



Title:

Clustering of Machinable Volumes For Tool Selection In 3-axis Milling

Authors:

Hariharan Krishnamurthy, hariharan_k@hcl.com, HCL Technologies
 Manoj Bhonge, manoj.bhonge@hcl.com, HCL Technologies
 Tathagata Chakraborty, tathagata.chakr@hcl.com, HCL Technologies

Keywords:

Tool Selection, Triangulated meshes, 3-axis milling, Machinable volume, Voxelization, Theoretical model

DOI: 10.14733/cadconfP.2022.66-71

Introduction:

Three-axis machining is a commonly used method of part production. Here, operation planning involves stock selection, sequencing of operations, tool selection and toolpath computation. Tool selection is an important part of operation planning, and involves geometry-based factors and tool, stock material among many others. The set of suitable tools is selected from the multitude of tools available in a machine shop.

Optimal tool selection has been researched primarily as mathematical optimization. Most works aim to minimize the total machining time or the net energy consumed of a subset of tools. Despite considerable work, tool selection is left to the machinist today as researched methods are not available in commercial software. We present a method in this paper to aid tool selection by computing a non-linear measure of the machinable volume by a tool, and using it to cluster available tools into subsets relevant to the operation at hand. The method is general, independent of tool, stock shapes, and can be enhanced to include other factors pertinent to machining processes.

Prior art:

Prior research into tool selection mainly used optimization algorithms, such as combinatorial ones. In [9] different combinations of a feasible range of flatend tool sizes, determined from a pocket's geometry, are explored using a branch-and-bound method. In [13] the 'voronoi mountain', the extrusion of the medial-axis of a polygon, is used to compute planar-pocket toolpaths and the leftover for the next tool in a sequence, and an optimal sequence found using dynamic programming. This method is extended to non simply-connected pockets in [12]. The work in [3] uses adaptive slicing of part meshes to compute accessible volumes of tools, and selects tools from size-based clusters to minimize machining time.

A set of methods first computes the tool-accessible areas in planar slices of pockets, then creates a weighted graph of tool sequences with overlapping accessible areas, and finds the shortest path (using Dijkstra's algorithm) for the optimal sequence [5, 6, 11]. In [15], the graph-based method is used for optimizing milling time across parts considering slices from multiple parts together.

In [4] several meta-heuristic optimization methods for tool selection are compared. In [7], the energy consumed in flexible manufacturing is optimized by mixed-integer-programming. In [14] the energy consumption, usage cost for milling tools is optimized; In [2] the average MRR is greedily optimized by computing the toolpaths from voxel-based offsets.



Fig. 1: Examples of input shapes - (left) target part, and (right) flatend tool of 10mm diameter

Most of the reviewed methods cast tool selection as evaluating a graph of tool sequences, and require toolpath computation which can be expensive. The methods also require features and are restricted to 2.5D ones. The present method estimates the tool-accessible volume discretely on the whole part, and doesn't require features or toolpath computation, and applies to 2.5-axis, 3-axis milling.

Machinable Volumes - Computation:

The set of all points accessible by the tool in the volume to be removed, without penetrating the part, is termed as the *machinable volume* by the tool, indicating the maximum material removable. The set of non-accessible points in the delta volume represent the leftover by the tool. The tight bounding box of the target part (see 1(left) for example part) is the stock used. Regions with an 'overhang' are considered inaccessible i.e. 'undercut' regions are not machinable. The tool is modelled by the flute's 'envelope', such as a cylinder for flatend tool, such as in figure 1(right).

The part shape is enclosed by a uniform grid larger than the bounding box, and grid-cells intercepting the part surface, called boundary voxels hence, are identified using triangle-box overlap[1]. The cells between a pair of boundary voxels in a column are classified as interior, as they occur between boundaries. All other exterior cells are outside, denoting removable material. Part and tool grids are separately classified, using the same grid-cell size.

The voxel containing the tool tip, is placed one voxel (called a query voxel) above a part boundary voxel, and the overlap between *corresponding* tool, part voxel columns is checked (see 2(left)). If no overlap is found, then the query voxel is accessible. In case of overlap, the height to escape penetration is calculated as the maximum difference between tool and part *boundary* voxels among all corresponding voxel columns (see 2(right)). This 'safe' height is the Hausdorff distance between input shapes at the query voxel, considering the machining direction as Z.

All the exterior voxels subsumed by the tool placed at a query voxel are considered machinable. The union of machinable voxels at all query voxels represents the machinable volume, while the union of non-accessible voxels represents the leftover volume. The actual volume is the product of voxel count by the unit cell volume .

A surface representation of voxels is convenient for visualization. The voxel class (leftover/machinable) is used to assign the fraction of neighbourhood voxels to a given one, as a scalar; this ranges from 0.0 for isolated to 1.0 for enclosed voxels. The iso-surface of the scalar (got by marching cubes) for a very small value is the leftover volume; the machinable volume is obtained similarly. The leftover volume for the part in 1(left) with a 10mm flatend tool is shown in figure 3.

Clustering Of Machinable Volumes:

Clustering is the grouping of semantically common elements from a set of data points; the meaning

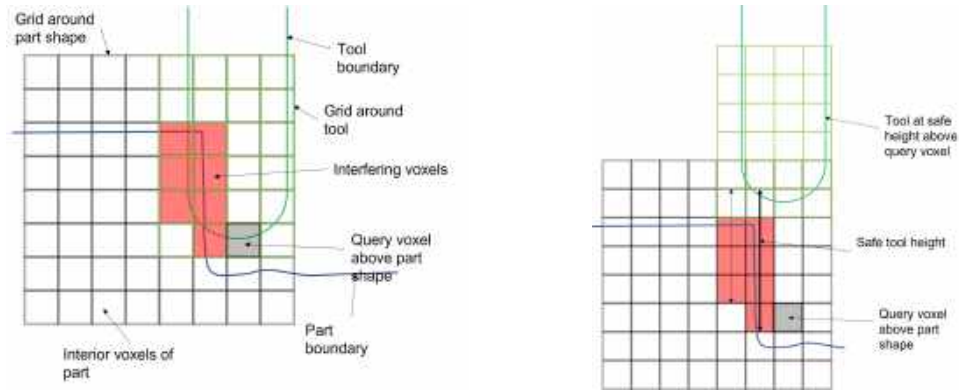


Fig. 2: Schematic depiction of tool placement and interference check - (left) tool interference with target part when placed above part voxel, (right) calculation of safe height by ‘lifting’ tool along machining direction.

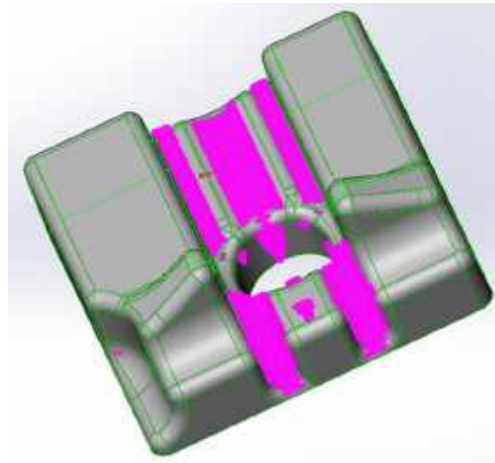


Fig. 3: Bounding surface (shaded translucent pink) of the leftover volume with a 10mm flatend tool on the part surface. The surface is the iso-surface for a small value of the scalar assigned to leftover voxels.

depends on the application domain. The clustering exercise can be explored along many aspects such as method chosen, number of expected clusters, the dimensionality of the ‘feature’ space and so on. While the number of clusters is a ‘hyperparameter’ depending on the method, the features used are described in the following.

The machinable volume is a feature used to represent the tool in a part-aware manner. Apart from this, the average Material Removal Rate (MRR) is used, calculated from linear speed and other parameters from tool data tables. The machinable volume and MRR are used to estimate machining time, juxtaposes machinable volume. The combinations of these features are tested for clustering from singleton ones, such as machinable volume or MRR only, to all three features together.

There are a multitude of clustering techniques developed over the years; Three out of these, namely the k-Means, mean shift and agglomerative clustering are explored here. K-means is an old, standard clustering technique, while mean shift and agglomerative clustering are newer, general methods. The

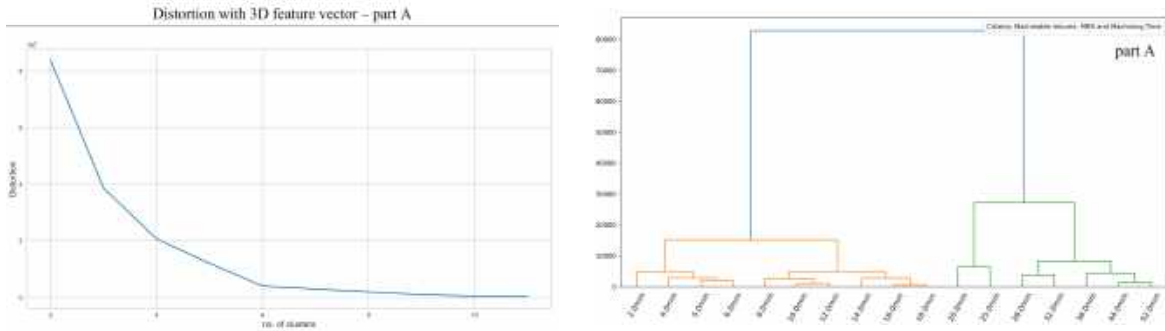


Fig. 4: Methods for choosing the ‘hyperparameter’ of the number of clusters - (left) ‘distortion’ score for k in k-Means, (right) dendrogram for agglomerative clustering.

k-Means method initializes and updates the centroids of k clusters iteratively. The hyperparameter k is chosen using distortion or silhouette scores usually; presently, the distortion score (see figure 4(left)) is used as it yields distinct ‘elbows’. Mean shift clustering repeatedly updates centroids of regions, and as such obtains the clusters.

In agglomerative clustering, clusters are formed by progressively merging atomic clusters into larger ones. A dendrogram, a tree-like visualization of merged clusters, (see figure 4(right)) is inspected to decide the number of clusters. The merging or linking is based on a ‘distance’ measure; presently ‘ward’ linking is used.

Results and Discussion:

The clustering methods were experimented on three parts, considering 17 flatend tools sized 2mm to 52 mm. The python packages scikit-learn ([10]) and matplotlib ([8]) were used for clustering and plotting respectively.

For the part shown in figure 1(left), k was chosen as 5 for both agglomerative and k-Means using distortion score and dendrograms respectively. The results with all three features are discussed for brevity; the results with k-Means are shown in figure 5. The five clusters obtained were - one with tools <10mm, two for those 10<25mm and two for those >25mm with both the k-Means and agglomerative methods. The mean-shift method identified three clusters - <20mm, between 20 and 30mm and >30mm.

The other parts also showed results agreeing with the expected number of tool groups for the given part shape. The clusters from mean shift were similar across all three parts, while agglomerative and k-Means complied with each other for each part.

Some of the singleton features, such as machining time alone, led to different clustering than others. This could mean that, by itself, machining time does not distinguish tools well. The clustering results agreeing with expected grouping of tools would mean that the method is generally applicable; for example, three distinct groups were obtained for a part with three nested pockets. Tools which cannot machine any volume on a given part were also identified as having 0 volume, and separated by the clustering, helping eliminate such tools quickly from consideration.

Conclusions:

The method of computation of machinable volume seems useful and distinctive for clustering. The machinable volume is general. The geometric operations are independent, and thus parallelizable. The clustering using features like machinable volume resulted in intuitive grouping of tools. A combination, rather than individual, of features seems more suitable for clustering as they represent different machining

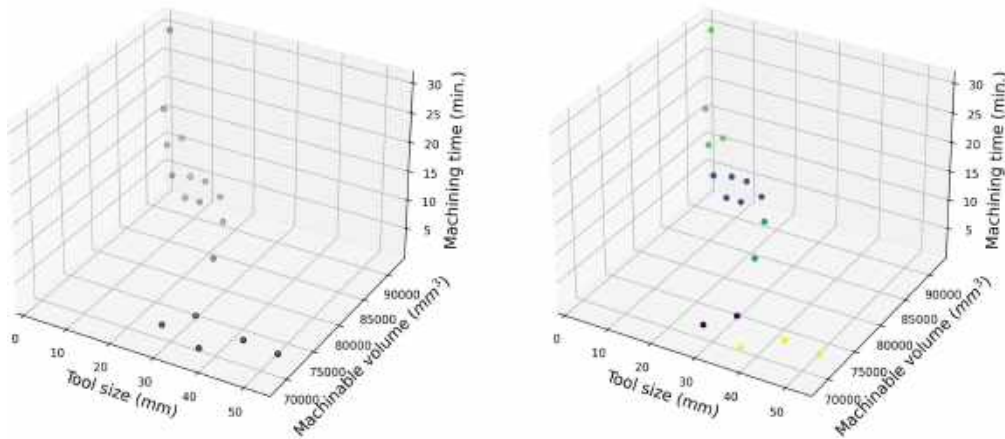


Fig. 5: Clusters identified for the example part with k-Means method seeking 5 clusters, with all three features used to represent the tool.

aspects. Agglomerative clustering seems more suitable, being visual and interpretable,

Acknowledgement:

We thank Anand M., Manish R. and Swadhin B., Vivek G. for help with the work and the initiation respectively. We thank HCL Technologies for the opportunity and environment supporting the work.

References:

- [1] Akenine-Moller, T.: Fast 3D triangle-box overlap testing. *Journal of graphics tools*, 6(1), 2001 , 29-33.<https://doi.org/10.1145/1198555.1198747>
- [2] Amuer, A.: Voxel-based tool sequence optimization for 5-axis machining using high performance computing. M.Sc. thesis, Georgia Institute of Technology, 2017.<https://smartech.gatech.edu/bitstream/handle/1853/60655/AMEUR-THESIS-2017.pdf>
- [3] Balasubramaniam, M.: Tool selection and path planning in 3-axis rough machining. M.Sc. thesis, Massachusetts Institute of Technology, 1999.<https://dspace.mit.edu/bitstream/handle/1721.1/80500/43183908-MIT.pdf?sequence=2>
- [4] Churchill, A. W., Husbands, P., and Philippides, A.: Metaheuristic approaches to tool selection optimisation. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, 2012, 1079-1086.<https://doi.org/10.1145/2330163.2330313>
- [5] D'Souza, R., Wright, P., and Sequin, C.: Automated microplanning for 2.5-D pocket machining. *Journal of Manufacturing Systems*, 20(4), 2001 , 288-296.[https://doi.org/10.1016/S0278-6125\(01\)80048-0](https://doi.org/10.1016/S0278-6125(01)80048-0)
- [6] D'Souza, R. M.: On setup level tool sequence selection for 2.5-D pocket machining. *Robot Comput Integr Manuf.*, 22(3), 2006 , 256-266.<https://doi.org/10.1016/j.rcim.2005.06.001>
- [7] He, Y., Li, Y., Wu, T., and Sutherland, J. W.: An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops. *Journal of Cleaner Production*, 87, 2015 , 245-254.<https://doi.org/10.1016/j.jclepro.2014.10.006>
- [8] Hunter, J. D.: Matplotlib: A 2D graphics environment. *Computing in science and engineering*, 9(03), 2007 , 90-95.<https://doi.org/10.1109/MCSE.2007.55>

- [9] Kyoung, Y. M., Cho, K. K., and Jun, C. S.: Optimal tool selection for pocket machining in process planning. *Computers and industrial engineering*, 33(3-4), 1997, 505-508.[https://doi.org/10.1016/s0360-8352\(97\)00179-4](https://doi.org/10.1016/s0360-8352(97)00179-4)
- [10] Pedregosa, F., Varoquaux, G., Gramfort, A., et al: Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12, 2011, 2825-2830.<https://doi.org/10.5555/1953048.2078195>
- [11] Ramaswami, H., Shaw, R. S., and Anand, S.: Selection of optimal set of cutting tools for machining of polygonal pockets with islands. *Int. J. Adv. Manuf. Technol.*, 53(9), 2011 , 963-977.<https://doi.org/10.1007/s00170-010-2909-7>
- [12] Seth, A., and Stori, J. A.: Optimal tool selection for 2.5 D milling, part 2: a Voronoi mountain approach for generalized pocket geometries. *Int. J. Comput. Integr. Manuf.*, 18(6), 2005 , 463-479.<https://doi.org/10.1080/09511920512331319636>
- [13] Veeramani, D., and Gau, Y. S.: Selection of an optimal set of cutting-tool sizes for 2D pocket machining. *Computer-Aided Design*, 29(12), 1997 , 869-877.[https://doi.org/10.1016/S0010-4485\(97\)00042-0](https://doi.org/10.1016/S0010-4485(97)00042-0)
- [14] Wu, L., Li, C., Tang, Y., and Yi, Q.: Multi-objective tool sequence optimization in 2.5 D pocket CNC milling for minimizing energy consumption and machining cost. *Procedia Cirp*, 61, 2017 , 529-534.<https://doi.org/10.1016/j.procir.2016.11.188>
- [15] Yao, Z., Gupta, S. K., and Nau, D. S.: Algorithms for selecting cutters in multi-part milling problems. *Computer-Aided Design*, 35(9), 2003 , 825-839.[https://doi.org/10.1016/s0010-4485\(02\)00110-0](https://doi.org/10.1016/s0010-4485(02)00110-0)