On the Generation of Spiral Paths Within Planar Shapes

Martin Held[*] Stefan de Lorenzo[*]

Abstract

We simplify and extend prior work by Held and Spielberger [CAD 2009, CAD&A 2014]: We use a linearization to derive a simple algorithm that computes a spiral path inside of planar shape bounded by straight-line segments and circular arcs. Our spiral paths are continuous and without self-intersection, respect a user-specified maximum step-over distance, start in the interior and end at the boundary of the shape. We also extend this basic algorithm to double spirals that start and end at the boundary.

1 Introduction

Several applications require to cover a planar shape by moving a circular disk along a path. E.g., in machining applications the disk models the cross-section of a tool and the area models a so-called pocket. Similarly, the disk may represent the area covered by a spray nozzle or the area of visibility of a camera device used for aerial surveillance.

Traditional strategies for path generation include zigzag patterns and the use of offset curves to form contour-parallel patterns. Common to these traditional strategies is the fact that the resulting paths contain lots of sharp corners, i.e., abrupt changes of the direction. The higher the speed or the moment of inertia of the moving object represented by the disk is, the more these directional discontinuities become a problem. E.g., for a high speed machining (HSM) application, an abrupt change of direction requires the tool to slow down to near-zero speed, change its direction and then accelerate until the desired maximum speed is reached again.

In order to avoid sharp directional discontinuities, (spiral) paths have been studied. Bieterman and Sandstrom [2] present an approach based on partial differential equations (PDEs) to compute spiral tool paths inside star-shaped pockets. Abrahamsen [1] constructs a polygonal spiral inside a pocket bounded by straight-line segments. Held and Spielberger [4, 5] used the medial axis of the pocket to generate spiral paths for general non-convex pockets with or without islands. The key ingredient of their approach are circles centered on the medial axis whose radii increase as time progresses. Portions of these circles are interpolated and connected by other circular arcs to form a G^1-continuous path.

Since the algorithm by Held and Spielberger [4, 5] is difficult to analyze theoretically and even more difficult to implement reliably, in this work we pick up their overall idea but simplify it significantly: A linearization of the medial axis allows to come up with an algorithm for a raw G^0-continuous spiral path that is easy to implement. (And, indeed, an implementation of this algorithm is already in commercial use at our industrial partner.) The spiral path is continuous, without self-intersection, and respects a user-specified maximum step-over distance. This raw path can then be boosted to G^1-continuity or C^2-continuity by using a (one-sided) approximation by biarcs or cubic B-splines.

As in the work by Held and Spielberger [4, 5], our spiral path starts in the interior of the pocket and ends at its boundary. The simplicity of our approach allows to generalize this scheme and to devise double-spiral paths that start and end at arbitrary points on the boundary. This makes it easier to cover a complex shape by one continuous spiral path by (1) decomposing the shape into simpler sub-areas, (2) computing a (double) spiral within every sub-area, and (3) linking the individual spirals to form one continuous path. While such a double-spiral path is unsuited for machining, it may find use in other applications, such as spray painting, aerial surveillance, or path finding algorithms for rescue missions.

2 Preliminaries

If a disk of radius ρ that moves has to stay within a shape during the entire movement then it is obvious that its center can never get closer to the boundary then ρ, even if this constraint results in some areas of the shape being uncovered. (E.g., at convex corners of the boundary.) The loci of all permissible positions of the center can be obtained as the Minkowski difference of the shape and a disk of radius ρ centered at the origin. We call this set a *pocket*, P. Hence, the area P' to be covered by our moving disk is given by the Minkowski union of P with a disk of radius ρ centered at the origin. It is well-known that (1) the boundary of P also consists of $O(n)$ straight-line segments and circular arcs if the initial shape was bounded by n straight-line segments and circular arcs, and (2) that it can be obtained in $O(n \log n)$ time via

[*] Universität Salzburg, FB Computerwissenschaften, 5020 Salzburg, Austria.

This is an extended abstract of a presentation given at EuroCG 2017. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

Voronoi-based offsetting.

We assume that P is path-connected and simply-connected. If P were disconnected then we would run our algorithm separately for every connected component. If P contains islands (i.e., is multiply-connected) then we convert it to a simply-connected area by introducing bridges [5].

It is natural to break a spiral that winds around a point r for k times into a sequence of k individual portions, where each portion corresponds to one full turn around r. We call such a portion of a spiral a *lap*. Then the *step-over distance* at point p of lap ℓ_{i+1} is the minimum distance from p to the next inner lap ℓ_i. It is obvious that, in general, the step-over distance has to be less than the diameter of the disk that is being moved in order to avoid regions of P' that are not covered. In practice, considerably smaller step-over distances are used, though. For HSM, a good step-over value is a fraction of the diameter that depends on the material of the cutter as well as the workpiece. (It is largely independent of the geometry of the pocket.) E.g., for aluminum or (non-hardened) steel a typical maximum step-over is given by about 15% of the diameter. In any case, it is important that the user can specify a maximum step-over Δ that the spiral path has to keep.

3 The Medial Axis Tree

In order to simplify the algorithm by Held and Spielberger [4] we approximate every edge of the medial axis $\mathcal{MA}(P)$ of the pocket P by a polygonal chain. The vertices of such a polygonal chain are obtained by placing uniformly distributed sample points on the edge such that the maximum length of a segment of the chain is less than a user-supplied or heuristically determined value λ. This process yields the discrete medial axis $\mathcal{MA}'(P)$. We refer to the sample points on $\mathcal{MA}(P)$ and the original nodes of $\mathcal{MA}(P)$ as *vertices* of $\mathcal{MA}'(P)$.

As usual, the *clearance*, $\mathrm{clr}(p)$, of a point p on $\mathcal{MA}'(P)$ is the radius of the largest disk centered at p that fits into P. For every vertex p of $\mathcal{MA}'(P)$ we consider the points $p_1, p_2, \ldots, p_k$ where the clearance disk of p touches the boundary ∂P of P, and construct the clearance line segments $\overline{pp_1}, \overline{pp_2}, \ldots, \overline{pp_k}$. (If p happens to be the center of a circular arc a of ∂P then we select finitely many points on a which are uniformly spaced, with a spacing less than λ.)

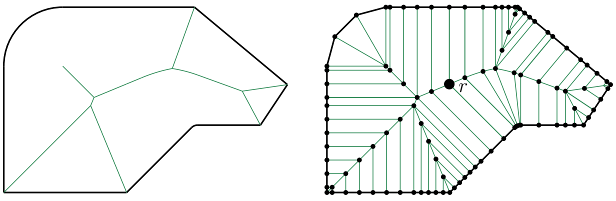
Let $\mathcal{L}$ be the set of all clearance line segments defined by vertices of $\mathcal{MA}'(P)$. We add $\mathcal{L}$ to $\mathcal{MA}'(P)$ and get $\mathcal{MA}''(P) := \mathcal{MA}'(P) \cup \mathcal{L}$.

Both $\mathcal{MA}'(P)$ and $\mathcal{MA}''(P)$ form a tree because P does not contain islands. By choosing one vertex r as root we can turn $\mathcal{MA}''(P)$ into a rooted tree $\mathcal{T}_r(P)$, the *discrete medial axis tree* derived from $\mathcal{MA}''(P)$.

Since all edges of $\mathcal{T}_r(P)$ correspond to line segments, it is easy to compute the (Euclidean) length $d_{\mathcal{T}_r(P)}(p,q)$ of the path between two vertices p, q of $\mathcal{T}_r(P)$. This allows to define the *Euclidean height* of a vertex p of $\mathcal{T}_r(P)$ as

$$h_{\mathcal{T}_r(P)}(p) := \max_q \left(d_{\mathcal{T}_r(P)}(p,q) + \mathrm{clr}(q)\right),$$

where the maximum is taken over all vertices q of the sub-tree of $\mathcal{T}_r(P)$ rooted at p. As in [5] we assume that $\mathcal{T}_r(P)$ is *height-balanced*, that is, that $h_{\mathcal{T}_r(P)}(r)$ is assumed for two different leaves of $\mathcal{T}_r(P)$. (If no such vertex exists then we insert a new vertex within an edge of $\mathcal{T}_r(P)$ in order to achieve a perfect balance.) See Figure 1. (For the sake of visual clarity we show this toy example with a very course discretization and an unrealistically large step-over distance, which result in spiral paths that do not look smooth.)



(a) (b)

Figure 1: (a) Medial axis. (b) Height-balanced discrete medial axis tree $\mathcal{T}_r(P)$.

4 Impulse Propagation

As in [4] we consider an impulse which starts at the root r of the discrete medial axis tree $\mathcal{T}_r(P)$, which is active during the time interval $[0,1]$, and which discharges concurrently at the leaves of $\mathcal{T}_r(P)$. Let e be an edge between the nodes p and q of $\mathcal{T}_r(P)$, where p is the parent node of q inside $\mathcal{T}_r(P)$. The velocity v_e of the impulse at e is given by

$$v_e := \frac{h_{\mathcal{T}_r(P)}(q) + l_e}{1 - t_p},$$

where l_e is the length of e and t_p is the time when the impulse reached p. Since the impulse starts at time $t_r = 0$ at r, we can recursively compute the time when the impulse reaches a specific vertex or any point within an edge of $\mathcal{T}_r(P)$.

As the impulse flows through $\mathcal{T}_r(P)$, it covers an increasing portion of $\mathcal{T}_r(P)$. The point which the impulse reaches at time t on its way from r to some leaf of $\mathcal{T}_r(P)$ is called *corner* at time t. Clearly, there exist at most as many corners as there are leaves in $\mathcal{T}_r(P)$.

By computing all corners at a specific moment in time, and arranging them in the order in which they appear when $\mathcal{T}_r(P)$ is traversed in depth-first manner, it is feasible to construct a closed polygonal chain, a so-called *wavefront* $w(t)$ at time t.

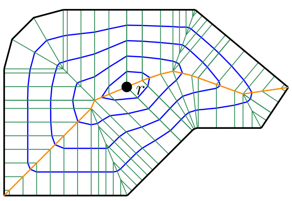
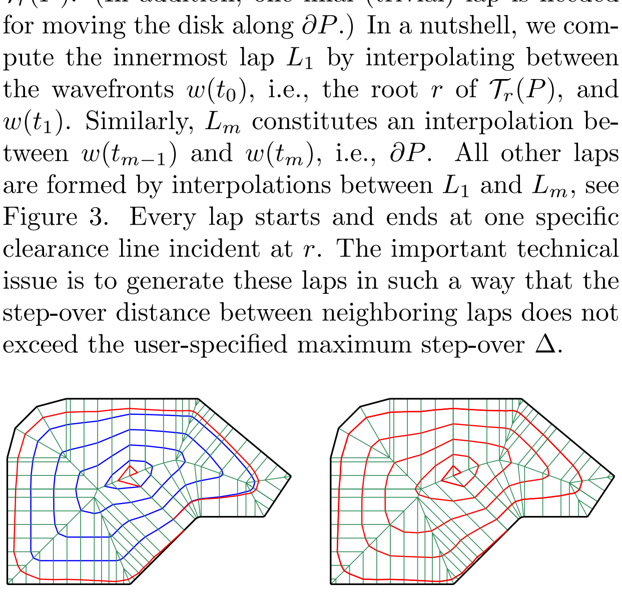


Figure 2: Wavefronts.

In order to guarantee that the user-specified maximum step-over Δ is respected, the spacing of the wavefronts has to be chosen carefully. Roughly, we divide the Euclidean height $h_{\mathcal{T}_r(P)}(r)$ of r by Δ in order to get a lower bound on the number $m+1$ of wavefronts. (In Figure 2, the two longest paths in $\mathcal{T}_r(P)$ between r and leaves of $\mathcal{T}_r(P)$ that correspond to $h_{\mathcal{T}_r(P)}(r)$ are shown in orange.) This gives us a uniform decomposition of time $T := (t_0, t_1, ..., t_m)$, for $m \in \mathbb{N}_0$, with $0 = t_0 < t_1 < ... < t_m = 1$. Then the corners of the wavefront $w(t_i)$ are given by the positions of the impulse at time t_i. (We omit formulas due to lack of space.) Note that this construction implies that the longest paths are split by the wavefronts into sections with length at most Δ. In particular, our construction ensures that the (symmetric) Hausdorff distance between $w(t_i)$ and $w(t_{i+1})$ is at most Δ and, thus, that the maximum step-over Δ is respected. Note that $w(t_m)$ equals ∂P while $w(t_0)$ degenerates to r.

5 One Spiral Path

We now focus on the generation of the actual spiral path, which is fundamentally different to the strategy applied by Held and Spielberger [4]. The spiral path $\mathcal{S}(P, \Delta)$ is made up of m laps $L_1, L_2, \ldots, L_m$. Each of these laps is a polygonal chain whose corners lie on $\mathcal{T}_r(P)$. (In addition, one final (trivial) lap is needed for moving the disk along ∂P.) In a nutshell, we compute the innermost lap L_1 by interpolating between the wavefronts $w(t_0)$, i.e., the root r of $\mathcal{T}_r(P)$, and $w(t_1)$. Similarly, L_m constitutes an interpolation between $w(t_{m-1})$ and $w(t_m)$, i.e., ∂P. All other laps are formed by interpolations between L_1 and L_m, see Figure 3. Every lap starts and ends at one specific clearance line incident at r. The important technical issue is to generate these laps in such a way that the step-over distance between neighboring laps does not exceed the user-specified maximum step-over Δ.



(a)

(b)

Figure 3: (a) First and last lap. (b) All laps.

We start with explaining how L_1 is generated, see Figure 4. Recall that $w(t_0)$ degenerates to r. Suppose that q_0 is the vertex of $w(t_1)$ that is intersected by the clearance line $\overline{rv_0}$, on which all laps start and end. Thus, L_1 starts at r and ends at q_0. If we want to generate counter-clockwise (CCW) laps then we number the vertices of $w(t_1)$ in CCW order, starting at q_0. Now consider some vertex of $w(t_1)$, e.g., q_4 in Figure 4. Let d_4 denote the length of the polygonal chain $q_0 q_1 \ldots q_4$, let d denote the circumference of $w(t_1)$, and let δ_4 denote the distance (along $\mathcal{T}_r(P)$) from q_4 to r. Then a candidate corner c of L_1 is placed on the path from q_4 to r at a distance (along $\mathcal{T}_r(P)$) of

$$\left(1 - \frac{d}{d_4}\right) \cdot \delta_4$$

from q_4. We store c at the corresponding edge of $\mathcal{T}_r(P)$. Note that some vertices of $w(t_1)$ might end up storing candidate corners on the same edge or path towards r.

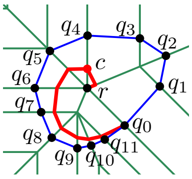


Figure 4: Lap generation.

After setting the weight d to the circumference of ∂P and letting the corners of $w(t_{m-1})$ play the role of r, we obtain candidate corners for L_m in a similar way by moving from the vertices of $w(t_m)$, i.e., ∂P, towards the vertices of $w(t_{m-1})$. If required, we can also let L_m wind around r a bit more than once, and let it end at some point on ∂P other than v_0, by making d larger than the circumference of ∂P.

In order to actually generate L_1 we scan $\mathcal{T}_r(P)$ in a depth-first order, starting at r and moving along $\overline{rv_0}$ as first branch of $\mathcal{T}_r(P)$. The recursive scan stops whenever a candidate corner for L_1 is encountered. This depth-first scan establishes all corners of L_1 in the desired (CCW or CW) order. Then we we start a new depth-first scan towards the leaves of $\mathcal{T}_r(P)$ at every corner q of L_1. The recursion of the depth-first scan from q is stopped whenever we get to a distance $(m-2)\Delta$ from q along $\mathcal{T}_r(P)$. At every such stopping point of the recursion a new candidate corner for L_m is placed. Then another depth-first scan starting at r reveals all corners of L_m by stopping the recursion whenever a candidate corner for L_m is encountered.

We note that this construction guarantees the following distances, where $H(\cdot,\cdot)$ denotes the symmetric Hausdorff distance: $H(r, L_1) \leq \Delta$ and $H(\partial P, L_m) \leq \Delta$ and $H(L_1, L_m) \leq (m-2) \cdot \Delta$.

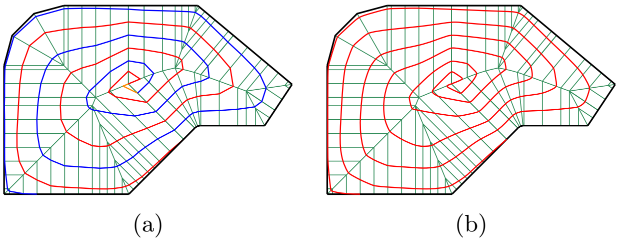
The remaining laps $L_2, \ldots, L_{m-1}$ can be produced similar to the generation of the initial wavefronts if we take the freedom to regard one lap as a special type of wavefront between L_1 and L_m: Again we let an impulse propagate along $\mathcal{T}_r(P)$. However, this modified impulse propagation starts at time $t = 0$ at the corners of L_1, and ends at time $t = 1$ at the corners of L_m. Then, for properly chosen velocities of the impulse on the edges of $\mathcal{T}_r(P)$, the "wavefront" that corresponds to the time $i/m-2$ forms the lap L_{i+1}, for $i \in \{1, 2, \ldots, m-2\}$.

By connecting all laps in the natural way we obtain a polygonal path $\mathcal{S}(P, \Delta)$ inside P. Trivially, $\mathcal{S}(P, \Delta)$ starts at r and ends on ∂P. Furthermore, $\mathcal{S}(P, \Delta)$ is not self-intersecting, because we are gradually moving outwards, starting at r, until we arrive at ∂P. And due to the construction, $\mathcal{S}(P, \Delta)$ respects the maximum step-over Δ.

6 Generating a Double Spiral

We now generalize our approach to a double spiral that starts and ends at the boundary ∂P. As in the case of a single spiral, the user-specified step-over Δ implies a certain minimum number of wavefronts. For the sake of descriptional simplicity, suppose that this number is odd and that we have $2k + 1$ wavefronts $w(t_0), w(t_1), \ldots, w(t_{2k})$, with $w(t_0)$ equal to r and $w(t_{2k})$ equal to ∂P. We use the algorithm of Section 5 to compute one single spiral with maximum step-over 2Δ which starts at r and ends at v_0 on ∂P. Let $L_1, L_3, \ldots, L_{2k-1}$ denote the successive laps of this spiral. Hence, L_1 starts at r and ends at the intersection q of $w(t_2)$ with $\overline{rv_0}$, L_3 starts at q and ends on $w(t_4)$, and so on. In particular, L_{2k-1} ends at v_0 on ∂P.

Let L_{2k+1} be identical to ∂P. At every corner of lap L_i, for $i \in \{1, 3, \ldots, 2k-1\}$ we plant an impulse that moves towards the leaves of $\mathcal{T}_r(P)$, starting on L_i at time $t = 0$ and reaching L_{i+2} at time $t = 1$. Stopping the impulse at time $t = 1/2$ yields the laps $L_2, L_4, \ldots, L_{2k}$, where L_2 starts at q and L_{2k} ends at v_0 on ∂P. As for a single spiral, the positions of the end-points of L_{2k-1} and L_{2k} on ∂P can be adjusted to meet specific needs. In Figure 5(a), the two sequences of laps are shown in red and blue.


(a)

(b)

Figure 5: (a) Double spiral, and (b) its approximation by cubic Bézier curves.

In order to connect the start of L_2 at q with the start of L_1 at r we move from the corners of L_1 towards r for a distance of Δ, thus obtaining corners of a polygonal path that connects L_1 and L_2. In Figure 5(a), this path is shown in orange. This construction ensures that the resulting double spiral is not self-intersecting and respects the maximum step-over Δ.

7 Extensions

As explained in [5], we can decompose a complex (possibly multiply-connected) shape into simpler sub-shapes and then compute spiral paths within these sub-shapes: Two single spirals and several double spirals can be linked to form one continuous multi-spiral path that spirals through the entire shape, see Figure 6. We omit details due to lack of space.

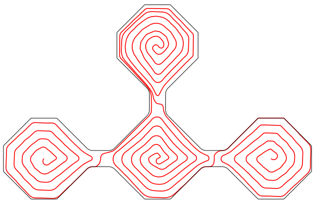


Figure 6: Shape covered by one multi-spiral path.

Furthermore, the polygonal spirals can be approximated by higher-order primitives. For instance, Figure 5(b) shows an approximation by cubic Bézier curves of the double spiral of Figure 5(a). (For the approximation we use the PowerApx package [3].)

References

[1] M. Abrahamsen. Spiral Toolpaths for High-Speed Machining of 2D Pockets With or Without Islands. In *Proc. ASME IDETC/CIE 2015 Conf.*, 2015.

[2] M.B. Bieterman and D.R. Sandstrom. A Curvilinear Tool-Path Method for Pocket Machining. *ASME J. Manufac. Science Eng.*, 125(4):709–715, November 2003.

[3] M. Heimlich and M. Held. Biarc Approximation, Simplification and Smoothing of Polygonal Curves by Means of Voronoi-Based Tolerance Bands. *Internat. J. Comput. Geom. Appl.*, 18(3):221–250, June 2008.

[4] M. Held and C. Spielberger. A Smooth Spiral Tool Path for High Speed Machining of 2D Pockets. *Comput. Aided Design*, 41(7):539–550, July 2009.

[5] M. Held and C. Spielberger. Improved Spiral High-Speed Machining of Multiply-Connected Pockets. *Comput. Aided Design & Appl.*, 11(3):346–357, 2014.