Title:
A GPU Method and error analysis for Multi-Point Positioning of a Toroidal Tool for 5-Axis Machining

Authors:
Mukhmeet Singh, mukhmit.singh@gmail.com, University of Waterloo
Sanjeev Bedi, sbedi@uwaterloo,ca, University of Waterloo
Stephen Mann, smann@uwaterloo.ca, University of Waterloo

Keywords:
CNC Machining, 5-Axis Machining, Tool Positioning, GPU, Error Analysis

DOI: 10.14733/cadconfP.2023.206-210

Introduction:
The placement of the tool over the surface for 5-axis machining is challenging as it must be placed to avoid gouging or overcutting. The tool and modeled surface involve geometry that results in complex simultaneous transcendental equations that must be solved to get gouge-free placement. In Multipoint machining methods (MPM) the toroidal tool touches the modeled surface (to be machined) at multiple points of contact. There are number of methods to find MPM toolpaths [1, 2, 3, 4]. All these methods involve numerical methods and use the CPU to solve the non-linear transcendental-polynomial equations. These method of obtaining the solutions for positioning the tool over the surface are time consuming [2]. In addition, the solution results in multiple answers that satisfy the equations, but some of them gouge the surface (at other points on the tool) and necessitate the use of a gouge checker to isolate the gouge free solution.

The method proposed in this paper is based on a graphical model and does not require the solution of simultaneous equations or initial guesses. The new method emulates the tool dropping on a surface [3]. The drop distance is calculated using a GPU based algorithm. The method presented here evolved from an earlier work, where the earlier method was used for gouge checking (a minor role in tool positioning). This gouge checking method is modified to speed up its computational efficiency and to produce 3-axis tool positions. In addition, the GPU based tool drop method is repeatedly used with a rotation algorithm to obtain multiple contacts between the tool and the design surface and produce gouge-free 5-axis tool positions.

Many of these tool positioning methods use a triangular approximation for surface and tool. These approximations impact the tool position relative to the surface. The paper studies the impact approximation can have and identifies limits to the approximations. These limits can be used as guides for setting the accuracy of triangulation for curved surfaces.

Multi-point Method:
In the MPM method [4], positioning the tool is done in two steps. In the first step, the first point of contact is obtained by dropping a tool over the modeled surface from a certain height atop of the

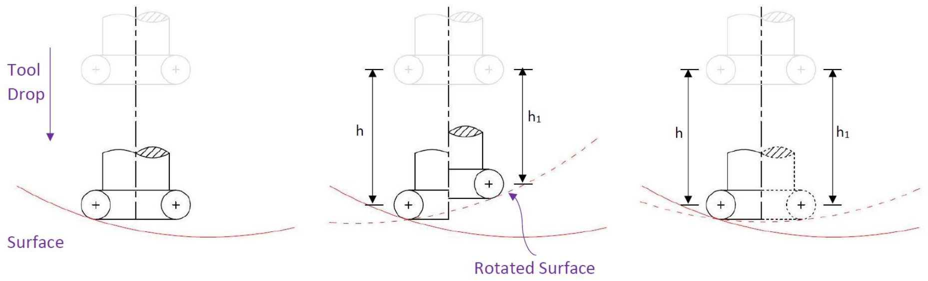

Fig. 1: Tool Drop used for computing tilt angle.

surface. The drop distance for calculation of gouge-free tool position is computed using the GPU. The rendering requires the definition of the view volume and the view window. Since the goal is accurate tool positioning, the view direction is set along the tool axis and the view volume is set to enclose the tool. The window size impacts the time taken to compute the tool positions and the accuracy of the tool position. In this paper, we derive the appropriate size window to use to achieve a desired level of accuracy.

The tool is rendered using Orthographic projection and the resulting z-buffer is saved; a second rendering with the same view direction and view volume is done for the surface and again the resulting z-buffer is saved. These two z-buffers are compared with each other and the minimum difference in z-value or depth value gives the drop distance with which the tool is moved towards the surface to touch it without any gouging. This drop distance concept is fast to compute, and its accuracy can be determined based on the discretization and the world modelling parameters. After an initial drop to obtain the first point of contact, the center and axis of the pseudo insert at that point of contact are obtained.

To calculate the second point of contact, the surface is rotated around the axis of the pseudo insert, the tool is dropped onto this rotated surface, and the process is repeated until a gouge-free contact is established between tool and surface at the second point of contact. The rotation of surface around the axis of pseudo insert ensures that the tool maintains the contact with the surface at first point of contact during the rotation.

The process is illustrated in figure 1. Figure 1a shows the tool dropping on the surface to determine the first point of contact at a height h. In figure 1b the surface is rotated about the pseudo insert axis, and the tool is dropped onto this rotated surface at a height h_1. The surface is rotated iteratively until a rotation angle, θ, is found where $h_1 = h$. Rotating the surface by the angle θ results in two points of contact as shown in figure 1c. The surface rotation about the pseudo-insert ensures the toroidal tool axis stays aligned with the vertical. In this method tool drop with the z-buffer method is used repeatedly to find both points of contact.

Multi-level Rendering:

Achieving the desired level of error may require a large window, resulting in large computation time. To reduce this cost, we used a multi-leveled approach. In the Multi-level rendering approach, the torus is rendered in two stages with lower window resolutions. In the first stage, the orthographic projection is kept the same, as explained in the previous section, except the window size is reduced. The above procedure is followed to obtain the drop distance and associated pixel position. Next the discretization is locally refined around this approximate point of contact to improve the accuracy of point of contact.

Proceedings of CAD'23, Mexico City, Mexico, July 10-12, 2023, 206-210

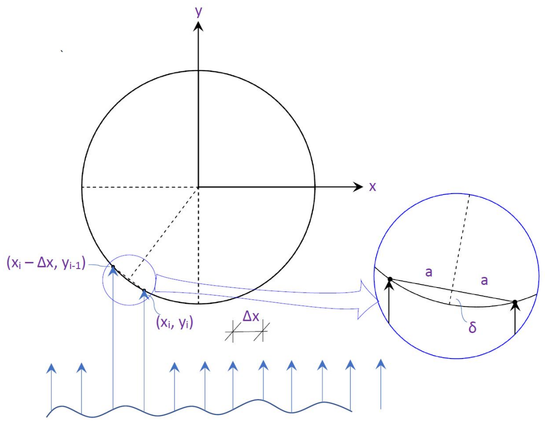

Fig. 2: Error in piecewise linear approximation to a circle, δ, where Δx is the distance between samples,

The approximate contact point is surrounded by other points in the grid; however, the algorithm ensures that the drop distance at these points is higher than at the approximate contact point. As both the torus and the surface are continuous bodies, the mean-value theorem tells us that the lowest value lies in the region surrounded by the neighboring pixels. So, both the torus and the surface are rendered again with a zoomed view of the surrounding area that focuses on a 3×3 area around the contact point. In this second stage of rendering, the camera is moved to the estimated point of contact, but the view vector is retained. The *ToolDropMethod* is re-applied for the newly rendered sections of torus and surface.

One issue to resolve is what resolution (in pixels) to make the windows used in rendering? I.e., more pixels results in higher accuracy, but longer render times. The next section looks at the error resulting in the pixel approximation, and derives a method for determining the window resolution based on the maximum error deemed acceptable by the user.

Error Analysis:

In this work the tool and the part surface are approximated by planar facets. The triangulation simplifies the mathematical formulation but requires it to be executed many times. Accuracy requires fine triangulation and computational efficiency favours coarse triangulation. This is an ideal optimization problem in which user acceptable error can be used to determine triangular accuracy. Facet approximation introduces error in the tool position. The impact of the error due to approximation of the torus is discussed here.

The triangle rendering of the torus is a piecewise-linear approximation to the torus. Looking at a circular cross-section of the torus, and referring to figure 2, the error δ in the approximation is a function of the spacing between the pixels (Δx) and the position on the circle (x_i). Since orthographic projection is used, the rendering of each pixel can be thought of as casting a ray parallel to the tool axis. Two neighboring rays are shown in figure 2. Essentially, the z-buffer algorithm is making a linear approximation to the circle on the line segment between the two points of intersection of these rays and the circle. This approximating line deviates from the circular insert. From right triangles, we can show that the maximum distance δ from this approximating line to the circle is

$$\delta = \frac{a^2 + \delta^2}{2R_i}$$

and since $\delta \ll 1$ we can drop the δ^2 term and still have a tight bound on δ:

$$\delta < \frac{a^2}{2R_i}. \tag{2.1}$$

Knowing the equation of the circle, the length of the line segment between the intersection of two neighbring rays that are Δx apart can be calculated. The rays fired from x_j and at x_{j-1} intersects the circle at y_j and at y_{j-1}. Then

$$\begin{aligned}(y_j - R_i)^2 &= R_i^2 - x_j^2 \\ (y_{j-1} - R_i)^2 &= Ri^2 - (x_j - \Delta x)^2\end{aligned}$$

and thus

$$\begin{aligned}\Delta y = y_j - y_{j-1} &= R_i - \sqrt{R_i^2 - x_j^2} - \left(R_i - \sqrt{R_i^2 - (x_j - \Delta x)^2}\right) \\ &= -\sqrt{R_i^2 - x_j^2} + \sqrt{R_i^2 - (x_j - \Delta x)^2} \end{aligned} \tag{2.2}$$

In this equation, Δx is constant and Δy is small at the bottom of the circular insert and is largest at the side of the insert (at $x = R_i$) where it is given by

$$\Delta y = \sqrt{2R_i \Delta x - (\Delta x)^2}.$$

Further, we know

$$2a = \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2} = \sqrt{(\Delta x)^2 + (\Delta y)^2}. \tag{2.3}$$

Substituting a and Δy (equations (2.3) and (2.2)) in (2.1) gives us the following expression for δ:

$$\delta < \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{8R_i} = \frac{\sqrt{(\Delta x)^2 + \left(-\sqrt{R_i^2 - x_j^2} + (\sqrt{R_i^2 - (x_j - \Delta x)^2})\right)^2}}{8R_i} \tag{2.4}$$

and at $x_j = R_i$, where the error is maximum, we have

$$\delta < \frac{2R_i \Delta x}{8R_i} = \frac{\Delta x}{4} \tag{2.5}$$

as a bound on the worse case error in our approximation.

We can use (2.5) to determine how many pixels to use in our GPU algorithm to obtain a desired error bound. For example, with the machine in our lab, using a tool with $R_o = 6.7mm$, $R_i = 6mm$, where we want $0.01mm$ accuracy, using a view volume of $25mm \approx 2(R_o + R_i)$, we can see that using a width and height of 625 pixels gives a $\Delta x = 25mm/626 = 0.04mm$ and thus δ has the desired accuracy of $\delta < 0.01mm$.

However, note that this worst case error occurs with the outer portion of the tool. While this outer portion of the tool will cut away stock, for many types of machining, the finish machining will be cut with the bottom of the tool. Equation (2.4) gives the error δ as a function of x_j and Δ_x. We plot this error function (generalizing the discrete locations x_j to continuous locations x) in Figure 3. In this plot, we see that away from the edge of the tool, the error is more than an order of magnitude lower than the maximum error. Thus, if the finish machine is performed with the bottom of the tool, a larger pixel size should give acceptable error bounds in our GPU simulation.

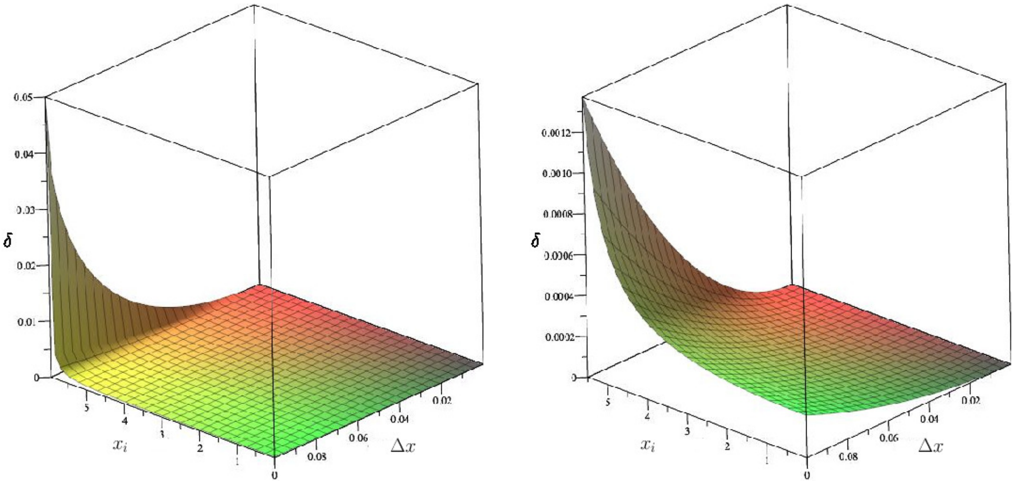

Fig. 3: Approximation error δ as a function of x and Δx, with x ranging from 0 to R_i in mm.

Conclusions:

The tool positioning method proposed in this paper does not require the solution of simultaneous equations or initial guesses. The new method emulates the tool dropping on a surface [3]. In addition, the GPU based tool drop method is repeatedly used with a rotation algorithm to obtain multiple contacts between the tool and the design surface and produce gouge-free 5-axis tool positions.

Many of these tool positioning methods use a triangular approximation for surface and tool. These approximations impact the tool position relative to the surface. The paper studies the impact approximation can have and identifies limits to the approximations. These limits can be used as guides for setting the accuracy of triangulation for curved surfaces. In the final paper, we give additional details of our GPU algorithm and we extend our analysis to determine an appropriate tessellation level to use for the design surface.

Acknowledgement:

This research is funded by the NSERC Discovery grants of the authors.

References:

[1] Gray, P.; Ismail, F.; Bedi, S.; Graphics-assisted Rolling Ball Method for 5-axis surface machining, Computer-Aided Design, 36(7), 2004, 653–663, https://doi.org/10.1016/S0010-4485(03)00141-6

[2] Duvedi, R.K.; Bedi, S; Mann, S.; Numerical implementation of drop and tilt method of five-axis tool positioning for tensor product surfaces, in the Internation Journal of Advanced Manufacturing Technology, 95, 2018, 219–232. https://doi.org/10.1007/s00170-017-1193-1

[3] Duvedi, R.K.; Bedi, S.; Batish, A.; Mann, S.; A multipoint method for 5-axis machining of triangulated surface models, Computer-Aided Design, 52, 2014, 17-26. https://doi.org/10.1016/j.cad.2014.02.008

[4] Duvedi, R.K.; Singh, M.; Bedi, S.; Mann, S.; Multipoint tool positioning of a toroidal end mill for five-axis machining of generalized tensor product Bézier surfaces, in the International Journal of Advanced Manufacturing Technology, 110, 2020, 495–503, https://doi.org/10.1007/s00170-020-06006-1