euspen's 24th International Conference & Exhibition, Dublin, IE, June 2024
www.euspen.eu

Adaptive dexelisation approach for material removal simulation in milling

Yigit Ozcan[1,2], Shashwat Kushwaha[1,2], Jun Qian[1,2], Dominiek Reynaerts[1,2]

[1]Department of Mechanical Engineering, KU Leuven, Celestijnenlaan 300, Leuven 3001, Belgium
[2]Member Flanders Make, Belgium

yigit.oezcan@kuleuven.be

Abstract

Minimizing lead time without sacrificing accuracy is a major challenge in various industries. In CNC milling, geometrical inspection takes one of the biggest efforts. Process-parallel simulations have the potential to reduce down-time and effort spent in inspection. Especially for real-time workpiece quality estimation, the cutter workpiece engagement methods must be improved to achieve effective computation without sacrificing precision. According to the literature, the dexelisation method is one of the best contenders for high-fidelity material removal simulations.
The literature shows promising results for offline part geometry estimation considering cutting loads. However, academic studies and industrial software tools are scarce for process-parallel material removal simulation. This is due to the need of high computation power. Therefore, in this study, an adaptive dexelization approach for workpiece is proposed. The method is accelerated by using a Ray Tracing algorithm and GPU cores without comprising the accuracy of inspection considering cutting loads. Around five times reduction on computation time has been reached with this method.

Material removal simulation, dexelization, Tridexel, milling, cutter workpiece engagement.

1. Introduction

Currently, the manufacturing field is experiencing the fourth industrial revolution, which enables value-adding interconnected assets to communicate with each other so that digital data can provide greater added value. This digital revolution allows faster reporting and hence efficient and timely decision making and intervention. Like in other manufacturing methods, in CNC milling, there is a trend through the digitalisation of the process to generate a digital model of the physical product [1]. In that sense, virtual machining is critically important to evaluate whether the physical workpiece is in the desired condition during the process.

Three primary methods exist for virtual workpiece representation for milling. (1) Solid Modelling, including Constructive Solid Geometry (CSG) or Boundary Representation (B-rep) [2], uses primitive volumes updated through Boolean operations. However, both CSG and B-rep have limitations; Brep's reliability in updating the workpiece for every cutter location is uncertain, and CSG struggles with supporting free-form objects [3].

(2) Space Partitioning, representing the workpiece through voxels, discrete volume elements defined as binary (1 for material, 0 for empty) [4]. Here the memory allocation is directly proportional to $O(n^3)$, where O is memory allocation, and n is the number of voxels. Dexel method overcomes the memory requirements. Here, the workpiece is represented by vectors, where memory allocation is proportional to $O(n^2)$ [5]. On top, tri-dexel methods address fidelity problems, e.g. on high slope surfaces by sending line segments from three perpendicular axis-aligned planes [6]. The tri-dexelisation can be sped by running ray tracing algorithm on a GPU [7].

Despite progress in virtual workpiece representation for CNC milling, achieving high-fidelity and low-latency in the digital twin context remains a challenge. This study focuses on enhancing Tri-dexel method efficiency by adaptively adjusting dexel density based on known cut regions and their expected tolerance. This step is pre-processing for process-parallel geometry estimation intended to save memory and reducing simulation time while improving the precision at required locations.

2. Tridexel algorithm

2.1. Step -1: Adaptive Workpiece Dexelisation

In this section, details of adaptive Tri-dexel algorithm are given. Our method requires the input of tessellated CAD model of the stock material. The user divides the CAD model into sub-geometries with respect to required tolerances in every x, y, and z directions separately while triangulating the geometry (.STL conversion). Each sub-geometry is to be defined in a specific dexel resolution and corresponding spatial grids are generated along the three axes. From each grid, rays are cast to make intersection with triangles of sub-geometries. This procedure is applied only once before the material removal is calculated. It should be noted that each grid execution is done in one GPU block. The results from every ray-triangle intersection are transferred to the CPU to find the starting and ending points of

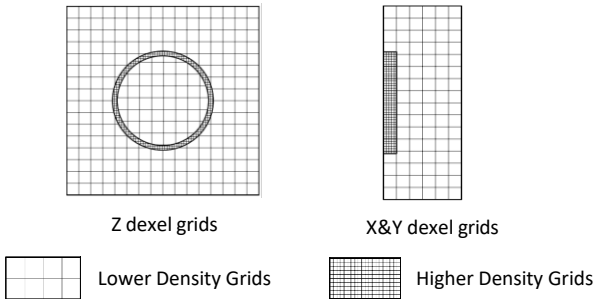

Figure 1. Adapted Grids for WP in X,Y,Z directions

every intersection calculation between ray and triangles. Depending on the intersection distance and grid IDs, the starting and ending points are found by sorting for each grid IDs. In this study, an 80x80x30 mm workpiece material was dexelised adaptively. Referring to Figure 1 the inner 40 mm diameter cylindrical pocket is dexelised with higher density grids. Here a 1 mm from the cylindrical surface is chosen for high density dexels. Similar procedure is applied in X and Y direction independently. In all directions, remaining part is kept with relatively lower grid density, i.e., 0.5 mm grid resolution, which is represented in Figure 1. Regarding to these grids, adaptive dexels in X and Z directions can be seen in the Figure 2. Note that because of the symmetricity, Y dexels are basically the same figure with dexelised workpiece (WP) in X direction.

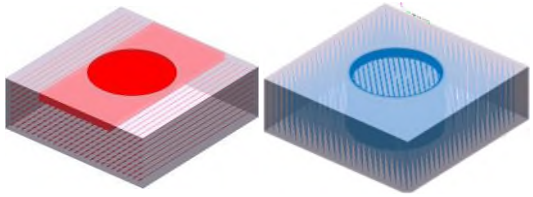

Figure 2. Dexelised WP in X and Z directions.

2.2. Step 2: Finding Axis Aligned Bounding Box (AABB)

At this stage, another improvement on calculation can be done by using an axis aligned bounding box (AABB). First, multiple cutter locations are collected and AABB is calculated by the finding an envelope for the collected cuter locations. The AABB allows to locate the dexels within this envelop. For the material removal simulation to proceed, only the dexels inside the AABB are loaded for further calculations. That means only the rays inside of the extreme point of cutter regions is to be activated for ray to tool geometry intersection. In the Figure 3, the blue lines represent the tool path and corresponding tool locations in grey. By finding the extreme points in XY plane (for Z direction dexel intersection) the AABB is created. In this way, only the dexels that fit into this box are activated for ray-cylinder/tool intersection.

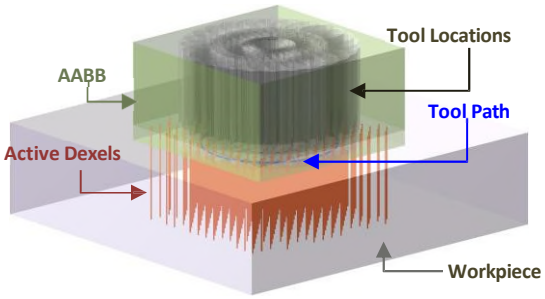

Figure 3. Active Z dexels by finding AABB of tool locations.

2.3. Step 3: Ray-tool intersection

To further simplify the calculations, the tool (end mill) is represented as a cylinder. The ray cylinder intersection calculation was used, which is computationally effective as opposed to intersection problem with a triangulated tool. If the intersection distance is lower than the end point of corresponding active dexel element's end point, then the dexel element end point updated with the intersected point.

The algorithm is developed on MATLAB by using the parallel computing tool of CUDA cores. NVidia GeForce RTX3070 graphics card with 8GB dedicated memory was used for ray-triangle/cylinder intersection. In this simulation a 10 mm flat end tool is selected. Cutter is simulated as a cylinder geometry with total 553 cutter locations for the circular (high density) region as 40 mm diameter and 5 mm depth.

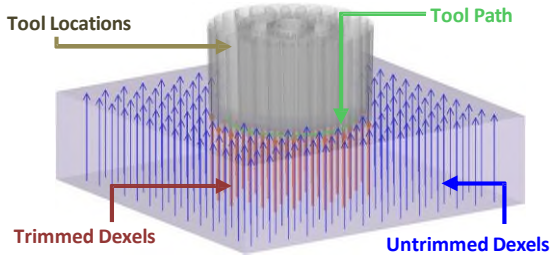

Figure 4. Dexel trimming

3. Results and Conclusion

The comparison of computation time for uniform dexelisation and adaptive dexelisation is shown in Figure 5. The low density region in the adaptive dexelisation case is always set to 0.5 mm. Whereas, the high density region is varied from 15 to 250 µm. On the other hand, for uniform density dexelisation, the whole workpiece is dexelised with the same density. The GPU memory imposed a limit in dexel resolution, at 15 µm uniform density case utilised the whole 8 GB of GPU.

It is seen that at the highest resolution a 5 times simulation time improvement is achieved. The calculation cost is compared based on Step 2 and 3, because adaptive dexelisation (Step 1, pre-process) is a one-time operation and it is not the part of process-parallel geometry estimation during the milling process.

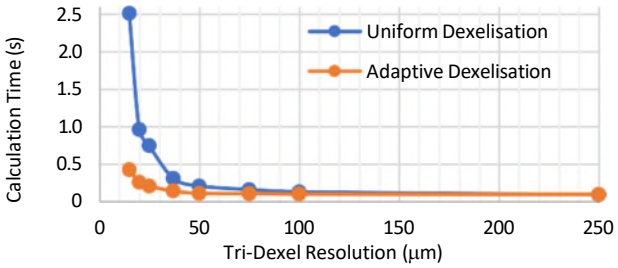

Figure 5. Comparison of uniformly and adaptively WP calculation

As a conclusion, adaptive dexelisation provides a potential use with GPU parallelized dexelisation. Since part geometry features can be read from PMI data of the CAD by using some standardized formats like QIF or STEP 242, these standardised formats allow the user to reach features of the CAD data with their IDs in .xml format. Therefore, finding these regions by using standardized formats for adaptive dexelisation is the potential future study.

References

[1] Frenz W 2022 *Handbook Industry 4.0: Law, Technology, Society*. Springer Nature

[2] Requicha AAG, Rossignac JR 1992 Solid Modeling and Beyond *IEEE Comput Graph Appl* **12**, no. 5,

[3] Benouamer MO,Michelucci D 1997 Bridging the gap between CSG and Brep via a triple ray representation *Proceedings of the Symposium on Solid Modeling and Applications*, pp. 68–79

[4] Miers JC, Tucker T,Kurfess T,Saldana C 2021 Voxel-based modeling of transient material removal in machining *International Journal of Advanced Manufacturing Technology* **116**, no. 5–6, pp. 1575–1589

[5] Altintas Y, Kersting P, Biermann D, Budak E, Denkena B, Lazoglu I, 2014,Virtual process systems for part machining operations *CIRP Ann Manuf Technol* **63**, no. 2

[6] Lee SW,Nestler A 2012 Virtual workpiece: Workpiece representation for material removal process *International Journal of Advanced Manufacturing Technology* **58**, no. 5–8, pp. 443–463

[7] Abecassis F, Lavernhe S, Tournier C, Boucard PA 2015 Performance evaluation of CUDA programming for 5-axis machining multi-scale simulation *Comput Ind* **71**