Segmentation and Coverage Planning of Freeform Geometries for Robotic Surface Finishing

Stefan Schneyer[1], Arne Sachtler[1,2], Thomas Eiband[1], and Korbinian Nottensteiner[1]

Abstract—Surface finishing such as grinding or polishing is a time-consuming task, involves health risks for humans and is still largely performed by hand. Due to the high curvatures of complex geometries, different areas of the surface cannot be optimally reached by a simple strategy using a tool with a relatively large and flat finishing disk. In this paper, a planning method is presented that uses a variable contact point on the finishing disk as an additional degree of freedom. Different strategies for covering the workpiece surface are used to optimize the surface finishing process and ensure the coverage of concave areas. Therefore, an automatic segmentation method is developed to find areas with a uniform machining strategy based on the exact tool and workpiece geometry. Further, a method for planning coverage paths is presented, in which the contact area is modeled to realize an adaptive spacing between path lines. The approach was evaluated in simulation and practical experiments on the DLR SARA robot. The results show high coverage for complex freeform geometry and that adaptive spacing can optimize the overall process by reducing uncovered gaps and overlaps between coverage lines.

Index Terms—Intelligent and Flexible Manufacturing, Contact Modeling, Motion and Path Planning

I. INTRODUCTION

SURFACE FINISHING is a crucial step of many manufacturing processes [1] and includes tasks like smoothing rough surfaces, polishing, and varnishing/coloring materials. Sanding is an example of surface finishing in the manufacturing of wood products [2]. The quality of the product depends on the surface treatment [3], which is a complex task for concave shapes and fine geometric features. Therefore, surface finishing is largely performed manually by human workers and is one of the least automated processes [1], [4]. However, automation of surface finishing offers many benefits to both workers and the economy. It can reduce the health risk for workers, as they are no longer exposed to hazardous dust and noise. Further, it can increase efficiency, reduce costs and offers opportunities for higher accuracy and quality in the manufacturing process. Robotic systems provide the general flexibility to automate surface finishing tasks with complicated

Fig. 1. Overview of the planning approach for surface finishing tasks. The process is divided into subprocesses: analyzing possible solutions for different contact points, partitioning the surface into segments and planning a suitable coverage path.

geometries [5]–[7]. Nevertheless, for small batch or one-of-a-kind production, robotic solutions should be intuitive to operate and easy to reprogram or adapt to new tasks. This work presents an approach to reduce the manual planning and programming effort for robotic surface finishing.

Problem Statement: In surface finishing, the geometry of the surface in combination with the tool geometry plays a crucial role. It is straightforward to program a path for flat or weakly curved surfaces, but complex for large and non-constant curvatures, such as concave shapes, edges, and other fine geometric features. Simple strategies like following the surface normal with the center point of a finishing disk are typically not possible anymore. In addition, the area of contact changes depending on the curvature strength, causing the treated area to vary even for convex shapes. Thus, methods are required to find suitable tool trajectories and facilitate robotic surface processing.

Contribution: We propose an approach for surface segmentation and coverage planning for robotic surface processing of complex-shaped geometries (Fig. 1). The focus is on varying the contact point on the finishing disk as an additional degree of freedom to machine different parts of the surface. The first contribution is a planning algorithm to determine the optimal contact point and orientation of the tool based on the local geometry, and divide the surface accordingly into segments with similar processing strategies. For example, fine concavities can be handled through edge contact, although a relatively large finishing disk is used. The geometry of the workpiece and finishing disk is taken into account in the segmentation process, instead of using only the normal

vectors and the curvature as approximate features [8]–[10]. In addition, this work goes beyond related work by considering variations of the contact points, including the center and edge contacts, rather than just one predefined contact point for processing the entire surface. The second contribution is a method to model the expected contact area and use it for a coverage planning approach with variable spacing between parallel tool paths. In contrast to related approaches [5], [11], no constant curvature of the surface is assumed. An adaptive approach is used to optimally determine the spacing between parallel coverage lines, with the goal of completely covering the surface, avoiding overlapping, and minimizing machining time. Both major contributions are evaluated in simulation and in real experiments.

The structure of this work is as follows. Related work is presented in Section II. Then, we describe the approach for surface segmentation in Section III and for coverage planning in Section IV. The results of the evaluation are provided and discussed in Section V. Finally, we conclude the work in Section VI and give an outlook for future work.

II. Related Work

Segmentation is a pre-processing step that divides the surface into segments that can be handled with similar strategies, and is typically applied for complex-shaped geometries. The approach in [9] decomposes a geometry given as a CAD model into segments that are simple to process and focuses on flat regions only. Segmentation based on curvature is described by [10], and [8] uses k-means clustering to partition the surface points using their coordinates and normal vectors as feature vectors. A different segmentation approach is shown in [12] and [13] for a spraying task, where curved surfaces are segmented into topologically simple and monotonic cells. This results in segments with simple geometries without holes and is achieved by an algorithm based on the Boustrophedon decomposition method [14] generalized to non-planar surfaces. In [6], for the segmentation of eyeglass frames for polishing tasks, workpiece symmetry is considered as a criterion in addition to the surface normal vector and surface edges. A metric is proposed in [7], which includes the deviation of normal vectors besides the distance of surface points. Then, subregions with relatively small curvature can be extracted by identifying the points that lie below a threshold given this metric. In the works listed above, only features of the workpiece geometry are taken into account for segmentation, but not the geometry of the tool.

Coverage path planning aims to efficiently cover a surface in tasks such as sanding, grinding, or painting, considering factors like collision-free traveling, minimal time and energy cost, and low overlap rate [15]. General approaches are presented in [16] and include heuristic, randomized, and cellular decomposition methods. An important step is to find an appropriate representation of the surface or grid system on which the coverage path can be planned using typically the assumption of a constant contact area. In [17], various methods for covering surfaces and generating raster paths for polishing are presented. The surface is first mapped into a two-dimensional space for planning and later transferred back to the three-dimensional space. Space-filling curves such as raster paths, Lissajous curves and Peano curves are used as patterns for planning in the two-dimensional parametrization. Using a projection to obtain a two-dimensional surface parametrization will result in a distorted distance measure on highly curved surfaces and a non-uniform grid on the surface. This leads to difficulties in coverage path planning since orientations and spacing are not maintained, with the consequence that the contact area is distorted and varies significantly. In [18], a method is presented to generate a uniform grid on freeform surfaces by investigating three scenarios where an in-plane freeform curve is transformed into a surface. However, uniform grid mappings can not be found for arbitrary surface shapes. A more general approach for finding a good parametrization can be achieved by optimization. The method As-Rigid-As-Possible [19] applies optimization techniques to find a mapping that is as isometric as possible [20]. While efficient methods for surface representation exist, the approaches listed above still make the simplifying assumption of constant contact areas.

Instead of assuming constant contact area during surface processing, further related works explore the idea of varying areas of contact in coverage planning. As shown in [11] and [5], the contact area can be determined under the assumption of constant curvature, and can be considered further in path planning, using an adaptive spacing between path lines. In these approaches, no segmentation is performed and only the center contact point is used, whereby a workpiece geometry may only be weakly curved relative to the tool size. In [21], in addition to the area of contact, the removal depth is studied and a non-constant pressure distribution model is considered in the area of contact. The approach in [22] presents a model to estimate the area of contact and predict surface quality based on process parameters like sandpaper roughness and rotation speed. In our approach, no constant curvature is assumed and the contact area can be approximated for arbitrary contact points on the finishing disk.

III. Surface Segmentation

For simplicity, a constant reference point on the tool is typically used as the contact point in robotic surface processing [5], [7], [17]. A common choice is the center of the finishing disk. However, using a constant contact point constrains the possible configurations relative to the surface and thus leads to suboptimal solutions for geometries with strong variations in shape. Varying the contact point location relative to the tool as an additional degree of freedom can therefore handle more complicated geometries, for example, by using an edge contact instead of the center point contact. In the following, the geometric surface and tool properties will be analyzed in more detail to obtain a segmentation for different contact points.

A. Solution Map Creation for Grid Points on Mesh

The uniform and circular finishing disk of the considered tool is rotationally symmetric. Thus, the set of possible contact points can be reduced from the entire surface of the tool disk to a line between the center and an edge point. In the extreme case of an edge contact, the tool can be tilted so

that a second undesired collision with the surface is avoided. Consequently, the line of possible contact points can further be reduced to the two extreme points representing edge and center contacts (Fig. 2a). While the center contact point features a fixed configuration of the symmetric disk relative to the tangent plane, an edge contact adds two degrees of freedom and opens up new possibilities for processing. In this case, the tool configuration is specified by the direction angle β (Fig. 2b), which describes the rotation around the normal vector of the contact point and is measured with respect to the x-axis of the surface parametrization. In addition, the angle α represents the tilt of the tool with respect to the tangent plane in the contact point. Note that a third degree of freedom is the rotation around the normal vector of the disk and is used in practice to avoid collisions of the overall tool with the workpiece, but is not further considered at this point due to the symmetry of the tool disk.

(a) Center (left) and edge point contact (right) with the tilt direction (blue arrow). The orange line represents the set of possible contact points, which is reduced to two extreme points.

(b) Edge contact orientation   (c) Sampling β orientations
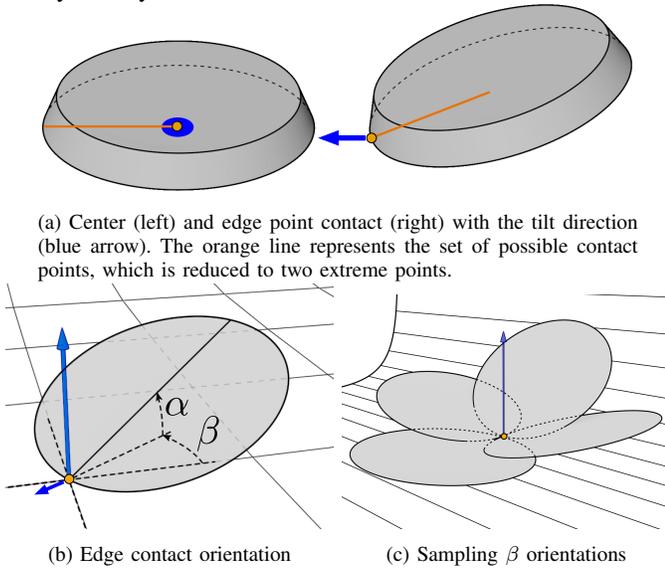
Fig. 2. Different contact points on the tool disk (a), defining orientation for edge contact (b) and illustration of sampling different β orientations (c)

The geometry of the workpiece is analyzed using a uniform grid $\mathcal{G}$ on the two-dimensional surface parametrization, which is generated using the As-Rigid-As-Possible method [19]. The grid points with indices $i, j \in \mathbb{N}_0$ are generated at equal distances. Further information can be appended at each point and stored together in a map. Based on that, we can specify a class map $C_{\mathcal{G}}$ which assigns the preferred strategy to each grid point:

$$C_{\mathcal{G}} : \mathcal{G} \to \{None, Center, Edge\}, \qquad (1)$$

where $\mathcal{G}$ is the set of all grid points.

Class *None*: The grid point is not on the surface or no solution was found.
Class *Center*: The grid point is machined with the center point contact of the finishing disk (preferred).
Class *Edge*: The grid point is treated with edge contact of the finishing disk (only if *Center* is not possible).

Further, a solution map $\Omega_{\mathcal{G}} : \mathcal{G} \times \mathbb{D}_{\beta} \to \mathbb{D}_{\alpha}$ assigns for each grid point $(i, j) \in \mathcal{G}$ and all β orientation angle values (index

$k \in \mathbb{D}_{\beta} = \{1, \cdots, N\}$ out of N sample steps) the optimal angle $\alpha_{\text{opt}} \in \mathbb{D}_{\alpha} = [0, 45°]$ in case of edge contact as

$$\Omega_{\mathcal{G}} : (i, j, k) \mapsto \alpha_{\text{opt}}. \qquad (2)$$

The optimal angle α_{opt} is the smallest angle that does not cause a second collision besides of the edge contact. A small angle ensures a stable contact and has a larger contact area which leads to a more uniform machining of the surface. For a given β_k value an optimal angle α_{opt} can be determined within a limit $\alpha_{\max} = 45°$:

$$\alpha_{\text{opt}} = \min \{\alpha \in \mathbb{D}_{\alpha} \mid (\alpha, \beta_k) \in \mathcal{C}\}. \qquad (3)$$

The set $\mathcal{C}$ contains all feasible configurations (α, β) of the disk, in which the geometry touches the workpiece only at the specified contact point.

To find the optimal solution, we propose a method that uses collision checks of the exact geometries instead of simplified metrics. We use the Flexible Collision Library[1] to determine a collision between the finishing disk and the workpiece. Since the library cannot distinguish between desired and undesired contacts, we move the disk away from the surface by a minimal distance along the normal vector in the contact point. In order to solve (3), we then uniformly sample β orientations (Fig. 2c) and search for the smallest angle α that avoids collision using binary search.

B. Surface Segmentation

The goal of surface segmentation is to find segments that can be processed with similar strategies. However, since the best value α_{opt} per grid point and the associated value of β are not necessarily optimal for the neighboring points, several solutions per grid point are kept to prepare the segmentation.

For each grid point (i, j) the smallest angle for all directions is given by

$$\alpha_{\min} := \min_k \Omega_{\mathcal{G}}(i, j, k). \qquad (4)$$

From this, a threshold value can be calculated as

$$\alpha_{\text{thres}} = \alpha_{\min} + \lambda(\alpha_{\max} - \alpha_{\min}), \qquad (5)$$

given a certain percentage limit λ. The set of best solutions per grid point is called $\mathcal{B}_{(i,j)}$ and contains all direction angle steps of β for which the tilt angles α are within the threshold α_{thres}:

$$\mathcal{B}_{(i,j)} = \{k \mid \Omega_{\mathcal{G}}(i, j, k) < \alpha_{\text{thres}}\}. \qquad (6)$$

In order to establish a consensus among the different solutions, we first construct a graph $G_{\text{parts}} = (V, E, w)$, where V and E are nodes and edges of the graph, and w represents the weights of the edges. The nodes V denote the grid points. The edges $E \subseteq (V \times V)$ are the set of tuples of two nodes that are neighbors in the surface grid, where each node has at most four direct neighbors. Each edge is assigned a weight by a function $w : E \to \mathbb{R}$, which corresponds to the Intersection over Union (IoU) of the best solutions and can be defined for an edge $e = (v_1, v_2)$ as

$$w(e) = \frac{|\mathcal{B}_{v_1} \cap \mathcal{B}_{v_2}|}{|\mathcal{B}_{v_1} \cup \mathcal{B}_{v_2}|}. \qquad (7)$$

[1] https://github.com/BerkeleyAutomation/python-fcl

Besides this definition, there are some exceptional cases. For the center point contact, the weight between two grid points is always one, and for two different strategies, the weight becomes zero to prevent both points from being part of the same segment. In addition, the weight is set to zero if an edge is detected in the geometry of the workpiece between the two grid points. This is the case when the angle between the normal vectors of the grid points exceeds a threshold.

Based on the information of the graph, an algorithm is developed which finds a consensus of solutions among multiple nodes, i.e., equal processing directions, and thus generates a segmentation where the parts can be treated uniformly (Algorithm 1). The primary objective of the algorithm is to reduce the number of segments while ensuring that each segment includes at least one common processing direction for all the points it contains. Initially, each node is considered as an own segment of the surface. In each step, two segments are merged, but only if they have at least one common direction ($w(e) > 0$). After each merge, the set of possible solutions is reduced to the intersection, which also changes the IoU to the neighboring segments.

Algorithm 1 Consensus merging

1: **function** CONSENSUSMERGING(graph (V, E, w))
2: $\quad S \leftarrow V$ $\qquad$ ▷ set of segments: initially the nodes
3: $\quad \forall s \in S : D_s \leftarrow \mathcal{B}_s$ ▷ directions $\mathcal{B}_v$ for each segment
4: $\quad E \leftarrow$ SORTDESCENDINGBYWEIGHT(E, w)
5: $\quad$ **for all** $e = (s_1, s_2) \in E$ **do**
6: $\qquad$ **if** $w(e) = 0$ **then**
7: $\qquad\quad$ exit loop $\qquad$ ▷ since the edges are sorted
8: $\qquad$ apply redirection to segments s_1, s_2
9: $\qquad$ **if** $s_1 = s_2$ or $D_{s_1} \cap D_{s_2} = \{\}$ **then**
10: $\qquad\quad$ continue to next element in loop
11: $\qquad$ $(S, s_{\text{new}}) \leftarrow$ MERGESEGMENTS(s_1, s_2)
12: $\qquad$ $D_{s_{\text{new}}} \leftarrow D_{s_1} \cap D_{s_2}$
13: $\qquad$ redirect s_1, s_2 to new segment s_{new}
14: $\quad$ **return** (S, D)

The order in which the segments are merged is essential for the final result of the segmentation. Moreover, it can be observed that this is a special kind of graph partitioning problem. However, finding an optimal solution to a graph partitioning problem is known to be NP-hard [23]. Since this problem cannot be solved efficiently for a guaranteed optimal solution, a heuristic is used to decide on the merging order of graph nodes. The heuristic assures that graph nodes with a higher IoU metric w are merged first.

IV. COVERAGE PLANNING

In general, a planning procedure is required to ensure efficient surface finishing, avoid overlaps, reduce unprocessed gaps, and minimize the length of the processing path. The challenge of finding the path with minimum cost through different surface points corresponds to the Traveling Salesman Problem (TSP), which is known to be NP-hard [24]. The use of an approximate TSP solver is not appropriate in this situation because heuristics are difficult to tune for this process and can

affect the quality of surface finishing. Moreover, in the case of surface finishing, it is not enough to minimize only the path length. We also need smooth tool paths and intend to take into account the changes in the contact area during the process. Therefore, a grid path, where parallel lines are connected to form an overall path, is used as the underlying pattern in the following, which reduces the search space and can lead to tool paths that are more suitable for surface finishing.

Two methods will be investigated for finding coverage paths for surface segments: constant contact area with uniform spacing, and variable contact area to tackle overlap and gaps.

A. Processing Direction on Segments

We assume that there is a constant processing direction for each segment. Therefore, raster paths are used to cover segments because they feature in general a preferred motion direction and can be adapted to different hull geometries. For the two different strategies, edge contact and center point contact, we choose the following processing directions:

1) When machining a part with the edge of the finishing disk, the direction of movement influences the machined area. Choosing the direction along the tilt of the tool results in a wider machining track.
2) For using the center of the disk as the contact point, the optimal machining direction is found by maximizing the length of the coverage lines and minimizing their amount. This is achieved by aligning the direction of motion with the longest side of the smallest rectangle that encompasses the segment shape.

A coordinate system is introduced for each segment, with the x-axis aligned to the processing direction and the y-axis perpendicular to it. The segments are represented by polygons in this frame defining the set of surface points included in each segment in the surface parametrization space.

B. Coverage with Constant Spacing

To simplify the planning algorithm, the first approach assumes a constant contact area during surface processing. Under this assumption, a good solution for a coverage path can be found by cutting the contour in the two-dimensional planning space of each segment layer by layer with a line along the x-axis and constant spacings in y-direction (Fig. 3). To reduce overlapping with neighboring segments, a small distance between the path endpoint and segment boundary is ensured.

C. Model of Varying Contact Area

To improve accuracy and planning efficiency, we analyze and model the contact area of the finishing tool to determine the layer spacing. In practice, the disk is flexible to a certain degree and adapts to the shape of the surface. Therefore, we introduce an elasticity distance ϵ, which represents the compliance of the tool and defines how far the tool and abrasive material can deform under the process forces to conform to the surface. We assume that this distance remains constant even with varying forces. To approximate the contact area,

Fig. 3. Illustration of the method to find coverage lines for a constant spacing. The lines (light green) intersecting the contour (red) with a small distance subtracted from the intersection point to avoid overlap with neighboring segments result in the coverage lines (green).
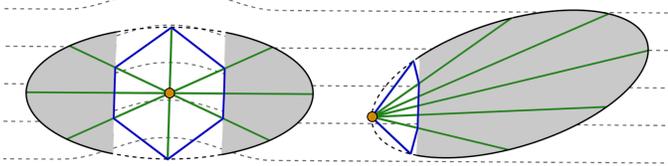
Fig. 4. Illustration of the rays used to approximate the area of contact for a center point contact (left) and an edge contact (right)

we move the finishing disk along its height by ϵ towards the surface and then determine the intersection with the workpiece geometry. Rays are sampled in different directions from the shifted contact point to find the intersection points with the geometry, which describes the edge contour of the contact area as visualized in Fig. 4.

D. Cover Segment of the Varying Contact Area

A raster-like path also serves as a basis to find a coverage path per segment with variable spacing. Coverage lines are created in layers one after the other, using the width of the area of contact as spacing to cover the entire segment step by step. We start with the smallest possible y-values and layer up in the direction of the y-axis. For this purpose, a so-called frontier line is used to describe the current state of the coverage and the starting point for the next layer. The procedure is summarized in Algorithm 2 and an example is shown in Fig. 5.
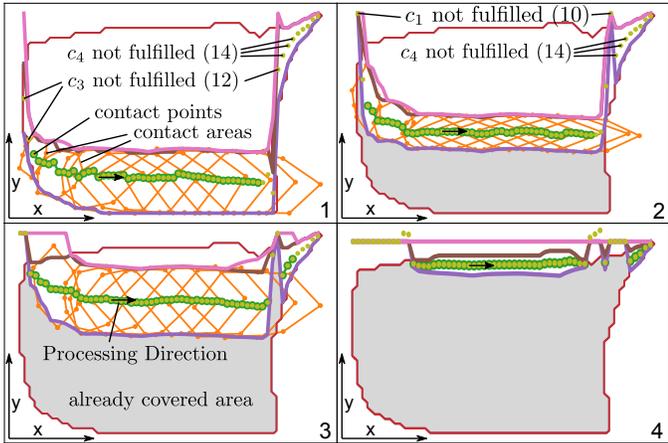
Fig. 5. Covering a segment with variable spacing. For this method, four diagrams of sequential steps are shown, where the contour of the contact area (orange) is placed with an adaptive spacing to the already covered area (frontier line) so that overlaps and gaps are avoided. Each diagram shows the coverage lines for one layer of the method.

In the first step, a list for the lines of the coverage path is initialized and the frontier is set to the boundary of the segment polygon $\mathcal{P}$ from which the first layer should start. The layers are built up starting from the frontier in a loop, whereby further layers are added as long as the smallest frontier value has not yet exceeded the maximum y-value of the segment. For each iteration, the frontier for the next layer is calculated for all x-values in a first step using the contact area. The x-values are discretized equidistantly and are denoted by the set $\mathcal{X}$.

Algorithm 2 Find coverage path for segment

1: **function** FINDCOVERAGEPATH($\mathcal{X}$, $\mathcal{P}$)
2:     $\mathcal{L} \leftarrow \{\}$ ▷ Initialize a set of all coverage lines
3:     $\forall x \in \mathcal{X} : f_x \leftarrow \min\{y \mid (x,y) \in \mathcal{P}\}$
4:     $y_{\max} \leftarrow \max(\{y \mid (x,y) \in \mathcal{P}\})$
5:     **while** $\min(\{f_x \mid x \in \mathcal{X}\}) < y_{\max}$ **do**
6:         $f_{\text{pred}} \leftarrow$ PREDICTNEXTFRONTIER($\mathcal{X}$, f)
7:         $\mathcal{Y} \leftarrow$ FINDLAYERPOINTS($\mathcal{X}$, f_{pred})
8:         $(l, f_{\text{next}}) \leftarrow$ GETLAYERCOVPATH($\mathcal{Y}$, f_{pred})
9:         $\mathcal{L} \leftarrow \mathcal{L} \cup l$
10:       $\forall x \in \mathcal{X} : f_x \leftarrow f_{\text{next}}(x)$
11:     **return** $\mathcal{L}$

In the prediction step, the width of the next layer is determined based on the contact areas to find the next frontier line and the contact points of the coverage line. However, there is a cyclic dependency, since the contact area and the displacement in the y-direction are to be calculated, but they are mutually dependent. Iterative optimization is used to solve this problem because the contact area varies too much to be considered constant. The optimization problem can be formulated as

$$\min \mid f_x - \min(\{y \mid (x,y) \in \mathcal{I}_x\}) \mid \qquad (8)$$

where f_x denotes the y-coordinates of the frontier line and $\mathcal{I}_x$ the contact area for the x-coordinate x. We want to minimize the distance between the frontier and the lower boundary of the contact area in y-direction. The distance is first initialized as half the tool radius, and an iterative procedure adjusts the distance of each x-value to minimize the remaining deviation such that it converges to zero.

Next, the coverage line for the current layer can be created. However, combining all points into one path line is problematic because the points may vary significantly due to segment boundaries and contact areas of different sizes. Four criteria are defined to decide which points along the x-direction should belong to the next path line. The first criterion indicates if for an x-value the segment is already completely covered, which is given as

$$f_{\max} := \max(\{f_x \mid \forall x \in \mathcal{X}\}), \qquad (9)$$
$$c_1 :\Leftrightarrow f_x \geq f_{\max}. \qquad (10)$$

Furthermore, a point (x, y) outside the segment polygon $\mathcal{P}$ should not be added to the line:

$$c_2 :\Leftrightarrow (x, y) \notin \mathcal{P}. \qquad (11)$$

A distance d should be left to the edges in the x-direction, so that points (x, y) closer to the side border should be excluded:

$$c_3 :\Leftrightarrow (x - d, y) \in \mathcal{P} \wedge (x + d, y) \in \mathcal{P}. \qquad (12)$$

Points with a significantly smaller frontier value f_x should be processed first to compensate for large jumps of the tool in y-direction, which can be caused by rapidly changing contact area or segment boundaries. Lines too far from the minimum frontier value should be excluded. The criterion is

$$x_{\min} := \arg \min_x (\{ f_x \mid \forall x \in \mathcal{X} \}), \tag{13}$$

$$c_4 :\Leftrightarrow f_x - f_{x_{\min}} > \text{width}(\mathcal{I}_{x_{\min}}). \tag{14}$$

Points that do not fulfill any of the four conditions are added to the coverage path of the layer, with possible interruptions in the line. The frontier f_{next} differs from f_{pred} due to the exclusion of points by the criteria and also takes into account the width in x-direction of the contact area.

V. EXPERIMENTAL EVALUATION

A. Segmentation Evaluation

Simple Geometries: First, we evaluate the segmentation method with simple geometries as shown in Fig. 6a. For geometries such as a convex or concave cylinder, a single segment is created using only a *Center* or an *Edge* processing strategy, respectively. The height profile of a Gaussian curve with a varying curvature results in multiple segments. The two outer segments (1 and 4) can be machined with the center of the disk, while the middle zone is divided into two segments (2 and 3), which are reachable best using the edge contact from opposite directions.

Freeform Geometry: The segmentation of a complex freeform geometry is shown in Fig. 6b. The surface features various shapes such as soft and hard edges, convex and concave sections, and double-curved areas. A grid resolution of 5mm is used for the workpiece, which has a size of 50x50x15cm, and we choose λ at 30% for the sets of best solutions, compare (5). For example, regions B and C can be machined with center point contact due to convexity, as can region A, which is further subdivided by edge detection of the surface geometry. The segments 8 and 2 are examples of using the edge contact. Area D has fine features resulting in a large number of small segments, which are challenging to treat efficiently with the relatively large tool and will be excluded for further automated processing. In area E, one segment can be machined with center point contact, and three further segments use edge contact. Segments in the concave double-curved region use edge contact, with the orientation of the tool tilt towards the center (point F).

B. Path Coverage Evaluation

The path coverage is evaluated in simulation and practical experiments on the real robot. In the simulation, a coverage map, as shown in Fig. 7, is generated to analyze and visually represent how the finishing process treats various points on the surface. The intensity of the processing is indicated at each point of the map with a treatment measure $T(p, \Delta t)$ which is a function of the applied pressure p and the total duration Δt of processing at a point. A simulated trajectory is first created in which a constant pressure is applied during the process. In addition, the simulated velocity is chosen so that the time,

(a) Simple Geometries

(b) Freeform Geometry
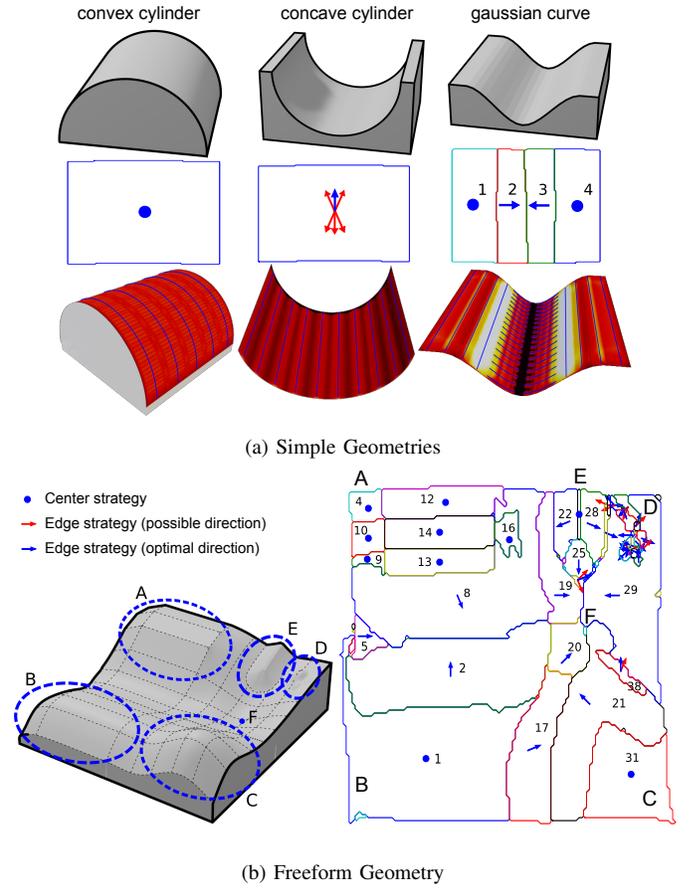
Fig. 6. Segmentation of simple geometries (a) and a more complex freeform shape (b). The blue circle denotes segments processed with the *Center* strategy and the arrow represents the main processing direction of the *Edge* strategy, with the red arrow indicating a possible direction and the blue one denoting the most optimal of all directions of a segment. For the simple geometries, the simulation results with constant spacing are shown as coverage maps. The color of each map indicates the intensity of processing, with dark colors indicating no processing and bright colors indicating high processing intensity.

the disk acts on a surface point, remains approximately the same for different sizes of contact area. These two constraints of constant pressure and processing time make it possible to investigate overlapping areas in the theoretical model since uniform machining is simulated for contact areas of different sizes. To generate the coverage map, the simulated trajectory is sampled at equal time intervals. Then, for each time step, an incremental value is added to all points of the coverage map that are part of the approximated contact area.

The coverage maps for the simple geometries are shown in Fig. 6a below the corresponding segmentation results. The simulated path is generated by constant spacing, which covers the cylindrical shapes efficiently since constant contact areas are obtained over the entire surface. For the freeform geometry, the path coverage is evaluated in more detail, where three different scenarios are compared: the coverage method with constant spacing with two different parameterizations (large and small distances) and the approach with adaptive spacing. First, we present the results of the theoretical coverage from the simulation. Then, we compare them with the actual coverage achieved in experiments with the real robot.

TABLE I
COMPARISON OF EXPERIMENTS WITH DIFFERENT SPACING

	Simulated Experiment for Whole Surface			Practical Experiment of a Surface Section		
	Large Spacing	Small Spacing	Variable Spacing	Large Spacing	Small Spacing	Variable Spacing
Center point contact spacing	0.067 m	0.045 m	variable	0.067 m	0.045 m	variable
Edge contact spacing	0.042 m	0.014 m	variable	0.042 m	0.014 m	variable
Coverage	88.9 %	96.2 %	91.6 %	89.5 %	96.6 %	94.6 %
Overlapping area	18.9 %	49.2 %	17.7 %	16.4 %	48.5 %	27.0 %
Calculated path length on surface	5.49 m	13.43 m	6.99 m	2.46 m	5.11 m	2.94 m
Planning time (Segmentation)	297 s (for 12769 grid points; 5mm step size; 16 threads)					
Planning time (Trajectory)	22 s	54 s	113 s	7 s	12 s	34 s
Runtime on real robot	-	-	-	183 s	321 s	223 s

(a) Large Spacing (b) Small Spacing (c) Variable Spacing
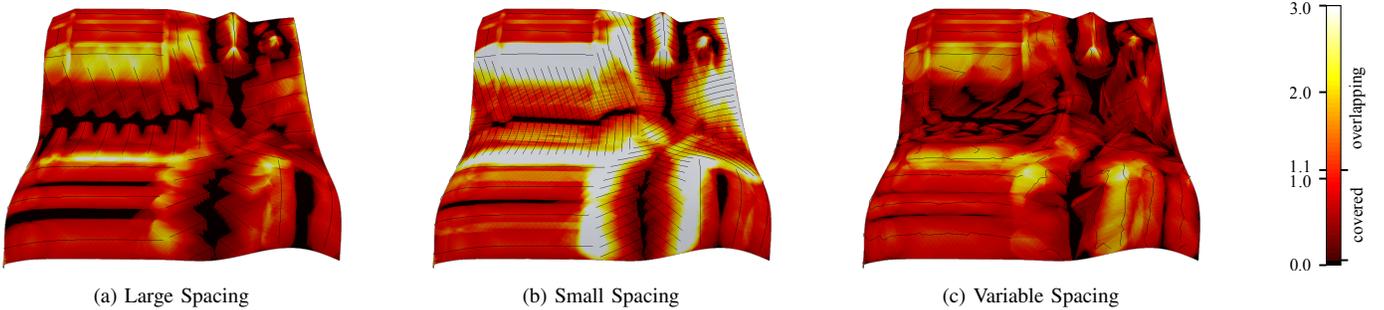
Fig. 7. Simulation results for the entire workpiece in the three different scenarios. Scenarios (a) and (b) show the results for the method with constant spacing, one with large and one with small spacing. In scenario (c), the method with adaptive spacing was comparatively simulated. The value of 1 corresponds to optimal processing, while values at least 10% above this value are considered overlapping and values close to zero are classified as unprocessed.

Simulated Coverage of Freeform Geometry: Table I (Simulated Experiment) provides an overview of the quantitative evaluation criteria for the entire freeform surface for constant spacing with large and small distances, as well as variable spacing. Fig. 7 shows the coverage maps resulting from the simulation. The connections between coverage lines are removed to highlight the effects of the individually planned coverage lines. Variable spacing closes gaps between different paths and reduces overlapping at the same time. However, all approaches still experience overlapping and unprocessed areas at segment boundaries and concave regions. In the regions of strongly changing contact area, e.g. as in the concave shape between region A and B (Fig. 6b), small uncovered gaps between the coverage lines still occur for the variable spacing approach. Thus, the highest coverage of the surface can be achieved by a small constant spacing.

Comparison of Simulated and Practical Coverage: For the surface finishing experiment, the torque-controlled DLR SARA robot with 7 degrees of freedom is used. The primary focus of this work is on path planning, while details of the force control on the SARA robot can be found in [25]. As an end effector of the robot, a Festool Random orbital sander is used (Fig. 8) to which different abrasive materials such as sandpaper or fleece can be attached for surface treatment. The pose of the workpiece can be determined by measuring three tangential planes of the geometry with the end effector of the robot. The workpiece plane is described using the position of the end effector and the normal vector of the finishing disk. To obtain uniform machining, constant pressure is maintained during execution. With changing size of contact area, the force is adjusted so that the corresponding pressure remains constant, given by $F = p_{\mathrm{const}} \cdot A_{\mathrm{contact}}$. In the theoretical and practical comparisons, only a part of the entire geometry

Fig. 8. Surface finishing tool and workpiece of the experiment

is evaluated, since the workspace of the robot did not allow the entire surface of the freeform to be machined without repositioning. To evaluate the effects of the surface treatment on the real robot, grass modeling particles are fixed on the surface which are to be removed by the surface treatment, creating a visual contrast to identify unprocessed areas. An overview of the results is given in Table I (Practical Experiment). A larger spacing results in a shorter runtime and fewer overlaps, but less coverage. A smaller spacing produces longer processing and more overlaps, but better coverage. Variable spacing achieves high coverage and low overlapping with a short runtime, making it a good solution across categories. Further, constant spacing requires expert estimates, while the variable approach is automatically parameterized. Unlike in the purely simulated scenario, the individual coverage lines are connected for efficiency to prevent the robot from lifting off the surface after each line, resulting in additional overlap, which explains the discrepancy between the simulated and practical experiment for variable spacing.

Fig. 9 shows simulated and experimental results for the variable spacing approach. The adaptive method accurately

Fig. 9. Comparison between the simulated coverage map (top) and the result of the real robot execution (bottom) with variable spacing.

determines the spacing and minimizes both overlaps and unprocessed areas in the simulation and experiments on the real robot setup. However, segment boundaries remain a challenge, requiring a compromise between overlap and untreated areas when planning the last path of a segment. In general, it seems impossible to completely avoid overlaps and achieve full coverage with a single tool geometry.

VI. Conclusion and Future Work

An automation method focusing on path planning for robotic surface finishing was presented, which was evaluated in simulation as well as in real-world experiments using a 7-degree-of-freedom compliant manipulator showing almost complete machining of a complicated freeform surface with few exceptions and reduced overlaps between paths. The method can segment the surface and find efficient paths for surface treatment procedures, including varying the contact point on the finishing disk for angled and large concave curvatures. A novel segmentation method has been developed to find regions of the surface that can be processed uniformly and take into account different contact points of the finishing disk. Further, tool path planning methods are presented to cover the segments, including a constant spacing raster path and a method to minimize overlaps and unmachined areas with adaptively determined spacing based on the area of contact.

A limitation of the presented approach is that it cannot be applied to fully closed surfaces or surfaces with holes, as the parameterization is not applicable in these cases. Further, it is assumed that the geometry of the workpiece is known beforehand. Future research could investigate methods to estimate geometry during execution, such as by haptic feedback or a vision system. Additionally, different contact points on the finishing disk could be combined in one segment leading to a smooth transition. Small uncovered areas between segments in concave regions might be encountered with an additional dedicated machining strategy or with specialized tools for precision finishing.

Acknowledgments

References

[1] Y. Li, H. Chen, and N. Xi, "Automatic programming for robotic grinding using real time 3d measurement," in *Int. Conf. Cyber Technol. Autom., Control Intell. Syst.* IEEE, 2017, pp. 803–808.

[2] Y. Huo, P. Li, D. Chen, Y.-H. Liu, and X. Li, "Model-free adaptive impedance control for autonomous robotic sanding," in *Transactions Autom. Sci. Eng.* IEEE, 2021.

[3] Y. Zhao, J. Zhao, L. Zhang, L. Qi, and Q. Tang, "Path planning for automatic robotic blade grinding," in *Proc. Int. Conf. Mechatron. Autom.* IEEE, 2009, pp. 1556–1560.

[4] Y. Dong, T. Ren, K. Hu, D. Wu, and K. Chen, "Contact force detection and control for robotic polishing based on joint torque sensors," *Int. Journal Adv. Manuf. Technol.*, vol. 107, pp. 2745–2756, 2020.

[5] Y. Wen, D. J. Jaeger, and P. R. Pagilla, "Uniform coverage tool path generation for robotic surface finishing of curved surfaces," *IEEE Robot. Autom. Letters*, vol. 7, no. 2, pp. 4931–4938, 2022.

[6] I. Mohsin, K. He, Z. Li, and R. Du, "Path planning under force control in robotic polishing of the complex curved surfaces," *Appl. Sci.*, vol. 9, no. 24, 2019.

[7] M. Xiao, Y. Ding, and G. Yang, "A model-based trajectory planning method for robotic polishing of complex surfaces," *IEEE Transactions Autom. Sci. Eng.*, vol. 19, no. 4, 2022.

[8] X. Liu, Y. Li, and Q. Li, "A region-based 3 + 2-axis machining tool-path generation methodfor freeform surface," *Int. Journal Adv. Manuf. Technol.*, vol. 97, p. 1149–1163, 2018.

[9] W. Sheng, N. Xi, M. Song, and Y. Chen, "CAD-guided robot motion planning," in *Ind. Robot*, vol. 28, no. 2, 2001, pp. 143–152.

[10] J. Liu, X. Huang, S. Fang, H. Chen, and N. Xi, "Industrial robot path planning for polishing applications," in *Proc. Int. Conf. Robot. Biomimetics.* IEEE, 2016, pp. 1764–1769.

[11] M. Rososhansky, F. Xi, and Y. Li, "Coverage based tool path planning for automated polishing using contact stress theory," in *Conf. Autom. Sci. Eng.* IEEE, 2010, pp. 592–597.

[12] P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset, and A. A. Rizzi, "Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage," *IEEE Transactions Autom. Sci. Eng.*, vol. 6, no. 1, pp. 107–120, 2009.

[13] W. Sheng, H. Chen, N. Xi, J. Tan, and Y. Chen, "Optimal tool path planning for compound surfaces in spray forming processes," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 2004, pp. 45–50 Vol.1.

[14] H. Choset, E. Acar, A. Rizzi, and J. Luntz, "Exact cellular decompositions in terms of critical points of Morse functions," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 2000, pp. 2270–2277.

[15] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119 310–119 342, 2021.

[16] H. Choset, "Coverage for robotics – a survey of recent results," in *Annals Math. Artif. Intell.*, vol. 31, no. 1, 2001, pp. 113–126.

[17] H. yuen Tam, O. C. hang Lui, and A. C. Mok, "Robotic polishing of freeform surfaces using scanning paths," *Journal Mater. Process. Technol.*, pp. 191–200, 1999.

[18] S. McGovern and J. Xiao, "UV grid generation on 3d freeform surfaces for constrained robotic coverage path planning," in *IEEE Int. Conf. Autom. Sci. Eng.*, 2022, pp. 1503–1509.

[19] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, "A local/global approach to mesh parameterization," in *Eurographics Symp. Geom. Process.*, vol. 27, no. 5, 2008.

[20] M. Dyck, A. Sachtler, J. Klodmann, and A. Albu-Schäffer, "Impedance control on arbitrary surfaces for ultrasound scanning using discrete differential geometry," *IEEE Robot. Autom. Letters*, vol. 7, no. 3, pp. 7738–7746, 2022.

[21] D. Feng, Y. Sun, and H. Du, "Investigations on the automatic precision polishing of curved surfaces using a five-axis machining centre," *Int. Journal Adv. Manuf. Technol.*, vol. 72, no. 9-12, pp. 1625–1637, 2014.

[22] A. Fernandez, J. Dieste, C. Javierre, and J. Santolaria, "Surface roughness evolution model for finishing using an abrasive tool on a robot," *Int. Journal Adv. Robot. Syst.*, vol. 12, p. 1, 2015.

[23] T. N. Bui and C. Jones, "Finding good approximate vertex and edge partitions is NP-hard," *Inf. Process. Letters*, vol. 42, no. 3, 1992.

[24] J. Hess, G. Tipaldi, and W. Burgard, "Null space optimization for effective coverage of 3d surfaces using redundant manipulators," in *IEEE Int. Conf. Intell. Robots Syst.*, 2012, pp. 1923–1928.

[25] M. Iskandar, O. Eiberger, A. Albu-Schäffer, A. De Luca, and A. Dietrich, "Collision detection, identification, and localization on the DLR SARA robot with sensing redundancy," in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 3111–3117.