

Automatic Spatial Planning for Machining Operations

Saigopal Nelaturi, Gregory Burton, Christian Fritz and Tolga Kurtoglu
System Sciences Laboratory, Palo Alto Research Center
Palo Alto, CA 94304, USA

Email: {Saigopal.Nelaturi, Gregory.Burton, Christian.Fritz, Tolga.Kurtoglu}@parc.com

Abstract—We present a novel spatial planning system that automatically generates machining plans with stable fixtures to fabricate complex geometries. Given a cutting tool, the system initially finds all tool poses that do not cut into the desired part geometry, and constructs the maximal volume removable from a raw stock according to the tool’s degrees of freedom. The planning space is then defined as the product space of all available tools and their associated maximal removal volumes. A sequence of machining operations to iteratively remove material from a raw stock is generated by searching the planning space and finding successive machining steps that minimize manufacturing time or cost. For each step, a vise or modular fixture is automatically assembled such that the part is stable and the fixture assembly does not interfere with tool motion. Process planning is an inverse problem and multiple feasible solutions exist to machine the part if it is deemed manufacturable by the system. The system generates and visualizes several qualitatively distinct fabrication plans for an engineer to choose from. The chosen plan is sent to a tool path planner to generate machining instructions. We will show several examples of rapid and automatic end-to-end machining process planning on complex geometries to demonstrate the scalability and practicality of our system.

I. INTRODUCTION

A. Machining Process Planning

Computer aided manufacturing process planning is the mapping of design specifications into a sequence of manufacturing operations and accompanying instructions to fabricate a product. In machining, the sequence consists of individual setups where a part is fixtured and a cutting tool is programmed to strategically remove material without colliding with surrounding tooling. Given the nominal (perfect form) design and manufacturing constraints, process planning is an inverse problem that attempts to find a sequence of setups such that the manufactured part is equivalent (up to tolerance) to the designed shape. The problem is ill-posed because a solution may not exist if the part is non-manufacturable, or multiple solutions may exist if the part is manufacturable.

It was recognized quite early [19] that subsets of solid models called *features* could be parameterized and annotated with manufacturing information including datums, tolerances, machining operations, and material properties. Consequently machining process planning using features evolved into a vast area of academic research over the last three decades, and has been considered the de-facto approach to map CAD geometry to CAM instructions. However, despite these research efforts, today’s commercial product lifecycle management systems have limited support for automatic

feature based process planning, and production floors largely still rely on operator expertise to manually transform design models into manufacturing plans. This is due in part to shortcomings of feature-based process planning approaches.

B. Challenges in feature based process planning

There is no consistent definition of a feature across design and manufacturing, but in practice features are assumed to represent ‘engineering intent’ interpreted in an application specific manner [21], [22], [4]. For example assembly design features seen in CAD systems represent mating surfaces between parts and are not in one-to-one correspondence with features in machining process planning, where boundaries of features are assumed to coincide with machined surfaces. Process planners depend on recognizing and extracting machining features from a solid model so that they may be mapped to a relevant set of manufacturing processes. Automatic feature recognition for machining comprises a large body of work with notable techniques including volume decomposition [28], [11], [5], analysis of the boundary representation graph [10], [30], and rule based pattern recognition [1], [27] among others. See [9], [8] for a comprehensive review of challenges in recognizing manufacturing features.

Most feature recognition techniques are restricted to pocket milled, drilled, and turned features. For such features, accessibility analysis [23], [29] outlines the candidate set of orientations in which a part may be machined. Process planners are usually rule based systems [6], [24] that assign a variety of cutting tools and operations to each accessible feature. When parts can be fabricated by independently machining all recognized features, for example in the case of 2.5d milling, such process planners are quite effective. However, the main challenge in feature based process planning is scalability to a large class of parts, especially when designs consist of interacting features [26], machinable regions not classified as manufacturing features, or regions requiring combinations of machining operations in multiple orientations. In these cases, when the material volume to be machined is decomposed into features, the subvolumes created may not be manufactureable, or may be difficult to identify as manufactureable, while an alternative decomposition could yield different results.

C. Contributions and paper outline

In this paper, we demonstrate a new, feature-free approach for automatic process planning that adopts an alternative

viewpoint of computing the *maximal* machinable volume in any accessible orientation. Using properties of the configuration space of tool motions, the approach scales to complex part and tool geometry, dramatically reduces the search space using appropriate heuristics, and facilitates rapid computations. As a result, process plans can be computed using commodity compute hardware in less than a minute even for very complex parts with many features. The paper is organized as follows: 1. A set theoretic formulation of the maximal volume machinable from a fixtured stock by a tool moving in its configuration space (Section III). 2. Efficient algorithms to compute maximal machinable volumes (Section IV). 3. Set covering and search algorithms over maximal machinable volumes to determine qualitatively distinct process plans (Section V). 4. Automatic selection and assembly of fixtures for individual machining steps in a plan (Section VI).

II. MATHEMATICAL PRELIMINARIES

We define and compute collision free rigid motions of cutting tools in the configuration space $SE(3)$ of rigid motions. Subsets of $SE(3)$ are termed *filters* and readers are referred to [15], [14] for rigorous analysis of properties involving the action of filters on solids for spatial planning. We use two important operations relating solids and filters:

- *Dilation* [16], [14], or sweep, of a solid X by a filter \mathcal{M} is defined as $\delta(\mathcal{M}, X) = \bigcup_{m \in \mathcal{M}} X_m$ where X_m represents the solid X transformed by a rigid transformation m .

- *Configuration space obstacle* [13], [23] $S \boxtimes T$ of solids S, T is the filter consisting of all transformations of the moving solid T that cause collision with the stationary solid S . Every $h \notin S \boxtimes T$ is said to be *feasible*. When the moving solid T is translating in a fixed orientation, the filter $S \boxtimes T = S \oplus T^{-1}$ is the translational configuration space obstacle of S and T , seen in robot motion planning problems. Here \oplus denotes the *Minkowski sum* [20] and T^{-1} represents the scaling of all coordinates in T by -1 .

III. MAXIMAL MACHINABLE VOLUMES

In this Section, we consider the problem of determining the maximal volume that is machinable from raw stock by a collection of tools, with the goal of manufacturing a part of interest. Each *staging model*, which is the remaining uncut metal before a machining operation, is typically re-fixtured in several orientations to enable the cutting tool to machine the entire part surface. We use standard techniques of accessibility analysis [23], [29] to determine the orientations to be used in the planner. Given a pair (S, P) where S represents a fixtured raw stock or a staging model and P represents the part to be manufactured, let T represent the tool that is selected to machine P from S .

A. Translating Tools

We now formulate the problem of determining the largest subset of S that may be removed by T without interfering with P . To simplify exposition, we will initially consider the problem of determining the maximal machinable volume

from a stationary stock using a translating tool. This situation is manifested in practice, for example, when three axis milling machines are used to cut metal from fixtured stock. The configuration space of the unconstrained tool is \mathbb{R}^3 .

For the part P to be manufacturable, it is a basic requirement that S is placed within the set of positions that a reference point on the tool T can reach in this orientation. The maximal filter of feasible transformations applicable to the tool to avoid cutting into P is the complement of the configuration space obstacle

$$\mathcal{F} = (P \oplus T^{-1})^c \quad (1)$$

Equation 1 outlines the largest subset of \mathbb{R}^3 that is applicable to the tool in the chosen orientation. However, in practice machining is going to be limited by the workspace, or the set of positions and orientations achievable by the tool. In the current situation involving a fixed orientation, we are only concerned with the set of all reachable positions, or the *reachable workspace* of the tool. Denoting the reachable workspace as \mathcal{R} , the maximal set $\mathcal{K} \subset \mathbb{R}^3$ of translations applicable to the tool is revised as

$$\mathcal{K} = \mathcal{F} \cap \mathcal{R} \quad (2)$$

\mathcal{K} is the maximal filter representing all translations applicable to the tool T without interfering with the part P . Therefore the dilation $\delta(\mathcal{K}, T)$ represents the sweep of the tool by the maximal filter within the workspace without interfering with P . Finally, the maximal volume machinable from the raw stock (or staging model) S is given by

$$\mathbb{V} = \delta(\mathcal{K}, T) \cap S \quad (3)$$

B. Rotating and Translating Tools

We now assume a tool T moves according to transformations in a filter \mathcal{W} called the *workspace*, which is a full-dimensional subset of $SE(3)$. The reachable workspace \mathcal{R} is the projection of \mathcal{W} into \mathbb{R}^3 and represents all positions that can be achieved by some orientation of the tool. The positions and orientations achievable by T during its motion are described in terms of the location of a coordinate frame attached to the tool.

Assuming this frame is initially aligned with the global reference frame with respect to which transformations are described, \mathcal{W} also describes positions and orientations of the tool with respect to the absolute frame of reference. The origin of each frame is the reference point of T and indicates the special point of contact between the tool and the stock which will be used to outline the allowable spatial locations of the tool for manufacturability.

The maximal volume that is machinable from a raw stock by a rotating and translating T is formulated by extending Equations 1 and 2 to $SE(3)$. Computations in $SE(3)$ can be broken down into smaller three dimensional computations owing to the product structure of $SE(3) = \mathbb{R}^3 \times SO(3)$ [14] where $SO(3)$ represents the group of spatial rotations. In practice this implies the maximal filter applicable to T to

avoid cutting into P is

$$\mathcal{F} = \bigcup_{r \in SO(3)} (P \oplus T_r^{-1})^c \quad (4)$$

Equation 4 suggests that to compute all (6d) non-colliding tool motions we rotate the tool according to orientations r in the workspace and apply Equation 2 repeatedly to compute and accumulate 3d maximal machinable volumes.

In high axis machining, typically available machine tool orientations are provided per axis. Five axis machines have rotational degrees of freedom about two axes of rotation, in addition to the three translational motions. If rotations about the available axes are denoted \mathcal{O}_θ and \mathcal{O}_ϕ , the subset of $SO(3)$ for which Equation 4 needs to be computed is given by $\mathcal{O} = \bigcup_{\theta \in \mathcal{O}_\theta, \phi \in \mathcal{O}_\phi} (\theta \cdot \phi) \cup (\phi \cdot \theta)$ owing to the non-commutativity of spatial rotation composition. Here \cdot represents composition of rotations, for example by matrix or quaternion multiplication.

IV. COMPUTING MAXIMAL MACHINABLE VOLUMES

Minkowski sum, dilation, and configuration space obstacle calculations form a large body of research in solid modeling, because of applications in spatial planning [15], [12]. In general, boundary representations of these shapes are harder to compute than lower level representations such as sampled point sets. Using implicit representation of shapes on sampled regular grids provides a unified framework to use properties of convolutions, Fourier transforms, and other signal and image processing algorithms to implement these operations [16]. In the general case where no assumptions are made on part and tool geometry, these algorithms can be efficiently applied to compute maximal machinable volumes by directly implementing Equations 3 and 4. Figure 1 shows an example of a maximal machinable volume computed using this approach for a complex part and tool combination.

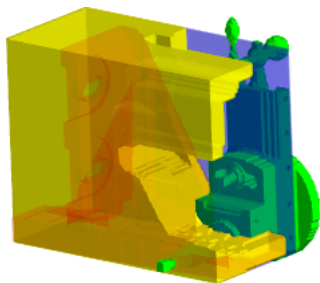


Fig. 1. Computing the maximal machinable volume that a translating shaping tool can machine without colliding with the final part

We now describe an algorithm to compute Equation 1 when T is a translating axisymmetric milling tool and does not possess undercut features when viewed along its spin axis. Many cutting tools such as end mills and ball mills are built this way and so the special case has significant practical value. Furthermore the implementation of the algorithm exploits dimensional reduction making it faster in practice than evaluating full dimensional computations.

Consider a feasible translation of an oriented undercut-free tool T . At every point $v \in \partial T$ there exists a $u \in \partial P$ such that the vector $v - u$ is parallel to the spin axis and the segment $\lambda v + (1 - \lambda)u$, $\lambda \in [0, 1]$ is entirely in P^c . See Figure 2. Every such u is said to be *visible* from the tool.

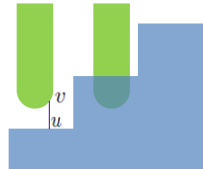


Fig. 2. Left: u is visible from tool. Right: not visible from tool, $\lambda \notin [0, 1]$

We will use the fact that the distance between v and u can be computed easily using *height maps*. The value of the height map at a point q on a plane with normal parallel to the spin axis is the distance from q to the point $p \in \partial P$ visible from q . We implement height maps using standard z -buffer hidden surface removal with orthographic projection [3], where the height map is computed on the camera plane located at infinity.

An infinitesimally thin tool can be represented as a ray, in which case the height map of the part represents the maximum collision-free translation along the tool spin axis for every point (x, y) on the camera plane. For a tool with thickness, the height map of the part is therefore an upper bound on the possible collision-free translation. The important insight is that by moving away from the part an infinitely long, undercut free tool can only transition from a colliding to a non-colliding state. Therefore, we may determine the maximum collision-free translation of the tool along z by starting at the visible surface of the part determined by the z -buffer algorithm and translating away from the part by the maximum penetration depth. The algorithm is presented graphically in Figure 3.

After computing the maximum feasible translation of the tool for every (x, y) (which implements Equation 2), we dilate that heightmap by the heightmap of the tool to get the heightmap of the sweep of the tool, i.e. the heightmap $R_H(x, y)$ of the removal volume. By ignoring rotation about the tool's spin axis due to axisymmetry, or when the tool is represented by its rotational sweep about the spin axis, we remove a degree of freedom and implement the height map in two dimensions using standard image processing algorithms. The maximal machinable volume is the set of all points (x, y, z) where $z \leq R_H(x, y)$. Therefore we obtain a three dimensional voxel representation of the maximal machinable volume (Equation 3) using image calculations.

Note that while z -buffer algorithms have been used to compute rasterized tool paths *after* process planning ([3], [25]), we use the voxel representation of maximal machinable volume as the input to a search algorithm that rapidly evaluates process plans that may be then fed to a tool path planning algorithm.

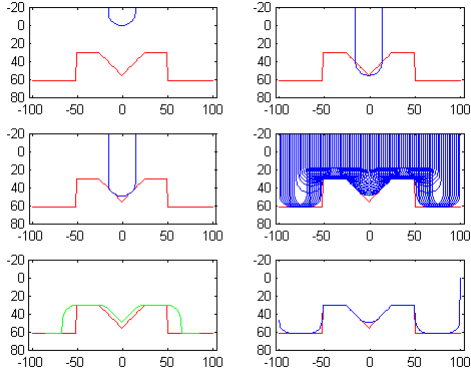


Fig. 3. 1D Example of tool sweep calculation using height maps. Upper-left: tool at $(X, Y, Z = 0)$, Upper-right: tool Z initialized to part heightmap value, Middle-left: tool translated away from part by the penetration depth, Middle-right: tool at (X, Y, Z_{max}) for all (X, Y) , Bottom-left: Z_{max} for all (X, Y) , Bottom-right: the removal heightmap, i.e. the Z_{max} heightmap dilated by the shape of the tool

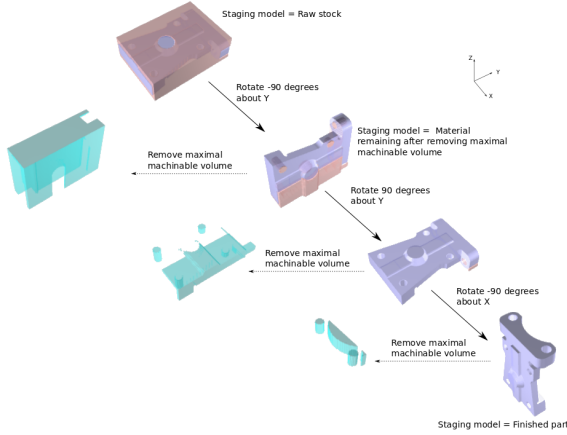


Fig. 4. An automatically computed process plan. The plan shows how material is iteratively removed from staging models to the finished part. Each step of the plan involves reorienting the staging model from the previous step and removing the maximal machinable volume in the specified orientation.

V. COMPUTING PROCESS PLANS FROM MAXIMAL MACHINABLE VOLUMES

A. Problem Definition

We cast the problem of generating a valid process plan as a planning problem, as studied in the artificial intelligence community [7], and solve it using a variant of weighted A* search [18]. That is, we cast it as the problem of finding a sequence of *actions* from a set of available actions A , called the *plan*, that, starting in a known *initial state* I , reaches a state where a *goal condition* $G(s)$ is met. We do so while minimizing a *cost function* $C(a, s)$ that describes the cost of executing a particular action a in a particular state s .

A state S is a tuple $\langle V, T, \theta \rangle$ where V is a set of voxels (“voxel volume”) describing the remaining material to be removed, T is the current tool or \perp to indicate that no tool is being held by the machine currently, and θ is the current rotation applied to the tool with respect to the voxel volume.

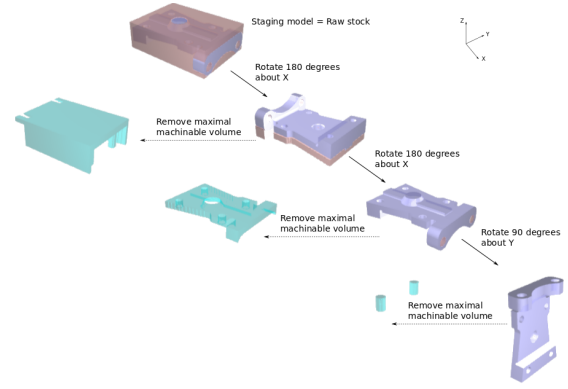


Fig. 5. Qualitatively distinct plan compared to Figure 4.

The voxel volume V_I in the initial state I is the negative of the part S with respect to its bounding box¹, $T = \perp$, and $\theta = (0^\circ, 0^\circ)$.

An action is a tuple $\langle V, T, \theta \rangle$ where V is the set of voxels corresponding to Equation 3, i.e., the maximal sub-volume of the initial removal volume that can be removed using the tool T in orientation θ . The effects of an action are described by the transition function $\tau(s, a)$ that, given a state and an action returns the *successor state*, s' .

$$\tau(\langle V_s, T_s, \theta_s \rangle, \langle V_a, T_a, \theta_a \rangle) = \langle V_s \setminus V_a, T_a, \theta_a \rangle$$

The action removes from the set of remaining voxels V all the voxels the given tool can reach in its given orientation. Note that the successor state assumes the tool and tool orientation of the action. This means that an action can imply a reorientation of the tool holder relative to the part, or, equivalently, a part orientation relative to the tool holder. This implicit change is taken into account by the cost function, which recognizes that reorienting the part takes time, denoted $t_\theta(\theta_a, \theta_s)$, as an additional setup will be required that needs to be prepared. Similarly, if the machine does not have an automatic tool changer, then a tool change can incur additional time-cost, $t_T(T_a, T_s)$. The cost function is

$$C(\langle V_s, T_s, \theta_s \rangle, \langle V_a, T_a, \theta_a \rangle) = t_V(V_s \cap V_a, T_a) + t_T(T_a, T_s) + t_\theta(\theta_a, \theta_s)$$

where $t_V(V, T)$ is a function that estimates the time to remove the volume V using tool T [2]. The goal condition $G(s)$ is simply that the set of remaining voxels is empty: $G(\langle V_s, T_s, \theta_s \rangle) \equiv V_s = \emptyset$.

B. Solution Approach

The search space for process planning problems is too large to search exhaustively in practice. Considering only 8 required access directions, and 5 tools, we get a ramification factor of 40, which means there are $40^{10} > 1e16$ different plans of length 10. We solve the process planning problem in practice using a variant of weighted A* search, using the number of remaining voxels to construct a heuristic function

¹For cylindrical parts a bounding cylinder can be used, but for simplicity of exposition in this paper we limit ourselves to boxes.

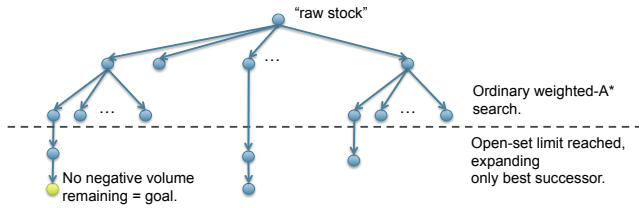


Fig. 6. Notional shape of the search tree resulting in our search approach.

that can effectively guide the search. Additionally, we prune the search by imposing an upper limit on the size of the open-set (search frontier) to 1000. This causes the search tree to take a shape as described in Figure 6. The search will branch out quickly at first, but then become greedy and only consider the best successor (i.e. most cost effective action) of each state. This produces near-optimal plans in practice because the heuristic is very good at estimating the actual cost of removing the remaining material. Additionally, we run a pure greedy search before we run this search, to establish an upper bound on the cost of the best plan. Greedy search is extremely fast, on the order of 0.1 seconds in practice for grids with 200^3 voxels, and the result can be used as a bound for branch-and-bound, i.e., for pruning candidate plans whose aggregate cost plus heuristic value already exceed the bound. Combining these measures we are able to compute process plans in usually less than 60 seconds for realistic 3-axis machined parts. Figures 4 and 5 show simplified examples found using this algorithm.

Also note that in comparison to feature based approaches, where Boolean operations must be performed on boundary representations, in our approach Booleans are reduced to operations on binary sets, which are computed very quickly.

VI. FIXTURE PLANNING

Fixture planning is a largely experience driven activity that focuses on determining precise workpiece clamping according to process constraints such as collision avoidance with the moving tool, accessibility to specific locations, tolerances etc. It is a key part of machining process planning, and offers significant scope for automation. We now briefly discuss automatic workpiece holding using modular fixtures and vises with emphasis on collision avoidance with moving tools. In particular we show the algorithms to compute maximal machinable volumes may be applied with some modifications to compute vise fixtures.

A. Modular Fixtures

Rigorous analysis to automatically generate modular fixtures are discussed in our previous work [17]. Here we briefly describe the salient points of automatic modular fixture configuration relevant to process planning. Assuming a polyhedral representation of staging models and the maximal machinable volume, we apply principles of force and form closure to assert that seven points on the boundary of the part (outside the machinable volume) are necessary and sufficient to grasp the staging model such that it remains immobile

under external forces and moments. Essentially, unit forces applied to sample locations on the part boundary have an associated moment about the part centroid, and the six-tuple of the force and moment components defines a point or a *wrench* in a six dimensional *wrench space*. Form closure is equivalent to the condition that the 6×7 *wrench matrix* (whose columns are sampled wrenches) has a nullspace with purely positive coefficients. In [17] we show how the wrench space may be partitioned and searched efficiently to rapidly generate wrench matrices satisfying form closure.

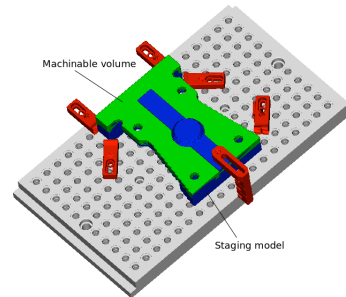


Fig. 7. Automatic assembly of modular fixtures in form closure while avoiding tool collision during machining. The base plate is extensible.

When friction between clamps and the part surface is considered, four points are sufficient to guarantee form closure. Tangent forces due to Coulomb friction may be used to generate nullspaces with positive coefficients. The coefficients of the wrench matrix may be thought of as scaling factors applied to clamping forces, which are always bounded in practice. Therefore under operating conditions we use a restriction of form closure to force closure, which bounds the total forces and moments that the fixtured part can withstand.

We then map the chosen form/force closure configuration to a catalog of modular fixtures including several types of clamps (edge clamps, strap clamps, side clamps) and locator pins to choose the proper arrangement of clamps to avoid tool collisions. Modular fixtures offer the significant advantage of reconfigurability but are often more expensive, labor-wise, for a shop to assemble compared to solutions like vises.

B. Vise Fixturing

Vise fixturing is more popular in practice because of setup convenience but only provides force closure which is usually sufficient for practical situations. A pair of parallel plates are mated against flat surfaces on the part and stock and positioned to maximize the frictional clamping forces while ensuring that the cutting tool will not collide with any part of the vise. Given an orientation for the vise relative to the part, we treat the vise jaws like undercut free tools so that we may use z-buffer algorithms as in Section IV, with the addition of a counter to record the number of surface points of the vise that are coincident with the surface of the part.

The orientation of the vise relative to the part can be chosen based on many criteria; we assume the vise base lies on a plane orthogonal to the tool access direction and

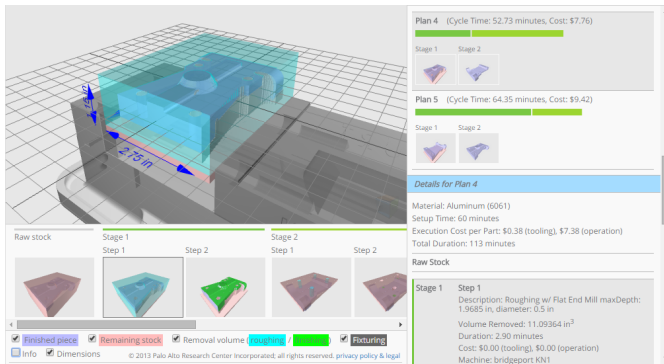


Fig. 8. GUI for process planning system showing multiple process plans (top-right), graphical and textual descriptions of plan steps (bottom-left and bottom-right, respectively), and 3d visualization of removal volumes, staging models, and fixturing for each stage of a plan (top-left)

test orientations are chosen from the surface normals of the part which are parallel to the base plane. Heuristics such as surface area can be used to greedily sort the test orientations. Furthermore, the fact that a vise jaw is flat and, generally, large relative to the part enables significant optimizations over checking of all surface points of the vise against all corresponding surface points of the part (such as described for tools in [25])

To guarantee collision avoidance with the cutting tool, we perform this z-buffer max depth and overlap count from the perspective of both vise jaws and on both the part and the removal volume. The (X, Y) translation, and corresponding vise opening, with the largest overlap with the part and zero overlap with the removal volume is selected as the ideal placement of the vise chosen from a catalog.

VII. CONCLUSION

We have demonstrated algorithms for automatic process plan generation with fixtures for machining operations. The algorithms have been combined into a spatial planning system that solves the problem of detailing machining plans ranked by time/cost for a user specified design. The proposed formulations do not rely on recognizing information from specific part representations e.g. through feature detection, and implementations are performed rapidly in voxel representation either using convolutions or with (faster) dimensional reduction and visibility analysis when undercut free tools are usable. The system scales to complex geometries and generates process plans using a search algorithm that quickly evaluates time/cost optimal manufacturing actions. Fixtures are then automatically generated for manufacturable parts such that the fixture assembly does not collide with the moving tool. Inaccessible regions for non-manufacturable parts are highlighted and the output of the spatial planning may be directly sent to tool path generating software. To the best of our knowledge this is the first automatic process planning system for milling operations that combines design geometry with manufacturing capabilities to outline shop specific process plans.

REFERENCES

- [1] B. Babic, N. Nestic, and Z. Miljkovic. A review of automated feature recognition with rule-based pattern recognition. *Comput. Ind.*, 59(4):321–337, 2008.
- [2] G Boothroyd, P Dewhurst, and W Knight. *Product design for manufacture and assembly*. CRC Press, 2010.
- [3] J Carter, T Tucker, and T Kurfess. 3-axis cnc path planning using depth buffer and fragment shader. *Comput.-Aided Des. Applic.*, 5(5):612–621, 2008.
- [4] C Chu and R Gadh. Feature-based approach for set-up minimization of process design from product design. *CAD*, 28(5):321–332, 1996.
- [5] M Eftekharian, Aand Campbell. Convex decomposition of 3d solid models for automated manufacturing process planning applications. In *ASME 2012 IDETC/CIE*, pages 727–735. American Society of Mechanical Engineers, 2012.
- [6] W. Fu, P. Eftekharian, A. Radhakrishnan, M. Campbell, and C. Fritz. A graph grammar based approach to automated manufacturing planning. In *ASME 2012 IDETC/CIE*, pages 77–88. American Society of Mechanical Engineers, 2012.
- [7] M Ghallab, D Nau, and P Traverso. *Automated planning: theory & practice*. Elsevier, 2004.
- [8] SK Gupta, W Regli, D Das, and DS Nau. Automated manufacturability analysis: a survey. *Research in Engineering Design*, 9(3):168–190, 1997.
- [9] J-H Han, M Pratt, and WC Regli. Manufacturing feature recognition from solid models: a status report. *Robotics and Automation, IEEE Transactions on*, 16(6):782–796, 2000.
- [10] S Joshi and T-C Chang. Graph-based heuristics for recognition of machined features from a 3d solid model. *Comput.-Aided Des.*, 20(2):58–66, 1988.
- [11] Y-S Kim. Recognition of form features using convex decomposition. *Comput.-Aided Des.*, 24(9):461–476, 1992.
- [12] J-C Latombe. Robot motion planning. 1991.
- [13] T Lozano-Perez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, 100(2):108–120, 1983.
- [14] M. Lysenko, S. Nelaturi, and V. Shapiro. Group morphology with convolution algebras. In *ACM Symposium on Solid and Physical Modeling 2010*, 2010.
- [15] S. Nelaturi. Configuration modeling. *Ph.D. Thesis, University of Wisconsin-Madison*, 2011.
- [16] S Nelaturi, M Lysenko, and V Shapiro. Rapid mapping and exploration of configuration space. *J. Comput. Inf. Sci. Eng.*, 12(2), 2012.
- [17] S. Nelaturi, A. Rangarajan, C. Fritz, and T. Kurtoglu. Automated fixture configuration for rapid manufacturing planning. *Comput.-Aided Des.*, 46:160–169, 2014.
- [18] Ira Pohl. Heuristic search viewed as path finding in a graph. *Artif. Intell.*, 1(3):193–204, 1970.
- [19] AAG Requicha. *Representations of rigid solid objects*. Springer, 1980.
- [20] J. Serra. *Image Analysis and Mathematical Morphology- vol 1*. Academic Press, 1982.
- [21] JJ Shah. Assessment of features technology. *Comput.-Aided Des.*, 23(5):331–343, 1991.
- [22] JJ Shah. *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*. John Wiley & Sons, 1995.
- [23] AJ Spyridi and AAG Requicha. Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines. In *Proceedings of IEEE ICRA*, pages 1284–1289, 1990.
- [24] S Subrahmanyam and M Wozny. An overview of automatic feature recognition techniques for computer-aided process planning. *Comput. Ind.*, 26(1):1–21, 1995.
- [25] JA Tarbuton, T Kurfess, and T Tucker. Graphics based path planning for multi-axis machine tools. *Comput.-Aided Des. Applic.*, 7(6):835–845, 2010.
- [26] Y-J Tseng and SB Joshi. Recognizing multiple interpretations of interacting machining features. *CAD*, 26(9):667–688, 1994.
- [27] JH Vandenbrande and AAG Requicha. Spatial reasoning for the automatic recognition of machinable features in solid models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(12):1269–1285, 1993.
- [28] TC Woo. Feature extraction by volume decomposition. In *Conf. on CAD/CAM technology in mechanical engineering*, pages 76–94, 1982.
- [29] TC Woo. Visibility maps and spherical algorithms. *Comput.-Aided Des.*, 26(1):6–16, 1994.
- [30] M-C Wu and CR Lit. Analysis on machined feature recognition techniques based on b-rep. *Comput.-Aided Des.*, 28(8):603–616, 1996.