Journal for Geometry and Graphics
Volume 17 (2013), No. 2, 223–235.

Geometric Simulation of Locally Optimal Tool Paths in Three-Axis Milling

Márta Szilvási-Nagy[1], Gyula Mátyási[2], Szilvia Béla[1]

[1] *Department of Geometry, Budapest University of Technology and Economics*
Egry József u. 1. H. ép. 22, H-1111 Budapest, Hungary
emails: {szilvasi,belus}@math.bme.hu

[2] *Department of Manufacturing Science and Technology,*
Budapest University of Technology and Economics
email: matyasi@manuf.bme.hu

Abstract. The most important aim in tool path generation methods is to increase the machining efficiency by minimizing the total length of tool paths while the error is kept under a prescribed tolerance. This can be achieved by determining the moving direction of the cutting tool such that the machined stripe is the widest. From a technical point of view it is recommended that the angle between the tool axis and the surface normal does not change too much along the tool path in order to ensure even abrasion of the tool. In this paper a mathematical method for tool path generation in 3-axis milling is presented, which considers these requirements by combining the features of isophotic curves and principal curvatures. It calculates the proposed moving direction of the tool at each point of the surface. The proposed direction depends on the measurement of the tool and on the curvature values of the surface. For triangulated surfaces a new local offset computation method is presented, which is suitable also for detecting tool collision with the target surface and self intersection in the offset mesh.

Key Words: tool path simulation, isophotes, offset, triangular mesh

MSC 2010: 65D17, 68U20

1. Introduction

This paper deals with three-axis milling, and presents computations on triangulated and analytic surfaces for determining the optimal moving direction of the tool at a given point of the surface. Basic requirements, such as the tool does not remove material from the target surface and the error remains under a prescribed tolerance, can be fulfilled by keeping the processed surface between the target (part) surface and its outer parallel (offset) surface at distance of a given tolerance. In the focus of these investigations are two problems of tool path planning,

ISSN 1433-8157/$ 2.50

- how to compute the offset of the mesh in a neighborhood of the actual point, and
- how to determine the optimal moving direction at any point of the surface represented by a triangular mesh.

In the following short survey we present different mathematical approaches dealing with geometrical factors in surface milling. The most frequently used tool path generation methods cut the surface by parallel driving planes in equal intervals or extract isoparametric curves in equally spaced parametric steps. If the surface has regions of different curvatures, both methods lead to uneven distances between tool paths, consequently, to fluctuating errors. The machining error on the processed surface is measured by the height of the cusps (scallops) left between two adjacent tool paths (Fig. 1). Constant scallop height tool path generation methods for ball-end milling of analytic surfaces are described in [24, 6, 10], however, the required computations are expensive, and some technical questions (e.g., self-intersections) are not solved in all cases. For triangular meshes, in [12] a constant scallop height method is presented for computing the distances between slicing planes in three-axis milling with a ball-end cutter.

In [5] the application of two different tool path generation methods (using parallel driving planes or z-level contour parallel curves) is suggested by splitting the surface into "flat" and "steep" regions according to the angle between the tool axis and the surface normal .

In [7] and [21] a method is presented for approximating the processed part of the surface around a contact point, for generating the widest stripe along a tool path, and for investigating the local and global millability with a given cutting tool. The condition for collision-free manufacturing is expressed in terms of the curvature indicatrices of the part surface and the tool surface at their touching point. This local millability can be extended to global millability along an isophotic line. The angle between a given direction (the tool axis) and the surface normal does not change along an isophote; consequently the area of the contact surface of the tool is constant, what is optimal for the abrasion of the tool end. Isophotes are applied to partition the surface into regions, in which a generic side step (the distance between two neighboring tool paths) can be computed from the prescribed maximal scallop height [4]. In [23] such regions are computed for the detection of interference between the surface and the tool (called usually gouging). In [22] an isophote-based tool path generation is proposed and in [8] an application to NC-machining. A numerical method for the computation of isophotic curves on an analytic surface is presented in [14]. Despite of the mentioned advantageous properties of isophotes these curves are not suitable milling paths due to their uneven density causing uncontrollable scallop heights. In general, each set of surface curves fulfills only some conditions for tool paths, but not all at the same time.

The investigation of the surface curvatures at the touching point of the target surface and the cutting tool implies that the principal direction of maximal normal curvature is the most efficient tool feed direction in three-axis milling as it minimizes the total length of necessary tool paths ([1, 2, 21]). Namely, this "steepest ascending" tool path leaves the widest track (processed stripe) on the machined surface, as also shown in [21]. A flexible strategy for parallel plane milling considering the surface roughness is presented in [16] by introducing a fitness function proportional to the feed rate and the transversal step. Then the acceptable machining directions are computed by optimizing the fitness function.

In surface offsetting algorithms a difficult problem is to detect and to remove self-intersections. For triangulated surfaces different offsetting methods have been published. Faces, edges and vertices are offset by using half cylinders and spheres; then the gaps are filled with blending methods, and overlaps are removed. By slicing this corrected offset triangular mesh

with a series of planes the three-axis tool paths are calculated [13]. By a local offsetting scheme the facets are translated in the normal direction; convex edges are converted into trimmed cylinders and convex vertices into trimmed spheres. Concave edges are removed from offsetting. Then in tool path generation, the intersection with a driving plane results in circular arcs and line segments which are sorted and linked; overlapping portions are removed in order to generate a compound curve for the cutting path [9, 11]. In [15] each triangle is offset, the sliced segments in the drive plane are swept and chained into monotone polygonal tool paths.

In this paper a new, more effective offsetting method will be presented (see Section 2). In Section 3 the tool paths on triangulated surfaces are simulated by moving the tool in principal directions. Section 4 presents proposals for the optimal moving direction at a given point as well as its computation on analytic surfaces. Section 5 shows examples.

2. Mesh offsetting and computing the processed part of the surface at a point

In three-axis milling the direction of the tool axis is fixed, and the tool end is moved in three mutually orthogonal directions, the axis direction and two other directions in the perpendicular plane. The result of the milling process with differently shaped tools is an approximation to the target surface, where the processed shape shows *scallops* (cusps). The maximal height of these scallops has to be kept within a given tolerance, i.e., the machined surface must lie between the target (part) surface and its offset for the given tolerance (Fig. 1). The piece of the tool surface lying in this region will be named *contact surface* of the tool. The curve of intersection of the tool surface and the offset surface is the boundary of this contact surface. The processed patch around a given touching point of the tool and the target surface is bounded by the projection of this boundary curve onto the part surface. The set of all processed patches should cover the entire part surface. In Fig. 2 a smooth surface is shown together with its offset, with a touching ball-end tool and the processed patch on the surface.

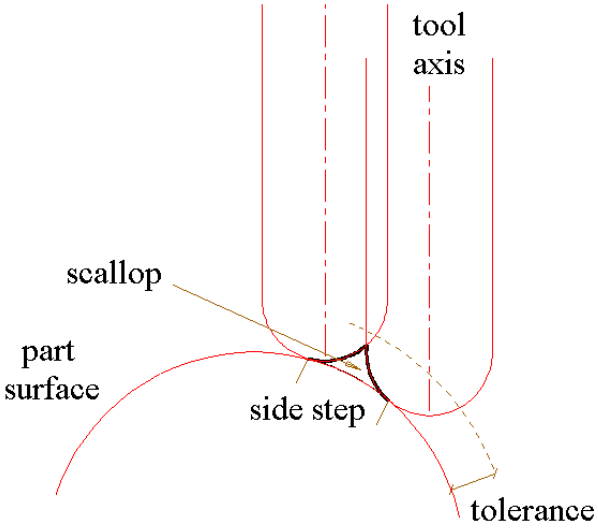

Figure 1: Milling with given tolerance

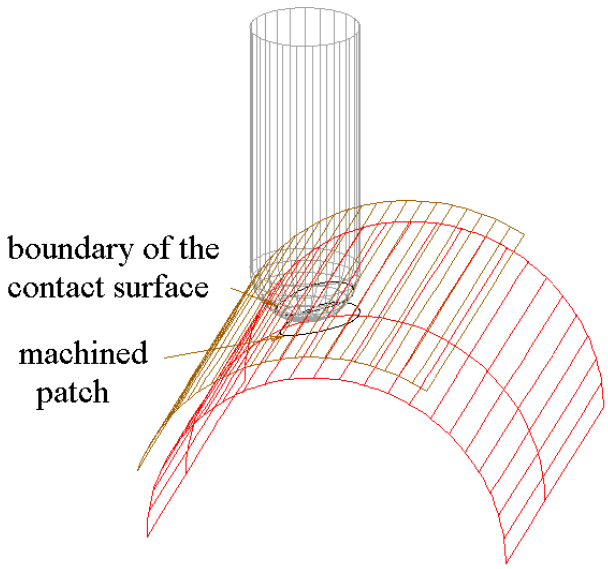

Figure 2: The processed patch on the part surface

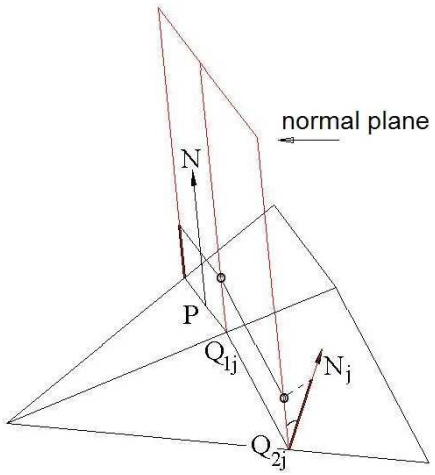

Figure 3: Offset construction in a normal plane

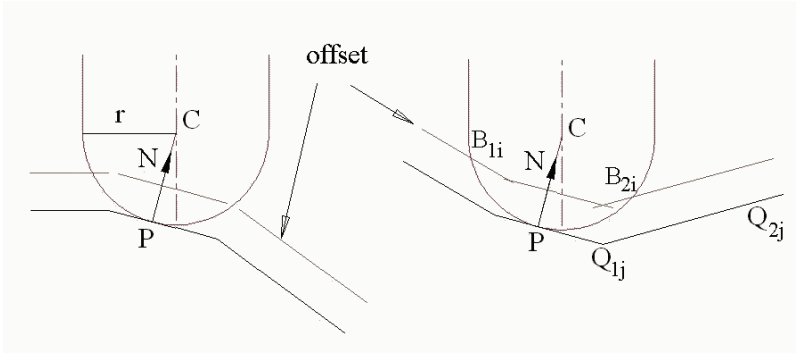

Figure 4: Computation of the intersection with the tool in a convex and a concave region

In order to compute the processed patch at a given point, we first have to generate the offset surface.

When the surface is represented by a triangular mesh, we consider an actual position of the tool which touches a facet of the mesh with normal vector $\mathbf{N}$ at the point P. The offset will be computed in n normal planes. Each normal plane passes through the normal line of the actual triangular facet at the touching point P and it is determined by an arbitrary direction in the plane of this triangle. The offsetting is carried out on the segments which are the intersections between the normal plane and the triangular mesh (Fig. 3). If the i-th normal plane intersects the j-th triangle with normal vector $\mathbf{N}_j$ along the segment $[\mathrm{Q}_{1j}, \mathrm{Q}_{2j}]$, its offset is $[\mathbf{Q}_{1j}+\varepsilon'\,\mathbf{N}_j,\ \mathbf{Q}_{2j}+\varepsilon'\,\mathbf{N}_j]$ marked by circles in Fig. 3.[1] Here ε is the given tolerance and $\varepsilon' = \varepsilon\cos\alpha$ with α denoting the angle of inclination between the normal vector $\mathbf{N}_j$ and the i-th section plane. ε is shown by a thick segment and α is denoted by one arc in Fig. 3.

In a convex region gaps arise between the offset segments, but in a concave region the adjacent segments intersect. Filling the gaps and removing self-intersections of the offset mesh are carried out by two-dimensional methods in each section plane. The obtained polygonal line intersects the normal section of the tool end in two points, B_{1i} and B_{2i} which are lying on the boundary curve of the contact surface of the tool and the offset mesh (Fig. 4). The projections of B_{1i} and B_{2i} onto the mesh give the endpoints of a diameter of the processed patch. This computation in the normal planes ($i = 1, \ldots, n$) results in $2n$ points lying on the boundary curve of the processed patch.

In Fig. 5 the part of the offset surface within the ball-end of the tool is shown by 24 polygonal lines of intersection above the base triangle. The boundary of the processed patch is drawn by a polygonal line on the mesh. Additional section planes can be included easily, if necessary.

This local offsetting method is novel in the literature. It has the advantage that only two-dimensional algorithms are necessary in the computation. Self intersection in the offset mesh and tool collision with the target surface can be detected by simple computation just checking the corresponding arcs and segments for intersection in each normal plane. Moreover, registering the directions of the normal planes, where such problems arise, a method for par-

[1] We generally denote the position vector of any point by the corresponding boldface letter.

titioning the surface could be developed for different milling tools and strategies. The applied edge-oriented polyhedral data structure defined on the mesh is very effective in determining planar sections of the mesh [17].

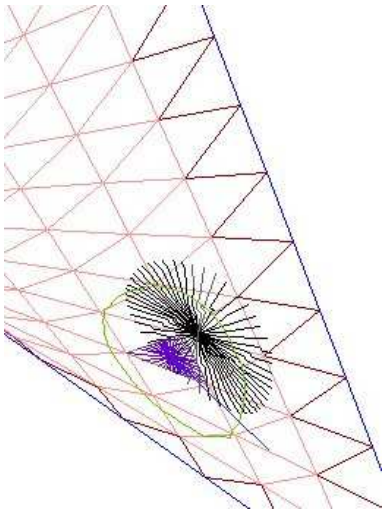

Figure 5: Part of the computed offset within the ball-end and boundary of the processed patch on the mesh of a quadratic surface. The actual triangle is lined.

3. Moving the tool in principal direction on meshes

First, the requirement is considered that the machined stripe along a tool path is the widest. Such tool paths will usually minimize the total length of milling paths [1, 2]. The patch processed by the tool around the touching point is determined by its n diameters lying on the surface. Now the tool will be moved in the direction perpendicular to the largest diameter (except at umbilical points). The next touching point of the tool will be chosen in this direction on the boundary of the processed patch. A similar moving direction is suggested in [21]; nevertheless, our computational approach is different, because in [21] the authors used Dupin-indicatrices.

In Figs. 6 and 7 tool paths of the widest machined stripe are displayed on a cylindrical surface. Also a floating patch above one base triangle is shown which is the part of the offset mesh within the ball-end of the tool. The moving directions are shown by straight line segments. Each is emanating from the touching point of the actual triangle. The diameters of the processed patches are the shortest in the directions perpendicular to the generators of the cylinder in the case, when the tool touches the cylinder on the convex side, and they are parallel to the generators, when the tool is on the concave side of the surface. In the case of these "synthetic" meshes of a cylindrical surface the largest and shortest diameters of a processed patch are perpendicular to each other, and they are practically the principal directions. Investigations have been made on different synthetic meshes which revealed that the direction of the widest processed patch is very close to the principal direction of the largest normal curvature computed in a sufficiently large region around the contact point. For the estimation of the principal directions at a point of the triangulated surface, a method is described in [18] and [19] by specifying a circular disc on the mesh and computing the minimal and maximal chord lengths between the end points of its curved diagonals. In

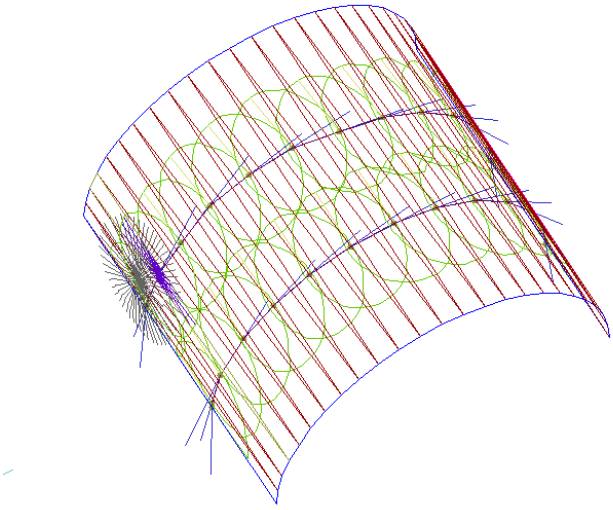

Figure 6: Tool paths on a cylindrical surface machined from the convex side

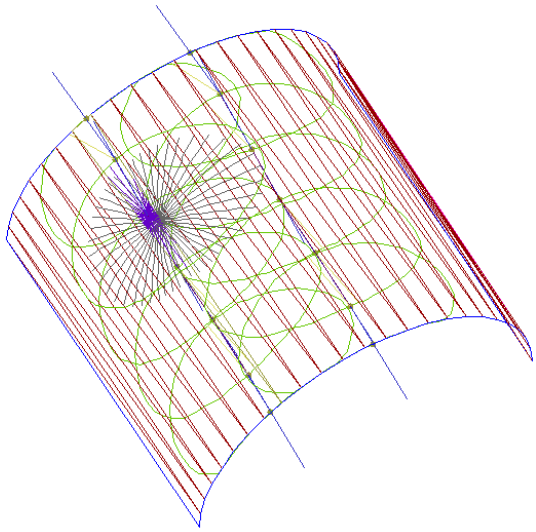

Figure 7: Tool paths on the same cylindrical surface machined from the concave side

this concept the curvature values are assigned to the triangular facets of the mesh, and the approximated surface normals are replaced by facet normals.

The applied normal curvature estimation method constructed for triangular meshes is suitable for the computation on meshes also in such cases, where the known vertex-oriented computations do not work. Namely, in the mesh of the shown cylinder all the vertices are situated on the surface boundary. However, the frequently used vertex-oriented methods require 'good' vertices in the interior of the actual region.

Even if the total tool path computed in the widest stripe direction can be the shortest, these curves are not always appropriate tool paths due to technical reasons. Namely, the angle between the tool axis and the surface normal changes to much on a curved surface, which leads to uneven abrasion of the tool. Hence, the computed moving direction should be combined with other tool path generation algorithms as suggested also in [21].

4. Locally optimal moving direction and its computation on analytic surfaces

Now we consider the technical requirement that the abrasion of the tool should be even during the milling process. This requirement is fulfilled, if the angle between the tool axis and the surface normal remains constant along a tool path. Such curves are the isophotes on smooth surfaces.[2]

As the isophotes are not suitable tool paths, a compromise will be suggested, and the tool will be moved from the instantaneous contact point neither in the direction of the widest processed patch, nor along an isophote. A first idea was to choose the bisector of these two directions. The computation has been made on an analytic surface given by the function $f(x, y)$, assuming that the piece of the surface to be machined is visible from the tool axis ($=$ z-axis) direction [20]. In such cases the z-projection method can be applied, when the processed patch is computed by projecting the boundary of the contact surface parallel to the z-direction, and the step size is given in the xy-plane [3]. For the points $\mathrm{B}\,(x, y, f(x, y))$

[2]The investigation of other material and kinematical factors is not object of this paper.

of the boundary curve of the contact surface the equation

$$r = \text{distance}(\mathbf{C},\, \mathbf{B} + \varepsilon \mathbf{N}(x,y)), \quad \mathbf{N}(x,y) = \frac{\big(f'_x(x,y), f'_y(x,y), -1\big)}{\sqrt{f'^2_x(x,y) + f'^2_y(x,y) + 1}}$$

holds, where r is the radius and C is the center point of the ball-end of the tool, and ε is the offset distance. $\langle\,,\rangle$ denotes the dot product and f'_x, f'_y are the partial derivatives. The points of the isophote passing through the point $\mathrm{P}\,(x_p, y_p, f(x_p, y_p))$ satisfy the equation

$$\langle \mathbf{N}_p,\, \mathbf{a} \rangle = \langle \mathbf{N}(x,y),\, \mathbf{a} \rangle\,, \tag{1}$$

where $\mathbf{a}$ is the unit vector of the tool axis direction. In the specific coordinate system with $\mathbf{a} \parallel z$, this equation is equivalent to

$$f'^2_x(x_P, y_P) + f'^2_y(x_P, y_P) = f'^2_x(x,y) + f'^2_y(x,y)\,. \tag{2}$$

Equations (1) and (2) can be solved numerically resulting in a finite number of points which lie on the required surface curves, i.e., on the boundary of the processed patch and on the isophote, respectively.

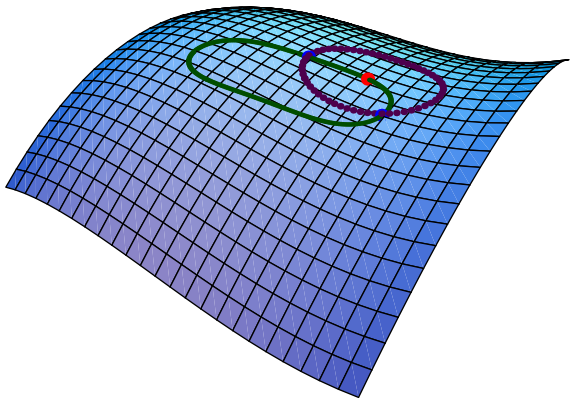

Figure 8: A processed patch around the contact point of the tool and the isophote through this point on a cubic surface

Figure 8 shows on a cubic surface how the processed patch around a contact point and the isophote through this point are situated. From the two common points of the curves two bisector directions can be determined (Fig. 9). One of them (which doesn't lead to a zig-zag curve) has been chosen for the tool path direction by checking "turn left" and "turn right" in the chain of the computed segments. In Fig. 10 a few points of three generated tool paths with the processed patches are shown. The step size is constant. The problem of varying distances between the tool paths occurs here as well as also in the frequently used methods [4].

A more reasonable idea for finding the optimal direction at any point of the surface is to consider the change of the angle between the tool axis and the surface normal (i.e., the inclination angle of the isophote passing through the actual point) while the tool is moving into a next position. This will determine the correction factor of the declination from the widest stripe direction, i.e., the direction, where the absolute value of the normal curvature is maximal. If the surface is nearly flat, the surface normal doesn't change very much while

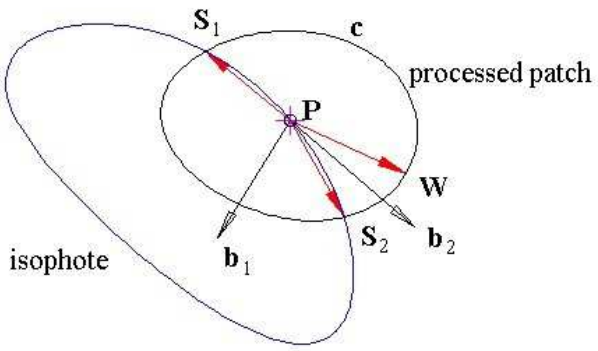

Figure 9: Bisector directions $\mathbf{b}_1$ and $\mathbf{b}_2$ of the angle $\langle \mathrm{W}, \mathrm{P}, \mathrm{S}_1 \rangle$ and $\langle \mathrm{W}, \mathrm{P}, \mathrm{S}_2 \rangle$, respectively.

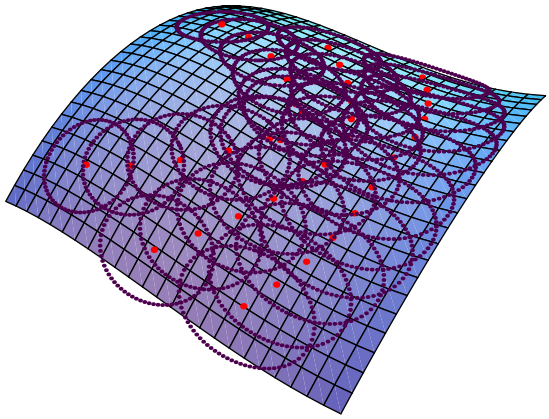

Figure 10: Tool paths generated in the bisector of the widest stripe and the isophote directions. To each point the processed patch is shown.

moving in the widest stripe direction. Therefore, the tool can be kept on this surface curve, because the abrasion will be even. In the case of larger curvature values the tool should be moved rather towards the corresponding isophote direction.

In the generic case the suggested moving direction will be determined between the widest stripe direction and the direction pointing to the isophote point on the boundary of the processed patch $\mathbf{c}$ in such a way that it divides the arc between these two directions in the ratio $(1 - \cos\beta) : \cos\beta$. Here β denotes the difference of the inclination angles of the surface normals to the tool axis at the two corresponding points on the processed patch boundary. In Fig. 11 the vector $\mathbf{w}$ is pointing into the direction of the widest stripe and $\mathbf{s}$ is pointing to the point S of the isophote passing through the contact point P. The boundary $\mathbf{c}$ is represented by $2n$ points S_i which are the projections of the points lying on the boundary of the contact surface denoted by B_{1i} and B_{2i} in Fig. 4 (n is the number of the section planes). The isophote point S on $\mathbf{c}$ can be approximated by one of the points S_i, in which the angles satisfy

$$\angle(\mathbf{N}_{S_i}, \mathbf{a}) \approx \angle(\mathbf{N}_P, \mathbf{a}),$$

which can be computed from the condition

$$|\arccos\langle \mathbf{N}_S, \mathbf{a}\rangle - \arccos\langle \mathbf{N}_P, \mathbf{a}\rangle| \approx \min\left\{|\arccos\langle \mathbf{N}_{S_i}, \mathbf{a}\rangle - \arccos\langle \mathbf{N}_P, \mathbf{a}\rangle|\,,\ \mathbf{S}_i \in \mathbf{c},\ i = 1, \ldots, 2n\right\}$$

where $\mathbf{N}_{S_i}$ is the normal vector of the surface at the point S_i. This means, the direction of $\mathbf{N}_S$ should fit into the cone determined by the isophote's angle of $\mathbf{N}_P$ (marked by a single arc in Fig. 11). The vectors $\mathbf{N}_S$, $\mathbf{N}_P$, $\mathbf{N}_{S_i}$, and $\mathbf{a}$ are unit vectors.

Let β denote the difference of the angles which are formed by the surface normals $\mathbf{N}_W$ and $\mathbf{N}_S$ to the axis direction $\mathbf{a}$.

$$\beta = |\arccos\langle \mathbf{N}_W, \mathbf{a}\rangle - \arccos\langle \mathbf{N}_S, \mathbf{a}\rangle|, \quad \|\mathbf{N}_W\| = 1,\ \|\mathbf{N}_S\| = 1,\ \|\mathbf{a}\| = 1.$$

The proposed new moving direction $\mathbf{q}$ is determined by the arc length of the processed patch boundary between $\mathbf{w}$ and $\mathbf{s}$:

$$\mathrm{arc}(\mathbf{q}, \mathbf{w}) = (1 - \cos\beta) \cdot \mathrm{arc}(\mathbf{s}, \mathbf{w}).$$

Consequently, a zero or a very small β does not change the moving direction $\mathbf{w}$. This β depends also on the radius of the ball-end. The bigger the tool, the bigger is the processed

patch. Therefore, the change of the inclination angle of the normals is larger. We remark that the factors included in the fitness function introduced in [16] have the same effect on the moving strategy.

This path correction does not influence the abrasion of a ball-end tool, it has importance in milling with other cutters. The computation for a toroidal cutter does not differ from the presented computation above, but the demonstrating figures have been made with ball-end tools.

5. Examples

Figure 12 shows two tool paths on a synthetic mesh. The path in the widest stripe direction (practically a parallel circle of the cylinder) is shown with the processed patches around the computed contact points. The corrected tool path computed by the suggested method, situated lower, shows a declination. The radius of the ball-end of the tool is in this case eight times the average size of the triangles. The declination of the new path is less with a smaller radius. In the case, when the radius is only four times the average size of the triangles, the tool will move approximately along the path of the widest stripe. The tolerance was one third of the tool radius in this case, which is the distance of the floating piece of the offset and the part surface (see left-hand side in the figure). Note that the tool paths computed on a cylinder surface in [16] in a completely different way show a similar shape.

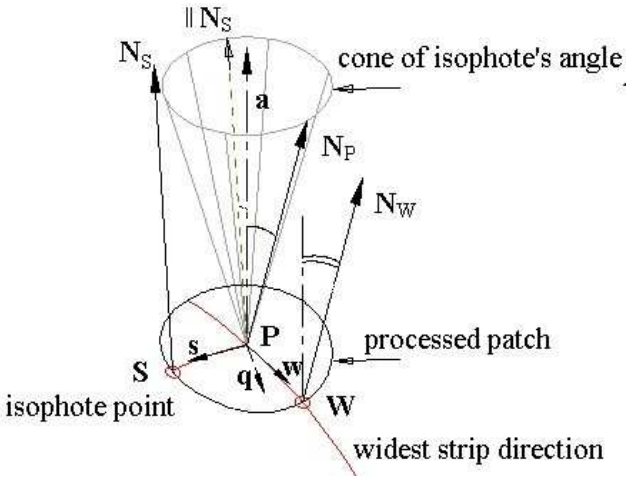

Figure 11: The proposed moving direction **q** is computed between the widest stripe direction **w** and the isophotic direction **s**

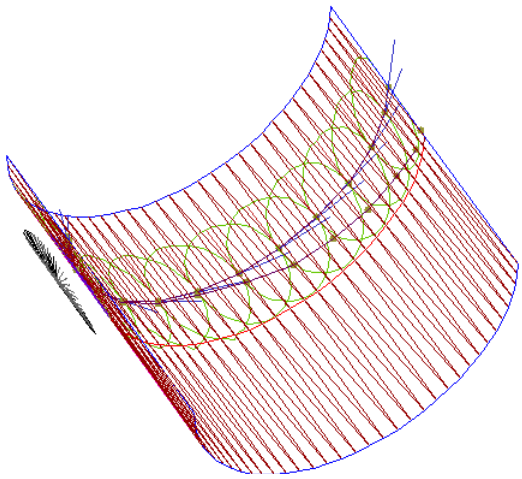

Figure 12: The upper path is computed in the widest stripe direction and the lower one in the proposed moving direction

The next examples are computed on parametric surfaces. Though the surface is presented by a vector function $\mathbf{r}(u, v)$, $(u, v) \in [u_1, u_2] \times [v_1, v_2]$, which is at least twice differentiable in a sufficiently large neighborhood of P, the computation is carried out in n normal planes producing $2n$ points of the surface curve **c** bounding the processed patch. In this way, the time consuming and unstable process of computing the projection of a point onto the surface will be replaced by a very effective method.

Suppose, the actual tangent direction vector of a surface curve through the point P is $\mathbf{s}_i$, $i \in \{1, \dots n\}$. The normal plane containing the required point S_i is determined by the vectors

$\mathbf{s}_i$ and the surface normal $\mathbf{N}_P$. The decomposition of $\mathbf{s}_i$ in the tangent plane is

$$\mathbf{s}_i = a_i \left.\frac{\partial \mathbf{r}(u,v)}{\partial u}\right|_P + b_i \left.\frac{\partial \mathbf{r}(u,v)}{\partial v}\right|_P .$$

Let B_i in this normal plane be the point to be projected onto the surface. First, a_i and b_i are computed as solutions of the system of linear equations from the decomposition of $\mathbf{s}_i$. Then series of points are generated in the parameter domain with an appropriate Δt, which are

$$(u_k, v_k) = (u_P, v_P) + k \cdot \Delta t \frac{(a_i, b_i)}{\sqrt{a_i^2 + b_i^2}}, \quad k = 1, \ldots, K .$$

The corresponding points on the surface are determined by the vectors $\mathbf{r}(u_k, v_k)$. The required point S_i is determined by the parameter values (u_k, v_k) for which the distance of the surface point $\mathbf{r}(u_k, v_k)$ to B_i is minimal ($k \in \{1, \ldots, K\}$).

Numerical tests have shown that the results of the presented approximation of the projection onto the surface are satisfying compared to the "real" projections computed by the program package Mathematica by searching for a nearest surface point to the one to be projected without any constraints. The difference between the results of the two methods is within the error bounds of the computation in other procedures of the complete task.

In Figs. 13 and 14 the results of this computation on a torus are shown. The smaller patches are generated with a smaller tool radius. Here the tool path deviates little from the parameter curve of the meridian circle, which is the path with the widest stripe with tangents in the principal directions of the bigger principal curvatures. The isophotic curves with respect to the axis of the torus (which is the axis of the tool) are the parallel circles. A larger tool radius shows a significantly bigger deviation from the steepest moving directions.

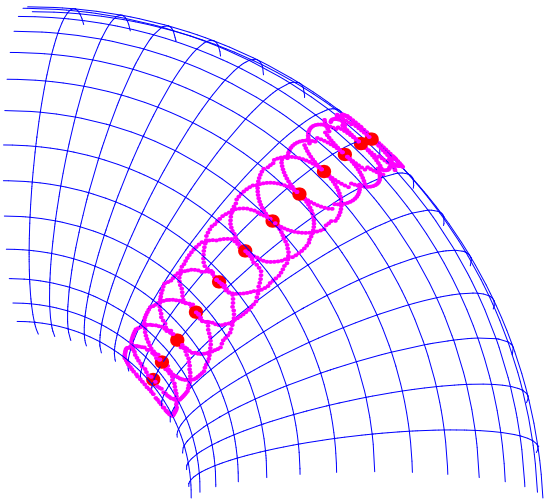

Figure 13: Tool path on the torus with the processed patches around the computed points

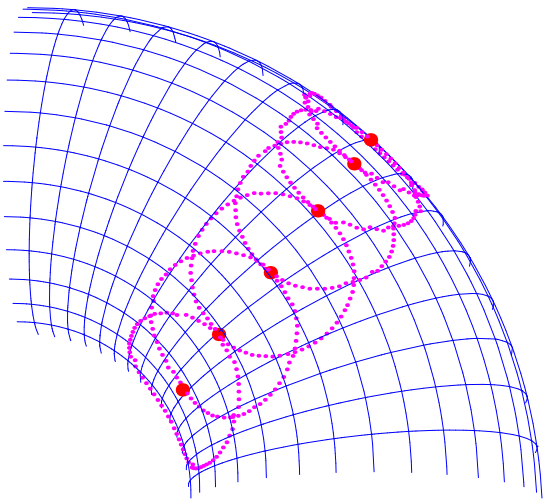

Figure 14: Tool path on the same torus with larger tool radius

The next surface in Figs. 15, 16 and 17 is described by a trigonometric function. The steepest ascending direction crosses the bump (not shown), and an isophotic curve with respect to the tool axis is shown in Fig. 15. The computed tool path goes between them, and Fig. 16, 17 show that the computed moving directions are stable also in more curved regions.

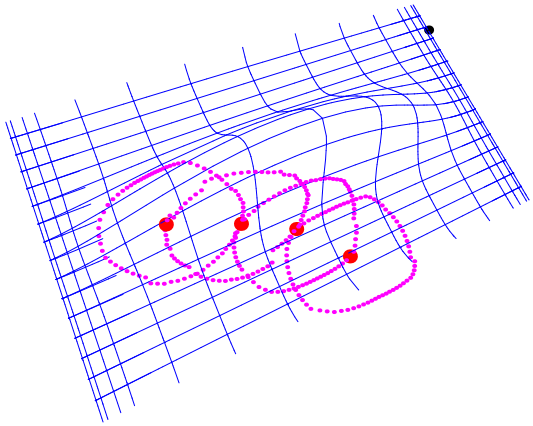

Figure 15: Isophotic curve with respect to the direction perpendicular to the base plane on a trigonometric surface

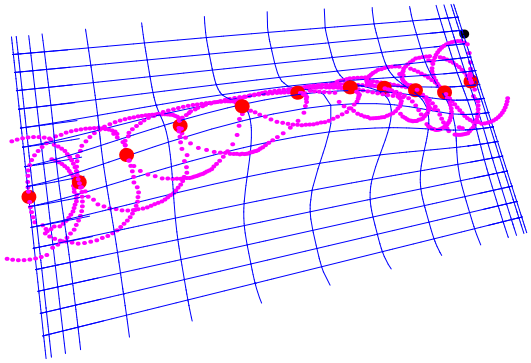

Figure 16: Tool path in a more curved region

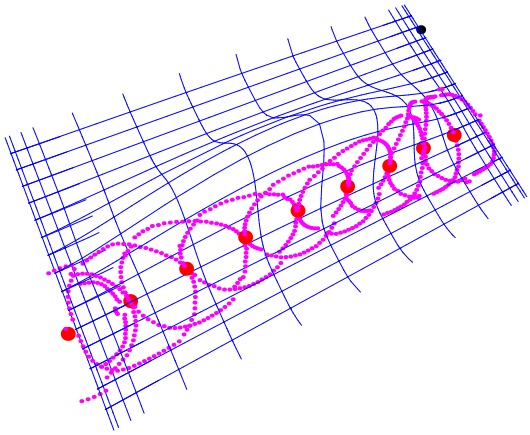

Figure 17: Tool path in a less curved region

The results in the examples correspond to expected ones; the direction choice optimizes the tool inclination angle for maximal material removal rate in each cutter contact point. The suggested locally optimal moving direction is depending on the surface curvatures at the actual point as well as on the tool's size.

The analysis of the variation of the angle between the tool axis and the surface normal shows the following values. On the torus with Gaussian curvature between -0.01 and 0.08 the change of this angle of the axis of the smaller tool (Fig. 13) along the path in the maximal curvature direction would be between 0.32 and 0.4, while along the path in the proposed direction between 0.28 and 0.4. The width of the processed patches has decreased by $\approx 0.16\%$. (Remember that this angle does not change when the tool is moving in the isophotic direction, but the change is maximal in the maximal normal curvature direction.) By our method in the case of the bigger tool (Fig. 14) the variation of the angle of the axis to the surface normal has been reduced even more, to the interval $[0.1, 0.34]$, and the width of the processed stripe has been reduced by $\approx 0.3\%$. On the trigonometric surface with Gaussian curvature between -0.001 and 0.0006 the variation of the observed angle along the steepest direction would be between 0.15 and 0.44, while along the proposed path between 0.0 and 0.4 (Figs. 16 and 17). The width's decrease on the processed patches is in this case 2%.

These values show that the variation of the angle between the tool axis (the direction of

which is fixed in three-axis milling) and the surface normal has been decreased while moving the tool along the proposed tool path. The loss in width of the processed stripe is not significant.

6. Conclusions

Geometric considerations about generating milling tool paths have been presented. The local moving direction of the tool has been computed, when the processed stripe is the widest, and the change of the inclination angle of the tool axis is minimal. A correction factor to the steepest ascending direction has been determined from the change of the isophotic angle of the surface normal with respect to the tool axis while moving the tool from a given point into the next position. This strategy provides to develop a new milling strategy with possible wide processed stripes and small variation of the angle between the tool axis and the surface normals. The presented new local offsetting method for triangular meshes solves basic problems, as computing the processed patch boundary and detecting self intersection or gauging. For the solution of other technical problems further investigations are necessary.

On analytic surfaces the computations and the figures have been made by the symbolic algebraic program package Mathematica. The program developed by the first-named author for the presented computations on discrete surfaces is not designed for professional applications. A lot of numerical problems may arise in the computation with real triangular meshes. But the solution of such problems was not the task of this work.

Acknowledgements

The work was supported by a joint project between the Technische Universität Berlin and the Budapest University of Technology and Economics (BUTE). In addition, the authors are grateful for the valuable remarks of the reviewers.

References

[1] Z.C. Chen, Z. Dong, G.W. Vickers: *Most efficient tool feed direction in three-axis CNC machining.* Integrated Manufacturing Systems **14**/7, 554–566 (2003).

[2] Z.C. Chen, G.W. Vickers, Z. Dong: *A new principle of CNC tool path planning for three-axis sculptured part machining – a steepest-ascending tool path.* J. Manufacturing Science and Engineering **126**, 515–523 (2004).

[3] C.M. Chuang, H.T. Yan: *A new approach to z-level contour machining of triangulated surface models using fillet endmills.* Computer-Aided Design **37**, 1039–1051 (2005).

[4] S. Ding, M.A. Mannan, A.N. Poo, D.C.H. Yang, Z. Han: *Adaptive iso-planar tool path generation for machining of free-form surfaces.* Computer-Aided Design **35**, 141–153 (2003).

[5] J. Dong, J.H. Kao, J.M. Pinilla, Y.C. Chang, F.B. Princ: *Automated planning for material shaping operations in additive/subtractive solid free form fabrication.* In Proc. Solid Freeform Fabrication Symposium, Univ. of Texas at Austin 1999, pp. 121–128.

[6] H.Y. Feng, H. Li: *Constant scallop-height tool path generation for three-axis sculptured surface machining.* Computer-Aided Design **34**, 647–654 (2002).

[7] G. Glaeser, J. Wallner, H. Pottman: *Collision-free 3-axis milling and selection of cutting tools*. Computer-Aided Design **31**, 225–232 (1999).

[8] Z. Han, D.C.H. Yang, J.J. Chuang: *Isophote-based ruled surface approximation of free-form surfaces and its application in NC machining*. Int. J. Production Research **39**, 1911–1930 (2001).

[9] C.S. Jun, D.S. Kim, S. Park: *A new curve-based approach to polyhedral machining*. Computer-Aided Design **34**, 379–389 (2002).

[10] T. Kim: *Constant cusp height tool paths as geodesic parallels on an abstract Riemannian manifold*. Computer-Aided Design **39**, 477–489 (2007).

[11] D.S. Kim, C.S. Jun, S. Park: *Tool path generation for clean-up machining by a curve-based approach*. Computer-Aided Design **37**, 967–973 (2005).

[12] S.J. Kim, M.Y. Yang: *A CL surface deformation approach for constant scallop height tool path generation from triangular mesh*. Int. J. Adv. Manuf. Technol. **28**, 314–320 (2006).

[13] S.J. Kim, M.Y. Yang: *Incomplete mesh offset for NC machining*. J. Materials Processing Technology **194**, 110–120 (2007).

[14] J. Lang: *Zur Konstruktion von Isophoten im Computer Aided Design*. CAD – Computergraphik und Konstruktion **34**, 1–7 (1984).

[15] S.C. Park: *Sculptured surface machining using triangular mesh slicing*. Computer-Aided Design **36**, 279–288 (2004).

[16] Y. Quinsat, L. Sabourin: *Optimal selection of machining direction for three-axis milling of sculptured parts*. Int. J. Adv. Manuf. Technol. **27**, 1132–1139 (2006).

[17] M. Szilvási-Nagy, Gy. Mátyási: *Analysis of STL files*. Mathematical and Computer Modelling **38**, 945–960 (2003).

[18] M. Szilvási-Nagy: *About curvatures on triangle meshes*. KoG **10**, 13–18 (2006).

[19] M. Szilvási-Nagy: *Face-based estimations of curvatures on triangle meshes*. J. Geometry Graphics **12**/1, 63–73 (2008).

[20] M. Szilvási-Nagy, Sz. Béla, Gy. Mátyási: *About the geometry of milling paths*. Annales Mathematicae et Informaticae **35**, 135–146 (2008).

[21] J. Wallner, G. Glaeser, H. Pottman: *Geometric contributions to 3-axis milling of sculptured surfaces*. In G. Olling, B. Choi, R. Jerard (eds.): *Machining Impossible Shapes,* Kluwer Academic Publ., Boston 1999, pp. 33–41.

[22] H.Y. Xu, H.Y. Tam, J.J. Zhang: *Isophote interpolation*. Computer-Aided Design **35**, 1337–1344 (2003).

[23] D.C.H. Yang, Z. Han: *Interference detection and optimal tool selection in 3-axis NC milling of free-form surfaces*. Computer-Aided Design **31**, 303–315 (1999).

[24] J.H. Yoon: *Fast tool path generation by the iso-scallop height method for ball-end milling of sculptured surfaces*. Int. J. Production Research **43**, 4989–4998 (2005).

Received December 10, 2012; final form October 10, 2013