

Measurement and Inspection of Roundness Using Computer Vision

قياس وفحص الاستدارة باستخدام الرؤية بالحاسب

E.S. Gadelmawla*, W.M. Khalifa, I.M. Elewa

Production Engineering and Mechanical Design Department, Faculty of Engineering,
Mansoura University, Mansoura 35516, Egypt

* Email: esamy@mans.edu.eg

الملخص العربي:

يعتبر قياس وفحص الاستدارة للمنتجات الدائرية من المشاكل الشائعة في اختبارات الجودة لأن الخطأ في الاستدارة يمكن أن يؤثر بشكل كبير على دقة ووظيفة الأجزاء الدائرية المجمع. لذلك توجد حاجة مستمرة إلى تطوير طرق الفحص الآلي الخاصة بقياس وفحص الاستدارة. ويقدم هذا البحث نظاما للرؤية بالحاسب تم تطويره في معمل القياسات الميكانيكية بكلية الهندسة جامعة المنصورة كأسلوب جديد لقياس وفحص استدارة المنتجات. ويستخدم هذا النظام الالتقاط صور للمنتجات المراد قياسها أو فحصها، ثم يتم تحليل هذه الصور باستخدام برنامج تم تطويره خصيصا لقياس وفحص الاستدارة للصور الملتقطة. وقد تم تطوير هذا البرنامج كاملا في المعمل باستخدام لغة البرمجة (Visual C++) بحيث يعمل بكفاءة مع أنظمة التشغيل ويندوز. ويتميز النظام المقدم بإجراء عملية القياس والفحص للمنتجات الدائرية بدون تلامس، وهذا عكس معظم الأنظمة التقليدية المستخدمة لنفس الغرض. بالإضافة إلى ذلك يتميز هذا النظام بإمكانية قياس وفحص كل الأجزاء الدائرية التي يمكن أن توجد في الصور الملتقطة للمنتجات، مثل الثقوب الداخلية. وقد تم معايرة النظام لاستخدام النظام المترى أو النقطة (Pixel) المستخدمة في نظام الصور. واختبار دقة النظام المقدم تم قياس الاستدارة لمجموعة من العينات باستخدام هذا النظام، ثم تم قياس نفس العينات باستخدام أحد الأنظمة التجارية المعروفة لقياس الاستدارة (Roundtest RA-100). وبمقارنة النتائج التي تم الحصول عليها من النظامين وجد أن نسبة الاختلاف بين النتائج كانت في حدود $\pm 7.22\%$.

Abstract:

A common problem of quality control and inspection of circular parts is the measurement of their roundness. Errors in rounded objects can greatly affect the accuracy and function of assemblies. Therefore, there is a requirement to develop automatic inspection methods for the inspection of roundness. This paper introduces a new approach for measurement and inspection of roundness using computer vision. The vision system has been established in the metrology lab and it is used to capture images for objects to be measured or inspected. A software (named *RoundVision*) has been especially developed in-house using Microsoft Visual C++ to analyze the captured images and perform the measurement and inspection processes. Unlike the conventional methods, the proposed system has the advantage of being non contact method and much near to the reality because all points on the contour of the objects to be measured are considered. In addition, the system is capable of measuring and inspecting multiple rounded features in the same part, whether it represents the external frame or internal holes. The system has been calibrated for both metric and pixel units. To verify the introduced system, sample circular parts were used to measure their roundness error by the introduced system and a commercial system (Roundtest RA-100), then the results were compared. The maximum difference between the results of the two systems was $\pm 7.22\%$.

Keywords: Circularity, Computer vision, Image processing, Inspection, Roundness.

1. Introduction

Circular feature is one of the most basic geometric elements of mechanical parts. Whether or not circularity error can be evaluated accurately and efficiently it will directly influence the mechanical products performance and life [1].

According to the geometric roundness tolerance standard, the whole profile of rounded parts must be measured to fully describe the characteristics of roundness [2]. For conventional instruments, the number of sampling points is limited in practical measurement. Therefore, it is very difficult, even impossible for these instruments to perform the roundness measurements according to the present standards. Furthermore, measuring the whole profile by these instruments complicates the measurement process and increase the time of the measurement process. To meet the geometric roundness tolerance standard, the coordinate measuring machines (CMMs) have been recently used in automated inspection for both the on-line and off-line inspection of manufactured components. It can be used to measure roundness errors by collecting a large number of sampling points from the profile of the rounded parts. However, the coordinate measuring process of CMMs is expensive and very time-consuming if a desirable large set of measured points is collected.

With the advent of low-cost digital computers, machine vision systems have emerged as a financially viable device in automated manufacturing systems. In addition, the developments in image processing, computer vision, artificial intelligence and other related fields have significantly improved the capability of visual inspection techniques [3, 4]. A machine vision system with a typical 512x512 image processor can collect a huge amount of boundary points within a reasonably short time if compared with the CMMs. Furthermore, machine vision can realize the non-contact inspection of products while most roundness measuring

instruments are contact-type. The development of a machine vision system for inspection has received considerable attention [5-9]. The advantages of using such automatic systems include a decrease in the time required for measurement as well as the greater accuracy of measurement and better flexibility than the conventional methods.

Because machine vision is only an accurate and efficient data antenna, it requires assessment algorithms for distinct inspection functions. Therefore, the study of effective algorithms specific to manufactured parts and error analysis becomes imperative in vision-based inspection. This work aims to introduce a set of computer vision algorithms that can be used to measure and inspect the roundness error from the captured images of circular parts.

2. Background of Roundness

Roundness measurements are primarily classified into two categories [10]: diametral roundness measurements and radial roundness measurements. Diametral roundness measurements can be conducted using two-point measuring instruments such as micrometers, calipers or indicating snap gages. The primary error of two-point measuring process is that the measured points are limited in practice, and the midpoints are not located at the same position in a plane. Radial roundness measurements are conducted using precision spindle instruments, which are sophisticated, expensive, and time-consuming.

According to the ISO 1101 standard, a roundness tolerance band is the zone bounded by two concentric circles at the same cross-section whose radii difference is equal to the roundness tolerance value [11]. To realize the measurement of this distance, a reference circle must be established from the measurement data to minimize the maximum deviation between the reference feature and the actual feature. Four

reference circles are internationally accepted for roundness measurements as shown in Fig. 1 [12]. These are: Maximum inscribing circle (MIC), Minimum circumscribing circle (MCC), Minimum zone circles (MZC) and Least squares circle (LSC).

The MIC is the largest circle just contained by the profile of a workpiece. It is known as the plug gauge circle and is intended for shafts. The MCC is the smallest circle that just contains the profile of a workpiece and it is also known as the ring gauge circle and is intended for holes. The requirement for finding the MZC is to determine a pair of concentric circles, the annular zone between which is the narrowest and which will just contain the profile of a workpiece. The roundness error of the MZC is then the width of the zone. The LSC is the circle chosen such that the sum of the squares of the radial distance of all data points from the fitting circle is a minimum.

The center position and the radius of the reference circle are important parameters to be estimated from the standpoint of their function in an assembly. In general, all four reference circles will have different parameters for any given profile.

In a previous work [13], the algorithms and the mathematical calculations of the first three methods (MIC, MCC, and MZC) were discussed in details. In this paper, the algorithms and the mathematical calculations of the fourth method (LSC) is explained and applied to calculate the roundness error using computer vision.

3. Least Square Mathematics

The method of least squares is used to solve a set of linear equations having more equations than unknown variables. If the equations are linear, the least-squares process will produce a direct solution for the unknowns. If the equations are not linear, an initial guess of the unknowns is required, and the result is an adjustment to the initial parameters. This is repeated until the results converge (the adjustments become very close to zero). The linear case is an adjustment using zero as the initial guess of all parameters.

Many techniques were developed to evaluate the circularity problem [14-19]. The most commonly used methodology involves the fitting of the least square circle [20-22] to the coordinate measuring data. This methodology is coded on most coordinate measurement machines (CMM) software.

The general equation of the circle is defined by equation 1. Where (x_0, y_0) is the center of the circle and r is the radius of the circle.

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} = r \quad (1)$$

For a given set of two-dimensional points denoted by (x_i, y_i) , the least-squares method for circularity evaluation is to find the right center location to minimize the objective function value, which is the sum of the squared errors of the data points from the fitted circle [23]. Thus, the objective function could be written as follows:

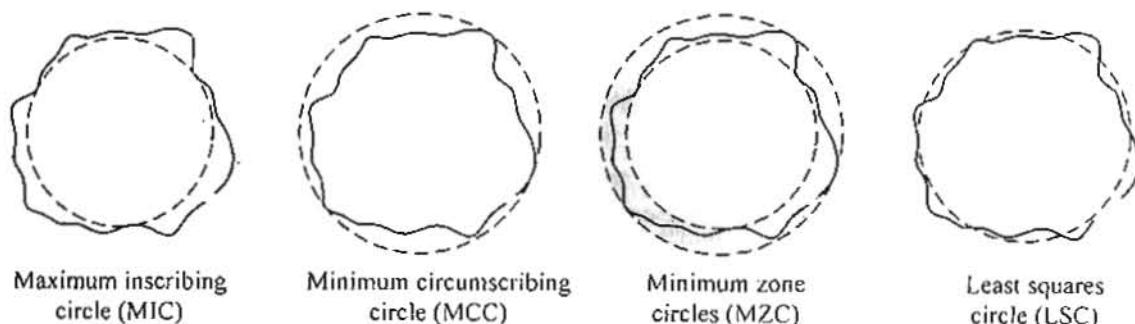


Fig.1: Reference circles accepted for roundness measurements

$$F_i(x_0, y_0, r) = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r = 0 \quad (2)$$

3.1 Linear solution:

The equation of a circle is not linear in the unknown values x_0 , y_0 and r and it can be written in the linear form as follows:

$$A(x^2 + y^2) + Bx + Cy = 1 \quad (3)$$

Where:

$$A = \frac{1}{r^2 - x_0^2 - y_0^2}, \quad B = \frac{-2x_0}{r^2 - x_0^2 - y_0^2},$$

$$C = \frac{-2y_0}{r^2 - x_0^2 - y_0^2}$$

The original unknowns can be calculated from equations 4, 5, and 6:

$$x_0 = \frac{-B}{2A} \quad (4)$$

$$y_0 = \frac{-C}{2A} \quad (5)$$

$$r = \frac{\sqrt{4A + B^2 + C^2}}{2A} \quad (6)$$

The unknowns A , B , and C can be solved using equation 7.

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_i (a_i^2) & \sum_i (a_i b_i) & \sum_i (a_i c_i) \\ \sum_i (a_i b_i) & \sum_i (b_i^2) & \sum_i (b_i c_i) \\ \sum_i (a_i c_i) & \sum_i (b_i c_i) & \sum_i (c_i^2) \end{bmatrix}^{-1} \begin{bmatrix} \sum_i a_i \\ \sum_i b_i \\ \sum_i c_i \end{bmatrix} \quad (7)$$

Where $a_i = (x_i^2 + y_i^2)$, $b_i = x_i$, $c_i = y_i$

Once the unknowns are calculated, the center and the radius of the circle can be calculated using equations 4, 5, and 6.

3.2 Nonlinear solution:

Instead of solving for the linear parameters A , B , and C , it is more desirable to solve for the circle parameters x_0 , y_0 , and r using an equation that minimizes the distance from the points to the circle. For the nonlinear solution, equation 8 is used.

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} - r = 0 \quad (8)$$

The solution of the circle parameters x_0 , y_0 , and r require multiple iterations. A starting guess for these values is required. In this paper, the calculated values obtained from the linear solution are used as the starting values. The adjustments of the unknowns Δx_0 , Δy_0 , and Δr can be solved using equation 9.

$$\begin{bmatrix} \Delta x_0 \\ \Delta y_0 \\ \Delta r \end{bmatrix} = \begin{bmatrix} \sum_i (a_i^2) & \sum_i (a_i b_i) & \sum_i (a_i c_i) \\ \sum_i (a_i b_i) & \sum_i (b_i^2) & \sum_i (b_i c_i) \\ \sum_i (a_i c_i) & \sum_i (b_i c_i) & \sum_i (c_i^2) \end{bmatrix}^{-1} \begin{bmatrix} \sum_i (a_i k_i) \\ \sum_i (b_i k_i) \\ \sum_i (c_i k_i) \end{bmatrix} \quad (9)$$

Where:

$$a_i = \frac{x_0 - x_i}{\sqrt{x_i^2 - 2x_0x_i + y_i^2 - 2y_0y_i + x_0^2 + y_0^2}}$$

$$b_i = \frac{y_0 - y_i}{\sqrt{x_i^2 - 2x_0x_i + y_i^2 - 2y_0y_i + x_0^2 + y_0^2}}$$

$$c_i = -1,$$

$$k_i = 0 - \left(\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r \right)$$

Once the unknowns are calculated, the circle parameters are adjusted to $(x_0 + \Delta x_0)$, $(y_0 + \Delta y_0)$, $(r + \Delta r)$, respectively. The new circle parameters are used to recalculate the adjustments of the unknowns Δx_0 , Δy_0 , and r . This iteration process is performed until the adjustment values become close to zero.

4. The Proposed System

Fig. 2 shows a photograph of the proposed vision system. It consists of two main parts, hardware and software as follows:

4.1. System Hardware

By referring to Fig. 1, the capturing system consists of three items. The first item is the backlighting table (1), which is a lighting box with diffusing surface at its front, and it is used to produce a back lighting for the product to be measured (2). The second item is a CCD color video camera (3) and a set of lenses with different

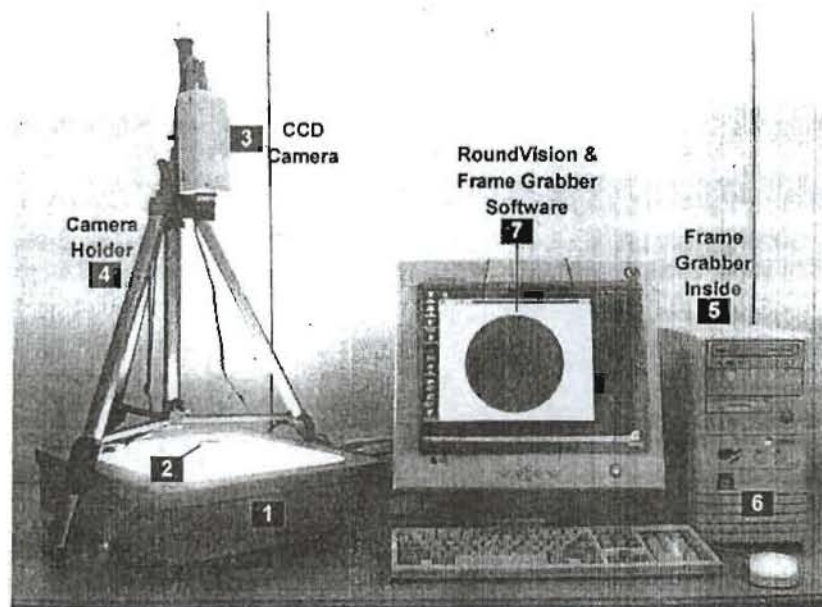


Fig 2: Photograph of the proposed vision system

focal lengths. The camera is carried by a camera holder (4). The third item is a 24 bit per pixel (ELF VGA) frame grabber video card (5), which is installed inside the PC computer (6) and connected to the CCD camera. A capturing software (7) is provided with the frame grabber to acquire images and save it to files with various types of file formats.

4.2. The Developed Software

The developed software (named *RoundVision*) is fully written in-house using Microsoft Visual C++ as a 32-bit Windows application. It features many image processing and computer vision algorithms to measure and inspect the roundness error of circular parts from their captured images. Fig. 3 shows the main interface of the *RoundVision* software.

To perform the measurement or inspection process, the object to be measured or inspected is set on the backlighting table then an image is captured and saved to a BMP file using the capturing software. The captured image is then opened by the *RoundVision* software and analyzed to perform the measurement and inspection processes.

5. Software Algorithms

To perform the measurement and the inspection processes, two inputs are required from the user through the main interface shown in figure 3. The first input is the diameter of the object to be measured and it is used to calibrate the system. The second input is the maximum allowable value for roundness error and it is used for inspection. Several image processing and computer vision algorithms are applied to the captured image to perform the measurement and inspection processes. Fig. 4 shows a process flowchart of these algorithms and the next sections describe them.

5.1. Image Segmentation

The captured images obtained by the frame grabber are usually color images. To measure the roundness of circular objects, only the edge pixels of these circular objects should be considered. This can be done by converting the color images into binary images, in which only two colors are presented (black for the object and white for the background), then finding the edge pixels using an edge detection algorithm as it will be discussed in the next section.

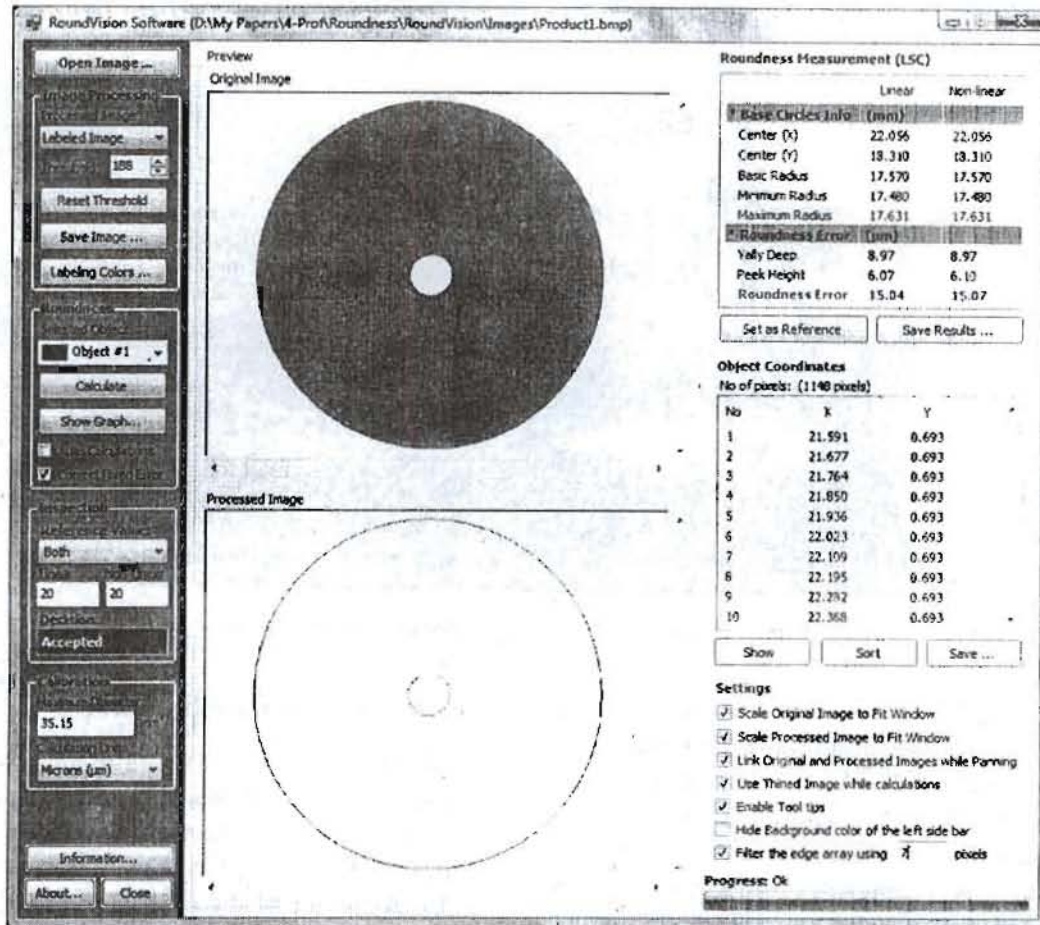


Fig. 3: Main interface of the RoundVision software

Two steps are required to convert color images into binary images: (1) converting the color image to gray image and (2) calculating a threshold value to distinguish object pixels from background pixels. The color image can be converted into gray image by setting the three components of each pixel's color (Red, Green, and Blue) to the same value according to the following equation:

$$G_I = .299R_c + .587G_c + .114B_c$$

Where: G_I is the gray level of the pixel; R_c , G_c , and B_c are the red, green, and blue components of the pixel color, respectively.

Once the color image is converted to gray image, the frequency of each gray level is calculated then the two gray levels having the maximum frequencies are used to calculate a threshold value as follows:

$$T = Abs((G_{I_{max1}} - G_{I_{max2}}) / 2)$$

Where: T is the threshold value, $G_{I_{max1}}$ and $G_{I_{max2}}$ are the two gray levels having the maximum two frequencies.

To obtain the binary image, all pixels having gray levels greater than the threshold are converted to black pixels and other pixels are converted to white pixels.

5.2. The Edge Detection Algorithm

The edge detection algorithm creates an edge image contains only the edge pixels in the binary image. It scans the binary image and compares the color of each pixel with the colors of its eight neighbor pixels. If the color of the checked pixel is 0 (black) and the color of any of the eight neighbor pixels is 255 (white), then the pixel is marked as edge pixel. Otherwise, the pixel is marked as background pixel. This algorithm can be programmed as follows:

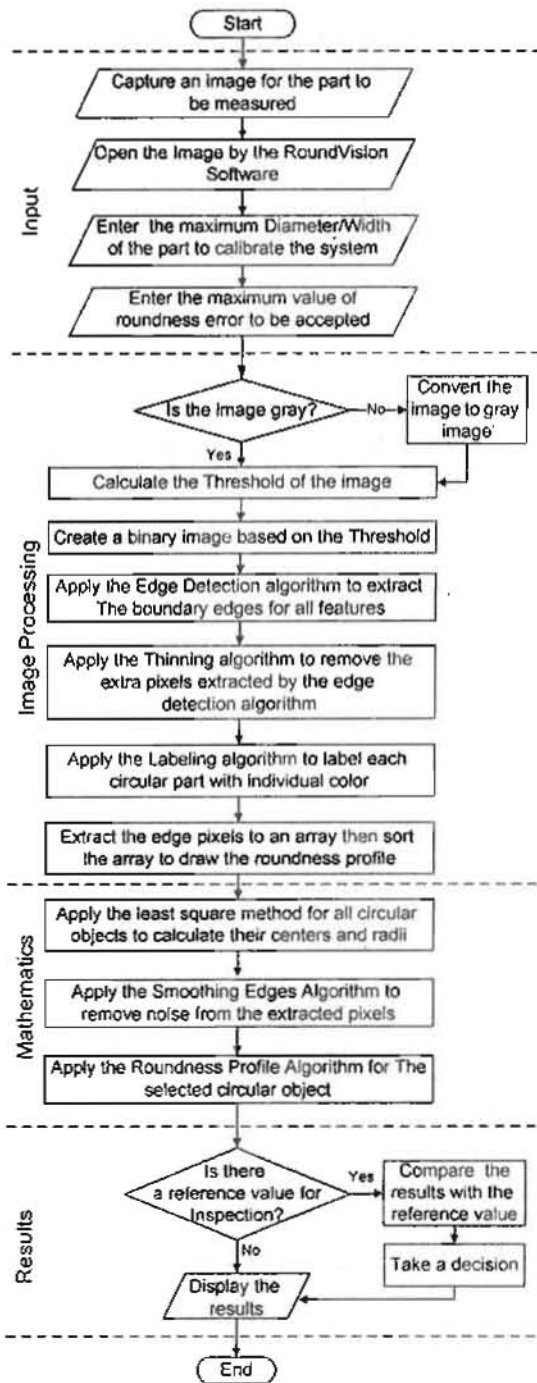


Fig 4. Process flowchart of the RoundVision algorithms

If $(C_{cp} = 0 \text{ AND } C_{n1} \neq 255)$ Then
 Mark the pixel as an edge ($C_{cp}=0$)
 Else
 Mark the pixel as a background
 ($C_{cp}=255$)

Where C_{cp} : is the color of the current pixel (pixel to be checked), $C_{n1:8}$ is the

color of any of the eight neighbor pixels, 0 represents the black color, and 255 represents the white color.

Fig. 5-a shows an enlarged part of a circular part image obtained by the edge detection algorithm. As shown in the figure, the obtained edges are relatively thick due to the detection of some pixels which satisfy the above condition and located inside the circular part. These pixels will increase the calculated roundness error when the least square method is applied. To overcome this problem, the edge thinning algorithm was developed to remove these extra pixels from the detected edges.

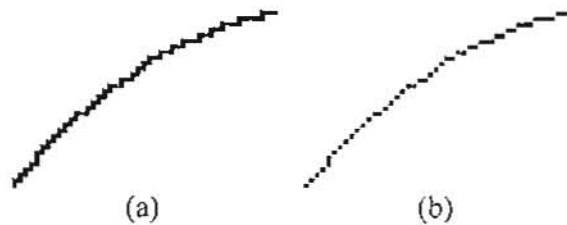


Fig 5: Results of the Edge Detection (a) and the Edge Thinning (b) algorithms

5.3. The Edge Thinning Algorithm

The edge thinning algorithm searches the edge image and compares the color of each pixel with the colors of their eight neighbor pixels. If any two successive neighbor pixels (figure 6) have the same color as the pixel to be checked, then the pixel is marked as a background pixel. This condition can be written programmatically as follows:

If $(C_{cp}=0 \text{ AND } (C_{n1}=0 \text{ OR } C_{n2}=0) \text{ OR } (C_{cp}=0 \text{ AND } (C_{n3}=0 \text{ OR } C_{n4}=0) \text{ OR } (C_{cp}=0 \text{ AND } (C_{n5}=0 \text{ OR } C_{n6}=0) \text{ OR } (C_{cp}=0 \text{ AND } (C_{n7}=0 \text{ OR } C_{n8}=0)))$

Then

Mark the pixel as a background
 $(C_{cp}=255)$

Where: C_{cp} is the color of the current pixel (pixel to be checked), $C_{n1}, C_{n2}, \dots, C_{n8}$ are the colors of the eight neighbor pixels.

Fig. 5-b shows the results of applying the edge thinning algorithm to the same image in figure 5-a.

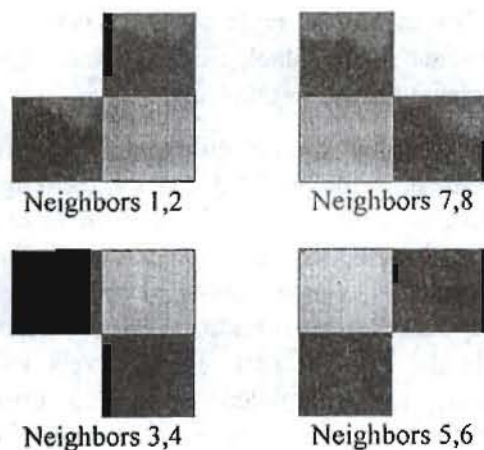


Fig. 6: Gray pixels are removed by the edge thinning algorithm in these cases

5.4. The Edge Labeling Algorithm

The captured images may contain more than one circular part; therefore, the edge pixels of each circular part should be labeled individually so that the roundness error of each part can be calculated separately. The edge labeling algorithm is used to create a labeled image from the edge image, in which the pixels of each circular part are labeled with different color. This algorithm uses the general procedure outlined in reference [24], which can be summarized as follows:

- 1- Apply Run-length algorithm to encode the edge pixels image.
- 2- Scan the runs and assign preliminary labels to each run, then record the label equivalences in a local equivalence table.
- 3- Resolve the equivalence classes.
- 4- Relabel the runs based on the resolved equivalence classes.

5.5. The Roundness Error Algorithm

The roundness error algorithm applies the least square method, discussed in section 3, to calculate the roundness error as follows:

- 1- The labeled image is scanned using left to right and top to bottom order, then the coordinates (x, y) of the edge pixels of a selected part (color) are extracted and stored to an array called *EdgePixels*.

- 2- The least square method is applied to the pixels stored in the *EdgePixels* array to calculate the center and radius of the least square circle.
- 3- The distances between the calculated center and all pixels in the *EdgePixels* array are calculated, then the minimum and the maximum distances are obtained (R_{min}, R_{max}).
- 4- The roundness error is calculated by subtracting R_{min} from R_{max} ($R_{max} - R_{min}$).

5.6. The Roundness Profile Algorithm

The pixels stored in the *EdgePixels* array are usually not sorted due to the scanning process discussed above. This does not affect the calculation of the roundness error; however, the roundness profile cannot be drawn unless these pixels are sorted sequentially. Therefore, the roundness profile algorithm is used to perform this task.

If *EP* represents the *EdgePixels* array. *N* is the number of pixels in this array, *GetDist* is a function to calculate the distance between two pixels; *i, j*, *CurDist*, *MinDist* and *tempP* are local programming variables, then the following c code can be used to sort the pixels in *EP* array sequentially:

```
for (i = 0; i < N-1; i++)
{
  MinDist = GetDist(EP[i],EP[i+1]);
  for (j = i + 2; j < N; j++)
  { CurDist = GetDist(EP[i],EP[j]);
    if (CurDist < MinDist)
    { tempP = EP[i + 1];
      EP[i + 1] =EP[j];
      EP[j] = tempP;
      MinDist = CurDist;
    }
  }
}
```

Once the *EdgePixels* array is sorted, the roundness profile is drawn by connecting each two successive pixels by a short line. Figure 7 shows the roundness profile of a sample circular part drawn by this algorithm.

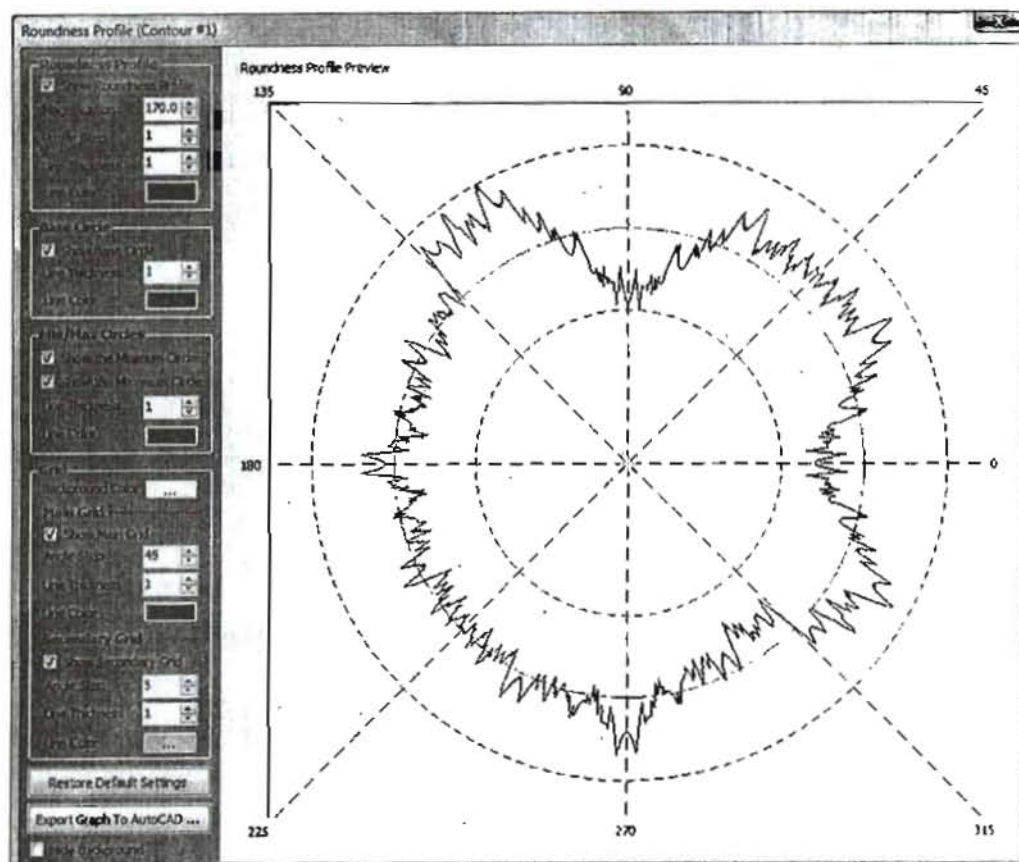


Fig. 7: Roundness profile of a sample circular part

5.7. The Edge Smoothing Algorithm

Due to the digitization (capturing images) of the actual profile of the object to be measured, the extracted edge pixels may include some noise, which affect the roundness error and the roundness profile. The edge smoothing algorithm is used to filter the edge pixels by adjusting their coordinates so that the coordinates of each pixel are adjusted to be equal to the average of the coordinates of two or more successive pixels. The number of pixels used to calculate the average coordinates is called smoothing factor. A smoothing factor equal to five means that the coordinates of each pixel will be set to the average of the coordinates of five successive pixels. Figure 8 shows the effect of the smoothing factor on the roundness profile of a sample circular part. The following c code can be used to apply the smoothing edges algorithm.

```

for (i = 0; i < N; i++)
{
    SumX = 0; SumY = 0;
    for (j = i; j < i + SF; j++)
    {
        if (j < 0)
        { SumX += Xi[N - j];
          SumY += Yi[N - j];
        }
        else if (j > N - 1)
        { SumX += Xi[j - N];
          SumY += Yi[j - N];
        }
        else
        { SumX += Xi[j];
          SumY += Yi[j];
        }
    }
    tempX[i] = SumX / SF;
    tempY[i] = SumY / SF;
}
for (i = 0; i < N; i++)
{
    Xi[i] = tempX[i];
    Yi[i] = tempY[i];
}

```

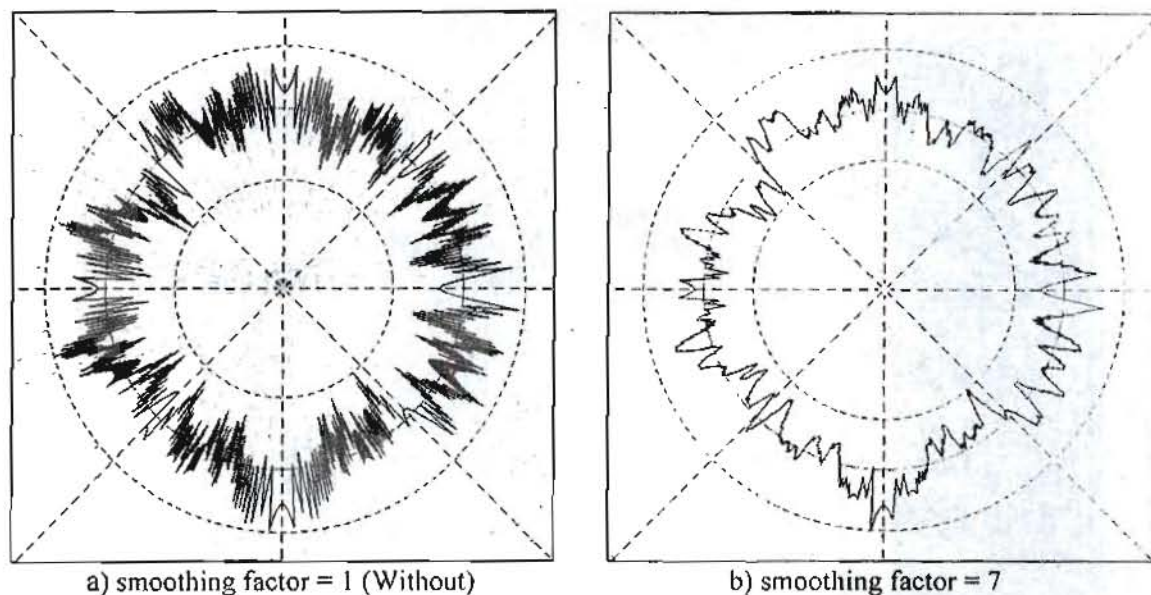


Fig. 8: Effect of the smoothing factor on the roundness profile

Where: SF is the smoothing factor; i, j , $SumX$, $SumY$ are local programming variables, N is the number of edge pixels of the circular part, $Xi[]$, $Yi[]$ are the arrays store the x and y coordinates of the edge pixels; $tempX$ and $tempY$ are temporary arrays to store the calculated averages.

To study the effect of the smoothing factor on the roundness error, a perfect circle (roundness error = zero, theoretically) was draw by an image processing software then the image of this circle was opened by the RoundVision software and the roundness error was calculated for various values of the smoothing factor. Figure 9 shows the effect of the smoothing factor on the calculated roundness error. It is clear that increasing the smoothing factor decreases the calculated roundness error up to a certain value (15 pixels), then its effect can be neglected.

Although the expected roundness error for a perfect circle is theoretically zero, there was a fixed error for each value of the smoothing factor. The fixed error ranged between 1.978 (for $SF=1$) and 1.136 (for $SF=15$) pixels. Therefore, this fixed error should be taken into consideration while calculating the actual roundness error. The equation of correlation between the fixed

error (FE) and the smoothing factor was obtained from Excel to be used during the calibration process.

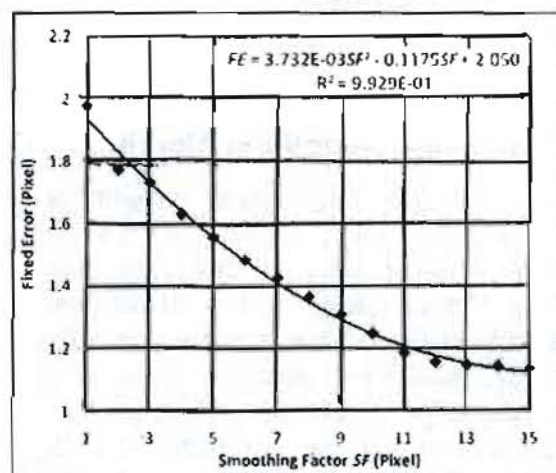


Fig. 9: Effect of the smoothing factor on the calculated roundness error

6. System Calibration

The system calibration is performed by calculating the pixel size in both x and y directions according to the actual size of the object to be measured as follow:

- i- The user enters the actual diameter (D_{actual}) of the measured object (in millimeter) to the RoundVision software. If the measured object contains more

than one circular part, the size of the outer part (maximum diameter) should be entered.

- 2- The software searches the edge pixels of the outer contour to find the two pixels having the minimum and the maximum x coordinates, then it calculates the maximum diameter of the captured image in x direction (D_{max_x}) as follows:

$$D_{max_x} = Abs(X_{max} - X_{min})$$

Where X_{max} and X_{min} are the maximum and minimum x coordinates of the edge pixels of the outer contour, respectively.

- 3- A calibration factor in x direction (CF_x) is calculated as follows:

$$CF_x = D_{actual} / D_{max_x}$$

- 4- Similarly, a calibration factor in y direction (CF_y) is calculated by applying step 2 to find the two pixels having the minimum and the maximum y coordinates, then CF_y is calculated as follows:

$$CF_y = D_{actual} / D_{max_y}$$

- 5- All x coordinates of the edge pixels are multiplied by CF_x and all y coordinates are multiplied by CF_y .
6. A primary roundness error ($RE_{primary}$) is calculated by applying the least square circle technique to the edge pixels.
- 7- Using the equation of correlation between the fixed error (FE) and the smoothing factor (SF), which was obtained from Excel, the fixed error is calculated in pixel according to the smoothing factor entered by the user as follows:

$$FE_{pixel} = 3.732E^{-3}SF - 0.1175SF + 2.050$$

8. The fixed error is then calibrated and calculated in millimeters (FE_{mm}) as follows:

$$FE_{mm} = FE_{pixel} \sqrt{CF_x * CF_y}$$

9. The final roundness error (RE_{final}) is calculated as follows:

$$RE_{final} = RE_{primary} - FE_{mm}$$

7. System Verification

To verify the introduced system, three specimens with 25.4 mm diameter were measured by a commercial roundness instrument (Roundtest RA-100) and the introduced system. The results are listed in table 1. It can be seen that the difference between the results obtained by the two systems does not exceed $\pm 4.4\%$.

8. Inspection of Roundness

The inspection process is performed by comparing the calculated roundness error with a reference value, which can be set through the Inspection section in the main interface shown in figure 3. The reference value could be selected as linear, non-linear, or both least square methods from the Reference Value combo box, then the maximum allowable error for each method is entered. If the calculated roundness error is less than or equal to the reference value, then the inspection decision will be "Accepted", otherwise it will be "Rejected".

The calculated roundness error of any circular part can be set as a reference value for inspection by clicking the Set as Reference button in the Roundness Measurement section.

Table 1: Comparison between the results obtained by Roundtest RA-100 and the introduced system

Specimen No.	Roundtest instrument (μm)	Vision system (μm)	Difference (μm)	Error %
1	3.89	4.062	-0.172	-4.42
2	4.12	3.941	0.179	4.34
3	4.50	4.175	0.325	7.22

9. Conclusions

A vision system has been introduced as a new approach for measurement and inspection of roundness error of circular parts from their captured images. The system has the advantages of being non-contact method and satisfying the geometric roundness tolerance standard. A software has been developed in-house to analyze the captured images and perform the measurement and inspection processes using many image processing and computer vision algorithms.

The system has been calibrated and verified by measuring the roundness error of sample circular parts using the introduced system and a commercial system (Roundtest RA-100), then the results were compared. The maximum difference between the results of the two systems was $\pm 7.22\%$.

References

1. Xiulan Wen, Qingguan Xia, Yibing Zhao, "An effective genetic algorithm for circularity error unified evaluation", *International Journal of Machine Tools & Manufacture*, Vol. 46, pp 1770-1777, 2006.
2. Xiong F., Gao Y. and Li Z., "Statistical Roundness Tolerance and the Parameters for Assessment", *International committee for Measurement and Instruments, 5th International Symposium on Measurement Tech. and Intelligent Instruments*, pp. 419-424, 25-27 September, Cairo, Egypt, 2001.
3. Rao B.P.C., "Visual techniques in non-destructive testing, encyclopedia of materials: science and technology", Elsevier Science Ltd, pp 6043-6046, 2001.
4. Fletcher M.J., "Fully automatic inspection of welds now a reality", *NDT & E International*, Vol. 29, No. 3, pp 191, 1996.
5. Bayro-Corrochano E., "Review of automated visual inspection", *Proceedings of SPIE: Intelligent Robots and Computer Vision XII 2055*, pp 128-172, 1993.
6. Nobel J.A., "Form inspection to process understanding and monitoring: a view on computer vision in manufacturing". *Image and Vision Computing*, Vol. 13, pp 197-214, 1995.
7. Harding K., "Machine vision-based gauging of manufactured parts", *Vision, MVA/SME quarterly* 12, pp 1-6, 1996.
8. Guerra E. and Villalobos J.R., "A three dimensional automated visual inspection system for SMT assembly", *Computers and Industrial engineering*, Vol. 40, pp 175-190, 2001.
9. Harvey D.M., Kshirsagar S.P., and Hobson C.A., "Low cost scalable parallel image processing system". *Microprocessors and Microsystems*, Vol. 25, pp143-157, 2001.
10. Griffith, G.K., "Measuring and Gauging Geometric Tolerances", Prentice-Hall, Englewood Cliffs, NJ, 1994.
11. ISO/R 1001-1983, "Technical drawings-Geometrical Tolerancing Guidelines", ISO, Geneva.
12. Sy-Ming G. and Du-Ming T., "Measurement of Roundness: a Nonlinear Approach", *Proc. Natl. Sci. Counc. ROC(A)*, Vol. 23, No. 3, pp 348-352, 1999.
13. Gadelmawla E.S., "Computer Aided Measurement software for roundness evaluation from the coordinate measurement data", *9th Cairo university international conference on mechanical design & production (mdp-9)*, Cairo - Egypt, January 8-10, 2008, pp 270-285.
14. Jyunping H., "An exact solution for the roundness evaluation problems", *Precision Engineering*, Vol. 23, pp 2-8, 1999.
15. Jyunping H., "A new strategy for circularity problems", *Precision Engineering*, Vol. 25, pp 301-308, 2001.
16. Li Xiuming, Shi Zhaoyao, "Development and application of convex hull in the assessment of

- roundness error", *International Journal of Machine Tools and Manufacture*, Vol. 48, No. 1, January, pp 135-139, 2008.
17. Ye Ding, LiMin Zhu, Han Ding, "A unified approach for circularity and spatial straightness evaluation using semi-definite programming", *International Journal of Machine Tools & Manufacture*, Vol. 47; pp 1646-1650, 2007.
 18. Venkaiah N., Shunmugam M.S., "Evaluation of form data using computational geometric techniques—Part I: Circularity error", *International Journal of Machine Tools & Manufacture*, Vol. 47, pp 1229-1236, 2007.
 19. Muralikrishnan B., Venkatachalam S., Raja J, Malburg M, "A note on the three-point method for roundness measurement", *Precision Engineering*, Vol. 29, pp 257-260, 2005.
 20. Choi W., Kurfess T.R. and Cagan J., "Sampling uncertainty in coordinate measurement data analysis", *Precision Engineering*, Vol. 22, No. 3, pp153-163, 1998.
 21. Murthy T.S.R., "A comparison of different algorithms for circularity evaluation", *Precision Engineering*, Vol. 8, No. 1, pp 19-23, 1986.
 22. Yeralan S., Ventura J., "Computerized roundness inspection", *International Journal of Production Researches*, Vol. 26, No. 10, pp 1921-1935, 1988.
 23. David Manthey, "General Least-Squares - Direct Solutions and Bundle Adjustments", On 20/09/2008, <http://www.orbitals.com/self/least/least.htm>.
 24. Haralick, Robert M., and Linda G. Shapiro, "Computer and Robot Vision", Volume 1, Addison-Wesley, pp 28-48, 1992.