PAPER

A Practical Algorithm for Computing the Roundness

Hiroyuki EBARA†, Noriyuki FUKUYAMA††, Hideo NAKANO† *and* Yoshiro NAKANISHI†††, *Members*

SUMMARY Roundness is one of the most important geometric measures for circular objects in the process of mechanical assembly . It is the amount of variation in a circular size which can be permitted. To compute roundness, the authors have already proposed an exact polynomial-time algorithm whose time complexity is $O(n^2)$. In this paper, we show that this roundness algorithm can be improved more efficiently, by introducing the deletion of the unnecessary points, in practical applications. In addition, the computational experience of this revised algorithm is also presented.

***key words**: computational geometry, roundness, computational experience*

1. Introduction

In the process of mechanical assembly, the roundness[5] is one of the most important geometric measures for objects with circular geometry (for example, ball bearings, cylinders of engines, or rotary magnetic heads of video tape recorders (see Fig. 1)). It is the geometric tolerance for circular objects, that is, the amount of variation in a circular size which can be permitted, and is provided by ISO (International Organization for Standardization)[5].

We define the roundness problem to determine the roundness as follows: Given n points in the Euclidean plane, find the center of the concentric circles enclosing all given points between outer and inner circles and minimizing the difference between radii of the outer and inner circles. This definition is consistent with the definition of the roundness by ISO, except that objects are represented by a set of discrete points instead of a closed curve, and it is well adapted to practical roundness measurements.

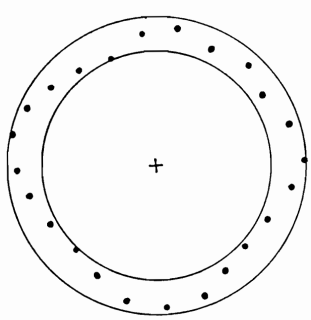


Fig. 1 The roundness.

For this problem, we have proposed an exact polynomial-time algorithm whose time complexity is $O(n^2)$[2]. This algorithm employs the technique of taking the union of the nearest-point Voronoi diagram[9],[10] and the farthest-point Voronoi diagram[9],[10]. Recently, Le and Lee proposed new algorithms for a different problem related to the roundness[6]. Moreover, other methods (for example, the least squares method, the Min-Max method, or the Simplex method) have been applied to compute the roundness[8],[12]. These are, however, approximate algorithms or exact but nonpolynomial time algorithms, with the exception of ours and that of Le and Lee.

In this paper, we develop a practical fast algorithm for the roundness problem, maintaining the exactness. The concept of this algorithm is the deletion of the unnecessary points. Our algorithm must be efficient, if the input data are randomly distributed almost on a circle as are practical data. Further, we confirm its remarkable efficiency through the computational experience of practical roundness data.

2. Exact Roundness Algorithm

We have shown the following theorem in Ref. (2). This theorem embodies the key concept of our previous roundness algorithm.

Theorem 1: The exact roundness can be computed in polynomial time by examining all vertices in the union of the nearest-point Voronoi diagram and the farthest-point Voronoi diagram.

Moreover, we present an additional result in the following proposition in this paper.

Proposition 1: There exists no center of the concentric circles determining the exact roundness on the Voronoi vertices in the nearest-point Voronoi diagram and the farthest-point Voronoi diagram, except for the case that the nearest-point (farthest-point) Voronoi vertices are on Voronoi edges or Voronoi vertices in

Manuscript received April 16, 1991.
Manuscript revised September 10, 1991.

† The authors are with the Faculty of Engineering, Osaka University, Suita-shi, 565 Japan.
†† The author is with FUJITSU LTD., Kawasaki-shi, 211 Japan.
††† The author is with Nara National College of Technology, Yamato-Koriyama-shi, 639-11 Japan.

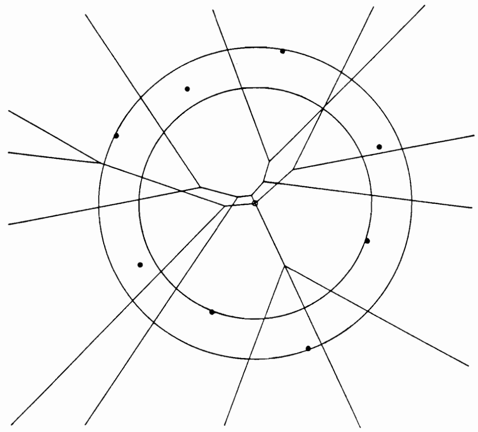


Fig. 2 The concentric circles determining the optimal roundness.

the farthest-point (nearest-point) Voronoi diagram.
[Sketch of proof] We assume that the center of the concentric circles determining the exact roundness is on the Voronoi vertex, which is not on a Voronoi edge or a Voronoi vertex in the other Voronoi diagram. Thus, there exists the center of the concentric circles determining the smaller roundness by moving the center a minute distance in a certain direction. This is a contradiction. □

Based on this theorem and proposition, we propose an exact algorithm—Algorithm 1—for the roundness problem. Algorithm 1 is the same as the algorithm found in Ref. (2) except that Algorithm 1 finds no Voronoi vertices. Figure 2 shows the concentric circles determining the exact roundness as constructed by Algorithm 1.

[Algorithm 1]

step 1 Construct the nearest-point Voronoi diagram.

step 2 Construct the farthest-point Voronoi diagram.

step 3 Find all intersecting points between the Voronoi edges in the nearest-point Voronoi diagram and the Voronoi edges in the farthest-point Voronoi diagram.

step 4 Compute the difference of the distances from these intersecting points to the nearest points and the farthest points.

step 5 Find the point with the minimum difference of the distances. The minimum difference of the distances is the roundness, and this point is the center of the concentric circles determining the roundness.

The time complexity of Algorithm 1 is $O(n^2)$, and there exists an instance where Algorithm 1 takes $O(n^2)$ time (see Fig. 3). This instance shows that the time complexity of Algorithm 1 is tight. However, if we were to implement this algorithm using the technique of Ref. (1) or that of Ref. (7), we could reduce the time complexity to $O(n \log n + k)$, where k is the number of intersecting points, and $O(n^2)$ in the worst case.

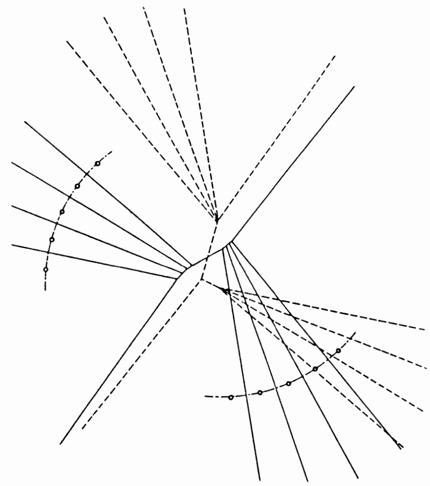


Fig. 3 Instance where there are $O(n^2)$ intersecting points.

3. Practical Roundness Algorithm

In practical applications of the roundness algorithm, a set of sampling points usually contains a number of points which do not contribute to the determination of the roundness. Therefore, if we can delete the unnecessary points in advance, maintaining the exactness, it is possible to reduce the computing time of the roundness algorithm.

We assume that the practical data are successively obtained in counterclockwise order, and the center of the concentric circles determining the exact roundness exists in the interior of the input data sequence. These assumptions are suitable for practical roundness measurements.

We will present two lemmas, two corollaries, and two algorithms for deleting the unnecessary points. The unnecessary points, those which do not contribute to the determination of the roundness, are defined in Corollary 1 and Corollary 2. First, we consider the deletion of the unnecessary points in constructing the inner circle from given points.

Lemma 1: Suppose the farthest pair for a set of n given points. Let D be the distance of this farthest pair, O_F the center point of the farthest pair, R_{in} the distance from O_F to the nearest point, R_{out} the distance from O_F to the farthest point, $r = D/2$, and $d = R_{\text{out}} - r$. There exist no concentric circles determining the exact roundness, such that the radius R' of the inner circle is less than $R_{\text{in}} - d$ (see Fig. 4).

Proof: Let R_{opt} be the exact roundness.

$$R_{\text{opt}} \leqq R_{\text{out}} - R_{\text{in}} = (R_{\text{out}} - d) - (R_{\text{in}} - d)$$
$$= r - (R_{\text{in}} - d).$$

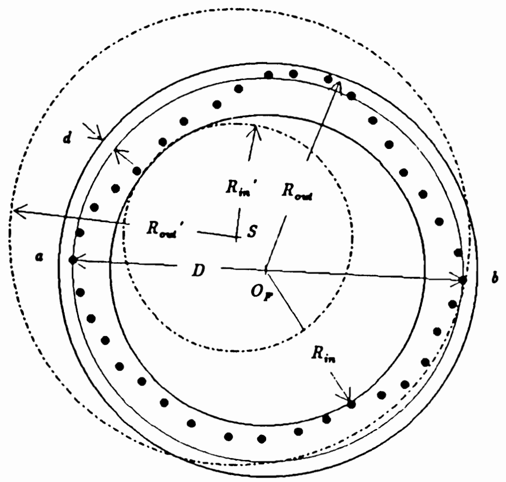


Fig. 4 Illustration for the proof of Fact 1.

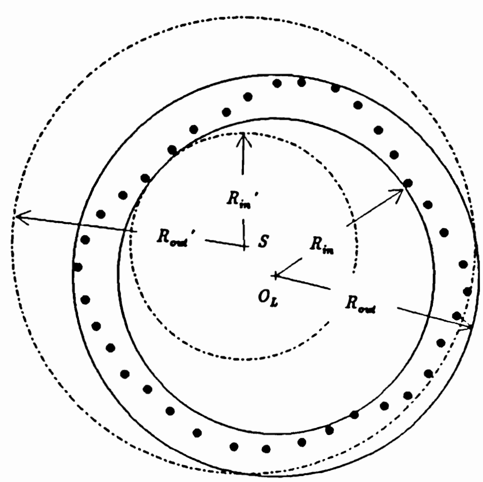


Fig. 5 Illustration for the proof of Fact 3.

Consider concentric circles with a center S such that the radius R_{in}' of the inner circle is less than $R_{in}-d$. Let R_{out}' be the radius of the outer circle. Since r is less than or equal to the radius of the smallest enclosing circle, $R_{out}'>r$. Therefore, since $R_{in}'<R_{in}-d$,

$$R_{opt} \leqq r-(R_{in}-d)<R_{out}'-R_{in}'. \qquad \square$$

Lemma 1 leads to the following corollary—Corollary 1—immediately.

Corollary 1: Suppose a circle formed by three points taken successively in counterclockwise order which form a left turn. If the radius of this circle is less than $R_{in}-d$, then the second point is an unnecessary point.

Using Corollary 1 and reforming the technique of Graham's convex hull algorithm[3], we construct an algorithm—Algorithm 2—deleting the unnecessary points for given points, in constructing the inner circle.

[**Algorithm 2**]

step 1 Construct the convex hull, and find the farthest pair using the caliper method[11].

step 2 Find the center point O_F of the farthest pair and compute the distance D of the farthest pair.

step 3 Compute R_{in} and R_{out}.

step 4 Let $R_L=R_{in}-(R_{out}-D/2)$.

step 5 Arrange the given points into a circular doubly linked list, with *RLINK* and *LLINK* associated with a node pointer, respectively, to the node on the counterclockwise side and the node on the clockwise side.

step 6 Let v_{START} be the nearest point from O_F.

step 7 Start at $v=v_{START}$, and repeat step 8 until $RLINK[v]=v_{START}$.

step 8 If three successive points from v form a left turn and the radius of the circle formed by these points is less than R_L, then delete the second point, and if $v \neq v_{START}$ let $v=LLINK[v]$ (the node on the clockwise side); otherwise, let $v=RLINK[v]$ (the node on the counterclockwise side).

Next, we consider the deletion of the unnecessary points in constructing the outer circle from given points.

Lemma 2: Suppose the largest empty circle for a set of n given points. Let R_{in} be the radius of this circle, and R_{out} the distance from the center O_L of this circle to the farthest point. There exist no concentric circles determining the exact roundness, such that the radius R' of the outer circle is more than R_{out} (see Fig. 5).

Proof: Consider an empty circle constituting the concentric circles with an enclosing circle of the radius R_{out}' ($>R_{out}$). Let R_{in}' be the radius of this circle. Then $R_{in} \geqq R_{in}'$, because R_{in} is the radius of the largest empty circle. Since $R_{out}<R_{out}'$,

$$R_{out}'-R_{in}'>R_{out}-R_{in}. \qquad \square$$

Lemma 2 leads to the following corollary—Corollary 2—immediately.

Corollary 2: Suppose a circle formed by three points taken successively on the convex hull in counterclockwise order. If the radius of this circle is more than R_{out}, then the second point is an unnecessary point.

Using Corollary 2 and reforming the technique of Graham's convex hull algorithm, we construct a similar algorithm—Algorithm 3—deleting the unnecessary points for given points, in constructing the outer circle.

[**Algorithm 3**]

step 1 Arrange the given points on the convex hull into a circular doubly linked list, with *RLINK* and *LLINK* associated with a node pointer, respectively, to the node on the counter-clockwise side and the node on the clockwise side.

step 2 Let v_{START} be the farthest point from O_L, and let R_{out} be this distance.

step 3 Start at $v=v_{START}$, and repeat step 4 until

$RLINK[v] = v_{START}$.

step 4 If the radius of the circle formed by three successive points is more than R_{out}, then delete the second point, and if $v \neq v_{START}$ let $v = LLINK[v]$ (the node on the clockwise side); otherwise, let $v = RLINK[v]$ (the node on the counterclockwise side).

The worst-case time complexity of Graham's convex hull algorithm is $O(n \log n)$. However, the time complexity of both algorithms—Algorithm 2 and Algorithm 3—is linear as the input data have been sorted by polar angle.

Now, we can improve the exact roundness algorithm by introducing the deletion of the unnecessary points. We propose such an algorithm—Algorithm 4—as a practical version of Algorithm 1.

[Algorithm 4]

step 1 Delete the unnecessary points among the given points by Algorithm 2 in constructing the nearest-point Voronoi diagram.

step 2 Construct the nearest-point Voronoi diagram for the rest of the given points.

step 3 Find the center O_L of the largest empty circle from the nearest-point Voronoi diagram.

step 4 Delete the unnecessary points among the given points by Algorithm 3 in constructing the farthest-point Voronoi diagram.

step 5 Construct the farthest-point Voronoi diagram for the rest of the given points.

step 6 Compute the roundness by taking the union of these Voronoi diagrams.

We are convinced that this algorithm can delete many unnecessary points and run much faster than Algorithm 1, taking into the consideration the fact that the input data are distributed almost on a circle in practical roundness measurements.

4. Computational Experience

In this section, we provide computational results for practical roundness measurements.

The input data in this experience are all real data sampled from cylinders with 3 cm radii by roundness measuring instruments, and the number of sampling points is 1800. The input data are distributed almost on a circle, and its roundness is about 10 μm. We use a Unix workstation with coding in C language.

In Algorithm 1 and Algorithm 4, we adopt the incremental algorithm of Ref. (4) to construct the Voronoi diagrams and an exhaustive search to find intersecting points.

Table 1 gives the number of points on the convex hull and the numbers of points after deleting the unnecessary points. Table 2 presents a comparison of the computational time between Algorithm 1 and Algorithm 4.

The computational results show that the revised roundness algorithm reduces the necessary points in constructing the nearest-point Voronoi diagram in about 1/20 the time of the original one and in constructing the farthest-point Voronoi diagram in about 1/200 the time. Furthermore, the overall computational process for computing the roundness proceeds more than 20 times as fast. Since the boundaries for the deletion in constructing the nearest-point Voronoi diagram are worse, being computed from the farthest pair, the deleting points in constructing the nearest-point Voronoi diagram are fewer.

Table 1 The number of points on the convex hull and the numbers of points after deleting the unnecessary points.

data No.	# of points on the convex hull	# of points after deletion	
		Algorithm 2	**Algorithm 3**
1	1273	107	5
2	1297	114	12
3	1273	102	7
4	1254	111	13
5	1233	22	4
6	1201	112	7
7	1170	33	3
8	1630	9	4
average	1291	76.3	6.9

Table 2 A comparison of the computational time between Algorithm 1 and Algorithm 4.

data No.	computational time (sec.)	
	Algorithm 1	**Algorithm 4**
1	116.2	6.26
2	142.3	7.00
3	156.5	6.41
4	138.4	6.71
5	159.1	4.86
6	177.9	9.39
7	141.3	7.06
8	165.5	7.89
average	149.7	6.95

5. Conclusions

We proposed a practical, fast roundness algorithm by introducing the deletion of the unnecessary points, maintaining the exactness. In addition, we made sure of its remarkable efficiency by employing practical roundness data. It is also clear that our technique of deleting the unnecessary points can be applied to the roundness algorithms which have typically been used in practical roundness measurements.

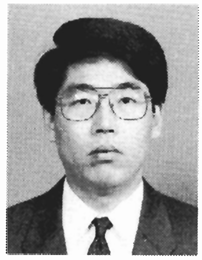
We can estimate the number of the deleted points by our algorithm experimentally; however, we cannot estimate them theoretically. The theoretical estimation of the number of deleted points is a problem requiring further study.

Acknowledgements

We would like to thank Dr. D. T. Lee and Dr. Te. Asano for useful discussions, and the unknown referee for several insightful comments. We also wish to thank Mr. T. Sanada for offering us many practical roundness data.

References

(1) Chazelle B. and Edelsbrunner H.:"An Optimal Algorithm for Intersecting Line Segments in the Plane", Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science, pp. 590-600 (1988).
(2) Ebara H., Nakano H., Nakanishi Y. and Sanada T.:"A Roundness Algorithm Using the Voronoi Diagrams", Trans. IEICE, **J70-A**, 4, pp. 620-624 (1987).
(3) Graham R. L.:"An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set", Information Processing Letters, **1**, pp. 132-133 (1972).
(4) Green P. J. and Sibson R.:"Computing Dirichlet Tessellations in the Plane", The Computer Journal, **21**, 2, pp. 168-173 (1978).
(5) ISO R1101/I, 1969.
(6) Le V. B. and Lee D. T.:"Out-Of-Roundness Problem Revisited", IEEE Trans. Pattern Analysis and Machine Intelligence, **13**, 3, pp. 217-223 (1991).
(7) Mairson H. G. and Stolfi J.:"Reporting and Counting Intersections Between Two Sets of Line Segments", Theoretical Foundations of Computer Graphics and CAD, pp. 307-325, Springer-Verlag (1987).
(8) Murthy T. S. R. and Abdin S. Z.:"Minimum Zone Evaluation of Surfaces", International Journal of Machine Tool Design & Research, **20**, pp. 123-137 (1980).
(9) Preparata F. P. and Shamos M. I.:"Computational Geometry—An Introduction", Springer-Verlag (1985).
(10) Shamos M. I. and Hoey D.:"Closest-Point Problems", Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science, pp. 151-162 (1975).
(11) Toussaint G. T.:"Solving Geometric Problems with the Rotating Calipers", Proceedings of IEEE MELECON '83 (1983).
(12) Tsukada T. and Kanada T.:"Measurement of Cylindrical Form Errors Using a Non-Contact Detector", Precision Engineering, **4**, pp. 153-158 (1982).



Hiroyuki Ebara was born in Osaka, Japan, on April 28, 1958. He received the B.E., M.E., and Dr.Eng. degrees in communication engineering from Osaka University, Osaka, Japan, in 1982, 1984, and 1987, respectively. Since 1987 he has been with Osaka University, where he is Assistant Professor. His main research interests are in computational geometry, combinatorial optimization, and parallel computing. Dr. Ebara is a member of IEEE, ACM, and IPSJ.

Noriyuki Fukuyama was born in Osaka, Japan, on May 4, 1963. He received the B.E. and M.E. degrees in communication engineering from Osaka University, Osaka, Japan, in 1986 and 1988, respectively. He has been with the Fujitsu Ltd. since 1988. His research interests are in the area of telecommunication.

Hideo Nakano was born in Osaka, Japan, on January 1, 1948. He received the B.E., M. E., and Dr. Eng. degrees in communication engineering from Osaka University, Osaka, Japan, in 1970, 1972, and 1975, respectively. Since 1975 he has been with Osaka University, where he is now Associate Professor. His main research interests are in combinatorial optimization, computational geometry, network security, and software engineering. Dr. Nakano is a member of IEEE, ACM, and IPSJ.

Yoshiro Nakanishi was born in Wakayama, Japan, on January 10, 1926. He received the B.E. and Ph.D. degrees in communication engineering from Osaka University, Osaka, Japan, in 1952 and 1962, respectively. After being a professor at Osaka University, in 1989 he became the president of Nara National College of Technology, Nara, Japan, where he has been to the present. He engages in study and research on computer science and system engineering. Dr. Nakanishi is a member of IEEE.