching and Object Recognition Using Shape Contexts, IEEE Trans . Pattern Analysis and Machine Intelligence. VOL.24, NO.24, April 2002
9. Paul J. Besl,Neil D.McKay, A Method for Registration of 3-D Shapes, IEEE Trans .Pattern Analysis And Machine Intelligence,VOL.14,NO.2,February 1992
10. Yang Chen, Gerard Medioni, Object Modeling by Registration of Multiple Range Image, Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, Califormia, April 1991
11. S Ghosal,R Mehrotra, Orthogonal Moment Operators for Subpixel Edge Detection, Pattern Recognition, Volume 26,Issue 2,February 1993
12. M. Kisworo, S. Venkatesh, G. West, Modeling edges at subpixel accuracy using the local energy approach, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 405–410,
13. C. Steger, Subpixel-precise extraction of lines and edges, in: International Archives of Photogrammetry and Remote Sensing, vol. XXXIII, 2000, pp. 141–156, (part B3).
14. R Jonker and A Volgenant, A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems, Computing, VOL.38, pp. 325-340, 1987
15. K S Arun, T S Huang, and S D Blostein. Least-Squares Fitting of Two 3-D Point Sets. Pattern Analysis and Machine Intelligence, IEEE Trans, 9(5):698–700, 1987.
16. Berthold K P Horn. Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A, 4(4):629, 1987.
17. Berthold K P Horn, Hugh M Hilden, and Shahriar Negahdaripour. Closedform solution of absolute orientation using orthonormal matrices. Journal of the Optical Society of America A, 5(7):1127–1135, 1988.
18. Michael W Walker, Lejun Shao, and Richard a Volz. Estimating 3-D location parameters using dual number quaternions. CVGIP: Image Understanding,54(3):358–367, November 1991.