www.cadanda.com
Computer-Aided Design
AND APPLICATIONS

Improved Spiral High-Speed Machining of Multiply-Connected Pockets

Martin Held[1] and Christian Spielberger[2]

Universität Salzburg, FB Computerwissenschaften, A-5020 Salzburg, Austria
[1]held@cosy.sbg.ac.at
[2]cspiel@cosy.sbg.ac.at

ABSTRACT

We introduce a geometric heuristic for decomposing an arbitrarily complex pocket with or without islands into simpler sub-pockets that are better suited for efficient spiral high-speed machining. Within every sub-pocket we apply a second heuristic for selecting a "good" start point of the spiral tool path. Several machining parameters such as the step-over distance and the engagement angle are considered as measures and indicators for a good tool path. Our heuristics are based on the Voronoi diagram of the pocket contours, and we can handle contours consisting of straight-line segments and circular arcs. The resulting new algorithm for high-speed spiral pocket machining was implemented and tested successfully on real-world data. Our experiments provide strong evidence that our heuristics reduce the total length of the tool path, while also reducing the variation of the curvature and of the engagement angle over the entire tool path, and decreasing the ratio between the maximum and the minimum step-over distance.

Keywords: pocketing, spiral path, HSM, pocket decomposition, tool path optimization

1. INTRODUCTION

1.1. High-Speed Pocket Machining

Pocket machining is a manufacturing operation which arises frequently in the CAD/CAM industry. In this operation, a numerically controlled milling machine (NC machine) is used to cut a cavity out of a solid material a layer at a time by moving the tool along a path. Most cutting moves are constrained to planes parallel to two coordinate axes, e.g., parallel to the *xy*-plane, thus rendering the computation of tool paths for pocketing a geometric problem in two dimensions (2D): The tool can be regarded as a circular disk, and computing a tool path means finding a path in 2D such that the swept volume of the disk covers the 2D shape that models the pocket.

An inwards (or interior) offset with offset distance ρ of a pocket is the set of curves traced out by the center of a disk with radius ρ that rolls along the boundary of the pocket in its interior. Similarly, an outwards (or exterior) offset is traced out if a disk with radius rolls along the boundary of the pocket in its exterior. A conventional contour-parallel tool path consists of portions of inwards offset curves that are linked together in order to form one path. (See, e.g., [9].) These tool paths tend to contain many sharp corners which are "inherited" from the pocket's boundary.

When using high-speed machining (HSM), the spindle rotation speed and the feed rate are higher than for conventional milling in order to minimize the manufacturing time without a decrease of the part quality [10]. The high rotation speeds and feed rates of HSM impose new constraints on the tool path: A sharp corner requires the tool to slow down, change its direction and accelerate again until the desired maximum speed is regained again. Also, when cutting hard material, a rapidly changing tool load may result in an increased tool wear. Since sharp corners or, more generally, points of high curvature of the tool path often also result in a rapid change of the tool load it is obvious that they should be avoided for HSM tool paths.

1.2. Prior Work

Interestingly, little is known on the generation of high-quality HSM tool paths. One possibility to avoid sharp corners in a contour-parallel tool path is to replace the corners with arcs, loops, or other curves (e.g., [11,17,18]). Wang et al. [14] start with

Computer-Aided Design & Applications, 11(3), 2013, 346–357, http://dx.doi.org/10.1080/16864360.2014.863508

Taylor & Francis
Taylor & Francis Group

a contour-parallel path, place control points onto the offset curves, and slightly move these points to improve the value of a self-defined metric. Machining along maximal circles (e.g., [3]) provides tool paths with minimum curvature but with a high variation of the tool load: Roughly 50% of the tool path corresponds to non-cutting moves.

Bieterman and Sandstrom [1] realized that a spiral-out tool path is most suitable for general-purpose HSM. They use the solution of an elliptic partial differential equation (PDE) boundary value problem to morph a point - the center point of the pocket - to the boundary of the pocket. Level curves are combined to a smooth spiral tool path by means of a radial interpolation around the center point. They also suggest to subdivide general pockets prior to the generation of the spiral tool-path, but give no details. While one may assume that their method works nicely for simple "nearly convex" pockets, our own tests indicated that it is difficult to apply to more general pockets.

Yao and Joneja [15] propose to use a combination of Archimedean and clothoid spirals, but it remains unclear how material left over close to the pocket boundary is removed efficiently and whether their approach is applicable to complex pockets. Chuang and Yang [2] also employ a spiral tool path, based on a Laplace (re-)parameterization. Since the isoparametrics obtained via the Laplace PDE tend to be unevenly distributed if the pocket has bottlenecks, fairly non-steady tool loads may occur along their tool paths. Also, sharp corners of the pocket boundary are clearly reflected by their tool paths.

In [8] we presented an algorithm for generating spiral-out tool paths for HSM. Our algorithm ensures that the step-over distance is bounded above by a user-specified value. The tool path generation tries to maximize the curvature radius and to hold the step-over as stable as possible. The tool paths of [8] are G^1-continuous, but can be boosted to C^2-continuity by means of an error-controlled approximation by uniform cubic B-spline curves [7].

1.3. Results Achieved

In this paper we present geometric optimization heuristics that can be applied as pre-processing heuristics to improve the suitability of spiral tools paths for HSM. While our heuristics have been built upon our own prior work [8], they are rather general in nature and, thus, are meaningful also for other (future) approaches as long as the spiral paths generated are similar to those of [8], no matter whether up-cut or down-cut milling is used. That is, our optimization heuristics could be used as a higher-order machining strategy, while the actual spiral tool path is computed by an algorithm other than the one in [8].

It is obvious that the shape of the pocket has a great influence on the suitability of a spiral tool path for HSM: If the pocket is long but very narrow or contains bottlenecks then one spiral path may be less than ideal, and it might be better to find a decomposition of the pocket and to machine the resulting parts by separate spirals. It is also obvious that the choice of the starting point of the spiral ought to be chosen deliberately, leaving room for a second improvement.

Therefore, we introduce heuristics for decomposing a complex pocket bounded by straight-line segments and circular arcs into sub-pockets that are better suited for HSM, and for choosing a start point of the spiral tool path within such a sub-pocket. See Fig. 1 for a sample pocket, machined with one spiral path and two different starting points, and alternatively split up into two sub-pockets that can be machined more efficiently. The tool path in Fig. 1(b) reduces the path length to 64% of the length of the path depicted in Fig. 1(a) and the number of laps from 30 to 20. The sum of the tool path lengths in Fig. 1(c) is only 63% of the length of the tool path in Fig. 1(b). We emphasize that all paths respect the same maximum step-over distance. Our pocket decomposition can also be applied if the pocket is multiply-connected, i.e., if the pocket is bounded by one outer contour and one or more island contours.

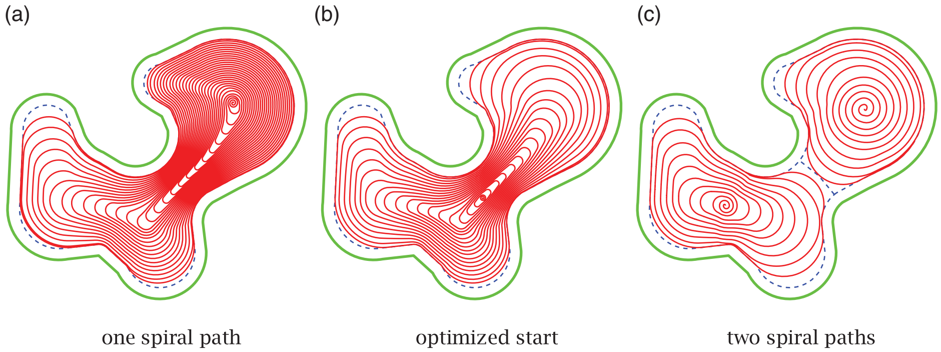


Fig. 1: Three possible tool paths for machining a non-convex pocket. (a) Machining with one spiral path [8]. (b) Machining with one spiral path which has an optimized start point. (c) Adequate splitting and machining with two spiral paths. In each sub-figure the dashed curve represents the outermost offset curve $\partial\mathcal{S}$ which is traversed by the tool at the end of the machining.

Whether or not there is freedom to choose the start point of $\mathcal{P}$ deliberately depends on the specific machining application. E.g., for the 2.5D machining of a mold the machining is carried out within successive parallel layers. If the tool cannot plunge directly into the material then one has to pre-drill holes. In this case there is an obvious trade-off between the time spent on drilling multiple holes, one for each layer, and the time saved by optimized start points of the spiral paths within the individual layers. Similar considerations apply to replacing one spiral path by several paths, after a decomposition of the pocket into sub-pockets that can be machined more efficiently.

In this paper we do not argue for or against the applicability of either optimization to some specific machining application. Rather, in Sec. 3 we focus on the geometric reasoning that allows to carry out these optimizations. The experiments with our optimization heuristics are discussed in Sec. 4. The results reported show clearly that, in general, our heuristics improve several machining parameters while hardly any parameter changes for the worse. Thus, if applicable to a specific machining situation, a spiral tool path is likely to improve if our optimization heuristics are employed.

2. SPIRAL TOOL PATH GENERATION

Let $\mathcal{S}$ denote the planar shape inside of a pocket within which the center of the tool's cross section is allowed to travel. We assume $\mathcal{S}$ to be connected and simply-connected. That is, $\mathcal{S}$ has no islands and is bounded by one simple curve $\partial\mathcal{S}$. (A curve is "simple" if it has no self-intersections.) Typically, $\partial\mathcal{S}$ will be obtained by inwards offsetting of a pocket by a distance that is equal to the radius of the tool T. We assume that $\partial\mathcal{S}$ consists of straight-line segments and circular arcs. The spiral tool path, $\mathcal{P}$, starts somewhere inside $\mathcal{S}$ at a point r and loops counterclockwise outwards until the boundary $\partial\mathcal{S}$ is met. (Of course, our heuristics can also be adapted to clockwise spiraling.) The machining of the original pocket can be finished by machining along $\partial\mathcal{S}$. In the following we will often regard $\mathcal{P}$ as partitioned into *laps*, where one lap corresponds to one full move of the tool around r.

2.1. Tool Engagement and Cutting Force

The cutting force is the force that has to be applied for the cutting. It is influenced by a variety of technical parameters, such as the feed rate, the rotation speed of the spindle, chip geometry, tool diameter and by the engagement angle. The engagement angle α at a particular point in time is the angle which spans the part of the tool surface that performs the cutting, see Fig. 2.

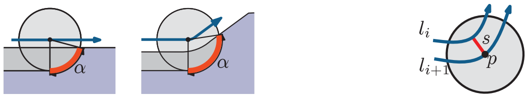


Fig. 2: The engagement angle α at a straight and curved segment of the tool path, and the step-over s. The disk depicts the tool.

The engagement angle has a direct influence on the cutting force [13]: The greater the engagement angle is, the greater is the cutting force, because more material (volume) has to be removed per time. To prevent the tool from chipping it is favorable to keep the cutting force as stable as possible. Since keeping the engagement angle constant over the whole tool path seems impossible for general (non-convex) pockets, our goal is to avoid sharp increases or decreases of the engagement angle, and we attempt to keep its variation as small as possible along the entire tool path.

Controlling the engagement angle mathematically is difficult because it is influenced not only by the current location and direction of the tool movement, but also by the previous movement of the tool. Thus, we do not attempt a global optimization that minimizes the maximum or average engagement angle. Rather, we focus on the *step-over* distance which is easier to control mathematically. The step-over at a point p on lap l_{i+1} of $\mathcal{P}$ is the shortest (Euclidean) distance to the next inner lap l_i, see Fig. 2. We denote by $s_{\max}$ the maximum step-over and by $s_{\min}$ the minimum step-over that occurs along $\mathcal{P}$, and take the ratio $s_{\max}/s_{\min}$ as *step-over variation*. We will demonstrate experimentally that a low step-over variation tends to imply a low variation of the engagement angle.

2.2. The Medial Axis Tree

As in [8] we use the Voronoi diagram and the medial axis of $\partial\mathcal{S}$ as algorithmic tool for our tool path optimization. Very roughly, the Voronoi diagram of $\partial\mathcal{S}$ (inside $\mathcal{S}$) partitions $\mathcal{S}$ into cells ("Voronoi cells") such that each cell is defined by a "site" - straight-line segment, circular arc or (reflex) vertex of $\partial\mathcal{S}$ - and consists of all points of $\mathcal{S}$ closer to that site than to any other site. The medial axis is a subset of the Voronoi diagram of $\partial\mathcal{S}$ and it can be derived easily from the Voronoi diagram by removing all Voronoi edges incident at reflex vertices of $\partial\mathcal{S}$. The red structure in Fig. 3(a) depicts the medial axis of $\partial\mathcal{S}$ for the sample pocket shown in Fig. 1(a); together with the blue arcs it forms its Voronoi diagram (within $\mathcal{S}$).

Both the Voronoi diagram and the medial axis of $\partial\mathcal{S}$ consist of $O(n)$ edges and nodes if $\partial\mathcal{S}$ is formed by n line segments and circular arcs; they can be computed in in $O(n \log n)$ time and $O(n)$ space, both in the worst case [16] and in the expected case [6]. Reference is given to [5] for an up-to-date survey on the computation and application of Voronoi diagrams of straight-line segments and circular arcs.

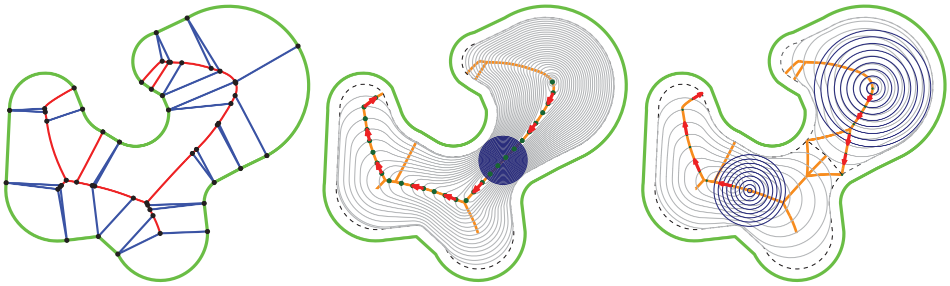


Fig. 3: (a) Voronoi diagram and medial axis of the sample pocket of Fig. 1(a), (b) large step-over variation and (c) small step-over variation.

It is well-known that the medial axis of $\partial\mathcal{S}$ forms a tree, $\mathcal{T}$. As usual, a *path* in $\mathcal{T}$ between two nodes u and v of $\mathcal{T}$ is a sequence of edges of $\mathcal{T}$ that share common nodes such that all edges are distinct. If a path shall start (or end) at a point p which lies on an edge e of $\mathcal{T}$ but which is no node of $\mathcal{T}$ then we imagine that a temporary dummy node is positioned at p such that e is temporarily split into two sub-edges.

We take the arc length of the curve that models a Voronoi edge e as length (or weight) of e. Following standard terminology for edge-weighted graphs, the *length* of a path P in $\mathcal{T}$ is given by the sum of the lengths of its edges. The *distance* $d_{\mathcal{T}}(u,v)$ of two nodes u and v of $\mathcal{T}$ is given by the length of the path in $\mathcal{T}$ between u and v. (Recall that every such path is unique since $\mathcal{T}$ forms a tree.)

We call a straight-line segment that connects a point p within $\mathcal{S}$ with one of its closest points on $\partial\mathcal{S}$ a *clearance line*. By the definition of the medial axis, if p lies on $\mathcal{T}$ then there exist at least two different clearance lines at p (with equal length). The (boundary) *clearance* of p on $\mathcal{T}$, denoted by $clr(p)$, is given by the length of a clearance line of p.

Suppose that we are given a point r in $\mathcal{S}$ which is to form the start point of the spiral tool path $\mathcal{P}$. If r lies on an edge e of $\mathcal{T}$ but does not coincide with a Voronoi node then we split e at r and introduce a new (degree-two) Voronoi node at r. If r lies in the interior of the Voronoi cell of a site s of $\partial\mathcal{S}$ then we extend the clearance line through r towards $\mathcal{T}$ until it intersects the boundary of the Voronoi cell of s. The start and end points of this straight-line segment and r itself become new Voronoi nodes, thus adding three Voronoi nodes and two Voronoi edges. In any case, we can modify $\mathcal{T}$ (and $\partial\mathcal{S}$) such that the point r becomes a Voronoi node.

We now regard $\mathcal{T}$ as a tree rooted at r and call it *medial axis tree*. As usual, a node v of $\mathcal{T}$ is called *descendant* of a node u, with $u \neq v$, if the path from the root to v contains u. If that path consists only of the edge (u,v) then v is a child of u. The node u and all its descendants, together with the appropriate edges of $\mathcal{T}$, form the *subtree* of $\mathcal{T}$ that is rooted at u. A *leaf* is a node without descendants, and a node v is a *descendant leaf* of a node u if v is a leaf of $\mathcal{T}$ and a descendant of u.

We define the (Euclidean) *height* $h_{\mathcal{T},r}(u)$ of a node u of $\mathcal{T}$ rooted at r as follows: If u is a leaf then $h_{\mathcal{T},r}(u) := clr(u)$. Otherwise,

$$h_{\mathcal{T},r}(u) := \max\{d_{\mathcal{T}}(u,v) + clr(v) : \\ v \text{ is descendant leaf of } u\}.$$

We emphasize that the Euclidean height of a node depends on the choice of the root r of $\mathcal{T}$, since the set of descendants changes if r changes.

3. OPTIMIZATION HEURISTICS

3.1. Finding a Good Root of a Spiral Tool Path

Suppose that we have the freedom to choose the start point r of a spiral tool path $\mathcal{P}$ within $\mathcal{S}$. Intuition tells us that it makes no sense to place r close to $\partial\mathcal{S}$, since in this case the laps would become rather dense between r and its closest point on $\partial\mathcal{S}$. Rather, r should be on or close to the medial axis of $\partial\mathcal{S}$. But where on $\mathcal{T}$ should we choose r?

Let $s^\star$ be the maximum step-over specified by the user. A point p_i on lap i around r is at a distance of at most $s^\star$ from a point p_{i-1} on lap $i-1$, which is at a distance of at most $s^\star$ from a point p_{i-2} on lap $i-2$, and so on. Thus, by repeated application of the triangle inequality, we conclude that the center of the tool is at most at a distance $s^\star \cdot i$ from r during the i-th lap. (If each point of $\mathcal{P}$ were always precisely at distance $s^\star$ from the next inner lap then $\mathcal{P}$ would form an involute of a circle.)

Now consider a rectangularly shaped pocket $\mathcal{S}$ and suppose that its length l is much larger than its width w. The medial axis of $\mathcal{S}$ is given by a long central line segment c of length $l-w$ and two pairs of short line segments attached to the start and end of c. Suppose that we place r on c, at a distance x from the short side of $\mathcal{S}$. The upper bound on the distance that the tool center can be away from r during the i-th lap implies that we need approximately

$$\left\lceil \max\left\{\frac{x}{s^\star}, \frac{l-x}{s^\star}\right\}\right\rceil$$

laps to machine that pocket such that the step-over does never exceed $s^\star$. Since $s^\star$ is constant and the ceiling function $\lceil . \rceil$ is monotonically increasing, the number of laps is minimized if $x = l/2$. That is, for a rectangular pocket we may expect a close-to-minimum number of laps if the root is chosen "in the middle" of $\mathcal{T}$, where "middle" needs to be defined in terms of distance along $\mathcal{T}$ rather than by a purely combinatorial measure (such as the number of nodes of $\mathcal{T}$).

We formalize this idea as follows: We call the root r of the medial axis tree $\mathcal{T}$ *height-balanced* if $h_{\mathcal{T},r}(r) = d_{\mathcal{T}(r,u)} + clr(u) = d_{\mathcal{T}(r,v)} + clr(v)$ for two leaves u and v that belong to disjoint subtrees of $\mathcal{T}$ rooted at r. (Recall that we allow to split an edge e of $\mathcal{T}$ by placing the root r in the relative interior of e.) A path from r to a leaf u with $h_{\mathcal{T},r}(r) = d_{\mathcal{T}(r,u)} + clr(u)$ is called a *longest branch* of $\mathcal{T}$.

Let r be a height-balanced root of $\mathcal{T}$. We now prove that

$$h_{\mathcal{T},r}(r') < h_{\mathcal{T},r}(r) < h_{\mathcal{T},r'}(r') \text{ for all nodes } r \neq r' \text{ of } \mathcal{T}. \tag{1}$$

Let u, v be two leaves of $\mathcal{T}$ rooted at r that define longest branches of $\mathcal{T}$ within disjoint subtrees of $\mathcal{T}$. Hence, $h_{\mathcal{T},r}(r) = d_{\mathcal{T}}(r,u) + clr(u) = d_{\mathcal{T}}(r,v) + clr(v)$. Let r' be a node of $\mathcal{T}$ that is different from r. Hence, $d_{\mathcal{T}}(r,r') > 0$.

Case: The node r' is on the path from u to v, i.e., on one of the paths from r to u or from r to v. W.l.o.g., r' is on the path from r to u. Then

$$\begin{aligned} h_{\mathcal{T},r}(r') &= d_{\mathcal{T}}(r',u) + clr(u) \\ &< d_{\mathcal{T}}(r',u) + clr(u) + d_{\mathcal{T}}(r,r') \\ &= d_{\mathcal{T}}(r,u) + clr(u) = h_{\mathcal{T},r}(r), \end{aligned}$$

and

$$\begin{aligned} h_{\mathcal{T},r'}(r') &= d_{\mathcal{T}}(r',v) + clr(v) \\ &= d_{\mathcal{T}}(r,r') + d_{\mathcal{T}}(r,v) + clr(v) \\ &= d_{\mathcal{T}}(r,r') + h_{\mathcal{T},r}(r) > h_{\mathcal{T},r}(r). \end{aligned}$$

Case: The node r' does not belong to the path from u to v. Hence, r' is in a different subtree of $\mathcal{T}$ than both u and v, and $h_{\mathcal{T},r}(r')$ is not given by the distance of r' to either u or v. We get

$$h_{\mathcal{T},r}(r') < h_{\mathcal{T},r}(r') + d_{\mathcal{T}}(r,r') \leq h_{\mathcal{T},r}(r),$$

and

$$h_{\mathcal{T},r}(r) < h_{\mathcal{T},r}(r) + d_{\mathcal{T}}(r,r') \leq h_{\mathcal{T},r'}(r).$$

Thus, the inequalities (1) are correct. This allows us to conclude that the height-balanced root of $\mathcal{T}$ is uniquely determined.

So, how can we compute the height-balanced root of $\mathcal{T}$? Structural induction on $\mathcal{T}$ immediately reveals the following recursive formula for $h_{\mathcal{T},r}(u)$ for every node u of $\mathcal{T}$ which is not a leaf if $\mathcal{T}$ is rooted at r:

$$h_{\mathcal{T},r}(u) = \max\{h_{\mathcal{T},r}(v) + d_{\mathcal{T}}(u,v) : v \text{ is child of } u\}. \tag{2}$$

We now declare an arbitrary non-leaf node of $\mathcal{T}$ to be the root r of $\mathcal{T}$. Equation (2) tells us that a simple recursive procedure suffices to compute the height of every node of $\mathcal{T}$ rooted at r in $O(n)$ time. In particular, the recursion yields

$$h_{\mathcal{T},r}(r) = \max\{h_{\mathcal{T},r}(v) + d_{\mathcal{T}}(r,v) : v \text{ is child of } r\}.$$

By comparing $\delta(v) := h_{\mathcal{T},r}(v) + d_{\mathcal{T}}(r,v)$ for all children v of r to $h_{\mathcal{T},r}(r)$ we can check whether r is already height-balanced. If r is not height-balanced then we pick the unique child r' of r such that $\delta(v)$ is maximum for r' among all children v of r. The node r' lies on the paths from r to all leaves u of $\mathcal{T}$ for which $h_{\mathcal{T},r}(r) = d_{\mathcal{T}}(r,u) + clr(u)$.

Which heights would we obtain if we had rooted $\mathcal{T}$ at r' rather than r? We get

$$h_{\mathcal{T},r'}(r') = \max\{h_{\mathcal{T},r}(r'), h_{\mathcal{T},r}(r) + d_{\mathcal{T}}(r,r')\}.$$

If $h_{\mathcal{T},r'}(r') < h_{\mathcal{T},r}(r)$ then the height-balanced root of $\mathcal{T}$ lies in the subtree $\mathcal{T}'$ of $\mathcal{T}$ rooted at r which contains r'. In this case we declare r' as new root. Note that $h_{\mathcal{T},r'}(u) = h_{\mathcal{T},r}(u)$ for all nodes u of $\mathcal{T}'$ other than r'. Hence, we do not need to recompute the height values and can apply the same scheme to r' instead of r, with $\mathcal{T}$ now being rooted at r'.

Since there is only a finite number of nodes in $\mathcal{T}$, this scheme terminates once either r' is a height-balanced root of $\mathcal{T}$ or $h_{\mathcal{T},r'}(r') \geq h_{\mathcal{T},r}(r)$. In the latter case we know that we have gone too far, and that the height-balanced root $r^\star$ of $\mathcal{T}$ lies on the Voronoi edge (r, r') and fulfills the following equation:

$$h_{\mathcal{T},r}(r) + d_{\mathcal{T}}(r, r^\star) = h_{\mathcal{T},r'}(r') + d_{\mathcal{T}}(r', r^\star).$$

We conclude that the height-balanced root $r^\star$ of $\mathcal{T}$ can be found in time linear in the number of Voronoi edges, which in turn means linear in the number of boundary segments of $\mathcal{S}$, i.e., in $O(n)$ time.

We emphasize that we do not claim (in strict mathematical terms) the uniqueness or optimality of the height-balanced root as start point of a spiral tool path with a minimum number of laps. First of all, the ceiling function $\lceil . \rceil$ is a step function, which suggests that the number of laps might not change if the start point is changed slightly. Hence, it would be inappropriate to claim a uniqueness of the "optimum" start point. Second, our algorithm measures distance along the (curved) edges of $\mathcal{T}$ while the step-over distance at a point p on the intersection of lap $i+1$ with $\mathcal{T}$ is given by the distance from p to a point q on lap i. This point q need not lie on $\mathcal{T}$, though. Even if q lies on $\mathcal{T}$ then the length of the curved arc between p and q forms only an upper bound on the distance between p and q. We note, however, that the latter problem becomes the more negligible the smaller the maximum step-over $s^\star$ is relative to the size of the pocket. Hence, the fact that HSM tends to use rather small step-over distances is in our favor.

From a practical point of view it is necessary to discuss how the length of an edge of the Voronoi diagram can be computed. The theory of Voronoi diagrams tells us that every Voronoi edge is a portion of a conic, i.e., a straight-line segment, parabolic arc, hyperbolic arc, or elliptic arc. The length of a line segment and a parabolic arc can be computed analytically. The arc length of a portion of an ellipse or hyperbola is given by an incomplete elliptic integral, which can be evaluated numerically. Alternatively, and simpler for an actual implementation, one may consider straight-line approximations and compute the length of a Voronoi edge up to a specified precision. Since our geometric optimizations are heuristics without claim of a strict optimality, a fairly moderate precision will be sufficient in practice.

3.2. Decomposition of Complex Pockets

The time needed for machining a pocket decomposed into several sub-pockets includes the individual machining times, times for retraction moves and, possibly, times for tool changes. Often the feed rate is adapted to the tool path curvature or to the step-over. This makes the accurate estimation of the machining time for a (sub-)pocket rather difficult. On the other hand, it is known that a pocket can be machined more efficiently if the step-over variation of the tool path is kept low. In addition, the tool load is kept more stable, which is an important criterion for HSM. Another beneficial effect of a pocket decomposition is that the total length of the tool path can be reduced, too. (Splitting a pocket into two sub-pockets will, in general, not reduce the total number of laps, though.)

Hence, as in Sec. 3.1, we focus on general geometric aspects of the optimization of the pocket decomposition: We try to find a pocket decomposition which allows to optimize, among other parameters, the step-over variation. Once again, we use several simplifications and approximations and, therefore, do not claim strict optimality.

We start by examining how $\mathcal{S}$ can be partitioned. For a point p on $\mathcal{T}$, we call a pair of two different clearance lines of p a *split curve* at p. Note that a Voronoi node of degree n defines $n(n-1)/2$ different split curves, and that each split curve does indeed split $\mathcal{S}$ into two parts. We call a maximal connected portion of $\mathcal{S}$ that is bounded by parts of $\partial\mathcal{S}$ (or by $\partial\mathcal{S}$ completely) and zero or more split curves a *sub-pocket* of $\mathcal{S}$. A particular partition $\mathcal{D}$ of $\mathcal{S}$ by split curves is called a *pocket decomposition*.

Figure 1 shows the tool paths for a sample pocket and for its decomposition. Finishing is done by machining along the dashed lines that consist of the offset curve $\partial\mathcal{S}$ and the split curve between the parts. Both parts can be machined with a smaller step-over variation, and the total length of the tool path is reduced. We exclude the final machining along $\partial\mathcal{S}$ from the mathematical modeling (and our experimental tests) since at the end point of a spiral tool path both the step-over and the engagement angle are zero, thus rendering the computation of their minima for a spiral tool path meaningless.

Our main goal is to decompose $\mathcal{S}$ such that the step-over variation is minimized. But there are also other criteria to consider for this optimization: A split curve that forces a sharp corner of a sub-pocket should be avoided because sharp corners tend to lead to points of high curvature in the tool path. Also, if a pocket is machined by machining separate parts along individual tool paths then retraction moves (or other non-cutting moves) between the sub-pockets are unavoidable. It is obvious that the number of those moves and, thus, the number of sub-pockets generated by the decomposition, cannot grow unboundedly. Thus, we formulate the following three criteria that a pocket decomposition should meet as good as possible.

(C1) The pocket decomposition should minimize the maximum of the step over variations within the sub-pockets.
(C2) The number of sub-pockets should be kept low.
(C3) Sharp corners at the middle vertices of the split curves should be avoided.

Note that a split curve will always be perpendicular to $\partial\mathcal{S}$ and, therefore, we only need to worry about the *split angle* at the point of $\mathcal{T}$ that defines the split curve. (We take the angle at one side of the split curve and check how much it deviates from π.)

The main remaining hurdle on the road to a optimized pocket decomposition is the determination of the step-over variation within some sub-pocket $\mathcal{S}'$ of $\mathcal{S}$: The step-over variation within $\mathcal{S}'$ is only known once a tool path has been computed for $\mathcal{S}'$. But we cannot afford the computational efforts of computing spiral tool paths within $\mathcal{S}'$. After all, different combinations of split curves will induce different sub-pockets of $\mathcal{S}$, and it would be far too time-consuming to compute tool paths for all the resulting sub-pockets.

Hence, we replace the step-over variation of an unknown tool path by a parameter that can be obtained by inspecting only $\mathcal{T}$, without a need to compute actual tool paths. After determining the height-balanced root of $\mathcal{S}$, we replace $s_{\max}$ by the maximum distance $d_{\max}$ between neighboring laps along a longest branch of $\mathcal{T}$. Similarly, we replace $s_{\min}$ by the minimum distance $d_{\min}$ between neighboring laps. If we assume that an algorithm that generates a spiral tool path will attempt to keep the distances between neighboring laps locally as uniform as possible then one needs to know only the start point r of the tool path and the (user-specified) maximum step-over in order to be able to extract both $d_{\max}$ and $d_{\min}$ from $\mathcal{T}$. For instance, $d_{\max}$ can be obtained by determining the height $h_{\mathcal{T},r}(r)$ of the root r of $\mathcal{T}$, and setting $d_{\max} := h_{\mathcal{T},r}(r)/m$, with $m := \lceil h_{\mathcal{T},r}(r)s^{\star}\rceil$ being the estimated number of laps necessary to machine $\mathcal{S}$. We denote the ratio $d_{\max}/d_{\min}$ of the estimated step-over variation of a hypothetical spiral tool path within a sub-pocket $\mathcal{S}'$ of $\mathcal{S}$ by $SOV(\mathcal{S}')$.

Figure 3(b) shows a pocket with a large estimated step-over variation, while the step-over variations of Fig. 3(c) are considerably smaller. In both figures, $d_{\max}$ is visualized by the red arrows, while the difference in the radii of the concentric blue circles visualizes $d_{\min}$. In terms of our prior work [8], the ratio $d_{\max}/d_{\min}$ can be interpreted as the maximum impulse speed divided by the minimum M-disk growth rate; we refer to [8] for further details on how to compute $d_{\max}$ and $d_{\min}$.

We are now ready to formulate the problem of finding a good pocket decomposition as an objective function that involves penalty coefficients. We define the objective function o that evaluates a pocket decomposition $\mathcal{D}$ as

$$o(\mathcal{D}) := \max_{\mathcal{S}'\in\mathcal{D}} SOV(\mathcal{S}') + \mu(|\mathcal{D}| - 1) + \frac{\nu}{|\mathcal{D}| - 1} \sum_{\beta\in\text{splitangles}} \left|\frac{\pi}{\beta} - 1\right|, \qquad (3)$$

where $|\mathcal{D}|$ denotes the number of sub-pockets of $\mathcal{S}$ that are contained in $\mathcal{D}$. The first summand of $o(.)$ represents the estimated step-over variation which we want to minimize. The second summand is the penalty for the number of sub-pockets, with the parameter $\mu \geq 0$ being the penalty coefficient for criterion (C2). The third summand is the penalty for the deviation of the split angles (of the split curves used for partitioning $\mathcal{S}$ into $\mathcal{D}$) from π, with $\nu \geq 0$ being the penalty coefficient for criterion (C3).

It remains to discuss how to choose a suitable set of split curves. Since there are $k!/i!(k-i)! = O(k^i)$ many ways to choose i split curves out of a set of k split curves, it is obvious that we cannot afford to try all possible combinations of up to i split curves out of a large set of k split curves; such an exhaustive search would require $O(k^i)$ evaluations of the objective functions. Hence, generating candidates for the split curves by naïvely putting lots of sample points on all edges of $\mathcal{T}$ is no option in practice.

The major question for reducing the processing time is "where are the split curves located that result in a minimization of the objective function?" And can we possibly detect the split curves on the fly, as the optimization proceeds, rather than attempting to fix them prior to the optimization?

Experiments with our code indicated that there is a fair chance that a simple recursive scheme will achieve a near-optimal solution for the objective function (3). The idea is to split $\mathcal{S}$ into two sub-pockets that are obvious candidates for being machined separately.

We determine a point p on $\mathcal{T}$ for which our analysis of $\mathcal{T}$ predicts a step-over distance equal to $d_{\min}$ to occur along two clearance lines of p. (Likely, it will turn out that p is the middle point of a bottleneck of $\partial\mathcal{S}$.) Let $\mathcal{D} := \{\mathcal{S}_1, \mathcal{S}_2\}$ be the decomposition of $\mathcal{S}$ obtained by using that pair of clearance lines as a split curve.

In order to detect an improvement that occurs in only one sub-pocket of $\mathcal{S}$ we replace the term $\max\mathcal{S}' \in \mathcal{D}SOV(\mathcal{S}')$ by the term $1/2(SOV(\mathcal{S}_1) + SOV(\mathcal{S}_2))$, i.e., we consider the average of (our estimates of) the maximum step-over variations within $\mathcal{S}_1$ and $\mathcal{S}_2$. Hence, for a decomposition $\mathcal{D} = \{\mathcal{S}_1, \mathcal{S}_2\}$ we get

$$o_{rec}(\{\mathcal{S}_1, \mathcal{S}_2\}) := \frac{1}{2}(SOV(\mathcal{S}_1) + SOV(\mathcal{S}_2)) + \mu + \nu \left|\frac{\pi}{\beta} - 1\right| \qquad (4)$$

as new objective function for the recursive decomposition, where β is the split angle of the split line (at either side of the split line). Of course, we set $o(\{S\}) := o_{rec}(\{\mathcal{S}\}) := SOV(\mathcal{S})$.

If $o_{rec}(\{\mathcal{S}_1, \mathcal{S}_2\}) < o_{rec}(\{\mathcal{S}\})$ then the decomposition of $\mathcal{S}$ into $\mathcal{S}_1$ and $\mathcal{S}_2$ is accepted, and we proceed recursively within $\mathcal{S}_1$ and $\mathcal{S}_2$. The recursion stops if either the number of sub-pockets exceeds some user-specified bound or if the reduction of the step-over variation does not outweigh the penalty imposed on the increase of the number of sub-pockets. (Hence, if no upper bound on the number of sub-pockets is specified then choosing $\mu > 0$ is important in order to guarantee a termination of the recursion.)

3.3. Handling Pockets with Islands

The basic geometric tool for determining a good root of a spiral tool path and for the pocket decomposition is the medial axis $\mathcal{T}$ of $\partial\mathcal{S}$. Both of our heuristics are only applicable if the medial axis is indeed a geometric tree. However, it is obvious that the medial axis of a pocket with islands contains cycles and is no longer a tree.

Suppose that $\mathcal{S}$ is a multiply-connected pocket that has k islands. The individual closed curves that form $\partial\mathcal{S}$ are called *contours* and we denote them by $C_0, C_1, \ldots, C_k$, with C_0 being the outermost contour that contains all other contours in its interior. Note that islands are not allowed to be nested. This restriction is of no practical relevance since an island nested inside an island would correspond to an area that is to be machined, but since it is disconnected from the rest it would have to be machined on its own, anyway.

We now explain how to insert bridge edges for linking the islands contours with each other or with C_0, thus breaking up the cycles. We call a point p on the Voronoi diagram of $\partial\mathcal{S}$ within $\mathcal{S}$ a *bridge point* if

- p has a locally minimum clearance, and if
- the two clearance lines starting at p end on different contours of $\partial\mathcal{S}$.

The two clearance lines of a bridge point are called *bridge*. One can prove easily that the two line segments of a bridge lie on the same supporting line. Hence, twice the clearance of a bridge point equals the length of the bridge, i.e., the width of a bottleneck of $\mathcal{S}$. Figure 4 shows a pocket with islands (green contours) together with the bridges (red and blue lines).

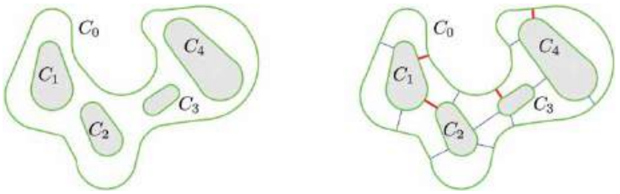


Fig. 4: A pocket with four islands and bridges.

Let n again denote the number of straight-line segments and circular arcs of $\partial\mathcal{S}$. The theory of Voronoi diagrams tells us that we get at most $O(n)$ many bridge points, and that all bridge points can be determined easily in $O(n)$ time, provided that the Voronoi diagram of $\partial\mathcal{S}$ is known. (This Voronoi diagram can be computed in $O(n \log n)$ time, see [6,16].)

For every bridge we consider a quadruple (i, j, δ, p), where i and j are the indices of the contours linked by the bridge with bridge point p and length δ. Experiments with our pocket decomposition suggest that narrow bottlenecks of $\mathcal{S}$ are likely places where a decomposition will occur. Sorting these quadruples lexicographically allows to keep only the shortest bridge between each pair of contours. (This sort takes $O(n \log n)$ time.)

In order to choose a set of "good" bridges that correspond to narrow bottlenecks we define an edge-weighted graph $\mathcal{G}$ as follows: The $k+1$ nodes of $\mathcal{G}$ represent the outermost contour C_0 and the k island contours $C_1, \ldots, C_k$ of $\partial\mathcal{S}$. If there exists a bridge between contours C_i and C_j then we insert an edge between the nodes of $\mathcal{G}$ that correspond to C_i and C_j, with the length of the bridge taken as weight of the edge.

Obviously, the graph $\mathcal{G}$ is connected and planar. Since it has $O(n)$ nodes and edges, we can run a standard algorithm to compute a minimum spanning tree on $\mathcal{G}$ in $O(n \log n)$ time. (E.g., we can use Prim's algorithm [12].) That is, we can determine a set of k bridges such that each island is linked (possibly via a sequence of bridges) with C_0 and such that the sum of the lengths of the bridges is minimized. (These bridges are depicted by red lines in Fig. 4.)

By traversing each of these k bridges twice, in both directions, we can convert $\partial\mathcal{S}$ into one curve C that contains C_0 and all island contours. Technically speaking, the curve C is not simple, but the heuristics outlined in Sec. 3.1–3.2 remain applicable. Since all individual steps of the bridge finding run in $O(n \log n)$ time, a pocket with islands can be transformed into a pocket without islands in $O(n \log n)$ time.

4. EXPERIMENTAL EVALUATION

We implemented our heuristics in C++, based on the Voronoi code VRONI/ArcVRONI [4,6]. Our implementation was used for an extensive series of tests which are summarized in this section. The actual spiral tool paths (relative to specific settings of variables for our heuristics) were computed as described in our prior work [8].

Let $\mathbb{D}$ be a domain (to be specified later) that represents the range of permissible values for a variable which influences the computation of the spiral tool path $\mathcal{P}$, and let $d \in \mathbb{D}$ be a specific value of this variable. We will regard $\mathcal{P}$ as a curve parameterized over the time interval $[0,1]$. Table 1 lists the quality parameters, defined as functions over $\mathbb{D}$, which we examined in our analysis. We denote the maximum step-over allowed by the user by $s^\star$ and the diameter of the (cross-section of the) tool by ϕ_{tool}.

$s_m : \mathbb{D} \to [0,1]$	with $s_m(d) := s_{\min}/\phi_{\text{tool}}$, where $s_{\min}$ is the minimum step-over on the first m laps of $\mathcal{P}$.
$s_M : \mathbb{D} \to [0,1]$	with $s_M(d) := s_{\max}/\phi_{\text{tool}}$, where $s_{\max}$ is the maximum step-over along $\mathcal{P}$.
$V_s : \mathbb{D} \to \mathbb{R}^+$	with $V_s(d) := s_M(d)/s_m(d) = s_{\max}/s_{\min}$ as the step-over variation.
$l : \mathbb{D} \to \mathbb{R}^+$	is the length of $\mathcal{P}$.
$\alpha_m : \mathbb{D} \to [0,\pi]$	is the minimum engagement angle on the first m laps of $\mathcal{P}$.
$\alpha_M : \mathbb{D} \to [0,\pi]$	is the maximum engagement angle on $\mathcal{P}$.
$\alpha_D : \mathbb{D} \to [0,\pi]$	with $\alpha_D(d) := \alpha_M(d) - \alpha_m(d)$.
$\dot{\alpha}_M : \mathbb{D} \to \mathbb{R}^+$	with $\dot{\alpha}_M(d) :=$ $\sup\left\{\lvert \frac{d}{dt}\alpha(t)\rvert : t \in (0,1)\right\}$, and $\alpha(t)$ is the engagement angle at time t of $\mathcal{P}$. This is the maximum absolute value of the first derivative of α.
$\kappa_M : \mathbb{D} \to \mathbb{R}^+$	is the maximum curvature along α.

Tab. 1: Quality parameters of the tool path analysis.

4.1. Impact of Height-Balanced Root

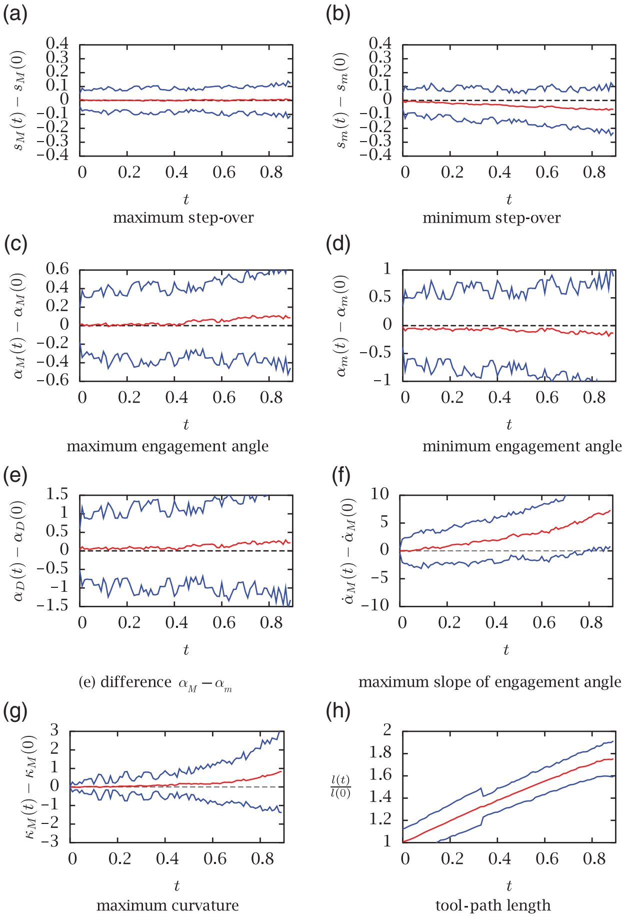
We started our experiments with an extensive series of tests to investigate the impact of the start point of a spiral tool path on the quality parameters detailed in Table 1: For a set of 290 pockets we examined how these quality parameters change when the start point of $\mathcal{P}$ is moved along a longest branch away from the height-balanced root $r^\star$ of $\mathcal{T}$. The measure of how far some start point r of $\mathcal{P}$ is away from $r^\star$ is specified by

$$t = \frac{d_{\mathcal{T}}(r^\star, r)}{h_{\mathcal{T},r^\star}(r^\star)}.$$

Hence, t can range between 0 and 1. We get sample positions for r along a longest branch of $\mathcal{T}$ by incrementing t in steps of 0.01, starting with $t = 0$, i.e., with the height-balanced root $r^\star$. Thus, the domain $\mathbb{D}$ is given by $\mathbb{D} = \{0, 0.01, 0.02, \ldots, 0.99, 1\}$. The nearer t gets to 1 the closer r is to the boundary $\partial\mathcal{S}$ of $\mathcal{S}$.

Figure 5 shows the results of this experiment. We put $t = d_{\mathcal{T}}(r^\star, r)/h_{\mathcal{T},r^\star}(r^\star)$ as a measure of the relative distance of r from $r^\star$ on the x-axis. The red line is the average over the set of pockets, and the blue curves illustrate the standard deviation. In each plot the area between the two blue curves represents the 2σ-environment around μ (in red), where σ is the standard deviation from the mean μ. If $\mathcal{P}$ is computed such that it uses the minimum number of laps to ensure that $s_{\max} \leq s^\star$, where $s^\star$ is the maximum step-over allowed by the user, then it is not surprising that the maximum step-over s_M in Fig. 5(a) is mostly independent of the choice of t. (We set $s^\star := 0.8 \cdot \phi_{\text{tool}}$.) The minimum step-over s_m does depend on t: As Fig. 5(b), shows it tends do decrease as r is moved away from $r^\star$. Thus, the step-over variation V_s increases as t increases.

Similarly, the maximum engagement angle increases and the minimum engagement angle decreases as t is increased, see Fig. 5(c)-5(d). This leads to an increase of α_D in Fig. 5(e), which is a measure for the range of the cutting forces arising during machining. Hence, we get experimental evidence that a minimization of the step-over variation implies a minimization of the variation of the engagement angle. Except for a few outliers the maximum slope of the engagement angle $\dot{\alpha}_M$ and the maximum curvature κ_M also increase as t increases, see Fig. 5(f)-5(g). Most strikingly, Fig. 5(h) shows clearly that the length l of $\mathcal{P}$ increases rapidly as t increases: Since at least one longest branch of $\mathcal{T}$ grows as r is moved away from $r^\star$, more and more laps are necessary to accomplish a tool path that obeys $s_{\max} \leq s^\star$. Summarizing, all quality parameters of a spiral tool path deteriorate if the root r is moved away from the height-balanced root $r^\star$.



maximum step-over

minimum step-over

maximum engagement angle

minimum engagement angle

(e) difference $\alpha_M - \alpha_m$

maximum slope of engagement angle

maximum curvature

tool-path length

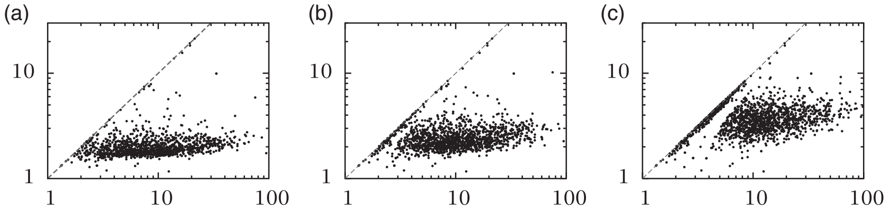
Fig. 5: Plots of the quality parameters of a spiral tool path in dependence on the distance from the height-balanced root $r^\star$.

4.2. Impact of Pocket Decomposition

We started the experimental evaluation of our heuristic for decomposing a pocket by a closer examination of the penalty coefficients used in the objective function (4). Figure 6 shows the reduction of the step-over variation for $\nu = 0.5$ and three different values of μ. In the plots the x-axis corresponds to the step-over variation of the original pocket, while the y-axis plots the average step-over variation for a pocket decomposition. A dot on the line $y = x$, depicted by a blue dashed line, stands for a pocket which was not decomposed. No maximum number of sub-pockets was specified by us. Rather, we used the parameter μ to influence how early the recursion would terminate. Within every sub-pocket we used our heuristic for finding a good root of the spiral tool path. Most pocket decompositions took less than one second on a dual-core PC with a clock rate of 2.7 GHz. (Our code does not benefit from hyper-threading.)

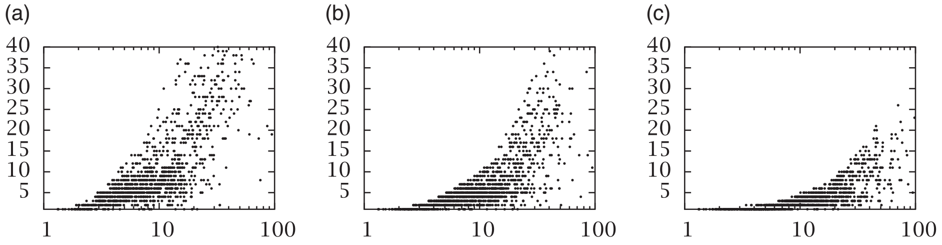
Figure 7 shows the number of sub-pockets in relation to the step-over variation of the original pocket. It comes as no surprise that smaller values of μ lead to a larger number of sub-pockets but also allow a more impressive reduction of the step-over variation.

Determining values of μ and ν that work best for a particular application will likely require a bit of experimentation on the user's side. We suggest to use $\mu := 0.8$ and $\nu := 0.5$ as initial settings for the penalty coefficients. One may increase μ and, e.g., set $\mu := 2$ if one wants to reduce the number of sub-pockets. Figure 8 shows sample pockets decomposed by our pocket decomposition, with the penalty coefficients $\mu = 2$ and $\nu = 0.5$.



$\mu=0.3, v=0.5$ $m=0.8,\ v=0.5$ $m=2.0,\ v=0.5$

Fig. 6: The reduction of the step-over variation achieved by a pocket decomposition.



$\mu=0.3, v=0.5$ $m=0.8,\ v=0.5$ $m=2.0,\ v=0.5$

Fig. 7: The number of sub-pockets in relation to the step-over variation of the original pockets.

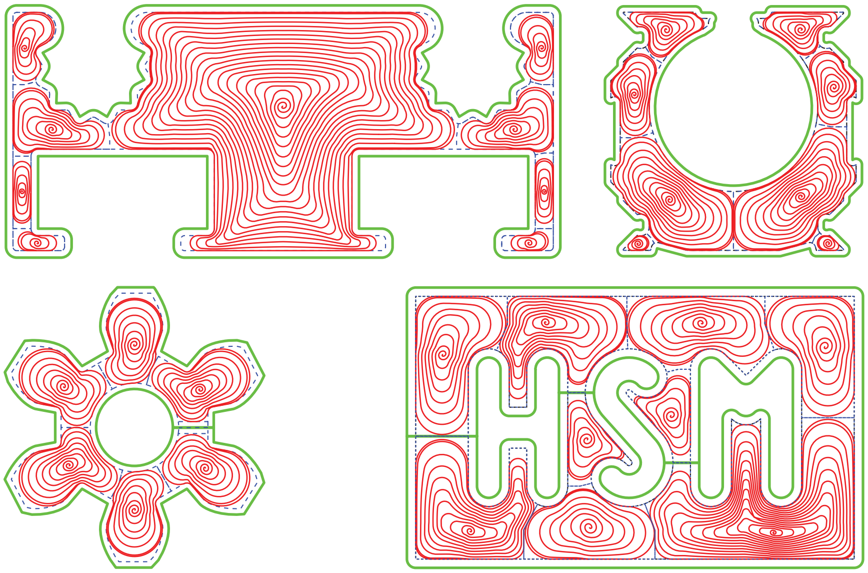


Fig. 8: Examples of our pocket decomposition for the penalty coefficients $\mu = 2$ and $\nu = 0.5$.

We now turn our attention to an experimental evaluation of pocket decompositions with respect to the quality parameters listed in Table 1. We carried out our tests on over one thousand pockets for which a decomposition makes sense. For this set of experiments we set the penalty coefficients of the objective function (4) to $\mu := 0.8$ and $\nu := 0.5$. For every pocket we compared the quality parameters for a spiral path within the undecomposed $\mathcal{S}$ to the maximum (or sum) of the quality parameters of spiral paths within the sub-pockets suggested by our pocket decomposition.

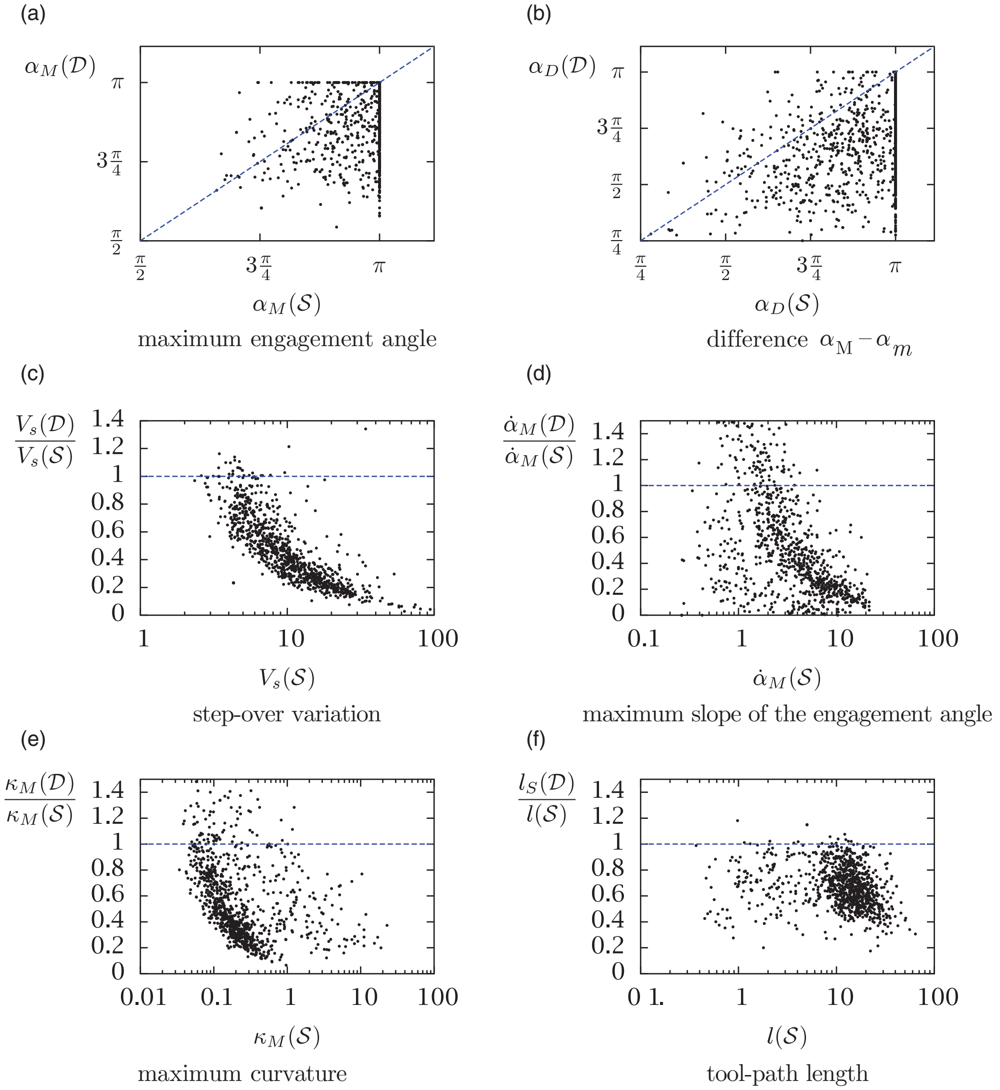
That is, for every pocket $\mathcal{S}$ the domain $\mathcal{D}$ in plots of Fig. 9 is given by $\mathcal{S}$ and its decomposition $\mathcal{D}$. If $\mathcal{D}$ is a decomposition of $\mathcal{S}$, then we compute the quality parameters for $\mathcal{D}$ as follows:

$$s_M(\mathcal{D}) := \max\left\{s_M(\mathcal{S}') : \mathcal{S}' \in \mathcal{D}\right\}$$

$$V_s(\mathcal{D}) := \max\left\{\frac{s_M(\mathcal{S}')}{s_m(\mathcal{S}')} : \mathcal{S}' \in \mathcal{D}\right\}$$

$$\alpha_M(\mathcal{D}) := \max\left\{\alpha_M(\mathcal{S}') : \mathcal{S}' \in \mathcal{D}\right\}$$

$$\alpha_D(\mathcal{D}) := \max\left\{\alpha_M(\mathcal{S}') - \alpha_m(\mathcal{S}') : \mathcal{S}' \in \mathcal{D}\right\}$$



maximum engagement angle

difference $\alpha_{\mathrm{M}} - \alpha_m$

step-over variation

maximum slope of the engagement angle

maximum curvature

tool-path length

Fig. 9: Results for tool paths computed on decomposed pockets. The original pocket is denoted by $\mathcal{S}$, whereas $\mathcal{D}$ constitutes the pocket decomposition determined by our heuristic.

$$\dot\alpha_M(\mathcal{D}) := \max\left\{\dot\alpha_M(\mathcal{S}') : \mathcal{S}' \in \mathcal{D}\right\}$$
$$\kappa_M(\mathcal{D}) := \max\left\{\kappa_M(\mathcal{S}') : \mathcal{S}' \in \mathcal{D}\right\}$$
$$l_S(\mathcal{D}) := \sum_{\mathcal{S}' \in \mathcal{D}} l(\mathcal{S}')$$

Every pocket and its decomposition is represented as a data point in the plots in Fig. 9. On the x-axis we put the parameter value of the spiral tool path for the undecomposed pocket $\mathcal{S}$. Since α_M and α_D have a bounded image domain ($[0, \pi]$), we put the corresponding parameter values for the decomposition directly on the y-axis. A data point below the line $y = x$, depicted by a blue dotted line in the Fig. 9(a)-(b), signals that the pocket decomposition improved this quality parameter. It can be seen that for most pockets the decomposition results in a reduction of α_M and α_D, especially if $\alpha_M(\mathcal{S}) = \pi$, i.e., if the tool is fully immersed during a so-called "slotting cut" somewhere on the tool path. Our experience with pocket decompositions tells us that slotting cuts can often be avoided by a decomposition. However, there are some pockets out of the set of over one thousand pockets for which the maximum engagement angle of $\mathcal{D}$ is higher than for $\mathcal{S}$. This may happen since sometimes the maximum step-over on a tool path within a sub-pocket comes closer to the maximum allowed value $s^\star$. And a higher step-over locally also implies a higher engagement angle.

For the quality parameters with unbounded image domain we put the ratios of the parameter values of the decomposition $\mathcal{D}$ to the parameter value of the original pocket $\mathcal{S}$ on the y-axis, see Figs. 9(c)-(f). A value below 1, illustrated by the blue horizontal line, signals an improvement. As it can be seen, for the vast majority of pockets tested the pocket decomposition results in an improvement of the stepover variation $\mathcal{V}_S$, the maximum slope on the engagement angle $\dot{\alpha}_M$, the maximum curvature κ_M and the tool path length l_S. Typically, the improvement is substantial. There are only relatively few pockets for which a decomposition resulted in a deterioration of at least one of the quality parameters examined.

5. CONCLUSION

Computing a spiral tool path for a general pocket is a non-trivial task if some form of "optimality" of the tool path is sought. It seems impossible to come up with one general-purpose algorithm that will generate the best path(s) for every application. Rather, some application-dependent fine tuning will always be necessary. Still, it is possible to devise general-purpose heuristics that have a wide range of applicability: One can attempt to optimize the start point of a spiral, and one can decompose a complex pocket into subpockets such that the machining is improved. Based on the Voronoi diagram of a pocket we present two geometric heuristics which can be used to carry out such an optimization. Again based on the Voronoi diagram, we extend our heuristics to pockets with islands by computing appropriate bridges to connect the island contours with the outermost contour.

Our experiments provided clear evidence that our heuristics reduce the overall tool-path length and avoid excessive variations of the tool-path curvature, the engagement angle and of the step-over distance. Thus, if applicable to a specific machining situation, a spiral tool path is likely to improve if our optimization heuristics are employed.

REFERENCES

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

[2] Chuang, J.-J.; Yang, D. C. H.: A Laplace-Based Spiral Contouring Method for General Pocket Machining, Int. J. Avd. Manuf. Technology., 34(7-8), 2007, 714–723. doi:10.1007/s00170-006-0648-6.

[3] Elber, G.; Cohen, E.; Drake, S.: MATHSM: Medial Axis Transform Toward High Speed Machining of Pockets, Comput.-Aided Design, 37(2), 2004, 241–250. doi:10.1016/j.cad.2004.05.008.

[4] 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), 2001, 95–123. doi:10.1016/S0925-7721(01)00003-7.

[5] 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, pages 3-12, Qingdao, China, June 2011. doi:10.1109/ISVD.2011.9.

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

[7] Held, M.; Kaaser, D.: C2 Approximation of Planar Curvilinear Profiles by Cubic B-Splines, Computer-Aided Design & Applications, 11(2), 2014, 206–219.

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

[9] Held, M.; Lukács, G.; Andor, L.: Pocket Machining Based on Contour-Parallel Tool Paths Generated by Means of Proximity Maps, Computer-Aided Design, 26(3), 1994, 189–203. doi:10.1016/0010-4485(94)90042-6.

[10] Marinac, D.: Tool Path Strategies For High Speed Machining. Modern Machine Shop, 72(9), 2000, 104–110.

[11] Pateloup, V.; Duc, E.; Ray, P.: Corner Optimization for Pocket Machining, Int. J. Avd. Manuf. Technology., 44(12-13), 2004, 1343–1353. doi:10.1016/j.ijmachtools.2004.04.011.

[12] Prim, R. C.: Shortest Connection Networks and Some Generalizations, Bell System Technical J., 36, 1957, 1389–1401.

[13] Stori, J. A.; Wright, P. K.: Constant Engagement Tool-Path Generation for Convex Geometries, J. Manuf. Syst., 19(3), 2000, 172–183. doi:10.1016/S0278-6125(00)80010-2.

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

[15] Yao, Z.; Joneja, A.: Path Generation for High Speed Machining Using Spiral Curves, Computer- Aided Design & Applications, 4(1), 2007, 191–198. doi:10.3722/cadaps.2007.191-198.

[16] Yap, C. K.: An O(n log n) Algorithm for the Voronoi Diagram of a Set of Simple Curve Segments, Discrete Comput. Geom., 2(4), 1987, 365–393.

[17] Zhao, Z.; Liu, B.; Zhang, M.; Zhou, H.; Yu, S.: Toolpath Optimization for High Speed Milling of Pockets, In Proc. ICIC'09, pages 327–330. IEEE CS, 2009. doi:10.1109/ICIC.2009.90.

[18] Zhao, Z. Y.; Wang, C. Y.; Zhou, H. M.; Qin, Z.: Pocketing Tool-path Optimization for Sharp Corners, J. Materials Proc. Techn., 192-193(0), 2007, 175–180. doi:10.1016/j.jmatprotec.2007.04.096.