Trochoidal Tool Paths for Pocket Machining with Full Control of the Tool Engagement Angle

Martin Held[1] , Josef Pfeiffer[2]

[1]Universität Salzburg, held@cs.sbg.ac.at
[2]Universität Salzburg, jpfeiffer@cs.sbg.ac.at

Corresponding author: Martin Held, held@cs.sbg.ac.at

Abstract. We extend and improve the MATHSM pocketing strategy by Elber, Cohen and Drake (Comput. Aided Des. 2005) to obtain trochoidal tool paths inside of planar pockets bounded by straight-line segments and circular arcs: We compute a dynamic spacing of the machining circles such that the engagement angle stays below a user-specified maximum engagement angle along the entire tool path. Since the amount of material removed is directly linked to the engagement angle, this effectively bounds the material removal along the entire path (for any given maximum feedrate). As a further improvement, we maintain a model of the area already machined by the tool as the tool moves along the path. Experiments clearly show that our improvements tend to result in substantially shorter tool paths compared to our implementation of the original MATHSM method, while guaranteeing that the engagement angle does not exceed the user-specified limit.

Keywords: pocket machining, engagement angle, trochoidal path, MATHSM, Voronoi diagram
DOI: https://doi.org/10.14733/cadaps.2025.731-747

1 INTRODUCTION

1.1 Cutting Width and Engagement Angle

Pocket machining is a roughing operation that clears out material within prescribed boundaries by the movement of a tool (or the part to be machined) within a plane. The underlying geometric path planning problem requires us to move a circular disk within a planar region such that the region has been fully swept at the end of the movement. Traditional strategies for generating such paths include the use of zigzag patterns and the use of offset curves to form contour-parallel patterns. See, e.g., Held [4] for a detailed discussion of both strategies. More recently, smooth spiral-based paths (e.g., [1, 8]) and trochoidal-like paths [3] have been applied.

Virtually all CAD/CAM software packages offer options for generating tool paths according to one or more of these strategies. However, while these paths can be expected to be correct from a purely geometric point of view — in the sense that the tool clears out the material completely without any gouging — they do not necessarily take into account key process parameters like the cutting width and the tool engagement angle. The *radial width of cut*, υ, frequently simply called *cutting width*, is commonly defined as the radial amount of the tool that is engaged in the material; see Fig. 1a. (A more formal definition, based on directed Hausdorff distances, is given in [11].) Since the actual value of υ only makes sense relative to the tool diameter, some machining handbooks prefer to regard the cutting width as the percentage s of the tool diameter D that is engaged in the material:

$$s = \frac{\upsilon}{D} \tag{1}$$

For a movement of the tool along straight lines, the cutting width equals the *immersion depth* δ. It depends linearly on the step-over distance between parallel lines of the tool path. Furthermore, it determines the material removal rate and reflects the cutting forces experienced by the tool.

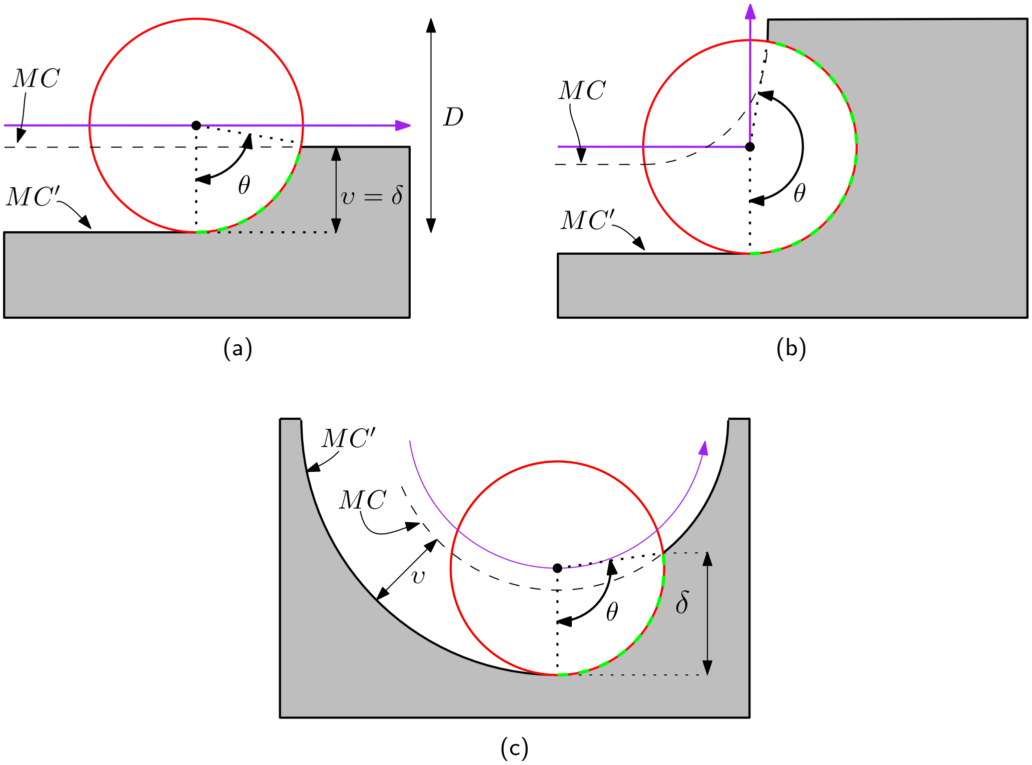


Figure 1: Cutting width υ and engagement angle θ for linear and circular motion of a tool (depicted by a red disk) along a tool path (shown in purple). The old machining contour is denoted by MC and the new contour is denoted by MC'. The unmachined material is shaded grey.

Unfortunately, the cutting width is no adequate proxy for the actual amount of material removed and for the resulting cutting forces when the tool path exhibits an abrupt change of direction or when the tool moves along circular portions of the tool path, see Fig. 1b and Fig. 1c. For instance, when the tool moves along a straight line segment into a 90° corner, initially the material removal rate is constant, then it increases monotonously until the tool reaches the actual corner vertex of the tool path, and then it drops instantaneously to the same original constant value. Similarly, in Fig. 1c we see that the immersion depth δ may be significantly larger than the cutting width υ. Note that υ is the same in Fig. 1a and in Fig. 1c. This problem pops up even if the corresponding straight-line segments and circular arcs of the tool path are parallel or concentric.

The actual cutting forces are better reflected by the *tool engagement angle* θ: It is the angle subtended by the circular arc that corresponds to the contact surface of the tool disk with the material being machined. (In Figs. 1a to 1c this circular arc is shown in dashed green.) That is, for a (stationary) tool the engagement angle is the angular amount of sweep subtended by each cutting edge of the tool as it engages and leaves the material. The higher this angle is, the more cutting edges of the tool are involved in the process which increases the material removal rate and, as a consequence, implies an increased cutting force.

Obviously, the engagement angle attains is absolute maximum, 360°, when the tool plunges vertically down into the material. During standard cutting moves the maximum of 180° is attained during full slotting moves. For straight-line moves of the tool, the engagement angle is fairly easy to compute. Standard mathematics yields for $0 \leq \theta \leq 180$ and $0 \leq s \leq 1$

$$\theta = \arcsin(2s-1) + 90 \tag{2}$$

and

$$s = \frac{1+\sin(\theta - 90)}{2}. \tag{3}$$

The computation of the engagement angle gets significantly more complicated as soon as the current segment of the tool path and the old machining contour are not two straight-line segments that are parallel. We emphasize that the engagement angle of a tool centered at a point c on the tool path does not only depend on the position of c relative to the old machining contour but also on the tangent vector of the tool path at c.

1.2 Prior Work

Since pocket machining is a basic task in manufacturing, a large number of studies on tool path generation have been published in this context. Still, little is known about global strategies to generate tool paths such that a user-specified maximum engagement angle is not exceeded. Most pocketing papers either ignore the engagement angle completely or provide only heuristics and, at best, try to argue experimentally that the engagement angle can be controlled.

For instance, Bieterman and Sandstrom [1] employ second-order partial differential equations to generate spiral tool paths that start out along a nearly circular segment and slowly morph towards the pocket boundary. Their experiments show that these spirals extend tool life significantly when cutting hard materials. This observation can be taken as an indication that the engagement angle stays within reasonable bounds. Similarly, the spiral tool paths by Held and de Lorenzo [8] guarantee to respect a user-specified maximum cutting width but no formal guarantee is given regarding the engagement angle. However, also their experiments show that a fairly uniform engagement angle is achieved. Wang, Jang and Stori [19] use metrics to evaluate key process parameters. These metrics allow them to optimize contour-parallel tool paths such that the average engagement angle is improved, but in the worst case the engagement angle might still jump (close) to 180°.

Stori and Wright [17] study spiral-in tool paths for convex pockets such that the engagement angle can be controlled along most portions of the path. General pockets have to be decomposed into convex pieces. The restriction to convex geometries is removed by an extension of their approach described by Ibaraki et al. [13]. However, Dumitrache, Borangiu and Dogar [2] report that their own reproduction of the tool paths of [13] witnessed switching between conventional and climb milling (and vice versa). Kim, Lee and Yang [16] use pixel-based computations to assess the material-removal rate of standard contour-parallel tool paths. The material removal rate is kept within bounds by replacing sharp corners of the paths with one or more trochoidal segments. Uddin et al. [18] focus their study of engagement angles on the last finishing paths along the pocket boundary.

More recently, Jacso, Matyasi and Szalay [14] published the "fast constant engagement offsetting method" (FACEOM). Their algorithm allows them to achieve a constant engagement angle relative to a given parametric curve. However, it remains completely unclear how their approach could be applied on a more global level such that a tool path can be planned for an entire pocket. (The follow-up paper [15] also does not contain figures of full tool paths for a pocket.)

1.3 Our Contribution

We provide an extension of the (one-sided) MATHSM trochoidal pocketing strategy by Elber, Cohen and Drake [3] for pockets (without holes) bounded by straight-line segments and circular arcs. Rather than blindly resorting to some fixed constant step-over distance, we adapt the step-over distance between each pair of subsequent circular path segments such that the tool engagement angle reaches but never exceeds a user-specified limit. Since the amount of material removed is directly linked to the engagement angle, this effectively bounds the material removal along the entire path for any given maximum feedrate. As a more global optimization, we keep track of the area already machined. This allows to increase the step-over distance even further if previous machining operations in other parts of the pocket have already covered some portion of the material that is currently to be removed. Furthermore, by analyzing the pocket geometry we are able to dynamically adapt the limit on the engagement angle depending on the "narrowness" of parts of the pocket. Experiments clearly show that our improvements tend to result in substantially shorter tool paths compared to our implementation of the original MATHSM method, while guaranteeing that the engagement angle does not exceed the user-specified limit.

2 TOOL PATH COMPUTATION

2.1 Basics

We study tool paths for pockets bounded by straight-line segments and circular arcs. A pocket $\mathcal{P}$ is assumed to be a simply-connected region of the plane. That is, no holes or islands are allowed. Our approach can be extended to multiply-connected pockets as described in [3], or by introducing "bridges" [12, 8].

The pocket boundary $\partial\mathcal{P}$ is assumed to be one Jordan curve that is oriented counter-clockwise (CCW). This orientation imposes an orientation of the straight-line segments and circular arcs of the boundary in a natural way. We call a circular arc *concave* if it is oriented clockwise (CW), and *convex* otherwise.

As usual, a disk centered at a point p within (the closure of) $\mathcal{P}$ is called a *clearance disk* if the entire disk is completely contained inside (the closure of) $\mathcal{P}$ and if its radius cannot be enlarged without protruding outside of $\mathcal{P}$. Hence, a clearance disk touches $\partial\mathcal{P}$ in at least one point. The radius of the clearance disk of a point p is called the *clearance distance* of p. Roughly, the medial axis of $\partial\mathcal{P}$ (within $\mathcal{P}$) is given by the union of the centers of all those clearance disks of $\mathcal{P}$ that touch $\partial\mathcal{P}$ in at least two disjoint points. The medial axis is a subset of the Voronoi diagram of $\partial\mathcal{P}$, and it can be derived easily from the Voronoi diagram. We refer to Held [4, 7] for a detailed discussion of Voronoi diagrams, medial axes and their use for offsetting.

It is obvious that only those parts of a pocket can be machined that can be covered by the tool disk with radius r without resulting in gouging. We call a pocket *machinable* if it equals the union of the disks of radius r centered at all points inside of $\mathcal{P}$ whose clearance distance is at least r. To convert an input pocket to a machinable pocket, one could (1) compute an interior offset of the pocket boundary for offset distance r and (2) compute an exterior offset with offset distance r of that interior offset. If the original pocket was machinable then the resulting exterior offset would be identical to the boundary of the input curve.

Since our implementation relies heavily on Voronoi diagrams and medial axes computed by means of VRONI/ARCVRONI [6, 9], we ensure the machinability of the input geometry by computing offsets by means of the Voronoi diagram of the pocket boundary and subsequent modification of the input geometry and its

Voronoi diagram. The basic ideas for obtaining the boundary of the machinable area (relative to a tool radius r) from an input geometry are sketched in Fig. 2.

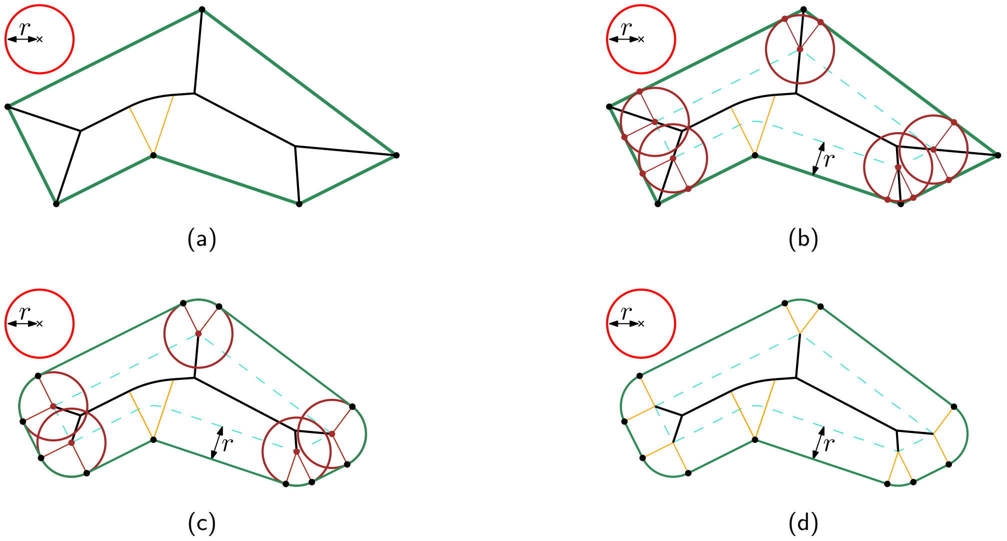


Figure 2: Illustration of the transformation from original input to the machinable pocket boundary. In (a) the medial axis (black) and the other Voronoi edges (orange) of the pocket boundary (dark green) are shown. The red tool disk (with radius r) is depicted in the upper-left corner. (For the sake of visual clarity, a fairly large tool is used.) In (b) clearance disks (brown) are centered on the intersection points of the offset curve (dashed turquoise) for radius r with the medial axis. In (c) and (d) the modified Voronoi diagram and the new pocket geometry can be seen. The offset curve in (d) for the modified contour is the same as in (b) for the original input.

For the sake of simplicity, in the sequel, we assume that the entire pocket $\mathcal{P}$ is machinable relative to the tool radius r. Actually, to avoid special cases in our algorithm, we assume that all convex arcs of the pocket boundary $\partial\mathcal{P}$ have radii that are strictly greater than r. (That is, the transformation sketched in Fig. 2 is carried out for an offset distance $r + \varepsilon$, for some small $\varepsilon > 0$.) When talking about Voronoi diagrams and medial axes we will be sloppy and simply refer to the Voronoi diagram of $\partial\mathcal{P}$ restricted to the interior of $\mathcal{P}$ as the Voronoi diagram of $\mathcal{P}$; similarly for the medial axis of $\mathcal{P}$.

Our tool paths are suitable for conventional milling, also known as up-cut milling. It would be straightforward to modify our approach such that climb milling is supported.

2.2 MATHSM Algorithm

In a nutshell, the "medial axis transform toward high-speed machining of pockets" (MATHSM) algorithm by Elber, Cohen and Drake [3] uses the medial axis of $\mathcal{P}$ to compute clearance disks. The circles of these clearance disks are then combined to form a trochoidal path. The path is an alternating series of "machining circles" and tangential "transition elements" between pairs of machining circles. A "central MATHSM" algorithm and a "one-sided MATHSM" algorithm are introduced in [3]. Since our algorithm generalizes the one-sided MATHSM algorithm, we will only sketch this variant.

Consider a point c_{i-1} with clearance disk A_{i-1} and clearance distance $\rho_{i-1} + r$ within $\mathcal{P}$ (for $\rho_{i-1} > 0$). (In the sequel we will also take the liberty to regard A_{i-1} as a clearance circle rather than a disk.) The circle M_{i-1} centered at c_{i-1} with radius ρ_{i-1} is the *machining circle* of c_{i-1}, and c_{i-1} is its *machining center*. By the definition of clearance disks, this circle is fully contained within the interior of $\mathcal{P}$. If c_{i-1} does not lie on the medial axis of $\mathcal{P}$ then A_{i-1} touches $\partial\mathcal{P}$ in exactly one point p_{i-1}. Similarly for some other machining center c_i and its machining circle M_i; cf. Fig. 3. In the sequel we will frequently need to talk about moving the tool disk in such a way that its center is moved along some curve. We simplify diction and simply talk about moving the tool (disk) along that curve.

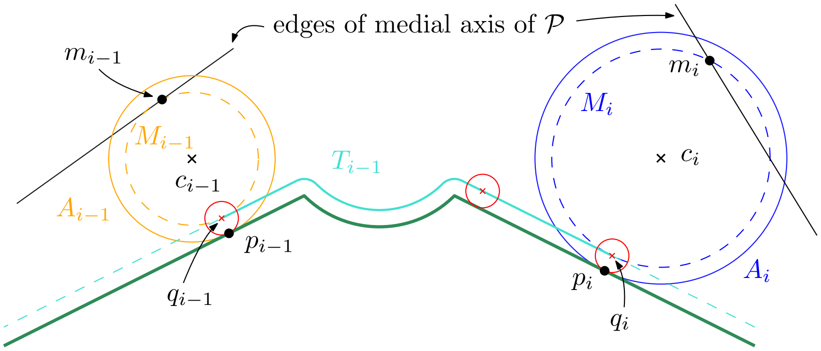


Figure 3: The basic building blocks of a MATHSM path: Two subsequent machining circles M_{i-1} and M_i and the transition element T_{i-1} (in turquoise) for linking them. The pocket boundary $\partial\mathcal{P}$ is indicated by a dark green line and its offset curve (for offset distance r) is drawn as a dashed turquoise line. The three small red circles depict the tool disk.

These machining circles are the main curves along which the tool disk is moved. In order to end up with one continuous path, two subsequent machining circles M_{i-1} and M_i are linked by a *transition element* T_{i-1} as follows: The offset curve for offset distance r is intersected with the line segment between c_{i-1} and p_{i-1}, yielding a point q_{i-1}. (Note that we get indeed one offset curve rather than a family of offset curves since $\mathcal{P}$ is machinable.) Similarly we get q_i as the intersection of the line segment between c_i and p_i with the offset curve. Then T_{i-1} is obtained by moving along the offset curve from p_{i-1} to p_i in CCW manner.

The construction described above yields the following part of a trochoidal tool path: The center of the tool starts at q_{i-1}, moves along M_{i-1} once in CCW direction until it returns to q_{i-1}, and then proceeds along the transition element T_{i-1} to q_i. From there it would continue CCW along M_i, etc. Since the area swept by the tool when moved along M_{i-1} is a subset of the clearance disk A_{i-1}, no gouging can occur during this motion. And, by construction, no gouging can occur when the tool moves along T_{i-1}. We conclude that any tool path built by such an alternating sequence of machining arcs and transition steps is free of gouging.

It is obvious that a distant spacing of the machining circles as shown in Fig. 3 would not be suitable for a real machining process. For the one-sided MATHSM, Elber et al. [3] place a machining center c_i such that it is the midpoint of q_i and the (closest) intersection point m_i of the medial axis of $\mathcal{P}$ with the line through p_i and c_i. Thus, the line segment between m_i and q_i forms a diameter of M_i with center c_i; see Fig. 3.

Suppose that all material within A_{i-1} has been removed. Then c_i should be placed such that the region formed by the union of A_{i-1} and the area swept by the tool disk when moved along M_i is path connected and simply connected. This goal is achieved if during the entire move of the tool along M_i the tool disk and A_{i-1} intersect. Standard high-school mathematics leads to the following condition on the maximal spacing of c_i relative to c_{i-1}:

$$\|c_i - c_{i-1}\| + \rho_i - \rho_{i-1} \leq 2r \tag{4}$$

The actual spacing of the machining centers is not discussed in [3]. However, comments in its section on extending the basic MATHSM algorithm suggest that some (unknown) constant spacing is applied, either with $\|c_i - c_{i-1}\|$ or with $\|m_i - m_{i-1}\|$ being constant.

In order to plunge the tool into the material at the start of the tool path, Elber et al. [3] suggest spiraling down along some machining circle M_0 until the required depth of cut is reached. The final result of their algorithm is a trochoidal tool path that allows clearing all material within a pocket without gouging.

The obvious advantage of MATHSM is that the tool path is G^1 continuous. Furthermore, since it consists predominantly of circular arcs with fairly large radii, the radial acceleration exerted on the tool is minimized [3]. The material removal rate changes along a MATHSM path: It decreases and increases, with actual machining taking place along about one half of the path. (This is a feature inherently linked to the nature of MATHSM paths.) Worse, depending on the unclear constant spacing of the machining centers and the geometry of the pocket, the material removal rate may change drastically and may be fairly high. But it is important to note that this change is continuous at any position on a MATHSM path.

At the end of their paper, Elber et al. [3] comment that a dynamic strategy that adapts the spacing distances according to machining parameters can be expected to be beneficial. In the rest of this paper, we pick up this lead and explain how to extend their MATHSM algorithm such that a spacing of the machining centers is obtained that guarantees to keep the tool engagement angle below a user-specified limit.

2.3 Computing the Engagement Angle

Suppose that the tool with radius r has moved along the machining circle M_{i-1} with radius ρ_{i-1} centered at c_{i-1}, and suppose that all material within the circle A_{i-1} has been removed. Recall that A_{i-1} has radius $r + \rho_{i-1}$ and is also centered at c_{i-1}. After traversing the transition step T_{i-1}, the tool will move in CCW direction along the next machining circle M_i centered at c_i. In the rest of this subsection, we regard r, c_{i-1}, and ρ_{i-1} as fixed. The MATHSM construction implies that ρ_i is also fixed once c_i has been fixed.

Let q be a position of the tool center on M_i for which cutting occurs. We denote the intersections of the tool circle with A_{i-1} by a and b, with b being that point the tool has not yet swept over, cf. Fig. 4a. (These two intersections have to exist for any meaningful spacing of c_{i-1} and c_i; otherwise, material would be left uncut between A_{i-1} and the region swept by moving the tool along M_i.) The intersection point of the ray from c_i through q with A_i is denoted by w. Then the engagement angle of the tool centered at q is given by the angle $\theta := \angle bqw$ at the vertex q of the triangle $\Delta(b, q, w)$. We note that fixing the position of q on M_i also fixes the position of b on A_{i-1}, and vice versa.

For which position q of the tool center on M_i is θ maximized? Trivially, maximizing θ is equivalent to minimizing the angle $\angle c_i q b$ at the vertex q of the triangle $\Delta(c_i, q, b)$. By construction, we have $\|q - c_i\| = \rho_i$ and $\|q - b\| = r$. That is, these two edges of $\Delta(c_i, q, b)$ have fixed constant lengths. We conclude that the angle $\angle c_i q b$ is minimum exactly if the edge length $\|b - c_i\|$ is as short as possible. This happens when the points c_{i-1}, c_i and b are collinear and occur in that order along the common line, cf. Figure 4b.

However, naïvely placing q on M_i such that c_{i-1}, c_i and b are collinear may lead to an overestimation of the maximum engagement angle that occurs while moving the tool along M_i: Figure 4c shows a setting where w ends up within A_{i-1}. Since the tool does not interact with the material at w, the angle $\angle bqw$ exceeds the true engagement angle for this tool position. Now recall that moving b away from the line through c_{i-1} and c_i (along A_{i-1}) causes the engagement angle to decrease. Hence, we move b in CCW direction along A_{i-1} just far enough to allow w to coincide with the intersection of A_{i-1} and A_i.

We conclude that the maximum engagement angle can be determined as follows: (1) Consider the ray u that starts at c_{i-1} and passes through c_i, and obtain the intersection point b by intersecting u with the circle A_{i-1}, cf. Figure 4b. (2) Intersect the tool circle centered at b with the circle M_i; the intersection point that is right of the ray u is the point q sought. (3) Intersect the ray from c_i through q with the circle A_i, thereby obtaining w. If w is outside of A_{i-1} then let $\theta := \angle bqw$. Otherwise, as illustrated in Fig. 4d, in (4) obtain w as the intersection of A_{i-1} and A_i. (Again pick the intersection point right of the ray u.) (5) Compute q as the (first) intersection of the ray from w through c_i with M_i. (6) Obtain b as the other intersection of A_{i-1} with the tool circle centered at q, and let $\theta := \angle bqw$.

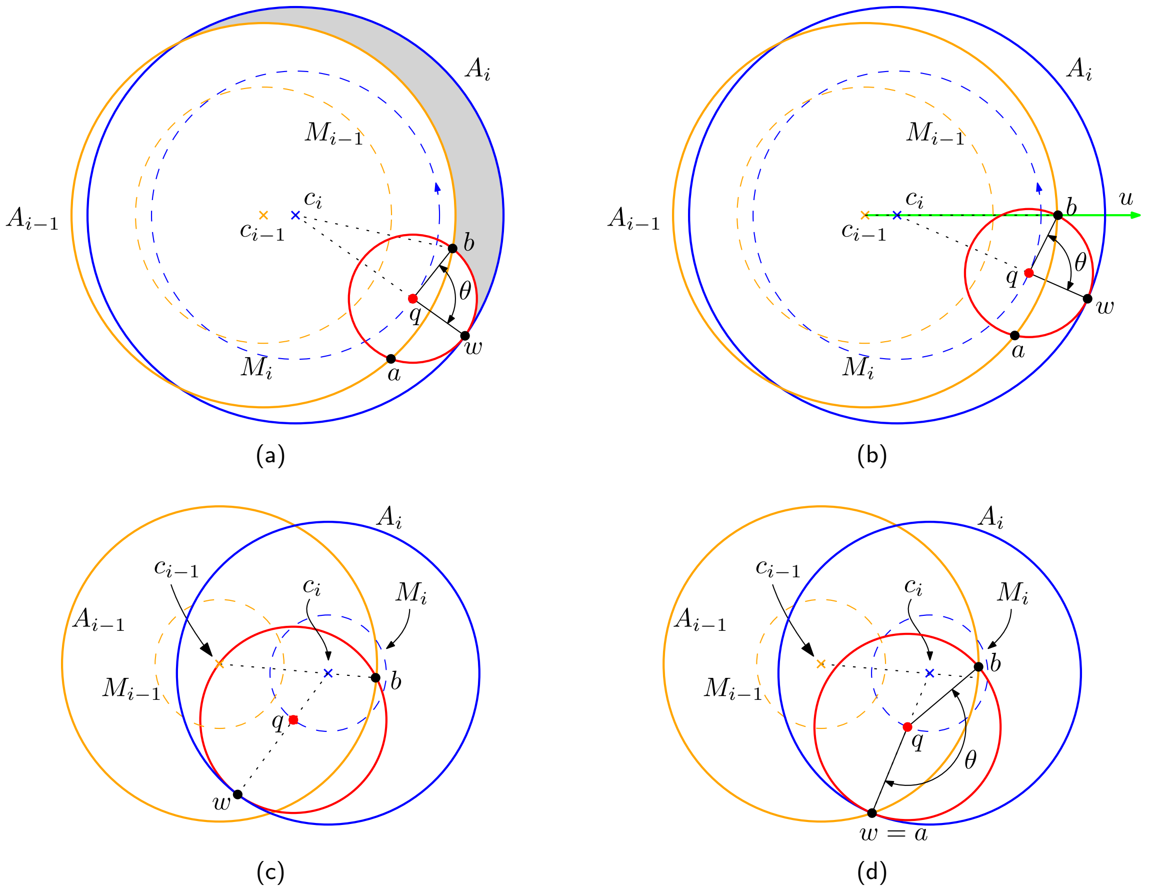


Figure 4: Computing the maximum tool engagement angle when the (red) tool disk is moved CCW along the machining circle M_i, under the assumption that no material is left within A_{i-1}. In (a) the setting is shown for a position q of the tool center; the engagement angle θ is given by the angle $\angle bqw$. The material yet to be removed is shaded grey. For the same geometric setting of M_{i-1} and M_i and the same tool radius, in (b) the tool position q for which the maximum engagement angle relative to A_{i-1} is assumed is shown. Subfigure (c) shows a setting for which the engagement angle would be overestimated if $\angle bqw$ would be considered because w lies within A_{i-1}. For this setting, the correct maximum engagement angle is shown in (d).

These computational steps can be cast into an explicit formula for θ in dependence on the machining center c_i and radius ρ_i. We sketch how the formula for the setting shown in Fig. 4b is derived: Without loss of generality, suppose that c_i coincides with the coordinate origin and that b lies on the positive x-axis. (This can be achieved by applying a suitable rigid motion.) Let b_x denote the x-coordinate of b. (It is given by the x-coordinate of c_{i-1} plus $(\rho_{i-1}+r)$.) The point q is an intersection point of the circle $x^2+y^2=\rho_i^2$ (centered at the origin c_i) with the circle $(b_x-x)^2+y^2=r^2$ (centered at b). We get

$$q_x = \frac{b_x^2 - r^2 + \rho_i^2}{2b_x} \tag{5}$$

and

$$q_y = -\sqrt{\rho_i^2 - \frac{(b_x^2 - r^2 + \rho_i^2)^2}{4b_x^2}}. \tag{6}$$

The point w is given by q plus the unit position vector of q times r. Since the dot product of the unit vector from q to w with the unit vector from q to b equals the arccosine of θ, we get (with the help of Mathematica) the following simple formula for θ:

$$\theta = \arccos\left(\frac{b_x^2 - r^2 - \rho_i^2}{2r\rho_i}\right) \tag{7}$$

The formula for θ for the setting of Fig. 4d can be derived similarly, but it is a bit more complicated. We refrain from stating the formula because it provides little additional insight.

2.4 Determining the Next Feasible Position of a Machining Circle

We are now ready to describe how the next machining center c_i is determined such that the maximum engagement angle θ stays below a user-specified limit $\theta_{\max} < 180$. (If $\theta_{\max} := 180$ then even full slotting moves were allowed and no tool path would exceed this limit.) Obviously, no machining occurs if $\|c_i - c_{i-1}\| = 0$ and, thus, $\theta = 0$. The maximum engagement angle starts to grow as soon as c_i is moved away from c_{i-1} in the direction of the unmachined material.

The explanation given in the previous subsection allows to compute the maximum engagement angle θ for any machining center c_i (relative to c_{i-1} and ρ_{i-1}). Unfortunately, we have not been able to find a closed-form solution for the inverse problem: Given $\theta_{\max}$, compute c_i such that moving the tool along M_i centered at c_i results in a maximum engagement angle $\theta = \theta_{\max}$.

As illustrated in Fig. 3, the MATHSM algorithm places c_i on a "middle" curve between the medial axis of $\mathcal{P}$ and $\partial\mathcal{P}$: The machining center c_i is at a distance ρ_i from the point m_i on the medial axis and at a distance $\rho_i + r$ from its normal projection p_i onto $\partial\mathcal{P}$. Inequality 4 implies a trivial upper bound on the maximum permissible distance d of c_i from c_{i-1}:

$$d < 2r + \rho_{i-1} - \rho_i < 2r + \rho_{i-1} \tag{8}$$

If the distance d equals or exceeds $2r + \rho_{i-1}$ then a full slotting move occurs and the engagement angle is guaranteed to be $180°$.

Summarizing, for $d := 0$ we have $\theta = 0$ and for $d := 2r + \rho_{i-1}$ we have $\theta = 180$. Hence, we apply bisection to find a suitable spacing distance d between c_{i-1} and c_i such that $\theta = \theta_{\max}$. In our implementation, we stop the bisection routine as soon as (in radian) $\theta_{\max} - \varepsilon \leq \theta \leq \theta_{\max}$ for $\varepsilon := 0.001$. Experience drawn from myriads of invocations of the bisection routine tells us that the bisection needs 9–18 iterative steps to converge.

Of course, we do not attempt to model the "middle" curve between the medial axis and $\partial\mathcal{P}$ explicitly as the loci of all potential machining centers. (Since the edges of the medial axis are portions of conics it would be rather awkward to come up with a precise mathematical expression for this curve.) Rather, in parallel we move away from p_{i-1} along $\partial\mathcal{P}$ (in CCW direction) towards p_i and accordingly from m_{i-1} along the medial axis towards m_i. This traversing of the medial axis required for locating m_i is very similar to the traversing required for offsetting, and we refer to literature on Voronoi-based offsetting for details; see, e.g., [4].

Since our implementation relies on Vroni/ArcVroni [6, 9] for the computation of the medial axis, we benefit from its specific way of storing Voronoi diagrams and medial axes as planar graphs. In particular, Vroni/ArcVroni stores (via a formula) the clearance distances of the points of the medial axis. Thus, once m_i has been fixed, the distance from m_i to p_i is known without any need for further computation. Furthermore, for every edge e of the medial axis and, thus, also for every point m_i on e, the corresponding boundary segments of $\partial\mathcal{P}$ (on the left and right side of e) are known.

3 IMPROVING THE TOOL PATH

3.1 Keeping Track of the Machined Area

Recall that m_i lies on some edge e of the medial axis and that the circle M_i passes through m_i; cf. Fig. 3. Hence, the disk A_i will protrude over e. Since the interior of A_i models (a super-set of) the area machined by the tool when moving along M_i, it is important that A_i contains points on both sides of e. That is, this overlap ensures that the pocket gets machined completely.

On the other hand, this overlap implies that it makes a difference whether or not the other side of e has already been machined when driving the tool along M_i: If the other side of e has already been machined then a smaller portion of M_i will result in the actual removal of material. In particular, in this case the maximum engagement angle along M_i may be smaller than without previous machining on the other side of e, and our scheme may overestimate this angle. As a consequence, we may end up picking an unnecessarily small distance between c_{i-1} and c_i. Such an overestimation of the engagement angle would not render the tool path invalid but it should be expected to be a source of inefficiency because it will make the tool path longer than necessary.

As a remedy, we model the area machined by the tool so far and restrict the computation of the engagement angle to that portion of M_i along which genuine machining occurs. Since we are mostly interested in the neighborhood of the edges of the medial axis, we may neglect the transition elements and focus only on the machining circles for modeling the machined area. Thus, when moving the tool along M_i, the area machined so far is given by (a super-set of) the union of the disks $A_0, A_1, \ldots, A_{i-1}$. (In general, the area machined is a genuine super-set of the union of these disks since cutting also takes place when the tool is moved along the transition elements.) This is a union of disks whose foot-points $p_0, p_1, \ldots, p_{i-1}$ are sorted in CCW order along $\partial\mathcal{P}$. Similarly, the order of these footpoints along $\partial\mathcal{P}$ induces an order on $m_0, m_1, \ldots m_{i-1}$. That is, the important insight is that we do not have to struggle with the union of a random arrangement of disks but with disks that are sorted relative to $\partial\mathcal{P}$.

The boundary of the union of these disks is given by a sequence of circular arcs that are sorted according to the order of their underlying disks. In order to update the boundary when A_i is added, it suffices to scan the sequence of the underlying disks "backwards" in CW direction. If an arc s of the boundary is contained completely within A_i then s is not required for any future computations and we remove it from the boundary. If, however, the arc s intersects the circle A_i then we compute the point of intersection, shorten s accordingly and split A_i into two arcs; one of these arcs can also be discarded. Thus, either s is shortened and the check for intersections may stop, or it can be discarded entirely. This is the key property that allows us to maintain the boundary of the union of these disks efficiently. In terms of complexity, maintaining this boundary results in an amount of extra work that is linear in the number of machining circles.

The contour of the area already machined is used during the computation of the engagement angle for a newly positioned machining circle, recall Fig. 4: If the point b determined by our algorithm lies within the area already machined then it is moved in CW direction along A_{i-1} until it matches the intersection of A_{i-1} with that contour. This decreases the engagement angle to its correct value. We call this improvement of our "standard" approach the "contour-aware" approach.

3.2 Dynamic Adjustment of the Maximum Engagement Angle

Discussions of the second author of this paper with NC engineers who use trochoidal paths for practical machining applications made it apparent that expecting a user to set a fixed limit $\theta_{\max}$ on the engagement angle may not be good enough in practice when machining hard metal. Apparently, if the tool is immersed deeply and if the material removal is constrained to a small region bounded by uncut material then some chips may be not thrown out of the pocket. However, in that case those chips may end up being cut a second time, which may cause them to become extremely hot, thus welding them to the surface. As a consequence, in such a case it may be desirable to reduce the material removal rate, i.e., to decrease the limit on the engagement angle.

Typical places where such a problem is more likely to occur are bottlenecks of $\partial\mathcal{P}$. Fortunately, identifying bottlenecks of $\partial\mathcal{P}$ is straightforward if its Voronoi diagram is known. Due to VRONI's parameterization of the edges of a Voronoi diagram, a simple graph search suffices to identify all bottlenecks in time linear in the number of segments of $\partial\mathcal{P}$. Furthermore, the widths of the bottlenecks can also be obtained easily; see [5] for details.

Thus, we proceed as follows: We scan the medial axis and determine all bottlenecks together with their widths. The bottlenecks get marked on the medial axis and on $\partial\mathcal{P}$. When the tool path moves for the first time through a bottleneck — that is, if no machining has already taken place on the other side of the current edge of the medial axis — then the limit for the engagement angle is reduced. In our implementation, we use a simple linear reduction of the limit relative to the width of the bottleneck. But, of course, more elaborate user-supplied functions could be applied instead. If, however, the tool passes through the bottleneck for the second time and, thus, the area on the other side of the edge of the medial axis has already been machined, then we stay with the user-specified limit $\theta_{\max}$. (Depending on the specific machining situation, one may want to reduce $\theta_{\max}$ in this case, too.)

4 RESULTS OBTAINED

We implemented our algorithm in C++. As already stated, Voronoi diagrams and medial axes are computed by means of VRONI/ARCVRONI [6, 9]. A constant spacing of the machining centers rather than a spacing based on the maximum engagement angle allows us to use our implementation to generate paths that mimic the original MATHSM algorithm.

In our experiments, we studied the lengths of the tool paths and the distributions of the engagement angles along the paths. While summing the lengths of the individual straight-line segments and circular arcs suffices to compute the length of a path, assessing the engagement angles along a path requires a higher effort. For the sake of implementational simplicity, we resorted to a discretization and computed the engagement angles for a myriad of densely spaced positions of the tool center along a path. (The engagement angles for different tool positions can be computed similarly to the formula given in Eq. (7).)

Since there is no apparent relation between the constant spacing d of the machining centers and the resulting maximum engagement angle for the MATHSM algorithm, we resorted to a brute-force solution: We varied the value d in tiny increments from small to large (relative to the tool radius and the geometry of a pocket), computed for every value of d the MATHSM path and recorded its maximum engagement angle (and its length). Of course, if different spacing distances resulted in (roughly) the same maximum engagement angle for the MATHSM paths then we recorded the length of the shorter path.

This allowed us to compare our paths to the MATHSM paths such that all paths respect the same maximum engagement angle $\theta_{\max}$. In Fig. 5, sample paths are shown for $\theta_{\max} := 80$. In all our figures of tool paths both the maximum engagement angle and the tool size were chosen reasonably large for the sake of visual clarity; we appreciate the fact that machining hard materials might make it necessary to use smaller values. In the figures, the start chosen for the tool path is depicted by a red tool circle and a red cross; it would be suitable for a spiral-down motion of the tool within a disk whose diameter matches roughly the tool diameter. (Since it is application-dependent how the tool is immersed into the material at the start of a path, our implementation picks the starting points of the paths with little attempt to be particularly smart.) Glancing at these three paths makes it immediately apparent that the two paths generated by our approach are substantially shorter than the (shortest) MATHSM path (shown in Fig. 5d) that respects the same value of $\theta_{\max}$.

It is not surprising that a smaller maximum engagement angle results, in general, in a longer tool path. In Fig. 6 we plot how the length of a tool path for the sample pocket (and tool size) shown in Fig. 5 depends on the maximum engagement angle chosen. We see that the difference in the path lengths decreases if the

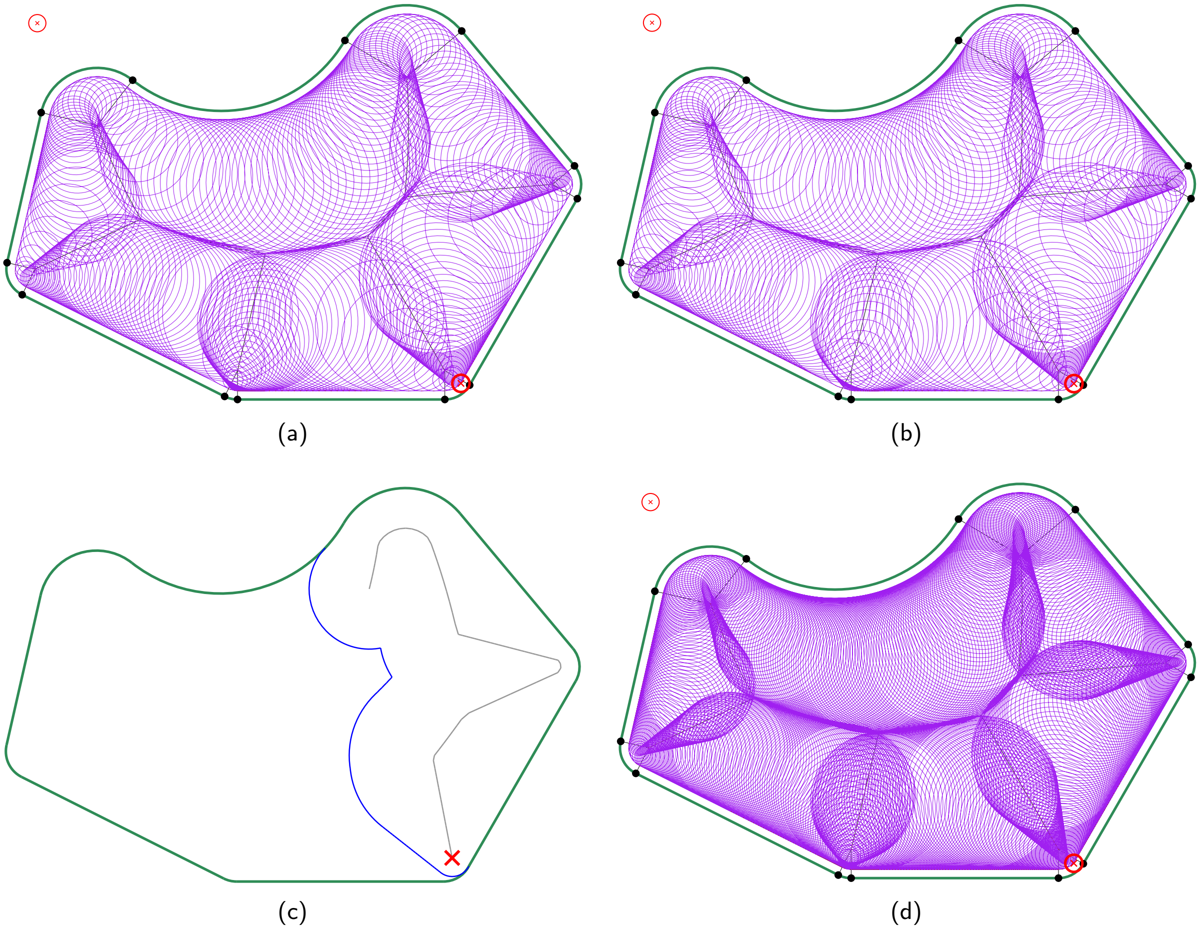


Figure 5: Sample tool paths for $\theta_{\max} := 80$ and the tool shown in the upper-left corners of the figures: The path in (a) was generated by our standard algorithm, while (b) shows the contour-aware path. For a specific point in time during machining, the (blue) contour of the area machined can be seen (c); the gray curve is the "middle" curve between the medial axis and the pocket boundary that links the centers of the machining circles. The path in (d) shows the result for our implementation of the original MATHSM algorithm. The start is the same for all paths.

maximum engagement angle is increased. In particular, the greater $\theta_{\max}$ the less the difference in length between our two variants of the tool path gets: Taking the area already machined into account hardly helps to shorten the tool path for larger values of $\theta_{\max}$. Of course, for $\theta_{\max} := 180$ we get identical lengths for all three paths since the same spacing of the machining centers can be used.

Figure 6 clearly shows that even for very large values of $\theta_{\max} < 180$ the original MATHSM algorithm generates paths that are longer than our paths. To probe this issue further we examined the distribution of the engagement angles along the tool paths. Figure 7 plots the engagement angles for hundreds of consecutive tool positions along the tool paths for our standard method and for the original MATHSM approach, for the setting of Fig. 5. For every position (of the center) of the tool a color-coded point indicates the engagement angle. No engagement angles were assessed during the spiral-down move within the white disk at the start

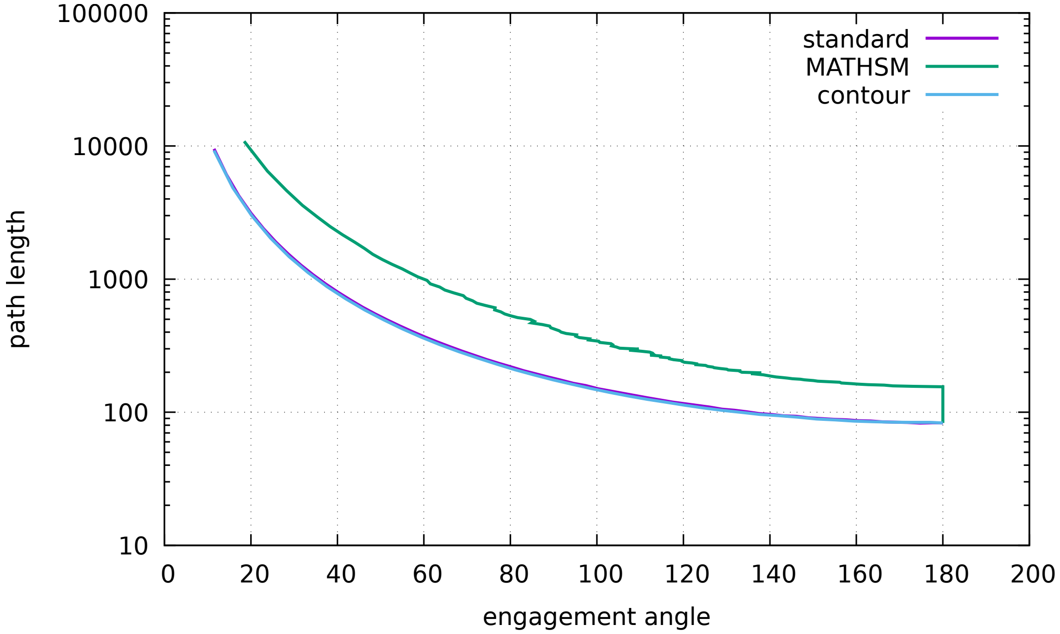
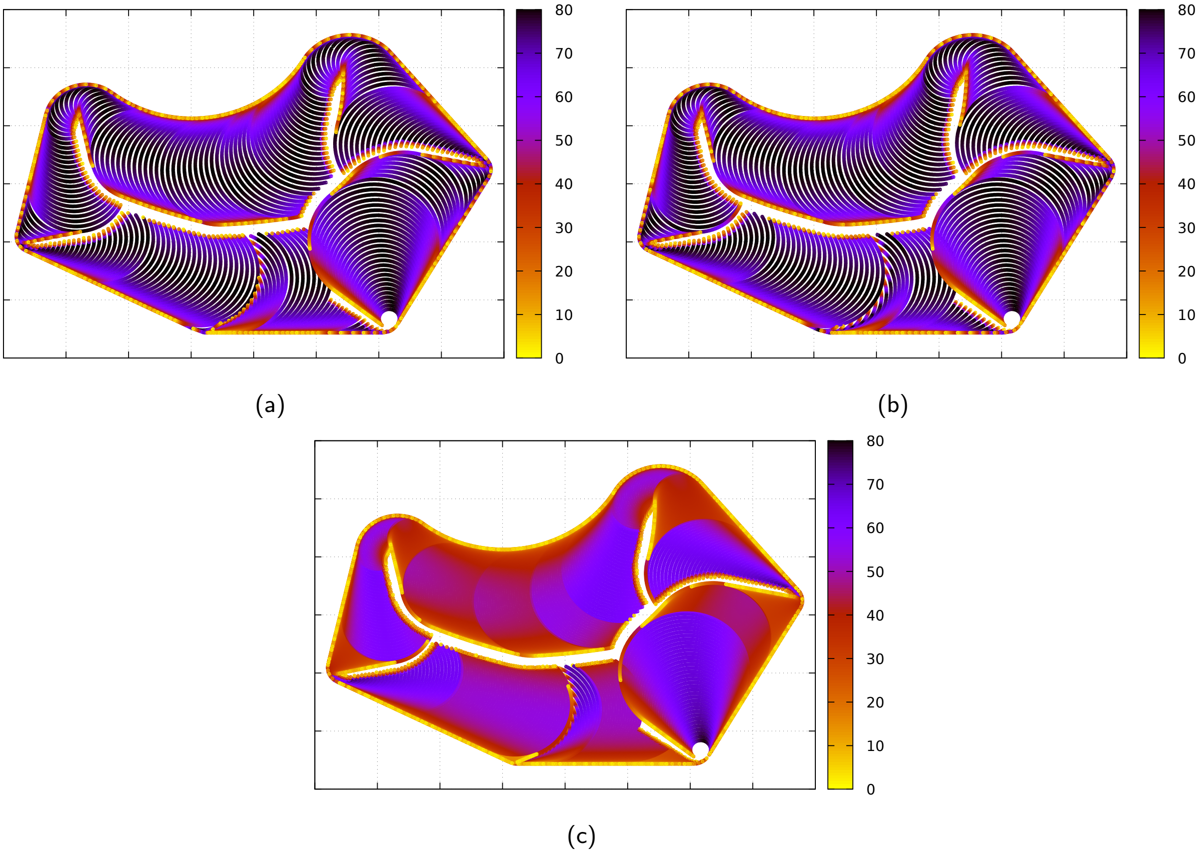


Figure 6: Plot that shows the distribution of the path lengths relative to different engagement angles for the pocket and tool size shown in Fig. 5. We use the labels "standard" for our standard algorithm, "contour" for the improved contour-aware version, and "MATHSM" for the original MATHSM algorithm.

of the tool path. We admit that the color coding is not entirely reliable in the close neighborhood of the edges of the medial axis due to multiple overlaps of the tool disk (which make it difficult to place the correctly colored point "on top" in the plots). Still, the plots make it evident why the original MATHSM paths are significantly longer than our paths: While our paths have engagement angles in the range $70°$ to $80°$ along large portions of the cutting moves, the MATHSM path has angles mostly in the range $30°$ to $60°$. Only around the start of the path and in the very left region and very right region of the pocket the angles reach $80°$. These regions enforce a small value for the spacing of the machining circles. As it can be seen, such a small spacing constitutes a waste for most portions of the path, thus leading to an excessively long path.

The plots in Fig. 7 also help to shed light on the benefits of maintaining the contour of the machined area: Compare the standard path in (a) to the contour-aware path in (b) within the cell in the first row (from bottom to top) and sixth column (from left to right) of the dashed grid. We see that the engagement angles along the contour-aware path quickly approach $80°$ as the tool moves away from the transition elements towards an area already machined at the start of the tool path. For the same portions of the standard path the angles remain mostly in the range $40°$ to $60°$. A similar difference can be spotted in the fifth column at the border between the first and second rows. That is, taking the contour of the area already machined into account allows to space some of the machining centers further apart and, thus, results in a shorter tool path. Indeed, the path shown in Fig. 5b is about 3% shorter than the one in Fig. 5a. We note that maintaining that contour has no disadvantage. Thus, while the savings (in general) are moderate, they come for free.

The run-time of a tool path computation based on our approach is small enough to allow its use in practice. But, of course, it increases as $\theta_{\max}$ is decreased: The smaller $\theta_{\max}$, the more machining circles have to be generated, thus also resulting in more computations of the engagement angle. On a standard desktop PC and for the setting shown in Fig. 5, our code (without I/O and computation of the Voronoi diagram) needs



(a)

(b)

(c)

Figure 7: Plots that show the distribution of the engagement angles along the tool paths for our standard method (a), for the contour-aware method (b), and for the original MATHSM approach (c). The pocket and the tool size are the same as in Fig. 5.

about 100ms when $\theta_{\max} := 20$, about 30ms for $\theta_{\max} := 40$, and about 9ms for $\theta_{\max} := 80$. Increasing $\theta_{\max}$ even further causes the run-time to stay in the range $3\,\text{ms}$ to $6\,\text{ms}$. The additional run-time consumed for computing the Voronoi diagram is negligible: It is less than 1ms for the pocket geometry shown in Fig. 5.

Of course, the results obtained depend on the geometry of the pocket and on the size of the tool. For instance, if the pocket is a simple circular disk then no difference in length will occur between our paths and the MATHSM path if all paths respect the same value of $\theta_{\max}$. (But recall that this would require trying lots of distances to find the appropriate constant spacing of the machining centers for the MATHSM path by trial and error.) On the other hand, just one convex arc with a small radius (relative to the tool radius) will cause the MATHSM path to be longer.

Similarly, the reduction of the length of a tool path achieved by the contour-aware method depends on how much overlap occurs during machining, which in turn depends on the geometry of the pocket and the tool size. And it depends on the start of the tool path, too. We have seen reductions of up to 10% of the length as well as truly marginal improvements. Furthermore, the length of a tool path may also depend on how the tool happens to get to a "critical" region such as a convex arc with a small radius (relative to the radius of the tool). At such regions of the pocket it is a matter of chance whether a few machining circles more or less are needed. (This also explains slight zigzagging in plots of the tool path length, e.g., for the MATHSM paths in Fig. 6.)

Still, overall the results for other pockets, tool sizes and start points of the paths are similar to the results presented for the setting of Fig. 5. Therefore we refrain from including detailed studies for other pockets, but conclude this section with a few additional sample tool paths; see Fig. 8.

5 DISCUSSION AND CONCLUSION

Our generalization of the MATHSM algorithm allows us to generate tool paths such that a user-specified maximum engagement angle $\theta_{\max}$ is not exceeded along the entire path. In particular, there is no need to try different constants for the spacing of the machining centers to find such a path. As this angle is directly related to the force acting on the tool, maintaining and controlling the maximum engagement angle can be expected to increase the tool life and improve the surface quality. Furthermore, our adaptive spacing reduces the length of the tool paths and, thus, cuts down the time spent on machining a pocket. Considering areas that were already machined allows us to adaptively increase the spacing even further, thus resulting in yet another shortening of the tool paths and savings in machining time.

The very nature of the MATHSM construction implies that no milling occurs along about half of the tool path. One may want to save machining time by increasing the feedrate along these portions of the path. Alternatively, one could shorten the tool path by replacing the non-cutting portions of the machining circles with shorter primitives, e.g., with elliptic arcs. In that case our approach to controlling the engagement angle would remain applicable because these non-cutting portions have no impact on the engagement angle. As a matter of principle, also the cutting portions of the tool path could be replaced. However, then the underlying mathematics would become more demanding since computing circle-circle and circle-line intersections would no longer suffice for calculating the engagement angle.

Same as for the original MATHSM paths, it is trivial to convert a tool path computed by our algorithm to C^1 continuity by resorting to an arc-length parameterization of the path. And, of course, C^2 continuity could be achieved by approximating it by a sequence of higher-order primitives. While algorithms are known for approximating curvilinear paths by cubic splines within prescribed tolerances [10], this raises a non-trivial theoretical question: Suppose that the maximum engagement angle along a tool path $\mathcal{TP}$ is $\theta_{\max}$. Given a tolerance $\varepsilon > 0$, what is a suitable approximation threshold $\delta > 0$ such that an approximation of $\mathcal{TP}$ by cubic splines (or similar higher-order primitives) yields a new tool path $\mathcal{TP}'$ such that the maximum engagement angle that occurs when moving along $\mathcal{TP}'$ is at most $\theta_{\max} + \varepsilon$ if $\mathcal{TP}$ and $\mathcal{TP}'$ are within a Hausdorff distance of at most δ? Intuition tells us that any "reasonable" approximation of $\mathcal{TP}$ will result in a small ε if we just pick δ small enough. But what is the precise mathematical relation between ε and δ for some specific type of approximation?

Martin Held, https://orcid.org/0000-0003-0728-7545
Josef Pfeiffer, https://orcid.org/0009-0003-7865-782X

REFERENCES

[1] Bieterman, M.; Sandstrom, D.: A Curvilinear Tool-Path Method for Pocket Machining. ASME J. Manufac. Science Eng., 125(4), 709–715, 2003. http://doi.org/10.1115/1.1596579.

[2] Dumitrache, A.; Borangiu, T.; Dogar, A.: Automatic Generation of Milling Toolpaths with Tool Engagement Control for Complex Part Geometry. IFAC Proc. Volumes, 43, 252–257, 2010. http://doi.org/10.3182/20100701-2-PT-4011.00044.

[3] Elber, G.; Cohen, E.; Drake, S.: MATHSM: Medial Axis Transform toward High Speed Machining of

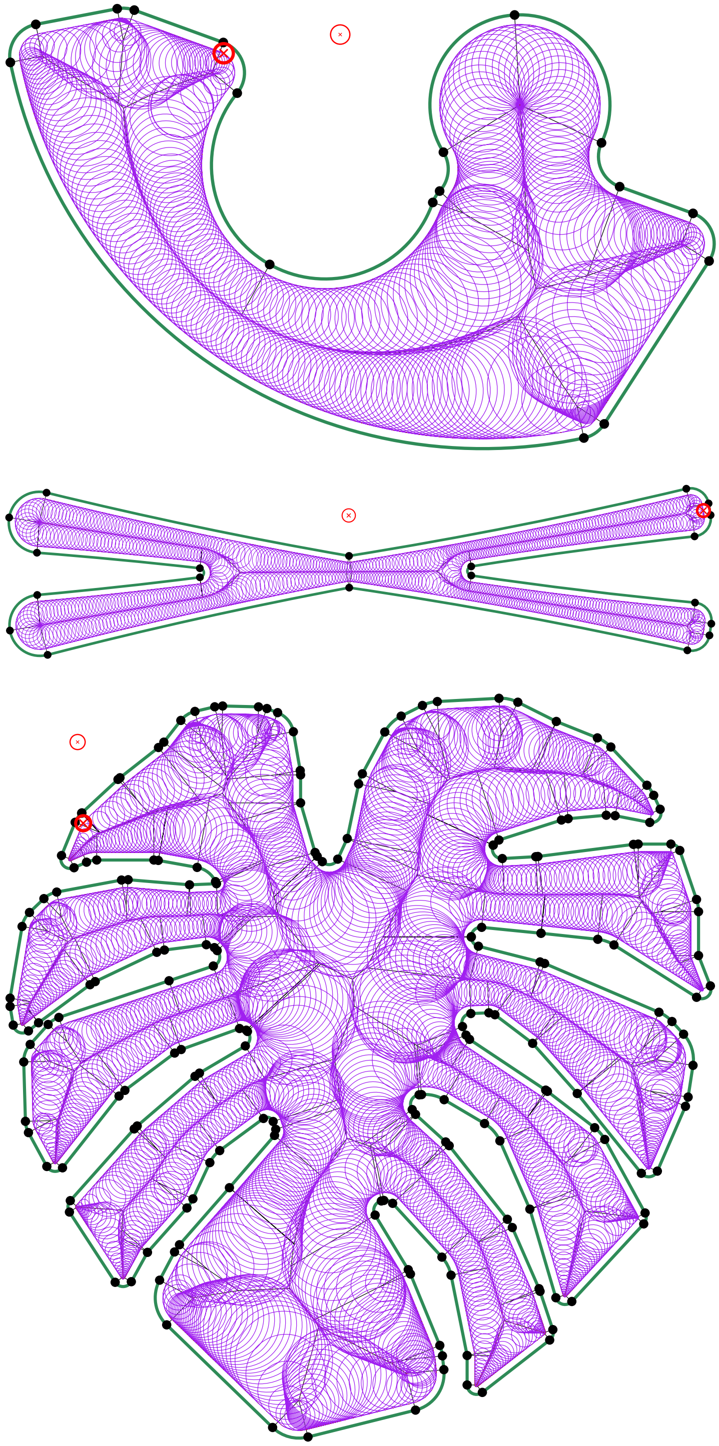


Figure 8: Sample tool paths for $\theta_{\max} := 80$ for another real-world pocket, crossed skis and a leaf of the plant Monstera deliciosa. All paths are contour-aware, i.e., they take the already machined areas into account.

Pockets. Comput. Aided Design, 37(2), 241–250, 2005. http://doi.org/10.1016/j.cad.2004.05.008.

[4] Held, M.: On the Computational Geometry of Pocket Machining, vol. 500 of Lecture Notes Comput. Sci. Springer-Verlag, 1991. ISBN 3-540-54103-9.

[5] Held, M.: A Versatile Geometric Data Structure for Handling Clearance Problems of CAD/CAM and Robotics. In N. Ko; S. Tan, eds., Proc. Internat. Conf. Manufact. Automat., 11–18, 1992.

[6] Held, M.: VRONI: An Engineering Approach to the Reliable and Efficient Computation of Voronoi Diagrams of Points and Line Segments. Comput. Geom. Theory and Appl., 18(2), 95–123, 2001. http://doi.org/10.1016/S0925-7721(01)00003-7.

[7] Held, M.: VRONI and ArcVRONI: Software for and Applications of Voronoi Diagrams in Science and Engineering. In Proc. 8th Int. Symp. Voronoi Diagrams in Science & Engineering, 3–12, 2011. http://doi.org/10.1109/ISVD.2011.9.

[8] Held, M.; de Lorenzo, S.: On the Generation of Spiral-Like Paths Within Planar Shapes. J. Computat. Design Engg., 5(3), 348–357, 2018. http://doi.org/10.1016/j.jcde.2017.11.011.

[9] Held, M.; Huber, S.: Topology-Oriented Incremental Computation of Voronoi Diagrams of Circular Arcs and Straight-Line Segments. Comput. Aided Design, 41(5), 327–338, 2009. http://doi.org/10.1016/j.cad.2008.08.004.

[10] Held, M.; Kaaser, D.: C2 Approximation of Planar Curvilinear Profiles by Cubic B-Splines. Comput. Aided Design & Appl., 11(2), 206–219, 2014. http://doi.org/10.1080/16864360.2014.846092.

[11] Held, M.; Spielberger, C.: A Smooth Spiral Tool Path for High Speed Machining of 2D Pockets. Comput. Aided Design, 41(7), 539–550, 2009. http://doi.org/10.1016/j.cad.2009.04.002.

[12] Held, M.; Spielberger, C.: Improved Spiral High-Speed Machining of Multiply-Connected Pockets. Comput. Aided Design & Appl., 11(3), 346–357, 2014. http://doi.org/10.1080/16864360.2014.863508.

[13] Ibaraki, S.; Ikeda, D.; Yamaji, I.; Kakino, Y.; Nishida, S.: Tool Path Design for Constant Cutting Engagement in End Milling Processes. In Proc. 2004 Spring JSPE Semiannual Meeting, 549–550, 2004.

[14] Jacso, A.; Matyasi, G.; Szalay, T.: The Fast Constant Engagement Offsetting Method for Generating Milling Tool Paths. Int. J. Adv. Manuf. Technology., 103, 4293–4305, 2019. http://doi.org/10.1007/s00170-019-03834-8.

[15] Jacso, A.; Szalay, T.: Optimizing the Numerical Algorithm in Fast Constant Engagement Offsetting Method for Generating 2.5D Milling Tool Paths. Int. J. Adv. Manuf. Technology., 108, 2285–2300, 2020. http://doi.org/10.1007/s00170-020-05452-1.

[16] Kim, H.C.; Lee, S.G.; Yang, M.Y.: An Optimized Contour Parallel Tool Path for 2D Milling with Flat Endmill. Int. J. Adv. Manuf. Technology., 31, 567–573, 2006. http://doi.org/10.1007/s00170-005-0228-1.

[17] Stori, J.; Wright, P.: Constant Engagement Tool-Path Generation for Convex Geometries. J. Manuf. Syst., 19(3), 172–184, 2000. http://doi.org/10.1016/S0278-6125(00)80010-2.

[18] Uddin, M.; Ibaraki, S.; Matsubara, A.; Nishida, S.: Constant Engagement Tool Path Generation to Enhance Machining Accuracy in End Milling. JSME Int. J. Series C, 49(1), 43–49, 2006. http://doi.org/10.1299/jsmec.49.43.

[19] Wang, H.; Jang, P.; Stori, J.: A Metric-Based Approach to 2D Tool-Path Optimization for High-Speed Machining. ASME J. Manufac. Science Eng., 127(1), 33–48, 2005. http://doi.org/10.1115/1.1830492.