Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2023) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Industry 4.0 and Smart Manufacturing

Autonomous robotic polishing of free-form poly-surfaces: planning from scanning in realistic industrial setting

Alessandra Tafuro[a,b,*], Carla Ghalloub[a], Andrea Maria Zanchettin[a], Paolo Rocco[a]

[a] *Politecnico di Milano, Piazza Leonardo da Vinci, Milano 20133, Italy*
[b] *Istituto Italiano di Tecnologia, Via Morego 30, Genova 16163, Italy*

Abstract

We propose a novel solution for generating robot tool paths to achieve uniform surface polishing of free-form poly-surfaces, addressing the limitations of imperfect surface reconstruction in real industrial settings and of abruptly varying curvatures. Existing approaches assume perfect 3D surface knowledge and overlook pieces with abrupt changes in principal curvatures, leading to unsatisfactory outcomes like non-smooth paths and incomplete coverage. Our research aims to develop an algorithm that adapts the polishing path for uniform coverage despite imperfect reconstruction or aggressive varying curvatures of the objects to be polished. Users can control the path points density, overlapping, edge proximity, and scanning direction. By segmenting the surface based on curvature similarity and connecting the separately computed trajectories, we minimize polishing time and achieve optimal polishing quality of free-form poly-surfaces in real industrial settings.

Keywords: Robotic polishing; Smart manufacturing; Path planning; Non-optimal surface reconstruction; Free-form poly-surfaces; Uniform polishing

1. Introduction

Given their versatility, lightweight design, enhanced performance, and aesthetic appeal, free-form surfaces are becoming progressively ubiquitous in some of the most demanding industries and fields including medical, optical, electrical, astronomical, aerospace, and automotive. Despite their ubiquity, their manufacturing process still poses complex and various challenges, hindering automation efforts. This leads to a heavy dependence on human operators, and results in reduced manufacturing efficiency. The introduction of CNC machining has made significant strides in automating drilling, turning, and cutting processes, but failed to make comparative results for operations involving

* Corresponding author. Tel.: +393341417542 .
E-mail address: alessandra.tafuro@polimi.it

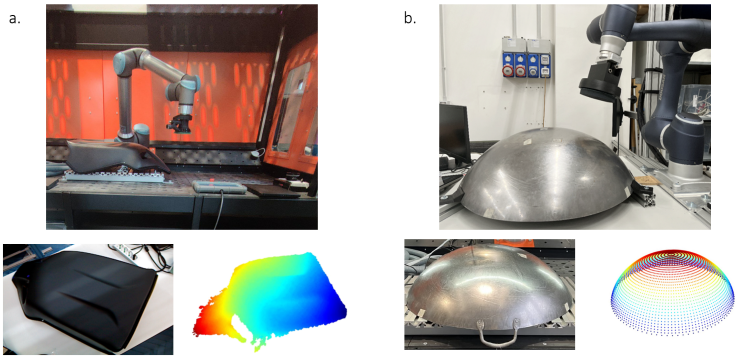


Fig. 1: a. Back of car seat (example of free-form poly-surface); b. Wok (example of free-form mono-surface).

more complex physics such as polishing, grinding, and deburring. Such operations are vital for achieving the desired surface roughness, which leaves no room for compromise on the final product's quality, and makes indispensable the expertise and adaptability of human operators. This is especially significant in the case of polishing, which serves as the final step in machining to ensure the desired surface quality and consistent roughness. Traditionally, human operators have been relied upon for such manufacturing tasks, especially when it comes to polishing freeform surfaces. However, manual polishing is labor-intensive, involving the use of abrasives and potentially hazardous materials for long durations. Moreover, it demands a high level of expertise, making it a costly operation. In fact, polishing activities, such as for molds and dies, can account for approximately 15-20% of overall costs and consume up to 50% of total manufacturing time [1]. Various polishing methods (e.g., laser, magnetorheological, ultrasonic vibration, jet, and bonnet polishing) can be used to improve surface quality [2]. Each method's parameter (feedrate, pressure, intensity, material, time, rotation speed) affect the outcome, requiring customization for each object and material. Manual polishing challenges large-scale production, favoring robotic automation which instead ensures process repeatability, consistent quality, higher yield, cost reduction, and improved efficiency. Along all the models and strategies that should be developed to automate polishing, the robot path planning stage is critical for uniform material removal [2], and this is the motivation behind our intent to develop a methodology to identify the optimal path for consistent surface coverage. The document is structured as follows: Section 2 explores the previous works in this field; Section 3 addresses limitations and problems encountered in existing approaches for polishing free-form poly-surfaces and surface reconstruction; Sections 4 and 5 detail the surface subdivision methodology; Section 6 explains the path generation approach; Finally Section 7 presents experimental results, while Section 8 provides concluding remarks.

2. Related works

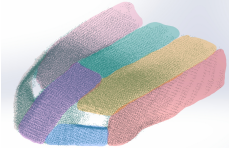


Fig. 2: Human Operator surface subdivision for optimal polishing.

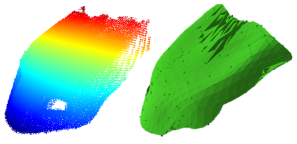


Fig. 3: Pointcloud to NURBS surface fitting.

Planning a robot's path on a curved surface aims to achieve uniform material removal while minimizing overlap and ensuring consistent coverage. Various path generation strategies have been proposed for this purpose [3]. Some applications, like machining, focus on constant path spacing and predicting material removal rate [4]-[5]. Contact models, such as Hertzian and others, are used to minimize overlap and determine spacing between paths [6]-[7]. Direction-parallel paths (zig-zag) [8] and spiral paths [9] are commonly used for surface coverage. Zig-zag paths with minimal overlap can achieve complete coverage on nearly flat surfaces, with spacing based on the tool diameter. Zig-zag patterns are preferred in tool paths due to their ease of generation [10], allowing optimization of path direction and spacing [11]. But if we consider free-form surfaces, using a constant spacing path is often inadequate, as it can lead to incomplete areas with large spacing or excessive overlap, resulting in uneven finishing quality [12]. This is because the tool's contact shape with the surface is influenced by local geometric features and tool deformation caused by applied normal force. As the tool moves, the contact shape and area change due to the surface's varying geometric features and curvature. To ensure optimal contact, path spacing should be determined based on the tool's contact model, which highly depends on the

Metric	Wok	Back of car seat
RMS Point-to-Surface Distance (mm)	0.12	0.45
Hausdorff Distance (mm)	0.32	1.4
Normal Deviation (rad)	0.12	0.52

Table 1: Pointcloud to surface fitting accuracy metrics.

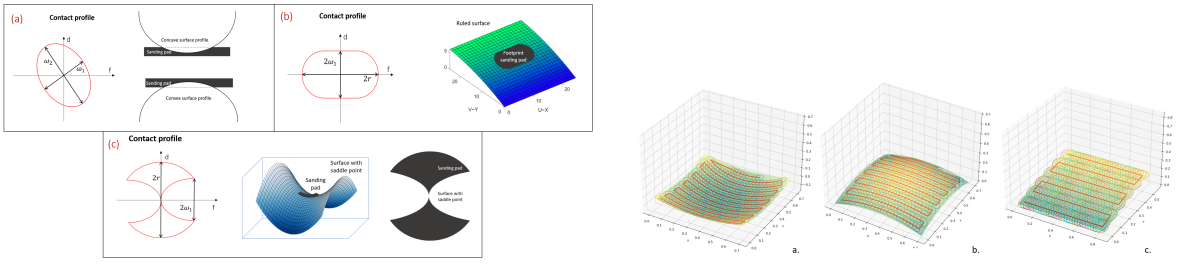
Surface	Covered (%)	Uncovered (%)	Overlap (%)
MATLAB convex surface	105%	2.2%	2.8%
MATLAB concave surface	103%	1%	2%
Wok	108%	4.3%	3.7%
Back of car seat	143%	16.9%	26.1%

Table 2: [12] method performance on different surfaces (synthetic and real).

tool position. In literature, [12] presents a path planning strategy that uses a contact shape descriptor to achieve uniform coverage of curved surfaces with minimal overlap. It utilizes principles from differential geometry to analyze and approximate the contact area between the sanding pad and the surface, determining the optimal path interval distance. They introduced an iterative bisection technique to generate a uniform coverage path on a 2D manifold in R3. The planner also includes normal vector fields along the path for applying contact force perpendicular to the surface during finishing. While achieving satisfactory outcomes on **free-form mono-surfaces**, we faced limitations when applying their methodology to **free-form poly-surfaces**, which consist of adjacent parts with contrasting principal curvature and orientation in space. Indeed the methodology was tested on two surface types (Fig. 1), namely the back of a car seat (a free-form poly-surface) and a wok (a free-form mono-surface). Comparing the surface behavior of the examined objects reveals a clear distinction. Free-form mono-surfaces (Fig. 1 b.) demonstrate gradual variation in principal curvature along their main direction, making them identifiable as a single surface. However, objects like the car seat back (Fig. 1 a.) do not exhibit this characteristic. Instead, the entire surface displays abrupt changes in principal curvature within specific regions. However, the surface is composed of progressive changes in principal curvature within different parts. This indicates that the surface can be divided into distinct parts with gradual principal curvature variation. Indeed a human operator would proceed by dividing the process into sequential polishing of the individual sections, as depicted in Fig. 2. Despite this [12] considers the total object surface as a single one for the computation of the polishing path and additionally they assume complete knowledge of the 3D representation of the surface to be polished. This understanding can be attained through access to the CAD model of the object or accurate real-time surface reconstruction via scanning. However real-life industrial scenarios pose challenges that deviate from this assumption. Indeed imperfect or noisy surface reconstruction is intrinsic to the scanning devices and to the fitting algorithms used. For the aforementioned reasons [12]'s method achieves favourable results on perfectly reconstructed free-form mono-surfaces,however free-form poly-surfaces and the absence of CAD models can complicate the use of this method for path planning. Imperfect or noisy surface reconstructions hinder algorithm performance, leading to irregular paths, overlapping, and incorrect extension or omission of the polishing tool at surface edges. To address sensor-generated noise in industrial surface reconstruction and the optimal machining of free-form poly-surfaces our proposed method proceeds by dividing the process into sequential polishing, clustering sub-parts with similar principal curvature or gradual curvature changes. To ensure high-quality polishing results, multiple paths are generated for sub-surfaces instead of a single path for the entire work-piece, since neglecting abrupt changes in principal direction during path generation can result in defects. The operator's possibility to choose the polishing path feed direction for optimal material removal volume [13] is also added. This work contributes in the following ways: 1) Subdividing a free-form poly-surface based on a smoothness criterion; 2) Generating separate polishing trajectories on individual sub-surfaces using an improved local contact shape descriptor from [12]. The enhancements allow control over path point density, penalty for polishing strips overlapping, and proximity of the polishing pad to surface edges; 3) Adding operator-selectable path feed direction to optimize material removal volume and finishing quality [13]; 4) Connecting individual trajectories into a single global path to minimize total polishing time, ensuring a reasonable and optimal global polishing path for free-form poly-surfaces; 5) Validating the approach through numerical simulations and real-time experiments using a ten-degree-of-freedom robot equipped with a sanding tool. The experiments involved both free-form mono-surfaces and free-form poly-surfaces (Fig. 1).

3. Insights on polishing path generation on free-form poly-surfaces

NURBS (Non-Uniform Rational Basis-Splines) is commonly chosen as the mathematical model to represent free-form surfaces since it offers continuity, accurate calculation of geometric properties, and flexibility in representing complex geometries [14]. NURBS surfaces use a bivariate parametric equation with parameters $\boldsymbol{u}$ and $\boldsymbol{v}$ ranging from 0 to 1 to map the 3D Cartesian coordinates of the surface. Control points and their weights determine the shape of the



(a) (b)

Fig. 4: (a) Contact area of the sanding pad with concave, convex and ruled surfaces and in case of a saddle point. (b) [12] method is successful on synthetic surfaces (convex, concave, ruled) created in simulation (MATLAB).

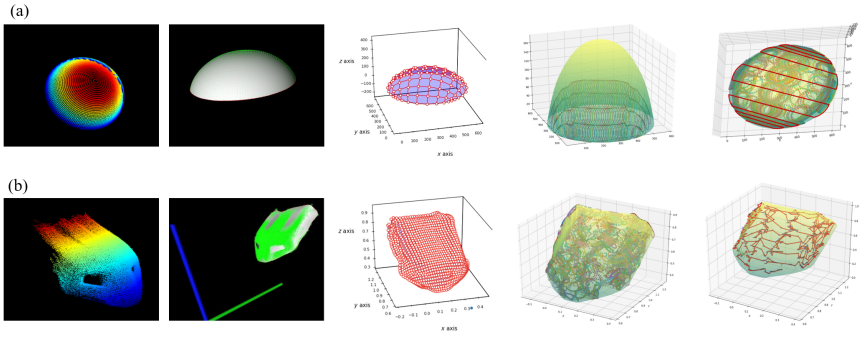


Fig. 5: [12] method for polishing path generation on: a. free-form mono-surface b. free-form poly-surface

base curves that form the surface. In Figure 3, the car seat surface is reconstructed using a B-spline fitting algorithm applied to a point cloud (further described in Section 5), resulting in a smooth parametric representation of the surface. The strategy proposed by [12] to generate the polishing path estimates the contact shape and boundary between the curved surface and the tool while applying a normal force. They classify contact shapes based on a stiffness model derived from Gauss and mean curvatures of the curved surface (see Figure 4a). The contact boundary finally helps in adjusting the tool path spacing for uniform surface coverage. We demonstrated with numerical simulations that the method presented by [12] shows successful performance on simulated synthetic surfaces (e.g., using MATLAB) as depicted in Figure 4b and it is effective for convex, concave, and ruled surfaces, minimizing ribbon overlap and achieving full surface coverage. The method can also be used for components composed of free-form mono-surfaces, like the wok in Figure 5 a., for which the predicted polishing path shows smoothness, minimal overlap, and complete surface coverage. However, when applying the same method to the back of the car seat (Figure 5 b.), accuracy, smoothness, overlap, and coverage are inadequate. Table 2 shows the algorithm difference in performances if applied to MATLAB synthetic surfaces or real free-form mono or poly-surfaces, comparing the coverage percentage, uncovered area and overlapping percentage. Another important aspect to be considered is that complex parts pose challenges for accurate surface reconstruction using NURBS compared to simpler objects. Adjusting B-spline fitting parameters can improve results, but achieving a perfect fit to the actual surface is inherently impossible. Inaccuracies stem from limitations of the camera during point cloud acquisition and the NURBS reconstruction process. Table 1 compares surface reconstruction accuracy for the wok and the back of the car seat using three metrics: Point-to-Surface Distance, Hausdorff Distance, and Normal Deviation. The point cloud data was obtained using an ASUS-Xtion depth camera with 1mm depth accuracy at a distance of 1 meter. Based on these metrics, the back of the car seat exhibits higher inaccuracies compared to the wok, which aligns with expectations. We claim that algorithm [12] low performance on the car seat can be attributed to both the inaccuracies in surface reconstruction and the deviation from the appropriate polishing technique for objects of this nature. Indeed, for optimal polishing results, the human operator's wold proceed by dividing the surface into parts, as depicted in Figure 2, and polishing all of them sequentially with polishing strips aligned with a specific direction, while keeping the feed direction perpendicular to the strips. This technique guarantees maximum material removal volume [13] and optimal quality. In the following, the solution we propose to address these limitations is described. Our developed polishing path generation algorithm takes into account both the optimal way in which free-form poly surfaces should be machined and the inherent limitations in surface reconstruction.

4. Pointcloud clustering

The initial stage involves subdividing the point cloud of the object intended for polishing into clusters by merging points that satisfy the smoothness criterion and are in close proximity to each other. As a result, the output of this algorithm is a collection of clusters, where each cluster represents a group of points deemed to belong to the same smooth surface. An initial step to mitigate the acquisition errors involves using an *Outliers removal function*, denoted as $\epsilon(.)$, which utilizes a statistical filter to compute the distribution of distances between points and their neighbors in the input data-set. Points with mean distances outside the interval defined by the global mean and standard deviation are identified as outliers and removed. Then, utilizing a *Normal finding function* $\Xi(.)$ and a *Curvature finding function* $\Gamma(.)$, the point normals and curvatures are computed for evey point in the pointcloud. Clusters are computed from sorted points based on their curvature values. The algorithm begins with the point of minimum curvature, adding it to a *seeds* set. For each seed point, neighboring points are identified using a *Neighbour finding function* $\Omega(.)$ and evaluated based on the angle between their normals and the current seed point's normal. Points within a specified angle threshold Θ_{th} are added to the current region. The curvature of each neighbor is examined, and if it falls below a certain threshold C_{th}, the point is added to the *seeds* set. The current seed point is then removed from the *seeds* set. When the *seeds* set becomes empty, it indicates completion of the region growth for the current segment, and the process restarts (Eq. 1).

$$\mathcal{K} = \Omega(\tilde{\mathbf{PC}}, C, \mathcal{B}, \Theta_{th}, C_{th}) \quad (1)$$

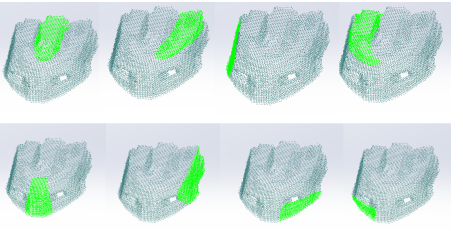


Fig. 6: Car seat pointcloud subdivision in single clusters.

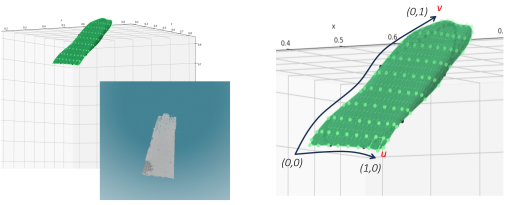


Fig. 7: Pointcloud to NURBS fitting.

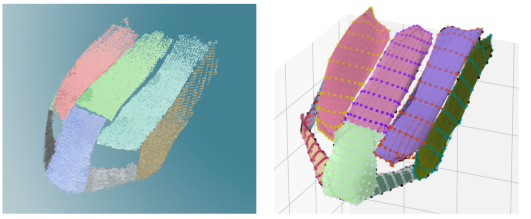


Fig. 8: Free-form poly surface subdivision in N_k single NURBS surfaces.

The result of the process applied to the pointcloud of the back of the car seat is displayed in Fig. 6.

5. NURBS surface reconstruction

The N_k point cloud clusters, denoted as $\mathbf{PC_k}$, are used to fit the points into sub-surfaces $NURBS_k$ using a NURBS fitting function, represented as $\Lambda(.)$. This function utilizes Principal Component Analysis (PCA) to establish an initial B-spline surface, taking into account the rough planar nature of the point cloud with two primary orientations. The B-spline surface is then refined and fitted to improve its representation of the data ($NURBS_k = \Lambda(\mathbf{PC_k})$). Then, to determine the boundary edges of the sub-surfaces, a NURBS boundary evaluation function denoted as $\Delta(.)$ is employed. This function utilizes a circular initialization and fits a B-spline curve to accurately capture the shape and characteristics of the data points ($Bound_k = \Delta(NURBS_k)$). Figure 7 illustrates the fitting outcome for a randomly selected cluster, showcasing the conversion from point clouds to NURBS surface. As prevoiusly mentioned a NURBS surface is composed of a bivariate parametric equation in two parameters u and v, which vary in the interval [0, 1]. Additionally, Figure 8 provides a visualization of the N_k sub-surfaces $NURBS_k$ composing the back on the car seat.

6. Trajectory generation

The core part of the algorithm deals with the polishing trajectory generation on each of the N_k sub-surfaces $NURBS_k$ separately. By leveraging the contact shape information [12], the distance between consecutive paths of the zig-zag pattern is modified to ensure the desired coverage while having a controllable level of overlapping. In the process, several parameters play a crucial role: an *overlap factor* O and *four edge proximity factors* ($F_{prox,0}, F_{prox,1}, F_{prox,2}, F_{prox,3}$).

These parameters determine the extent of overlap between consecutive ribbons and how close the polishing tool should approach the four edges. The reason for considering a certain amount of overlap is to ensure the complete surface coverage, as relying solely on perfectly tangent polishing strips (with zero overlap) may not achieve this goal.

6.1. Path feed direction selection

In some cases, full surface coverage may not be necessary, especially when different polishing techniques are used for the edges with respect to the rest of the surface, so it's important to set the allowed distance of the polishing tool from the edges. The path planning strategy involves creating sequential path segments along a chosen edge of the workpiece surface. The scanning direction parameter D (set as "H" for horizontal or "V" for vertical) determines the selected edge (Fig. 9).

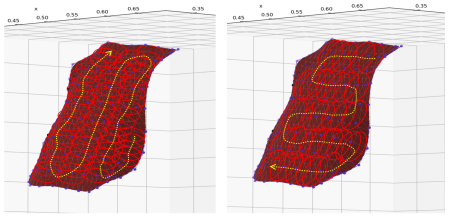


Fig. 9: The two possible path scanning directions.

6.2. Initialization and path generation procedure

A zig-zag pattern with an offset is used, and each subsequent path segment is determined by shifting the boundary of the previous segment. The shifting distance is determined by the effective contact shape and fine-tuned using the overlap factor O. Other parameters required for determining the contact shape between the tool and the workpiece include the normal polishing force $F[N]$, tool radius $r[m]$, and surface-tool compliance factor $k_c[N/m]$. The output of this process is a series of tool positions ($\mathbf{Pos_k}$) and orientations ($\mathbf{Ori_k}$) to polish all the sub-surfaces while maintaining perpendicularity to the surface. These vectors are then unified in a single global polishing path. The path generation algorithm starts with the operator choosing the feed direction, considering for maximum material removal and optimal ribbon orientation. In the following D = "H" is considered, but the same procedure applies with D = "V", just switching the u and v coordinates. The path generation process begins by using the $Bound_k$ (an iso parametric curve on the NURBS surface for $v = 0$) and the parameter $F_{prox,0}$. An initial $iso - v$ spline S_0 is created, which maintains a distance from $Bound_k$ based on the value of $F_{prox,0}$. In Fig. 10, the orange line represents $Bound_k$, the red line represents S_0, while $F_{prox,0}$ determines the distance between these two lines. From S_0, a set of N_i base points $pu_{i,0}$ is sampled. The number of sampled points is determined by the *sampling factor* S, which controls the density of polishing path points for each polishing ribbon. In Fig. 10 the base points are shown in white. On the other hand, $F_{prox,1}$ and $F_{prox,2}$ define the limits on the right and left sides of S_0, respectively, where sampling points are not allowed. This influences the proximity of the polishing path to the lateral edges of the sub-surface. This set of points is localized by the a pair of $\{(u^{up}_{i,0}, v^{up}_{i,0})\}_{1=1..N_i}$ coordinates At each point of the trajectory, a local reference system is defined, denoted as $d - n - f$. Here, n represents the surface normal (used to set the orientation of the polishing tool $\mathbf{N_{i,j}}$), f represents the advancement direction (partial derivative of the surface with respect to v), and d represents the lateral step direction (partial derivative of the surface with respect to u). The goal of the trajectory generation algorithm is to find the optimal offset between successive polishing tapes. For each point $pu_{i,0}$ on the initial curve, the algorithm determines the offset in the advancement direction, resulting in a point $\mathbf{P_{i,1}}$. To implement this method, a search interval is defined, denoted as $[a = 0, b = 3r]$, where r is the radius of the polishing tool. Within this interval, a bisection method is used to determine a candidate point $\tilde{\mathbf{P}}_{i,1}$. At this point, the contact shape between the tool and the surface is computed based on the mean and Gaussian curvature values, utilizing the contact model developed by [12] and described in Section 6.3. For each contact profile, the points of minimum ($pl_{i,1}$) and maximum ($pu_{i,1}$) advancement are considered to calculate the upper and lower boundary curves of the polishing strip, represented by the minimum and maximum

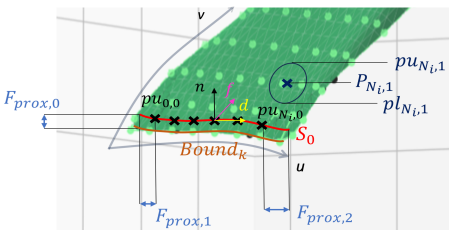


Fig. 10: Initialization of the path generation procedure.

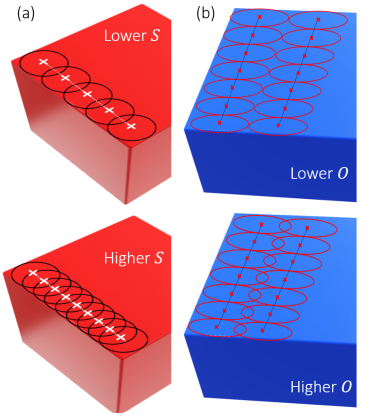


Fig. 11: (a) Polishing path points density; (b) Overlapping regulation

coordinates along the v axis. After defining the search interval, the next step is to iterate to find the acceptable offset, given the amount of acceptable overlap defined by the parameter O. At each iteration step, the average between the upper and lower limits of the search interval $mid = (a+b)/2$ is considered as a candidate offset for the new $\tilde{\mathbf{P}}_{i,1}$. If the error, defined as the distance between the upper boundary curve of the previous tape $pu_{i,j-1}$ and the lower boundary curve of the next tape $pl_{i,j}$, is less than a certain tolerance, defined by the overlap parameter O, the offset is valid (Eq. 2 and Fig. 11 b.).

$$e_{i,j} \;<=\; (pl_{i,j} - pu_{i,j-1}) \bullet \vec{f}\left\|pl_{i,j} - pu_{i,j-1}\right\| \;<=\; O \tag{2}$$

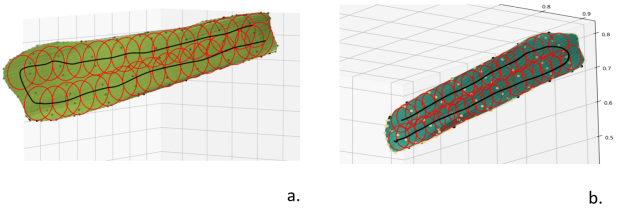
If the error is greater than the tolerance or negative, the search interval is appropriately adjusted (if the error norm is negative $a_{new} = mid, b_{new} = b$, if the error norm is positive and $e_{i,j} > O$, $a_{new} = a, b_{new} = mid$). Once the end of the current polishing strip is reached ($P_{N_i,j}$ defined) the upper boundary points are fitted in a Spline $\{pu_{i,j}\}_{i=1..N_i,j}$ and then, based on the *sampling factor S* a set of points to be considered to determine the next polishing strip offset is defined (Fig. 11 a.). This procedure is repeated for all the polishing strips increasing the j index until the opposite surface boundary is reached. In particular the condition of boundary reaching is defined in Eq. 3.

$$v^{up}_{0,N_j} == F_{prox,3} \quad if\ D == \text{“}H\text{”} \qquad\qquad u^{up}_{0,N_j} == F_{prox,3} \quad if\ D == \text{“}V\text{”} \tag{3}$$

$F_{prox,3}$ determines the maximum coordinate of the upper boundary curve in the direction of advancement, indicating the distance that should be maintained from the opposite edge of the surface. The algorithm concludes by providing the Cartesian coordinates of the polishing path relative to the base frame of the robot: $\mathbf{Pos_k} = \{\mathbf{P_{i,j}}\}_{\mathbf{i=1...N_i,j=1...N_j}} = \{X_{i,j}, Y_{i,j}, Z_{i,j}\}^k_{i=1...N_i,j=1...N_j}$, as well as the orientation vectors along the polishing path: $\mathbf{Ori}_k = \{\mathbf{N_{i,j}}\}_{\mathbf{i=1...N_i,j=1...N_j}} = \{Q^1_{i,j}, Q^2_{i,j}, Q^3_{i,j}\}^k_{i=1...N_i,j=1...N_j}$, for all N_k sub-surfaces (Fig. 12). After calculating individual polishing paths for each sub-surface, the paths are merged into a global trajectory that minimizes polishing time. The initial and final points of each sub-surface path $E^1_k = (X_{0,0}, Y_{0,0}, Z_{0,0})$, $E^2_k = (X_{N_i,N_j}, Y_{N_i,N_j}, Z_{N_i,N_j})$ are extracted. The sequence of polishing is determined by finding the shortest path based on Cartesian distance between the closest path extremities. The transition from E^1_k to E^2_k follows a gate-type path. Initially, the end effector is raised from $(X_{0,0}, Y_{0,0}, Z_{0,0})$ using a user-defined parameter H. Then, it moves linearly at the same height until it reaches and descends to $(X_{N_i,N_j}, Y_{N_i,N_j}, Z_{N_i,N_j})$. The overall trajectory coputed for the back of the car seat is shown in Figure 13. The whole algorithm is provided in Appendix A together with the **Nomenclature**.

6.3. Contact shape model

The local contact shape descriptor employed in this study [12] utilizes concepts from differential geometry to estimate the shape of contact between the finishing tool and the workpiece surface when a normal force F is applied. By using a contact stiffness model, the depth of penetration corresponding to a given contact force can be estimated based on the Gauss curvature $kg = k_2k_2$ and mean curvatures $km = (k_2 + k_2)/2$ of the curved surface. Here, k_1 and k_2 represent the principal curvatures of the surface at a specific point. According to [12], the penetration depth of the sanding pad h follows a linear contact model: $k_1u^2 + k_2v^2 = 2h$. Given the known normal force F applied by the sanding pad and a compliance factor k_c determined through experiments [15], the deformation of the sanding pad onto the surface can be determined as $h = F/k_c$. Based on the signs of the principal curvatures, the contact boundary is described with a:



a. b.

Fig. 12: Sub-surfaces polishing paths samples ($k = 4, k = 5$)

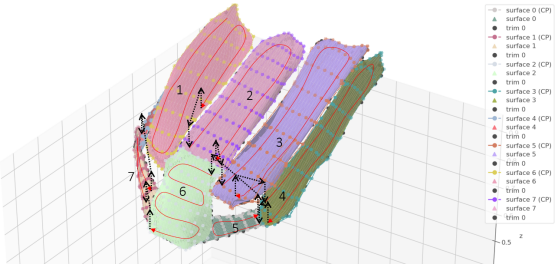


Fig. 13: Global polishing path assembly.

1. **Ellipse** (Fig. 4a) if $k_g > 0$ and $k_m > 0$ (convex region) or $k_g > 0$ and $k_m < 0$ (concave region);
2. **Parabola** (Fig. 4a) if $k_g = 0$ (ruled surfaces have one null principal curvature);
3. **Hyperbole** (Fig. 4a) if $k_g < 0$ (a surface with saddle points has the principal curvatures of opposite sign).

The magnitude of the axes defining these three profiles is $\omega_i = \sqrt{(1/k_i)^2 - (|1/k_i| - h)^2}$ if $\omega_i <= r$ and $\omega_i = r$ otherwise (look at Fig.4a for reference).

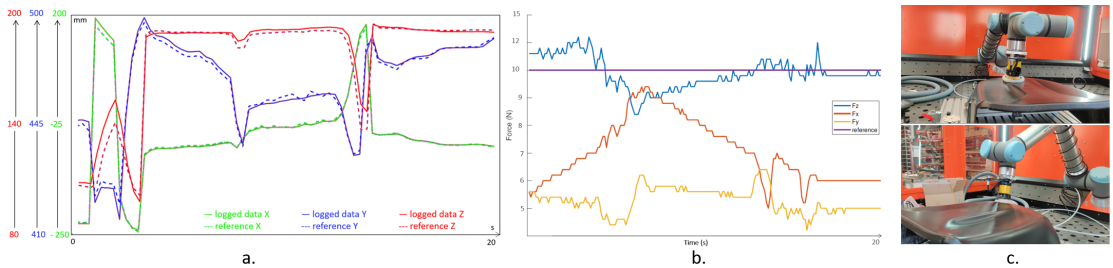


Fig. 14: a-b) TCP position and force measurements. c) Experimental setup: UR10e following the polishing planned trajectory.

7. Experimental results

To validate the uniform coverage path planner, real-time experiments were conducted using a UR10e robot (Fig. 14 c.) equipped with a Mirka AIROS orbital sander (650CV input power, 60mm radius). The robot utilized the UR10e's embedded force control algorithm to regulate the sander's normal force on the surface, set at 10N. The robot followed the generated path, ensuring the tool axis remained perpendicular to the surface at each point. It achieved this through a series of *movel* movements while the force control operated simultaneously. An ASUS-Xtion depth camera mounted on the robot's end effector scanned the back of the car seat, generating a point cloud with a 1mm depth accuracy at a 1-meter distance. Camera calibration establishes the base reference system for the robot, enabling the acquisition of a point cloud with coordinates relative to it. This ensures that the polishing path positions are already expressed in the same reference frame. For the pointcloud clustering part, curvature threshold value of $C_{th} = 1mm^{-1}$ and angle threshold value of $\Theta_{th} = 0.1rad$ have been used. The surface-tool compliance factor k_c has been set to $58 \cdot 10^6 N/m$. For all the sub-surfaces these are the values used: overlap factor $O = 0.05$, sampling factor $S = 0.1$, edge proximity factors $F_{prox,0} = 0.1, F_{prox,1} = 0.1, F_{prox,2} = 0.95, F_{prox,3} = 0.95$. Both scanning directions were tested. Fig. 14 a. compares the reference and actual positions of the tool during polishing of surface 2 in Fig. 13. The tool closely followed the trajectory, with minor variations mainly in the z-axis due to tool vibration. Fig. 14 b. displays the forces (F_x, F_y, F_z) relative to the local reference system at the TCP (Tool Center Point). The desired force was set to 10 N along the z-axis. Non-zero forces along the x and y directions result from friction between the tool and the workpiece during polishing. The z-axis force shows variations ranging from 12 N to 8 N, with an average of 10.46 N. The graph indicates some force jumps along the z-direction due to tool vibration during surface polishing, as the tool did not glide smoothly on the object. This uncertainty is partly attributed to the use of a load cell instead of a force sensor in the robot, which may provide less accurate results. Furthermore, the specific force control algorithm implemented in the robot controllers (Universal Robots) adds to the uncertainty. This limitation relates to the robot chosen for the experiments, and not to the developed polishing path planner. Nonetheless, the overall trajectory was smoothly followed, resulting in a satisfactory polishing path. The key determinant for deeming the polishing path satisfying was its ability to faithfully adhere to the operator's desires and requirements in terms of edge proximity, overlapping of polishing ribbons and coverage of the surface. Moreover, the possibility to select the scanning direction is revealed to be crucial for such objects, as it replicates the approach of a human operator performing the task.

8. Conclusions and future work

Our innovative contribution provides an autonomous path generation algorithm that achieves uniform surface polishing of free-form poly-surfaces, even with sub-optimal surface reconstruction, common in real industrial settings. It ensures uniform coverage, accommodates varying curvatures, and offers user control. Segmentation by curvature and trajectory optimization minimize polishing time. Users can control stability, overlapping, edge proximity, and scanning direction. Real-time experiments validate our method. An interesting future addition would be a material removal model and a force controller for polishing free-form surfaces, considering factors like cutting process, material removal, feedrate, and spindle speed. Future research could also explore additional path patterns for circular based objects such as the Archimedes's spiral. In the envisioned future of path generation tasks, humans will transition from direct operators to supervisors. Automation will handle routine operations while humans oversee, set objectives, and solve complex problems. Effective communication and adaptability will be pivotal, requiring humans to convey intentions and robots to provide feedback. This collaboration will enable the best of both human expertise and robotic precision to be leveraged for optimal results, ensuring efficiency, safety and adaptability in a wide range of scenarios.

Appendix A.

Algorithm 1 Free-form poly-surfaces path generation algorithm.

Input: Pointcloud representation **PC**, Curvature threshold $C_{th}[rad]$, Angle threshold $\Theta_{th}[rad]$, Normal polishing Force $F[N]$ Tool radius $r[m]$, Surface-tool compliance factor $k_c[N/m]$ Overlap factor O, Sampling factor S, Scanning direction D, 4 Edge proximity factors $F_{prox,0}, F_{prox,1}, F_{prox,2}, F_{prox,3}$.
Outputs: Polishing path in terms of: position $\{\mathbf{X_t}, \mathbf{Y_t}, \mathbf{Z_t}\}$ and orientation $\{\mathbf{Q_t^1}, \mathbf{Q_t^2}, \mathbf{Q_t^3}\}$ for t = {1...N_t}.

Pointcloud Clustering:

−Neighbour finding function $\Omega(.)$ −Normal finding function $\Xi(.)$ −Remove outliers function $\epsilon(.)$ −Curvature finding function $\Gamma(.)$
Outliers removal: $\tilde{\mathbf{PC}} = \epsilon(\mathbf{PC})$
Normal computation: $\mathcal{B} = \Xi(\tilde{\mathbf{PC}})$
Curvature computation: $\mathcal{C} = \Gamma(\tilde{\mathbf{PC}})$
Clusters computation: $\mathcal{K} = \Omega(\tilde{\mathbf{PC}}, \mathcal{C}, \mathcal{B}, \Theta_{th}, C_{th})$

NURBS reconstruction:

−NURBS fitting function $\Lambda(.)$ −NURBS boundary evaluation function $\Delta(.)$

for $\mathbf{PC_k} \in \mathcal{K}$ **do**
NURBS fitting: $NURBS_k = \Lambda(\mathbf{PC_k})$
Surface boundary computation: $Bound_k = \Delta(NURBS_k)$
end for

Trajectory Generation:

for all $NURBS_k$: **do**
if D == "H": **then**
$Limit\_reached\_condition : v^{up}_{0,j} <= F_{prox,3}$
end if
if D == "V": **then**
$Limit\_reached\_condition : u^{up}_{0,j} <= F_{prox,3}$
end if
Initialization: $j = 0, i = 1...N_i$
$S_0 = Initial\_spline\ (F_{prox,0}, Bound_k)$
Sampling Upper boundary points: $pu_{i,0} = Sampling\ (S_0, F_{prox,1}, F_{prox,2}, S) = (u^{up}_{i,0}, v^{up}_{i,0})$
while $Limit\_reached\_condition$ **do**
for $i = 0 : N_i$: **do**
Bisection method to find the polishing path: $\mathbf{P_{i,j}} = Bisection\ (S_j, O, r)$
Contact area computation and upper boundary extraction: $A_{i,j} = Contact\_area\ (F, r, k_c, P_{i,j}, NURBS_k)$ $pu_{i,j} = Upper\_boundary\ (A_{i,j})$
Surface normal vector extraction along the path: $\mathbf{N_{i,j}} = Normal\_computation\ (P_{i,j}, NURBS_k)$
end for
$S_j = Spline\_interpolation\ (pu_{i,j})$
Sampling Upper boundary points: $pu_{i,j} = Sampling(S_j, F_{prox,1}, F_{prox,2}, S) = (u^{up}_{i,j}, v^{up}_{i,j})$
Next polishing ribbon: $j+ = 1$
end while
Polishing path of the k^{th} sub-surface : $\mathbf{Pos_k} = \{\mathbf{P_{i,j}}\}_{\mathbf{i=1...N_i\, j=1...N_j}} = \{X_{i,j}, Y_{i,j}, Z_{i,j}\}^k_{i=1...N_i, j=1...N_j}$ $\mathbf{Ori}_k = \{\mathbf{N_{i,j}}\}_{\mathbf{i=1...N_i\, j=1...N_j}} = \{Q^1_{i,j}, Q^2_{i,j}, Q^3_{i,j}\}^k_{i=1...N_i, j=1...N_j}$
end for

Global trajectory assembly :

for $k\ in\ 0 : N_k$: **do**
Extraction of initial and final points of the N_k polishing paths: $E^1_k, E^2_k = Estremities(\mathbf{Pos_k})$
end for
Shortest global polishing path computation: $Pos = Shortest\_path(\{E^1_k, E^2_k\}_{k=1...N_k})$

return Global polishing Trajectory: $\mathbf{Pos} = \{\mathbf{X_t}, \mathbf{Y_t}, \mathbf{Z_t}\}_{\mathbf{t=1...N_t}}$, $\mathbf{Ori} = \{\mathbf{Q_t^1}, \mathbf{Q_t^2}, \mathbf{Q_t^3}\}_{\mathbf{t=1...N_t}}$

Nomenclature

PC	Pointcloud of the object to be polished
$\tilde{\mathbf{PC}}$	Outliers filtered pointcloud
$\mathbf{PC_k}$	k^{th} pointcloud cluster
C_{th}	Curvature treshold
Θ_{th}	Angle treshold
F	Normal polishing force
r	Polishing tool radius
k_c	Surface-tool compliance factor
O	Overlap factor
S	Sampling factor
D	Scanning direction
$F_{prox,0}$	Starting edge proximity factor
$F_{prox,1}$	Left edge proximity factor

$F_{prox,2}$	Right edge proximity factor
$F_{prox,3}$	Terminal edge proximity factor
$\Omega(.)$	Neighbour finding function
$\epsilon(.)$	Pointcloud outliers removal function
$\Xi(.)$	Pointcloud normal finding function
$\Gamma(.)$	Pointcloud curvature finding function
$\Lambda(.)$	pointcloud-NURBS fitting function
$\Delta(.)$	NURBS boundary evaluation function
$\mathbf{P_{i,j}}$	Polishing path points $(X_{i,j}, Y_{i,j}, Z_{i,j})$ over the bi-parametrized NURBS surface
$\mathbf{N_{i,j}}$	Polishing tool orientation vectors $(Q^1_{i,j}, Q^2_{i,j}, Q^3_{i,j})$ normal to the bi-parametrized NURBS surface
$pu_{i,j}$	Upper boundary points $(X^u_{i,j}, Y^u_{i,j}, Z^u_{i,j})$ of the polishing ribbon
$A_{i,j}$	Tool-surface contact area in $P_{i,j}$
$NURBS_k$	k^{th} fitted NURBS surface
S_j	j^{th} Polishing ribbon upper boundaty curve
Ori	Sequence of polishing tool orientation vectors along the global polishing path
Pos	Sequence of polishing tool position vectors along the global polishing path
$\mathbf{Ori_k}$	Sequence of polishing tool orientation vectors along the polishing path on the k^{th} sub-surface
$\mathbf{Pos_k}$	Sequence of polishing tool position vectors along the polishing path on the k^{th} sub-surface
N_j	Number of polishing ribbons
N_i	Number of evaluation points on the j^{th} polishing ribbon.
N_k	Number of pointclouds clusters
E^1_k, E^2_k	Initial and final point of the k^{th} polishing path
$u_{i,j}, v_{i,j}$	$P_{i,j}$ coordinates along $\bar{\mathbf{j}}$ and $\bar{\mathbf{i}}$ on the bi-parametrized NURBS surface
$X_{i,j}, Y_{i,j}, Z_{i,j}$	$P_{i,j}$ coordinates wrt the robot base frame
X_t, Y_t, Z_t	Sequence of polishing tool position vectors along the global polishing path wrt the robot base frame
$Q^1_{i,j}, Q^2_{i,j}, Q^3_{i,j}$	Orientation vectors as Euler angles in $P_{i,j}$
Q^1_t, Q^2_t, Q^3_t	Sequence of polishing tool orientation vectors along the global polishing path as Euler angles
$\mathcal{K}$	Set of N_k sub-surfaces described as NURBS
$\mathcal{B}$	Set of normal vectors of the pointcloud **PC**
C	Set of curvature values of the pointcloud **PC**

References

[1] S. Husmann et al. “Model Predictive Force Control in Grinding based on a Lightweight Robot”. In: IFAC-PapersOnLine 52.13 (2019). 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019, pp. 1779–1784. issn: 2405-8963. doi: https : / / doi . org / 10 . 1016 / j. ifacol.2019.11.459. url: https://www.sciencedirect.com/science/ article/pii/S2405896319314405.

[2] Yanjun Han et al. “Tool paths generation strategy for polishing of freeform surface with physically uniform coverage”. In: The International Journal of Advanced Manufacturing Technology 95.5-8 (Nov. 2017), pp. 2125–2144. doi: 10.1007/s00170- 017- 1281- 2. url: https://doi.org/10.1007/ s00170-017-1281-2.

[3] A. Lasemi, D. Xue, and P. Gu, “Recent development in cnc machining of freeform surfaces: a state-of-the-art review,” Computer-Aided Design, vol. 42, no. 7, pp. 641–654, 2010.

[4] P. N. Atkar, H. Choset, and A. A. Rizzi, “Towards optimal coverage of 2-dimensional surfaces embedded in ir/sup 3: choice of start curve,” in Proceedings 2003 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2003), vol. 4, 2003, pp. 3581–3587.

[5] T. Kim and S. E. Sarma, “Optimal sweeping paths on a 2-manifold: a new class of optimization problems defined by path structures,” IEEE

[6] M. Rososhansky and F. J. Xi, “Coverage based tool-path planning for automated polishing using contact mechanics theory,” Journal of Manufacturing Systems, vol. 30, no. 3, pp. 144–153, 2011.

[7] Y. Lv, Z. Peng, C. Qu, and D. Zhu, “An adaptive trajectory planning algorithm for robotic belt grinding of blade leading and trailing edges based on material removal profile model,” Robotics and Computer- Integrated Manufacturing, vol. 66, p. 101987, 2020.

[8] W. H. Huang, “Optimal line-sweep-based decompositions for coverage algorithms,” in Proceedings 2001 ICRA. IEEE Intl. Conf. on Robotics and Automation, vol. 1, 2001, pp. 27–32.

[9] Y. Han, L. Zhang, M. Guo, C. Fan, and F. Liang, “Tool paths generation strategy for polishing of freeform surface with physically uniform coverage,” The Intl. Journal of Advanced Manufacturing Technology, vol. 95, no. 5, pp. 2125–2144, 2018.

[10] S. E. Sarma, “The crossing function and its application to zig-zag tool paths,” Computer-Aided Design, vol. 31, no. 14, pp. 881–890, 1999.

[11] M. Xiao, Y. Ding, and G. Yang, “A model-based trajectory planning method for robotic polishing of complex surfaces,” IEEE trans. on Automation Science and Engineering, 2021.

[12] Wen, Yalun and Jaeger, Daniel J. and Pagilla, Prabhakar R. (2022) ”Uniform Coverage Tool Path Generation for Robotic Surface Finishing of Curved Surfaces” *IEEE Robotics and Automation Letters* **7** (2): 4931–4938.

[13] Z.-Y. Liao, J.-R. Li, H.-L. Xie, Q.-H. Wang, and X.-F. Zhou, “Regionbased toolpath generation for robotic milling of freeform surfaces with stiffness optimization,” Robotics and Computer-Integrated Manufacturing, vol. 64, p. 101953, 2020.

[14] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi, “Uniform coverage of automotive surface patches,” The Intl. Journal of Robotics Research, vol. 24, no. 11, pp. 883–898, 2005.

[15] Y. Wen, J. Hu, and P. R. Pagilla, “A novel robotic system for finishing of freeform surfaces,” in 2019 Intl. Conf. on Robotics and Automation (ICRA), 2019, pp. 5571–5577.