

# C<sup>1</sup> Continuous Toolpath Generation Toward 5-axis High Speed Machining

Gershon Elber<sup>1</sup>, Elaine Cohen<sup>2</sup> and Sam Drake<sup>3</sup>

<sup>1</sup>Technion-Israel Institute of Technology, [gershon@cs.technion.ac.il](mailto:gershon@cs.technion.ac.il)

<sup>2</sup>University of Utah, [cohen@cs.utah.edu](mailto:cohen@cs.utah.edu)

<sup>3</sup>University of Utah, [drake@cs.utah.edu](mailto:drake@cs.utah.edu)

## ABSTRACT

We present a scheme to construct  $C^1$  continuous toolpath for 5-axis pocketing of high speed machining. A continuous toolpath synthesis is proposed, that forms out circular segments of maximally inscribed sizes and linear segments connecting these circular segments.

Extending earlier work that employed the Medial Axis Transform to compute these maximally inscribed circles for 2D planar pockets, this work examines general pockets in  $\mathfrak{R}^3$  represented as NURBS surfaces, and in 5-axis machining setup.

Results are demonstrated by employing the proposed scheme toward the machining of impellers. Possible extensions toward a  $C^2$  continuous toolpath is discussed as well.

**Keywords:** Medial Axis Transform, Surface Composition, Lines and Arcs, Bi-arcs.

## 1. INTRODUCTION

NC machining is one of the most common manufacturing scheme available nowadays [1-3],[10-12],[15],[17-18],[20]. For example, forming die-cavities [2] is one such application. It requires machining at different levels of several parallel cutting planes that define various pockets. A closed shape, denoted the *outline* of the pocket, is prescribed in a pocketing operation, possibly with islands, and a toolpath is generated to machine the interior of the shape, cleaning it to a prescribed depth. In some applications, the bottom of the pocket is assumed planar. A more complex scenario necessitates the support of freeform shaped bottom of pocket, a case common in molding.

Two major approaches exist to generate toolpaths for pocketing operations. The first is based on the intersection of the outline of the pocket with parallel and equally spaced planes. By connecting the parallel adjacent segments that trim the pocket domain into a zig-zag motion, a complete toolpath is constructed [1],[11],[17-18]. This approach is similar to the scan conversion scheme used in computer graphics to render polygonal domains and fill them with pixels [9]. The different scanline segments are then connected into a  $C^0$  continuous zigzag toolpath.

The second approach offsets the outline(s) of the pocket by equally spaced offset distances, and employs these successive offsets to derive a toolpath [3],[10],[15]. Offsets of a  $C^1$  continuous outline can yield  $C^1$  discontinuities in regions where the radius of curvature of the curve is smaller than the offset distance. Then, self-intersections might occur and the trimming of these self-intersections introduces  $C^1$  discontinuities. Hence, in general, this offset-based toolpath is also only  $C^0$  continuous.

Other related works on NC pocket operations have attempted to look at optimization issues. Examples include consideration of successive application of tools with decreasing radii [20], so as to reduce the overall machining time, or alternatively, minimizing the number of retractions in zigzag [1], [17-18]) and offset [3] type toolpaths. The latter has an exponential-expected time complexity due to the ability of reducing the toolpath traversal problem to the Traveling Salesman Problem (TSP).

Some work could be found on NC machining of pockets over freeform geometry. Typically, either a ball end must be used or the tool must be oriented so as to minimize the scallop residue, such as the curvature matched machining [13] approach. All such schemes, again, use the offset schemes over the surface, employ parallel planes' intersections, or alternatively, exploit the surface parametrization and extract isoparametric curves to guide the tools. In this work, we propose another approach, one that employs as-large-as-possible circular controlled motions over the freeform geometry.

High speed machining (HSM) is an NC machining procedure that is becoming more and more common [14],[16],[19]. Unfortunately, both the zigzag toolpath and the offsetting toolpath generation approaches introduce multiple  $C^1$  discontinuities into the toolpath, making it unsuitable for high speed machining applications. Recently, several attempts have been made to alleviate these difficulties. One approach employs a spiral motion that starts at the center of the pocket and spirals out until it reaches the outline [14],[16]. If the spiral starts at the center, the spiral toolpath will present an infinite curvature, at the limit. Furthermore, it is unclear how to initialize the spiraling process as well as bring the tool to its proper cutting depth. While the spiraling scheme can work reasonably well in almost circular pockets, it also becomes arbitrarily inefficient when elongated pockets are considered. Similar difficulties might result when pockets with complex shapes are considered. Other approaches include trochoidal milling [19] where the toolpath is following a trochoidal path. This scheme is used mainly to handle local  $C^1$  discontinuities in the outline of the pocket, converting  $C^1$  discontinuities into small loops that continuously connect the adjacent pieces, thereby forming a  $C^1$  continuous toolpath.

In our work, we propose a different scheme to handle high speed machining of arbitrary  $C^1$  continuous freeform surface pockets, in a 5-axis machining context. The generated toolpath consists of arcs of circular arcs of as-large-as-possible radii, and segments that connect between consecutive arcs. This procedure automatically generated toolpaths that are especially suited for narrow and elongated pocket regions that are difficult to machine even using traditional toolpath generation schemes. Further, the constructed toolpath is guaranteed to be  $C^1$  continuous, maximizing the benefits that can be drawn from the high speed machining schemes. In [8], a scheme to support toolpaths formed out of arcs of as-large-as-possible circular arc was presented over flat bottom pockets. This work extends [8] to support arbitrary  $C^1$  continuous freeform surface pockets.

The rest of this paper is organized as follows. In Section 2, the basic toolpath generation algorithm is considered. In Section 3, several examples are presented, including NC machining tests we have conducted. Finally, in Section 4, we consider possible extensions and conclude.

## 2. THE 5-AXIS HSM TOOLPATH ALGORITHM

Turbine blades and impellers are prime examples where 5-axis machining can serve well by allowing full access to the entire geometry, geometry that could be quite challenging (see Figure 1(a)). Consider regular  $C^1$  continuous parametric surface  $S(u, v) : D \subset \mathcal{R}^2 \rightarrow \mathcal{R}^3$ ,  $D = [0, 1]^2$ , and  $C^1$  vector field  $V(u, v)$  that prescribes and orients the tool at each location on the surface. Further, assume  $S(u, v)$ , as a one pocket of the impeller object, is open in one of its four boundaries. Without loss of generality, let this open boundary be  $u = 0$  or  $S(0, v)$ ; see Figure 1(b).

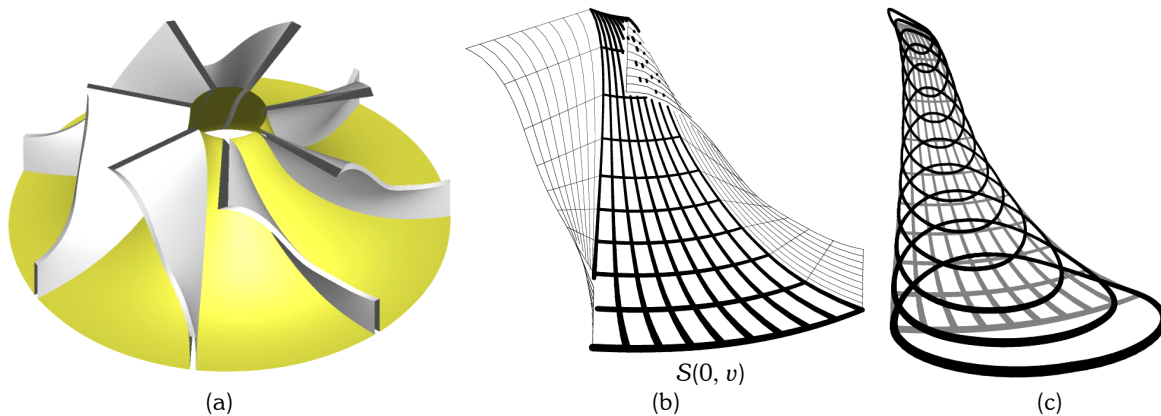


Fig. 1. Given the impeller model in (a), consider one freeform surface pocket  $S$  to be high speed machined in 5-axis context (b). The composition of adjacent circular curves in the parametric space of  $S$ , yields adjacent rounded curves on the surface (c).

Now consider parametric circle  $C_1(t) = (u_1(t), v_1(t))$ , positioned in the parametric space of surface  $S$ ,  $D$ . The composition of  $S(C_1) = S(u_1(t), v_1(t))$  would yield a rounded shape in Euclidean space, over  $S$ , that is  $C^1$ , because  $S$  is  $C^1$ . Position several such circles adjacent to each other in  $D$ , and in Euclidean space, one will get a sequence of rounded curves; see Figure 1(b). By continuously connecting these rounded curves into one long toolpath, one is creating a complete  $C^1$  continuous toolpath that covers the entire domain of the freeform surface's pocket.

In the coming subsections, we will expand on the necessary details of this approach. In Section 2.1, we describe the composition process and possible alternatives to the composition computation. In Section 2.2, these composed shapes will be combined into one long toolpath, discussing the bi-tangency computations and the meaning of offsets over surface  $S$ . Finally, in Section 2.3, the tool orientation is considered, for both cylindrical and tapered tools.

## 2.1 The Composition Computation and Alternatives

Given two polynomial/rational functions,  $f$  and  $g$ , their composition  $f \circ g$  is clearly a polynomial/rational (of multiple degree). Hence, given a Bézier curve embedded in the parametric domain of a Bézier surface, their composition is also a Bézier curve on the surface. This, since Bézier curves and surface are polynomials (or rationals). Direct computation of the composition of Bézier freeforms has also been considered and discussed, for example in [6]. The composition of B-spline and NURBS freeform is not much more complex conceptually as one can always subdivide the piecewise polynomial/rational into its polynomial/rational pieces, compute the composition and reconstruct the final composed shape. More on composition can be found in [4],[6].

A crucial property of the composition operation is that if both  $f$  and  $g$  are  $C^k$  continuous so is  $f \circ g$ . Hence, any shape  $C(t)$  that is  $C^1$  continuous and is composed with a  $C^1$  continuous surface  $S$  would yield a  $C^1$  continuous curve on  $S$ . Specifically, when we compose circles with  $S$ , the rounded curve over  $S$  would be  $C^1$  continuous at every location for which  $S$  is  $C^1$ . The fact that  $S$  is not an isometry (distance preserving, see [5]) means that circles are not preserved under the mapping. Ideally, and since the first fundamental form [5] measures distance changes due to the  $S$  mapping, one should derive the inverse of the first fundamental form function, denoted  $\mathfrak{T}^{-1}$ , as a pre-warping function from  $\mathfrak{R}^2$  to  $\mathfrak{R}^2$  and pre-warp circles  $C_i$  through  $\mathfrak{T}^{-1}$  only to be mapped through  $S$  to yield good approximation of circular arcs on the surface of  $S$ , as  $S(\mathfrak{T}^{-1}(C_i))$ . While  $\mathfrak{T}^{-1}$  is not the inverse function of  $S$  but rather compensates for the non-isometric behavior of  $S$ , reversing the effects of stretch and scale in  $S$ , this process is extremely demanding computationally. Fortunately, we are not really in need of having precise circles on  $S$ . We are only required that:

1. the mapped shape,  $S(C_i)$ , will be  $C^1$  continuous.
2. the mapped shape,  $S(C_i)$ , will be close to its adjacent shape,  $S(C_{i+1})$ , with a bounded distance.

Requirement 1 is already established, provided  $S$  and  $C$  are  $C^1$ . In order to handle Requirement 2, we should establish a bound on the relative distance change between the parametric domain from  $(u_0, v_0)$  to  $(u_1, v_1)$  and the Euclidean space from  $S(u_0, v_0)$  to  $S(u_1, v_1)$ . The first fundamental form again provides the answer here, and bounds on the coefficients of the first fundamental form also bound the distance changes one can expect between the parametric and Euclidean spaces of  $S$ . Therefore, and provided  $S$  is a Bézier or a NURBS surface, one could compute

$$E = \left\langle \frac{\partial S}{\partial u}, \frac{\partial S}{\partial u} \right\rangle, \quad F = \left\langle \frac{\partial S}{\partial u}, \frac{\partial S}{\partial v} \right\rangle, \quad G = \left\langle \frac{\partial S}{\partial v}, \frac{\partial S}{\partial v} \right\rangle,$$

as (piecewise) rational Bézier or NURBS fields and use the extreme coefficients of  $E$ ,  $F$ , and  $G$  as the established distance-change bounds. Let these extreme distance-change bounds be  $(\alpha_{min}, \alpha_{max})$ . That is, an arbitrary small step  $\delta$  in the parameter domain of  $S$ ,  $D$ , would be mapped to a path on  $S$  of an arc-length between  $\delta\alpha_{min}$  and  $\delta\alpha_{max}$ . Denote by  $d(P_1, P_2)$  the Euclidean distance between  $P_1$  and  $P_2$ , and let  $D_h$  denote the Hausdorff distance between two curves  $C_i(t)$  and  $C_{i+1}(r)$ ,

$$D_h(C_i(t), C_{i+1}(r)) = \max \left( \max_t \left( \min_r \left( d(C_i(t), C_{i+1}(r)) \right) \right), \max_r \left( \min_t \left( d(C_i(t), C_{i+1}(r)) \right) \right) \right).$$

Then, the Hausdorff distance between two curves on  $S$ ,  $S(C_i(t))$  and  $S(C_{i+1}(r))$ , is bounded to be between  $D_h(C_i(t), C_{i+1}(r))\alpha_{min}$  and  $D_h(C_i(t), C_{i+1}(r))\alpha_{max}$ . By placing adjacent circular shapes in the parametric space of  $S$  so that their Hausdorff distance is bounded by  $\delta$ , the  $\delta\alpha_{min}$  to  $\delta\alpha_{max}$  bounds will hold. Bounding the Hausdorff distance between two curves in the parametric domain is a simpler problem as the domain is planar now. Herein, however, the problem is even simpler. If the curves,  $C_i$ , are circles, then the Hausdorff distance between two circles is trivial to compute. Furthermore, even if  $C_i$  is not a circle,  $C_i$  and  $C_{i+1}$  are typically going to be the same shape of curves, affinely transformed. Hence, a bound on the Hausdorff distance could be computed from this transform.

Seeing the tool path shown in Figure 1(c), it is clear the tool is going to be in contact with material and actually machining only for about half of the time. Hence, one can, for instance, cut and round the back half of the circular arc, in each circle, making the tool return to its next cutting position faster. Figure 2 presents several possibilities for rounded curves that can be used. In (c), the cut and rounded back half circle shape just proposed is presented.

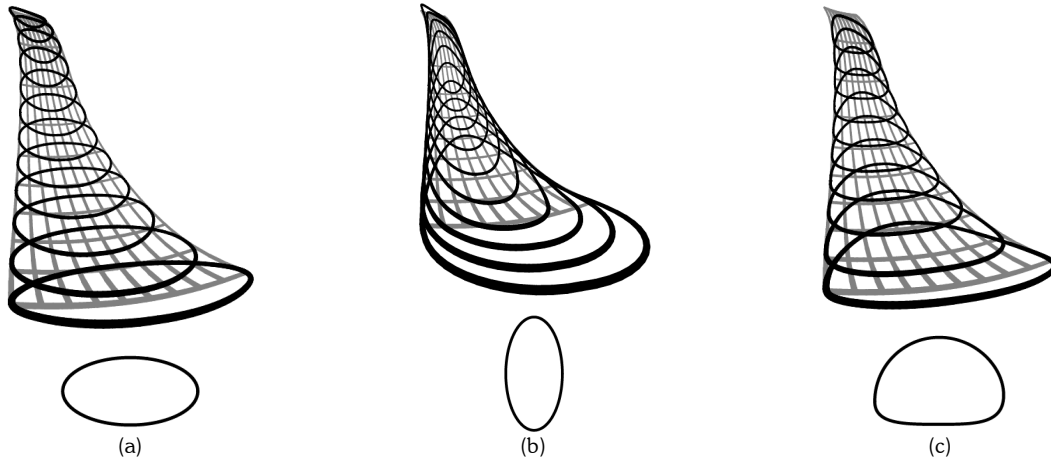


Fig. 2. Several examples of rounded shapes one can use to compose and map through  $S$  to yield  $C^1$  continuous toolpath. Compare with Figure 1(c), which is the result of composing circles

The precise composition yields a piecewise polynomial or rational. Very few NC machine controllers support the direct processing of spline curves. In fact, few are supporting the helical motions that are necessary for following the piecewise-arcs approximations of 5-axis toolpaths. If the toolpath is to be approximated as a piecewise linear curve, then the composition could clearly be approximated by simply sampling points along the curve  $C(t)$ , at  $t_i, i = 1, \dots, n$ , only to be then evaluated through  $S$  as  $S(C(t_i)), i = 1, \dots, n$ .

In the next section, we consider two additional problems that must be solved in order to make the set of rounded composed shapes into a complete toolpath: how to connect adjacent rounded shapes into one complete  $C^1$  continuous toolpath and how to offset this toolpath.

**2.2 Making a Complete Toolpath**

Given two adjacent rounded shapes on  $S, C_i = S(C_i)$  and  $C_{i+1} = S(C_{i+1})$ , one needs to connect the two (and all other adjacent shapes) into one long toolpath. Any bi-tangent line segment,  $L_{i,i+1}$  in  $D$ , between  $C_i$  and  $C_{i+1}$  could serve as the bridge connection between the two shapes, that is  $C^1$  continuous in the parametric domain. Then, and since  $S$  is  $C^1$  continuous,  $L_{i,i+1} = S(L_{i,i+1})$  is going to be tangent to both  $C_i$  and  $C_{i+1}$  over  $S$ , or the toolpath  $C_i, L_{i,i+1}, C_{i+1}$  is going to be  $C^1$  continuous.

Two disjoint convex planar shapes have four common bi-tangents: two crossing bi-tangents and two bi-tangents on two different sides. Two intersecting convex shapes will only have two bi-tangents on the sides; see Figure 3.



Fig. 3. (a) Two disjoint planar convex shapes have four bi-tangents. (b) Two intersecting planar convex shapes can have two bi-tangents only.

In our case, two adjacent rounded shapes are interesting and hence we should expect two bi-tangents on two different sides. One can explicitly solve for these bi-tangents by solving for the two polynomial constraints of

$$0 = \langle C_i(t) - C_{i+1}(r), N_i(t) \rangle,$$

$$0 = \langle C_i(t) - C_{i+1}(r), N_{i+1}(r) \rangle,$$

and two unknowns,  $t$  and  $r$ , where  $N_i$  denotes the normal field of the curve  $C_i$ . This approach is clearly feasible (see [7] and, in fact, the bi-tangents in Figure 3 were derived by explicitly solving Equations (1)). However, in our case, this can be unnecessary, at times. The two adjacent rounded shapes are, in many cases, tangent to the boundary of  $S$ . This holds, due to the fact that we place the rounded shapes in the parametric domain of  $S$  so that they cover maximum domain and hence make them tangent to the boundaries. Hence, one can simply use segments of the boundary curve of  $S$  to bridge between adjacent rounds, in a  $C^1$  manner. This simplified tangency-to-the-boundary approach was employed in the impellers' examples shown in this work.

Another difficulty we must handle before having a valid toolpath, stems from the need to perform offsets of the toolpath away from the boundary of the pocket, by the tool radius. One can foresee several options. One option is for the original surface pocket  $S$  to be properly clipped at all boundaries, by the offset radius, before we apply the presented approach. The problem with this first solution is that now  $S$  must be represented as a *trimmed surface* as the offset edges, in the parametric domain, could be quite arbitrary.

An alternative approach would directly offset points on  $S(C_i)$  that are too close to the boundary. For each location along the toolpath, find the closest location to one of the boundaries and determine the direction of the offset, in the tangent plane of  $S$ , away from that boundary. If  $S$  is regular, the offset direction could be expressed in terms of the partial derivatives of  $S$  and hence the offset, in terms of parametric coordinates, could be determined as well.

To determine the minimal distance to one of the boundaries, one is required to compute geodesic distances over  $S$ . One typically initiates the process by building a discrete distance map over the  $D$  domain. Starting from the boundaries themselves with zero distance, a breadth first search (BFS) is applied to the discrete distance map until all distances are determined. Interestingly enough, the offset amount should be such that after the offset operation, the tool will no longer collide with the boundaries. This means that any point that is far from the boundaries more than the tool radius need not move. In practice, we found that while boundary points should be moved an amount that is equal to the tool radius, better and more stable behavior is achieved if interior points are moved an amount that is proportional to their distance to the boundary, with points along the bisector (that are equally distant from two opposite boundaries) are indeed left stationary; see Figure 4 for an example. The next section considers another question one must address, in the context of 5-axis machining, and that is the question of orienting the tool.

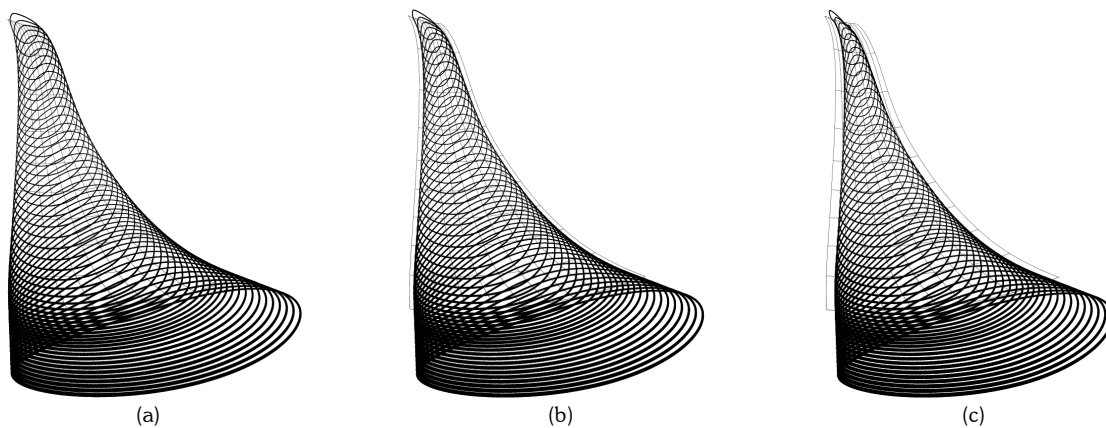


Fig. 4. The machined impeller using the proposed tool path scheme and a cylindrical tool. The original tool path (a) is being offset 1mm in (b) and 2mm in (c) from the boundary. Note this offset is varying and only points near the boundary are moved the full amount.

### 2.3 Orienting the Tool

The presented approach can employ arbitrary tool orientation scheme. The simplest can be to employ the surface normal to orient a flat end tool. Yet, in many cases the surface normal is not the proper orientation to use. For example, the use of surface normals to orient the tool in the case of the impeller model, in Figure 1, would yield walls of varying thickness. Better accessibility in complex environments might demand a tailored tool orientation field. Similarly, curvature matched machining [13] is another motivation to use an orientation field, other than the surface normals' field.

A more general orientation prescription scheme allows for arbitrary setting of a vector field,  $V(u, v)$ , to govern the tool orientation. In the impeller case,  $V(u, v)$  is derived as the difference surface between the top pocket surface and the base pocket surface; see Figure 5.

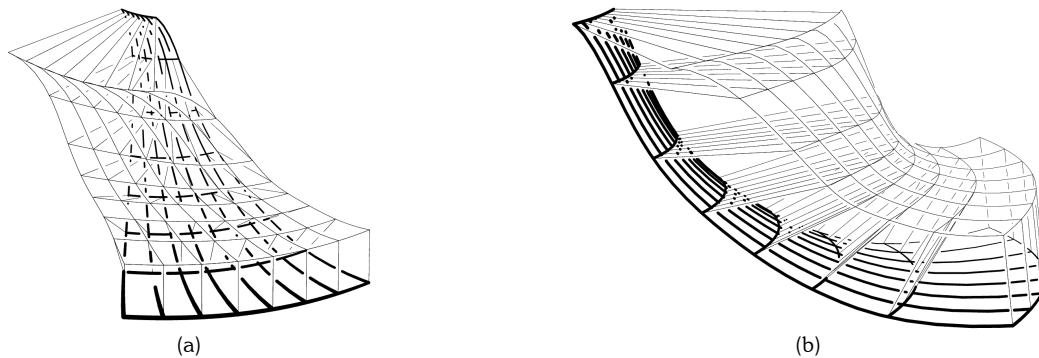


Fig. 5. Two views on the vector field that is computed to orient the tool, as the difference of the top and bottom surfaces of the 5-axis machined pocket. Note this field can be different from the normal field of the bottom pocket surface,  $S$ , as is the case here.

If the used tool is tapered, further measures must be taken to ensure the tool is properly tangent to the side walls of the pockets. Much like the variable offset we used in the previous section, herein, one can use variable rotations. Let the tapered angle be  $\theta$ . Then, tool orientation vectors that are on the boundary will be rotated in (around the tangent vector of the boundary curve) by  $\theta$  degrees. Interior tangents will be rotated less and less until the bisector locations at which no rotations take place. In the next section, we present the results of using the proposed toolpath scheme, in machining impeller models, one of which is from Figure 1.

### 3. 5-AXIS HSM EXAMPLES

In order to machine the impellers, which were 125mm in diameter, two types of tools were used. The first was a 5mm (diameter) cylindrical (rounded) flat end tool and the second was a (rounded) flat end 4mm (diameter) tapered tool with 3 degrees tapering.

The toolpath was created with a 1/3 of a mm size steps yielding a very good finish. Figure 6 shows the final generated toolpath for the impeller in Figure 1, and using a cylindrical flat end tool. Figure 7 shows the actually machined impeller, from two different views. Figure 8 shows a similar, six-blade impeller that was machined using the tapered tool, from two different views.

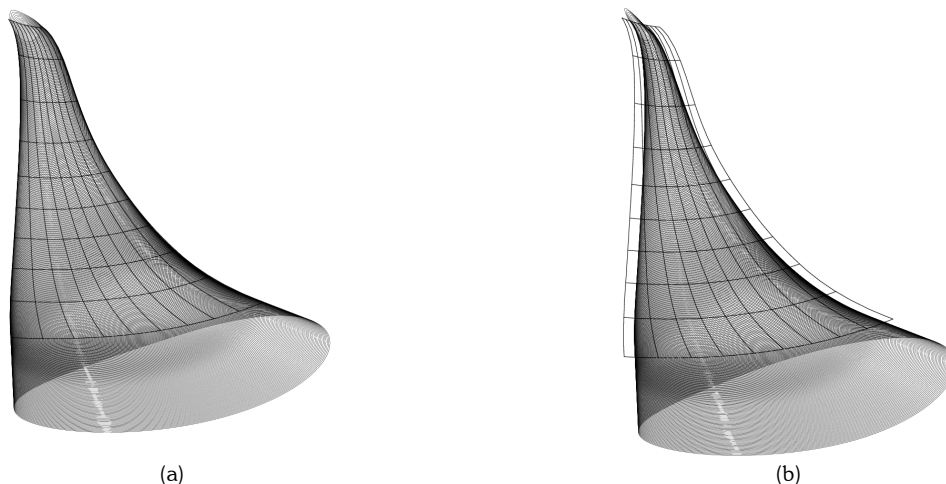


Fig. 6. The final tool path for the impeller in Figure 1, using a cylindrical flat end tool. In (a), the toolpath before the offset is shown whereas (b) presents the tool path after the (variable) offset was applied.

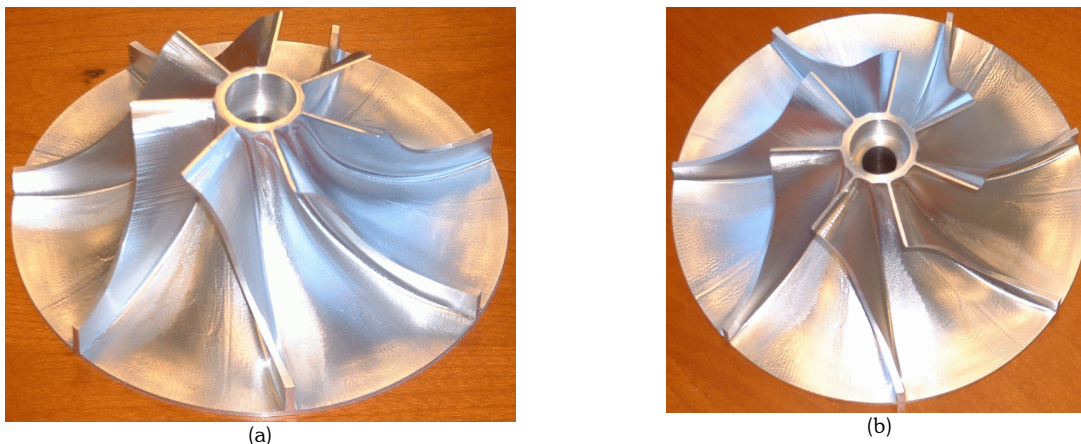


Fig. 7. The machined impeller using the proposed tool path scheme and a cylindrical tool.

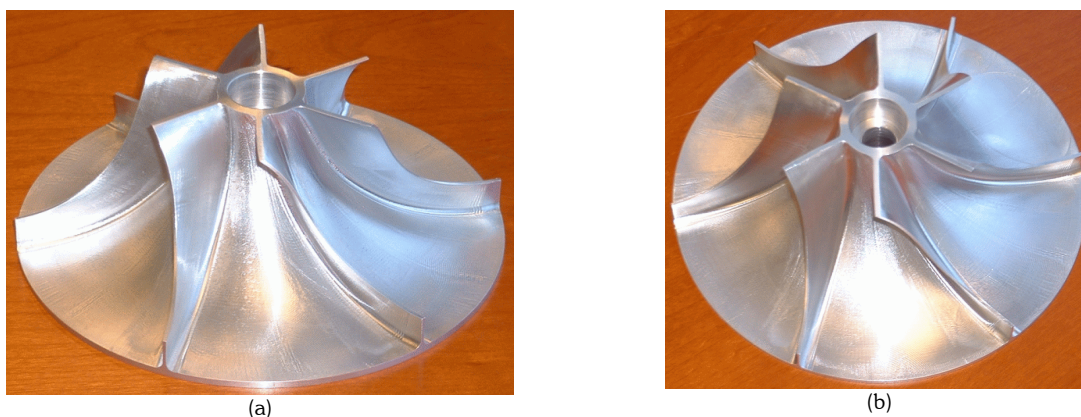


Fig. 8. A different, 6 blades, impeller machined using the proposed tool path scheme and a tapered tool.

#### 4. DISCUSSION, FUTURE WORK AND CONCLUDING REMARKS

This work presented an approach to construct  $C^1$  continuous toolpaths that are highly suitable to high speed machining. By connecting the adjacent rounded shapes by bi-tangent lines,  $C^1$  continuity has been achieved.

One can consider the use of higher order contact curves between adjacent rounded shapes. For example, by using a quintic Hermite to bridge between adjacent rounded shapes, one can construct a  $C^2$  continuous toolpath. Composed over a  $C^2$  surface, the resulting toolpath will be  $C^2$  as well.

Another open basic question could be found in the construction of the rounded shapes of the toolpath. Better distance bounds, than globally using the first fundamental form, to bound the distance changes and  $D_h$ , must be sought. One clear alternative is to try and get estimates on the distance changes in the local neighborhood instead of globally. Such local bound is likely to be tighter. Similarly, better ways to control the rounded shape after it is composed with  $S$  should be sought as well. One possibility would be to prewarp curve  $C_i$  so that  $S(C_i)$  would achieve the desired result.

Yet another clear deficiency of the presented scheme is that in about half the time, the tool is in free motion, cutting nothing. Yet, since this free motion is easily identifiable, being the back half portion of every rounded curve  $S(C_i)$ , one can make the tool move along these half back portions much faster. In effect, the time now spent in free motion will be minimized.

The presented approach could be combined with other techniques to minimize the scallop height such as curvature matched machining [13]. The quality of this result closely depends on the maximal distance between adjacent

toolpaths, herein, adjacent composed circular shapes and should probably be compared to other toolpath generation schemes.

## 5. ACKNOWLEDGMENTS

The research was supported in part by ARO(DAAD19-01-1-0013) and the Fund for Promotion of Research at the Technion, Haifa, Israel. All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

## 6. REFERENCES

- [1] Arkin, E. M., Held, M. and Smith, C. L., Optimization problems related to zigzag pocket machining, *Algorithmica*, Vol. 26, No. 2, 2000, pp 197-236.
- [2] Choi, B. K. and Kim, B. H., Die-cavity pocketing via cutting simulation, *Computer Aided Design*, Vol. 29, No. 12, 1997, pp 837-846.
- [3] Chou, J. J., *NC Milling Machine Toolpath Generation for Regions Bounded by Free Form Curves and Surfaces*, PhD thesis, University of Utah, UT, 1989.
- [4] DeRose, T., Goldman, R., Hagen, H., and Mann, S., Functional composition via blossoming, *ACM Transactions on Graphics*, Vol. 12, No. 2, 1993, pp 113 – 135.
- [5] DoCarmo, M. P., *Differential Geometry of Curves and Surfaces*, Prentice-Hall, 1976.
- [6] Elber, G., *Free Form Surface Analysis using a Hybrid of Symbolic and Numeric Computation*, Ph.D. thesis, University of Utah, UT, 1992.
- [7] Elber, G., and Kim, M.-S., Geometric constraint solver using multivariate rational spline functions, *Proc. of ACM Symposium on Solid Modeling and Applications*, Ann Arbor, MI, June 4--8, 2001.
- [8] Elber, G., Cohen, E., and Drake, S., MATHSM: Medial axis transform toward high speed machining of pockets, *Computer Aided Design*, Vol. 37, No. 2, 2005, pp 241-250.
- [9] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F., *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Publishing Company, second edition, 1990.
- [10] Hansen, A. and Arbab, F. An algorithm for generating NC tool paths for arbitrarily shaped pockets with islands, *ACM Transactions on Graphics*, Vol. 11, No. 2, 1992, pp 152-182.
- [11] Held, M., A geometry-based investigation of the tool path generation for zigzag pocket machining. *The Visual Computer*, Vol. 7, No. 5-6, 1991, pp 296-308.
- [12] Held, M. Pocket machining based on contour-parallel tool paths generated by means of proximity maps, *Computer Aided Design*, Vol. 26, No. 3, 1994, pp 189-203.
- [13] Jensen, C. G., Red, W. E., and Pi, J. Tool selection for five-axis curvature matched machining, *Computer Aided Design*, Vol. 34, No. 3, 2002, pp 251-266.
- [14] Lee, E. Contour offset approach to spiral toolpath generation with constant scallop height, *Computer Aided Design*, Vol. 35, No. 6, 2003, pp 511-518.
- [15] Persson, H., NC machining of arbitrary shaped pockets. *Computer Aided Design*, Vol. 10, No. 3, 1978, pp 169-174.
- [16] Spiral high speed machining, <http://www.cimatron.com>, Cimatron.
- [17] Tang, K., Chou, S. Y., and Chen, L. L., An algorithm for reducing tool retractions in zigzag pocket machining, *Computer Aided Design*, Vol. 30, No. 2, 1998, pp 123-129.
- [18] Tang, K. and Joneja, A. Traversing the machining graph of a pocket, *Computer Aided Design*, Vol. 35, No. 11, 2003, pp 1023-1040.
- [19] Trochoidal high speed machining, <http://www.delcam.com>, <http://www.mmsonline.com>, Delcam.
- [20] Veeramani, D. and Gau, Y. S., Selection of an optimal set of cutting-tool sizes for 2 ½ D pocket machining, *Computer Aided Design*, Vol. 29, No. 12, 1997, pp 869-877.