

An efficient, accurate approach to medial axis transforms of pockets with closed free-form boundaries

Zezhong C. Chen · Qiang Fu

Received: 14 February 2012 / Accepted: 17 October 2012 / Published online: 2 November 2012
© Springer-Verlag London 2012

Abstract Medial axis transform of a pocket with free-form closed boundaries is a completed, compact representation of the pocket geometric shape and topology. It is very useful to multiple cutters selection and their tool paths generation for CNC machining of complex pockets. In the past decades, much research has been successfully conducted on the topic of finding the medial axis of a shape domain bounded with a polygon or simple geometries, e.g., lines and circles. Currently, more pockets with free-form boundaries are adopted in mechanical parts; however, the prior medial axis generation methods cannot handle this type of pockets well, resulting in long computation time and low medial axis accuracy. To address this problem, an efficient, accurate approach to calculating the medial axis transforms of these pockets is proposed in this work. An original optimization model of bisectors is established, and a new optimization method—the hybrid global optimization method—is developed to efficiently and accurately solve the optimization model of bisectors. The new optimization model and solver have been applied to many examples, and the testing results have demonstrated the advantages of this innovative approach over the prior medial axis methods. It can be an effective solution to the medial axis transforms of complex pockets.

Keywords Medial axis transform · Free-form pocket medial axis · Global optimization free-form pocket machining

1 Introduction

As a new type of geometric feature, pockets with closed free-form boundary curves are often adopted in mechanical parts, and it is always in high demand to efficiently machine these pockets in industry. However, due to their complex shape, it is quite challenging to select appropriate cutters and plan tool paths for them to achieve high material removal rates [1]. Medial axis (MA) of the pockets can be an effective tool to solve this technical challenge. Basically, the medial axis of a planar domain is the locus of centers of all the largest circles that are bounded by the domain boundaries at the contact points and are scattered in the domain along its boundaries. In this paper, the medial axis transform (MAT) refers to these circles centers, radii, and contact points. Due to its wide applications, MA has been a topic under extensive research in the past decade. For pockets with polygon boundaries, many works adopted Voronoi diagrams of the pockets to easily and accurately find their MAs, and the results were acceptable. But, for free-form boundaries, they often encountered the problem of low accuracy of medial axis points. Then the analytical methods were proposed to find the bisector equations between two curves. However, these methods are very difficult to implement and computing time is long. The objective of this research is to develop a new approach to generating the medial axis transforms of free-form pockets, which include non-uniform rational B-spline (NURBS) boundary curves.

Among the accurate mathematical methods of finding MA [2–7], the differential and topological properties of the MAT were studied and the exact MATs of pocket boundaries with piecewise lines and circular arcs were constructed in an analytical way [2, 3]. A trimming procedure was proposed to identify parametric sub-segments of variable

Z. C. Chen (✉) · Q. Fu
Department of Mechanical and Industrial Engineering,
Concordia University, Montreal, QC H3G 1M8, Canada
e-mail: zcchen@encs.concordia.ca

distance offsets that constitute true bisectors of point/curve and curve/curve [4, 5]. Elber and Kim [6] represented the correspondence between the foot points on two planar rational curves as a bi-variate B-spline function. Moreover, Culver et al. [7] developed a tracing algorithm to compute the exact medial axes of polyhedrons. The approaches are accurate; however, their solutions could be complicated and the computing time normally is quite long.

Many researchers have conducted research on generating Voronoi diagrams to approximate MAs. In computational geometry, the technique of computing the Voronoi diagram of a group of point sites is matured [8, 9], and several efficient algorithms of complexity $O(n \cdot \log n)$ are available together with their free source codes [10]. Brandt [11] found that the Voronoi diagram of a shape can properly converge if its boundary is continuous. Dey and Zhao [12] approximated a medial axis with a Voronoi diagram in a scale- and density-independent manner, and convergence of this approximation was guaranteed. Chou [13] developed a tracing algorithm to calculate Voronoi diagrams for closed planar shapes with boundaries of free-form curves. Ramamurthy and Farouki [14, 15] presented an approach for computing Voronoi diagrams by adding a single boundary segment in each step, based on point/curve and curve/curve bisectors. They clearly stated in [14] that “it might seem that an easy ‘practical’ approach to domains with free-form boundary curves is to first approximate the boundaries, within a prescribed tolerance, by polygonal or piecewise-linear/circular curves, and then invoke the currently available algorithms for such boundary curves. However, this approach yields qualitatively incorrect results.” Alt et al. [16] presented a randomized incremental algorithm to compute the Voronoi diagrams of curved objects by breaking up the curves into “harmless” pieces, which include line segments, circular arcs and spiral arcs. Unfortunately, the methods of approximating bisector curves with Voronoi diagrams are challenged for shape domains with free-form boundaries. Seong et al. [17] calculated the critical structures of the Voronoi diagrams by solving a system of nonlinear piecewise rational equations in the parametric space. However, this method is very complicated.

Besides the main stream MA methods mentioned above, Gursoy and Patrikalakis [18, 19] introduced a wave front propagation method by determining inward offset distances and the associated branch points at which the contour topology changes. Their method analytically defines the MA in terms of conic sections, and free-form boundary curves are approximated within a prescribed tolerance using line and circular arc segments. Likewise, Hassouna and Farag [20] rendered a wave front propagation method. The front propagates at each object point in a speed that is proportional to its Euclidean distance from the boundary. The motion of the front is governed by a nonlinear partial differential equation

and is solved using level set methods. Then, the medial axis is computed by tracking the propagating front points of the maximal positive curvature. Aichholer et al. [21] approximated the boundary spline with spiral biarcs and found the medial axis based on the biarcs. Kosinka and Juttler [22] computed the medial surfaces of spatial boundary surfaces in the Minkowski space. However, a major problem of these methods is that the medial axis of the fitting biarcs could largely deviate from the ideal medial axis of the spline boundary even though they fit the boundary very well. The main reason is that the fitting arcs can be quite different from the boundary with regard to the tangent at two points on them, the corresponding medial axis points could be far from each other. This is an inherited problem of these methods and has not received much attention.

To address the technical challenge of the prior MA methods, an efficient, accurate approach to the medial axis transforms of pockets with closed free-form boundary curves is proposed. As the basics, the properties of the free-form boundary curves of pockets and terminologies and definitions of MAT are provided in Sect. 2. The optimization model of bisector point is formulated in Sect. 3, and a new global optimization method—a hybrid global optimization method—is developed in Sect. 4 to solve the optimization problem for bisector points of the MA. In Sect. 5, several examples are provided to compare the results of MAs found with a Voronoi diagram generation method and this new approach, in order to demonstrate advantages of this approach. This work is concluded in Sect. 6.

2 Medial axis transforms of pockets with closed free-form boundary curves

Since the main research objective is to generate medial axis transforms of pockets of mechanical parts for their CNC machining, the geometric properties and definitions of the pocket boundaries and their medial axis transforms are first rendered in this section.

2.1 Properties of the free-form boundaries of pockets of mechanical parts

To design pockets of mechanical parts with CAD software, composite curves including line, arc, and NURBS curve segments are often used to define the external boundaries of the recess domains and boundaries of the internal islands, if any [1]. The external boundary is closed with its component curves head-to-toe connected one-by-one. All component curves are regular curves, i.e., their first derivatives exist, and are represented in a parametric form. In this work, a pocket boundary generally includes external and internal boundaries, and it is defined in the following.

Definition 2.1 A pocket boundary is defined as a composite parametric curve $\mathbf{B}(u) : \mathbb{R} \rightarrow \mathbb{R}^2$ consisting of n component curves, $\mathbf{B}^i(u) := [Bx^i(u), By^i(u)]^T, i = 1, 2, \dots, n$, where their parameter $u \in [u_{\min}^i, u_{\max}^i]$. The boundary generally can be represented as $\mathbf{B}(u) := \bigcup_{i=1}^n \mathbf{B}^i(u)$. The component curves can be lines, circular arcs and NURBS curves with the following properties:

- $\mathbf{B}^i(u_{\max}^i) = \mathbf{B}^{i+1}(u_{\min}^{i+1}), i = 1, 2, \dots, (n - 1);$ and $\mathbf{B}^1(u_{\min}^1) = \mathbf{B}^n(u_{\max}^n);$
- The unit tangent vector of a component curve as $\mathbf{T}^i(u) := [Tx^i(u), Ty^i(u)]^T;$ and
- The unit normal vector of a component curve as $\mathbf{N}^i(u) := [Nx^i(u), Ny^i(u)]^T,$ and the vector $\mathbf{N}^i(u)$ pointing inside of the pocket.

For the external boundary of a pocket, its component curves and their directions are arranged counter-clockwise. For an internal boundary of the pocket, its component curves and their directions are arranged clockwise. Each of these boundaries is stored as an ordered list of the component curves.

Generally, at a point on a component curve, there exist infinite circles tangent to this curve and on the pocket side, which can be defined in the following way. First, for any point \mathbf{P} with parameter u_p on a component line $\mathbf{B}^i(u), u_{\min}^i \leq u \leq u_{\max}^i,$ its position vector \mathbf{B}_p^i can be computed as $\mathbf{B}_p^i = \mathbf{B}^i(u_p),$ and the corresponding unit normal of $\mathbf{B}^i(u)$ is $\mathbf{N}_p^i := \mathbf{N}^i(u_p),$ where $\mathbf{N}^i(u)$ is the unit normal function. If the point \mathbf{P} is the starting point $\mathbf{B}_S^i (u_p = u_{\min}^i),$ the unit normal vector at this point is on its right-hand side, denoted as $\mathbf{N}^i(u_p) := \mathbf{N}^i(u_p^+) := \mathbf{N}^i(u_{\min}^i);$ likewise, if the point \mathbf{P} is the ending point $\mathbf{B}_E^i (u_p = u_{\max}^i),$ the unit normal at this point is on its left-hand side as $\mathbf{N}^i(u_p) := \mathbf{N}^i(u_p^-) := \mathbf{N}^i(u_{\max}^i).$ Therefore, the center of a tangent circle $\Theta_p(r)$ is represented as $\mathbf{O}^{\Theta_p} := \mathbf{B}_p^i + r \cdot \mathbf{N}_p^i,$ where the circle radius $r > 0.$ This type of tangent circles is defined as $\Theta_p(r) := \{(u_p, r) : \mathbf{B}_p^i, \mathbf{O}^{\Theta_p}, r\}.$

Second, for a point \mathbf{P} on a circular arc $\mathbf{B}^i(u)$ of center $\mathbf{O}^{\mathbf{B}^i}$ and radius $r^{\mathbf{B}^i},$ the point parameter u_p is within its domain, $u_{\min}^i \leq u_p \leq u_{\max}^i.$ It is easy to understand that the largest circle tangent to the arc at this point \mathbf{B}_p^i and inside the arc is the corresponding circle of the arc. Thus, the largest tangent circle Θ_p^{\max} is the circular arc $\mathbf{B}^i(u),$ whose center $\mathbf{O}^{\Theta_p^{\max}}$ is $\mathbf{O}^{\mathbf{B}^i}$ and radius $r^{\Theta_p^{\max}}$ is $r^{\mathbf{B}^i}.$ If the point \mathbf{P} is at the starting point, the unit normal vector \mathbf{N}_p^i can be calculated as

$$\mathbf{N}_p^i := \mathbf{N}^i(u_p^+) = \mathbf{N}^i(u_{\min}^i) = \begin{bmatrix} -\cos(u_{\min}^i) \\ -\sin(u_{\min}^i) \end{bmatrix}, \tag{1}$$

and if the point \mathbf{P} is at the ending point, the unit normal vector \mathbf{N}_p^i can be calculated as $\mathbf{N}_p^i := \mathbf{N}^i(u_p^-) = \mathbf{N}^i(u_{\max}^i).$

Third, for a point \mathbf{P} of parameter u_p on a NURBS component curve $\mathbf{B}^i(u),$ its unit tangent vector $\mathbf{T}_p^i := \mathbf{T}^i(u_p)$ and unit normal vector $\mathbf{N}_p^i := \mathbf{N}^i(u_p)$ can be calculated. The curvature $\kappa_p := \kappa(u_p)$ of this curve at this point can be found using the following equation,

$$\frac{d}{du} [\mathbf{T}^i(u)] = \kappa(u) \cdot \left| \frac{d}{du} \mathbf{B}^i(u) \right| \cdot \mathbf{N}^i(u), \quad \text{and} \quad u = u_p. \tag{2}$$

When $\kappa_p > 0,$ the largest circle Θ_p^{\max} tangent to the NURBS curve at \mathbf{B}_p^i is the osculating circle of the NURBS curve at this point, and its radius $r^{\Theta_p^{\max}}$ is fixed at $1/\kappa_p,$ and the equation of the circle center is

$$\mathbf{O}^{\Theta_p^{\max}} := \mathbf{B}_p^i + r^{\Theta_p^{\max}} \cdot \mathbf{N}_p^i = \mathbf{B}_p^i + \frac{1}{\kappa_p} \cdot \mathbf{N}_p^i. \tag{3}$$

When $\kappa_p < 0,$ the tangent circle $\Theta_p(r)$ is not unique, and the circle center is $\mathbf{O}^{\Theta_p} := \mathbf{B}_p^i + r \cdot \mathbf{N}_p^i,$ where the circle radius is $r > 0.$

As all the pocket component curves are regular curves, at an interior curve point, whose parameter is $u_p,$ the unit tangent vectors on both sides of the point are the same, i.e., $\mathbf{T}^i(u_p^-) = \mathbf{T}^i(u_p^+).$ So there is only one unit normal vector \mathbf{N}_p^i at this point. In this work, the composite curves of the pocket boundaries connect one by one with at least position continuity. Some pair of curves is smoothly connected, sharing the same tangent direction at their joint; while some pair is not smoothly connected, having different tangent directions at their joint. Thus, for the unit tangent vectors of the two curves at the left and the right of the joint are the same, i.e., $\mathbf{T}^i(u_{\max}^i) = \mathbf{T}^{i+1}(u_{\min}^{i+1}),$ there is only one unit normal vector, i.e., $\mathbf{N}^i(u_{\max}^i) = \mathbf{N}^{i+1}(u_{\min}^{i+1}).$ However, for the non-smooth link of two component curves, the joint is a cusp and the unit normal vector at the joint is defined here. If the tangent vectors of the two curves at the joint meet the following condition,

$$[\mathbf{T}^i(u_{\max}^i) \times \mathbf{T}^{i+1}(u_{\min}^{i+1})]_z = \left| \begin{matrix} Tx^i(u_{\max}^i) & Ty^i(u_{\max}^i) \\ Tx^{i+1}(u_{\min}^{i+1}) & Ty^{i+1}(u_{\min}^{i+1}) \end{matrix} \right| > 0, \tag{5}$$

which means this joint is a concave corner, no unit normal vector at the joint is defined. Otherwise, the angle θ_{\min} starting from $\mathbf{N}^i(u_{\max}^i)$ to $\mathbf{N}^{i+1}(u_{\min}^{i+1})$ can be calculated as

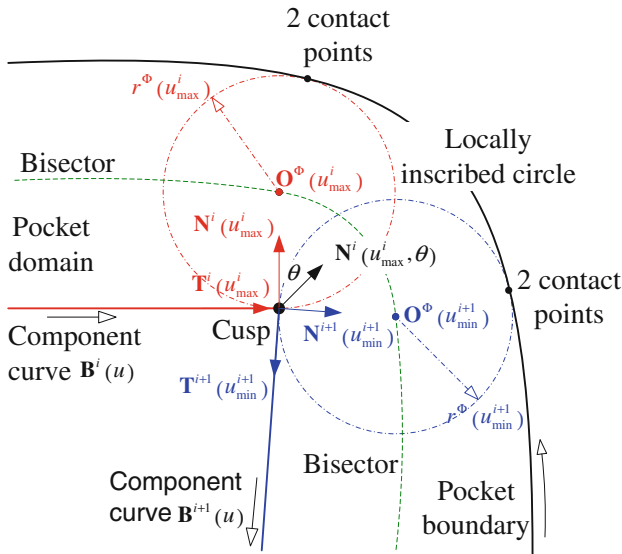


Fig. 1 An illustrative diagram for finding valid unit normal vectors at a cusp between two adjacent component curves

$$\theta_{\min} = -\cos^{-1}[\mathbf{T}^i(u_{\max}) \cdot \mathbf{T}^{i+1}(u_{\min})], \tag{6}$$

where $-\pi < \theta_{\min} < 0$. The valid unit normal vector at the joint can be defined as

$$\mathbf{N}^i(u_{\max}, \theta) := \mathbf{N}^{i+1}(u_{\min}, \theta) : \\ = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \mathbf{N}^i(u_{\max}), \tag{7}$$

where $\theta \in [\theta_{\min}, 0^\circ]$. Hence, any unit normal vector between $\mathbf{N}^i(u_{\max})$ and $\mathbf{N}^{i+1}(u_{\min}^{i+1})$ is a valid unit normal vector at the joint (see Fig. 1).

2.2 Terminologies and definitions of medial axis transform

The medial axis transform (MAT) of the domain of a pocket is a completed shape descriptor, which concisely represents the domain shape and is very useful to manipulate the complex pocket domain (see Fig. 2). The MAT is based on the medial axis (MA), which geometrically bisects the pocket domain and becomes the pocket skeleton. Although MA has been defined in the published articles, we update its definition in our work for clarity. For this purpose, locally inscribed circle (LIC) is first introduced. As before, the external and internal boundaries are denoted as $\mathbf{B}^i(u)$ with parameter u , and a locally inscribed circle at a boundary point u , its center and radius are denoted as $\Phi(u)$, \mathbf{O}^Φ , and r^Φ , respectively.

Definition 2.2 Given a pocket domain $\Omega \subset \mathbb{R}^2$ and its boundary $\mathbf{B}(u)$, at a boundary point \mathbf{P} with parameter u , a locally inscribed circle $\Phi(u)$ of center \mathbf{O}^Φ and radius r^Φ is defined if and only if a circle meets four conditions: (1) $\Phi(u) \subseteq \Omega$, (2) $\forall [\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_j, \dots, \mathbf{P}_m] := \Phi(u) \cap \mathbf{B}(u)$, $m \geq 1$, $|\mathbf{O}^\Phi \mathbf{P}_1| = |\mathbf{O}^\Phi \mathbf{P}_2| = \dots = |\mathbf{O}^\Phi \mathbf{P}_j| = \dots = |\mathbf{O}^\Phi \mathbf{P}_m|$, (3) $\forall \mathbf{P}_j := \mathbf{B}^i(u_j)$, $\mathbf{O}^\Phi \mathbf{P}_j \cdot \mathbf{T}^i(u_j) = 0$, $j = 1, 2, \dots, m$, and (4) the radius r^Φ is the maximum. Here, the locally inscribed circle (LIC) is represented as $\Phi(u) := \{\mathbf{O}^\Phi(u), r^\Phi(u), \Phi(\mathbf{O}^\Phi, r^\Phi), [\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_m]\}$ (see Fig. 2). At any boundary point, its LIC is the largest circle tangent to the boundary inside the pocket without any intersection with the boundary.

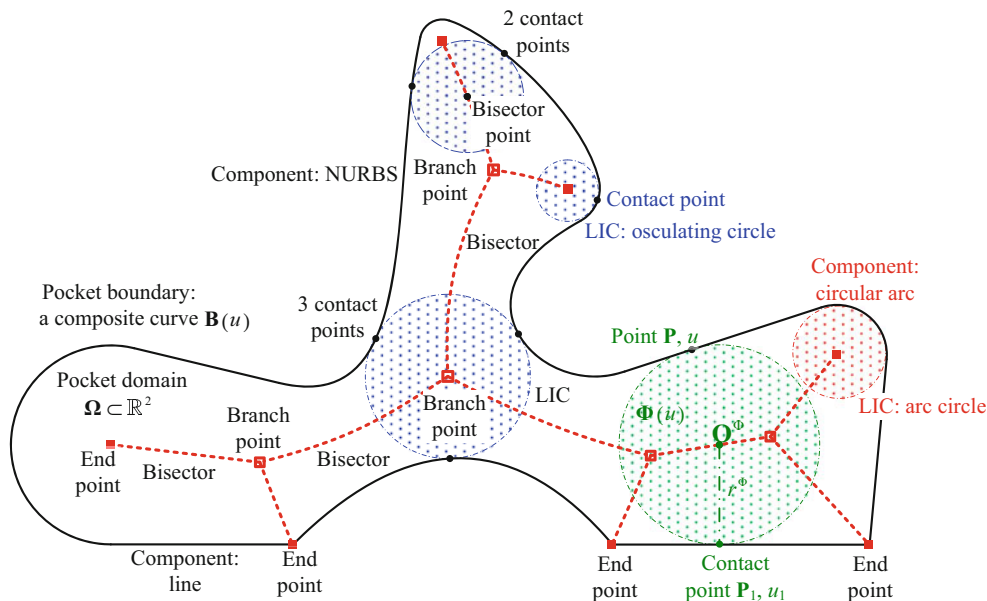


Fig. 2 Illustration of local inscribed circles and the basic elements of the MA of a pocket

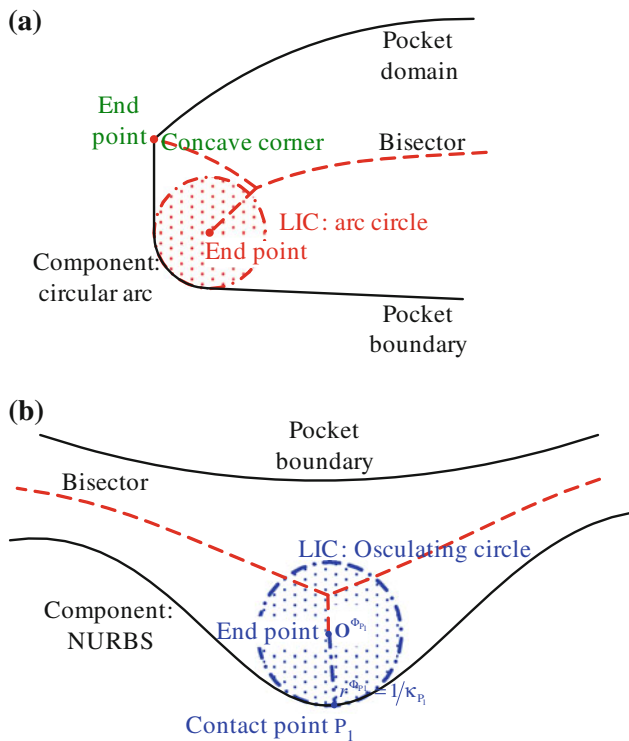


Fig. 3 Diagram of end points in three conditions: **a** for the first and the second conditions, and **b** for the third condition

Definition 2.3 The MA definition is that the MA of a pocket domain Ω is the locus of the center of the locally inscribed circle (disk) $\Phi(u)$ moving along the pocket boundary $\mathbf{B}(u)$ to cover the whole domain, which is represented as $\mathbf{MA} := \{\mathbf{O}^\Phi : \mathbf{O}^\Phi \in \Phi(u), \cup \Phi(u) = \Omega\}$ (see Fig. 2).

Remarks The LIC varies in size at different locations as it best fits the local area bounded by the pocket boundary at any location. The MA is a “middle” axis bisecting the pocket domain. In this work, the pocket MA is represented with a set of end and branch points connected by bisector curves, which are defined in the following.

Definition 2.4 The definition of end point \mathbf{V}^E is that the center of the LIC $\Phi(u)$ is an end point if and only if it meets one of the three conditions: (1) the component curve $\mathbf{B}^i(u)$ that the LIC $\Phi(u)$ touches is a circular arc, (2) at a concave corner, the radius r^Φ of the LIC is zero at \mathbf{P}_1 , where the unit normal vector is invalid, and (3) $\forall \mathbf{P}_1 \in \Phi(u) \cap \mathbf{B}(u)$ and $\mathbf{P}_1 := \mathbf{B}_{P_1}^i$, its curvature κ_{P_1} is positive and maximal in the local region of \mathbf{P}_1 (see Fig. 3).

Remarks The three conditions in the definition of end point infer to three special cases of LIC. These special circles are the circle of a component arc, the degenerated circle—a point—at the joint of two component curves

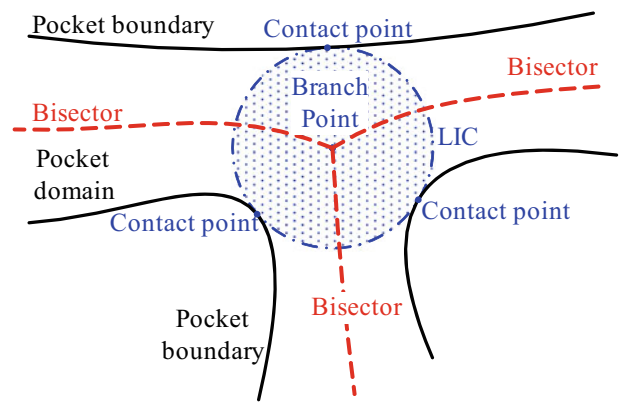


Fig. 4 Diagram of the branch point definition

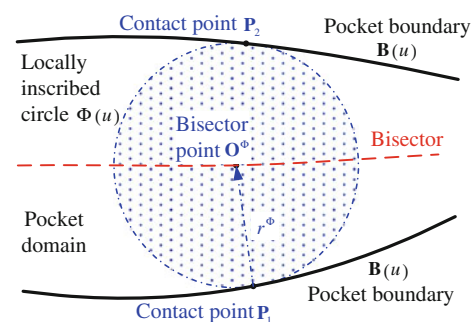


Fig. 5 Diagram of the bisector point definition

forming a sharp concave corner, and the osculating circle of a NURBS component curve at a point having the maximum curvature in its neighborhood.

Definition 2.5 The definition of branch point \mathbf{V}^B is that the center of an LIC $\Phi(u)$ is a branch point if and only if this circle touches the pocket boundary at more than two points (see Fig. 4).

Remarks If an LIC contacts the boundary n times ($n > 2$), the circle center is a branch point where n bisectors are connected (see Fig. 4). Generally, the LIC at a branch point touches the pocket boundary at three discrete points (Fig. 5).

Definition 2.6 The definition of bisector point is that the center \mathbf{O}^Φ of the LIC $\Phi(u)$ is a bisector point if and only if this circle contacts the pocket boundary $\mathbf{B}(u)$ at two points $[\mathbf{P}_1, \mathbf{P}_2]$. Thus, the bisector curve $\mathbf{O}^\Phi(u)$ is a set of bisector points and connects a pair of end-and-branch or branch points.

Definition 2.7 The medial axis transform (MAT) refers to the MA, the radii and centers of the LICs and the contact points between these circles and the boundaries. Thus, the

MAT is defined as $\mathbf{MAT} := [\Phi(u) \cup \Phi(u) = \Omega]$, and it can be simply represented as $\cup[(\mathbf{V}^E, \mathbf{V}^B), \mathbf{O}^\Phi(u)]$.

To find the MAT of a pocket, it is crucial to calculate the LIC of a bisector point accurately and efficiently. This work proposes a new mathematical model of the LIC.

3 A mathematical model of LIC

The essential element of the MAT is the LICs, and an innovative mathematical model of LIC is established in the section.

3.1 The maximal tangent circle

At a point of the pocket boundary where various tangent circles Θ exist, the LIC Φ is unique among the circles. These circles Θ can be found using the equations rendered in Sect. 2. To facilitate finding the LIC Φ , the maximal tangent circle is defined.

Definition 3.1 For a point \mathbf{P}_1 on a boundary component curve $\mathbf{B}^i(u)$ with parameter u_{P_1} , the unit normal at this point is valid and the boundary curve is not a circular arc. The normal vector intersects the opposite boundary $\mathbf{B}(u)$ first at point \mathbf{Q}_1 with parameter u_{Q_1} . Sometimes, the opposite boundary $\mathbf{B}(u)$ could be the curve $\mathbf{B}^i(u)$ itself. The tangent circle $\Theta_{P_1}(r)$ determined with the diameter line $\mathbf{P}_1\mathbf{Q}_1$ is defined as the maximal tangent circle $\Theta_{P_1}^{\max} := \Theta_{P_1}(\frac{1}{2}|\mathbf{P}_1\mathbf{Q}_1|)$ in this work. If the component curve $\mathbf{B}^i(u)$ is a circular arc of radius r^{B^i} , the arc circle is the maximal tangent circle, $\Theta_{P_1}^{\max}(r^{B^i}) = \mathbf{B}^i(u)$.

Lemma 3.2 At a point \mathbf{P}_1 with parameter u_{P_1} on a component curve $\mathbf{B}^i(u)$, its LIC $\Phi_{P_1} := \Phi(u_{P_1})$ is within or, in some special cases, is its maximal tangent circle $\Theta_{P_1}^{\max}$ (see Fig. 6).

Proof The proof covers two cases: the first case is that the component curve $\mathbf{B}^i(u)$ of the pocket boundary $\mathbf{B}(u)$ is not a circular arc, while the second case is that curve $\mathbf{B}^i(u)$ is a circular arc. For the first case, at the point \mathbf{P}_1 on curve $\mathbf{B}^i(u)$ with parameter u_{P_1} , its unit normal vector $\mathbf{N}_{P_1}^i$ and its tangent circles $\Theta_{P_1}(r)$ in different size can be found. Suppose the unit normal vector intersects the boundary at point \mathbf{Q}_1 , the maximal tangent circle $\Theta_{P_1}^{\max}$ is determined with diameter $\mathbf{P}_1\mathbf{Q}_1$. Assume the boundary traverses the maximal tangent circle once with two intersection points \mathbf{Q}_1 and \mathbf{Q}_2 , leaving a boundary segment within the circle and bounded by these points. By continuously reducing radius r of the tangent circle $\Theta_{P_1}(r)$ at the point \mathbf{P}_1 to shrink it, the intersection points $\mathbf{Q}_1(r)$ and $\mathbf{Q}_2(r)$ gradually come closer and the boundary segment inside the circle

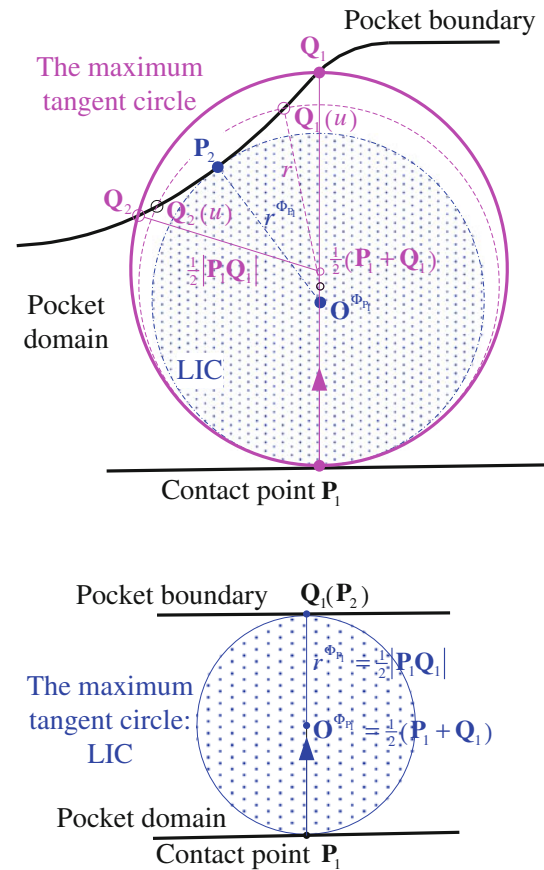


Fig. 6 Diagram of the maximal tangent circle and its relationship with the LIC

becomes shorter (see Fig. 6). By doing this, it is guaranteed that the radius r can reach a critical value r^Φ so that the points $\mathbf{Q}_1(r)$ and $\mathbf{Q}_2(r)$ converge to one point \mathbf{P}_2 and the boundary touches the tangent circle. According to Definition 2.2, this circle is the LIC Φ_{P_1} , $(\{\mathbf{O}^\Phi(u_{P_1}), r^\Phi(u_{P_1}), \Phi(\mathbf{O}^\Phi, r^\Phi), [\mathbf{P}_1, \mathbf{P}_2]\})$. Therefore, it is within the maximal tangent circle (see Fig. 6). If the boundary traverses the maximal tangent circle m times, there are m pairs of intersection points, $(\mathbf{Q}_1, \mathbf{Q}_2), (\mathbf{Q}_3, \mathbf{Q}_4), \dots, (\mathbf{Q}_{2m-1}, \mathbf{Q}_{2m})$. For each pair of the intersection points, a critical circle of radius $r^j, j = 1, 2, \dots, m$ can be found. Among them, if only one circle is the minimal, it is the LIC and within the maximal tangent circle. The center of this circle is a bisector point of the MA. In some special cases, several circles are the same and in the minimal size; thus, this minimal circle is the LIC whose center is a branch point of the MA. Occasionally, if the maximal tangent circle contacts the boundary right at the point \mathbf{Q}_1 without any boundary crossing, this maximal circle itself is an LIC. The center of this circle is a bisector point of the MA.

For the second case that the component curve is a circular arc, the maximal tangent circle of a point on this curve is the arc circle. Similarly, if the boundary traverses

this circle, an LIC can be found by reducing the radius of the tangent circle. It is evident that the LIC is within the maximal tangent circle (see Fig. 6). Otherwise, the maximal tangent circle is the LIC, whose center is an end point of the MA. \square

Remarks The maximal tangent circle can facilitate finding the LIC. By checking the boundary segments inside the maximal tangent circle, the LIC can be efficiently found, instead of exhaustively searching the whole boundary.

3.2 Optimization model of calculating a bisector point

A mathematical model of LIC is very important for the accuracy and efficiency of computing the MA of a pocket. In general, a MA consists of a set of end and branch points and a group of bisector curves, and each curve connects a pair of end and branch points or two branch points. Since closed-form equations of the bisector curves cannot be derived, discrete bisector points should be computed to represent the bisector curves. Thus, a new and accurate model of calculating a bisector point is proposed in this section, in which an innovative idea of formulating the LIC at a boundary point is to shrink the maximal tangent circle until the tangent circle touches the boundary.

To formulate this model, assuming a point \mathbf{P}_1 of parameter u_{p_1} on the component curve $\mathbf{B}^i(u)$ of the boundary $\mathbf{B}(u)$, the position vector is $\mathbf{P}_1 = [Bx^i(u_{p_1}), By^i(u_{p_1})]^T$, and the unit normal vector $\mathbf{N}_{p_1}^i = [Nx^i(u_{p_1}), Ny^i(u_{p_1})]^T$ can be found using the equations in Sect. 2. The first intersection point \mathbf{Q}_1 between the unit normal and the boundary $\mathbf{B}(u)$ can be easily calculated; its parameter is u_{Q_1} and the position vector is $\mathbf{Q}_1 = [Bx(u_{Q_1}), By(u_{Q_1})]^T$. Therefore, the diameter of the maximal tangent circle is $\mathbf{P}_1\mathbf{Q}_1$, and the center is $\mathbf{O}^{\Theta_{P_1}^{\max}} = \frac{1}{2}(\mathbf{P}_1 + \mathbf{Q}_1)$. The maximal tangent circle $\Theta_{P_1}^{\max}$ ($\frac{1}{2}|\mathbf{P}_1\mathbf{Q}_1|$) is found (see Fig. 7). For a bisector point, the simple, general situation is that the boundary traverses the maximal tangent circle once with a segment inside the circle. Based on Lemma 3.2, the LIC is within the maximal tangent circle.

To attain a smaller tangent circle at point \mathbf{P}_1 , select a test point \mathbf{P} with parameter u on the boundary segment inside the maximal tangent circle; $\mathbf{P} = [Bx(u), By(u)]^T$ and $|\mathbf{PO}^{\Theta_{P_1}^{\max}}| < \frac{1}{2}|\mathbf{P}_1\mathbf{Q}_1|$. Here, the test point \mathbf{P} cannot be the point \mathbf{P}_1 , so $u \neq u_{p_1}$. A tangent circle of point \mathbf{P}_1 and passing through point \mathbf{P} can be constructed after finding its center in two steps: (1) connecting the two points with \mathbf{PP}_1 , and (2) drawing the mid-section line of \mathbf{PP}_1 and intersecting $\mathbf{P}_1\mathbf{Q}_1$ at point $\mathbf{O}^{\Theta_{P_1}}$. Thus, the center of the tangent circle $\Theta_{P_1}(r(u))$ is $\mathbf{O}^{\Theta_{P_1}}$, and the circle radius $r(u)$ can be calculated as

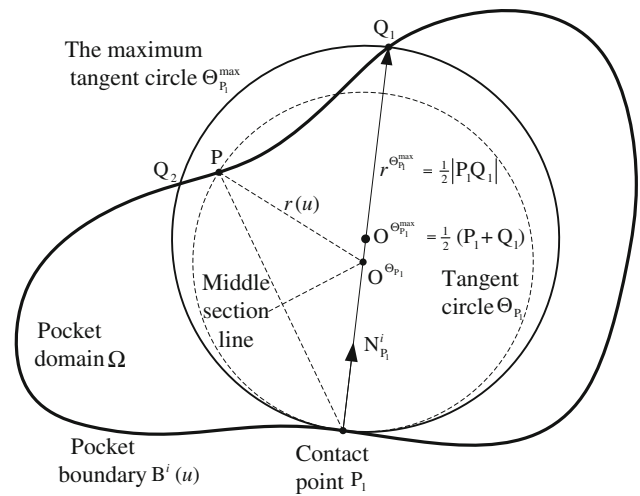


Fig. 7 Diagram of formulating the mathematical model of calculating a bisector point

$$r(u) = \frac{1}{2} \cdot \frac{[Bx(u) - Bx^i(u_{p_1})]^2 + [By(u) - By^i(u_{p_1})]^2}{[Bx(u) - Bx^i(u_{p_1})] \cdot Nx^i(u_{p_1}) + [By(u) - By^i(u_{p_1})] \cdot Ny^i(u_{p_1})} \tag{17}$$

By changing the point \mathbf{P} along the boundary segment inside the maximal tangent circle, the radius of the tangent circle can gradually be reduced. Ultimately, when point \mathbf{P} reaches a special point \mathbf{P}_2 , the radius is the minimum, and, besides point \mathbf{P}_1 , the tangent circle only contacts the boundary at this point. Therefore, this tangent circle is the LIC Φ_{P_1} of point \mathbf{P}_1 . The optimization model of the LIC is

$$\begin{aligned} &\text{Minimize } r(u) = \frac{1}{2} \cdot \frac{(Bx(u) - Bx^i(u_{p_1}))^2 + (By(u) - By^i(u_{p_1}))^2}{(Bx(u) - Bx^i(u_{p_1})) \cdot Nx^i(u_{p_1}) + (By(u) - By^i(u_{p_1})) \cdot Ny^i(u_{p_1})} \\ &\text{Subject to } \begin{cases} u \neq u_{p_1} \\ |\mathbf{PO}^{\Theta_{P_1}^{\max}}| < \frac{1}{2}|\mathbf{P}_1\mathbf{Q}_1| \end{cases} \end{aligned} \tag{18}$$

However, in many cases, the boundary traverses the maximal tangent circle several times and leaves several segments inside the circle (see Fig. 8). According to Lemma 3.2, the LIC is within the maximal tangent circle and touches one of the boundary segments, without the boundary traversing inside the circle. As the test point \mathbf{P} moves along the boundary segments, the radius of the tangent circle changes in a highly non-linear way. It is clear that in this case the LIC model is a difficult global optimization problem. To solve this problem efficiently and consistently, the hybrid global optimization method is developed and introduced in the next section.

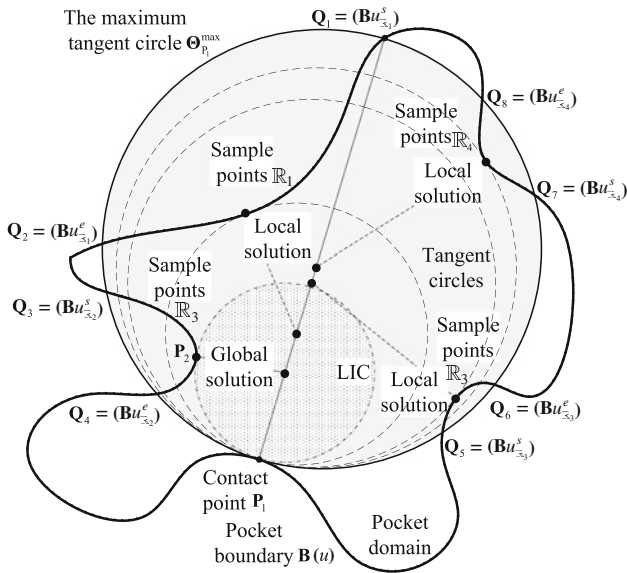


Fig. 8 Diagram of the discrete particle swarm optimization method in this work

4 The hybrid global optimization method

For the LIC at a boundary point P_1 , solving the above optimization problem actually is to search the boundary for point P_2 that the circle contacts the boundary. However, due to the complex pocket shape, finding point P_2 can take a long time and result in non-consistent solutions by using either a prior global or a local optimization method. With regard to the current global optimization methods, their salient feature is that they can intelligently and efficiently search for the region of the exact solution; however, it often takes intolerably long time to pin-point the solution. As comparison, the local optimization methods are not able to globally search the solution; while, if the solution region is initially provided, they can converge at the solution in very short time. To take the advantages of the global and local optimization methods, this work develops an effective hybrid optimization method that integrates a global and a local optimization method. In the first step, the discrete particle swarm optimization (PSO) method is employed to narrow down the solution region from the whole boundary; and in the second step, Newton’s gradient method is applied to locate the exact solution.

4.1 The discrete particle swarm optimization method

The problem of locating point P_2 on the boundary for the LIC at point P_1 actually is to search for P_2 in the 1-D continuous range of the boundary parameter. As an outstanding feature, the population-based intelligent search of the present PSO method is to have a swarm of particles scattering across the problem domain, searching and sharing the objective function

values in an iterative process, in order to efficiently find out the global solution [23, 24]. Thus, the PSO method is an effective tool to solve the problem. To push the limit of searching efficiency in this work, an improvement is to represent the whole continuous domain with a fairly large number of discrete points and modify the current PSO method to search among these points for the finite boundary piece around point P_2 .

To find out the finite boundary piece where point P_2 lies, a large number of sample points (the number is N_S) are taken on the pocket boundary $B(u)$, $u \in [u_{min}, u_{max}]$. For example, the points are evenly distributed along the boundary with the same parameter interval $\Delta u = (u_{max} - u_{min}) / (N_S - 1)$, and the point parameters are $[u_1, u_2, \dots, u_j, \dots, u_{N_S}]$ where $u_j = u_{min} + (j - 1) \cdot \Delta u$. In this work, the points are sampled on the boundary with approximately the same arc length between two adjacent points. The points are denoted as $B_j : = B(u_j)$ where $j = 1, 2, \dots, N_S$. The finite boundary piece around point B_j is denoted as B_j . In this problem, due to the constraint that only the boundary points within the maximal tangent circle are the candidates of the solution in this step, all the points B_j are checked against the in-equation,

$$|B_j O_{P_1}^{\Theta_{P_1}^{max}}| < \frac{1}{2} |P_1 Q_1|. \text{ Then, the boundary points inside the maximal tangent circle define the problem sub-domains } \mathbb{R}_1 = [u_{\mathbb{R}_1}^s, \dots, u_{\mathbb{R}_1}^e], \dots, \mathbb{R}_l = [u_{\mathbb{R}_l}^s, \dots, u_{\mathbb{R}_l}^e], \dots, \mathbb{R}_{N_D} = [u_{\mathbb{R}_{N_D}}^s, \dots, u_{\mathbb{R}_{N_D}}^e], \text{ for example, } [u_1, \underbrace{u_2, u_3, u_4}_{\mathbb{R}_1}, \dots, \underbrace{u_{j-1}, u_j, u_{j+1}}_{\mathbb{R}_l}, \dots, \dots, \underbrace{u_{N_S-1}, u_{N_S}}_{\mathbb{R}_{N_D}}]. \text{ In}$$

order to quickly find the piece of the solution point P_2 , a discrete particle swarm optimization (PSO) method is developed in this work (see Fig. 8).

Here, the discrete PSO method is introduced. Given the boundary $B(u)$ in non-continuous sub-domains as $\mathbb{R}_1, \dots, \mathbb{R}_l, \dots, \mathbb{R}_{N_D}$, and many discrete points B_j in the sub-domain are calculated (see Fig. 8). At beginning of the search, a swarm of particles with a number of N_P is randomly selected among the discrete points in the domains. The particles are denoted as $B_{s(k)}$ and their parameters are denoted as $u_{s(k)}$ where $k = 1, 2, \dots, N_P$ refers to the particles and $s(k)$ refers to the sequence number of the sample points. The initial velocity v_k of each particle is also determined with a random number. In the searching iteration, the velocity of each particle v_k and a parameter value for this particle u are computed using the equation below,

$$\begin{cases} v_k = w \cdot v_k + c_1 \cdot rand \cdot (u_k^* - u_{s(k)}) \\ \quad + c_2 \cdot rand \cdot (u^* - u_{s(k)}) \\ u = u_{s(k)} + v_k \end{cases}, \tag{19}$$

where the inertia weight w is 0.9, the acceleration coefficients c_1 and c_2 are 0.2 and 2.0, respectively, and r and is the uniform distribution function that can generate random numbers between 0 and 1. If the in-equation

$$u \in \mathbb{R}_l \quad \text{and} \quad \frac{u_{j-1} + u_j}{2} < u \leq \frac{u_j + u_{j+1}}{2} \tag{20}$$

is true, $s(k) = j$ and $u_{s(k)} = u_j$. If the in-equation

$$u_{s(k)} \in \mathbb{R}_l \quad \text{and} \quad u_{\mathbb{R}_l}^e \leq u \leq u_{\mathbb{R}_{l+1}}^s \tag{21}$$

is true, $s(k)$ is the sequence number of the point with parameter $u_{\mathbb{R}_{l+1}}^s$, and $u_{s(k)} = u_{\mathbb{R}_{l+1}}^s$. If the in-equation

$$u_{s(k)} \in \mathbb{R}_l \quad \text{and} \quad u_{\mathbb{R}_{l-1}}^e \leq u \leq u_{\mathbb{R}_l}^s \tag{22}$$

is true, $s(k)$ is the sequence number of the sample point with parameter $u_{\mathbb{R}_{l-1}}^e$, and $u_{s(k)} = u_{\mathbb{R}_{l-1}}^e$. Then, by using Eq. 18, the radii $r(u_{s(k)})$ of the tangent circles for all the particles are calculated. For each particle, if the radius is less than $r(u_k^*)$ in the previous iteration, the best parameter value of the particle $u_k^* = u_{s(k)}$. If the radius is less than $r(u^*)$, the best parameter value of the swarm $u^* = u_{s(k)}$, where $k = 1, 2, \dots, N_p$. After many iterations, the particles will converge to point \mathbf{B}_j with parameter u_j in the sub-domains.

Since the boundary domain is divided into several sub-domains with discrete sample points based on the constraint, the search range of the particles has been significantly reduced, thus, it is very quick to find out the solution point \mathbf{P}_2 . The finite boundary piece of this point is the piece where the exact solution lies, and this point is the input for searching with Newton’s gradient method.

4.2 Accurate Newton’s optimization method

By using the discrete PSO method, the finite boundary piece of the exact solution can be found. Because the piece is very short, the objective function in the region of this piece is in a uniform concave shape, and the Newton’s gradient method can guarantee converge to the solution. The first derivative of the objective function in terms of the parameter u is

With the closed-form equation of the first derivative of the objective function, the linear search method is applied, and the solution can be found in several steps.

To show the super efficiency of this new optimization solver, the NURBS boundary curve shown in Fig. 8 is employed as an example, and our method and a related program of commercial geometric kernel software [25] are applied to this example to find the LICs at 1,000 locations on the boundary. The commercial program takes 4,286 ms to compute all the LICs. In our optimization method, 2,000 sample points are taken from the boundary. It takes 1,919 ms to compute 1,000 LICs at the same locations. Although we do not know the algorithm used in the commercial program, the testing result shows that the hybrid optimization method is more efficient than this program.

This newly proposed hybrid global optimization method uses the discrete PSO method to globally search for the local region of the solution in the large problem domain, and feed the search result to the Newton’s local optimization method for the exact solution. This method takes the advantages of the global and local optimization methods and gets rid of their weakness. It is very efficient to solve highly non-linear optimization problems, including the problem in this work.

5 Medial axis approximation

By using this new approach, a bisector point of an LIC at a boundary point can be accurately found; which is the main contribution of this work. Now, to generate the MA with high accuracy, a practical method is proposed, which includes (1) a large number of bisector points are computed and are fit with a curve to approximate the MA, and (2) a number of different bisector points are calculated and are compared to the MA approximation for the MA errors; if an error is larger than the tolerance, these bisector points, together with the previous bisector points, are used to re-generate the MA approximation. This is a numerical way to globally bind the error of the MA approximation. Since the MA equations cannot be established, this problem is very difficult to address, and further study is needed.

$$r'(u) = \frac{[Bx(u) - Bx^i(u_{P_1})] \cdot Bx(u)' + [By(u) - By^i(u_{P_1})] \cdot By(u)'}{[Bx(u) - Bx^i(u_{P_1})] \cdot Nx^i(u_{P_1}) + [By(u) - By^i(u_{P_1})] \cdot Ny^i(u_{P_1})} - \frac{[(Bx(u) - Bx^i(u_{P_1}))^2 + (By(u) - By^i(u_{P_1}))^2] \cdot [Nx^i(u_{P_1}) \cdot Bx(u)' + Ny^i(u_{P_1}) \cdot By(u)']}{2 \cdot [(Bx(u) - Bx^i(u_{P_1})) \cdot Nx^i(u_{P_1}) + (By(u) - By^i(u_{P_1})) \cdot Ny^i(u_{P_1}))^2]} \tag{23}$$

6 Applications

This new approach has been implemented into our in-house software by using the C++ programming language. A Windows PC with an AMD Athlon 2.7 GHz CPU is used. To demonstrate that it is more efficient and accurate than the prior MA methods, this new approach is applied to a thin, long pocket with a closed free-form boundary shown in Fig. 9. Among the current MA methods, the most popular MA method, the Voronoi diagram generation method, is selected to find the MA of the pocket. The results by using the two methods are explained in detail and are compared to confirm the advantage of our approach.

By using the new approach to finding the bisector of the pocket, 40 points are sampled on the pocket boundary, and 40 bisector points are accurately calculated, at which the location errors are less than $1.0e-7$. Then, a curve is fit to the bisector points to represent the bisector curve of the pocket shown in Fig. 9a. So, the location error of the bisector curve with regard to every boundary points, instead of at the 40 sample points, can be calculated. The error graph is plotted in Fig. 9b, and the maximum error is $5.84e-3$ mm. Since the location errors at the 40 discrete bisector points are much less than the maximum error, the new approach to calculating bisector points is accurate, and the bisector curve error mainly is the bisector fitting error. Therefore, when more boundary points are sampled, more

bisector points are attained, and the bisector curve is more accurate.

Here, a group of tests has been conducted by using this new approach. In each test, by sampling a different number of points on the boundary, their corresponding bisector points are computed, and the bisector curve is found. Then, the maximum error of the bisector curve is calculated. The results are listed in Table 1. When 390 points are sampled, the 390 bisector points are very accurate, thus, the bisector curve accuracy ($8.11e-7$ mm) is quite high and the computing time (359 ms) is very quick.

In this work, the Voronoi diagram generation method is employed to find the MA of the pocket. First, 82 points are sampled on the pocket boundary to form a polygon as its new discrete representation, and the Voronoi diagram is then found with 136 Voronoi points. In Fig. 10, the pocket boundary points and the corresponding Voronoi diagram points are plotted. Among all the Voronoi points, 80 bisector points inside the pocket can be found and shown in this diagram.

However, the 80 computed Voronoi points, which are used to approximate 80 bisector points, deviate from their corresponding true “middle” points (true medial axis points), so the distance between a computed Voronoi point and its corresponding true medial axis point is defined as a Voronoi location error. To show this problem with the

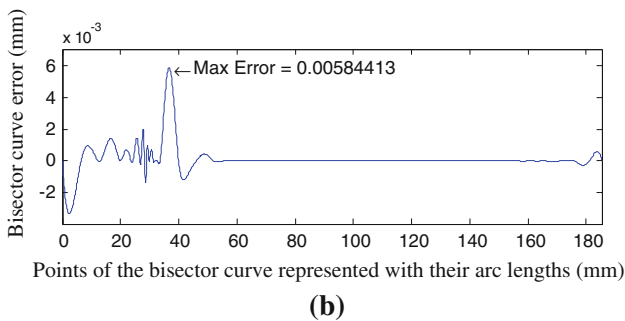
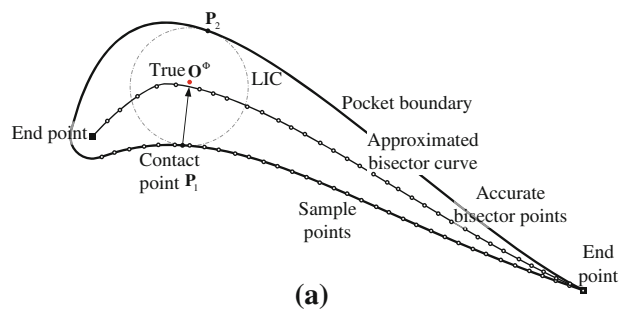


Fig. 9 a A group of accurate bisector points calculated with this new approach and an approximated bisector curve fitting these points, and b the error between the approximate bisector curve and the true bisector curve

Table 1 Results of a group of tests by using this new approach

Bisector points	Computing time with our new approach (ms)	Max error of bisector curve (mm)
40	38	$5.84e-3$
98	86	$1.68e-4$
196	171	$1.08e-5$
390	359	$8.11e-7$

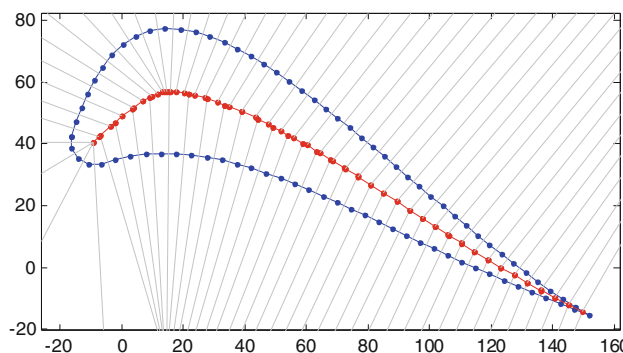
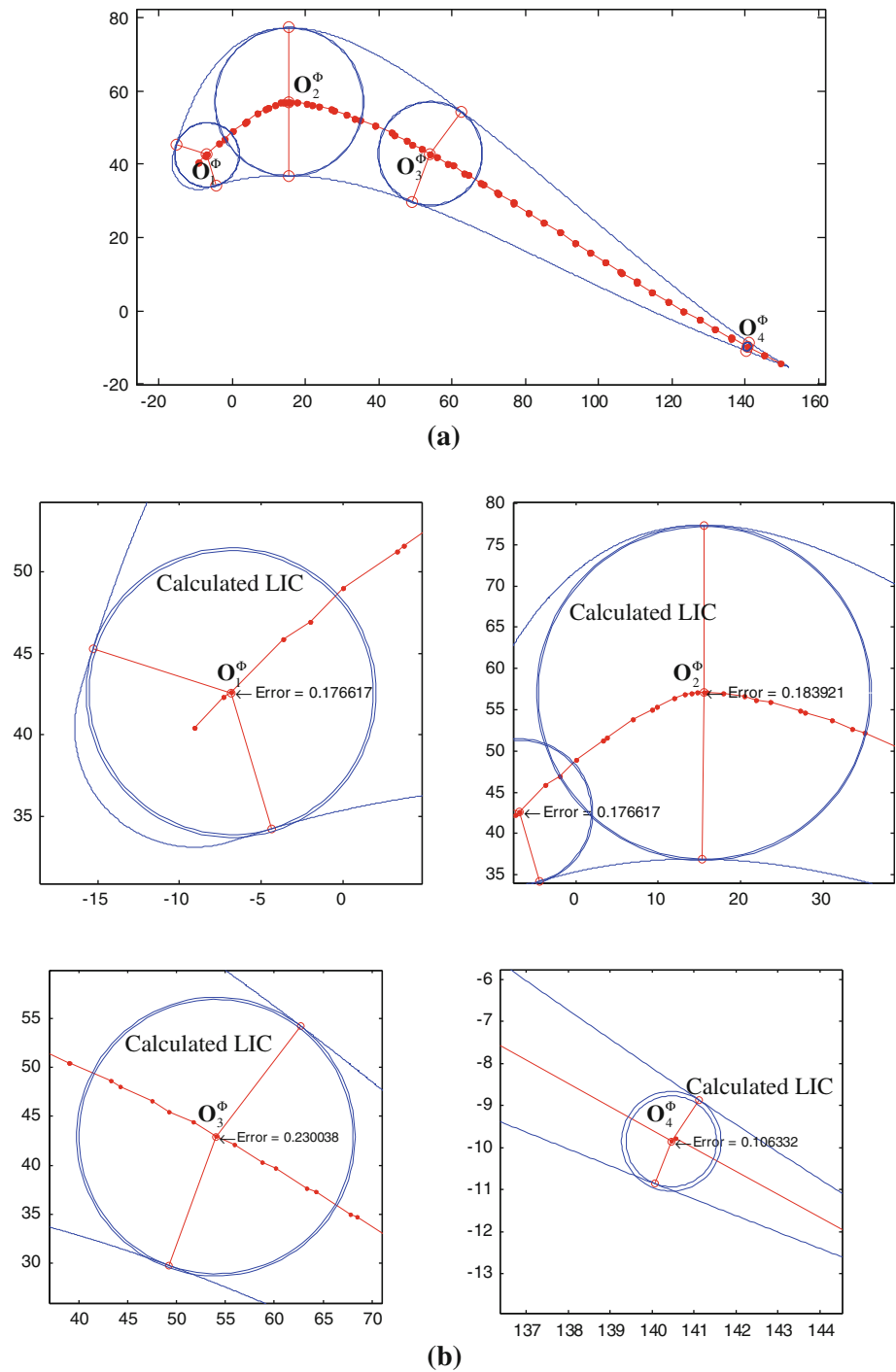


Fig. 10 The Voronoi diagram of the polygon approximating the pocket boundary

Fig. 11 The bisector point errors by using the Voronoi diagram and their detailed illustrations



Voronoi diagram, four Voronoi points and their errors are closely examined as shown in Fig. 11. At Voronoi point O_1^ϕ , the distances from this point to the boundary curves are not equal, and no circle centered at this point can be drawn to contact the boundary at two points. In Fig. 11, the location error is 0.176617 mm. Similarly, Voronoi points O_2^ϕ , O_3^ϕ and O_4^ϕ are off their true medial axis points, and their errors are 0.18392, 0.230038, and 0.183921 mm, and

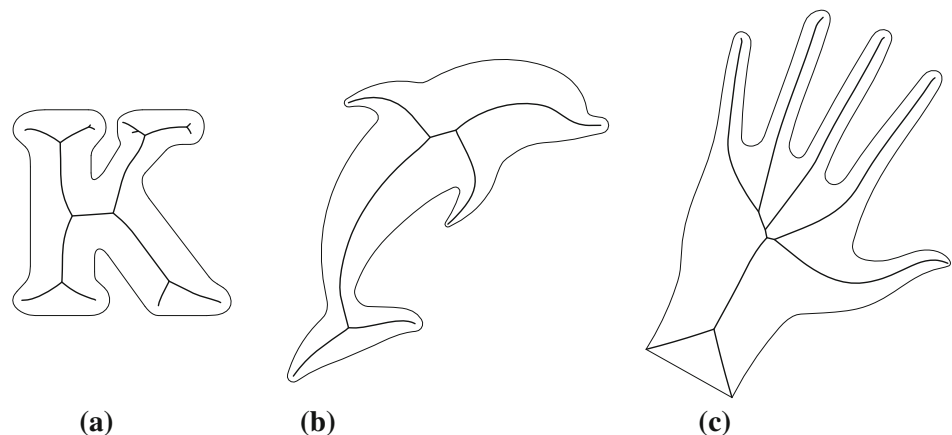
shown in Fig. 11, respectively. As a consequence, the location errors of the Voronoi points in terms of the true bisector points are large, and the maximum error of the bisector curve fitting these Voronoi points are large. Unfortunately, these bisector curves cannot be used for high precision tool path planning.

To fully represent the bisector curve, a group of discrete Voronoi points are computed to approximate bisector

Table 2 Results of a group of tests by using the Voronoi diagram generation method

Sample points of the boundary polygon	Bisection points (Voronoi points)	Computing time with MATLAB (ms)	Max error of the bisection curve (mm)
41	39 (67)	93	9.22e-1
82	80 (136)	120	2.92e-1
203	201 (343)	171	7.54e-2
405	403 (686)	317	3.18e-2
807	805 (1,370)	764	1.24e-2
2,014	2,012 (3,429)	3,441	4.34e-3
4,026	4,024 (6,931)	13,484	1.79e-3
8050	8,047 (14,274)	57,562	9.06e-4

points, and then a B-spline curve is fit to these points as the result. Thus, the bisector error is defined as the maximum deviation between the calculated and the true bisector curves. Eight tests with a different number of bisector points, which are extracted from the Voronoi points, are carried out; unfortunately, the bisector errors are quite large, which are listed in Table 2. For example, when the number of the bisector points is 41, the bisector error is 0.922 mm, while the error of the bisector curve that is fit to 40 bisector points found with our approach is 0.00584 mm. It is evident that the error of the bisector calculated with the Voronoi method is much greater than that of the bisector found with our approach. Since the curve fitting method is the same, the main reason for the large errors of the bisector points generated with the Voronoi method is that the Voronoi points are not accurate and the bisector points computed with our method are accurate, with regard to the true bisector points. It is easy to see that the computing time by using our new approach is much less than the time needed by using a MATLAB program of the Voronoi diagram method.

Fig. 12 The MAs of a group of testing examples with closed free-form pocket boundaries**Table 3** Data of finding the MAs of the three pockets by using our in-house software

Pockets	Boundary arc length (mm)	Bisector points	Number of bisector curves
(a)	553	322	19
(b)	556	316	7
(c)	1,068	604	11

By using our in-house software, the MATs of three pockets with very complex shape are calculated and shown in Fig. 12, and the results are listed in Table 3. The above examples demonstrate the robustness and effectiveness of the proposed algorithms when dealing with complex shapes.

7 Conclusions

In this work, an efficient, accurate approach to medial axis transforms of pockets with closed free-form boundary has been proposed. Compared to the prior medial axis generation methods, this new approach is able to handle pocket domains bounded with lines, circular arcs, and B-spline curves. It does not need to identify and eliminate curve/curve intersections, thus, it takes less time to compute MATs, compared to the analytical MA methods. It uses accurate boundary curves information to achieve much higher accuracy, compared to the Voronoi diagram approximation methods. The main contributions of this work include (1) an innovative optimization model of bisector points, which is able to accurately calculate any bisector point, and (2) a new global optimization method, a hybrid optimization method of the particle swarm optimization and the gradient optimization methods, which is able to efficiently pin-point any bisector point. This approach

substantially improves the technique of computing the medial axis transforms of pockets. It can be directly used to multiple cutters selection and their path generation for aggressive rough machining of pockets.

References

- Chen ZC, Fu Q (2011) An optimal approach to multiple tool selection and their numerical control path generation for aggressive rough machining of pockets with free-form boundaries. *Comput Aided Des* 43(6):651–663
- Sherbrooke EC, Patrikalakis NM, Wolter F (1996) Differential and topological properties of medial axis transforms. *Graphical Model Image Process* 58(6):574–592
- Choi HI, Choi SW, Moon HP (1997) Mathematical theory of medial axis transform. *Pac J Math* 181:57–88
- Farouki RT, Johnstone JK (1994) The bisector of a point and a plane parametric curve. *Comput Aided Geom Des* 11:117–151
- Farouki RT, Ramamurthy R (1998) Specified-precision computation of curve/curve bisectors. *Int J Comput Geom Appl* 8(5–6):599–617
- Elber G, Kim MS (1998) Bisector curves of planar rational curves. *Comput Aided Des* 30(14):1089–1096
- Culver T, Keyser J, Manocha D (2004) Exact computation of the medial axis of a polyhedron. *Comput Aided Geom Des* 21(1):65–98
- Aurenhammer F (1991) Voronoi diagrams—a survey of a fundamental geometric data structure. *Comput Surv* 23(3):345–405
- O'Rourke J (1998) *Computational geometry in C*. Cambridge University Press, New York
- Fabri A (2007) Voronoi diagrams in CGAL, the computational geometry algorithms library. In: *The 4th International Symposium on Voronoi diagrams in science and engineering* July 9–11, pp 8–11
- Brandt JW (1994) Convergence and continuity criteria for discrete approximations of the continuous planar skeleton. *CVGIP Image Underst* 59(1):116–124
- Dey TK, Zhao W (2004) Approximate medial axis as a Voronoi subcomplex. *Comput Aided Des* 36(2):195–202
- Chou JJ (1995) Voronoi diagrams for planar shapes. *IEEE Comput Graphics Appl* 15(2):52–59
- Ramamurthy R, Farouki RT (1999) Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I. Theoretical foundations. *J Comput Appl Math* 102(1):119–141
- Ramamurthy R, Farouki RT (1999) Voronoi diagram and medial axis algorithm for planar domains with curved boundaries II: detailed algorithm description. *J Comput Appl Math* 102(2):253–277
- Alt H, Cheong O, Vigneron A (2005) The Voronoi diagram of curved objects. *Discret Comput Geom* 34(3):439–453
- Seong JK, Cohen E, Elber G (2008) Voronoi diagram computations for planar NURBS curves. In: *Proceedings of the 2008 ACM Symposium on solid and physical modeling*, pp 67–77
- Gursoy HN, Patrikalakis NM (1992) An automatic coarse and fine surface mesh generation scheme based on medial axis transform I. Algorithms. *Eng Comput* 8(3):121–137
- Gursoy HN, Patrikalakis NM (1992) An automatic coarse and fine surface mesh generation scheme based on medial axis transform II. Implementation. *Eng Comput* 8(4):179–196
- Hassouna MS, Farag AA (2005) Robust centerline extraction framework using level sets. In: *The 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition June 20–25, vol I*, pp 458–467
- Alichholer O, Aigner W, Aurenhammer F, Hackl T, Juttler B, Rabl M (2009) Medial axis computation for planar free-form shapes. *Comput Aided Des* 41:339–349
- Kosinka J, Juttler B (2007) MOS surfaces: medial surface transforms with rational domain boundaries. In: *Proceedings of the 12th IMA International Conference on mathematics of surface, vol XII*, pp 245–262
- Kennedy J, Eberhart R (1995) Particle swarm optimization. *Anon IEEE, Perth, vol 4*, pp 1942–1948
- Bratton D, Kennedy J (2007) Defining a standard for particle swarm optimization. In: *The 2007 IEEE Swarm Intelligence Symposium April 1–5*, pp 120–127
- OpenCASCADE. <http://www.OpenCASCADE.Org>