Precise Gouging-free Tool Orientations for 5-Axis CNC Machining

Yong-Joon Kim[a], Gershon Elber[a], Michael Bartoň[b], Helmut Pottmann[c]

[a]*Department of Computer Science, Technion, Israel*
[b]*King Abdullah University of Science and Technology, Saudi Arabia*
[c]*Institute of Discrete Mathematics and Geometry, TU Wien, Austria*

Abstract

We present a precise approach to the generation of optimized collision-free and gouging-free tool paths for 5-axis CNC machining of freeform NURBS surfaces using flat-end and rounded-end (bull nose) tools having cylindrical shank. To achieve high approximation quality, we employ analysis of hyper-osculating circles (HOC) [26, 27], that have third order contact with the target surface, and lead to a locally collision-free configuration between the tool and the target surface. At locations where an HOC is not possible, we aim at a double tangential contact among the tool and the target surface, and use it as a bridge between the feasible HOC tool paths. We formulate all such possible two-contact configurations as systems of algebraic constraints and solve them. For all feasible HOCs and two-contact configurations, we perform a global optimization to find the tool path that maximizes the approximation quality of the machining, while being gouge-free and possibly satisfying constraints on the tool tilt and the tool acceleration. We demonstrate the effectiveness of our approach via several experimental results.

Keywords: free-form surfaces, curvature matching, hyper-osculating circles, third order approximation, collision and gouging detection

1. Introduction

Free-form NURBS surfaces are the de facto industry standard representation for 3D modeling and have been widely used in a variety of applications. Consequently, CNC machining of NURBS surface has been studied extensively in the past few decades. 5-axis machining, which provides two additional degrees of freedom, compared to 3-axis machining, attracted much attention due to the flexibility of 5-axis tool path planning, in contrast with the difficulties that 5-axis tool path generation entails.

There are two major technical challenges in 5-axis machining. One is how to optimally orient the tool so that it can approximate the local target surface properly. The other challenge is how to avoid collisions and gouging into the target surface, S, and other check surfaces, or other surfaces of the object. Collision and gouging[1] avoidance in 5-axis machining is more challenging compared to other applications, because the tool tip is typically in a tangential contact with S at the milling contact point, denoted P_C.

Consider 5-axis machining planning using a flat end cylindrical tool, T, having a bottom circle, C_T, as a cutting edge.[2] For each contact point P_C, two orientation
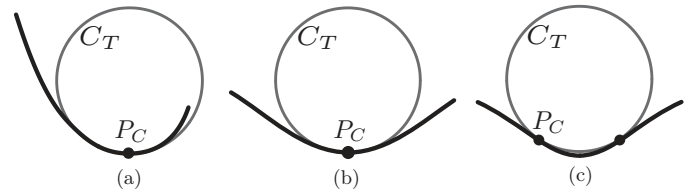

Figure 1: Osculating, hyper-osculating and two-contact circles: (a) The osculating circle at P_C having increasing curvature. (b) The HOC, located at curvature extreme point P_C of the curve, having a locally collision-free configuration. (c) The two-contact circles having two different tangential contacts.

degrees of freedom must be determined for T, while maintaining tangential contact between C_T and S at P_C. One natural approach to determine the orientation degrees of freedom uses a 2nd order approximation, to examine the osculating circle of a planar section of S using the plane through C_T. From Meusnier's theorem [4], and using the two orientation degrees of freedom, one can attempt to match the curvature of C_T and this planar section of S at P_C, which is also known as *curvature matched machining* [3, 18, 17, 19, 32, 33]. Curvature matched machining provides a 2nd order approximation and is a simple way to define the tool orientation, and it has been widely adapted in tool path computation. However and as a side effect of curvature matched machining, T is likely to locally gouge into S, near P_C, because the planar section curve of S typically has an increasing or decreasing curvature at P_C, while C_T presents a constant curvature (see Figure 1(a)).

Hyper-osculating circles (HOCs) [26, 27] alleviate this

---

[1]We distinguish two types of interferences between the cutting tool and the surface: the local one, denoted *gouging*, when the tool tip penetrates the surface in the local neighborhood of the contact point, and the global one, called *collision*, when the conflict occurs elsewhere on the tool.

[2]Path planning for a bull-nose or rounded tool having a cylindrical shank can be reduced to path planning for a flat end tool by offsetting the (target) surfaces by the rounding radius of the tool.

difficulty and vastly reduce the possibility of C_T gouging into S. Based on a 3rd order differential analysis, HOCs share the same position, tangent direction, curvature, and curvature derivative (which is zero for a circle), with S's planar section at P_C. In other words, an HOC is an osculating circle that is located at a curvature extreme point of the planar section of S and thus it resolves the local potential gouging of T into S (see Figures 1(b) and 5). However, this 3rd order approximation also has a limitation. Since the HOCs should satisfy both curvature and curvature derivative constraints, typically only a limited number of candidate configurations exist. As a consequence, a hyper-osculating configuration is not always feasible, and even less so when various constraints in practical situations are imposed, e.g., angular orientation limits on the CNC machine, global collisions, etc.

To help overcome this limitation of the HOCs, we also consider cases of two-tangential contacts between C_T and S (see Figure 1(c)). Henceforth and unless otherwise stated, 'contact' will denote a *tangential contact*. The key idea of our approach is based on the fact that there always exist two-contact configurations in the local neighborhood of hyper-osculating configurations[3]. From this observation, we can augment the HOCs and switch to nearby two-contact configurations whenever necessary, while providing good approximation quality in terms of the curvature difference between the bottom circle and the surface section at P_C and being gouging-free.

The rest of this paper is organized as follows. In Section 2, we briefly review previous related work. Section 3 introduces HOCs and establishes the algebraic conditions for two-contact configurations between T and S. Then, we propose a global optimization algorithm for the tool path that maximizes the approximation quality of the HOCs and the two-contact configurations. Experimental results are reported in Section 4, and the paper is concluded in Section 5.

2. Related Work

Related work on 5-axis milling mostly appeared in the CAD and mechanical engineering literature (see reference works [5, 23]). There are numerous papers contributing geometric methods to freeform surface CNC machining, including multi-axis (4- and 5-axis) machining. While this work focuses on contact analysis at the tip of the tool, in 5-axis machining, side or flank milling is one method that attracted much attention. See a recent survey in [14] on flank milling.

Numerous efforts were made to optimize the position of the tip of the tool, in 5-axis tool path planning (see, e.g., [6, 16, 20]). However, state-of the-art results have not yet resolved the fundamental question satisfactorily; namely, how to select the best tool size and design an optimal tool motion so that a given freeform surface can be machined

with given accuracy, without gouging and in a minimal amount of time. While the optimal tool orientation is the question that this work aims to resolve, the state-of-the-art of tool orientation is curvature matched machining [3, 18, 17, 19, 32, 33], which we already showed to be problematic as it is likely to cause small gouging of T into S, in the neighborhood of P_C. Multi point machining (MPM) [28, 30, 29] provides an alternative approach to compute an optimized tool orientation by positioning the tool in a manner that maximizes the number of contact points. The MPM method produces a considerable reduction in scallop heights and a wide machining strip width compared to the other conventional approaches [29]. However, the MPM tool path computation involves a non-linear optimization and finding a gouging-free solution is not always guaranteed.

Methods that aimed to select the best tool radius for the task are known. In [11], the surface is divided into regions that are accessible to different tool radii, by globally examining the curvature properties of the surface. That said, the tool radii being examined are predetermined. Recently, a scheme that reduces the machining time by segmenting the input surface into smaller sub-patches, called zones, was introduced [24]. Nevertheless, no explanation was given for where to segment the surface and an optimal size of the milling tool was not considered at all.

A general key question in planning of surface machining is the selection of a proper set of curves that form a *covering* for S, the surface to be machined. That is, when the tool traverses these covering curves, all locations of S are guaranteed to be visited to within some epsilon, by T. Tool path generation methods include contouring S by parallel planes, use of parallel (in the domain of S) isoparametric curves which can be chosen adaptively [12] and the use of space filling curves [1, 7]. See [10] for a survey on tool path generation. This work is focused on the optimal contact analysis between T and S along some given tool path and hence any synthesis method for tool path curves can be employed here.

Collision detection also plays a major role in 5-axis tool path planning. See [25, 31] for recent surveys on collision avoidance and simulations. While previous work includes configuration (C-) space analysis methods, among others, in this work, we perform the C-space analysis algebraically, gaining high precision.

Another optimality question that has recently been getting more attention is the aim for different tool shapes (beyond ball-end, flat-end and bull-nose tools). The reader is referred to [22] as an example for barrel-type tools. Examining the problem in reverse, an algorithm that looks for semi-kinematic parts of freeform objects was recently introduced in [2] in order to directly detect large segments that can be efficiently manufactured by a single sweep of a planar (tool) profile.

3. The Algorithm

Consider a C^3 continuous target surface $S(u, v)$ and let $S_u(u, v)$, $S_v(u, v)$ and $N(u, v)$ be the partial derivatives

---

[3]In fact, HOC is a singular case of a two-contact circle when the two-contact locations coalesce, as will be discussed later.
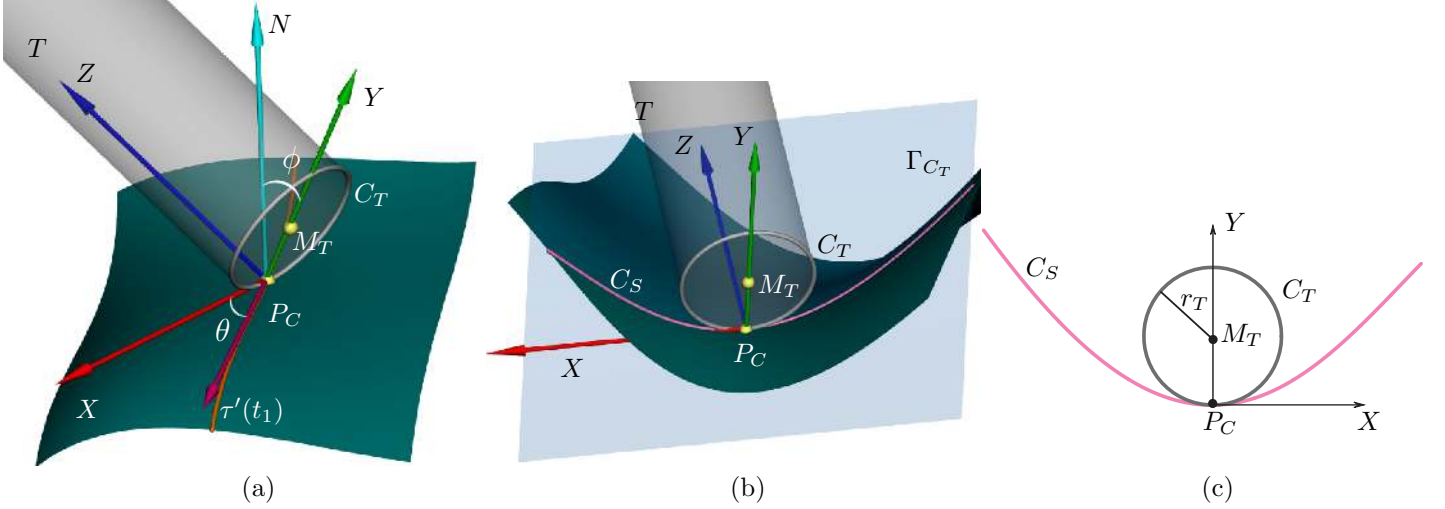
Figure 3: Coordinate system of T and a section of S, C_S, along the plane containing C_T, Γ_{C_T}: (a) The two degrees of freedom (θ, ϕ) for the orientation of T. (b) The surface section C_S on plane Γ_{C_T}. (c) The view in plane Γ_{C_T}.
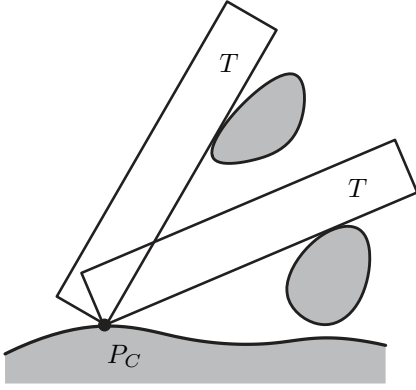

Figure 2: Two collision-free two-contact configurations.

and the normal field of S. Now consider one tool path in S, $\tau(t) = S(u(t), v(t))$, of a flat end tool T with a bottom circular edge C_T, and let $P_C = \tau(t_1) = S(u_1, v_1)$ be the current contact point. Further let, r_T be the radius of T, M_T the center point of C_T and Γ_{C_T} the plane containing C_T. Finally, let $C_S = S(u, v) \cap \Gamma_{C_T}$ be the intersection curve between plane Γ_{C_T} and $S(u, v)$; see Figure 3.

Recall that while we preserve the tangential contact between C_T and S at P_C, we also have two degrees of freedom to best adapt T to S by *rotating* and *tilting* the tool's position. We denote these two angular degrees of freedom by θ and ϕ, respectively. Consider an ortho-normal coordinate system (X, Y, Z) (see Figure 3(a)-(b)), of T, where X is the tangent direction of C_T at P_C, $Y = \frac{M_T - P_C}{||M_T - P_C||}$, and $Z = X \times Y$, where $\times$ denotes the cross-product of two vectors. Based on this coordinate system of T, θ is now defined as the angle between X and the tangent direction of the tool path, $\tau'(t_1)$, and similarly, the tilting angle ϕ is defined as the angle between Y and the unit normal vector N of S at P_C (see Figure 3). We limit the range of these angles to be $\theta \in [0, \pi]$ and $\phi \in [0, \frac{\pi}{2}]$.

The contact configurations between T and S form a

C-space obstacle boundary. Because the gouging of T into S always starts at a second event of contact, say at point $P_2 = S(u_2, v_2)$, this second contact point together with $P_C = S(u_1, v_1)$ defines a two-contact configuration. There can be several two-contact configurations (θ_i, ϕ_i), typically one of which is gouging-free. However, in exceptional cases one can have multiple collision-free two-contact configurations as is shown, for example, in Figure 2. In this work, we seek a collision-free two-contact configuration in the C-space of T, and hence stop after one is found. Figure 4(a) shows one example of a C-space obstacle boundary consisting of two-contact configurations (black curve), which divide the (θ, ϕ) domain into two regions, a collision-free and a gouging region.

This C-space obstacle boundary of T plays an important role in our algorithm. Consider a tilting degree of freedom ϕ of T for a fixed rotation angle θ^*. Following Meusnier's theorem [4] which states that the locus of osculating circles of a surface, passing through a given point P_C and sharing the same tangent line at P_C, form a sphere, the radius of curvature of C_S (Figure 3(b)-(c)) at P_C is given by:

$$r_p(\phi) = \frac{1}{k_n} \cos \phi. \tag{1}$$

Here k_n is the normal curvature of the surface for a fixed rotation angle θ^* and it must hold that $\frac{1}{k_n} > r_T$, since otherwise there is no gouging-free configuration of T, for any $\phi \in [0, \frac{\pi}{2}]$. Let ϕ^* be a tilting angle for which $r_p(\phi^*) = r_T$. In general, curvature matched configurations ($\phi = \phi^*$) lead to some penetrations of T into S and thus while tilting T from $\phi = 0$ to $\frac{\pi}{2}$, there exists a two-contact configuration on the C-space obstacle boundary at $\phi = \phi^t < \phi^*$, before it reaches the curvature matched configuration (see the red and yellow dots in Figure 4(a) and the circles in Figure 4(b) for fixed $\theta = \theta^*$, which shows a curvature matched C_T and a two-contact C_T, respectively). Since $r_p(\phi)$ is a monotonically decreasing function for $\phi \in [0, \frac{\pi}{2}]$,
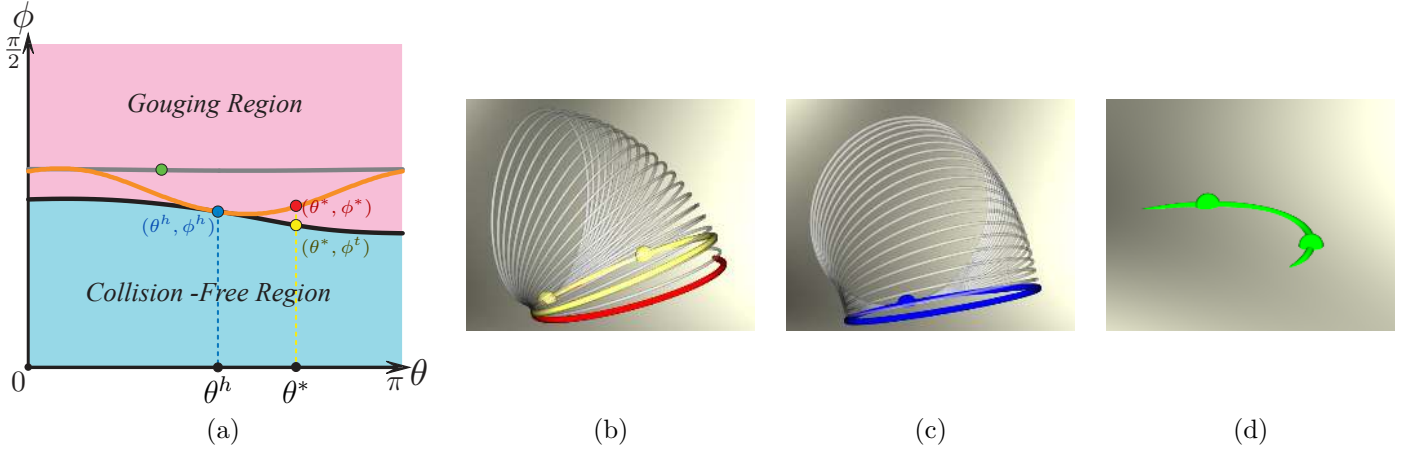
3

Figure 4: Two-contact configurations and HOCs: (a) Two-contact configurations (the black and gray curves) in the (θ, ϕ) domain and curvature matched configurations (orange curve). (b) The yellow circle shows a collision-free two-contact configuration (small yellow spheres) while the red circle presents a curvature matched configuration of C_T that gouges into S. (c) The blue circle shows a gouging-free hyper-osculating configuration of C_T. (d) The green circle shows a different two-contact configuration of C_T having inter-penetration into S elsewhere. Colors in (b),(c) and (d) correspond to the colors of the dots in (a). Dashed lines in (a) and transparent circles in (b),(c) both correspond to the tilting motions of T for fixed rotation angles θ^h, θ^*.

the difference in the radius of curvature, $||r_p(\phi) - r_T||$, is minimized while being gouging-free, when $\phi = \phi^t$. Consequently, the best collision free approximation in terms of curvature difference is realized at a two-contact configuration. On the other hand, when an HOC is feasible for a fixed rotation angle θ^h, the best collision free approximation, that is, a two-contact configuration on the C-space obstacle boundary, and the curvature matched configuration for this rotation angle θ^h coalesce, which leads us to the following lemma:

Lemma 1. *If a hyper-osculating circle (HOC) at (θ^h, ϕ^h) is a collision-free configuration, (θ^h, ϕ^h) is on the C-space obstacle boundary of T.*

Proof: From the definition of HOC, $r_p(\phi^h) = r_T$ and (θ^h, ϕ^h) is locally collision-free. Hence, for arbitrary small positive real value ϵ, $r_p(\phi^h + \epsilon) < r_T$, by Meusnier's theorem. If $r_p(\phi^h + \epsilon) < r_T$, T penetrates into S. Therefore, and due to the monotonicity of $r_p(\phi)$, (θ^h, ϕ^h) is on the C-space obstacle boundary of T. ∎

As is shown in Figure 4(a), the hyper-osculating configuration (blue dot) is realized on the tangential intersection between the two-contact configuration curve (black curve) and the curvature matched configuration curve (orange curve). This shows the known geometric fact that a hyper-osculating configuration is a special type of a two-contact configuration that has two coalescing contact points. Therefore, we can treat the HOCs as a special type of two-contact configurations and can perform a global optimization for the tool path on the combined set of configurations (done in Section 3.4).

Our algorithm consists of these two key contact analysis components, HOCs and two-contact configurations. In the rest of this section, we present details of our approach to compute these two-contact components, HOCs

in Sections 3.1 to 3.2 and two-contact configurations in Section 3.3.

3.1. Hyper-osculating circles (HOCs)

Recall that the HOC possesses 3rd order contact with S, i.e., the contact point P_C is a root of multiplicity *four*, considering the intersection of C_T and S as two algebraic manifolds. Whereas Meusnier's theorem gives rise to a one-parameter family of osculating circles of given radius at a general point, the hyper-osculating configuration additionally requires extremal curvature and cannot be expected to be achievable everywhere on the surface. The natural relaxation is to split this one quadruple contact point into a pair of double (tangential contact) points. The expansion of the space of admissible circles, as will be seen in Section 4, gives us sufficient freedom to cover S.

In this section, we perform 3rd order contact analysis to initialize the HOCs and again emphasize their benefit of not only having higher order contact but also of being locally "one sided" and hence locally gouging-free approximations (see Figure 5). This makes them perfect candidates for CNC machining purposes.

3.1.1. Higher order differential analysis of a surface

Given a non-umbilical point P_C and a tangent vector X of a surface $S(u, v)$, let D_1 and D_2 be the principal directions of S at P_C and α be the angle spanned by X and D_1 as is shown in Figure 6. There exists a *unique* section plane Γ such that the osculating circle of the intersection curve, $C_S = S \cap \Gamma$, has a 3rd order contact with S; see Figure 5 and [21], pp. 118-124, or alternatively [26].

Our computation follows [13], where the 3rd order invariants of a surface are classified and, using them, the intrinsic quantities, e.g., the derivative of the curvature, are computed. In particular, we exploit Lemma 11 in [13], where the derivative of the curvature with respect to the
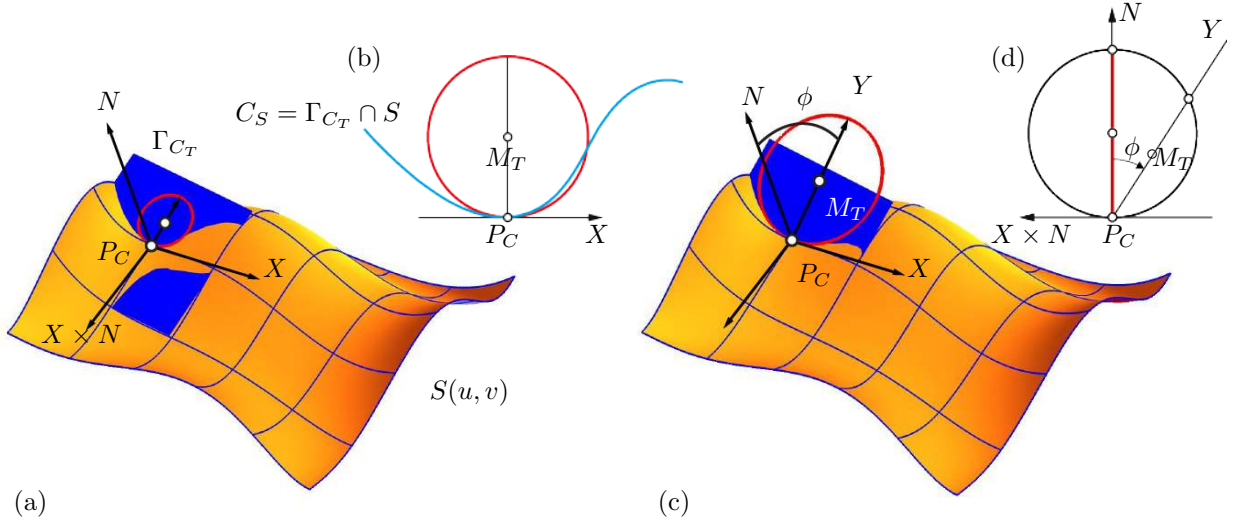
4

Figure 5: Hyper-Osculating vs. osculating circles: (a) Local fit of the surface $S(u,v)$ at point P_C by the osculating circle (red) of the intersection curve between S and plane Γ_{C_T}, $\{P_C, X\} \subset \Gamma_{C_T}$. (b) View in plane Γ_{C_T}; the osculating circle C_T locally penetrates C_S (and hence T penetrates S) near the contact point P_C. (c) The proper inclination of Γ_{C_T}, however, gives rise to the hyper-osculating configuration. Keeping the tangent direction X at P_C fixed, the position of the HOC (red) lying in Γ_{C_T} is uniquely determined by the tilting angle ϕ. (d) Once the hyper-osculating angle ϕ is computed from Equation (3), the center M_T of the hyper-osculating circle is found using Meusnier's theorem, in the Meusnier's sphere (in black).

arc-length of the intersection curve C_S is computed using the 3rd order invariants. We define

$$
\begin{aligned}
I_1 &= l_u + \langle 2S_{uu}, N_u \rangle, \\
I_2 &= l_v + \langle 2S_{uv}, N_u \rangle, \\
I_3 &= n_u + \langle 2S_{uv}N_v \rangle, \\
I_4 &= n_v + \langle 2S_{vv}N_v \rangle,
\end{aligned}
\tag{2}
$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, $l = \langle S_{uu}, N \rangle$, $n = \langle S_{vv}, N \rangle$ and N is the unit surface normal and subscripts represent the derivatives with respect to the corresponding parameter. Then, the curvature derivative with respect to arc-length s is

$$
\begin{aligned}
\frac{\mathrm{d}k}{\mathrm{d}s} = {} & \frac{1}{\cos\alpha}(I_1 \cos^3\alpha + 3I_2 \cos^2\alpha \sin\alpha + \\
& + 3I_3 \cos\alpha \sin^2\alpha + I_4 \sin^3\alpha + \\
& + 3\tan\phi(k_1 \cos^2\alpha + k_2 \sin^2\alpha)(k_2 - k_1)\cos\alpha \sin\alpha),
\end{aligned}
\tag{3}
$$

where k_1 and k_2 are the principal curvatures at non-umbilical point P_C, $(k_1 \neq k_2)$, and ϕ is the tilting (or Meusnier's) angle between the surface normal N and its orthogonal projection on plane Γ (see Figure 5).

The phenomenon of uniqueness of the plane Γ containing the HOC can also be observed from Equation (3), originally Equation (36) in [13], where for a fixed tangent direction X, the constraint on the extremal curvature ($\mathrm{d}k/\mathrm{d}s = 0$) is linear in $\tan\phi$ and gives a one-to-one mapping between the oriented Meusnier's angle ϕ and the value of $\mathrm{d}k/\mathrm{d}s$. We are interested in the planes (angles ϕ) where $\mathrm{d}k/\mathrm{d}s = 0$, i.e., C_S has the extremal curvature and the osculating circle becomes hyper-osculating.

3.2. The hyper-osculating radial function

At a given point P_C, using Equation (3), a univariate discrete *radial* function $r_{P_C}(\phi)$ is constructed, which assigns the value of the radius of the hyper-osculating circle
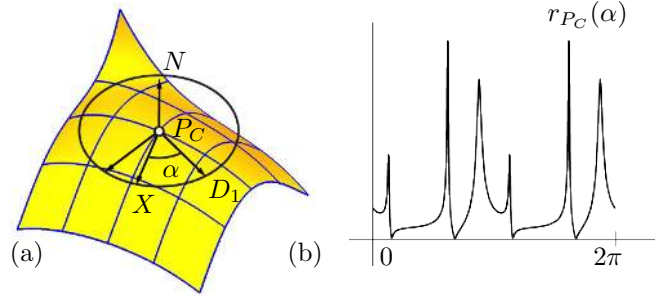

Figure 6: (a) The black circle corresponds to the set of unit tangent vectors of a surface $S(u,v)$ at an elliptical (convex) point P_C. The position of a tangent vector X is determined by α, the Euler angle between X and the principal direction D_1. (b) The radial function $r_{P_C}(\alpha)$, (see Equation (5)) assigns the value of the radius of the HOC to the tangent direction X, determined by the Euler angle α. The zeros of $r_{P_C}(\phi)$ correspond to the principal directions.

("hyper-osculating radius") to the tangent direction X, determined by the Euler angle α (see Figure 6). From Meusnier's theorem, the radial function $r_{P_C}(\alpha)$ is

$$
r_{P_C}(\alpha) = \frac{1}{k_n}\cos\phi = \frac{1}{(k_1 \cos^2\alpha + k_2 \sin^2\alpha)\sqrt{1 + \tan^2\phi}}.
\tag{4}
$$

Here, $\tan\phi$ is a function of α, computed from Equation (3) by requiring $\mathrm{d}k/\mathrm{d}s = 0$. In this way, we obtain the final formula for $r_{P_C}(\alpha)$,

$$
r_{P_C}(\alpha) = \frac{1}{(k_1 \cos^2\alpha + k_2 \sin^2\alpha)\sqrt{1 + F^2}},
\tag{5}
$$

where

$$
F = \frac{I_1 \cos^3\alpha + 3I_2 \cos^2\alpha \sin\alpha + 3I_3 \cos\alpha \sin^2\alpha + I_4 \sin^3\alpha}{3(k_1 \cos^2\alpha + k_2 \sin^2\alpha)(k_2 - k_1)\cos\alpha \sin\alpha}.
\tag{6}
$$

Assuming the tool radius r_T is given, we would like to identify HOCs for this radius, i.e., we have to solve
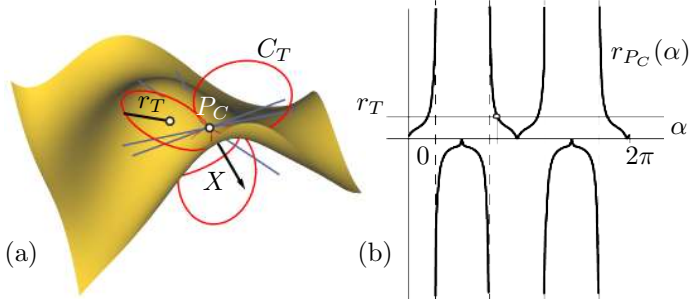
5

Figure 7: The reconstruction of the HOC from the radial function:
(a) Given the radius r_T, the feasible HOCs at P_C are obtained from
the radial function and Meusnier's theorem. (b) The signed radial
function $r_{P_C}(\alpha)$ at a hyperbolic point P_C. For a given radius r_T, the
hyper-osculating Euler angle α, associated with the hyper-osculating
tangent direction X, is obtained by solving $r_{P_C}(\alpha) = \pm r_T$. Two so-
lutions from above and two from below the surface are found for this
P_C. Note $r_{P_C}(\alpha)$ is a π-periodic function, hence eight intersection
points exist in $[0, 2\pi]$.

$r_{P_C}(\alpha) = \pm r_T$ for α, and obtain the (possibly multi-
ple) "hyper-osculating tangent directions" at point P_C (see
Figure 7).

Let us first see how many solutions we have to expect:
Rationalization of Equation (5) by substituting $\sin \alpha =
2t/(1 + t^2)$, $\cos \alpha = (1 - t^2)/(1 + t^2)$ yields that $r_{P_C}^2(t)$ is
a rational function of degree 16 over $\mathbb{R}$, which implies the
upper bound 8 on the number of hyper-osculating tan-
gent directions. Note that two different real numbers t
correspond to the same angle α. Since the degree of the
problem is greater than 4, we numerically solve constraint
$r_{P_C}(\alpha) = \pm r_T$.

3.3. Two-contact tool path

Two-contact configurations can be formulated as a sys-
tem of algebraic constraints and be solved. The solu-
tions of these algebraic equations may contain invalid so-
lutions that cause gouging elsewhere into S (recall Fig-
ure 4(d)). Here, we present the algebraic constraints of
the two-contact configurations and introduce an efficient
and accurate collision detection algorithm that facilitates
the robust elimination of these invalid solutions.

3.3.1. Algebraic constraints

For a given target surface $S(u, v)$ and a fixed con-
tact point P_C, we need to find a second contact point
$P_2 = S(u_2, v_2)$ and a tool orientation which is determined
by rotating and tilting angles θ and ϕ. Recall the ortho-
normal coordinate system of T, (X, Y, Z) that is shown
in Figure 3 (a). The X, Y, Z axes can be parameterized
using θ and ϕ, as $X(\theta, \phi), Y(\theta, \phi), Z(\theta, \phi)$. We also pa-
rameterize the center of T as $M_T(\theta, \phi) = r_T Y(\theta, \phi) + P_C$.
Depending on the location of P_2 in the different regions of
T (the bottom disk of T, the shank of T, or the C^1 discon-
tinuity between them – along C_T), we have three different
algebraic conditions to handle:

1. A first condition, which is the most typical case, is
   realized when P_2 is also located on C_T (Figure 8(a)).

We can formulate this condition using the following
algebraic constraints:

$$||S(u_2, v_2) - M_T(\theta, \phi)||^2 - r_T^2 = 0, \tag{7}$$

$$\langle (S(u_2, v_2) - P_C), Z(\theta, \phi) \rangle = 0, \tag{8}$$

$$\langle (S(u_2, v_2) - M_T(\theta, \phi)) \times Z(\theta, \phi)), N(u_2, v_2) \rangle = 0, \tag{9}$$

with u_2, v_2, θ, ϕ as unknowns.

Equations (7) and (8) prescribe the condition for $P_2 =
S(u_2, v_2)$ to be on C_T. Then, $(S(u, v) - M_T(\theta, \phi)) \times
Z(\theta, \phi)$ in Equation (9) denotes the tangent direction
of the circle at $S(u_2, v_2)$ and Equation (9) makes sure
that the tool tip is in contact with S at $S(u_2, v_2)$ as
well.

2. A second condition is realized when the second con-
   tact point P_2 is located on the bottom disk of T, as
   is shown in Figure 8(b). The algebraic constraints for
   this condition are as follows:

$$\langle (S_u(u_2, v_2), Z(\theta, \phi) \rangle = 0, \tag{10}$$

$$\langle (S_v(u_2, v_2), Z(\theta, \phi) \rangle = 0, \tag{11}$$

$$\langle (S(u_2, v_2) - P_C), Z(\theta, \phi) \rangle = 0, \tag{12}$$

with u_2, v_2, θ, ϕ as unknowns.

Equations (10) and (11) force the bottom disk of T
to be tangential to S. Then, Equation (12) locates
$P_2 = S(u_2, v_2)$ on the plane containing the bottom
disk of T.

3. $P_2 = S(u_2, v_2)$ can also be located on the shank of
   T (Figure 8(c)), which leads us to the last condition
   for the two-contact configuration. This last condition
   can be formulated as follows:

$$\langle N(u_2, v_2), Z(\theta, \phi) \rangle = 0, \tag{13}$$

$$\langle (S(u_2, v_2) - M_T(\theta, \phi)) \times N(u_2, v_2), Z(\theta, \phi) \rangle = 0, \tag{14}$$

$$||S(u_2, v_2) - M_T(\theta, \phi)||^2 -
|| \langle S(u_2, v_2) - M_T(\theta, \phi), Z(\theta, \phi) \rangle ||^2 - r_T^2 = 0, \tag{15}$$

with u_2, v_2, θ, ϕ as unknowns.

Equations (13) and (14) force a *bi-normal condition*,
where the two normals of the two surfaces in hand,
T and S, are collinear. Equation (15) constrains the
distance between S and the axis of T to be r_T so that
P_2 is on the shank of T.

3.3.2. Collision detection

Not all the two-contact configurations computed in Sec-
tion 3.3.1 are valid. There are invalid solutions that cause
gouging of T into S elsewhere. Recall Figure 4(d) that
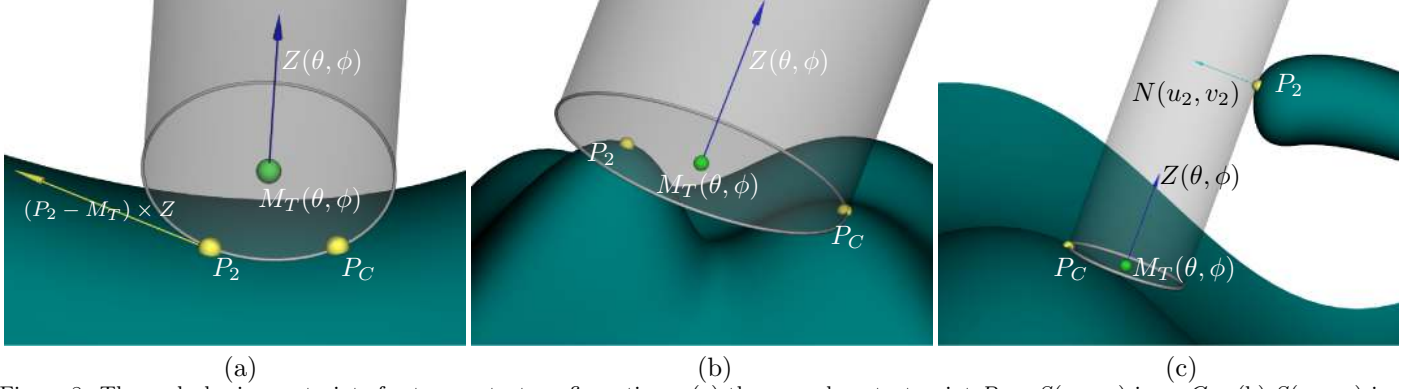
Figure 8: Three algebraic constraints for two-contact configurations: (a) the second contact point $P_2 = S(u_2, v_2)$ is on C_T, (b) $S(u_2, v_2)$ is on the bottom disk of T, and (c) $S(u_2, v_2)$ is on the shank of T.
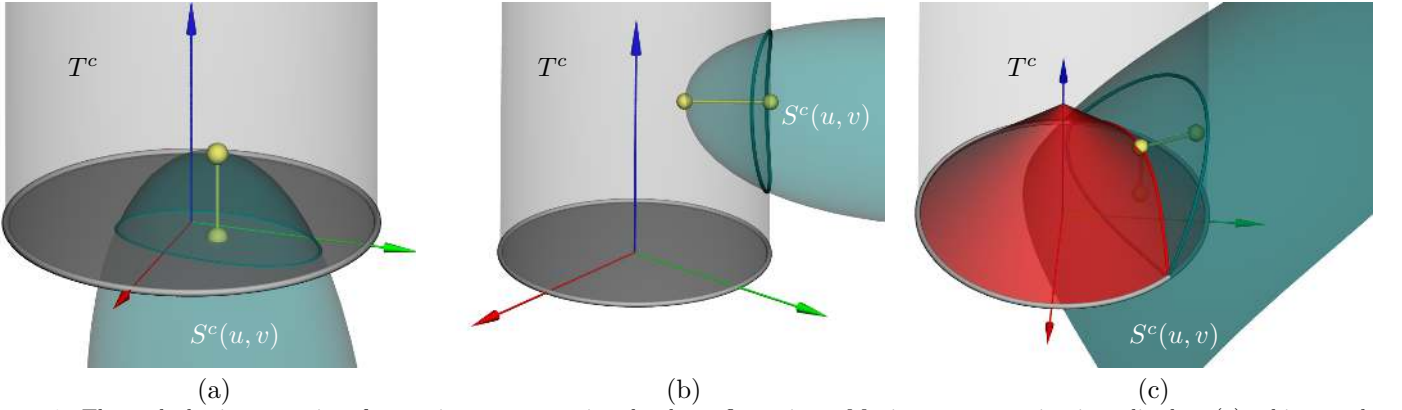

Figure 9: Three algebraic constraints for maximum penetration depth configurations: Maximum penetration is realized at (a) a bi-normal location with the bottom disk of T^c, (b) a bi-normal with the shank of T^c, or (c) an intersection with the self-bisector of T^c with S^c.

showed an example of such invalid two-contact configurations. To eliminate these invalid solutions, we perform some collision detection tests between T and S, a nontrivial task because $C_T \subset T$ is already in contact with S. Furthermore, these contact configurations might cause small gouging(s) of T into S, for which a simple point sampling approach to collision detection cannot guarantee a gouging-free answer. To address this issue, we precisely compute the maximum penetration depth of T into S. The conditions for the maximum penetration can also be formulated as a set of algebraic equations, which are presented in this section. By solving these algebraic constraints we can compute the precise penetration depth, enabling a robust and accurate detection of both global collision and local gouging.

Similar to the two-contact configurations analysis, there are three cases for the maximum penetration depth to be considered. We perform the collision detection test for a fixed configuration of T, in the coordinate system of T (Figure 3), where $X(\theta, \phi), Y(\theta, \phi), Z(\theta, \phi)$ becomes our x, y, z coordinate system and the origin is at $P_C(\theta, \phi)$. The geometry in this (transformed) canonical space will be denoted by superscript c. For example, $(S_x^c(u,v), S_y^c(u,v), S_z^c(u,v))$ and $(N_x^c(u,v), N_y^c(u,v), N_z^c(u,v))$ denote the coefficients of S and N in the canonical space, respectively.

For two smooth surface regions, the maximum penetration depth is realized at the bi-normal events. Herein,

the two surfaces are T and S (see Figure 9) and hence these bi-normal locations can be located as follows:

1. If S^c has a bi-normal location with the bottom disk of T^c, this event can be formulated as follows:

$$N_x^c(u,v) = 0, \qquad (16)$$
$$N_y^c(u,v) = 0, \qquad (17)$$

seeking vertical normals.

2. If the bi-normal event occurs on the shank of T^c, this event can be formulated as follows:

$$N_z^c(u,v) = 0, \qquad (18)$$
$$N_y^c(u,v)S_x^c(u,v) - N_x^c(u,v)S_y^c(u,v) = 0. \qquad (19)$$

3. Finally, the extrema can also occur at C^1 discontinuities or herein along the intersection curve between S^c and the self-bisector of T^c that emanates from C_T^c. While the self-bisector of a general freeform surface can be quite difficult to compute, herein T^c is a flat end tool and has a simple self-bisector emanating from C_T^c in the shape of a cone (see Figure 9(c)). The constraints identifying this last case are as follows:

$$S_x^c(u,v)^2 + S_y^c(u,v)^2 - (r_T^c - S_z^c(u,v))^2 = 0, \quad (20)$$
$$N_y^c(u,v)S_x^c(u,v) - N_x^c(u,v)S_y^c(u,v) = 0. \quad (21)$$

Equation (20) prescribes the intersection condition between the target surface and the self-bisector of the tool, which can be represented by the implicit equation $x^2 + y^2 - (r_T - z)^2 = 0$. Then, Equation (21) forces the tangent direction of this intersection curve to be orthogonal to the (canonical) z-axis so that the solution maximizes the penetration depth into S^c.

3.4. Global Optimization

For a given tool path $\tau(t) = S(u(t), v(t))$, $0 \leq t \leq 1$, we discretely compute the optimized orientations $O(t) = (\theta(t), \phi(t))$ for $\tau(t)$. As a pre-processing step of this discrete optimization, we compute collision-free two-contact configurations and HOCs using the algebraic constraints and the collision detection algorithms that were presented in the previous sections. We sample at $t = t_i$, $i = 1, 2, ..., n$ along the tool path and similarly sample along rotation angle $\theta = \theta_j$, $j = 1, 2, ..., m$. For each such (t_i, θ_j) pair, we compute $\phi = \phi_k$ that are collision-free two-contact configurations. For each t_i we also find a finite number of HOCs at (θ_i^h, ϕ_i^h). These collision-free two-contact and HOC configurations create (samples of) two/one-dimensional manifolds in the three dimensional space of (θ, ϕ, t). We accelerate this pre-computation, which is the main bottleneck of our algorithm, by taking advantage of the coherence of the solution set. Before solving the two-contact and HOC configurations, we numerically improve the previously computed adjacent solution, and then check if the numerically improved solution is collision free. If the solution is collision free we can use it as a solution, because we assume that there exist only one collision-free two-contact or HOC.

When there is neither a two-contact configuration nor an HOC for (t_i, θ_j), domain (t_i, θ_j, ϕ), $0 \leq \phi \leq \pi$ is either entirely collision-free or completely gouging. We distinguish between the two cases simply by conducting a single collision detection test at one point in the domain, say at $(t_i, \theta_j, \frac{\pi}{2})$. If we detect a collision we void this configuration completely and if no collision is found, we set ϕ so that T and S have the best geometric matching. If P_C is an elliptic (convex) point, we set $\phi = \frac{\pi}{2}$, aligning T along S's normal N, which is the best curvature matching we can hope for, for a convex location.

We treat these two-contact configurations and HOCs as vertices in a directional graph in (θ, ϕ, t) space. The directional edges for the graph are defined by only connecting from $(\theta_{j_1}, \phi_{k_1}, t_i)$ to $(\theta_{j_2}, \phi_{k_2}, t_{i+1})$ if the angular distance is within some given tolerance. The weight of each edge reflects on the curvature matching quality, measuring the difference $||r_{C_S}(\theta, \phi, t) - r_T||$, where r_{C_S} is a radius of curvature of C_S at P_C.

Given this graph structure, we perform a Dijkstra [9] search to find the shortest path between some start $(t = t_1)$ and terminal $(t = t_n)$ vertices. By weighing the edges in the graph following $||r_{C_S}(\theta, \phi, t) - r_T||$, the shortest path globally minimizes the total curvature difference in the tool path. One can also easily embed various additional constraints into the optimization, by simply removing the vertices or edges from the graph that violate the constraint.

4. Experimental Result

In this section, we present results of using the tool contact planning algorithms on several free-form surfaces. Figure 10 illustrates the limitations when only HOCs are considered. In order to fairly evaluate the potential of HOCs, we computed optimized contact paths with a level set approach (see figure caption). Even then, the coverage of the surface by HOC strips is too low.

In the coming examples, the optimization algorithm computes 200 samples along the tool path and 200 samples along θ, finding the best tilting angle, ϕ_k, for each such (t_i, θ_j) pair and HOCs, (θ_i^h, ϕ_i^h), for each t_i. All solutions for the HOCs and the two-contact configurations are computed with high numerical accuracy (10^{-8} in a unit-size parameter space), preventing possible tool axis oscillations due to numerical errors. Figure 13 (a) shows one example of this (t_i, θ_j) graph with ϕ_k as the elevations. The admissible HOCs are also shown Figure 13 (a) in blue. In Figure 13 (b), the optimal path, computed using the Dijkstra search over this graph and taking into account collision detection and constraints etc., is presented in green.

The graph search example shown in Figure 13 is for the cyan tool path shown in Figures 11 (a) and (b). Figure 11 (a) shows the tool path used for this biquintic surface, where 99 isoparametric curves are used as the tool path. Figure 11 (b) shows several tool positions along the cyan curve in (a) where a blue P_C denotes an HOC contact and a green P_C depicts a two-contact case with P_2 as the second contact location in magenta. Figures 11 (c) presents the resulting machined surface, simulated using Moduleworks' (www.moduleworks.com) 5-axis machining simulator.

To gain some insight into the difference between the 3rd order tool contact orientation approach proposed here in comparison to curvature-matched machining, we provide Figure 15. The color coding in this figure hints on gouging (in red) and on under cuts (in blue), while green hints on the precise surface location. As can be seen in Figure 15, curvature matched machining gouges into the surface, whereas the C^3 approach presented in this work stays in the 'green'. The flat-end tool used here is 5% of the maximum edge-length of the bounding box.

Figure 16 presents two additional surfaces that were tested using the presented algorithms under similar conditions (number of tool path curves and samples in the curves). Then, Table 1 shows statistics for all three surfaces, results that are fairly consistent among all three surfaces.

A careful inspection of Figure 11 (c) and Figure 15 (a) reveals a small zone where the machining quality is not in the green level. We now discuss the reason for this and a possible remedy. Figure 12 (a) shows a zoom-in on the problem. In Figure 12 (b), the reason is depicted. Since we allow θ to vary from zero to π, it can clearly happen that at two adjacent tool contact locations, on two adjacent tool paths, one P_C will present $\theta = 0$ and the other P_C will have $\theta = \pi$. As a direct result, the region between the
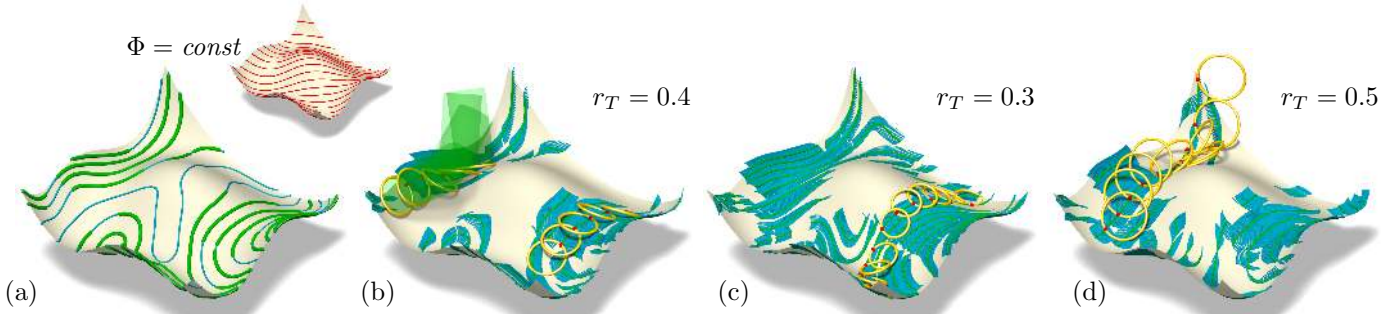
Figure 10: Limitations of machining with only hyper-osculating tool contact. (a) Aiming at tool contact paths with hyperosculating tool bottom, we developed an algorithm where these paths are represented as level sets of a function Φ defined on the (u, v) parameter domain of S. Initialized by a linear function Φ (red curves), we optimize Φ so that the HOC's contact direction is as perpendicular as possible to the path tangent direction. This leads to the blue curves. In green we show those contact path segments whose tool position is both gouging free and collision free. (b-d) The coverage of the reference surface by strips consisting of only hyper-osculating arcs (cyan) for various radii is incomplete.
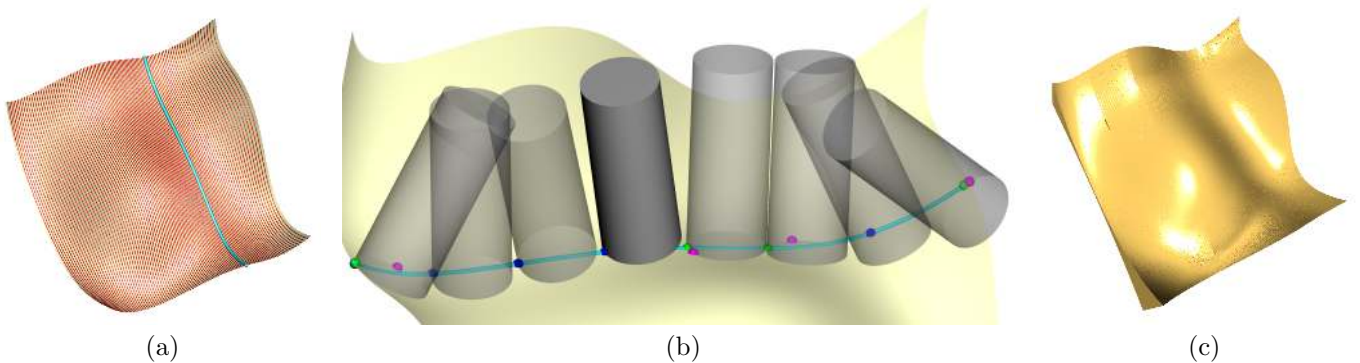
Figure 11: Simulation results for the optimized tool paths. (a) S, a biquintic B-spline surface with (6×6) control points and its tool paths, $\tau(t)$, in red. (b) A few tool positions and orientations along the cyan tool path in (a). HOC contacts are in blue, and two-contacts are in green (P_C) and magenta (P_2). (c) Result of the entire machining simulation.

two adjacent tool paths is not machined! To remedy this, one can limit θ to a smaller domain, avoiding the problem as is shown in Figures 12 (c) and (d).

Because the HOCs computation, in Section 3.1.1, requires a third order analysis, we assume that the input surface has at least C^3 continuity. Working with NURBS surfaces, when the input surface is not C^3, the algorithm is applied independently to each polynomial patch only to connect the paths via the optimization algorithm introduced in 3.4. Figure 14(a) shows the discontinuous HOC paths for a bicubic B-spline surface and Figure 14(b) shows the result of the optimization algorithm which combines separated HOC paths and two-contact configurations.

5. Conclusion and Future Work

The proposed optimization algorithm naturally combines HOCs and two-contact configurations by treating the hyper-osculating case as a special type of the two-contact configuration, and by simply minimizing the curvature difference between T and S. The tool path orientations generated by the global optimization procedure tend to maintain the hyper-osculating configurations when it is accessible, and smoothly switch to nearby two-contact configurations if there is no hyper-osculating position or it is not accessible due to some constraints. All this is done while precisely analyzing and avoiding any gouging.

The assumption was, while computing HOCs, that P_C is not an umbilical point. While umbilical points are typically isolated, even if S contains umbilical points, the two-contact solution can always be used in the neighborhood of the umbilical location(s) and hence, no real difficulty is posed by umbilical locations.

While this work assumed a cylindrical shank for T, the extension of this work to handle conical tools is quite simple. For the most part, it amounts to handling two-contacts over conical shanks. One has to add support for possible two-contact with the conical shanks, extending Equations (13) to (15), and add support for collision detection for conical shanks, extending Equations (18) and (19).

The computation cost of the presented tool paths is high and can be in hours for a whole surface S. Improvements of the performance of the presented algorithms must be explored as it will clearly ease their acceptance and use.

Even though we sample the tool path, we can easily interpolate the two adjacent solutions by representing them as unit quaternions [8]. However, the advantage of using quaternions in this context is yet to be explored.

While we ensure gouging-free arrangements at the sampled points, we can guarantee very little in between samples. The continuous collision detection processing of tool path $\tau(t)$, and the continuous collision detection/tests remain a challenging task, even though some work on this continuous processing has already been done, for example in [15].
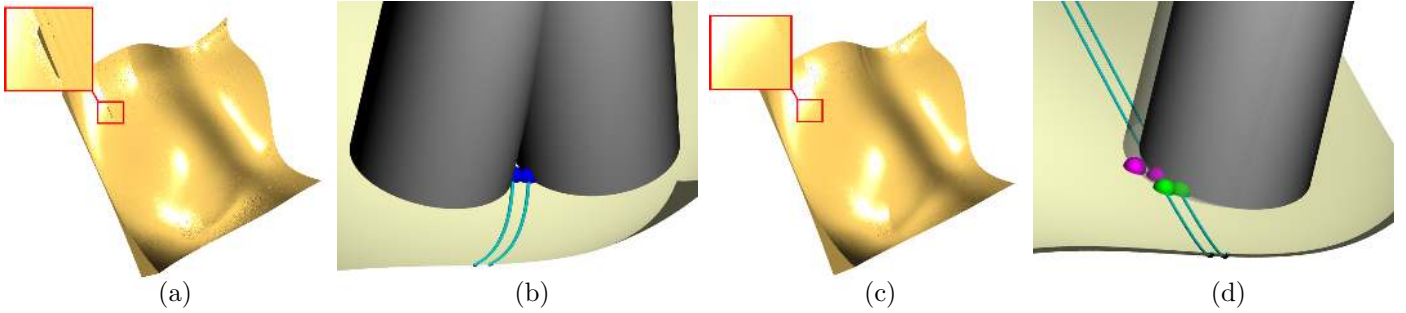
Figure 12: The surface from Figure 11 has a small problem that is shown in (a). The reason for this problem can be found in adjacent tool paths that present completely opposite θ orientations (b), resulting in a zone between the two adjacent positioned that is not machined. By limiting the range of θ, this problem can be avoided, as shown in (c) and (d).

Example	HOC (Secs)	Collision (Secs)	Solver (Secs)	Total (Secs)	Optimization (Secs)	Memory (MB)	Max. Gouging (Rel.)
Fig. 11	1.88	71.38	321.52	562.52	15.97	68.184	2.03×10^{-11}
Fig. 16 (a)	2.07	80.81	27 4.25	494.84	20.89	67.193	2.16×10^{-11}
Fig. 16 (b)	2.67	73.84	463.94	591.14	20.56	54.784	4.39×10^{-11}

Table 1: Timing (in seconds), memory consumption (in Megabytes) and maximum gouging with respect to the maximum edge-length of the bounding box for the examples in Figures 11 and 16, for one tool path curve with 200 sampled points.

In the present work, we considered the tool paths as input. Computing these paths as part of an optimization algorithm is left as a topic for future research. Finally, being aware of the fast progress of engineering abilities, considering a machining tool that can adaptively modify its milling radius and shape introduces an interesting area for research.

6. Acknowledgements

[1] W. Anotaipaiboona and S. S. Makhanovb. Curvilinear space-filling curves for five-axis machining. *Computer Aided Design*, 40:350–367, 2008.

[2] M. Bartoň, H. Pottmann, and J. Wallner. Detection and reconstruction of freeform sweeps. *Computer Graphics Forum*, 33(2):23–32, 2014.

[3] S. Bedi, S. Grevelle, and Y. H. Chen. Principal curvature alignment technique for machining complex surfaces. *Journal of Manufacturing Science and Engineering*, 119(4):756–765, 1997.

[4] M.P. Do Carmo. *Differntial geometry of curves and surfaces.* Prentice-Hall, 1976.

[5] B. K. Choi and R. B. Jerard. *Sculptured Surface Machining: Theory and Applications.* Kluwer, 1998.

[6] B. K. Choi, J. W. Park, and C. S. Jun. Cutter-location data optimization in 5-axis surface machining. *Computer Aided Design*, 25(6):377–386, 1993.

[7] J. J. Cox, Y. Takezaki, H. R. P. Ferguson, K. E. Kohkonen, and E. L. Mulkay. Space-filling curves in tool-path applications. *Computer Aided Design*, 26(4):215–224, 1994.

[8] Erik B. Dam, Martin Koch, and Martin Lillholm. Quaternions, interpolation and animation. Technical report, 1998.

[9] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Math*, 1:269–271, 1959.

[10] D. Dragomatz and S. Mann. A classified bibliography of literature on NC tool path generatio. *Computer Aided Design*, 29(4):239–247, 1997.

[11] G. Elber and E. Cohen. Second order surface analysis using hybrid symbolic and numeric operators. *ACM Trans. on Graphics*, 12(2):160–178, 1993.

[12] G. Elber and E. Cohen. Tool path generation for freeform surface models. *Computer Aided Design*, 26(7):490–496, 1994.

[13] J. Gravesen and M. Ungstrup. Constructing invariant fairness measures for surfaces. *Advances in Computational Mathematics*, 17:67–88, 2002.

[14] R. F. Harik, H. Gong, and A. Bernard. 5-axis flank milling: A state-of-the-art review. *Computer Aided Design,*, 45(3):796–808, 2013.

[15] O. Ilushin, G. Elber, D. Halperin, and R. Wein. Precise global collision detection in multi-axis NC-machining. *Computer Aided Design*, 37(9):233–242, 2005.

[16] S. A. Ivanenko, S. S. Makhanov, and M. A. Munlin. New numerical algorithms to optimize cutting operations of a five-axis milling machine. *Appl. Numer. Math.*, 49(3–4):395–413, 2004.

[17] C. G. Jensen. Analysis and synthesis of multi-axis sculptured surface machining. Dissertation, Purdue University, 1993.

[18] C. G. Jensen and D. C. Anderson. Accurate tool placement and orientation for finish surface machining. In *Proc. Symposium on Concurrent Engineering*, pages 127–145, 1993.

[19] C. G. Jensen, W. E. Red, and C. Ernst. Machining free-form surface cavities using a combination of traditional and non-traditional multi-axis machining methods. *Computer Aided Design and Applications*, 5:241–253, 2008.

[20] C. S. Jun, K. Cha, and Y. S. Lee. Optimizing tool orientations for 5-axis machining by configuration-space search method. *Computer Aided Design*, 35(6):549–566, 2003.

[21] E. Kruppa. *Analytische und konstruktive Differentialgeometrie.* Springer, Wien, 1957.

[22] C. Zhitong M. Fanjun, X. Rufeng, and L. Xun. Optimal barrel cutter selection for the CNC machining of blisk. *Comput. Aided Design. To appear.*, 2014.

[23] G. J. Olling, B. K. Choi, and R. B. Jerard. *Machining Impossible Shapes.* Kluwer, Boston, 1999.

[24] J. Senatore, S. Segonds, W. Rubio, and G. Dessein. Correlation between machining direction, cutter geometry and stepover distance in 3-axis milling: Application to milling by zones. *Computer Aided Design*, 44:1151–1160, 2012.

[25] T. D. Tang. Algorithms for collision detection and avoidance for five-axis NC machining: A state of the art review. *Computer Aided Design*, 51(1):1–17, 2014.

[26] X. C. Wang, S. K. Ghosh, Y .B. Li, and X. T. Wu. Curvature catering – a new approach in manufacture of sculptured surfaces (part 1. theorem). *Journal of Materials Processing Technology*, 38(1–2):159–175, 1993.
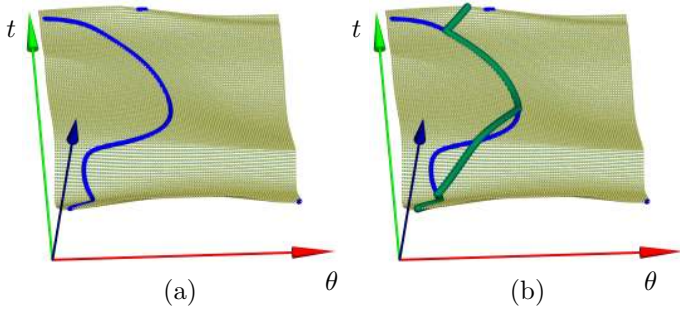
Figure 13: Dijkstra optimization over samples in (t_i, θ_j) space. In (a), the samples are shown in yellow along with the HOCs in blue. In (b), the optimal result, following Dijkstra, is presented in green. The bivariate scalar function (yellow) over the (t, θ) domain displays the computed values of the tilting angle ϕ.
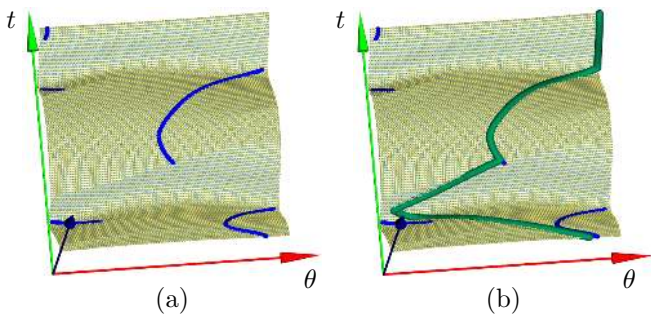

Figure 14: Dijkstra optimization result for a C^2 NURBS surface. (a) Shows discontinuous HOC paths for a bicubic B-spline surface and (b) presents the optimal result that combines separated HOC paths and two-contact configurations.
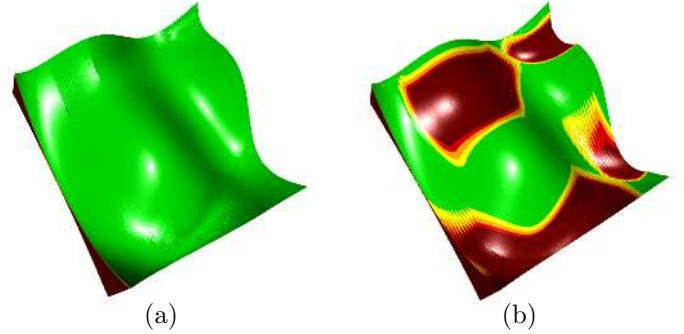

Figure 15: Quality comparison of the tool path, using the 5-axis simulator of Moduleworks (www.moduleworks.com). The colors of red, denote gouging (inside S), to green, that depict being on the surface, to blue, that hint on under-cuts. (a) Presents the result of using HOCs/two-contacts, (b) curvature matched machining. Full red color represents an over-cut (gouging) by 1% of the maximum edge-length of the bounding box. The flat-end tool used here is 5% of the maximum edge-length of the bounding box.
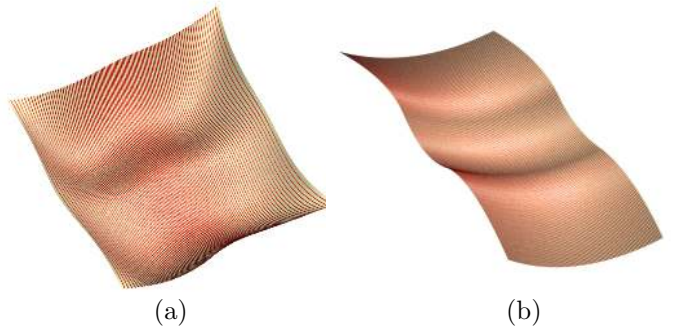
[27] X. C. Wang, S. K. Ghosh, Y .B. Li, and X. T. Wu. Curvature catering – a new approach in manufacture of sculptured surfaces (part 2. methodology). *Journal of Materials Processing Technology*, 38(1–2):177–193, 1993.

[28] A. Warkentin, F. Ismail, and S. Bedi. Intersection approach to multi-point machining of sculptured surfaces. *Computer Aided Geometric Design*, 15(6):567–584, 1998.

[29] A. Warkentin, F. Ismail, and S. Bedi. Comparison between multi-point and other 5-axis tool positioning strategies. *Int. Jounal of Machine Tools & Manufacture*, 40:185–208, 2000.

[30] A. Warkentin, F. Ismail, and S. Bedi. Multi-point tool positioning strategy for 5-axis mashining of sculptured surfaces. *Computer Aided Geometric Design*, 17(1):83–100, 2000.

[31] X. Xu Y. Zhang and Y. Liu. Numerical control machining simulation: a comprehensive survey. *Int. J. Computer Integrated Manufacturing*, 24(7):593–609, 2011.

[32] J. H. Yoon. Tool tip gouging avoidance and optimal tool positioning for 5-axis sculptured surface machining. *Intl. J. Production Research*, 41:2125–2142, 2003.

[33] J. H. Yoon, H. Pottmann, and Y. S. Lee. Locally optimal cutting positions for five-axis sculptured surface machining. *Computer Aided Design*, 35:69–81, 2003.

Figure 16: Two additional surfaces that where tested using the presented 5-axis contact algorithms: A bicubic B-spline surface with (a) (5×5) control points and (b) (4×8) control points.