

ポロノイ図を応用した真円度を求める解法

著者	榎原 博之, 中野 秀男, 中西 義郎, 真田 友宏
雑誌名	電子情報通信学会論文誌A
巻	J70-A
号	4
ページ	620-624
発行年	1987-04
権利	copyright (C)1987 IEICE
その他のタイトル	A Roundness Algorithm Using the Voronoi Diagrams
URL	http://hdl.handle.net/10112/1877

ボロノイ図を応用した真円度を求める解法

正員 榎原 博之[†] 正員 中野 秀男[†]
 正員 中西 義郎[†] 非会員 真田 友宏^{††}

A Roundness Algorithm Using the Voronoi Diagrams

Hiroyuki EBARA[†], Hideo NAKANO[†], Yoshiro NAKANISHI[†], *Members*
 and Tomohiro SANADA^{††}, *Nonmember*

あらまし 真円度は機械部品の精度として重要な幾何公差（形状公差）の一つであり、これまでのところ、これを求める時間計算量が多項式時間である厳密解法は知られていない。本論文は、計算幾何学の中心課題の一つであるボロノイ図を利用することによって、真円度を多項式時間で求める解法が考えられることを指摘したものである。筆者らは、真円度を求める解法として、最近点のボロノイ図と最遠点のボロノイ図を構成し、それらの結び（Union）をとるといふ考えに基づいた解法を提案し、その時間計算量が $O(n^2)$ であることを明らかにしている。

1. まえがき

真円度⁽¹⁾は、機械部品の精度として重要な幾何公差（形状公差）の一つであり、「円筒断面の輪郭を二つの同心円ではさんだとき、両円の間隔が最小となる両円の半径差」と定義される（図1）。

真円度を求める手法としては、これまでのところ、最小2乗法による解法、Min-Max法を適用した解法、シンプレックス法を用いた解法等が考えられているが、近似解法に過ぎないとか、厳密解法であっても時間計算量に難点があるものであって、時間計算量が多項式時間である厳密解法は知られていない。

本論文は、ボロノイ図⁽²⁾を利用することによって、真円度を多項式時間で求める解法が考えられることを指摘したものである。

まず、2.で、最近点および最遠点のボロノイ図およびそれらの構成算法について述べ、ついで、3.で、これらの算法を利用した真円度を求める解法を提案し、その最適性と時間計算量を明らかにしている。

2. 最近点および最遠点のボロノイ図とそれらの構成算法

2.1 最近点および最遠点のボロノイ図

ボロノイ図（勢力圏図）⁽²⁾とは、次のように定義されるものである。

「平面上の n 個の点 $p_i = (x_i, y_i) (i=1, 2, \dots, n)$ に対して、点 p_i の‘勢力圏’ $V(p_i)$ を

$$V(p_i) = \bigcap_{j=1}^n \{p = (x, y) \mid d(p_i, p) \leq d(p_j, p)\}$$

と定め、 $V(p_i) (i=1, 2, \dots, n)$ による平面分割をボロノイ図と呼ぶ（図2）」。ここで、 $d(p_i, p)$ は2点 p_i, p のユ

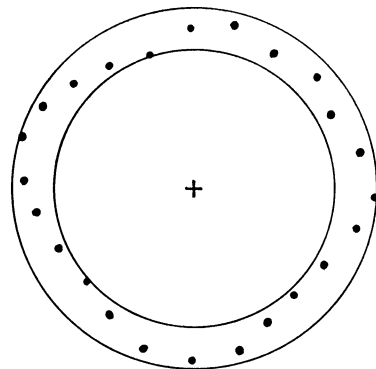


図1 真円度
 Fig. 1 Roundness.

[†] 大阪大学工学部通信工学科，吹田市
 Faculty of Engineering, Osaka University, Suita-shi, 565 Japan

^{††} 株式会社東京精密，土浦市
 Tokyo Seimitsu Co., Ltd., Tsuchiura-shi, 300 Japan

ユークリッド距離，すなわち， $d^2(p_i, p) = (x - x_i)^2 + (y - y_i)^2$ である。

ポロノイ図において，‘勢力圏’ $V(p_i)$ を点 p_i のポロノイ領域と呼び，これは凸多角形である。また，ポロノイ領域の頂点を最近点のポロノイ点，辺を最近点のポロノイ辺と呼ぶ。

以上は，最近点（勢力圏）を表すポロノイ図であるが，これに対して，最遠点（非勢力圏）を表すポロノイ図⁽²⁾も考えられ，次のように定義される。

「平面上の n 個の点 $p_i = (x_i, y_i) (i = 1, 2, \dots, n)$ に対して，点 p_i の‘非勢力圏’ $U(p_i)$ を

$$U(p_i) = \bigcap_{j=1}^n \{p = (x, y) | d(p_i, p) \geq d(p_j, p)\}$$

と定め， $U(p_i) (i = 1, 2, \dots, n)$ による平面分割を最遠点のポロノイ図と呼ぶ(図3)」。最遠点のポロノイ点およびポロノイ辺も同様に定義される。

最遠点のポロノイ図の場合はすべての点が領域を持つわけではなく，与えられた点集合の凸包上の点のみ領域を持つ。

2.2 最近点および最遠点のポロノイ図の構成算法

ポロノイ図の構成算法は，分割統治法を用いる手法が考えられている。最近点のポロノイ図の構成算法をアルゴリズム1に示す⁽²⁾。

[アルゴリズム1]

step 1 n 個の点を x 座標の値に関してソートし，その順に p_1, \dots, p_n とする。

step 2 これらの点を左側の点の集合 $L = \{p_1, \dots, p_{\lfloor n/2 \rfloor}\}$ と右側の点の集合 $R = \{p_{\lfloor n/2 \rfloor + 1}, \dots, p_n\}$ に二等分し，それぞれに対するポロノイ図

$V(L), V(R)$ を再帰的に構成する。

step 3 点集合 L, R から等距離な点の集合の軌跡(折れ線)を求め，その線より左にある $V(R)$ ，その線より右にある $V(L)$ ，それぞれのポロノイ点，ポロノイ辺を除くことによって，二つのポロノイ図 $V(L), V(R)$ を併合する(図4)。

このアルゴリズムのデータ構造は，与えられた n 個

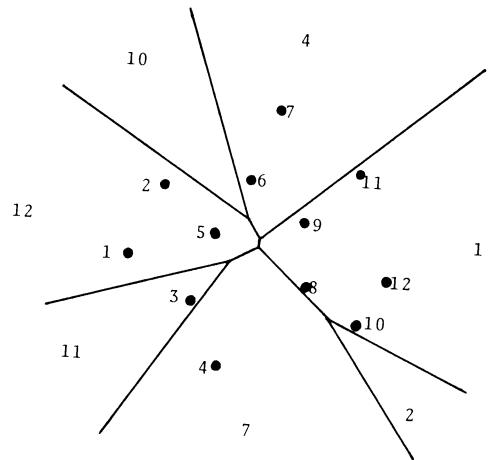


図3 最遠点のポロノイ図
Fig. 3 Farthest-point Voronoi diagram.

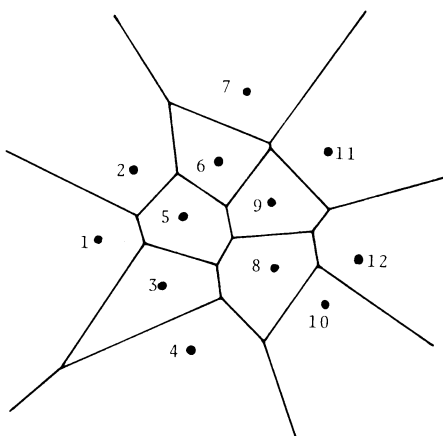
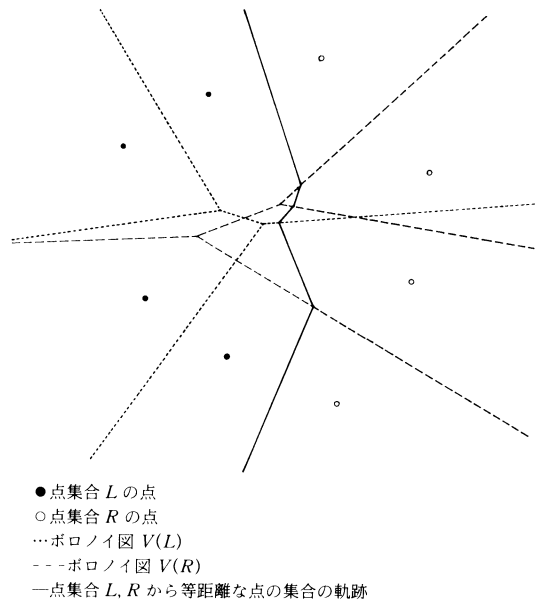


図2 最近点のポロノイ図
Fig. 2 Nearest-point Voronoi diagram.



●点集合 L の点
○点集合 R の点
…ポロノイ図 $V(L)$
---ポロノイ図 $V(R)$
—点集合 L, R から等距離な点の集合の軌跡
図4 最近点のポロノイ図の併合
Fig. 4 Merge of nearest-point Voronoi diagrams.

の点とボロノイ辺に関するものであり、各点はその点に対応する領域を境界しているボロノイ辺を指し、ボロノイ辺はその両側の領域の点とそのボロノイ辺に隣接しているボロノイ辺を指す。

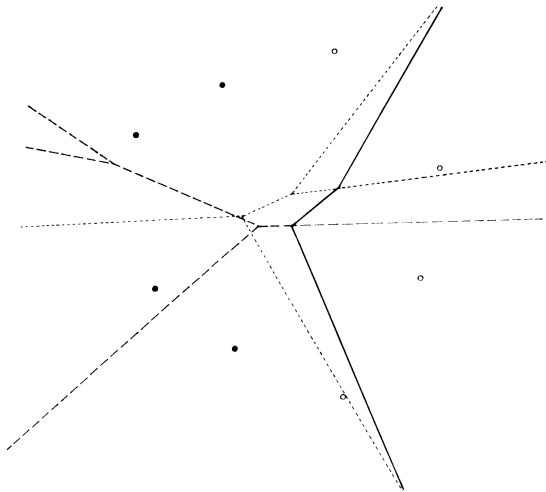
点集合 L, R から等距離な点の集合の軌跡を求める操作は、点集合 L, R の凸包の下側の共通接線上の2点の垂直二等分線から始め、この直線が左右どちらかのボロノイ辺と交われば、その交点から次の垂直二等分線を引き、この操作をボロノイ辺と交わらなくなるまで行う。この操作は、ボロノイ辺のリストをたどることによって、点の数に比例する手間で行うことができる。

このようにして、step 3の併合は併合する図の点の数に比例する手間で行えるので、アルゴリズム1の時間計算量は $O(n \log n)$ である⁽²⁾。

次に、最遠点のボロノイ図の構成にあたっては、アルゴリズム1で等距離な点の集合の軌跡より遠い側を取り除いたのに対して、近い側を取り除けば済む。従って、最遠点のボロノイ図の構成算法はこの点を除いてアルゴリズム1と同じである。この算法をアルゴリズム2に示す⁽²⁾。

[アルゴリズム2]

step 1 n 個の点を x 座標の値に関してソートし、その順に p_1, \dots, p_n とする。



●点集合 L の点
○点集合 R の点
---ボロノイ図 $U(L)$
---ボロノイ図 $U(R)$
—点集合 L, R から等距離な点の集合の軌跡

図5 最遠点のボロノイ図の併合

Fig. 5 Merge of farthest-point Voronoi diagrams.

step 2 これらの点を左側の点の集合 $L = \{p_1, \dots, p_{[n/2]}\}$ と右側の点の集合 $R = \{p_{[n/2]+1}, \dots, p_n\}$ に二等分し、それぞれに対するボロノイ図 $U(L), U(R)$ を再帰的に構成する。

step 3 点集合 L, R から等距離な点の集合の軌跡(折れ線)を求め、その線より左にある $U(L)$ 、その線より右にある $U(R)$ 、それぞれのボロノイ点、ボロノイ辺を除くことによって、二つのボロノイ図 $U(L), U(R)$ を併合する(図5)。

アルゴリズム2の時間計算量も同様に $O(n \log n)$ である⁽²⁾。

3. 真円度を求める解法

円筒断面の輪郭が n 個の点で与えられているとする。このとき、真円度を決める同心円の中心はその中心点からの最も近い点(1点とは限らない)と最も遠い点(1点とは限らない)の距離の差が最小となる点である。

ところで、 n 個の点についてのボロノイ図を考えると、最近点のボロノイ領域内の任意の点はその領域に対応する点が最近点であり、最遠点のボロノイ領域内の任意の点はその領域に対応する点が最遠点になっている。ここで、最近点のボロノイ図と最遠点のボロノイ図の結び(Union)をとると、各ボロノイ辺はただか $O(n)$ 個の領域しか通らないので、平面はただか $O(n^2)$ 個の領域に分割され、各領域内の任意の点は最近点と最遠点が唯一に決められている。

なお、最近点と最遠点のボロノイ領域はいずれも凸領域であり、二つの凸領域の交わり(Intersection)は凸領域であることから、二つのボロノイ図の結びとして作られる各領域は凸領域であることに注意しておく。

以上のことから、真円度を決める同心円の中心は、与えられた n 個の点に対する二つのボロノイ図の結びの領域について、その領域内の点で最近点および最遠点への距離の差を最小とする点を見出せばよいことになる。すなわち、結びの領域 r 内の点を $p(x_p, y_p)$ 、その領域に対する最近点および最遠点をそれぞれ $v(x_v, y_v), u(x_u, y_u)$ とすれば、問題は領域 r 内で距離の差の関数

$$f = \sqrt{(x_u - x_p)^2 + (y_u - y_p)^2} - \sqrt{(x_v - x_p)^2 + (y_v - y_p)^2}$$

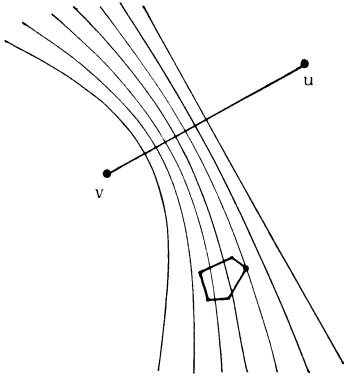
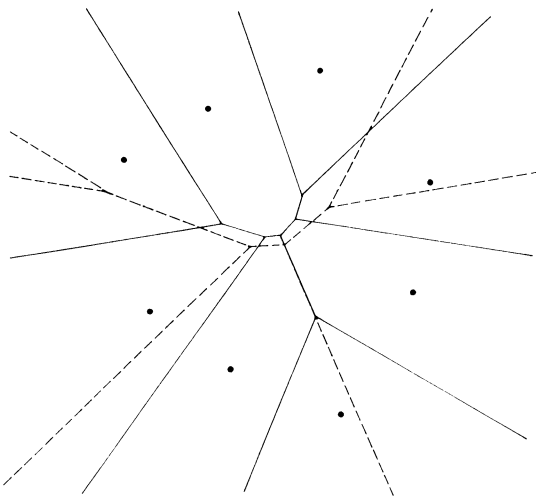


図6 距離の差の関数と凸領域

Fig. 6 Function of the difference in distance and convex region.



—最近点のボロノイ図
 ---最遠点のボロノイ図

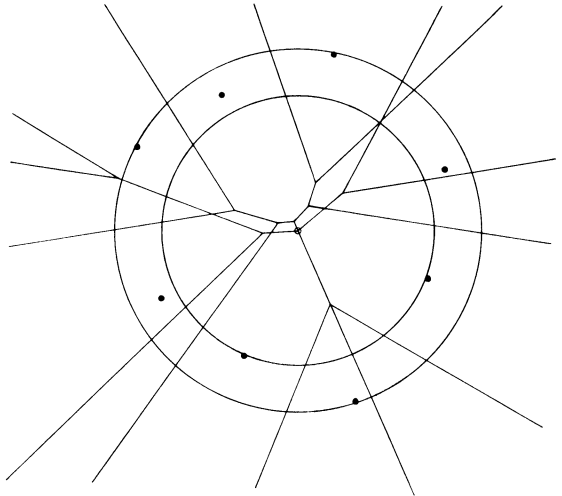
図7 最近点と最遠点のボロノイ図の結び

Fig. 7 Union of nearest-point and farthest-point Voronoi diagrams.

を最小とすると定式化できる。

関数 f は双曲線となるので、結びの領域側（最近点側）からみると凹関数であり、結びの領域は先に指摘したように凸領域であるので、最適解は必ず結びの領域の頂点にある（図6）。従って、問題の最適解の探索は、二つのボロノイ図の結びについて、最近点のボロノイ点、最遠点のボロノイ点、および、最近点のボロノイ辺と最遠点のボロノイ辺との交点のみを調べることと済む。

上述の論拠に基づいて真円度を求める解法としてアルゴリズム3を提案する。



○真円度の同心円の中心

図8 真円度の同心円

Fig. 8 Concentric circles of roundness.

[アルゴリズム3]

- step 1 アルゴリズム1を行い、最近点のボロノイ図を構成する。
- step 2 アルゴリズム2を行い、最遠点のボロノイ図を構成する。
- step 3 step 1, 2で求めた二つのボロノイ図の結び (Union) をとる (図7)。
- step 4 二つのボロノイ図の結びの図の各点 (最近点のボロノイ点, 最遠点のボロノイ点, 最近点のボロノイ辺と最遠点のボロノイ辺との交点) に対して、その点からの最近点と最遠点の距離の差を求める。
- step 5 最近点と最遠点の距離の差が最小となる点とその値を求める。最小となる距離の差が真円度であり、最小となる点が真円度の同心円の中心である (図8)。

このアルゴリズムの step 3 の二つのボロノイ図の結びをとる操作は、step 4 で結びの図の各点 (最近点のボロノイ点, 最遠点のボロノイ点, 最近点のボロノイ辺と最遠点のボロノイ辺との交点) に対してのみ操作を行うので、最近点のボロノイ点と最遠点のボロノイ点を列挙し、各ボロノイ辺の交点を調べてその交点を列挙すればよい。

[アルゴリズム3の時間計算量]

アルゴリズム3における step 1, step 2 の最近点と最遠点のボロノイ図の構成は、前章で示したように $O(n \log n)$ の時間計算量でできる。二つのボロノイ図

は平面グラフであり、頂点の数、辺の数、領域の数はすべて $O(n)$ である。従って、step 3 での二つのポロノイ図の結びは $O(n^2)$ の手間で作ることができ、できあがった結びの図の頂点、辺、領域の数はたかだか $O(n^2)$ である。step 4 では、たかだか $O(n^2)$ 個の各点に対してその点からの最近点と最遠点の距離の差を求めるが、これは結びをとるときに各点の最近点と最遠点の一つずつ記憶しておくことによって各点に対して $O(1)$ の手間のできる。step 5 は、step 4 で計算された各点についての最近点と最遠点の距離の差のうちで、最小となる点を見出す手続きであり、これに要する手間はたかだか $O(n^2)$ である。

以上のことから、アルゴリズム 3 全体の時間計算量は $O(n^2)$ ということになる。

4. むすび

本論文では、機械部品の精度にとって重要な因子の一つである真円度を求める解法として、最近点のポロノイ図と最遠点のポロノイ図の結び(Union)をとるという考えに基づいた解法を提案し、その時間計算量が $O(n^2)$ であることを明らかにした。

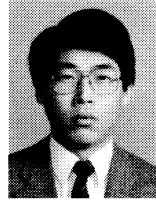
本解法は、計算幾何学の中心課題の一つであるポロノイ図の一つの応用を指摘するだけでなく、真円度の決定問題に対して実用的にも有力な解法になるものと評価している。

また、本問題を 3 次元に拡張すると、真円度は真球度となり、真球度も 3 次元のポロノイ図を構成することによって同様に多項式時間で求めることができると考えている。

文 献

- (1) 塚田, 金田, 奥田: “最適化技法を用いた最小領域法真円度の評価法”, 精密機械, **49**, 10, pp. 1351-1357 (昭 58).
- (2) M. I. Shamos and D. Honey: “Closest-point problems”, Proc. 16th IEEE Symp. Foundations of Computer Science, pp. 151-162 (1976).
- (3) 浅野孝夫: “計算幾何学とその応用”, 情報処理, **25**, 3, pp. 208-221 (昭 59-03)

(昭和 61 年 8 月 21 日受付, 11 月 13 日再受付)



榎原 博之

昭 57 阪大・工・通信卒。昭 59 同大学院修士課程了。現在、同大学院博士課程在学中。計算幾何学、組合せ最適化問題等の研究に従事。情報処理学会会員。



中野 秀男

昭 45 阪大・工・通信卒。昭 50 同大学院博士(通信)課程了。同年阪大通信工学科助手となり現在に至る。離散最適化問題の近似解法の評価、ソフトウェア工学等の研究に従事。工博。情報処理学会、IEEE、ACM 各会員。



中西 義郎

昭 27 阪大・工・通信卒。昭 37 阪大助教授。工博。昭 47 年同教授となり現在に至る。システム工学、ソフトウェア工学に関心をもち、組合せ最適化問題、システム故障診断、プログラムテストなどの研究に従事。IEEE 会員。



真田 友宏

昭 45 東海大・工・電子卒。同年(株)東京精密入社。真円度測定機、表面粗さ計の開発に従事。現在、同社土浦工場技術部技術 3 課課長。