

Article

Research on Image Measurement Method of Flat Parts Based on the Adaptive Chord Inclination Angle Algorithm

Tao Liu *, Xingchen Lv  and Min Jin

College of Mechanical and Electrical Engineering, Inner Mongolia Agricultural University, Hohhot 010018, China
* Correspondence: liutao@imau.edu.cn; Tel.: +86-153-9112-4176

Abstract: To accurately measure the critical dimensions of flat parts using machine vision, an inspection method based on the adaptive chord inclination angle progressively screening the segmentation points of graphic elements was proposed in this study. The method doubled the size of the part image using bicubic interpolation, extracted the single-pixel contour with more detailed information, and designed an adaptive step size to obtain the front and back chord inclination angles of the contour. The method of complementing the front and back chord inclination angles was employed to avoid the negative effects of contour jaggedness, thereby obtaining the contour segmentation points after the initial screening. The segmentation points obtained in the initial sieving were divided into different point clusters according to the distance, and the contour, which was segmented by two segmentation points in different point clusters, was fitted using the least squares. The fitting results were evaluated, and all the fitting results were selected using the improved non-maximum suppression (NMS) algorithm to obtain the precisely selected segmentation points of the graphic elements. Consequently, the segmented individual graphic elements were fitted with the segmentation points as constraints to obtain the key dimensions of the closed part. The developed method could accurately find the contour segmentation points, and the relative error was less than 0.6%.

Keywords: feature points; feature extraction; contour segmentation; machine vision; part inspections



Citation: Liu, T.; Lv, X.; Jin, M. Research on Image Measurement Method of Flat Parts Based on the Adaptive Chord Inclination Angle Algorithm. *Appl. Sci.* **2023**, *13*, 1641. <https://doi.org/10.3390/app13031641>

Academic Editors:
Manuel Rodríguez-Martín,
João Ribeiro and Roberto
García Martín

Received: 29 December 2022
Revised: 22 January 2023
Accepted: 23 January 2023
Published: 27 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of the modern manufacturing industry, the requirements for the contour inspection of complex flat parts are becoming higher than ever, and the traditional dimensional inspection methods can hardly meet the requirements of measurement accuracy and speed. Rapid and accurate dimensional measurement methods have become an urgent problem. The application of machine vision to measure the sizes of parts has been a research hotspot and a major direction of development in the machinery precision machining and manufacturing industry [1].

Part size measurement based on machine vision mainly includes three technical points: part image contour extraction, graphic element segmentation point detection, and graphic element fitting. To date, numerous scholars have conducted research on these topics. Research on the direction of image contour extraction started early, and early studies mainly exploited the feature that contour points are discontinuous in the gray intensity in an image. An image was convolved using a local filter to determine the contour pixel point with the largest local gradient. Several linear filters [2–4] have been widely used, such as the Sobel, Perwitt, and Canny operators. However, Perona and Malik [5] pointed out that the real image contours are usually not step functions but a combination of step, peak, and roof situations. To cope with this situation, the grayscale gradient features were combined with other gradient features, including color and texture gradients, to obtain a probabilistic detector (Pb) [6]. Ren [7] significantly improved the detection accuracy of Pb using sparse coded gradient (SCG) features instead of hand-designed gradients in Pb. SCG features can be automatically learned from image data using sparse coding, making

the contour detection closer to that of humans. In recent years, many researchers have introduced deep learning techniques to the contour detection problem, with deep features instead of traditional hand-designed features. Xie [8] proposed holistically nested edge detection (HED) based on deeply supervised nets (DSN) [9], which implemented holistic image training and prediction, and multi-scale and multi-level feature learning.

In the scenario of part size measurement, the influence of complex background environments and textures was circumvented in image acquisition, which made contour extraction less difficult. Therefore, mature algorithms were usually adopted. Tao et al. [10] used the algorithm proposed by Suzuki [11] for contour extraction while measuring the lengths of plastic part notches in a video. This algorithm was adopted in this paper.

The detection of the segmentation point of the graphic element is the most critical part of the entire dimensional measurement system, and the detection accuracy directly determines the accuracy of the graphic element fitting and even the final dimensional measurement. This research can be divided into four directions: the approximate curvature method, the Hough transform and its improvement algorithm, the contour approximation method, and the auxiliary detection method.

Mokhtarian et al. [12] proposed a curvature scale space (CSS) corner point detector that employed local curvatures in different directions for corner point detection and improved localization accuracy. Subsequently, many studies have been conducted based on the approximate curvature for graphic element segmentation point detection. For instance, Wang et al. [13] proposed an eleven-point method to approximate the contour curvature, which improved the noise resistance of the traditional three-point method. Hou et al. [14] adopted the eleven-point method to find the approximate locations of feature points and then calculated the curvature differences for feature points within those locations to find the accurate locations of feature points.

The Hough transform was proposed by Paul Hough in 1962 to detect the features of specific shapes (such as lines or circles) in digitized images. It has an excellent detection effect on the graphic elements, which can be described by parametric features. Mo and Wu [15,16] achieved the segmentation of different figure elements by adopting the traditional Hough transform and its improved algorithm; however, Zhou [17] proposed a combination of the improved corner point algorithm and the point Hough transform circle detection method to achieve different feature point detection. Qu [18] optimized the Hough transform using the K clustering algorithm. Rachmawati et al. [19] proposed a polygon approximation technique combining high-speed corner point detection with Freeman chain code, which constructed an approximate polygon of a string of ordered line segments using a set of image corner points, and the approximate polygon could be further used in the field of shape representation and analysis as a contour description of digital images. Liu [20] adopted a contour approximation algorithm to achieve graphic element segmentation. Similarly, Wang [21] employed a polygon contour approximation method to achieve graphic element segmentation.

The auxiliary detection method mainly relies on other geometric elements and features near the segmentation point to detect the segmentation point of the graphic element. Chen [22] achieved contour intersection and tangent point detection by comparing the segmentation occupancy of the background and the part to the circular region around the contour point.

Dou [23] employed chord-to-point distance accumulation [24](CPDA) to achieve corner point detection. Shi [25] proposed the accumulation of point to tangent distance (APTD) for detecting the inflection points of image contours, and Huang [26] determined the segmentation point of a graphic element by studying the characteristics of the local line angle of the contour and established a graphic element error model to achieve accurate segmentation of the graphic element and fit.

The least-squares method is the most popular fitting algorithm for the final dimensional part measurement. To solve the problem of non-closure after the graphic element fitting, Wang [27] proposed the least-squares method with constraints. Li [28] introduced

Tukey [29] weights during the least-squares fitting of each contour point and to suppress the influence of large outliers on the fitting results. Moreover, in order to better evaluate the roundness error, Khilil [30] proposed an improved algorithm to find the region with the least roundness error using the alternating exchange method.

The existing research has made breakthroughs in many aspects, but there are still some shortcomings in the dimensional measurement of complex plane parts:

1. A large number of studies only measured the sizes of specific parts, and the algorithms are difficult to apply to complex parts.
2. Some studies did not pay enough attention to the reality that parts have complex contours, which makes it difficult for the algorithms to accurately locate segmentation points such as tangent points and inflection points.
3. Some studies required a lot of human intervention.

In this paper, we design an algorithm to address the defects of existing algorithms. The chord inclination angle feature used in the initial screening of segmentation points utilizes the jagged contour, thus eliminating the need for the smoothing of the contour, avoiding errors caused by smoothing and at the same time being sensitive enough to tangent points and inflection points. We also design the adaptive step to solve the impact caused by the support area, design the segmentation point evaluation mechanism, and improve the NMS algorithm to make it more suitable for segmentation point screening.

2. Image Preprocessing

2.1. The Image Pretreatment Process

The images used in this study were captured and acquired using an MV-VD200SC (1600 × 1200) camera (Figure 1) manufactured by *Microvision* (Xi'an, Shaanxi, China).



Figure 1. Camera for image acquisition.

The adaptive chord inclination angles depend on the single-pixel contour of the flat part; therefore, it was necessary to convert the acquired part image to grayscale after reading it. We doubled the grayscale image size using bicubic interpolation to obtain more detailed information, binarized the grayscale image through the Otsu method, and finally captured and obtained the single-pixel contour of the image. The workflow is illustrated in Figure 2.

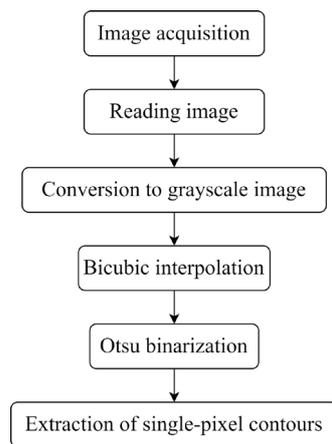


Figure 2. Flow chart of image preprocessing.

The operation effect of each step of preprocessing is shown in Figure 3. In addition to the original image, the effects of other steps are visually demonstrated by the red-shaded part in Figure 3a. It can be seen in Figure 3 that the jagged part image was effectively improved after interpolation.

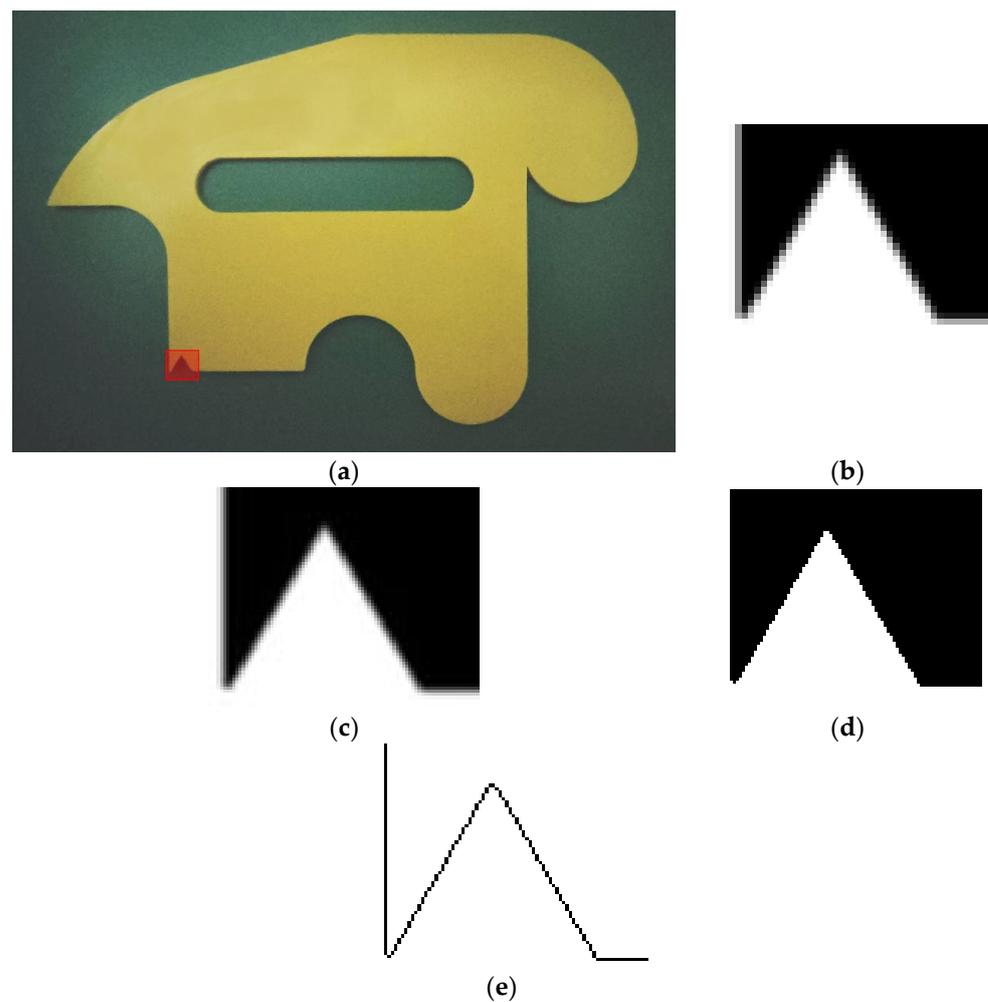


Figure 3. Image preprocessing. (a) Part image; (b) Grayscale image; (c) Bicubic interpolation; (d) Otsu binarization; (e) Single-pixel contours.

2.2. Design of Pending Inspection Image

During actual industrial production, various complex situations are involved in the detection of flat parts. However, it is difficult to cover all complex situations in one part. Therefore, we designed a pending inspection image to ensure comprehensive and credible results for the experimental verification in this study. We employed an image (Figure 4) as the pending inspection image, which contained the following types of segmentation points: an intersection between lines, an intersection between a line and an arc, a tangent point between a line and an arc, a tangent point of same-direction arcs, a tangent of reverse arcs, a tangent of a large arc, a large angle intersection, and segmentation points between shorter graphic elements. The graphic element information of this image was rich and comprehensive and could basically cover the common inspection requirements in an actual industrial production.

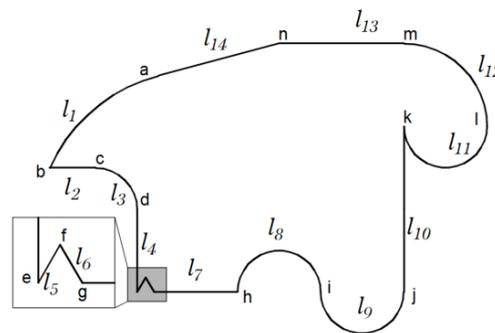


Figure 4. Pending inspection image in this paper.

The equivalence relations between the graphic elements in the image to be examined were designed to analyze the relative error of the algorithm processing results. The relationships among the graphic elements are listed in Table 1.

Table 1. Relationships between the graphic elements.

Type of Relationship	Content
Equal length	$l_4 = l_7 = l_{10}/2, l_{14} = l_{13}, l_5 = l_6$
Parallel	$l_4 // l_{10}, l_7 // l_2 // l_{13}$
Equal radius	$R_{l_3} = R_{l_8} = R_{l_9} = R_{l_{11}} = R_{l_{12}}/2 = R_{l_1}/4$
Other relationships	$\angle l_{13}l_{14} = 165^\circ$

The categories of the segmentation points involved in the pending inspection image are listed in Table 2.

Table 2. Categories of segmentation points.

Type of Segmentation Points	Segmentation Points
Intersection between line and line	e, f, g, n
Intersection between line and arc	b, h, j, k
Tangent point between line and arc	a, c, d, m
Tangent point of same-direction arcs	l
Tangent of the reverse arcs	i
Tangent of large arc	a
Large angle intersection	n
Segmentation points between shorter graphic elements	e, f, g

2.3. Bicubic Interpolation Process

The bicubic interpolation algorithm employed the grayscale values of 16 points around the sampled point as three interpolations, which took into account not only the grayscale

influence of the four directly adjacent points but also the impacts of the rates of change for the grayscale values between each neighboring point. The three interpolations could yield a magnification effect that was closer to the resolution of the image.

The distance between the pixels was combined using Equations (1)–(8).

$$d_{x_1} = \left| \frac{x'}{a \cdot x} - (x - 1) \right| \tag{1}$$

$$d_{y_1} = \left| \frac{x'}{a \cdot y} - (y - 1) \right| \tag{2}$$

$$d_{x_2} = \left| \frac{x'}{a \cdot x} - x \right| \tag{3}$$

$$d_{y_2} = \left| \frac{x'}{a \cdot y} - y \right| \tag{4}$$

$$d_{x_3} = \left| \frac{x'}{a \cdot x} - (x + 1) \right| \tag{5}$$

$$d_{y_3} = \left| \frac{x'}{a \cdot y} - (y + 1) \right| \tag{6}$$

$$d_{x_4} = \left| \frac{x'}{a \cdot x} - (x + 2) \right| \tag{7}$$

$$d_{y_4} = \left| \frac{x'}{a \cdot y} - (y + 2) \right| \tag{8}$$

The weights were obtained using a distance-based function, taking a as -1 in most cases. In general, the distances of the blue pixels in Figure 5 were $|t| \leq 1$, and the distances of the green pixels were $1 < |t| < 2$. The weights could be calculated using Equation (9).

$$h(t) = \begin{cases} (a + 2) \cdot |t|^3 - (a + 3) \cdot |t|^2 + 1 & \text{if } |t| \leq 1 \\ a \cdot |t|^3 - 5 \cdot a \cdot |t|^2 + 8 \cdot a \cdot |t| - 4 \cdot a & \text{if } 1 < |t| < 2 \\ 0 & \text{else} \end{cases} \tag{9}$$

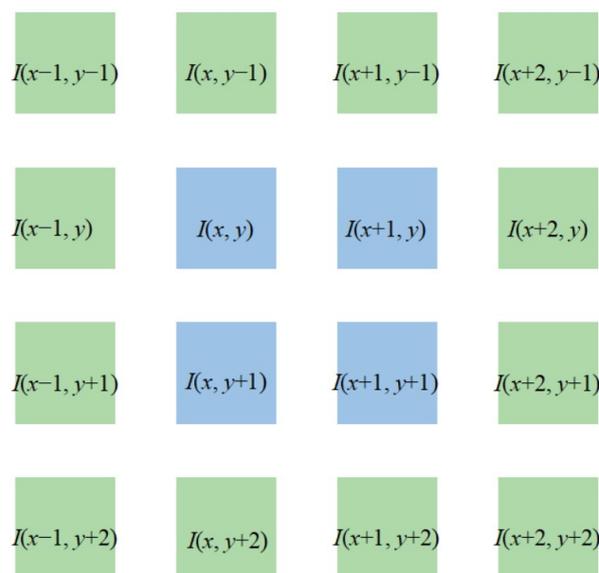


Figure 5. Support pixels for the bicubic interpolation algorithm.

The value of the interpolated pixel was obtained by dividing the sum of the products of each pixel and the weights by the sum of the weights, as in Equation (10).

$$I'(x', y') = \sum_{j=1}^4 \sum_{i=1}^4 (I(x+i-1, y+j-2) \cdot h(d_{x_i}) \cdot h(d_{y_j})) \quad (10)$$

Bicubic interpolation was adopted to expand the length and width of the grayscale image of the original flat part to twice the size of the original image so that the graphic element of the part had more contour points as support.

3. Detection Algorithm Design of Segmentation Points of Graphic Elements

3.1. Obtaining the Initial Screening Segmentation Points

When acquiring single-pixel contours from a part image, the part contours need to be distributed in a pixel-based grid, which inevitably causes jagged boundary contours (Figure 6).

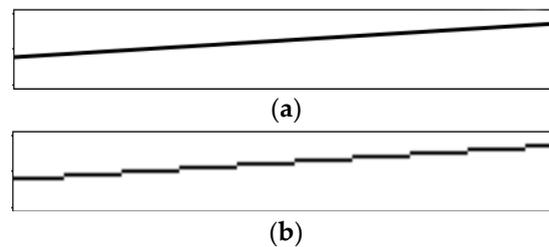


Figure 6. Contour jaggedness error. (a) Ideal boundary contour; (b) True boundary contour.

The impact of the jagged contour on segmentation point detection is mainly attributed to two aspects: on one hand, the jagged contour causes noise-like effects that lead to the formation of pseudo-segmentation points; on the other hand, some local features of contour serrations are stronger than the segmentation points of the pending inspection [31], thus resulting in missed segmentation point detection.

Some methods have been studied and reported for the above situations. For instance, Huang et al. [26] adopted Gaussian functions to smoothen the contour, resulting in an increase in the support area of the detection operator [13]; however, smoothing the contour causes leak detection of segmentation points with insignificant local features as well as spatial location changes of the segmentation points, and the smoothing treatment is irreversible. In addition, increasing the support area of the detection operator also leads to the leak detection of segmentation points that are closer.

Therefore, in this paper, we proposed a method to remove the effect of a jagged contour using the complementary front and back chord inclination angles of the contour point. For a point (P_i) on the contour, the front and back chord inclination angles are shown in Figure 7.

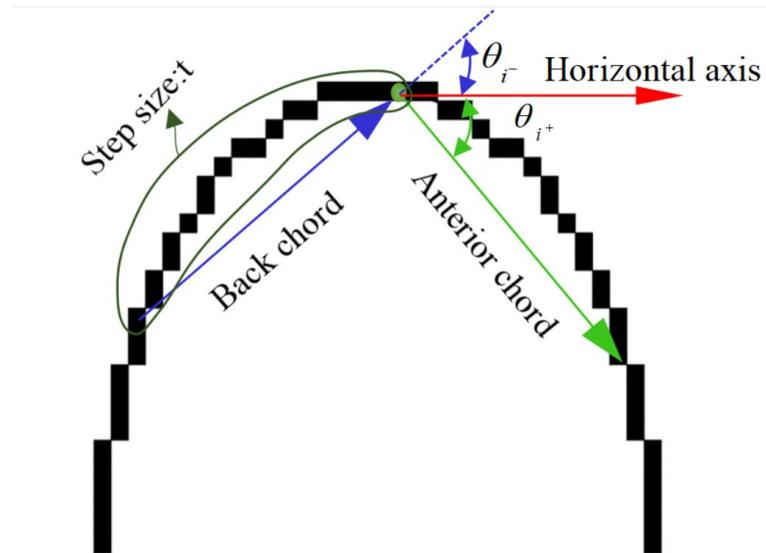


Figure 7. Diagram of the front and back chord inclination angles.

The front chord inclination angle (the angle between the green arrow and the red arrow) was calculated using Equation (11).

$$\theta_{i+} = \arctan \left(\frac{\left| \vec{P}_{i+t} P_i \times \vec{P}_0 \right|}{\left| \vec{P}_{i+t} P_i \cdot \vec{P}_0 \right|} \right) \tag{11}$$

The back chord inclination angle (the angle between the blue arrow and the red arrow) was calculated using Equation (12).

$$\theta_{i-} = \arctan \left(\frac{\left| P_i P_{i-t} \times \vec{P}_0 \right|}{\left| P_i P_{i-t} \cdot \vec{P}_0 \right|} \right) \tag{12}$$

where $P_0 = (0, 1)$ is the unit horizontal vector and t is the length of the chord across the contour point.

A jagged contour has a significant effect on the unidirectional chord inclination angles, which was studied using the contour at a tangent point, as shown in Figure 8.

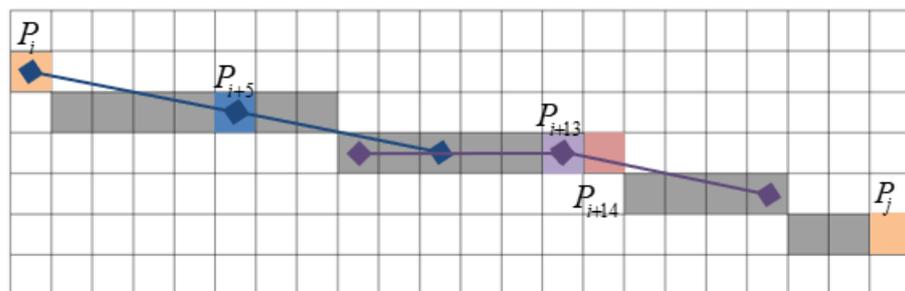


Figure 8. Effect of contour serration on chord dip angle.

The front and back chord inclination angles at the tangent point in Figure 8 are listed in Table 3.

Table 3. Variation in front and back chord inclination angles.

Index	Front Chord Inclination Angle	Back Chord Inclination Angle	Index	Front Chord Inclination Angle	Back Chord Inclination Angle
P_i	21.8		P_{i+11}	21.8	21.8
P_{i+1}	0		P_{i+12}	21.8	21.8
P_{i+2}	0		P_{i+13}	21.8	0
P_{i+3}	21.8		P_{i+14}	30.96	0
P_{i+4}	21.8		P_{i+15}	21.8	21.8
P_{i+5}	21.8	21.8	P_{i+16}	21.8	21.8
P_{i+6}	21.8	0	P_{i+17}		21.8
P_{i+7}	21.8	0	P_{i+18}		21.8
P_{i+8}	0	21.8	P_{i+19}		30.96
P_{i+9}	0	21.8	P_{i+20}		21.8
P_{i+10}	21.8	21.8	P_j		21.8

The effect of a jagged contour can be suppressed by complementing the front and back chord inclination angles.

The contour points had a uniform distribution of the jagged contour from P_{i+5} (the blue point in Figure 8) to P_{i+13} (the purple point in Figure 8). As can be seen in Table 3, the anterior chord inclination and posterior chord inclination of all contour points in this section always fluctuated between 0° and 21.8° , with a value (21.8) that was always the same between $(\theta_{i+5+}, \theta_{i+5-})$ and all $(\theta_{i+n+}, \theta_{i+n-})$, $n = 5, 6 \dots 13$. Once there was a point where the jaggedness distribution was clearly uneven (near the tangent point: (the red point P_{i+14} in Figure 8), there was a chord inclination angle (30.96) different from 0° and 21.8° , which could be considered a segmentation point.

Benefits of this property include:

1. This property makes it possible to find the contour segment with the same chord inclination angle and the longest length and to consider the two endpoints as the segmentation points to be selected to suppress the pseudo-segmentation points caused by the jagged contour.
2. The method of complementary front and back chord inclination angles is sensitive to the tangent point that can effectively respond to the tangent point and precisely locate the tangent point.

Experimental validation using the pending inspection image could effectively verify the detection effect of the algorithm on graphic elements of different scales and different types of segmentation points. The segmentation points initially screened for their contours by adopting the method of complementary front and back chord inclination are shown in Figure 9.

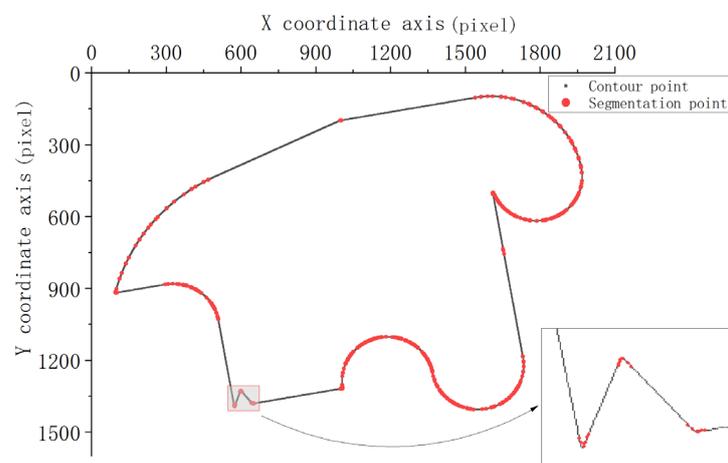


Figure 9. Segmentation points from initial screening.

3.2. Adaptive Step Size

For the method of complementary front and back chord inclination angles of the contour, the step size (t) determines the sensitivity of the chord inclination angle to changes in the jagged contour distribution. A smaller value of t leads to a lower sensitivity of the chord inclination angle, which results in a larger deviation in the detection position of the tangent point or leak detection. As can be seen in Figure 10 for the tangent of the large arc in the pending inspection image, a larger value of t leads to a higher sensitivity of the chord inclination angle, which results in all the shorter graphic elements in the contour being considered as segmentation points, making it more difficult to differentiate them, as shown in Figure 11 for the segmentation points between shorter graphic elements: e, f, and g.

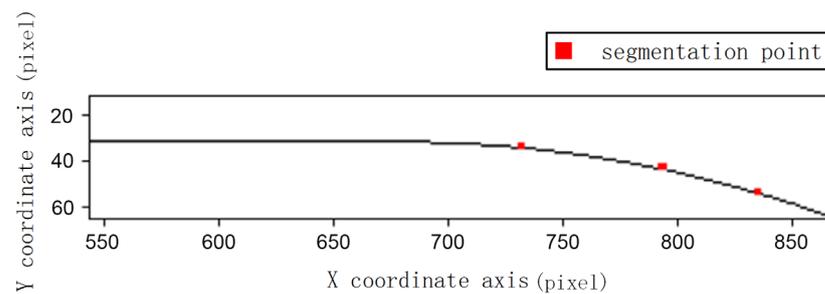


Figure 10. Deviation in the positions of the segmentation points at the tangent point of the great circle.

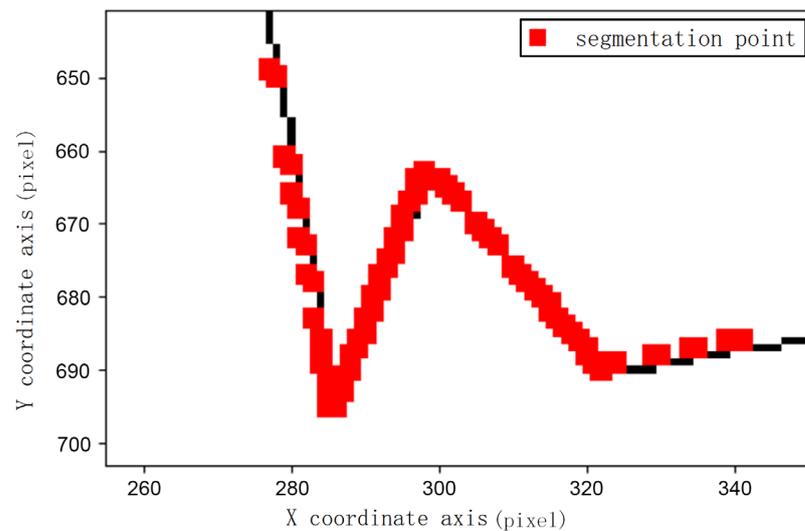


Figure 11. All shorter graphic elements being considered as segmentation points.

In summary, in the initial screening of the segmentation points by the chord inclination angle, the crossover points of graphic elements could be detected using a short step size, but the tangent points and inflection points could not be detected. In this case, fewer segmentation points were obtained, and the operation cost of the subsequent algorithm was low. Using a long step size, all segmentation points could be detected. In this case, the number of segmentation points was too large, and the operation cost of the subsequent algorithm was too large.

The goal of the adaptive step size design is to obtain an accurate and complete segmentation point detection ability while costing less and increasing the adaptive ability of the algorithm.

When the chord inclination angle is complementary to obtain the segmentation points, if the pixel length (s) is taken as the step length, there are $1 \sim s$ pixel segmentation points at the crossover points of primitives. When the step size is shorter, there is only one isolated

segmentation point at the tangent point and the inflection point. Therefore, the adaptive step size is detection using a shorter step and then using a longer step near the isolated segmentation point.

In this way, the value of a short step size only determines the operation cost of the algorithm but does not affect its detection performance. Therefore, the minimum value of the short step size was set to two pixels.

The contour intercepted by the nearest segmentation points to the left and right of the isolated segmentation point were detected using a longer step size, and the number of contour points between the isolated segmentation point and the nearest segmentation points to the left and right were recorded as $len1$ and $len2$, respectively. The larger step length should not be greater than $\min(len1, len2)$. If $\max(len1, len2) \gg \min(len1, len2)$, the expected segmentation points tended to be on the longer side, and to make sure they were properly positioned, we needed to set the step size to $\min(len1, len2)$. Based on the above description, we temporarily set the longer step size to $\min(\min(len1, len2), \max(len1, len2)/50)$. To avoid the calculated value of the longer step being too small to be used, the longer step was defined using Equation (13).

$$Stepsize = \max(10, \min(\min(len1, len2), \max(len1, len2)/50)) \tag{13}$$

The use of adaptive steps can effectively avoid all the shorter tuples being considered as segmentation points and solve the bias of tangent point position detection. These effects are presented in Figures 12 and 13.

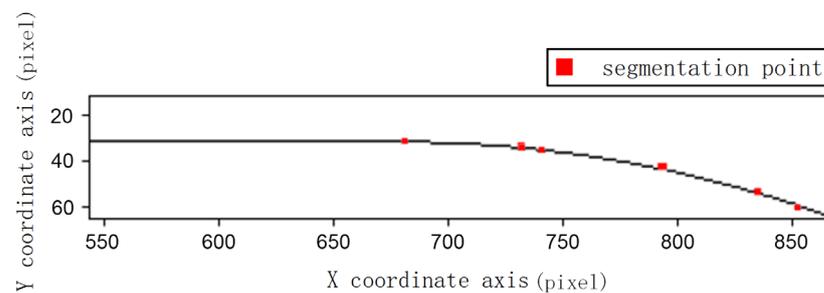


Figure 12. Segmentation points detected at large arc tangents by adaptive steps.

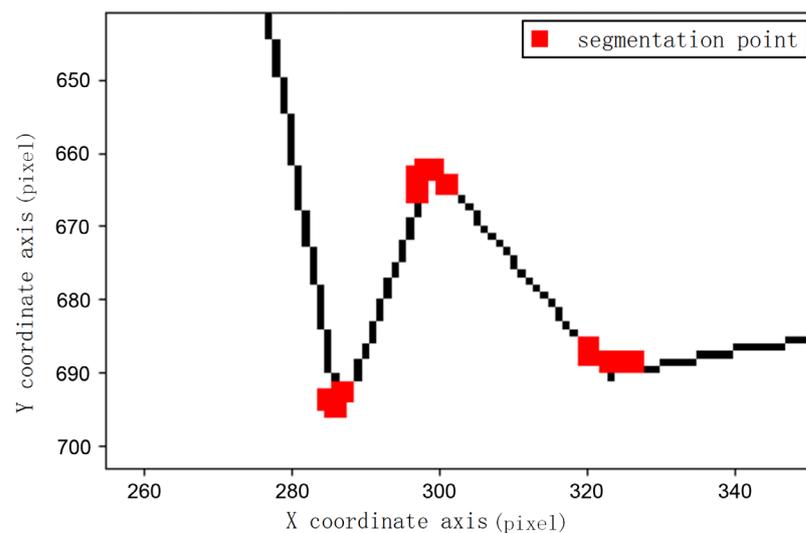


Figure 13. Segmentation points detected at shorter graphic elements by adaptive steps.

3.3. Improved NMS Algorithm to Precisely Select Segmentation Points

As shown in Figure 9, there were many redundant points near the crucial segmentation points after the initially screening by complementing the front and back chord inclination

angles. For redundant points, Wu et al. [16] regarded the point with the maximum curvature among the redundant points as the segmentation point. Hou et al. [17] calculated the curvature difference from the point cluster near the circular arc direction and regarded it as the segmentation point when a point was greater than the threshold. However, as we know, the curvature is susceptible to noise, and a screening approach based on curvature makes it difficult to accurately locate the segmentation point. For the pseudo-segmentation points on the arc, the previous study used the projection height method to treat them as points on the same arc.

Regarding the above analysis, we proposed a segmentation point selection method based on the improved NMS algorithm. The main process of this selection method is as follows:

1. Segmentation points are classified as point clusters.

In some cases, the segmentation points after the initial screening are adjacent or extremely close to each other. When fitting these segmentation points, the fitting error is small owing to the few contour points being included. However, this fitting is meaningless and even weakens the results of the effective fitting. Therefore, the segmentation points that are extremely close to each other are considered as the same point cluster, and the point clusters are numbered after the initial screening. Then, the subsequent fitting is not performed in the segmentation points that are in the same point cluster.

2. Evaluation of each contour segment and the fitting results.

From a segmentation point, we search for another segmentation point in different point clusters from proximal to distal and perform least-squares fitting on the contour segment between the two segmentation points until the fitting error is greater than the fitting tolerance threshold (T). Subsequently, the same operation is performed on the next segmentation point until all segmentation points are processed.

When evaluating the results of each fitting, it is crucial to ensure that the best segmentation points are used to obtain the best evaluation results. Therefore, the selected evaluation metrics must be the most sensitive to the contour points of the non-plot elements.

Figure 14 shows l_8 of an arc from the pending inspection image to compare the sensitivity of the maximum error, mean error, and mean square error to the wrong segmentation point, and the results are presented in Figure 15.

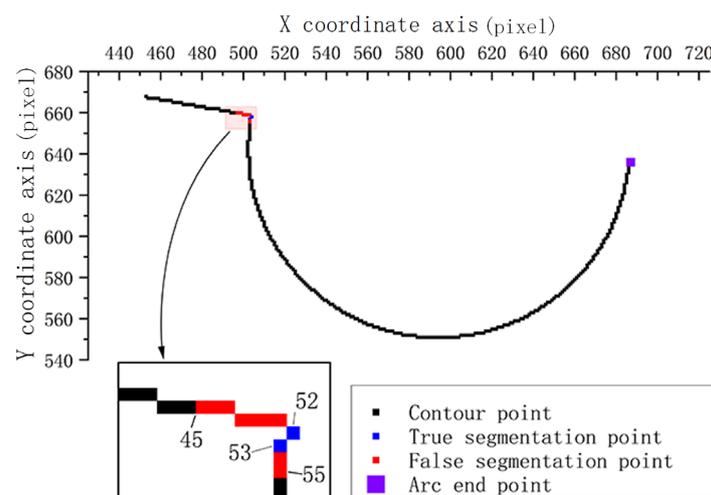


Figure 14. Arc of l_8 .

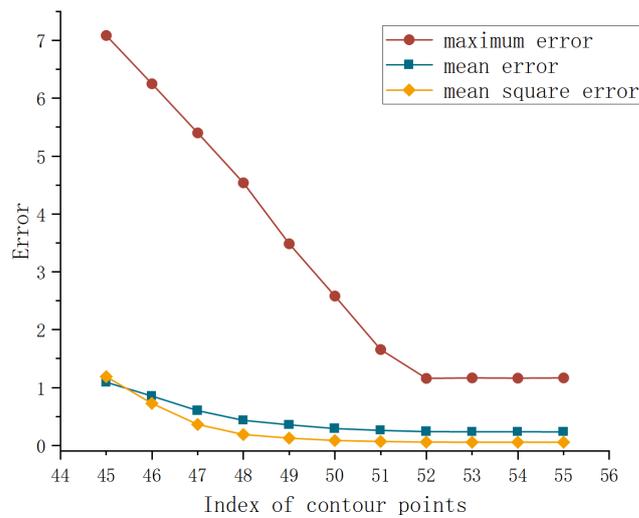


Figure 15. Fitting error of using 45–55 contour points as split points.

As shown in Figure 15, the maximum fitting error was the most sensitive to the false segmentation points. However, only adopting the maximum fitting error to evaluate the contour segment fitting results resulted in the local fitting results outperforming the overall fitting results for the graphic element, which contributed to the misdetection of segmentation points and finally caused the complete graphic element to be error-segmented.

As shown in Figure 16, points of the same color belong to the same cluster. There were multiple point clusters in one graphic element, and the maximum fitting error of the contour fitting between two points in the point cluster with close distance (‘maximum fitting error of contour fitting between two points’ is denoted as *Max_Error*) was smaller than the *Max_Error* of the real segmentation points. If only the maximum fitting error was used to evaluate the fitting results of the contour segments, the local fitting results were better than the overall fitting results of the primitives, resulting in the misdetection of segmentation points and the missegmentation of the complete graphic element.

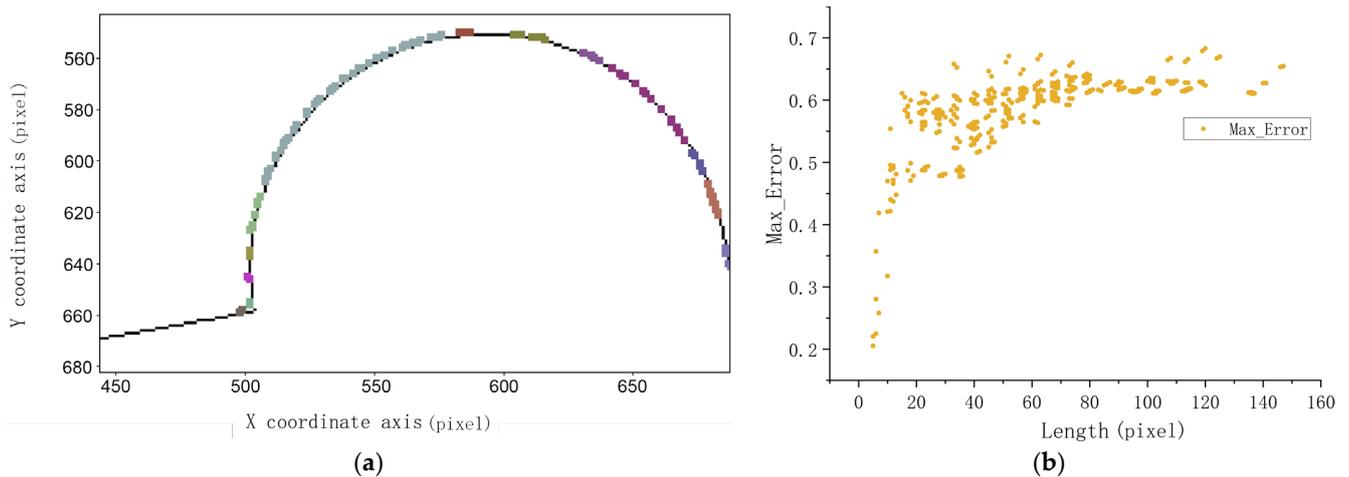


Figure 16. (a) l_8 distribution of point group. (b) Distribution of the length of the fitted contour segment in relation to the fitting error.

Therefore, the length (L) of the contour segment was introduced during the evaluation of the fitting results, and the length and maximum error (E) of each fitted con-

four segment were normalized to obtain l and e , and they could be calculated using Equations (14) and (15).

$$l = \frac{L - L_{\min}}{L_{\max} - L_{\min}} \tag{14}$$

$$e = \frac{E - E_{\min}}{E_{\max} - E_{\min}} \tag{15}$$

The evaluation of the fitting results for each time was recorded as $result = n \times l - e$.

As for the compensation effect of length on evaluation, we expected that if there was a point in the graphic element whose Max_Error was smaller than the Max_Error of the true segmentation point, the length compensation would make the fitting evaluation of the true segmentation point higher. The length difference was denoted as ΔS , and the Max_Error difference was denoted as ΔE . Then, $n \cdot l_{+\Delta S} - e_{-\Delta E} > n \cdot l - e$ could be obtained according to the expectation of length compensation:

$$n > \frac{L_{\max} - L_{\min}}{E_{\max} - E_{\min}} \cdot \frac{\Delta E}{\Delta S} \tag{16}$$

We did not need the length compensation to act on the point cluster where the split point was located so we could know that $\Delta S_{\min} = 5$, according to the definition of the point cluster.

The initial screening segmentation points were fitted by the least square method, and the Max_Error values are shown in Figure 17. Under the effect of the fitting tolerance threshold (T), the fitting mainly focused on the interior of the graphic element and the junction of the graphic element, which were, respectively, located at the two peaks of the frequency distribution histogram. The Otsu method was used to find the threshold to distinguish the two types of fitting results. that is, the distinguishing threshold (t) was found to maximize the variance between classes (Equation (17)).

$$\begin{aligned} (S_b)^2 &= w_0 \cdot (M_0 - M_t)^2 + w_1 \cdot (M_1 - M_t)^2 \\ &= w_0 \cdot w_1 \cdot (M_0 - M_1)^2 \end{aligned} \tag{17}$$

where ω_0 and ω_1 and are the ratios of the number of pixels in the two classes, separated by the threshold (t), to the total number of pixels (satisfied by $\omega_0 + \omega_1 = 1$); M_0 and M_1 are the averages of the pixel values of these two classes. From this, $t = 0.98$ was obtained.

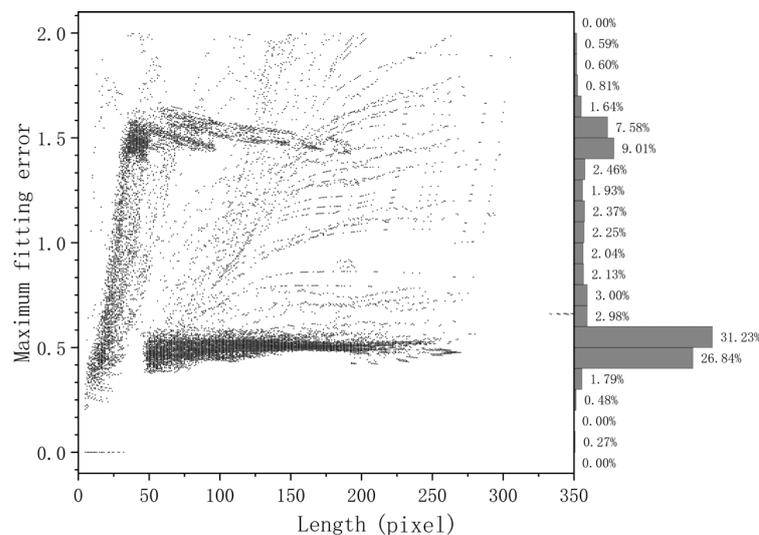


Figure 17. Contour segment length and its maximum fitting error distribution diagram.

The dividing value between the fitting results within the graphic element and the fitting results at the junction of the graphic element obtained using the Otsu method was set to the maximum value of ΔE , which was $\Delta E_{\max} = t = 0.98$:

$$inf(n) \leq \frac{L_{\max} - L_{\min}}{E_{\max} - E_{\min}} \cdot \frac{\Delta E_{\max}}{\Delta S_{\min}} \tag{18}$$

No length compensation was performed for the fitting result of $Max_Error > \Delta E_{\max}$ ($n = 0$) so that the operation could avoid the false detection of segmentation points caused by length overcompensation.

To summarize, the *result* could be calculated using Equation (19).

$$result = n \cdot l - e; n = \begin{cases} \frac{L_{\max} - L_{\min}}{E_{\max} - E_{\min}} \cdot \frac{\Delta E_{\max}}{\Delta S_{\min}} & \text{if } Max_Error > \Delta E_{\max} \\ 0 & \text{if } Max_Error \leq \Delta E_{\max} \end{cases} \tag{19}$$

3. Precise selection of segmentation points

The improved NMS algorithm processed a set of contour segments: first, we needed to search the contour segment with the highest evaluation and consider the two endpoints as the segmentation points; then, we searched the contour segments that overlapped with the highest rated one, subsequently eliminating both the highest-rated contour segment and the overlapping parts of the contour segments that overlapped with it; and finally, we repeated the above process until all contour segments were eliminated.

The segmentation points were precisely selected by adopting the improved NMS algorithm for experimental use. The segmentation points precisely selected by the NMS algorithm were accurately located. Meanwhile, there were no false or missed detections. The results are shown in Figure 18, each point of a different color represents a segmentation point between different graphic elements.

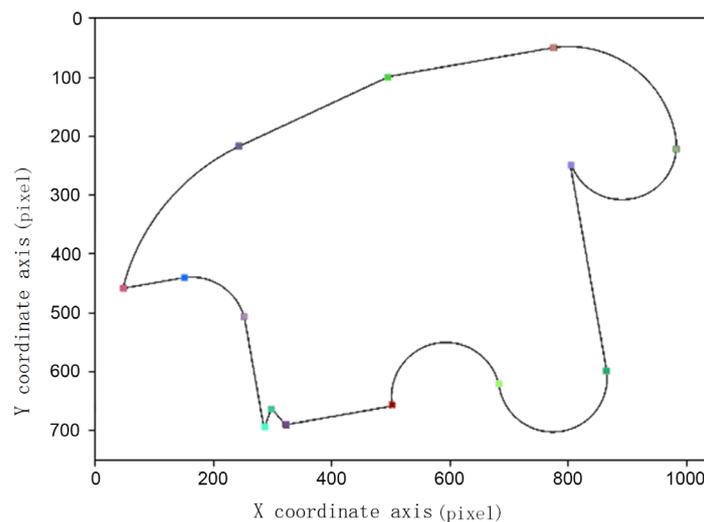


Figure 18. Results for selected segmentation points on image to be tested.

The execution effect of the improved NMS algorithm is shown in Figure 19. The gray and black points are contour points, and the points in the color coil constitute the contour segments after evaluation. The red points are the segmentation points filtered by the NMS algorithm.

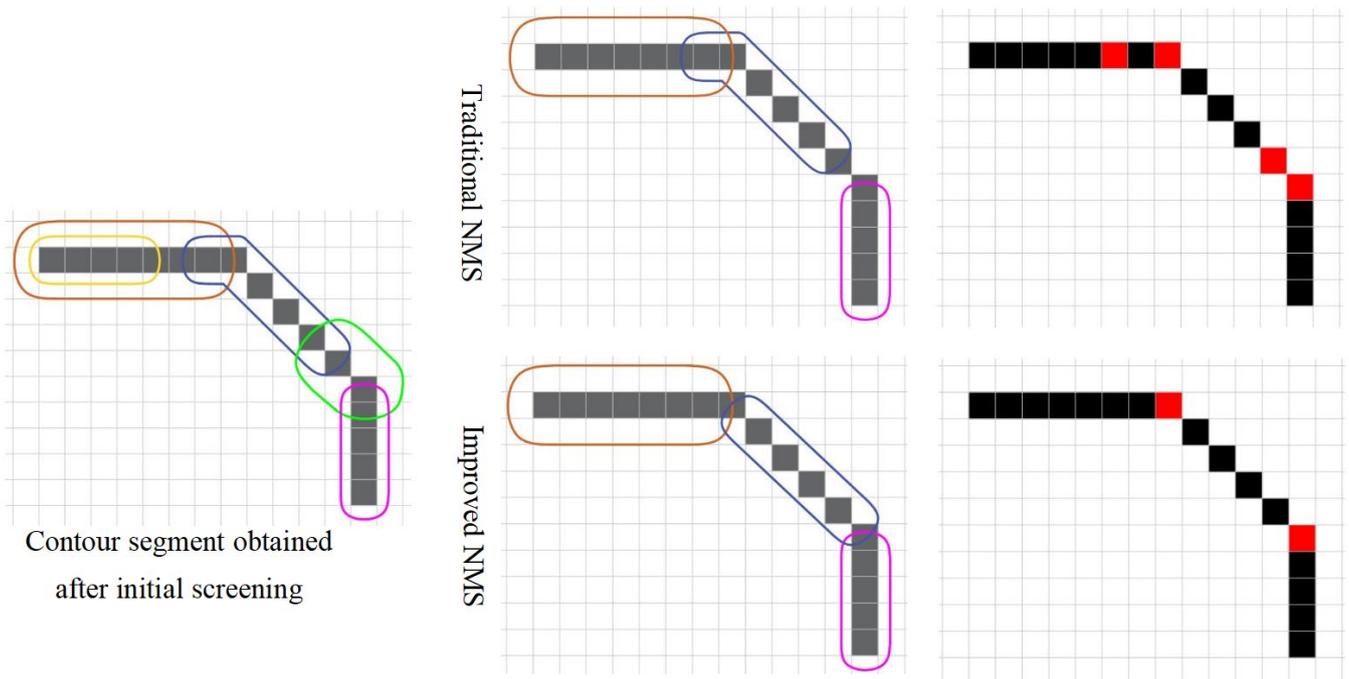


Figure 19. Schematic of the effect of the improved NMS algorithm.

The workflow of the improved NMS algorithm is shown in Figure 20.

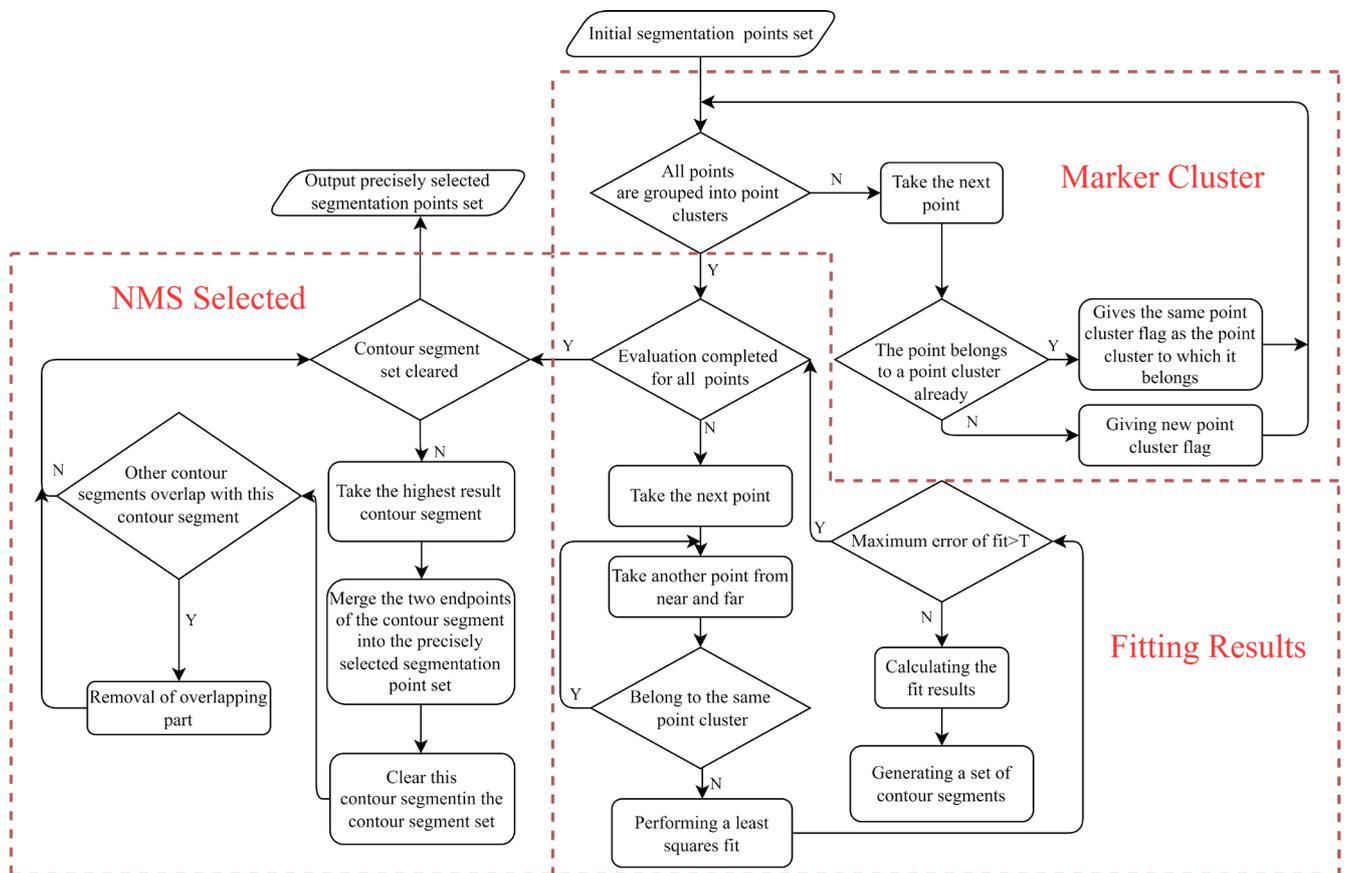


Figure 20. Flow chart of precise selection.

4. Results

The initial screening results of the segmentation points based on the adaptive chord inclination angle are shown in Figure 9 and Section 3.2. The precise screening results of the segmentation points based on the improved NMS are shown in Figure 18 and Section 3.3.

After confirming the location of the segmentation point of the graphic element, the graphic element needed to be fitted to determine its key parameters. If we directly fitted the graphic element contour, it resulted in a situation where the fitting result was not closed, as shown in Figure 21. Therefore, the graphic element was fitted with a two-point constraint. The two endpoints were noted as (x_0, y_0) and (x_1, y_1) .



Figure 21. The graphic element fitting results were not closed.

For a straight-line segment, two points can determine all the parameters of the straight line. The equation of a straight line is given in Equation (20):

$$\begin{cases} y = y_0 & (y_0 = y_1) \\ x = x_0 & (x_0 = x_1) \\ y = k \cdot x + b \end{cases} \quad k = \frac{y_1 - y_0}{x_1 - x_0}, \quad b = y_0 - k \cdot x_0 \quad (20)$$

The inclination angle of a straight-line segment is given in Equation (21):

$$angle = \arctan \left(\frac{\vec{P_0P_1} \times \vec{x}}{\vec{P_0P_1} \cdot \vec{x}} \right) \times 180^\circ \div \pi \quad (21)$$

where \vec{x} is the positive direction of the x-axis.

For a circular arc segment, the equation of a circle is as follows: $x^2 + y^2 + a \cdot x + b \cdot y + c = 0$. Then, the coordinates of the center of the circle are $(-\frac{a}{2}, -\frac{b}{2})$, and the radius is $R = \frac{\sqrt{a^2 + b^2 - 4c}}{2}$. We consider x as the set of x-coordinates of all contour points in the element and y as the set of y-coordinates of all contour points in the element.

When $y_0 \neq y_1$, the coefficients in the circle equation can be calculated by Equations (22)–(24).

$$a = -\frac{J}{\sum H^2} \quad (22)$$

$$b = D + a \cdot E \quad (23)$$

$$c = F + a \cdot G \quad (24)$$

$$D = \frac{(x_0^2 - x_1^2) + (y_0^2 - y_1^2)}{y_1 - y_0}$$

$$E = \frac{x_0 - x_1}{y_1 - y_0}$$

$$F = \frac{(y_1 \cdot x_0^2 - y_0 \cdot x_1^2) + (y_1 \cdot y_0^2 - y_0 \cdot y_1^2)}{y_0 - y_1}$$

$$G = \frac{(x_0 \cdot y_1 - x_1 \cdot y_0)}{y_0 - y_1}$$

$$H = x + E \cdot y + G$$

$$J = \sum \left((x^2 + y^2 + D \cdot y + F) \cdot H \right)$$

When $y_0 = y_1$, the coefficients in the circle equation can be calculated by Equations (25)–(27).

$$a = \frac{(x_0^2 - x_1^2) + (y_0^2 - y_1^2)}{x - x_0} \tag{25}$$

$$b = \frac{E - F}{\sum y^2 - \frac{1}{n}(\sum y)^2} \tag{26}$$

$$c = -\sum \frac{D + b \cdot y}{n} \tag{27}$$

$$D = x^2 + y^2 + a \cdot x$$

$$E = \frac{1}{n} \sum D \cdot \sum y$$

$$F = \sum (D \cdot y)$$

$$n = \text{lenght}(x)$$

The results of fitting to the pending inspection image are shown in Table 4.

Table 4. Results of fitting on experimental plots.

Arc Graphic Elements			Straight-Line Graphic Elements		
Index	Circle Center	Radius	Index	Tilt Angle	Length
l_1	(401.11, 544.51)	362.35	l_2	9.91	104.56
l_3	(164.15, 532.68)	92.48	l_4	−79.39	190.24
l_8	(594.37, 643.38)	92.38	l_5	69.86	32.31
l_9	(775.13, 611.13)	91.66	l_6	−46.12	35.38
l_{11}	(891.96, 216.48)	92.26	l_7	10.04	182.01
l_{12}	(803.02, 230.08)	181.11	l_{10}	99.75	354.12
			l_{13}	−169.87	284.42
			l_{14}	−155.09	277.83

5. Experiment and Analysis

5.1. Relative Error of Graphic Element Fitting

Given Section 2.2, there were a large number of equal relationships in graphic elements for the pending inspection image, and the fitting results in Table 4 were analyzed for relative errors.

From the fitting results, the average radius of the circular arc segment was 91.224. The average length of l_4, l_7, l_{10} was 181.5925. The average length of l_5, l_6 was 33.845. The average length of l_{13}, l_{14} was 281.125. The design tilt angle of l_4, l_{10} was 100° . The design tilt angle of l_2, l_7, l_{13} was 10° . The relative errors are listed in Table 5.

5.2. Effect of Bicubic Interpolation on the Relative Error of the Fit after Processing

The original flat part image was doubled in length and width using bicubic interpolation so that the part element had more contour points as support, which could further improve the fitting accuracy of the element. The effect of interpolation processing on the relative error is shown in Figure 22. It can be seen that the relative error of the measurement results was generally reduced and tended to be stable after the interpolation process. In particular, the relative error was significantly decreased at the short graphic element (indices 4–8).

Table 5. Relative errors of fitting results.

Index	Average Radius (Pixel)	Average Length (Pixel)	Design Tilt (°)	Clamping Angle (°)	Fitted Value(Pixel/°)	Relative Error
l_1	364.896				362.35	0.703%
l_2			10		9.91	0.90%
l_3	91.224				92.48	1.35%
l_4			100		100.61	0.61%
l_4		181.5925			190.24	4.54%
l_5		33.845			32.31	4.75%
l_6		33.845			35.38	4.33%
l_7			10		10.04	0.4%
l_7		181.5925			182.01	0.22%
l_8	91.224				92.38	1.25%
l_9	91.224				91.66	0.47%
l_{10}			100		99.57	0.43%
l_{10}		363.185			354.12	2.55%
l_{11}	91.224				92.26	1.12%
l_{12}	182.448				181.11	0.73%
l_{13}			10		10.13	1.3%
l_{13}		281.125			284.42	1.15%
l_{14}		281.125			277.83	1.18%
l_{14}				165	165.22	0.13%

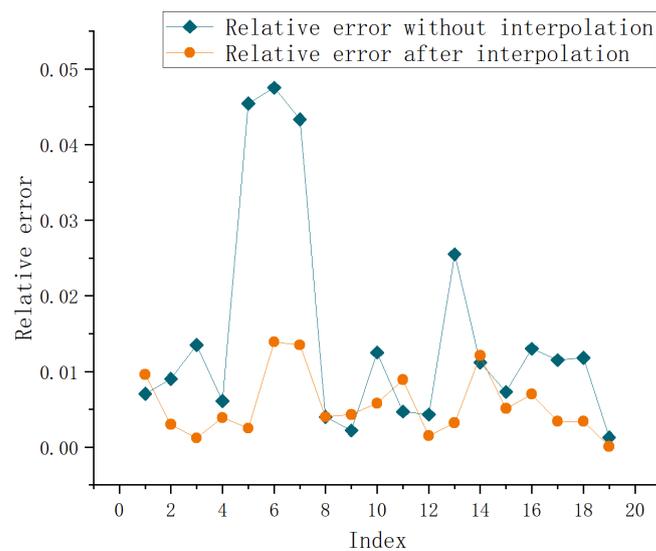


Figure 22. The effect of interpolation processing.

5.3. Robustness and Generalizability

The method proposed in this study could effectively screen the correct segmentation points and accurately identify the tangent points and inflection points. In particular, for the parts with different placement angles that had the same detection effect, the detection method had a certain degree of robustness. The experimental results are shown in Figure 23.

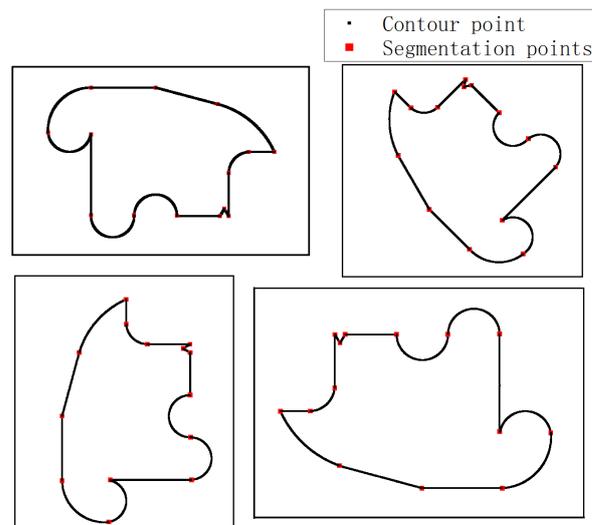


Figure 23. Segmentation points selected from different placement angles.

The effects of contour defects on the segmentation points could be effectively removed by adjusting the fitting tolerance threshold (T) when NMS algorithm selection was adopted. Rectangular parts with more severe boundary contour defects were used to verify the noise immunity capability of this detection method. The results are shown in Figure 24.

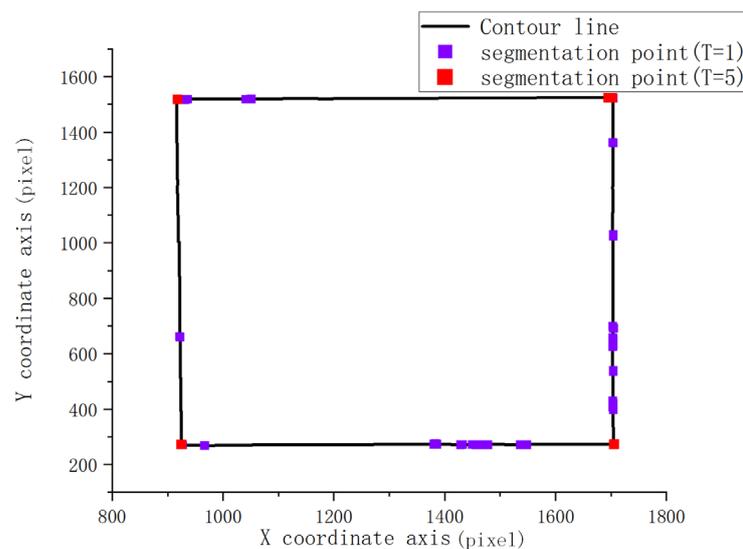


Figure 24. Noise suppression effect of NMS segmentation point selection.

In this study, we processed the following two parts using a laser cutting machine (Figures 25 and 26). After image preprocessing, front and back chord tilt complementation, NMS algorithm selection, and fitting with constraints, the detection effect of the proposed method is shown in Table 6. Four sizes were selected for comparison.

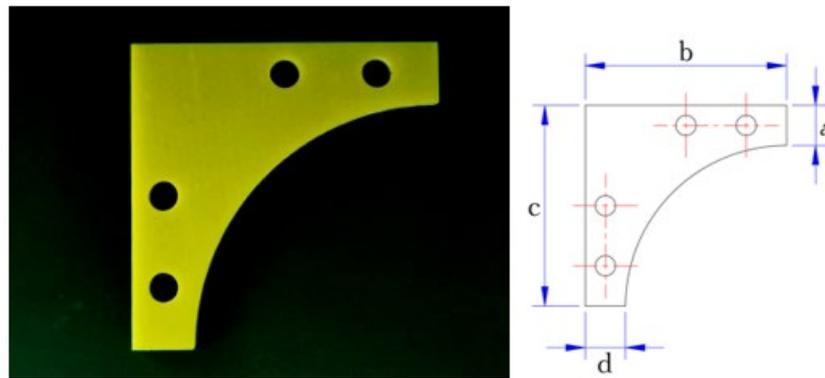


Figure 25. Test part 1.

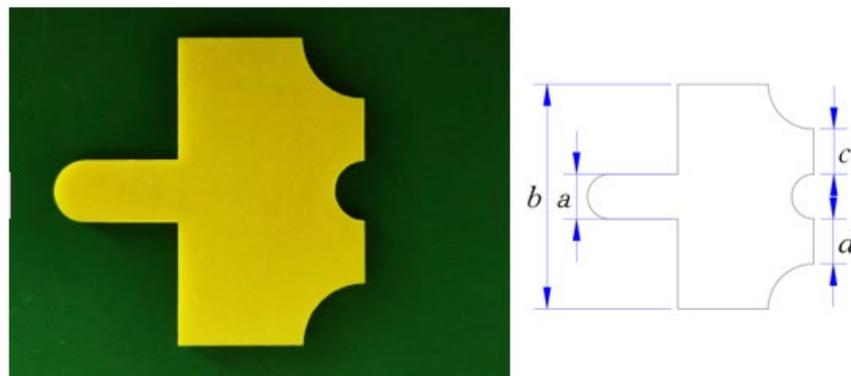


Figure 26. Test part 2.

Table 6. Manual versus visual measurement.

Part	Size	Part Size Value (mm)	Visual Correction Values (mm)	Error (mm)	Relative Error
part 1	a	9.96	10.02	0.06	0.60%
	b	50.10	50.08	0.02	0.03%
	c	50.04	50.08	0.04	0.07%
	d	9.90	9.87	0.03	0.30%
part 2	a	9.90	9.86	0.04	0.40%
	b	49.84	49.90	0.06	0.12%
	c	9.92	9.86	0.06	0.60%
	d	9.96	9.91	0.05	0.50%

As can be seen in Table 6, there was no significant difference between the visual inspection results in this study and the value of part size in practice, and the relative error was less than 0.6%.

6. Conclusions

Segmentation point detection for flat part image contours can segment different elements in the contour to facilitate subsequent graphic element fitting, and the segmentation point, as a constraint for fitting, can effectively avoid the problem of the non-closure of the element fitting results. The main advantages of the segmentation point detection method proposed in this paper include two aspects: on one hand, the initial screening method is sensitive to the points with inconspicuous local features, such as tangent points and inflection points, and there is no leak detection for segmentation points; on the other hand, the selection method can effectively eliminate the pseudo-segmentation points that result from noise and jagged contours, and there is no need for a contour smoothing process,

ensuring that the segmentation point position is accurate, which provides a guarantee for the accuracy of the subsequent element fitting.

The original image was processed by bicubic interpolation, which increased the support pixels of the graphic element and significantly reduced the relative error of the image element fitting. After actual measurement verification, the machine-vision-based flat part measurement method proposed in this paper could obtain high-precision part dimensions, and the relative error was less than 0.6%, which can meet the accuracy requirements of most measurement scenarios.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app13031641/s1>. The supplementary file contains mainly algorithms written in python and experimental data. Code: Algorithmic Program, Image: Pictures used in the experiment, Data: Experimental Data.

Author Contributions: Conceptualization, T.L. and M.J.; methodology, X.L.; software, X.L.; validation, T.L.; formal analysis, T.L.; investigation, X.L.; resources, T.L.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L.; visualization, X.L.; supervision, T.L.; project administration, T.L.; funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Research Program of Science and Technology at Universities of the Inner Mongolia Autonomous Region (2022.12–2025.12) (Funding number: NJZZ23041).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are contained within the article or the Supplementary Materials.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Moru, D.; Borro, D. A Machine Vision Algorithm for Quality Control Inspection of Gears. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 105–123. [CrossRef]
2. Wang, X. Laplacian Operator-Based Edge Detectors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 886–890. [CrossRef] [PubMed]
3. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 679–698. [CrossRef] [PubMed]
4. Nixon, M.S.; Aguado, A.S. *Feature Extraction & Image Processing for Computer Vision*; Academic Press: Cambridge, MA, USA, 2012.
5. Perona, P.; Malik, J. Detecting and Localizing Edges Composed of Steps, Peaks and Roofs. In Proceedings of the Third International Conference on Computer Vision, Osaka, Japan, 4–7 December 1990; pp. 52–57.
6. Martin, D.R.; Fowlkes, C.C.; Malik, J. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 530–549. [CrossRef] [PubMed]
7. Xiaofeng, R.; Bo, L. Discriminatively Trained Sparse Code Gradients for Contour Detection. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
8. Xie, S.; Tu, Z. Holistically-Nested Edge Detection. *Int. J. Comput. Vis.* **2017**, *125*, 3–18. [CrossRef]
9. Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; Tu, Z. Deeply-Supervised Nets. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, San Diego, CA, USA, 9–12 May 2015; pp. 562–570.
10. Tao, J.; Wei, H.; Zhang, K. Measurement Algorithm of Notch Length of Plastic Parts Based on Video. *Meas. Sci. Technol.* **2022**, *33*, 075008. [CrossRef]
11. Suzuki, S.; Abe, K. Topological Structural Analysis of Digitized Binary Images by Border Following. *Comput. Vis. Graph. Image Process.* **1985**, *30*, 32–46. [CrossRef]
12. Awrangjeb, M.; Lu, G. Robust Image Corner Detection Based on the Chord-to-Point Distance Accumulation Technique. *IEEE Trans. Multim.* **2008**, *10*, 1059–1072. [CrossRef]
13. Wang, Y.; Wu, W.; Zhang, J. Segmentation and Recognition Techniques for Planar Contour. *J. Comput.-Aided Des. Comput. Graph.* **2002**, *14*, 1142–1145+1151.
14. Hou, X.; Yang, P.; Zhao, Y. Contour Feature Point Detection Algorithm of CCD Image. *J. UEST China* **2004**, *33*, 446–448.
15. Mo, Q. Research on Machine Vision Measurement in Geometry of Sheet Metal Parts. Master's Thesis, Huazhong University of Science & Technology, Wuhan, China, 2011.
16. Wu, J. Research on Key Technologies in Dimensional Inspection System of Thin Sheet Parts Based on Machine Vision. Ph.D. Thesis, Huazhong University of Science & Technology, Wuhan, China, 2008.

17. Zhou, C. Research of Parts Size Measurement System Based on Vision Technology. Master's Thesis, Zhejiang Sci-Tech University, Hangzhou, China, 2016.
18. Qu, Y.; Liu, R.; Luan, R. Recognition of machining position of mechanical parts based on shape and position feature. *Sci. Technol. Eng.* **2019**, *19*, 276–282.
19. Rachmawati, E.; Supriana, I.; Khodra, M.L. FAST Corner Detection in Polygonal Approximation of Shape. In Proceedings of the 2017 3rd International Conference on Science in Information Technology (ICSITech), Bandung, Indonesia, 25–26 October 2017; pp. 166–170.
20. Liu, Y. Research on feature and constraint based visual reverse technology for two dimensional parts. Master's Thesis, Harbin Institute of Technology, Harbin, China, 2013.
21. Wang, Y. Research of reverse technology for accurate parts based on two dimensional computer vision. Master's Thesis, Harbin Institute of Technology, Harbin, China, 2013.
22. Chen, Y.; Qi, F. A new approach to extract feature points on the boundary. *J. Infrared Millim. Waves* **1998**, *17*, 13–18.
23. Dou, Y.; Huang, R.; Li, T.; Zhao, X. Representation method of outlines by using critical points. *Appl. Res. Comput.* **2015**, *32*, 1524–1526+1539.
24. Chetverikov, D. A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves. In Proceedings of the Computer Analysis es and Patterns; Petkov, N., Westenberg, M.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 746–753.
25. Shi, J.; Tian, Y.; Sun, Y.; Zhang, J. Corner Detection Algorithm Based on Image Contour Curve. *Meas. Control. Technol.* **2020**, *39*, 91–96+105. [[CrossRef](#)]
26. Huang, S.; Qin, H.; Zhou, Z. Vision Measurement for Flat Parts Based on Local Line-Angle Contour Segmentation. *Meas. Sci. Technol.* **2022**, *33*, 085012. [[CrossRef](#)]
27. Li, Y.; Du, W.; Wang, J.; Duan, N.; Guan, B.; Tang, H.; Li, B. Research on the method of planar contour segmentation and recognition of the geometric primitives. *Manuf. Technol. Mach. Tool* **2016**, 85–88. [[CrossRef](#)]
28. Wang, Y.; Wu, W. Accurate Reconstruction for Planar Contour Based on Segmentation and Recognition Techniques. *J. Eng. Graph.* **2007**, 43–48. [[CrossRef](#)]
29. Joiner, B.L. Data Analysis and Regression. *J. Qual. Technol.* **1979**, *11*, 102–103. [[CrossRef](#)]
30. Khilil, A.; Shi, Z.; Umar, A.; Ma, B. Improved Algorithm for Minimum Zone of Roundness Error Evaluation Using Alternating Exchange Approach. *Meas. Sci. Technol.* **2022**, *33*, 045003. [[CrossRef](#)]
31. Jing, J.; Sun, J.; Zhang, W.; Bai, M. Image corner detection method based on multi-scale discrete curvature. *J. Xi'an Polytech. Univ.* **2021**, *35*, 27–33.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.